



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE
COMUNICACIÓN**

**“TARJETAS DE ENTRENAMIENTO PARA EL APRENDIZAJE DE
ELECTRÓNICA PROGRAMADA CON INTEL GALILEO BASADA
EN LA PLATAFORMA ARDUINO”**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERA EN ELECTRÓNICA Y REDES DE
COMUNICACIÓN**

AUTORA: SAMANTHA MARISOL MESA TAPIA

DIRECTOR: ING. DANIEL JARAMILLO

IBARRA-ECUADOR

2016



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA

UNIVERSIDAD TÉCNICA DEL NORTE

1.- IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional, determinó 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 DEL CONTACTO	
Cédula de Identidad	10039735-1
Apellidos y Nombres	Mesa Tapia Samantha Marisol
Dirección	San José de Cananvalle
E-mail	sammysolmesa@hotmail.com
Teléfono Fijo	062542027
Teléfono Móvil	0999063232
DATOS DE LA OBRA	
Título	“TARJETAS DE ENTRENAMIENTO PARA APRENDIZAJE DE ELECTRÓNICA PROGRAMADA CON INTEL GALILEO G2 BASADO EN LA PLATAFORMA ARDUINO.”
Autora	Mesa Tapia Samantha Marisol
Fecha	28-10.2016
Programa	Pregrado
Título por el que se aspira:	Ingeniería en Electrónica y Redes de Comunicación
Director	Ing. Daniel Jaramillo

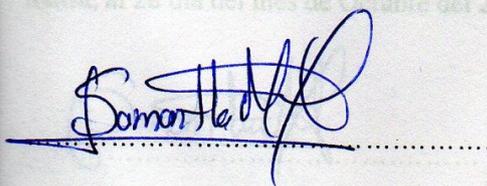
2.- AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, SAMANTHA MARISOL MESA TAPIA, con cédula de identidad Nro. 100339735-1, en calidad de autora y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma 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 de material y como apoyo a la educación, investigación y extensión, en concordancia con la ley de Educación Superior artículo 144.

3.- CONSTANCIAS

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

Ibarra, al 28 del mes de Octubre del 2016



Samantha Marisol Mesa Tapia

100339735-1



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DECLARACIÓN

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

Yo, Samantha Marisol Mesa Tapia, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, y que éste no ha sido previamente presentado para ningún grado o calificación profesional.

Yo, SAMANTHA MARISOL MESA TAPIA, con cédula de identidad Nro. 100339735-1, 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 y 6, en calidad de autor del trabajo de grado denominado: "TARJETAS DE ENTRENAMIENTO PARA APRENDIZAJE DE ELECTRÓNICA PROGRAMADA CON INTEL GALILEO G2 BASADO EN LA PLATAFORMA ARDUINO.", que ha sido desarrollado para optar el título de Ingeniería en Electrónica y Redes de Comunicación, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos concedidos anteriormente. En mi condición de autora 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, al 28 día del mes de Octubre del 2016

Samantha Marisol Mesa Tapia

100339735-1



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DECLARACIÓN

Yo, Samantha Marisol Mesa Tapia, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; y que éste no ha sido previamente presentado para ningún grado o calificación profesional.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Técnica del Norte, según lo establecido por las Leyes de Propiedad Intelectual, Reglamentos y Normatividad vigente de la Universidad Técnica del Norte

Samantha Marisol Mesa Tapia

100339735-1



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

CERTIFICACIÓN

Certifico que la Tesis “TARJETAS DE ENTRENAMIENTO PARA APRENDIZAJE DE ELECTRÓNICA PROGRAMADA CON INTEL GALILEO G2 BASADO EN LA PLATAFORMA ARDUINO.”, ha sido realizada en su totalidad por la señorita: SAMANTHA MARISOL MESA TAPIA portadora de la cédula de identidad número:

100339735-1

.....

Ing. Daniel Jaramillo

Director de Tesis

DEDICATORIA

Quiero dedicar este proyecto de titulación con mucho cariño a los seres más importantes en mi vida, mis padres Marisol y Patricio, y a mi hermano Israel, por su apoyo incondicional y su confianza depositada en mí en todo momento, también a las personas que se han cruzado en mi vida familiares y amigos que han aportado en mi vida para llegar a culminar mi carrera universitaria.

AGRADECIMIENTO

Gracias a Dios por guiar mis pasos y permitirme tener a mi hermosa familia unida, siempre con una voz de aliento, gracias a mis padres por todo su amor, por su lucha, constancia, y gran ejemplo de superación, gracias a mi familia, a mi tía Janeth porque es la muestra de que el amor de familia perdura a pesar del tiempo y la distancia, gracias a Byron Granda, que siempre estuvo a mi lado apoyándome, gracias a un gran ejemplo de mujer y profesional Hilda Herrera, gracias a los buenos maestros que han aportado con sus conocimientos profesionales y sobre todo a los que dejando la hora de clase, dedicaron un poco de su tiempo para brindar un consejo sincero, gracias a los buenos amigos, y a todos los que de una u otra forma fueron parte de este crecimiento, gracias.

ÍNDICE

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	ii
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	¡Error! Marcador no definido.
DECLARACIÓN	¡Error! Marcador no definido.
CERTIFICACIÓN.....	¡Error! Marcador no definido.
DEDICATORIA.....	vii
AGRADECIMIENTO.....	viii
ÍNDICE.....	ix
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE TABLAS.....	xv
RESUMEN.....	xvii
ABSTRACT	xviii
PRESENTACIÓN	xix
CAPITULO I.....	1
ANTECEDENTES	1
1.1. PROBLEMA.....	1
1.2. OBJETIVOS	2
2.1.1. Objetivo General:	2
2.1.2. Objetivos Específicos:	3
1.3. ALCANCE.....	3
1.4. JUSTIFICACIÓN	5
CAPÍTULO II.....	7

MARCO TEÓRICO	7
2.1. TEORÍAS Y PRÁCTICAS DE LA ENSEÑANZA	7
2.1.1. Enseñanza	7
2.1.2. Aprendizaje.....	8
2.1.3. Didáctica.....	11
2.1.4. Métodos de enseñanza modernos	13
2.1.5. Programación de la enseñanza. (Davini, 2008).	19
2.1.6. Definición de objetivos de aprendizaje. (Davini, 2008).....	20
2.2. ELECTRÓNICA.....	21
2.2.1. Fundamentos.....	21
2.2.2. Elementos pasivos	22
2.2.3. Sensores	28
2.2.4. Actuadores y visualizadores	35
2.3. SISTEMAS EMBEBIDOS	36
2.3.1. Introducción.....	36
2.1.1. Hardware	38
2.3.2. Firmware.....	38
2.3.3. Sistema operativo	38
2.3.4. Aplicación.....	38
2.3.5. Diseño del sistema (Vahid & Tony, 1999).....	39
2.3.6. Software de diseño	45
2.4. INTEL GALILEO GEN 2	49
2.4.1. Introducción.....	49
2.4.2. Características importantes (INTEL, 2015).....	52
2.4.3. Intel Quark SoC x1000(INTEL, 2015).....	56
2.4.4. Desarrollo(INTEL, 2015)	58

2.4.5.	Arduino IDE y otros lenguajes	58
2.4.6.	IoT	59
2.4.7.	Linux Embebido	61
CAPÍTULO III		64
3.	DISEÑO Y DESARROLLO DEL SISTEMA	64
3.1.	DISEÑO.....	64
3.1.1.	Análisis de tarjetas de entrenamiento	64
3.1.2.	Análisis de guía	74
3.1.3.	Requerimientos de tarjetas de entrenamiento	76
3.2.	Requerimientos de guías	86
3.3.	DESARROLLO	87
3.3.1.	Diseño de placas	87
3.3.2.	Elección de sensores	88
3.3.3.	Desarrollo de placas.....	91
3.3.4.	Desarrollo de placas.....	103
3.3.5.	Desarrollo de guías	105
3.3.6.	Diagramas de flujo de funcionamiento.....	105
3.3.7.	Costo-beneficio.....	106
CAPITULO IV		113
PRUEBAS		113
4.1.	Pruebas del sistema.....	113
4.1.1.	Pruebas de placas	113
4.2.	Pruebas de guías.....	115
4.3.	Resultados	120
4.3.1.	Estudiantes principiantes sin guías	120
4.3.2.	Estudiantes principiantes con guías	120

4.3.3. Estudiantes con conocimientos medios sin guías	121
4.3.4. Estudiantes con conocimientos medios con guías.....	121
4.3.5. Estudiantes con experiencia sin guías	122
4.3.6. Estudiantes con experiencia con guías	122
4.4. Análisis de resultados	123
BIBLIOGRAFÍA	126

ÍNDICE DE FIGURAS

Figura 1: Modelo de Estudio de casos	13
Figura 2. Secuencia de programación de enseñanza.	19
Figura 3. Resistencia de carbón	23
Figura 4. Código de colores de resistencias	24
Figura 5. Resistencia con película de	24
Figura 6. Esquema de un resistor variable.....	25
Figura 7. Tipos de condensadores	26
Figura 8. Efecto de una inductancia en corriente continua.....	27
Figura 9. Funcionamiento del Diodo	28
Figura 10. Sensor HC-SR04	31
Figura 11. Sensor DHT 11.....	33
Figura 12. Diagrama de conexión.....	34
Figura 13. Diagrama de conexión PIR	35
Figura 14. Sensor PIR.....	35
Figura 15. Uso de transistores en CI	37
Figura 16. Tipos de procesadores	40
Figura 17. Distribución de pines Arduino Uno	91
Figura 18. DISEÑO PUERTOS DIGITALES ENTRADA SALIDA.....	92
Figura 19. PLACAS PUERTOS ENTRADA SALIDA DIGITAL.....	93
Figura 20. DISEÑO DE PLACA CAD ENTRADA	94

Figura 21. PLACA ENTRADA CADS	94
Figura 22. DISEÑO MOTORES ANÁLOGOS DIGITALES	95
Figura 23. PLACAS MOTORES A/D	95
Figura 24. DISEÑO PLACA DISPLAY	96
Figura 25. DISEÑO DE PLACA DISPLAY	96
Figura 26. DISEÑO DE ENTRENADOR ANÁLOGO DIGITAL.....	97
Figura 27. DISEÑO PLACA ENTRENADOR ANÁLOGO DIGITAL.....	98
Figura 28. DISEÑO PLACA ENTRENADOR ANÁLOGO DIGITAL.....	98
Figura 29. DISEÑO DE PLACA LCD.....	99
Figura 30. DISEÑO DE PLACA LCD.....	99
Figura 31. DISEÑO PLACAS TECLADO Y BLUETOOTH	100
Figura 32. DISEÑO DE PLACAS TECLADO Y BLUETOOTH.....	100
Figura 33. DISEÑO DE PLACAS SENSORES	101
Figura 34. DISEÑO DE PLACAS SENSORES	101
Figura 35. DISEÑO DE PLACA IOT	102
Figura 36. DISEÑO PLACA IOT	102
Figura 37. DISEÑO DE PLACA IOT.....	103
Figura 38. PLACAS TERMINADAS	104
Figura 39. Diagrama de flujo de funcionamiento.....	106

ÍNDICE DE TABLAS

Tabla 1: Secuencia básica del método por casos	13
Tabla 2. Secuencia básica para método de solución de problemas	15
Tabla 3. Secuencia básica para método de proyectos.....	16
Tabla 4. Secuencia básica de método de simulación.....	18
Tabla 5. Tipos de objetivos	20
Tabla 6. Tipos de resistores	22
Tabla 7. Características módulo HC-SR04.....	31
Tabla 8. Características sensor DTH11	32
Tabla 9. Formas para diagramas de flujo	41
Tabla 10. Análisis de tarjetas de entrenamiento	65
Tabla 11. Análisis de guía	74
Tabla 12. Requerimientos de arquitectura puertos como salida.....	78
Tabla 13. Requerimientos de arquitectura puertos como entrada	79
Tabla 14. Requerimientos de arquitectura CAD's	80
Tabla 15. Requerimientos de arquitectura motores/servos/chicharra.....	81
Tabla 16. Requerimientos de arquitectura teclado/lcd	82
Tabla 17. Requerimientos de arquitectura sensores	83
Tabla 18. Requerimientos de arquitectura aplicaciones IoT	84
Tabla 19. Requerimientos de arquitectura tarjetas completas	85
Tabla 20. Requerimientos de guía	86
Tabla 21. Elección de software de edición.....	87
Tabla 22. Elección de sensores.....	88
Tabla 23. Lista de materiales	106
Tabla 24. Costo por placa.....	109
Tabla 25. Consumo de corrientes	113

Tabla 26. Resultados estudiantes principiantes sin guías	120
Tabla 27. Resultados estudiantes principiantes con guías	120
Tabla 28. Resultados estudiantes con conocimientos medios sin guías	121
Tabla 29. Resultados estudiantes con conocimientos medios con guías.....	121
Tabla 30. Resultados estudiantes con experiencia sin guías	122
Tabla 31. Resultados estudiantes con experiencia con guías	122

RESUMEN

El sistema educativo existente está en constante cambio, el avance de la tecnología se vuelve un factor determinante para la educación de las personas, los métodos tradicionales dejan de tener efectividad cuando se tiene gran acceso a la información desde otras fuentes no certificadas y hacen que muchas personas puedan desorientarse ocasionando la pérdida de oportunidades, el desarrollo de plataformas libres permiten crear conocimiento de una mejor manera, permite que el estudiante se afronte con la realidad y permita prototipar con gran rapidez.

Específicamente en la universidad que ofertan carreras relacionadas con la tecnología es necesario que el estudiante tenga los recursos adecuados para un correcto aprendizaje, Intel Galileo Gen 2 es una herramienta de estudio que permite una rápida curva de aprendizaje por sus diferentes aplicaciones, el trabajo realizado permite enlazar el software proporcionado gratuitamente por Arduino para la gestión de aplicaciones del Intel Galileo y un hardware educativo que con guías apropiadas a la edad y el nivel del estudiante pueda trabajar en los laboratorios sin la necesidad de incurrir a demasiados costos y sobre todo permite que el material existente sea compartido con más generaciones ansiosas de conocimientos.

Los sistemas embebidos electrónicos son parte de la vida diaria y con este trabajo permite un rápido y mejor desarrollo asociados con una guía que brinde información y además rete al estudiante a probar sus conocimientos.

ABSTRACT

The existing education system is constantly changing, the advancement of technology becomes a determining factor for the education of the people. Traditional methods fail to be effective when you have great access to information from non-certified sources which cause many people to become disoriented and miss opportunities. The development of free platforms allows creating knowledge in a better way which allows the student to confront the reality and prototype very quickly.

Specifically in Universities that offer careers in technology is necessary that the student has adequate resources for proper apprentice. Intel Galileo Gen 2 is a study tool that allows the rapid learning for different applications. Its ultimate work permits the liaison of the free software, provided by Arduino for application management of Galileo Intel hardware, and the educational guides. These are age appropriate and adequate to the level of students who can work in laboratories without incurring too many costs, and over all, it allows the existing material to be a shared knowledge to future generations.

Electronic embedded systems are part of our daily lives and this work allows a faster and better development associated with a guide that provides information and also challenges the students to test their knowledge.

PRESENTACIÓN

El presente trabajo se divide en diferentes secciones que permiten una mejor comprensión de la información, así como demuestra el avance de todo el proceso realizado.

El capítulo 1 se enfoca a los antecedentes presentados para la aprobación del proyecto de titulación de tercer nivel, donde demuestra la problemática, sus objetivos, su alcance y su justificación.

El capítulo 2 se basa en el marco teórico necesario para el desarrollo de la parte práctica, haciendo un resumen de los métodos de enseñanza hasta los actuales y como se enfocan a la realidad tecnológica que vivimos, hace referencia a los elementos electrónicos utilizados tanto como activos y pasivos haciendo un especial recuento de la basta información y de aplicaciones que se pueden generar con la placa de desarrollo Intel Galileo Gen 2.

En el capítulo 3 se encuentra todo el desarrollo del Trabajo de Grado desde el levantamiento de requerimientos con el estándar IEEE 29148 hasta la conclusión de las placas, el capítulo 4 son las pruebas de funcionamiento, el plan de clase con la información adecuada y la forma de evaluación del estudiante, finalmente en el capítulo 5 son las conclusiones y recomendaciones del trabajo realizado.

CAPITULO I

ANTECEDENTES

1.1.PROBLEMA

En la materia de Sistemas Microprocesados se aprende el manejo de puertos, comunicación serial, conversión análoga/digital, visualizadores, interrupciones y timers realizando programas para destacar las características de cada uno de los temas y también para poder trabajar en proyectos haciendo uso de los mismos en conjunto.

Para cumplir con los requerimientos de la materia los estudiantes deben comprar materiales electrónicos como: teclado matricial, lcd, cable ttl, leds, resistencias, microcontroladores, pulsadores, cables y programadores para PIC o AVR's pero no todos pueden acceder al uso de estos por su costo, el mal uso de grabadores PIC o AVR's en algunas ocasiones bloquean a los microcontroladores lo que se convierte en la reincidencia de compra de dichos materiales electrónicos, y a la vez no se puede tener escalabilidad en este tipo de hardware ni realizar modificaciones según las necesidades de los proyectos que desarrollan los estudiantes, lo cual da como resultado que los estudiantes no lleguen a cumplir con sus objetivos esperados. La Universidad Técnica del Norte necesita obtener licencias de programas para la programación de microcontroladores brindando herramientas a los estudiantes, pero estas licencias deben estar en constante renovación lo cual también demanda de mayor capital estimado. Estas razones hacen que la educación de electrónica en cierta forma se quede un poco desactualizada, ya que no se utiliza herramientas nuevas para mejorar el conocimiento de

los estudiantes, en un mundo en donde la tecnología avanza de manera rápida y con algunos beneficios que aún no han sido explotados.

Con los entrenadores electrónicos y la placa Intel Galileo G2 que ofrece un entorno de desarrollo de hardware y software de código abierto basados en la plataforma Arduino se pueden lograr mejoras como: optimizar el tiempo de armado de los proyectos, ahorro de recursos económicos para los estudiantes, fomentar el desarrollo de nuevas formas de aprendizaje evitando el consumismo y desarrollando un criterio responsable sobre la tecnología. Estos entrenadores al ser escalables se diferencian de los boards existentes en el mercado porque estos no están diseñados con el fin de educar sino solo para ser usados en aplicaciones, su diseño es muy rígido y tienen limitaciones.

Al utilizar esta placa de última generación combinada con los boards como herramientas de aprendizaje se pueden realizar aplicaciones que utilicen IOT, estas aplicaciones pueden ser monitoreadas, controladas o verificadas desde cualquier lugar logrando cambios en el proceso de aprendizaje de electrónica, los beneficios van enfocados directamente a los estudiantes y de la misma manera realiza mejoras en la metodología de enseñanza de la materia.

1.2.OBJETIVOS

Objetivo General:

Diseñar e implementar tarjetas de entrenamiento para mejorar el aprendizaje de electrónica utilizando la placa Intel Galileo G2 basada en la plataforma Arduino.

Objetivos Específicos:

- Describir las características técnicas de los dispositivos electrónicos como sensores, módulos y de la placa Intel Galileo G2 para el manejo de sus funciones tanto en hardware y software.
- Realizar un estudio para determinar los sensores más óptimos para la realización de boards más complejos.
- Describir el procedimiento que se debe seguir para el diseño de los boards (tarjetas de entrenamiento), y las funciones que cumplen cada uno.
- Realizar el diseño de los boards básicos y de los más complejos, para tener variedad en la utilización de la placa Intel Galileo G2.
- Implementar cada board en la placa Intel Galileo G2 para probar el funcionamiento individual de cada uno y de los mismos en conjunto.
- Describir la metodología para el mejoramiento del aprendizaje al utilizar las tarjetas de entrenamiento.
- Verificar si existen mejoras en el aprendizaje de electrónica una vez utilizadas las tarjetas de entrenamiento.

1.3.ALCANCE

Utilizando la placa Intel Galileo G2 se realizará el diseño y la implementación de tarjetas de entrenamiento (boards), las cuales tendrán diferentes funciones, para hacer énfasis en el manejo de cada tema como: manejo de puertos, comunicación serial, conversión análoga/ digital, visualizadores, interrupciones y timers.

Cada tema mencionado tendrá una tarjeta de entrenamiento ya que son los principales temas vistos en la materia de Sistemas Microprocesados, y son la base para realizar

proyectos electrónicos más grandes y con más complejidad, por esta razón se realizarán 10 tarjetas entrenadoras que cumplan con estas con funciones principales (manejo de puertos, comunicación serial, conversión analógica/ digital, visualizadores, interrupciones y timers) y 5 boards más complejos que combinen las funciones de los boards sencillos con otros elementos electrónicos como sensores los mismos que estarán enfocados en ciertas variables como la comunicación inalámbrica, condiciones ambientales y otros aspectos que sirvan de uso en el proceso de aprendizaje.

Para determinar los sensores a usar en los boards más complejos se realizará un estudio, que determine los sensores más óptimos, de esta forma se obtiene variedad en la funcionalidad de la placa y se pueden cumplir con los objetivos de este anteproyecto.

Al utilizar estas tarjetas se realiza un ahorro de tiempo en la realización de los proyectos porque se suprime el armado con cables y hace que se vuelvan dinámicas las prácticas en electrónica ya que se conectarán directamente los pines de los boards a los pines de la placa Intel Galileo G2.

Una vez realizadas las tarjetas entrenadoras se aplicará una metodología para verificar el mejoramiento del aprendizaje, este proceso comienza con la captación de la información, la misma que será brindada como una guía para el uso de cada una de las tarjetas de entrenamiento por las personas que desarrollaran esta actividad. A continuación, los estudiantes o docentes tienen en cuenta, analizan y tratan de relacionar con el conocimiento existente en sus modelos mentales, lo procesan y sintetizan. Seguidamente, y dentro de la fase de interpretación, se internaliza la nueva información ya reflexionada, y se examina según sus conocimientos y habilidades. Como parte de una comprobación de la viabilidad de este proyecto se tendrá en cuenta parámetros medibles como: tiempo que se demoran los estudiantes en asociarse con nuevas tecnologías, tiempo

de armado de los proyectos, tiempo de desarrollo y solución de las aplicaciones y su visión para seguir creando a futuro proyectos diferentes.

Este proyecto es una herramienta para realizar aplicaciones que utilicen IOT, así las tarjetas de expansión pueden ser usadas en beneficio de la educación y la creación de aplicaciones.

Al ser compatible totalmente con la plataforma Arduino se obtendrá escalabilidad en las aplicaciones y de la misma manera se reutilizará el código para realizar cambios o mejoras en las mismas, un aspecto muy importante es que tiene como fin aportar al proceso de educación en la Universidad Técnica del Norte y será entregado en la Carrera de Ingeniería en Electrónica y Redes de Comunicación para que pueda usarse por docentes y estudiantes que consideren oportuno hacer uso del mismo.

1.4.JUSTIFICACIÓN

Al hacer un proyecto enfocado en la innovación de tecnología y en el mejoramiento de aprendizaje, se brinda herramientas para un continuo desarrollo tecnológico, el mismo que puede ayudar a resolver muchos de los problemas actuales de la sociedad, incentivando a estudiantes a realizar proyectos novedosos con diferentes fines.

El aporte de este proyecto para la Universidad Técnica del Norte, se relaciona directamente con la misión de la misma, ya que promueve el desarrollo y crecimiento del conocimiento de los estudiantes y le da un valor agregado a la institución por ser el origen de dichos conocimientos, los cuales sirven para realizar proyectos destinados al aprendizaje y a la resolución de problemas en diferentes campos.

La placa Intel Galileo Gen 2 es la primera en una familia de placas de desarrollo y prototipos certificadas por Arduino basadas en la arquitectura Intel y especialmente

diseñadas para fabricantes, estudiantes, educadores y entusiastas de la electrónica tipo "Hágalo usted mismo". Posee características técnicas nuevas en comparación a los dispositivos electrónicos tradicionales utilizados en clases, utiliza una arquitectura totalmente compatible con Arduino, ofrece a los usuarios un entorno de desarrollo de hardware y software de código abierto, complementa y amplía la línea de productos Arduino para ofrecer funciones informáticas más avanzadas.

Un aspecto importante es la escalabilidad que se puede tener al usar estas tarjetas de entrenamiento de electrónica, permitiendo que se puedan tener avances y nuevos desarrollos, este proyecto es muy rentable para los estudiantes, permite ampliar los conocimientos adquiridos en el transcurso de la carrera, también sirve para fomentar el desarrollo de nuevos proyectos que se enfoquen en nuevas herramientas para mejorar la calidad de enseñanza de una materia y de la carrera en sí.

CAPÍTULO II.

MARCO TEÓRICO

2.1.TEORÍAS Y PRÁCTICAS DE LA ENSEÑANZA

2.1.1. Enseñanza

(Davini, 2008) Indica que la enseñanza es una necesidad, una constante en la historia donde los adultos contaban sus historias a los jóvenes en diferentes prácticas sociales y en distintos ámbitos, las escuelas de hoy en día son creadas para la misma finalidad de transferir información de una persona a otra, la enseñanza es una práctica social e interpersonal, todos pueden ser maestros o instructores. A mediados del siglo XIX se comienza a preocupar por tener un sistema o una forma de enseñar, se crean los primeros sistemas escolares y es así que en el siglo XX se logra tener distintos niveles y especialidades. La existencia de la educación requiere de personas formadas para enseñar, es un acto especializado con un orden y que consta de algunas reglas básicas. En la sociedad humana una de las formas más importantes de aprender es cuando una persona o un grupo ayudan a otros a aprender, desde un profesor que enseña a leer o un padre enseña un correcto comportamiento.

La intención de enseñar es una acción voluntaria y conscientemente dirigida de una forma bidireccional que se pueden aplicar una gran variedad de recursos para promover resultados de transmisión de conocimientos. Habilidades y capacidades donde

ayudan principalmente a la persona, pero influye notablemente en la sociedad, donde su gran importancia es eliminar las brechas sociales existentes. (Davini, 2008).

2.1.1.1. La enseñanza como una secuencia metódica de acciones. (Davini, 2008).

La enseñanza es una forma de comunicación que implica una intención con un resultado de aprendizaje, lo que planteó Herbat a mediados del siglo XIX es una secuencia de pasos para llevar a cabo la enseñanza, durante décadas los pasos eran formales y estrictos, con el paso de los años se entiende que son muy variables y dependen del propósito de la enseñanza, del contenido de la materia, la característica de los alumnos y del contexto o ambiente, sin embargo siempre es necesario llevar un ordenamiento lógico que se va midiendo en función del grado de aceptación del grupo.

2.1.1.2. La enseñanza como mediación social y pedagógica. (Davini, 2008).

En los primeros años de la existencia del hombre su forma de supervivencia era el contar a las generaciones sus hallazgos para ser emulados por las siguientes generaciones, de esta manera la cultura y su desarrollo social fueron conservados. La idea de mediación pedagógica se basa en entender que la persona no es la que tiene el conocimiento, es el encargado de transmitir hacia el resto de personas y llegar a ampliarlo dependiendo del punto de vista del grupo que está aprendiendo.

2.1.2. Aprendizaje

El aprendizaje es considerado a la cantidad de información que se adquiere en las escuelas y colegios para poderlos aplicar y llegar a un conocimiento, pero en verdad los seres humanos necesitan aprender siempre en todas sus etapas de la vida para satisfacer necesidades o adquirir nuevas herramientas para la interacción social y adecuarse a las circunstancias cambiantes. El aprendizaje es la necesidad de adaptación para la interacción social de una manera activa y creativa que en muchos casos debe ser de un modo espontáneo que se relacionan directamente con el medio, aunque muchos aprendizajes requieren de que otras personas tengan la intención de enseñar para la asimilación de conocimientos, conceptos y herramientas para llegar a tener cierta autonomía en sus acciones.(Davini, 2008).

2.1.2.1.La dinámica individual y social del aprendizaje.(Davini, 2008).

En siglos pasados se pensaba que el aprendizaje era efectivo en un aislamiento entre el individuo y el medio, donde dificultó sus resultados principalmente en niños y jóvenes ya que no era un proceso totalmente activo ni participativo, el aprendizaje social se basa en que las personas adquieren conocimientos cuando una persona los orienta o un grupo, donde el profesor dirige el aprendizaje de cada estudiante y cada uno aprende del profesor y de la interacción del grupo donde adquiere habilidades cognitivas y nuevas formas de pensamiento.

2.1.2.2.El aprendizaje requiere de esfuerzo. (Davini, 2008).

El aprendizaje se presenta como un proceso feliz, donde permite el desarrollo personal y la mejor interacción con la sociedad, pero es necesario conocer que el

aprendizaje no es un proceso fácil, requiere de esfuerzo y dedicación, en algunas ocasiones es necesario eliminar malos hábitos o de realizar sacrificios para ganar una recompensa, es un proceso que en ocasiones genera resistencia dependiendo del interés y la complejidad, la única forma de superar estos limitantes es tomar consciencia y tomarlo en serio ya que nadie nos puede decir como pensar.

2.1.2.3.El profesor y el aprendizaje escolarizado (Davini, 2008).

Muchas investigaciones han dado como resultado que el ambiente escolar planteado en muchos sistemas de educación son los adecuados, son sistemas basados en el siglo XXI ya que como manifiesta (Robinson, 2008) “Las escuelas matan la creatividad”, aunque en cierto contexto las escuelas, colegios y universidades han cumplido y cumplen una función importante en la sociedad con la democratización de la educación su estructura lo aleja de una ambiente fértil para el aprendizaje por el aislamiento en aulas distanciándoles de las experiencias reales, con un grupo de muchos alumnos con un profesor que no puede estar bien preparado con una relación de marcada asimetría entre los alumnos e interacciones rígidas por textos y manuales donde definen que es importante aprender y hasta dónde con tiempos de aprendizaje reglados por horarios fijos y calendarios estrictos.

Los profesores son los encargados en relacionar los conocimientos del aula con enfoque en contextos realistas estimulando y facilitando las múltiples formas de conocimiento y herramientas culturales para involucrar al alumno en la reflexión, la búsqueda y el hallazgo, para obtener resultados autónomos y promoviendo actividades colaborativas.

2.1.3. Didáctica

Las personas que enseñan necesitan tener un dominio sobre el conocimiento a impartir para no llegar a inferencias, nadie puede enseñar aquello que no conoce, pero el saber no es suficiente, es necesaria una forma para transmitir y que sea entendible para profundizar y cumplir los objetivos de enseñanza. (Davini, 2008).

Existen 3 revoluciones que afectan directamente al acto de aprender; la interactividad, la cognición¹ humana y la gestión del sistema educativo, la didáctica se basa en el dominio de estas revoluciones para implementar una estrategia en clase y se puede generar aprendizaje en las personas cumpliendo sus expectativas, hoy en día haciendo especial énfasis en la interactividad sobre la tecnología y como se pueden realizar aprendizajes electrónicos, es decir tener una didáctica con medios informáticos que son formas de gran impacto para aprender. (Delacote, 1997)

La didáctica se maneja en relación experto/novato para garantizar la calidad del aprendizaje, donde el individuo presta mayor atención a la información de un docente capacitado donde mejora el desempeño individual y colectivo. (Delacote, 1997)

2.1.3.1. Didáctica y los profesores. (Davini, 2008).

Todas las personas pueden enseñar, pero los docentes ejercen como profesión tanto escuelas, colegios o universidades. Una profesión consiste en la búsqueda de

¹ Cognición: Capacidad del ser humano para procesar información

métodos adecuados y precisos para alcanzar sus fines, la práctica docente es disponer de medios adecuados para la transmisión educativa conforme a su intención.

Toda profesión no puede basarse solo en algo práctico, es necesario el apoyo de conocimientos con criterios y reglas de acción práctica para ser utilizados en ámbitos diferentes y puedan construir una didáctica. En base a su propia experiencia los profesores podrán:

- Analizar, adoptar o combinar distintos métodos según las intenciones educativas, como marcos generales, andamios o herramientas sistemáticas para la organización del proceso de enseñanza.
- Analizar e integrar los métodos generales y otros aportes derivados de los desarrollos de las didácticas específicas, considerando aspectos particulares de los contenidos, los sujetos y los contextos de enseñanza.
- Construir estrategias propias, a partir de aquellas herramientas de sostén, considerando las características de los alumnos, el contexto cultural y el ambiente de aprendizaje.
- Priorizar las enseñanzas relevantes en el marco del currículo, contribuyendo a su desarrollo y mejora, y considerando su valor para la formación de los alumnos.
- Reflexionar sobre sus propios enfoques o estilos y en qué medida ellos inhiben el desarrollo de experiencias de enseñanza significativas, en especial cuando éstos impliquen prejuicios o etiquetamientos sociales, o cuando expresen la tendencia a mantener cómodas rutinas. En cualquier caso, intentarán abrirse a la experimentación y a la búsqueda, evitando que sus preferencias constituyan un verdadero límite para la enseñanza y el desarrollo de aprendizajes relevantes.

(Davini, 2008, p. 74)

De esta forma como explica el autor, el docente tiene una capacidad adecuada para enfrentarse al reto de enseñar con una didáctica adecuada a las circunstancias planteadas en cada entorno.

2.1.4. Métodos de enseñanza modernos

El desarrollo de sistemas educativos conjuntamente con la tecnología ha dado origen a nuevas formas de enseñanza para lograr una transferencia de conocimiento, enfrentándoles a desafíos de todo nivel prácticos. Estos nuevos métodos están orientados a la guía docente y no tanto a la instrucción del mismo, enfocados a la producción de alternativas de solución a un problema planteado, son métodos participativos y colaborativos.

2.1.4.1. Método de estudio de casos. (Davini, 2008).

El estudio por casos se refiere a situaciones de la vida real. Que es necesario profundizar y tomar alguna posición sobre la solución a plantear donde es un proceso de entender, interpretar e intervenir en la realidad y lo sistematiza para la enseñanza, el objetivo es generar posibles maneras de intervenir en la situación con el fin de mejorar, constituye un metodología de análisis-interpretación acción, de este modo llama la atención de los participantes, el esquema básico se lo demuestra en la figura 1 y su secuencia básica como se muestra en la tabla 1.

