



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN

CARRERA: ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA: Estudio, diseño y construcción de un prototipo autónomo selector y transportador de placas coloreadas, activado por una aplicación Android para el Laboratorio de Electrónica de la Universidad Tecnológica Israel de la ciudad de Quito.

AUTOR: Jefferson Javier Pilapaña Tituaña

TUTOR: Ing. Cañete Bajuelo Rene Alberto

AÑO 2015

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de titulación certifico:

Que el trabajo de titulación **“ESTUDIO, DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO AUTÓNOMO SELECTOR Y TRANSPORTADOR DE PLACAS COLOREADAS, ACTIVADO POR UNA APLICACIÓN ANDROID PARA EL LABORATORIO DE ELECTRÓNICA DE LA UNIVERSIDAD TECNOLÓGICA ISRAEL.”**, presentado por el Sr. Jefferson Javier Pilapaña Tituaña, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. Enero del 2015

TUTOR

.....

ING. CAÑETE BAJUELO RENE ALBERTO

UNIVERSIDAD TECNOLÓGICA ISRAEL

AUTORÍA DEL TRABAJO DE TITULACIÓN

El abajo firmante, en calidad de estudiante de la Carrera de Electrónica Digital y Telecomunicaciones, declaro que los contenidos de este Trabajo de Titulación, requisito previo a la obtención del Grado de Ingeniería en Electrónica Digital y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito D.M. Enero del 2015

.....

Jefferson Javier Pilapaña Tituaña

CC: 1718365941

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de grado, aprueban el Trabajo de Titulación para la graduación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para títulos de pregrado.

Quito D.M. Enero del 2015

Para constancia firma:

TRIBUNAL DE GRADO

.....

PRESIDENTE

.....

MIEMBRO 1

.....

MIEMBRO 2

AGRADECIMIENTO

Agradezco a Dios por haberme guiado por el camino de la felicidad, brindándome la sabiduría y fortaleza para poder cumplir con mi objetivo de culminar mi carrera, a mis ingenieros por permitirme ser más que un amigo, a quienes les debo gran parte de mis conocimientos, gracias por su paciencia y enseñanza, al Ing. Mauricio Alminati gracias por su ayuda y tiempo, por la paciencia y disposición para brindarme sus conocimientos es un gran ejemplo a seguir lo admiro y respeto, a mi prestigiosa Universidad Tecnológica Israel por abrir sus puertas para prepararme para un futuro competitivo y formarme como una persona de bien.

DEDICATORIA

Dedico este Proyecto de Grado a Dios por estar conmigo en cada paso que doy y por haberme permitido llegar hasta este punto, a mis padres que me han dado la existencia, quienes día a día me dieron un motivo para culminar mi carrera su sacrificio, esfuerzo y sobre todo la confianza que pusieron en mí, gracias padre y madre por desearme lo mejor su presencia han ayudado construir y a forjar la persona que soy ahora, a mi hijo Matías por ser la fuente de motivación para poder superarme cada día más y así poder luchar por nuestro futuro.

ÍNDICE

Contenido

| | |
|--|-----|
| APROBACIÓN DEL TUTOR | II |
| AUTORÍA DEL TRABAJO DE TITULACIÓN | III |
| APROBACIÓN DEL TRIBUNAL DE GRADO | IV |
| AGRADECIMIENTO | V |
| DEDICATORIA | VI |
| ÍNDICE | VII |
| ÍNDICE FIGURAS..... | IX |
| ÍNDICE DE TABLAS..... | XI |
| INTRODUCCIÓN..... | 1 |
| Objetivos | 1 |
| Objetivo General: | 1 |
| Objetivos Específicos: | 2 |
| CAPÍTULO I | 3 |
| FUNDAMENTACIÓN TEÓRICA..... | 3 |
| 1.1 Introducción | 3 |
| 1.2 Marco Teórico | 3 |
| 1.2.1 Arduino Uno..... | 3 |
| 1.2.2 Shields | 5 |
| 1.2.3 Servomotor | 5 |
| 1.2.4 Electroimán..... | 6 |
| 1.2.5 Sensor de Color..... | 6 |
| 1.2.6 Bluetooth..... | 7 |
| 1.2.7 APP BlueTerm | 9 |
| CAPÍTULO II | 11 |
| BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO PARA LA CONSTRUCCIÓN DEL SELECTOR Y TRANSPORTADOR DE PLACAS COLOREDAS | 11 |
| 2.1 Problema Principal..... | 11 |

