UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO
INGENIERO EN MECATRÓNICA

TEMA:

SISTEMA MODULAR PARA REALIZAR PRÁCTICAS DE SERVOMECANISMOS Y CONTROL MEDIANTE DESLIZADOR LINEAL E INTERFAZ CON LABVIEW

AUTOR: Ernesto Vladimir Palacios Merino

DIRECTOR: Ing. Diego Terán

Ibarra - Ecuador
2014

UNIVERSIDAD TÉCNICA DEL NORTE
AUTORIZACIÓN DE USO Y PUBLICACION A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.

1. IDENTFICACIÓN DE LA OBRA

La universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determino la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CEDULA DE IDENTIDAD	110405921-5		
APELLIDOS Y NOMBRES	Palacios Merino Ernesto Vladimir		
DIRECCIÓN:	Andrés Bello 12-04 y Miguel Ángel Suarez (Loja -Ecuador)		
EMAIL:	mecatronica.mid@gmail.com		
TELÉFONO FIJO:	07-2573-855	TELÉFONO MÓVIL:	09-91519642

DATOS DE LA OBRA	
TÍTULO:	**Sistema Modular para realizar prácticas de servomecanismos y control mediante deslizador lineal e interfaz con LabVIEW**
AUTOR:	**PALACIOS MERINO ERNESTO VLADIMIR**
FECHA:	**2014-02-06**
PROGRAMA:	**PREGRADO**
TÍTULO POR EL QUE OPTA:	**INGENIERO EN MECATRÓNICA**
DIRECTOR	**ING. DIEGO TERÁN**

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Ernesto Vladimir Palacios Merino con cédula de identidad Nro. 110405921-5, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ernesto V. Palacios Merino
110405921-5

UNIVERSIDAD TÉCNICA DEL NORTE
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Ernesto Vladimir Palacios Merino, con Cédula de identidad Nro. 110405921-5, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículos 4, 5,6, en calidad de autor del trabajo de grado denominado: SISTEMA MODULAR PARA REALIZAR PRÁCTICAS DE SERVOMECANISMOS Y CONTROL MEDIANTE DESLIZADOR LINEAL E INTERFAZ CON LABVIEW, que ha sido desarrollada para optar por el título de: Ingeniero en Mecatrónica en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

Ibarra, a los 15 días del mes de julio del 2014

Ernesto V. Palacios Merino
110405921-5

3. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrollo sin violar derechos de autor de terceros, por lo tanto la obra es original, y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 15 días del mes de julio del 2014

Ernesto V. Palacios Merino
110405921-5

CERTIFICACIÓN

Certifico que la Tesis previa a la obtención del título de Ingeniero en Mecatrónica con el tema: **SISTEMA MODULAR PARA REALIZAR PRÁCTICAS DE SERVOMECANISMOS Y CONTROL MEDIANTE DESLIZADOR LINEAL E INTERFAZ CON LABVIEW**, ha sido desarrollada y terminada en su totalidad por el Sr. Ernesto Vladimir Palacios Merino, con cédula de identidad: 110405921-5, bajo mi supervisión para lo cual firmo en constancia.

Atentamente,

Ing. Diego Terán

DIRECTOR DE PROYECTO

AGRADECIMIENTO

Tras la culminación de este trabajo, tengo que agradecer mucho a mis padres por su apoyo, a mi hermano por su paciencia, a todos mis familiares por siempre tenerme presente y por su aliento.

Quiero agradecer también a todos mis amigos quienes de una u otra forma han sabido apoyarme a lo largo de este proceso de aprendizaje, gracias por su compañía y compañerismo.

Un especial agradecimiento a mi director de carrera Ing. Diego Terán, por su ayuda y guía en la realización de este trabajo.

DEDICATORIA

La vida es un camino de conocimiento y sabiduría; el ser humano jamás deja de aprender, no importa la edad que tenga, lo importante es tener la mente y el corazón abiertos a los cambios que exige la nueva sociedad.

Por tal razón dedico este trabajo a mis padres, quienes supieron orientarme para llevar una formación permanente. A mi hermano, a mi tía, Hna. Rosarito y a su comunidad Carmelitana, quienes me apoyaron moral y espiritualmente.

Ernesto Vladimir

INDICE GENERAL

1. IDENTFICACIÓN DE LA OBRA ii
2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD**Error! Bookmark not defined.**
3. CONSTANCIAS ..... **Error! Bookmark not defined.**
CERTIFICACIÓN ..... **Error! Bookmark not defined.**
AGRADECIMIENTO vii
DEDICATORIA viii
INDICE GENERAL ix
INDICE DE FIGURAS xi
INDICE DE TABLAS xiii
INDICE DE ECUACIONES xiii
RESUMEN xv
ABSTRACT xvi
PRESENTACIÓN xvii
CAPITULO I 18
ANTECEDENTES 18
1.1. EL MICROCONTROLADOR ARM MBED 18
1.1.1. INTERFAZ MBED CON RPC 21
1.2. EL SERVOMOTOR 22
1.2.1 EL SERVOMOTOR 110SJT 25
1.3. PROTOCOLO DE COMUNICACIÓN ETHERNET 26
1.3.1. PROTOCOLO RPC 34
1.4. PROGRAMACIÓN GRÁFICA EN LABVIEW 34
CAPITULO II 36
ARQUITECTURA DEL SISTEMA 36
2.1. DESCRIPCIÓN GENERAL 36
2.2. CONCEPCIÓN A BLOQUES 37
2.3. DETERMINACIÓN DE SUBSISTEMAS 39
2.3.1. ESTRUCTURA DEL DESLIZADOR 39
2.3.2. CONTROLADOR MICROPROCESADO 39
2.3.3. API DEL CONTROLADOR EN LABVIEW 41
2.3.4. RESPUESTA 42
2.3.4.1. Control de Posición 43
2.3.4.2. Control de Velocidad 43
CAPITULO III 46
DISEÑO DEL SISTEMA 46
3.1. ESTRUCTURA DEL DESLIZADOR 46
3.1.1. ALCANCE 46
3.1.2. SISTEMA MECÁNICO 46
3.1.2.1. Transmisión de potencia 47
3.1.2.2. Rodamientos lineales 51
3.1.3. DESLIZADOR 57
3.1.3.1. Rodamiento Lineal 58
3.1.3.2. Tornillo de bolas 60

3.2. CONTROLADOR MICROPROCESADO 64
3.2.1. DIAGRAMA DE CONTROL 64
3.2.2. ESQUEMA DEL CIRCUITO 66
3.2.2.1. Fuentes de alimentación 67
3.2.2.2. Encoder de cuadratura 68
3.2.2.3. Salida analógica 69
3.2.3. PROGRAMACIÓN DEL MICROCONTROLADOR 72
3.3. API PARA EL CONTROL Y MONITOREO EN LABVIEW 76
3.3.1. API DE BAJO NIVEL 76
3.3.1.1. Protocolo serial 77
3.3.1.2. Protocolo Ethernet 80
3.3.2. LIBRERÍA EN LABVIEW 81
3.3.2.1. API Serial - Configuración 81
3.3.2.2. API Serial – Encender Servomotor 83
3.3.2.3. API Serial – Manejo de errores 86
3.3.2.4. API Serial - Tren de pulsos 90
3.3.2.5. API Serial - Dirección 94
3.3.2.6. API Serial – Velocidad de Pulsos 97
3.3.2.7. API Serial – Definir Pulsos 100
3.3.2.8. API Serial – Leer Encoder 103
3.3.2.9. API Serial – Limpiar Encoder 107
3.3.2.10. API Serial – Leer RPM. 110
3.3.2.11. API Serial – Ir a Inicio 113
3.3.2.12. API Serial – Voltaje de salida 116
3.3.2.13. API Ethernet – Configuración 120
3.3.2.14. API Ethernet – Manejo de errores 121
CAPITULO IV 124
IMPLEMENTACIÓN Y PRUEBAS 124
4.1. IMPLEMENTACIÓN FÍSICA DEL MÓDULO 125
4.1.1. ESTRUCTURA DE SUJECIÓN 125
4.1.2. Mesa del deslizador 126
4.1.3. El servomotor 127
4.2. CALIBRACIÓN DEL MÓDULO 127
4.2.1. Calibración Mecánica 127
4.2.2. Calibración eléctrica 128
4.3. PRUEBAS DE FUNCIONAMIENTO 129
4.3.1. Pruebas de posición 129
4.3.2. Pruebas de velocidad 131
CAPITULO V 134
CONCLUSIONES Y RECOMENDACIONES 134
5.1. CONCLUSIONES 134
5.2. RECOMENDACIONES 135
BIBLIOGRAFÍA 138
ANEXO 1. MANUAL DE OPERACIONES 140
1. Generalidades 142
2. Armado y desarmado del módulo 142
2.1. Estructura del deslizador 143

2.2. Tornillo central y base 147
2.3. Guías y rodamientos 148
2.4. Encoder 149
2.5. Servomotor 150
3. Electrónica 151
3.1. Placa de Conexión Servodrive 152
3.2. Placa de encoder y alarmas 154
3.3. Placa microcontrolador 155
4. Mantenimiento 157
4.1. Aplicación de grasa 158
ANEXO 2. MANUAL DE PRÁCTICAS 160
MANUAL DE PRÁCTICAS 162
PRACTICA UNO: Generación de frecuencia 164
PRACTICA DOS: Generación de Voltaje 170
PRACTICA TRES: Control del posición del deslizador (Lazo Abierto) 176
PRACTICA CUATRO: Control de velocidad del deslizador (Lazo Cerrado) . 184
ANEXO 3. PLANOS MECÁNICOS 192
ANEXO 4. DIAGRAMA ELÉCTRICO ..... **Error! Bookmark not defined.**
ANEXO 5. CIRCUITOS IMPRESOS ..... **Error! Bookmark not defined.**
ANEXO 6. HOJA DE DATOS MICROCONTROLADOR ..... **Error! Bookmark not defined.**
ANEXO 7. HOJA DE DATOS DEL SERVOMOTOR ..... **Error! Bookmark not defined.**
ANEXO 8. CARACTERÍSTICAS DEL ACERO ASTM A-500 **Error! Bookmark not defined.**

INDICE DE FIGURAS

Fig. 1. Esquema del microcontrolador ARM-mbed 19
Fig. 2. Esquema del Servomotor. 24
Fig. 3. Subsistemas del módulo 38
Fig. 4. Distribución de las librerías 41
Fig. 5. Clasificación de tornillos 47
Fig. 6. Diseño interno de tornillo de bolas 48
Fig. 7. Tubo de retorno bolas re circulantes 48
Fig. 8. Paso del tornillo 49
Fig. 9. Tipos de tornillos dependiendo de su paso 49
Fig. 10. Elementos de un tornillo de bolas 50
Fig. 11. Error de paso en los tornillos de bolas 50
Fig. 12. Tipos de rodamiento lineal. 52
Fig. 13. Tipos de rodamientos lineales de bolas 54
Fig. 14. Partes de un rodamiento lineal 55
Fig. 15. Rodamiento de riel. 56
Fig. 16. Rodamiento lineal de rodillo cruzado 57
Fig. 17. Rodamiento LBBR 20 58
Fig. 18. Distribución de los rodamientos lineales 59
Fig. 19. Características del tornillo de bolas escogido 61

Fig. 20. Vista superior del deslizador lineal. 63
Fig. 21. Esquema de las conexiones para el deslizador. 65
Fig. 22. Sistema en lazo abierto 65
Fig. 23. Sistema en lazo cerrado 66
Fig. 24. Esquemático del circuito de alimentación, placa principal 67
Fig. 25. Encoder de cuadratura Hohner 68
Fig. 26. Esquemático de fuente dividida para alimentación del amplificador instrumental 70
Fig. 27. Niveles de voltaje para la salida analógica del microcontrolador 71
Fig. 28. Esquemático del circuito de amplificación para la salida analógica 71
Fig. 29. Niveles de voltaje resultantes 72
Fig. 30. Lazo principal en la programación del microcontrolador 73
Fig. 31. Subrutina de Configuración para la comunicación vía Ethernet 74
Fig. 32. Subrutina de configuración serial para la comunicación del microcontrolador 75
Fig. 33. Formato de trama serial 77
Fig. 34. Visualización CONFIG_SERIAL 81
Fig. 35. Diagrama de Bloques - CONFIG_SERIAL 82
Fig. 36. Flujo grama CONFIG_SERIAL 83
Fig. 37. Visualización SERVO_ON 84
Fig. 38. Diagrama de bloques - SERVO_ON 84
Fig. 39. Flujo grama - SERVO_ON 85
Fig. 40. Visualización VERIFICAR_ERROR_MBED 86
Fig. 41. Visualización INTERPRETAR_ERROR 89
Fig. 42. Visualización PTO_SERIAL 91
Fig. 43. Diagrama de bloques - PTO_SERIAL 92
Fig. 44. Flujo grama - PTO_SERIAL 93
Fig. 45. Visualización DIR_SERIAL 94
Fig. 46. Diagrama de bloques - DIR_SERIAL 95
Fig. 47. Flujo grama - DIR_SERIAL 96
Fig. 48. Visualización PULSOS_VELOCIDAD 97
Fig. 49. Diagrama de bloque - PULSOS_VELOCIDAD 98
Fig. 50. Flujo grama - PULSOS_VELOCIDAD. 99
Fig. 51. Visualización PULSOS 100
Fig. 52. Diagrama de bloques - PULSOS_SERIAL 101
Fig. 53. Flujo grama - PULSOS_SERIAL 102
Fig. 54. Visualización ENCODER_SERIAL 104
Fig. 55. Diagrama de bloques - LEER_ENCODER 105
Fig. 56. Flujo grama - LEER_ENCODER 106
Fig. 57. Visualización LIMPIAR_ENCODER 107
Fig. 58. Diagrama de bloques - LIMPIAR_ENCODER 108
Fig. 59. Diagrama de bloques - LIMPIAR_ENCODER 109
Fig. 60. Visualización LEE_RPM 110
Fig. 61. Diagrama de bloques - LEER_RPM 111
Fig. 62. Flujo grama - LEER_RPM 112
Fig. 63. Visualización HOME_SERIAL 113
Fig. 64. Diagrama de bloques - HOME_SERIAL 114
Fig. 65. Flujo grama - HOME_SERIAL 115
Fig. 66. Visualización VOUT_SERIAL 116
Fig. 67. Diagrama de bloques - VOUT_SERIAL 118

Fig. 68. Flujo grama - VOUT_SERIAL 119
Fig. 69. Visualización HTTPRPC 120
Fig. 70. Visualización HTTPRPC_DELETE 121
Fig. 71. Visualización ALARMA_HTTP 122
Fig. 72. Diagrama de bloques - ALM_HTTP 122
Fig. 73. Flujo grama - ALM_HTTP 123
Fig. 74. Diseño final del deslizador 124
Fig. 75. Sujeción del extremo de la placa de soporte 125
Fig. 76. Placas de soporte con sus ejes 125
Fig. 77. Sujeción de la mesa del deslizador 126
Fig. 78. Sujeción a la tuerca del tornillo de bolas 126
Fig. 79. Sujeción del servomotor 127
Fig. 80. Ubicación de potenciómetros para calibración 129
Fig. 81. Diagrama de flujo - Práctica 03. 181
Fig. 82. Configuración e Ir a Inicio 182
Fig. 83. Lazo de espera de cambios en la interfaz 182
Fig. 84. Diagrama del control del tiempo 183
Fig. 85. Diagrama de Bloques - Práctica 04. 189
Fig. 86. Flujo grama - Practica 04 190

INDICE DE TABLAS

Tabla 1. Especificaciones del microcontrolador ARM mbed 19
Tabla 2. Especificaciones Servomotor GSK 110SJT 26
Tabla 3. Tecnologías Ethernet 30
Tabla 4. Precisión en los tornillos de bolas 51
Tabla 5. Ecuaciones básicas para un rodamiento lineal 53
Tabla 6.Características del rodamiento LBBR 20 59
Tabla 7. Características del tornillo de bolas escogido 61
Tabla 8. Características del tornillo de bolas utilizado 61
Tabla 9. Características del encoder Hohner 68
Tabla 10. Lista de posibles comandos seriales 78
Tabla 11. Lista de comandos seriales detallada 79
Tabla 12. Relación voltaje - comando 80
Tabla 13. Réplicas del microcontrolador 87
Tabla 14. Error lineal medido para el servomotor 130
Tabla 15. Resultados de la prueba de velocidad máxima 2000 RPM 132
Tabla 16. Resultados de la prueba de velocidad máxima 1000 RPM 132
Tabla 17. Conector de Encoder 155

INDICE DE ECUACIONES

Ecuación 1. Conversión de kilogramos a Newtons 59
Ecuación 3. Cálculo de vida útil en metros 60
Ecuación 2. Carga soportada por el rodamiento 60

Ecuación 4 Aceleración angular 62
Ecuación 5. Inercia en el tornillo de bolas 62

RESUMEN

El presente trabajo de grado describe el diseño e implementación de un sistema modular para realizar prácticas de servomecanismos y control, el cual brinda a los estudiantes de la Facultad de Ingeniería en Ciencias Aplicadas tener mayor flexibilidad a la hora de realizar sus proyectos, ya que cuentan con una interfaz que les permite controlar uno de los servomotores con los que cuenta el laboratorio de mecatrónica. Este sistema, al ser modular, cuenta con diferentes medios de comunicación y diferentes tipos de control, con lo cual el estudiante tiene muchas más opciones al momento de realizar sus prácticas.

El circuito electrónico permite controlar el servomotor, adicionalmente se implementó librerías de comunicación y control con el entorno de desarrollo LabVIEW, esto permite que la comunicación con el controlador del motor se realice de una manera muy fácil, y aún más, estas librerías cuentan con tres niveles de abstracción, liberando al estudiante de los detalles de implementación del sistema, y de esta manera permitiéndole centrarse en el desarrollo de su aplicación.

Esto no quiere decir que el estudiante esté cerrado a utilizar únicamente este entorno de desarrollo, el sistema puede ser implementado en cualquier lenguaje de programación que permita comunicarse con un puerto serial, algunos de estos programas incluyen: Matlab, python, C/C++, java, entre otros. Con lo cual la flexibilidad a la hora de configurar y utilizar el sistema es muy grande, y las aplicaciones que se pueden realizar son muy variadas.

Esto es importante a la hora de realizar prácticas de control ya que se pueden utilizar diferentes tipos de programación, el control puede ser implementado utilizando LabVIEW, Matlab o un microcontrolador, y ser desplegado en el deslizador de una manera rápida. Es posible modificar el código del controlador y volverlo a ejecutar de una manera rápida, si el servomotor estuviera controlado por un PLC se necesitaría volver a programar el dispositivo con cada cambio del programa, en cambio al utilizar este sistema eso no es necesario ya que el control del sistema lo realiza el programa del estudiante directamente.

ABSTRACT

This paper describes the design and implementation of a modular system for servo-mechanism and control projects, this system allows the students of the Engineering Faculty for Applied Science to have greater flexibility to carry out their projects, because they will have an interface that enables them to control the servo motors in the laboratory of mechatronics, and this system is modular, allowing different communication protocols and different types of control, which the student has many more options when making their practices.

Besides the electronic circuit, which controls the actuator, it has been implemented libraries for communication and control with the LabVIEW development environment, this means communication with the motor controller is done very easily, moreover, these libraries have come in three levels of abstraction, freeing the student from the implementation details of the system, thus allowing you to focus on developing your application.

This does not mean that the student is locked to only use this development environment, the system can be implemented in any programming language that can communicate with a serial port, and some of these programs include: Matlab, Python, C/C++ and java, among others. The flexibility to configure and use the system is very large, and applications that can be performed vary widely.

This is important when making control practices that can use different types of programming, the control can be implemented using LabVIEW, Matlab or a dedicated microcontroller, and be deployed quickly on the slider, you can modify the code driver and re-run it, if the actuator was controlled by a PLC it would be required to reprogram the device with each program change, however when using this system it is not necessary because the system control is performed by the student's program directly on the computer.

PRESENTACIÓN

La carrera de ingeniería en mecatrónica es muy variada en sus campos de estudio, en ella se abarcan diferentes áreas de ingeniería, por lo tanto es importante que al realizar prácticas en los laboratorios, se cuente con el equipamiento necesario para el estudio de estas áreas.

La adición de un deslizador lineal sirve a este propósito brindando al estudiante la capacidad de realizar diversas aplicaciones utilizando el sistema modular que se implementó en el presente trabajo de grado. Las aplicaciones que puede realizar son muy variadas, ya que la modularidad del sistema permite utilizar sus distintas partes de forma aislada, o también, aumentar las capacidades del sistema aumentando sus módulos.

Todo se diseñó tomando en cuenta la participación del estudiante al realizar sus aplicaciones. La programación, la comunicación, los sensores, el tipo de control sobre el motor, todo es susceptible de personalización adecuándose a las necesidades y al nivel de complejidad del proyecto.

Para poder lograr lo aquí expuesto se ha aplicado todos los conocimientos adquiridos a lo largo de la carrera en mecatrónica, el diseño de las librerías, el sistema de control y el diseño del deslizador, todo esto y más es el testimonio de la culminación de años de estudio y dedicación.

CAPITULO I

ANTECEDENTES

Lo que busca este trabajo de grado es ampliar las funcionalidades del módulo de servomecanismos del laboratorio de mecatrónica, para que sus elementos puedan ser utilizados de mejor manera por un amplio grupo de estudiantes, permitiéndoles interactuar con diversos elementos, y brindarles las herramientas necesarias para poder desplegar sus proyectos de una manera rápida, y desarrollar sus aplicaciones utilizando ideas y conceptos con los que ya están familiarizados.

Un deslizador lineal tiene un diseño muy simple, pero a pesar de la sencillez de su diseño brinda la oportunidad a los estudiantes de estudiar la respuesta de un sistema servo controlado y analizar el comportamiento del sistema bajo distintos tipos de cargas mecánicas, estudiar la inercia producida por la carga y modelar un sistema de compensación, todo esto dentro de un mismo proyecto, desde la programación básica del deslizador, hasta la implementación de un sistema de control, integrado en un sólo sistema modular que permite un estudio completo y no como partes separadas.

La programación actual del microcontrolador permite la utilización de todos sus puertos e interfaces de manera dinámica a través de la implementación del protocolo Remote Procedural Call, (RPC) sobre una red Ethernet. Esto posibilita que se le pueda dar muchas aplicaciones más allá del alcance que este trabajo de grado presenta, todo esto sin necesidad de reprogramar el microcontrolador, directamente desde las librerías de LabVIEW que se ofrecen junto con el microcontrolador en su sitio web. Esto convierte al microcontrolador mbed en una muy poderosa tarjeta de adquisición, sin necesidad de hardware adicional.

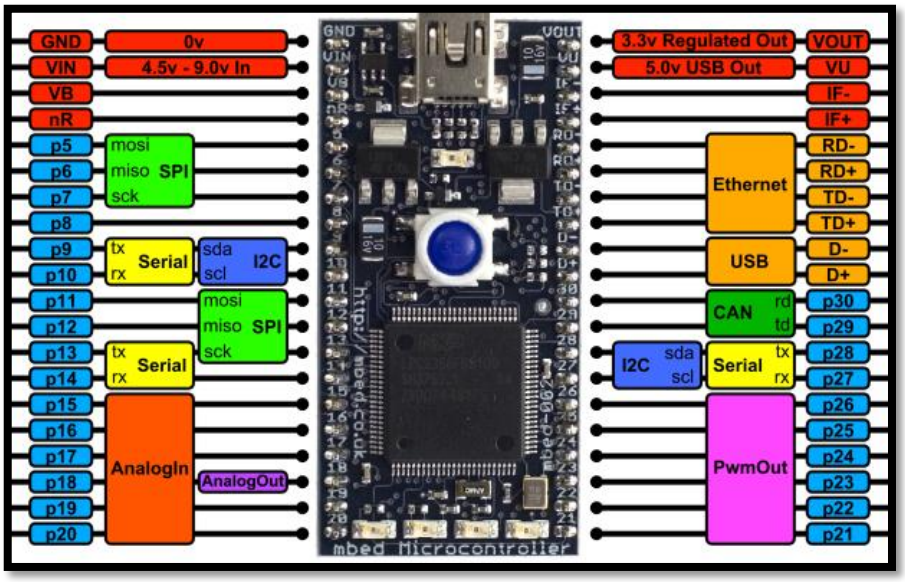
1.1. EL MICROCONTROLADOR ARM MBED

El microcontrolador es la parte más importante del sistema, posee las características necesarias para su implementación, y es el elemento que provee de modularidad y flexibilidad al momento de controlar el deslizador, por tal motivo

se escogió el microcontrolador ARM-mbed el cual reúne todas las prestaciones que el sistema necesita.

Sin las herramientas adecuadas, los detalles de la implementación pueden rápidamente entorpecer el desarrollo de prototipos y la experimentación. El microcontrolador mbed resuelve este problema ya que cubre las bases sobre las cuales desarrollar diversos proyectos.

Fig. 1. Esquema del microcontrolador ARM-mbed



Fuente: (ARM Holdings, 2014)

Está basado en el chip NXP LPC1768, con un núcleo ARM Cortex-M3 de 32 bits corriendo a 96 MHz. Incluye 512KB de memoria FLASH, 32KB RAM y muchas interfaces incluyendo Ethernet incorporado, USB Huésped y dispositivo, CAN, SPI, I2C, ADC, DAC, PWM, y otras interfaces de E/S. El diagrama arriba muestra las interfaces más comunes y su disposición. Note que todos los pines numerados (p5, p30) pueden también ser usados como interfaces de Entrada y Salida digital

Tabla 1. Especificaciones del microcontrolador ARM mbed

Núcleo:	ARM Cortex-M3 LPC1768
Frecuencia:	96 MHz
Memoria Flash	512 KB
Memoria RAM	32KB

Especificaciones del microcontrolador ARM mbed	
Consumo energético	60-120 mA
Conexión Ethernet	1
USB Host	1
USB Device	1
Módulo SPI	2
Módulo I2C	2
Módulo CAN	1
Entradas Analógicas	6
Salidas de PWM	6
Salidas Analógicas	1

Fuente: (ARM Holdings, 2014)

"El microcontrolador ARM mbed está diseñado para desarrollar proyectos de una manera rápida, flexible de bajo costo y de una manera profesional. Para ello cuenta con un el microprocesador ARM Cortex-M3 antes mencionado montado sobre una placa de cuarenta pines, la cual se puede montar en una protoboard.

Se puede alimentar externamente aplicando un voltaje entre 4.5v a 9.0v a la entrada VIN o directamente utilizando el conector mini USB incorporado". (ARM Holdings, 2014)

También se tiene soporte para un puerto serial virtual usando la misma conexión USB, permitiendo comunicación serial con un computador ya sea en una consola serial como Hiperterminal, LabVIEW, Matlab, o cualquier otro tipo de lenguaje de programación que pueda comunicarse con un puerto serial (COM).

Existen dos formas de escribir programas para el dispositivo, una cuando se encuentre conectado a internet, usando el compilador en línea (mbed.org/compiler), o caso contrario se puede utilizar cualquiera de los siguientes compiladores los cuales no requieren una conexión constante a internet: Keil uVision, Code Red, o GCC-ARM.

Para el deslizador se hizo uso de las capacidades de comunicación serial y ethernet, así como la interfaz QED y DAC para comunicarse con el encoder y el control de velocidad respectivamente.

1.1.1. INTERFAZ MBED CON RPC

Hay muchas ocasiones en las cuales es útil ser capaz de comunicarse con mbed desde un ordenador. Esto podría ser para controlar actuadores que utilizan las salidas de mbed, recopilar datos utilizando sensores o crear aplicaciones remotas a través de una red. Crear una interfaz entre mbed y un ordenador puede ser difícil ya que requiere que se especifique un formato de comunicación y luego escribir el código tanto en mbed y en el equipo con el cual se desea conectar.

El microcontrolador mbed es capaz de recibir e interpretar comandos RPC esto puede ser usado para simplificar en gran medida la creación de un interfaz. Los comandos RPC se encuentran en un formato predefinido y se pueden enviar a través de cualquier mecanismo de transporte que pueda enviar una secuencia de texto. Ellos le permiten interactuar directamente con los objetos de mbed.

También se han creado bibliotecas de varias lenguas populares que le permiten utilizar RPC a través de varios mecanismos de transporte sin necesidad de tener que hacer ningún trabajo para establecer el transporte mecanismo o el formato de sus mensajes. Las bibliotecas se han desarrollado para: MATLAB, LabVIEW, Python, Java y .NET.

El microcontrolador es capaz de levantar un servidor HTTP el cual cuenta con un controlador RPC, así que al usar la librería HTTP Server se pueden enviar los comandos RPC sobre HTTP. Estos comandos son enviados al ir agregándolos al URL del microcontrolador en un navegador de la siguiente manera:

```
http://<url de mbed>/rpc/<Nombre del Objeto>/<Nombre del método>
    <Argumentos separados por espacios>
```

Una vez que se establece una dirección IP para el microcontrolador este la transmite mediante el puerto USB serial.

LabVIEW RPC le permite controlar directamente a mbed mediante el protocolo RPC. Se crearon algunos bloques de construcción de la interfaz RPC de mbed con LabVIEW, permitiendo que los programas de LabVIEW puedan interactuar con el mundo real. Esto podría ser usado para cosas como:

- Adquisición de datos en LabVIEW a través de sensores conectados a mbed.
- Actuadores de control conectados a mbed desde LabVIEW.

Se pueden crear programas de LabVIEW con hardware en el bucle, donde los sensores y actuadores se interconectan con mbed pero los cálculos y el control son en LabVIEW

Utilizando estas herramientas se creó una API para el sistema, controlando el deslizador desde LabVIEW mediante una red de datos local.

1.2. EL SERVOMOTOR

Para la implementación del sistema se utiliza un servomotor de corriente alterna, estos motores combinan la potencia de un motor de corriente alterna, con la precisión de un motor de pasos. Lo que nos permite tener una muy alta precisión y velocidades de rotación bajas manteniendo el torque del motor.

Un Servomotor podría definirse genéricamente como un motor utilizado para obtener una salida precisa y exacta en función del tiempo. Dicha salida está expresada habitualmente en términos de posición, velocidad y/o torque.

La aplicación industrial de dichos motores se está desarrollando significativamente por múltiples razones entre las que podemos mencionar: nuevos y más potentes componentes magnéticos para los motores como los imanes de tierras raras, reducción de costo de los motores y los equipos electrónicos necesarios para el control de los mismos, incorporación en dichos equipos electrónicos de nuevas funciones para un control preciso y confiable del movimiento que permiten utilizarlos eficientemente e incorporar nuevas áreas a su dominio de aplicación.

Esencialmente, un motor sin escobillas a imán permanente es una máquina sincrónica con la frecuencia de alimentación, capaz de desarrollar altos torques (hasta 3 o 4 veces su torque nominal) en forma transitoria para oponerse a todo esfuerzo que trate de sacarla de sincronismo. La denominación sin escobillas es una forma de diferenciarlo de sus predecesores, los servomotores a imán permanente alimentados con corriente continua.

En comparación con motores asíncronos de jaula de ardillas (que eroguen el mismo torque/velocidad en su eje) la inercia de un servomotor sin escobillas es sustancialmente menor. Ambas características: sobre torques importantes e inercias reducidas son características apreciadas y útiles para el control del movimiento pues permiten rápidas aceleraciones y deceleraciones así como control preciso de posición en altas velocidades.

Constructivamente el servomotor sin escobillas posee un estator parecido al de un motor de jaula con un núcleo laminado y un bobinado trifásico uniformemente distribuido. El rotor está constituido por un grupo de imanes permanentes fijados en el eje de rotación. La forma de los rotores a imanes varía de acuerdo al diseño y puede clasificarse en cilíndricos o de polos salientes.

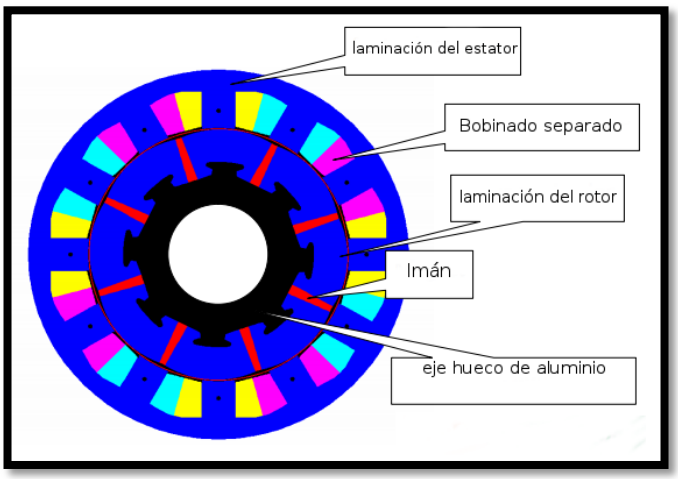
La fijación de los imanes al rotor ha sido uno de los puntos críticos en la construcción de estos motores debido a las altas fuerzas centrífugas a las que se encuentran sometidos durante los procesos de aceleración y frenado. Actualmente se combinan fijaciones mecánicas de diferentes tipos (atadura con fibra de vidrio, chaveteado con diferentes materiales, etc.) con pegado utilizando adhesivos especiales. (León Aguirre & Tapia Vaca, 2009)

Cuando circula corriente alterna en las fases del bobinado de estator se produce un campo magnético rotante en el entrehierro del motor. Si en cada instante el campo magnético generado en el estator intersecta con el ángulo correcto al campo magnético producido por los imanes del rotor generamos torque para lograr el movimiento del motor y la carga acoplada a él.

La utilización de un dispositivo electrónico denominado servodrive o driver para el servomotor para alimentar el estator con la tensión y frecuencia correcta, permite en cada instante, generar un campo magnético en el estator de magnitud

y posición correctamente alineada con el campo magnético del rotor. De esta forma obtenemos el torque necesario para mantener la velocidad y posición deseada del eje del motor.

Fig. 2. Esquema del Servomotor.



Fuente: Autor

El proceso implica conocer en todo instante la posición del rotor para lo cual se equipan los servomotores con dispositivos tales como resolvers, encoders u otros. Los mismos rotan solidariamente con el eje del servomotor e informan al servodrive la posición del rotor. Dichos dispositivos de realimentación de posición se diferencian en la robustez, resolución, capacidad de retener la información de posición ante cortes de alimentación y número de conexiones necesarias entre otras. Por ejemplo en una servo-máquina de tracción directa que rota normalmente a una velocidad nominal de algunas centenas de RPM deberemos seleccionar dispositivos con un alto número de pulsos por revolución a fin de tener control de torque durante la partida y parada del ascensor.

Actualmente los servodriver operan por técnicas de modulación de ancho de pulso (PWM) con configuraciones de hardware (básicamente en la parte de potencia) parecidas a los inversores para el control de motores asincrónicos. De hecho existen en el mercado drivers que permiten controlar ambos tipos de motores.

Debe puntualizar que para la operación normal de un servomotor se necestita un servodrive, el motor no puede ser operado directamente de la red de suministro. El servodrive es el encargado del control rotacional del eje del servomotor. Por tal motivo se normalmente se tienen dos conexiones entre el servomotor y el servodrive, una de ellas se utiliza para llevar la corriente a los devanados del motor, mientras que la otra sirte para el control de posición a través del encoder acoplado al eje del servomotor.

La selección de un servomotor para una determinada aplicación requiere conocer el torque de pico necesario para acelerar y frenar la carga impulsada por el motor así como el torque eficaz requerido por la aplicación. Básicamente el conjunto servodrive - servomotor deben estar en condiciones de satisfacer los requerimientos de torque de pico solicitados por el sistema y el motor debe soportar sin deterioro el régimen térmico impuesto por manejar el torque eficaz requerido por la aplicación.

La utilización de servomotores se está popularizando en todas las ramas de la industria. En el transporte vertical vemos cada vez más frecuentemente aplicaciones que aprovechan la alta capacidad de sobre torque y la baja inercia del motor para lograr un perfecto control del viaje y nivelación aun en muy altas velocidades en máquinas de tracción o posicionamientos perfectos con alto control del torque en operadores de puerta.

Los conjuntos son más eficientes desde el punto de vista rendimiento y consumen menos energía que algunas aplicaciones tradicionales. Por lo tanto es de esperar en un futuro cercano una mayor difusión de este tipo de soluciones acompañada por una baja de su costo, producto de la mayor cantidad de unidades manufacturadas y número de proveedores presentes en el mercado.

1.2.1 EL SERVOMOTOR 110SJT

El servomotor utilizado es el GSK 110SJT el cual está controlado por el driver GSK DA98D. Las características de este servomotor son las siguientes:

Tabla 2. Especificaciones Servomotor GSK 110SJT

Un	**220V**
In	8A
Tn	2 N.m
Nn	3000 RPM
Nmax	3300 RPM
INS	CLASS B
M	2500 r/rev

Fuente: Manual Driver GSK

La unidad DA98D AC servodrive es una nueva generación de la plena unidad de disco digital de AC servo producido por GSK. Este producto incluye dos modos de control de velocidad y posición. Se pueden combinar con diversos sistemas de control en lazo abierto y lazo cerrado y ha sido ampliamente aplicado a máquinas herramientas CNC y la industria de automatización.

Las conexiones eléctricas entre el servomotor y el driver están realizadas y listas en los laboratorios, por lo tanto es necesario solamente realizar la conexión entre el driver y el elemento de control. Este elemento de control tradicionalmente ha sido el PCL Siemens S7-200, pero ahora también existe la posibilidad de controlar el servomotor a través del microcontrolador ARM-mbed.

1.3. PROTOCOLO DE COMUNICACIÓN ETHERNET

En los años 40, la instrumentación de campo todavía se apoyaba en señales de presión para la monitorización de los procesos. En los 60 se introdujo la señal estándar 4-20 mA en las aplicaciones de instrumentación. A pesar de su éxito, señales de diferentes niveles se utilizaban en dispositivos no adecuados al estándar, defendidos por unos u otros fabricantes. El primer autómata programable aparece en 1969. A mediados de los 70, la empresa Honeywell anuncia el primer sistema de control de procesos distribuido (DCCS). En los años 80 aparecieron los sensores inteligentes basados en microprocesador, esto potenció la aparición de los buses de campo que comunicaran los distintos dispositivos de la instalación entre sí.

Desde entonces, tal como ocurrió con la señalización analógica, se realizaron grandes esfuerzos en el control de procesos para unificar tanto las comunicaciones entre dispositivos como los perfiles a los que estos debían responder para garantizar el comportamiento estandarizado. Los bocetos del estándar propuesto por el comité IEC/ISA SP50 se centraron en definir las siguientes funciones:

- **Capa física**. Especifica el medio de transmisión, sería el sustituto digital de la señal 4-20 mA en el entorno de proceso.
- **Capa de enlace**. Especifica las comunicaciones entre dispositivos de un mismo bus, el método de acceso a éste y chequea posibles errores.
- **Capa de aplicación**. Encargada de dar formato de mensaje a los datos, de forma que sean entendibles por el dispositivo receptor y emisor. También ofrece servicios a la capa de usuario.
- **Capa de usuario**. Ofrece a las aplicaciones finales funciones específicas de control e identificación automática de dispositivos.

Sin embargo, ante el retraso en la salida del estándar, cada fabricante abogó de nuevo por implementaciones propietarias. Es el momento de ISP (Interoperable Systems Project) y WorldFIP, que dieron lugar a la actual Fielbus Foundation, o de la Profibus User Organization (PNO). ModBus aparece en 1979; Interbus-S en 1984 y CAN (especifica capas 1 y 2) en 1986. FieldBus Foundation especificó el bus H1 en 1996; un año antes, PNO especifica Profibus PA. AS-Interface (1993) surge como bus especializado en señales todo-nada y posteriormente intentará mejorar sus prestaciones en transmisión de datos analógicos. DeviceNET aparece en 1994. Dentro del campo de la automatización de edificios aparecen BatiBUS, EIB (1990), LonWorks (1991) o BACNet (1995).

Así surgieron multitud de soluciones de comunicación industrial. Tanto el cliente final como integradores y fabricantes debían apostar por una solución u otra sin tener demasiado claras las perspectivas de futuro de dicha solución. La apuesta por cualquiera de ellas suponía además formar a personal especializado para su instalación, puesta en marcha y mantenimiento. Tiempos difíciles para el diseño de redes de control en las que, por ejemplo, la formación previa del

personal propio condicionaba drásticamente las soluciones ofrecidas para los siguientes proyectos.

El estándar Ethernet a 10 Mbps es publicado por el IEEE (802.3) en 1985 y rápidamente conquistó el terreno de las comunicaciones de área local en el entorno ofimático. En 1993 aparecen los primeros conmutadores Full Duplex y Fast Ethernet (100 Mbps) se estandariza en 1995. Una vez asentada esa prevalencia en el sector no industrial, el estándar Ethernet empezó a ser visto como una posible solución a la falta de unificación práctica en las capas física y de enlace de los equipos de bus de campo provenientes de diferentes fabricantes. Los estándares de calidad de servicio en capa 2 no aparecen hasta 1997.

La implantación de Ethernet como soporte para los protocolos de nivel superior era clara a nivel de empresarial (nivel ERP en la estructura de producción) y rápidamente bajó al nivel de Sala de Información (niveles MES y SCADA). El salto al nivel de Control (comunicación entre DCSs, autómatas y sistemas HMI locales) se convirtió en una realidad a medida que la electrónica de red se implementó en las unidades de control de proceso. El paso a nivel de dispositivos de campo es claro en algunos campos de aplicación (Transport Automation) pero lento en procesos continuos y discretos. El motivo lo debemos buscar en la naturaleza de los dispositivos finales empleados

La interoperabilidad en capa 1 y 2 da a la electrónica de red Ethernet un impulso industrial notable. A los fabricantes les ofrece la posibilidad de ofrecer soluciones basadas en diferentes protocolos superiores y por lo tanto acceder a mayores mercados. A los instaladores y diseñadores les facilita la vida al permitir unificar el medio físico independientemente de la red que estén tirando (de oficina o de producción). La gestión del conocimiento también se facilita por no ser necesario personal extremadamente especializado en un sistema de comunicación propietario para poner en marcha o mantener instalaciones muy concretas. Incluso el personal encargado de las redes empresariales podría llegar a hacerse cargo de la red industrial. El personal de automatización también se beneficia del hecho de contar con un conocimiento estable en el tiempo y compatible con todos los protocolos superiores que se puedan plantear en cada aplicación. A nivel de mantenimiento la posibilidad de reducir dificultades para

centralizar las islas de automatización de sistemas heredados supone sin duda una ventaja importante.

Según Smith (2005) El protocolo ethernet es la red que más se ha difundido en oficinas e industrias globalmente. Su infraestructura, interoperabilidad y escalabilidad aseguran un fácil desarrollo. Una vez que un equipo ha sido integrado a una red ethernet es posible su monitoreo a través de internet.

Las tecnologías Ethernet que existen se diferencian en estos conceptos:

Velocidad de transmisión.- velocidad a la que transmite la tecnología.

Tipo de cable.- Tecnología del nivel físico que usa la tecnología.

Longitud máxima.- Distancia máxima que puede haber entre dos nodos adyacentes (sin estaciones repetidoras).

Topología.- Determina la forma física de la red. Bus si se usan conectores T (hoy sólo usados con las tecnologías más antiguas) y estrella si se usan hubs (estrella de difusión) o switches (estrella conmutada).

“En un sentido general, la comunicación entre ordenadores se lleva a cabo por el intercambio de datos -información codificada de alguna manera la cual depende del sistema. Podemos considerar este cambio como llevado a cabo en pasos discretos, que llamaremos comunicaciones elementales, en cada uno de los cuales un mensaje es transmitido” (Sharp, 2008)

A continuación se especifican los anteriores conceptos en las tecnologías más importantes: (Tabla 3.)

Ethernet es un estándar de redes de área local para computadores con acceso al medio por contienda CSMA/CD. CSMA/CD (Acceso Múltiple por Detección de Portadora con Detección de Colisiones), es una técnica usada en redes Ethernet para mejorar sus prestaciones.

El protocolo CSMA/CD incorpora dos mejoras que aumentan el rendimiento en una red: en primer lugar, no se transmite si hay otra estación hablando, y en segundo, si mientras se está transmitiendo detecta que otra estación transmite (es decir, se produce una colisión), la estación se calla, en lugar de seguir transmitiendo inútilmente al final de la trama.

Tabla 3. Tecnologías Ethernet

Tecnologías Ethernet

Tecnología	Velocidad de transmisión	Tipo de cable	Distancia máxima	Topología
10Base2	10 Mbps	Coaxial	185 m	Bus (Conector T)
10BaseT	10 Mbps	Par Trenzado	100 m	Estrella (Hub o Switch)
10BaseF	10 Mbps	Fibra óptica	2000 m	Estrella (Hub o Switch)
100BaseT4	100Mbps	Par Trenzado (categoría 3UTP)	100 m	Estrella. Half Duplex (hub) y Full Duplex (switch)
100BaseTX	100Mbps	Par Trenzado (categoría 5UTP)	100 m	Estrella. Half Duplex (hub) y Full Duplex (switch)
100BaseFX	100Mbps	Fibra óptica	2000 m	No permite el uso de hubs
1000BaseT	1000Mbps	4 pares trenzado (categoría 5e ó 6UTP)	100 m	Estrella. Full Duplex (switch)
1000BaseSX	1000Mbps	Fibra óptica (multimodo)	550 m	Estrella. Full Duplex (switch)
1000BaseLX	1000Mbps	Fibra óptica (monomodo)	5000 m	Estrella. Full Duplex (switch)

Fuente: (Llano, 2014)

Se produce una “colisión” cuando dos o más estaciones empiezan a transmitir simultáneamente o con una separación en el tiempo de propagación que las separa. Por ejemplo, en una red donde el tiempo de ida y vuelta es igual a 5.06 ms se producirá una colisión siempre que los dos nodos transmiten con una separación en el tiempo menor de 2.53 ms. Si la separación es mayor que 2.53 ms, no se producirá colisión.

La trama en una red Ethernet puede variar entre un mínimo de 72 bytes y un máximo de 1526. De este modo, la máxima tasa de colisiones para una trama Ethernet variará de acuerdo con el tamaño de la trama. Las operaciones en Ethernet necesitan un “tiempo muerto” entre tramas de 9,6 ms. El tiempo por bit para una Ethernet a 10 Mbps es 1/10-7 o 100 ns. Con base en lo anterior se puede calcular el máximo número de tramas por segundo para tramas de 1526 bytes:

9.6 ms + 1526 bytes * 8 bits/bytes
9.6 ms + 12208 bits * 100ns/bit
1.23 ms

Así en un segundo puede haber un máximo de 1/1.23 de tiempo por bit u 812 bytes de tamaño máximo de trama. Para un mínimo de tamaño de trama el tiempo por trama es:

9.6 ms +72 bytes * 8 bits/bytes * 100 ns/bit
67.2 x 10-6 s

De esta forma, en un segundo puede haber un máximo de 1/67.2 x 10-6 ms (en tiempo por bit) o 14880 bytes y un tamaño mínimo de trama de 72 bytes.

Según IDC (International Data Corporation), a finales de 1997 más del 85% de las conexiones de red instaladas en el mundo eran Ethernet, lo cual representa unos 118 millones de ordenadores. El 17% restante está formado por Token Ring, FDDI, ATM y otras tecnologías. Todos los sistemas operativos y aplicaciones populares son compatibles con Ethernet, así como las pilas de protocolos de niveles superiores, tales como TCP/IP, IPX, NetBEUI y DECnet.

Así pues, la hegemonía en el mundo de las redes locales que Ethernet ha disfrutado desde su debut comercial en 1981 no sólo se mantiene sino que parece ir más lejos. Todos sus competidores han quedado en el camino. ATM, que durante algún tiempo parecía ser el futuro de las redes locales, no sólo no ha conquistado al usuario final sino que al parecer está desplazando rápidamente al backbone de campus por Gigabit Ethernet. Más aún, las últimas tendencias en redes de área extensa de muy alta velocidad basadas en DWDM (Dense Wavelenght Division Multiplexing) estudian la posibilidad de sustituir las tecnologías tradicionales ATM y SONET/SDH como medio de transporte de tráfico IP por una versión de Ethernet que funcione a 10 Gbps". (Márquez Días, Pardo Sánchez, & Pizarro Valencia, 2014)

Seguridad de red.- Las pegas a la inseguridad ofrecida por las redes apoyadas en Ethernet son fácilmente solventables mediante el uso de técnicas ya integradas en la electrónica estándar. Estas técnicas se pueden aplicar a distintos niveles:

• Nivel de puerto. Posibilidad de especificar qué equipos pueden comunicar a través de qué puertos. Se basan en las direcciones MAC o IP del equipo conectado.

• Nivel de nodo. La implementación de protección contra ataques de denegación de servicio (DoS) en base a limitadores de broadcast o de listas de control de acceso (ACL) para especificar con quién puede hablar un dispositivo concreto y con quién no.

• Nivel de diseño. Definición de reglas de acceso en los accesos remotos mediante dispositivos cortafuegos específicamente diseñados para esta función y fácilmente integrables en estructuras redundantes.

• Nivel de dispositivo final. Los equipos conectados a una red Ethernet deben ser capaces de reaccionar adecuadamente ante un ataque externo. Esto no afecta tanto a la electrónica de red como a la implementación del equipo final (PC, PLC, DCI...). Un ejemplo típico es un autómata con procesador único que sufre un ataque de denegación de servicio. La sobrecarga de comunicaciones podría llegar

a afectar al ciclo de procesamiento. La mayoría de los controladores industriales actuales están preparados para este tipo de contingencia.

Interfaz hombre-máquina.- Decíamos que una de las grandes ventajas de los buses de campo basados en Ethernet es ese conocimiento común en los niveles 1 y 2. En nivel 1, las particularidades de instalación de los enlaces serán los mismos para todos los proyectos. En el nivel 2 nos encontraremos con la configuración de la electrónica de red. Se ha realizado un esfuerzo en facilitar el manejo dichos equipos. El resultado es el interfaz gráfico apoyado en explorador http. Esto elimina la necesidad de herramientas especiales de configuración (basta con tener Internet Explorer en el equipo desde el que configuramos).

Una vez puesta en marcha la red, el personal de mantenimiento se enfrentará a la labor de conocer y diagnosticar el estado de la red. La solución tradicional para esto han sido las herramientas de gestión SNMP. Sin embargo ya existen herramientas software que, apoyándose en los mismos protocolos estándar, ofrecen una apariencia de sistema SCADA, haciendo transparente para el usuario el protocolo de gestión de red. Por último, las herramientas de integración apoyadas en OPC permiten incluir el estado de los dispositivos de red en cualquier sistema SCADA preexistente.

El avance de la tecnología Ethernet en el sector industrial es fuerte gracias a su adaptación a los retos que su nuevo ambiente plantea. La permanencia de Ethernet como solución en el tiempo está garantizada por el tremendo apoyo que recibe por parte de los fabricantes de equipos de control, integradores y clientes de sistemas industriales.

Sin duda el paso definitivo de Ethernet hacia el nivel de dispositivo final vendrá de la mano de la integración generalizada de interfaces Ethernet en éstos. A medida que el dispositivo va siendo más simple, la integración de un interfaz más complejo puede suponer un coste significativo añadido al equipo. En este sentido los módulos de entrada-salida distribuidas, tan habituales en las soluciones de control, agrupan conjuntos I/O que utilizan un único interfaz cuyo coste adicional está más justificado. (Llano, 2014)

1.3.1. PROTOCOLO RPC

“El protocolo de Llamada de Procedimiento Remoto (RPC, por sus siglas en inglés, Remote Procedural Call) de Microsoft es una tecnología de gran alcance para crear programas distribuidos cliente/servidor. RPC es una técnica de comunicación entre procesos que permite que el software cliente y servidor se comuniquen.

Los sistemas operativos y programas se han ido volviendo cada vez más complejos a lo largo de los años. Con cada lanzamiento, hay más características. La complejidad creciente de los sistemas hace que sea más difícil para los desarrolladores evitar errores durante el proceso de desarrollo.

RPC está diseñado para mitigar estos problemas al proporcionar una interfaz común entre aplicaciones. RPC sirve como intermediario para hacer la interacción cliente/servidor más fácil y segura mediante la factorización de tareas comunes, como la seguridad, sincronización y manejo de flujo de datos, en una biblioteca común para que los desarrolladores no tengan que dedicar su tiempo y esfuerzo en desarrollar sus soluciones.

RPC de Microsoft es un mecanismo de comunicación entre procesos (IPC), que permite el intercambio de datos y la invocación de la funcionalidad que reside en un proceso diferente. Ese proceso puede estar en el mismo equipo, en la red de área local (LAN), o a través de internet. [...] Con RPC, la lógica esencial de la programación y el código de procedimiento relacionado pueden existir en diferentes equipos, lo cual es importante para las aplicaciones distribuidas” (Microsoft, 2013)

1.4. PROGRAMACIÓN GRÁFICA EN LABVIEW

“LabVIEW es una herramienta diseñada especialmente para monitorizar, controlar, automatizar y realizar cálculos complejos de señales analógicas y digitales capturadas a través de tarjetas de adquisición de datos, puertos serie y GPIBs (Buses de Intercambio de Propósito General).

Es un lenguaje de programación de propósito general, como es el Lenguaje C o Basic, pero con la característica que es totalmente gráfico, facilitando de esta manera el entendimiento y manejo de dicho lenguaje para el diseñador y programador de aplicaciones tipo SCADA.

Incluye librerías para la adquisición, análisis, presentación y almacenamiento de datos, GPIB y puertos serie. Además de otras prestaciones, como la conectividad con otros programas, por ejemplo de cálculo, y en especial Matlab.

Está basado en la programación modular, lo que permite crear tareas muy complicadas a partir de módulos o sub-módulos mucho más sencillos. Además estos módulos pueden ser usados en otras tareas, con lo cual permite una programación más rápida y provechosa.

También ofrece la ventaja de “debugging” en cualquier punto de la aplicación. Permite la posibilidad de poner “break points”, ejecución paso a paso, ejecución hasta un punto determinado y se puede observar como los datos van tomando valores a medida que se va ejecutando la aplicación. Además también lleva incorporado generadores de señales para poder hacer un simulador.

LabVIEW tiene la característica de descomposición modular ya que cualquier VI que se ha diseñado puede convertirse fácilmente en un módulo que puede ser usado como una sub-unidad dentro de otro VI. Esta peculiaridad podría compararse a la característica de procedimiento en los lenguajes de programación estructurada.

Es un sistema abierto, en cuanto a que cualquier fabricante de tarjetas de adquisición de datos o instrumentos en general puede proporcionar el driver de su producto en forma de VI dentro del entorno de LabVIEW. También es posible programar módulos para LabVIEW en lenguajes como C y C++, estos módulos son conocidos como SubVI’s y no se difieren a los VI creados con LabVIEW salvo por el interfaz del lenguaje en el que han sido programados. Además estos SubVI’s son muy útiles por ejemplo en el campo de cálculos numéricos complejos que no se encuentran incluidos en las librerías de LabVIEW.” (Zuñiga Tufiño, 2008)

CAPITULO II

ARQUITECTURA DEL SISTEMA

2.1. DESCRIPCIÓN GENERAL

El sistema se plantea como una herramienta de fácil manejo para los estudiantes. Por tal motivo se diseñó un sistema en el cual las posibilidades de cometer errores sean mínimas, y que a la vez brinde flexibilidad para que los estudiantes puedan experimentar con el módulo.

Este sistema está pensado para que el usuario pueda programar el módulo y elija el método de comunicación que deseé ya sea vía ethernet o serial, y que además que pueda hacer uso de los diferentes modos de funcionamiento del servomotor. Por lo tanto se han definido las funciones básicas que permiten controlar al motor, lo que es llamado la interfaz entre el programador, en este caso el estudiante, y la aplicación. Estas funciones básicas o rutinas, están concebidas como bloques funcionales los cuales permiten programar la aplicación.

La interfaz programador/aplicación, “Aplication Programmer Interface (API) está implementada de tal manera que sea de fácil manejo para un amplio rango de estudiantes, con esto en mente se creó tres niveles de programación. El nivel bajo, con el cuál se entra en contacto directamente con el microcontrolador, el nivel medio se construye sobre el nivel bajo y sirve para controlar el motor, y el nivel alto, con el cual se puede controlar directamente el deslizador.

El microcontrolador recibe los comandos de la librería de nivel bajo, los procesa y ejecuta, al ejecutarlos lo que hace es enviar las señales eléctricas necesarias al driver del servomotor para que, este a su vez, ponga en funcionamiento el motor y accione el deslizador.

La protección del deslizador es importante, ya que al ser utilizado por muchos estudiantes para realizar diversos tipos de prácticas es posible que se lo utilice de manera equivocada. Es de especial importancia la protección de la base del deslizador, esto se lo realiza a través de sensores ópticos que impiden que la

base choque contra los extremos del deslizador, enviando una señal de alerta al usuario y modificando el comportamiento del deslizador para evitar estas colisiones. Este control se lo implementa dentro del circuito electrónico, lo cual hace que el sistema esté protegido independientemente de la aplicación que se haya programado por el estudiante.

2.2. CONCEPCIÓN A BLOQUES

El sistema cuenta de tres partes definidas, las cuales son: el subsistema electrónico, el subsistema de control y el subsistema de mecánico.

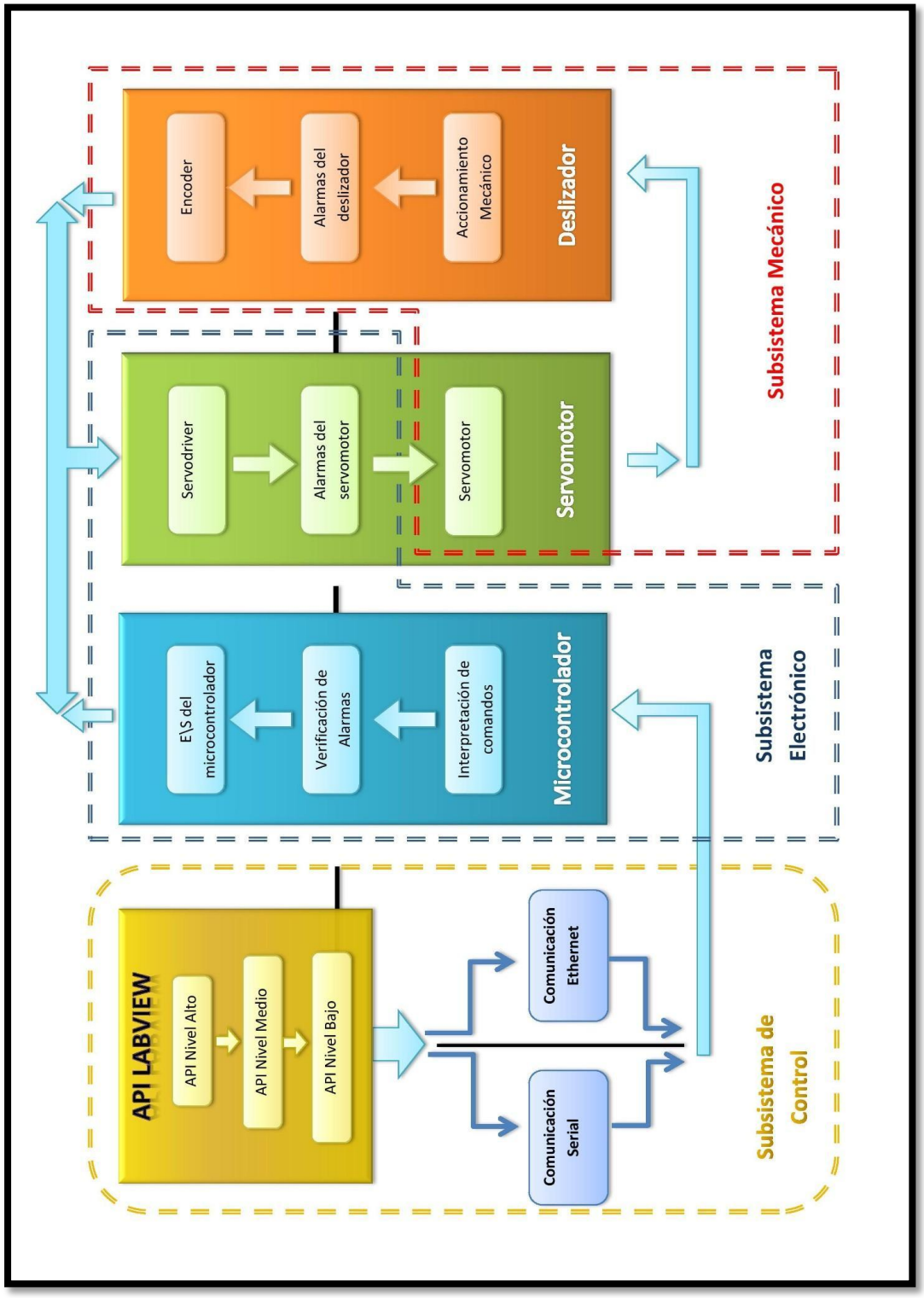
El subsistema electrónico es el encargado de hacer llegar al servodriver los comandos enviados desde el computador o microcontrolador. Aquí es importante que la conexión entre el microcontrolador y el driver se realice de manera correcta, pues es aquí donde existe mayor riesgo de dañar el equipo por su mal uso, por lo tanto se han tomado las medidas de seguridad necesarias para proteger el circuito tanto del microcontrolador como del servomotor etiquetando los conectores que son requeridas para el control del servomotor. En este subsistema esta implementada la seguridad del deslizador, la cual al recibir la señal de proximidad de la base del deslizador a cualquiera de sus extremos detiene el motor sin la necesidad de intervención del usuario.

El sistema de control lo componen la interfaz de programación y el mecanismo de comunicación con el microcontrolador. El usuario puede elegir entre una conexión vía ethernet o serial, dentro de la comunicación serial además puede elegir la velocidad en baudios de la conexión serial y si se utiliza el puerto Serial-USB integrado en el microcontrolador, o una conexión serial estándar a través de un conector DB-9. La interfaz de programación está desarrollada en LabVIEW, pero al ser este un sistema abierto, se tiene acceso al código fuente tanto de la interfaz como del microcontrolador, permitiendo controlar el servomotor usando cualquier otro lenguaje de programación si el usuario así lo desea.

El sistema mecánico es la salida del sistema sobre el cual realizaremos las diferentes prácticas. Lo componen el deslizador, el servomotor, y el encoder encargado de leer la posición del deslizador, para transferir el torque del motor en

deslizamiento lineal se implementó un tornillo de bolas re circulantes, el cual permite un alto grado de precisión, la base del deslizador cuenta con rodamientos lineales el cual permite a la base del deslizador tener un movimiento suave y de baja fricción.

Fig. 3. Subsistemas del módulo



Fuente: Autor

2.3. DETERMINACIÓN DE SUBSISTEMAS

2.3.1. ESTRUCTURA DEL DESLIZADOR

El deslizador es el elemento mecánico del sistema, transforma el movimiento rotacional del motor en movimiento lineal mediante la utilización de un tornillo de bolas re circulantes, el cual está conectado con la base del deslizador y permite su movimiento lineal. Desde el punto de vista mecánico su diseño es muy sencillo, posee un grado de libertad y cuenta con un encoder de posición relativa para su eje principal.

El tornillo desliza la base la cual está apoyada en dos guías laterales las cuales soportan el peso de la base, los apoyos usan rodamientos lineales para mejorar el deslizamiento y reducir la fricción.

El motor se encuentra atornillado a la estructura del deslizador y se une al tornillo mediante un acople o matrimonio, este acople está diseñado para permitir cierta desalineación entre el eje del motor y el tornillo, la cual se mantuvo en un mínimo durante la construcción del deslizador.

2.3.2. CONTROLADOR MICROPROCESADO

Este dispositivo permitirá reemplazar al PLC que actualmente se ocupa y amplía sus funciones al permitir mecanismos de comunicación más flexibles, lo cual permite crear programas y ejecutarlos en el mismo computador, reduciendo el tiempo de prueba y error inherente al aprendizaje del funcionamiento de un servomotor.

Las funciones que el microcontrolador debe desempeñar son: controlar el driver del servomotor, registrar los cambios del encoder del deslizador, y establecer comunicación con el programa del usuario.

Existen dos tipos de control del servomotor, el control de posición y de velocidad, el microcontrolador se encarga de ambos tipos de control. Cuando el motor se encuentra en el modo de control de posición el microcontrolador se encarga de generar un tren de pulsos para generar movimiento en el motor,

mientras que para el control de velocidad el microcontrolador genera una salida de voltaje variable que permite controlar la velocidad del motor.

La generación de este tren de pulsos puede llegar a ser a una alta frecuencia, por tal motivo se aprovechó las características del dispositivo de tal manera que la generación del tren de pulsos se lo realiza en los módulos del temporizador propio del microcontrolador, liberando de esta manera su núcleo de procesamiento para atender los comandos enviados por el usuario y las alarmas generadas por el módulo.

Para el control de velocidad se utiliza el conversor digital analógico propio del microcontrolador, logrando con esto una resolución de 12 bits para la generación analógica de este voltaje.

Otra característica destacada del microcontrolador es que posee un decodificador de cuadratura el cual es de gran utilidad en conjunción con el encoder, ya que permite llevar un registro de la posición relativa del mismo sin la necesidad de que el núcleo central intervenga, esto quiere decir que el conteo se lleva de forma independiente por un módulo especializado dentro del mismo microcontrolador.

Para la comunicación con el programa creado por el usuario se tienen dos mecanismos de transporte de los comandos, uno de manera serial, a través del protocolo RS-232 y otro mediante una red local Ethernet, a través del protocolo RPC.

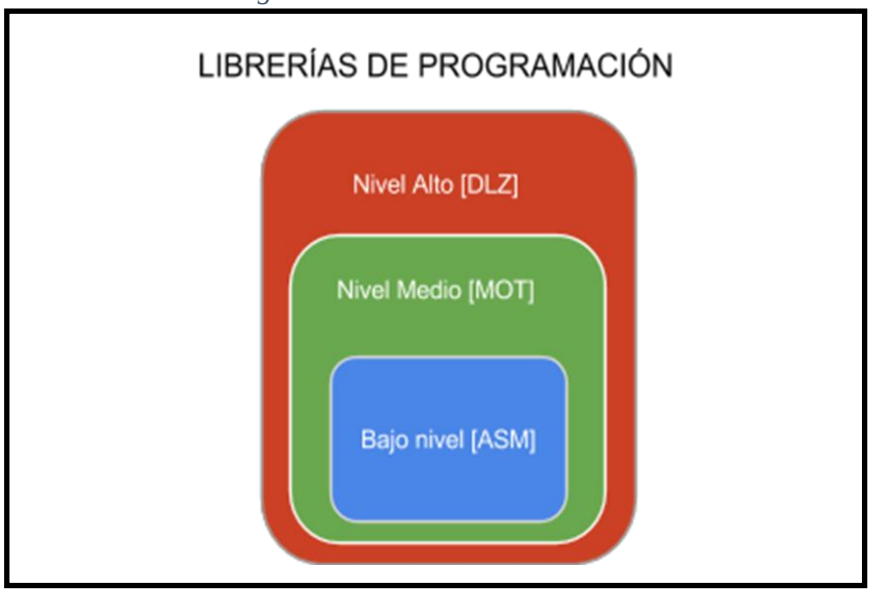
Dentro del protocolo RS-232 se han creado configuraciones dependiendo del tipo de aplicación y la plataforma en la que se esté desarrollando la aplicación. Se puede utilizar el puerto Serial-USB incluido en el microcontrolador, o se puede optar por usar un puerto serial con lógica TTL, y finalmente se brinda la opción conectar este mismo puerto TTL a una salida RS-232 estándar mediante el uso de un conversor MAX232 y un conector DB9.

Todas estas opciones de comunicación son configuradas por el usuario de una manera muy fácil mediante el uso de un switch de configuración que se encuentra en la placa principal del microcontrolador.

Por parte de la comunicación Ethernet, el microcontrolador cuenta con un módulo interno dedicado a este fin, el cual puede establecer la comunicación necesaria para facilitar el alcance que se le pueden dar a las prácticas al crear una red local en la cual se puede interactuar con el deslizador.

En el corazón del controlador tenemos al microcontrolador ARM-mbed en el cual se implementó las funcionalidades anteriormente descritas. Este microcontrolador es el principal elemento del sistema, las características con las que cuenta lo hacen ideal para este tipo de aplicaciones en las cuales se requiere de varias interfaces de comunicación y una ejecución rápida del código.

Fig. 4. Distribución de las librerías



Fuente: Autor

2.3.3. API DEL CONTROLADOR EN LABVIEW

Para controlar al deslizador se creó una interfaz de programación en LabVIEW. Esta permite controlar distintas variables del microcontrolador y consecuentemente del deslizador.

De este modo existen tres niveles: bajo nivel o del microcontrolador, nivel medio o del motor, y nivel alto o del deslizador. Las librerías de nivel medio y nivel alto se construyen sobre la librería de bajo nivel, liberando al estudiante de los cálculos específicos del servomotor y permitiendo un rápido desarrollo del código.

El nivel bajo es el nivel básico, el cual envía los comandos al microcontrolador. Es el nivel que habla directamente con el protocolo de comunicación sea este serial o ethernet. Las variables a controlar en este nivel son, el voltaje de salida para el control de velocidad, la frecuencia de salida para el control de posición, la dirección de giro del eje del motor, el encendido o apagado del servomotor, y lectura de la posición del encoder.

El nivel medio se construye sobre el nivel bajo, es decir, se utiliza la librería de nivel bajo para controlar directamente las variables el motor, como su velocidad y posición, éstas variables dependen de la configuración de los parámetros del servodriver, por lo que esta librería toma en cuenta estos datos y calcula cuales son los valores de las variables del microcontrolador para que la respuesta del motor sea la deseada. Finalmente utiliza la librería de bajo nivel para comunicar estos valores al microcontrolador.

2.3.4. RESPUESTA

El servomotor es nuestro elemento actuador, aquel que genera la respuesta del sistema, debido a su construcción y prestaciones ofrece un alto nivel de precisión en sus dos modos de operación, tanto para el control de posición como de velocidad. Especialmente para el control de posición ya que brinda una precisión de 1/10000 Pasos por revolución, lo cual se traduce en una precisión de 0.036° por revolución.

El control de velocidad por otra parte, aunque preciso del lado del servomotor pierde precisión debido a que su modo de control a través de voltaje es analógico y sujeto interferencias al momento de llevar la señal de voltaje desde el controlador microprocesado hasta el servodrive. Se puede mitigar en algo estos efectos mediante una configuración adecuada de los parámetros del servodriver. Esto no necesariamente es algo malo ya que brinda la posibilidad de ser discutido y analizado en las clases de servo-mecanismos. Es en este tipo de control de velocidad donde la presencia de un encoder rotacional tiene gran importancia, ya que se puede controlar la velocidad del servomotor de una manera más precisa y aplicar diversos métodos de control que aseguren que la velocidad deseada se

mantenga, permitirá crear lazos de control para ser estudiados por los estudiantes.

El control de velocidad o posición es configurado en el servodriver, esta configuración no pueden ser cambiados remotamente desde el programa del usuario. Es importante que los parámetros sean los correctos de acuerdo a la operación que se va a realizar con el servomotor.

2.3.4.1. Control de Posición

El control de posición para del servomotor se lo realiza mediante un tren de pulsos, similar al funcionamiento de un motor de pasos cada pulso enviado al servodrive hace que el servomotor recorra un paso.

El servomotor tiene una resolución de 10 000 pasos por revolución, si se envía 10 000 pulsos al servodrive este rotará el eje del motor en una revolución completa.

La velocidad con la que estos pulsos llegan al servomotor determina la velocidad con la que éste rota. Un punto importante a considerar es el factor de engranaje electrónico con que cuenta el encoder interno, ya que este puede ser modificado mediante dos parámetros del servodriver [PA:12 y PA:13], el cual, esencialmente, modifica electrónicamente la resolución del servomotor, permitiendo que se pueda dar una revolución completa con un menor número de pulsos, esto sirve para aumenta la velocidad a la que gira el eje del motor sin aumentar la velocidad del tren de pulsos a costa de perder la precisión angular máxima con la que cuenta el servomotor.

La frecuencia máxima con que se puede alimentar el servodriver es mediante un tren de pulsos a una frecuencia de 250kHz, esto fue tomado en cuenta en el diseño de la placa electrónica logrando alcanzar esta frecuencia y por lo toda el rango de trabajo del servomotor.

2.3.4.2. Control de Velocidad

En el modo de control de velocidad del servomotor el servodriver gira el eje del servomotor acorde al nivel de tensión detectado entre los terminales VCMD y

AGND de su conector, este voltaje puede ser de más menos diez voltios, cuando es cero, el motor permanece detenido, si el voltaje es positivo con respecto a AGND el motor girará en el sentido de las manecillas del reloj, si es negativo girará al contario de las manecillas del reloj.

Para poder tener un control de todo el rango de velocidades del servomotor se implementó el circuito necesario para llegar a estos valores de voltaje. El control de velocidad del servomotor resulta bastante práctico para realizar el estudio de lazos de control, ya que al contar con un control analógico, y utilizando el encoder como sensor de velocidad rotacional, se tiene los elementos necesarios para un lazo de control cerrado, permitiendo la implementación de diversas técnicas de control, ya sea, PID, Fuzzy, o similar.

CAPITULO III

DISEÑO DEL SISTEMA

3.1. ESTRUCTURA DEL DESLIZADOR

Para poder transformar el movimiento rotacional del eje del servomotor en movimiento lineal es necesario diseñar una estructura mecánica que permita realizar esta acción. En el presente capítulo, se expresan los diferentes requerimientos, limitaciones, consideraciones y especificaciones que se dispondrán para el dimensionamiento del deslizador lineal.

3.1.1. ALCANCE

El sistema mecánico del deslizador será diseñado, construido y montado de tal manera que permita cumplir con los siguientes requerimientos:

- La longitud del desplazamiento lineal debe ser de 70cm
- Se debe tener una precisión de hasta 0.1 mm
- La carga máxima que debe soportar es de 30 Kg

3.1.2. SISTEMA MECÁNICO

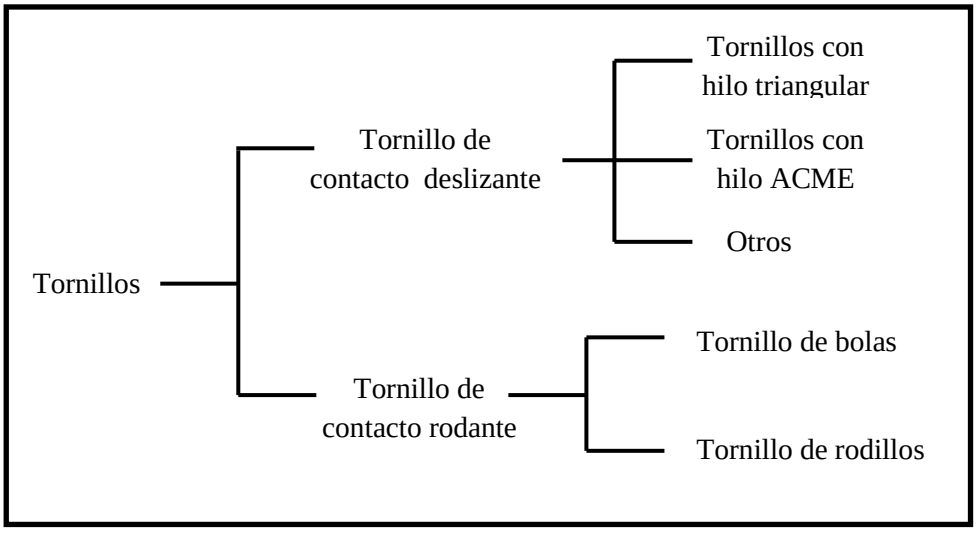
La forma más sencilla y práctica para trasmitir el torque del servomotor en movimiento lineal es mediante la utilización de un tornillo y tuerca, al restringir el movimiento de la tuerca esta se desplazará linealmente a medida que gire su tornillo.

Para facilitar el movimiento lineal es necesario que el peso de la base y la carga no se aplique directamente al tornillo, para este fin se implementó guías las cuales permitan sostener el peso de la carga. Es necesario tomar en cuenta que mecanismo nos servirá para que la base se desplace sobre las guías, para ello el elemento que permita el movimiento lineal debe tener un bajo coeficiente de fricción.

3.1.2.1. Transmisión de potencia

La conversión de movimiento rotacional en movimiento de traslación se logra mediante la implementación de un tornillo y tuerca. Existen diferentes tipos de tornillos para la transmisión del torque, estos se clasifican de acuerdo al contacto que existe entre la tuerca y el tornillo.