*Figura 1: Modelo de Estudio de casos
Referencia:(Davini, 2008, p. 120)*

Tabla 1: Secuencia básica del método por casos

Fases	Actividades
--------------	--------------------

Apertura y organización	Presentación de propósito del trabajo y presentación del caso con informaciones básicas y preguntas de orientación. Posible organización de grupos.
<hr/>	
Análisis	Estudio del caso, búsqueda de nuevas informaciones, consultas o testimonios. Intercambio y debate progresivo Organización de la información.
<hr/>	
Integración	Intercambio y comparación de resultados de diferentes grupos y puesta en común de posibles soluciones o formas de investigación.
<hr/>	
Síntesis	Desarrollo de conclusiones finales de la tarea y revisión del proceso seguido por alumnos para alcanzarlas. Análisis de nuevas preguntas y problemas con el contexto específico de los alumnos.

Referencia:(Davini, 2008, p. 122)

2.1.4.2.Método de solución de problemas. (Davini, 2008).

Los problemas generan dudas o cierta incertidumbre, donde es necesario analizar y comprenderlo para tomar una decisión y tratar de resolverlo o por lo menos reducir la incertidumbre que se genera. Las variedades de problemas son muy amplios con una complejidad diferente, al tratar de resolverlos se genera una capacidad crítica y la inventiva con sentido práctico, el razonamiento es aplicado a la realidad y genera nuevas inquietudes donde es necesario adquirir nuevas formas de adquirir información.

Los problemas pueden ser bien definidos o estructurados donde pueden resolverse utilizando un número limitado de pasos, en la enseñanza generalmente los parámetros están explicados en el enunciado; por otra parte, los problemas parcialmente estructurados necesitan de un análisis que puede dar como resultado varias soluciones y finalmente los problemas de débil estructuración donde es necesario utilizar diferentes criterios y son dependientes de muchas variables como su entorno, cultura, contexto, etc.

Todos los problemas generan preguntas y sus respuestas no son establecidas en su mayoría, implica un análisis para determinar la solución correcta, la tabla 2 demuestra la secuencia básica de este método.

Tabla 2. Secuencia básica para método de solución de problemas

Fases	Actividades
Apertura y organización	<p>Presentación de propósito del trabajo y problemas a tratar.</p> <p>Diálogo y desarrollo de preguntas entre el profesor y alumnos para seguir un procedimiento y encontrar las incógnitas.</p>
Análisis	Análisis de las cuestiones y dimensiones del problema, búsqueda de información y ensayo de soluciones alternativas
Integración	Intercambio y comparación de resultados de diferentes grupos y puesta en común las diferentes posibles soluciones y forma de investigación
Síntesis	Verificación y generación de nuevas preguntas con la revisión del proceso seguido.

Referencia:(Davini, 2008, p. 125)

2.1.4.3.Método de proyectos. (Davini, 2008).

Apunta a aprender mediante la elaboración de un plan o proyecto que tiene un plan de trabajo, un cronograma, un proceso y resultados; generalmente usado en la formación profesional de jóvenes y adultos. Las dificultades de este método es la rigidez que puede ser impuesto por las unidades académicas en relación a su campo de acción, el tiempo puede ser un limitante por estas mismas reglas de trabajo pero en sí representa un importante valor educativo y sobre todo genera conocimientos teóricos y prácticos favoreciendo al autoaprendizaje, en la tabla 3 se detalla la secuencia básica del método.

Tabla 3. Secuencia básica para método de proyectos

Fases	Actividades
Apertura	Proponer un proyecto de trabajo viable, considerando las necesidades de aprendizaje y las capacidades de los alumnos tanto en conocimiento, como materiales y de apoyo.
Organización	Transmitir y buscar informaciones, datos y conocimientos. Elaborar un plan de trabajo por etapas Distribuir las tareas Organizar los grupos de trabajo de acuerdo con intereses y capacidades, en forma cooperativa
Puesta en marcha y monitoreo	Desarrollar actividades previstas en el plan por los distintos grupos. Apoyar y monitorear el desarrollo de actividades, ayudando en dificultades. Apoyar a la búsqueda de conocimientos y transmitir otros a lo largo de las actividades.
Difusión y socialización	Incentivar la comunicación del proyecto, Organizar muestras del producto.

Integración y evaluaciónEvaluar el conjunto el proceso seguido y sus logros de aprendizaje

Referencia:(Davini, 2008, p. 131)

Existen dudas de que si este método da resultados o no, cuando se habla de un aprendizaje en un marco de innovación, es importante comprender que existen dos procesos para aprender, el primero es aprender estudiando, y el segundo proceso es aplicar creativamente lo aprendido en la solución de problemas, esto da lugar a la expansión del uso de capacidades. (Judith Sutz, Héctor del Castillo, Martínez, 2011)

Cuando una institución educativa implementa propuestas y proyectos innovadores debe estar preparada para equivocarse, mejorar y volver a equivocarse, es un ciclo de aprendizaje en donde el mayor valor no es el resultado final sino el desarrollo de competencias asociadas a la práctica de nuevas formas de enfrentar problemas que de forma tradicional no se podrían resolver, dicho de esta forma este método no proporciona perdidas, lo que hace que sea viable. El paso del tiempo se asocia a la adquisición de nuevas habilidades, si bien se dice que el tiempo no determina que el aprendizaje haya mejorado pero es una variable que sirve para que en un rango determinado se logren resultados propuestos y luego estos puedan ser evaluados como por ejemplo: la adaptación a esta nueva forma de desarrollar proyectos, calidad de software y conclusión de los proyectos con éxito. (Judith Sutz, Héctor del Castillo, Martínez, 2011)

2.1.4.4. Modelos tutoriales (Davini, 2008).

Es un modelo que apoya al razonamiento durante y después de la acción donde el docente debe intervenir en el aprendizaje del estudiante mediante ejemplos de soluciones

desarrolladas por el profesor, de esa manera crea un mejor estímulo al estudiante con una retroalimentación de conocimientos de una forma personalizada.

2.1.4.5. Simulación (Davini, 2008).

Propone un método de enseñanza que acerca al estudiante a casos reales pero en un entorno virtual, es una opción para una gran variedad de contenidos de enseñanza principalmente en campos de la tecnología, el caso particular el uso de simuladores ayuda al uso de instrumentos para una toma de decisión, en la tabla 4 se puede observar su secuencia básica.

Tabla 4. Secuencia básica de método de simulación

Fases	Actividades
Apertura	Presentación de los propósitos de trabajo. Explicación conjunta de principios y conocimientos que sostienen la simulación, de acuerdo con las capacidades de los alumnos.
Puesta en marcha	Organización del ambiente de aprendizaje y de los materiales, recursos y herramientas. Distribución de tareas y organización de pequeños grupos, dialogar e integrar para verificar su comprensión, o solucionar dificultades
Desarrollo en la acción	Desarrollo de la simulación. Apoyar, dar pistas y corregir. Solucionar dudas y ampliar información, brindando nuevos ejemplos prácticos. Retroalimentar en forma permanente.
Integración	Análisis del proceso y reflexión sobre las habilidades puestas en práctica. Analizar nuevos ejemplos de la vida real relacionados con la experiencia realizada.

Referencia: (Davini, 2008, p. 150)

2.1.5. Programación de la enseñanza. (Davini, 2008).

La programación del estudio es algo fundamental, generalmente llamado un currículo, el hecho de programar indica la secuencia y la figura 2.

- Clarificar los propósitos educativos y definir los objetivos del aprendizaje específico.
- Organizar los contenidos.
- Diseñar la estrategia particular de enseñanza, apoyándose o integrando aportes de distintos métodos, pertinentes para los objetivos planteados.
- Diseñar una secuencia de actividades de aprendizaje y su distribución en el tiempo, incluyendo las formas de seguimiento y evaluación.
- Organizar el ambiente, seleccionando los materiales y los recursos.

(Davini, 2008, pp. 170,171)



Figura 2. Secuencia de programación de enseñanza.
Referencia:(Davini, 2008, p. 171)

2.1.6. Definición de objetivos de aprendizaje. (Davini, 2008).

Los propósitos expresan las finalidades de estudio, los objetivos expresan lo esperado por el alumno, lo que se desea que alcance a conocer o desarrolle su capacidad, los objetivos indican a dónde y cómo se quiere llegar. Los objetivos claros con una simple secuencia eficiente ayudan a los docentes a llevar un plan de clase y una buena programación de enseñanza con todos sus factores, en la tabla 5 se indican los diferentes objetivos que se pueden plantear.

Tabla 5. Tipos de objetivos

Tipos	Significado y proceso
Objetivos terminales o de productos	Definen los resultados de aprendizaje al completar una secuencia de enseñanza. Atendiendo a las recomendaciones anteriores, deberán privilegiarse resultados significativos y relevantes, más que largas listas de aprendizaje atomizado o puntual. Estos objetivos refieren integración de conocimientos o la elaboración de un producto de trabajo.
Objetivos de proceso	Plantean un proceso abierto, sin indicar un resultado único, valorando la diversidad de alternativas por alumnos, la riqueza de los procedimientos seguidos, las interacciones, las formas de expresión, la iniciativa, la inventiva, etc.
Objetivos expresivos	Enfatizan la expresión libre del alumno alrededor de actividades, situaciones o experiencias propuestas en la enseñanza, sin definir las condiciones o características que deber tener, son particularmente valiosos para el

2.2.ELECTRÓNICA

2.2.1. Fundamentos

2.2.1.1.Voltaje.

La tensión eléctrica o diferencial de potencial (llamado también voltaje) es una magnitud física que mide el trabajo por unidad de carga ejercido por el campo eléctrico sobre una partícula cargada al moverla entre dos terminales de diferente valor. (Cogdell, 2000).

2.2.1.2.Corriente

La intensidad eléctrica es el flujo de carga por unidad de tiempo que recorre un material, se debe al movimiento de electrones dentro de un material.(Cogdell, 2000).

2.2.1.3.Conductores

Son materiales cuya oposición al paso de electrones es muy baja, son metales como el cobre, el hierro, el aluminio, el oro y la plata, existen materiales no metálicos como el grafito; tienen de 1 a 3 electrones en su última órbita. (Cogdell, 2000)

2.2.1.4.Semiconductores

Es un elemento que se comporta como un conductor o un aislante² dependiendo de diversos factores, uno de ellos es el campo eléctrico y/o magnético, la presión, la radiación que le incide, o la temperatura del ambiente en el que se encuentre; estos elementos tienen impurezas que son átomos de otro elemento, pueden ser dopados para evitar esas impurezas y ser más o menos conductores, tienen 4 electrones de su última órbita.(Boylestad, 1997)

2.2.2. Elementos pasivos

2.2.2.1.Resistores

Se define como la oposición de corriente de un material, se fabrican de materiales conductores de electricidad pero que poseen una resistencia mayor al resto de componentes en el circuito.

El uso de resistores es amplio y existen de muchos tipos en la tabla 6 se muestran los tipos existentes con sus demás características.

Tabla 6. Tipos de resistores

Tipo de resistor	Rango de resistencia	Tolerancia [%]	Coefficiente [%/C]	Potencia máxima
Carbón	1Ω a 22MΩ	5 a 20	0,1	Hasta 2W

² Aislante: Material no conductor de cargas.

Alambre	1 Ω a 100k Ω	0,0005 y mayor	0,0005	Hasta 200W
Película de metal	0,1 Ω a 10 ¹⁰ Ω	0,005 y mayor	0,0001	Hasta 1W
Película de carbón	10 Ω a 100M Ω	0,5 y mayor	-0,015 a 0,05	Hasta 2W
Acero	0,1 Ω a 1 Ω	20	-	Hasta 250kW

Referencia:(Rodriguez, 2001, p. 7)

Las resistencias de carbón son las más comunes por su forma como la figura 3 y por los varios valores que pueden tomar como se muestra en la figura 4, los resistores de carbón para poder observar su valor óhmico poseen una película de metal como la figura 5.

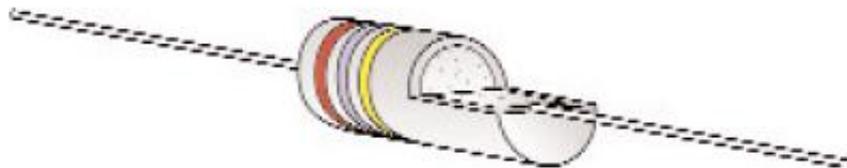


Figura 3. Resistencia de carbón
Referencia:(Rodriguez, 2001, p. 7)

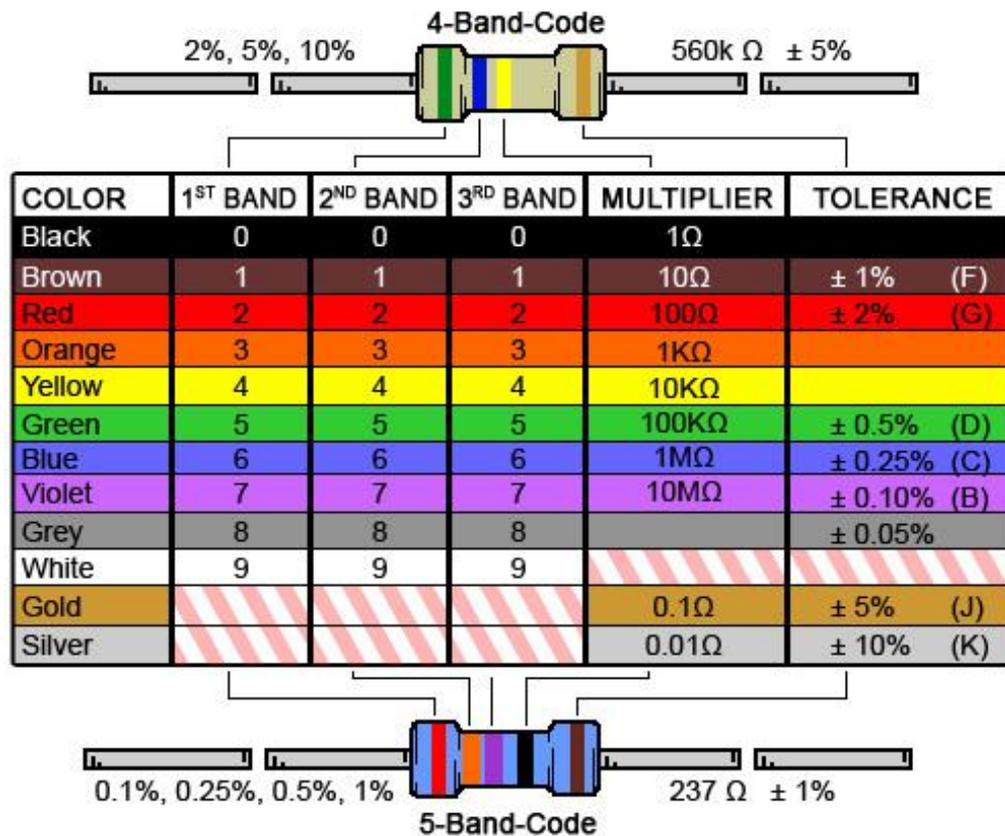


Figura 4. Código de colores de resistencias
Referencia: (Electronics, 2015)

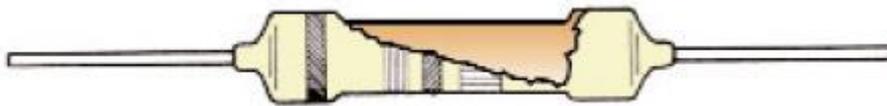


Figura 5. Resistencia con película de carbón
Referencia: (Rodríguez, 2001, p. 8)

2.2.2.2. Resistores variables

Los resistores variables permiten cambiar su valor óhmico entre diferentes rangos establecidos por el mismo material, para cambiar este valor generalmente poseen una perilla para poder variar su valor, los materiales generalmente de los que son fabricados es el alambre y los de carbón. (Rodríguez, 2001)

Los resistores variables de alambre son los más utilizados, sus valores resistivos van desde $2,2 \Omega$ hasta los $50 \text{ k}\Omega$, con potencias desde los 2 W hasta los 200 W , su ángulo de giro es de 300 grados como se muestra en la figura 6. (Rodríguez, 2001)

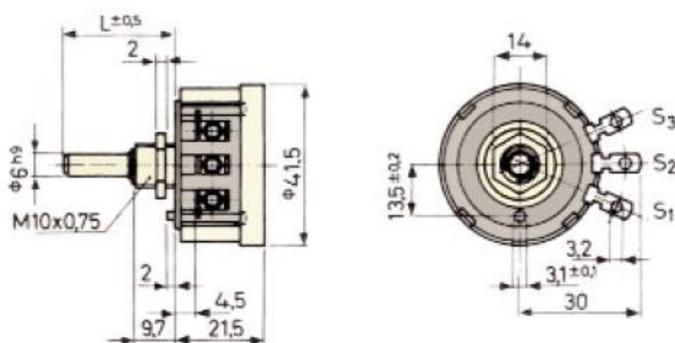


Figura 6. Esquema de un resistor variable
Referencia:(Rodríguez, 2001, p. 11)

2.2.2.3.Capacitores

El capacitor o condensador eléctrico es un dispositivo capaz de poder almacenar una carga eléctrica. Su estructura básica consiste en dos placas metálicas separadas entre sí y aisladas una de la otra por un material aislante llamado dieléctrico.

Los cuerpos materiales que poseen cargas eléctricas opuestas se atraen entre sí por una fuerza cuya intensidad se calcula mediante la Ley de Coulomb³, esta fuerza dependerá de la intensidad del campo eléctrico \mathbf{q} y del voltaje \mathbf{v} entre esos cuerpos, observando que para cada configuración particular de dos cuerpos cargados (con los mismos cuerpos y con una separación entre ellos constante), la relación existente entre carga y voltaje resulta constante.(Rodríguez, 2001, p. 20).

Si se construye un componente de forma tal que posea en forma deliberada un valor determinado de capacitancia, a ese elemento se lo llama capacitor o condensador. Su unidad de medida es el Faradio (F). El Coulomb resulta una cantidad muy grande de carga y la cantidad de carga almacenada en la mayoría de los capacitores reales es mucho más pequeña que ella. Esto hace que el Faradio como unidad de medida sea demasiado elevada para describir los valores de capacidad. Por ello, es común ver la capacidad expresada en fracciones del Faradio. (Rodriguez, 2001, p. 21).

En la figura 7 se puede observar los diferentes tipos de condensadores.



*Figura 7. Tipos de condensadores
Referencia: (Cogdell, 2000, p. 25)*

2.2.2.4. Inductores

La inductancia es la propiedad que poseen determinados dispositivos, llamados inductores, los cuales reaccionan ante cualquier variación en la corriente que los atraviesa. Los inductores son componentes diseñados para ser empleados en determinadas aplicaciones con el propósito de oponerse a cambios bruscos en la corriente que los atraviesan (principio de Inducción) y así efectuar una función de control.

Un inductor primario consiste en una o varias espiras de alambre conductor (generalmente de cobre o de aluminio, bobinado sobre un núcleo magnético o no), las cuales son atravesadas por una corriente. Michael Faraday (1791-1867) investigó los campos magnéticos y los concibió como líneas de fuerza partiendo de un polo magnético y retornando hacia el otro.

La cantidad total de líneas de fuerza generadas por el magneto definen su flujo magnético. La cantidad de flujo por unidad de superficie se denomina densidad de flujo B . (Rodríguez, 2001, p. 34)

La figura 8 muestra el flujo de corriente continua con el campo magnético.

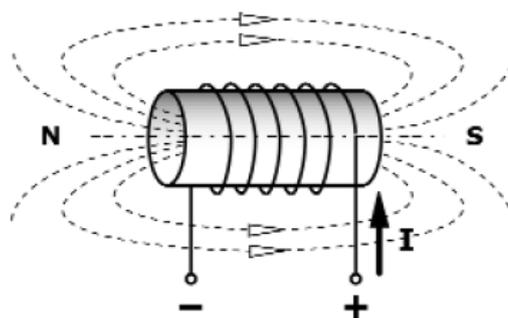
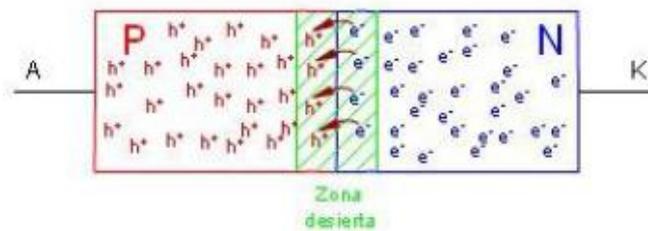


Figura 8. Efecto de una inductancia en corriente continúa
Referencia: (Cogdell, 2000, p. 34)

2.2.2.5. Diodos

Al unir dos materiales (P-N) se produce una recombinación de electrones (e^-) y huecos (h^+), un diodo tiene estos dos materiales separados por una zona desierta, cuando existen un voltaje que influya en un diodo, existe un traspaso de electrones de un material a otro haciendo la zona desierta cada vez más pequeña permitiendo el paso de corriente como se muestra en la figura 9. (Boylestad, 1997).



*Figura 9. Funcionamiento del Diodo
Referencia: (Boylestad, 1997, p. 85)*

2.2.3. Sensores

Los sensores son dispositivos que se encargan de capturar y convertir un tipo de energía física, química o biológica a otro tipo de energía que por lo general es eléctrica. Estos dispositivos posibilitan la comunicación entre el mundo físico y los sistemas de medición o de control, tanto eléctricos como electrónicos, utilizándose extensivamente en todo tipo de procesos industriales y no industriales para propósitos de monitoreo, medición, control y procesamiento. (López, 2006)

En la actualidad los sensores son muy usados en las aulas para las prácticas en la enseñanza de electrónica digital y analógica. (López, 2006)

2.2.3.1. Sensores analógicos

Los sensores analógicos son aquellos que dan como salida un valor de tensión o corriente variable en forma continua en el tiempo dentro del campo de medida. Normalmente en electrónica este tipo de sensores proporcionan salidas analógicas entre 0 y 5 voltios. (Cabrera, 2014)

Ejemplos de Sensores Analógicos:

- Sensor de temperatura
- Sensor de presión
- Sensor de caudal
- Sensor de proximidad
- Sensor de velocidad
- Sensor de aceleración
- Sensor de fuerza

2.2.3.2.Sensores digitales

Los sensores digitales son aquellos que dan como resultado una salida codificada en forma de pulsos o en forma de una palabra digital codificada en binario, BCD⁴ u otro sistema cualquiera. Es decir de un sensor digital únicamente se puede obtener dos valores lógicos 1 o 0, estos sensores se basan en un umbral o límite para arrojar cualquiera de los dos resultados dependiendo de la función y configuración del sensor. En electrónica generalmente se conoce a 5 voltios como un 1 lógico y a 0 voltios como un 0 lógico. (López, 2006).

Ejemplos de Sensores Digitales:

- Sensor de presencia
- Sensor contador de pulsos
- Sensores codificadores de posición (Encoders).
- Sensor Hall

⁴BCD= Decimal codificado en binario

- Sensor de nivel

2.2.3.3.Sensores Ultrasónicos de Distancia

Los sensores ultrasónicos o de ultrasonido como también son conocidos son detectores de proximidad que perciben objetos a distancias de hasta 8 metros sin necesidad de un contacto físico directo. El sensor emite una señal de sonido que al llegar a un objeto se refleja y regresa hacia el sensor, el tiempo que transcurre desde el momento en que se emite la señal hasta que se recibe la señal reflejada permite obtener la distancia a la que está situado el objeto detectado. (Cabrera, 2014)

Este tipo de sensor puede detectar cualquier tipo de objeto siempre y cuando este tenga un índice mínimo de reflexión que permita obtener una señal acústica reflejada hacia el sensor. Los materiales de los objetos pueden ser sólidos, líquidos o polvorientos de cualquier forma, color o material.(Cabrera, 2014)

El Sensor HC-SR04 es un módulo compatible con la plataforma Arduino y la mayoría de los microcontroladores. HC-SR04 es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 2 a 450 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su

bajo consumo, gran precisión y bajo precio por lo que está reemplazando a los sensores polaroid en los robots más recientes. Su forma se muestra en figura 10.(Cabrera, 2014)



Figura 10. Sensor HC-SR04
Referencia: (Micropik, 2014, p. 1)

El sensor HC-SR04 consta de un emisor de sonido de alta frecuencia y un receptor tipo micrófono. Para iniciar la medición se envía un pulso de 5 voltios durante 10 microsegundos en la entrada del pin TRIG del módulo. Esto hace que el módulo inicie la transmisión de 8 pulsos de un sonido a 40Khz a través del emisor, el pin ECHO se coloca en 5 voltios y mantiene este estado por un tiempo proporcional a la distancia. La distancia puede calcularse en metros o en centímetros con las siguientes fórmulas:

$$\text{distancia[metros]} = (\text{tiempo ECHO [segundo]}) * \frac{\text{velocidad del sonido } \left[\frac{m}{s}\right]}{2}$$

Ecuación 1. Cálculo de distancia en metros.
Referencia: (Cabrera, 2014)

Las características más importantes se muestran en la tabla 7.

Tabla 7. Características módulo HC-SR04

Característica	Detalle
Voltaje de trabajo	5 Vdc
Consumo de corriente	15 mA

Frecuencia de trabajo	40Hz
Distancia mínima	4 metros
Distancia máxima	2 centímetros
Ángulo de medición	15 grados
Señal TRIG	Pulso de 10 microsegundos
Dimensión	45*20*15 cm

Referencia: (Cabrera, 2014)

2.2.3.4. Sensor de temperatura y humedad DTH 11

La serie DHT11 cuenta con un sensor de temperatura y humedad complejo con una salida de señal digital calibrada. Mediante el uso de la exclusiva señal de adquisición digital técnica de la temperatura y la tecnología de detección de humedad, que garantiza una alta fiabilidad y excelente estabilidad a largo plazo. Este sensor incluye una medición de la humedad de tipo resistivo con un componente de medición de temperatura con un alto rendimiento.

Su información es de 8 bits, que ofrece una excelente calidad, respuesta rápida, interferencias de baja capacidad y gran rentabilidad. Sus características principales se muestran en la tabla 8.

Tabla 8. Características sensor DTH11

Característica	Detalle
Humedad	1%RH
Temperatura	0-50C
Rango de medidas	0-25 C Y 25-50 C
Respuesta en segundos	6 s y 10 s

Resolución	8 bits
Tiempo de duración	1 año

Referencia: (Micropik, 2014, p. 10)

Cada elemento DHT11 se calibra estrictamente en el laboratorio que es extremadamente precisa. Los coeficientes de calibración se almacenan como programas en la memoria OTP⁵, que son utilizados por el proceso de detección de la señal interna del sensor. La interfaz en serie de un solo cable hace que la integración del sistema sea rápida y fácil. Su pequeño tamaño, bajo consumo de energía y la transmisión de datos es la mejor opción para diversas aplicaciones, incluidos para sistemas exigentes. El componente es de 4 pines de una sola fila. Su forma y su diagrama de conexión como se muestra en la figura 11 y 12.(Micropik, 2014)

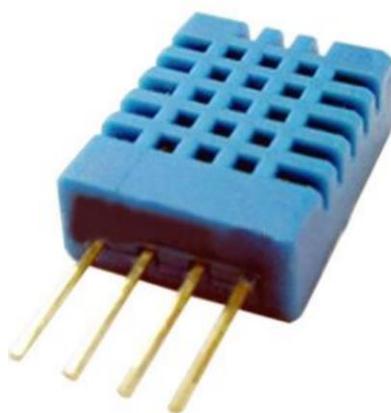
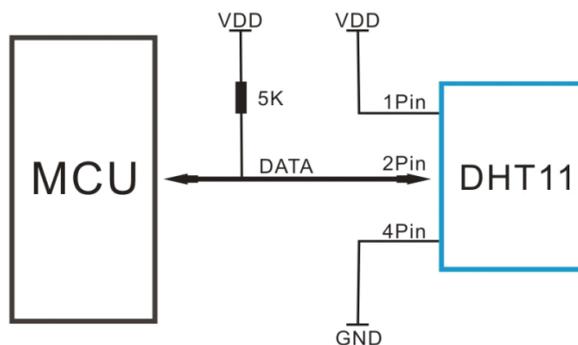


Figura 11. Sensor DHT 11
Referencia: (Micropik, 2014, p. 10)

⁵OTP: Memoria programable solo una vez.



*Figura 12. Diagrama de conexión
Referencia: (Micropik, 2014, p. 10)*

2.2.3.5.Sensor Pir

El sensor PIR permite detectar el movimiento, casi siempre se utiliza para detectar si un ser humano se ha movido dentro o fuera de una zona establecida. Son pequeños, de bajo costo, bajo consumo de energía, fáciles de usar y no se desgastan, por esa razón, se encuentran comúnmente en los electrodomésticos y aparatos utilizados en los hogares o negocios.

Ellos se refieren a menudo como PIR, "PassiveInfrared". PIR se compone básicamente de un sensor piroeléctrico (metal redondo con un cristal rectangular del centro), que puede detectar niveles de radiación infrarroja. Todo emite algo de radiación de bajo nivel, y ese algo más caliente, más radiación es emitida. Su diagrama de conexión y su forma se indican en las figuras 13 y 14 junto con la tabla 9 de características importantes.(Micropik, 2014)

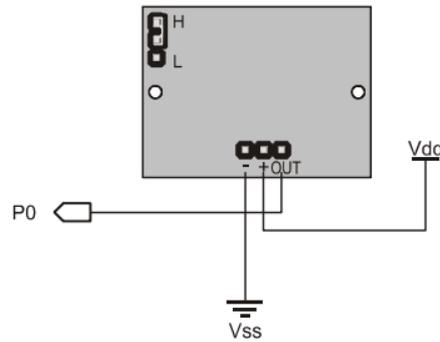


Figura 13. Diagrama de conexión PIR
Referencia: (Micropik, 2014, p. 10)

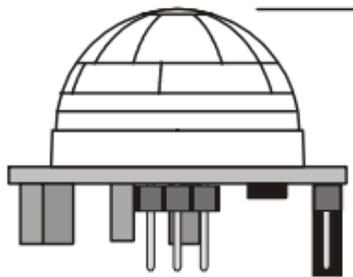


Figura 14. Sensor PIR
Referencia: (Micropik, 2014, p. 10)

2.2.4. Actuadores y visualizadores

Los actuadores en sistemas electrónicos son los elementos que generalmente ejercen de interfaces para convertir magnitudes físicas normalmente de carácter eléctrico a otro tipo de magnitud que permita actuar sobre el medio o proceso a controlar, por ejemplo un motor convierte los pulsos eléctricos en movimiento, un LED convierte los pulsos eléctricos en luz y así se puede tener otros ejemplos; en otras palabras los actuadores permiten interactuar o ejercer acciones ante las señales eléctricas. (López, 2006).

Ejemplos de actuadores y visualizadores

- LCD

- Display de 8 segmentos
- Chicharra/buzer
- Leds
- Relé
- Motor

2.3.SISTEMAS EMBEBIDOS

2.3.1. Introducción

Desde hace mucho tiempo se ha manejado un concepto sobre lo que es un sistema embebido, el cual en la actualidad aún se utiliza ya que no se desapega de su objetivo general que es cumplir con una o varias funciones específicas, su diseño de hardware no puede ser modificado pero su software si, pueden haber sistemas embebidos que requieran diseños muy sencillos hasta otros un poco más complejos, lo importante es tener claro en fin con el que este se crea para que el sistema no sea inexacto.

El avance de la tecnología ha hecho dispositivos muy potentes usando cada vez mayor cantidad de transistores, esto permite que su capacidad de procesamiento sea de mejor forma, como se muestran en la figura 15 el incremento exponencial de transistores en función de los años.(Chiesa & Bassi, 2010)

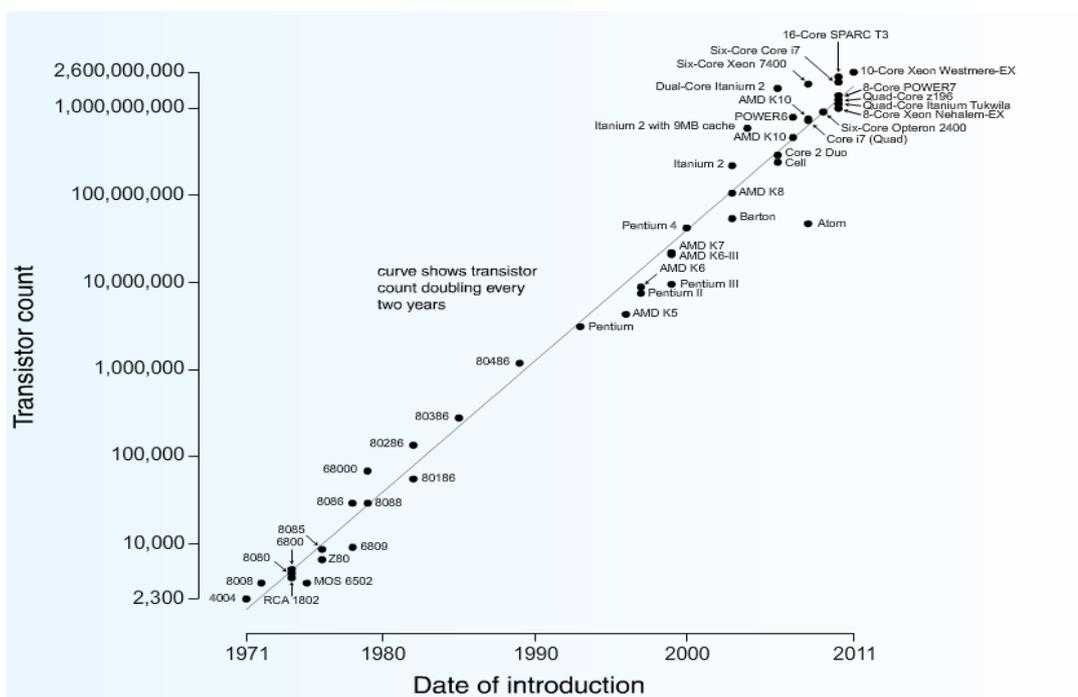


Figura 15. Uso de transistores en CI
Referencia: (CISCO, 2014)

Este aumento de transistores ha cambiado las arquitecturas de los procesadores que se encuentran dentro de un sistema embebido, existen una gran variedad de procesadores en relación a las diferentes aplicaciones que se pueden desarrollar.