| | |
|---|----|
| 2.2 Propósito de los Objetivos..... | 11 |
| 2.3 Hipótesis | 11 |
| 2.4. Fundamentación Teórica..... | 11 |
| 2.5 Metodología de Investigación..... | 12 |
| 2.5.1 Método de análisis y síntesis | 12 |
| 2.5.2 Método de modelación | 12 |
| 2.5.3 Método experimental | 12 |
| 2.6 Resultados Esperados | 16 |
| CAPÍTULO III | 17 |
| PRESENTACIÓN DE RESULTADOS..... | 17 |
| 3.1 Propuesta de Solución del Problema..... | 17 |
| 3.2 Diseño del Prototipo | 17 |
| 3.2.1 Fase 1 | 18 |
| 3.2.2 Fase 2 | 19 |
| 3.2.3 Fase 3..... | 22 |
| 3.2.4 Diagrama circuital general del proyecto..... | 24 |
| 3.2.5 Circuito impreso del proyecto..... | 25 |
| 3.2.6 Circuito impreso Bluetooth FC ID:Z4T | 25 |
| 3.2.7 Elementos utilizados para la construcción del prototipo..... | 26 |
| 3.3 Implementación del Proyecto | 27 |
| 3.4 Evaluación Técnica..... | 33 |
| 3.5 Pruebas de funcionamiento..... | 34 |
| 3.5.1 Prueba 1: Identificación y selección de placas coloreadas | 34 |
| 3.5.1 Prueba 2: Distancia entre las placas y el electroimán | 34 |
| 3.6 Análisis de Resultados | 35 |
| 3.6.1 Análisis de Resultados de la Prueba 1 | 35 |
| 3.6.2 Análisis de Resultados de la Prueba 2 | 35 |
| CONCLUSIONES Y RECOMENDACIONES..... | 36 |
| BIBLIOGRAFÍA | 38 |
| ANEXOS | 40 |

ÍNDICE FIGURAS

| | |
|---|----|
| Figura 1.1 Placa Arduino Uno..... | 3 |
| Figura 1.2 Shields para Arduino | 5 |
| Figura 1.3 Sensor de color TCS230 (Acoplamiento con placa Arduino) | 7 |
| Figura 1.4 Bluetooth FCC ID: Z4T-BTSHIELD..... | 8 |
| Figura 1.5 Plataforma de Programación Arduino | 8 |
| Figura 1.6 Aplicación BlueTerm (terminales que se conectan a cualquier dispositivo Bluetooth) | 10 |
| Figura 2.1 Gráfica de pastel referida a la entrevista (pregunta 1) | 14 |
| Figura 2.2 Gráfica de pastel referida a la entrevista (pregunta 2) | 15 |
| Figura 2.3 Gráfica de pastel referida a la entrevista (pregunta 3) | 15 |
| Figura 2.4 Gráfica de pastel referida a la entrevista (pregunta 4) | 16 |
| Figura 3.1 Diagrama General | 17 |
| Figura 3.2 Fase de Selección y Arranque | 18 |
| Figura 3.3 Regular de Voltaje | 19 |
| Figura 3.4 Circuito Arduino | 20 |
| Figura 3.5 Circuito Sensor de Color | 21 |
| Figura 3.6 Circuito Módulo bluetooth..... | 21 |
| Figura 3.7 Circuito de Potencia – Puente H | 22 |
| Figura 3.8 Circuito Electroimán | 23 |
| Figura 3.9 Diagrama General del Circuito | 24 |
| Figura 3.10 Circuito Impreso del proyecto..... | 25 |
| Figura 3.11 Circuito Impreso del Módulo Bluetooth | 26 |
| Figura 3.12 PCB Placa Complementaria..... | 27 |
| Figura 3.13 Elementos del Circuito Complementario..... | 28 |
| Figura 3.14 Placa Arduino | 28 |
| Figura 3.15 Sensor de Color TSC230 | 29 |