Fig. 5. Clasificación de tornillos



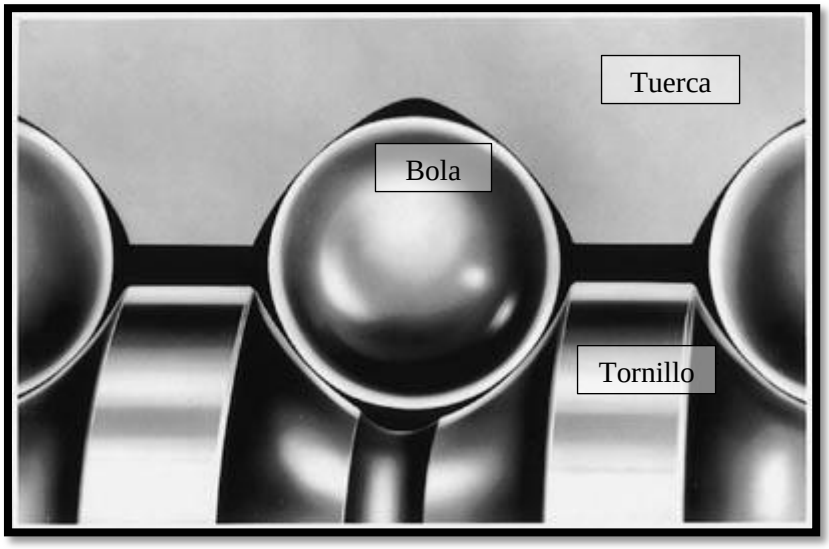
Fuente: (NSK Corporation, 2005)

Los tornillos de hilo triangular y trapezoidal tienen varias desventajas comparados a los tornillos de bolas y rodillos, principalmente su menor tiempo de vida útil debido a la fricción a la que están sometidos y su precisión es inferior a la necesaria, por esta razón se han escogido un tornillo de bolas.

“La mayoría (90% o más) de la fuerza usada para rotar el eje del tronillo puede ser convertida a la fuerza para mover la tuerca de bolas. (Debido a que la perdida por fricción es extremadamente baja, la cantidad de fuerza usada para rotar el eje del tornillo es tan baja como un tercio de la necesaria para el tornillo tipo acme).

Al proveer bolas de acero entre el eje del tornillo y la tuerca (con ranuras), y las bolas ruedan dentro de las ranuras (es decir, cambiando el contacto deslizando por uno rodante se reduce la fricción)” (NSK Corporation, 2005).

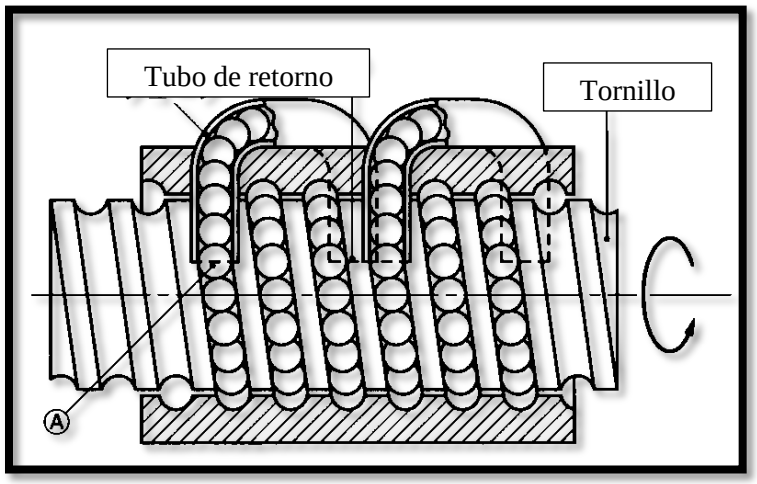
Fig. 6. Diseño interno de tornillo de bolas



Fuente: (NSK Corporation, 2005)

Para prevenir que las bolas se salgan de la ranura donde se encuentran los tornillos implementan tubo el cual sirve para la recirculación de las bolas.

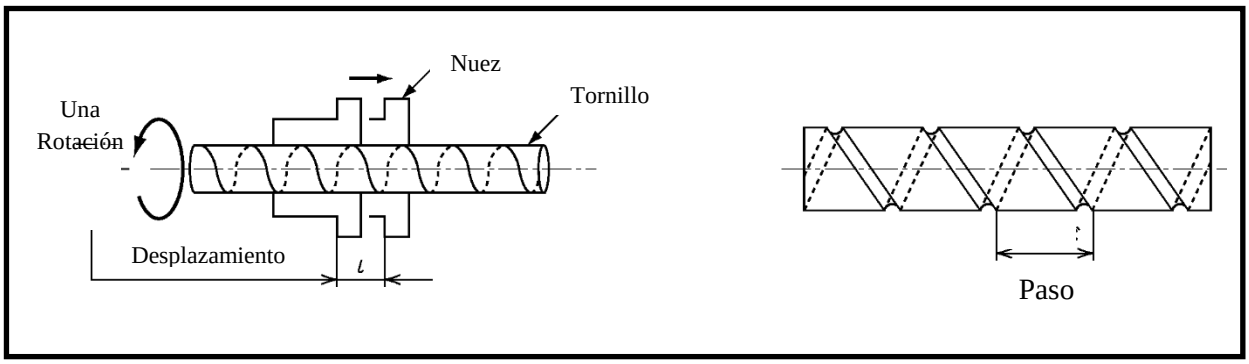
Fig. 7. Tubo de retorno bolas re circulantes



Fuente: (NSK Corporation, 2005)

La distancia lineal recorrida al rotar el eje una revolución se conoce como el paso del tornillo. En nuestro caso, para el tornillo escogido en el trabajo se tiene un paso de cinco milímetros.

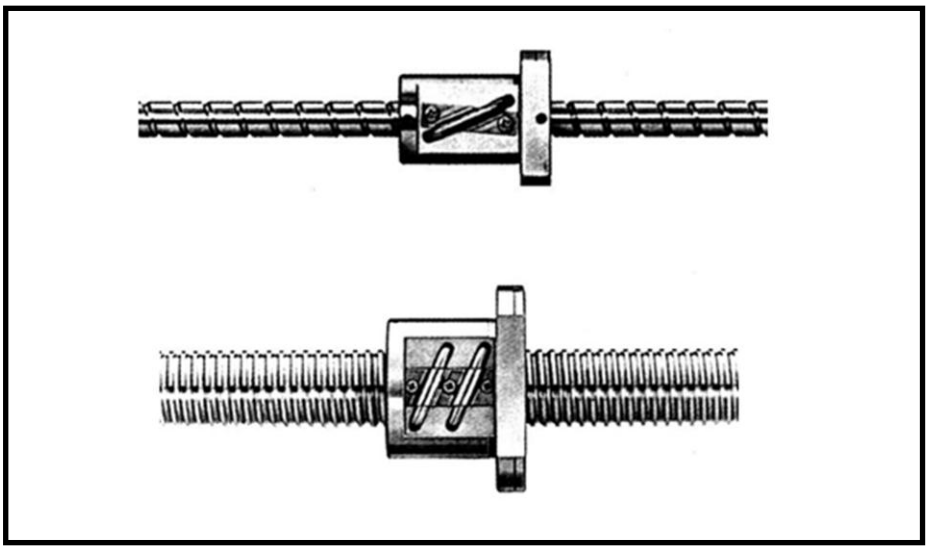
Fig. 8. Paso del tornillo



Fuente: (NSK Corporation, 2005)

Los tornillos de bolas se clasifican en dos categorías dependiendo de su paso, de paso largo la cual permite una mayor velocidad lineal y de paso fino, la cual permite una mayor precisión de posicionamiento.

Fig. 9. Tipos de tornillos dependiendo de su paso

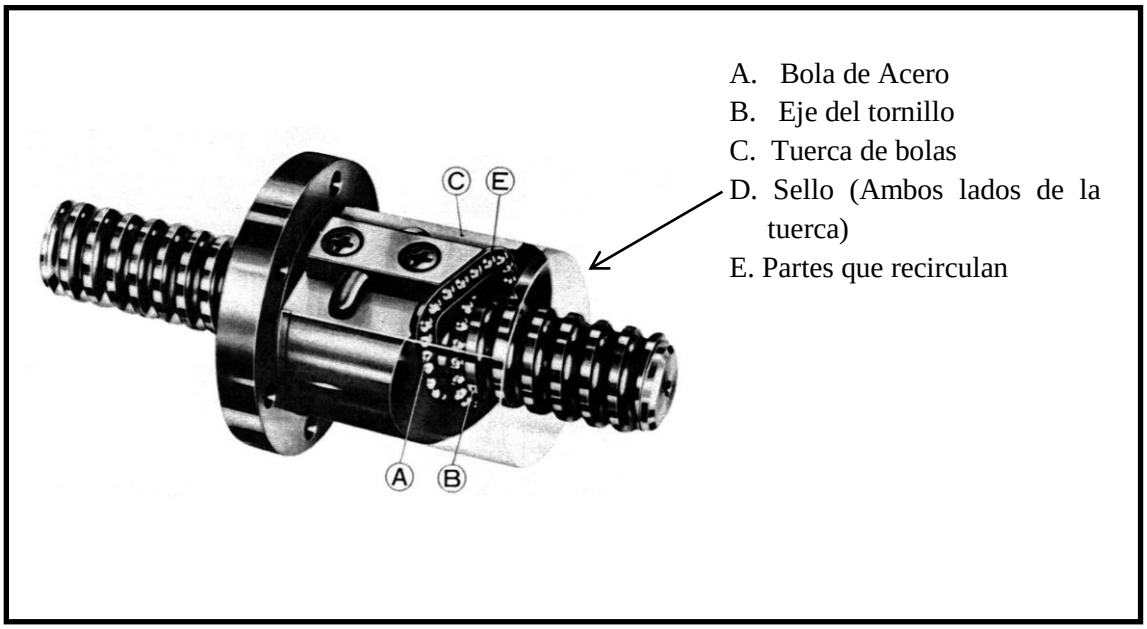


Fuente: (NSK Corporation, 2005)

La precisión de los tornillos se define como: “la exactitud de la distancia (con una precisión de viajes de tuerca) que la tuerca ha viajado cuando el eje de tornillo ha girado” Es completamente dependiente de la precisión en la manufactura de las ranuras para las bolas en su dirección de alimentación. Cierta aplicaciones requieren precisiones de hasta 3.5 (um). Por lo general para la industria aeroespacial.

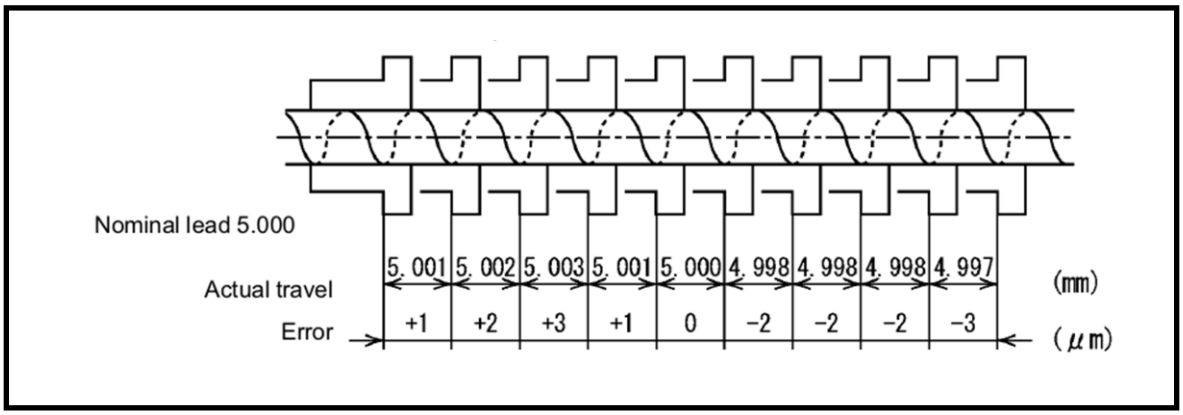
Un tornillo de bolas re circulantes cuenta con los siguientes elementos:

Fig. 10. Elementos de un tornillo de bolas



Fuente: (NSK Corporation, 2005)

Fig. 11. Error de paso en los tornillos de bolas



Fuente: (NSK Corporation, 2005)

Por ejemplo, suponiendo que un tornillo de bolas ha sido fabricado con la intención de paso de L=5.00 [mm], aun así podría ser fabricado con L=4.998 o L=5.005. Como tal, un error positivo o negativo es parte del tornillo.

Como esos errores de paso afectan directamente el sistema conductor en variación con la velocidad de alimentación imprecisión en la posición, existen reglas detalladas y criterios para la precisión de tornillos de bolas en estándares de relevancia industrial.

Tabla 4. Precisión en los tornillos de bolas

Categoría / Ítem	Serie de Posicionamiento					Serie de transportación	
Grado de Precisión	C0	C1	C2	C3	C5	C7	C10
V300	3.5um	5um	7um	8um	18um	52um	210um
Calidad	Alta precisión ←						

Fuente: (NSK Corporation, 2005)

V300: Esta es la variación más grande (variación de viaje) en errores de paso sobre cualquier intervalo de 300mm, dentro del largo de viaje efectivo. (NSK Corporation, 2005)

La precarga es crear deformaciones elásticas (deflexiones) en las bolas de acero y las ranuras de la tuerca y el eje del tornillo por adelantado al dotar de carga axial.

El propósito de esta deformación es eliminar el juego axial entre el eje del tornillo y la tuerca de bolas. Minimizar la deformación elástica causada por fuerzas externas.

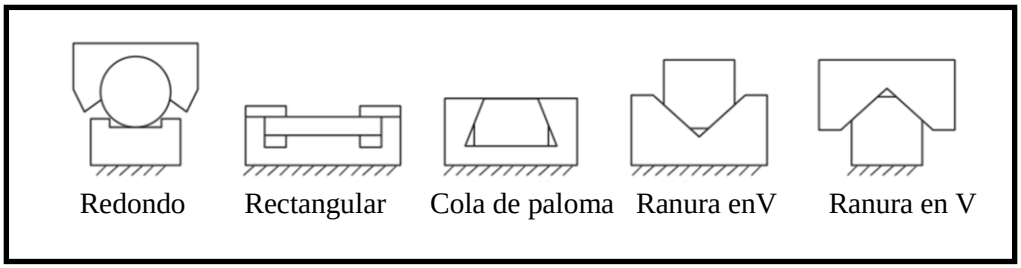
3.1.2.2. Rodamientos lineales

Existe una gran variedad de rodamientos lineales. Algunos ejemplos son bujes lineales, bujes lineales con bolas, rieles lineales, y ejes de bolas. Los rodamientos llanos son simples y de bajo costo, pero están limitados en velocidad y vida por la fricción entre los elementos deslizantes. Los rodamientos de bolas o rodillos reemplazan a los deslizantes con contactos rodantes para una mayor velocidad, carga y vida útil. Los rodillos tienden a tener una capacidad de carga más alta que los de bolas y son usados para aplicaciones de cargas elevadas. Para todos los rodamientos lineales, la vida útil es medida en desplazamiento lineal y es determinada por los materiales que lo componen, dureza, lubricación y condiciones de carga.

Lubricación.- Los rodamientos lineales de fricción se utilizan generalmente en un estado de lubricación mínima. Grasa u otros lubricantes espesos se aplican al eje y deben ser re aplicado periódicamente para reducir la fricción y prolongar la vida del rodamiento. Los contaminantes tienen fácil acceso a los rodamientos lineales de fricción, por lo que las medidas preventivas adecuadas o materiales suaves deben ser utilizados.

Los rodamientos lineales suelen incorporar vías de acceso a la caja de cojinetes para lubricación interna. Las conexiones de engrase están normalmente presentes en la caja del rodamiento para fines de lubricación. Estos rodamientos a menudo tienen sellos de eje para reducir la pérdida de lubricante y prevenir la entrada de contaminantes. Los rodamientos lineales deben ser lubricados periódicamente, ya que la pérdida de lubricante es de esperar.

Fig. 12. Tipos de rodamiento lineal.



Fuente: (Marrs, 2011)

Rodamientos lineales planos.-Los rodamientos lineales planos son esencialmente lo mismo que los rodamientos rotacionales planos en diseño, materiales, y clasificación de presión estática. Los rodamientos lineales planos pueden tener una variedad de formas incluyendo las mangas redondas, placas planas, y otras formas. En muchos casos, una disposición de cojinete liso lineal puede ser la de un deslizador y guía de deslizamiento; esta disposición permite el movimiento a lo largo de un solo eje. No es raro tanto para el deslizador y la guía de ser hechas de acero para herramientas endurecido para una larga vida y recubierto con lubricantes de película seca, así como engrasado a fondo. Las cargas pesadas pueden ser llevadas en algunas configuraciones, y la alta precisión se logrará siempre que el desgaste se mantenga a un mínimo. Deslizadores de alta resistencia y guías de deslizamiento se pueden encontrar en

centros de mecanizado, y versiones de poca potencia se encuentran comúnmente en el montaje de máquinas y accesorios. Muchas versiones están disponibles comercialmente, pero también se diseñan fácilmente “en casa” si es necesario. A continuación se muestran algunas configuraciones comunes:

Rodamientos lineales con componentes rodantes.- Cuando se especifica rodamientos lineales, los principales criterios son precisión, juego interno, capacidad de carga estática, capacidad de carga dinámica, la velocidad y la esperanza de vida. Se debe usar la carga equivalente en el cálculo de la carga dinámica, y los catálogos del fabricante deben ser consultados al hacerlo. La vida de los rodamientos lineales a menudo se da en distancia lineal recorrida. El estándar ISO, que es comúnmente utilizado por la mayoría de los fabricantes, es una carga de la esperanza de vida dinámica de 50 km. Las fórmulas de la siguiente tabla se pueden utilizar para calcular de carga dinámica para rodamientos lineales. La distancia lineal recorrida puede ayudar a dar una idea del tiempo de vida útil de los rodamientos.

Tabla 5. Ecuaciones básicas para un rodamiento lineal

<table>
<tr><td colspan="2">L = esperanza de vida en distancia lineal Rodamiento de bolas: k = 3
F = carga equivalente en el rodamiento Rodamiento de rodillos k = 3.33</td></tr>
<tr><td>Carga estática básica

$C_0 = carga\ estática\ básica\ requerida$
$SF_0 = Factor\ de\ seguridad$
$F_{MAX} = Carga\ máxima\ equivalente$</td><td>$C_0 = (SF_0)F_{MAX}$</td></tr>
<tr><td>Vida Útil vs. Carga</td><td>$\frac{L_2}{L_1} = \left(\frac{F_2}{F_1}\right)^k$</td></tr>
<tr><td>Esperanza de vida
$h = horas\ de\ funcionamiento$
$S = longitud\ del\ recorrido$
$n = número\ de\ recorridos\ por\ minuto$
Para cargas variables
$R_1 = Distancia\ bajo\ carga\ L_1$
$L_1 = Vida\ (distancia) bajo\ L_1$
$R_T = Distancia\ total\ esperada\ para\ el\ rodamiento$</td><td>Conversión de tiempo de vida a horas.
$L = 60h(2Sn)$

Cargas variables
$L = \frac{1}{\frac{R_1}{R_T L_1} + \frac{R_2}{R_T L_2} + \dots}$</td></tr>
</table>

Carga dinámica básica	Cargas dinámica usando la confiabilidad nominal:
$L_{nominal} = 50km\ \ típico$ $K_s = factor\ de\ servicio\ por\ choque$ Los siguientes valores pueden ser aplicados si la clasificación de la carga se basa en una eficiencia del 90%: $K_R = 0.62\ para\ confiabilidad\ del\ 95\%$ $K_R = 0.33\ para\ confiabilidad\ del\ 98\%$ $K_R = 0.21\ para\ confiabilidad\ del\ 99\%$	$C = FK_s \sqrt[k]{\frac{L}{L_{nominal}}}$ $(Si\ L = L_{nominal},\quad C = FK_s)$ Carga dinámica usando el ajuste de confiabilidad: $C = FK_s \sqrt[k]{\frac{L}{K_R L_{nominal}}}$

Fuente: (Marrs, 2011)

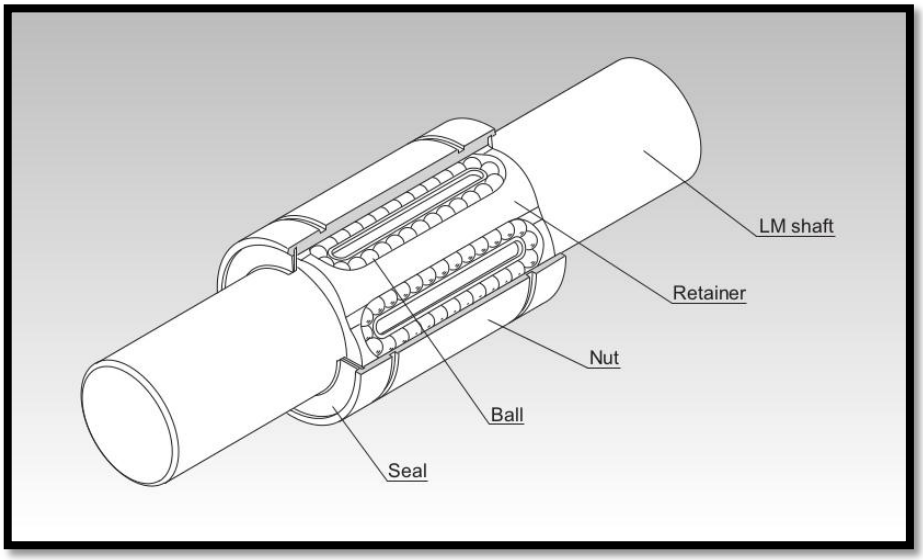
Hay muchos tipos de rodamientos lineales de elementos de rodadura disponibles comercialmente. Los siguientes son algunos tipos comunes y sus descripciones.

Fig. 13. Tipos de rodamientos lineales de bolas

Fuente: (Marrs, 2011)

Cojinetes de bolas.- Estos rodamientos son de uso general y consisten en un cojinete que contiene los rodamientos de bolas que se desplazan sobre un eje liso. Generalmente su propósito es para condiciones de carga ligera. Algunos diseños permiten la rotación del rodamiento en el eje, mientras la mayoría no son diseñados para rotar. En general, debe evitarse torque rotacional. Usualmente los rodamientos lineales son usados en pares, corriendo sobre ejes paralelos. Las bolas dentro de los rodamientos comúnmente re circulan para proveer un continuo contacto con el eje. Los cojinetes de bolas pueden conseguirse tanto en configuraciones abiertas como cerradas, con y sin tapas que permiten conservar la lubricación. Se puede conseguir cojinetes de bolas con pre carga. Los cojinetes de bola tienen una capacidad limitada para momentos de carga (alrededor de un eje perpendicular al eje de movimiento).

Fig. 14. Partes de un rodamiento lineal

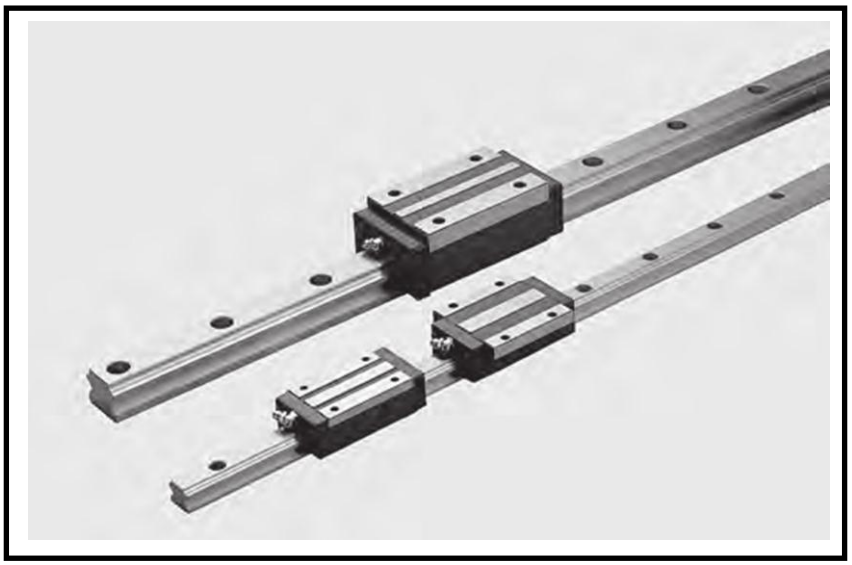


Fuente: (THK, 2010)

Rodamientos ranurados.- Los rodamientos lineales ranurados son recomendados cuando el ensamblaje debe resistir la rotación del rodamiento en el eje durante una carga torsional. El anclaje de las bolas con el eje previene la rotación. Los rodamientos ranurados usualmente son calificados de acuerdo a su capacidad de capacidad de torque, y generalmente se usan en diseños que permiten transmitir el torque mientras permiten el movimiento lineal.

Rieles lineales.- Las rieles lineales son una solución compacta al problema de prevenir la rotación mientras se permite el movimiento lineal. Estas configuraciones normalmente consisten de un riel rectangular y un rodamiento tipo patín que se mueve sobre la riel. En general, el patín está enganchado al riel y puede soportar cargas de todas las direcciones. Este diseño pueden generalmente soportar momentos de carga, el uso de dos guías en una riel, o dos rieles lado a lado deben ser considerados. Comercialmente existen dos versiones, con bolas y con rodillos.

Fig. 15. Rodamiento de riel.



Fuente: (NSK Corporation, 2005)

Guías de rodillos cruzados.- Estos rodamientos son altamente compactos y rígidos, los cuales consisten en dos filas de rodillos en ángulo entre dos carriles lineales. Son capaces de transportar cargas muy pesadas. Estos ensamblajes pueden soportar cargas de momentos significativas y son generalmente usados como soportes para mesas en máquinas y herramientas. Los rodillos no son re circulantes, por lo que el movimiento es generalmente más suave que el de cojinetes con bolas re circulantes. Debido a su configuración, las guías de rodillo cruzado deben ser el doble de largo del recorrido esperado. (Marrs, 2011)

La selección y dimensionamiento de los elementos lineales de rodadura se realiza de una manera similar a aquella de rodamientos rotacionales. (Marrs, 2011)

Fig. 16. Rodamiento lineal de rodillo cruzado


Muy altas cargas


Rodillo de aguja


Rodillo curvado

Fuente: (Marrs, 2011)

3.1.3. DESLIZADOR

El deslizador lineal está destinado a ser una herramienta para el laboratorio de mecatrónica, por lo cual no tendrá usos en exteriores ni capacidades industriales. Para el diseño y dimensionamiento se han tomado las siguientes consideraciones

- La carga será estática sobre la mesa del trabajo del deslizador, esto quiere decir que no existirán momentos de torsión rotacional.

- Las guías laterales se deben dimensionar de acuerdo a diámetro interno de los rodamientos lineales, los cuales a su vez dependen de la carga estática y dinámica a la que se encuentren sometidos.

- La carga dinámica será soportada mediante los rodamientos lineales en sus guías laterales, por tal motivo para el dimensionamiento del tornillo de bolas se deben considerar el torque transmitido y la longitud del recorrido.

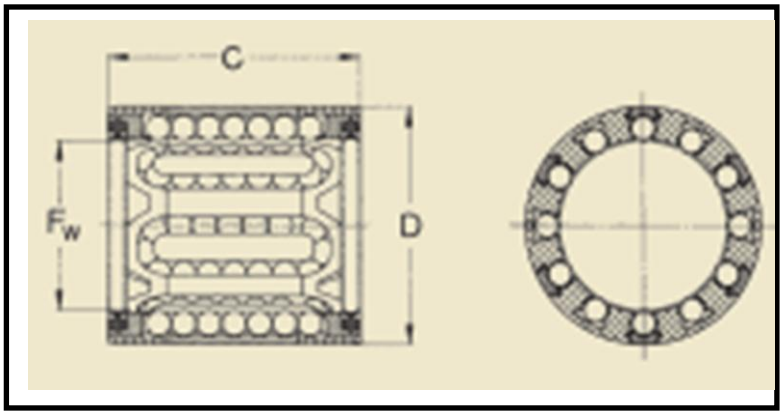
- La estructura del deslizador debe ser lo suficientemente robusta como para soportar el manejo y portabilidad del módulo dentro y fuera del laboratorio de mecatrónica.

Tomando en cuenta estas consideraciones se han escogido los siguientes elementos:

3.1.3.1. Rodamiento Lineal

El rodamiento lineal escogido es el SKF LBBR 20. Es un rodamiento que puede ser conseguido en el mercado local, en caso de que sea neceario su reemplazo por desgaste mecánico. Este rodamiento tiene las siguientes características:

Fig. 17. Rodamiento LBBR 20



Fuente: (Marrs, 2011)

Tabla 6.Características del rodamiento LBBR 20

Dimensiones			N° de filas de bolas	Cargas básicas		Masa	Código
Fw	D	C		Dinámica	Estática		
mm	mm	mm	-	N	N	Kg	
20	28	30	6	1160	800	0,021	LBBR20

Fuente: (SKF Group, 2014)

Este rodamiento se seleccionó debido a que cumple con los requerimientos mecánicos del sistema y se puede encontrar en el mercado local.

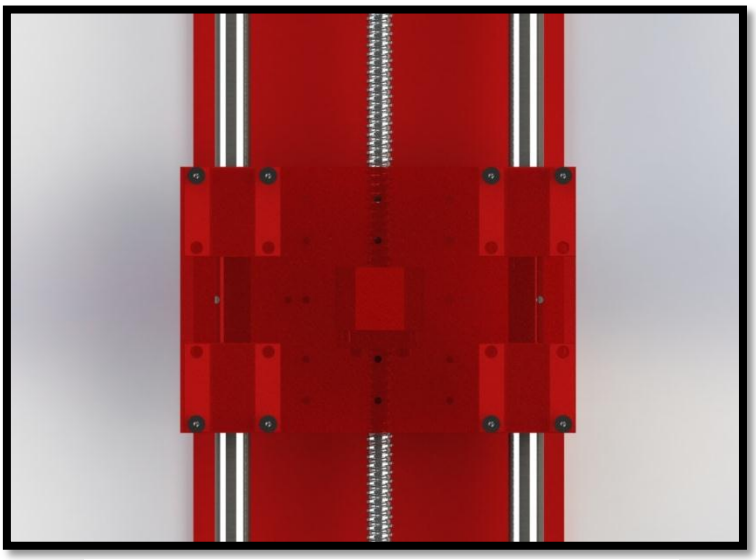
Para poder determinar la carga que pude soportar el deslizador debemos transformar la capacidad de carga de kilogramos a Newtons, esto se logra con la siguiente relación.

Ecuación 1. Conversión de kilogramos a Newtons

$$1\ (Kg) = 9.8\ (N)$$
$$30\ (Kg) = 294\ (N)$$

Se utilizan cuatro cojinetes de rodamientos en la siguiente configuración:

Fig. 18. Distribución de los rodamientos lineales



Fuente: Autor, Software: SolidWorks 2013

Se debe anticipar la posibilidad de que la carga no se encuentre uniformemente distribuida sobre los cuatro rodamientos, además se considera un factor de seguridad de dos. Tomando en cuenta estos datos se puede calcular la carga estática básica soportada en los rodamientos de la siguiente manera:

Para el cálculo de la vida útil del rodamiento lineal tenemos la siguiente fórmula:

Ecuación 2. Cálculo de vida útil en metros

$$L_{10} = \frac{l_s}{l_t}\left(\frac{C}{P}\right)^p$$
$$L_{10} = \frac{0.7}{1}\left(\frac{800}{588}\right)^3$$
$$L_{10} = 1.76 \cdot 10^5 (m)$$

Fuente: (SKF Group, 2014)

Ecuación 3. Carga soportada por el rodamiento

$$C_0 = (SF_0)F_{MAX}$$
$$C_0 = (2) * (294)\ [N]$$
$$C_0 = 588\ [N]$$

Fuente: (Marrs, 2011)

El recorrido total de vida útil esperado para los rodamientos es de 176 Km de recorrido.

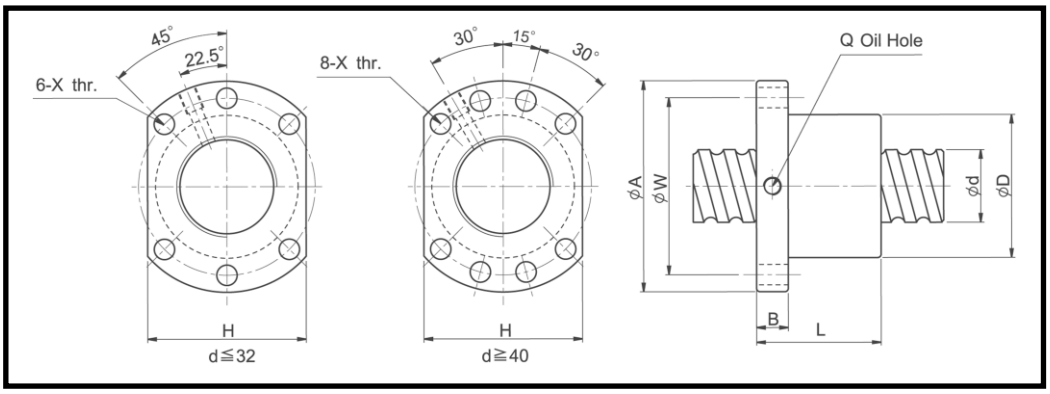
3.1.3.2. Tornillo de bolas

Para esta aplicación se escogió el tornillo de bolas re circulantes:

BSC-AAM-SFUR-016-05-G-C7-P0-1050.

El cual cuenta con las siguientes características:

Fig. 19. Características del tornillo de bolas escogido



Fuente: Anaheim Automatation (2012)

Tabla 7. Características del tornillo de bolas escogido

l:Paso, Da: diám de bola, n: Núm de circuitos, K: Rigidez (kgf) H: Hélica Ca: Carga dinámica básica (kgf) Coa: Carga estática básica (kgf)															
d	l	Da	D	A	B	L	W	X	H	Q	n	Ca	Coa	K	H
16	5	3.18	28	48	10	50	38	5.5	40	M6	1x4	1380	3052	32	R

Fuente: Anaheim Automatation (2012)

La carga axial soportada por este rodamiento es mucho mayor que la carga especificada para el deslizador, la carga útil descansa sobre los rodamientos lineales, cualquier carga normal al eje puede ser soportada por el mismo.

Tabla 8. Características del tornillo de bolas utilizado

Diámetro:	16 (mm)
Longitud:	1000 (mm)
Paso:	0,2 (rev/mm)
Eficiencia	90% (Aerotech, 2014)
Coeficiente de fricción:	0,05 (Aerotech, 2014)
Carga:	30 (Kg)
Vel. Max. (angular):	3000 (rpm)
Tiempo aceleración:	0.1 (seg)

Fuente: (Aerotech, 2014)

Cálculo de la aceleración angular:

Ecuación 4 Aceleración angular

$$\alpha = \frac{4.5\,\theta}{t^2}$$

Dónde.

α = Aceleración angular

θ = Distancia en radianes

t = tiempo total del movimiento

Fuente: (Aerotech, 2014)

$$\alpha = \frac{4.5\,(25.13)}{0.1^2} = 11310\ \frac{rad}{seg^2}$$

Inercial en el eje:

Ecuación 5. Inercia en el tornillo de bolas

$$J_{eje} = (7.57x10^{-13})D^4L\ \ kg - m^2$$

Dónde.

J = Inercia en el eje

D = Diámetro del eje

L = Longitud del eje

Fuente: (Aerotech, 2014)

$$J_{eje} = (7.57x10^{-13})(16)^4(1000)\ \ (kg.m^2)$$

$$J_{eje} = 4.96x10^{-5}\ (kg.m^2)$$

Inercia en la carga:

Ecuación 6. Inercia en el tornillo de bolas

$$J_{eje} = (2.25x10^{-8})\frac{m_{carga}}{P^2}\ \ kg - m^2$$

Dónde.

J_{carga} = Inercia de la carga reflejada en kg.m2

m_{carga} = Carga útil en kg a:

P = paso del eje expresado en rev/mm

Fuente: (Aerotech, 2014)

$$J_{eje} = (2.25x10^{-8})\frac{30}{0.2^2} \ (kg.m^2)$$

$$J_{eje} = 1.6875\ x\ 10^{-5} \quad (kg.m^2)$$

La inercia total el sistema es igual a:

$$J_{TOTAL} = (1.6875 + 4.96)x10^{-5} \ (kg.m^2)$$

$$J_{TOTAL} = \ 6{,}6475 \ (kg.m^2)$$

El torque para la aceleración se calcula con la siguiente fórmula:

Ecuación 7. Torque del motor.

$$T_\alpha = \left(J_{motor} + \frac{J_{eje}}{e} + \frac{J_{carga}}{e}\right)\alpha \quad N.m$$

Dónde:

e = Eficiencia

J_{motor} = Inercia del motor (típico 0.5)

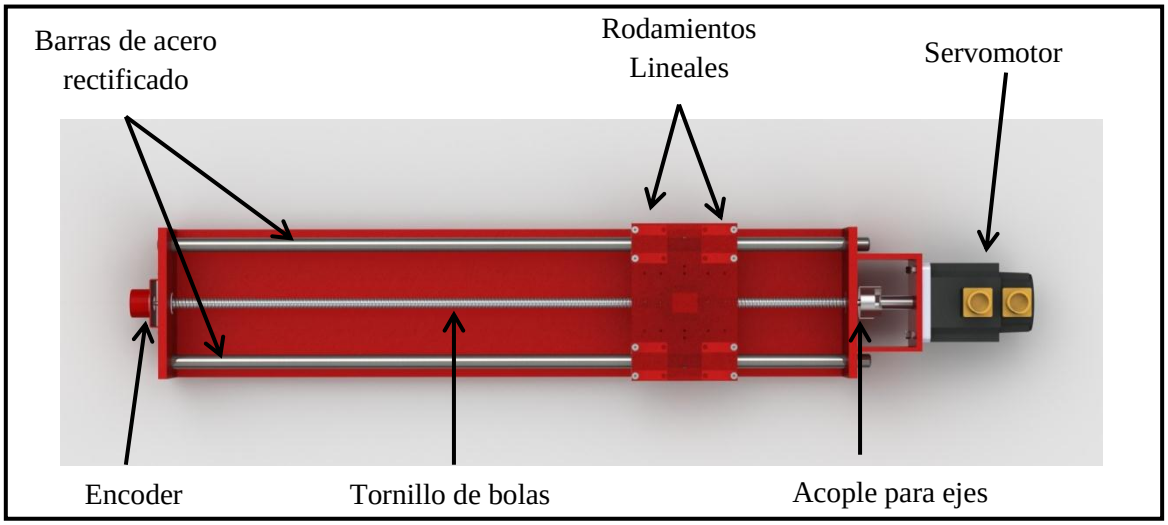
Fuente: (Aerotech, 2014)

$$T_\alpha = \left(0.5 + \frac{6.6475}{0.9}\right)x10^{-5}\ (11310) \ (N.m)$$

$$T_\alpha = 0{,}89 \ (N.m)$$

El torque calculado es menor al torque que el motor es capaz de entregar, el cual es de dos newton por metro. Por lo tanto el deslizador es capaz de soportar todas las cargas y fuerzas cumpliendo con las especificaciones de su diseño.

Fig. 20. Vista superior del deslizador lineal.



3.2. CONTROLADOR MICROPROCESADO

3.2.1. DIAGRAMA DE CONTROL

La forma en que se realiza el control sobre el sistema es a través de la programación en LabVIEW. En éste deben ser creados los lazos de control o monitoreo del sistema, y se realizan las llamadas a hardware utilizando el API para el deslizador.

De esta manera el microcontrolador se vuelve el puente por el cual pasan todas las comunicaciones, este ejecuta los comandos que le son enviados y envía la información del encoder y alarmas a petición del usuario, que es quién crea el control en su programación.

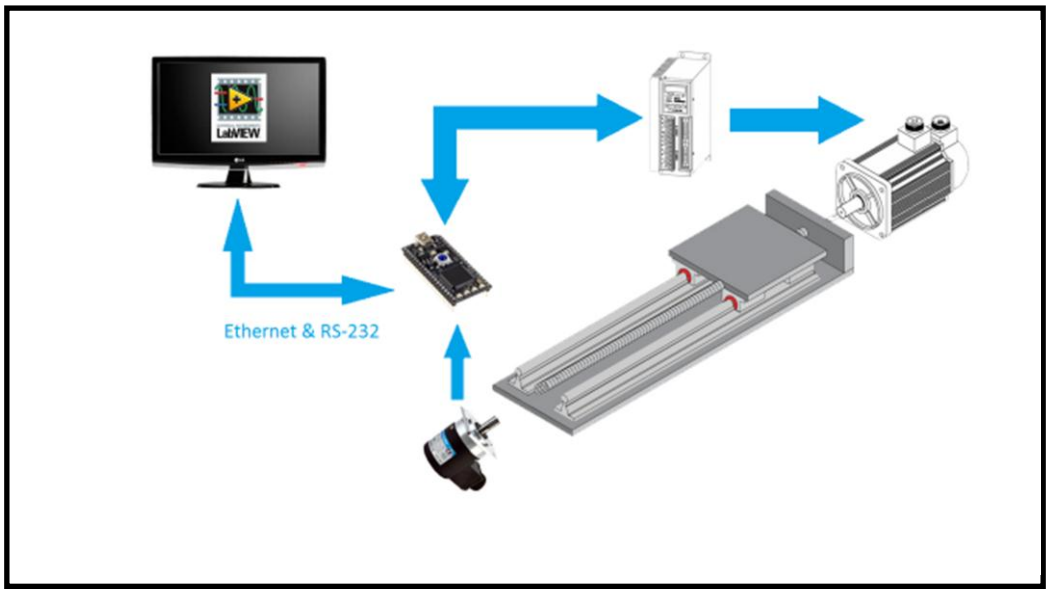
Sin embargo se deben tomar medidas de protección con el deslizador para evitar que una mala programación del usuario dañe el equipo. Para esto se han conectado cuatro sensores ópticos, dos a cada extremo del sensor, los cuales sirven para evitar una colisión de la base del deslizador.

Son dos los sensores que se utiliza en cada extremo ya que el primero envía una señal de alarma al programa del usuario para que se toman las medidas pertinentes a fin de evitar la colisión, si no se han tomado medidas correctivas y se llega al accionamiento del segundo sensor, entonces el microcontrolador detendrá el servomotor automáticamente, sin esperar el envió de un comando y de esta manera evitará que el deslizador sufra daños.

El microcontrolador también monitorea permanentemente la señal de alarma proveniente del servomotor, en el caso de llegar a producirse un error en el servodriver este detendrá el motor y accionará la señal de alarma, la cual será transmitida al programa principal.

El microcontrolador siempre envía una trama de respuesta cuando se le envía un comando, esta trama de respuesta contiene información acerca del estado actual del deslizador, esto se detalla en el API del sistema.

Fig. 21. Esquema de las conexiones para el deslizador.



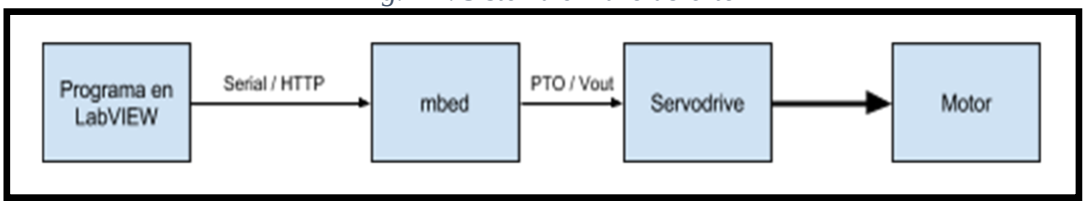
Fuente: Autor

El elemento de retroalimentación es el encoder, este puede enviar información de posición al programa principal para que actúe acorde a su programación. Al utilizar la información proporcionada por el encoder se logra un lazo de control cerrado.

Este equipo se diseño para ser modular, lo que quiere decir que lazo abierto o lazo cerrado depende del usuario en su programación. El sistema puede ser configurado con diversos elementos de entrada para su control y utilizar el encoder para monitorear el error de posición en caso de requerirse.

La configuración del sistema en lazo abierto es la siguiente:

Fig. 22. Sistema en lazo abierto

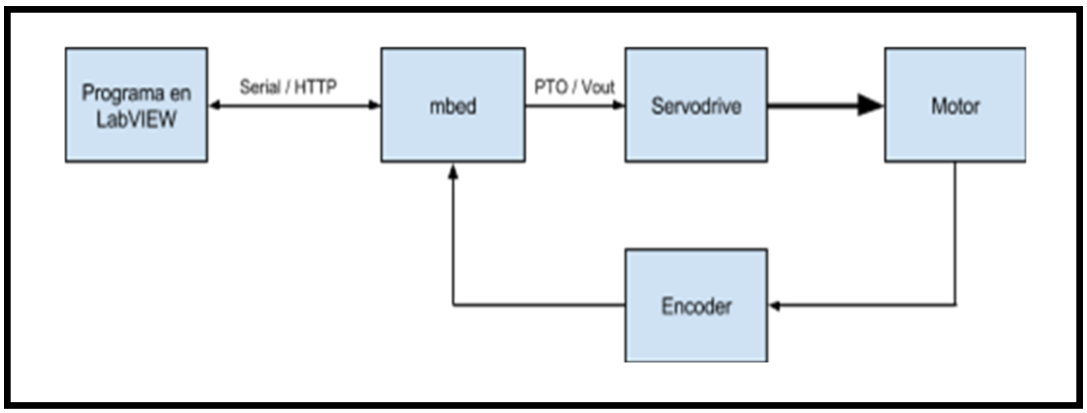


Fuente: Autor

En lazo abierto los comandos enviados al microcontrolador son ejecutados y aplicados al servomotor, el cual informará solamente las alarmas del servodriver y del deslizador.

La configuración del sistema en lazo cerrado es la siguiente:

Fig. 23. Sistema en lazo cerrado



Fuente: Autor

En el lazo de control cerrado se utiliza el encoder del deslizador para obtener la posición angular relativa del servomotor. El encoder utilizado es un encoder de cuadratura el cual se encuentra conectado al tornillo del deslizador, por tanto se puede conocer la posición y el sentido de rotación del servomotor.

3.2.2. ESQUEMA DEL CIRCUITO

El circuito que rodea al microcontrolador le sirve para ajustar los niveles de voltaje de sus entradas y salidas a las entradas y salidas del servodriver y del estándar Ethernet y Serial.

Se han implementado tres placas de circuitos, una principal que es donde se encuentra el microcontrolador, una placa de conexión con el servodriver y una placa de conexión con el encoder y las alarmas del deslizador.

La placa de conexión con el servodriver no contiene ningún circuito integrado, su función es la de dotar a la placa principal de conexión física con el servodriver a través de borneras tipo banana, permitiendo mantener la compatibilidad del conector, el cual utilizaba las borneras para conectarse al PLC de control.

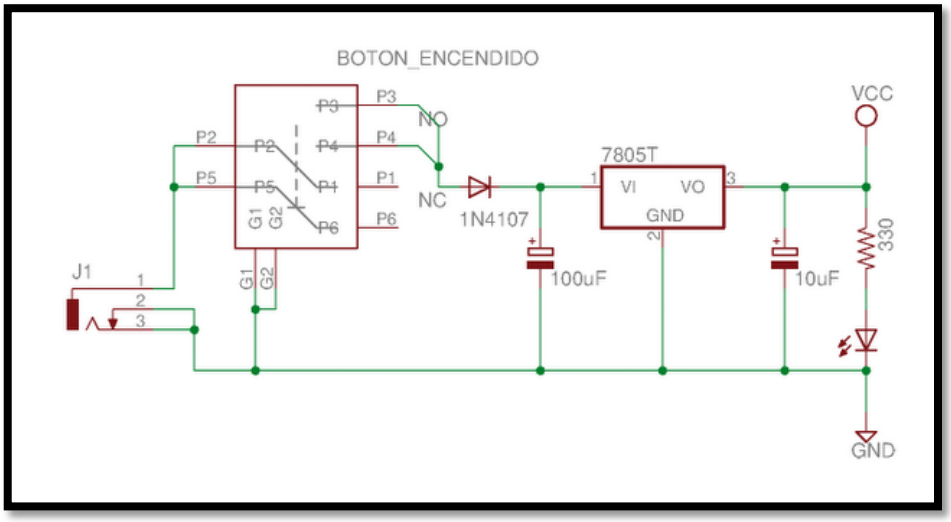
La placa de conexión con el encoder y alarmas del deslizador realiza la adaptación de los voltajes del encoder a los voltajes del microcontrolador. Además cuenta con compuertas lógicas que permiten simplificar y adecuar la señal proveniente de los opto acopladores para ser leídos por el microcontrolador.

3.2.2.1. Fuentes de alimentación

Es necesario contar con dos fuentes de alimentación para alimentar el subsistema electrónico. Esto se debe a que existen dos secciones en la tarjeta principal, se tiene una sección digital y una sección analógica.

La sección digital de la tarjeta se refiere a la alimentación del microcontrolador y de los circuitos de la placa de encoder y alarmas, todos estos circuitos trabajan a una tensión de 5Vdc, para alimentar estos circuitos se necesita una entrada de voltaje de 9Vdc la cual es regulada a 5Vdc mediante el siguiente circuito:

Fig. 24. Esquemático del circuito de alimentación, placa principal



Fuente: Autor, Software: Eagle 6.0

El voltaje de salida ofrece 5Vdc con una corriente máxima de 1000 mA. En los esquemas siguientes el terminal VCC se refiere a los 5Vdc con referencia a masa o GND. Se tiene además un botón para encender o apagar la placa del controlador, y un diodo de protección en caso de que la polaridad de la fuente se conecte de manera incorrecta. El conector de entrada se lo implementó con un conector de barril de 1/8 de pulgada muy común en adaptadores de corriente AC/DC.

La sección analógica de la placa se refiere a la generación y adecuación de los niveles de voltaje necesarios para que el servomotor funcione en el modo de control de velocidad, debido a que el rango de tensión es de 20Vdc, se utilizó una alimentación de 24Vdc, esta alimentación se encuentra ya instalada en el módulo

de servomecanismos con que cuenta actualmente el laboratorio de mecatrónica, lo cual facilita su conexión.

Para evitar que la polaridad de la fuente de voltaje de 24Vdc sea conectada de manera incorrecta se conectó a un puente de diodos, el cual servirá para asegurarse que la polaridad siempre sea la correcta.

La fuente de voltaje de 24Vdc sirve para alimentar el circuito analógico y también alimenta el encoder de cuadratura.

3.2.2.2. Encoder de cuadratura

El sistema hace uso de un encoder de cuadratura marca Hohner modelo: 21-231-360. Este elemento se tomó del laboratorio de mecatrónica. El cual cuenta con las siguientes características:

Fig. 25. Encoder de cuadratura Hohner

Fuente: (Hohner Corporation, 2014)

Tabla 9. Características del encoder Hohner

Marca:	Hohner
Modelo:	21-231-260
Resolución	360 ppr [pulsos por revolución]
Fases	A,B,Z
Alimentación	11v…30v
Protección	IP65

Características del encoder Hohner	
Temperatura de trabajo	-20°C … 60°C
Consumo de corriente	80mA max.
Eje	Eje de acero inoxidable
Cubierta	Aluminio
Velocidad	3000 RPM máx

Fuente: (Hohner Corporation, 2014)

En el circuito, este encoder es alimentado por 24Vdc, por lo tanto su voltaje de salida debe ser adaptada a las entradas del microcontrolador. Esto se logra a través de un divisor de tensión. El circuito es el siguiente:

“Los encoder de cuadratura entregan dos señales tipo tren de pulsos A y B por cada uno de sus dos canales desfasadas 90º. Cuando el encoder gira hacia una dirección el canal A adelanta al B, y cuando gira en dirección contraria el canal B adelanta al A.” (Hihglights, 2011)

Estas señales son enviadas al decodificador de cuadratura con el que cuenta el microcontrolador, este decodificador se encuentra implementado mediante hardware, lo que significa que el microcontrolador puede llevar el registro de la posición angular del eje sin que esto implique tiempo de procesamiento en el procesador central. Esta es una característica muy útil de este microprocesador, la cual permite simplificar la programación e implementación de esta funcionalidad.

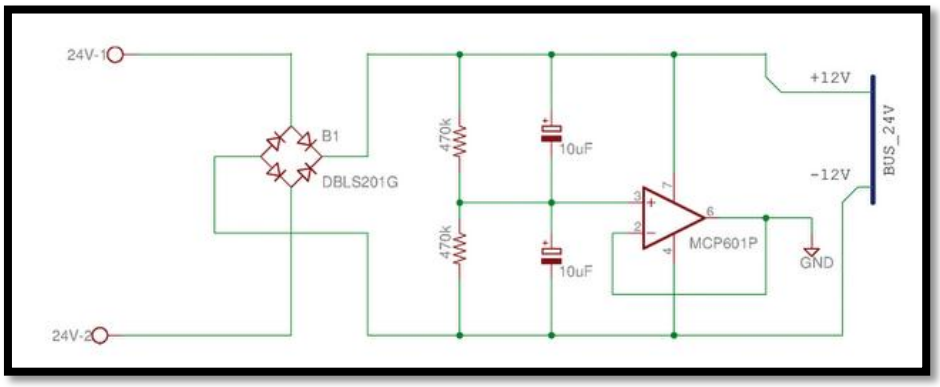
3.2.2.3. Salida analógica

Para alimentar al amplificador instrumental tenemos un circuito que permite obtener +/- 12.0v a partir de 24.0v de entrada aplicados a los terminales dispuestos para este propósito.

Este circuito cuenta con un divisor de tensión compuesto por dos resistencias de valores iguales, también se ha dispuesto un amplificador operacional configurado como seguidor (buffer) en el punto que será la masa virtual del amplificador. Luego esta masa virtual se conecta con la masa del circuito de alimentación del microcontrolador, con lo que se logra una fuente simétrica para

alimentar al amplificador instrumental y lograr una salida con voltajes positivos y negativos para el driver del servomotor. Una de las limitaciones de este circuito es que sólo puede entregar cantidades pequeñas de corriente, eso no es problema a que el amplificador no consume demasiada corriente, pero no es posible alimentar el resto del circuito directamente de este circuito.

Fig. 26. Esquemático de fuente dividida para alimentación del amplificador instrumental



Fuente: Autor (Software: Eagle 6.0)

Para evitar una mala conexión el circuito cuenta con un puente de diodos, de esta manera la polaridad del voltaje aplicado en las borneras es indiferente, al amplificador siempre llegará la polaridad correcta.

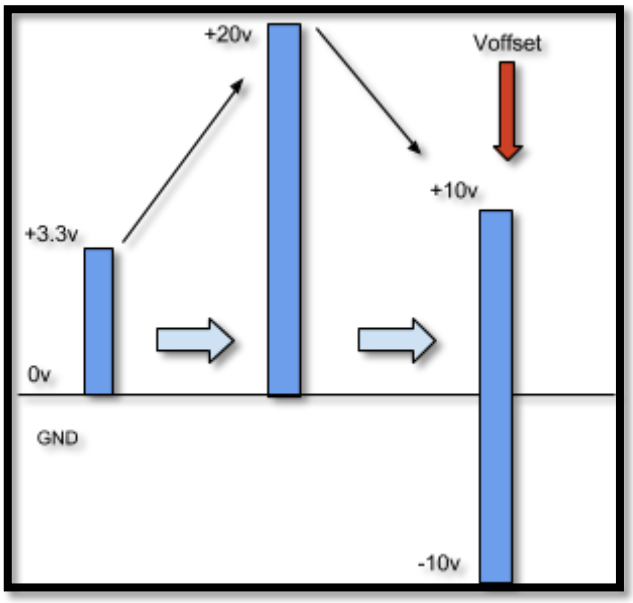
El microcontrolador cuenta con una conversor DAC de 10 bits de resolución. Las librerías necesarias para el manejo de este recurso están implementadas como una variable de tipo flotante entre cero y uno (0.0 - 1.0), es decir, al valor de uno le corresponde una salida de 3.3V, a un valor de 0.5 le corresponde una salida de 1.65V en una relación lineal. Cuando se envía el comando de un voltaje de salida de cero voltios, el microcontrolador tendrá una salida de 1.65V

El Driver del servomotor, por otro lado, tiene una entrada analógica para el control de velocidad desde -10V a +10V, dependiendo del sentido de giro. Para poder acoplar la salida del microcontrolador con la entrada del driver se utiliza el amplificador instrumental AD620N.

Se configura el amplificador con una ganancia ‘G = 6.06’ aproximadamente, luego se le da un offset de -10V para poder obtener la variación de -10V+10V a

partir de los 0V+3V del microcontrolador, est, al tener una precisión de 10 bits significa que tiene una precisión de salida luego de la amplificación de 4.8mV.

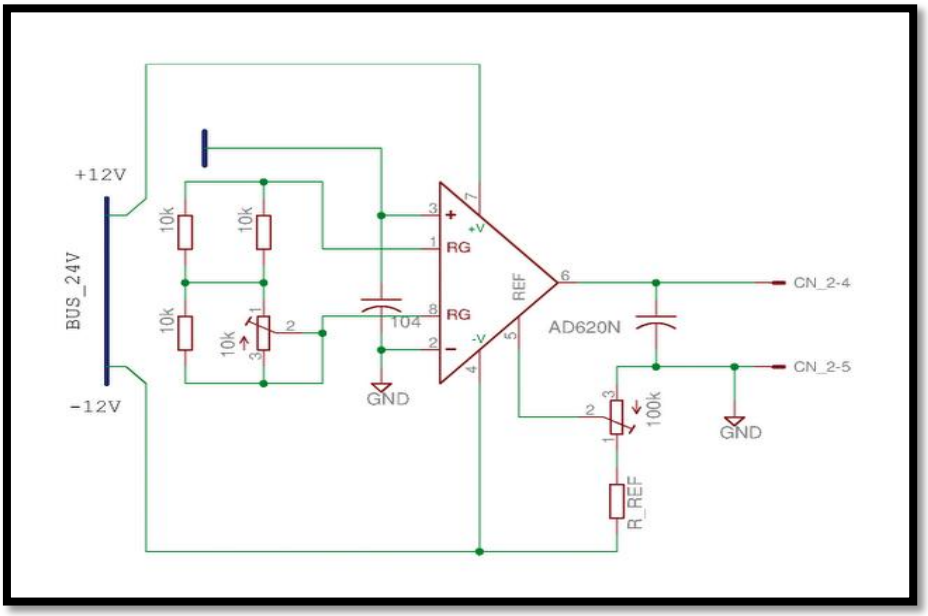
Fig. 27. Niveles de voltaje para la salida analógica del microcontrolador



Fuente: Autor

El siguiente circuito muestra la implementación de la amplificación a la salida analógica del microcontrolador mbed (pin_18) para controlar al servomotor en modo control de velocidad.

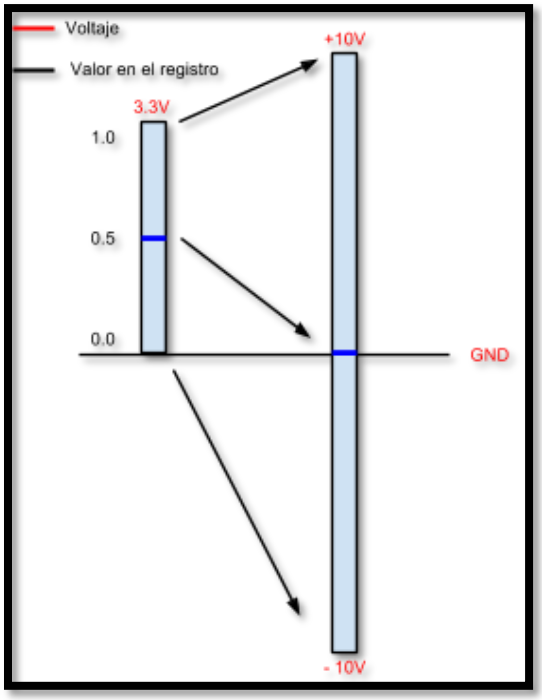
Fig. 28. Esquemático del circuito de amplificación para la salida analógica



Fuente: Autor (Software: Eagle 6.0)

El potenciómetro se encarga de regular la referencia del amplificador de manera que cuando el valor del microcontrolador se encuentra a la mitad (0.5 - 1.65V) la salida al driver sea de cero voltios. De esta manera cualquier valor para la salida analógica mayor a cero punto cinco se dará un valor positivo hasta los diez voltios, y valores menores tendrán valores de voltaje negativos hasta menos diez voltios. Por lo tanto se tiene la siguiente relación:

Fig. 29. Niveles de voltaje resultantes



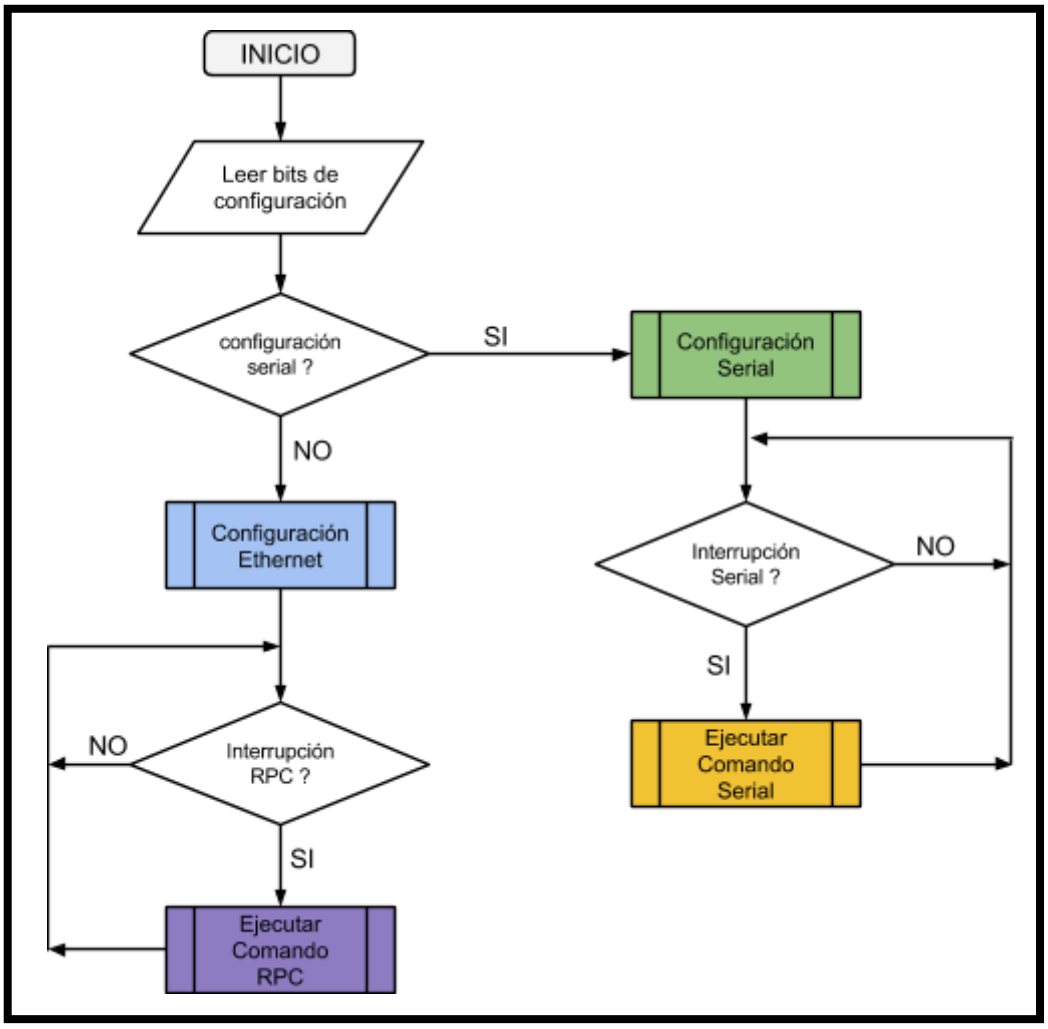
Fuente: Autor

3.2.3. PROGRAMACIÓN DEL MICROCONTROLADOR

La programación del microcontrolador se la realizó para que implemente una máquina de estados, estos estados son controlados por el usuario mediante el switch de configuración en la placa del microcontrolador. Estos estados configuran el mecanismo de transporte con el que se enviará los comandos al microcontrolador, estos deben ir acorde a la programación que se elige para el microcontrolador

Al inicio del programa el microcontrolador lee los bits del Switch de configuración y actúa acorde a como han sido configurados

Fig. 30. Lazo principal en la programación del microcontrolador

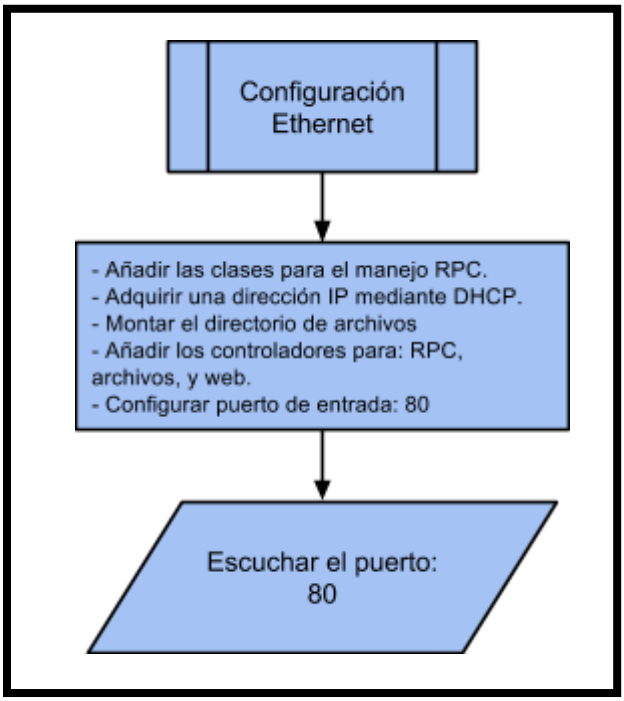


Fuente: Autor

Existen dos grandes divisiones: Configuración Ethernet y Configuración Serial. Se puede ejecutar solamente una a la vez, si se desea cambiar el modo de comunicación se debe cambiar la posición del bit uno del switch de configuración y luego resetear el microcontrolador. Estos pasos se describen en el manual de operación.

Si se elige la configuración ethernet el microcontrolador procede a inicializar las librerías y los objetos para recibir órdenes mediante el protocolo RPC. Si por el contrario se elige la comunicación serial el microcontrolador configura el puerto de salida y velocidad de transmisión especificada en los bits de configuración de la placa principal del microcontrolador.

Fig. 31. Subrutina de Configuración para la comunicación vía Ethernet



Fuente: Autor

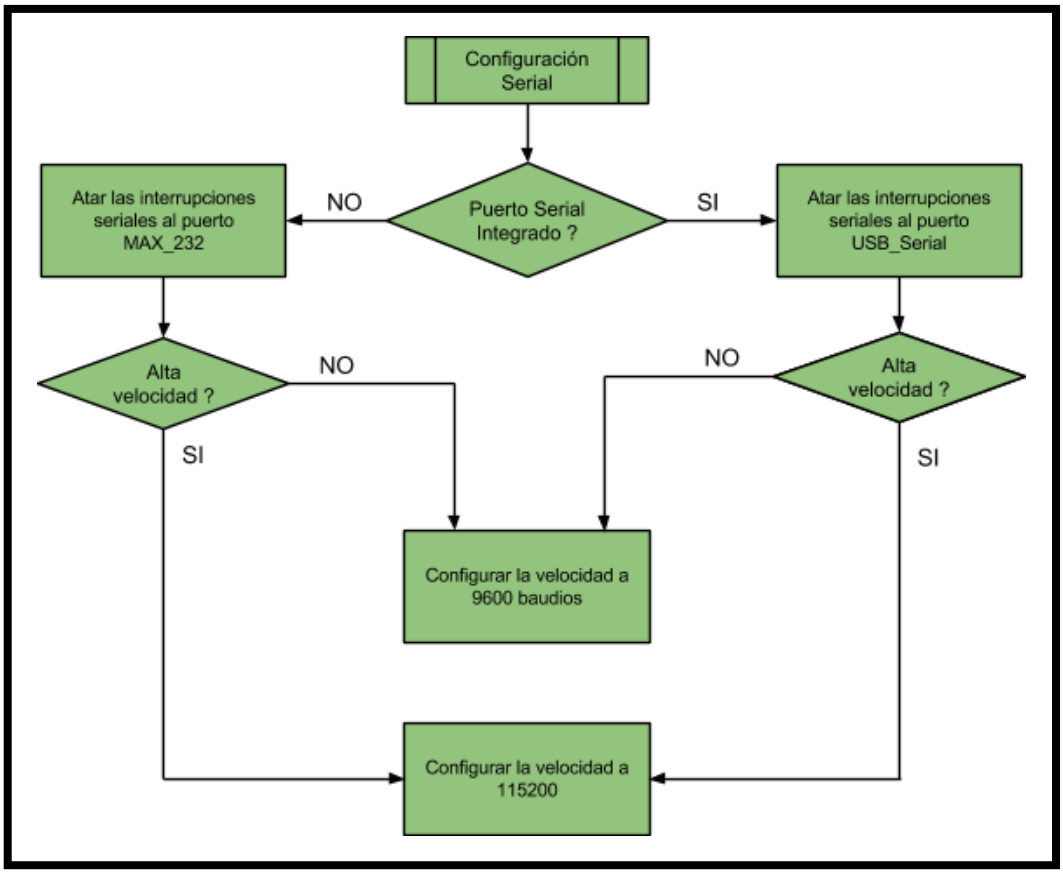
Para la configuración ethernet el microcontrolador añade las clases, funciones y objetos que van a ser controladas remotamente a través de RPC. Esto permite que se puedan realizar llamadas a funciones que ejecutan código en el microcontrolador desde el computador.

Estas funciones son las que controlan el servomotor, y pueden ser accesadas mediante el API de bajo nivel en LabVIEW. Sin embargo el control mediante protocolo RPC no sólo permite la ejecución de funciones sino también la creación de objetos de manera dinámica, esto quiere decir que se pueden acceder y utilizar las demás funciones del microcontrolador, como entradas y salidas digitales, salidas PWM y demás funciones que no han sido implementadas para el deslizador pero que el estudiante puede utilizar para el desarrollo de sus proyectos.

El uso de las funciones e interfaces del microcontrolador por medio del protocolo RPC queda fuera del alcance de este trabajo de grado, ya que su utilización no es necesaria para la implementación del sistema modular, pero es una función muy poderosa del microcontrolador la cual no requiere programación

extra en el microcontrolador, más información acerca de su funcionamiento puede ser encontrado en la página web del microcontrolador.

Fig. 32. Subrutina de configuración serial para la comunicación del microcontrolador



Fuente: Autor

Para la configuración serial se tiene dos puertos que pueden ser utilizados: El puerto serial integrado en el puerto USB, o el puerto serial con salida TTL.

Cualquiera de los dos puertos que se elija, la velocidad de transmisión puede cambiarse entre la genérica de 9600 baudios o una alternativa de alta velocidad, 115200 baudios. Además, para la salida a TTL se tiene un conversor MAX232 instalado y configurado en caso de que el usuario desee utilizarlo.

La configuración serial del sistema se diseñó para dar una gran variedad de posibilidades al momento de comunicarse con el microcontrolador, por lo tanto el usuario tiene la libertad de utilizar un microcontrolador, el computador o cualquier otro dispositivo con conexión RS-232 para el control del deslizador.

Una vez que el mecanismo de transporte de los comandos hacia el ARM-mbed se establece el microcontrolador entra en un estado de interrupciones, a la espera de que llegue un comando y procesarlo para nuevamente esperar la llegada de otro comando.

3.3. API PARA EL CONTROL Y MONITOREO EN LABVIEW

El sistema de control del deslizador está diseñado en el lenguaje de programación gráfica de LabVIEW, este cuenta con librerías en la forma de subrutinas (SubVI) las cuales actúan como una capa de abstracción entre el programador y el hardware del controlador.

Se tiene dos formas de hacer llegar los comandos al microcontrolador, de forma serial y mediante red ethernet, por lo cual se tiene dos librerías, las que están dedicadas a la comunicación serial y aquellas que sirven para comunicarse por medio de la red local. Sin embargo ambos tipos de librerías son simétricas, es decir, los mismos comandos se usan tanto para serial como para ethernet, la única diferencia es la forma de hacerlos llegar al microcontrolador.

Estas librerías están diseñadas dentro de la programación gráfica de LabVIEW, sin embargo debido a su simplicidad es posible que se puedan implementar en otros lenguajes de programación siempre y cuando sean capaces de manejar los protocolos serial y RPC. Estos programas pueden ser: Matlab, python, java, C/C++, entre otros, ejemplos de cómo implementar estos lenguajes pueden ser hallados en la página principal del microcontrolador (http://mbed.org), y de esta manera utilizar el API de LabVIEW como referencia para cualquier otro lenguaje de programación.

3.3.1. API DE BAJO NIVEL

Las librerías de bajo nivel establecen el mecanismo con el cual se pueden comunicar la aplicación del usuario con el microcontrolador. Forman la base sobre la cual se realizan programas para controlar el deslizador.

TIPOS DE CONTROL SOBRE EL DESLIZADOR:

1. Encender el Servo (SON, servo ON)

2. Cambiar la frecuencia de salida. (PULS+, tren de pulsos)
3. Cambiar el sentido de giro (SIGN+, salida digital)
4. Cambiar el voltaje de salida (VCMD, control de velocidad)
5. Leer la posición del encoder.
6. Leer las alertas del deslizador y del driver (ALM, alarmas)

Se han creado diferentes subrutinas para el control de estas variables, las cuales se describen a continuación.

3.3.1.1. Protocolo serial

El envío de datos desde y hacia el microcontrolador se lo define mediante caracteres imprimibles ASCII. Se ha establecido un formato para el envío de datos los cuales son interpretados por el controlador cuando se reciben de manera serial.

Fig. 33. Formato de trama serial

[int32]-[Comando][Enter]

[int32] = Cualquier número entero de 32bits.
[Comando] = Letra mayúscula que designa una acción a realizar.
[Enter] = Carácter Retorno de carro para enviar el comando.

Ejemplo de comandos aceptables:

1500-H
-125000000-P
0-E
-1-R

Ejemplo de comandos erróneos:

5.23-A
K-5000
2300-h
-1200A

Fuente: Autor

La trama cuenta con dos parámetros separados por un guion, el primer parámetro es un número entero de 32 bits, el segundo parámetro es un código

para la función que se va a ejecutar. Al final de la trama se debe enviar el carácter especial CR (Carriage Return: 0x13).

Si el comando ha sido enviado correctamente se obtendrá una respuesta afirmativa por parte del microcontrolador, caso contrario se presentará una respuesta negativa, existen ciertos comandos de consulta al microcontrolador los cuales devolver un valor numérico, se analizará detenidamente cada uno de ellos

A continuación una lista de los comandos aceptados por el microcontrolador.

Tabla 10. Lista de posibles comandos seriales

CARACTER	FUNCIÓN IMPLEMENTADA
S	Encender o apagar el servomotor
H	Cambiar la frecuencia de salida en hertzios
K	Cambiar la frecuencia de salidas en kilohertzios
D	Cambiar la dirección de giro
A	Cambiar el voltaje de salida.
P	Envía un número predefinido de pulsos
M	Envía un número predefinido de miles de pulsos
N	Envía un número predefinido de millones de pulsos
V	Cambiar la velocidad del envío predefinido de pulsos
I	Lleva el carro del deslizador a la posición inicial y encera el contador del encoder
E	Lee la posición del encoder
R	Lee la velocidad en RPM instantánea del encoder
Z	Limpiar el contador del encoder

Fuente: Autor

El valor del primer parámetro (int_32 - Número entero de 32 bits valores entre: + 2 147 483 647 y - 2 147 483 647) desempeña diferentes funciones dependiendo del comando que lo acompaña, de la siguiente manera.

Tabla 11. Lista de comandos seriales detallada

Comando	Valores aceptables	Valor por defecto	Detalle
S	0 a 1	0	0 Apagar el servomotor, 1 Encender servomotor.
H	0 a 500000	0	Frecuencia de salida del tren de pulsos, en hertzios.
K	0 a 500	0	Frecuencia de salida del tren de pulsos, en kilohertzios.
D	0 a 1	0	Sentido de giro: (CW = 1 – Hacia el motor). (CCW = 0 – Hacia el encoder).
A	-10000 a +10000	0	Voltaje de salida de +-10V con una resolución de ~4.8mV.
P	0 a 2 147 483 647	0	Número predefinido de pulsos
M	0 a 2 147 483 647	0	Generar un número predefinido de pulsos, multiplica el valor del comando por mil
N	0 a 2 147 483 647	0	Generar un número predefinido de pulsos, multiplica el valor del comando por un millón
V	10 - 500000	1000 Hz	La velocidad en tren de pulsos a la cual se va a llegar al ángulo deseado para el eje del motor.
I	0 a 2 147 483 647	0	Lleva el carro al inicio del recorrido. El valor numérico determina la frecuencia del tren de pulsos
E	0 a 1	0	0 Leer posición actual. 1 Leer velocidad actual.
Z	0	0	Limpia el contador del encoder

Fuente: Autor

El microcontrolador ejecutará estos comandos **sin importar cuál es la fuente de su procedencia**, siempre y cuando arriben de manera serial y tengan el formato de trama correcto.