Un sistema embebido generalmente todo su procesamiento lo reduce a un solo un chip sin la necesidad de añadir muchos elementos externos al núcleo. Se obtendrá con su uso, una central de proceso CPU, así como pequeñas cantidades de RAM y ROM internas. También ofrece la gestión de diferentes dispositivos externos, como buses de comunicación (UART, SPI, I2C) e incluso salidas programables (GPIO) que simplifican en gran medida el diseño y programación del sistema. Siempre a cambio de cierta lentitud computacional. (Fernández, 2009).

Un sistema embebido posee 4 partes importantes: hardware, firmware, sistema operativo y aplicaciones.

Hardware

Correspondiente al nivel más bajo, sobre el cual son controlados físicamente los dispositivos, son los elementos que se relacionan mediante pulsos eléctricos y son los que captan las señales del medio ambiente o genera señales que las personas puedan percibir.(Fernández, 2009)

2.3.2. Firmware

Es la parte de software, contiene los datos del programa para inicializar el hardware, es un programa específico desarrollado por cada empresa que desarrolle su propio hardware, usa un software de control (bootloader) para cargar las instrucciones desde una computadora con un lenguaje de programación y los inserta en la CPU del sistema para que pueda actuar de la forma programada.(Fernández, 2009)

2.3.3. Sistema operativo

Solo para sistemas embebidos avanzados, controla la utilización de la memoria y demás periféricos, son usados para generar servicios especializados con un conjunto de herramientas de software.(Fernández, 2009)

2.3.4. Aplicación

Es el que añade la funcionalidad a todas las partes antes mencionadas, es la más cercana al usuario, por ende, se encarga de la visualización de datos que transitan por el sistema embebido.(Fernández, 2009).

2.3.5. Diseño del sistema (Vahid & Tony, 1999)

Para considerar un buen desarrollo de sistemas embebidos las métricas incluyen:

- Costo: Costo de la realización del sistema y sus licencias.
- Costo de Ingeniería: Es el aporte tecnológico y el tiempo que le toma al desarrollador realizar el proyecto adecuadamente.
- Tamaño: El tamaño físico del sistema, el tamaño de datos que almacena.
- Rendimiento: Tiempo de procesamiento del software y su reacción a eventos.
- Poder: El consumo eléctrico del sistema, con su tiempo de vida de acuerdo a las hojas de datos de los procesadores principalmente.
- Flexibilidad: La facilidad de cambiar el funcionamiento en función de software.
- Tiempo del mercado: El tiempo para realizar el número de piezas que el usuario necesita.
- Tiempo de prototipado: El tiempo para construir la primera versión del sistema para verificar su funcionamiento.
- Correcciones: Después del prototipado el costo de corrección de errores.
- Seguridad: La probabilidad que el sistema no cause problemas de daños a los usuarios.
- Otros.

Todas estas métricas determinan un correcto diseño de un sistema embebido, el tiempo de mercado es uno de los más usados hoy en día por la rápida creación de nuevas herramientas y elementos electrónicos como sensores, el tiempo del mercado y su propósito ahora son muy cambiantes y restringen de gran manera el desarrollo de sistemas que puedan no estar a la par de la tecnología existente.

La correcta elección de la tecnología del procesador es una parte muy importante, ya que no solo se elige el procesador, también es su software de compilación y sus diferentes librerías de uso, cada uno tiene un diferente especialización con particulares aplicaciones, este concepto se muestra en la figura 16, donde indica los tipos de procesadores según su funcionalidad que son:

- Función deseada (a).
- Propósitos generales (b).
- Aplicaciones específicas (c).
- De una solo función (d).

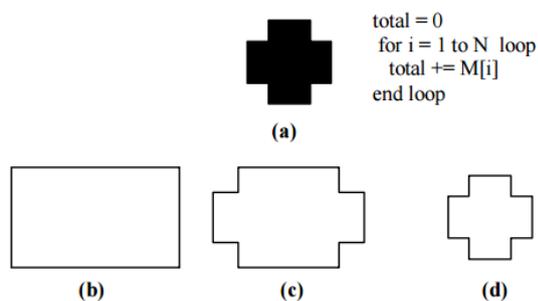


Figura 16. Tipos de procesadores
Referencia: (Vahid & Tony, 1999, p. 10)

2.3.5.1. Diagrama de bloques

Los diagramas de bloques es la representación de las conexiones físicas que tiene un sistema, no tiene parámetros establecidos de realización, todo depende del sistema que se quiera representar, pero debe tener entradas y salidas para demostrar el proceso del sistema. (Bolton)

2.3.5.2. Diagrama de flujo

Es la representación del código a realizar, los diagramas de flujo permiten al desarrollador poder socializar los programas para las debidas correcciones, es el nexo entre el software y el usuario. Las formas para la representación del diagrama de flujo se muestran en la tabla 9. (Cairo, 2005).

Tabla 9. Formas para diagramas de flujo

Figura	Nombre	Característica
	<i>Inicio</i>	<i>Inicio o fin de programa</i>
	<i>Proceso</i>	<i>Pasos, procesos o líneas de instrucción de programa de computo</i>
	<i>Entrada</i>	<i>Operaciones de entrada y salida</i>
	<i>Decisión</i>	<i>Toma decisiones y ramificación</i>
	<i>Conector</i>	<i>Conector para unir el flujo a otra parte del diagrama</i>
	<i>Base de datos</i>	<i>Almacenamiento de información</i>
	<i>Líneas de flujo</i>	<i>Sentido de dirección</i>
	<i>Salida</i>	<i>Salida de datos</i>
	<i>Documento</i>	<i>Refiere a un conjunto de documentos</i>

Referencia: (Cairo, 2005)

2.3.5.3. Estándar IEEE 830(IEEE, 1998)

Proporciona herramientas y formas para determinar los requisitos del sistema en el cual cuenta con un modelo para el desarrollo del mismo, en sí es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso, la ERS también contiene requisitos no funcionales (o complementarios).

Las características de una buena ERS son definidas por el estándar IEEE 830-1998. Una buena ERS debe ser:

- Completa. Todos los requerimientos deben estar reflejados en ella y todas las referencias deben estar definidas.
- Consistente. Debe ser coherente con los propios requerimientos y también con otros documentos de especificación.
- Inequívoca: La redacción debe ser clara de modo que no se pueda mal interpretar.
- Correcta: El software debe cumplir con los requisitos de la especificación.
- Trazable: Se refiere a la posibilidad de verificar la historia, ubicación o aplicación de un ítem a través de su identificación almacenada y documentada.
- Priorizable: Los requisitos deben poder organizarse jerárquicamente según su relevancia para el negocio y clasificándolos en esenciales, condicionales y opcionales.
- Modificable: Aunque todo requerimiento es modificable, se refiere a que debe ser fácilmente modificable.

- Verificable: Debe existir un método finito sin costo para poder probarlo.

2.3.5.4. Estándar IEEE 29148(IEEE, IEEE 29148-2011, 2011)

Esta norma reemplaza IEEE 830-1998, IEEE 1233-1998, IEEE 1362-1998. ISO / IEC / IEEE 29148: 2011 contiene disposiciones para los procesos y productos relacionados con la ingeniería de requisitos para los sistemas y productos de software y servicios en todo el ciclo de vida. Define la construcción de un buen requisito, ofrece atributos y características de los requisitos, y discute la aplicación iterativa y recursiva de los procesos de requisitos durante todo el ciclo de vida. ISO / IEC / IEEE 29148: 2011 proporciona orientación adicional en la aplicación de los procesos de requisitos de ingeniería y gestión de las actividades relacionadas con los requisitos de la norma ISO / IEC 12207 e ISO / IEC 15288. Elementos de información aplicables a la ingeniería de requisitos y su contenido están definidos. El contenido de la norma ISO / IEC / IEEE 29148: 2011 se puede añadir al conjunto existente de los procesos del ciclo de vida los requisitos relacionados definidos por la norma ISO / IEC 12207 o ISO / IEC 15288, o se puede utilizar de forma independiente.

Requisitos de ingeniería es una función interdisciplinaria que relaciona la necesidad del cliente con la solución del desarrollador para establecer y mantener los parámetros que debe cumplir por el sistema:

- Permite un entendimiento acordado entre las partes interesadas.
- Se valida con las necesidades del mundo real, se puede implementar
- Proporciona una base de diseños para verificar y aceptar soluciones
- Intenciones iniciales de las partes interesadas no sirven como requisitos de los interesados, ya que a menudo carecen de definición, análisis y, posiblemente la

consistencia y viabilidad, el análisis realizado nos permite tener toda la información necesaria transformarla en requisitos, por ende, es de gran importancia el correcto levantamiento de todos los stakeholder para no obviar detalles que retarden el progreso del proyecto.

Se debe formar los requerimientos en tres ámbitos que son los requerimientos de stakeholder⁶, requerimientos de sistema y requerimientos de arquitectura (hardware), esto contribuye a las necesidades del usuario y asegura que cada requerimiento los cubra, cada requerimiento debe ser necesario y útil donde defina la funcionalidad de sistema con una fácil implementación. Una de las herramientas que se puede utilizar dentro de este estándar es utilizar una entrevista que valide los requerimientos que se generan tanto desde el docente como el estudiante.

Un requerimiento cumple con lo siguiente:

- Puede ser verificado
- Debe ser cumplido
- Tiene condiciones medibles y con restricciones
- Define el rendimiento del sistema

Los requerimientos deben tener limitaciones que incluyen:

- Interfaces ya existentes
- Limitaciones de tamaño físico
- Leyes que interfieran al proyecto
- Duración o presupuesto disponible
- Plataformas ya existentes

⁶Stakeholder: Persona que influye directa o indirectamente en el Sistema.

- Usuarios y capacidades del sistema
- Existencia de materiales de trabajo
- Tiempo de entrega

2.3.6. Software de diseño

2.3.6.1.Eagle

Es un programa para el diseño de placas de una forma profesional, permite un desarrollo de una placa con grandes prestaciones de fiabilidad, es el software más completo y generalizado en el desarrollo de sistema electrónico, los cads de Intel Galileo Gen 2 se encuentran en los repositorios de Eagle con su libre distribución. (CadSoft, 2011).

Posee una gran cantidad de librerías, la mayoría de distribuidores y desarrolladores de sensores y placas de electrónica suben sus diseños de placas bajo Eagle para la réplica y mejora de los mismos y es multiplataforma. (CadSoft, 2011)

2.3.6.2.Arduino (Arduino, 2015)

Arduino se inició en el año 2005 como un proyecto para estudiantes en el Instituto IVREA, en Ivrea (Italia). En ese tiempo, los estudiantes usaban el microcontrolador BASICStamp, cuyo coste era de 100 dólares estadounidenses, lo que se consideraba demasiado costoso para ellos. Por aquella época, uno de los fundadores de Arduino, Massimo Banzi, daba clases en Ivrea.

El nombre del proyecto viene del nombre del Bar di Re Arduino (Bar del Rey Arduino) donde Massimo Banzi pasaba algunas horas. En su creación, contribuyó el estudiante colombiano Hernando Barragán, quien desarrolló la tarjeta electrónica Wiring, el lenguaje de programación y la plataforma de desarrollo.¹⁰ Una vez concluida dicha plataforma, los investigadores trabajaron para hacerlo más ligero, más económico y disponible para la comunidad de código abierto (hardware y código abierto). El instituto finalmente cerró sus puertas, así que los investigadores, entre ellos el español David Cuartielles, promovieron la idea. Banzi afirmaría años más tarde, que el proyecto nunca surgió como una idea de negocio, sino como una necesidad de subsistir ante el inminente cierre del Instituto de diseño Interactivo IVREA. Es decir, que, al crear un producto de hardware abierto, éste no podría ser embargado.

Posteriormente, Google colaboró en el desarrollo del Kit Android ADK (AccessoryDevelopment Kit), una placa Arduino capaz de comunicarse directamente con teléfonos móviles inteligentes bajo el sistema operativo Android para que el teléfono controle luces, motores y sensores conectados de Arduino.

Para la producción en serie de la primera versión se tomó en cuenta que el coste no fuera mayor de 30 euros, que fuera ensamblado en una placa de color azul, debía ser Plug and Play y que trabajara con todas las plataformas informáticas tales como MacOSX, Windows y GNU/Linux. Las primeras 300 unidades se las dieron a los alumnos del Instituto IVREA, con el fin de que las probaran y empezaran a diseñar sus primeros prototipos.

Es una plataforma de software y hardware libre que permite un desarrollo de sistemas electrónicos a bajo costo, posee un microcontrolador de marca AVR de diferentes distribuciones de pines para diferentes aplicaciones, con periféricos de:

- Salidas/entrada digitales.
- Entradas/ salidas analógicas
- Salidas de PWM⁷
- Comunicación serial
- Comunicación I2C.
- Pines de voltaje
- Interrupciones de software y hardware.

Es una de las herramientas con mayor cantidad de librerías gracias a la contribución de muchos desarrolladores que por su software libre permite crear y modificarlas en relación a la necesidad de la aplicación, sus librerías más usadas son:

- EEPROM
- Ethernet
- Firmata
- GSM
- LiquidCrystal
- SD
- Servo
- SPI
- SoftwareSerial
- TFT
- Wifi
- Wire

⁷PWM: Control de Ancho de Pulso

Arduino posee una sola distribución dedicada solo para Intel Galileo, Intel Galieo Gen 2 e Intel Edison por su diferente estructura y configuración de registros, este software es descargado de forma gratuita desde las páginas de Arduino o de la misma Intel.

2.3.6.3.Fritzing(Fritzing, 2015)

Fritzing es una iniciativa de hardware de código abierto que hace que la electrónica sea accesible como material creativo para cualquier persona. Ofrecen una herramienta de software, un sitio web de la comunidad y los servicios en el espíritu de Procesamiento y Arduino , fomentando un ecosistema creativo que permite a los usuarios documentar sus prototipos, comparten con los demás, enseñar electrónica en un salón de clases, y el diseño y fabricación de PCB profesionales.

2.3.6.4.Proteus(Labcenter, 5015)

El Programa ISIS, Intelligent Schematic Input System (Sistema de Enrutado de Esquemas Inteligente) permite diseñar el plano eléctrico del circuito que se desea realizar con componentes muy variados, desde simples resistencias, hasta alguno que otro microprocesador o microcontrolador, incluyendo fuentes de alimentación, generadores de señales y muchos otros componentes con prestaciones diferentes. Los diseños realizados en Isis pueden ser simulados en tiempo real, mediante el módulo VSM, asociado directamente con ISIS. ISIS posee una muy colección de librerías de modelos tanto para dibujar, simular o para placas. Además, permite la creación de nuevo

componentes, su modelización para su simulación e, incluso la posibilidad al fabricante de crear el modelo.

2.3.6.5.Dia(Developers, 2014)

Dia Diagram Editor es un software de dibujo de código abierto libre para Windows, Mac OS X y Linux. Dia soporta más de 30 tipos de diagramas diferentes, como diagramas de flujo, diagramas de red, modelos de bases de datos. Más de un millar de objetos readymade ayudan a dibujar diagramas profesionales. Dia puede leer y escribir un número de diferentes formatos de imagen vectorial y raster. Los desarrolladores de software y especialistas en bases de datos pueden utilizar Dia como una herramienta CASE para generar esqueletos de código de sus dibujos. Dia puede ser con guión y se extendió el uso de Python.

Dia es una aplicación para la creación de diagramas técnicos. Su interfaz y funciones se modelan libremente después del programa de Windows Visio. Características del Dia incluyen la impresión de varias páginas, la exportación a muchos formatos (EPS, SVG, CGM y PNG) y la capacidad de utilizar formas personalizadas creadas por el usuario descripciones XML como simples. Dia es útil para dibujar diagramas UML, mapas de la red, y diagramas de flujo.

2.4.INTEL GALILEO GEN 2

2.4.1. Introducción

La placa Intel® Galileo Gen 2 es la primera de una familia de placas de desarrollo y prototipos certificadas por Arduino* basadas en la arquitectura Intel®

y especialmente diseñadas para creadores, estudiantes, educadores y entusiastas de la electrónica tipo "Hágalo usted mismo".(INTEL, 2015).

Intel se compromete a proporcionar los últimos procesadores y herramientas para su comunidad. La primera iniciativa de Intel es la introducción de Intel e Intel Galileo Gen 2 que son compatibles con las cabeceras de Arduino y sus librerías actualizadas por miles de desarrolladores de la plataforma; Intel creó su propio IDE de Arduino de código y hardware abierto, es decir todo se encuentra online para la descarga gratuita, su propio IDE es por la robustez de sus microprocesador por una infraestructura x86, dando de esta forma un software completo y multiplataforma, al tener sus repositorios en la internet cualquier persona está en la capacidad de modificar las placas a su gusto o necesidad como lo ha realizado con Arduino, de esta forma la plataforma crece de una manera más rápida compartiendo conocimiento. (Makezine, 2015)

Walt Disney creó su primera película en 1923 en su garaje, Steve Jobs y Steve Wozniak desarrolló la primera computadora Apple en un garaje ubicado en California, Bill Hewlett y Dave Packard fundada HP con 500 dólares en un garaje, y la primera Harley-Davidson se construyó en un garaje de madera en 1903. Tal vez lasiguiente mayor invento se creará en el garaje de alguien.(Makezine, 2015)

Todas estas invenciones tienen un factor común, que es la convergencia de diferentes áreas, tales como el arte, la mecánica, la electrónica y la informática. Ellos participan creatividad y la pasión.(Makezine, 2015)

Los inventores son los responsables, y que han existido desde el principio de los tiempos, cuando los antepasados comenzaron a crear herramientas para pescar, cazar,

hacer fuegos, y para dibujar en las cuevas. Dependiendo de la naturaleza de su proyecto, usted puede conseguir el procesamiento de gran alcance y ahorrar mucho de dinero si utiliza las placas Arduino regulares basados sólo en microcontroladores; para tener una idea clara qué usted debe utilizar los tableros de Intel Galileo, suponga que usted necesita desarrollar un proyecto con los siguientes requisitos:

- Guardar la información en la tarjeta SD para el registro.
 - Conecte y transmitir los datos recogidos a través de Internet.
 - Los usuarios deben ser capaces de transmitir archivos de registro y los registros del monitor de archivos demanda. Así, un servidor web debe ser desarrollado.
 - Un periférico específico USB como una cámara web será utilizada y su
 - El acceso a Internet debe estar configurado a través de Ethernet o conexiones Wifi.
- Usted debe tener el tiempo y la fecha correcta de los datos está registrando en la tarjeta SD, incluso cuando sus reinicios de mesa y el sistema se restablecen, por lo que se necesita un reloj de tiempo real (RTC).

Si se piensa que estos requisitos son complejos, tenga en cuenta que son comunes requisitos cuando se desea crear una domótica, construir un robot que permite controlar de forma remota, construir un sistema de vigilancia, o monitorear el suelo del jardín, todos estos requerimientos aunque suenen complicados son factibles al utilizar la plataforma Intel Galileo.(Makezine, 2015).

Su fortaleza es la integración de un Linux embebido el cual permite tener una CPU⁸ capaz de levantar servicios de red, conectar diferentes tipos de redes y tecnologías y sobre todo desarrollar scripts en código de abierto para enlazar páginas web, servidores remotos etc. En la figura 16 se muestra la forma física de Intel Galileo Gen 2 desde una vista superior e inferior.

⁸CPU: Unidad Central de Procesamiento



*Figura 16. Vista frontal y posterior de la Placa Intel Galileo Gen 2
Referencia: (INTEL, 2015)*

2.4.2. Características importantes (INTEL, 2015)

- El procesador de aplicaciones Intel® Quark™ SoC X1000, una arquitectura de conjunto de instrucciones de procesador Intel® Pentium® de 32 bits, con un solo núcleo y un solo subproceso compatible con ISA, que funciona a velocidades de hasta 400 MHz.
- Compatibilidad con una amplia variedad de interfaces de E/S estándar en la industria, entre ellas la ranura mini-PCI Express de tamaño completo, el puerto Ethernet de 100 Mb, la ranura microSD, el host USB y el puerto cliente USB.
- DDR3 de 256 MB, SRAM de 512 kb integrada, Flash NOR de 8 MB y EEPROM de 8 kb estándar en la board, más compatibilidad con tarjeta microSD de hasta 32 GB.
- Compatibilidad de hardware y pines con una amplia variedad de shields Arduino Uno R3.

- Programable a través del entorno de desarrollo integrado (IDE) de software Arduino, que es compatible con los sistemas operativos host Microsoft Windows, Mac OS y Linux.
- Compatibilidad con la edición Yocto 1.4 Poky Linux.
- El cabezal TTL UART USB 3.3 V de 6 pines reemplaza al puerto de consola RS-232 con conector de 3,5 mm para depuración Linux. El nuevo conector de 6 pines puede usarse con el cable de serie USB FTDI estándar (TTL-232R-3V3) y las populares breakout boards de USB a serie. 12 GPIO totalmente nativos para mayor velocidad y una resistencia superior de la unidad.
- Modulación por amplitud de impulsos (PWM) de 12 bits para un control más preciso de servos y una respuesta más fluida.
- La consola UART1 se puede redireccionar a los cabezales Arduino en esquemas, eliminando la necesidad de soft-serial en muchos casos.
- Capacidad para alimentación sobre Ethernet (PoE) de 12 V (requiere instalación de módulo PoE).
- Se modificó el sistema de regulación de energía para admitir fuentes de alimentación de 7 V a 15 V.

Su diagrama de conexión general se muestra en la figura 17.

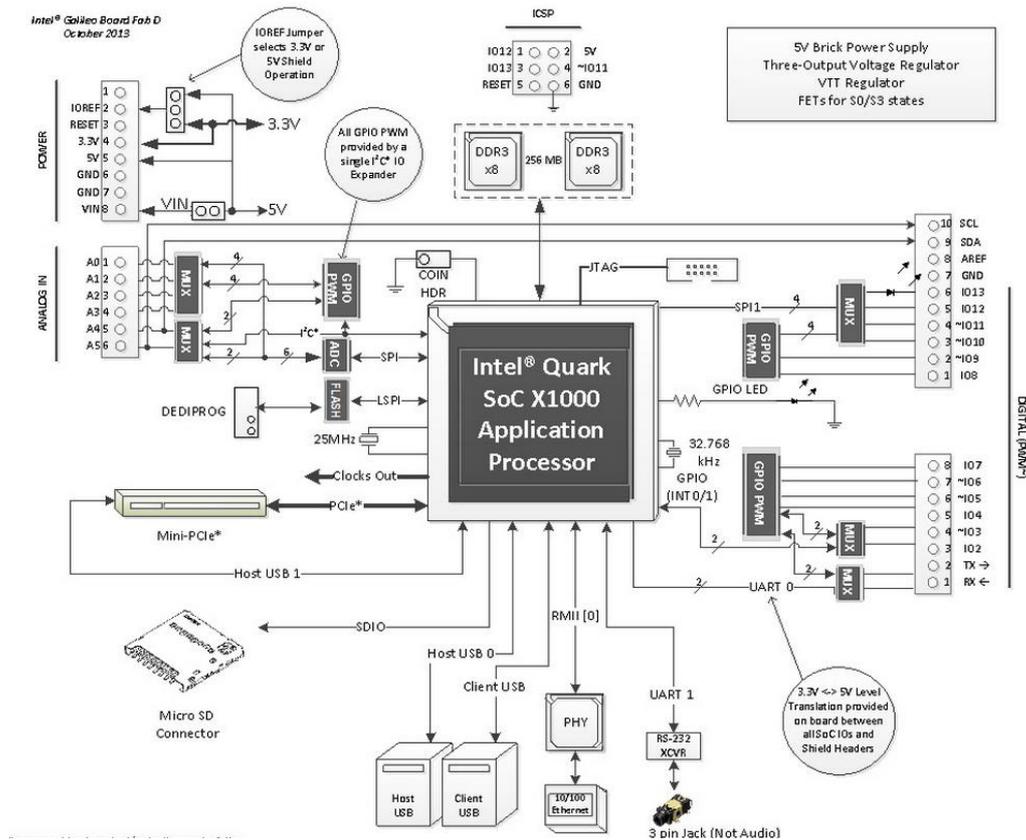


Figura 17. Diagrama de conexión Intel Galileo Gen 2
Referencia: (Ramon, 2014)

De la figura 17 se puede decir que:

- Intel Galileo Gen 2 ofrece un terminal con seis pines dedicados a la conexión del cable FTDI en el nivel TTL de 3.3V. Este terminal proporciona acceso a la consola de Linux.
- Al igual que la primera Intel Galileo, todos los pines que admiten PWM tienen una señal (~) Delante de sus etiquetas. Por ejemplo, el pasador IO3 tiene una señal en delante de su nombre, que significa que respalda PWM.

- Se utilizan los multiplexores (MUX) representados en el diagrama porque los pines pueden asumir diferentes funciones. Por ejemplo, el IO11 pin puede ser utilizado como entrada o salida digital o generar PWM, así el MUX cambia la conexión de este pin a los bloques de circuitos apropiados responsables de cada función.
- La cabecera IO es completamente diferente en comparación con la primera Intel Galileo. Ahora, la mayoría de los pines IO se conectan directamente a Intel Quark SOC con la excepción de IO7 y IO8, que están conectados al expansor PCA GPIO.
- Tenga en cuenta que los pines IO2 y IO3 ahora se puede utilizar como consolas de serie como
- Los puentes de VIN y la dirección I2C ya no existen.
- Hay dos conectores micro-USB que funcionan como cliente y el host interfaces de un conector JTAG de 10 clavijas, una ranura mini-PCIe, dos DDR3chips de memoria de 256 MB cada uno, y una interfaz SPI conectados directamente a Intel SoC Quark.
- La ranura para tarjetas SD utiliza un bus SDIO conectado a Intel y Quark apoya la tarjeta SD hasta 32GB.
- Existe una interfaz que le permite actualizar el firmware a través del protocolo de flash SPI usando una herramienta llamada SF100 DediProg.
- Aunque Intel Quark admite dos interfaces Ethernet sólo uno es expuesta en un conector Ethernet RJ45.(Ramon, 2014)

2.4.3. Intel Quark SoC x1000(INTEL, 2015)

Intel SoC Quark X1000 es un procesador de 32 bits diseñado para menor consumo de energía, con las instrucciones de código de operación esx86 compatibles pero implementa características como ACPI (Advanced Configuration and Power Interface) e incluye varias interfaces que proporcionar conexiones con periféricos externos. Intel Quark compete directamente con ARM.

Aunque el paquete sólo es 15x15mm, este pequeño procesador también ofrece una interfaz que le permite conectarse a varios periféricos, incluyendo los dispositivos Bluetooth, ZigBee, SD puertos host USB 2 y dispositivos / SDIO / eMMC tarjetas, dispositivos I2C y maneja GPIOs interrumpible o no por la configuración; soporta rangos de temperatura que alcanzan industrial, médica y aplicaciones militares; entre sus principales características se encuentran:

- Individual Quark núcleo de la CPU, solo hilo, 32 bits X86, a 400 MHz
- Controlador de memoria DDR3 de hasta 2GB en 800MTS y ECC-On-Chip
- Embedded RAM 512KB de caché y 16KB
- Compatible con los bloques de legado: PC compatible: puertos IO, PCI, ACPI
- Bajo costo de 10 pines JTAG
- Especificaciones de E / S
- Un 10 / 100MB MAC Ethernet
- Dos puertos host USB 2 (EHCI y OHCI)
- Dos controladores HS UART
- Un puerto SPI para periféricos
- Un puerto USB2 dispositivo HS
- / SDIO / interfaz de máster Erasmus Mundus Uno SD

- Controlador Uno I2C / GPIO
- GPIOs Dieciséis con interrupciones programables (borde)
- Soporte de Software
- Pentium ISA compatible (0.586 códigos de operación)
- compiladores estándar ICC / GCC / G ++
- El código abierto UEFI EDK II
- GRUB apoyo del gestor de arranque
- Soporte de depuración TOC
- Cumple con PCIe, USB, y las normas ACPI
- Ejecuta 3.8 kernel de Linux +
- Rangos de temperatura externa 40 grados centígrados
- sensor térmico programable Interna
- Seguridad
- Soporta la tecnología de arranque seguro
- Ejecución modo de Supervisión / protección
- Recuperación segura para UEFI FW

En la figura 17 se muestra su forma desde una vista superior



Figura 18. Quark SoC x1000
Referencia: (INTEL, 2015)

2.4.4. Desarrollo(INTEL, 2015)

Las placas de desarrollo de la Intel permite realizar un prototipado de sistemas electrónicos de una forma muy rápida y eficiente; este método permite a los desarrolladores de Google generar un solución o un concepto y poder trabajar con este de una forma real.

Intel tiene algunas plataformas aliadas para el adecuado desarrollo de sistemas electrónicos, entre ellos se encuentran las mismas páginas propietarias de la empresa, Fritzing para la simulación y creación de hardware electrónico, Wylodrhil para aplicaciones en la nube y monitoreo en tiempo real.

De la misma forma cuenta con simuladores en línea para la comprobación del programa albergado en la placa de desarrollo y poder corregir brevemente errores de funcionamiento.

2.4.5. Arduino IDE y otros lenguajes

Arduino es una plataforma electrónica de software y hardware libre, constituida por una placa con un microcontrolador y un entorno de desarrollo y diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. (Arduino, 2015)

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. (Arduino, 2015)

Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador, esto permite que se pueda combinar las diferentes

placas Arduino con otros sistemas o programas para mejorar los proyectos y sus utilidades.(Arduino, 2015)

2.4.6. IoT

Los primeros pasos de la Internet se recopilan desde hace 20 años, donde su principales usos eran las búsquedas, su acogida fue muy grande y dio paso para el comercio en línea, el cual permite realizar transacciones de bienes o servicios entre personas del todo mundo, finalmente las empresas vieron la gran importancia de conectarse de mejor manera con los clientes mediante redes sociales, esto permite una mejor interacción generando gran cantidad de datos e información que ya no son almacenados localmente, su almacenamiento se encuentra en la nube, este término se refiere la virtualización de servidores de grandes prestaciones para solo ocupar lo necesario por cada usuario, evitando la necesidad de tener uno propio.(CISCO, 2014).

Actualmente la comunicación entre personas no es suficiente, la automatización de procesos crean la necesidad de que las máquinas puedan comunicarse por sí solas hacia las personas para conocer su estado y actividades, esta comunicación permite un mejor desenvolvimiento de la máquina al conocer a tiempo real su funcionamiento, con este avance tecnológico es necesario tener mejores sistemas electrónicos que puedan comunicarse entre ellos y las personas de una forma local o remota; este proceso es considerado Internet de las Cosas con sus siglas en inglés IoT.(CISCO, 2014).

Para que IoT exista es necesario 4 pilares fundamentales: las personas, cosas, procesos y data; las personas son fundamentales ya que son las que reciben información, las analizan y las convierten en conocimiento para el mejor desarrollo de un bien o servicio, cuando este conocimiento es difundido y aplicado en diferentes ámbitos se

considera sabiduría. La sabiduría de las personas hace que la sociedad que hoy vivimos tenga soluciones a problemas existentes que por la carencia de información no pudieron ser resueltos.

Las cosas son las máquinas comunes y no comunes que se desean interconectar a la red, estos nuevos elementos de la red tienen un cerebro (controlador) para que interprete señales y pueda enviar respuestas o información de las mismas, para su programación se pueden usar diferentes lenguajes de programación y diferentes niveles.

Los procesos son las capacidades computacionales que poseen cada sistema para poder enviar datos de un lugar a otro, además como se analizan y se interpretan estos datos para tomar una decisión.

Datos son variables sin sentido que transitan por la red, los datos en sí no pueden ser útiles para ningún proceso o personas, cuando estos datos toman un sentido se transforman en información, la internet tiene millones de informaciones transcurriendo diariamente, la información cuando es procesada a diferentes niveles permite conocer una actividad con mayor exactitud y tomar decisiones correctas o predecir cambios en algún comportamiento, la información cuando toma este sentido se convierte en conocimiento; finalmente si este conocimiento es difundido a otras circunstancias es sabiduría. (CISCO, 2014).

IoT establece 3 niveles para su desarrollo M2M (machine to machine), se establece mediante la comunicación entre 2 máquinas sin la intervención del ser humano, poseen una configuración ya establecida y no necesitan de parámetros de accionamiento por humanos; M2P (machine to people) es el proceso donde una máquina recibe información de varias (M2M) y es entregada de una forma visual a las personas, la información ya es discriminada y se muestra solo lo solicitado por la persona; P2P es el paradigma de IoT, es

la interacción entre las personas a través de máquinas para su comunicación. Iot es funcional cuando los 3 niveles son desarrollados a cabalidad.(CISCO, 2014).

2.4.7. Linux Embebido

2.4.7.1.Introducción

Un sistema operativo es la interfaz que permite la interacción entre el software y hardware de una computadora, existen tres tipos de sistemas operativos, Windows, IOS y Linux, los dos primeros considerados de pago y el último gratuito bajo ciertas licencias libres.

Linux es un sistema operativo que da mayor flexibilidad en su instalación por sus bajos requerimientos del sistema y su elección de paquetes a instalar; con el avance de la electrónica y de sistemas embebidos es necesario generar servicios y controles para las redes que estos sistemas integran, por ende, el levantamiento de servicios de red son realizados en plataformas libres por su flexibilidad y gran número de desarrolladores.