| | |
|---|----|
| Figura 3.16 Parte Interna del Sistema | 29 |
| Figura 3.17 Vista Trasera del Sistema | 30 |
| Figura 3.18 Vista Frontal del Sistema | 31 |
| Figura 3.19 Vista Lateral del Sistema..... | 31 |
| Figura 3.20 Vista Lateral de las Placas | 32 |
| Figura 3.21 Producto final..... | 32 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 3.1 Evaluación Técnica..... | 33 |
| Tabla 3.2 Identificación y selección de placas coloreadas..... | 34 |
| Tabla 3.3 Distancia entre las placas y el electroimán | 34 |

INTRODUCCIÓN

La Universidad Tecnológica Israel es una institución del sistema de educación superior particular de derecho privado, nace en 1999 de la experiencia académica y profesional de dos institutos: el instituto Tecnológico Israel y el Instituto Tecnológico Italia, alianza de fortalecimiento para una educación superior de excelencia, con la finalidad de formar líderes con responsabilidad y pensamiento positivo, a través de los proyectos de investigación. Su Matriz Quito ubicada en la Av. Orellana y Francisco Pizarro E4-142, y su sede ubicada en la Av. Orellana 1762 y 9 de Octubre.

Esta noble institución tiene una oferta académica de varias carreras, una de estas es la carrera de Electrónica Digital y Telecomunicaciones encargada de formar profesionales en ingeniería con bases científicas, tecnológicas y humanísticas, con competencias para crear, manejar y aplicar modelos físico-matemáticos de la realidad, con conocimientos de computación, electromagnetismos, legales y administrativos; así como con competencias para el diseño e implementación de dispositivos electrónicos y de telecomunicaciones.

La carrera de Electrónica cuenta con laboratorios debidamente equipados para el desarrollo de las prácticas estudiantiles, pero no cuenta con proyectos reales que sirvan como guía de estudio.

Anteriormente el desarrollo de proyectos y procesos industriales se los realizaba por medio de microcontroladores y placas diseñadas según la necesidad del caso, con el paso del tiempo y el desarrollo tecnológico todo este proceso ha sido reemplazado por la tecnología Arduino, ya que recientemente se está incorporado en el mercado, existe poco conocimiento por los alumnos de la Universidad Israel acerca de las ventajas que tiene trabajar con este tipo de tecnología.

Objetivos

Objetivo General:

- Estudiar, diseñar y construir un prototipo autónomo selector y transportador de placas coloreadas, activado una aplicación Android para el laboratorio de electrónica de la Universidad Tecnológica Israel de la ciudad de Quito.

Objetivos Específicos:

- Estudiar los dispositivos electrónicos actuales para el desarrollo del sistema.
- Diseñar un sistema autónomo de fácil mantenimiento.
- Elaborar un sistema selector y transportador de placas coloreadas a bajo precio.
- Validar mediante pruebas que el prototipo está funcionando correctamente.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Para la realización del presente trabajo es indispensable tener en cuenta varios conceptos básicos que servirán para la construcción del prototipo autónomo selector y transportador de placas coloreadas, se conocerá las ventajas que se tiene al utilizar la placa Arduino y los acoplamientos como sensores, shields u otros elementos necesarios para un óptimo funcionamiento.

1.2 Marco Teórico

1.2.1 Arduino Uno

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

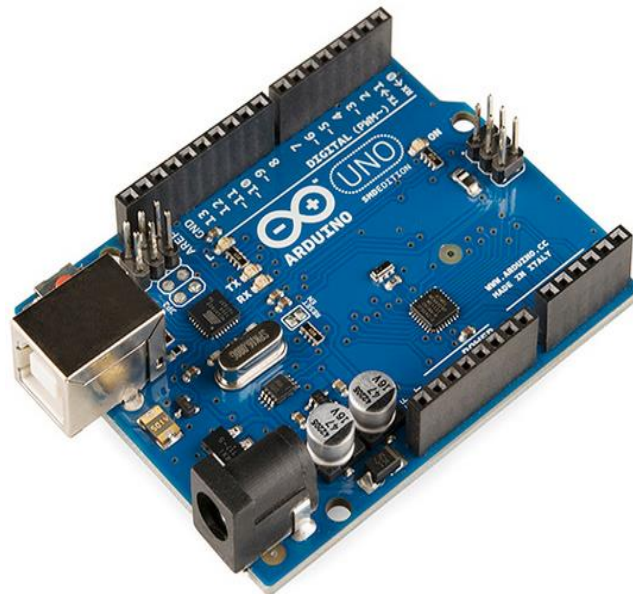


Figura 1.1 Placa Arduino Uno
Fuente: (Arduino Uno, 2015)

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing).