La salida analógica del microcontrolador será modifica para tener una mejor resolución sobre el voltaje de salida. Por esta razón el voltaje que se desee a la salida se debe lo debe ingresar en milésimas de voltio, por ejemplo:

Tabla 12. Relación voltaje - comando

Voltaje deseado (V)	Comando enviado
+ 5.000	5000-A
- 5.000	-5000-A
+ 9.874	9874-A
- 8.750	-8750-A
+10.000	10000-A

Fuente: Autor

El microcontrolador intentará reflejar el voltaje con la mayor precisión posible.

3.3.1.2. Protocolo Ethernet

El microcontrolador mbed cuenta con la capacidad de implementar una red Ethernet, en la cual se pueden ejecutar diferentes servicios como websockets o servidores HTTP. Para este trabajo de grado se utilizó el protocolo de comunicación RPC, el cual se monta sobre el servidor HTTP propio de mbed.

En lo que se traduce lo anteriormente mencionado es que se puede controlar el microcontrolador desde un navegador web, gracias a la implementación de la librería RPC.

La plataforma de desarrollo mbed cuenta con una librería propia desarrollada para LabVIEW en la cual se puede utilizar para enviar y recibir comandos de una manera muy sencilla. Aprovechando todas estas facilidades brindadas por la plataforma de desarrollo se han elaborado librerías específicas para el deslizador lineal, las cuales interactúan con el microcontrolador desde el lenguaje de programación de LabVIEW.

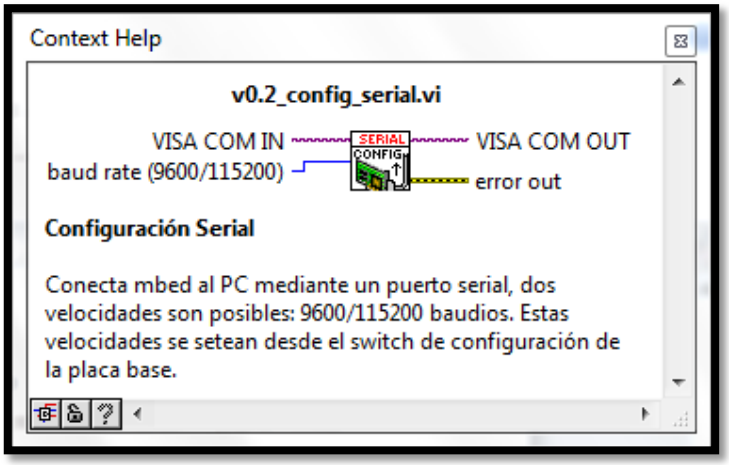
3.3.2. LIBRERÍA EN LABVIEW

En LabVIEW las librerías que envían comandos al microcontrolador son las librerías del API de bajo nivel, en las que se simplifica y se automatiza la comprobación de errores y el envío de comandos al microcontrolador. Estas librerías se han creado tanto para la comunicación serial como Ethernet, permitiendo de esta manera mayor flexibilidad al programador en cuanto a su comunicación con el dispositivo.

3.3.2.1. API Serial - Configuración

Para poder utilizar comunicación serial se debe primero configurar la velocidad de transmisión y el puerto en el que se va a conectar el microcontrolador, para esta configuración se utiliza la subrutina “config_serial.vi”

Fig. 34. Visualización CONFIG_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- VISA COM IN: El puerto de comunicación asignado al microcontrolador por parte de Windows (ejemplo: COM4),
- Baud rate: La velocidad en baudios con que se ha configurado al microcontrolador (ejemplo: 9600).

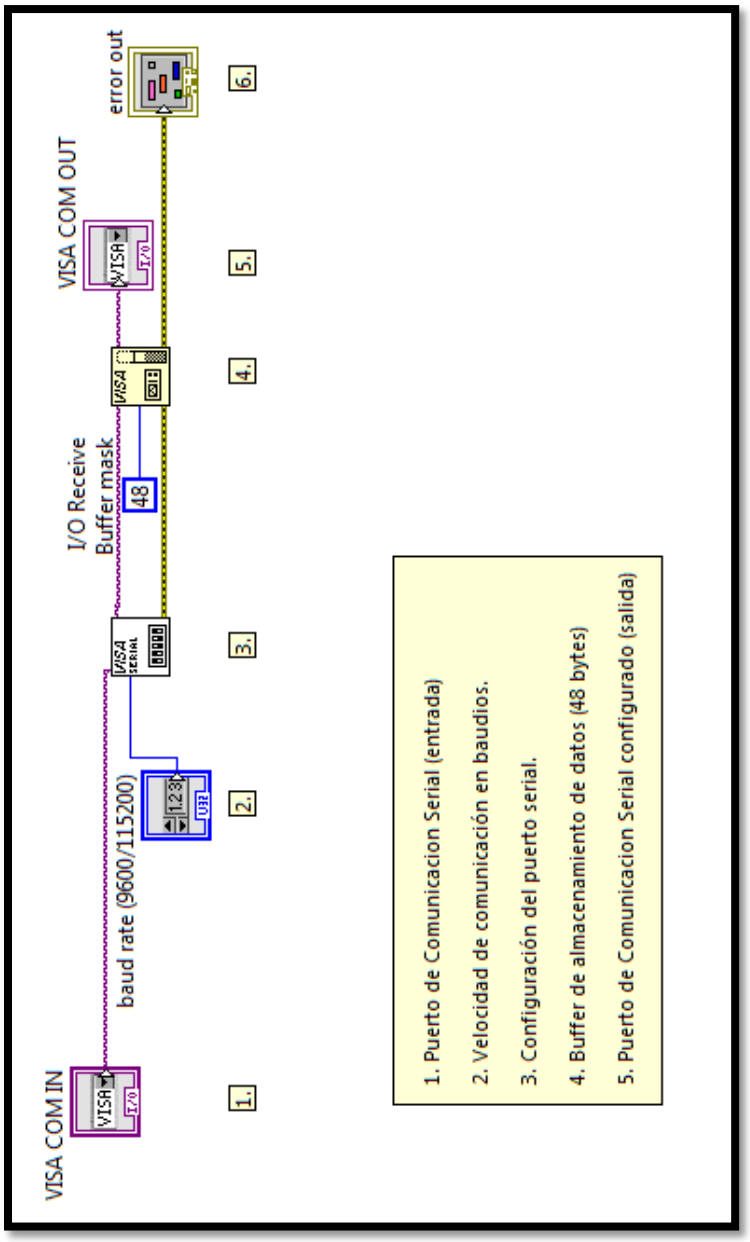
Salidas.-

- VISA COM OUT: El puerto correctamente configurado con para la comunicación
- Error out: Clúster de error.

Una vez ejecutado la subrutina devolverá el puerto serial configurado para que pueda ser utilizado por las demás subrutinas de la librería.

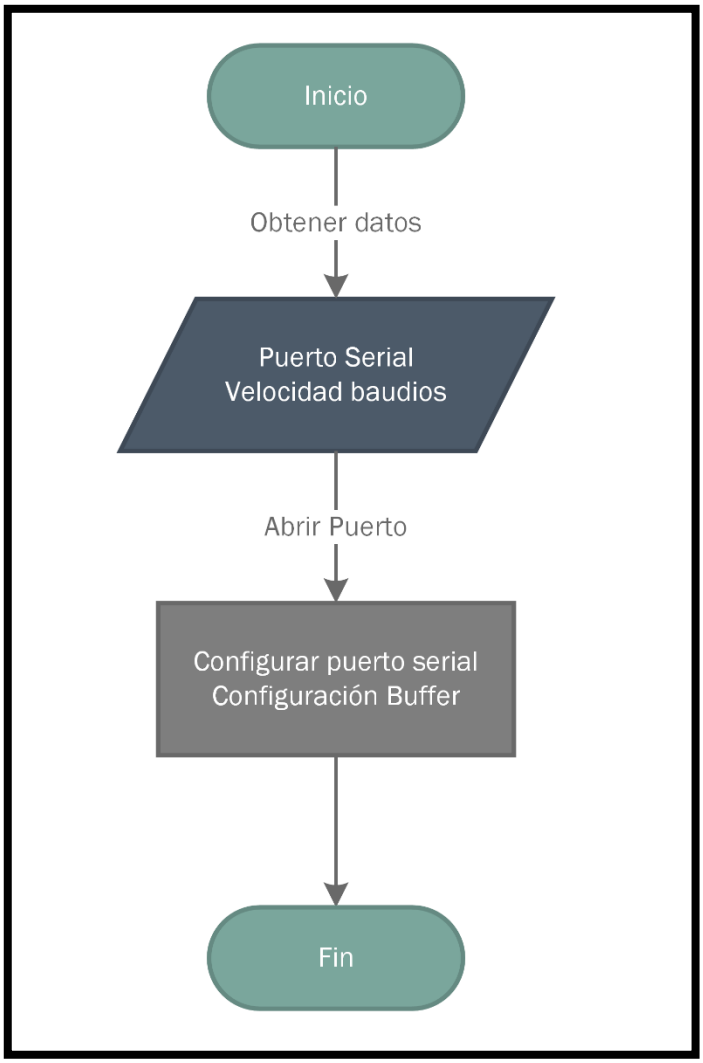
Es importante que la configuración serial se dé a través de esta subrutina debido a que el microcontrolador cuenta con un buffer de 48 bytes, una vez que se establece esta configuración se puede utilizar el puerto serial del microcontrolador.

Fig. 35. Diagrama de Bloques - CONFIG_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Fig. 36. Flujo grama CONFIG_SERIAL



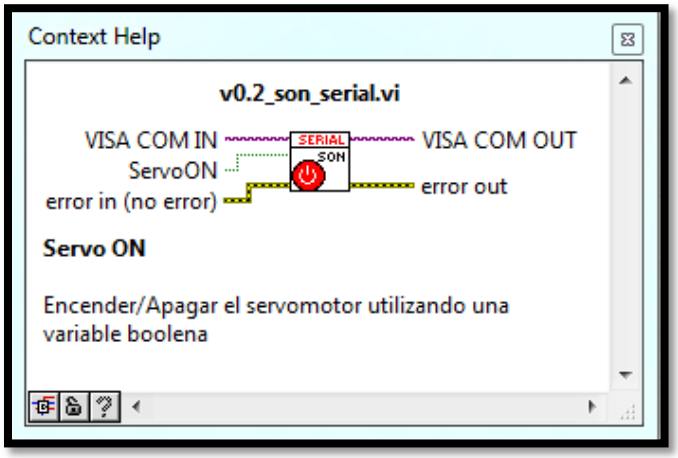
Fuente: Autor

3.3.2.2. API Serial – Encender Servomotor

La siguiente subrutina permite encender o apagar el servomotor. Para que el servomotor pueda encenderse de manera adecuada se debe comprobar que no existan alarmas en el servodrive y que el deslizador se encuentre en una posición válida. Se puede hacer una verificación visual rápida en la placa de conexiones del encoder, si existe algún led rojo encendido este significa una posición inválida. Es también importante verificar que el botón de paro de emergencia no se encuentre accionado

El estado del motor se controla a través de la variable booleana, verdadero significa encender el servomotor, falso apagarlo.

Fig. 37. Visualización SERVO_ON



Fuente: Autor (Software: LabVIEW 2013)

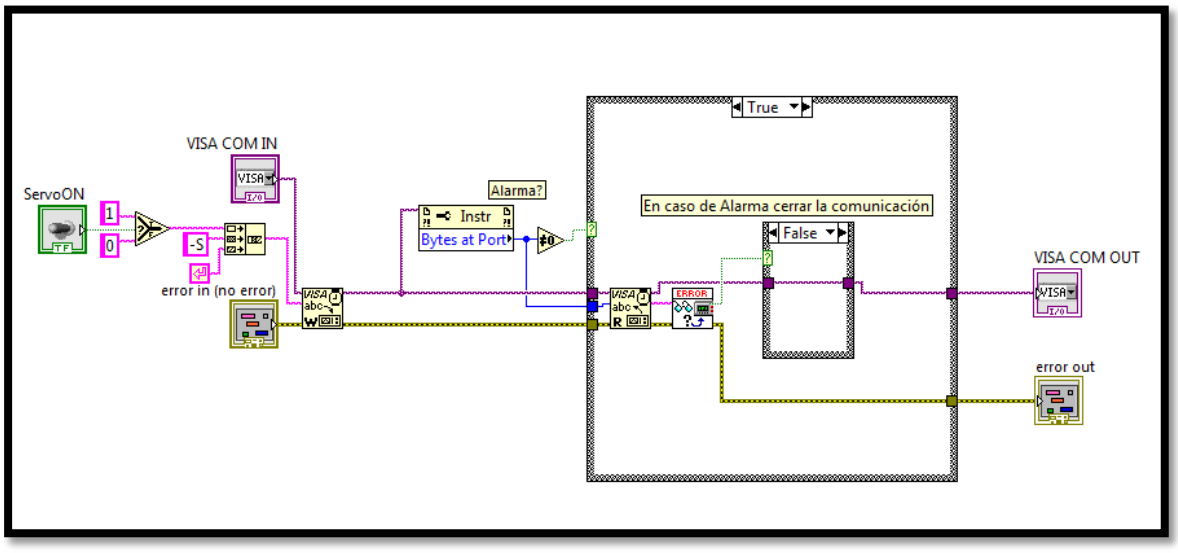
Entradas.-

- VISA COM IN: El puerto de comunicación configurado.
- ServoON: Variable booleana verdadero/falso.

Salidas.-

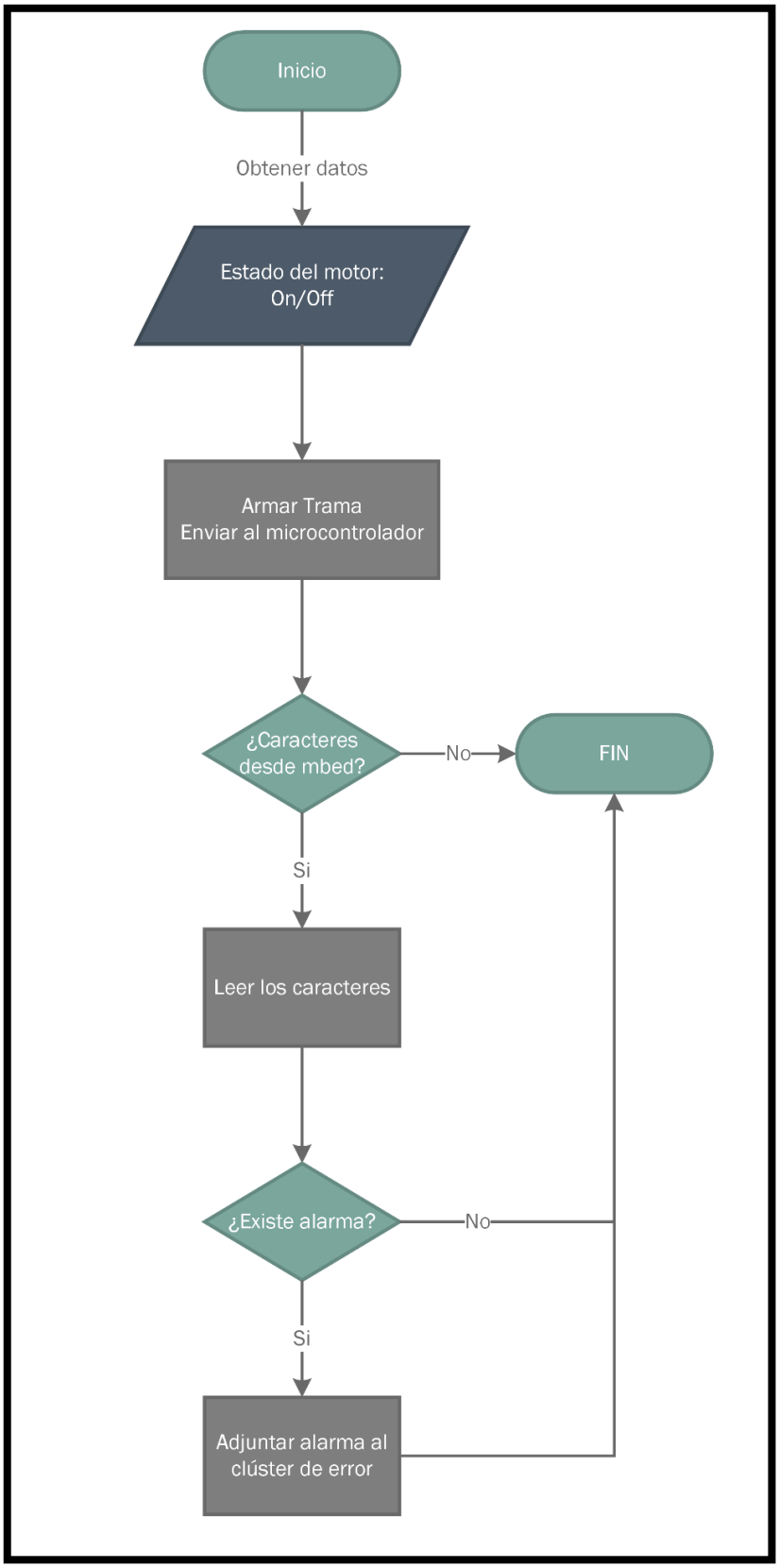
- VISA COM OUT: El puerto de comunicación configurado.
- Error out: Clúster de error.

Fig. 38. Diagrama de bloques - SERVO_ON



Fuente: Autor (Software: LabVIEW 2013)

Fig. 39. Flujo grama - SERVO_ON



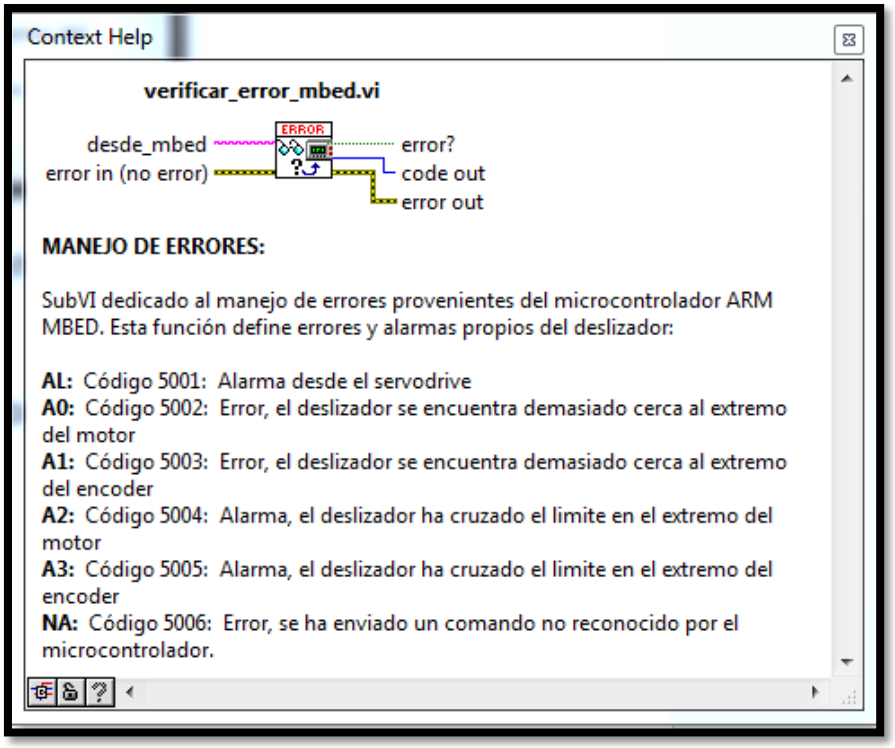
Fuente: Autor

3.3.2.3. API Serial – Manejo de errores

Se elaboró un clúster de error personalizado para la comunicación con el microcontrolador, la cual permite integrar los errores que se puedan generar en el deslizador con los errores nativos de LabVIEW. De esta manera se logra depurarlos de la misma manera, ya sea un error en programación, error en comunicación o inclusive un error mecánico del sistema.

El microcontrolador siempre devuelve una trama de respuesta cuando se le envía un comando. Esto cumple con varios propósitos, uno de ellos es tener un acuse de recibo que permita validar que la comunicación se estableció correctamente y que el comando enviado se ejecutó. El segundo propósito es monitorear posibles alarmas generadas en el servodrive, generalmente por un torque o frecuencia muy elevadas. Finalmente se pretende verificar que el carro del deslizador golpeé sus extremos, esto puede pasar debido a una mala programación y se han colocado advertencias y alarmas que permitan evitar esta colisión.

Fig. 40. Visualización VERIFICAR_ERROR_MBED



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- Desde_mbed: Caracteres devueltos por el microcontrolador
- Error in: Clúster de error de entrada.

Salidas.-

- Error?: Si existe una alarma devolverá verdadero, caso contrario falso.
- Code out: Código del error de existir, cero al no existir error.
- Error out: Clúster de error de salida.

Tabla 13. Réplicas del microcontrolador

Respuesta	Significado
OK	El microcontrolador responderá con estos dos caracteres cuando el comando enviado es reconocido y ejecutado. Sirve para comprobar que el comando ha sido recibido por el microcontrolador.
NA	El microcontrolador responderá con estos dos caracteres cuando el comando enviado NO es reconocido ni ejecutado. Si se recibe esta respuesta se debe revisar la sintaxis del comando enviado. Si se utiliza la subrutina “verificar_error_mbed” se asignara el código de error 5006 y se lo interpreta como una alarma, el cual puede detener la ejecución del programa en LabVIEW.
AL	El microcontrolador responderá con este comando cuando la salida digital de alarma del sevodrive se active. Si se utiliza la subrutina “verificar_error_mbed” se asignara el código de error 5001 y se lo interpreta como una alarma, la cual podrá detener la ejecución del programa en LabVIEW.

Respuesta	Significado
A0	El microcontrolador responderá con este comando cuando el carro del deslizador ha sobrepasado el primer sensor óptico del deslizador, es decir, se encuentra demasiado cerca al extremo del motor. Si se utiliza la subrutina "verificar_error_mbed" se asignara el código de error 5002 y se lo interpreta como un error, el cual no detendrá la ejecución del programa en LabVIEW.
A1	El microcontrolador responderá con este comando cuando el carro del deslizador ha sobrepasado el segundo sensor óptico del deslizador, es decir, se encuentra demasiado cerca al extremo del encoder. Si se utiliza la subrutina "verificar_error_mbed" se asignara el código de error 5002 y se lo interpreta como un error, el cual no detendrá la ejecución del programa en LabVIEW.
A2	El microcontrolador responderá con este comando cuando el carro del deslizador ha sobrepasado el tercer sensor óptico del deslizador, es decir, se encuentra en una posición inválida, demasiado cerca al extremo del motor. Si se utiliza la subrutina "verificar_error_mbed" se asignara el código de error 5002 y se lo interpreta como una alarma, la cual puede detener la ejecución del programa en LabVIEW.
A3	El microcontrolador responderá con este comando cuando el carro del deslizador ha sobrepasado el cuarto sensor óptico del deslizador, es decir, se encuentra demasiado cerca al extremo del motor. Si se utiliza la subrutina "verificar_error_mbed" se asignara el código de error 5002 y se lo interpreta como una alarma, la cual puede detener la ejecución del programa en LabVIEW

Fuente: Autor

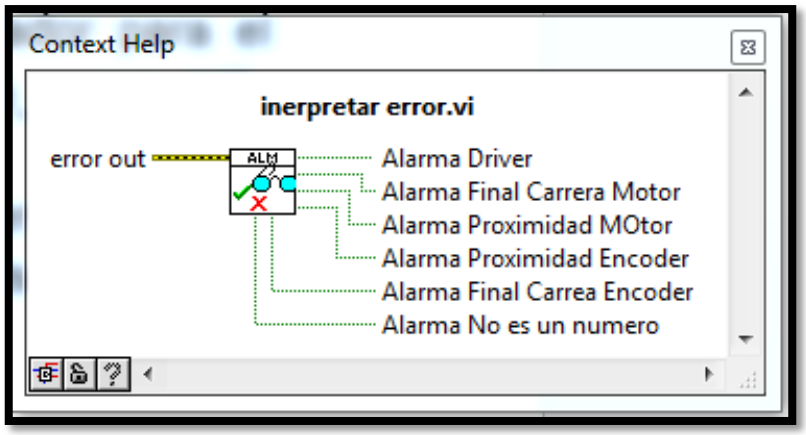
En el modo de comunicación serial se han creado interrupciones las cuales permiten que una vez que se genere una alarma se envíen los comandos inmediatamente. En cambio en el modo de comunicación Ethernet es necesario realizar una consulta para conocer el estado de las alarmas, esto se detalla más adelante.

La subrutina “verificar_error_mbed” interpreta los comandos devueltos por el microcontrolador y asigna el código de error correspondiente en caso de ser necesario, además añade un mensaje de error para ser mostrado en caso de que el programador escoja hacerlo.

Es importante aclarar que este SubVI solamente adjunta los errores al clúster general de LabVIEW, pero es labor del programador capturarlos, y elegir las acciones que se deben realizar.

Para complementar la función de manejo de errores se creó una subrutina que analice el clúster de error y permita el programador tomar acciones correctivas.

Fig. 41. Visualización INTERPRETAR_ERROR



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- Error out: Clúster de error para ser analizado.

Salidas.-

- Alarma Driver.- Verdadero si existe alarma del driver.

- Alarma Final de carrera motor: Verdadero si el carro del deslizador se encuentra en una posición inválida demasiado cerca al servomotor
- Alarma Proximidad motor: Verdadero si el carro del deslizador se encuentra demasiado cerca al extremo del servomotor.
- Alarma Proximidad encoder: Verdadero si el carro del deslizador se encuentra demasiado cerca al extremo del encoder.
- Alarma Final de carrera motor: Verdadero si el carro del deslizador se encuentra en una posición inválida demasiado cerca al encoder.
- Alarma No es un número: Verdadero si el formato del comando enviado no coincide con el que espera el microcontrolador.

3.3.2.4. API Serial - Tren de pulsos

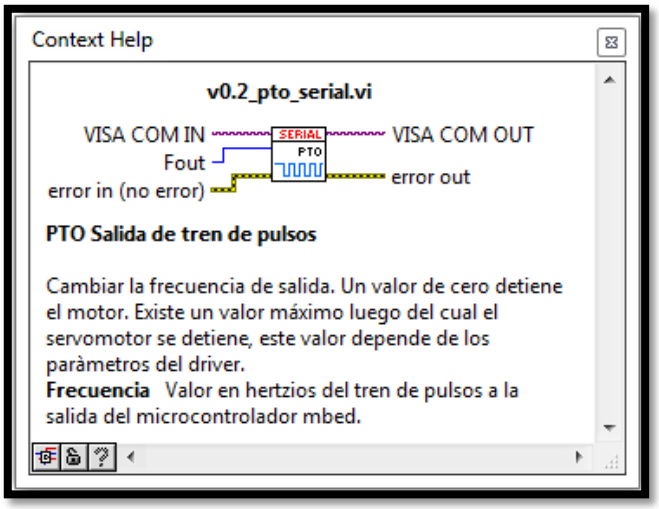
Para controlar el tren de pulsos generado por el microcontrolador para el control de posición del servomotor se utiliza la siguiente subrutina “pto_serial.vi”. Las siglas vienen de “Pulse Train Output (PTO)”.

El tren de pulsos sirve en el modo de “Control de Posición” del servomotor. El microcontrolador cuenta con la facultad de generar este tren de pulsos con una frecuencia que varía desde uno hasta 500 mil herztios.

Este tren de pulsos es alimentado al servodriver el cual hará rotar el servomotor, la velocidad de salida del servomotor dependerá de la configuración de los parámetros del mismo.

El sentido de giro del eje del servomotor depende del estado de la salida SIGN, la cual se controla a través de la subrutina “dir_serial.vi”. la frecuencia máxima aceptada por el servodrive es de 160 KHz. Más allá de este valor el servodrive detendrá al servomotor.

Fig. 42. Visualización PTO_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- VISA COM IN: El puerto de comunicación configurado.
- Fout: Frecuencia de salida para el generador de pulsos en hertzios.
- Error in: Clúster de error de entrada

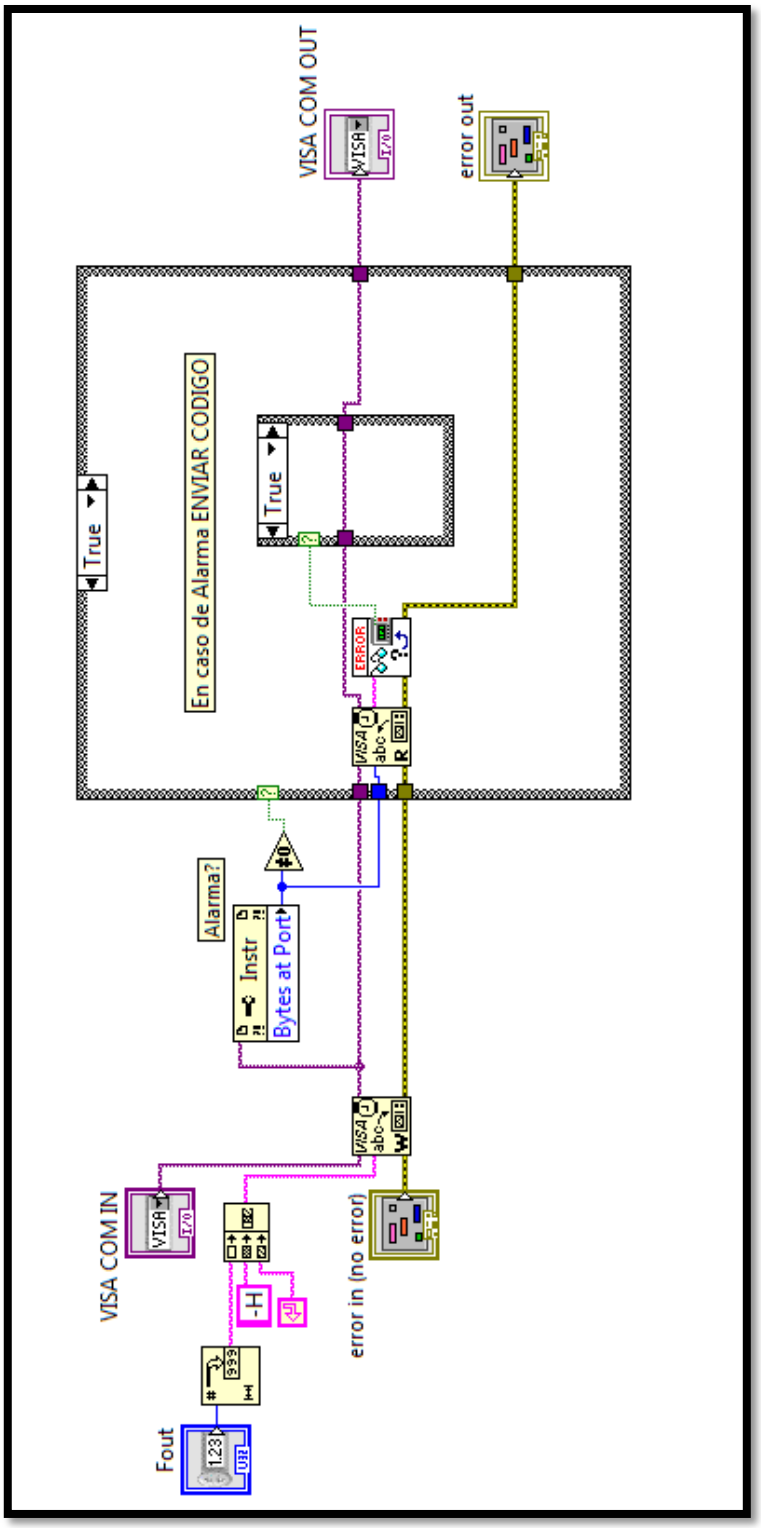
Salidas.-

- VISA COM OUT: El puerto serial configurado.
- Error out: Clúster de error de salida.

La resolución a la cual se pueden seleccionar una frecuencia para el tren de puolsos es menor a medida que se seleccionan frecuencias más alto, esto se debe a que la frecuencia de reloj del microcontrolador es de 96MHz. Este reloj alimenta al temporizador que genera los pulsos. El servodrive acepta frecuencias de hasta 160 KHz, en este rango de frecuencias se tiene una resolución de un kilohertzio, es decir se pueden seleccionar frecuencias de 157, 158,159 KHz.

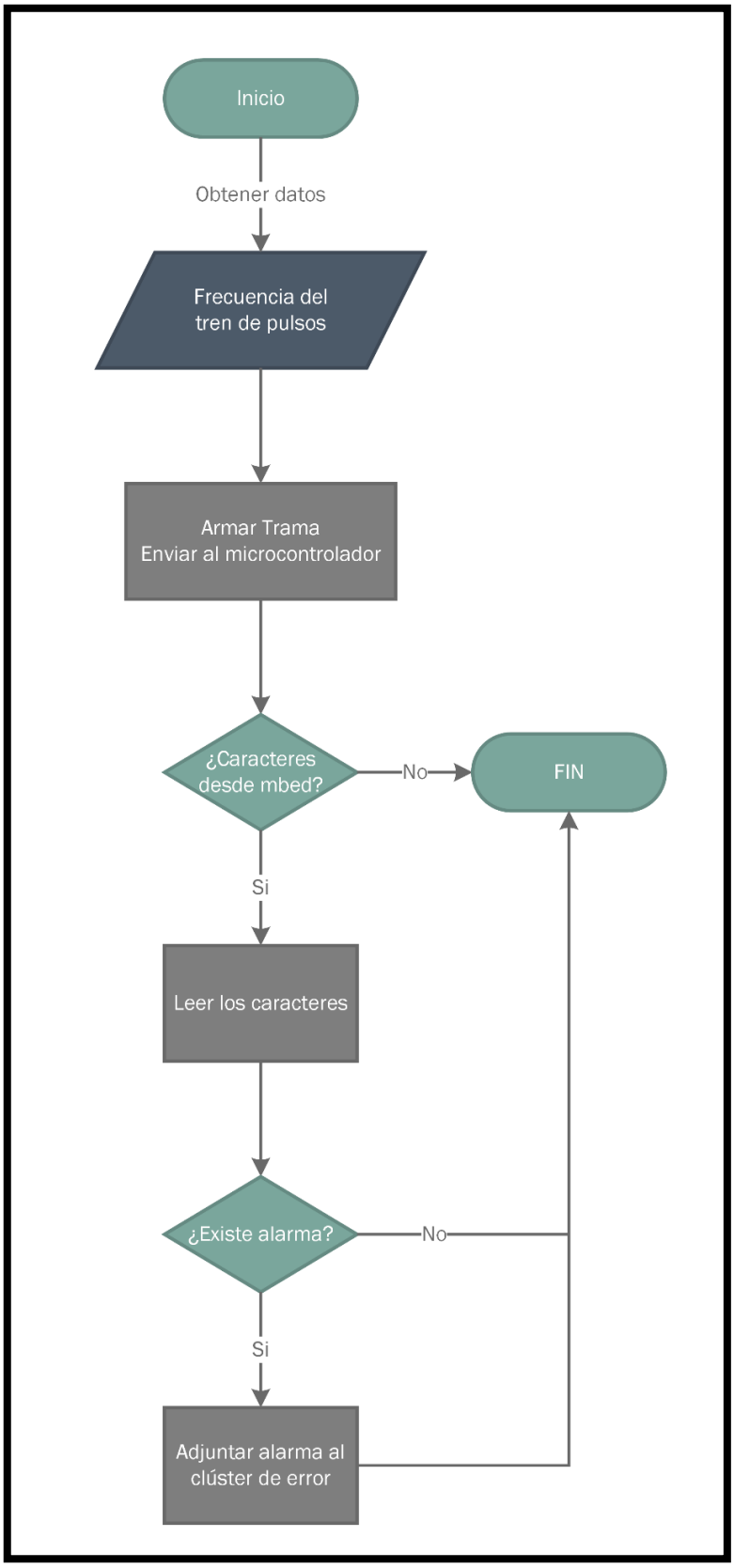
En el rango inferior de frecuencias se tiene una resolución de hertzios, 10,15,19 Hz son valores aceptables, como norma se sugiere que a partir de los 10KHz se seleccionen frecuencias en un rango de más menos 1Khz.

Fig. 43. Diagrama de bloques - PTO_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Fig. 44. Flujo grama - PTO_SERIAL



Fuente: Autor

3.3.2.5. API Serial - Dirección

Para cambiar la dirección de giro del eje del servomotor se utiliza el SubVI “dir_serial.vi” a través de una variable booleana, es decir, verdadero/falso. Este VI funciona cuando el servomotor se encuentra en el modo de control por posición, cambiando la salida digital SIGN +.

Fig. 45. Visualización DIR_SERIAL

Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- VISA COM IN: El puerto de comunicación configurado.
- Dirección: Valor verdadero o falso para controlar la dirección de giro.
- Error in: Clúster de error de entrada

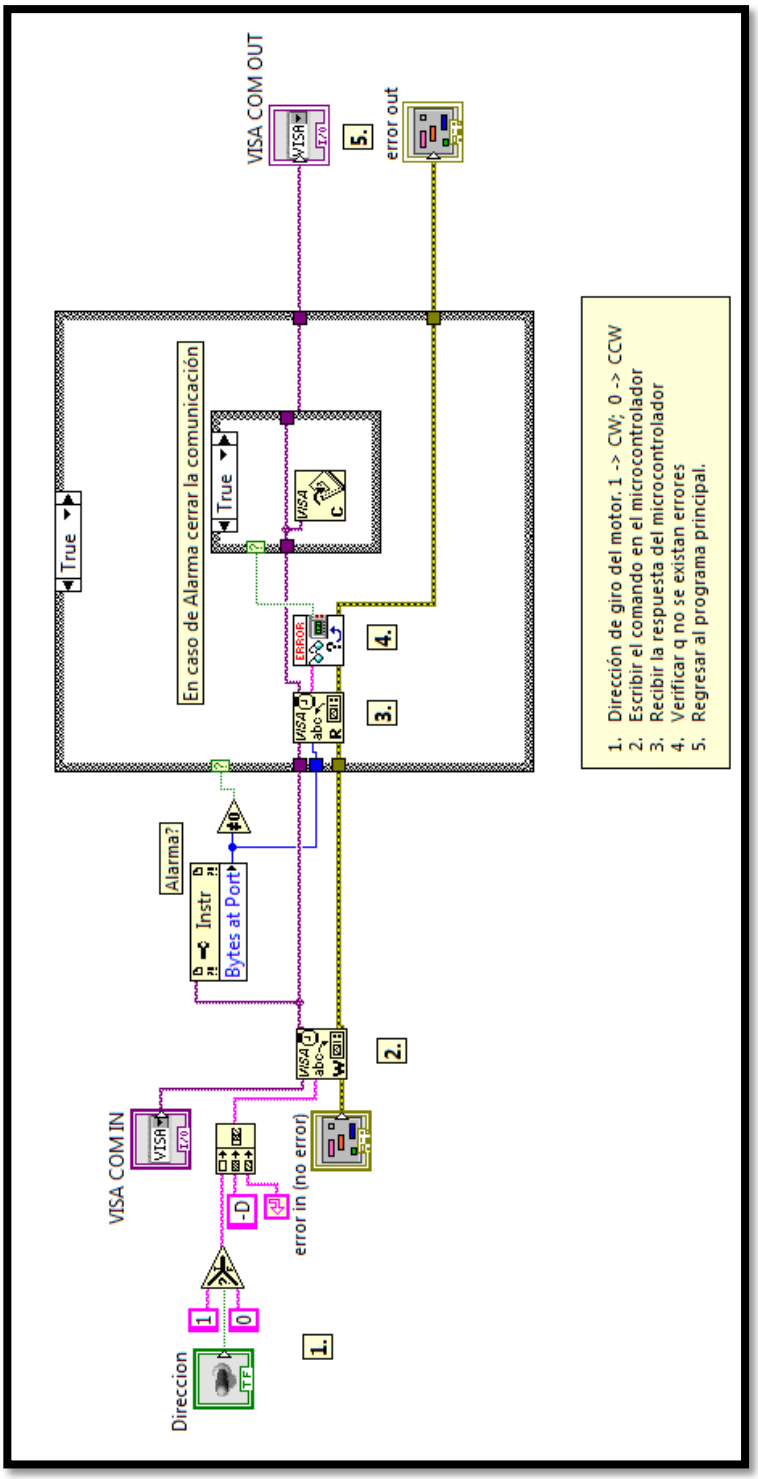
Salidas.-

- VISA COM OUT: El puerto serial configurado.
- Error out: Clúster de error de salida.

La dirección es controlada a través de una variable booleana, verdadero significa que el carro del deslizador se acercará al extremo donde está el servomotor, si la variable es falsa, significa que el carro deslizador se alejará hacia el extremo donde se encuentra instalador el encoder.

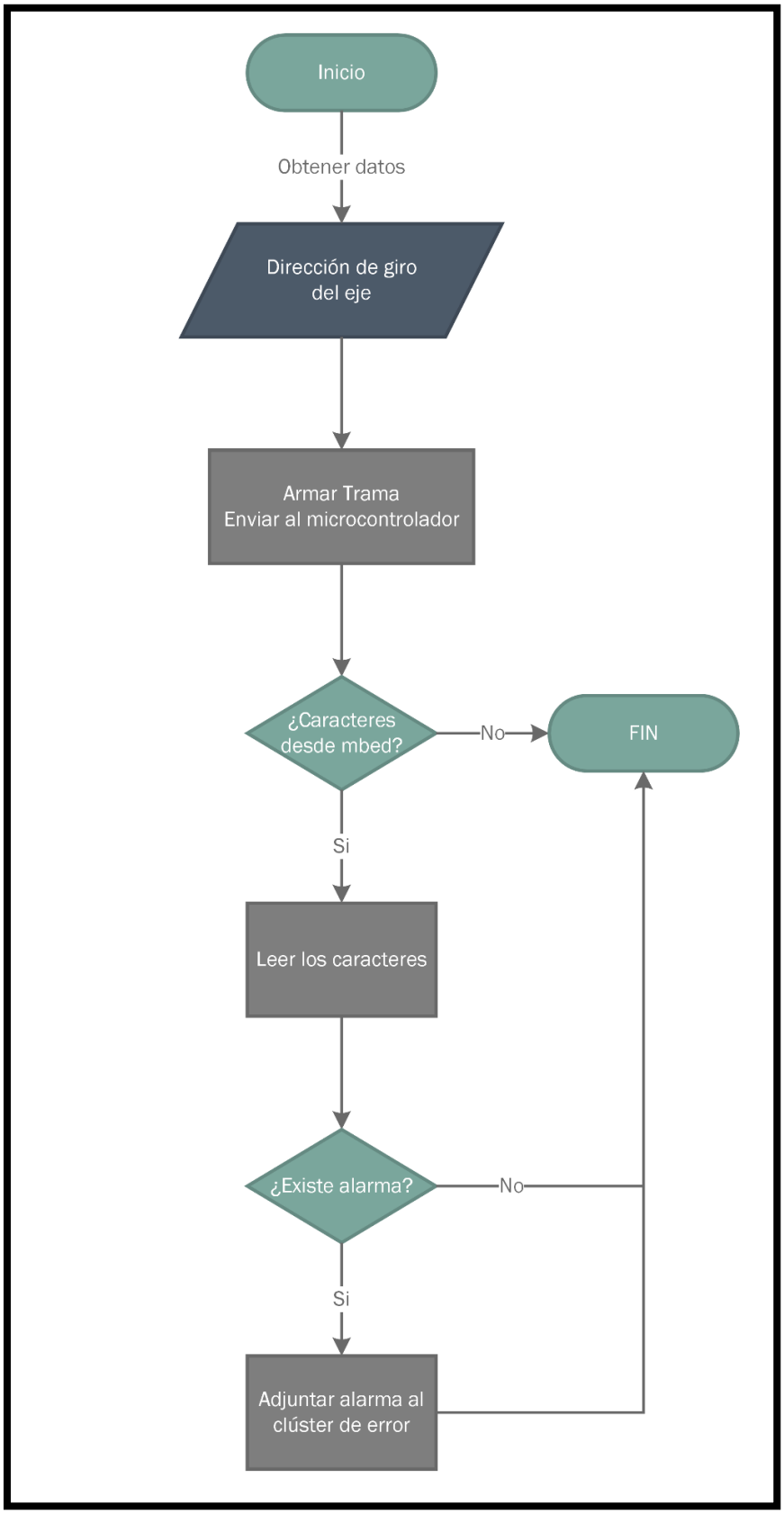
El microcontrolador comprueba si el valor enviado es mayor a cero, por lo tanto cualquier valor superior a cero es tomado como verdadero.

Fig. 46. Diagrama de bloques - DIR_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Fig. 47. Flujo grama - DIR_SERIAL

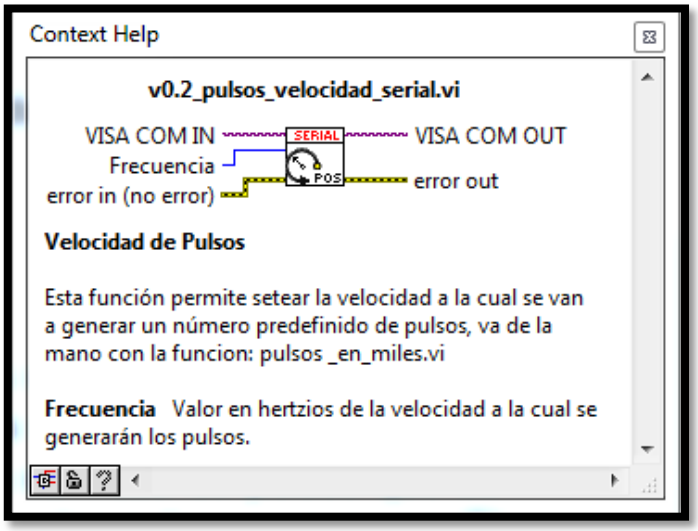


Fuente: Autor

3.3.2.6. API Serial – Velocidad de Pulsos

La generación de tren de pulsos genera pulsos de una manera constante, pero no se tiene un control preciso sobre la cantidad de pulsos que se va a generar. Para tener este control se deben definir dos variables, la cantidad de pulsos que se desea generar y la velocidad a la cual queremos que estos pulsos se generen. La presente subrutina se encarga de definir la velocidad a la cual se van a generar los pulsos, más adelante veremos la subrutina que permite determinar su cantidad.

Fig. 48. Visualización PULSOS_VELOCIDAD



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- VISA COM IN: El puerto de comunicación configurado.
- Frecuencia: Valor en hertzios a la cual se generarán los pulsos.
- Error in: Clúster de error de entrada

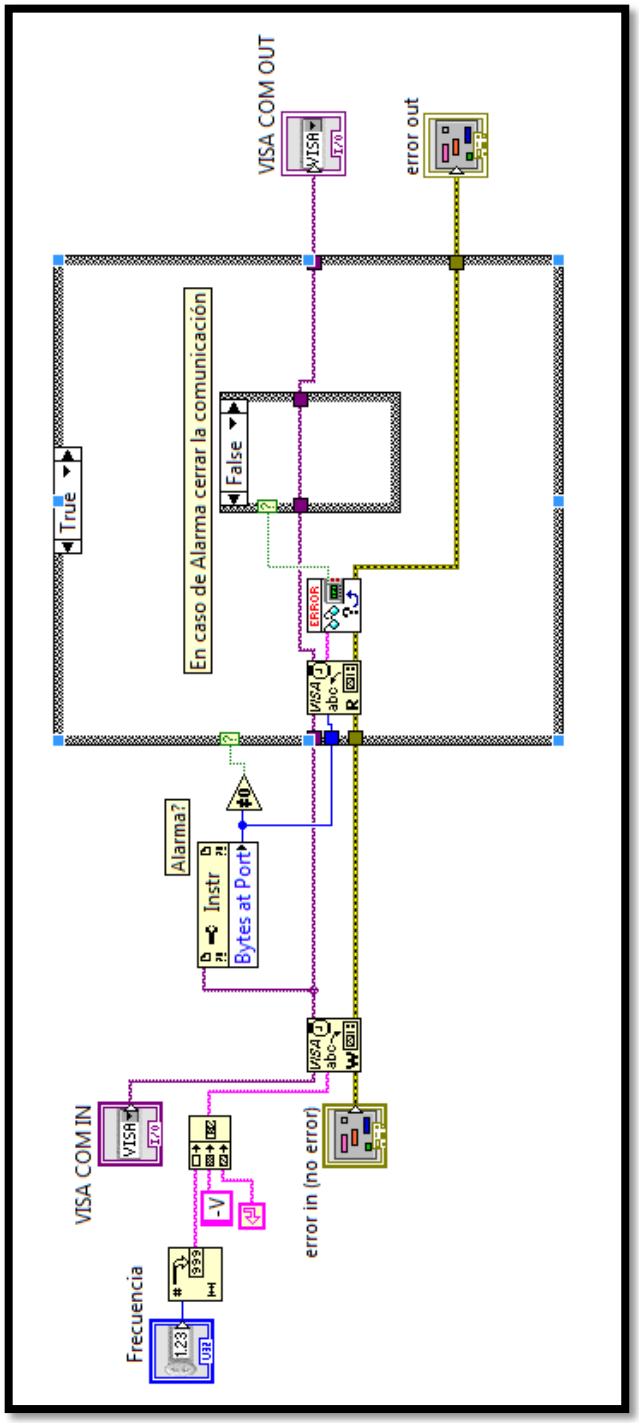
Salidas.-

- VISA COM OUT: El puerto serial configurado.
- Error out: Clúster de error de salida.

Esta subrutina se utiliza para que el carro deslizador se pueda desplazar una distancia determinada con la mayor precisión posible. La distancia final a desplazarse dependerá de la configuración de los parámetros del servodrive.

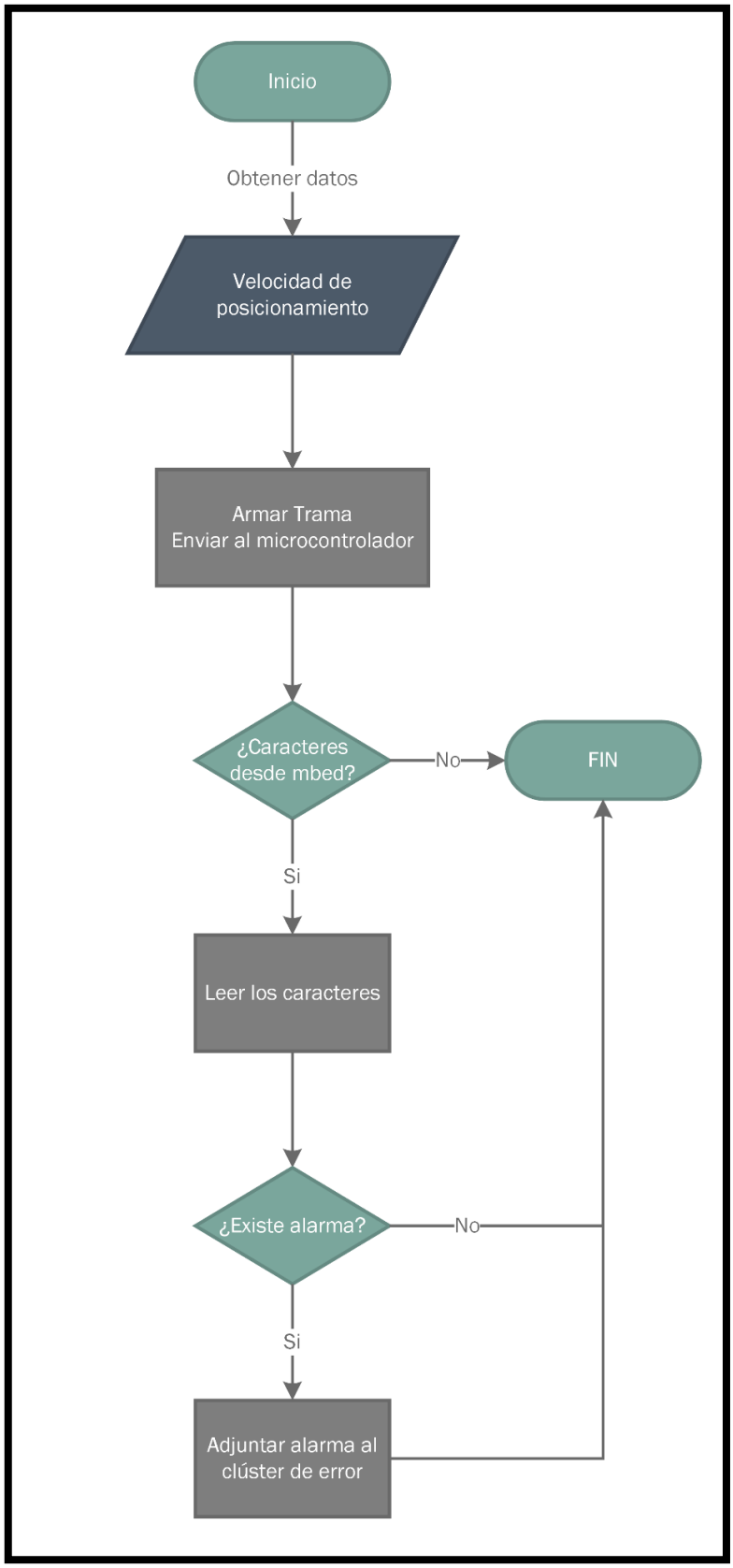
La variable frecuencia determinará la velocidad lineal a la cual se moverá el carro deslizador hasta llegar a su posición final, la dirección del giro sigue estando determinada por la variable de la subrutina “dir_serial.vi”.

Fig. 49. Diagrama de bloque - PULSOS_VELOCIDAD



Fuente: Autor (Software: LabVIEW 2013)

Fig. 50. Flujo grama - PULSOS_VELOCIDAD.

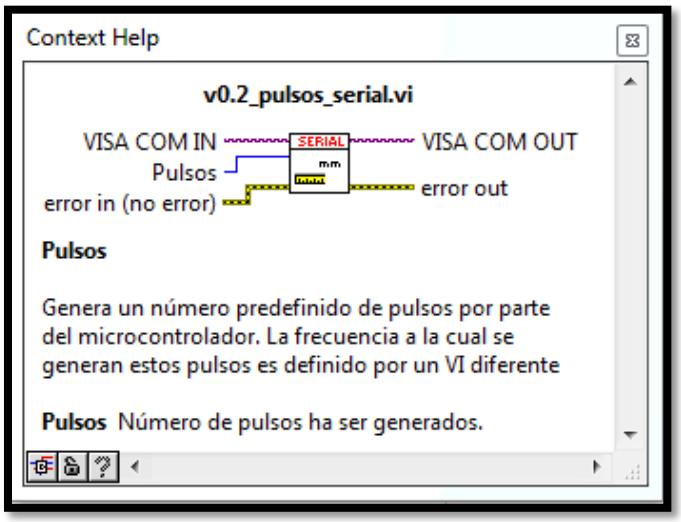


Fuente: Autor

3.3.2.7. API Serial – Definir Pulsos

Esta subrutina permite que el microcontrolador genere un número predefinido de pulsos, se utiliza para para desplazar el dispositivo una distancia predefinida en el modo de control de posición

Fig. 51. Visualización PULSOS



Fuente: Autor (Software: LabVIEW 2013)

La distancia recorrida depende de la configuración del servodrive, y la velocidad a la que se generarán dependerá del último comando de velocidad de posicionamiento enviado.

En la resolución por defecto se deben enviar 10 000 pulsos para recorrer cinco milímetros. Se tiene esta es la razón por la que se creó una subrutina adicional en la cual se utiliza un multiplicador por mil en el microcontrolador “pulsos_miles_serial.vi”, para evitar transmitir una elevada cantidad de caracteres.

Entradas.-

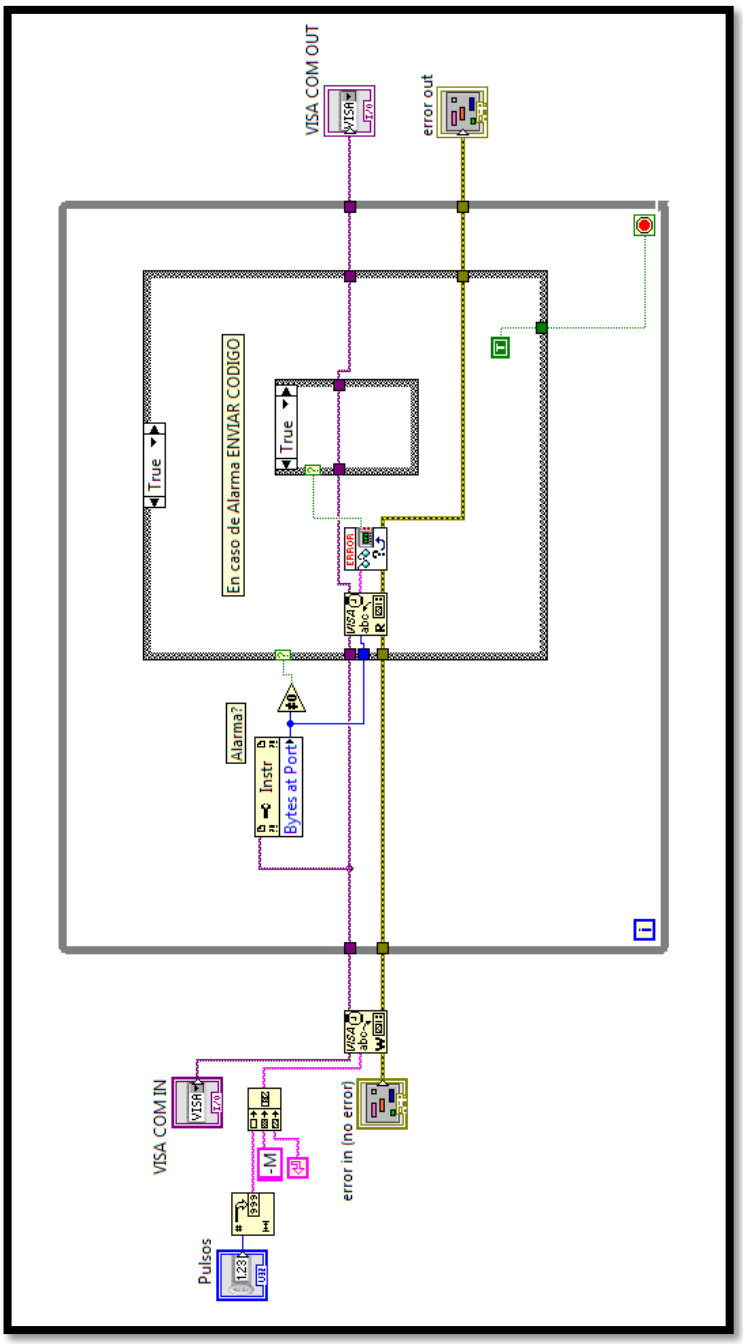
- VISA COM IN: El puerto de comunicación configurado.
- Pulsos: Valor pulsos a generar.
- Error in: Clúster de error de entrada

Salidas.-

- VISA COM OUT: El puerto serial configurado.
- Error out: Clúster de error de salida.

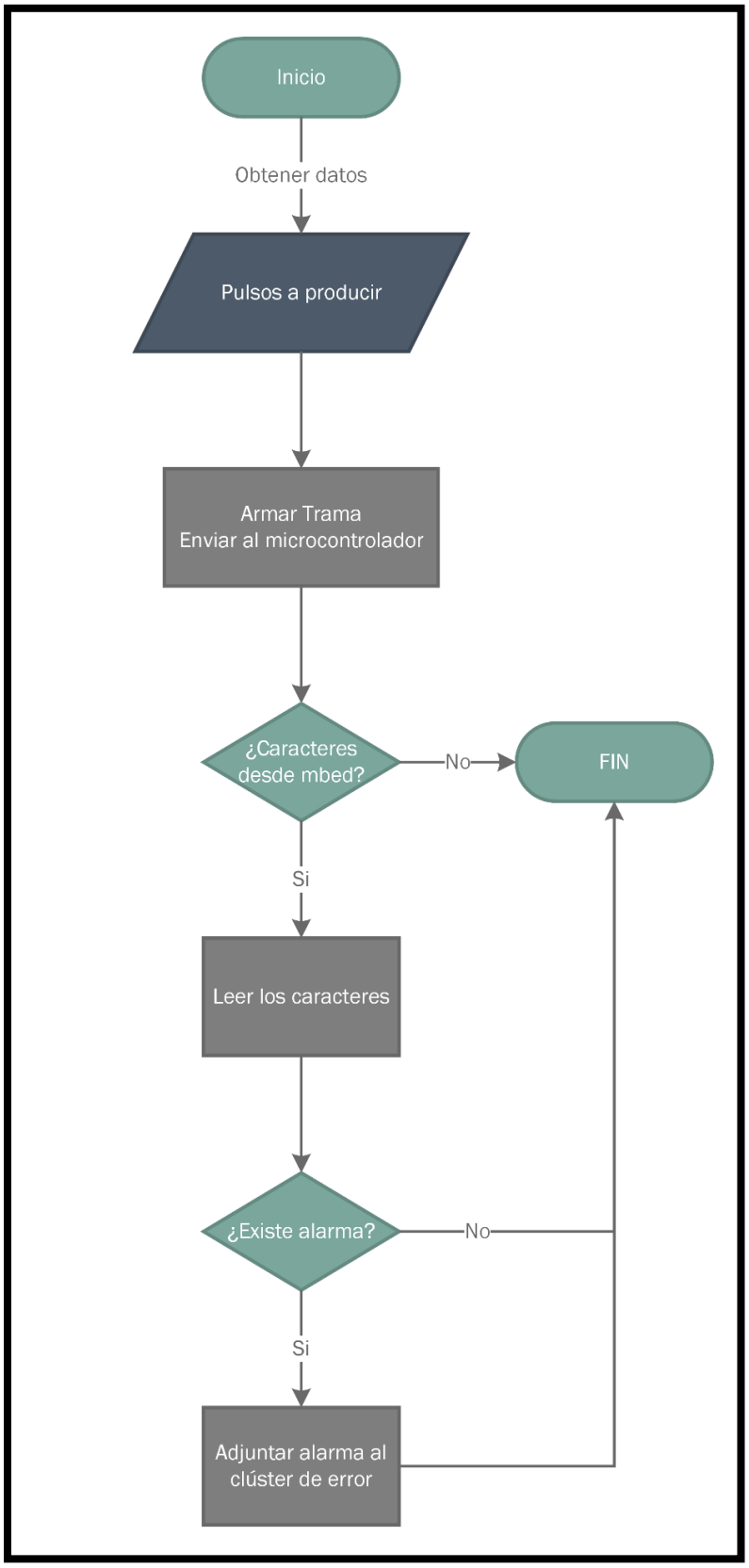
Esta es una función especial, en la cual durante el tiempo que se generan los pulsos no se enviará la respuesta de OK, durante este periodo de tiempo la subrutina esperará esta respuesta, esto significa que no se pueden ejecutar otras partes del código hasta que se llegue a la posición indicada. Una vez que se alcance esta posición se puede seguir con la ejecución de más código.

Fig. 52. Diagrama de bloques - PULSOS_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Fig. 53. Flujo grama - PULSOS_SERIAL



Fuente: Autor

3.3.2.8. API Serial – Leer Encoder

El deslizador cuenta con un encoder de cuadratura, este encoder se lo puede utilizar de dos maneras, para llevar un conteo de la posición relativa del deslizador y para obtener una medición instantánea de la velocidad rotacional del deslizador.

Se dice que la posición del deslizador es relativa ya que lo que hace el microcontrolador es llevar un conteo de los pulsos generados por el encoder a partir del momento en que el microcontrolador se enciende. La posición inicial del carro deslizador se la toma como cero, y se lleva un registro de los movimientos realizados por el deslizador a partir de esta posición, la cual es desconocida para el microcontrolador.

Para solucionar este problema se creó una subrutina “home_serial.vi” La cual veremos más adelante la cual hace que el carro del deslizador vaya hasta una posición conocida, la posición inicial u hogar del deslizador.

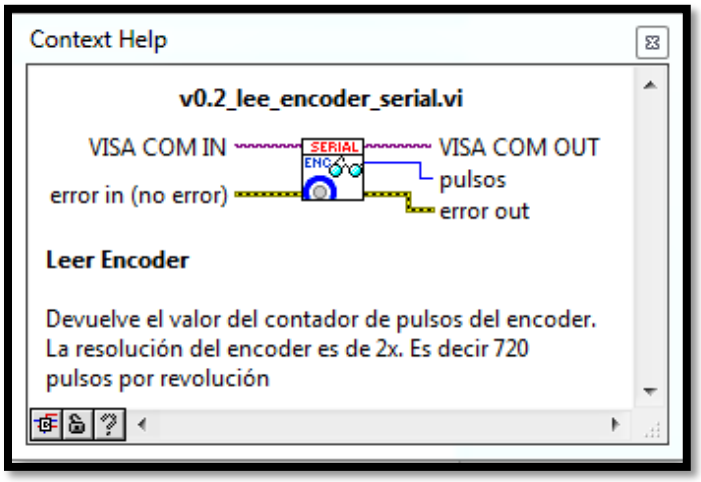
El encoder utilizado cuenta con una precisión de 1440 pulsos por revolución, es decir, si el contador del encoder aumenta su conteo en 1440, entonces el tornillo ha girado una revolución desplazando el carro del deslizador cinco milímetros hacia el extremo del encoder. Si por el contrario el conteo del encoder disminuye en 1440 quiere decir que se ha desplazado cinco milímetros hacía el extremo del servomotor.

El valor máximo que puede ser devuelto por el encoder es el de una variable entera de 32 bits, es decir, 2,147,483,647. Con ese valor se puede cubrir una distancia de 7 Km.

Para poder leer este registro contador del encoder se emplea la siguiente subrutina. Es importante tomar en cuenta que esta subrutina no devuelve los caracteres [OK], como si lo hacen la mayoría de las subrutinas, sino que devuelve el valor del encoder.

La resolución actual de 1440 (ppr), permite el desplazamiento de 0,0014 (mm) la cual es superior a la propuesta en el alcance de la tesis la cual es de 0,01 (mm).

Fig. 54. Visualización ENCODER_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- VISA COM IN: El puerto de comunicación configurado.
- Error in: Clúster de error de entrada

Salidas.-

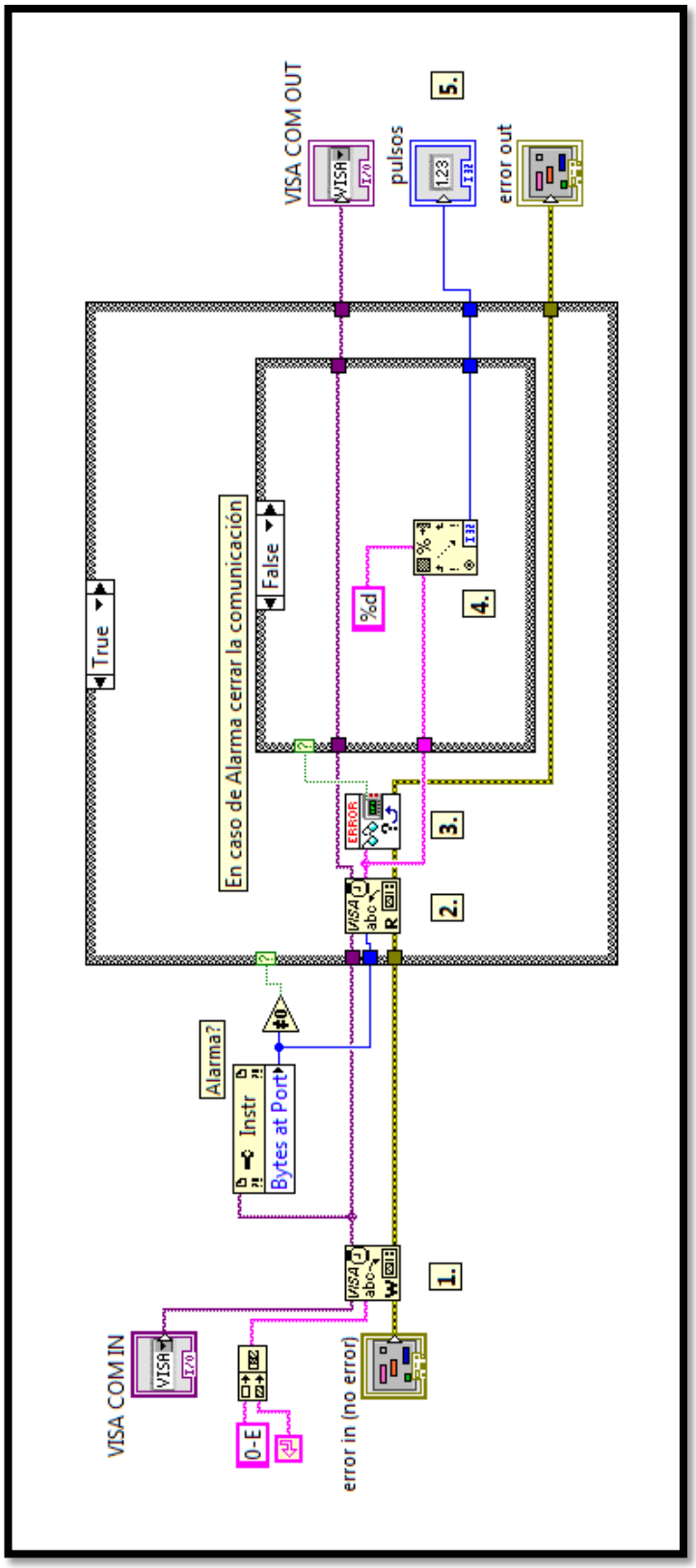
- VISA COM OUT: El puerto serial configurado.
- Pulsos.- Valor del registro contador del encoder
- Error out: Clúster de error de salida

Es importante tomar en cuenta que el valor devuelto es el valor instantáneo del registro, para aplicaciones de precisión se debe considerar el tiempo que toma la comunicación, pueden ser microsegundos, sin embargo es posible que puedan inducir errores.

En el caso de utilizar la comunicación serial, se recomienda que la velocidad de comunicación sea de 115200 baudios, para minimizar la latencia de la comunicación, y obtener mediciones más precisas.

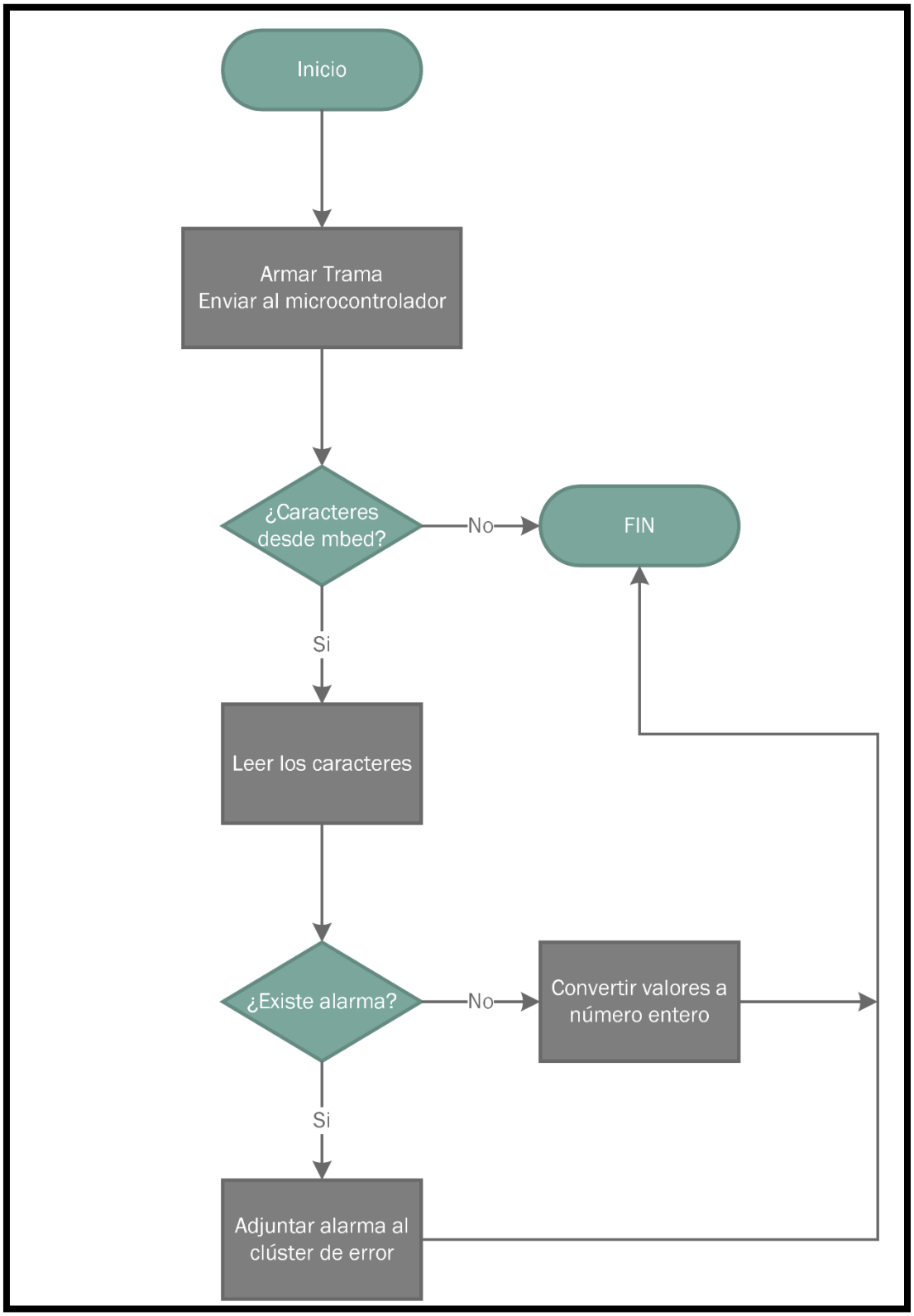
La vibración del servomotor provoca que a momentos se produzcan lecturas erróneas de la velocidad, la velocidad de actualización en el servodrive es menor a la leída por el microcontrolador, lo cual puede provocar variaciones pequeñas en la velocidad medida.

Fig. 55. Diagrama de bloques - LEER_ENCODER



Fuente: Autor (Software: LabVIEW 2013)

Fig. 56. Flujo grama - LEER_ENCODER

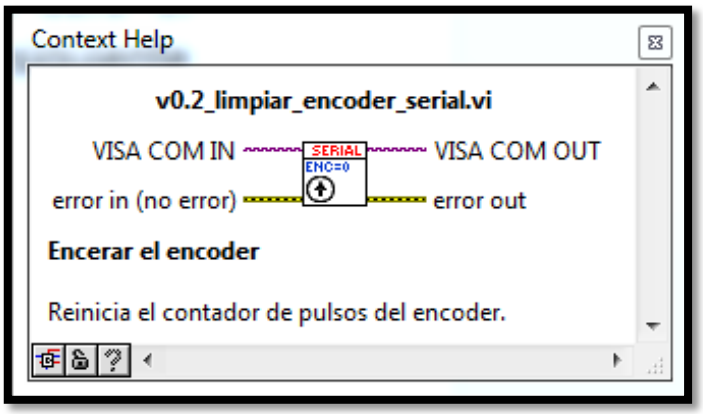


Fuente: Autor (Software: LabVIEW 2013)

3.3.2.9. API Serial – Limpiar Encoder

Es posible que para ciertas aplicaciones sea necesario volver a cero el contador del encoder, en tales circunstancias se puede usar la siguiente subrutina.