La distribución de Linux embebido son sistemas operativos de pequeños tamaños que son instalados en sistemas electrónicos con un procesador de gama media o alta capaz de poseer dos ambientes de desarrollo, tanto el electrónico como el de servicios de red.

2.4.7.2.Yocto

El Proyecto yocto es un proyecto de colaboración de código abierto que proporciona plantillas, herramientas y métodos para ayudarle a crear sistemas personalizados basados en Linux para los productos integrados,

independientemente de la arquitectura de hardware. Fue fundada en 2010 como una colaboración entre muchos fabricantes de hardware, de código abierto proveedores de sistemas operativos y compañías de electrónica para poner orden en el caos de desarrollo de Linux embebido. Es un completo entorno integrado de desarrollo de Linux con herramientas, metadatos y documentación. Las herramientas libres son fáciles para empezar, de gran alcance para trabajar con (incluidos los entornos de emulación, depuradores, un kit de herramientas de generador de aplicaciones, etc.) y permitir que los proyectos que se llevarán adelante en el tiempo sin causar que usted pierda optimizaciones y las inversiones realizadas durante el fase de prototipo del proyecto. El Proyecto Yocto fomenta la adopción de la comunidad de esta tecnología de código abierto que permite a sus usuarios centrarse en sus funciones y el desarrollo de productos específicos. El Proyecto Yocto proporciona recursos e información que atienden a los usuarios nuevos y experimentados, e incluye recetas básicas de componentes del sistema que proporciona el proyecto OpenEmbedded. (YOCTO, 2013).

2.4.7.3. Ubuntu embebido

Embedded Linux es una opción de rápido crecimiento del sistema operativo en el mundo de los dispositivos móviles. Por desgracia, los marcos existentes para incrustar Linux son o compleja (por ejemplo, usar demasiados hacks a fingir un entorno nativo) o utilizar código propietario en el lado de destino. Esto trae una gran cantidad de restricciones para portar aplicaciones de escritorio para el mundo de mano y haciendo un sistema completamente gratuito operativo integrado. Embedded Ubuntu es una iniciativa para construir una comunidad en torno a la elaboración de un marco muy flexible y simple para generar derivados

de Ubuntu a medida para las arquitecturas de destino se encuentran en los dispositivos móviles. En el corazón del proyecto Ubuntu se encuentra la creencia de que el software de código abierto y la tecnología pueden desempeñar un papel clave para que las personas alcancen su potencial. Un objetivo central ha sido la creación de un mundo de clase, el sistema operativo libre y de código abierto que se ha trabajado para hacer accesible a través de cuadernos, escritorios, clientes ligeros y servidores.(Ubuntu, 2009).

2.4.7.4. Debían embebido

La disponibilidad de dispositivos sin almacenamiento ampliable se quedó pequeño para mantener la cantidad de trabajo necesario para preparar la distribución. Cualquier dispositivo capaz de utilizar una tarjeta SD tiene más que suficiente de almacenamiento para ejecutar una distribución estándar de Debian. Emdebian está basada en Debian y por lo tanto utiliza los paquetes de Debian. Algunas instalaciones Emdebian hicieron demandas más bajas en el hardware de la máquina, pero esto era así porque la instalación Emdebianpaquetes que ya están en Debian que estaban destinados para tales fines seleccionados. Estos paquetes se han eliminado la mayoría, debido a la falta de actividad aguas arriba y una falta de dispositivos adecuados. Un viejo PC con un disco duro bastante grande (o una capacidad de usar un disco duro de varios gigabytes moderna u otros medios de comunicación) no era probable que se beneficien de Emdebian. Repositorios completos de paquetes para diferentes arquitecturas basadas en coreutils y Perl. Soporte para herramientas estándar de Debian como debian-installer. (DEBIAN, 2014).

CAPÍTULO III

3. DISEÑO Y DESARROLLO DEL SISTEMA

3.1.DISEÑO

3.1.1. Análisis de tarjetas de entrenamiento

El diseño de las placas de entrenamiento se enfoca en las materias de Sistemas Microprocesados y Optativa II (sistemas embebidos), con el fin de optimizar el tiempo de la construcción del hardware de un sistema electrónico para enfocarse en la hora de clase al aprendizaje del desarrollo del sistema y su rápida retroalimentación en la depuración de errores con las pruebas en el hardware, el estudiante cuenta con mayor cantidad de tiempo para aplicar sus conocimientos en el desarrollo integral del proyecto (software, hardware y pruebas) y mejorar su aprendizaje.

El manejo de periféricos más usados como puertos digitales de entrada/salida, puertos análogos de entrada/salida, comunicación serial, control de ancho de banda, control de servo motores, interrupciones, teclado matricial y LCD, esta información será validada por el docente a cargo de la materia en el uso del hardware en horas de clase y observar sus beneficios y desventajas; las primeras placas cumplirán las funciones antes mencionadas y tendrán sus propio requisitos, las siguientes placas son para la unión de varias funcionalidades del procesador que permita generar monitoreo de una forma real para aplicaciones en la asignatura de Optativa II.

En la tabla 10 se presenta el análisis inicial bajo los estándares IEEE 839 e IEEE 29148, como se había explicado anteriormente en el capítulo II, este estándar brinda

diferentes herramientas para determinar cuáles son los requerimientos y también servirá para tener un argumento por parte del docente que valide la elaboración de este proyecto, para esto se ha utilizado una entrevista al docente actual, la cual se encuentra contestada a mano por el docente en el Anexo 1.

Tabla 10. Análisis de tarjetas de entrenamiento

SITUACIÓN ACTUAL

Método: Los métodos seleccionados son la búsqueda de información y la entrevista para determinar las mejores prácticas en el desarrollo de las placas de entrenamiento y pueden ser validadas por el profesor.

Formato: Las preguntas a contestar para determinar los requisitos son:

1. ¿Cuáles son los temas que se llegan a impartir en las materias de Sistemas Microprocesados y Optativa II, se cumple el tiempo determinado?
 2. ¿Cuáles son los componentes de hardware que se pueden utilizar en cada materia?
 3. ¿Qué software se maneja en cada asignatura?
 4. ¿Qué inconvenientes se encuentra al realizar las prácticas de cada materia?
 5. ¿Qué proyectos se llevan a cabo a lo largo del semestre?
 6. ¿Cree que los gastos por cada materia se pueden ahorrar en función de la compra de materiales?
-

-
7. ¿Qué materiales son lo que más inversión representa a cada estudiante, y que se puede hacer para encontrar una forma de ahorro?
-

**ENTREVISTA AL ING. PAUL ROSERO
DOCENTE ACTUAL DE LAS MATERIAS DE
SISTEMAS MICROPROCESADOS Y
MICROCONTROLADORES AVANZADOS**

1. ¿Cuáles son los temas que se llegan a impartir en las materias de sistemas microprocesados y optativa II, se cumple el tiempo determinado?

Se llegan a impartir casi el 90% de los contenidos de cada materia, el tiempo para cada tema es un poco complicado en las prácticas por tiempo de armado de los estudiantes, en el caso de traer armado de la casa no se puede constatar si fue desarrollado por el estudiante

.

2. ¿Cuáles son los componentes de hardware que se pueden utilizar en cada materia?

En la materia de sistemas microprocesados se trabaja con lo siguiente:

- *Leds*
- *Resistencias*
- *Switchs*
- *Cables*
- *Microcontrolador ATMEGA164PA*
- *Un grabador de microcontroladores*
- *Protoboard*
- *Lcd*
- *Teclado matricial*
- *Potenciómetros*
- *Sensores análogos.*
- *Cable USB-TTL*

En la materia de sistemas embebidos hemos podido mejorar la gama del procesador a trabajar y los elementos son muy parecidos a los mencionados anteriormente, con el cambio de utilizar un Intel Galileo Gen 2, por sus mayores prestaciones, librerías y aplicaciones

3. *¿Qué software se maneja en cada asignatura?*

En sistemas microprocesados se trabaja con CodeVision, XtremeBurner, Proteus e Eagle

principalmente, solo en aplicaciones de comunicación serial se usa hyperterminal de Windows

En la materia de sistemas embebidos todo es software libre, se utiliza el IDE de Arduino para Intel, Processing, Eagle y lenguaje en HTML donde el programa lo elige el estudiante entre algunas herramientas que se le presenta en clases.

4. ¿Qué inconvenientes se encuentra al realizar las prácticas de cada materia?

El tiempo en armado de cada grupo de trabajo, realizar un programa y armar para probar en ciertas ocasiones no se puede llegar a completar en 2 horas clases, en algunos temas de mayor relevancia se realizan placas para evitar este inconveniente pero los estudiantes tienen problemas al trabajar con tantos pines de un microcontrolador y no trae las mismas probadas, esto genera un gran retraso y no permite culminar un trabajo.

En la materia de sistemas embebidos específicamente se tiene el problema de manipular los pines de plástico que tiene el Galileo, ya que por su uso se desgasta con la

inserción de cables y luego pierden la conectividad deseada.

5. ¿Qué proyectos se llevan a cabo a lo largo del semestre?

Los proyectos de sistemas microprocesados es el manejo de la mayoría de periféricos del microcontrolador, se generan algunas placas con diversos usos y se graban diferentes programas para verificar el funcionamiento. Sistemas embebidos está orientado al uso de sensores para una aplicación real de un sistemas con una interfaz de monitoreo o control.

6. ¿Cree que los gastos por cada materia se pueden ahorrar en función de la compra de materiales?

Los gastos en algunos casos son altos porque el estudiante daña sus propios elementos y reincide en la compra de los mismos materiales, no existe una buena organización y almacenamiento de elementos comunes, además de no tener la costumbre de probar los elementos antes de usarlos o desecharlos.

Al evitar estas compras demás se podría ahorrar el gasto del estudiante y sobre todo que los elementos queden en

la universidad para que nuevas promociones las puedan usar.

7. ¿Qué materiales son lo que más inversión representa a cada estudiante, y que se puede hacer para encontrar un forma de ahorro?

Son los procesadores los más costosos, un grupo de 3 personas dañan alrededor de 2 a 3 micros por semestre por no saber seguir las instrucciones y no asegurar los pines de voltaje adecuadamente.

En sistemas embebidos es menor el daño ya que el estudiante ya tiene mayor conciencia al trabajar con un sistema 10 veces más caro, todavía existen casos de daños pero en grupo muy reducidos.

Para tener un correcto desarrollo de las placas, se considera tener claros ciertos aspectos importantes como el propósito, los objetivos, la meta y las abreviaturas que se van a usar para referirse al sistema.

INTRODUCCIÓN

Propósito: El propósito de desarrollar las placas de entrenamiento es evitar la reincidencia de la compra de nuevos materiales y se más práctico el aprendizaje de electrónica programada.

Ámbito del sistema: Este proyecto se denomina Tarjetas de Entrenamiento para Aprendizaje de Electrónica Programada (T.E.A.E.P.)

Beneficios:

- Mejor didáctica en el curso.
 - Evitar la reincidencia de armados
 - Eliminar el tiempo perdido en armado de sistemas de pruebas
 - Generar sistemas propios de aprendizaje reutilizables
-

Objetivos:

- Determinar la mejor forma de conexión de cada tarjeta para su debida relación entre las mismas.
 - Desarrollar tarjetas de entrenamiento de fácil conexión para los estudiantes.
 - Realizar un sistema de prueba en menor tiempo de ensamblaje para la depuración del mismo.
-

Metas:

El sistema T.E.A.E.P. trabaje sin errores.

Definiciones, Acrónimos y abreviaturas: Tarjetas de Entrenamiento para Aprendizaje de Electrónica Programada (T.E.A.E.P.)

El proyecto a desarrollarse tiene sus propias especificaciones tanto en hardware y software, a continuación se realiza una descripción general de las funciones que debe cumplir.

DESCRIPCIÓN GENERAL

Perspectiva del sistema: El sistema tiene una función más específica que los shields o boards desarrollados en Arduino, estas tarjetas son generales y vienen generalmente con sensores robustos incorporados, el sistema se basa un esquema definido de estudios.

Funciones del proyecto: El sistema plantea diferentes tarjetas, cada tarjeta posee diferentes funciones para el manejo de diferentes periféricos, deben existir tarjetas que cumplan determinadas funciones y otras de una manera general que pueda abarcar más contenidos de las asignaturas previstas.

Interfaces: No tiene, se implementa guías.

Software:

- Programas de Arduino

Hardware:

- Puertos como salida
 - Puertos como entrada
 - Cads
 - Display
 - LCD y teclado
-

-
- Tarjeta completa
 - Motores/ Servo
 - Sensores
 - Tarjetas de IoT
-

Tipos de tecnología:

Software:

Hardware:

Lenguaje C.

Analógica y digital

Este proyecto maneja conceptos de electrónica para estudiantes que hayan pasado las materias básicas y que estén en la capacidad de formar un criterio propio para generar conocimiento a través de sus prácticas, por lo que está enfocado en estudiantes de sexto y séptimo nivel.

USUARIOS

Características: Son estudiantes de 6to y 7mo nivel de la Carrera de Ingeniería en Electrónica y Redes de Comunicación con conocimientos

avanzados de electrónica y programación con asesoría de profesores.

Restricciones:

- ✓ No todas las tarjetas se conectan entre sí
 - ✓ No se pueden modificar las tarjetas.
 - ✓ Cada tarjeta tiene un límite de aplicaciones
 - ✓ Algunas placas pueden usar alimentación externa.
 - ✓ Serán probadas son para la placa Intel Galileo Gen 2.
 - ✓ Solo se usa un servidor de IoT
 - ✓ Tendrán pines a la vista.
-

Referencia: Autoría

3.1.2. Análisis de guía

Actualmente no se cuenta con una guía determinada con la que se puedan apoyar los estudiantes, cada docente tiene sus documentos de apoyo pero no una guía estudiantil para el manejo de tecnologías nuevas. En la tabla 11 se explica el análisis para la realización de guías.

Tabla 11. Análisis de guía

SITUACIÓN ACTUAL

Método: No se cuenta con una guía de prácticas de la materia por no tener hardware para desarrollo.

Resumen: Los profesores de las materias enfocadas a electrónica tienen libros de referencia y diapositivas, no existen elementos actuales de uso para las materias de sistemas microprocesados y sistemas embebidos en los laboratorios de la Universidad que orienten al estudiante a nuevas tecnologías presentes.

Al generarse un proyecto que va a servir al aprendizaje de los estudiantes, es necesario crear una guía que sirva de apoyo y que explique el correcto manejo de las placas.

INTRODUCCIÓN

Propósito: El propósito de la guía es desarrollar un herramienta para el correcto uso de T.E.A.E.P

Ámbito del

sistema:

Beneficios:

Una mejor manipulación de la tarjetas y alargar su tiempo de vida

Objetivos:

Desarrollar una guía de apoyo al estudiante para el correcto uso de T.E.A.E.P.

Metas:

La guía debe ser aceptada como una herramienta de apoyo

Referencia: Autoría

3.1.3. Requerimientos de tarjetas de entrenamiento

En el desarrollo de las tarjetas solo se hace referencia a los requerimientos de arquitectura, ya que el software es modificable y puesto en consideración en la guía.

El sistema se divide en diferentes partes según las necesidades de aprendizaje una de ellas es el tipo de tarjetas que se va a tener, uno son de adquisición de datos y por otro lado el envío de datos (actuadores), además es necesario determinar que tarjetas necesitan los pines de GND o VCC o ambas.

En las siguientes tablas desde la 12 hasta la 19 se muestran los requerimientos de arquitectura para la realización de las placas, explicando que es lo que se necesita en cada caso.

3.1.3.1. Puertos salida

Tabla 12. Requerimientos de arquitectura puertos como salida

#	REQUERIMIENTO	REQUERIMIENTOS DE FUNCIONES			RELACIÓN	VERIFICACIÓN
		PRIORIDAD				
		Alta	Media	Baja		
REQUERIMIENTOS LÓGICOS						
SRSH 1	<i>Deben estar los pines declarados como salida</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE DISEÑO						
SRSH 2	<i>No pueden obstruir otros pines del Intel Galileo</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE SOFTWARE						
SRSH 3	<i>Solo necesita estados lógicos para funcionar.</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE HARDWARE						
SRSH 4	<i>Es necesario de 4 a 6 leds para generar secuencias</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 5	<i>Deben tener resistencias para limitar paso de corriente</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
SRSH 6	<i>Se debe considerar el tamaño para la selección de leds</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
SRSH 7	<i>Se debe determinar que pin determina el funcionamiento de cada led</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
REQUERIMIENTOS ELÉCTRICOS						
SRSH 8	<i>Necesitan conexión a GND</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

3.1.3.2. Puertos entrada

Tabla 13. Requerimientos de arquitectura puertos como entrada

REQUERIMIENTOS DE FUNCIONES						
#	REQUERIMIENTO	PRIORIDAD			RELACIÓN	VERIFICACIÓN
		Alta	Media	Baja		
REQUERIMIENTOS LÓGICOS						
SRSH 9	<i>Se deben configurar los pines como salida</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 10	<i>Necesita estructuras de control</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE DISEÑO						
SRSH 11	<i>El software de diseño de placas debe tener homologación con las páginas de referencia</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE SOFTWARE						
SRSH 12	<i>Solo recibe estados lógicos digitales</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE HARDWARE						
SRSH 13	<i>Se debe tener estados lógicos permanentes y variables 1 dipSwitch de 2 y dos botones, un pull-up y pull-down como mínimo</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 14	<i>Necesita resistencias para configuraciones de pull-up y pull-down</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS ELÉCTRICOS						
SRSH 15	<i>Necesita de conexiones de VCC y GND</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Referencia: Autoría

3.1.3.3. Conversores análogos digitales

Tabla 14. Requerimientos de arquitectura CAD's

#	REQUERIMIENTO	PRIORIDAD			RELACIÓN	VERIFICACIÓN
		Alta	Media	Baja		
REQUERIMIENTOS LÓGICOS						
SRSH 16	<i>Necesita una variable de tipo entero</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE DISEÑO						
SRSH 17	<i>Debe tener varios métodos de enrutamiento de la placa</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE SOFTWARE						
REQUERIMIENTOS DE HARDWARE						
SRSH 18	<i>Se usan mínimo de 2 CADs para ver funcionalidades</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS ELÉCTRICOS						
SRSH 19	<i>Necesita de conexiones de VCC y GND</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Referencia: Autoría

3.1.3.4.Motores/servos/chicharra

Tabla 15. Requerimientos de arquitectura motores/servos/chicharra

#	REQUERIMIENTO	PRIORIDAD			RELACIÓN	VERIFICACIÓN
		Alta	Media	Baja		
REQUERIMIENTOS LÓGICOS						
SRSH 20	<i>Necesita declarar pines como salida</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE DISEÑO						
SRSH 21	<i>Poseer librerías de Intel Galileo para desarrollo de placas</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE SOFTWARE						
SRSH 22	<i>Necesitan librerías</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE HARDWARE						
SRSH 23	<i>Necesita otra fuente de alimentación</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 24	<i>Necesita una fase de amplificación</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS ELÉCTRICOS						
SRSH 25	<i>Necesita de conexiones de VCC y GND</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Referencia: Autoría

3.1.3.5.Teclado/LCD

Tabla 16. Requerimientos de arquitectura teclado/lcd

#	REQUERIMIENTO	PRIORIDAD			RELACIÓN	VERIFICACIÓN
		Alta	Media	Baja		
REQUERIMIENTOS LÓGICOS						
SRSH 26	<i>Necesita declarar pines de datos</i>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE DISEÑO						
SRSH 27	<i>Debe ser libre el software de diseño</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE SOFTWARE						
SRSH 28	<i>Necesitan librerías</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE HARDWARE						
SRSH 29	<i>Necesita potenciómetro de control de luminosidad</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 30	<i>Ocupa todos los pines digitales</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS ELÉCTRICOS						
SRSH 31	<i>Necesita de conexiones de VCC y GND</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Referencia: Autoría

3.1.3.6.Sensores

Tabla 17. Requerimientos de arquitectura sensores

#	REQUERIMIENTO	PRIORIDAD			RELACIÓN	VERIFICACIÓN
		Alta	Media	Baja		
REQUERIMIENTOS LÓGICOS						
SRSH 32	<i>Necesita declarar pines de datos</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE DISEÑO						
SRSH 33	<i>El software de diseño debe contar con todos elementos necesarios para cada placa</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 34	<i>Se necesita tener datos análogos</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 35	<i>Se necesitan datos del medio ambiente/costo</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE SOFTWARE						
SRSH 36	<i>Necesitan librerías</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE HARDWARE						
SRSH 37	<i>Entorno de recolección de datos modificable</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS ELÉCTRICOS						
SRSH 38	<i>Necesita de conexiones de VCC y GND</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Referencia: Autoría

3.1.3.7.Aplicaciones IoT

Tabla 18. Requerimientos de arquitectura aplicaciones IoT

#	REQUERIMIENTO	PRIORIDAD			RELACIÓN	VERIFICACIÓN
		Alta	Media	Baja		
REQUERIMIENTOS DE FUNCIONES						
REQUERIMIENTOS LÓGICOS						
SRSH 39	<i>Solo envía caracteres de tipo string</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE DISEÑO						
SRSH 40	<i>Debe ser correctamente documentado</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 41	<i>Se necesitan datos del medio ambiente</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE SOFTWARE						
SRSH 42	<i>Necesitan librerías</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE HARDWARE						
SRSH 43	<i>Debe ser un entorno controlado con datos a tiempo real</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS ELÉCTRICOS						
SRSH 44	<i>Necesita de conexiones de VCC y GND externas</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Referencia: Autoría

3.1.3.8. Tarjetas completas

Tabla 19. Requerimientos de arquitectura tarjetas completas

#	REQUERIMIENTO	PRIORIDAD			RELACIÓN	VERIFICACIÓN
		Alta	Media	Baja		
		REQUERIMIENTOS DE FUNCIONES				
REQUERIMIENTOS LÓGICOS						
SRSH 45	<i>Necesita declarar pines para cada periférico</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE DISEÑO						
SRSH 46	<i>Las placas deben ser impresas ambos lados</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 47	<i>Necesita datos digitales y análogos</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
REQUERIMIENTOS DE HARDWARE						
SRSH 48	<i>Debe coincidir con todos los pines para poder trabajar</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
<i>Todos los pines deben especificar que periférico usa</i>						
REQUERIMIENTOS ELÉCTRICOS						
SRSH 49	<i>Usa los pines de VCC y GND del mismo Galileo</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Referencia: Autoría

3.2.Requerimientos de guías

El uso de guías para el manejo de las placas realizadas es importante en este proyecto, por lo que la tabla 20 muestra los requerimientos para su realización.

Tabla 20. Requerimientos de guía

REQUERIMIENTOS DE FUNCIONES						
#	REQUERIMIENTO	PRIORIDAD			RELACIÓN	VERIFICACIÓN
		Alta	Media	Baja		
REQUERIMIENTOS DE DISEÑO						
SRSH 50	<i>Debe indicar el cuidado de usar la placa Intel Galileo</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 51	<i>Debe indicar procesos de desarrollo de sistemas simples y avanzados</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 52	<i>Debe existir una forma de medir el conocimiento del estudiante</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 53	<i>Debe indicar la forma de conexión</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 54	<i>Debe indicar partes prácticas y teóricas</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
SRSH 55	<i>Debe indicar cuantas y que funciones tiene cada placa</i>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Referencia: Autoría

3.3.DESARROLLO

3.3.1. Diseño de placas

El diseño de una placa debe ser realizado mediante un software específico que permita determinar adecuadamente la ubicación de los elementos y el enrutamiento de las pistas de conducción de corriente.

Solo se hace referencia al software de placas, ya que el resto de software es propietario de Intel galileo y es restrictivo a solo los que nos indica su página oficial. En la tabla 21 se detalla la elección del software para el diseño de las placas.

Tabla 21. Elección de software de edición

SOFTWARE	REQUERIMIENTOS SRSH							VALORACIÓN TOTAL
	11	17	21	27	33	40	46	
Eagle	●	●	●	●	●	●	●	7 puntos
PCB Wizard	●	○	○	●	○	○	●	2 puntos
Fritzing	●	○	●	●	●	●	○	5 puntos

●Cumple

○ No cumple

Elección: Se selecciona como software de diseño a Eagle por cumplir con todos los requerimientos de las placas a desarrollar, no se descarta Fritzing para otras aplicaciones. Un manual básico para utilizar Eagle y realizar el diseño de las placas se encuentra en el Anexo 3 del trabajo de titulación.

Referencia: Autoría

3.3.2. Elección de sensores

Para este proyecto se tomaron en cuenta variables medibles con sensores, su elección se muestra en la tabla 22 de acuerdo a los requerimientos anteriormente realizados.

Tabla 22. Elección de sensores

HARDWARE	REQUERIMIENTOS SRSH							VALORACIÓN TOTAL
	32	33	34	35	36	37	38	
Sensor de corriente	○	●	●	○	○	○	○	3 puntos
Sensor de voltaje	○	●	●	○	○	○	●	3 puntos
foto receptor	○	●	○	○	○	○	●	2 puntos
Sensor infrarrojo	○	●	○	○	○	○	●	2 puntos
Sensor de luz	○	●	○	○	○	○	●	2 puntos
Sensor de imagen	○	●	●	○	○	●	●	4 puntos
Sensor de temperatura/humedad	●	○	●	●	●	●	●	7 puntos
Sensor de color	○	●	●	●	○	○	●	4 puntos
Sensor de humo	●	○	●	●	○	○	●	4 puntos
Sensor de distancia	●	●	●	●	●	●	●	7 puntos
Sensor de inclinación	○	●	●	○	○	●	●	4 puntos
Acelerómetro	●	●	●	●	●	●	●	7 puntos
Sensor de humedad de suelo	○	○	●	●	○	○	●	4 puntos
Sensor de gotas	○	●	●	●	○	○	●	4 puntos
Sensor de presencia	○	●	●	●	○	●	●	5 puntos
Reloj	○	○	●	●	○	○	●	3 puntos
RFID	○	○	●	●	○	○	●	3 puntos
GPS	○	○	●	●	○	●	●	4 puntos
Bluetooth	●	●	○	●	●	●	●	6 puntos
Zigbee	●	●	○	●	●	●	●	6 puntos
Sensor de golpes	○	●	○	○	○	○	●	2 puntos

●Cumple

○ No cumple

Elección: Se selecciona por su puntuación: sensor de temperatura, sensor de distancia, acelerómetro, sensor de presencia y bluetooth ya que el costo lo diferencia a Zigbee.

Referencia: Autoría

Una vez que se ha decidido las variables que se desean medir, es necesario realizar un estudio previo para saber que sensor utilizar en este trabajo de titulación, ya que existen varios en el mercado pero no todos cumplen con los requerimientos deseados, lo cual se explica a continuación en la tabla 23.

Para la elección de los sensores se han tomado en cuenta tres aspectos importantes: el costo, su disponibilidad en el mercado local y su aplicabilidad en este proyecto.

Tabla 23. Estudio de los sensores existentes para determinar los más óptimos a usar en el trabajo de titulación

SENSORES	CARACTERÍSTICAS			ELECTO
TEMPERATURA	MAX6675	AR837	DTH11	DTH11
	Costo: 20\$ Disponibilidad: Local Dato: Solamente mide temperatura	Costo: 70\$ Disponibilidad: Fuera del mercado local Dato: Mide temperatura y humedad, compatible con Arduino	Costo: 6\$ Disponibilidad: local DATO: Mide temperatura y humedad, estabilidad a largo plazo, compatible con Arduino	
ACELERACIÓN	ACELERÓMETRO ADXL345	ACELERÓMETRO GIROSCOPIO	ACELERÓMETRO MMA7455I	ACELERÓMETRO ADXL345
	Costo: 8\$ Disponibilidad: Local Dato: Funciona en los tres ejes, compatible con la Plataforma Arduino	Costo: 12 Disponibilidad: Fuera del mercado local Dato: Compatible con la Plataforma Arduino.	Costo: 10\$ Disponibilidad: Agotada Dato: Compatible con la Plataforma Arduino	

	ZIGBEE	BLUETOOTH RS232	BLUETOOTH ZS-040	
INALÁMBRICO	Costo: 35 Disponibilidad: Local Dato: Compatible con la Plataforma Arduino	Costo: 30\$ Disponibilidad: Fuera del mercado local Dato: Compatible con la Plataforma Arduino.	Costo: 8\$ Disponibilidad: Local Dato: Compatible con la Plataforma Arduino	BLUETOOTH ZS-040
	ULTRASÓNICO O HC-SR04	ULTRASÓNICO RS232	ULTRASÓNICO O DE NIVEL ENDRESS	
PROXIMIDAD	Costo: 6\$ Disponibilidad: Local Datos: Compatible con la Plataforma Arduino.	Costo: 100\$ Disponibilidad: Fuera del mercado local Dato: Es compatible con la Plataforma Arduino	Costo: 100\$ Disponibilidad: Local Dato: Aplicaciones industriales	ULTRASÓNICO HC-SR04
	INFRARROJO	PIR	SENSOR DE MOVIMIENTO INALÁMBRICO	
MOVIMIENTO	Costo: 17\$ Disponibilidad: Local Dato: Su aplicación es muy limitada.	Costo: 4,50 \$ Disponibilidad: Local Dato: Detecta movimiento, fácil de usar, no se desgasta. Compatible con Arduino	Costo: 80\$ Disponibilidad: Fuera del mercado local Dato: No es compatible con Arduino.	PIR

Referencia: Mercado local de elementos electrónicos

Los sensores elegidos son: DTH11, PIR, Ultrasonico HC-SR04, Acelerómetro ADXL345 y Bluetooth ZS-040, ya que tomándose en cuenta los tres aspectos importantes como costo, disponibilidad y funciones, son los más óptimos.

3.3.3. Desarrollo de placas

3.3.3.1. Introducción

El desarrollo de placas es necesario descargar los diseños de placa Arduino Uno, ya que la distribución de pines es la misma, estos documentos son libres desde la página oficial de Arduino, en algunos casos por el tamaño de las placas se utilizará un mismo board, para un desarrollo de placas en Eagle se generan dos archivos, un board para cargar todos los elementos y como se van a conectar eléctricamente y un Esquemático para la organización de los elementos en la placa. La distribución de pines se muestra en la figura 17.

Es importante mencionar que la placa Intel Galileo G2 soporta hasta 800mA de corriente, por lo tanto las placas de entrenamiento no pueden exceder o sobrepasar esta cantidad de corriente permitida.

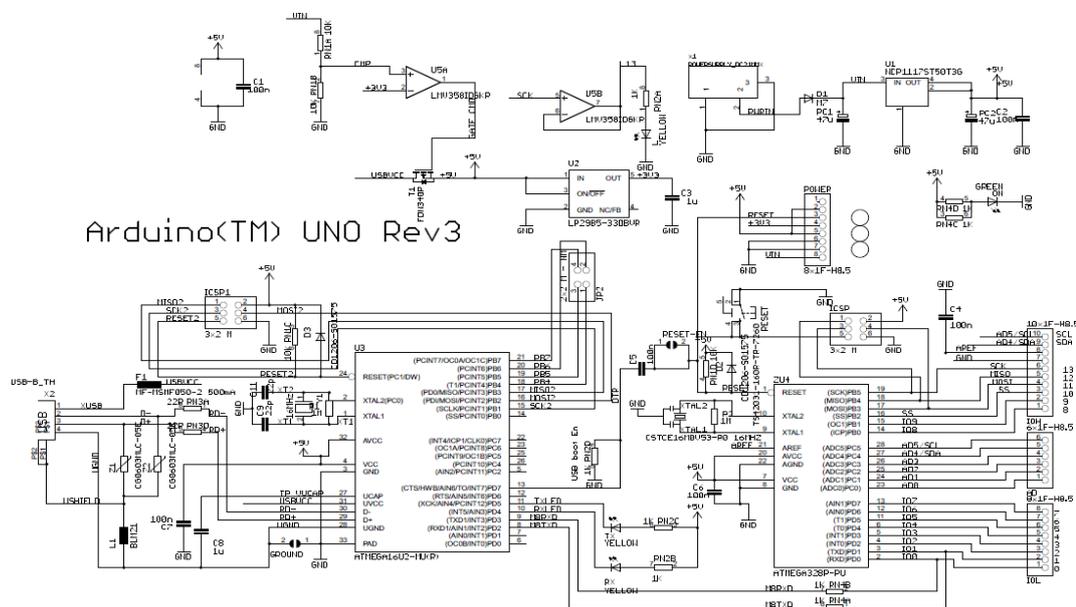


Figura 17. Distribución de pines Arduino Uno
Referencia: <https://www.arduino.cc/en/Main/arduinoBoardUno>

3.3.3.2. Pines digitales entrada/salida

Se usa un mismo Board y un mismo Esquemático por el tamaño de las placas y una mejor documentación de la misma. Cada placa debe tener pines de voltaje y tierra para realizar las diferentes conexiones, estas placas se trabajan por independiente. Se usan 4 leds para poder generar salidas visibles con sus respectivas resistencias para proteger los leds, esta placa está relacionada con la placa de entradas que contiene las dos configuraciones de estados lógicos pull-up y pull-down en pulsadores y un dipswitch de 2 posiciones con sus respectivas resistencias que son parte de la configuración para que funcionen correctamente, su diseño se encuentra en la figura 18 y figura 19.