Características

Microcontrolador ATmega328

Voltaje de funcionamiento 5V

Voltaje de entrada (recomendado) 7V - 12V

Voltaje de entrada (límites) 6V - 20V

Digital pines I / O 14 (de los cuales 6 proporcionan salida PWM)

Pines de entrada analógica 6

Corriente DC por Pin I / O 40 mA

Corriente DC de 3.3V Pin 50 mA

Memoria Flash 32 KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque

SRAM 2 KB (ATmega328)

EEPROM 1 KB (ATmega328)

Velocidad del reloj de 16 MHz

Longitud 68.6 mm

Ancho 53.4 mm

Peso 25 g

Las ventajas al utilizar Arduino Uno son:

- **“Multi – plataforma:** El software funciona en los sistemas operativos Windows, Macintosh OSX y Linux.
- **Entorno de programación simple y directa:** Basado en el entorno de programación de Processing fácil de programar.

- **Software ampliable y de código abierto:** El software Arduino está publicado bajo una licencia libre y preparada para ser ampliado por programas experimentados a través del lenguaje C++. (Ventajas Arduino, 2015)

1.2.2 Shields

Es una placa impresa que se pueden conectar en la parte superior de la placa Arduino para ampliar sus capacidades, pudiendo ser apilada una encima de la otra. (Shields Arduino, 2010)

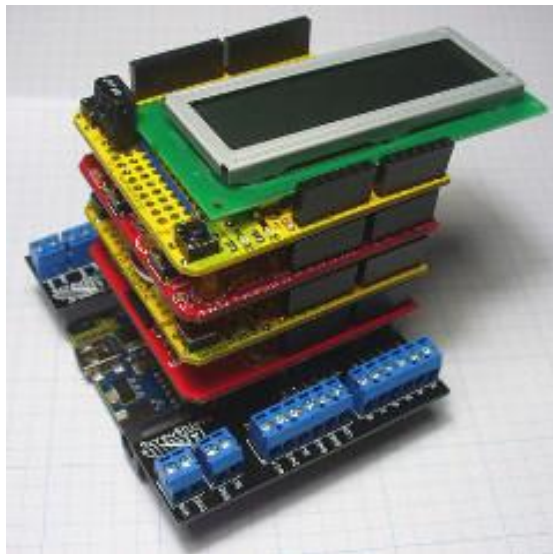


Figura 1.2 Shields para Arduino
Fuente: (Shields Arduino Uno, 2015)

Cabe mencionar que los Shields también pueden ser diseñados de acuerdo a los requerimientos de cada usuario de esta forma solo se utilizará las partes necesarias para iniciar el sistema.

1.2.3 Servomotor

Es un actuador mecánico basado en un motor y un conjunto de engranajes que permiten multiplicar el torque del sistema final, el cual posee elementos de control para monitorear de manera constante la posición de un elemento mecánico que será el enlace con el mundo exterior, los motores que forman parte de una impresora, junto a los sistemas de control de avance o retroceso del papel, forman un servomotor. (Servomotores, 2009)

Los servomotores se controlan mediante impulsos de ancho variable que deben refrescarse periódicamente. Esto significa que si se deja de enviar la señal de control en el tiempo en el que el servomotor lo necesita, éste dejará de mantenerse en la posición preestablecida y adoptará cualquier orientación regida por el esfuerzo al que esté sometido.

Ventajas de un servomotor:

- Potencia proporcional para cargas mecánicas.
- No consume mucha energía.
- Mayor precisión.