Fig. 57. Visualización LIMPIAR_ENCODER



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- VISA COM IN: El puerto de comunicación configurado.
- Error in: Clúster de error de entrada

Salidas.-

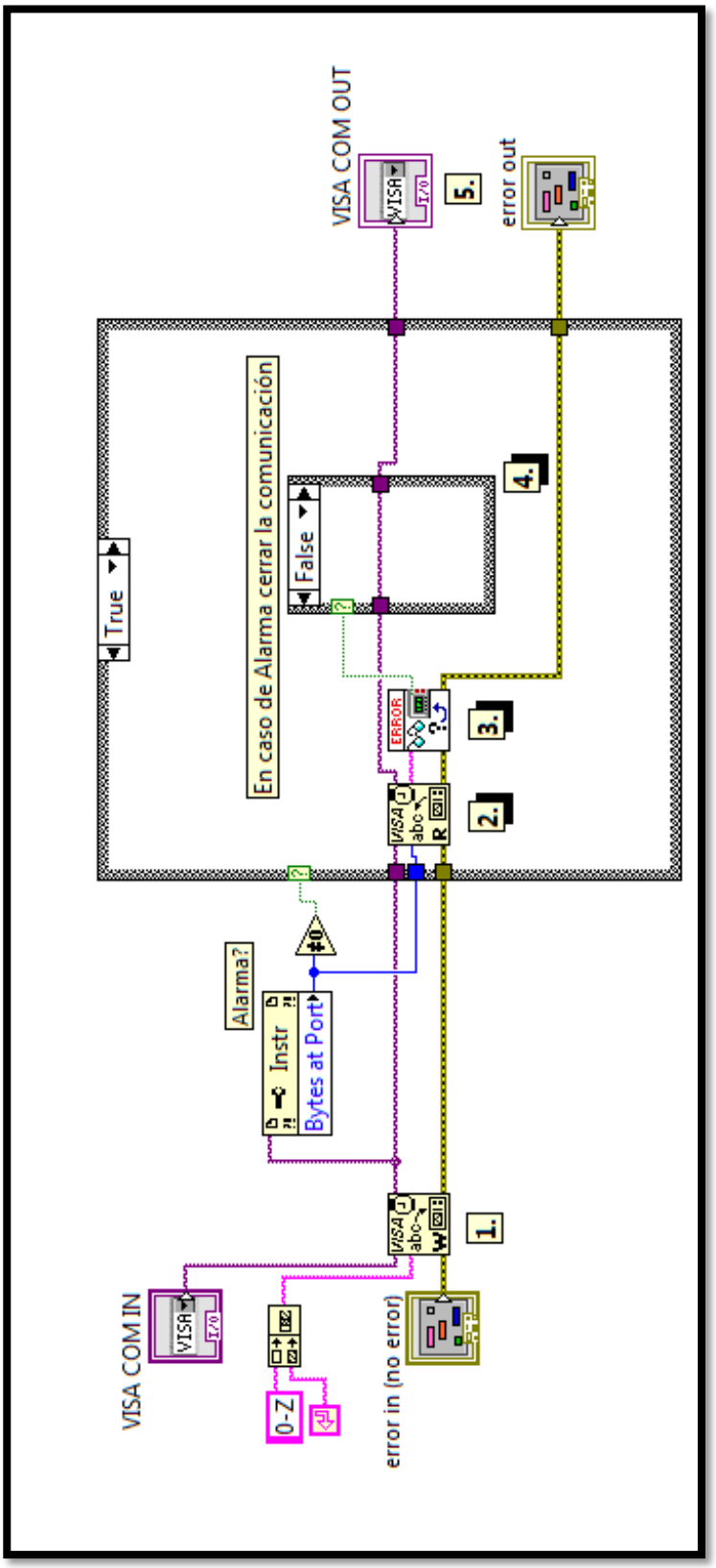
- VISA COM OUT: El puerto serial configurado.
- Error out: Clúster de error de salida

Esta subrutina no toma ningún argumento, el hecho de ejecutarla es suficiente para que el registro del conteo de pulsos desde el encoder se vuelva a cero.

Si el valor leído luego de haber ejecutado esta subrutina es negativo, quiere decir que el carro deslizador se encuentra más cerca al extremo del servomotor.

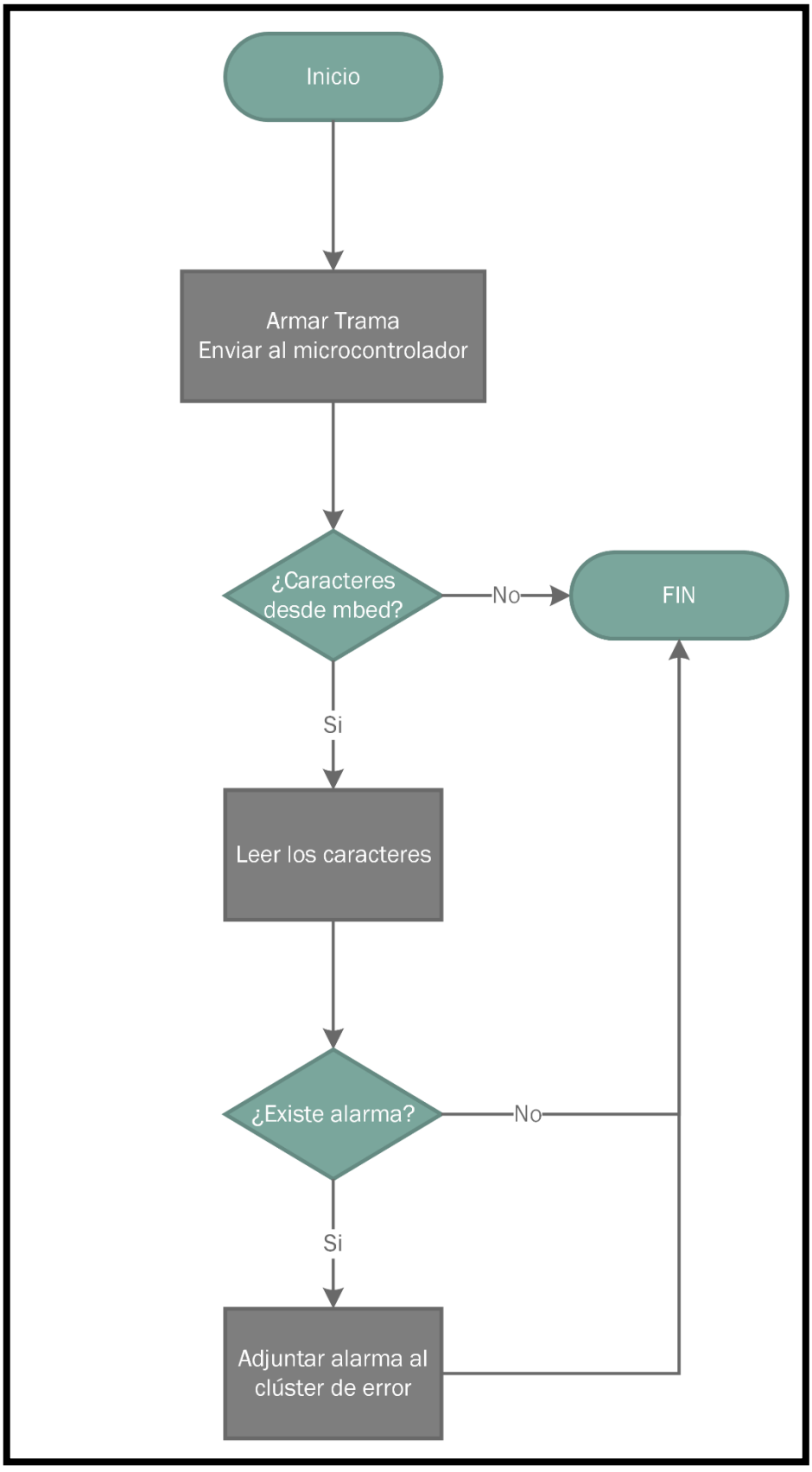
Existe una condición adicional en la cual el contador del encoder puede volver a cero, es en el caso de que se desee llevar el carro del encoder a la posición más cercana posible hacia el extremo del servomotor, esta condición especial es analizada más adelante en la sección Ir a Inicio.

Fig. 58. Diagrama de bloques - LIMPIAR_ENCODER



Fuente: Autor (Software: LabVIEW 2013)

Fig. 59. Diagrama de bloques - LIMPIAR_ENCODER

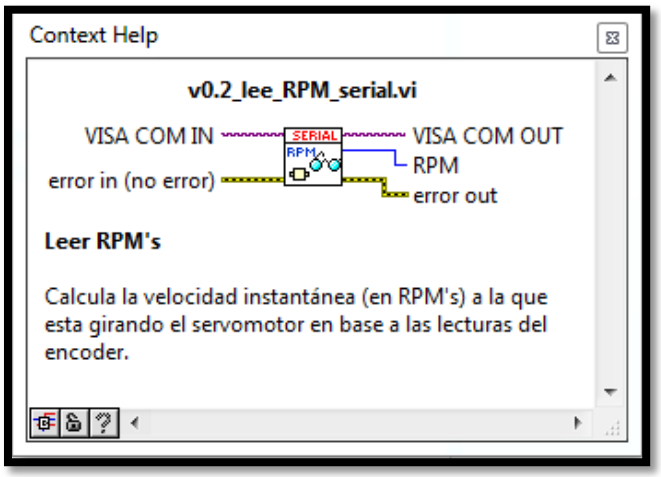


Fuente: Autor

3.3.2.10. API Serial – Leer RPM.

Dentro de las capacidades del microcontrolador se encuentra la de calcular la velocidad instantánea en revoluciones por minuto, para poder consultar esta velocidad se utiliza la siguiente subrutina.

Fig. 60. Visualización LEE_RPM



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

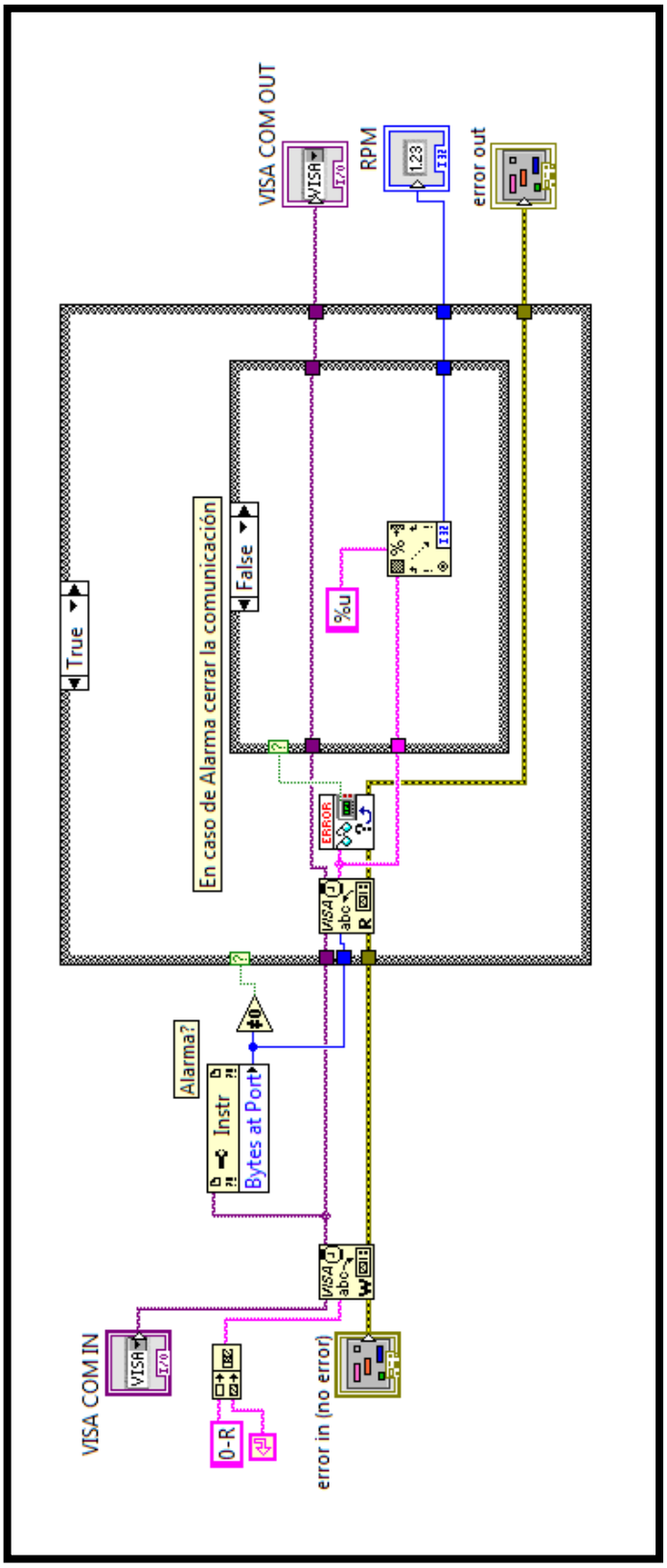
- VISA COM IN: El puerto de comunicación configurado.
- Error in: Clúster de error de entrada

Salidas.-

- VISA COM OUT: El puerto serial configurado.
- RPM: Velocidad instantánea en RPM's.
- Error out: Clúster de error de salida.

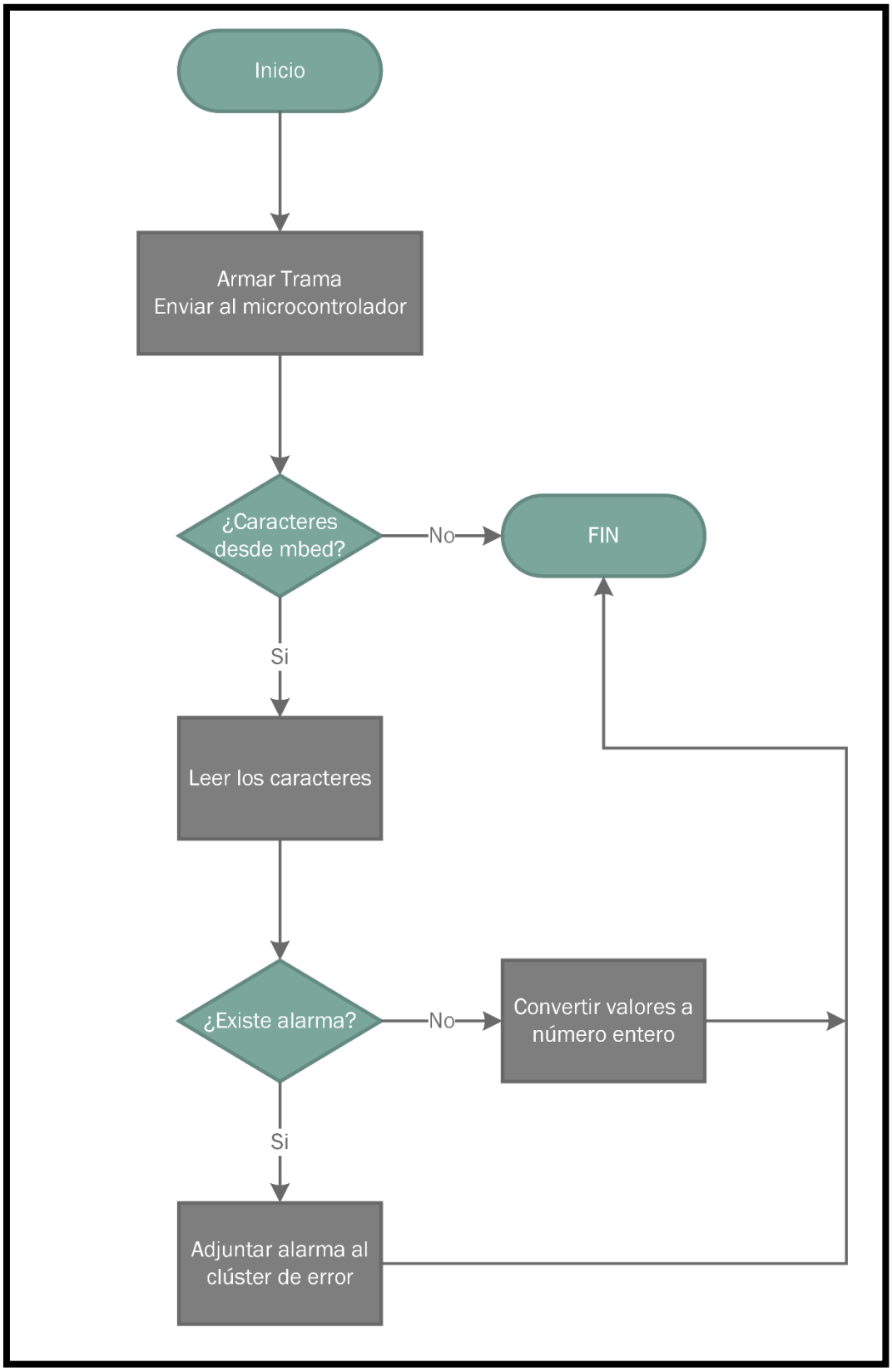
El microcontrolador cuenta con un temporizador específico para este fin, el mismo compara el registro los valores del registro del encoder cada micro segundo, luego toma en cuenta la resolución del encoder, en este caso 360 ppr y de esta manera puede calcular la velocidad rotacional del eje. En el caso de reemplazarse el encoder por uno que tenga una resolución diferente, ya sea mayor o menor, será necesario cambiar la programación del microcontrolador para que tome en cuenta la resolución del encoder en sus cálculos.

Fig. 61. Diagrama de bloques - LEER_RPM



Fuente: Autor (Software: LabVIEW 2013)

Fig. 62. Flujo grama - LEER_RPM

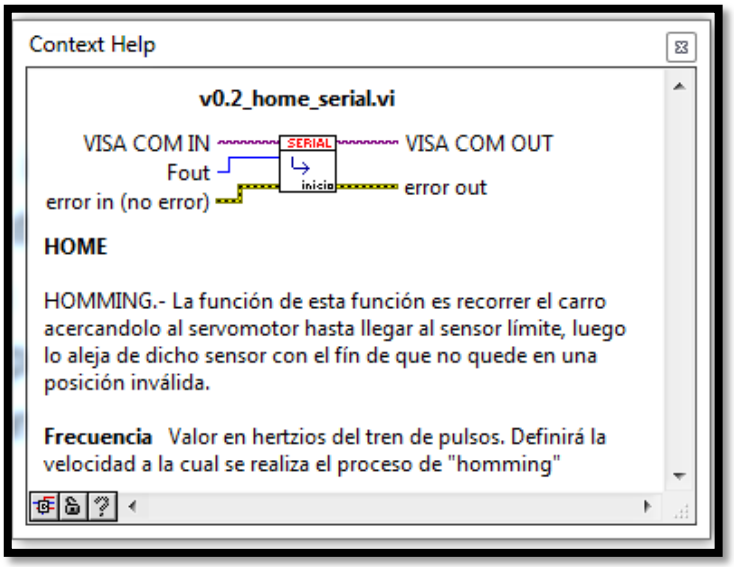


Fuente: Autor

3.3.2.11. API Serial – Ir a Inicio

Para poder utilizar el potencial del registro del encoder es importante tener un punto de referencia, un punto inicial el cual pueda ser definido como cero. Este punto se define en la posición en la cual el carro del deslizador se encuentra lo más próximo al sensor de proximidad del extremo del motor sin llegar traspasarlo. Para poder llegar a este punto de referencia se elaboró la siguiente subrutina.

Fig. 63. Visualización HOME_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- VISA COM IN: El puerto de comunicación configurado.
- Fout: Frecuencia del tren de pulsos para ir a inicio.
- Error in: Clúster de error de entrada

Salidas.-

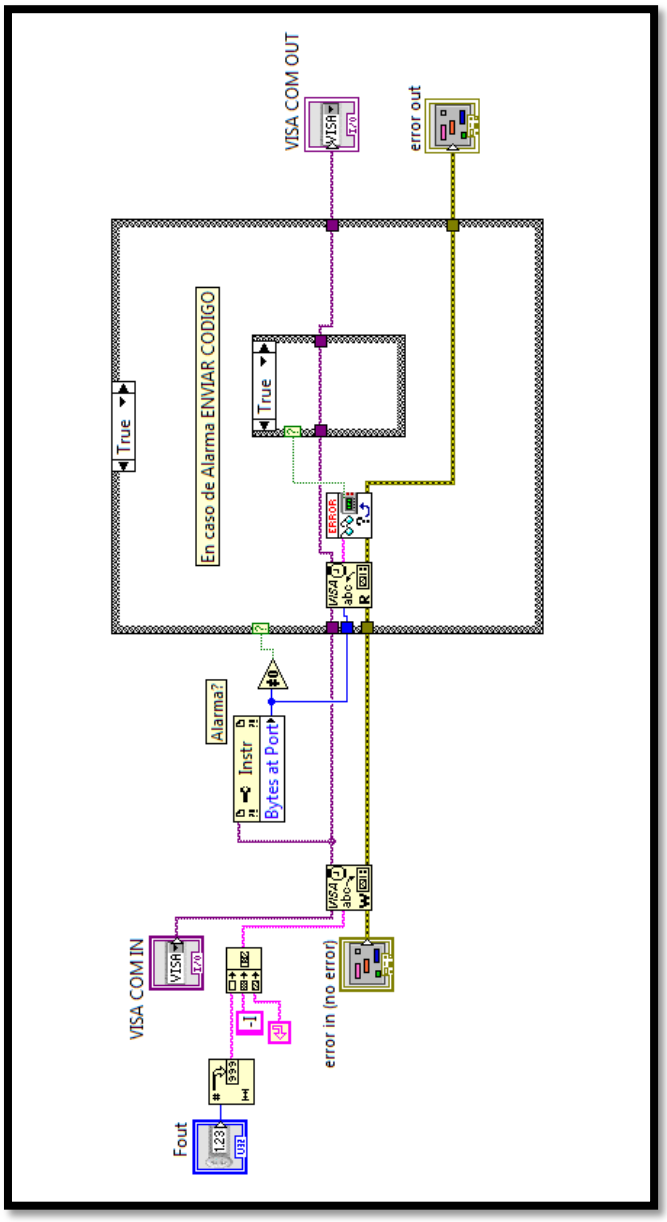
- VISA COM OUT: El puerto serial configurado.
- Error out: Clúster de error de salida

La velocidad a la que se irá hacia esta posición de inicio dependerá de la velocidad configurada en la variable de entrada ‘Fout’.

Es importante el comportamiento que tendrá dicho comando al momento de ser ejecutado. Una vez que se envié el comando el microcontrolador responderá

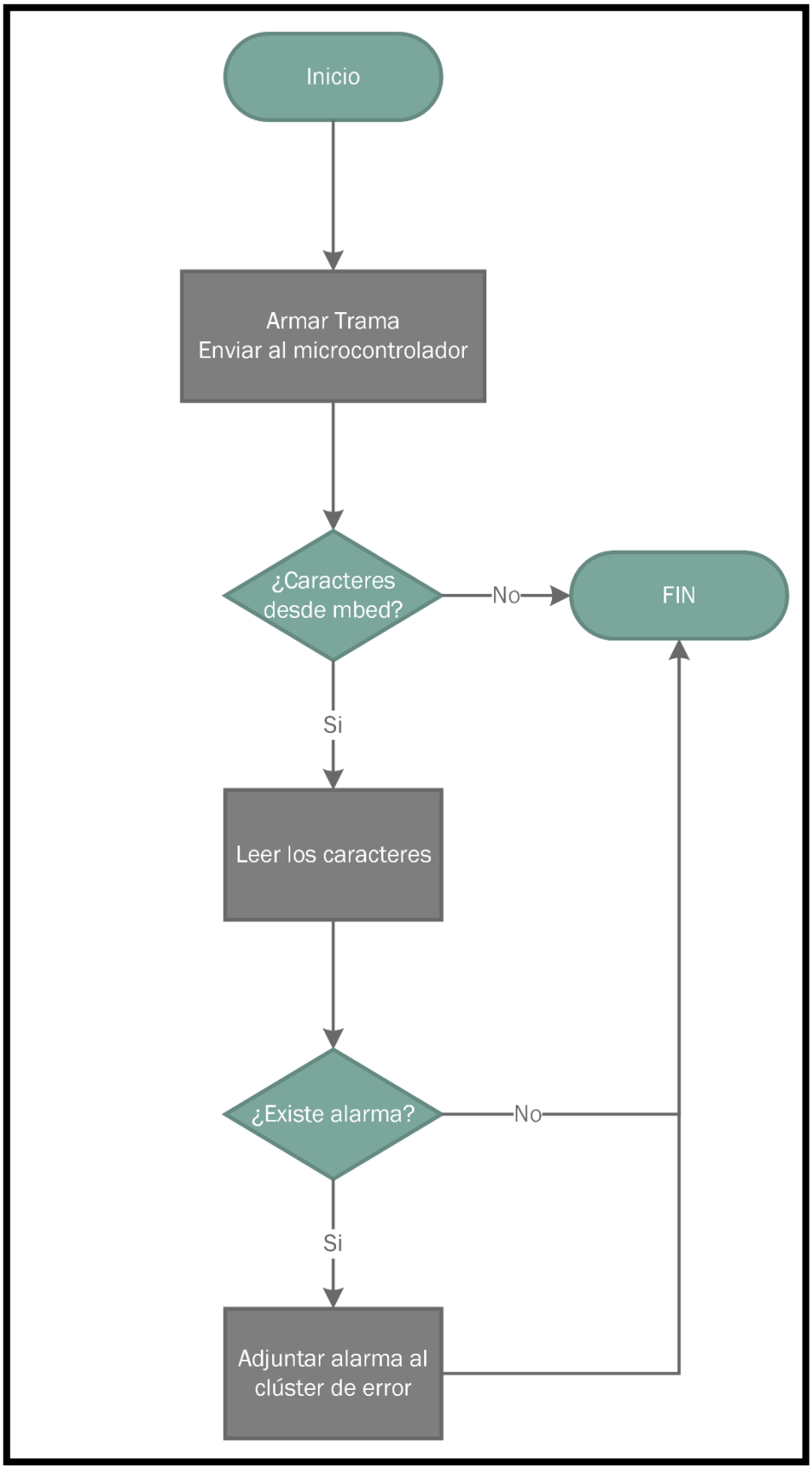
[OK], y empezara el proceso de regresar a su posición inicial a la frecuencia indicada. Una vez que se encuentre en su posición inicial devolverá los caracteres [IN], indicando que se encuentra al inicio del recorrido. En cualquier punto durante este trayecto se puede detener el proceso enviando cualquier comando que sea aceptado por el microcontrolador, por ejemplo el comando [0-H], el cual detendrá el avance del carro deslizador, el microcontrolador devolverá los caracteres [OK] y el proceso de volver al inicio se detendrá.

Fig. 64. Diagrama de bloques - HOME_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Fig. 65. Flujo grama - HOME_SERIAL

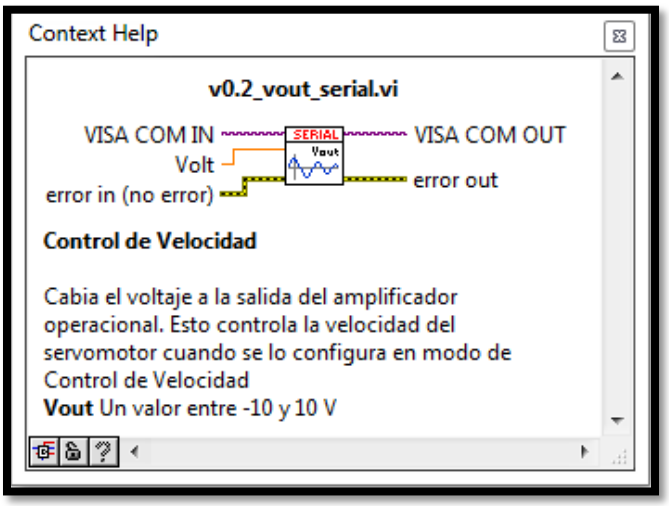


Fuente: Autor

3.3.2.12. API Serial – Voltaje de salida

En su modo de control de velocidad, el servodrive ignorará el tren de pulsos y en cambio su velocidad será determinada por el voltaje en el terminal VCMD. El cual a su vez es controlado con la siguiente subrutina.

Fig. 66. Visualización VOUT_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- VISA COM IN: El puerto de comunicación configurado.
- Volt: Variable tipo flotante con el voltaje a asignar al microcontrolador, puede ser un valor fraccionario, por ejemplo -5.75
- Error in: Clúster de error de entrada

Salidas.-

- VISA COM OUT: El puerto serial configurado.
- Error out: Clúster de error de salida

La variable a modificar representa directamente el voltaje que se espera a la salida del terminal VCMD. El microcontrolador cuenta con un conversor digital a analógico el cual generará el voltaje requerido, sin embargo este voltaje está sujeto a la calibración del amplificador operacional de la placa, específicamente controlado por los potenciómetros de ganancia y offset.

La velocidad final a la salida del servomotor depende del voltaje aplicado y de la configuración de los parámetros del servodrive. Específicamente el parámetro número 23, el cual configura la velocidad máxima a la que puede llegar el servomotor, para las practicas se utiliza un valor de 1000 RPM. Esto quiere decir que cuando el voltaje sea el máximo, diez voltios positivos o negativos el eje tendrá una velocidad de 1000 RPM y cualquier valor intermedio será proporcional, por ejemplo si se tienen cinco voltios la velocidad será de 500 RPM.

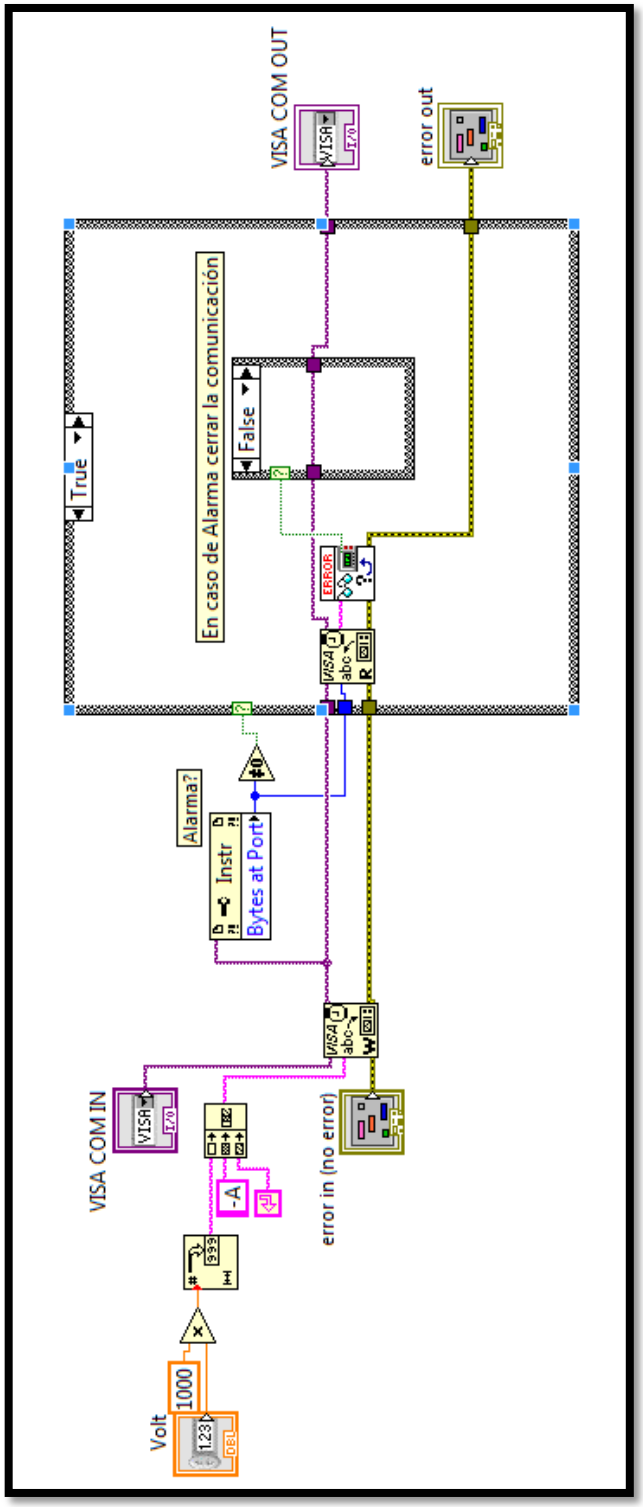
Debido a la longitud de los cables involucrados en el sistema y a la posible interferencia electromagnética existente en el laboratorio de mecatrónica, no se puede garantizar la precisión ni exactitud del voltaje leído por el servodrive. Esta situación no es del todo mala ya que permite que el estudiante realice prácticas de control que corrijan este error, brindando de esta manera la posibilidad de realizar diferentes tipos control sobre la velocidad rotacional del eje del deslizador.

El error depende en gran medida de la calibración del módulo, y es de particular importancia cuando se desea que el servomotor se encuentre detenido. En tal condición el voltaje leído por el servodriver debe ser lo más cercano al cero. Sin embargo esto no siempre es posible, debido en gran medida a que el amplificador instrumental no es un elemento ideal y no podrá llegar a un voltaje de cero absoluto. Este problema se solventa mediante la programación de los parámetros del servomotor, específicamente el parámetro número 33, el cual controla la tolerancia que se dará para considerar un voltaje cero.

Este parámetro se definió para las prácticas con un valor de 250, mediante el cual se puede llegar a un valor de tolerancia de +/- 50mV y el servodrive lo tomara a dicho voltaje como cero, se alienta al estudiante a comprobar y experimentar con este parámetro y su efecto en el servomotor.

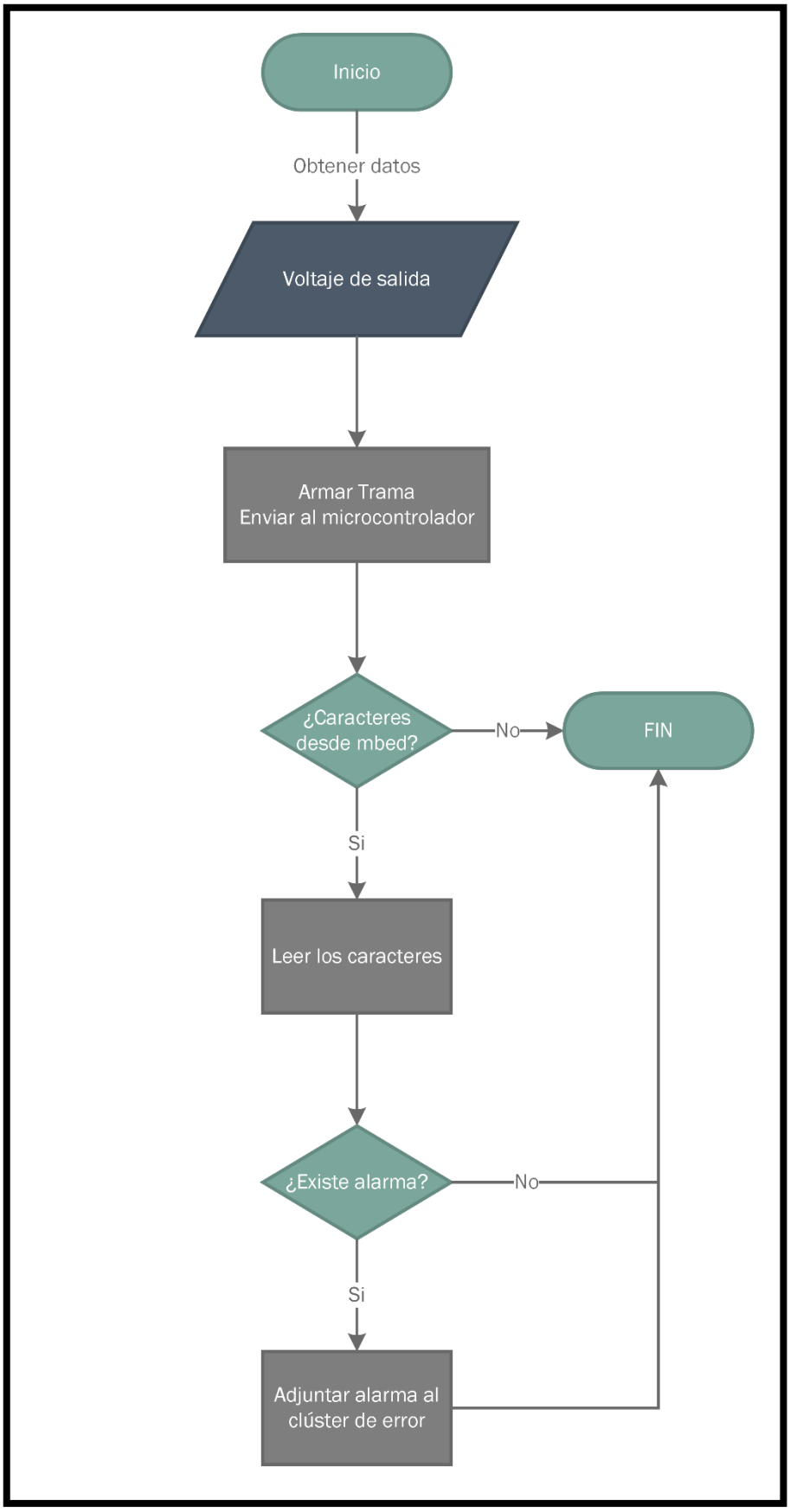
La resolución del circuito es de 4.8mV es decir que se puede aumentar o disminuir el voltaje analógico en pasos de 4.8mV. Esta cantidad está definida por la resolución del conversor digital / análogo del microcontrolador, el cual es de 10 bits, y por la ganancia del amplificador instrumental, la cual puede ser calibrada con el potencio destinado a este fin.

Fig. 67. Diagrama de bloques - VOUT_SERIAL



Fuente: Autor (Software: LabVIEW 2013)

Fig. 68. Flujo grama - VOUT_SERIAL



Fuente: Autor

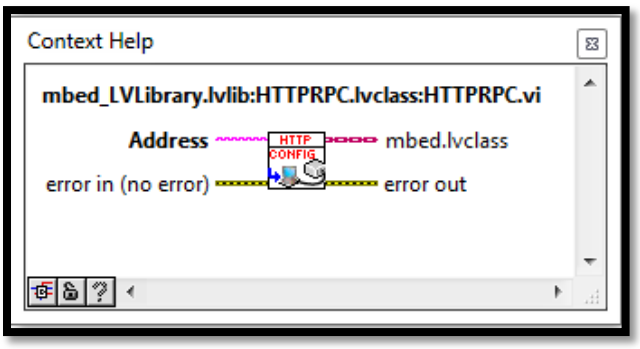
3.3.2.13. API Ethernet – Configuración

La comunicación Ethernet es muy similar a la comunicación serial, en el caso de la comunicación serial se necesita el puerto de comunicación, mientras que en el caso de la comunicación Ethernet se necesita la dirección IP del microcontrolador.

El microcontrolador está configurado para obtener una dirección IP dinámica, este proceso lo realiza cuando se enciende por primera vez el dispositivo, una vez que se asigne la dirección IP y la configuración se haya realizado exitosamente el microcontrolador comunicará su dirección IP a través del puerto USB Serial. Por lo tanto, en un inicio es necesaria la comunicación serial para conocer la IP asignada al microcontrolador. Una vez que se tiene la dirección la comunicación serial no es necesaria.

Una vez conocida esta dirección IP se debe utilizar la siguiente subrutina para configurar la comunicación Ethernet. En caso de ser necesario asignar una IP estática para el microcontrolador su dirección MAC es: **00:02:F7:F0:79:0D**.

Fig. 69. Visualización HTTPRPC



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

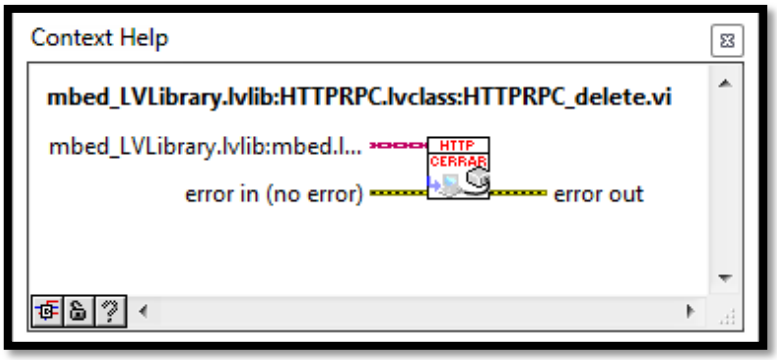
- Address: Dirección IP asignada al microcontrolador, ejemplo: 192.168.1.14
- Error in: Clúster de error de entrada

Salidas.-

- Mbed.lvclass: Clase de Ethernet configurada.
- Error out: Clúster de error de salida.

Una vez que se configura la clase puede ser utilizada de la misma manera que se usa el puerto serial configurado. De la misma forma como se utiliza una subrutina propia de LabVIEW para cerrar la comunicación serial se tiene una subrutina que cierra la comunicación Ethernet, esta debe utilizarse al final del programa.

Fig. 70. Visualización HTTPRPC_DELETE



Fuente: Autor (Software: LabVIEW 2013)

Entradas.-

- Mbed.lvclass: Clase de Ethernet configurada.
- Error in: Clúster de error de entrada

Salidas.-

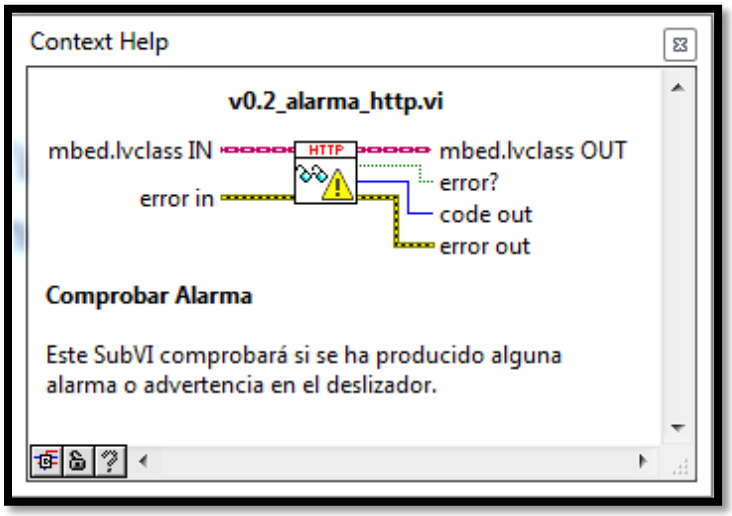
- Error out: Clúster de error de salida.

3.3.2.14. API Ethernet – Manejo de errores

El manejo de errores por parte de la librería Ethernet se lo realiza de la misma forma y con las mismas subrutinas que la comunicación serial.

Existe una diferencia importante entre la comunicación serial y Ethernet la cual consiste en que en la comunicación serial al momento de producirse un error este se informa inmediatamente al programa, mientras que en la comunicación Ethernet la cual implementa un paradigma cliente/servidor, se debe consultar constantemente el estado de las alarmas del deslizador, el microcontrolador no puede arbitrariamente enviar datos al programa en LabVIEW. Para solucionar este inconveniente se elaboró una subrutina la cual se encarga de revisar el estado de las alarmas en el deslizador.

Fig. 71. Visualización ALARMA_HTTP



Fuente: Autor (Software: LabVIEW 2013)

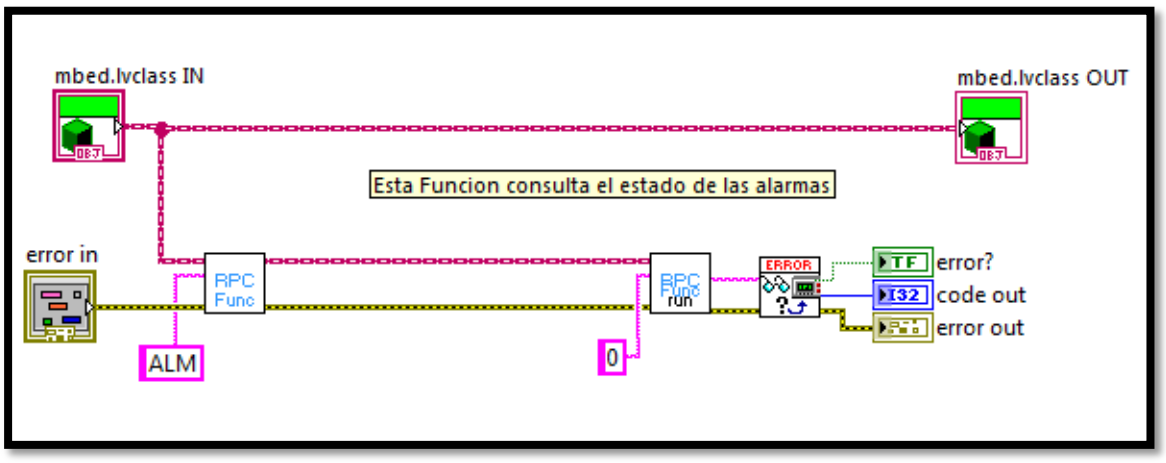
Entradas.-

- Mbed.lvclass: Clase de Ethernet configurada.
- Error in: Clúster de error de entrada

Salidas.-

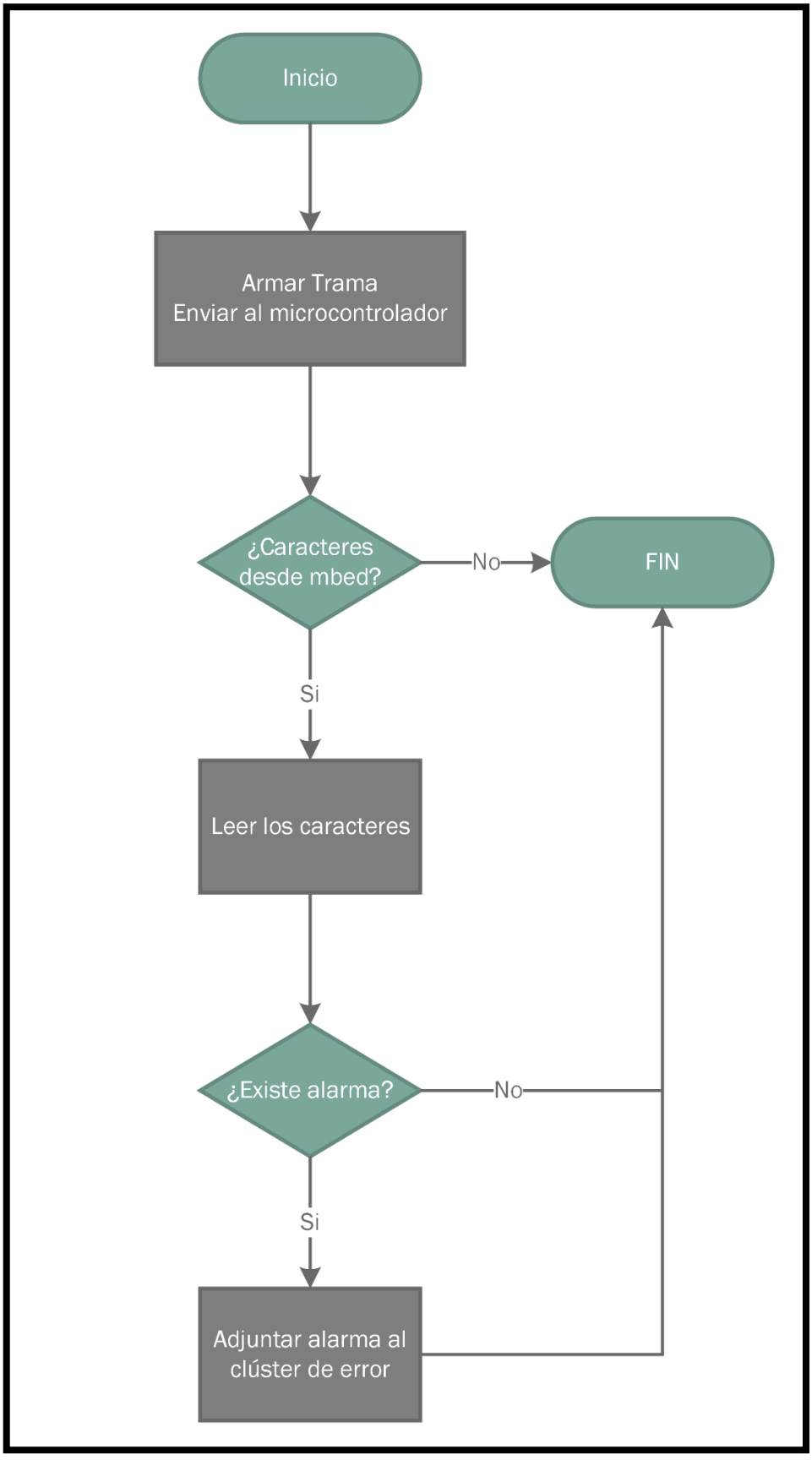
- Mbed.lvclass: Clase de Ethernet configurada.
- Error?: Devuelve verdadero de encontrarse un error en el deslizador.
- Code out: Código del error de haberse encontrado.
- Error out: Clúster de error de salida.

Fig. 72. Diagrama de bloques - ALM_HTTP



Fuente: Autor (Software: LabVIEW 2013)

Fig. 73. Flujo grama - ALM_HTTP



Fuente: Autor

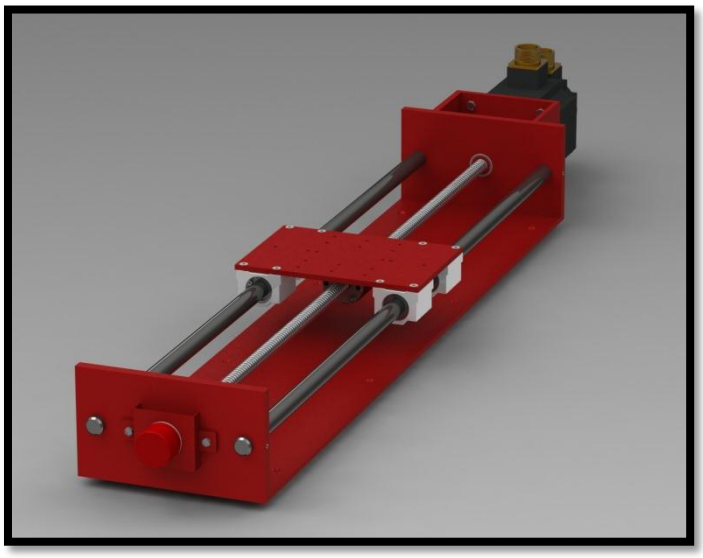
CAPITULO IV
IMPLEMENTACIÓN Y PRUEBAS

Una vez designados y adquiridos los materiales se procedió con el diseño e implementación de la estructura metálica. La elaboración de la misma se realizó en acero negro de seis y ocho milímetros de espesor, se escogió este material debido a su facilidad para trabajarlo y a su disponibilidad en una gran cantidad de talleres mecánicos de la ciudad. Las fuerzas mecánicas a las que estará sujeto el deslizador no son apreciables para justificar un diseño con diferentes materiales.

El diseño cuenta de una base inferior que cubre todo el recorrido del tornillo, a sus extremos se encuentran dos placas de soporte tanto para el extremo del tornillo que llevara el encoder, como para el extremo que se conecta al eje del servomotor. El extremo del servomotor cuenta con una jaula, la misma que permite soporte tanto para el tornillo central y las guías laterales como para el servomotor

Se diseñó que la sujeción de componentes tenga la mayor cantidad posible de uniones mediante pernos y tornillos con la finalidad de que el mantenimiento del módulo sea de fácil realización

Fig. 74. Diseño final del deslizador



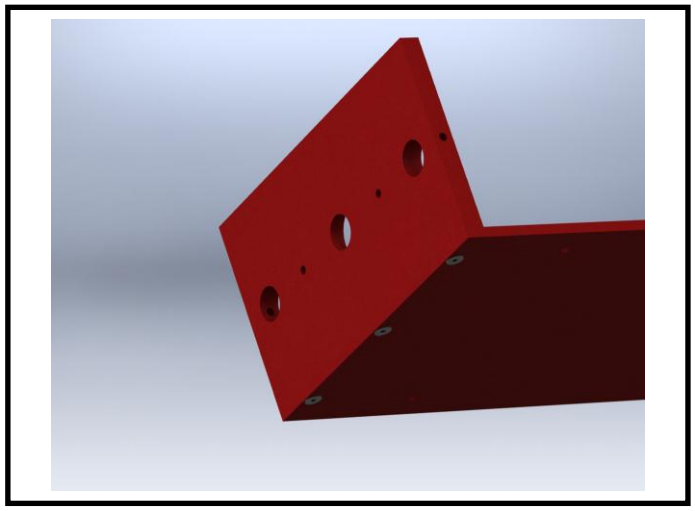
Fuente: Autor (Software: SolidWorks, 2013)

4.1. IMPLEMENTACIÓN FÍSICA DEL MÓDULO

4.1.1. ESTRUCTURA DE SUJECIÓN

Para el armado del módulo se debe empezar por sujetar las placas base a los extremo de la base del deslizador.

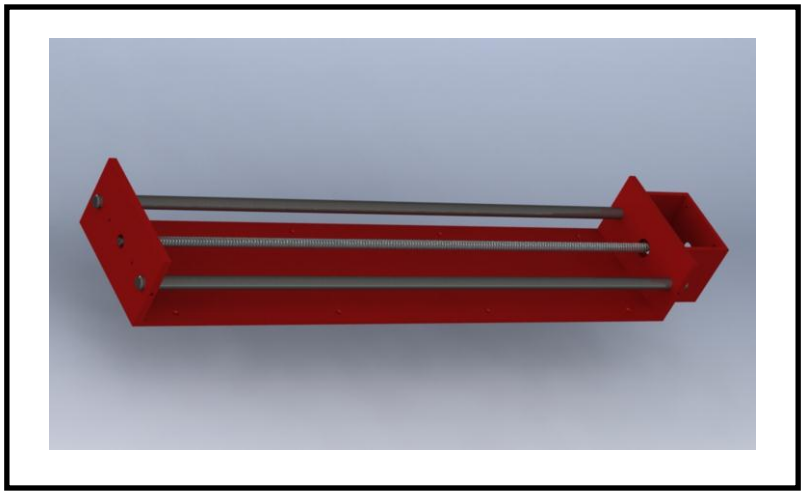
Fig. 75. Sujeción del extremo de la placa de soporte



Fuente: Autor (Software: SolidWorks, 2013)

Se debe sujetar primero un extremo para luego colocar los ejes donde se sujetará la mesa de trabajo del deslizador.

Fig. 76. Placas de soporte con sus ejes

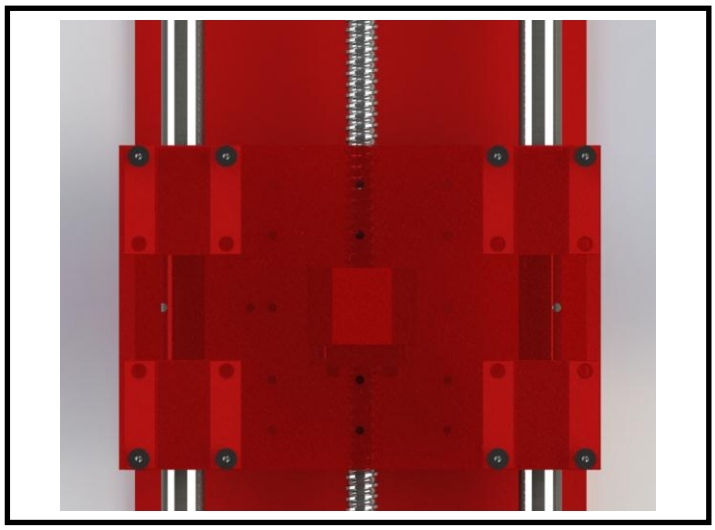


Fuente: Autor (Software: SolidWorks, 2013)

4.1.2. Mesa del deslizador

La parte móvil del deslizador, su base, se asienta sobre las guías laterales mediante los cuatro bloques de rodamientos lineales, se tiene tornillos con cabeza avellanada para su sujeción.

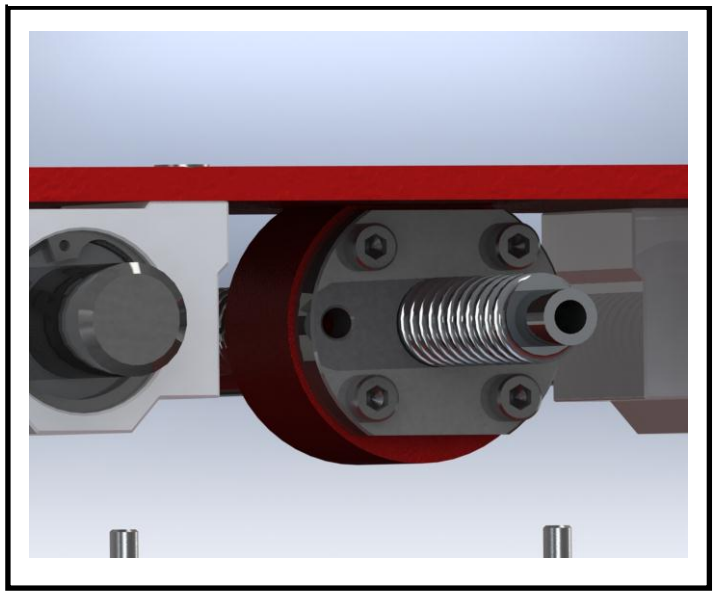
Fig. 77. Sujeción de la mesa del deslizador



Fuente: Autor (Software: SolidWorks, 2013)

Para que la mesa se pueda desplazar se debe sujetar la mesa a la tuerca del tornillo de bolas mediante cuatro tornillos, el resultado es el siguiente

Fig. 78. Sujeción a la tuerca del tornillo de bolas

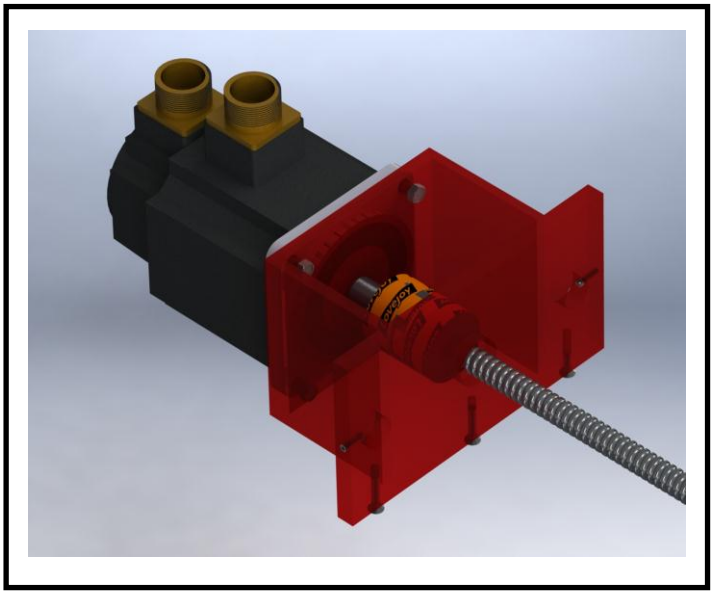


Fuente: Autor (Software: SolidWorks, 2013)

4.1.3. El servomotor

El servomotor se sujeta al deslizador mediante cuatro tornillos, para poder transmitir la potencia se utiliza un matrimonio el cual sirve para sujetar los dos ejes.

Fig. 79. Sujeción del servomotor



Fuente: Autor (Software: SolidWorks, 2013)

Es de vital importancia la correcta sujeción del servomotor ya que de esta depende la alineación de los ejes y la eliminación de vibraciones cuando el motor se revoluciona a altas velocidades.

4.2. CALIBRACIÓN DEL MÓDULO

4.2.1. Calibración Mecánica

Mecánicamente la calibración del módulo se enfoca en errores de posicionamiento de su deslizador, este error se minimizado mediante el uso de un tornillo de bolas re circulantes, este tornillo nos garantiza que el error de posicionamiento este en el orden de los micrómetros, muy por debajo de un error de posición que afecta la aplicación y el trabajo del deslizador bajo el alcance que se tiene. Un detalle importante a tomar en cuenta es garantizar que la sujeción entre el eje del servomotor y el tornillo de bolas se encuentre sujetado con firmeza por el matrimionio, de esto dependerá que la vibración de la máquina sea mínima.

Un posible punto que puede provocar error es la unión entre el eje del motor y del tornillo, se debe asegurar la correcta conexión de estos dos elementos mediante el matrimonio mecánico. Por tal motivo la calibración mecánica se reduce a un correcto acople de éstos dos ejes.

4.2.2. Calibración eléctrica

Como se ha venido indicando a lo largo de este documento, el módulo cuenta con dos modos de funcionamiento, control de posición y velocidad.

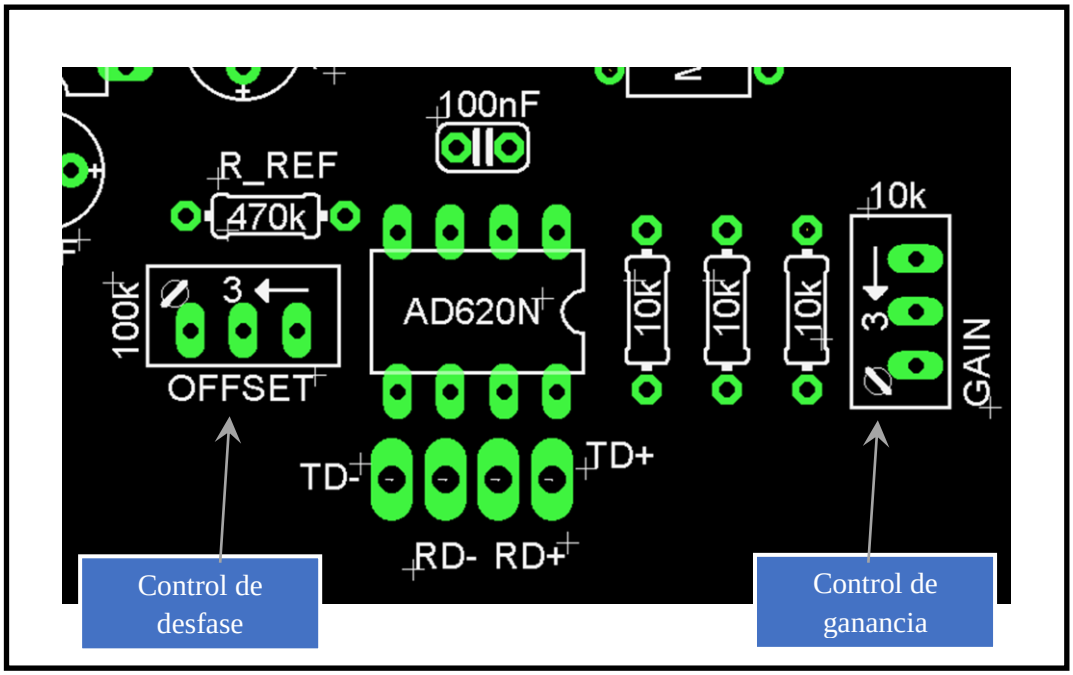
Dentro del control de posición se genera un tren de pulsos, esto quiere decir que es un control de carácter digital, por tal motivo, la generación de estos pulsos depende de la frecuencia a la que se encuentra funcionando el microcontrolador, esta frecuencia y su calibración dependerán exclusivamente del cristal oscilador, dicho cristal se encuentra soldado al microcontrolador por lo tanto no puede ser calibrado, solamente reemplazado.

El segundo modo de funcionamiento, de control de velocidad, depende del voltaje generado por la salida analógica del microcontrolador y de su etapa de amplificación. Es aquí donde es procedente la calibración de esta amplificación, al ser un control analógico es necesario que se calibre de manera adecuada para su correcto funcionamiento. Se diseñó el circuito electrónico para que la calibración se pueda dar una manera adecuada, mediante el uso de dos potenciómetros.

El circuito de amplificación para la salida analógica del microcontrolador cuenta con dos potenciómetros, los cuales controlan la ganancia y el desfase del voltaje de salida.

La calibración se la realiza utilizando un multímetro para poder medir el voltaje de salida. Cuando se alimenta el microcontrolador este tiene una salida hacia el servodrive de cero voltios, se debe ajustar este valor y verificar que es cero voltios midiendo con el multímetro el potencial en los terminales [VCMD] y [AGND] de la placa de conexiones de borneras.

Fig. 80. Ubicación de potenciómetros para calibración



Fuente: Autor (Software Eagle 6.0)

Seguidamente se configura el microcontrolador para una salida de voltaje de 10 Vdc, se utiliza el potenciómetro de ganancia para asegurarse que el voltaje es el correcto.

Se debe repetir estos dos pasos hasta calibrar correctamente el voltaje de salida hacia el servodrive.

4.3. PRUEBAS DE FUNCIONAMIENTO

4.3.1. Pruebas de posición

Para las pruebas de posición del módulo se tomó en cuenta el error de posición del módulo con diferentes cargas, partiendo desde cero hasta los 30 Kg.

El error de posición medido tras un recorrido ida y vuelta del módulo con un recorrido total de 140 cm. La medición se la realizó utilizando un calibrador pie de rey electrónico, el cual cuenta con una precisión de 0.01mm.

La precisión a la que opera el servomotor es de 1/10000 de revolución, tomando en cuenta el paso del tornillo de bolas este se traduce en:

Ecuación 8. Precisión del servomotor

$$Precisión\ Angular = \frac{1}{10000}\ (rev)$$

$$Precisión\ Lineal = \frac{5}{10000} = 0.0005\ (mm)$$

$$Precisión\ Lineal = 0{,}5\ (\mu m)$$

Fuente: Autor

El grado de precisión del tornillo (C7) seleccionado para la distancia de prueba es aproximadamente de 25um, es decir, tomando en cuenta que el recorrido es de ida y vuelta el error inducido por el tornillo es igual a cero.

A continuación se muestra los resultados obtenidos, en un recorrido de 140 cm ida y vuelta de la mesa del deslizador.

Tabla 14. Error lineal medido para el servomotor

N° Prueba	Carga (Kg)	Error medido (mm)
1	0	> 0.01
2	5	> 0.01
3	10	> 0.01
4	15	> 0.01
5	20	> 0.01
6	25	> 0.01
7	30	> 0.01

Fuente: Autor

En todas las pruebas realizadas el error, de existir, fue demasiado pequeño para poder ser medido, esto es de esperarse debido a la gran precisión con la que cuenta el motor.

Además de esta prueba de precisión se realizó una pruebe de la frecuencia máxima del tren de pulsos que es capaz de recibir el servodrive. Se encontró que

la máxima frecuencia leída por el servodrive es de 160 KHz. Una vez que se aumenta la frecuencia a 161 KHz el servodrive no identifica esta frecuencia.

La velocidad máxima que se puede obtener del motor para una precisión de 0.5 um es de 960 Revoluciones por Minuto. Se puede configurar el servomotor para que su precisión sea menor, esto quiere decir que se requieren menos pulsos para completar una revolución, esto se logra a través de los parámetros [PA:12] y [PA:13].

Ecuación 10. Cálculo de la velocidad.

$$F_{MAX} = 160\ (KHz)$$
$$V_{MAX} = \frac{160\ 000}{10\ 000}\left(\frac{rev}{seg}\right)$$
$$V_{MAX} = 960\ (RPM)$$

Fuente: Autor

Se debe mantener un balance, entre la velocidad máxima del servomotor y su precisión. Esto abre la puerta para diferentes configuraciones y prácticas para los estudiantes.

4.3.2. Pruebas de velocidad

Las pruebas de velocidad en el módulo del deslizador están ligadas a la calibración previa que tiene el servomotor, es decir, que si se ha enviado el comando al microcontrolador para que la salida sea de cinco voltios, comprobar que realmente esa es la lectura leída a su salida.

También se debe tomar en cuenta que al ser una señal analógica esta se ve sujeta a interferencias y caídas de potencial. El cable que lleva la señal hasta el servodrive es de tipo apantallado, por tanto las protecciones necesarias para su integridad ya han sido implementadas. La velocidad ha sido medida utilizando el sensor propio del servodrive obteniéndose los siguientes resultados.

Lo que se observa es una relación proporcional entre el voltaje aplicado y la velocidad máxima que puede desarrollar el servomotor.

Tabla 15. Resultados de la prueba de velocidad máxima 2000 RPM

Velocidad máxima en el parámetro 23. (RPM)	Voltaje aplicado (Vdc)	Velocidad medida (RPM)
2000	10.00	2000
2000	5.00	1150
2000	2.00	452
2000	1.00	218
2000	0.00	0
2000	-1.00	-217
2000	-2.00	-450
2000	-5.00	-1150
2000	-10.00	-2000

Tabla 16. Resultados de la prueba de velocidad máxima 1000 RPM

Velocidad máxima en el parámetro 23. (RPM)	Voltaje aplicado (Vdc)	Velocidad medida (RPM)
1000	10.00	1000
1000	5.00	575
1000	2.00	226
1000	1.00	109
1000	0.00	0
1000	-1.00	-110
1000	-2.00	-225
1000	-5.00	-570
1000	-10.00	-1000

Fuente: Autor

CAPITULO V
CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Se implementó un módulo que permite a los estudiantes realizar prácticas tanto de Control como de servomecanismos de una manera fácil, rápida y didáctica.

- El subsistema mecánico cumple con todas las especificaciones requeridas para su funcionamiento. Tanto en su longitud como su precisión.

- Se ha determinó que la velocidad máxima soportable por el mecanismo es de 1500 RPM, una velocidad superior por un periodo prolongado de tiempo puede provocar daños a los rodamientos del deslizador.

- El disponer de un módulo de deslizamiento lineal con una altísima precisión de posicionamiento abre las puertas a muy diversas aplicaciones que los estudiantes pueden investigar.

- Se diseño un sistema que puede ser usado por una amplia gama de estudiantes, no es estrictamente necesario tener conocimientos avanzados en servomecanismos.

- La utilización de un microcontrolador que cuenta con una plataforma de desarrollo permitió una gran agilidad a la hora de su programación, los módulos con los que cuenta se ajustan a la perfección a las necesidades del deslizador.

- Las librerías creadas en el entorno de desarrollo permiten una programación del deslizador que se acople a las diferentes prácticas del deslizador.

- Se logró un diseño robusto adecuado para soportar posibles abusos por parte de los estudiantes en sus experimentos.

- Se dotó de modularidad al servomecanismo, permitiendo futuras ampliaciones que lo doten de mayores grados de libertad.

- Los estudiantes han sido expuestos a nuevas tecnologías de desplazamiento lineal y sus aplicaciones.

- El módulo se implementó con medidas de seguridad que permiten la colisión de la mesa de trabajo con los extremos del deslizador. Además de un botón de paro de emergencia en caso de ser necesario.

5.2. RECOMENDACIONES

- Se deben observar todas las precauciones de seguridad al momento de operar el módulo para asegurar una larga vida útil del mismo.

- Se sugiere la implementación de puesta tierra para todos los equipos del laboratorio de electrónica y mecatrónica.

- Se debe dar el mantenimiento apropiado al módulo para asegurar su buen funcionamiento

- Se debe asegurar el correcto funcionamiento de los mecanismos de seguridad del módulo antes de su utilización.

- El encoder utilizado en el sistema tiene una precisión muy baja comparada a la del motor, esto puede mejorar adquiriendo un modelo actualizado del encoder.

- La programación tanto de la librería de funciones en LabVIEW como la del microcontrolador son de código abierto, se sugiere su estudio y mejoramiento de los algoritmos implementados en el módulo.

- El microcontrolador empleado cuenta con características muy superiores a las de microcontroladores tradicionales, se sugiere explorar estas características.

- La precisión del módulo abre la posibilidad de experimentar con sistemas de control en tiempo real y medir su desempeño, se sugiere la utilización de las capacidades de Visión Artificial, y ejecución en Tiempo Real del lenguaje LabVIEW.

- El módulo puede ser controlado desde Matlab, se sugiere la implementación de lazos de control en este programa como un ejemplo práctico de sus capacidades de programación.

- Para el sistema de control de velocidad en lazo cerrado se sugiere la implementación de diversos tipos de control y su comportamiento

- La mesa del deslizador tiene la capacidad de atornillar diferentes tipos de mecanismos adicionales, se sugiere el montaje de un brazo robótico que permita la manipulación de objetos.

- Se recomienda la adquisición de un juego completo de hexágonos por parte del laboratorio para poder dar mantenimiento al deslizador

- El sistema mecánico está sobredimensionado para que el módulo pueda ser ocupado como parte de un mecanismo aún mayor, se recomienda la implementación de una máquina CNC diseñada íntegramente en la Universidad Técnica del Norte.

BIBLIOGRAFÍA

Beer, F., & Johnston, R. (2010). *MECÁNICA VECTORIAL PARA INGENIEROS.* México: McGraw Hill.

Sharp, R. (2008). *Principles of Protocol Design.* Kongens Lyngby: Springer.

Blanding, D. L. (1995). *Machine Design Using Kinematics Processing.* New York: The American Socierty of Mechanical Engineers.

NSK Corporation. (2005). *What is a ball screw?* Franklin.

Scolum, A. H. (1992). *Precision machine design.* New Jersey: Prentice-Hall, Inc.

GSK CNC Equipment Co. (2007). *DA98D Digital AC Servo Drive Unit.* Guanzhou.

Hihglights. (2011). *CONVERSIÓN DE SEÑAL DE ENCODER DE CUADRATURA* . Cuenca, Ecuador.

León Aguirre, A., & Tapia Vaca, J. (2009). *Sincronismo y supervisión de posición y velocidad de un motor trifásico asíncrono con un sistema de encoders maestro-esclavo para aplicaciones de control de ejes.* Latacunga: Escuela Politécnica del Ejercito.

SKF Group. (20 de Enero de 2014). *www.skf.com.* Obtenido de http://www.skf.com/binary/12-55835/SKF4182_EN_Linear-ball-bearings.pdf

THK. (2010). *Guide Ball Bushing / Linear Bushing.* Agoura Hills: THK.

Hohner Corporation. (20 de Enero de 2014). *www.encoderonline.com.* Obtenido de http://www.encoderonline.com/Devicenet/Data-Sheets/Incremental/Data-33.htm

Llano, I. B. (20 de Enero de 2014). *https://www.icai.es.* Obtenido de https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCQQFjAA&url=https%3A%2F%2Fwww.icai.es%2Fpublicaciones%2Fanales_get.php%3Fid%3D1428&ei=w2XwUuWWGMKnkQeA_YGwDQ&usg=AFQjCNFPcZSb_WdVmMRb2yRTsXo4tinDTA&sig2=vX9HzJSU2MdaQ_AlCjTN9

Márquez Días, J., Pardo Sánchez, K., & Pizarro Valencia, S. (10 de Enero de 2014). *http://rcientificas.uninorte.edu.co/.* Obtenido de http://rcientificas.uninorte.edu.co/index.php/ingenieria/article/view/2272/1484

Microsoft. (24 de 11 de 2013). *http://technet.microsoft.com/.* Obtenido de http://technet.microsoft.com/en-us/library/cc759499(v=ws.10)

Zuñiga Tufiño, M. A. (2008). *ESTUDIO DISEÑO Y SIMULACIÓN DE UN SISTEMA DE*. Quito: ESCUELA POLITECNICA NACIONAL.

Aerotech. (20 de Enero de 2014). *www.aerotech.com*. Obtenido de http://www.aerotech.com/media/247221/section16_motor%20selection%20and%20sizing.pdf

ARM Holdings. (04 de Enero de 2014). *mbed*. Obtenido de http://www.mbed.org

ANEXO 1. MANUAL DE OPERACIONES

Deslizador Lineal UTN
Manual de operación

1. Generalidades

El presente documento tiene la finalidad de ayudar al estudiante a conocer y comprender el correcto manejo del sistema mecánico y electrónico del deslizador, y los pasos que debe seguir para su correcto mantenimiento.

Este documento complementa y extiende lo visto en los capítulos dos y tres del trabajo de grado que acompaña este anexo, para una comprensión completa de lo aquí expuesto se recomienda la lectura de estos capítulos.

2. Armado y desarmado del módulo

La estructura del sistema mecánico se encuentra dividida en tres partes, se tiene la estructura de soporte, luego tenemos la transmisión de potencia, y finalmente el carro deslizador.

La estructura de soporte es fácilmente reconocible ya que se encuentra pintada de rojo. Esta estructura se encuentra ensamblada mediante tornillos lo que permite poder desarmarla y cambiar componentes de ser necesario.

Los elementos de transmisión de potencia son, el tornillo de bolas, el matrimonio que acopla los ejes, y los rodamientos radiales que sostienen el tornillo. Estos elementos son los encargados de transmitir la potencia al carro deslizador.

El carro del deslizador se asienta sobre los rodamientos lineales, los cuales se deslizan a lo largo de sus guías quienes son las encargadas de soportar el la carga a la que se somete el deslizador.

El servomotor es el elemento actuador, se sujeta al tornillo a un extremo del mismo mientras que en el extremo opuesto se encuentra conectado el encoder de cuadratura.