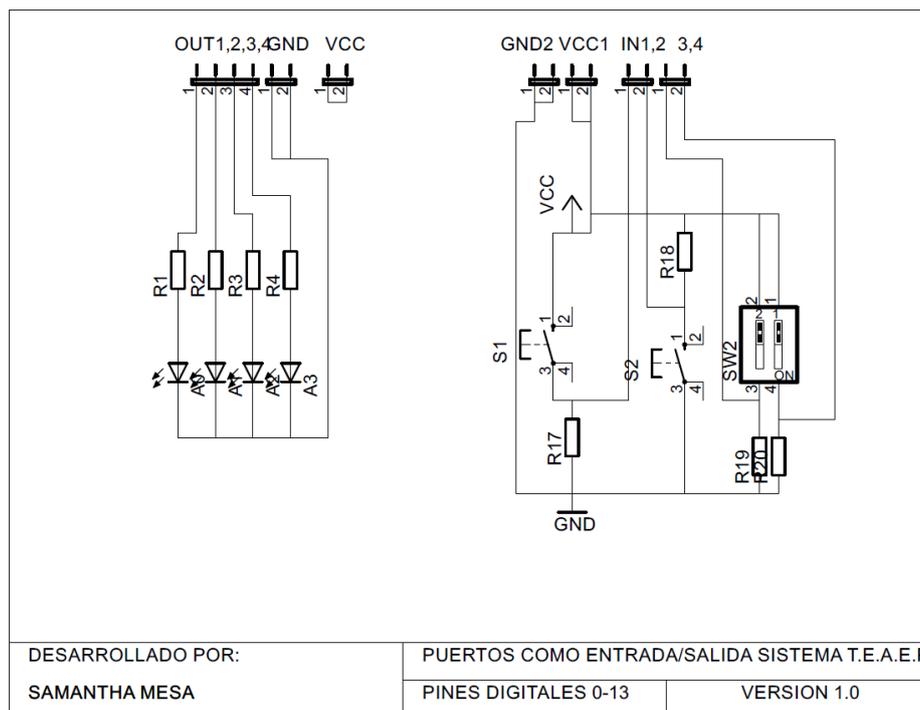


Figura 18. DISEÑO PUERTOS DIGITALES ENTRADA SALIDA
Referencia: Autoría

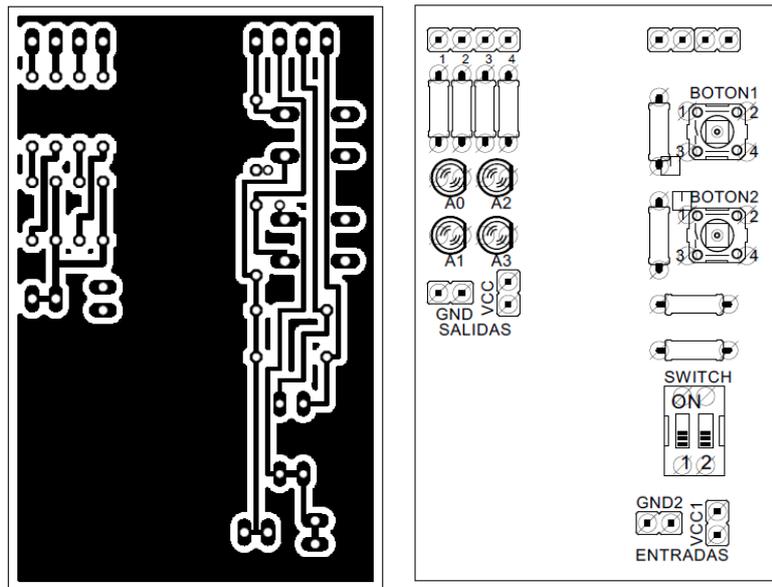


Figura 19. PLACAS PUERTOS ENTRADA SALIDA DIGITAL

Referencia: Autoría

3.3.3.3. Pines análogos de entrada

Para el diseño de la placa análoga solo se configurará como entrada ya que en la distribución de pines se encuentran en un sector aislado del resto de pines, para trabajar con pines análogos siempre es necesario conectar en con cada convertor voltaje y tierra, para el diseño se usarán dos potenciómetros de precisión y se dejará dos pines libres para la colocación de sensores como indican las figuras 20 y 21.

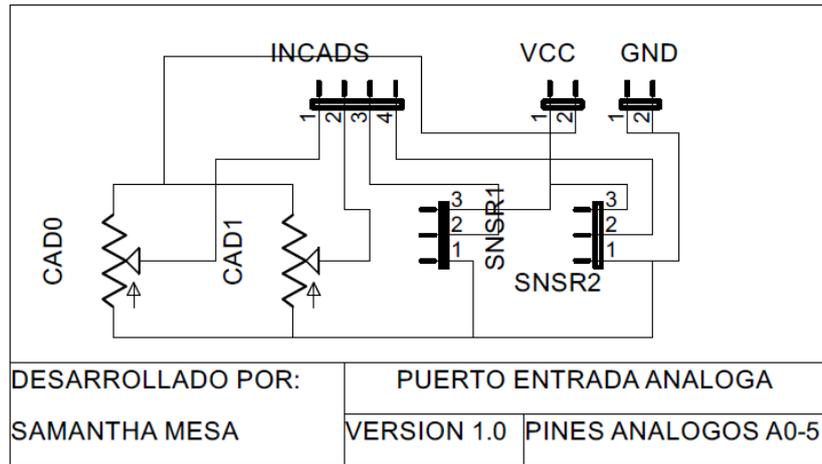


Figura 20. DISEÑO DE PLACA CAD ENTRADA
Referencia: Autoría

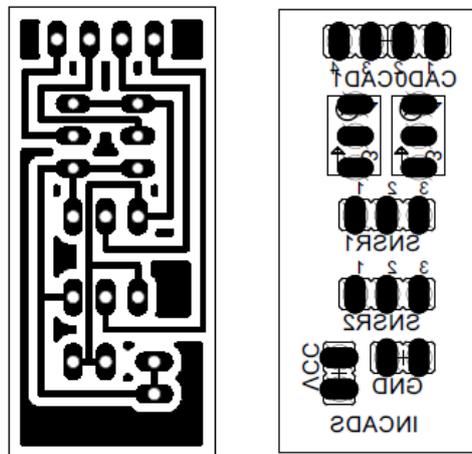


Figura 21. PLACA ENTRADA CADS
Referencia: Autoría

3.3.3.4. Motores analógicos/digitales

Se realiza una placa solo de motores por la necesidad de fuentes externas y etapas de amplificación mediante transistores, para esta placa se trabaja con un motor dc, un servo motor y un zumbador, en sí son salidas de mayor aplicación; tienen conexiones de voltaje y tierra del mismo Arduino y una externa compartida. En las figuras se 22 y 23 se observa su diseño.

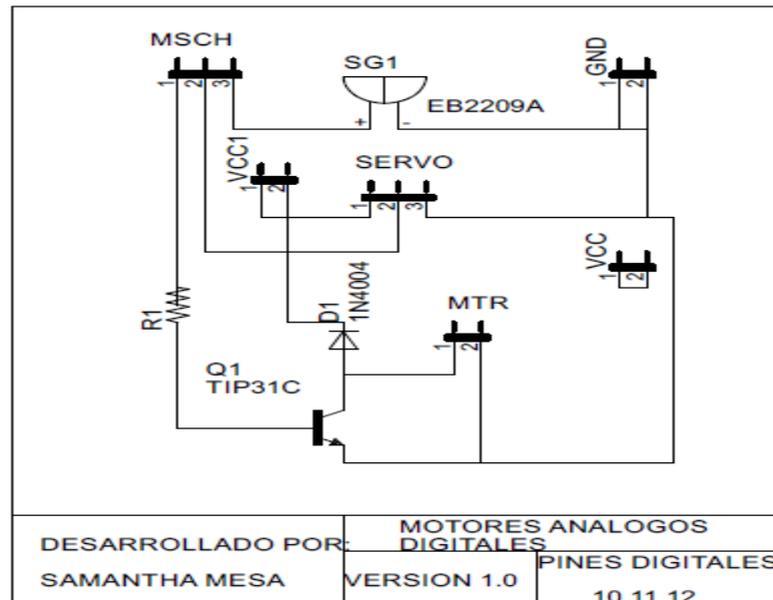


Figura 22. DISEÑO MOTORES ANÁLOGOS DIGITALES
Referencia: Autoría

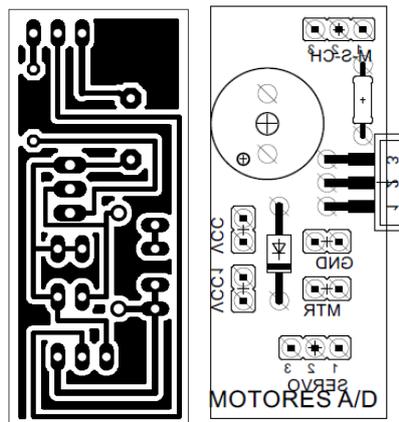


Figura 23. PLACAS MOTORES A/D
Referencia: Autoría

3.3.3.5. Display con decodificador

Se usa una placa con decodificador para evitar la utilización exagerada de pines, con el decodificador solo se usan 4 pines en vez de 7 pines, esto ayuda para insertar nuevas placas y generar más aplicaciones. Se utilizan pines de voltaje y tierra como se muestran en las figuras 24 y 25.

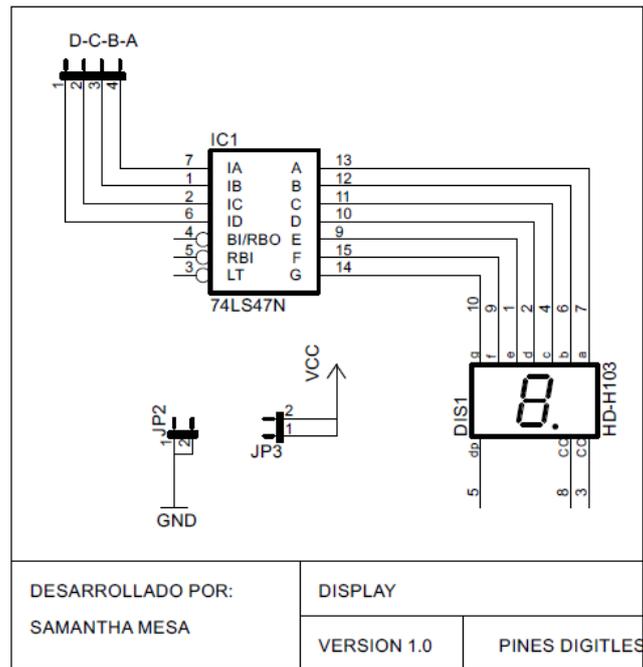


Figura 24. DISEÑO PLACA DISPLAY
Referencia: Autoría

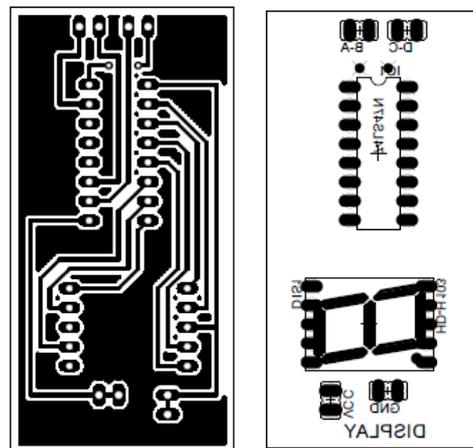


Figura 25. DISEÑO DE PLACA DISPLAY
Referencia: Autoría

3.3.3.6. Entrenador análogo digital completo

Para un trabajo desarrollo completo de aplicaciones con interacción entre pines análogos y digitales se desarrolló una placa que contenga potenciómetro como

convertidores analógicos digitales, dipswitch para entrada de datos digitales, uno en configuración pull-up y otro configuración pull-down, leds para salidas digitales y una chicharra, además de una salida para un motor de mayor prestaciones con entrada de voltaje mayores con una fase de amplificación como se muestran en las figuras 26, 27 y 28.

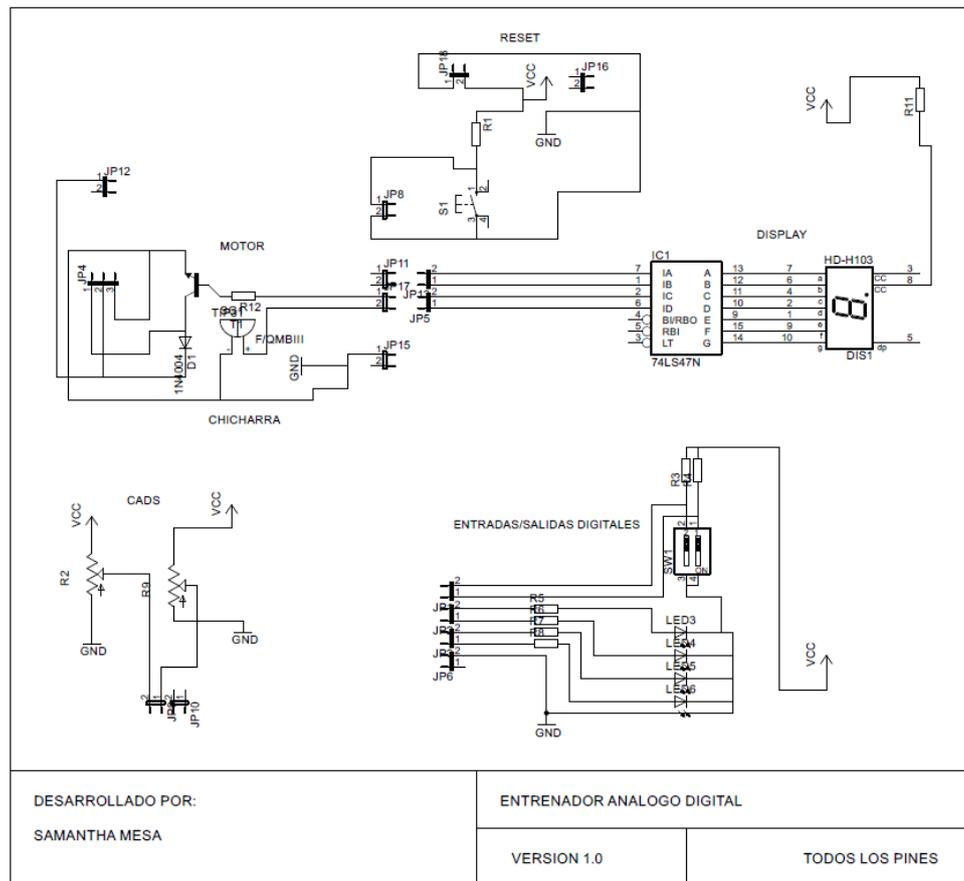


Figura 26. DISEÑO DE ENTRENADOR ANALOGO DIGITAL
Referencia: Autoría

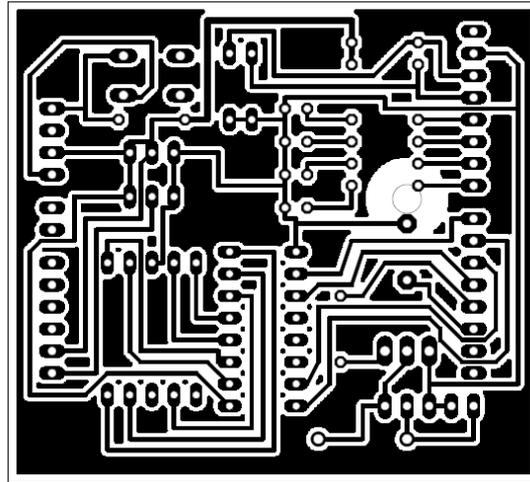


Figura 27. DISEÑO PLACA ENTRENADOR ANÁLOGO DIGITAL
Referencia: Autoría

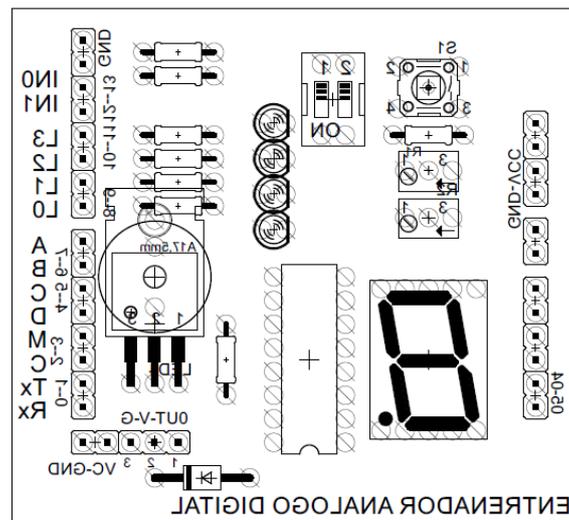


Figura 28. DISEÑO PLACA ENTRENADOR ANÁLOGO DIGITAL
Referencia: Autoría

3.3.3.7.Lcd

El diseño de la placa con lcd por su tamaño se dispondrá de la misma forma que la placa display que se encuentra en la parte interna del Intel Galileo para evitar interferir

con otras placas, tiene un potenciómetro para definir el brillo de la pantalla como se muestran en las figuras 29 y 30.

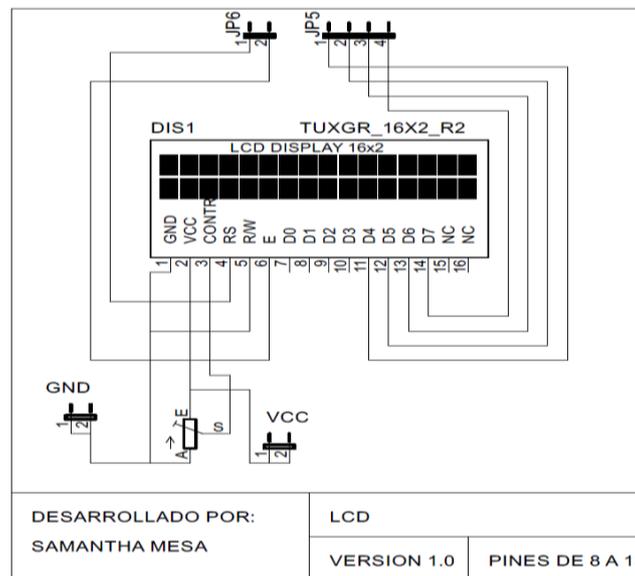


Figura 29. DISEÑO DE PLACA LCD
Referencia: Autoría

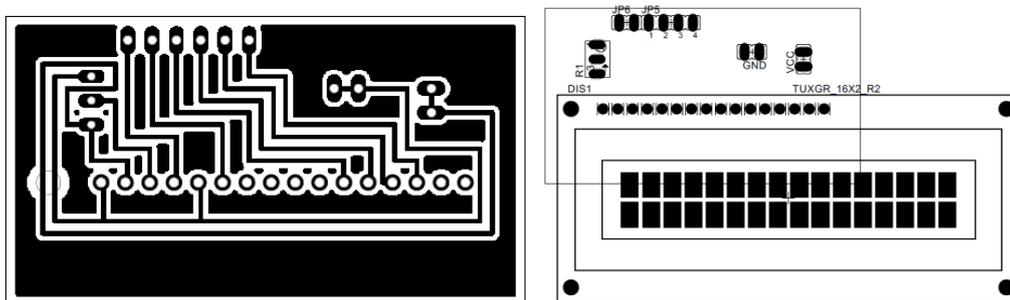


Figura 30. DISEÑO DE PLACA LCD
Referencia: Autoría

3.3.3.8. Teclado y bluetooth

La placa de teclado es muy sencilla ya que Arduino tiene una librería que simplifica el trabajo, solo es necesario conectar a los pines que se definan en la librería como se muestran en las figuras 31 y 32.

Al ser una placa mus similar la placa de bluetooth conecta con pines Tx y Rx con alimentación se realiza un solo diseño conjunto a teclado que se demuestran en las figuras 31 y 32, se hace referencia al módulo HC-05 por su popularidad en el medio.

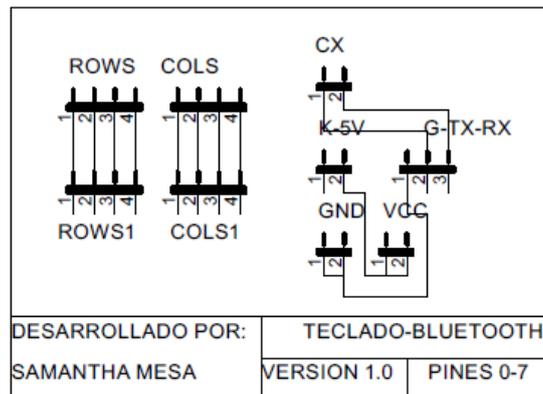


Figura 31. DISEÑO PLACAS TECLADO Y BLUETOOTH
Referencia: Autoría

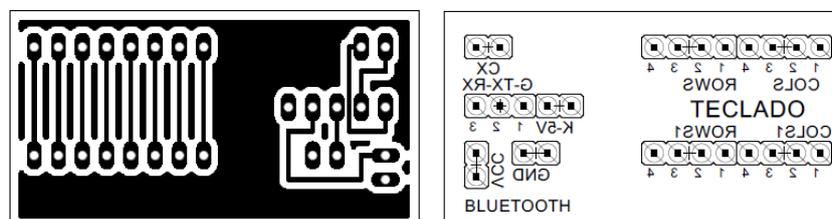


Figura 32. DISEÑO DE PLACAS TECLADO Y BLUETOOTH
Referencia: Autoría

3.3.3.9.Sensores

Los sensores establecidos para realizar las placas son: sensor ultrasónico, presencia y acelerómetro, cada placa viene soldado el sensor para solo colocar en la placa Intel Galileo.

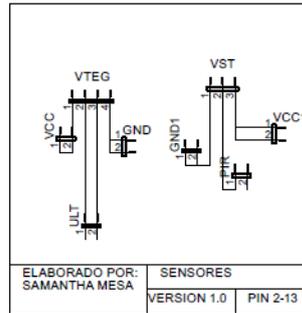


Figura 33. DISEÑO DE PLACAS SENSORES
Referencia: Autoría

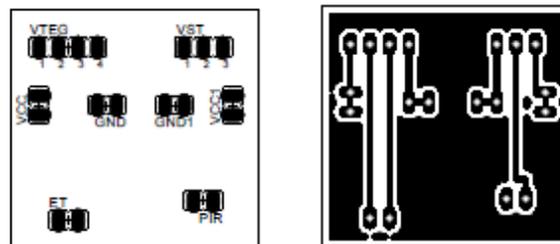


Figura 34. DISEÑO DE PLACAS SENSORES
Referencia: Autoría

3.3.3.10. IOT

Las placas relacionadas con IOT realizar una simulación de condiciones de ambientes reales para poder realizar un monitoreo con almacenamiento en la nube, se plantea una caja de pruebas donde se pueda modificar la temperatura de la misma para poder almacenar los datos y realizar sistemas de control de la caja. Por condiciones de diseño estas placas no están ligadas a la colocación de Boards dentro del Intel Galileo, por cuidados de la placa se realizar conexiones mediante cables y pines de conexión de Vcc y Gnd. Para poder cambiar la temperatura es necesario conexiones a 120 voltios para el foco, 12 voltios para el ventilador y 5 voltios para los sensores; se representa el diseño de las placas de la fuente y de las conexiones al Intel.

3.3.3.11. Relé a 120 y ventilador y temperatura

La placa contiene una fuente con una entrada a 120 VAC para tener salidas de 12 VDC y 5 VDC, con una conexión a relé para un foco a 120 VAC y genere calor a los sensores de temperatura, para la administración de la placa tiene conectores y se puedan conectar a los diferentes pines del Intel Galileo.

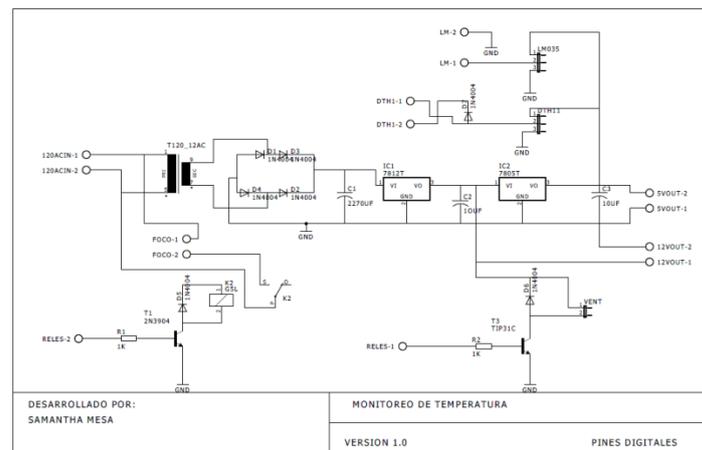


Figura 35. DISEÑO DE PLACA IOT
Referencia: Autoría

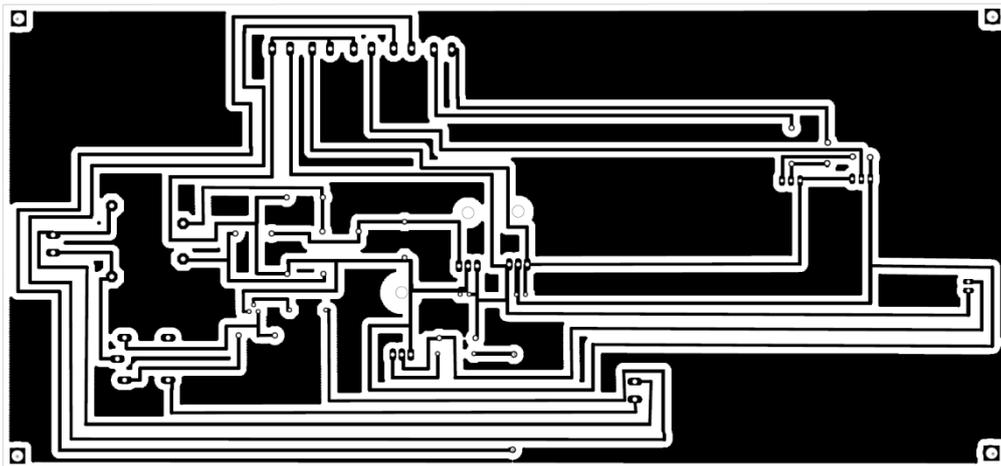
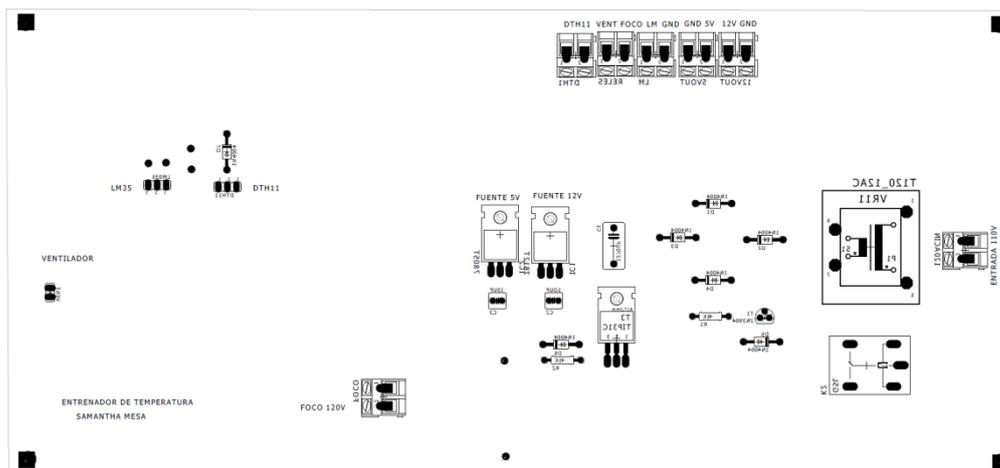


Figura 36. DISEÑO PLACA IOT
Referencia: Autoría



*Figura 37. DISEÑO DE PLACA IOT
Referencia: Autoría*

3.3.4. Desarrollo de placas

Desarrollados los diseños de placas se los puede materializar, el proceso es el siguiente:

- Se imprimen los diseños en hojas de papel termotransferibles en una impresora láser.
- Se recorta el papel y la placa de baquelita con las dimensiones de los diseños.
- Con una plancha a la máxima temperatura se repasa toda la placa para que la tinta se pegue en la baquelita.
- Se remoja el papel en agua para aflojarlo y evitar que se levante la tinta.
- Al tener la placa ya con las pistas impregnadas se la sumerge en una fundita de cloruro férrico en un recipiente de plástico, se agita unos 10 minutos hasta observar que el cobre restante de las pistas desaparezca.
- Se limpia impurezas y resto del ácido con alambre de limpieza con alcohol.
- Se lima partes excedentes de la placa hasta encontrar el tamaño adecuado de cada placa.

- Se verifica conexión entre las pistas y se eliminan puentes creados en el proceso de quemado con el ácido.
- Con pintura de impresora láser con resina se pinta ambas partes de la placa para que tenga un aislante en la placa, se deja secar por lo menos dos horas.
- Se colocan los elementos en sus posiciones de la placa y se procede a soldar, se imprime indicaciones en papel bond y se las pega en las placas.
- Se pasa nuevamente resina sin color para fijar los elementos adecuadamente, junto con la información de cada placa.
- Se realizar pruebas finales de conectividad.

Las placas desarrolladas después del proceso comentado anteriormente se pueden observar en la figura 38.



*Figura 38. PLACAS TERMINADAS
Referencia: Autoría*

3.3.5. Desarrollo de guías

El desarrollo de las guías está enfocadas a la metodología de proyectos donde el estudiante adquiere conocimiento y se propone retos prácticos que pueda solucionar, está enfocada a la configuración, programación y uso de las placas de entrenamiento con las diferentes integraciones que se pueden armar para presentar sistemas complejos. Las guías se encuentran en el Anexo 2 del trabajo de titulación.

3.3.6. Diagramas de flujo de funcionamiento

El diagrama de flujo demuestra el funcionamiento aplicado en clase.



Figura 39. Diagrama de flujo de funcionamiento
Referencia: Autoría

3.3.7. Costo-beneficio

Los costos de todos los materiales utilizados se detallan en la tabla 24.

Tabla 24. Lista de materiales

Cant.	Material	P. Unitario	TOTAL
2	Baquelita tamaño A4	10	20
4	Ácido cloruro férrico	0.50	2
1	Rollo de estaño	4	4
1	Chupa suelda	2	2
1	Cautín 40W	15	15
1	Sensor DTH11	15	15
1	Acelerómetro AXL345	10	10
1	Sensor PIR	5	5
1	Sensor Ultrasónico	5	5
1	LCD 16X2	8	8
1	Teclado	7	7
1	Módulo Bluetooth	15	15
6	Borneras	1	6
1	LM05	3	3
1	Capacitor de 3300 uF	0.5	0.5
6	Diodos 1N4007	0.10	0.6

1	LM7805	1	1
1	LM7812	1	1
2	Capacitores de 10 uF	0.10	0.20
2	Display ánodo común	0.8	1.6
2	Decodificador 7447	1	2
1	Relé	1	1
5	Filas de espadines macho	1.25	6.25
1	Fila de espadines hembra	1.25	1.25
2	Chicharras	1	2
4	Potenciómetros de precisión	1	4
2	Dipswtich de 2 posiciones	0.5	1
3	Pulsadores de 2 pines	0.25	0.75
3	TIP31C	1.25	3.75
25	Resistencias ¼ de Watio	0.10	2.5
1	Marcador de CD	1	1
1	Caja de plástico	10	10
8	Impresiones en papel termotransferible	0.5	4
1	Lata de pintura	4	4

TOTAL 165.4

Referencia: Autoría

El costo de individual de cada placa con los diferentes materiales se detalla en la tabla 25. Algunas placas vienen en un mismo kit por el hecho estar en la misma temática de estudio, del costo inicial de construcción es alto realizar una placa de cada sistema.

Cabe indicar que algunos precios de materiales son reutilizados del proceso de placas como lo es cobre o el ácido, es un trabajo para el estudiante pueda mejorar sus capacidades de desarrollo de sistemas electrónico y no se plantea un modelo de negocio del trabajo.

El precio de las placas de desarrollo no podía exceder demasiado en relación a la placa Intel Galileo Gen 2 que oscila entre los 115 dólares, con el precio indicado se puede tener estaciones de trabajo a bajo costo ya que no existen gasto por licencias de software.

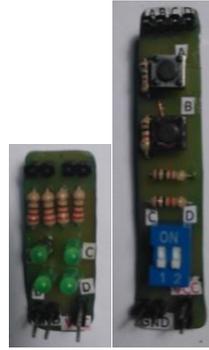
Todos los diseños se encuentran en repositorio de Github para el desarrollo de otras personas.

Tabla 25. Costo por placa

PLACA	COSTO	PLACAS
--------------	--------------	---------------

Pines digitales

10 usd



LEDS

PULSADORES

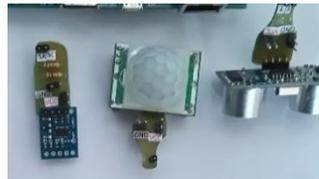
Display

5 usd



Sensores

30 usd

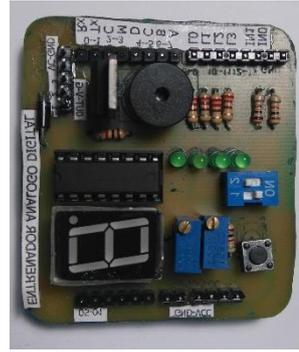


LCD y Teclado

25 usd



Entrenador análogo 20 usd
digital



Pines análogos 3 usd



Motores 3 usd



IOT 50 usd



TOTAL

150

Referencia: Autoría

El costo El costo por impresión de la guía es de 30 dólares de forma individual, existen descuentos por impresiones en volúmenes.