1.2.4 Electroimán

Es un tipo de imán en el que el campo magnético se produce mediante el flujo de una corriente eléctrica, desapareciendo en cuanto cesa dicha corriente, se dice que es un imán que atrae metales con ayuda de la electricidad. (Electroiman, 2015)

La principal ventaja de un electroimán sobre un imán permanente es que el campo magnético puede ser rápidamente manipulado en un amplio rango controlando la cantidad de corriente eléctrica. Sin embargo, se necesita una fuente continua de energía eléctrica para mantener el campo. (Electroiman, 2015)

1.2.5 Sensor de Color

El sensor de color con integrado TCS230 puede filtrar los datos RGB de la fuente de luz y la convierten en una onda cuadrada con una frecuencia directamente proporcional a la intensidad de la luz de radiación. Este sensor convierte el valor de $\mu\text{W}/\text{cm}^2$ en frecuencia en cada sensor que tiene. Está compuesto por 4 sensores, uno blanco sin filtro, otro azul, verde y uno rojo. (Sensor de color TCS230 electronilab, 2015)

“También hay una configuración para la frecuencia de respuesta, que va desde 10 hasta 600KHz. A velocidades muy lentas si el color es muy oscuro, puede tardar mucho en realizar un número grande de lecturas, en cambio sí se selecciona una frecuencia muy elevada, es posible que el microcontrolador no pueda procesarlo a tiempo.” (Colímetro TCS230, 2011)



Figura 1.3 Sensor de color TCS230 (Acoplamiento con placa Arduino)

Fuente: (Sensor de color TCS230 electronilab, 2015)

Características

Voltaje DC: 3V a 5V

Salida analógica y digital

Analógica: salida de voltaje lineal RGB

Leds incluidos en el PCB para iluminar el objeto a reconocer

Ventajas al utilizar un sensor de color Arduino TCS230:

- Alta resolución de conversión de la intensidad de luz a frecuencia.
- Tiempo rápido de respuesta
- Se comunica directamente con un microcontrolador (PIC, ARDUINO, etc.)

1.2.6 Bluetooth

Es una placa electrónica originalmente se basó en el ATmega 328, permite la comunicación serie inalámbrica a través de Bluetooth (pero no es compatible con auriculares Bluetooth). Contiene todo lo necesario para apoyar el microcontrolador y se puede programar de forma inalámbrica a través de la conexión Bluetooth. Este módulo es compatible con Arduino para comunicación serial Wireless. El shield funciona tanto con los módulos seriales por hardware y por software del Arduino. (Bluetooth, 2015)

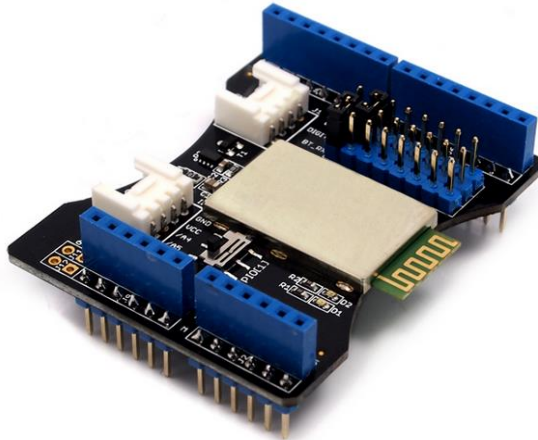


Figura 1.4 Bluetooth FCC ID: Z4T-BTSHIELD

Fuente:(Bluetooth, 2015)

El Arduino BT se puede programar con el software de Arduino, esto se lo hace descargando desde la plataforma libre.