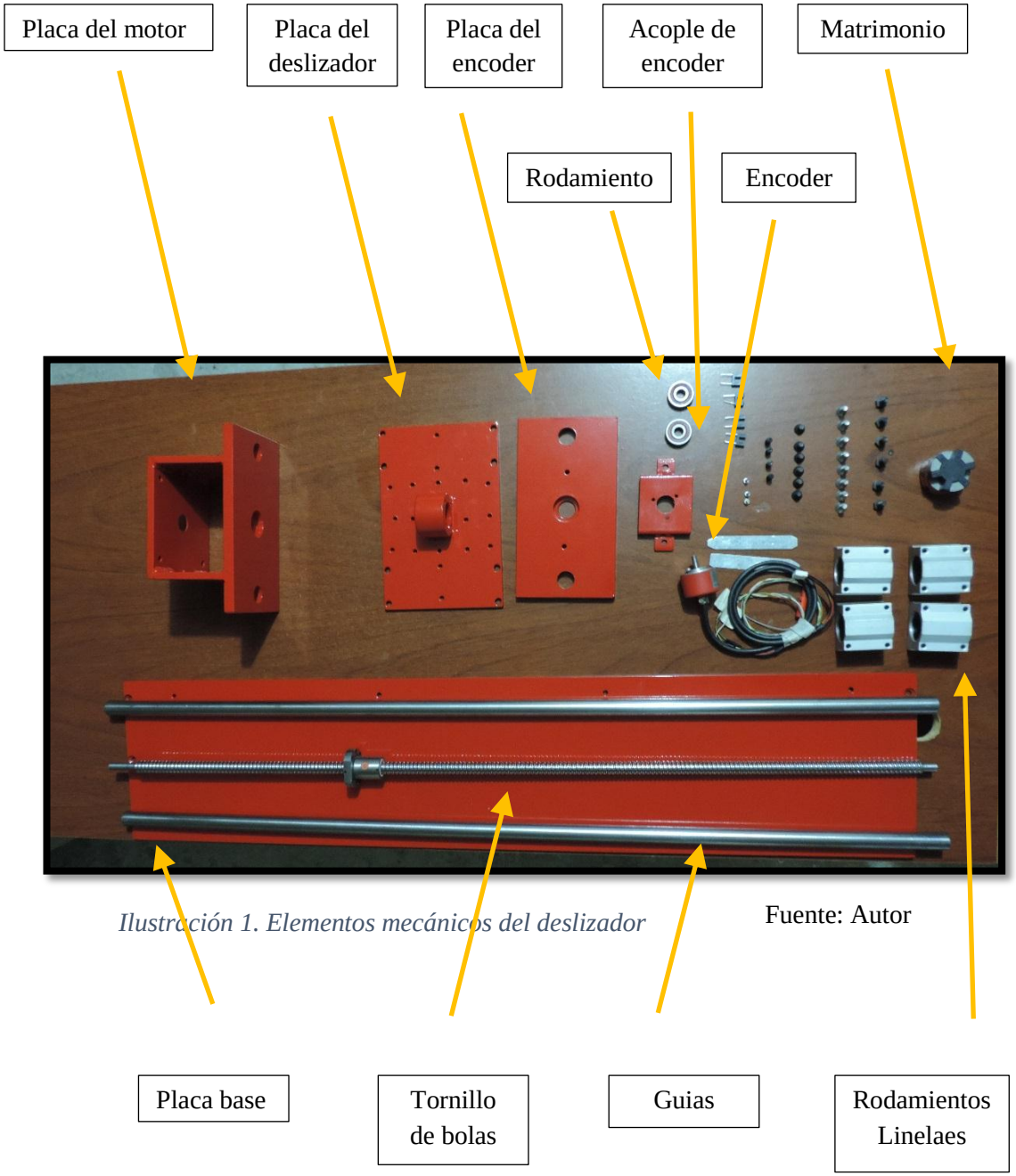


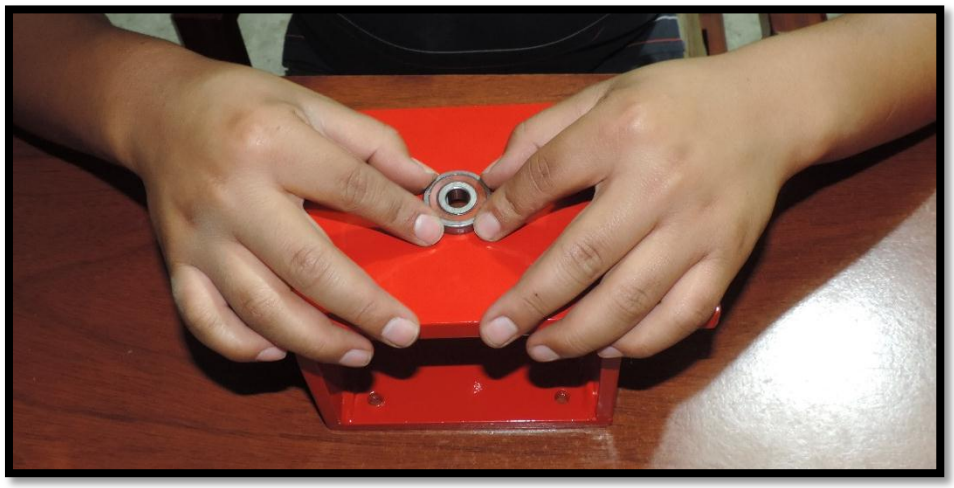
Ilustración 1. Elementos mecánicos del deslizador Fuente: Autor

A continuación se explicara detalladamente cada uno de estos componentes.

2.1. Estructura del deslizador

Las placas del deslizador cuentan en su centro con una abertura donde se coloca el rodamiento radial que se sujetará al tornillo de bolas. En caso de ser necesario se puede cambiar los mismos.

Ilustración 2. Rodamiento radial

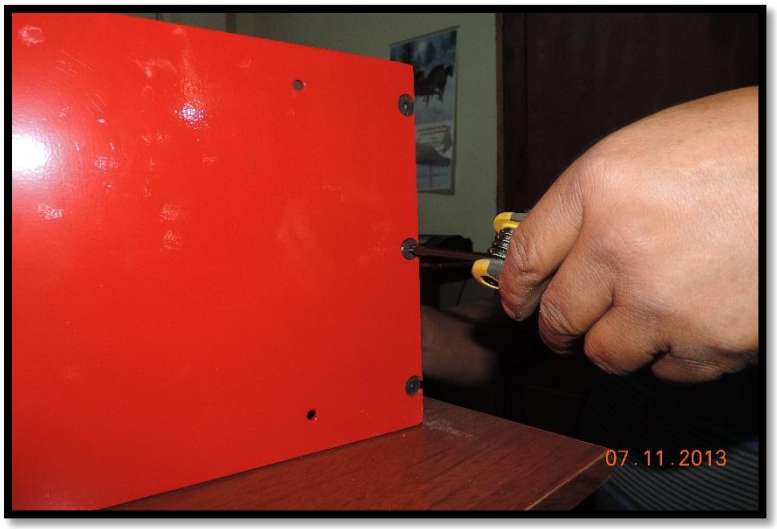


Fuente: Autor

Los rodamientos deben tener un diámetro interno de 10 mm y uno externo de 30 mm.

La base del deslizador es una placa de 970 x 210 x 6 milímetros, en sus extremos cuenta con perforaciones que permiten sujetar la placa del encoder en un extremo y la del motor en el otro.

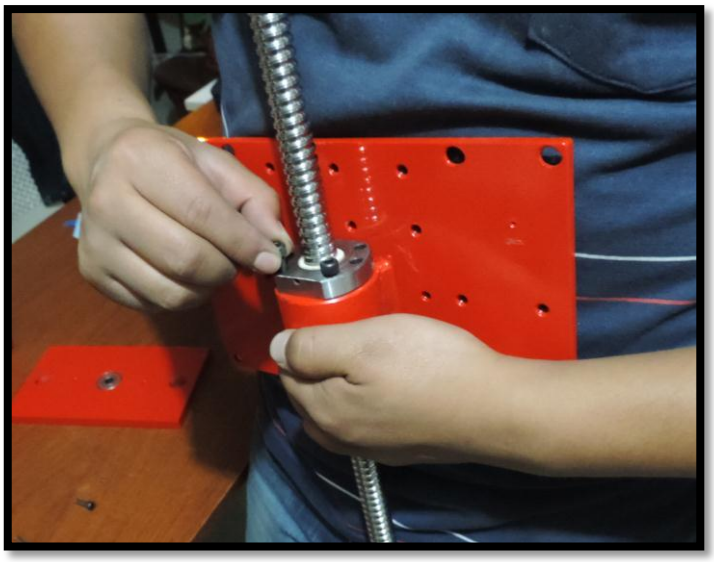
Ilustración 3. Sujeción de la placa base



Fuente: Autor

La sujeción se la realiza con tornillos avellanados de ¾''. Primero se debe sujetar la placa del deslizador al tornillo, luego se ajustan las guías y el tornillo central.

Ilustración 4. Sujeción de la placa del deslizador.



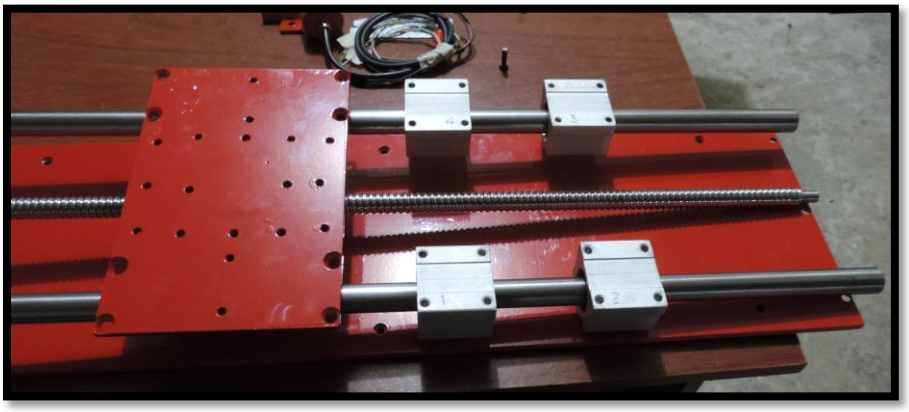
Fuente: Autor

Ilustración 5. Inserción de las guías

Fuente: Autor

Seguidamente se colocan los rodamientos lineales en las guías, los mismos tienen una numeración como se muestra en la imagen para que el ajuste sea el óptimo.

Ilustración 6. Insertar los rodamientos lineales



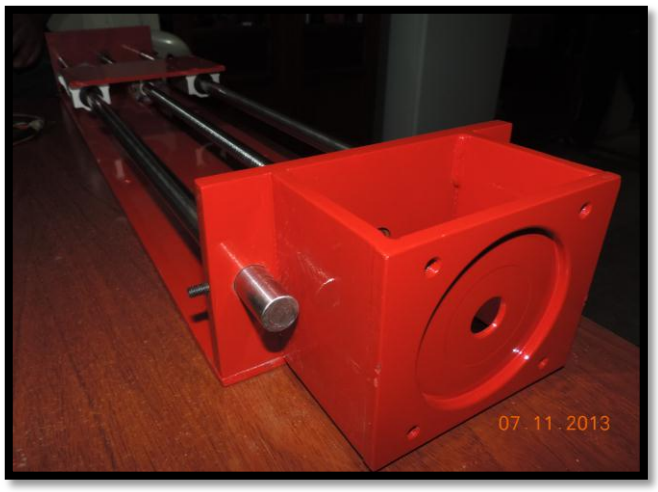
Fuente: Autor

Ilustración 7. Numeración de los rodamientos.

Fuente: Autor

Para poder asegurar las guías en su lugar se tiene prisioneros de 1/8’’

Ilustración 8.Seguro de la guía



Fuente: Autor

2.2. Tornillo central y base

El tornillo de bolas está compuesto por un tornillo central y su tuerca, ambos elementos de precisión, entre la tuerca y el tornillo existen pequeñas bolas las cuales hacen que el movimiento sea suave y preciso.

IMPORTANTE:

No se debe llevar la tuerca hasta el final del tornillo con la intención de sacarla, esto hará que las bolas se caigan del tornillo.

Ilustración 9. Tornillo de bolas

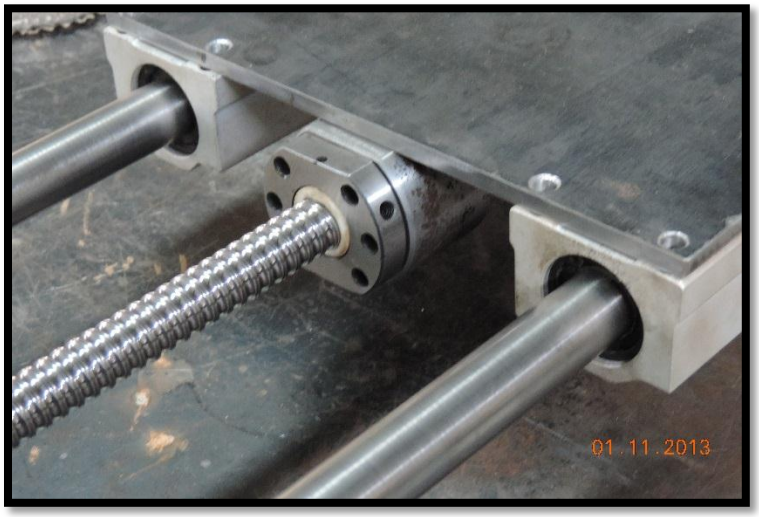
Fuente: Autor

En uno de sus extremos el tornillo de bolas cuenta con una perforación axial, esta sirve para que la espiga del encoder sea insertada y de esta forma pueda girar solidario al tornillo.

2.3. Guías y rodamientos

Las guías de los rodamientos son de acero extra duro templado y pulido. La superficie de la guía se sometió a un proceso de endurecimiento, esto con la finalidad de prolongar la vida útil de los rodamientos lineales.

Ilustración 10. Tornillos y carro deslizador.



Fuente: Autor

Es importante mantener la lubricación en las guías como en el tornillo para prolongar la vida útil de los componentes.

Los rodamientos lineales se componen de los rodamientos LBBR20 SKF, los cuales internamente también cuentan con bolas re circulantes para disminuir la fricción de los mismos.

Ilustración 11. Rodamientos SKF LBBR 20

Fuente: (SKF Group, 2014)

Estos rodamientos se encuentran dentro de sus carcasas de aluminio, las cuales facilitan la sujeción de la placa del deslizador, estas se realizan mediante tornillos avellanados de 1/8’’

Ilustración 12. Rodamientos en sus carcasas.

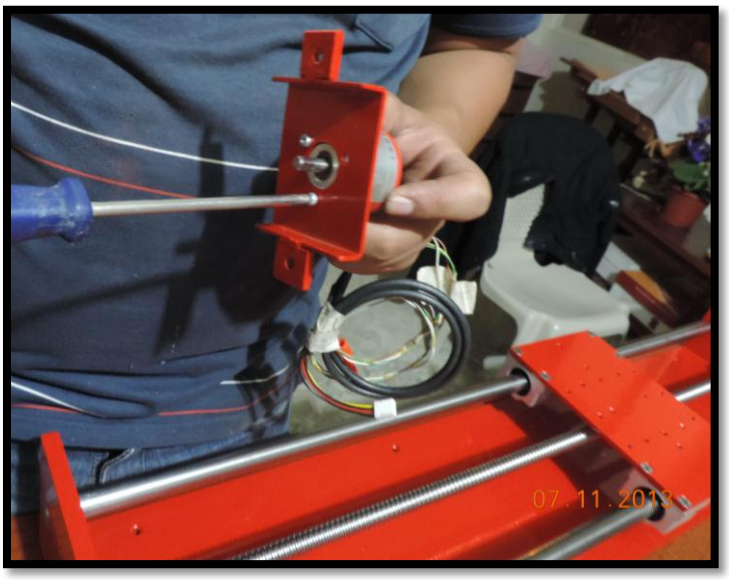
Fuente: Autor

En caso de ser necesario se puede cambiar el rodamiento manteniendo la misma carcasa.

2.4. Encoder

El módulo cuenta con un encoder de cuadratura Hohner, el cual es conecta al extremo del tornillo de bolas mediante su sujeción a una placa acopladora que le permite ser sujeta a la placa del deslizador.

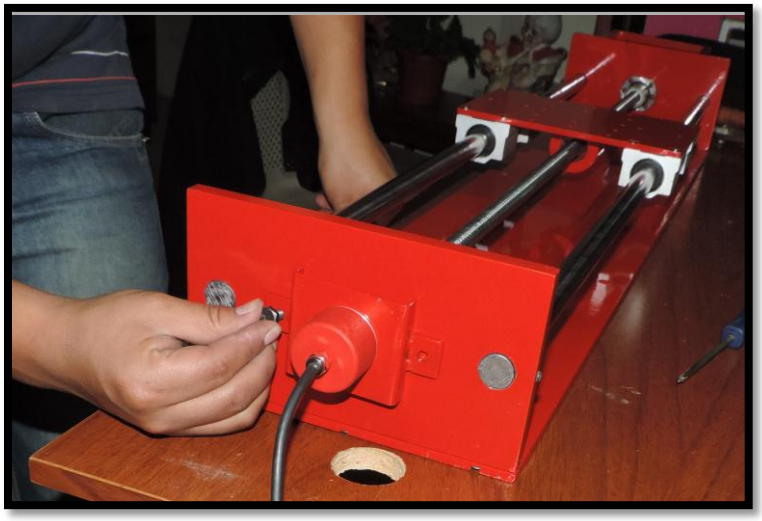
Ilustración 13. Sujeción del encoder.



Fuente: Autor

Los terminales del encoder se encuentran conectados a placa electrónica que cuenta con la leyenda “CONECTOR ENCODER”. Sus terminales de alimentación como sus salidas digitales se encuentran debidamente etiquetadas en la placa electrónica.

Ilustración 14. Acople del encoder al tornillo.

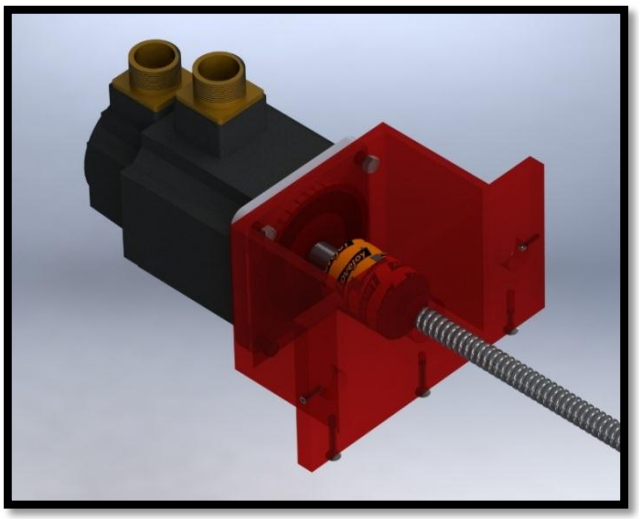


Fuente: Autor

2.5. Servomotor

El servomotor se sujeta al deslizador mediante cuatro tornillos de cabeza hexagonal de 5/16’’ de diámetro.

Ilustración 15. Sujeción del servomotor.



Fuente: Autor

El servomotor se une al tornillo de bolas mediante el matrimonio Lovejoy. Esta unión es de vital importancia para el servomotor, ya que de la misma depende la vibración que se produzca en el deslizador. El ajuste del matrimonio se realiza mediante prisioneros dentro del mismo acople.

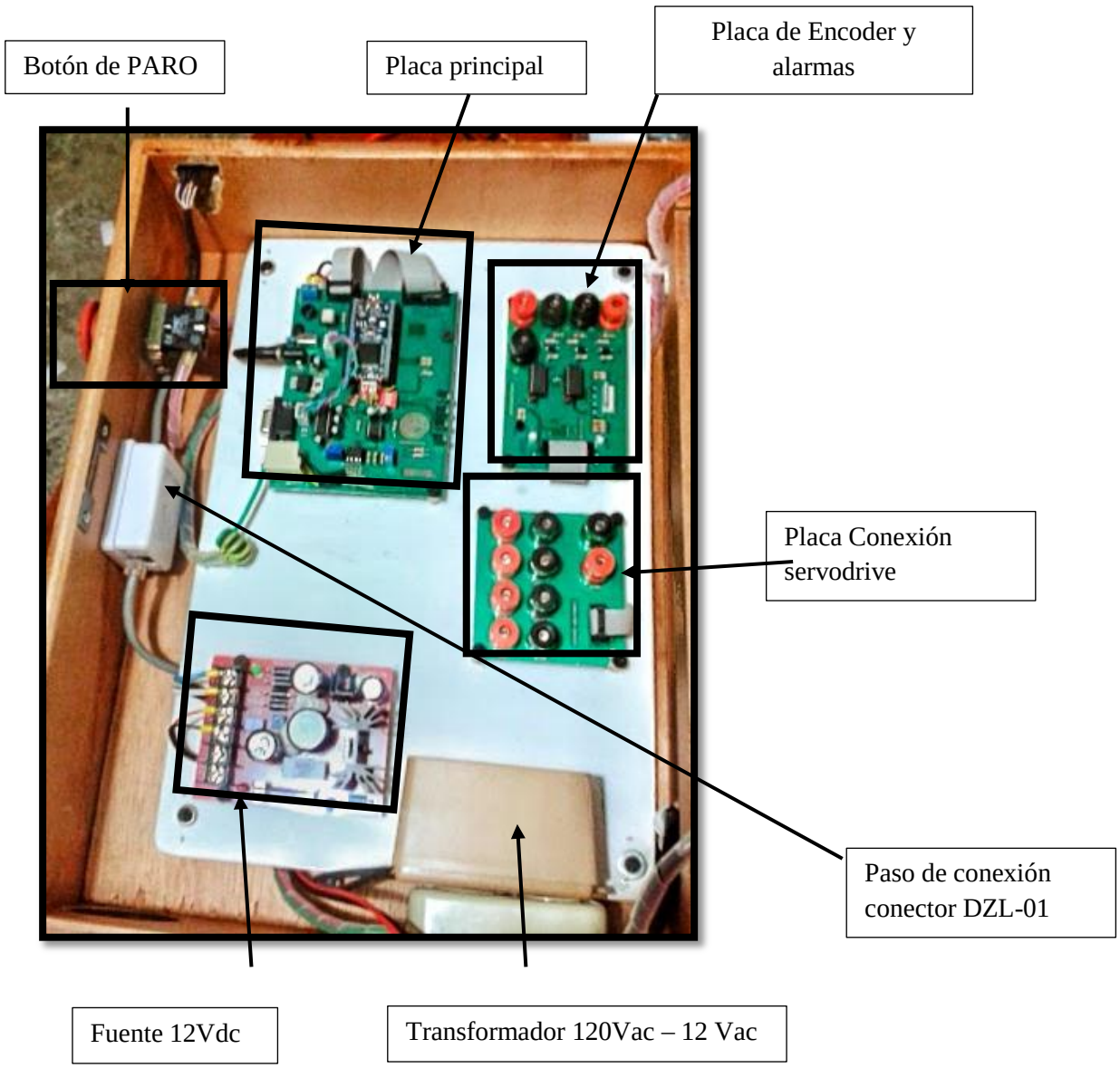
3. Electrónica

Para poder realizar el control del servomotor se han diseñado tres placas electrónicas las cuales se encuentran ubicadas dentro de una caja para mayor protección.

Dentro de esta caja se encuentran realizadas las conexiones necesarias para que el módulo funcione de manera adecuada. Los elementos con los que cuenta son:

- Fuente de voltaje de 12Vdc – 3A
- Transformador 120Vac-12Vac de 3A
- Botón de paro de emergencia
- Switch de encendido
- Luz piloto, indica el estado de la alimentación 110 Vac
- Conector para el servodrive.
- Cable de alimentación 24Vdc.
- Cable de comunicación serial RS-232.

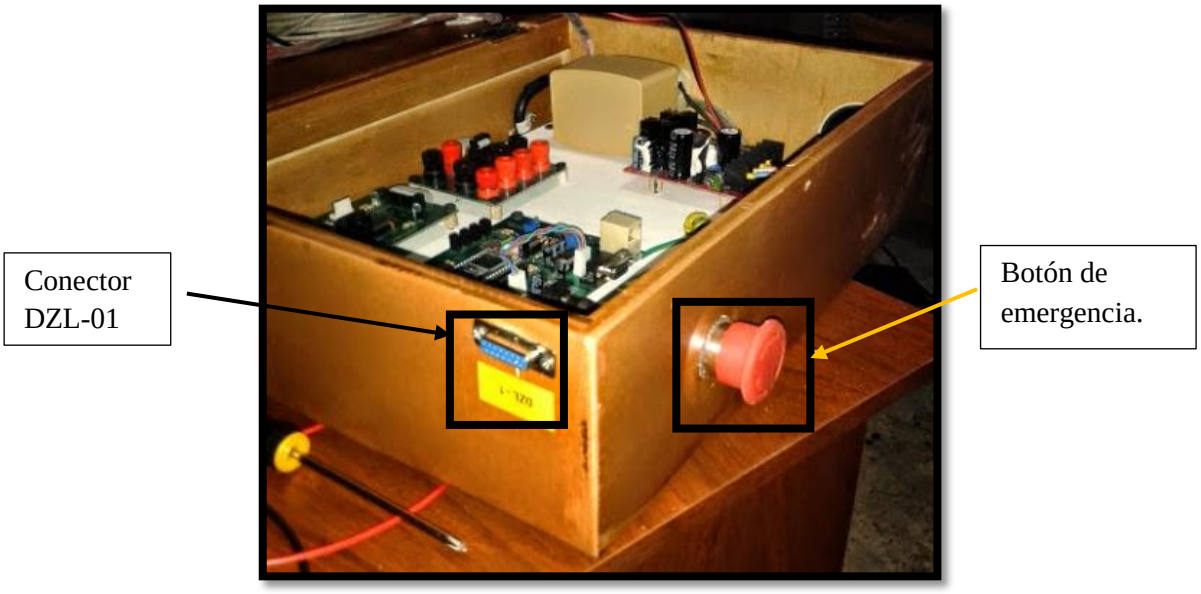
Ilustración 16. Caja de electrónica.



3.1. Placa de Conexión Servodrive

La placa de borneras tiene la finalidad de conectarse con el servodrive, esa diseñada para realizar esa conexión mediante conectores de banana, sin embargo, para mayor facilidad para el estudiante se ha incorporado un conector DB-15, el cual cuenta con un cable de conexión hacia el servodrive, para facilitar la instalación de la caja se ha instalado un paso de conexión al cual se conecta un cable de par trenzado UTP mediante un conector RJ-45, una examinación minuciosa del cableado facilitará conocer su funcionalidad.

Ilustración 17. Conector y botón de paro.



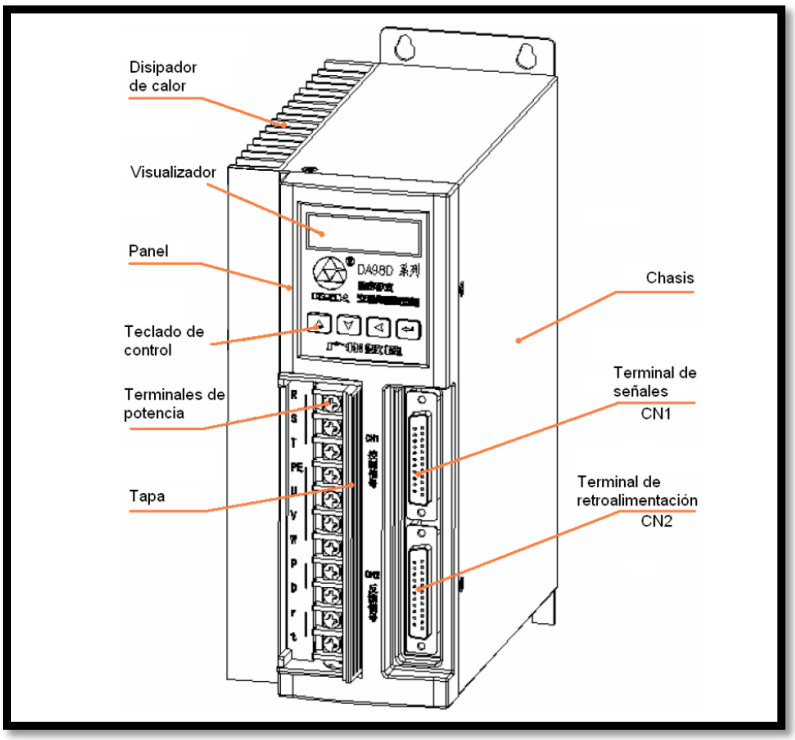
Fuente: Autor

El cable a utilizar para la conexión con el servodrive se encuentra debidamente etiquetado como DLZ-01.

Ilustración 18. Cable de comunicación DZL-01

Fuente: Autor

Ilustración 19. Servodrive GSK DA98D



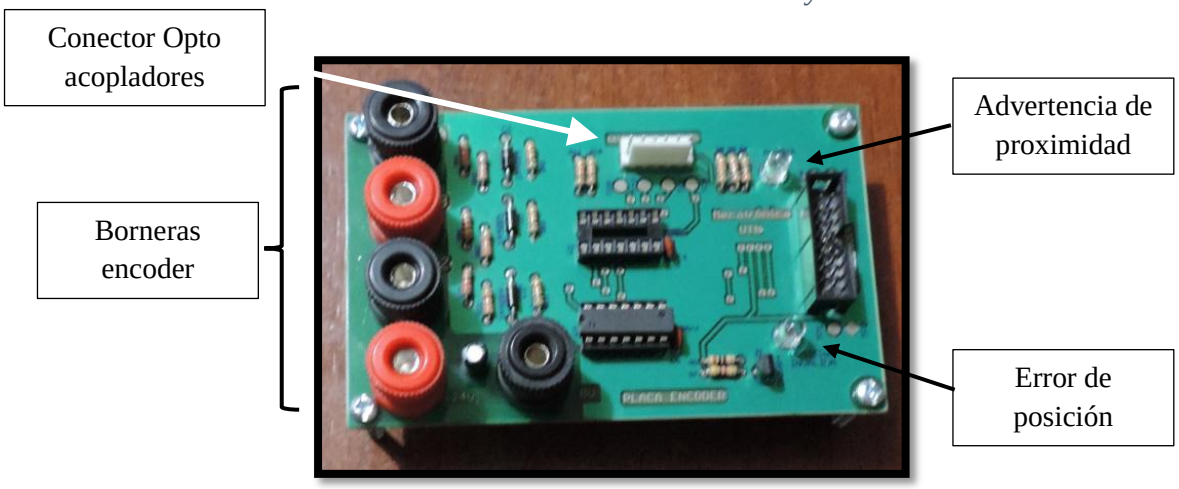
Fuente: (GSK CNC Equipment Co., 2007)

El cable de conectarse en el terminal CN1 del servodrive.

3.2. Placa de encoder y alarmas

La placa de encoder y alarmas está destinada a la adecuación de las señales provenientes tanto del encoder de cuadratura como de los opto-acopladores instalados para proveer señales de posición inválidas por parte del carro deslizador.

Ilustración 20. Placa de encoder y alarmas.



Fuente: Autor

El encoder cuenta con las siguientes conexiones.

Tabla 17. Conector de Encoder

Etiqueta	Función
A	Señal de cuadratura A
B	Señal de cuadratura B
Z	Señal de cruce por cero, un pulso por revolución
+	Alimentación positiva +24VDC
-	Alimentación negativa 0VDC

El conector para opto acopladores es un conector mólex de ocho pines, los opto acopladores ya se encuentran instalados en el deslizador.

3.3. Placa microcontrolador

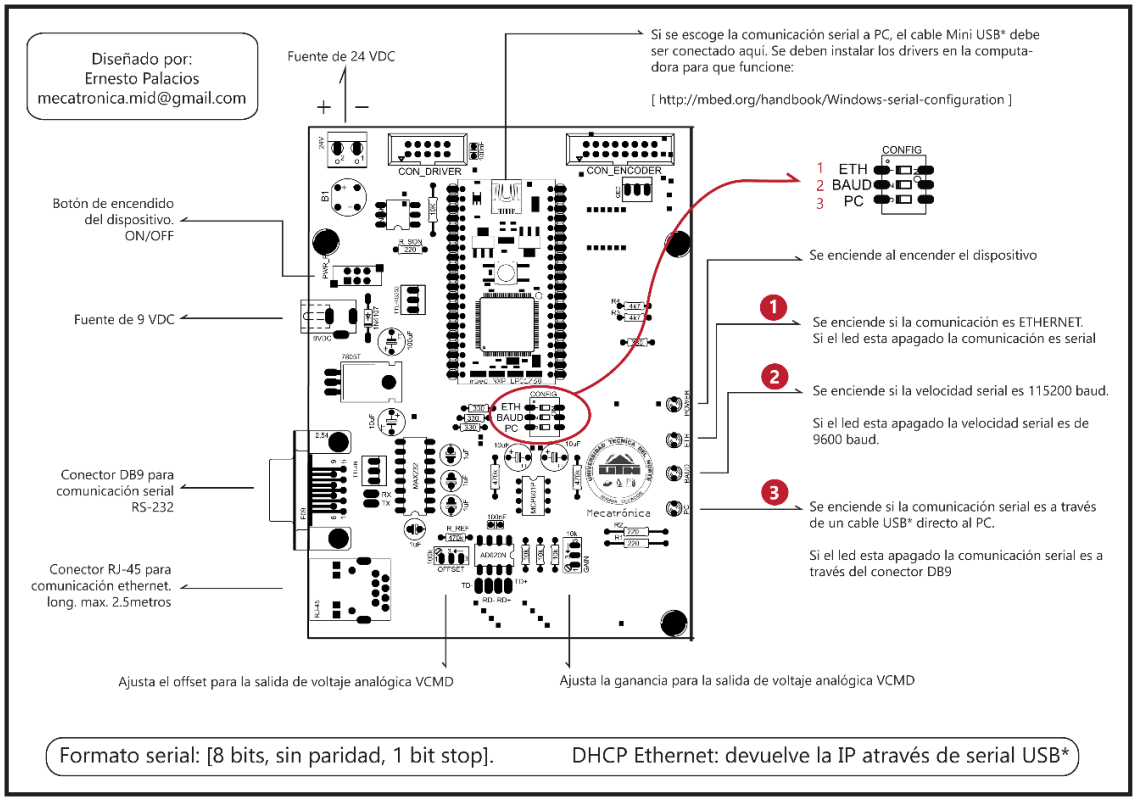
La placa principal cuenta con el microcontrolador para establecer comunicaciones con el computador, además en esta placa se encuentran las conexiones de voltaje que luego son repartidas a las demás placas.

La ilustración 21 se ha agregado a la caja del deslizador para una mejor y rápida visualización.

Se elaboró un cable RS-232 el cual permitirá conectase con el microcontrolador a través de su conector DB-9. En este se encuentran conectados los terminales 2,3 y 5, los cuales corresponden a RX, TX y Tierra para la comunicación serial.

En caso de ser necesario se puede construir un cable similar de mayor longitud, se debe tomar en cuenta la velocidad de comunicación serial, ya que es posible que una longitud elevada pueda deteriorar la transmisión de datos.

Ilustración 21. Conexiones de placa principal.



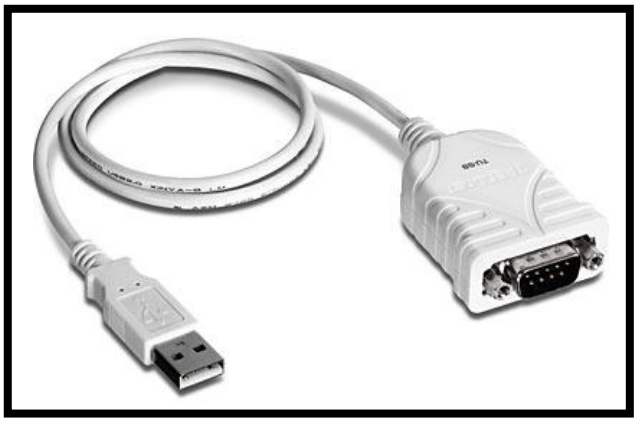
Fuente: Autor

Ilustración 22. Cable serial RS-232

Fuente: Autor

Para poder conectar a una computadora que no cuenta con un puerto serial nativo se debe utilizar

Ilustración 23. Cable USB - Serial. TRENDnet TU-S9



Fuente: Autor

Para la comunicación ethernet se puede utilizar un cable de red UTP Cat5e, sin embargo se debe tomar en cuenta que la longitud máxima no debe pasar los 2.5 metros de distancia.

4. Mantenimiento

El deslizador cuenta con partes móviles las cuales con el pasar del tiempo deberán ser reemplazadas, es difícil estimar su tiempo de vida útil ya que no se la frecuencia con que se utilizará el módulo, sin embargo, en su diseño se ha sobredimensionado sus elementos de tal manera que puedan durar un gran periodo de tiempo. El tiempo calculado de vida útil es de 176 (Km) de recorrido.

Los elementos que se deben verificar con frecuencia son los rodamientos, tanto los lineales que sostienen el deslizador, como los radiales que sostienen el tornillo. Estos elementos son los que tendrán un mayor desgaste y por lo tanto serán los primeros en ser reemplazados.

En el caso de ser necesario el reemplazo de algún elemento por favor referirse a la sección 3.1.3 DESLIZADOR. En esta se detallan la marca y modelo de los elementos del deslizador.

La velocidad máxima recomendada es de 1500 (RPM), el mantener una velocidad mayor puede causar daños a los rodamientos y el tornillo debido a la vibración.

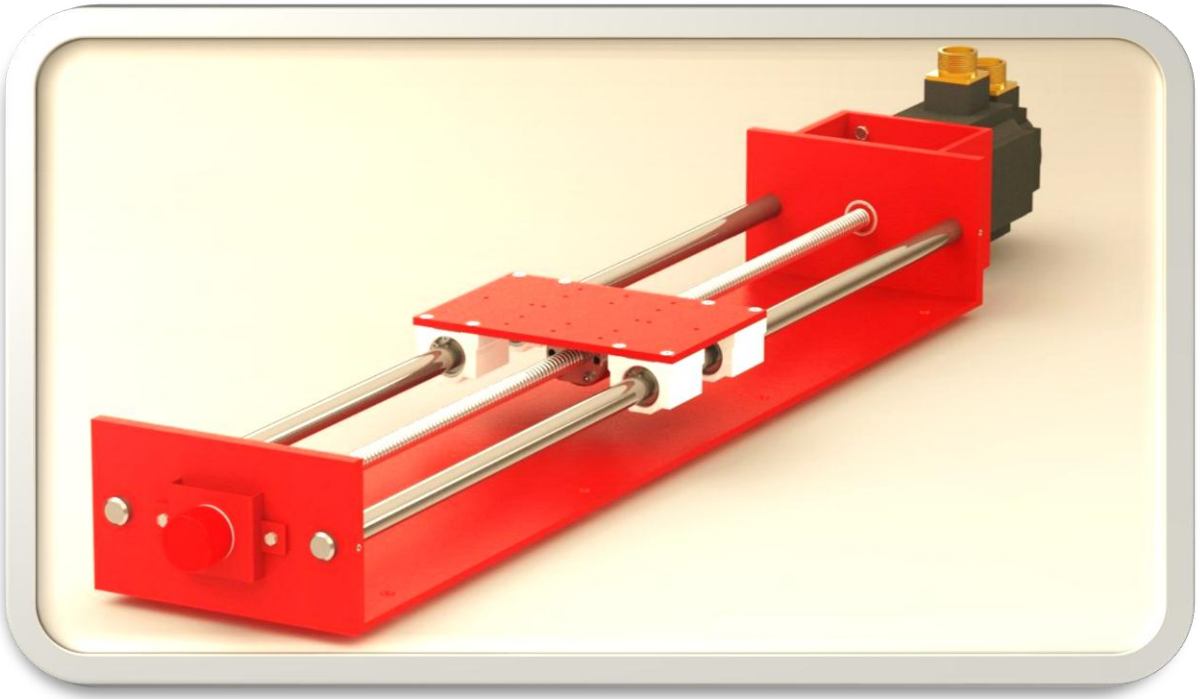
4.1. Aplicación de grasa

El mantenimiento del tornillo consiste en la aplicación periódica de grasa blanca 10-42 3M, este tipo de grasa ayudara a alargar el tiempo de vida del tornillo. Debido a que es difícil conseguir este tipo de grasa en el mercado local se sugiere el uso de “grasa azul” la cual es multipropósito y puede ser conseguida con facilidad.

Se debe aplicar la grasa antes de cualquier práctica que se realice con el deslizador, de esta manera se asegurará su correcto funcionamiento. Se debe evitar el contacto de las guías y el tornillo con las manos.

ANEXO 2. MANUAL DE PRÁCTICAS

DESLIZADOR LINEAL UTN



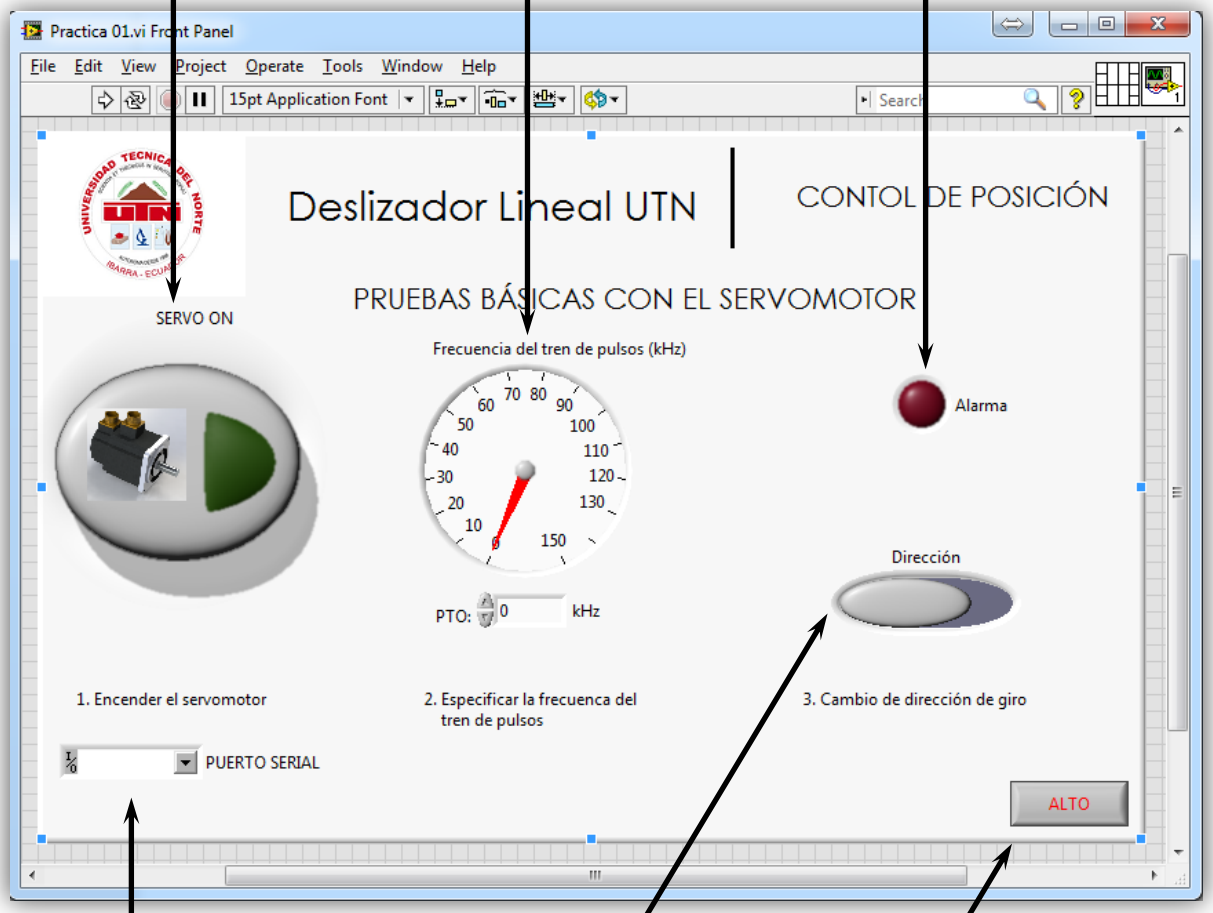
MANUAL DE PRÁCTICAS

PRACTICA UNO: Generación de frecuencia

CONTROL:
Encender Servo

CONTROL:
Frecuencia
generada

INDICADOR:
Alarmas



CONTROL:
Puerto serial
mbed

COTROL:
Dirección giro

CONTROL:
Detener
Programa

Objetivos.-

- Conocer las características, conexiones y capacidades básicas del sistema modular.
- Familiarizarse con los conceptos básicos, características y parámetros del servomotor en su modo de control de posición.

Introducción.-

Esta es una prueba básica que sirve para familiarizarse con el módulo y su funcionamiento. Permite el control del servomotor en su modo de posicionamiento a través de la generación de un tren de pulsos.

El control que se ejerce es de lazo abierto, es decir no existe retro-alimentación desde el deslizador al programa de LabVIEW, siendo el servodrive el elemento que se encarga de que los pulsos sean traducidos en movimiento rotacional del eje del servomotor.

La velocidad lineal del deslizador está determinada por la frecuencia que se genera desde el microcontrolador. La dirección de su giro está controlada por el estudiante desde el Panel Frontal de LabVIEW. **Es importante evitar que el carro del deslizador llegue hasta el final de su recorrido** ya que de darse esta condición el microcontrolador detendrá automáticamente al servomotor.

Materiales.-

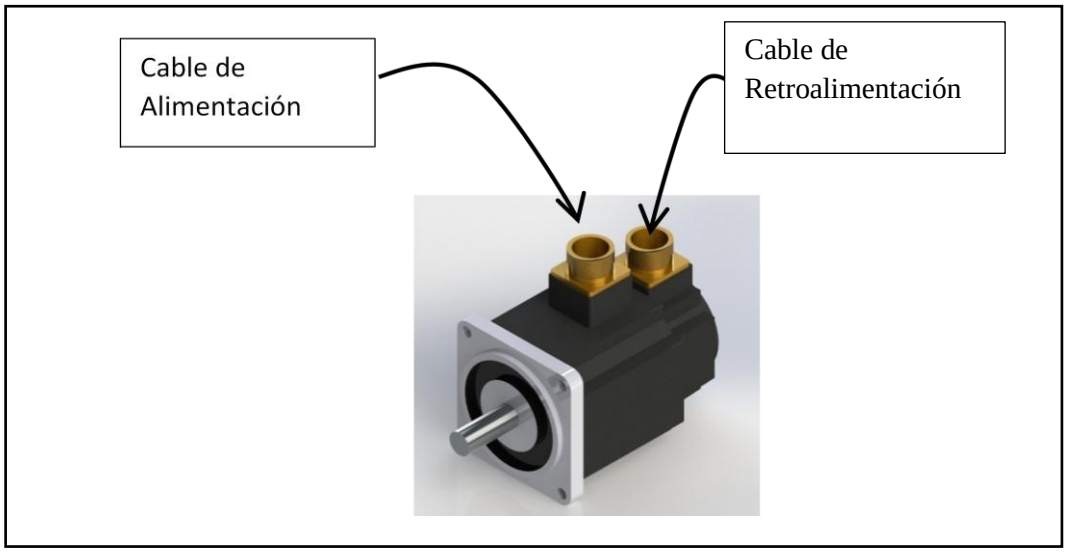
1	Servomotor GSK 110 SJT (Conectado al deslizador)
1	Servodrive DA98D
1	Deslizador Lineal
1	Computador instalado LabVIEW con las librerías del deslizador.

Procedimiento.-

CONEXIONES

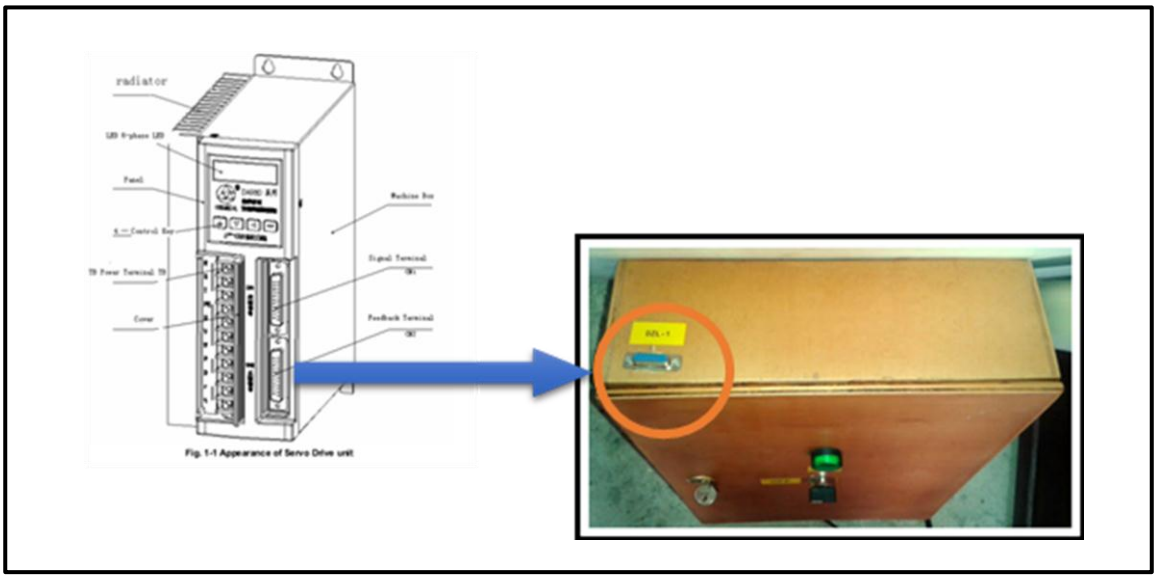
1. Se debe conectar los cables de alimentación y retroalimentación al servomotor.

Cuadro 1. Conexiones del servomotor

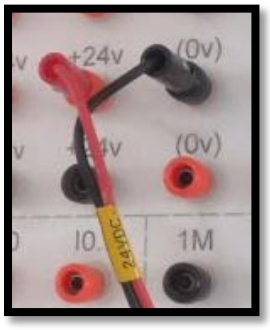


2. Se debe conectar el cableado de control del servodrive en el conector DZL-1 en la caja del deslizador. Tanto el cable de conexión como la tarjeta cuentan con etiquetas para su correcta conexión.

Cuadro 2. Conexión cable Servodrive - Deslizador



3. Se debe conectar el cable de alimentación de 24 VDC en el conector

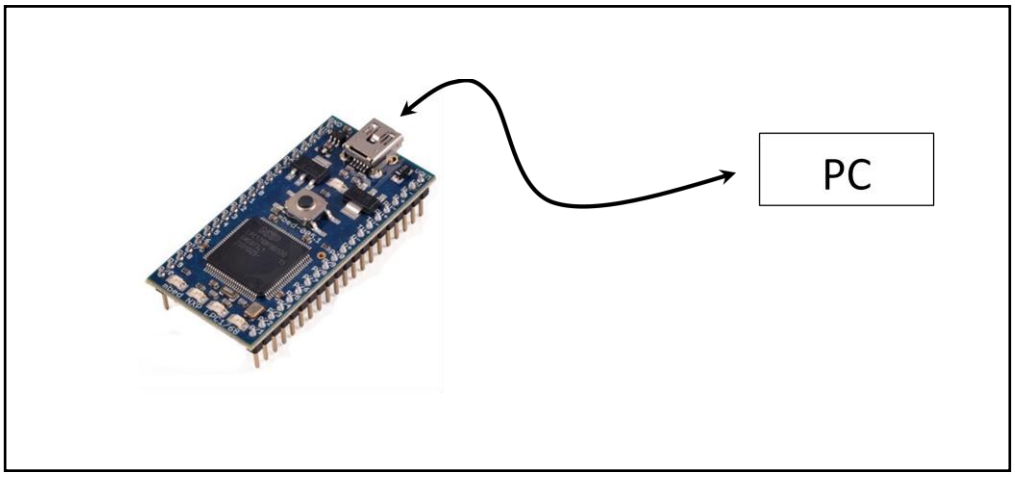


4. Se enciende la alimentación al servodrive, y se programan los siguientes parámetros.

PA	Valor
3	3
4	0
12	1
13	1
14	0
16	0
20	1

5. Se conecta la placa electrónica principal al computador, cuidando la configuración que se ha escogido. Se recomienda utilizar la comunicación Serial USB. Verificar el puerto de comunicación (COM) asignado al microcontrolador.

Cuadro 3. Conexión serial con el microcontrolador

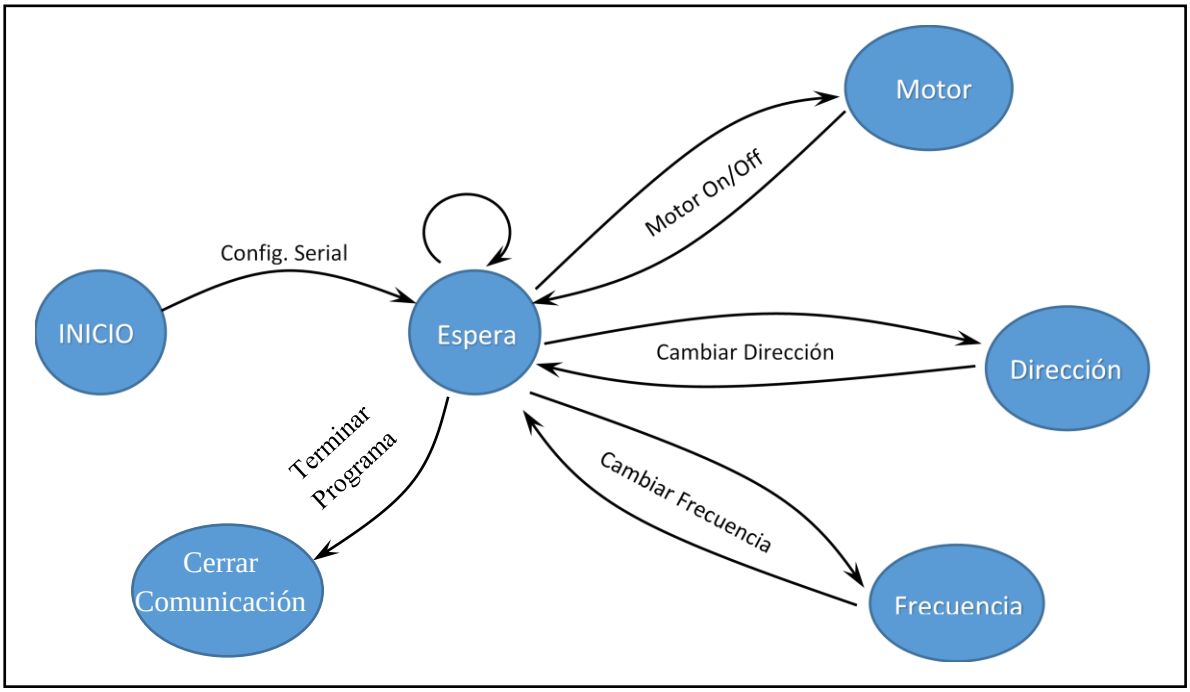


6. **IMPORTANTE:** Antes de ejecutar el programa en LabVIEW se debe escoger el puerto correcto en el panel frontal.

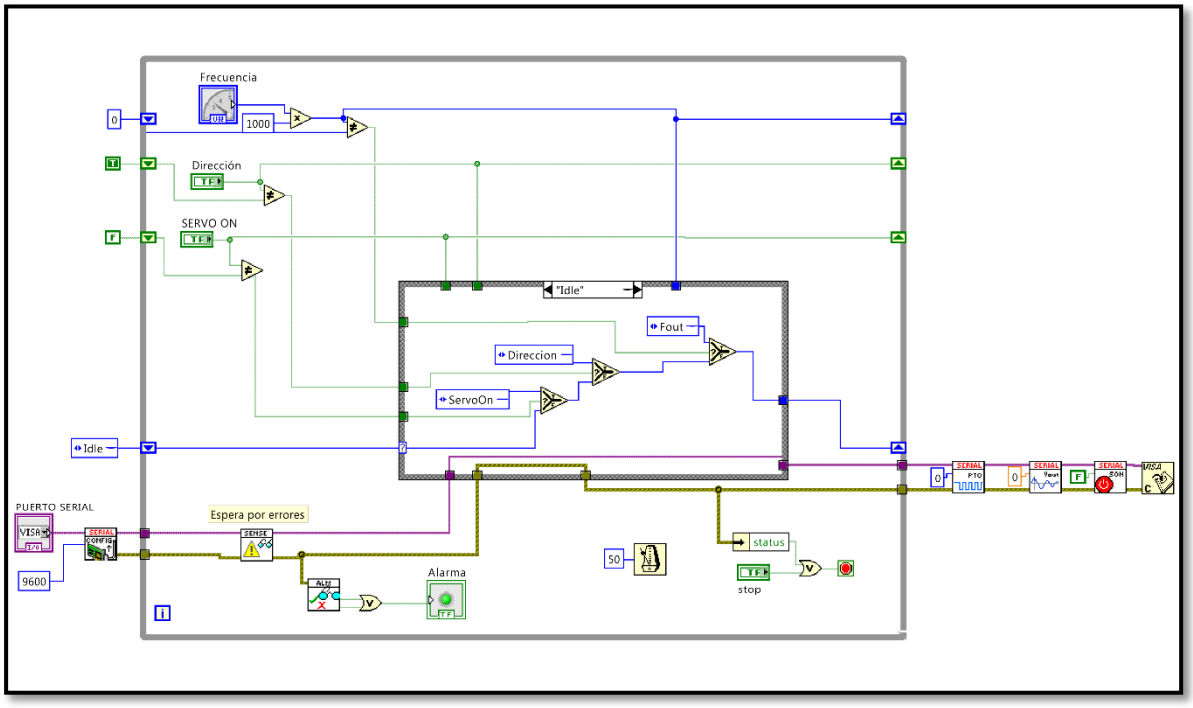
Programación.-

La programación se basa en una máquina de estados, la cual espera que el usuario realice una acción en su interfaz para actuar sobre el deslizador, una vez que se ha seleccionado una acción, encender el servo, cambio de la frecuencia, cambio en la salida de dirección. Se enviará el comando apropiado al microcontrilador.

Cuadro 4. Esquema de programación - Práctica 01



Cuadro 5. Diagrama de bloques - Practica 01



Resultados esperados.-

Una vez se ejecuta el programa el deslizador se desplazara a una velocidad constante determinada por la frecuencia del tren de pulsos y los parámetros del servodrive, esta velocidad se puede modificar cambiando su frecuencia. Si se cambia el estado de la variable “Dirección” se invertirá la dirección del movimiento.

Si el carro del servomotor se acerca demasiado al final de su recorrido el mismo se detendrá, y se mantendrá en este estado hasta que se lo aleje manualmente de esta posición.

Cuando se finalice el programa el servomotor detendrá su movimiento.

Ejercicios y recomendaciones.-

1. Se puede analizar el código fuente del programa de LabVIEW para entender como se ha programado la práctica.

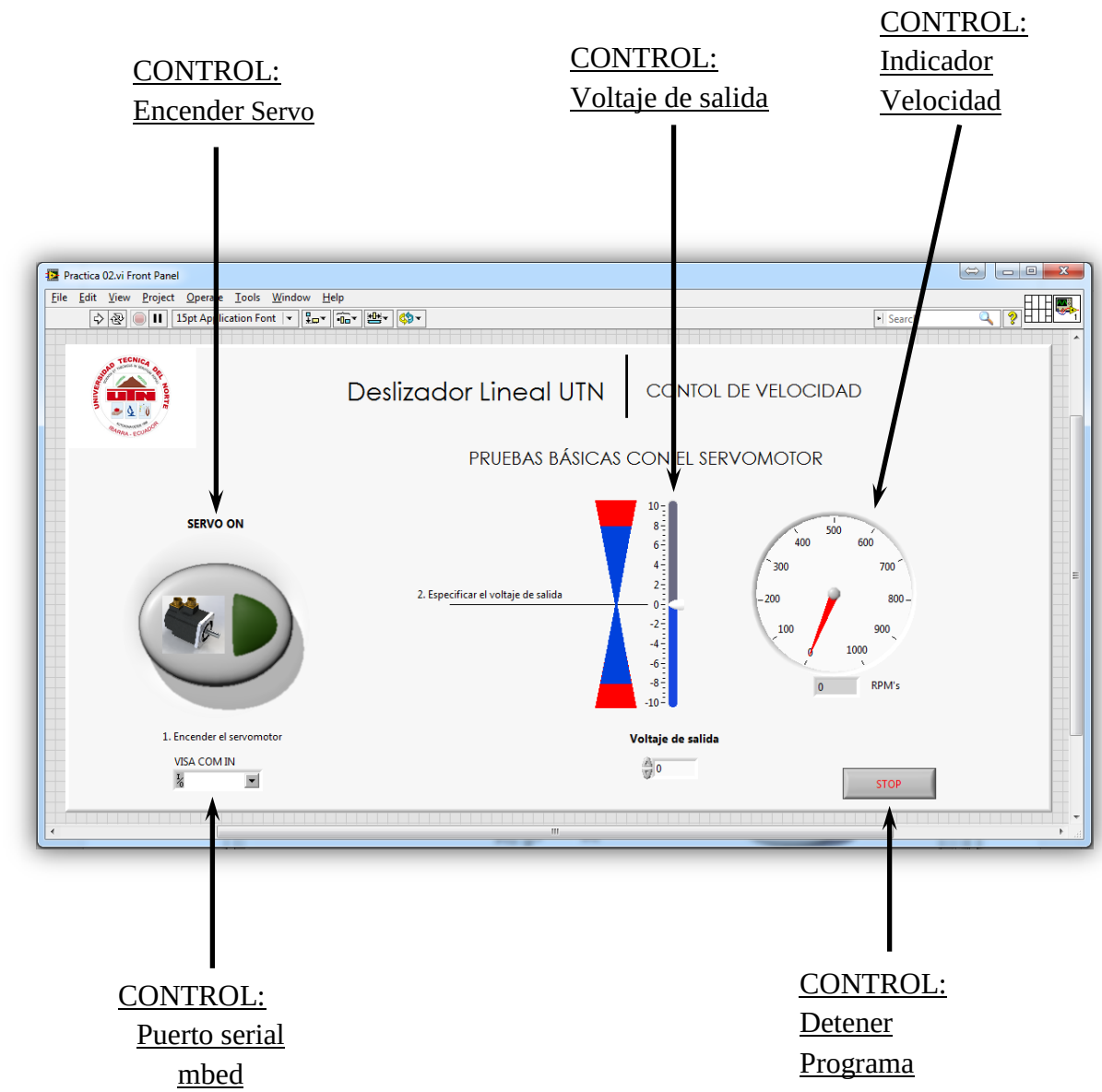
2. EJERCICIO: Se puede experimentar con los parámetros 12 y 13 del encoder y calcular cual es la velocidad en RPM’s obtenida al modificarlos.

3. EJERCICIO: Cambiar la programación en LabVIEW para que el módulo invierta su dirección automáticamente al llegar el final de su recorrido.

4. EJERCICIO: Cambiar la programación para ser ejecutada mediante comunicación ethernet del microcontrolador.

5. EJERCICIO: Cambiar la programación de la práctica para proveer al experimento de una rampa de aceleración y desaceleración

PRACTICA DOS: Generación de Voltaje.



IMPORTANTE:

Para realizar esta práctica se debe desconectar la mesa del deslizador de la tuerca del tornillo, de esta manera se puede rotar el servomotor sin que la base del deslizador se mueva

Objetivos.-

- Conocer las características, conexiones y capacidades básicas del sistema modular.
- Familiarizarse con los conceptos básicos, características y parámetros del servomotor en su modo de control de velocidad.

Introducción.-

Esta es una prueba básica que sirve para familiarizarse con el módulo y su funcionamiento. Permite el control del servomotor en su modo de control de velocidad a través de la generación de un voltaje variable.

El control que se ejerce es de lazo abierto, es decir no existe retro-alimentación desde el deslizador al programa de LabVIEW, siendo el servodrive el elemento que se encarga de que el voltaje sea traducido en movimiento rotacional del eje del servomotor.

La velocidad lineal del deslizador está determinada por la el voltaje que se genera desde el microcontrolador. La dirección de su giro está controlada por el estudiante desde el Panel Frontal de LabVIEW, si el voltaje es positivo se girará en un sentido, si el voltaje es negativo se invertirá su giro.

Materiales.-

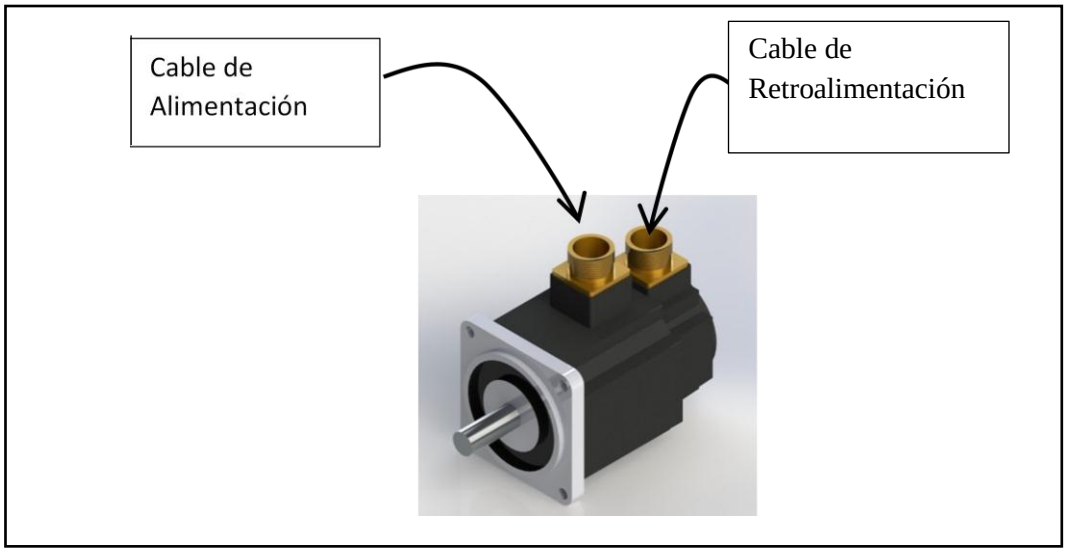
1	Servomotor GSK 110 SJT (Conectado al deslizador)
1	Servodrive DA98D
1	Deslizador Lineal
1	Computador instalado LabVIEW con las librerías del deslizador.
1	Destornillador de cabeza hexagonal 1/8’’

Procedimiento.-

CONEXIONES

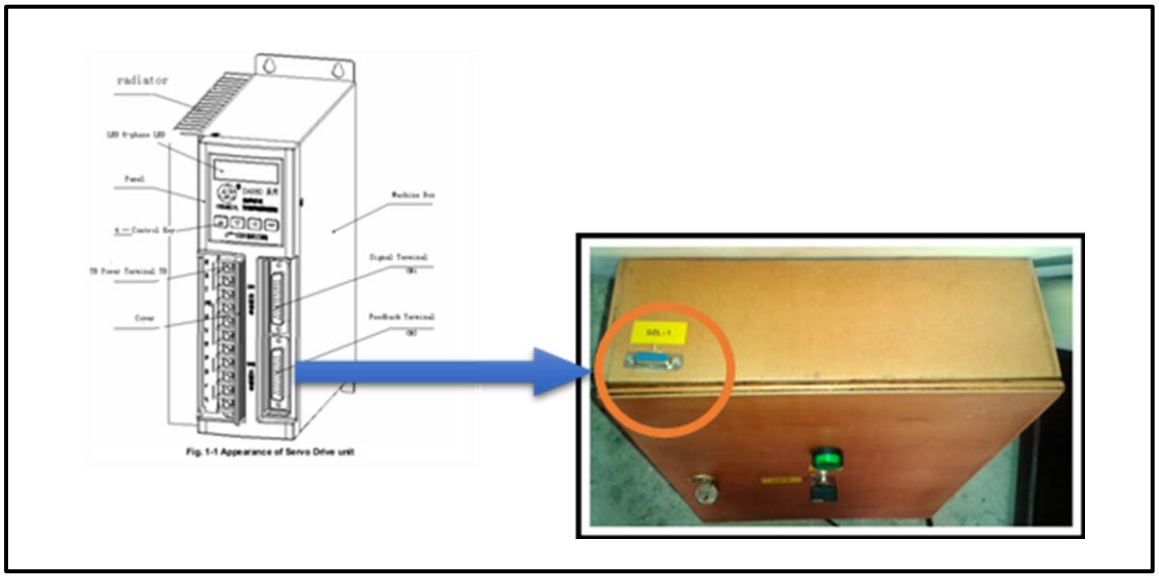
1. Se debe conectar los cables de alimentación y retroalimentación al servomotor.

Cuadro 6. Conexiones del servomotor



2. Se debe conectar el cableado de control del servodrive con la placa electrónica del deslizador destinada a este propósito. Tanto el cable de conexión como la tarjeta cuentan con etiquetas para su correcta conexión.

Cuadro 7. Conexión Servodrive - Deslizador



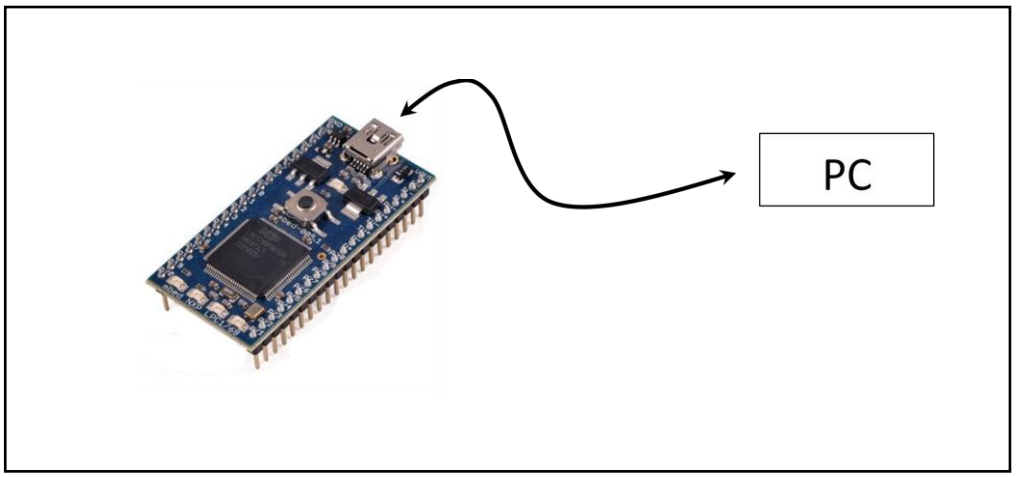
3. Se debe conectar el cable de alimentación de 24 VDC.

4. Se enciende la alimentación al servodrive, y se programan los siguientes parámetros.

PA	Valor
3	0
4	1
20	1
23	1000
33	250
39	0
40	0
43	1
46	0

5. Se conecta la placa electrónica principal al computador, cuidando la configuración que se ha escogido. Se recomienda utilizar la comunicación Serial USB.

Cuadro 8. Conexión serial con el microcontrolador

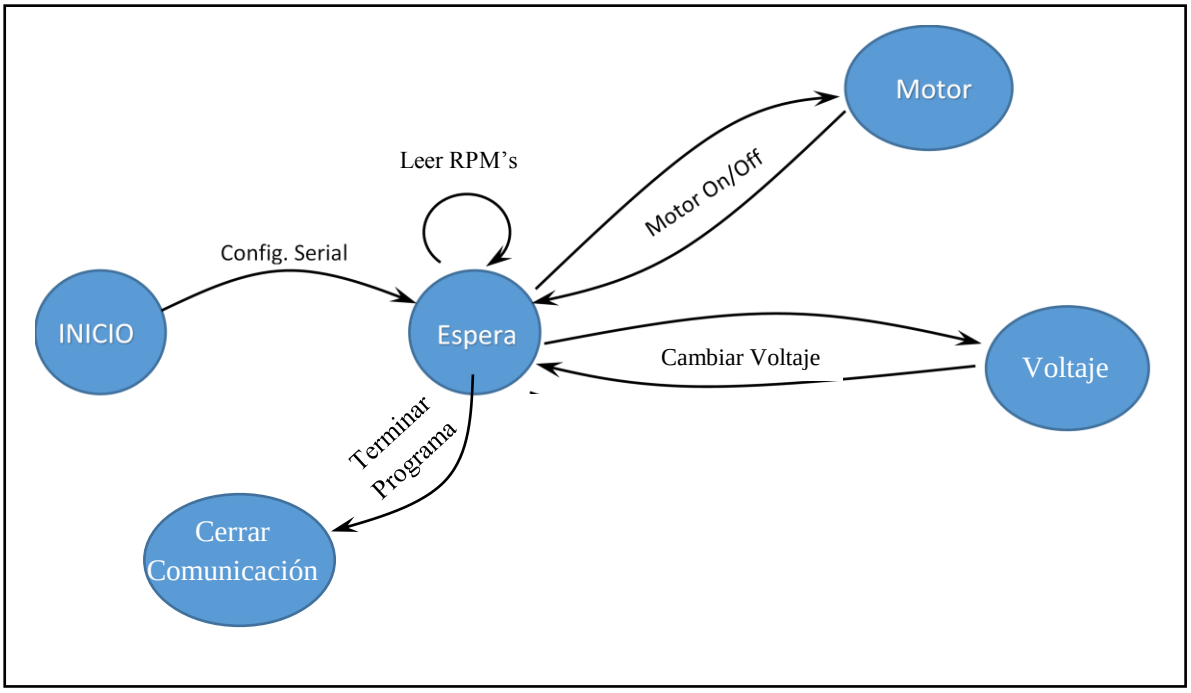


6. **IMPORTANTE:** Antes de ejecutar el programa en LabVIEW se debe escoger el puerto correcto en el panel frontal.

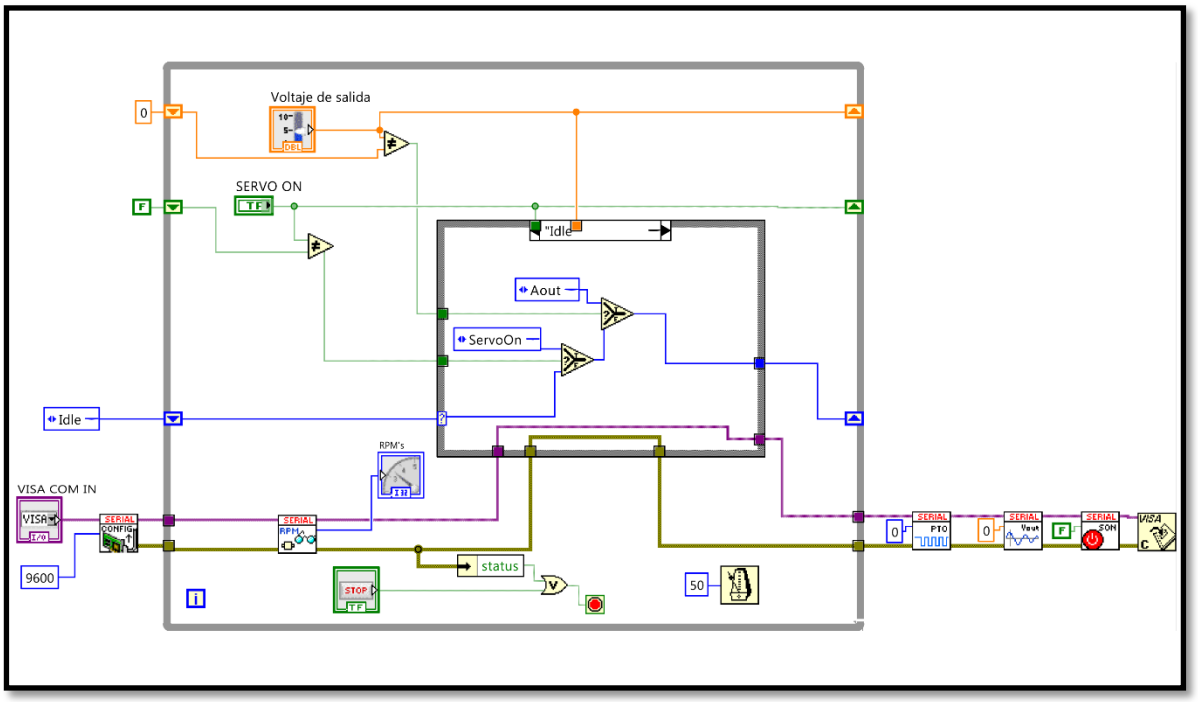
Programación.-

La programación se basa en una máquina de estados, la cual espera que el usuario realice una acción en su interfaz para actuar sobre el deslizador.

Cuadro 9. Diagrama de Máquina de estados



Cuadro 10. Diagrama de bloques - Practica 02



Resultados esperados.-

Una vez se ejecuta el programa el servomotor girará a una velocidad proporcional al voltaje aplicado, el sentido del giro se determina por la polaridad del voltaje. La velocidad máxima alcanzada será aquella configurada en el parámetro 23 del servomotor.

Si existe un error entre la velocidad medida por el encoder y la medida por el servodrive dependerá de la calibración del amplificador instrumental en la placa principal del microcontrolador. Para la calibración del mismo se necesita un multímetro y la medición de los terminales VCMD y AGND en la placa de conectores dentro del tablero de control del módulo.