El beneficio para los estudiantes es generar conocimiento a través de una práctica que no demanda demasiados recursos en relación al costo de los materiales que se encuentran en el mercado actual, al contar con un material propio para la carrera de Electrónica y Redes de Comunicación los estudiantes llegan a invertir un máximo de 180 dólares siendo mucho menor el costo de un entrenador propietario,

CAPITULO IV

PRUEBAS

4.1. Pruebas del sistema

4.1.1. Pruebas de placas

Las pruebas de las placas fueron parte del diseño de las guías donde cada una se validó su funcionamiento con los diferentes programas, se detalla las pruebas por cada placa con su respectivo programa, si no funcionaran adecuadamente no se hubiera podido realizar la guía. Un punto importante es el consumo de corriente que necesita las placas, en la tabla 26 se puede observar el consumo individual de cada shield, teniendo en cuenta que por la limitación de pines del Intel Galileo Gen 2 no se pueden ubicar todos los shields para un consumo total, además que dependiendo de las aplicaciones se pueden ir combinando los diferentes consumos de corriente que no deben superar los 800 mA que proporciona la placa (INTEL 2015)

Tabla 26. Consumo de corrientes

PLACA	CORRIENTE
Pines digitales	<p>10 mA por led= 40mA</p> <p>Potencia=2.2v*40 mA= 0.088 W</p>
Display	<p>10 mA por led = 70 mA</p> <p>Potencia= 70 mA * 2.2V= 0.154 W</p>

Sensores

PIR= 350 uA en reposo 10 mA activado

Potencia= 10 mA * 5 V= 0.050 W

HC-SR04 = 15 mA

Potencia= 15mA * 5 V= 0.075 W

ADXL335= 1 Ma

Potencia= 1 mA * 5 V= 0.005 W ADXL335= 1
mA

LCD y Teclado

TECLADO 8 resistencias de 10KΩ a 5v= 40 mA

Potencia= 40 mA * 5v= 0.02 W

LCD = 25 mA * 5v= 0.125W

Entrenador análogo digital

1 display, 4 leds, 2 switches, 1 chicharra = 200mA

Potencia= 200 mA * 5 V= 1 W

Pines análogos

2 potenciómetros de 10KΩ a 5v= 10mA

Potencia= 10 mA * 5 V= 0.05 W

Motores

Posee una conexión a fuente externa para
funcionamiento.

SERVO = 160 mA

IOT Tiene fuente externa y su consumo viene de un ventilador 0,3 A y un foco de 30 Watios. La fuente desarrollada tiene 5v y 12v con 1 amperio de salida.

Referencia: Autoría

No existe problema en el consumo de las placas ya que no exceden los 800 mA.

4.2. Pruebas de guías

Las pruebas de las guías se desarrollaron a diferentes estudiantes de la carrera de Electrónica y Redes de Comunicación, teniendo en cuenta diferentes niveles de programación y desarrollo de sistemas electrónicos, donde se escogieron 3 grupos: los principiantes (estudiantes de primera matricula de la asignatura de sistemas microprocesados), conocimientos medios (estudiantes de primera matricula de la asignatura de sistemas embebidos) y con experiencia (estudiantes tesistas de la carrera con temas afines a electrónica).

En el grupo de los principiantes se encuentran matriculados 21 estudiantes, en el grupo de conocimientos medios 25 estudiantes y los tesistas son 7, el número del universo es muy pequeño para parámetros estadísticos normales, ya que darían como resultado de la muestra el universo en sí, esto dificulta el tiempo de desarrollo ya que se cuenta con una sola guía y un grupo de shields por ello se cambió el error de estimación al 20% y un nivel de confianza del 80% que no son los adecuados en grandes poblaciones, pero si

logra una aprobación del 80% del estudiantado supera al 70% de lo establecido para una correcta evaluación docente, la muestra dió como resultado utilizando la fórmula:

Ecuación 1. Cálculo muestral

$$n = \frac{N\sigma^2 Z^2}{(N-1)e^2 + \sigma^2 Z^2}$$

Donde:

n = el tamaño de la muestra.

N = tamaño de la población.

Desviación estándar de la población que, generalmente cuando no se tiene su valor, suele utilizarse un valor constante de 0,5.

Z = Valor obtenido mediante niveles de confianza. Es un valor constante que, si no se tiene su valor, se lo toma en relación al 95% de confianza equivale a 1,96 (como más usual) o en relación al 99% de confianza equivale 2,58, valor que queda a criterio del investigador.

e = Límite aceptable de error muestral que, generalmente cuando no se tiene su valor, suele utilizarse un valor que varía entre el 1% (0,01) y 9% (0,09), valor que queda a criterio del encuestador.

Principiantes: 8

Medianos: 8

Experiencia: 5

Para las pruebas de las guías se planteó un plan de clases que se encuentra en el anexo 2, donde especifica las habilidades y competencias que debe adquirir el estudiante en relación a los objetivos planteados, se tomará el plan de clase en dos grupos, el primero solo tendrá información recolectada de las hojas de datos y libros sobre programación, además se contar con los materiales sin conexión y un protoboard junto con el galileo para que pueda realizar las diferentes conexiones, el otro grupo se le asignará la guía y los shields para que pueda trabajar; con este proceso al final de una clase de 2 horas, que se dividirá en secciones de 20 minutos de teoría y 40 minutos de práctica, la temática se escoge al azar en relación de los contenidos de la materia teniendo en cuenta el nivel de estudiante, por tiempo no se realizaron con todas las placas, según la dirección del docente se determinó grupos entre las placas donde son considerados entre el nivel de dificultad.

Se evaluarán 3 aspectos en cada grupo, una prueba práctica para resolver un reto propuesto, el tiempo de desarrollo y la facilidad de manejo del sistema en relación a software y hardware. Los temas se dieron en una función aleatoria que indicó lo siguiente:

Principiantes: Salidas y entradas digitales.

Mediano: Visualizadores.

Experiencia: salidas y entradas análogas.

El tiempo se medirá en minutos y el nivel de software y hardware en bajo, bueno y alto, en relación a un plan de clase propuesto.



UNIVERSIDAD TÉCNICA DEL NORTE

FICA-CIERCOM

PLAN DE CLASE

DATOS INFORMATIVOS

Nombre del Facilitador: Samantha Mesa

Tema: Aplicación y desarrollo de sistemas embebidos utilizando las tarjetas de entrenamiento

Objetivo de la clase:

Realizar prácticas sobre electrónica programada con Intel Galileo Gen 2 utilizando las tarjetas de entrenamiento con los periféricos de puertos como entrada/salida, comunicación serial e i2c, visualizadores y salidas a motores para reducir el tiempo de armado del circuito enfocando dicho tiempo en el aprendizaje y desarrollo del sistema.

¿Qué van aprender los estudiantes?	¿Cómo van aprender?	¿Con qué?	¿Cómo se va evaluar esos aprendizajes?	¿Qué va evaluarse?
DESTREZAS CON CRITERIO DE DESEMPEÑO	ACTIVIDADES	RECURSOS	TIPOS DE INSTRUMENTOS	INDICADORES DE EVALUACIÓN

Criterios en programación	Prácticas con las tarjetas de entrenamiento revisando su funcionamiento.	Guía para tarjetas de entrenamiento.	Evaluar al estudiante parte práctica aplicando varios shields.	Conocimiento teórico sobre periféricos que se presentan en la guía.
Criterios de periféricos en sistemas electrónicos	Retos con las tarjetas de entrenamiento enfocando aplicaciones reales.	Proyector	Realizar una evaluación de aprendizajes de la materia.	Conocimientos sobre programación.
Desarrollo de sistemas embebidos	Adquisición de parámetros del usuario para aplicar.	Computadora Puntero		Resultado del sistema electrónico.
Documentación de sistemas embebidos		Shields		

OBSERVACIONES: El material entregado no es personalizado y se debe hacer grupos de trabajo, la guía es entregada digitalmente.

FACILITADOR (A)

4.3. Resultados

4.3.1. Estudiantes principiantes sin guías

Tabla 27. Resultados estudiantes principiantes sin guías

Estudiante	Tiempo	Software	Hardware
#1	38 min	Bueno	Bajo
#2	32 min	Bueno	Bajo
#3	41min	Bueno	Bueno
#4	35 min	Alto	Bajo

Referencia: Autoría

4.3.2. Estudiantes principiantes con guías

Tabla 238. Resultados estudiantes principiantes con guías

Estudiante	Tiempo	Software	Hardware
#1	21 min	Alto	Alto
#2	23 min	Bueno	Alto
#3	25 min	Bueno	Alto
#4	18 min	Alto	Alto

Referencia: Autoría

4.3.3. Estudiantes con conocimientos medios sin guías

Tabla 29. Resultados estudiantes con conocimientos medios sin guías

Estudiante	Tiempo	Software	Hardware
#1	38 min	Bueno	Bueno
#2	32 min	Bueno	Bueno
#3	41min	Bueno	Bueno
#4	35 min	Alto	Bajo

Referencia: Autoría

4.3.4. Estudiantes con conocimientos medios con guías

Tabla 30. Resultados estudiantes con conocimientos medios con guías

Estudiante	Tiempo	Software	Hardware
#1	21 min	Alto	Alto
#2	23 min	Bueno	Alto
#3	25 min	Bueno	Alto
#4	18 min	Alto	Alto

Referencia: Autoría

4.3.5. Estudiantes con experiencia sin guías

Tabla 31. Resultados estudiantes con experiencia sin guías

Estudiante	Tiempo	Software	Hardware
#1	35 min	Alto	Alto
#2	32 min	Alto	Bueno
#3	32 min	Alto	Alto
#4	38 min	Alto	Bueno

Referencia: Autoría

4.3.6. Estudiantes con experiencia con guías

Tabla 242. Resultados estudiantes con experiencia con guías

Estudiante	Tiempo	Software	Hardware
#1	21 min	Alto	Alto
#2	20 min	Alto	Alto
#3	20 min	Alto	Alto
#4	20 min	Alto	Alto

Referencia: Autoría

4.4. Análisis de resultados

Los resultados que se muestran en la tabla 27 y 28 indicaron que los estudiantes principiantes llegaron casi el tiempo justo de la práctica ya que en su armado se demoraron bastante por la colocación de los cables de conexión, no fueron ordenados en su armado.

Los estudiantes con medianos conocimientos como se muestra en las tablas 29 y 30 mantienen sus problemas, aunque en su software no tuvieron problemas al usar programas de simulación antes de armar, el armado no fue complicado y pudieron realizarlos sin problemas, si hubo un tiempo considerable en terminación del sistema. Los estudiantes con experiencia como se muestra en la tabla 31 y 32 fueron los más complicados en el armado ya que al utilizar una lcd su tiempo fue muy largo en el armado y algunos tuvieron que realizar pruebas de conexión y observar en las hojas de datos para verificar su funcionamiento, esto limitó su tiempo en el desarrollo de su sistema. Como resultados de las pruebas determinaron que los estudiantes principiantes que cuentan con la guía fue una ayuda considerablemente al desarrollo de sistemas electrónicos al enfocarse netamente en el funcionamiento y su software, aunque nos indicaron en la hora de clase que el armado de los circuitos les permite conocer de mejor forma a los elementos electrónicos y sensores para dar una mejor solución, el objetivo de la guía es contar con una herramienta que mejore el tiempo de construcción de hardware para enfocarse en la solución integra de un sistema electrónico, al reducir el tiempo en una práctica se puede contar con una retroalimentación muy rápida y corrección de errores en un espacio en la hora de clase que antes no se contaba. Al brindar los diseños de las placas a los estudiantes las pueden realizar ellos mismos, utilizarlas y dar mejoras al proyecto en relación a su experiencia, comparten información entre ellos y permite generar conocimiento, en la actualidad lo llaman Cultura Maker. (Santiago Arango Sarmiento, 2016)

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Se diseñó e implementó las tarjetas de entrenamiento para mejorar el aprendizaje de electrónica utilizando la placa Intel Galileo G2 basada en la plataforma Arduino.
- La descripción técnica de los elementos electrónicos y el estudio de los sensores a utilizar en este proyecto fueron aportes importantes para el diseño de las placas tanto en dimensión como en funcionamiento.
- El aprendizaje puede ser llevado a cabo de diferentes formas, al utilizar un método orientado a proyectos y generando un conocimiento que no perjudica a un modelo actual de aprendizaje, lo hace viable y con expectativas de ganancia para quien lo desarrolla.
- Al realizar el análisis de resultados se puede ver las mejoras que se realizaron en algunos estudiantes como son: reducción en el tiempo de armado de acuerdo al reto que se propuso, más orden en la ubicación de las placas ya que al no usar cables se tiene mejor estética, desarrollo de software sin tanta dificultad ya que la Plataforma Arduino permite realizar cambios según se requiera, adaptación adecuada a esta nueva metodología de aprendizaje enfocada en mejorar el sistema de educación.

5.2. RECOMENDACIONES

- Promover en los estudiantes el interés en este tipo de proyectos, los beneficios no son sólo para ellos sino también para una sociedad en la que la tecnología avanza muy rápido, de esta forma se tendría una base, herramienta o apoyo para que el desarrollo que se realice posteriormente se pueda extender hacia aplicaciones más complejas.
- Se recomienda tener en cuenta que es lo que se quiere realizar en cada placa de entrenamiento, de acuerdo a eso se puede determinar que materiales utilizar.
- Realizar un correcto diseño de las placas, para ello en algunos casos se pueden realizar simulaciones previas, esto ayuda a que si existen errores se los corrija antes de que se impriman los circuitos y que cuando ya estén soldados los materiales electrónicos las placas funcionen correctamente.
- Realizar una distribución de pines que permita la escalabilidad de las placas de entrenamiento siendo esta una de las aplicaciones importantes de este proyecto al poder combinar varias placas juntas.

BIBLIOGRAFÍA

- Arduino. (7 de Agosto de 2015). *Arduino*. Obtenido de <https://www.arduino.cc>
- Bolton, W. (s.f.). *INGENIERÍA DE CONTROL*. <http://hdl.handle.net/123456789/4874>.
- Boylestad, R. (1997). *ELECTRÓNICA: TEORÍA DE CIRCUITOS*. Naucalpan de Juárez: Hall Hispanoamericana S.A.
- Cabrera, A. &. (2014). *Universidad del Azuay*. Obtenido de <http://dspace.uazuay.edu.ec/bitstream/datos/3610/1/10292.pdf>
- CadSoft. (2011). *EAGLE*. Obtenido de <http://www.cadsoftusa.com/>
- Cairo, B. (2005). *Metodología de la Programación: Algoritmos, diagramas de flujo y programas*. México: Alfaomega 464 p.
- Chiesa, L., & Bassi, G. (2010). *Workshop Sistemas embebidos*. Argentina: FIUBA.
- CISCO. (Marzo de 2014). *NETACAD*. Obtenido de <http://www.netacad.com/courses/internetforeverything>
- Cogdell, J. (2000). *Fundamentos de electrónica*. Hall, 423 páginas.
- Davini, M. C. (2008). *MÉTODOS DE ENSEÑANZA : Didáctica para maestros y profesoras*. Buenos Aires: Santillana 2008.
- DEBIAN. (17 de Noviembre de 2014). *EMDEBIAN*. Obtenido de <http://www.emdebian.org/>
- Delacote, G. (1997). *ENSEÑAR Y APRENDER CON NUEVOS MÉTODOS*. Barcelona: Gedisa, S.A. 251p.
- Developers, T. D. (2014). *Dia Installer*. Obtenido de <http://dia-installer.de/doc/index.html>
- Electronics, D.-k. (4 de Agosto de 2015). *Digi-key Electronics*. Obtenido de <http://www.digikey.com/>
- Fernández, C. (2009). *linux, Desarrollo de sistemas embebidos basados en linux*.
- Fritzing. (3 de Abril de 2015). *Fritzing*. Obtenido de <http://fritzing.org/home/>
- IEEE. (1998). *IEEE-STD-830-1998*. IEEE.
- IEEE. (2011). *IEEE 29148-2011*. IEEE.

- INTEL. (7 de Agosto de 2015). *Placas de desarrollo Intel Galileo Gen 2*. Obtenido de <http://www.intel.la/content/www/xl/es/do-it-yourself/galileo-maker-quark-board.html>
- Labcenter. (5015). *ISIS PROTEUS*. Obtenido de <http://www.labcenter.com/>
- López, J. (2006). *Electrónica analógica*. Valencia: Hall.
- Makezine. (7 de Agosto de 2015). *Makezine*. Obtenido de <http://makezine.com/>
- Micropik. (2014). *Micropik*. Obtenido de HCSR04.pdf: <http://www.micropik.com/PDF/HCSR04.pdf>
- Ramon, M. C. (2014). *Intel Galileo e Intel Galileo Gen 2 API Features and Arduino Projects for linux Programmers*. Apress.
- Robinson, S. K. (Dirección). (2008). *La educación mata la creatividad* [Película].
- Rodriguez, P. C. (2001). *COMPONENTES ELECTRÓNICOS*. Buenos Aires: LIBRERÍA Y EDITORIAL ALSINA.
- Ubuntu. (30 de Abril de 2009). *Wiki Ubuntu*. Obtenido de <https://wiki.ubuntu.com/EmbeddedUbuntu/>
- Vahid, F., & Tony, G. (1999). *EMbedded System Design: A unified Hardware/Sof5ware Approach*. California: University of California Department of Computer Science and Engineering.
- YOCTO. (2013). *YoctoProject*. Obtenido de <https://www.yoctoproject.org>

ANEXOS

ANEXO 2



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

GUÍA DE PRÁCTICAS DE LAS PLACAS DE ENTRENAMIENTO PARA MEJORAR EL APRENDIZAJE DE ELECTRÓNICA UTILIZANDO INTEL GALILEO G2 BASADO EN LA PLATAFORMA ARDUINO

DESARROLLADO POR SAMANTHA MESA

Contenido

GUÍAS PRÁCTICAS PARA PLACAS DE ENTRENAMIENTO	131
INTRODUCCIÓN DE SISTEMAS ELECTRÓNICOS	131
MATERIALES ELECTRÓNICOS	132
SOFTWARE DE DESARROLLO	134
GUIAS DE PRÁCTICA.....	135
GUÍA # 1: Puertos como Salida	136
GUÍA # 2: Puertos como Entrada.....	140
GUÍA # 3: DISPLAY	144
GUÍA # 4 Entradas y Salidas Análogas	151
GUÍA # 5: Comunicación Serial	153
Guía # 6: LCD	160
Guía # 7: Teclado	162
Guía # 8: Sensor PIR	168
Guía # 9: Sensor Ultrasónico.....	172
Guía # 10: Acelerómetro	176
Guía # 11: Bluetooth	181

GUÍAS PRÁCTICAS PARA PLACAS DE ENTRENAMIENTO

INTRODUCCIÓN DE SISTEMAS ELECTRÓNICOS

SISTEMA EMBEBIDO: Un sistema embebido es un sistema electrónico que cumple solo determinadas funciones, donde contiene las siguientes partes:

Entradas: Son los datos que traemos de nuestro entorno, de la vida real para poder realizar una actividad en nuestro Arduino, para ello debemos tener en cuenta que existen dos formas de adquirir datos de nuestro entorno: la primera es de forma análoga, esto indica que podemos adquirir información con miles de valores y algunos de ellos dependen de la apreciación de cada persona. Un ejemplo es el velocímetro de la Figura 1. Donde indica una velocidad de aproximadamente 65MPH, muchas personas pueden decir que es una velocidad de 64,8 y estarían muy cerca de lo correcto, otras 64,89 y sigue siendo un criterio adecuado.



Figura 1. Datos análogos

La segunda forma de adquirir datos es de forma digital, en este tipo de datos solo tienen dos valores establecidos, verdadero (1 lógico) y falso (0 lógico), de esta forma Arduino acepta instrucciones, un ejemplo es el foco de nuestros hogares, solo tiene dos opciones se encuentra encendido o apagado, otro es el celular donde puede estar encendido o apagado.

Unidad Central de Procesamiento: Más conocida como la CPU, es donde una vez obtenidos los datos del medio son procesados para dar cierta respuesta a los mismos. Es aquí donde funciona Arduino ya que al ser el cerebro del sistema donde es necesario programarlo mediante ciclos o variables de programación de manera que pueda actuar en función de los eventos censados.

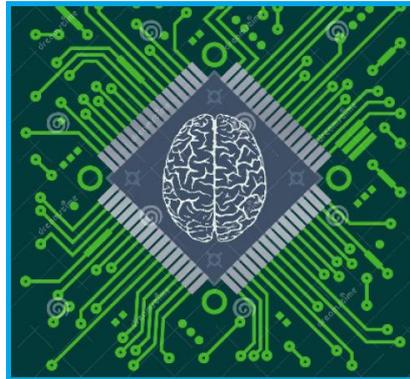


Figura 2. Unidad Central de Procesamiento

MATERIALES ELECTRÓNICOS

Ley de Ohm: La única ley que tomaremos en cuenta es la ley de Ohm que se refiere al diseño de un circuito y hace relación entre voltaje, corriente y resistencia.

El voltaje es la fuerza que empuja las cargas por un medio de transmisión como lo es un cable. La corriente es flujo de cargas que circulan por el medio y finalmente la resistencia es la oposición al paso de corriente, esta oposición ayuda para controlar el flujo de cargas y evitar sobre carga y daños a los elementos.

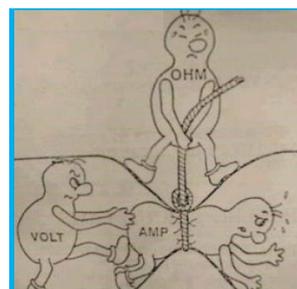


Figura 3. Ley de Ohm

LISTA DE ELEMENTOS COMUNES				
ELEMENTO	SIMBOLOGÍA	CARACTERÍSTICA	FORMA REAL	POLARIDAD
Resistencia		Limita el paso de corriente, tienen una gran cantidad de valores		No posee.
Potenciómetro		Es una resistencia variable.		No posee.
Leds		Es capaz de emitir luz. Puede ser diferente color.		En simbología la flecha indica el sentido de la corriente, la parte izquierda es el positivo, más bien conocido como ánodo, la parte negativa tiene la línea vertical y se llama cátodo. En la realidad el ánodo tiene la patita más larga
Diodo		Solo permite el paso de corriente en una sola dirección.		En simbología la flecha indica el sentido de la corriente, la parte izquierda es el positivo, más bien conocido como ánodo, la parte negativa tiene la línea vertical y se llama cátodo. En la realidad el cátodo tiene un franja blanca
Motor		Al circular corriente por los terminales del motor genera movimiento.		Depende de la polaridad tendrá el giro del motor.

Figura 4. Elementos electrónicos

Fuente (Vcc)		Proporcionar voltaje		Solo es Vcc
Tierra (Gnd)		Hace que circule la corriente al cerrar un circuito.		Solo es Gnd
Switch		Un interruptor pequeño		No posee.
Pulsador		Interruptor intermitente		No posee.

Figura 5. Elementos electrónicos

Los elementos mostrados son los más usados y son los que principalmente acompañan a todo sistema embebido, en otro apartado veremos el uso de sensores con la interacción de estos elementos, hay que considerar que los elementos básicos no requieren el uso de hojas de datos específicas.

SOFTWARE DE DESARROLLO

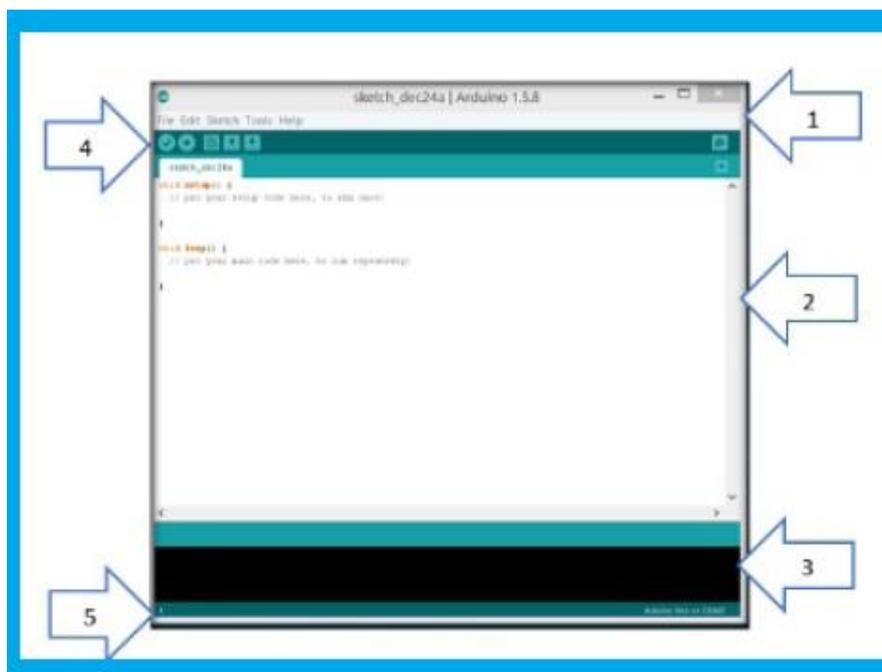


Figura 6. IDE de Arduino

IDE DE ARDUINO

Inicio de Programa: Una vez que hayamos entrado a la pantalla principal de Arduino es necesario conocer ciertas partes importantes para un mejor manejo del programa y no cometer errores de software al cargar un archivo equivocado.

1. Barra de Navegación: Permite al usuario acceder a programas, ejemplos y principalmente seleccionar el Arduino a usar y el puerto de acceso.
2. Cuadro de trabajo: Donde programamos al Arduino.
3. Cuadro de compilación: Nos indica el proceso de compilación y la ubicación de archivo temporal.hex
4. Barra de Acceso: Permite compilar al Arduino y cargar el programa.
5. Barra de Ubicación: Observaremos el tipo de Arduino conectado y el puerto de comunicación.

PROGRAMACIÓN:

1. **Librerías:** Son líneas de código ya construidas para el uso de ciertos aspectos de Arduino para una mayor facilidad y se llaman a estos subprogramas llamados métodos o subrutinas.
2. **Variables:** Es un espacio de memoria donde se almacenan datos, de forma temporal o fija, dependiendo del tipo de dato se debe elegir la variable a usar, además de ellos es necesario primero declarar la variable con un nombre que la represente, este nombre no puede albergar otra variable pero si puede tener algunos datos por escritura y lectura del mismo. Una consideración extra es que el tipo de variable debe ser según el tamaño de dato a usar.
3. **Void Setup:** Es el lugar donde configuramos al Arduino sobre que pines vamos a usar y cómo lo vamos a usar, en cada apartado de programación observamos la forma de configurar correctamente.
4. **Void Loop:** Es un espacio de programa que se repetirá infinitas veces hasta que alguna variable cambie de estado o se desconecte de la fuente, es donde irá el programa que dará el accionamiento al Arduino. Cada línea de código debe terminar con punto y coma (;), indicará al Arduino que hasta ese punto acaba la sentencia de programación para leer la siguiente, la lectura es secuencial, es decir el Arduino lee línea por línea en un orden de primera a última, cuando acaba de leer volverá al principio a leer nuevamente.

GUÍA # 1: Puertos como Salida

Tema: Puertos como Salida

Objetivo General:

Aprender el manejo de salidas digitales en la Placa Intel Galileo G2 para utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender a conectar las salidas digitales, en este caso son leds.
2. Utilizar las sintaxis para declaración de pines correspondientes.
3. Probar el funcionamiento de los dispositivos conectados con un programa básico.



Figura 7. Salidas digitales

Preparatorio:

1. ¿Qué son las salidas digitales?
2. ¿En qué pines de la Placa Intel Galileo G2 se pueden conectar las salidas digitales?
3. ¿Cuál es la sintaxis para declarar los pines como salidas?
4. ¿Es necesaria alguna librería para declarar pines como salida?
5. Enumere tres ejemplos de salidas digitales.

Desarrollo de la práctica

Aplicación de placa para puertos como salida se debe conectar en los pines 6, 7, 8 y 9.

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Placa de entrenamiento de salidas digitales
4. Cable USB
5. Cable macho-hembra para polarizar.

Proceso

Hardware:

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

1. La placa Intel Galileo debe estar alimentada y conectada a la PC.
2. Para esta práctica sólo se necesita conectar los leds a GND.
3. Esta placa de entrenamiento está diseñada para utilizar el espacio que hay entre el pin 7 y el pin 8 deberá conectarse en esos pines y posteriormente polarizar la placa con un cable macho-hembra.

Software:

1. Se debe tener instalado el IDE de Arduino
2. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
3. Declarar los pines como e/s en el **void setup**
4. Realizar la programación deseada en el **void loop**
5. Compilar
6. Cargar el programa al Intel Galileo G2
7. Verificar el funcionamiento

Diagrama de Conexión:

El diagrama muestra un Arduino Uno, pero como la distribución de pines es la misma, se puede ver la conexión que debe tener el ejemplo.

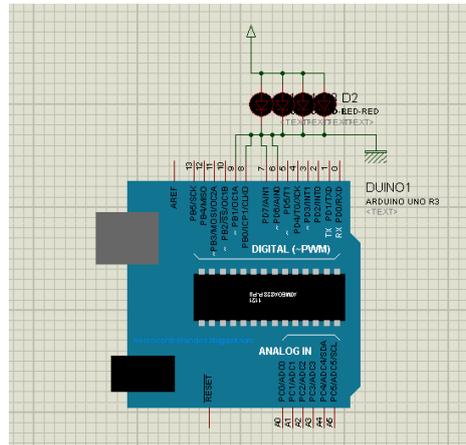


Figura 8. Diagrama de conexión de salidas digitales

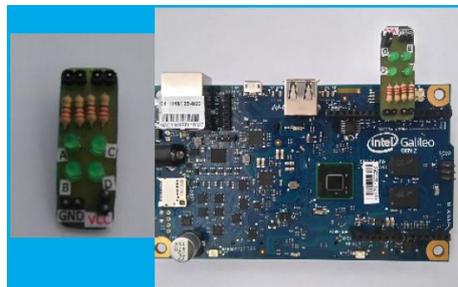


Figura 9. Salidas digitales

Programación

Código Fuente:

```
int A=6; //variable para led A
```

```
int B=7; //variable para led B
```

```
int C=8; //variable para led C
```

```
int D =9;//variable para led D

void setup()
{
pinMode(A, OUTPUT); // Declarar pin A como salida
pinMode(B,OUTPUT); // Declarar pin B como salida
pinMode(C,OUTPUT); // Declarar pin C como salida
pinMode(D,OUTPUT); // Declarar pin D como salida
}

void loop() {
digitalWrite(A, HIGH); // Encender led A
digitalWrite(B, HIGH); // Encender led B
digitalWrite(C, HIGH); // Encender led C
digitalWrite(D, HIGH); // Encender led D
delay (3000); // tiempo de encendido

digitalWrite(A, LOW); // Apagar led A
digitalWrite(B, LOW); // Apagar led B
digitalWrite(C,LOW); // Apagar led C
digitalWrite(D,LOW); // Apagar led D
delay (3000);// tiempo de apagado

digitalWrite(A, HIGH); // Encender led A
digitalWrite(B, HIGH); // Encender led B
delay (3000); // tiempo de encendido

digitalWrite(C,HIGH); // Encender led C
digitalWrite(D, HIGH); // Encender led D
```

```
delay (3000); // tiempo de encendido
```

```
digitalWrite(A,LOW); // Apagar led A
```

```
digitalWrite(B, LOW); // Apagar led B
```

```
delay (3000);// tiempo de apagado
```

```
digitalWrite(C,LOW); // Apagar led C
```

```
digitalWrite(D,LOW); // Apagar led D
```

```
delay (3000);// tiempo de retardo
```

Resultado:

Esta aplicación de puertos como salida permite observar un juego de luces de leds que se han conectado en los pines 6, 7, 8 y 9.

Reto:

Realizar un juego de luces en dónde se encienda uno por uno y se paguen uno por uno, hasta que se quede todo apagado.

Análisis de resultados:**Conclusiones:****Recomendaciones:****Bibliografía:**

GUÍA # 2: Puertos como Entrada

Tema: Puertos como Entrada

Objetivo General:

Aprender el manejo de entradas digitales en la Placa Intel Galileo G2 para utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender a conectar las entradas digitales, en este caso son pulsadores o un dip switch.
2. Utilizar la sintaxis para declaración de pines correspondientes como entradas
3. Combinar entradas y salidas digitales.
4. Probar el funcionamiento de los dispositivos conectados con un programa básico.

Preparatorio:

1. ¿Qué son las entradas digitales?
2. ¿En qué pines de la Placa Intel Galileo G2 se pueden conectar las entradas digitales?
3. ¿Cuál es la sintaxis para declarar los pines como entradas?
4. ¿Es necesaria alguna librería para declarar pines como entradas?
5. Enumere tres ejemplos de entradas digitales.

Desarrollo de la práctica

Encender leds que reciban la orden desde un pulsador como entrada para encenderse o apagarse, se debe conectar en los pines 6, 7, 8 y 9 como salida y como entrada los pines 2, 3, 4 y 5.

Materiales:

1. Placa Intel Galileo G2
2. PC

3. Placa de entrenamiento de entradas digitales en este caso pulsadores
4. Cable USB
5. Cables macho-hembra para polarizar.

Proceso

Hardware:

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

1. La placa Intel Galileo debe estar alimentada y conectada a la PC.
2. Para esta práctica sólo se necesita conectar los leds a GND y la placa de los pulsadores si debe polarizarse a VCC y GND
3. La placa de entrenamiento # 1 está diseñada para utilizar el espacio que hay entre el pin 7 y el pin 8 deberá conectarse en esos pines y posteriormente polarizar la placa con un cable macho-hembra.
4. La placa de entrenamiento # 2 se puede conectar en cualquier pin desde el 0 al 13 pero como 6, 7, 8 y 9 ya están ocupados se puede ocupar el 2, 3, 4 y 5.

Software:

8. Se debe tener instalado el IDE de Arduino
9. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
10. Declarar los pines como e/s en el **void setup**
11. Realizar la programación deseada en el **void loop**
12. Compilar
13. Cargar el programa al Intel Galileo G2
14. Verificar el funcionamiento

Diagrama de Conexión:

El diagrama muestra un Arduino Uno, pero como la distribución de pines es la misma, se puede ver la conexión que debe tener el ejemplo.