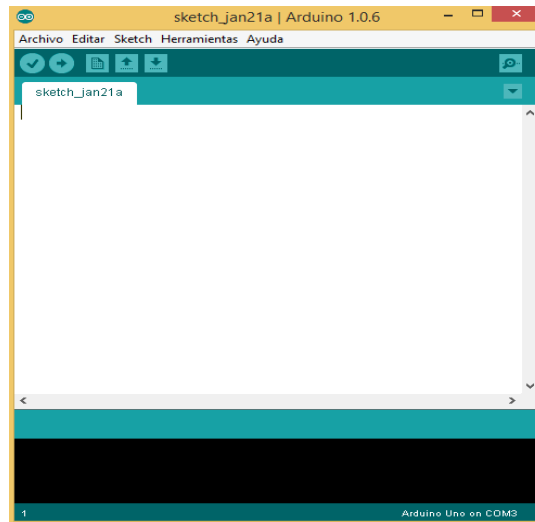


Figura 1.5 Plataforma de Programación Arduino

Fuente: (IDE Arduino, 2014)

Características

Compatible con Arduino

Alcance de hasta 10m sin obstáculos

Interfaz UART (TTL) con velocidad de transmisión programable (SPP firmware instalado)

Por defecto la velocidad de transmisión: 38.400, bits de datos: 8, bit de parada: 1,

Paridad: Sin paridad

CÓDIGO PIN predeterminado: "0000"

Un conjunto completo de comandos de configuración

Antena PCB On board

FCC ID certificada

Ventajas al utilizar una placa Arduino Bluetooth:

- IDE y lenguaje son fáciles de usar y aprender
- Tarjetas Arduino son económicas
- Software es open-source e independiente de la plataforma
- Hardware es open-source.
- Permite hacer todo tipo de proyectos

1.2.7 APP BlueTerm

BlueTerm permite controlar y comunicar un robot con una aplicación móvil a partir de un módulo Bluetooth. Dicha aplicación se puede descargar desde Play store con enfoque en móviles con plataforma Android. (BlueTerm, 2015)

Se empleará un software Arduino para desarrollar el programa de control del prototipo seleccionador y transportador de placas coloreadas, de esta manera se podrá ejecutar el sistema desde un celular moderno.

Adicionalmente, es un emulador que se conecta a cualquier dispositivo con puerto serie mediante un adaptador Bluetooth-serie, es una aplicación opensource que ayuda a enviar y recibir comandos rápidamente. (BlueTerm, 2015)

La principal ventaja es:

- Su fácil manipulación puede ser ejecutada desde cualquier móvil con acceso a Bluetooth.
- Se puede configurar cada tecla del mando para enviar el carácter deseado.

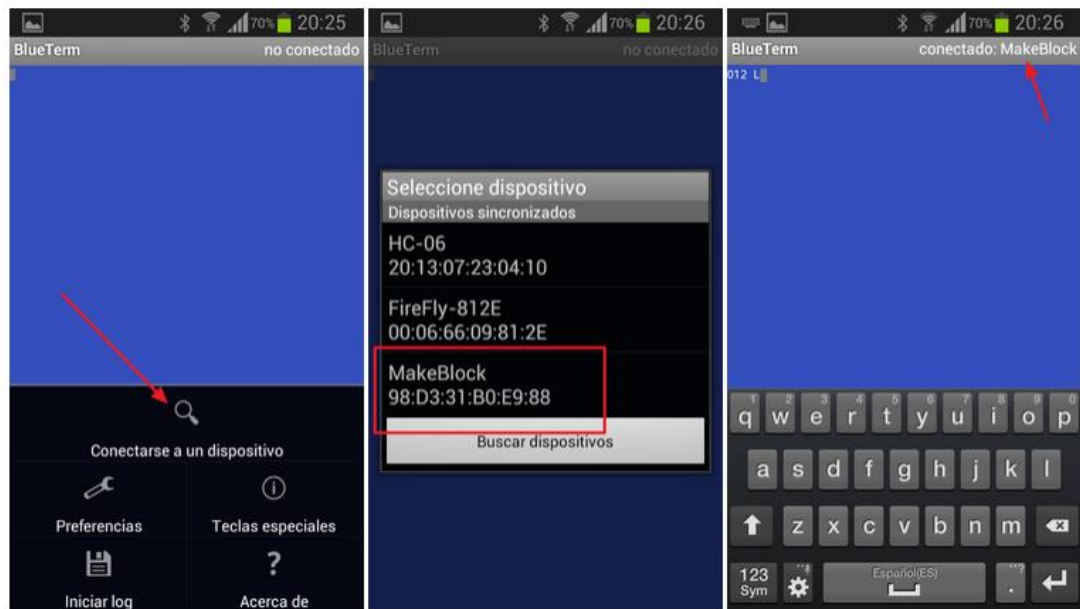


Figura 1.6 Aplicación BlueTerm (terminales que se conectan a cualquier dispositivo Bluetooth)