Ejercicios y recomendaciones

1. EJERCICIO: Se puede medir el error entre el voltaje de configurado y el que realmente alimenta al servomotor y determinar el error en su velocidad.

2. EJERCICIO: Se puede analizar el código fuente del programa de LabVIEW para entender como se ha programado la práctica.

3. EJERCICIO: Cambiar la programación para ser ejecutada mediante la comunicación ethernet del microcontrolador.

4. EJERCICIO: Cambiar la programación de la práctica para proveer al experimento de una rampa de aceleración y desaceleración.

5. EJERCICIO: Cambiar la programación para que permita la calibración del voltaje desde el programa de LabVIEW.

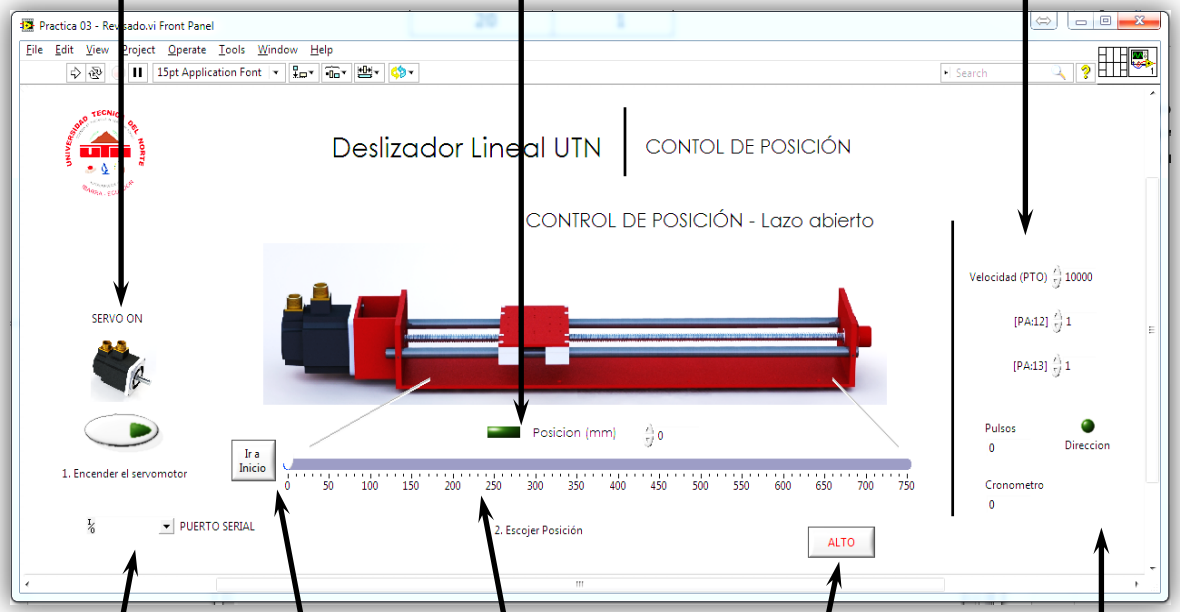
6. EJERCICIO: Determinar el voltaje mínimo al cual el servomotor detiene su marcha, modificar el parámetro 33 del servodrive y observar su efecto.

PRACTICA TRES: Control del posición del deslizador (Lazo Abierto)

CONTROL:
Encender Servo

INDICADOR:
Posición alcanzada

CONTROL:
Velocidad Pulsos
Parámetro 12 Sevodrive
Parámetro 13 Sevodrive



CONTROL:
Puerto serial
mbed

CONTROL:
Ir al inicio
del recorrido

CONTROL:
Seleccionar
Posición

CONTROL:
Detener
Programa

INDICADOR:
Pulsos a generar
Dirección
Tiempo Cronómetro

Objetivos.-

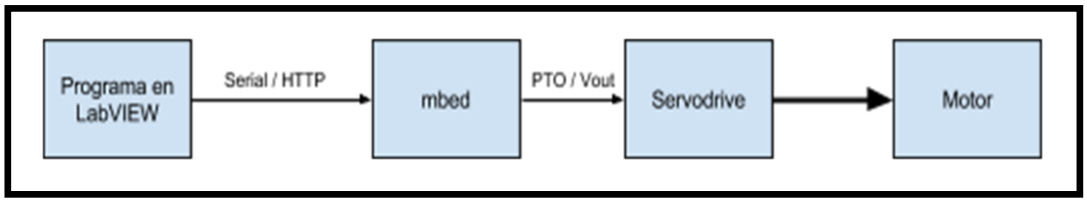
- Conocer las capacidades de programación del módulo.
- Experimentar con las capacidades de posicionamiento del módulo.

Introducción.-

Esta es una aplicación básica que sirve de punto de partida para las aplicaciones prácticas del módulo. Permite el control de la posición del carro del deslizador mediante el control de su velocidad lineal, su dirección y el tiempo que lleva desplazándose.

El control que se ejerce es de lazo abierto, es decir no existe retro-alimentación desde el deslizador al programa de LabVIEW, siendo el servodrive el elemento que se encarga de que los pulsos sean traducidos en movimiento rotacional del eje del servomotor.

Ilustración 24. Sistema en lazo abierto



Fuente: Autor

La velocidad lineal del deslizador está determinada por la frecuencia que se genera desde el microcontrolador y los parámetros 12 y 13.

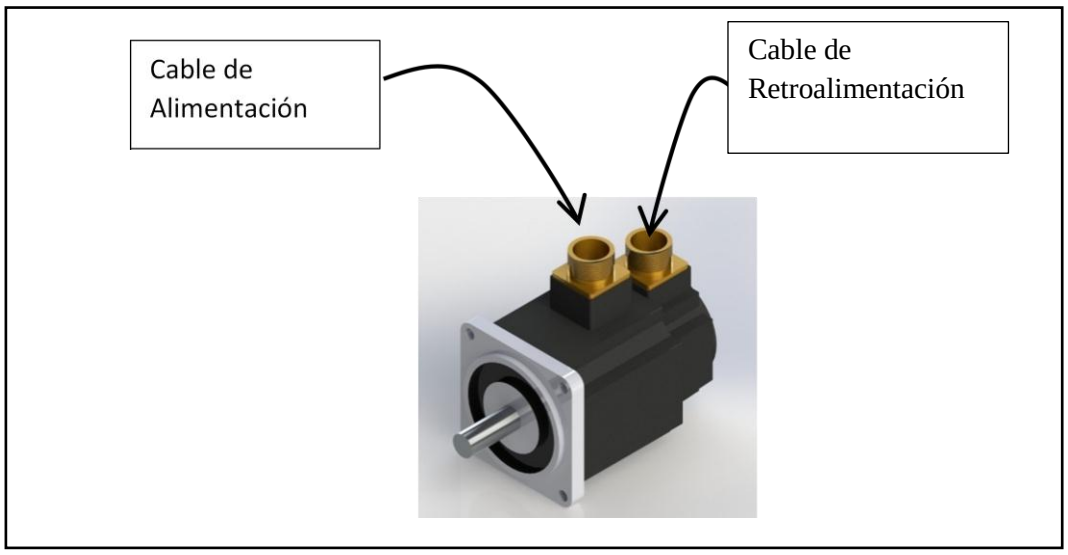
Materiales.-

1	Servomotor GSK 110 SJT (Conectado al deslizador)
1	Servodrive DA98D
1	Deslizador Lineal
1	Computador instalado LabVIEW con las librerías del deslizador.

Procedimiento.-

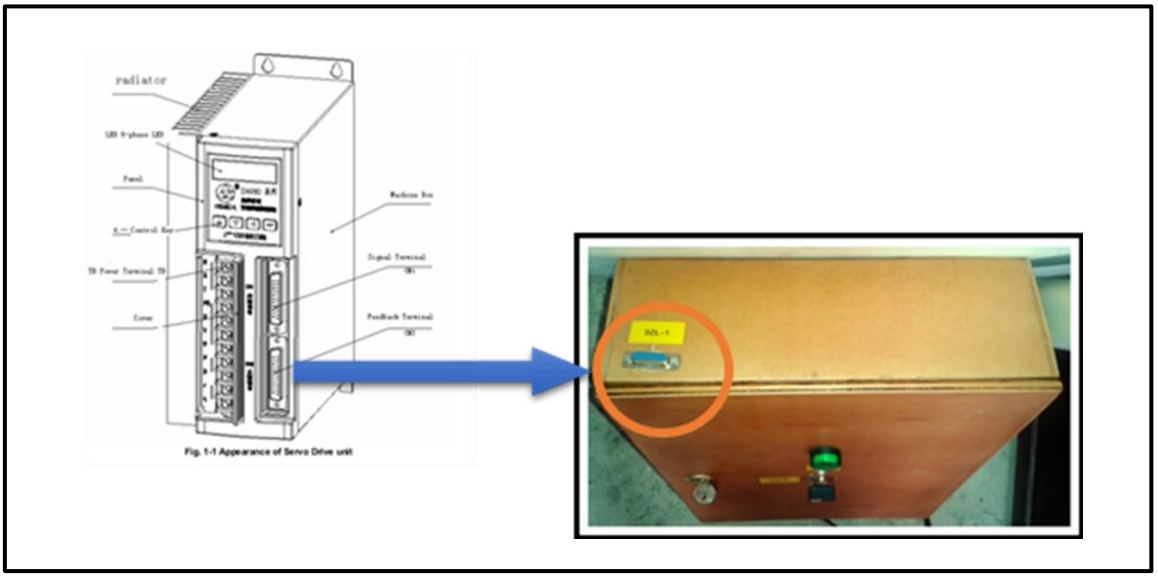
7. Se debe conectar los cables de alimentación y retroalimentación al servomotor.

Cuadro 11. Conexiones del servomotor

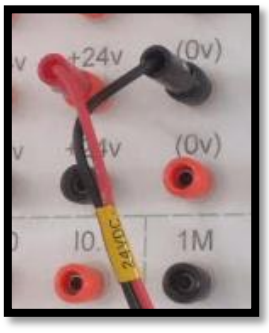


8. Se debe conectar el cableado de control del servodrive con la placa electrónica del deslizador destinada a este propósito. Tanto el cable de conexión como la tarjeta cuentan con etiquetas para su correcta conexión.

Cuadro 12. Conexión cable Servodrive - Deslizador



9. Conectar el cable de alimentación de 24VDC.

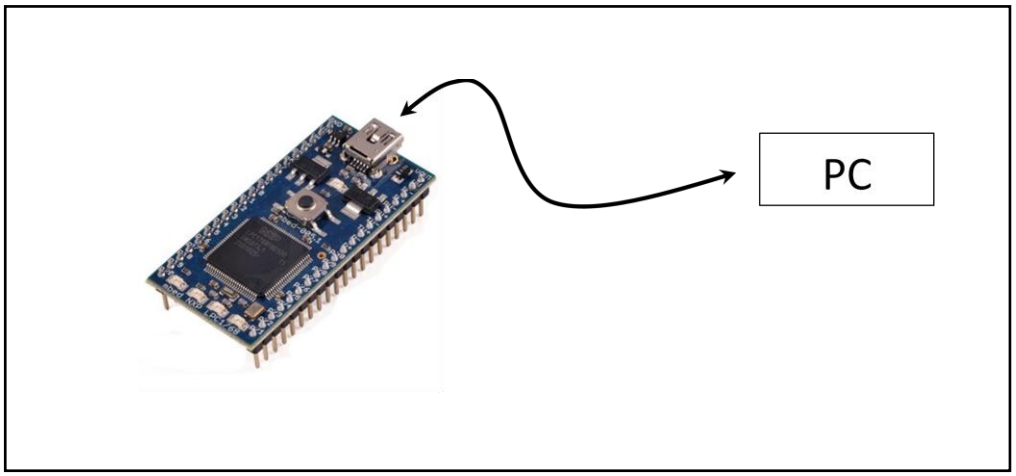


10. Se enciende la alimentación al servodrive, y se programan los siguientes parámetros.

PA	Valor
3	3
4	0
12	1
13	1
14	0
16	0
20	1

11. Se conecta la placa electrónica principal al computador, cuidando la configuración que se ha escogido. Se recomienda utilizar la comunicación Serial USB.

Cuadro 13. Conexión serial con el microcontrolador



12. **IMPORTANTE:** Antes de ejecutar el programa en LabVIEW se debe escoger el puerto correcto en el panel frontal.

Resultados esperados.-

Una vez que se ejecuta el programa automáticamente se llevara el carro del deslizador hacia su posición inicial de cero, si este proceso demora más allá de 35 segundos el programa desplegará un mensaje de advertencia.

Una vez que se ubica el deslizador ene esta posición se puede escoger una posición cualquiera en su recorrido, si la posición es cambiada antes de llegar a su destino el programa volverá a calcular la distancia que debe recorrer para llegar a su posición final, una vez que se llegue a esta posición se encenderá en el panel frontal el indicador de posición alcanzada.

PROGRAMACIÓN

La programación se basa en dos máquinas de estados ejecutándose en paralelo, la primera de ellas espera que el usuario realice una acción en su interfaz, especialmente cambios en la posición. La segunda máquina de estado lleva un control del tiempo transcurrido desde el último cambio de posición, calcula la posición actual del deslizador en base al tiempo transcurrido, y finalmente desplaza el deslizador hasta su posición final.

Debido a que se desea controlar la posición, la primera acción del programa es llevar el deslizador hasta la posición de inicio, durante este tiempo no se podrá realizar acciones sobre el deslizador.

Una vez en la posición cero se podrán escoger nuevas posiciones para el posicionamiento del deslizador.

El programa utiliza un temporizador para para llevar la cuenta de la distancia recorrida, si se escoge una nueva distancia antes que se llegue a completar la anterior, el programa recalcula el tiempo necesario para que llegue a su nueva posición.

Fig. 81. Diagrama de flujo - Práctica 03.

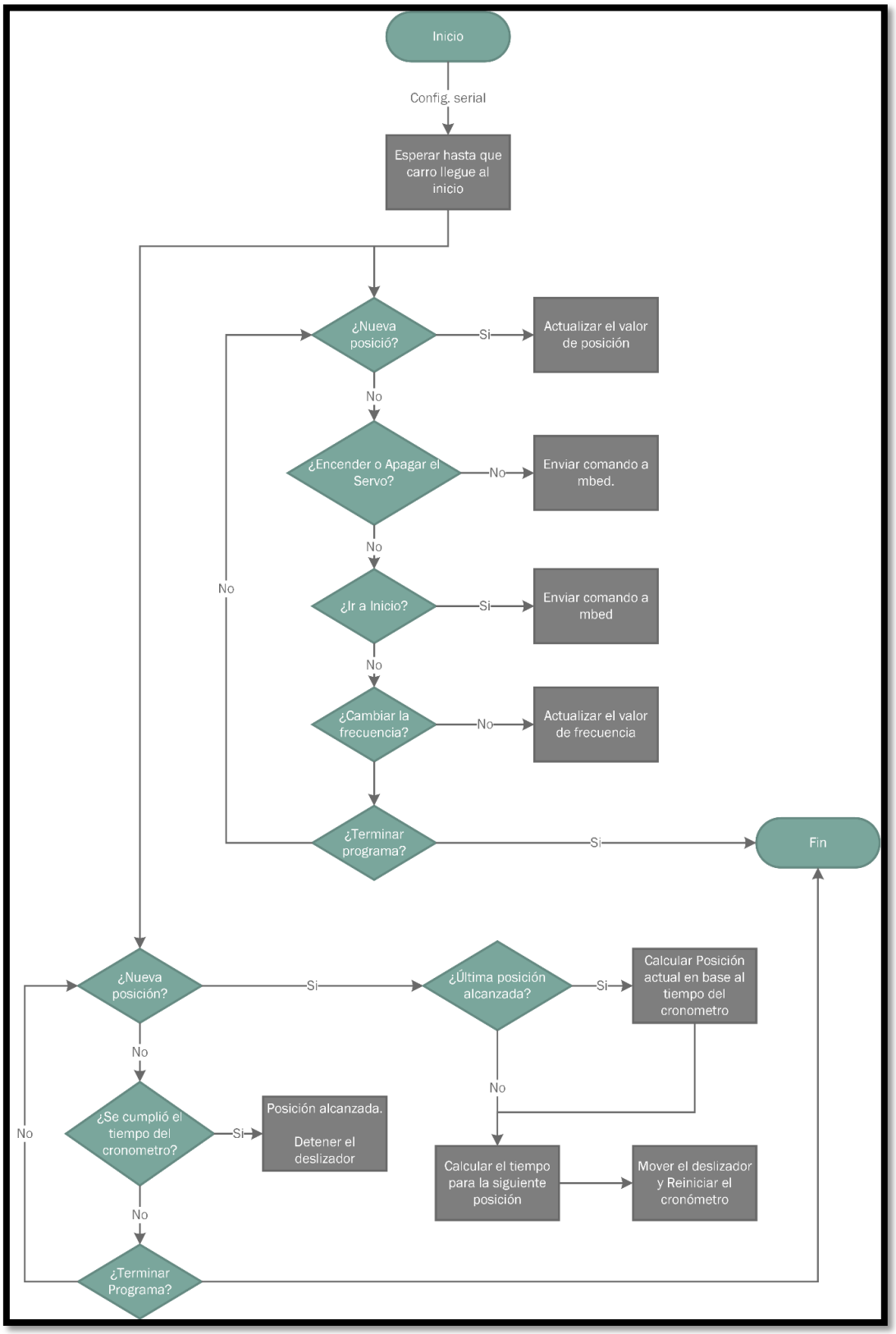


Fig. 82. Configuración e Ir a Inicio

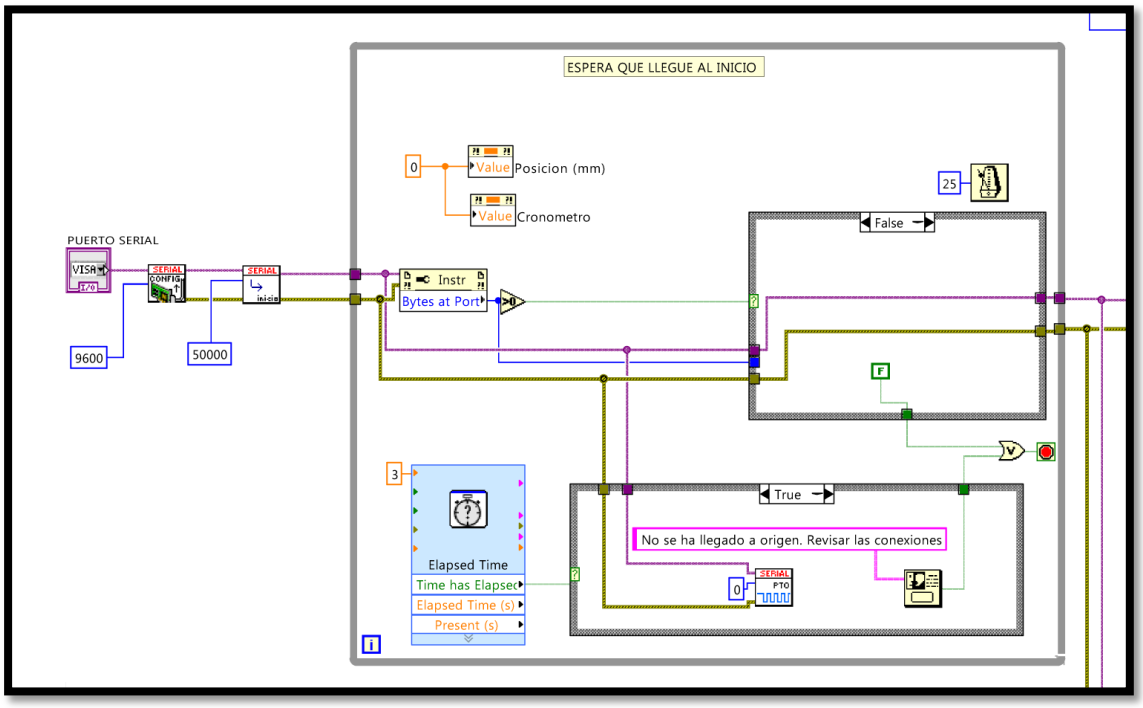


Fig. 83. Lazo de espera de cambios en la interfaz

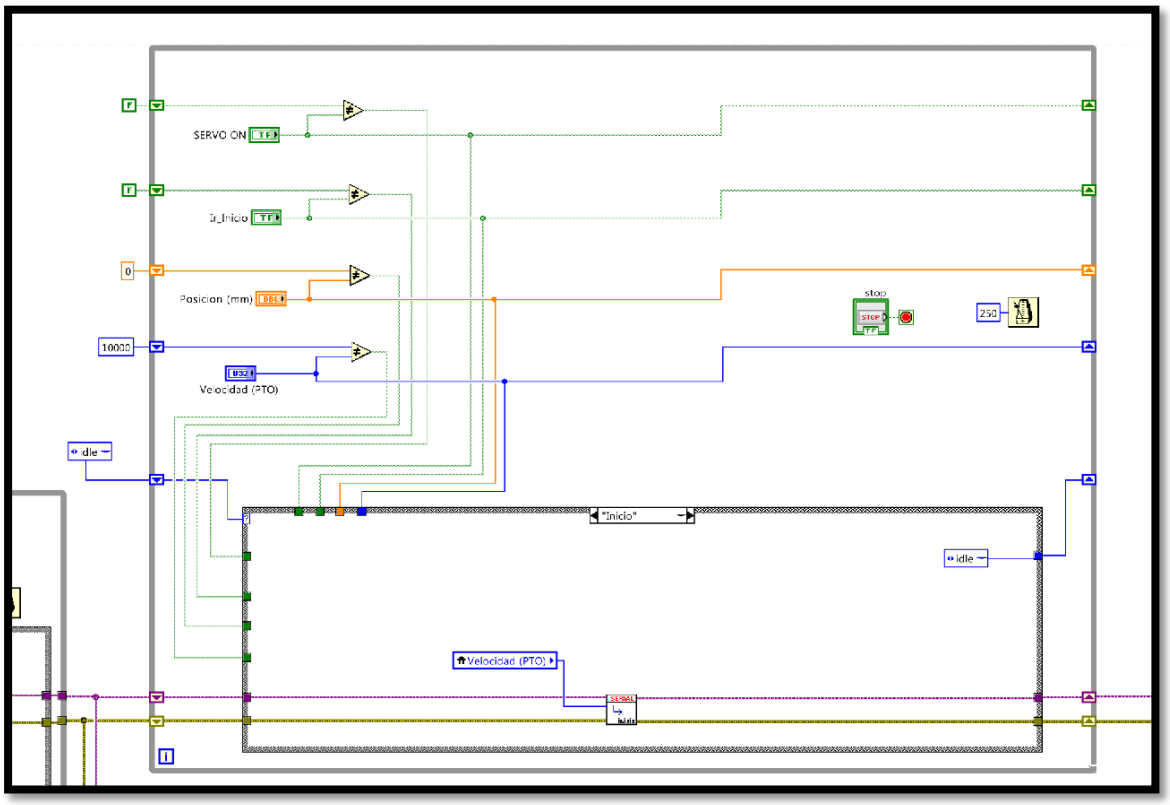
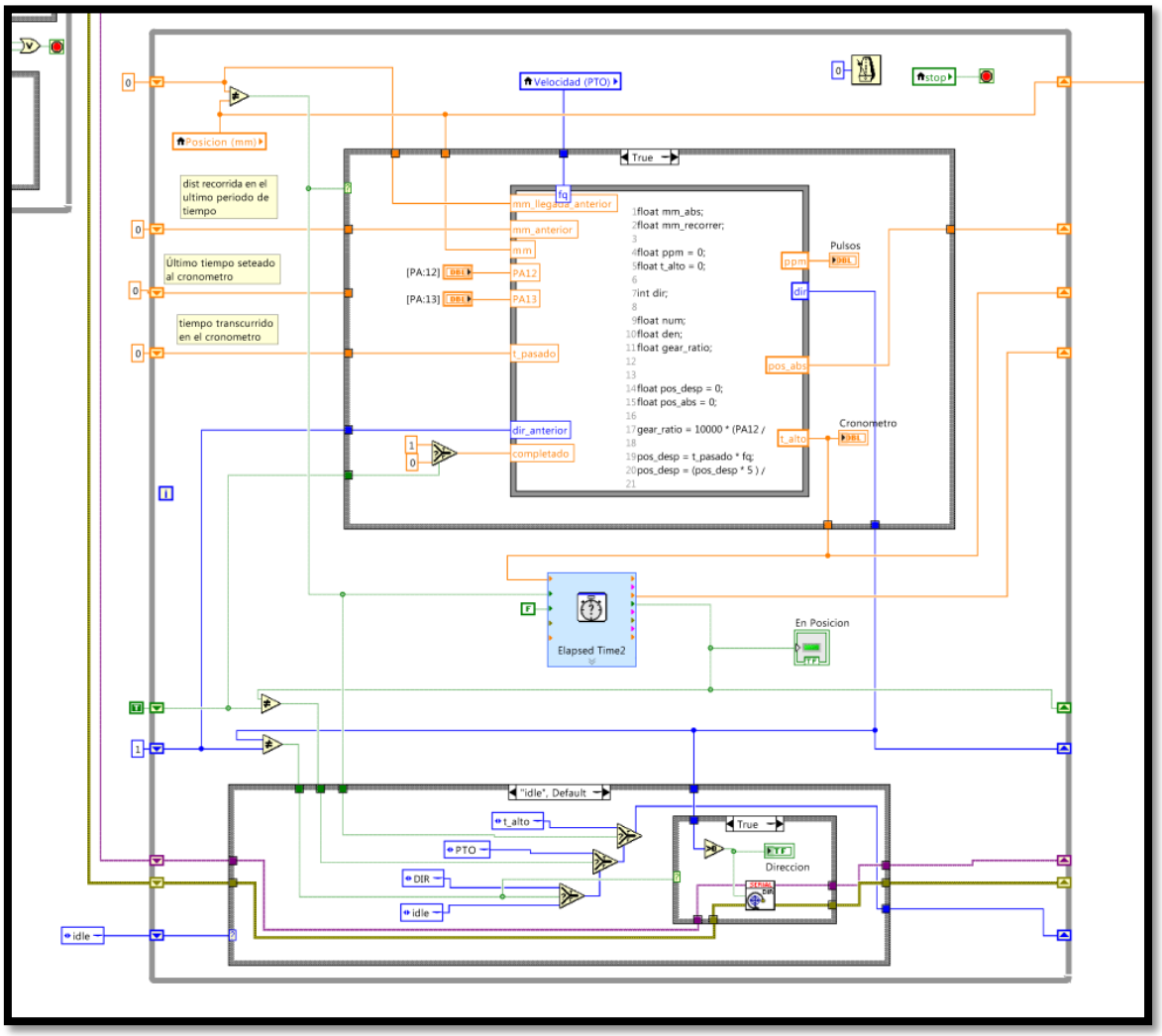


Fig. 84. Diagrama del control del tiempo



EJERCICIOS Y RECOMENDACIONES

1. EJERCICIO: Re-escribir el programa actual utilizando como medida de posición el contador en el registro del encoder.

2. EJERCICIO: Determinar cuál de los dos métodos de control es más precisión, control del tiempo (práctica actual), o control del registro del encoder (EJERCICIO 1).

3. EJERCICIO: Cambiar los parámetros del servomotor y observar si tienen algún efecto en la precisión del programa.

4. EJERCICIO: Cambiar la programación y proveer al sistema de Visión Artificial, que permita posicionar el módulo de manera dinámica.

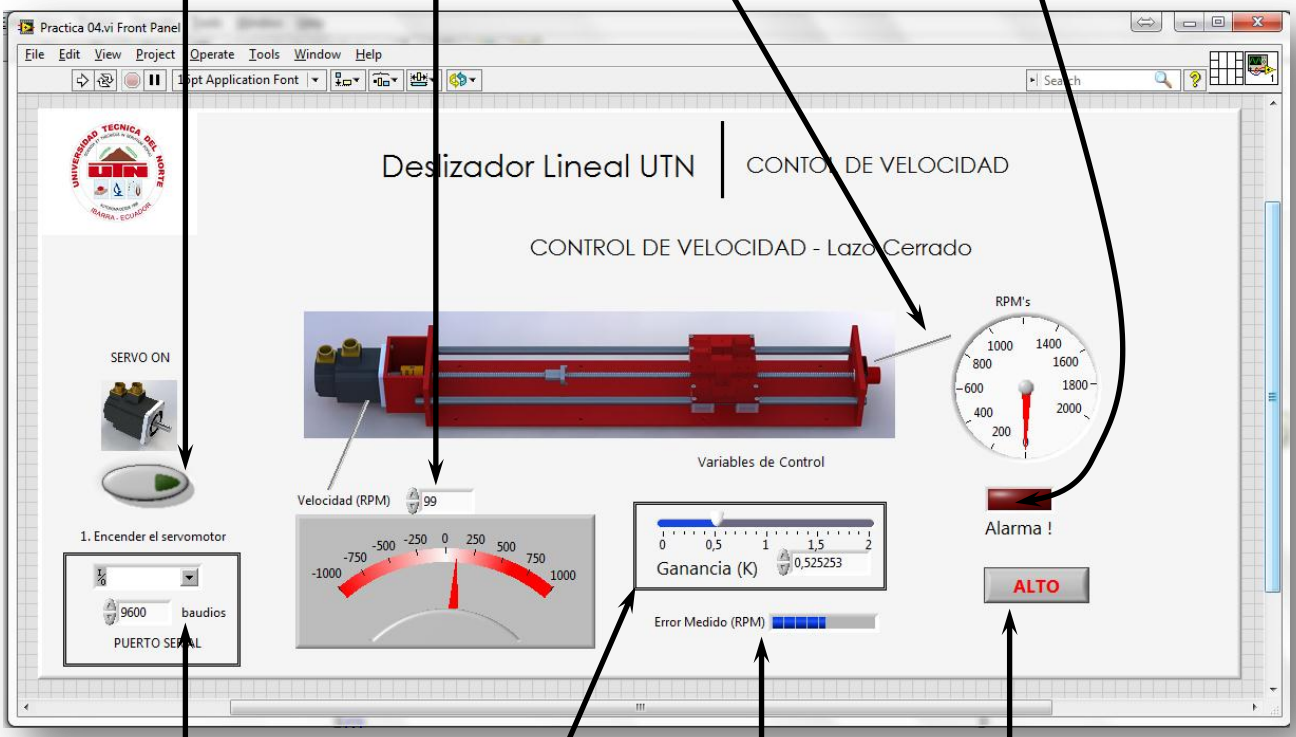
PRACTICA CUATRO: Control de velocidad del deslizador (Lazo Cerrado)

CONTROL: Encender Servo

CONTROL: Seleccionar Velocidad

INDICADOR: Velocidad Medida

INDICADOR: Alarma Servodrive



CONTROL: Puerto Serial mbed

CONTROL: Ganancia (Multiplica el error)

INDICADOR: Error medido en RPM's

CONTROL: Detener el programa

IMPORTANTE:

Para realizar esta práctica se debe desconectar la mesa del deslizador de la tuerca del tornillo, de esta manera se puede rotar el servomotor sin que la base del deslizador se mueva

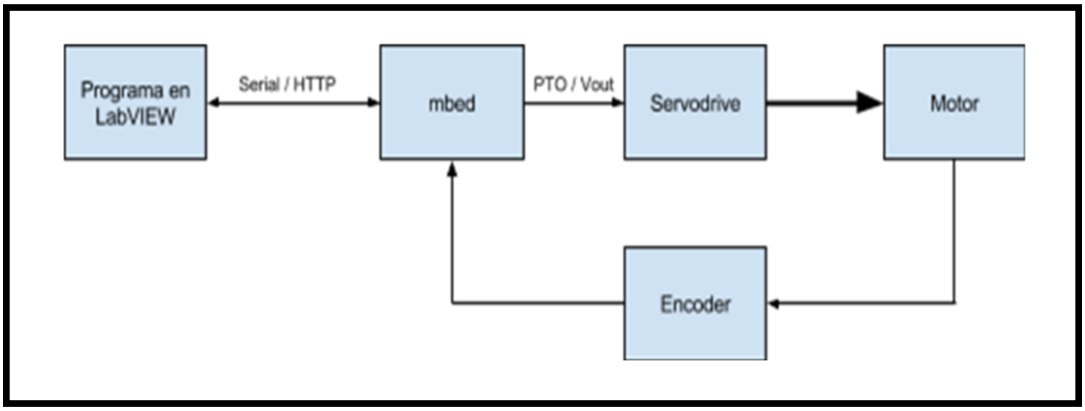
Objetivos.-

- Conocer las capacidades de control del sistema deslizador.
- Familiarizarse con las opciones de retro-alimentación del módulo

Introducción.-

Esta es una aplicación de control del módulo de servomecanismos en lazo cerrado. Permite el control de la velocidad del servomotor medida a través del encoder Hohner ubicado al final del tornillo del deslizador.

Ilustración 25. Sistema de control en lazo cerrado



Fuente: Autor

El control que se ejerce es simple, el error medido por el encoder es amplificado por una ganancia (K) la cual se controla desde el panel frontal, El programa rectifica este error para que la velocidad medida sea igual a la velocidad escogida por el estudiante.

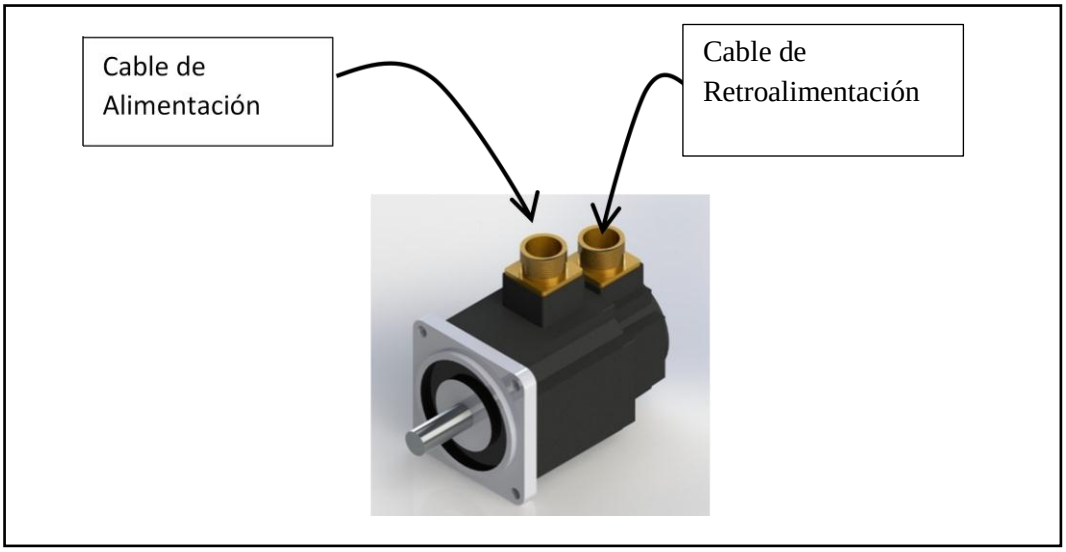
Materiales.-

1	Servomotor GSK 110 SJT (Conectado al deslizador)
1	Servodrive DA98D
1	Deslizador Lineal
1	Computador instalado LabVIEW con las librerías del deslizador.

Procedimiento.-

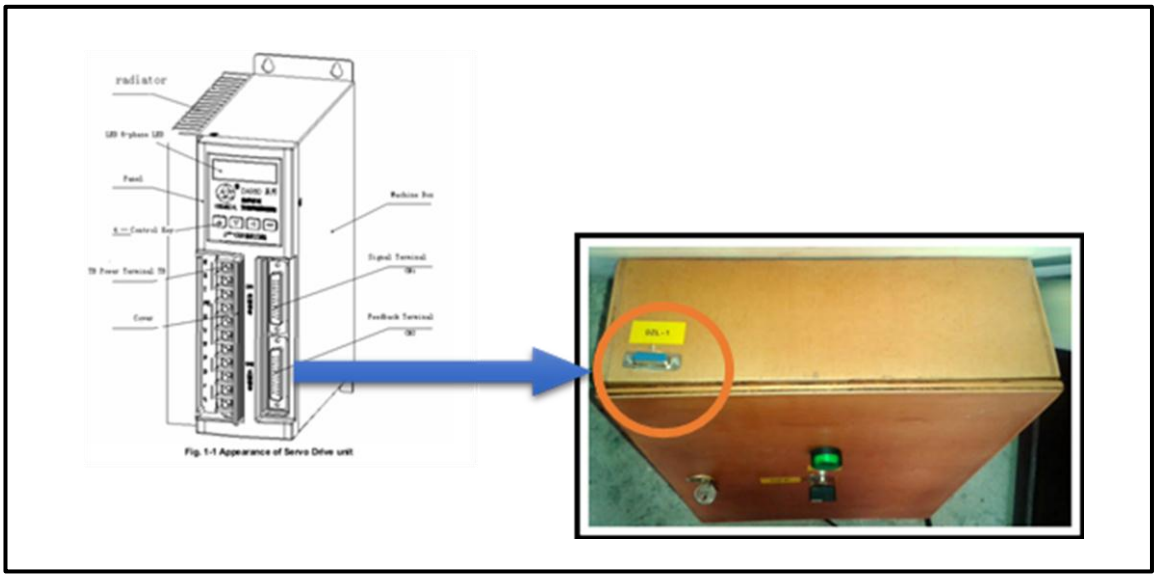
1. Se debe conectar los cables de alimentación y retroalimentación al servomotor.

Cuadro 14. Conexiones al servomotor

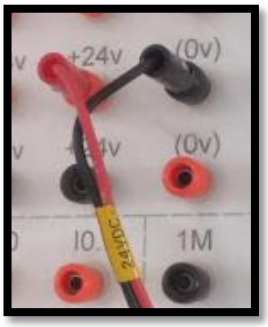


2. Se debe conectar el cableado de control del servodrive con la placa electrónica del deslizador destinada a este propósito. Tanto el cable de conexión como la tarjeta cuentan con etiquetas para su correcta conexión.

Cuadro 15. Conexión cable Servodrive - Deslizador



3. Se deben conectar los cables de alimentación de 24VDC.

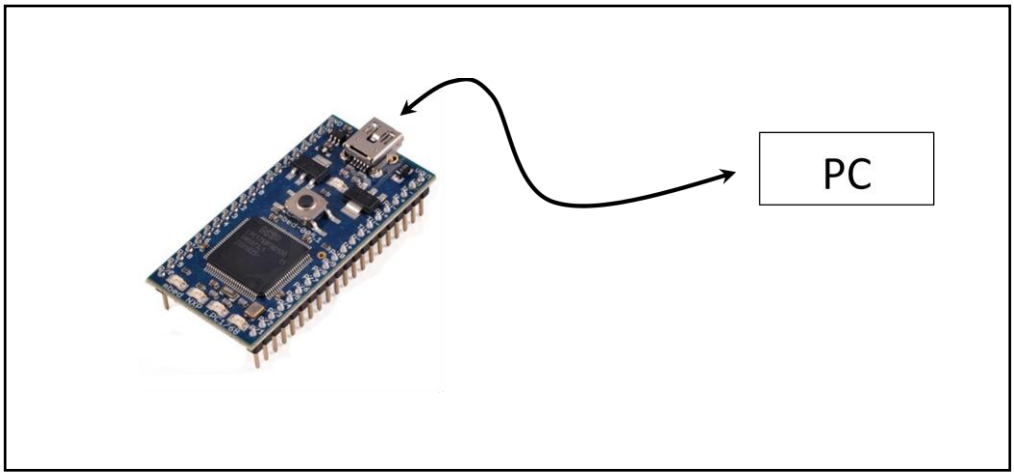


4. Se enciende la alimentación al servodrive, y se programan los siguientes parámetros.

PA	Valor
3	0
4	1
20	1
23	1000
33	250
39	0
40	0
43	1
46	0

5. Se conecta la placa electrónica principal al computador, cuidando la configuración que se ha escogido. Se recomienda utilizar la comunicación Serial USB.

Cuadro 16. Conexión microcontrolador - PC.



6. Se enciende la placa electrónica principal, en LabVIEW escogemos el puerto COM asignado al microcontrolador MBED y estamos listos para ejecutar la práctica.

Resultados esperados.-

El error medido por el encoder y el error medido por el servomotor variar en pequeña medida, esto se debe en parte a la mayor precisión con la que cuenta el encoder del servomotor y en parte a la vibración mecánica que experimenta el módulo en altas velocidades.

El funcionamiento de la práctica dependerá de la configuración de la velocidad máxima del servodrive, la cual debe ser de 1000 (RPM).

La velocidad a la cual se ejecutará el lazo de control depende de la velocidad de comunicación serial, se recomienda aumentar a 115200 baudios y observar la como afecta esto al programa.

Programación.-

La programación se centra en la velocidad escogida y la velocidad medida, el error se calcula como la diferencia de estos dos parámetros, este error luego se multiplica por la ganancia escogida para el sistema, una vez que se tiene calculado el error lo suma a la velocidad escogida, corrigiendo de esta forma la velocidad.

Fig. 85. Diagrama de Bloques - Práctica 04.

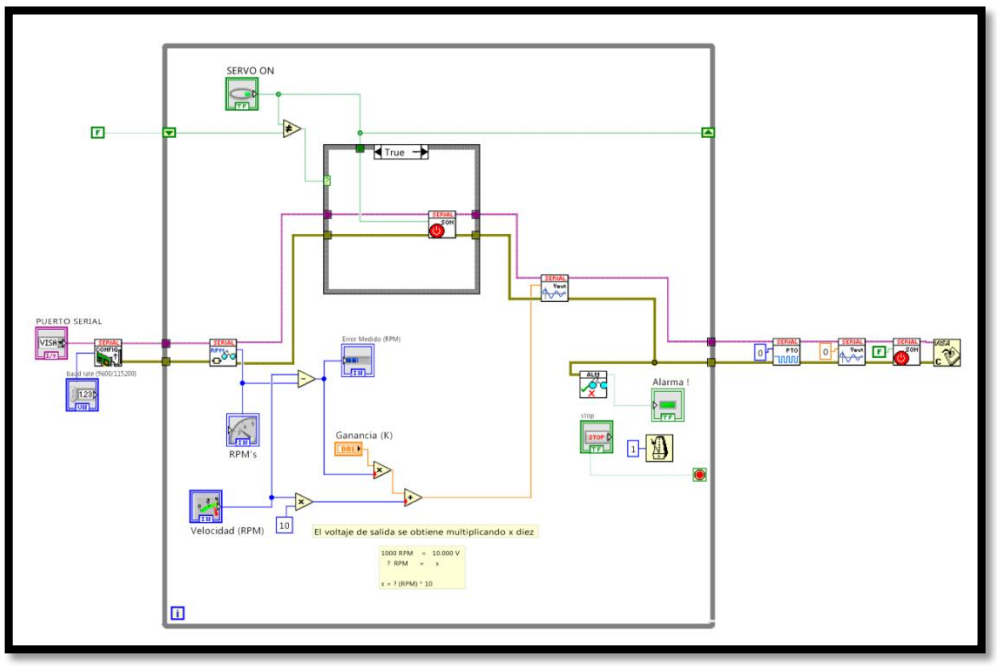
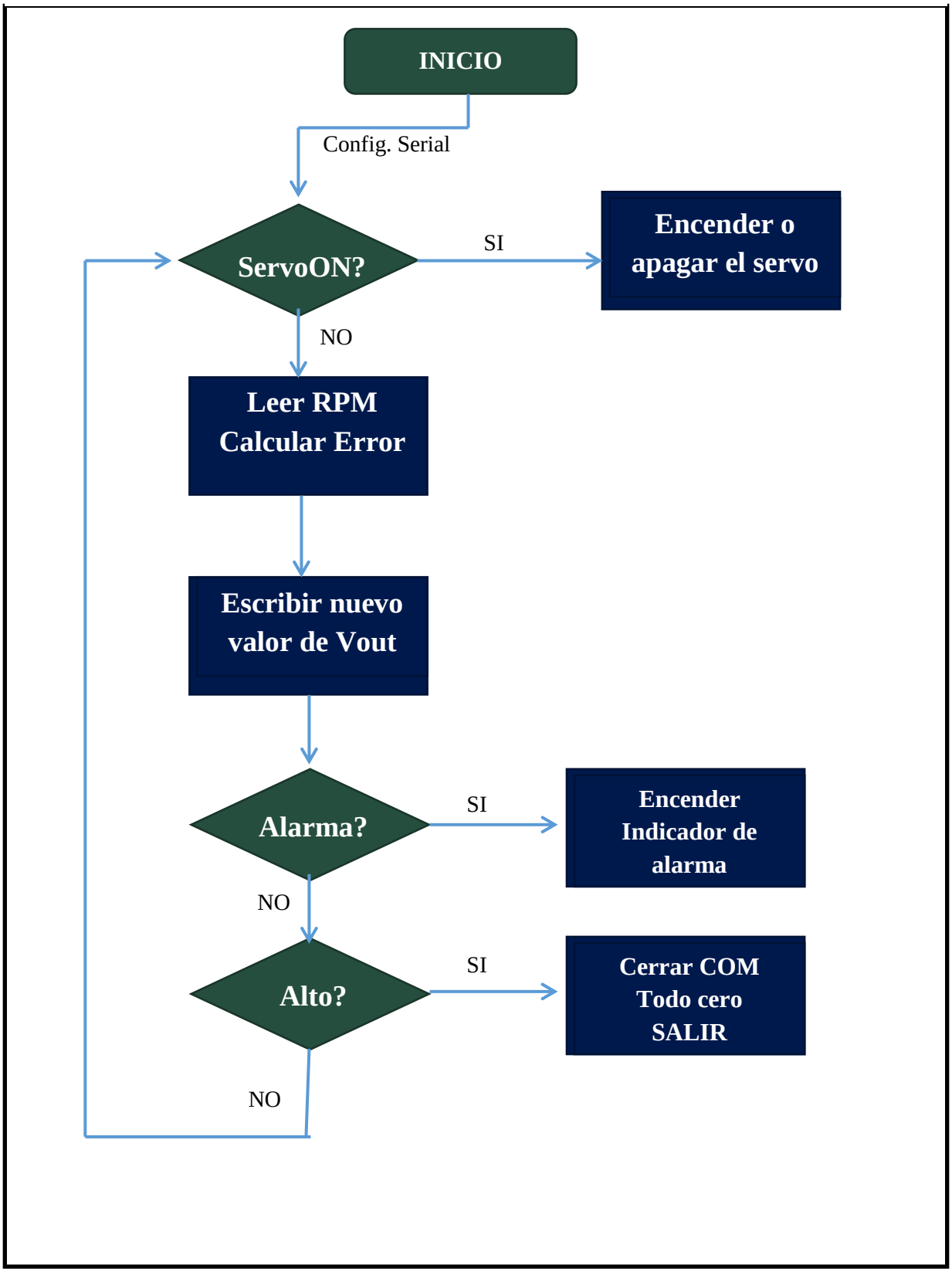


Fig. 86. Flujo grama - Practica 04



Fuente: Autor

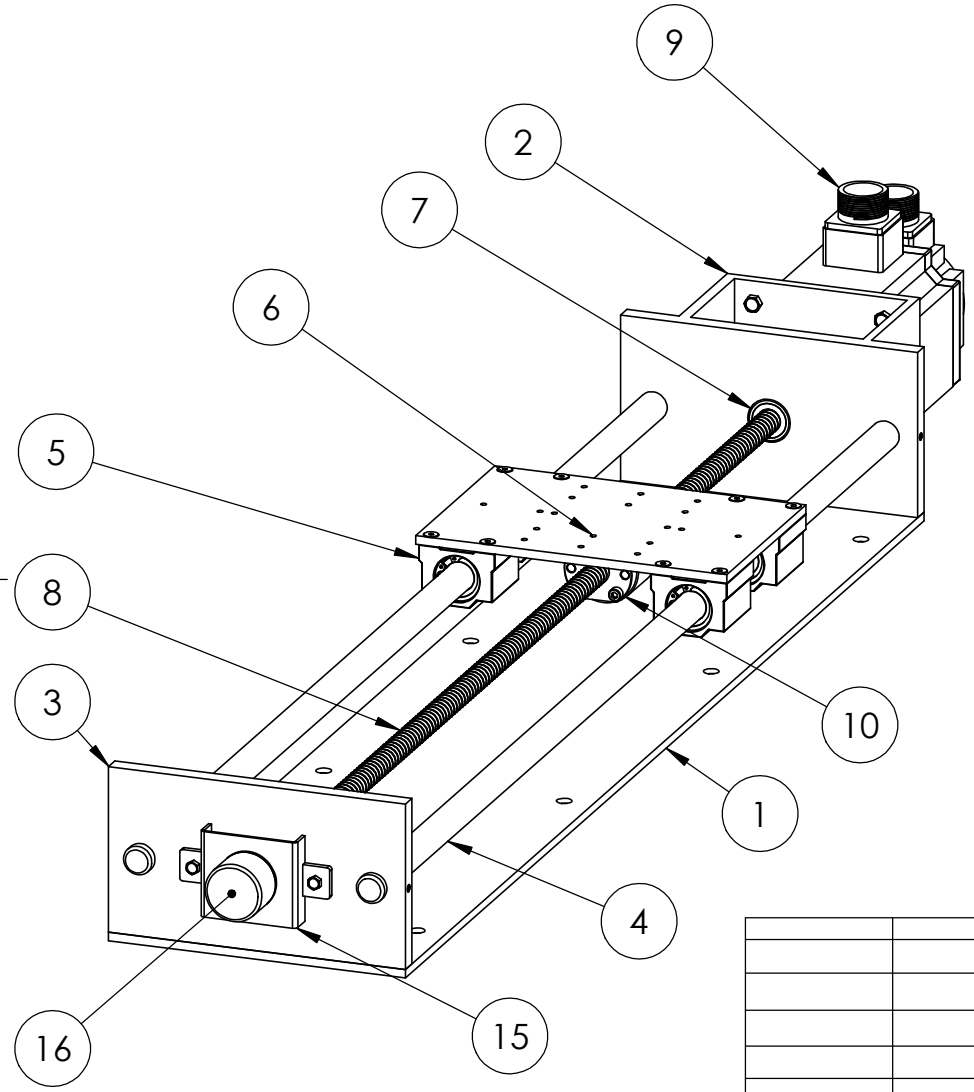
Ejercicios y recomendaciones.-

1. Se puede analizar el código fuente del programa de LabVIEW para entender como se ha programado la práctica.

2. EJERCICIO: Se pueden cambiar los parámetros tanto del servomotor como de la programación en LabVIEW y experimentar con diferentes velocidades, tomando en cuenta que no se debe superar las 1500 RPM.

3. EJERCICIO: Se puede añadir al programa una gráfica la cual indique la velocidad deseada con la velocidad medida y comprobar el error existente y el tiempo que demora en rectificarse.

ANEXO 3. PLANOS MECÁNICOS



ITEM NO.	PartNo	DESCRIPCIÓN	CANT.
1	placa_base	Base del deslizador	1
2	jaula_motor	Sirve para sujetar el servomotor	1
3	placa_encoder	Placa al extremo del encoder	1
4	Guia	Barra lateral	2
5	cojinete_rodamiento	Cojinete del rodamiento lineal	4
6	base_carro	base carro deslizador	1
7	rodamiento_10mm	Rodamiento radial	2
8	tornillo_16-5	Tornillo de bolas	1
9	Servo_110SJT	Servomotor	1
10	Nuez	Tuerca del tornillo	1
15	sujeta_encoder	Placa sujeción encoder	1
16	encoder_hohner	Encoder	1

Tolerancia: + - 1	Masa: 45 KG	Materiales: ACERO ASTM - A500

	Fecha:	Nombre:
Dib.	04/07/204	PALACIOS E.
Rev.	07/07/2014	TERÁN D.
Apro.	07/07/2014	TERÁN D.

Firma: *Ernesto P.*

Denominación: Deslizador Lineal UTN

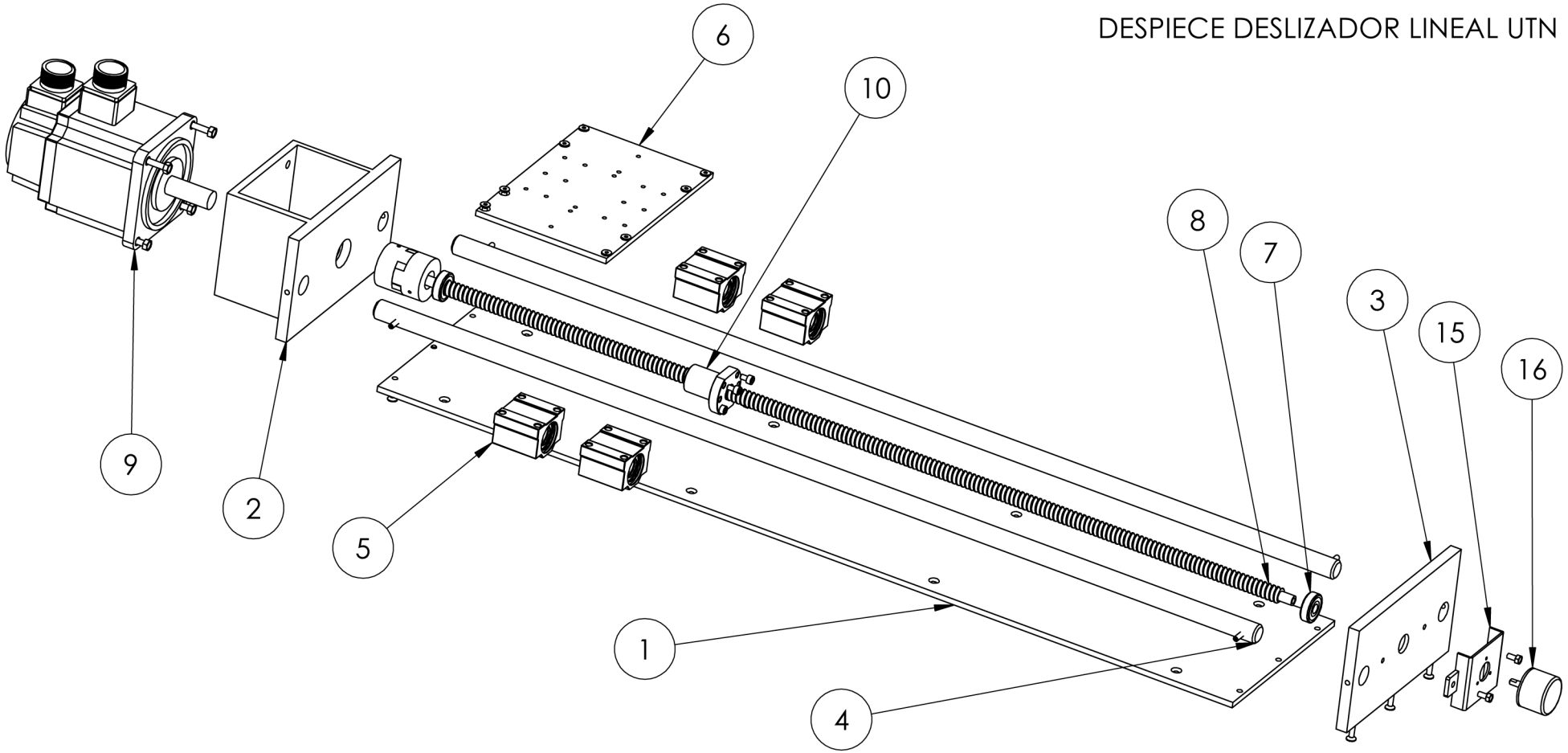
Escala: **1:5**

Número del dibujo: DESLIZADOR_LINEAL_UTN

Registro:

Sustitución:

DESPIECE DESLIZADOR LINEAL UTN



Nota: Lista de matrial en número de sustitución DESLIZADOR_LINEAL_UTN

Tolerancia: + - 1	Masa: 45 KG	Materiales: ACERO ASTM - A500

	Fecha:	Nombre:
Dib.	04/07/204	PALACIOS E.
Rev.	07/07/2014	TERÁN D.
Apro.	07/07/2014	TERÁN D.

Firma: *Ernesto P.*

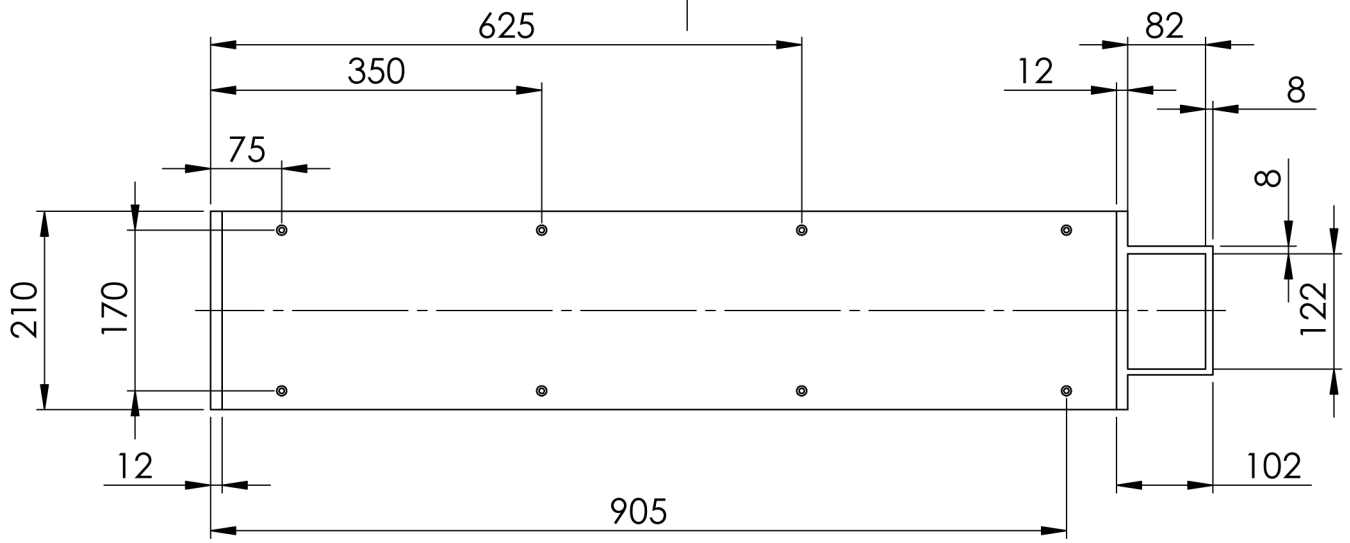
Denominación: **Deslizador Lineal UTN**

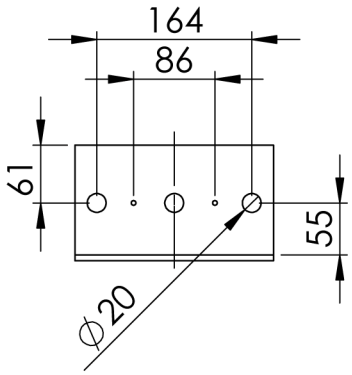
Escala: 1:5

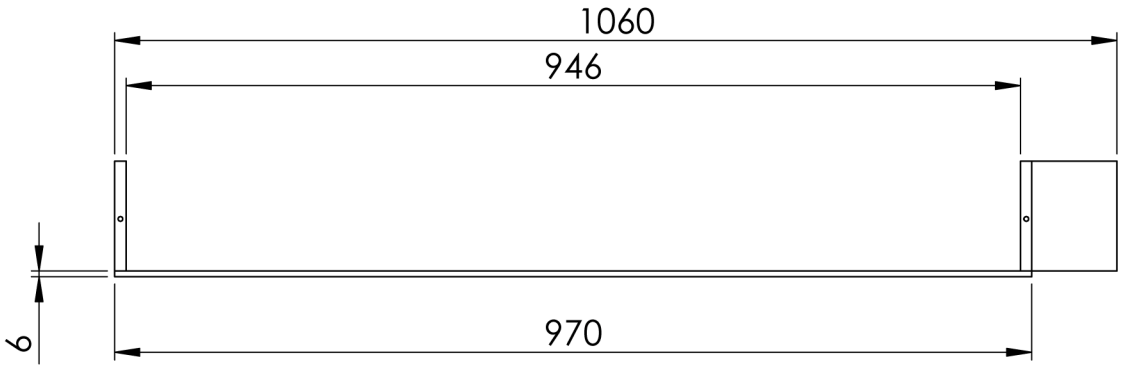
Número del dibujo: DESLIZADOR_DISSASEMBLY

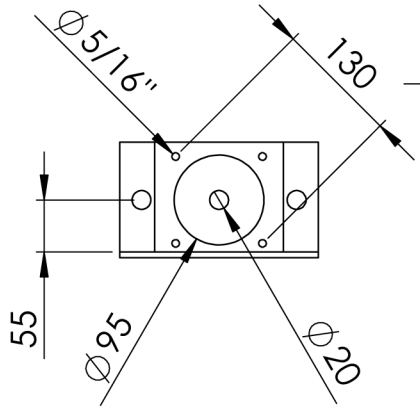
Registro:

Sustitución:



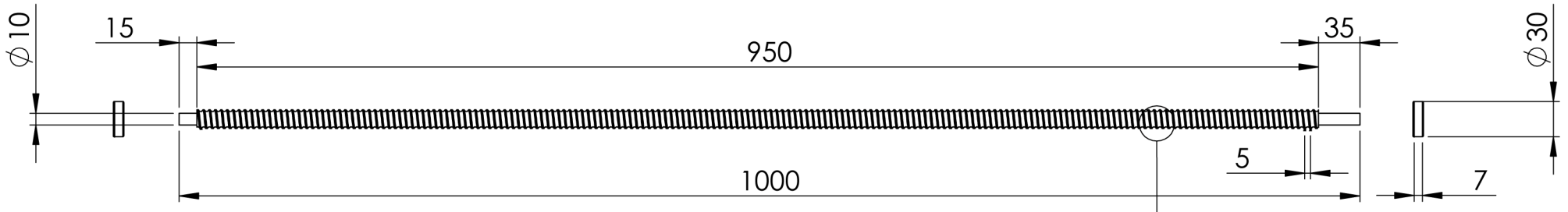


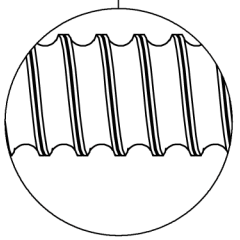




Tolerancia: + - 1			Masa: 40 KG	Materiales: ACERO ASTM - A500
	Fecha:	Nombre:	Denominación: Estructura deslizador	Escala: 1:10
Dib.	04/07/204	PALACIOS E.		
Rev.	07/07/2014	TERÁN D.		
Apro.	07/07/2014	TERÁN D.		
Firma: Ernesto P.			Número del dibujo: DZL-UTN-01	Registro:
			Sustitución:	Unidades: milímetros

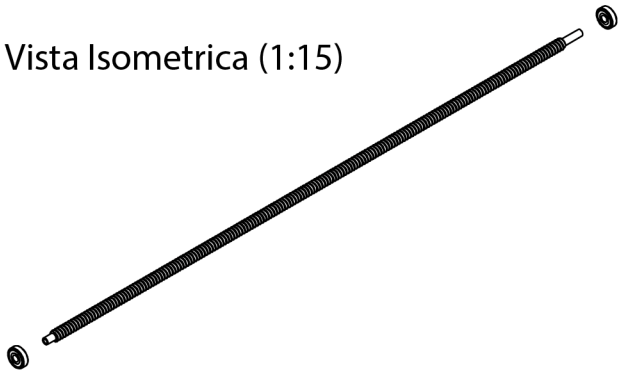
TORNILLO DE BOLAS





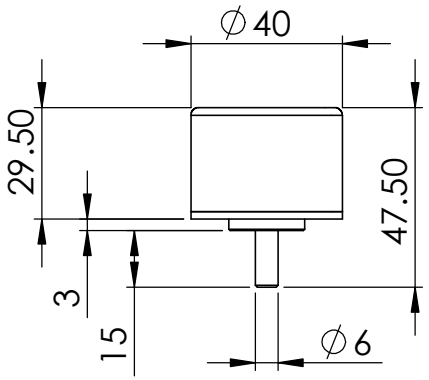
AA (1 : 1)

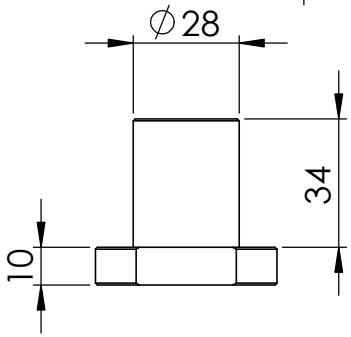
Vista Isometrica (1:15)

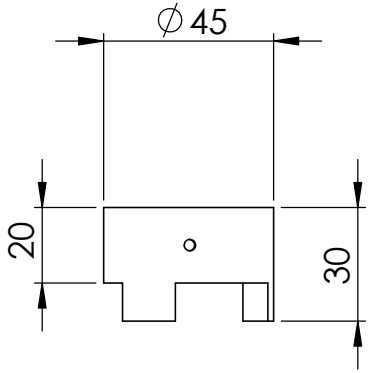


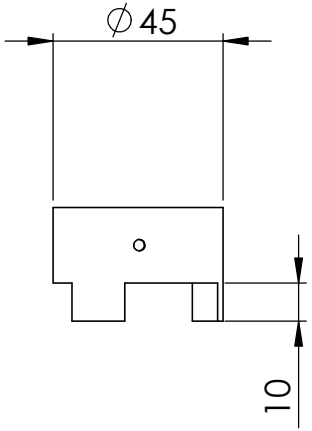
Tolerancia: + - 1			Masa: 2 KG	Materiales: ACERO ENDURECIDO ASTM - A500
	Fecha:	Nombre:	Denominación: Tornillo de bolas	Escala: 1:10
Dib.	04/07/204	PALACIOS E.		
Rev.	07/07/2014	TERÁN D.		
Apro.	07/07/2014	TERÁN D.		
Firma: Ernesto P.			Número del dibujo: DZL-UTN-02	Registro:
			Sustitución:	Unidades: milímetros

ELEMENTOS EJE VISTA SUPERIOR

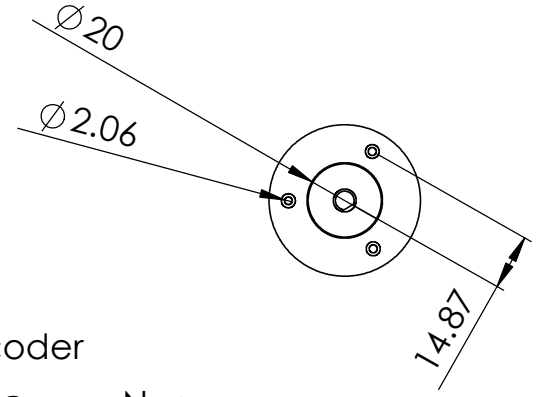


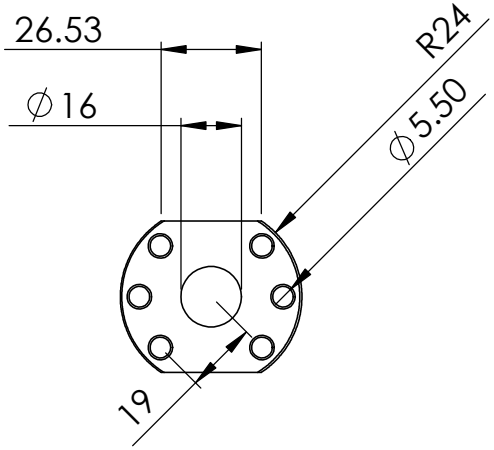


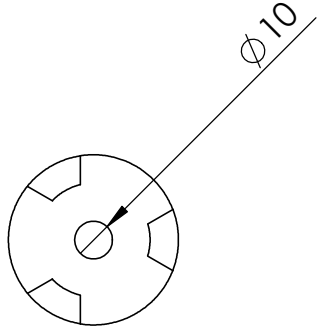




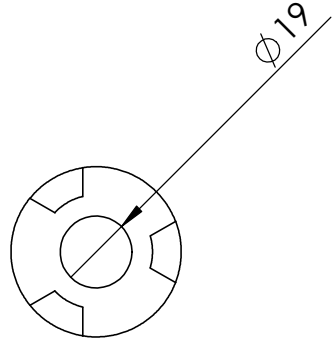
ELEMENTOS EJE VISTA FRONTAL







Extremo tornillo



Extremo Motor

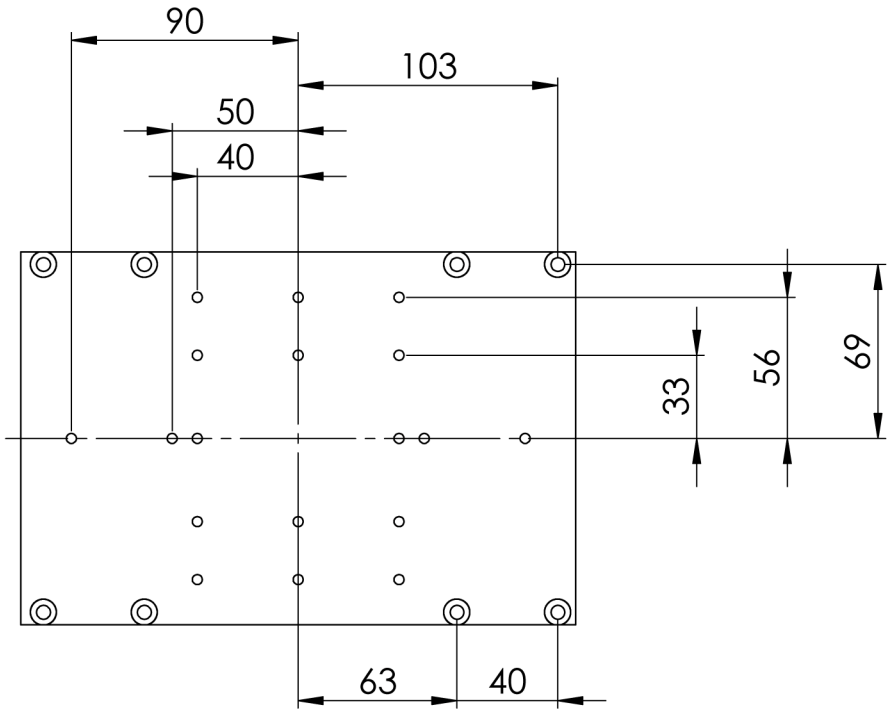
Encoder

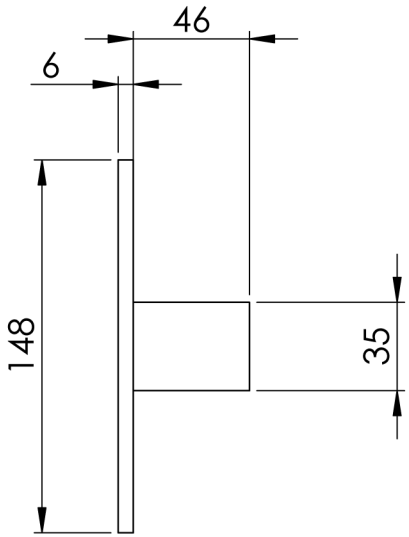
Nuez

Matrimonio

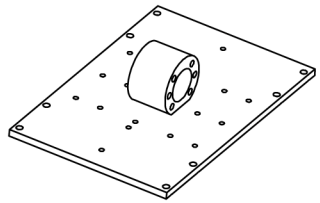
Tolerancia: + - 1		Masa: 2 KG	Materiales: ACERO ASTM - A500	
	Fecha:	Nombre:	Denominación: Elementos eje	Escala: 1:2
Dib.	04/07/204	PALACIOS E.		
Rev.	07/07/2014	TERÁN D.		
Apro.	07/07/2014	TERÁN D.		
Firma: Ernesto P.			Número del dibujo: DZL-UTN-03	Registro:
			Sustitución:	Unidades: milímetros

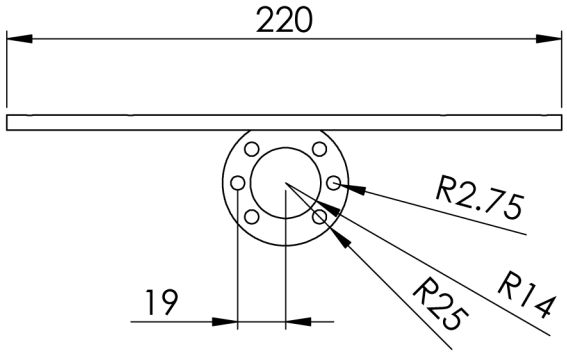
PLACA CARRO DESLIZADOR





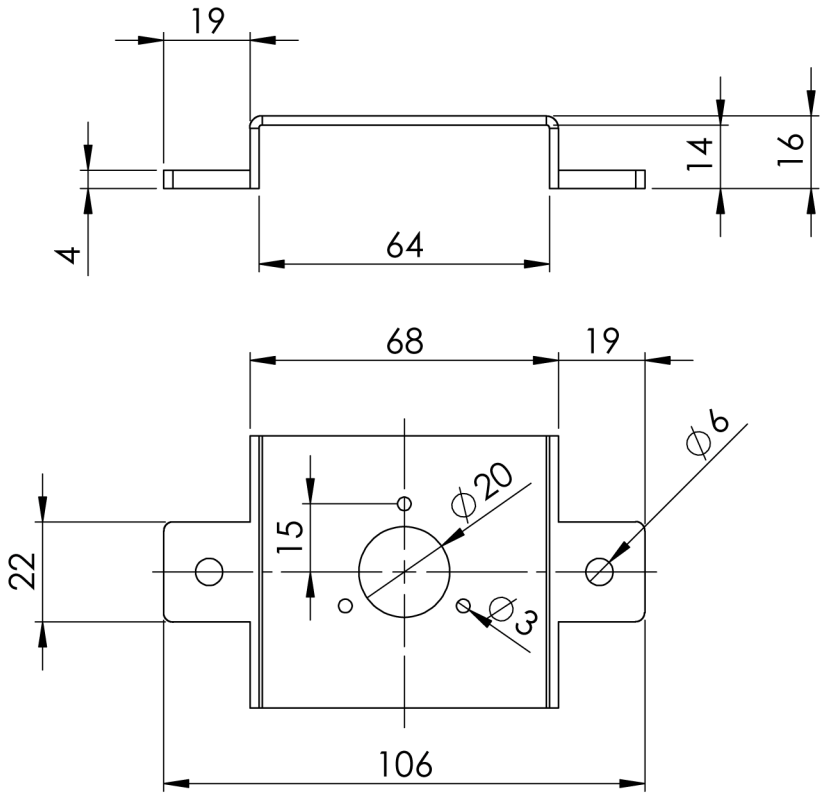
VISTA ISOMÉTRICA
(1:5)





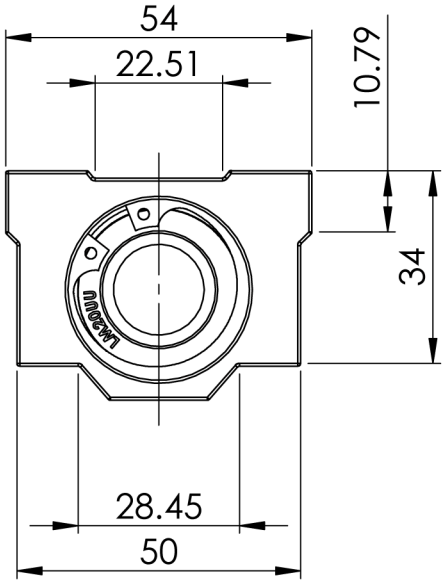
Tolerancia: + - 1		Masa: 3 KG	Materiales: ACERO ASTM - A500	
	Fecha:	Nombre:	Denominación: Placa deslizador	Escala: 1:5
Dib.	04/07/204	PALACIOS E.		
Rev.	07/07/2014	TERÁN D.		
Apro.	07/07/2014	TERÁN D.		
Firma: Ernesto P.			Número del dibujo: DZL-UTN-04	Registro:
			Sustitución:	Unidades: milímetros

SUJETA ENCODER

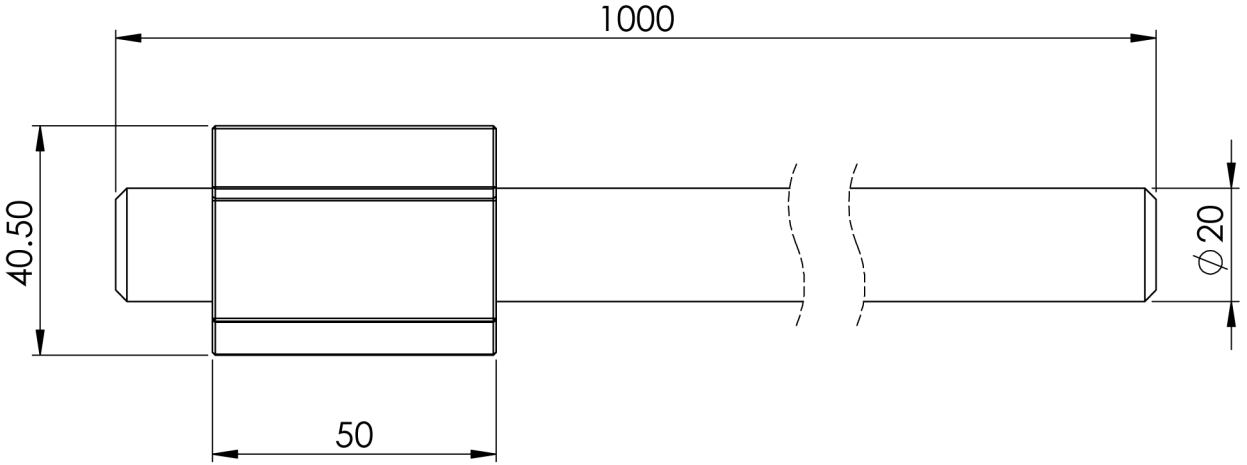


Tolerancia: + - 1	Masa: 0.1 KG	Materiales: ACERO ASTM - A500		
	Fecha:	Nombre:	Denominación: Sujeta Encoder	Escala: 1:2
Dib.	04/07/204	PALACIOS E.		
Rev.	07/07/2014	TERÁN D.		
Apro.	07/07/2014	TERÁN D.		
Firma: Ernesto P.			Número del dibujo: DZL-UTN-05	Registro:
			Sustitución:	Unidades: milímetros

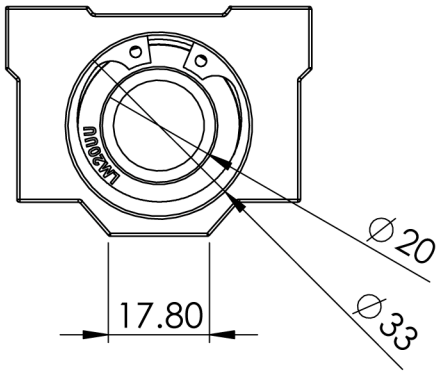
RODAMIENTO Y GUÍA



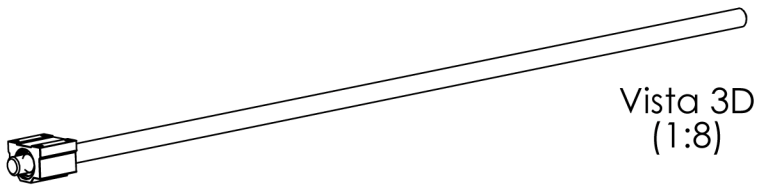
VISTA FRONTAL



VISTA LATERAL



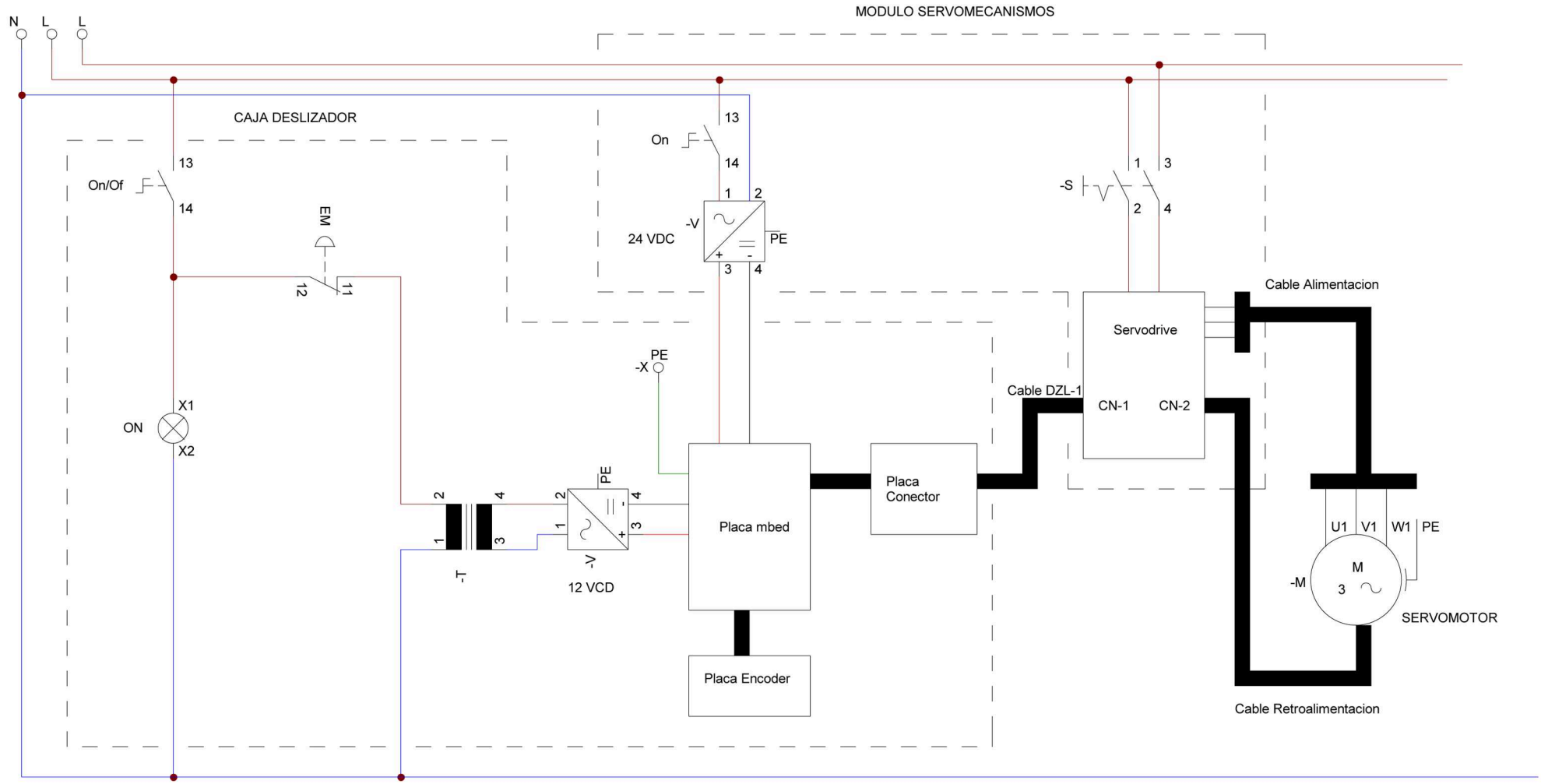
VISTA TRASERA



Vista 3D
(1:8)

Tolerancia: + - 1	Masa: 2 KG	Materiales: ACERO ASTM - A500		
	Fecha:	Nombre:	Denominación: Rodamiento y guía	Escala: 1:2
Dib.	04/07/204	PALACIOS E.		
Rev.	07/07/2014	TERÁN D.		
Apro.	07/07/2014	TERÁN D.		
Firma: Ernesto P.			Número del dibujo: DZL-UTN-06	Registro:
			Sustitución:	Unidades: milímetros

ANEXO 4. DIAGRAMA ELÉCTRICO



	Fecha	Nombre	Firmas	Entidad	Título	Fecha: 12-Jul-2014	Núm: 1 de 1
Dibujado	7/7/2014	E. Palacios		DZL-UTN-09	Diagrama eléctrico DZL	Archivo: DZL-UTN-09.cad	
Comprobado	11/7/2014	D. Terán					

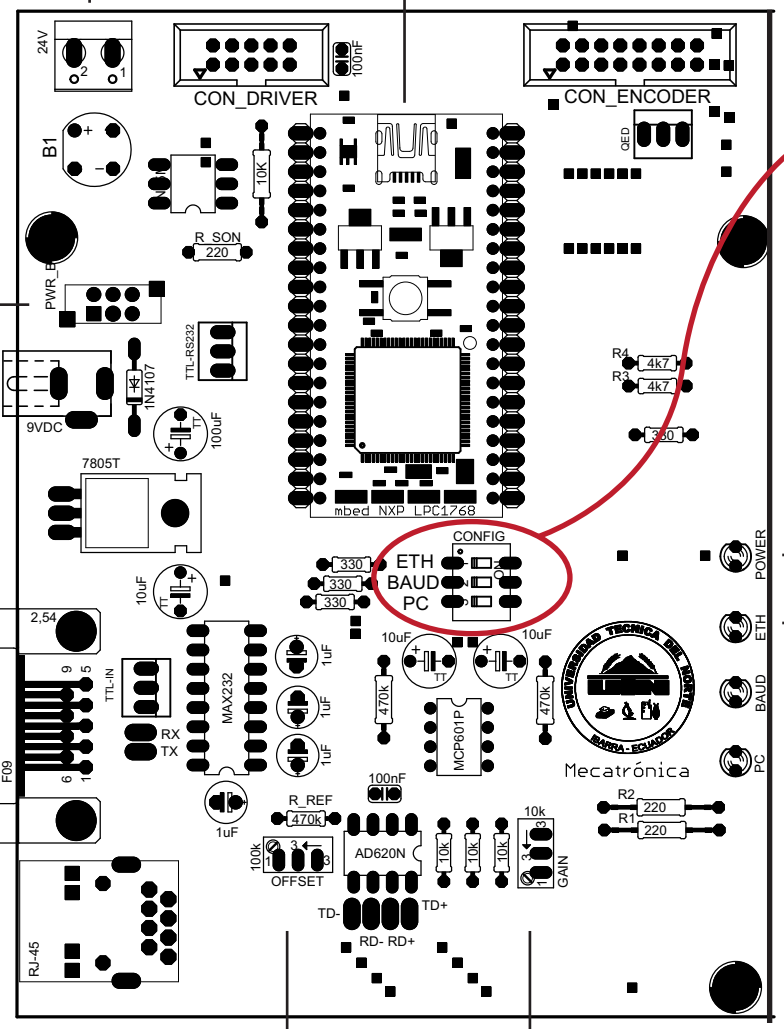
ANEXO 5. CIRCUITOS IMPRESOS

Diseñado por:
Ernesto Palacios
mecatronica.mid@gmail.com

Fuente de 24 VDC

Si se escoge la comunicación serial a PC, el cable Mini USB* debe ser conectado aquí. Se deben instalar los drivers en la computadora para que funcione:

[http://mbed.org/handbook/Windows-serial-configuration]



Botón de encendido del dispositivo. ON/OFF

Fuente de 9 VDC

Conector DB9 para comunicación serial RS-232

Conector RJ-45 para comunicación ethernet. long. max. 2.5metros

CONFIG
1 ETH
2 BAUD
3 PC

Se enciende al encender el dispositivo

1 Se enciende si la comunicación es ETHERNET.
Si el led esta apagado la comunicación es serial

2 Se enciende si la velocidad serial es 115200 baud.

Si el led esta apagado la velocidad serial es de 9600 baud.