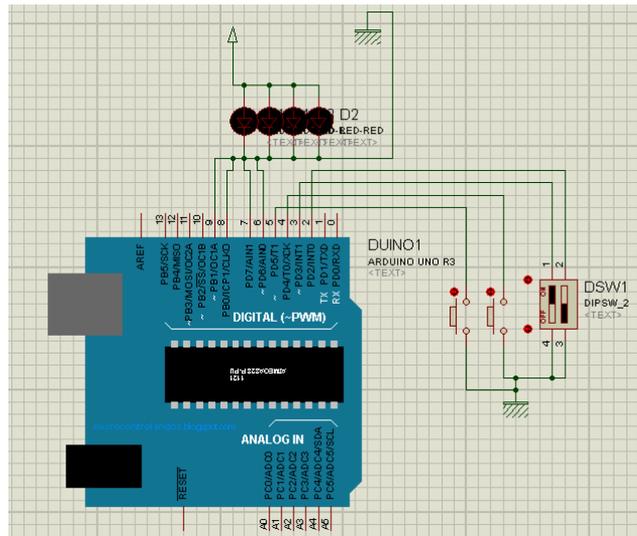


Figura 10. Diagrama de conexión de entradas y salidas digitales

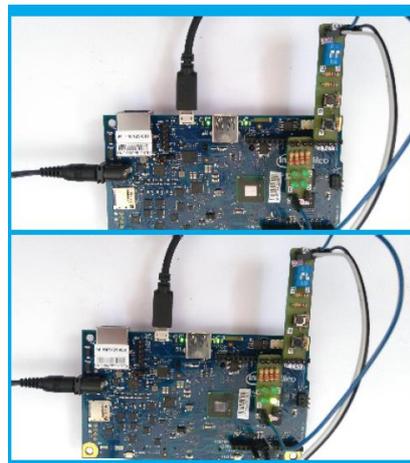


Figura 11. Entradas y salidas digitales

Programación

Código Fuente:

//Solo se usa un led.

int A=6; // Variable para led A

int pulsadorA=2;// variable para pulsador A

```
void setup()
{
pinMode(A,OUTPUT);// led A declarado como salida
pinMode(pulsadorA, INPUT); // pulsadorA declarado como
entrada
}
void loop()
{
if(digitalRead(pulsadorA)==HIGH)
{ // condicion de
lectura de dato
digitalWrite(A,HIGH); // si recibe 1 logico el led se
enciende
}
else
{
digitalWrite(A,LOW); // caso contrario el led se apaga.
}
}
```

Resultado:

Esta aplicación de puertos como entrada permite observar leds que reciban la orden desde un pulsador como entrada para encenderse o apagarse.

Reto:

Cambiar el programa para usar el pulsador y el dip switch restante y verificar su funcionamiento.

Análisis de resultados:

Conclusiones:

Recomendaciones:

Bibliografía:

GUÍA # 3: DISPLAY

Tema: Display

Objetivo General:

Aprender el manejo de salidas digitales en el display utilizando la Placa Intel Galileo G2 para utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender a conectar el display.
2. Utilizar la sintaxis para declaración de pines correspondientes como salidas
3. Probar el funcionamiento del dispositivo conectado con un programa básico que tenga varias combinaciones de software.

Preparatorio:

1. ¿Qué es un display?
2. ¿Qué es lo que hace un decodificador al conectarse al display?
3. ¿Cuántos tipos de display existen?
4. ¿En qué pines se puede conectar el display?
5. Enumere tres aplicaciones que se pueden hacer con el display.

Desarrollo de la práctica

Utilizando el display del entrenador de debe mostrar el número 1 y el 9 respectivamente.

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Placa de entrenamiento de display
4. Cable USB
5. Cables macho-hembra para polarizar.

Proceso

Hardware:

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

1. La placa Intel Galileo debe estar alimentada y conectada a la PC.
2. Para esta práctica sólo se necesita conectar los leds a GND y la placa de los pulsadores si debe polarizarse a VCC y GND
3. La placa de entrenamiento # 3 está diseñada para utilizar el espacio que hay entre el pin 7 y el pin 8 deberá conectarse en esos pines y posteriormente polarizar la placa con un cable macho-hembra.

Software:

1. Se debe tener instalado el IDE de Arduino
2. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
3. Declarar los pines como e/s en el **void setup**
4. Realizar la programación deseada en el **void loop**
5. Compilar
6. Cargar el programa al Intel Galileo G2
7. Verificar el funcionamiento

Diagrama de Conexión:

La placa entrenadora del display ya viene con su respectivo decodificador por lo que cuenta con solo cuatro pines que deben ser conectados en los pines 6, 7, 8 y 9.

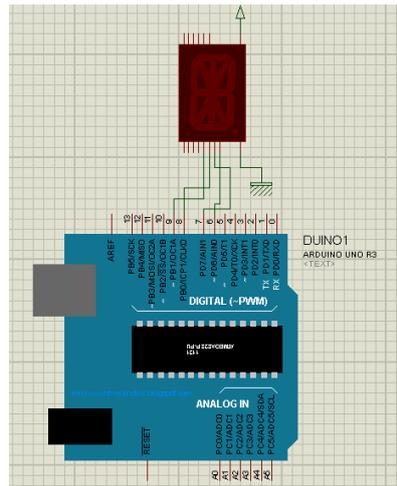


Figura 12. Diagrama de conexión del display

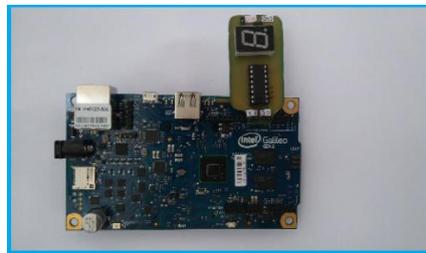


Figura 13. Display

Programación

Código Fuente:

```
int A=8; //SALIDA A
int B=9; //SALIDA B
int C=6; //SALIDA C
int D =7;//SALIDA D
```

```
void setup()
```

```
{  
pinMode(A,OUTPUT); // Declarar pin A como salida  
pinMode(B,OUTPUT); // Declarar pin B como salida  
pinMode(C,OUTPUT); // Declarar pin C como salida  
pinMode(D,OUTPUT); // Declarar pin D como salida  
}  
  
void loop()
```

```
{  
  
//numero #1  
  
digitalWrite(A,HIGH); // En alto A  
digitalWrite(B,LOW); // En bajo B  
digitalWrite(C,LOW); // En alto C  
digitalWrite(D,LOW); // En alto D  
delay(300);
```

```
//numero #9  
  
digitalWrite(A,HIGH); // En alto A  
digitalWrite(B,LOW); // En bajo B  
digitalWrite(C,LOW); // En bajo C  
digitalWrite(D,HIGH); // En alto D  
delay(300);  
}
```

Resultado:

Se visualiza el número 1 y el 9 respectivamente.

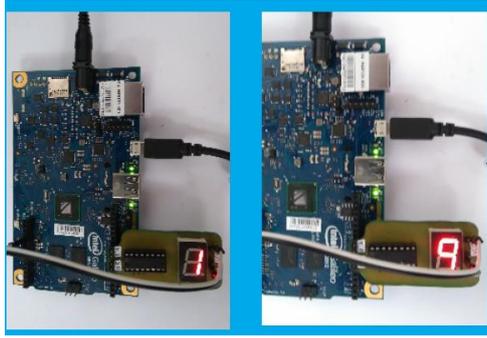


Figura 14. Funcionamiento del display

Reto:

Realizar combinaciones en la programación para que se visualicen todos los números del 0 al 9 de forma ascendente y luego de forma descendente.

Análisis de Resultados:

Conclusiones:

Recomendaciones:

Bibliografía:

GUÍA # 4 Entradas y Salidas Análogas

Tema: Entradas y salidas análogas

Objetivo General:

Manejar entradas y salidas análogas utilizando la Placa Intel Galileo G2 para utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender a conectar entradas y salidas análogas.
2. Utilizar las sintaxis para declaración de pines correspondientes.
3. Probar el funcionamiento de los dispositivos conectados con un programa básico.

Preparatorio:

1. ¿Qué son las salidas y entradas análogas?
2. ¿Cuál es la sintaxis que se utiliza para las salidas y entradas análogas?
3. ¿En qué pines se pueden conectar las salidas análogas?
4. ¿Cuál es la forma de declarar una variable para utilizar un motor a DC?
5. Enumere tres aplicaciones que se pueden hacer con entradas y salidas análogas.

Desarrollo de la práctica

Controlar la velocidad de un motor mediante el conversor del Intel Galileo, si sobrepasa la velocidad de 250 se activa una alarma.

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Placa de entrenamiento de salidas y entradas análogas en este caso potenciómetros.
4. Motor
5. Placa de entrenamiento de la chicharra
6. Cable USB
7. Cables macho-hembra para polarizar.

Proceso

Hardware:

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

4. La placa Intel Galileo debe estar alimentada y conectada a la PC.
5. Para esta práctica sólo se necesita conectar los leds a GND y la placa de los pulsadores si debe polarizarse a VCC y GND
6. La placa de entrenamiento de entradas y salidas análogas puede conectarse en los pines que están al frente de los que se venían utilizando y tienen la siguiente numeración A0, A1, A2, A3, A4, A5; de esta manera se puede utilizar cualquier de estos canales en este caso se ha utilizado A4 y A5.
7. Conectar el Motor.
8. Conectar la placa de la chicharra.
9. Posteriormente polarizar las placas con cables macho-hembra.

Software:

8. Se debe tener instalado el IDE de Arduino
9. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
10. Declarar los pines como e/s y las librerías necesarias en el **void setup**
11. Realizar la programación deseada en el **void loop**
12. Compilar
13. Cargar el programa al Intel Galileo G2
14. Verificar el funcionamiento

Diagrama de Conexión:

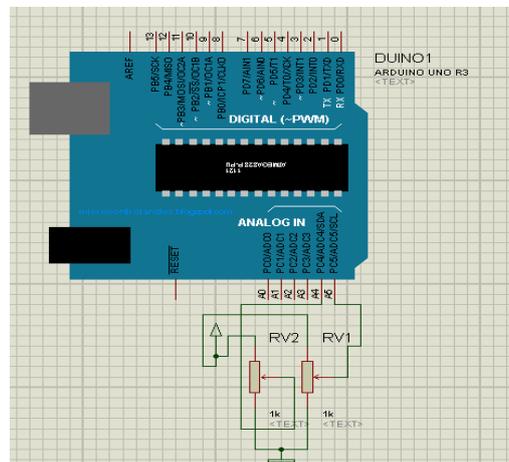


Figura 15. Diagrama de conexión de entradas analógicas

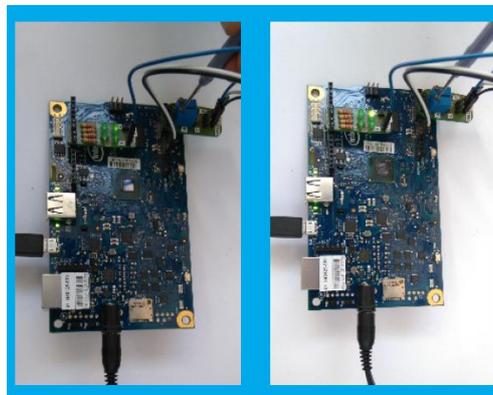


Figura 16. Entradas y salidas analógicas

Programación

Código Fuente:

// Aplicación de Conversores Análogos Digitales en el canal A0 y el Motor en DC pin 9

```
int motor =9;// Variable para el motor

int chicharra=11; // Variable para la chicharra

int conversor; // variable de almacenamiento

void setup()

{

pinMode(motor,OUTPUT); // pin para el motor declarado como salida

pinMode(chicharra,OUTPUT); // Pin de la chicharra como salida

//los conversores de entrada no es necesario declararlos.

}

void loop()

{

conversor= analogRead(0); // recepcion de dato analogo

analogWrite(motor, conversor/4); //conversor de 1023 dividido para 255

if(conversor/4>200){ // restriccion de advertencia de velocidad

digitalWrite(chicharra,HIGH); // encendido de chicharra

delay(100); // tiempo de encendido

digitalWrite(chicharra,LOW); // apagado de chicharra

delay(100); //tiempo de apagado

}

delay(20); //tiempo de estabilizados de velocidad.

}
```

Resultado:

Se ha controlado la velocidad de un motor mediante el conversor del Intel Galileo, cuando sobrepasa la velocidad de 250 suena la alarma.

Reto:

Con la chicharra modifique los valores de las salidas análogas para generar diferentes sonidos.

Análisis de Resultados:**Conclusiones:****Recomendaciones:****Bibliografía:**

GUÍA # 5: Comunicación Serial

Tema: Comunicación Serial

Objetivo General:

Aprender el manejo de comunicación serial utilizando la Placa Intel Galileo G2 para utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender la conexión para utilizar la comunicación serial.
2. Utilizar la librería que permite usar comunicación serial
3. Utilizar las sintaxis para declaración de pines correspondientes.
4. Probar el funcionamiento de los dispositivos conectados con un programa básico.

Preparatorio:

1. ¿Qué es la comunicación serial?
2. ¿Cuál es la sintaxis para usar comunicación serial?
3. ¿Cuáles son los pines de la Placa Intel Galileo G2 que permiten realizar esta comunicación
4. Enumere tres aplicaciones que se pueden hacer con comunicación serial.

Desarrollo de la práctica

Envío de mensajes por Comunicación Serial

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Cable USB
4. Cables macho-hembra para polarizar.
5. Hyperterminal

Proceso**Hardware:**

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

1. La placa Intel Galileo debe estar alimentada y conectada a la PC.
2. Para esta práctica sólo se necesita conectar los leds a GND y la placa de los pulsadores si debe polarizarse a VCC y GND
3. La placa de comunicación serial debe conectarse en los pines de rx y tx respectivamente.
4. Posteriormente polarizar la placa con cables macho-hembra.

Software:

1. Se debe tener instalado el IDE de Arduino
2. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
3. Declarar los pines como e/s y las librerías necesarias en el **void setup**
4. Realizar la programación deseada en el **void loop**
5. Compilar
6. Cargar el programa al Intel Galileo G2
7. Utilizar el Hyperterminal que proporciona el IDE de Arduino
8. Verificar el funcionamiento

Diagrama de Conexión:

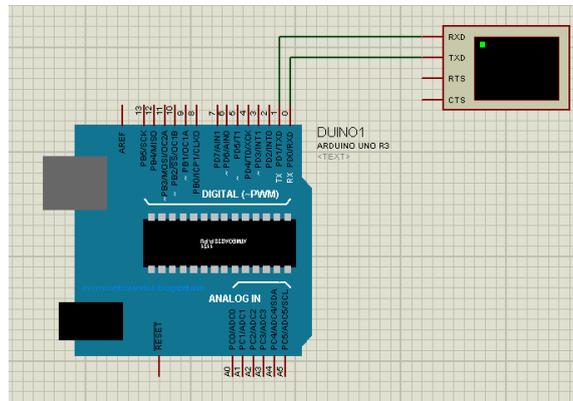


Figura 17. Diagrama de conexión de comunicación serial

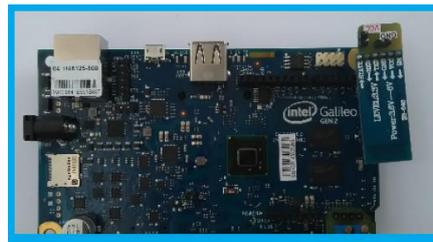


Figura 18. Comunicación Serial

Código Fuente:

```
void setup()

{
Serial.begin(9600); // configuración de comunicación
}

void loop()

{
```

```
Serial.print("hola"); // envio de frase
delay(5000); //retardo
Serial.print('A');// envio de letra
delay(5000); //retardo
}
```

// Al recibir la letra enciende un led de la placa usando comunicación serial.

```
int ledPin = 13;
char inByte ;

void setup()
{
Serial.begin(9600);
pinMode(ledPin, OUTPUT);
digitalWrite(ledPin, LOW);
}

void loop()
{
if (Serial.available() > 0) {
inByte = Serial.read();
if (inByte == 'a') {
if(digitalRead(ledPin)){
digitalWrite(ledPin, LOW);
}
else {
```

```
digitalWrite(ledPin, HIGH);  
}  
}  
}  
}
```

Resultado:

Se tiene como resultado la visualización de los caracteres enviados en el hyperterminal, y en el segundo caso se enciende o se apaga el led de acuerdo al dato enviado.

Reto:

Realizar diferentes juegos de luces cada vez que se envíen datos diferentes.

Análisis de Resultados:**Conclusiones:****Recomendaciones:****Bibliografía**

Guía # 6: LCD

Tema: Manejo de LCD

Objetivo General: Aprender el manejo de una LCD en la Placa Intel Galileo G2 y utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender a conectar la LCD.
2. Utilizar la librería que permite usar una LCD
3. Utilizar las sintaxis para declaración de pines correspondientes.

Preparatorio:

1. ¿Qué es una pantalla LCD?
2. ¿Cuál es la librería que se debe incluir para darle uso a la LCD?
3. ¿Cuántos tipos de LCD existen?
4. ¿En qué pines se puede conectar la LCD?
5. Enumere tres aplicaciones que se pueden hacer con una pantalla LCD.

Desarrollo de la práctica

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Placa de entrenamiento de la LCD
4. Cable USB
5. Cables macho-hembra para polarizar.



Figura 19. Pantalla LCD

Proceso de Software:

LiquidCrystal (): Esta función crea una variable del tipo LiquidCrystal.

Sintaxis:

LiquidCristal name(RS, E, D4, D5, D6, D7)

LiquidCristal name(RS, RW, E, D4, D5, D6, D7)

LiquidCristal name(RS, E, D0, D1, D2, D3, D4, D5, D6, D7)

LiquidCristal name(RS, RW, E, D0, D1, D2, D3, D4, D5, D6, D7)

Esta guía tiene su desarrollo en conjunto con la guía número 7.

Guía # 7: Teclado

Tema: Manejo del Teclado Matricial

Objetivo General: Aprender la programación para hacer uso del teclado matricial en la Placa Intel Galileo G2 y utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender la conexión para utilizar el teclado matricial.
2. Utilizar la librería que permite usar el teclado matricial
3. Utilizar las sintaxis para declaración de pines correspondientes.
4. Probar el funcionamiento de los dispositivos conectados con un programa básico.

Preparatorio:

1. ¿Qué es un teclado matricial?
2. ¿Cuál es la librería que se debe descargar para usar el teclado?
3. ¿Cuántos tipos de teclados matriciales existen?
4. ¿En qué pines se puede conectar el teclado matricial?
5. Enumere tres aplicaciones que se pueden hacer con el teclado matricial.

Desarrollo de la práctica

Ejemplo de LCD y Teclado: El programa envía caracteres presionados desde el teclado y se visualizan en la LCD.

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Placa de entrenamiento del Teclado y LCD
4. Cable USB

5. Cables macho-hembra para polarizar.

Proceso

Hardware:

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

1. La placa Intel Galileo debe estar alimentada y conectada a la PC.
2. Para esta práctica sólo se necesita conectar los leds a GND y la placa de los pulsadores si debe polarizarse a VCC y GND
3. La placa de entrenamiento de la LCD debe conectarse en los en pines 8,9,10,11,12 y el Teclado en pines 0,1,2,3,4,5,6 y 7, es necesario recordar que las placas deben ser polarizadas.
4. Posteriormente polarizar la placa con cables macho-hembra.

Software:

1. Se debe tener instalado el IDE de Arduino
2. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
3. Declarar los pines como e/s y las librerías necesarias en el **void setup**
4. Realizar la programación deseada en el **void loop**
5. Compilar
6. Cargar el programa al Intel Galileo G2
7. Verificar el funcionamiento

Diagrama de conexión:

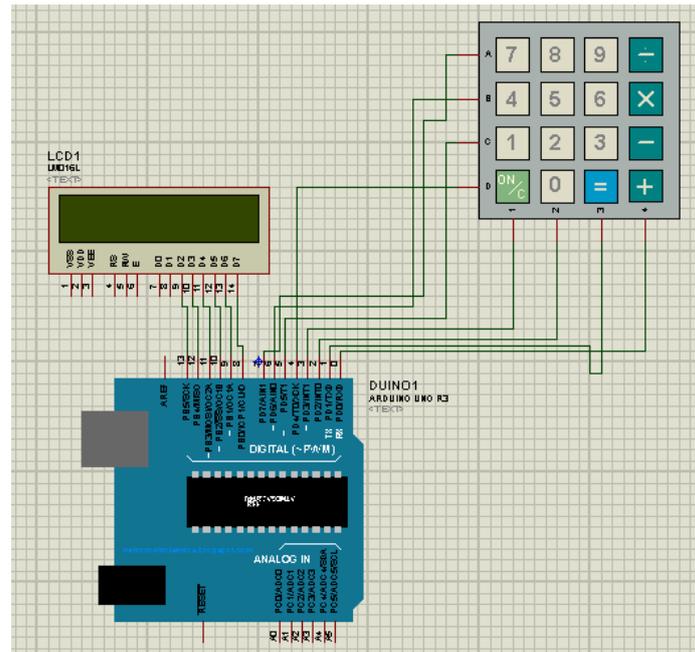


Figura 20. Diagrama de conexión de lcd y teclado



Figura 21. LCD y teclado

Programación

Código Fuente:

```
#include <Keypad.h> //lib de teclado
```

```
#include <LiquidCrystal.h> //lib de LCD
```

LiquidCrystal

```
lcd(13, 12, 11, 10, 9, 8 ); // Inicio de LCD
```

```
const byte ROWS = 4; //FILAS
```

```
const byte COLS = 4; //COLUMNAS
```

```
char
```

```
keys[ROWS][COLS] = { // Declarar matriz
```

```
{'1','4','7','0'},
```

```
{'2','5','8','M'},
```

```
{'3','6','9','='},
```

```
{'*', '#', '+', '-'}
```

```
};
```

```
byte rowPins[ROWS] = {7, 6, 5, 4}; // Definir Filas
```

```
byte colPins[COLS] = {3, 2, 1, 0}; // Definir columnas
```

```
// Lectura del teclado
```

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins,
```

```
ROWS, COLS );
```

```
void setup()
```

```
{
```

```
lcd.begin(20,4 );
```

```
}  
  
void loop()  
{  
  
char key = keypad.getKey();// almacena dato presionado  
if(key) // si tecla es presionada  
  
{  
lcd.setCursor(0,0);// ubicacion de la LCD  
lcd.print(key);  
}  
}
```

Resultado:

Cuando se presiona una tecla, se visualiza en la pantalla LCD.

Reto:

Visualizar en conversor análogo digital desde la LCD y comparar el valor CAD con uno ingresado desde el teclado.

Análisis de resultados:**Conclusiones:****Recomendaciones:****Bibliografía:**

Guía # 8: Sensor PIR

Tema: Manejo del sensor PIR

Objetivo General:

Aprender el manejo y función del sensor PIR en la placa Intel Galileo G2 para utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender la conexión para utilizar el sensor PIR.
2. Utilizar las sintaxis para declaración de pines correspondientes.
3. Probar el funcionamiento de los dispositivos conectados con un programa básico.

Preparatorio:

1. ¿Qué es un sensor?
2. ¿Qué hace y cómo funciona un sensor PIR?
3. ¿Utiliza alguna librería el sensor PIR?
4. ¿En qué pines se conecta el sensor PIR?
5. Enumere tres aplicaciones que se pueden hacer con el sensor PIR.

Desarrollo de la práctica

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Placa de entrenamiento del sensor PIR
4. Cable USB
5. Cables macho-hembra para polarizar.

Proceso

Hardware:

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

1. La placa Intel Galileo debe estar alimentada y conectada a la PC.
2. Para esta práctica se necesita conectar la placa de entrenamiento del sensor PIR a GND y a VCC y la placa de los leds a GND.

Software:

1. Se debe tener instalado el IDE de Arduino
2. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
3. Declarar los pines como e/s y las librerías necesarias en el **void setup**
4. Realizar la programación deseada en el **void loop**
5. Compilar
6. Cargar el programa al Intel Galileo G2
7. Verificar el funcionamiento

Diagrama de conexión:

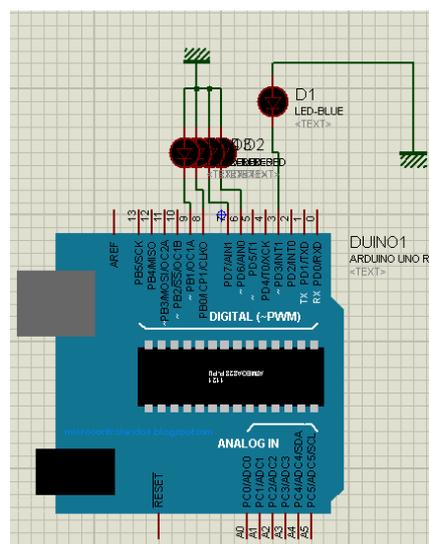


Figura 22. Diagrama de conexión del Sensor PIR



Figura 23. Sensor PIR



Figura 24. Funcionamiento del sensor PIR

Código Fuente:

```
int A=6; //variable para led A
int B=7; //variable para led B
int C=8; //variable para led C
int D =9;//variable para led D
int pir=3;

void setup()
{
    // put your setup code here, to run once:
    pinMode(A,OUTPUT); // Declarar pin A como salida
    pinMode(B,OUTPUT); // Declarar pin B como salida
    pinMode(C,OUTPUT); // Declarar pin C como salida
    pinMode(D,OUTPUT); // Declarar pin D como salida
    pinMode(pir,INPUT);
```

```
}  
void loop()  
{  
  // put your main code here, to run repeatedly:  
  if(digitalRead(pir)==HIGH){  
    digitalWrite(A,HIGH); // Encender led A  
    digitalWrite(B,HIGH); // Encender led B  
    digitalWrite(C,HIGH); // Encender led C  
    digitalWrite(D,HIGH); // Encender led D  
  }  
  else  
  {  
    digitalWrite(A,LOW); // Apagar led A  
    digitalWrite(B,LOW); // Apagar led B  
    digitalWrite(C,LOW); // Apagar led C  
    digitalWrite(D,LOW); // Apagar led D  
  }  
}
```

Resultado:

Cuando el sensor detecta la mano de una persona envía mensajes para que se enciendan los leds, de la misma manera cuando se retira la mano se apagan

Reto:

Observar en la LCD el tiempo que permanece una persona frente al sensor.

Análisis de Resultados:**Conclusiones:****Recomendaciones:****Bibliografía:**

Guía # 9: Sensor Ultrasónico

Tema: Manejo del sensor Ultrasónico

Objetivo General:

Aprender el manejo del sensor ultrasónico en la Placa Intel Galileo G2 para utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender la conexión para utilizar el sensor ultrasónico.
2. Utilizar las sintaxis para declaración de pines correspondientes.
3. Utilizar la sintaxis correspondiente para el cálculo de la distancia que se va a sensar
4. Probar el funcionamiento de los dispositivos conectados con un programa básico.

Preparatorio:

1. ¿Qué es un sensor?
2. ¿Qué hace un sensor ultrasónico?
3. ¿Utiliza alguna librería el sensor ultrasónico?
4. ¿Cuál es la fórmula para calcular la distancia del sensor ultrasónico?
5. ¿En qué pines se conecta el sensor ultrasónico?
6. Enumere tres aplicaciones que se pueden hacer con el sensor ultrasónico.

Desarrollo de la práctica

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Placa de entrenamiento del sensor ultrasónico
4. Cable USB
5. Cables macho-hembra para polarizar.

Proceso

Hardware:

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

1. La placa Intel Galileo debe estar alimentada y conectada a la PC.
2. Para esta práctica se necesita conectar la placa de entrenamiento del sensor Ultrasónico a GND y a VCC y la placa de los leds a GND.

Software:

1. Se debe tener instalado el IDE de Arduino
2. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
3. Declarar los pines como e/s y las librerías necesarias en el **void setup**
4. Realizar la programación deseada en el **void loop**
5. Compilar
6. Cargar el programa al Intel Galileo G2
7. Verificar el funcionamiento

Diagrama de conexión:

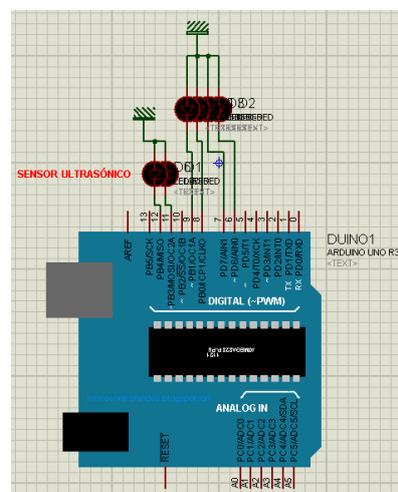


Figura 25. Diagrama de conexión del Sensor ultrasónico

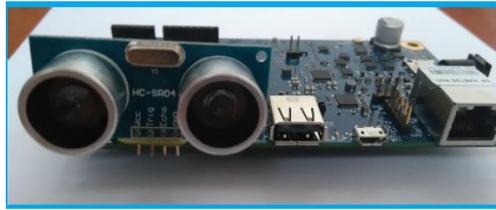


Figura 26. Sensor ultrasónico

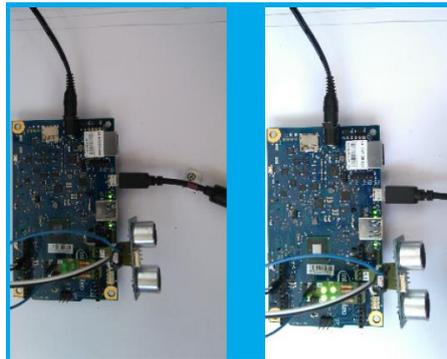


Figura 27. Funcionamiento del sensor ultrasónico

Programación

Código Fuente:

```
int duracion, distancia;
```

```
int eco=12;
```

```
int trig=11;
```

```
int A=6; //variable para led A
```

```
int B=7; //variable para led B
```

```
int C=8; //variable para led C
```

```
int D =9;//variable para led D
```

```
void setup()
```

```
{
```

```
  // put your setup code here, to run once:
```

```
Serial.begin(9600);

pinMode(trig,OUTPUT); // trig
pinMode(eco, INPUT); //eco // put your setup code here, to run once:
pinMode(A,OUTPUT); // Declarar pin A como salida
pinMode(B,OUTPUT); // Declarar pin B como salida
pinMode(C,OUTPUT); // Declarar pin C como salida
pinMode(D,OUTPUT); // Declarar pin D como salida
}
void loop()
{
  // put your main code here, to run repeatedly:
  digitalWrite(trig,HIGH);
  delayMicroseconds(1000);
  digitalWrite(trig,LOW);
  duracion=pulseIn(eco,HIGH);
  distancia=(duracion/2) /29;
  Serial.println(distancia);
  if(distancia<30){
    digitalWrite(A,HIGH);
    digitalWrite(B,HIGH);
    digitalWrite(C,HIGH);
    digitalWrite(D,HIGH);
  }
  else
  {
```

```
digitalWrite(A,LOW);  
digitalWrite(B,LOW);  
digitalWrite(C,LOW);  
digitalWrite(D,LOW);  
}  
delay(100);  
}
```

Resultado:

Cuando el sensor detecta la mano de una persona envía mensajes para que se enciendan los leds, de la misma manera cuando se retira la mano se apagan

Reto:

Visualizar en una LCD la distancia en centímetros del objeto en frente del sensor, si el objeto se acerca a menos de 20 centímetros encender la chicharra.

Análisis de Resultados:**Conclusiones:****Recomendaciones:****Bibliografía:**

Guía # 10: Acelerómetro

Tema: Manejo del Acelerómetro

Objetivo General:

Aprender el manejo del sensor de aceleración en la Placa Intel Galileo G2 para utilizar estos conocimientos en prácticas de clase.

Objetivos Específicos:

1. Aprender la conexión para utilizar el acelerómetro.
2. Utilizar la librería que requiere el sensor acelerómetro.
3. Utilizar las sintaxis para declaración de pines correspondientes.
4. Probar el funcionamiento de los dispositivos conectados con un programa básico.

Preparatorio:

1. ¿Qué es un sensor
2. ¿Qué hace un sensor acelerómetro?
3. ¿Cuáles son los ejes en los que trabaja?
4. ¿Cuál es la librería que utiliza en sensor acelerómetro?
5. ¿En qué pines se conecta el sensor acelerómetro?
6. Enumere tres aplicaciones que se pueden hacer con el sensor acelerómetro.

Desarrollo de la práctica

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Placa de entrenamiento del sensor acelerómetro
4. Cable USB
5. Cables macho-hembra para polarizar.

Proceso

Hardware:

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

1. La placa Intel Galileo debe estar alimentada y conectada a la PC.
2. Para esta práctica se necesita conectar la placa de entrenamiento del sensor acelerómetro a GND y a VCC.
3. Este sensor se conecta en los pines SC y SA respectivamente.

Software:

1. Se debe tener instalado el IDE de Arduino
2. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
3. Declarar los pines como e/s y las librerías necesarias en el **void setup**
4. Realizar la programación deseada en el **void loop**
5. Compilar
6. Cargar el programa al Intel Galileo G2
7. Verificar el funcionamiento

Diagrama de conexión:

Es un sensor que nos indica la posición en un plano x, y y z mediante una comunicación I2C, que permite que sea asincrónica.

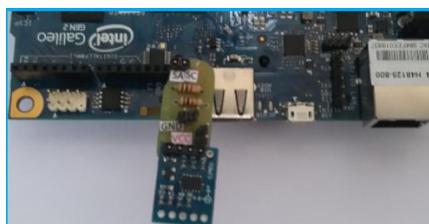


Figura 28. Sensor acelerómetro



Figura 29. Funcionamiento del sensor acelerómetro

Programación

Código Fuente:

```
#include "Wire.h"

#include "ADXL345.h"

//ADXL345 Accel;

ADXL345  adxl; //variable      adxl dela libreria      ADXL345

void  setup()

{

Serial.begin(9600); //configuracion de cx serial

adxl.powerOn(); //activar      sensor

}

void loop()

{

int x,y,z;      // inicializar variables de posicion

adxl.readAccel(&x, &y, &z); //lee datos del acelerometro //enviar      datos      a      la
comunicacion serial