3 Se enciende si la comunicación serial es a través de un cable USB* directo al PC.

Si el led esta apagado la comunicación serial es a través del conector DB9

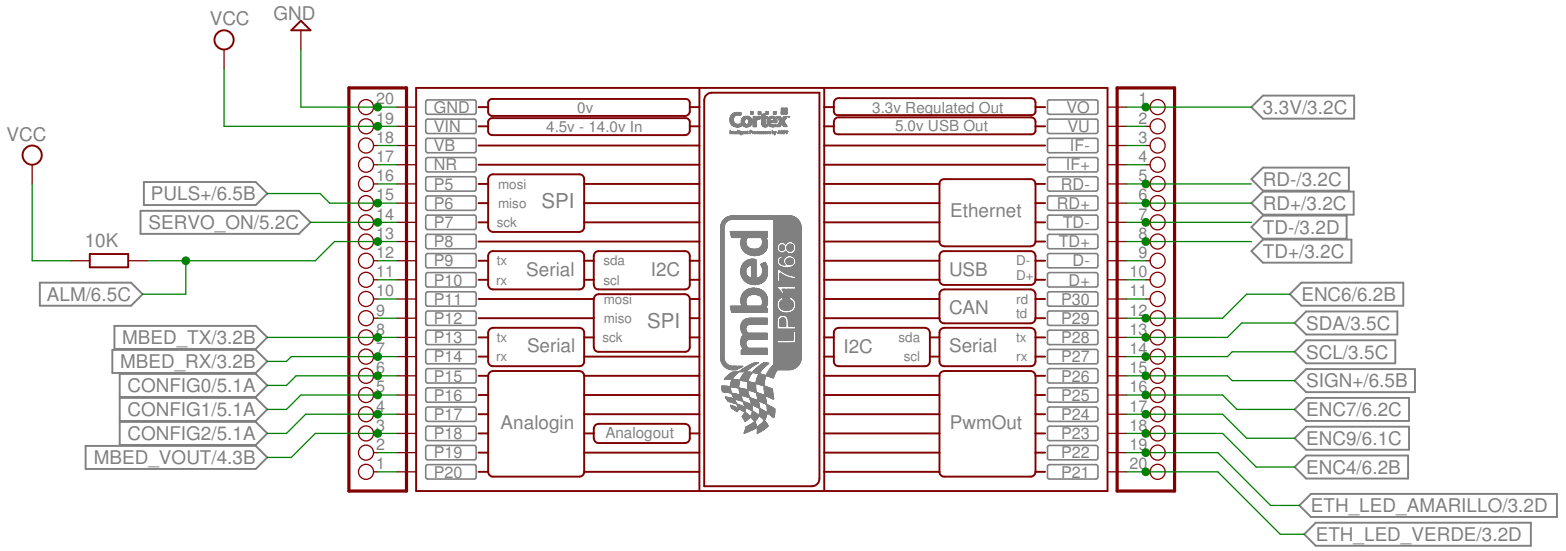
Ajusta el offset para la salida de voltaje analógica VCMD

Ajusta la ganancia para la salida de voltaje analógica VCMD

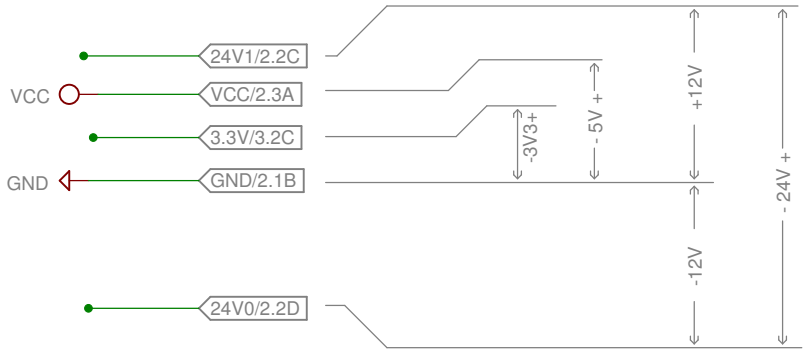
Formato serial: [8bits,Sin paridad, 1bit stop].

DHCP Ethernet: devuelve la IP através de serial USB*

MICRORONTROLADOR ARM MBED

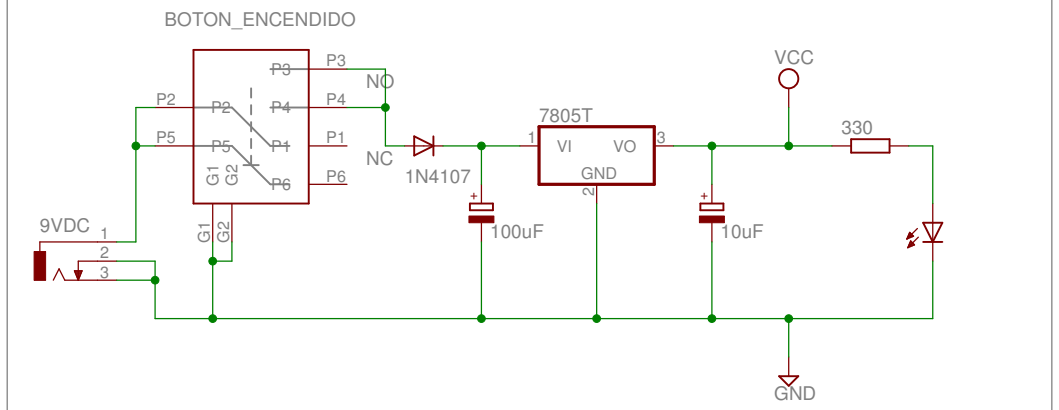


NIVELES DE VOLTAJE

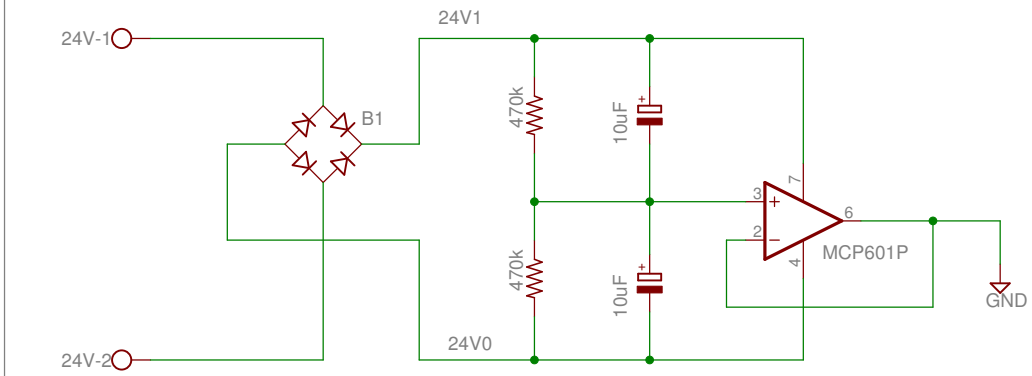


Autor: Ernesto Palacios	Microcontrolador
	Placa_mbed
	06/07/14 21:25
	Sheet: 1/6

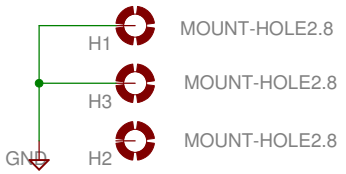
FUENTE ALIMENTACION 5V



FUENTE SIMÉTRICA +/- 12V

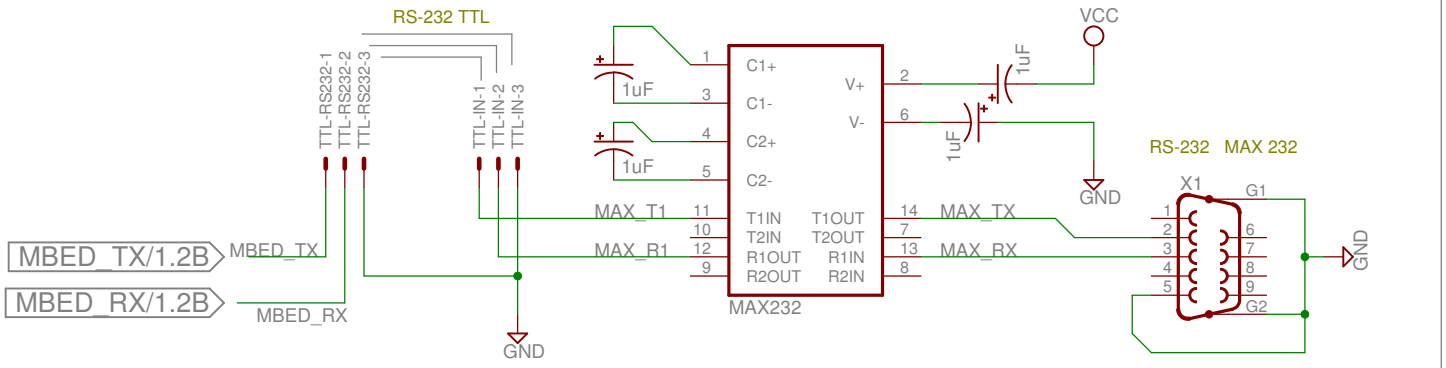


AGUJEROS DE MONTAJE

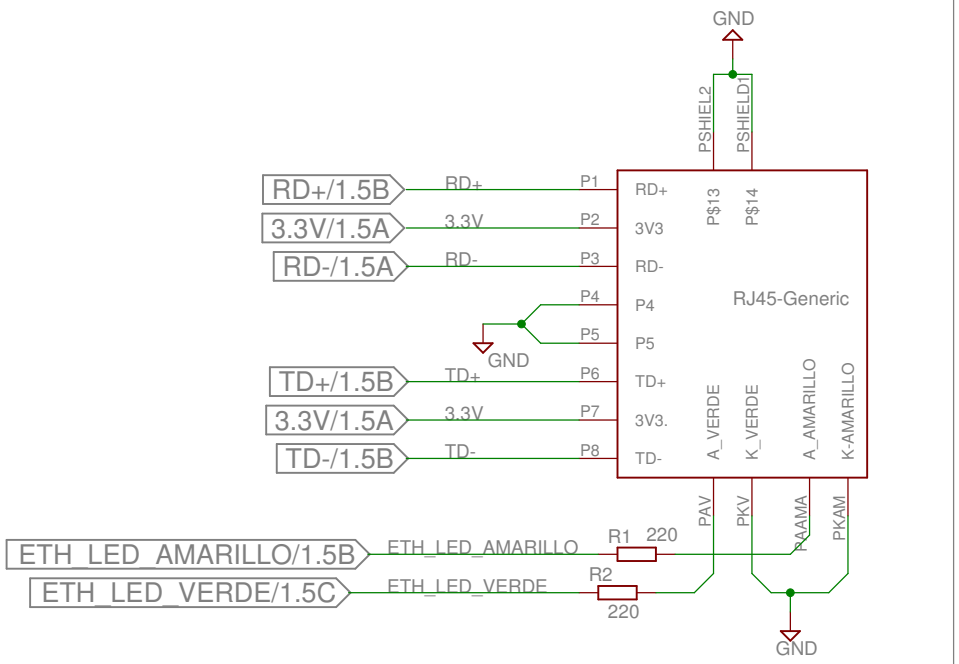


Autor: Ernesto Palacios	Fuentes de voltaje
	Placa_mbed
	06/07/14 21:25
	Sheet: 2/6

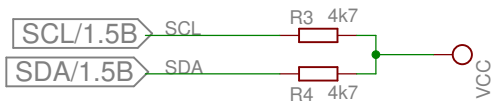
COMUNICACION SERIAL



CONECTOR RJ-45 MAG JACK

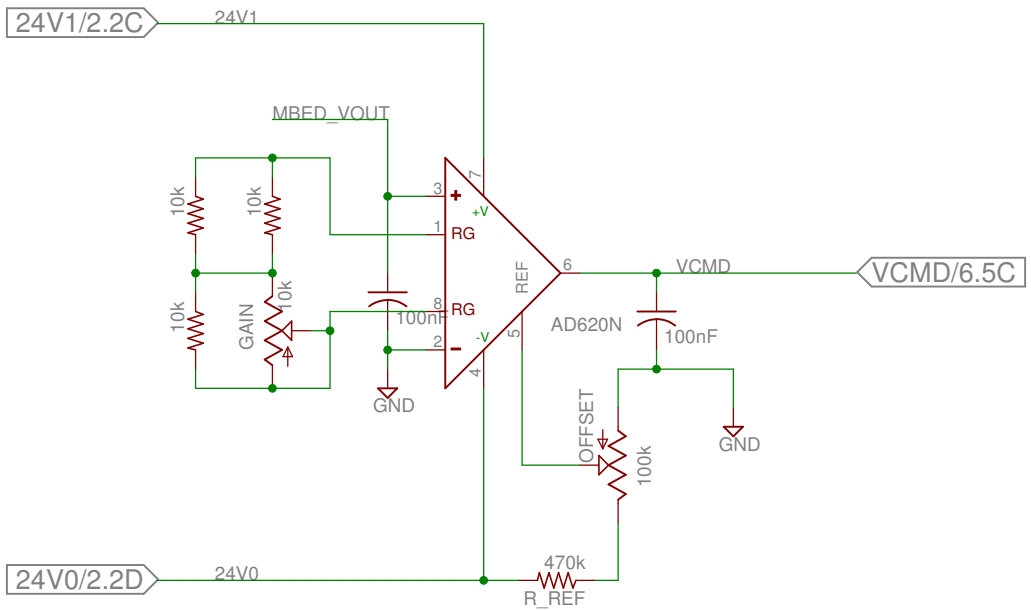


COMUNICACION I2C



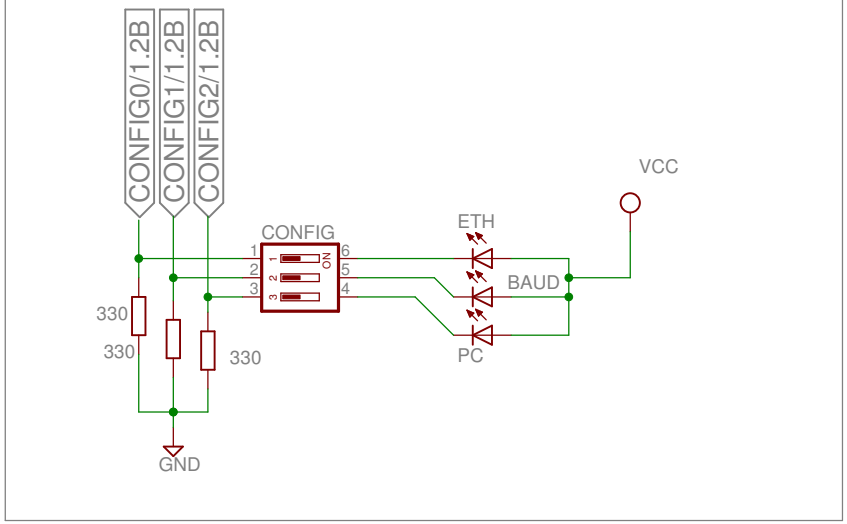
Autor: Ernesto Palacios	Comunicaciones
	Placa_mbed
	06/07/14 21:25
	Sheet: 3/6

AMPLIFICADOR OPERACIONAL

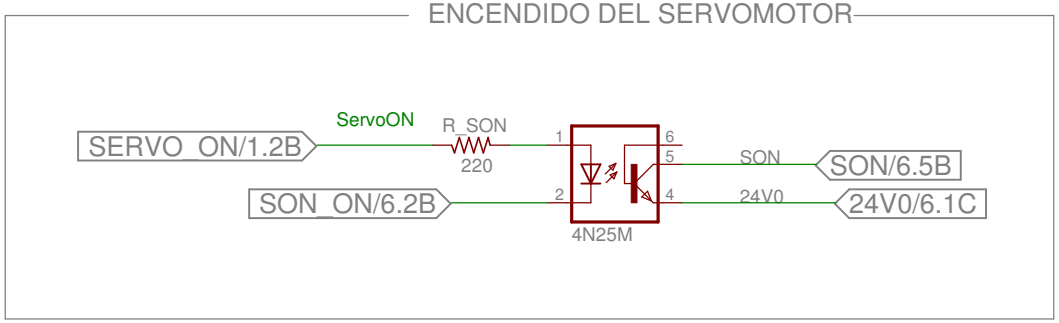


Autor: Ernesto Palacios	Salida de Voltaje
	Placa_mbed
	06/07/14 21:25
	Sheet: 4/6

SWITCH DE CONFIGURACIÓN

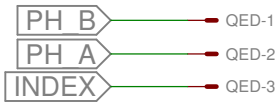


ENCENDIDO DEL SERVOMOTOR

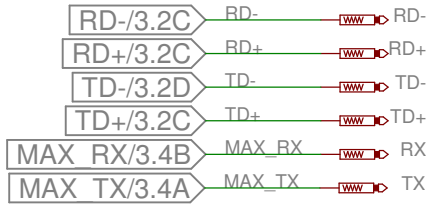


Autor: Ernesto Palacios	Configuraciones
	Placa_mbed
	06/07/14 21:25
	Sheet: 5/6

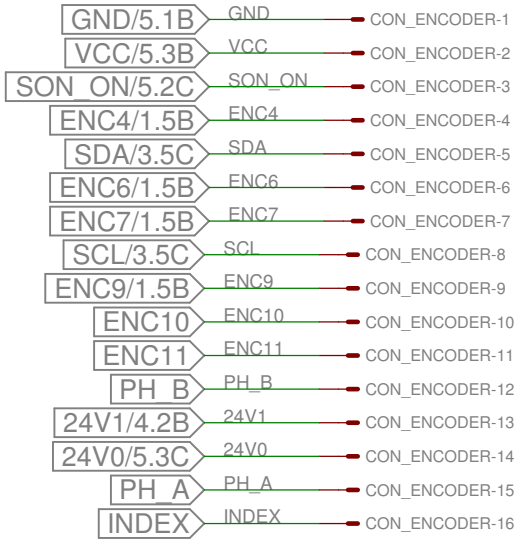
ENCODER DE CUADRATURA



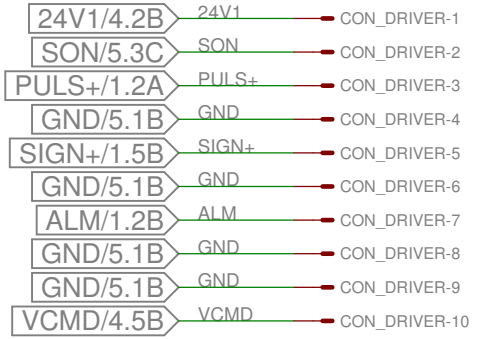
PUNTOS DE PRUEBA ETHERNET



CONECTOR CON PLACA DE ENCODERS

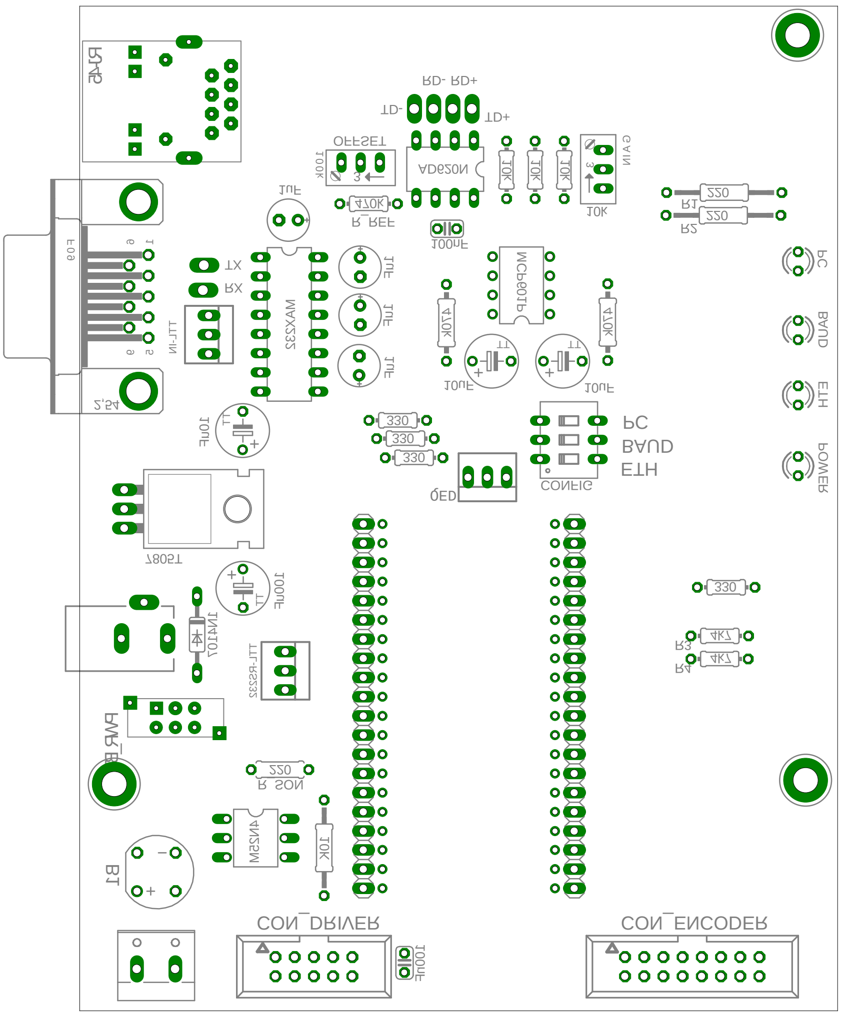


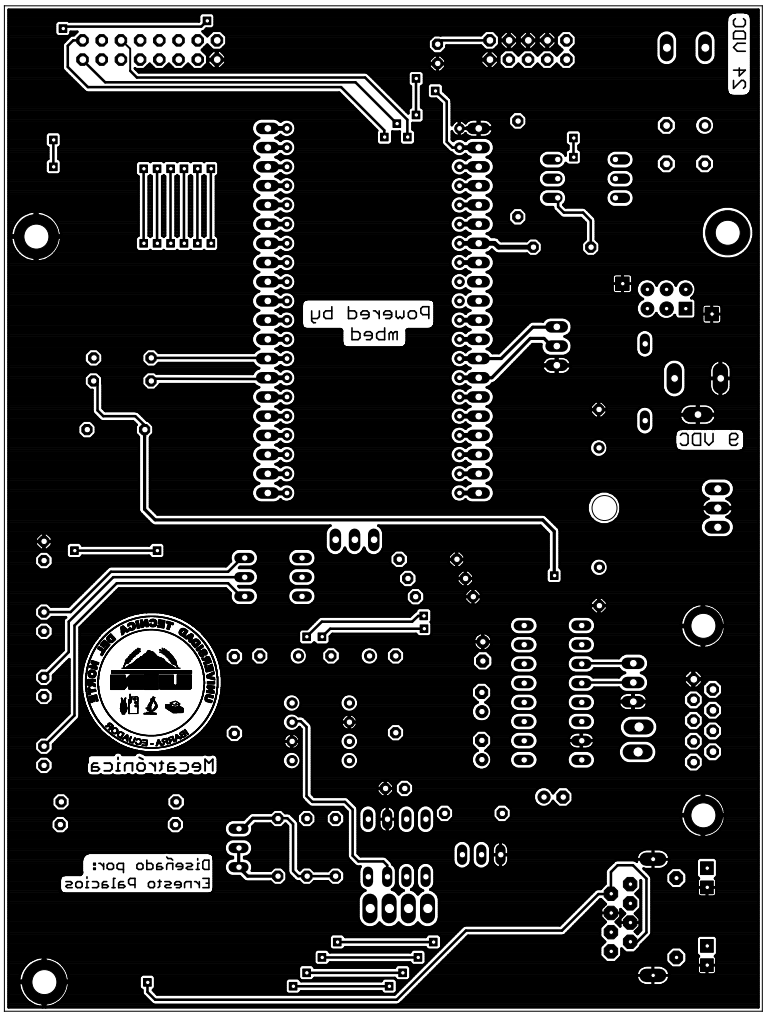
CONECTOR CON PLACA DEL DRIVER

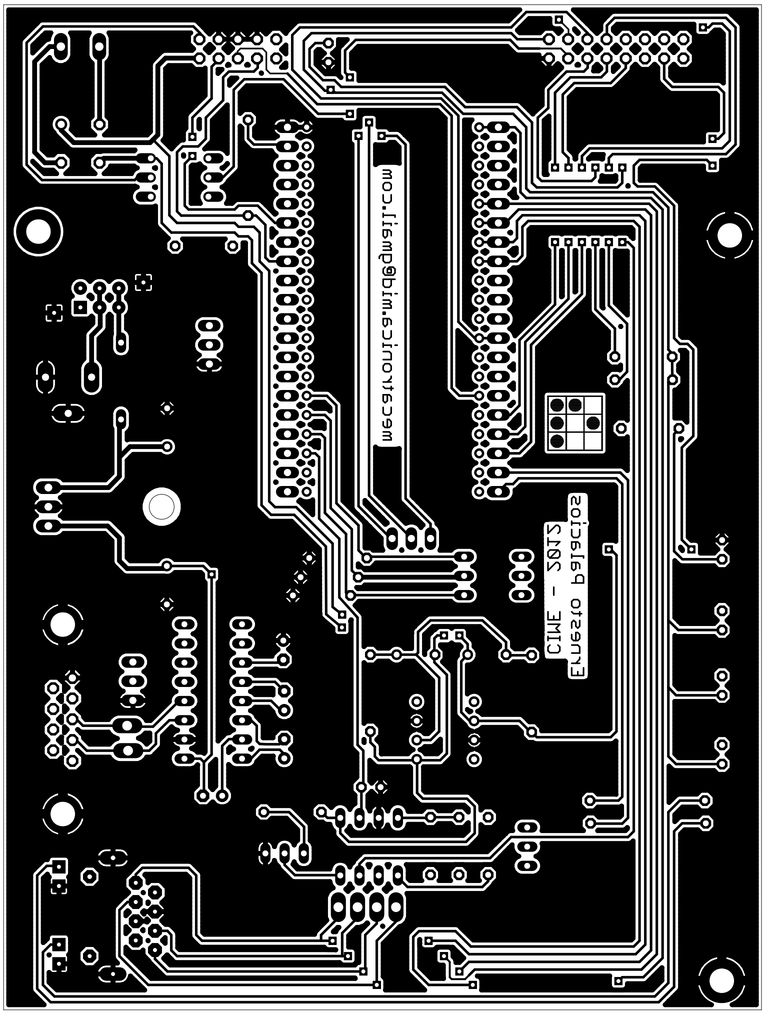


Autor: Ernesto Palacios	Conectores
	Placa_mbed
	06/07/14 21:25
	Sheet: 6/6

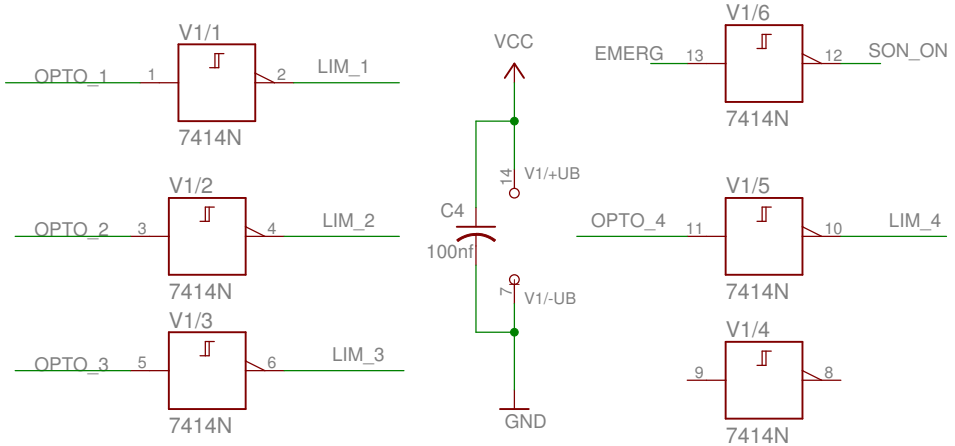
16/07/2014 16:54:38 mirrored D:\Proyecto de Tesis\mbed\Esquema_Circuito\version_final\Placa_mbed.brd



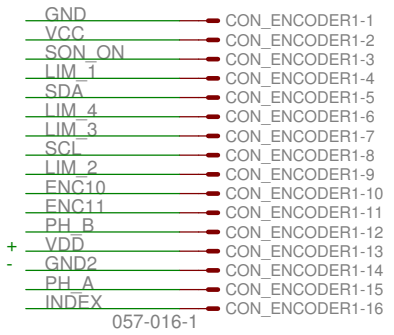

Powered by
mbed
9 VDC
Mecatrónica
Diseñado por:
Ernesto Palacios


mecatronica.mid@gmail.com
CIME - 2012
Ernesto Palacios

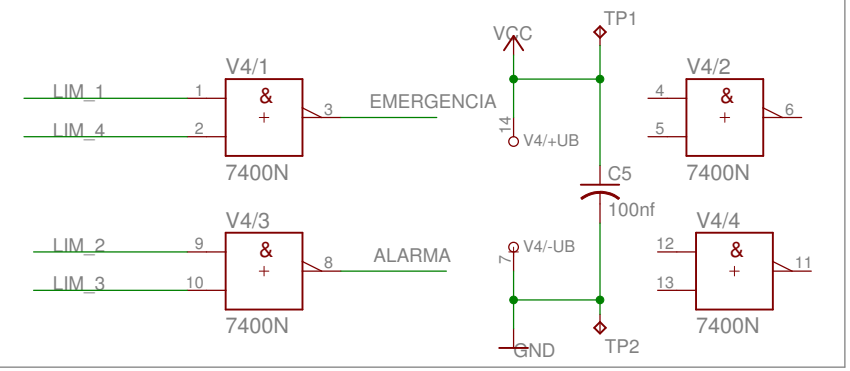
Compuerta NOT con Schmitt Trigger



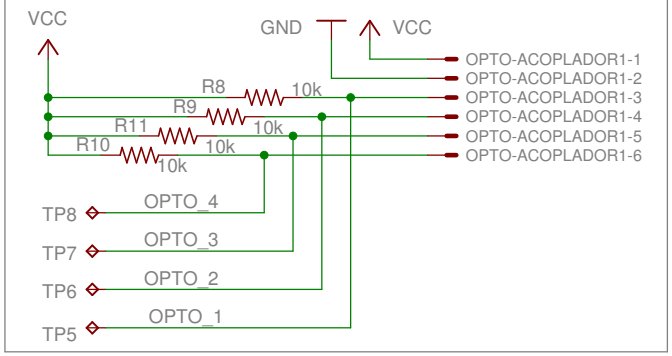
Conector IDC - 16 pines



Compuerta NAND

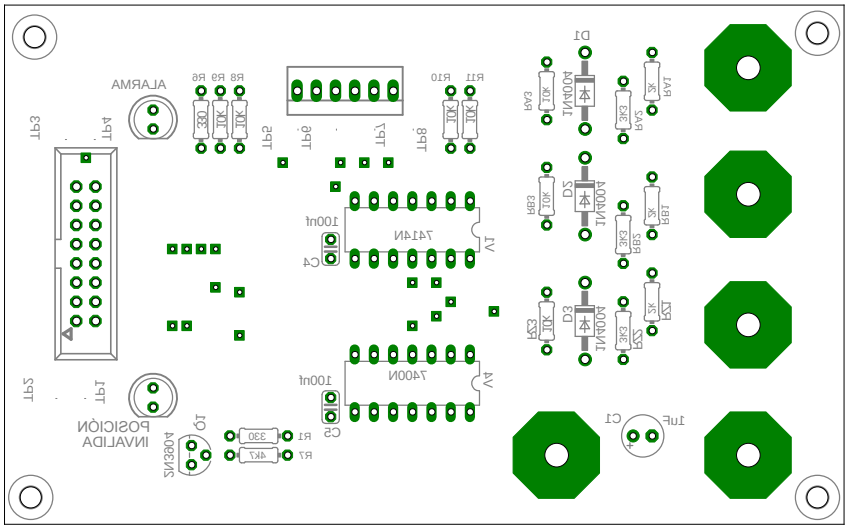


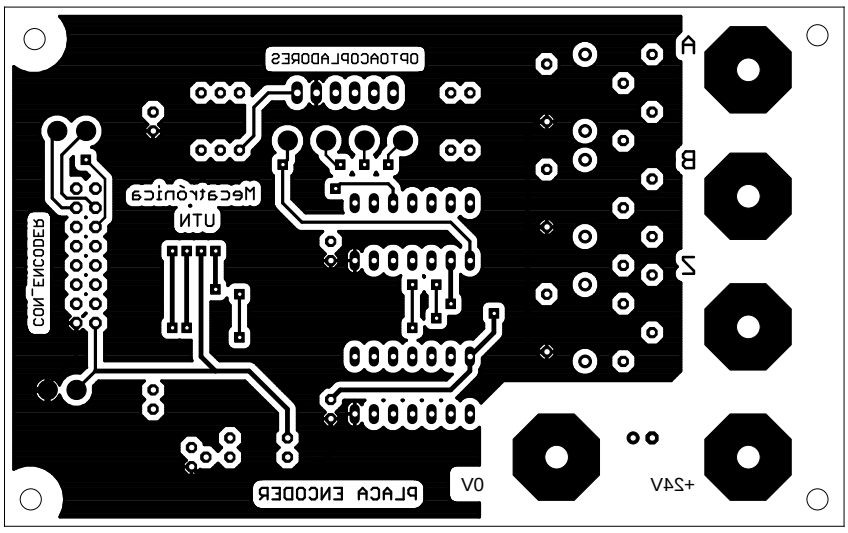
Salidas de sesores - optoacopladores

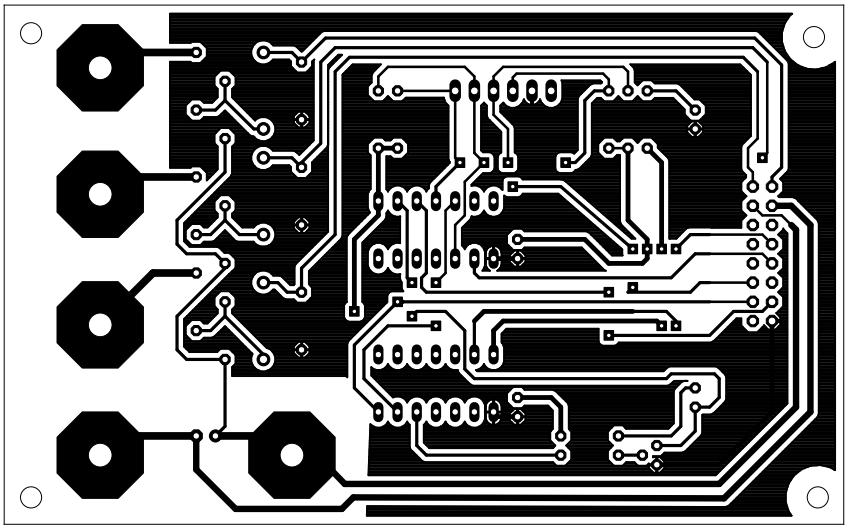


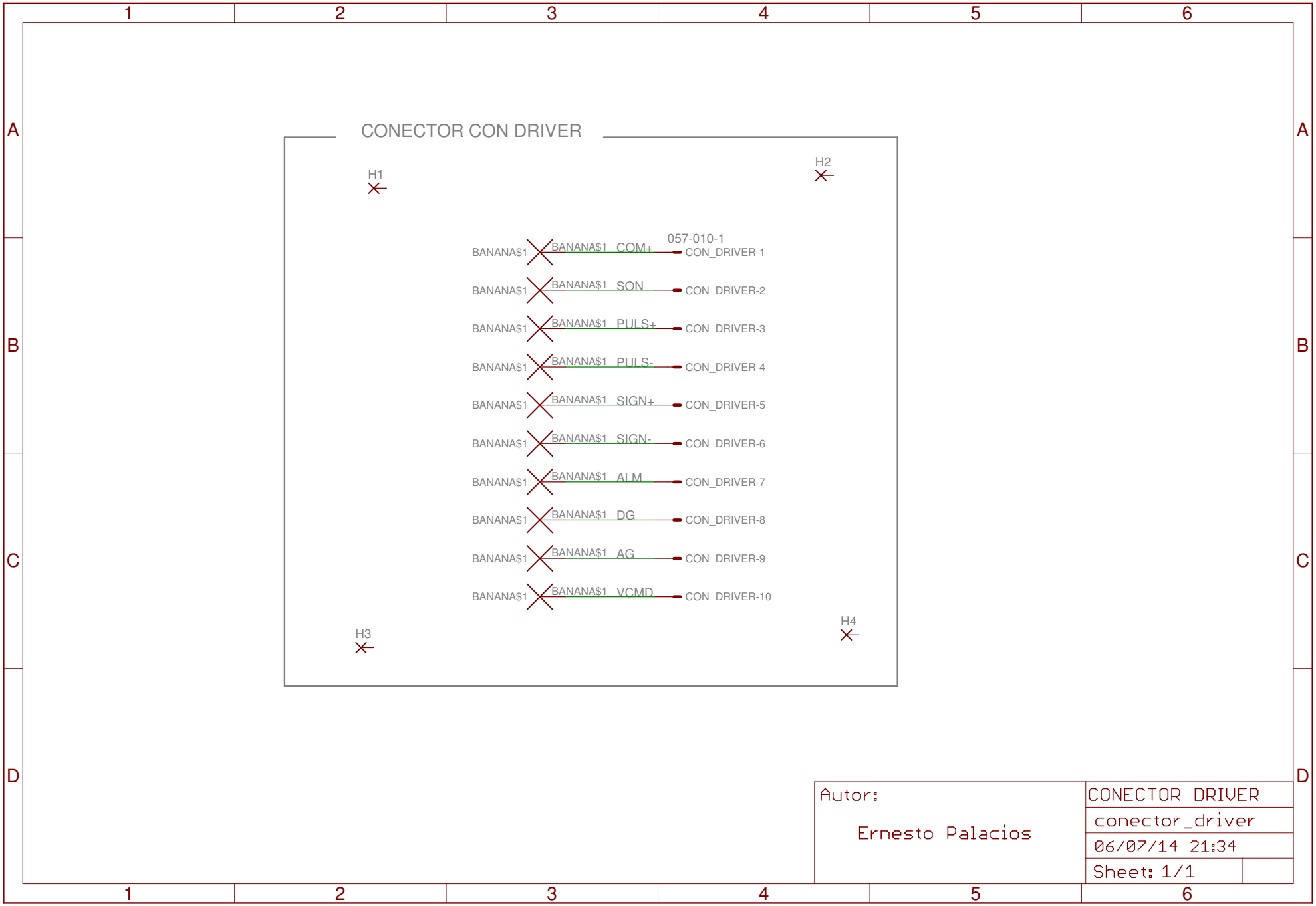
La salida de EMERGENCIA se activan con los opto-acopladores Lim_1 y Lim_4
La salida de ALARMA se activa con los optpacopladores Lim_2 y Lim_3

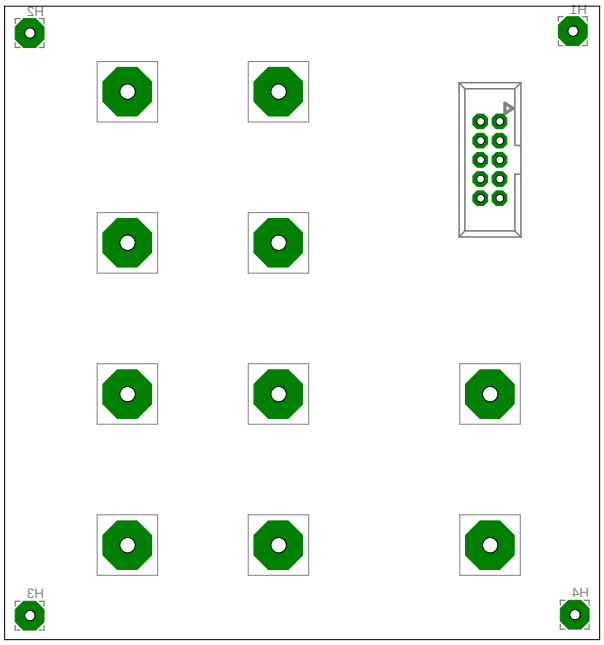
Autor: Ernesto Palacios	Placa Encoder
	Encoder_y_alarmas
	06/07/14 21:40
	Sheet: 2/2

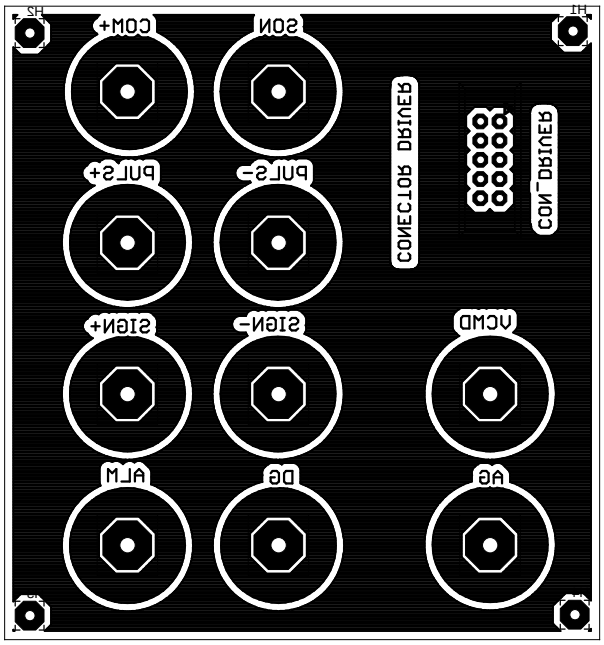

ALARMA
POSICIÓN
INVALIDA
74LS14N
74S00N
1N4004
1uF
100nf
330
4K7
10K
3K3
1K

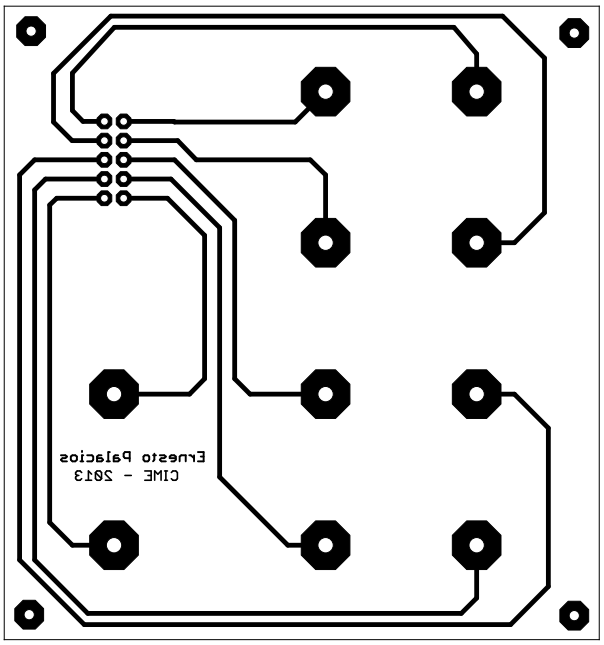

OPTOACOPLADORES
Mecatrónica
UTN
CON ENCODER
PLACA ENCODER
0V
+24V
A
B
Z




CONECTOR CON DRIVER
H1
H2
H3
H4
057-010-1
BANANA$1
BANANA$1 COM+
CON_DRIVER-1
BANANA$1 SON
CON_DRIVER-2
BANANA$1 PULS+
CON_DRIVER-3
BANANA$1 PULS-
CON_DRIVER-4
BANANA$1 SIGN+
CON_DRIVER-5
BANANA$1 SIGN-
CON_DRIVER-6
BANANA$1 ALM
CON_DRIVER-7
BANANA$1 DG
CON_DRIVER-8
BANANA$1 AG
CON_DRIVER-9
BANANA$1 VCMD
CON_DRIVER-10
Autor:
Ernesto Palacios
CONECTOR DRIVER
conector_driver
06/07/14 21:34
Sheet: 1/1


H5
H1
H3
H4


COM+
SON
PULS+
PULS-
SIGN+
SIGN-
VCMD
ALM
DG
AG
CONECTOR DRIVER
CON_DRIVER


Ernesto Palacios
CIME - 2013

ANEXO 6. HOJA DE DATOS MICROCONTROLADOR

26.1 Basic configuration

The QEI is configured using the following registers:

1. Power: In the PCONP register (Table 46), set bit PCQEI.

   Remark: On reset, the QEI is disabled (PCQEI = 0).
2. Peripheral clock: In the PCLKSEL0 register (Table 40), select PCLK_QEI.
3. Pins: Select QEI pins through the PINSEL registers. Select pin modes for port pins with QEI functions through the PINMODE registers (Section 8.5).
4. Interrupts: See Section 26.6.4. The QEI interrupt is enabled in the NVIC using the appropriate Interrupt Set Enable register.

26.2 Features

This Quadrature Encoder Interface (QEI) has the following features:

- tracks encoder position.
- increments/ decrements depending on direction.
- programmable for 2X or 4X position counting.
- velocity capture using built-in timer.
- velocity compare function with less than interrupt.
- uses 32-bit registers for position and velocity.
- three position compare registers with interrupts.
- index counter for revolution counting.
- index compare register with interrupts.
- can combine index and position interrupts to produce an interrupt for whole and partial revolution displacement.
- digital filter with programmable delays for encoder input signals.
- can accept decoded signal inputs (clock and direction).

26.3 Introduction

A quadrature encoder, also known as a 2-channel incremental encoder, converts angular displacement into two pulse signals. By monitoring both the number of pulses and the relative phase of the two signals, you can track the position, direction of rotation, and velocity. In addition, a third channel, or index signal, can be used to reset the position counter. This quadrature encoder interface module decodes the digital pulses from a quadrature encoder wheel to integrate position over time and determine direction of rotation. In addition, it can capture the velocity of the encoder wheel.

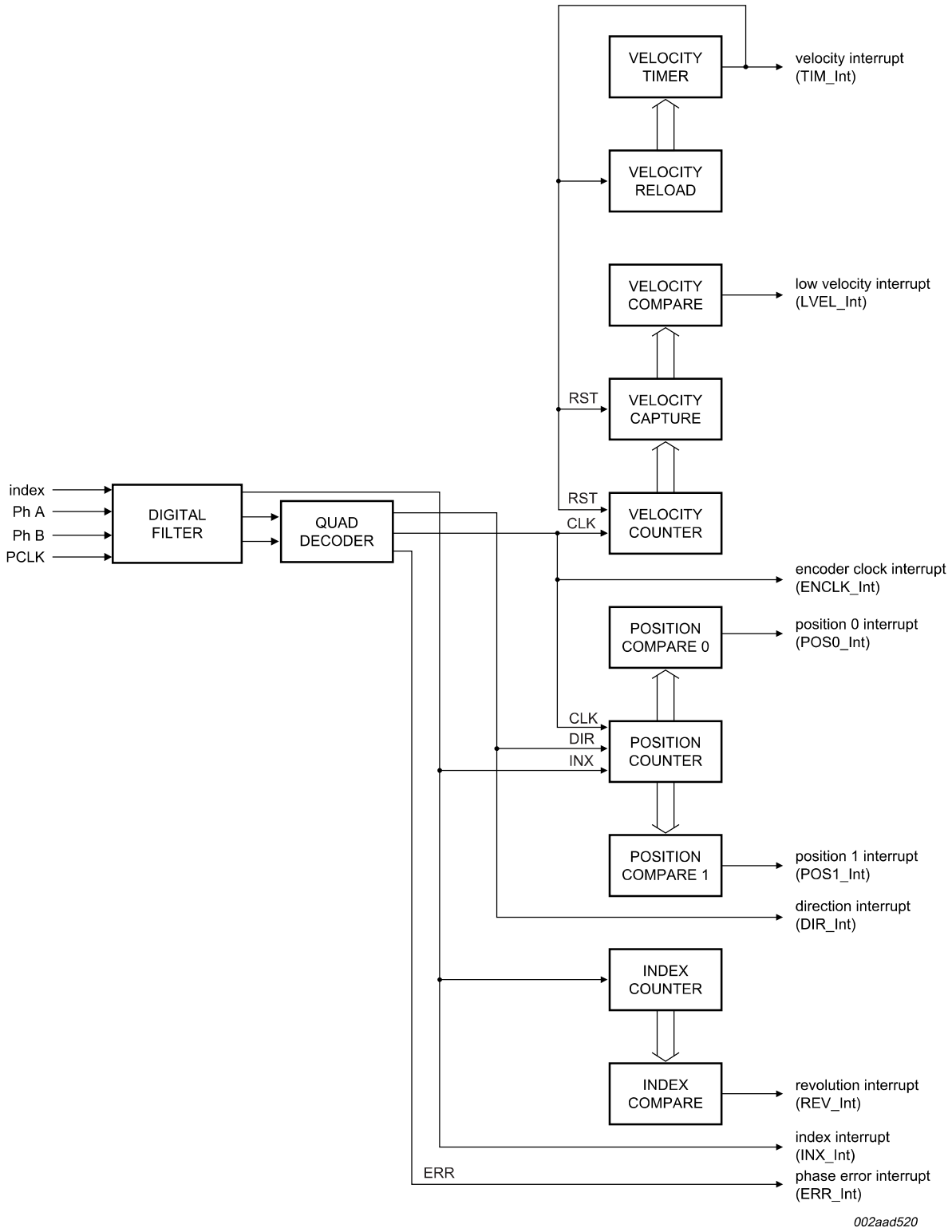


Fig 129.Encoder interface block diagram

26.4 Functional description

The QEI module interprets the two-bit gray code produced by a quadrature encoder wheel to integrate position over time and determine direction of rotation. In addition, it can capture the velocity of the encoder wheel.

26.4.1 Input signals

The QEI module supports two modes of signal operation: quadrature phase mode and clock/direction mode. In quadrature phase mode, the encoder produces two clocks that are 90 degrees out of phase; the edge relationship is used to determine the direction of rotation. In clock/direction mode, the encoder produces a clock signal to indicate steps and a direction signal to indicate the direction of rotation.).

This mode is determined by the SigMode bit of the QEI Configuration (QEICONF) register (See Table 486). When the SigMode bit = 1, the quadrature decoder is bypassed and the PhA pin functions as the direction signal and PhB pin functions as the clock signal for the counters, etc. When the SigMode bit = 0, the PhA pin and PhB pins are decoded by the quadrature decoder. In this mode the quadrature decoder produces the direction and clock signals for the counters, etc. In both modes the direction signal is subject to the effects of the direction invert (DIRINV) bit.

26.4.1.1 Quadrature input signals

When edges on PhA lead edges on PhB, the position counter is incremented. When edges on PhB lead edges on PhA, the position counter is decremented. When a rising and falling edge pair is seen on one of the phases without any edges on the other, the direction of rotation has changed.

Table 480. Encoder states

Phase A	Phase B	state
1	0	1
1	1	2
0	1	3
0	0	4

Table 481. Encoder state transitions[1]

from state	to state	Direction
1	2	positive
2	3	
3	4	
4	1	
4	3	negative
3	2	
2	1	
1	4	

[1] All other state transitions are illegal and should set the ERR bit.

Interchanging of the PhA and PhB input signals are compensated by complementing the DIR bit. When set = 1, the direction inversion bit (DIRINV) complements the DIR bit.

Table 482. Encoder direction

DIR bit	DIRINV bit	direction
0	0	forward
1	0	reverse
0	1	reverse
1	1	forward

Figure 130 shows how quadrature encoder signals equate to direction and count.

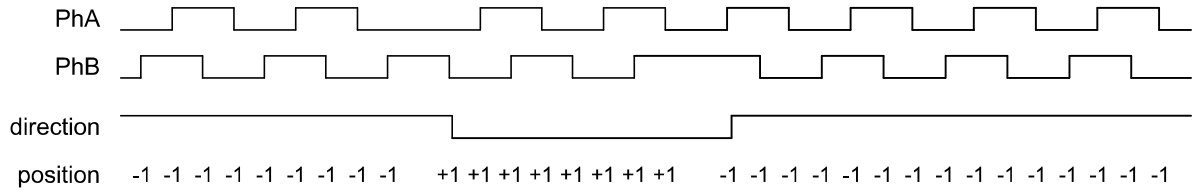


Fig 130.Quadrature Encoder Basic Operation

26.4.1.2 Digital input filtering

All three encoder inputs (PhA, PhB, and index) require digital filtering. The number of sample clocks is user programmable from 1 to 4,294,967,295 (0xFFFF FFFF). In order for a transition to be accepted, the input signal must remain in new state for the programmed number of sample clocks.

26.4.2 Position capture

The capture mode for the position integrator can be set to update the position counter on every edge of the PhA signal or to update on every edge of both PhA and PhB. Updating the position counter on every PhA and PhB provides more positional resolution at the cost of less range in the positional counter.

The position integrator and velocity capture can be independently enabled. Alternatively, the phase signals can be interpreted as a clock and direction signal as output by some encoders.

The position counter is automatically reset on one of two conditions. Incrementing past the maximum position value (QEIMAXPOS) will reset the position counter to zero. If the reset on index bit (RESPI) is set, sensing the index pulse will reset the position counter to zero.

26.4.3 Velocity capture

The velocity capture has a programmable timer and a capture register. It counts the number of phase edges (using the same configuration as for the position integrator) in a given time period. When the velocity timer (QEITIME) overflows the contents of the velocity counter (QEIVEL) are transferred to the capture (QEICAP) register. The velocity counter is then cleared. The velocity timer is loaded with the contents of the velocity reload register (QEILOAD). Finally, the velocity interrupt (TIM_Int) is asserted. The

number of edges counted in a given time period is directly proportional to the velocity of the encoder. Setting the reset velocity bit (RESV) has the same effect as an overflow of the velocity timer, except that the setting the RESV bit will not generate a velocity interrupt.

The following equation converts the velocity counter value into an RPM value:

```
RPM = (PCLK * QEICAP * 60) ÷ (QEILOAD * PPR * Edges)
```

where:

- **PCLK** is the peripheral clock rate for the QEI block. See Section 4.7.3 for more on the possibilities for PCLK).
- **QEICAP** is the captured velocity counter value for the last velocity timer period.
- **QEILOAD** is the velocity timer reload value.
- **PPR** is the number of pulses per revolution of the physical encoder used in the application
- **Edges** is 2 or 4, based on the capture mode set in the QEICON register (2 for CapMode set to 0 and 4 for CapMode set to 1)

For example, consider a motor running at 600 RPM. A 2048 pulse per revolution quadrature encoder is attached to the motor, producing 8192 phase edges per revolution (PPR * Edges). This results in 81,920 pulses per second (the motor turns 10 times per second at 600 RPM and there are 8,192 edges per revolution). If the timer were clocked at 10,000 Hz, and the QEILOAD was 2,500 (corresponding to ¼ of a second), it would count 20,480 pulses per update. Using the above equation:

RPM = (10000 * 1 * 20480 * 60) ÷ (2500 * 2048 * 4) = 600 RPM

Now, consider that the motor is sped up to 3000 RPM. This results in 409,600 pulses per second, or 102,400 every ¼ of a second. Again, the above equation gives:

RPM = (10000 * 1 * 102400 * 60) ÷ (2500 * 2048 * 4) = 3000 RPM

These are simple examples, real-world values will have a higher rate for PCLK, and probably a larger value for QEILOAD as well.

26.4.4 Velocity compare

In addition to velocity capture, the velocity measurement system includes a programmable velocity compare register. After every velocity capture event the contents of the velocity capture register (QEICAP) is compared with the contents of the velocity compare register (VELCOMP). If the captured velocity is less than the compare value an interrupt is asserted provided that the velocity compare interrupt enable bit is set. This can be used to determine if a motor shaft is either stalled or moving too slow.

26.5 Pin description

Table 483. QEI pin description

Pin name	I/O	Description
MCI0 [1]	I	Used as the Phase A (PhA) input to the Quadrature Encoder Interface.
MCI1 [1]	I	Used as the Phase B (PhB) input to the Quadrature Encoder Interface.
MCI2 [1]	I	Used as the Index (IDX) input to the Quadrature Encoder Interface.

[1] The Quadrature Encoder Interface uses the same pin functions as the Motor Control PWM feedback inputs and are connected when the Motor Control PWM function is selected on these pins. If used as part of motor control, the QEI is an alternative to feedback directly to the MCPWM.

26.6 Register description

26.6.1 Register summary

Table 484. QEI Register summary

Name	Description	Access	Reset value	Address
Control registers				
QEICON	Control register	WO	0	0x400B C000
QEICONF	Configuration register	R/W	0	0x400B C008
QEISTAT	Encoder status register	RO	0	0x400B C004
Position, index, and timer registers				
QEIPOS	Position register	RO	0	0x400B C00C
QEIMAXPOS	Maximum position register	R/W	0	0x400B C010
CMPOS0	position compare register 0	R/W	0	0x400B C014
CMPOS1	position compare register 1	R/W	0	0x400B C018
CMPOS2	position compare register 2	R/W	0	0x400B C01C
INXCNT	Index count register	RO	0	0x400B C020
INXCMP	Index compare register	R/W	0	0x400B C024
QEILOAD	Velocity timer reload register	R/W	0	0x400B C028
QEITIME	Velocity timer register	RO	0	0x400B C02C
QEIVEL	Velocity counter register	RO	0	0x400B C030
QEICAP	Velocity capture register	RO	0	0x400B C034
VELCOMP	Velocity compare register	R/W	0	0x400B C038
FILTER	Digital filter register	R/W	0	0x400B C03C
Interrupt registers				
QEIINTSTAT	Interrupt status register	RO	0	0x400B CFE0
QEISET	Interrupt status set register	WO	0	0x400B CFEC
QEICLR	Interrupt status clear register	WO	0	0x400B CFE8
QEIIE	Interrupt enable register	RO	0	0x400B CFE4
QEIIES	Interrupt enable set register	WO	0	0x400B CFDC
QEIIEC	Interrupt enable clear register	WO	0	0x400B CFD8

21.1 Basic configuration

The Timer 0, 1, 2, and 3 peripherals are configured using the following registers:

1. Power: In the PCONP register (Table 46), set bits PCTIM0/1/2/3.

   Remark: On reset, Timer0/1 are enabled (PCTIM0/1 = 1), and Timer2/3 are disabled (PCTIM2/3 = 0).
2. Peripheral clock: In the PCLKSEL0 register (Table 40), select PCLK_TIMER0/1; in the PCLKSEL1 register (Table 41), select PCLK_TIMER2/3.
3. Pins: Select timer pins through the PINSEL registers. Select the pin modes for the port pins with timer functions through the PINMODE registers (Section 8.5).
4. Interrupts: See register T0/1/2/3MCR (Table 430) and T0/1/2/3CCR (Table 431) for match and capture events. Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register.
5. DMA: Up to two match conditions can be used to generate timed DMA requests, see Table 544.

21.2 Features

Remark: The four Timer/Counters are identical except for the peripheral base address. A minimum of two Capture inputs and two Match outputs are pinned out for all four timers, with a choice of multiple pins for each. Timer 2 brings out all four Match outputs.

- A 32-bit Timer/Counter with a programmable 32-bit Prescaler.
- Counter or Timer operation
- Up to two 32-bit capture channels per timer, that can take a snapshot of the timer value when an input signal transitions. A capture event may also optionally generate an interrupt.
- Four 32-bit match registers that allow:
  - Continuous operation with optional interrupt generation on match.
  - Stop timer on match with optional interrupt generation.
  - Reset timer on match with optional interrupt generation.
- Up to four external outputs corresponding to match registers, with the following capabilities:
  - Set low on match.
  - Set high on match.
  - Toggle on match.
  - Do nothing on match.

21.3 Applications

- Interval Timer for counting internal events.
- Pulse Width Demodulator via Capture inputs.
- Free running timer.

21.4 Description

The Timer/Counter is designed to count cycles of the peripheral clock (PCLK) or an externally-supplied clock, and can optionally generate interrupts or perform other actions at specified timer values, based on four match registers. It also includes four capture inputs to trap the timer value when an input signal transitions, optionally generating an interrupt.

21.5 Pin description

Table 425 gives a brief summary of each of the Timer/Counter related pins.

Table 425. Timer/Counter pin description

Pin	Type	Description
CAP0[1:0] CAP1[1:0] CAP2[1:0] CAP3[1:0]	Input	Capture Signals- A transition on a capture pin can be configured to load one of the Capture Registers with the value in the Timer Counter and optionally generate an interrupt. Capture functionality can be selected from a number of pins. When more than one pin is selected for a Capture input on a single TIMER0/1 channel, the pin with the lowest Port number is used Timer/Counter block can select a capture signal as a clock source instead of the PCLK derived clock. For more details see Section 21.6.3.
MAT0[1:0] MAT1[1:0] MAT2[3:0] MAT3[1:0]	Output	External Match Output - When a match register (MR3:0) equals the timer counter (TC) this output can either toggle, go low, go high, or do nothing. The External Match Register (EMR) controls the functionality of this output. Match Output functionality can be selected on a number of pins in parallel.

21.5.1 Multiple CAP and MAT pins

Software can select from multiple pins for the CAP or MAT functions in the Pin Select registers, which are described in Section 8.5. When more than one pin is selected for a MAT output, all such pins are driven identically. When more than one pin is selected for a CAP input, the pin with the lowest Port number is used. Note that match conditions may be used internally without the use of a device pin.

Effective processing of the externally supplied clock to the counter has some limitations. Since two successive rising edges of the PCLK clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input can not exceed one quarter of the PCLK clock. Consequently, duration of the high/low levels on the same CAP input in this case cannot be shorter than 1/(2 x PCLK).

21.6 Register description

Each Timer/Counter contains the registers shown in Table 426 ("Reset Value" refers to the data stored in used bits only; it does not include reserved bits content). More detailed descriptions follow.

Table 426. TIMER/COUNTER0-3 register map

Generic Name	Description	Access	Reset Value[1]	TIMERn Register/ Name & Address
IR	Interrupt Register. The IR can be written to clear interrupts. The IR can be read to identify which of eight possible interrupt sources are pending.	R/W	0	T0IR - 0x4000 4000 T1IR - 0x4000 8000 T2IR - 0x4009 0000 T3IR - 0x4009 4000
TCR	Timer Control Register. The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.	R/W	0	T0TCR - 0x4000 4004 T1TCR - 0x4000 8004 T2TCR - 0x4009 0004 T3TCR - 0x4009 4004
TC	Timer Counter. The 32-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.	R/W	0	T0TC - 0x4000 4008 T1TC - 0x4000 8008 T2TC - 0x4009 0008 T3TC - 0x4009 4008
PR	Prescale Register. When the Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC.	R/W	0	T0PR - 0x4000 400C T1PR - 0x4000 800C T2PR - 0x4009 000C T3PR - 0x4009 400C
PC	Prescale Counter. The 32-bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented and the PC is cleared. The PC is observable and controllable through the bus interface.	R/W	0	T0PC - 0x4000 4010 T1PC - 0x4000 8010 T2PC - 0x4009 0010 T3PC - 0x4009 4010
MCR	Match Control Register. The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.	R/W	0	T0MCR - 0x4000 4014 T1MCR - 0x4000 8014 T2MCR - 0x4009 0014 T3MCR - 0x4009 4014
MR0	Match Register 0. MR0 can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC.	R/W	0	T0MR0 - 0x4000 4018 T1MR0 - 0x4000 8018 T2MR0 - 0x4009 0018 T3MR0 - 0x4009 4018
MR1	Match Register 1. See MR0 description.	R/W	0	T0MR1 - 0x4000 401C T1MR1 - 0x4000 801C T2MR1 - 0x4009 001C T3MR1 - 0x4009 401C
MR2	Match Register 2. See MR0 description.	R/W	0	T0MR2 - 0x4000 4020 T1MR2 - 0x4000 8020 T2MR2 - 0x4009 0020 T3MR2 - 0x4009 4020
MR3	Match Register 3. See MR0 description.	R/W	0	T0MR3 - 0x4000 4024 T1MR3 - 0x4000 8024 T2MR3 - 0x4009 0024 T3MR3 - 0x4009 4024
CCR	Capture Control Register. The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place.	R/W	0	T0CCR - 0x4000 4028 T1CCR - 0x4000 8028 T2CCR - 0x4009 0028 T3CCR - 0x4009 4028

Table 426. TIMER/COUNTER0-3 register map *…continued*

Generic Name	Description	Access	Reset Value[1]	TIMERn Register/ Name & Address
CR0	Capture Register 0. CR0 is loaded with the value of TC when there is an event on the CAPn.0(CAP0.0 or CAP1.0 respectively) input.	RO	0	T0CR0 - 0x4000 402C T1CR0 - 0x4000 802C T2CR0 - 0x4009 002C T3CR0 - 0x4009 402C
CR1	Capture Register 1. See CR0 description.	RO	0	T0CR1 - 0x4000 4030 T1CR1 - 0x4000 8030 T2CR1 - 0x4009 0030 T3CR1 - 0x4009 4030
EMR	External Match Register. The EMR controls the external match pins MATn.0-3 (MAT0.0-3 and MAT1.0-3 respectively).	R/W	0	T0EMR - 0x4000 403C T1EMR - 0x4000 803C T2EMR - 0x4009 003C T3EMR - 0x4009 403C
CTCR	Count Control Register. The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting.	R/W	0	T0CTCR - 0x4000 4070 T1CTCR - 0x4000 8070 T2CTCR - 0x4009 0070 T3CTCR - 0x4009 4070

[1] Reset Value reflects the data stored in used bits only. It does not include reserved bits content.

21.6.1 Interrupt Register (T[0/1/2/3]IR - 0x4000 4000, 0x4000 8000, 0x4009 0000, 0x4009 4000)

The Interrupt Register consists of 4 bits for the match interrupts and 4 bits for the capture interrupts. If an interrupt is generated then the corresponding bit in the IR will be high. Otherwise, the bit will be low. Writing a logic one to the corresponding IR bit will reset the interrupt. Writing a zero has no effect. The act of clearing an interrupt for a timer match also clears any corresponding DMA request.