Serial.print(x);
```

```
Serial.print(":");  
Serial.print(y);  
Serial.print(":");  
Serial.println(z);  
delay(200);  
}
```

Resultado:

Cuando se mueve el sensor acelerómetro se visualizan tres números diferentes que representan las coordenadas x, y, z.

Reto:

Visualizar en una LCD las coordenadas x,y,z; dependiendo de cómo se haga girar el sensor en el plano x, hacer girar un servo motor.

Análisis de Resultados:**Conclusiones:****Recomendaciones:****Bibliografía:**

Guía # 11: Bluetooth

Tema: Manejo de Bluetooth

Objetivo General:

Aprender el manejo del módulo bluetooth en la Placa Intel Galileo G2 para utilizar estos conocimientos en prácticas de clase.

Objetivo Específico:

1. Aprender la conexión para utilizar el módulo bluetooth.
2. Utilizar la librería que requiere el bluetooth.
3. Utilizar las sintaxis para declaración de pines correspondientes.
4. Probar el funcionamiento de los dispositivos conectados con un programa básico.

Preparatorio:

1. ¿Qué es un módulo bluetooth?
2. ¿Qué aplicaciones se pueden hacer con un módulo bluetooth?
3. ¿Tiene alguna librería para poder ser usado el módulo bluetooth?
4. ¿Hay pines específicos en donde debe conectarse el módulo bluetooth y cuáles son?

Desarrollo de la práctica

Materiales:

1. Placa Intel Galileo G2
2. PC
3. Placa de entrenamiento del módulo bluetooth
4. Celular con aplicación bluetooth
5. Cable USB
6. Cables macho-hembra para polarizar.

Proceso

Hardware:

Cada placa de entrenamiento tiene un pin para VCC y uno para GND, en este caso GND está al lado izquierdo de la placa, de donde debe salir el cable para conectarse al pin GND del Intel Galileo G2.

1. La placa Intel Galileo debe estar alimentada y conectada a la PC.
2. Para esta práctica se necesita conectar la placa de entrenamiento del módulo bluetooth a GND y VCC y también utilizar uno de los pines digitales.

Software:

1. Se debe tener instalado el IDE de Arduino
2. Una vez abierto el IDE de Arduino, verificar en la pestaña herramientas la Placa que se está utilizando y el puerto en el que está conectado el Intel Galileo G2.
3. Declarar los pines como e/s y las librerías necesarias en el **void setup**
4. Realizar la programación deseada en el **void loop**
5. Compilar
6. Cargar el programa al Intel Galileo G2
7. Configurar el acoplamiento con el dispositivo móvil en este caso un celular
8. Verificar el funcionamiento

Diagrama de conexión:

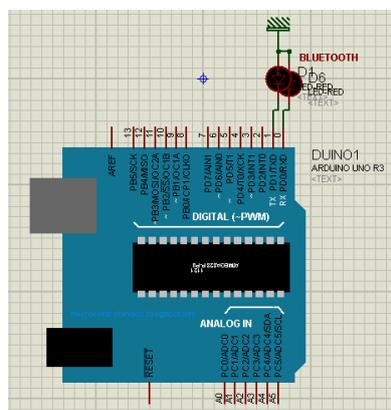


Figura 30. Diagrama de conexión del módulo bluetooth

Programación

Código Fuente:

```
#include <SoftwareSerial.h>

char rec;

SoftwareSerial blue(2,3);

void setup()
{
  pinMode(13,OUTPUT);
  blue.begin(9600);
  blue.println("Conectado");
}

void loop()
{
  if(blue.available()){
    rec=blue.read();
    switch(rec)
    {
      case '0':
        digitalWrite(13,LOW);
        delay(500);
```

```
    blue.println("Led off");  
  
    break;  
  
    case '1':  
  
        digitalWrite(13,HIGH);  
  
        delay(500);  
  
        blue.println("Led on");  
  
        break;  
  
    default:  
  
        delay(500);  
  
        blue.print(rec);  
  
        blue.println(" no es una orden valida. Introduzca 0 o 1");  
  
    }  
  
    }  
  
    }
```

Resultado:

Cuando se envían los números 1 o 0 desde una aplicación bluetooth, en este caso desde el celular, se apaga o prende un led respectivamente

Reto:

Aumentar el número de casos y hacer que suene una alarma cuando se reciba un número y de la misma manera se mueva un motor.

Análisis de Resultados:**Conclusiones:****Recomendaciones:****Bibliografía:**

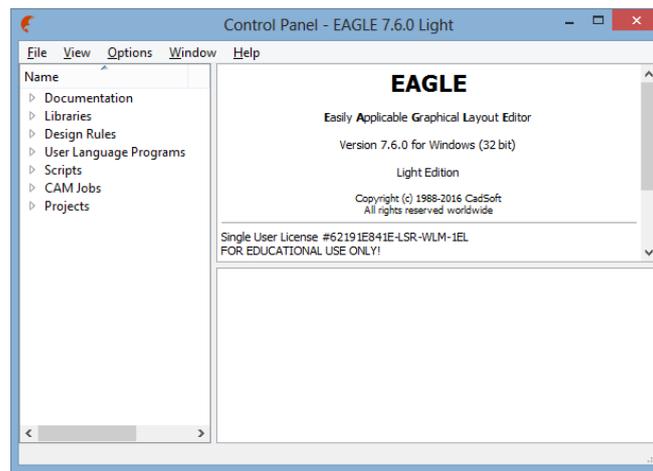
ANEXO 3

MANUAL DE EAGLE

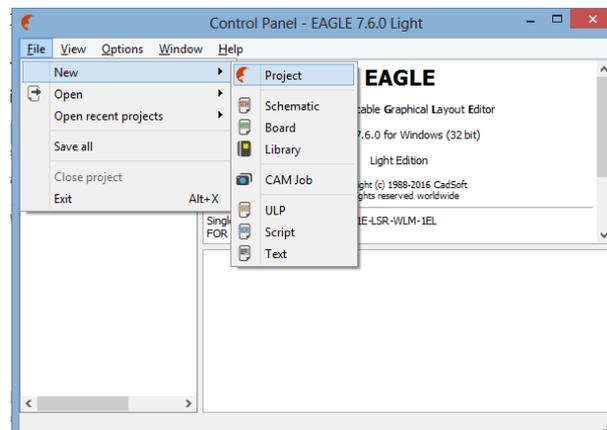
PROCESO QUE SE DEBE SEGUIR PARA EL DISEÑO DE LAS PLACAS DE ENTRENAMIENTO

Eagle sirve para crear circuitos esquemáticos y placas PCB.

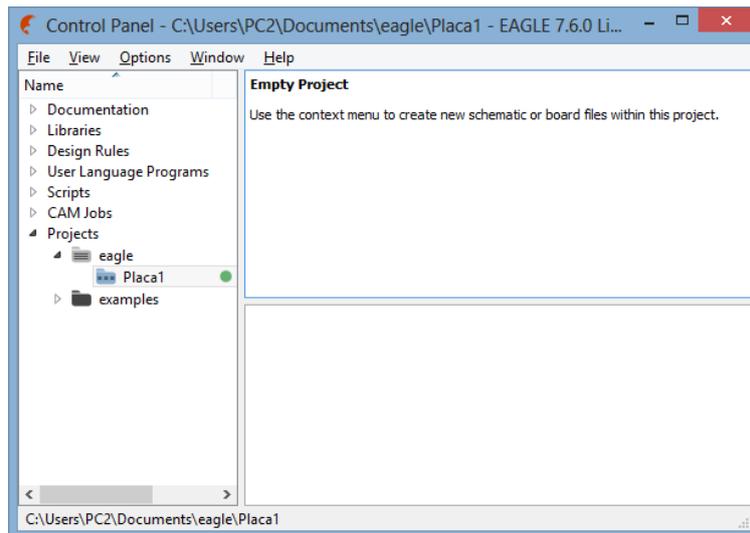
1. Primera interfaz de Eagle



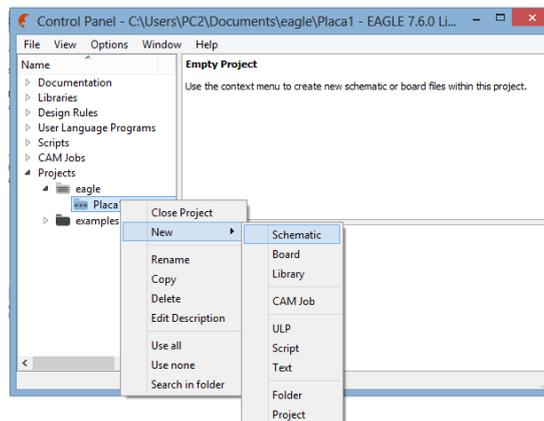
2. Para empezar un proyecto nuevo se debe ir primero a file, new, Project



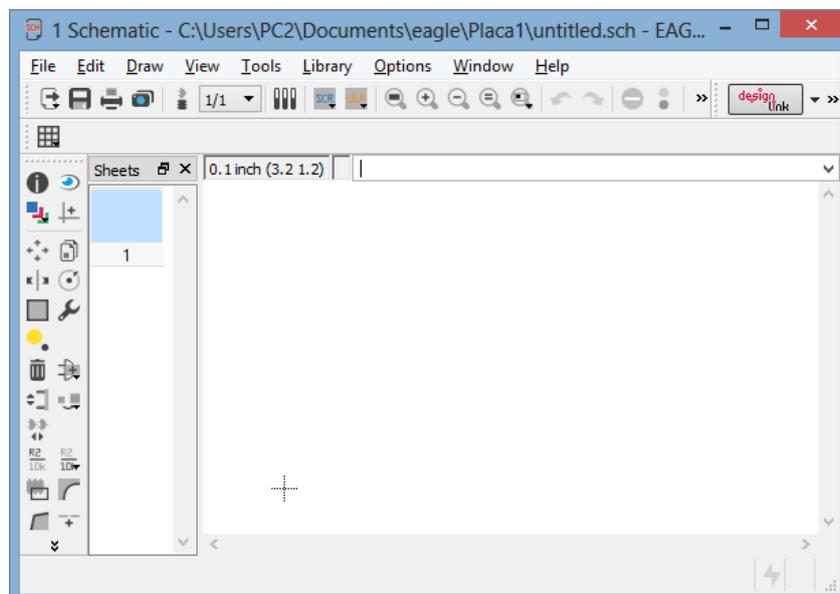
3. Se crea una carpeta de ese proyecto en donde se debe poner el nombre que se desee, en este caso se va a llamar Placa1.



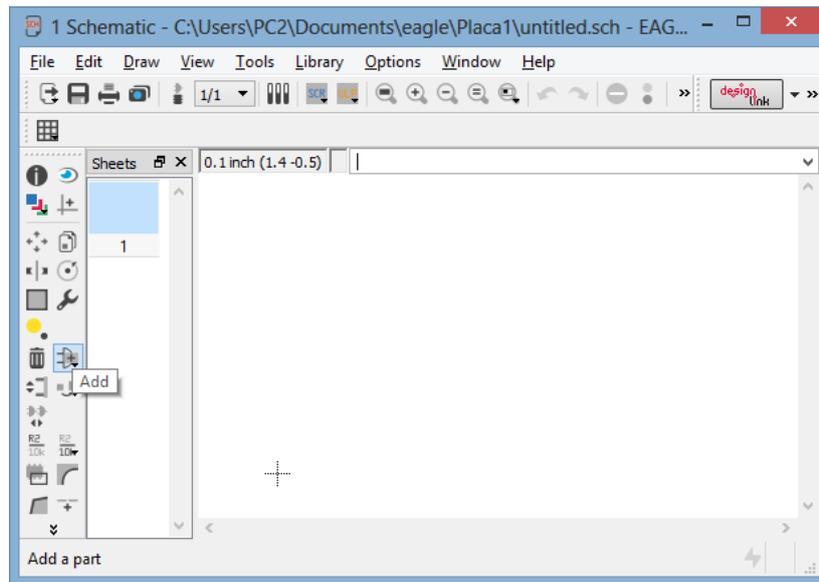
4. Dentro de la carpeta Placa1, se deben crear algunas cosas como el circuito esquemático, click derecho en Placa1, nuevo, esquemático.



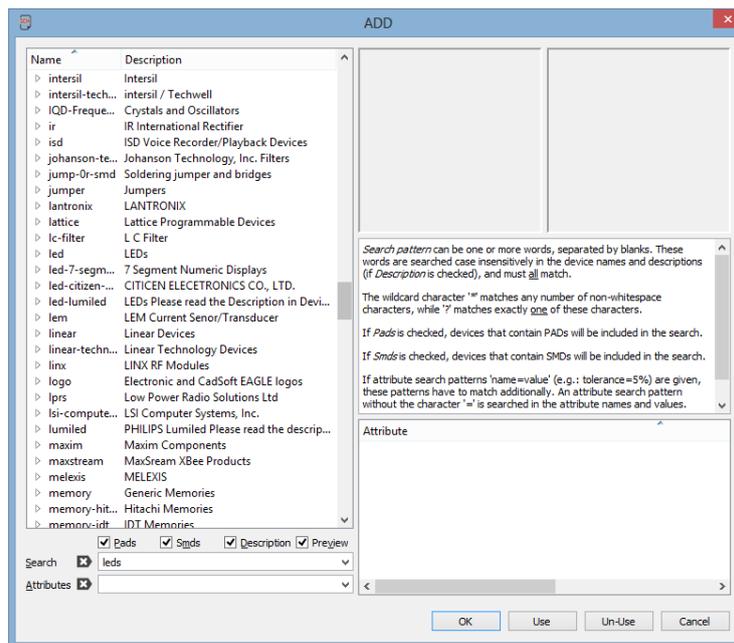
5. Aparecerá la siguiente pantalla, para luego empezar a seleccionar los componentes que sean necesarios.



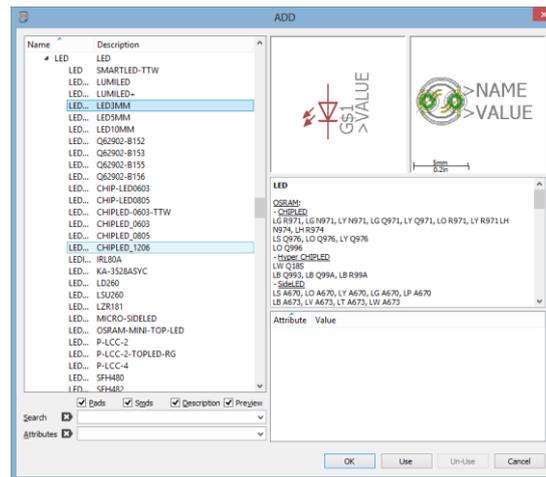
6. Se debe ubicar el icono para agregar los elementos llamado add.



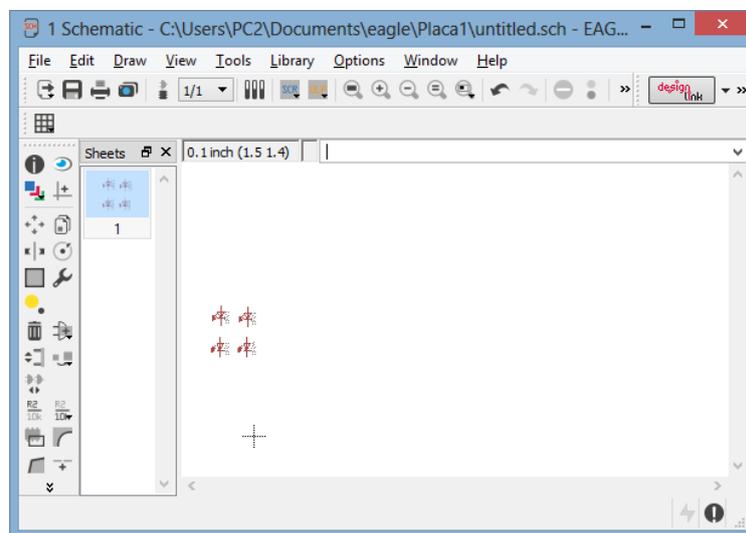
7. Aparecerá la siguiente pantalla en donde se deben buscar todos los elementos a utilizar.



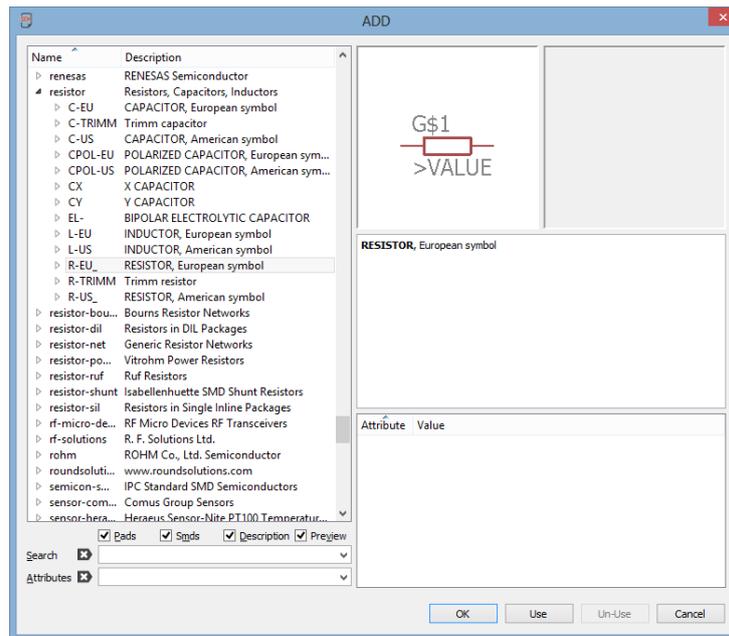
8. Una vez que se coloca la palabra LED, desplegar y seleccionar el tipo de led deseado, en este caso es el siguiente:



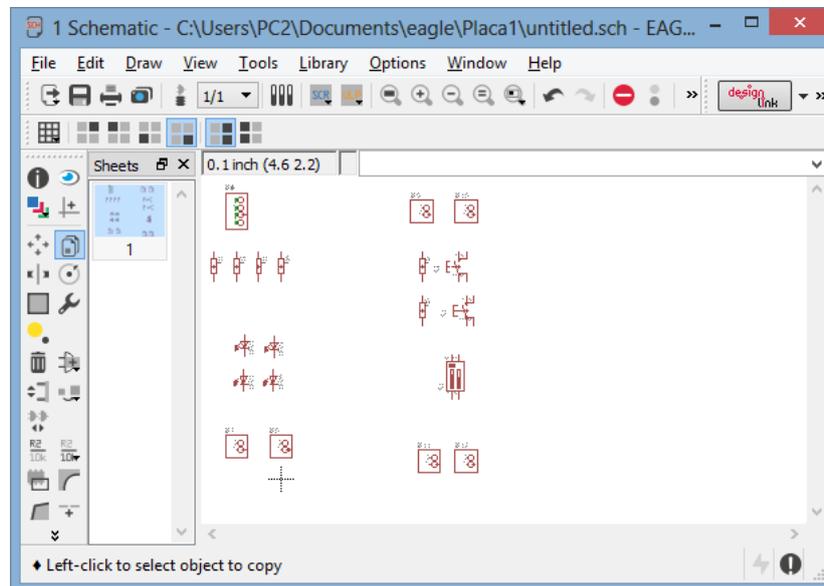
9. Al dar click en ok, aparece la pantalla para ubicar los elementos.



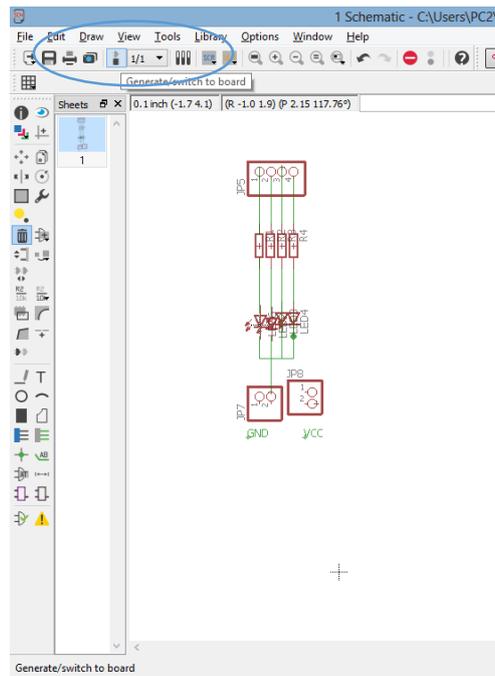
10. De la misma manera se procede con los otros elementos, primero se buscan y luego se ubican de acuerdo al diseño que se desee.
11. En este caso se necesitan, resistencias, pulsadores, dip switch y la polarización a VCC y GND.



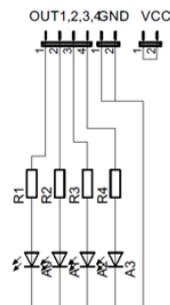
12. De la misma manera se continúa la búsqueda de todos los elementos a usar y posteriormente se ubica en el esquemático.
13. Para encontrar el pulsador se debe poner en la barra de búsqueda 10-xx y para el dip switch se debe poner switch y escoger el que se desee.
14. Una vez que se tengan los componentes necesarios, se va a tener la siguiente pantalla.



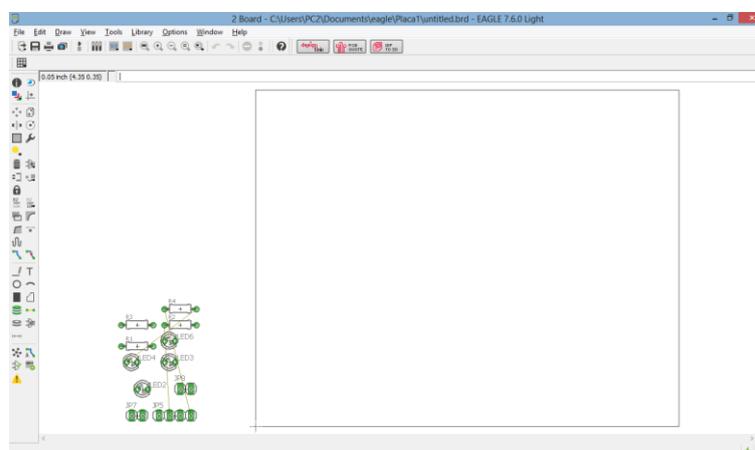
15. Ahora se debe realizar la conexión de los elementos.
16. Como el proyecto se llama placa 1, se va a continuar con el diseño de la placa 1, siendo así el diseño del lado derecho una demostración de cómo encontrar los componentes.
17. El siguiente paso es verificar que todo esté conectado, y luego generar el board.



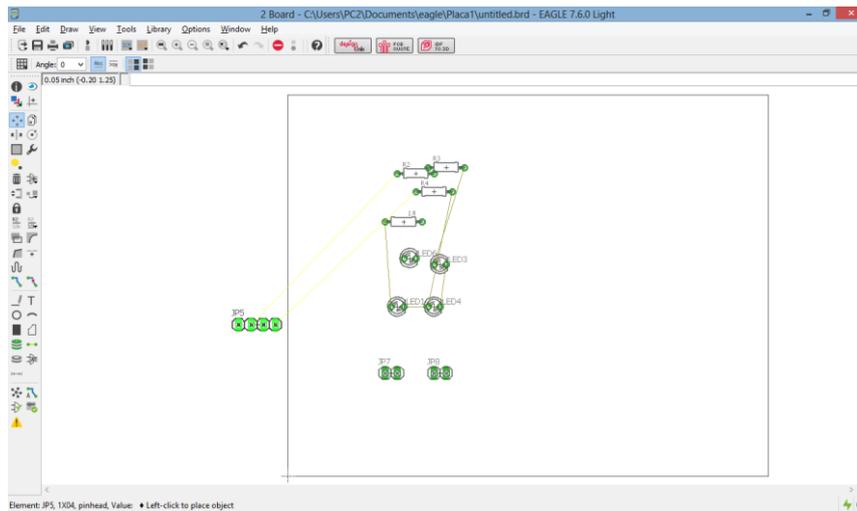
18. No existe conexión a VCC ya que la placa Intel Galileo G2 les proporciona a los leds pero en el caso de otras placas, sí es necesario.



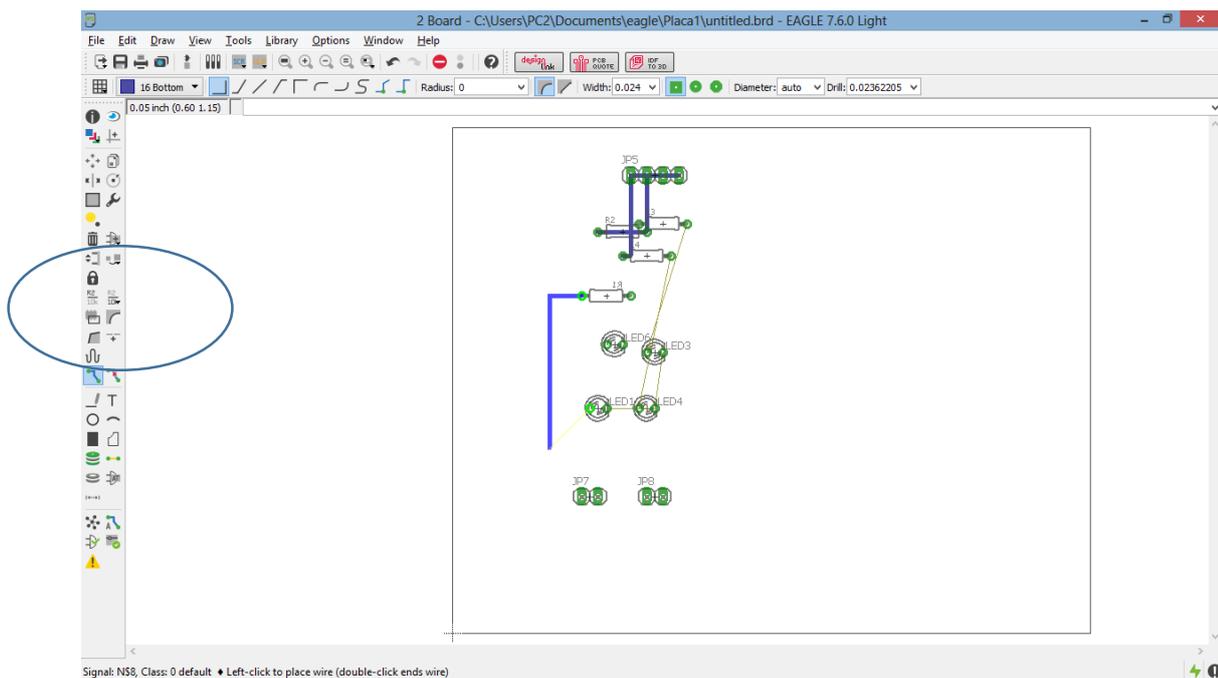
19. Esta es la nueva pantalla que se verá a continuación, en donde se deben arrastrar los elementos, es decir volver a ubicarlos.



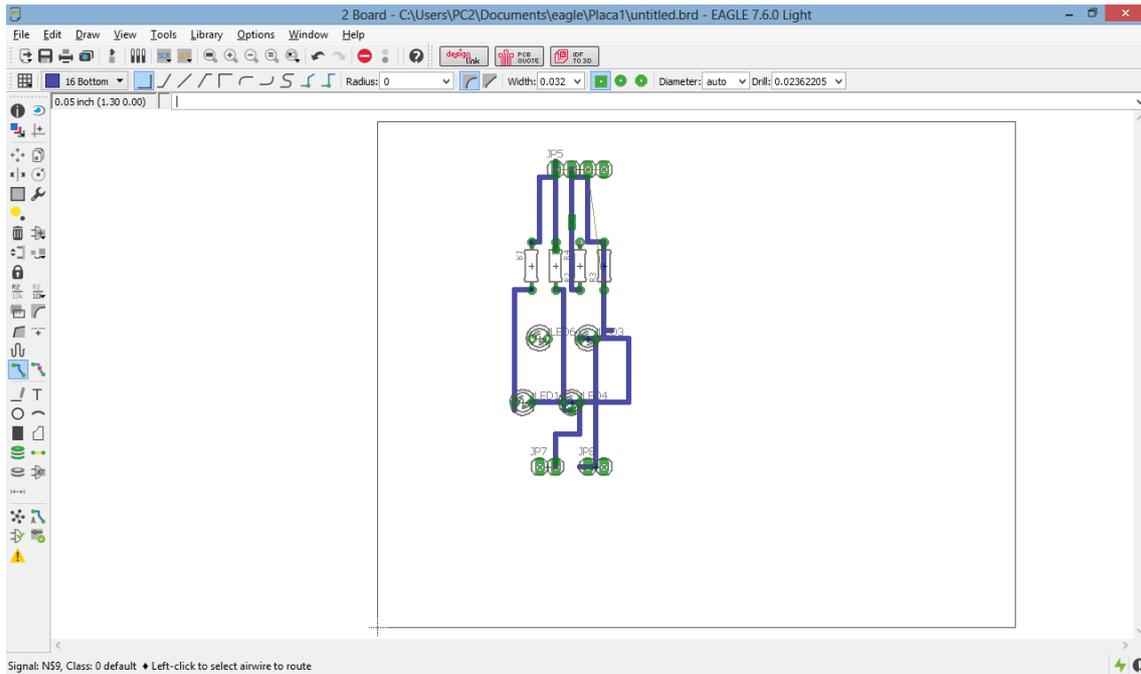
20. De esta manera se puede comprobar si todos los elementos están conectados o no



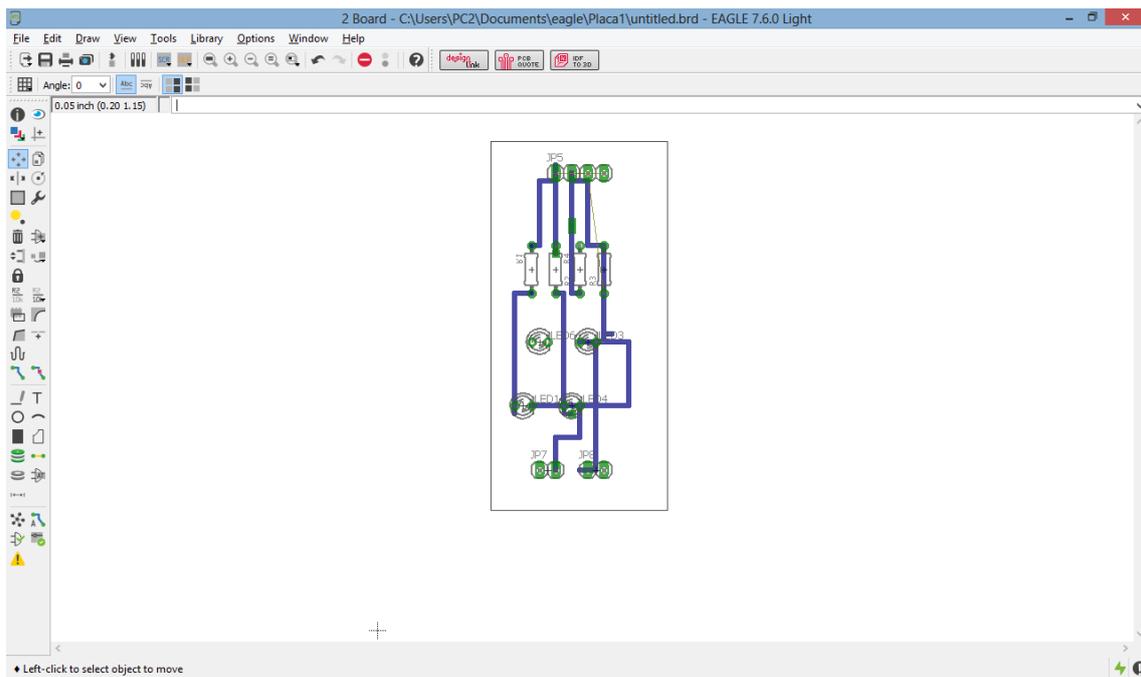
21. En el caso de que falte algún elemento, se cierra esa pantalla y aparece la pantalla anterior para corregir las conexiones.
22. Una vez que esté muy bien conectado, click en el botón Route y luego se den dibujar las pistas.



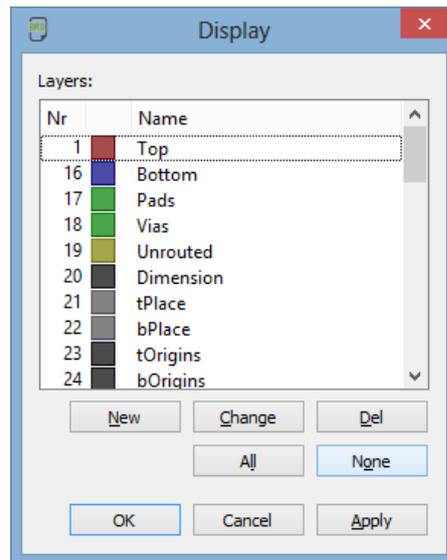
23. Una vez finalizado el ruteo se verá algo como lo que se muestra en la siguiente imagen dependiendo como se muevan los elementos van a ser diferentes los diseños.



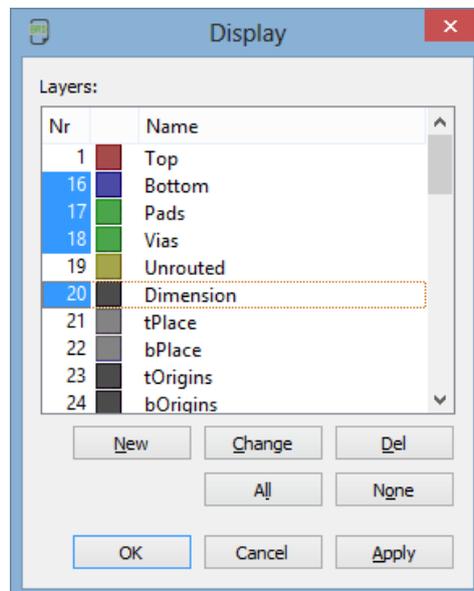
24. Posteriormente, se ajustan los bordes



25. El siguiente paso es ir hacia View, Layer Settings y click en none.



26. Se deben marcar las siguientes: Bottom, Pads, Vias y Dimension.



27. Para finalizar, se aplican los cambios antes mencionados y deben salir las pistas pero sólo con puntos de conexión, es así como se realiza el diseño de una placa de entrenamiento en Eagle.