Table 427. Interrupt Register (T[0/1/2/3]IR - addresses 0x4000 4000, 0x4000 8000, 0x4009 0000, 0x4009 4000) bit description

Bit	Symbol	Description	Reset Value
0	MR0 Interrupt	Interrupt flag for match channel 0.	0
1	MR1 Interrupt	Interrupt flag for match channel 1.	0
2	MR2 Interrupt	Interrupt flag for match channel 2.	0
3	MR3 Interrupt	Interrupt flag for match channel 3.	0
4	CR0 Interrupt	Interrupt flag for capture channel 0 event.	0
5	CR1 Interrupt	Interrupt flag for capture channel 1 event.	0
31:6	-	Reserved	-

21.6.2 Timer Control Register (T[0/1/2/3]CR - 0x4000 4004, 0x4000 8004, 0x4009 0004, 0x4009 4004)

The Timer Control Register (TCR) is used to control the operation of the Timer/Counter.

Table 428. Timer Control Register (TCR, TIMERn: TnTCR - addresses 0x4000 4004, 0x4000 8004, 0x4009 0004, 0x4009 4004) bit description

Bit	Symbol	Description	Reset Value
0	Counter Enable	When one, the Timer Counter and Prescale Counter are enabled for counting. When zero, the counters are disabled.	0
1	Counter Reset	When one, the Timer Counter and the Prescale Counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until TCR[1] is returned to zero.	0
31:2	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

21.6.3 Count Control Register (T[0/1/2/3]CTCR - 0x4000 4070, 0x4000 8070, 0x4009 0070, 0x4009 4070)

The Count Control Register (CTCR) is used to select between Timer and Counter mode, and in Counter mode to select the pin and edge(s) for counting.

When Counter Mode is chosen as a mode of operation, the CAP input (selected by the CTCR bits 3:2) is sampled on every rising edge of the PCLK clock. After comparing two consecutive samples of this CAP input, one of the following four events is recognized: rising edge, falling edge, either of edges or no changes in the level of the selected CAP input. Only if the identified event occurs and the event corresponds to the one selected by bits 1:0 in the CTCR register, will the Timer Counter register be incremented.

Effective processing of the externally supplied clock to the counter has some limitations. Since two successive rising edges of the PCLK clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input can not exceed one quarter of the PCLK clock. Consequently, duration of the high/low levels on the same CAP input in this case can not be shorter than 1/(2 PCLK).

Table 429. Count Control Register (T[0/1/2/3]CTCR - addresses 0x4000 4070, 0x4000 8070, 0x4009 0070, 0x4009 4070) bit description

Bit	Symbol	Value	Description	Reset Value
1:0	Counter/ Timer Mode		This field selects which rising PCLK edges can increment the Timer's Prescale Counter (PC), or clear the PC and increment the Timer Counter (TC).	00
		00	Timer Mode: the TC is incremented when the Prescale Counter matches the Prescale Register. The Prescale Counter is incremented on every rising PCLK edge.	
		01	Counter Mode: TC is incremented on rising edges on the CAP input selected by bits 3:2.	
		10	Counter Mode: TC is incremented on falling edges on the CAP input selected by bits 3:2.	
		11	Counter Mode: TC is incremented on both edges on the CAP input selected by bits 3:2.	

Table 429. Count Control Register (T[0/1/2/3]CTCR - addresses 0x4000 4070, 0x4000 8070, 0x4009 0070, 0x4009 4070) bit description *…continued*

Bit	Symbol	Value	Description	Reset Value
3:2	Count Input Select		When bits 1:0 in this register are not 00, these bits select which CAP pin is sampled for clocking.	00
		00	CAPn.0 for TIMERn	
		01	CAPn.1 for TIMERn	
		10	Reserved	
		11	Reserved	
			Note: If Counter mode is selected for a particular CAPn input in the TnCTCR, the 3 bits for that input in the Capture Control Register (TnCCR) must be programmed as 000. However, capture and/or interrupt can be selected for the other 3 CAPn inputs in the same timer.	
31:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

21.6.4 Timer Counter registers (T0TC - T3TC, 0x4000 4008, 0x4000 8008, 0x4009 0008, 0x4009 4008)

The 32-bit Timer Counter register is incremented when the prescale counter reaches its terminal count. Unless it is reset before reaching its upper limit, the Timer Counter will count up through the value 0xFFFF FFFF and then wrap back to the value 0x0000 0000. This event does not cause an interrupt, but a match register can be used to detect an overflow if needed.

21.6.5 Prescale register (T0PR - T3PR, 0x4000 400C, 0x4000 800C, 0x4009 000C, 0x4009 400C)

The 32-bit Prescale register specifies the maximum value for the Prescale Counter.

21.6.6 Prescale Counter register (T0PC - T3PC, 0x4000 4010, 0x4000 8010, 0x4009 0010, 0x4009 4010)

The 32-bit Prescale Counter controls division of PCLK by some constant value before it is applied to the Timer Counter. This allows control of the relationship of the resolution of the timer versus the maximum time before the timer overflows. The Prescale Counter is incremented on every PCLK. When it reaches the value stored in the Prescale register, the Timer Counter is incremented and the Prescale Counter is reset on the next PCLK. This causes the Timer Counter to increment on every PCLK when PR = 0, every 2 pclks when PR = 1, etc.

21.6.7 Match Registers (MR0 - MR3)

The Match register values are continuously compared to the Timer Counter value. When the two values are equal, actions can be triggered automatically. The action possibilities are to generate an interrupt, reset the Timer Counter, or stop the timer. Actions are controlled by the settings in the MCR register.

21.6.8 Match Control Register (T[0/1/2/3]MCR - 0x4000 4014, 0x4000 8014, 0x4009 0014, 0x4009 4014)

The Match Control Register is used to control what operations are performed when one of the Match Registers matches the Timer Counter. The function of each of the bits is shown in Table 430.

Table 430. Match Control Register (T[0/1/2/3]MCR - addresses 0x4000 4014, 0x4000 8014, 0x4009 0014, 0x4009 4014) bit description

Bit	Symbol	Value	Description	Reset Value
0	MR0I	1	Interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC.	0
		0	This interrupt is disabled	
1	MR0R	1	Reset on MR0: the TC will be reset if MR0 matches it.	0
		0	Feature disabled.	
2	MR0S	1	Stop on MR0: the TC and PC will be stopped and TCR[0] will be set to 0 if MR0 matches the TC.	0
		0	Feature disabled.	
3	MR1I	1	Interrupt on MR1: an interrupt is generated when MR1 matches the value in the TC.	0
		0	This interrupt is disabled	
4	MR1R	1	Reset on MR1: the TC will be reset if MR1 matches it.	0
		0	Feature disabled.	
5	MR1S	1	Stop on MR1: the TC and PC will be stopped and TCR[0] will be set to 0 if MR1 matches the TC.	0
		0	Feature disabled.	
6	MR2I	1	Interrupt on MR2: an interrupt is generated when MR2 matches the value in the TC.	0
		0	This interrupt is disabled	
7	MR2R	1	Reset on MR2: the TC will be reset if MR2 matches it.	0
		0	Feature disabled.	
8	MR2S	1	Stop on MR2: the TC and PC will be stopped and TCR[0] will be set to 0 if MR2 matches the TC.	0
		0	Feature disabled.	
9	MR3I	1	Interrupt on MR3: an interrupt is generated when MR3 matches the value in the TC.	0
		0	This interrupt is disabled	
10	MR3R	1	Reset on MR3: the TC will be reset if MR3 matches it.	0
		0	Feature disabled.	
11	MR3S	1	Stop on MR3: the TC and PC will be stopped and TCR[0] will be set to 0 if MR3 matches the TC.	0
		0	Feature disabled.	
31:12	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

21.6.9 Capture Registers (CR0 - CR1)

Each Capture register is associated with a device pin and may be loaded with the Timer Counter value when a specified event occurs on that pin. The settings in the Capture Control Register register determine whether the capture function is enabled, and whether a capture event happens on the rising edge of the associated pin, the falling edge, or on both edges.

21.6.10 Capture Control Register (T[0/1/2/3]CCR - 0x4000 4028, 0x4000 8028, 0x4009 0028, 0x4009 4028)

The Capture Control Register is used to control whether one of the four Capture Registers is loaded with the value in the Timer Counter when the capture event occurs, and whether an interrupt is generated by the capture event. Setting both the rising and falling bits at the same time is a valid configuration, resulting in a capture event for both edges. In the description below, "n" represents the Timer number, 0 or 1.

Note: If Counter mode is selected for a particular CAP input in the CTCR, the 3 bits for that input in this register should be programmed as 000, but capture and/or interrupt can be selected for the other 3 CAP inputs.

Table 431. Capture Control Register (T[0/1/2/3]CCR - addresses 0x4000 4028, 0x4000 8020, 0x4009 0028, 0x4009 4028) bit description

Bit	Symbol	Value	Description	Reset Value
0	CAP0RE	1	Capture on CAPn.0 rising edge: a sequence of 0 then 1 on CAPn.0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
1	CAP0FE	1	Capture on CAPn.0 falling edge: a sequence of 1 then 0 on CAPn.0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
2	CAP0I	1	Interrupt on CAPn.0 event: a CR0 load due to a CAPn.0 event will generate an interrupt.	0
		0	This feature is disabled.	
3	CAP1RE	1	Capture on CAPn.1 rising edge: a sequence of 0 then 1 on CAPn.1 will cause CR1 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
4	CAP1FE	1	Capture on CAPn.1 falling edge: a sequence of 1 then 0 on CAPn.1 will cause CR1 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
5	CAP1I	1	Interrupt on CAPn.1 event: a CR1 load due to a CAPn.1 event will generate an interrupt.	0
		0	This feature is disabled.	
31:6	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

21.6.11 External Match Register (T[0/1/2/3]EMR - 0x4000 403C, 0x4000 803C, 0x4009 003C, 0x4009 403C)

The External Match Register provides both control and status of the external match pins. In the descriptions below, “n” represents the Timer number, 0 or 1, and “m” represent a Match number, 0 through 3.

Match events for Match 0 and Match 1 in each timer can cause a DMA request, see Section 21.6.12.

Table 432. External Match Register (T[0/1/2/3]EMR - addresses 0x4000 403C, 0x4000 803C, 0x4009 003C, 0x4009 403C) bit description

Bit	Symbol	Description	Reset Value
0	EM0	External Match 0. When a match occurs between the TC and MR0, this bit can either toggle, go low, go high, or do nothing, depending on bits 5:4 of this register. This bit can be driven onto a MATn.0 pin, in a positive-logic manner (0 = low, 1 = high).	0
1	EM1	External Match 1. When a match occurs between the TC and MR1, this bit can either toggle, go low, go high, or do nothing, depending on bits 7:6 of this register. This bit can be driven onto a MATn.1 pin, in a positive-logic manner (0 = low, 1 = high).	0
2	EM2	External Match 2. When a match occurs between the TC and MR2, this bit can either toggle, go low, go high, or do nothing, depending on bits 9:8 of this register. This bit can be driven onto a MATn.2 pin, in a positive-logic manner (0 = low, 1 = high).	0
3	EM3	External Match 3. When a match occurs between the TC and MR3, this bit can either toggle, go low, go high, or do nothing, depending on bits 11:10 of this register. This bit can be driven onto a MATn.3 pin, in a positive-logic manner (0 = low, 1 = high).	0
5:4	EMC0	External Match Control 0. Determines the functionality of External Match 0. Table 433 shows the encoding of these bits.	00
7:6	EMC1	External Match Control 1. Determines the functionality of External Match 1. Table 433 shows the encoding of these bits.	00
9:8	EMC2	External Match Control 2. Determines the functionality of External Match 2. Table 433 shows the encoding of these bits.	00
11:10	EMC3	External Match Control 3. Determines the functionality of External Match 3. Table 433 shows the encoding of these bits.	00
15:12	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 433. External Match Control

EMR[11:10], EMR[9:8], EMR[7:6], or EMR[5:4]	Function
00	Do Nothing.
01	Clear the corresponding External Match bit/output to 0 (MATn.m pin is LOW if pinned out).
10	Set the corresponding External Match bit/output to 1 (MATn.m pin is HIGH if pinned out).
11	Toggle the corresponding External Match bit/output.

21.6.12 DMA operation

DMA requests are generated by a match of the Timer Counter (TC) register value to either Match Register 0 (MR0) or Match Register 1 (MR1). This is not connected to the operation of the Match outputs controlled by the EMR register. Each match sets a DMA request flag, which is connected to the DMA controller. In order to have an effect, the GPDMA must be configured and the relevant timer DMA request selected as a DMA source via the DMAREQSEL register, see Section 31.5.15.

When a timer is initially set up to generate a DMA request, the request may already be asserted before a match condition occurs. An initial DMA request may be avoided by having software write a one to the interrupt flag location, as if clearing a timer interrupt. See Section 21.6.1. A DMA request will be cleared automatically when it is acted upon by

the GPDMA controller.

Remark: Because timer DMA requests are generated whenever the timer value is equal to the related Match Register value, DMA requests are always generated when the timer is running, unless the Match Register value is higher than the upper count limit of the timer. It is important not to select and enable timer DMA requests in the GPDMA block unless the timer is correctly configured to generate valid DMA requests.

21.7 Example timer operation

Figure 115 shows a timer configured to reset the count and generate an interrupt on match. The prescaler is set to 2 and the match register set to 6. At the end of the timer cycle where the match occurs, the timer count is reset. This gives a full length cycle to the match value. The interrupt indicating that a match occurred is generated in the next clock after the timer reached the match value.

Figure 116 shows a timer configured to stop and generate an interrupt on match. The prescaler is again set to 2 and the match register set to 6. In the next clock after the timer reaches the match value, the timer enable bit in TCR is cleared, and the interrupt indicating that a match occurred is generated.

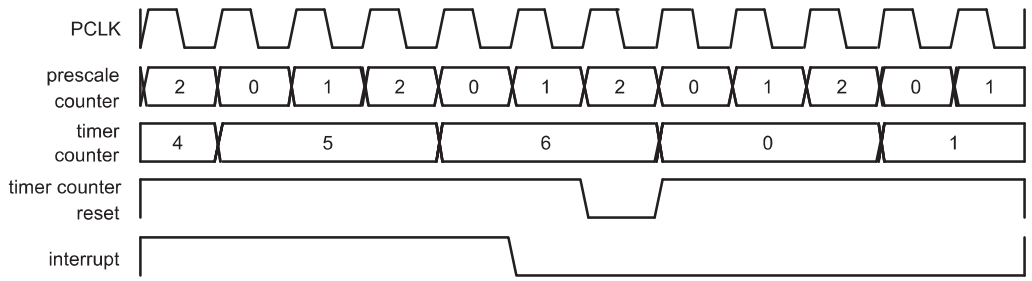


Fig 115. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled.

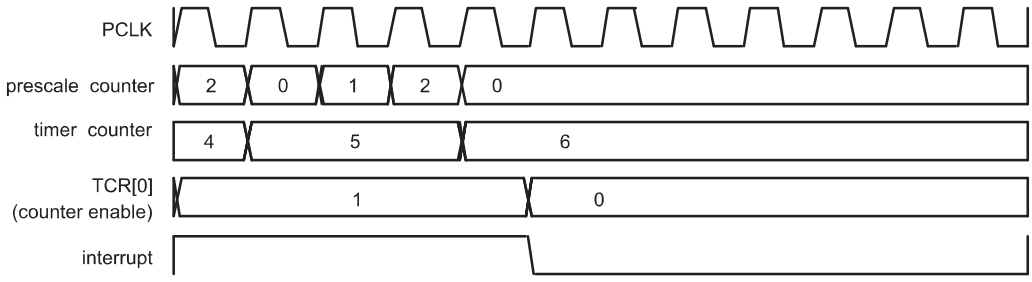


Fig 116. A timer Cycle in Which PR=2, MRx=6, and both interrupt and stop on match are enabled

21.8 Architecture

The block diagram for TIMER/COUNTER0 and TIMER/COUNTER1 is shown in Figure 117.

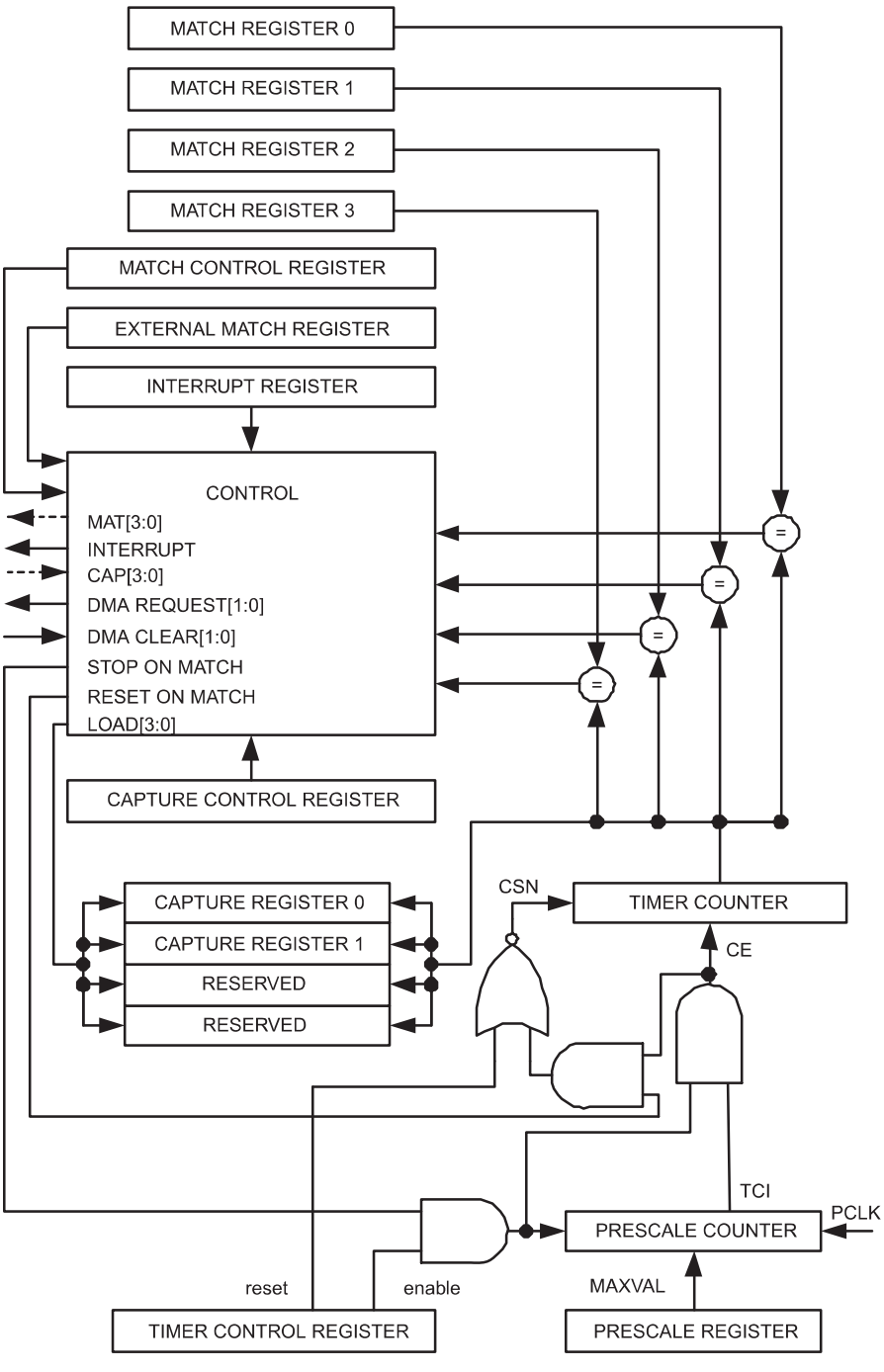


Fig 117. Timer block diagram

ANEXO 7. HOJA DE DATOS DEL SERVOMOTOR

DA98D Digital AC Servo Drive Unit

User Manual

(V5.00)

广州数控设备有限公司

GSK CNC EQUIPMENT CO., LTD.

In this User manual, we will exert ourselves to describe each item related to operation of this drive unit. But due to reasons like limit in space and specific product uses, detailed description of unnecessary or impossible operation of this drive unit will be not included. Therefore, items that are not specially indicated in this manual will all be regarded as “impossible” or “disallowed” operations.

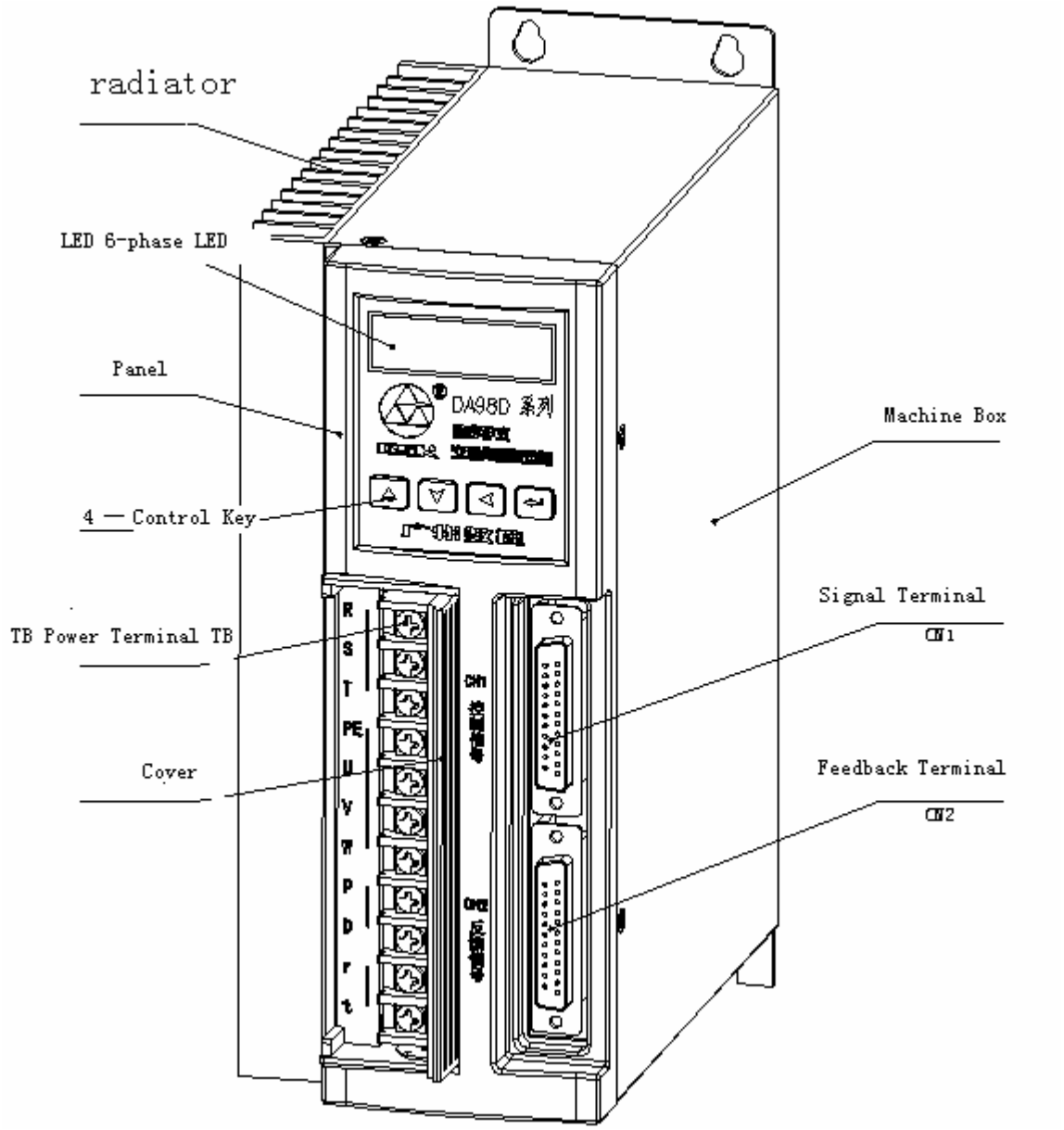


Fig. 1-1 Appearance of Servo Drive unit

2) Servo motor appearance

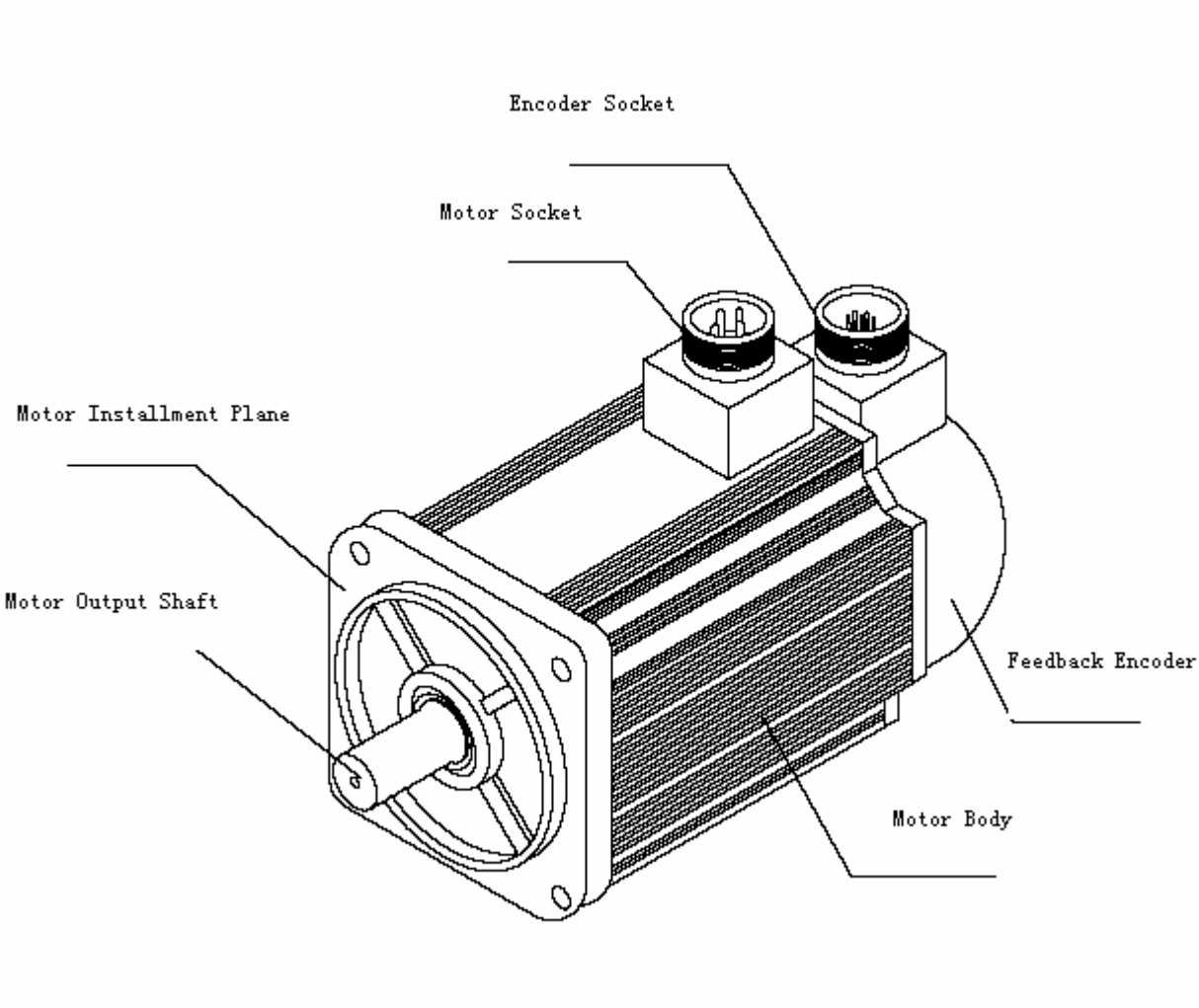


Fig. 1-2 Servo Motor Appearance

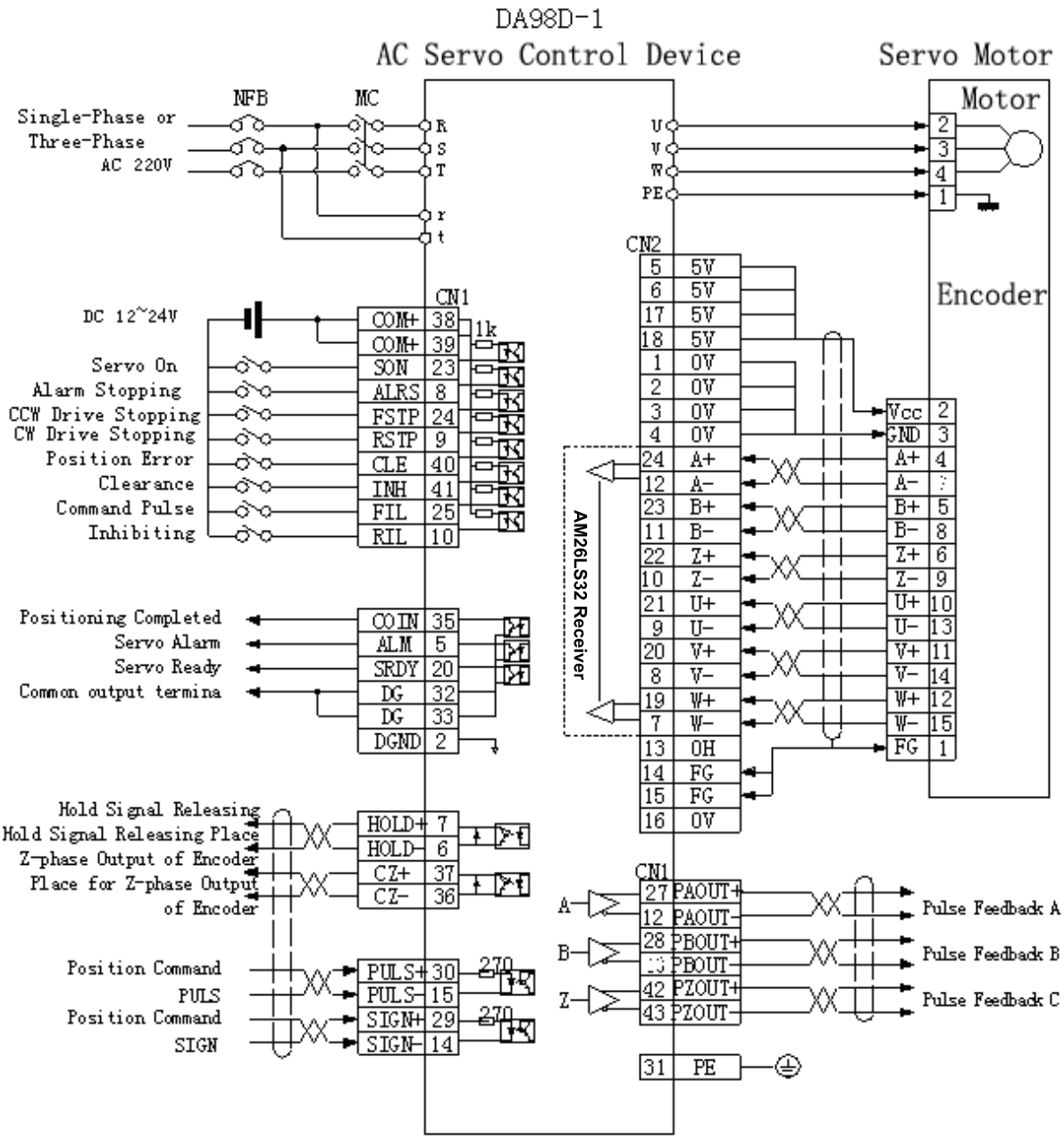


Fig. 3.1 Standard Wiring for Position Control Mode

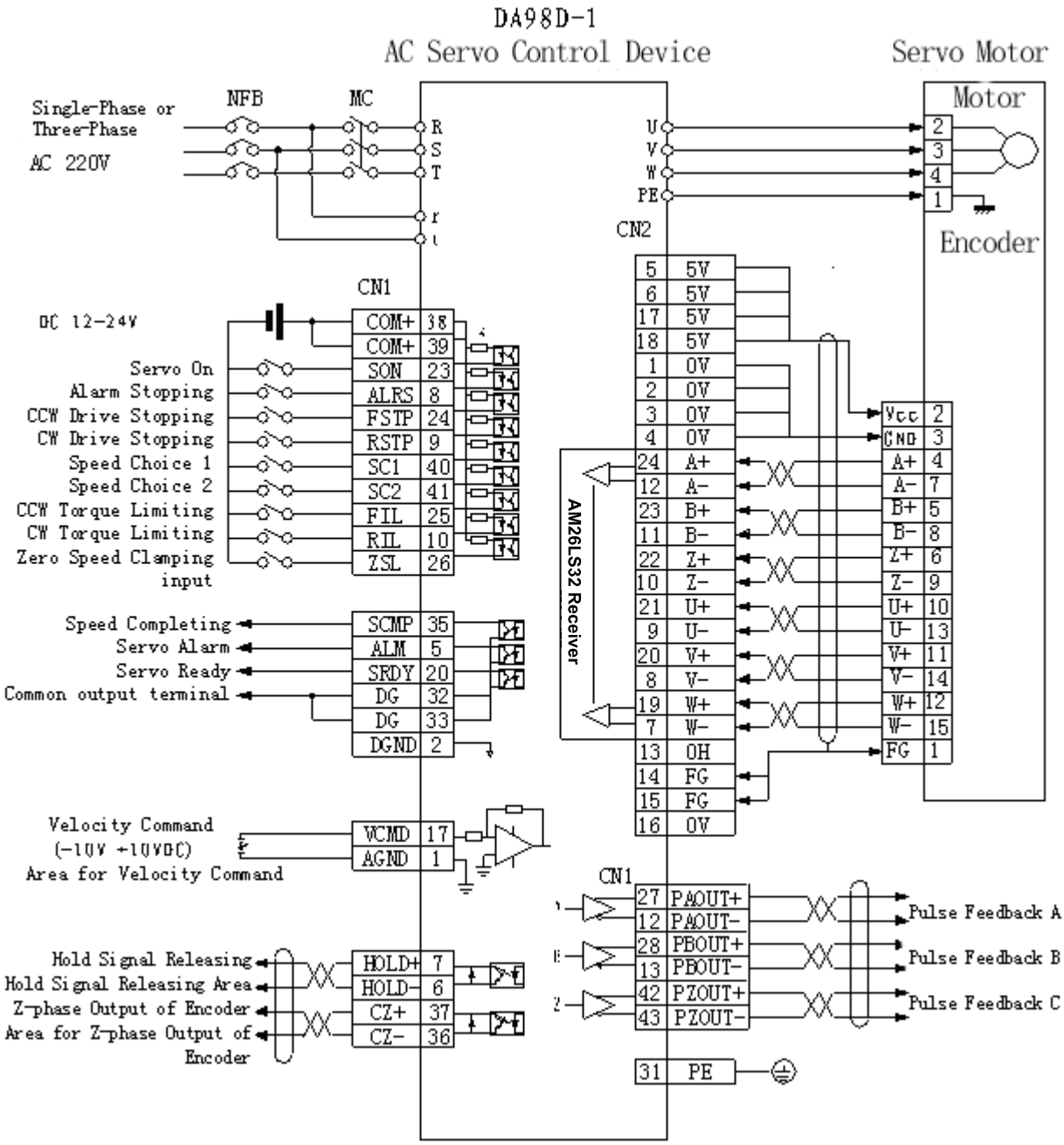


Fig. 3.2 Standard Wiring for Speed Control Mode

1) Switch Value Input Interface

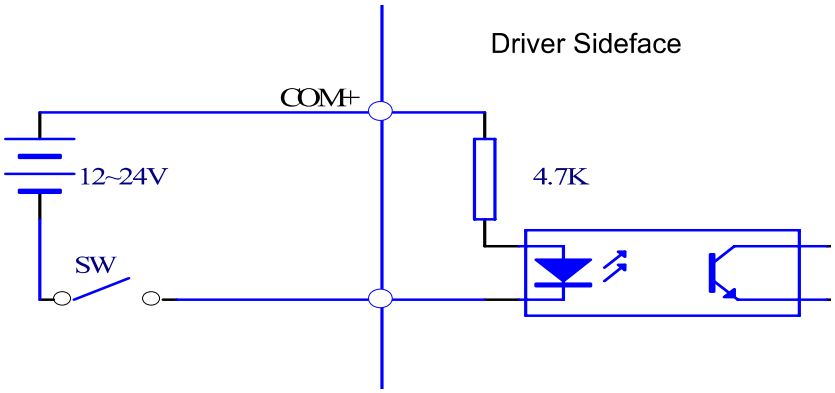


Fig. 3.4 Type1 Switch Value Input Interface

(1) Power supply is provided by the user, DC12～24V, current≥100mA;

(2) Note: if the electrodes are reversely connected, the servo driver will not work.

2) Switch Value Output Interface

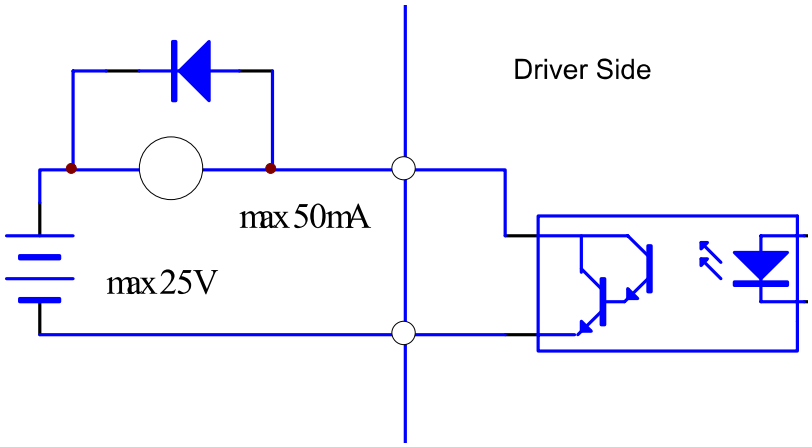


Fig. 3.5 Type2 Switch Value Ouput Interface

(1) The external power supply is provided by the user, but attention must be given to the case that if electrodes of the power supply are reversely connected, the servo drive unit may be damaged.

(2) The output is an open-circuit form of collector, with a maximal current of 50mA and a maximal external power voltage of 25V. Therefore, the load of switch value output signal must satisfy this limited requirement. If the limited requirement is surpassed or the output terminal is directly connected with the power supply, the servo drive unit will be damaged;

(3) If the loads are inductive ones like relay, two sides of the load must be reversely connected in parallel with the continuous current diode. If the continuous current diode is reversely connected, the servo drive unit will be damaged.

3) Analog Input Interface

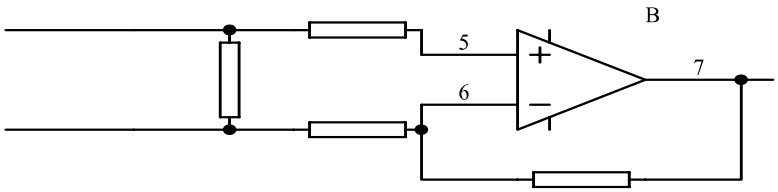


Fig. 3.6 Type4 Analog Command Input Interface

1. Input signal is connected with twisted-pair cable lines.
2. The circuit adopts the enlarged different-mode form, with an input resistance of 20K.

4) Pulse Output Interface

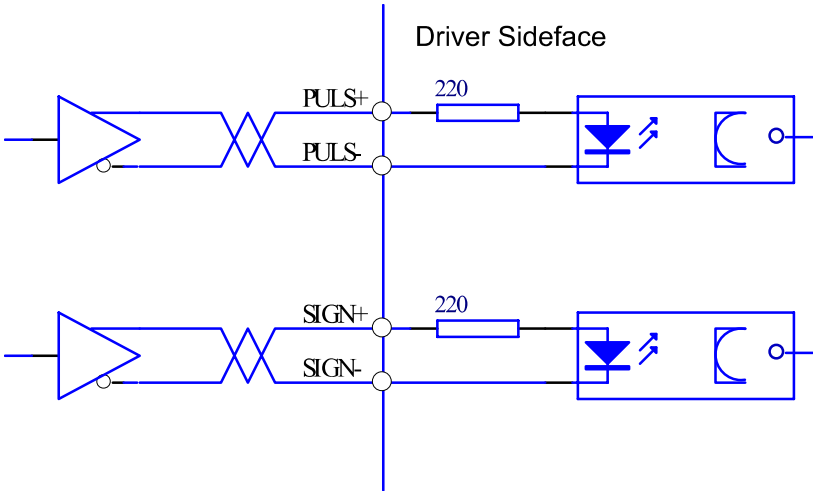


Fig. 3.7 Type3 Differential Drive Mode of Pulse Input Interface

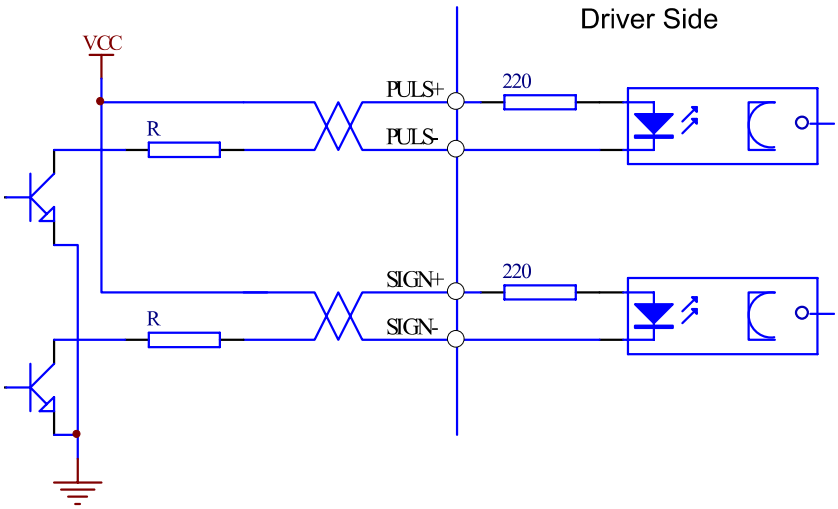


Fig. 3.8 Type4 Uni-polar Drive Mode of Pulse Input Interface

（1） For correctly transmitting pulse data, it is recommended to adopt the differential drive mode;

（2） Under differential drive mode, AM26LS31 and MC3487 or similar cable driver of RS422;

（3） The uni-polar drive mode will reduce the motion frequency. According to the requirements on the pulse amount input circuit: driving current 10～15mA and limited maximal external power voltage of 25V, empirical data are as follows: VCC=24V,R=1.3～2k;VCC=12V,R=510～

820Ω;VCC=5V,R=82～120Ω.

（4） When adopting uni-polar drive mode, the external power supply will be provided by the user, but attention must be given to the case that if the power electrodes are reversely connected, the servo drive unit may be damaged.

（5） Refer to Table 3.4 for details about the pulse input forms, in which the arrow means counting trend. Table 3.5 shows the time sequence and parameters for pulse input.

Table 3.4 Pulse Input Forms

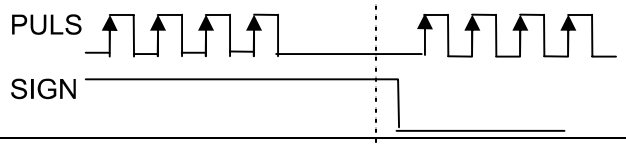
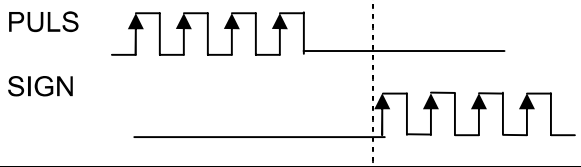
Forms of Pulse Command	CCW	CW	Set Parameter Values
Symbol for Pulse Train			0 Command Pulse＋Symbol
CCW Pulse Train CW Pulse Train			1 CCW Pulse/CCW Pulse

Table 3.5 Time sequence Parameters for Pulse Input

Parameter	Differential Drive Input	Uni-polar Drive Input
t_{ck}	>2μS	>5μS
t_h	>1μS	>2.5μS
t_l	>1μS	>2.5μS
t_{rh}	<0.2μS	<0.3μS
t_{rl}	<0.2μS	<0.3μS
t_s	>1μS	>2.5μS
t_{qck}	>8μS	>10μS
t_{qh}	>4μS	>5μS
t_{ql}	>4μS	>5μS
t_{qrh}	<0.2μS	<0.3μS
t_{qrl}	<0.2μS	<0.3μS
t_{qs}	>1μS	>2.5μS

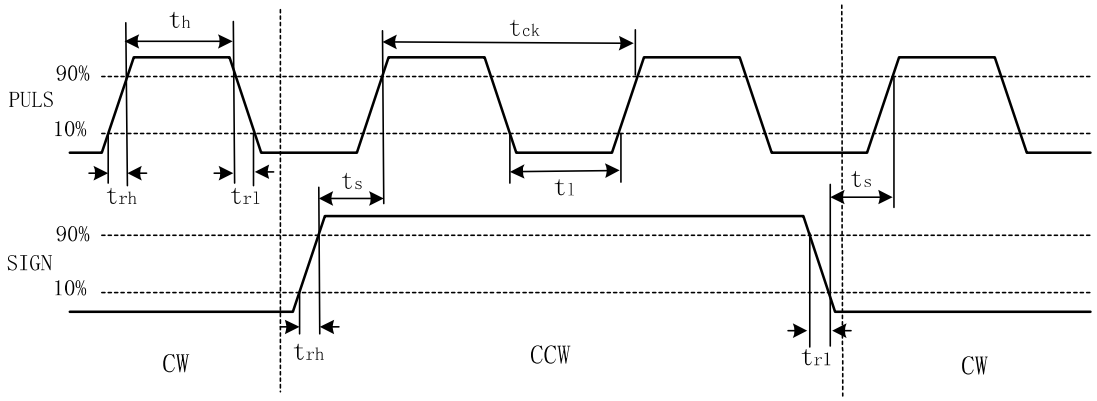


Fig. 3.9 Time sequence Diagram for Pulse＋Symbol Input Interface (Maximal Pulse Frequency:500kHz)

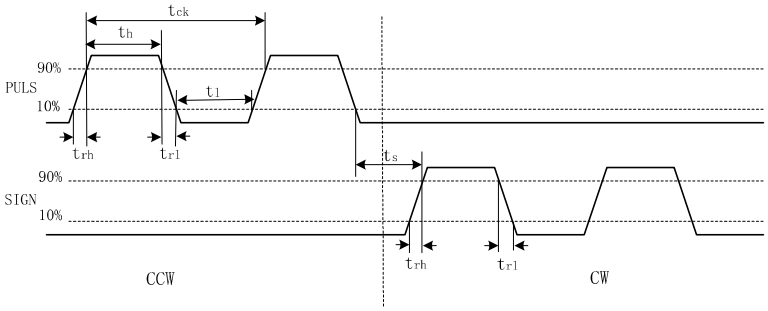


Fig. 3.10 Time sequence for CCW Pulse/CW Pulse Input Interface(Maximal Pulse Frequency:500kHz)

4) Driver Speed Signal Output Interface

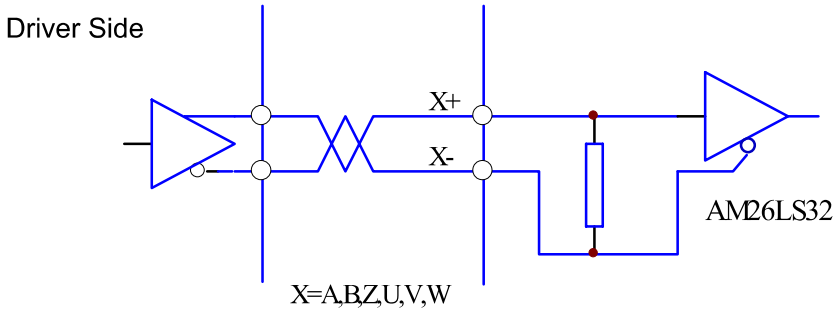


Fig. 3.11 Type5 Driver Speed Signal Output Interface

5) Input Interface for Servo Driver's Photoelectric Encoder

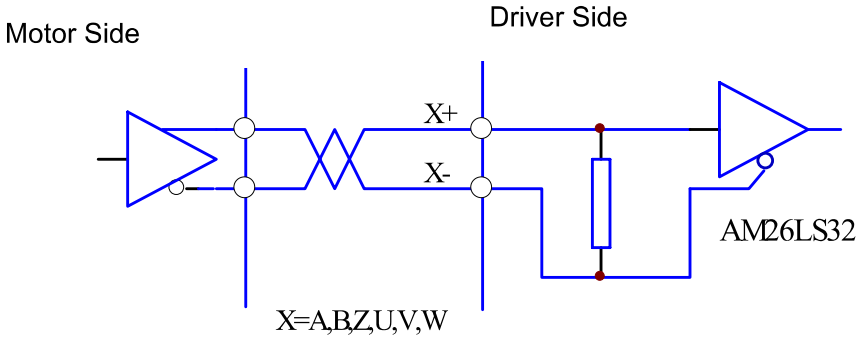


Fig. 3.12 Input Interface for Servo Driver's Photoelectric Encoder

4.2 Functions of Parameters

Table 4.2 Functions of Parameters

No	Name	Functions	Parameter range
0	Password	Prevent the parameters from being wrongly changed. Generally, when needing to set parameters, first set the parameter as the correct password, then set the parameters. After commissioning, reset the parameter back to 0 to ensure that the parameter will not be wrongly changed in the future. Passwords have different classes respectively for user parameters, system parameters and whole parameters. When changing model code parameter (PA1), the model code password must be used. Other passwords can not be used for changing this parameter. User password is 315. Model code password is 385.	0~9999
1	Model Code	Correspond to the same series of drive units and motors with different frequency classes. Different model codes correspond to different default parameter values. And when restoring the function of default parameter values, the parameter concerned must be correct. When EEPROM alarm (No.20) occurs, the parameter must be reset after the repair, then restore the default parameter value. Otherwise, the driver many not work normally or be damaged. When changing the parameter, first set the password PAO as 385, then the parameter concerned can be changed. Please refer to this chapter for detained meanings about parameters.	0~69
2	Software Version	The software version No. can be checked, but cannot be changed.	*
3	Initial Display State	Select display states of the driver screen after being electrified. 0: Display motor rotation rate; 1: Display that current position is lower by 5 (pulses); 2: Display that current position is higher by 5; 3: Display that current position command (means accumulated pulse value)is lower by 5; 4: Display that current position command (means accumulated pulse value)is higher by 5; 5:Display that position error is lower by 5; 6: Display that position error is higher by 5; 7:Display motor torque; 8:Display motor current; 9:Display linear velocity; 10:Display control mode; 11:Display position command pulse frequency; 12:Display speed command; 13:Display torque command; 14:Display absolute position of rotor during one round; 15:Display input terminal state; 16: Display output terminal state; 17:Display encoder input signal; 18:Display operation state; 19:Display alarm code; 20: Reservation.	0~20

39	Acceleration Time Constant	The set value means the acceleration time of motor ranging from 0r/min to 1000r/min The acceleration and deceleration have a linear feature. Only valid for speed control mode, invalid for position control mode. If the drive unit is used together with external position loop, this parameter shall be set as 0.	1ms ~10000ms
40	Deceleration Time Constant	The set value means the deceleration time of motor ranging from 0r/min to 1000r/min The acceleration and deceleration have a linear feature. Only valid for speed control mode, invalid for position control mode. If the drive unit is used together with external position loop, this parameter shall be set as 0.	1ms ~10000ms
41	Numerator of Output Electronic Gear Ratio	Feedback pulse from each coil of the encoder will be output through the gear within the drive unit. E.g., there are 2500 pulses in each coil of the encoder, setting PA41/42=4/5, then the A and B-phase signals output from the drive unit will be 2500 X PA41/PA42=2000 pulses/coil.	0～255
42	Denominator of Output Electronic Gear Ratio	This parameter must be more than or the same with parameter No.41.	0～255
43	Choice of Speed Command	Whether the operation speed is from internal speed or analog command: 0 Internal speed 1 Analog command	0~1
44	High Speed AD Zero Point	When restoring default value, this parameter will not be recovered.	412~1600
45	Low Speed AD Zero Point	When restoring default value, this parameter will not be recovered.	412~1600
46	Motor Rotation Direction Control	0 Normal 1 Opposite to the analog speed command 2 Opposite to the output pulse rotation direction 3 Opposite to both.	0～3
47	Analog Command Gain	Analog command is transited to speed gain.	20～3000
48	Anti-jamming Scope for Analog Command	The function is the same with parameter No.33, but the function range includes all speeds rather than zero speed only. It is recommended that this parameter not be used simultaneously together with parameter No.33.	0～1000
49	Choice of Zero Adjustment Channels for Analog Speed	0 Low speed AD for low speed and high speed AD for high speed. 1 High speed AD for both high and low speed This parameter is used only for AD zero adjusting: to improve the resolving power of analog command, AD switch with different multiplying factors are employed for high and low speed. First set this parameter as 1 to adjust zero point for high speed AD, then set the parameter as 0 to adjust zero point for low speed AD.	0～1
52	Analog Command Transition Mode	0 Speed AD is transited to speed command with curve of second order 1 Speed AD is transited to speed command with straight line.	0～1

5.2 Methods for Handling Alarms

Table 5.2 Methods for Handling Alarms

Alarm Code	Alarm Name	Operation State	Reasons	Handling Methods
1	Excessive Speed	Occur when switching on the control power	①Failure in control power board ②Encoder failure	①Replace servo driver. ②Replace servo motor.
		Occur during motor operation	Excessively high input command pulse frequency.	Correctly set the input command pulse.
			Excessively small acceleration/ deceleration time constant results in overshoot.	Increase acceleration/deceleration time constant.
			The input electronic gear ratio is excessively high.	Correctly set the ratio.
			Encoder failure.	Replace servo motor.
			Bad encoder cable.	Replace encoder cable.
			Instable servo system causes overshoot.	①Reset related gains. ②If the gains cannot be set at a proper value, reduce inertia rate for load rotation
		Occur immediately after the motor is started	Excessively large load inertia.	①Reduce load inertia. ②Replace it with driver and motor of greater frequency.
			Zero-point error in encoder.	①Replace servo motor ②Contact the manufacturer for readjusting the zero point.
			①Wrong connection of motor leads of U, V and W. ②Wrong connection of encoder cable leads	Correct wiring.
2	Over-voltage in Main Circuit	Occur when switching on control power	Failure in circuit board.	Replace servo drive unit.
		Occur when switching on main power supply	①Excessively high power voltage. ②Abnormal wave pattern of power voltage.	Examine the power supply source.
		Occur during motor operation	Disconnection of braking resistance.	Rewiring.
			①Braking transistor is damaged. ②Internal braking resistance is damaged.	Replace servo drive unit.
			Insufficient capacity in the braking return circuit.	①Reduce start/stop frequency . ②Increase acceleration/deceleration time constant. ③Reduce torque limiting value. ④Reduce load inertia. ⑤Replace it with driver and motor of greater frequency.

3	Voltage Shortage in Main Circuit	Occur when switching on main power supply	①Failure in circuit board. ②Power fuse is damaged. ③Failure in soft start circuit. ④Damaged rectifier.	Replace servo drive unit.
			①Excessively low power voltage. ②Temporary power failure for more than 20mS.	Examine the power supply.
3	Voltage Shortage in Main Circuit	Occur during motor operation	①Insufficient power capacity . ②instantaneous power failure.	Examine the power supply.
			Heat radiator overheating.	Examine load.
4	Position Excess	Occur when switching on control power	Failure in circuit board.	Replace servo driver.
		After switching on main power supply and control wire, the motor does not work when inputting command pulse	①Wrong connection of motor leads of U, V and W. ②Wrong connection of encoder cable leads.	Correct wiring.
			Encoder failure.	Replace servo motor.
			Set inspection range for position excess.	Extend inspection range for position excess.
			Position proportion gain is too small.	Increase the gain.
			Insufficient torque.	①Examine torque limiting value. ②Reduce loading capacity. ③Replace it with drive unit and motor of greater frequency.
			Excessively high command pulse frequency.	Reduce the frequency.
5	Motor overheating	Occur when switching on control power	Failure in circuit board.	Replace servo drive unit.
			①Cable disconnection . ②Internal temperature relay of motor is damaged.	①Examine cable. ②Examine motor.
		Occur during motor operation	Motor overload.	①Reduce load. ②Reduce start/stop frequency. ③Reduce torque limiting value. ④ Reduce related gain. ⑤Replace drive unit and motor of greater frequency.
			Internal failure in motor.	① Replace servo motor.
6	Saturation Failure of Speed Regulator	Occur during motor operation	Motor gets stuck by the machinery.	Examine machinery part of the load.
			Overload.	①Reduce load. ②Replace it with drive unit and motor of greater frequency.
7	Abnormal Drive Stopping		Disconnection of CCW and CW drive stopping input terminal.	Examine the wire connection and power of the input terminal.

8	Overflow of Position Error Meter		①The motor get stuck by the machinery. ②Abnormal input command pulse.	①Examine the machinery part of the load. ②Examine command pulse. ③.Examine whether the motor works after receiving command pulse
9	Encoder failure		Wrong connection of encoder.	Examine wire connection.
			Damaged encoder.	Replace motor.
			Bad encoder cable.	Replace cable.
9	Encoder Failure		Excessively long encoder cable results in low power voltage of encoder.	①Shorten cable. ②Supply power with multiple core wires connected in parallel.
10	Voltage Shortage in Control Power		Low input control power.	Examine control power.
			①Internal connectors of driver have bad performance. ②Abnormal switch power. ③Damaged chip.	①Replace drive unit. ②Examine connector. ③Examine switch power.
11	Encoder Failure	Occur when switching on control power	Failure in circuit board.	Replace servo drive unit.
		Occur during motor operation	①Excessively low power voltage. ②Overheating.	①Examine drive unit. ②Re-electrify. ③Replace drive unit.
			Short circuit between motor leads of U, V and W.	Examine wire connection.
			Bad ground contact.	Correct ground contact.
			Damaged motor insulation.	Replace motor.
			Be jammed.	①Add circuit filter. ②Keep away from jamming source
12	Excessive Current		Short circuit between motor leads of U, V and W.	Examine wire connection.
			Bad ground contact.	Correct ground contact.
			Damaged motor insulation.	Replace motor.
			Damaged driver.	Replace drive unit.
13	Overload	Occur when switching on control power	Failure in circuit board.	Replace servo drive unit.
		Occur during motor operation	Operation by exceeding rated torque.	①Examine load. ②Reduce start/stop frequency. Reduce torque limiting value. Replace it with drive unit and motor of greater frequency
			Hold brake cannot be opened.	Examine the hold brake.
			Instable motor with vibration.	①Increase gain. ②Increase acceleration/deceleration time. ③reduce load inertia.

			①One phase of U, V and W is disconnected. ②Wrong connection of encoder.	Examine wire connection.
14	Braking failure	Occur when switching on control power	Failure in circuit board.	Replace servo drive unit.
		Occur during motor operation	Disconnection of brake resistance.	Re-wiring.
			①Damaged brake transistor. ②Internal brake resistance is damaged.	Replace servo drive unit.
14	Braking failure	Occur during motor operation	Insufficient capacity in the brake loop.	①Reduce stop/start frequency. ②Reduce acceleration/deceleration time. ③Reduce torque limiting value. ④Reduce load inertia. ⑤Replace drive unit and motor of greater frequency.
			Excessively high main circuit power voltage.	Examine main circuit.
15	Counting Error of Encoder		Damaged encoder.	Replace motor.
			Wrong connection of encoder.	Examine wire connection.
			Bad ground contact.	Correct ground contact.
20	EEPROM Error		Damaged chip or circuit board.	Replace servo drive unit. .After repair, first reset driver model (parameter No.1), then restore default parameter value.
30	Z Pulse Losing in Encoder		Z pulse does not exist; damaged encoder Bad cable Bad cable shielding Bad connection between shielded wire and shielding layer Failure in encoder's interface circuit	Replace encoder Examine encoder's interface circuit
31	UVW Signal Error in Encoder		Damaged UVW signal of encoder Damaged Z signal of encoder Bad cable Bad cable shielding Bad connection between shielded wire and shielding layer Failure in encoder's interface circuit	Replace encoder Examine encoder interface circuit
32	Code Violation of Encoder's UVW Signal		Damaged UVW signal of encoder Bad cable Bad cable shielding Bad connection between shielded wire and shielding layer Failure in encoder's interface circuit	Replace encoder Examine encoder interface circuit

Chapter Six Display and Operation

6.1 Keyboard Operation

- The panel of drive unit is composed of six-phase LED nixie tube display and four keys of ▲ ▼ ◀ ↵, which are used to display states of drive units and set parameters. The specific functions of the keys are as follows :

  ▲: Increase serial number and value or forward operation.

  ▼ : Decrease serial number and value or backward operation.

  ◀: Return to the preceding layer of men or cancel the operation.

  ↵: Enter the next layer of menu or confirm the input.

Note: Press down ▲ or ▼ and hold on, the operation will be repeated. The longer the key is kept being pressed down, the faster the repetition frequency will be. .

- The six-phase LED nixie tube display can show various states and data about the system. If decimal point of all the nixie tubes or the nixie tube on the fastest right side keeps flashing, it means alarm.
- Operation is conducted according to multi-layer menu, in which the first layer is the main menu, including eight operation modes; the second layer is functional menu to the operation mode under first layer. Fig. 6.1 is a block chart for operations in the main menu, as shown in the following:

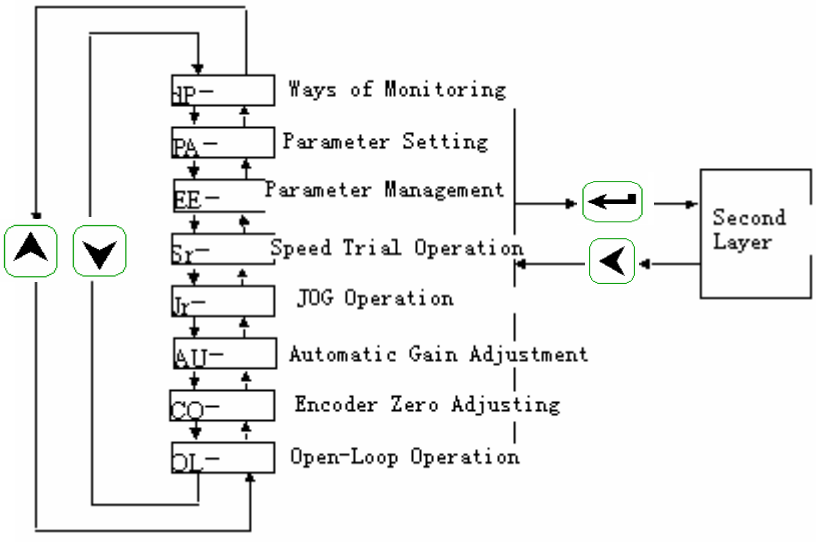


Fig. 6.1 Block Chart for Operation Mode Selection

6.2 Ways of Monitoring

Select “dP-” In the first layer and press down, then you will enter the ways of monitoring, in which there are 21 display states. Use and to select the display mode, then press down to enter the specific display state.

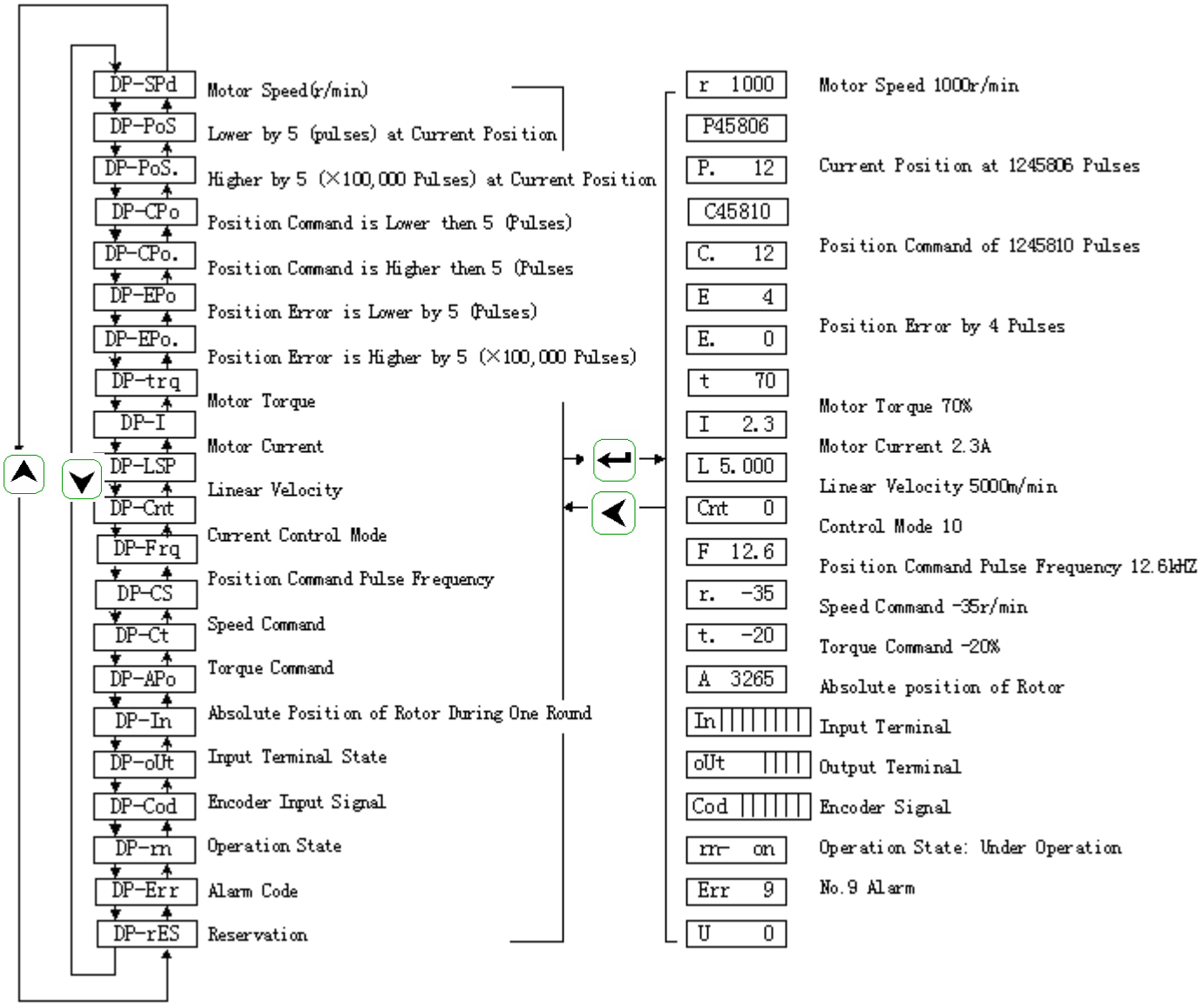


Fig. 6.2 Operation Chart for Ways of Monitoring

Note 1： Position and command pulse are both values multiplied by input electronic gear.

Note 2： Pulse unit is the internal pulse unit of the system, where it means 100,000 pulses/round. The pulse is expressed by value higher by 5 plus value lower by 5. The calculation method is: Pulse=(value higher by 5)×100,000 + value lower

130SJT-M150B	2.3	4	15	1500	8.5	3.1×10^{-3}	33	220(300)
130SJT-M150D	3.9	4	15	2500	14.5	3.6×10^{-3}	63	220(300)

Note: The user shall make a special indication if he or she wants to order motor with electricity-losing brake.

Table 8.9 Specifications on ST Series of Some Motors

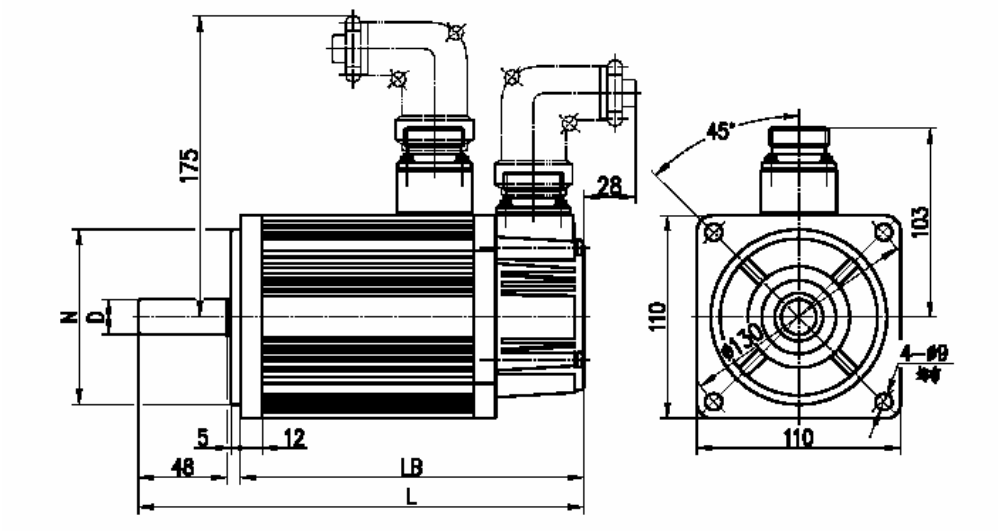
Models	Power (kw)	Zero Speed	Rated Rotation	Rated Current	Rotor Inertia	Machinery Time	Operating Voltage	Weight (kg)
110ST-M02030H	0.6	2	3000	4.0	0.33×10^{-3}	3.64	220(300)	4.2
110ST-M04030H	1.2	4	3000	7.5(5.0)	0.65×10^{-3}	2.32	220(300)	5.2
110ST-M05030H	1.5	5	3000	9.5(6.0)	0.82×10^{-3}	2.03	220(300)	5.8
110ST-M06020H	1.2	6	2000	8.0(6.0)	1.00×10^{-3}	1.82	220(300)	6.4
110ST-M06030H	1.8	6	3000	11.0(8.0)	1.00×10^{-3}	1.82	220(300)	6.4
130ST-M04025H	1.0	4	2500	6.5(4.0)	0.85×10^{-3}	3.75	220(300)	7.4
130ST-M05025H	1.3	5	2500	6.5(5.0)	1.06×10^{-3}	3.07	220(300)	7.9
130ST-M06025H	1.5	6	2500	8.0(6.0)	1.26×10^{-3}	2.83	220(300)	8.6
130ST-M07720H	1.6	7.7	2000	9.0(6.0)	1.58×10^{-3}	2.44	220(300)	9.5
130ST-M10015H	1.5	10	1500	9.0(6.0)	2.14×10^{-3}	2.11	220(300)	11.1
130ST-M10025H	2.6	10	2500	14.5(10.0)	2.14×10^{-3}	2.11	220(300)	11.1
130ST-M15015H	2.3	15	1500	13.5(9.5)	3.24×10^{-3}	1.88	220(300)	14.3

Note 1: The value within the bracket in the column of rated current is the rated current under high voltage

Note 2: The user shall make a special indication if he or she wants to order motor with electricity-losing brake

3) Figuration Dimension

(1) Figuration and installment dimension for SJT series of 110-stand model AC servo motors



Chapter Nine Order Instructions

9.1 Capacity Selecting

The determination of capacity of the servo system must comprehensively consider factors like load inertia, load torque, required positioning precision and maximal speed. The following steps are recommended for the considerations:

1）Calculating Load Inertia and Torque

By referring to related data, make a calculation on load inertia, load torque, acceleration/deceleration torque and effective torque, which will serve as the basis for choice at the next step.

2） Initial Determination of Machinery Gear Ratio

According to the required maximal speed and rotation rate of motor, make a calculation on the maximal machinery deceleration ration. This ratio and the minimal gyration unit of motor will be used to calculate whether the requirement for minimal position unit is satisfied. If a relatively high position precision is required, increase the machinery deceleration ratio (the actual maximal speed is decreased) or choose motor with higher rotation rate.

3）Calculating Inertia and Torque

The load inertia and torque can be converted to the motor shaft with machinery deceleration ratio. The converted inertial shall be no more than 5 times of the inertia of the motor rotor and the converted load torque and effective torque shall be not higher than the rated torque of motor. If these requirements cannot be satisfied, increase machinery ratio (the actual maximal speed is decreased) or choose motor with greater capacity.

9.2 Electronic Gear Ratio

Refer to Chapter Four (Table 4.2 Functions of Parameters), Chapter Six (Parameter Setting) and Chapter Seven (7.3 Adjustment) for the meaning and adjustment of electronic gear ratio G.

Under position control mode, the actual speed of load is:

command pulse speed×G×machinery deceleration ratio.

Under position control model, the actual minimal displacement of load is:

minimal command pulse itinerary×G×machinery deceleration ratio.

Note: When the electronic gear ratio G is not 1, the dividing operation of G may have a

remainder, which means that position errors exist. The maximal error is the minimal rotation value (minimal resolution) of the motor.

9.3 Stop Features

Under position control mode, there will be a difference between the command pulse and feedback pulse when using pulse train to control the servo motor. This difference will be accumulated in the position error meter and form the following relationships with the command pulse frequency, electronic gear ratio and position proportion gain:

$$\varepsilon = \frac{f^* \times G}{K_p}$$

in which,

ε: Lag Pulse (Puls);

f: Command Pulse Frequency (Hz);

Kp: Position Proportion Gain (1/S);

G: Electronic Gear Ratio.

Note: the above relationship is reached under the condition that position feed-forward gain is 0%. If the position feed-forward gain is >0%, the lag pulse will be less than that in the above calculation formula.

9.4 Calculation Method for Selecting Models of Servo System and Position Controller

1) Command displacement and actual displacement:

$$S = \frac{I}{\delta} \cdot \frac{CR}{CD} \cdot \frac{DR}{DD} \cdot \frac{1}{ST} \cdot \frac{ZD}{ZM} \cdot L$$

in which,

S: actual displacement mm;

I: command displacement mm;

δ: minimal unit of CNC mm;

CR: command frequency multiplication coefficient;

CD: command frequency division coefficient

DR: servo frequency multiplication coefficient;

DD: servo frequency division coefficient;

ST: grade division value per round of motor;

ZD: gear number of side gear of motor;

ZM: gear number of side gear of lead screw;

L: pitch of lead screw mm

Generally, S=I, which means command value equals actual value.

2) Maximal Command Speed of CNC:

$$\frac{F}{60 \times \delta} \cdot \frac{CR}{CD} \le f_{\max}$$

In which, F: command speed mm/min;

f_{max}: maximal output frequency of CNC Hz(GSK980 为 128000).

3) Maximal Speed of Servo System:

$$V_{\max} = n_{\max} \times \frac{DR}{DD} \times L$$

In which, V_{max}: maximal velocity of work bench permitted by the servo system mm/min;

n_{max}: maximal rotation rate permitted by the servo motor r/min;

Actual maximal speed of machine tools is restricted by maximal speed of CNC and the servo system.

$$\alpha = INT\left[INT\left(N \cdot \frac{CR}{CD} \right) \cdot \frac{DR}{DD} \right]_{\min} \cdot \frac{1}{ST} \cdot \frac{ZD}{ZM} \cdot \frac{L}{\delta}$$

in which, α: minimal displacement of machine tools mm;

N : natural number;

INT( ): integral number;

INT [ ]$_{min}$: minimal integral num

广州数控设备有限公司
GSK CNC EQUIPMENT CO., LTD.

Add: No.52, 1st . Street, Luochong North Road, Luochongwei, Guangzhou, 510165, China

Website: http://www.gsk.com.cn E-mail: gsk@gsk.com.cn

Tel: 86-20-81796410/81797922 Fax: 86-20-81993683

All specification and designs are subject to change without notice. Aug. 2007/Edition 1

Aug. 2007/Printing 1

ANEXO 8. CARACTERÍSTICAS DEL ACERO ASTM A-500

ASTM A500 Steel, grade B, shaped structural tubing

Categories: Metal; Ferrous Metal; ASTM Steel; Carbon Steel; Low Carbon Steel

Material Notes: The Cu content of 0.18% is a minimum content when copper steel is specified.

Key Words: copper steels, copper-steels, UNS K03000, ASTM A501

Vendors: **Click here to view all available suppliers for this material.**

Please click here if you are a supplier and would like information on how to add your listing to this material.

Physical Properties	Metric	English	Comments
Density	7.85 g/cc	0.284 lb/in³	Typical of ASTM Steel

Mechanical Properties	Metric	English	Comments
Tensile Strength, Ultimate	400 MPa	58000 psi	
Tensile Strength, Yield	315 MPa	45700 psi	
Elongation at Break	23.0 %	23.0 %	
Bulk Modulus	140 GPa	20300 ksi	Typical for steel
Shear Modulus	80.0 GPa	11600 ksi	Typical for steel

Material Components Properties	Metric	English	Comments
Carbon, C	<= 0.30 %	<= 0.30 %	
Copper, Cu	<= 0.18 %	<= 0.18 %	
Iron, Fe	99.0 %	99.0 %	
Phosphorous, P	<= 0.050 %	<= 0.050 %	
Sulfur, S	<= 0.0630 %	<= 0.0630 %	

References for this datasheet.