Fuente: (BlueTerm, 2015)

CAPÍTULO II

BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO PARA LA CONSTRUCCIÓN DEL SELECTOR Y TRANSPORTADOR DE PLACAS COLOREDAS

2.1 Problema Principal

La Universidad Tecnológica Israel no cuenta con proyectos reales construidos con la tecnología Arduino que sirvan como guía de estudio de este tipo de tecnología que recientemente se incorporó en el mercado ecuatoriano.

2.2 Propósito de los Objetivos

El objetivo principal se planteó con la finalidad de construir un prototipo autónomo seleccionador y transportador de placas coloreadas utilizando la plataforma libre hardware Arduino para aprovechar las ventajas que éste ofrece.

Los objetivos específicos se proyectaron con el fin de investigar los elementos necesarios para la implementación del proyecto de tal forma que sea de fácil mantenimiento, además para realizar el diseño que permitirá cumplir con todos los requerimientos necesarios para su funcionamiento.

2.3 Hipótesis

Al implementar el prototipo autónomo selector y transportador de placas coloreadas, activado por una aplicación Android para el laboratorio de electrónica de la Universidad Tecnológica Israel servirá como plataforma o guía de estudio en el laboratorio para los estudiantes.

Variable independiente: sistema autónomo y transportador de placas coloreadas.

Variable dependiente: servirá como guía de estudio para los estudiantes.

2.4. Fundamentación Teórica

Para el desarrollo del sistema se utilizó como base fundamental la tecnología Arduino implementada en la fase 2, que es la encargada de toda la parte lógica del producto,

pues a través del microcontrolador se realizan todas las actividades internas de mando, el sensor TCS230 Arduino permite identificar los diferentes colores de trabajo y por último la placa Bluetooth Arduino encargada de la comunicación serial. Todas estas actividades se las llevó a cabo mediante la programación de software libre Arduino.

2.5 Metodología de Investigación

Para realizar el proyecto fue necesario utilizar algunos métodos de investigación como son:

2.5.1 Método de análisis y síntesis

Este método se utilizó para adquirir información necesaria acerca de los dispositivos que se utilizaron en la construcción de prototipo seleccionador de placas coloreadas.

2.5.2 Método de modelación

El método de modelación se aplicó debido a que es necesario realizar un diseño electrónico en forma modular.

2.5.3 Método experimental

El método experimental se utilizó en la última etapa que es la elaboración ya que debido a la naturaleza del proyecto se deberá realizar pruebas y corrección de errores.

Se utilizó la técnica de la entrevista realizada a 5 estudiantes de la Carrera de Electrónica Digital y Telecomunicaciones de la Universidad Tecnológica Israel con el propósito de conocer los beneficios que se darán al construir el prototipo seleccionador y transportador de placas coloreadas.

El formato que se muestra a continuación es el que se utilizó para realizar las entrevistas y en el anexo 1 se puede observar las entrevistas realizadas.

UNIVERSIDAD TECNOLÓGICA ISRAEL
CARRERA DE ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

ENTREVISTA A LOS ESTUDIANTES DE LA UNIVERSIDAD TECNOLÓGICA ISRAEL

Nombre:

Cargo:

Fecha:

1.- ¿Con que regularidad utiliza usted los laboratorios de la Universidad Tecnológica Israel?

Siempre

Regularmente

Nunca

2.- ¿Piensa que los laboratorios cuentan con el equipamiento necesario para realizar las prácticas?

SI

NO

3.- ¿Los laboratorios disponen de proyectos reales que sirvan como guía de estudio?

SI

NO

4.- ¿Cree que este tipo de proyectos ayudarán a impulsar el interés académico?

SI

NO

Aquí un detalle de los resultados de la entrevista que se realizó a 5 estudiantes de la Universidad Tecnológica Israel.

Pregunta 1

La entrevista realizada a los estudiantes de la carrera de Electrónica Digital y Telecomunicaciones, aseguran que el 100% de los laboratorios se ocupan regularmente.

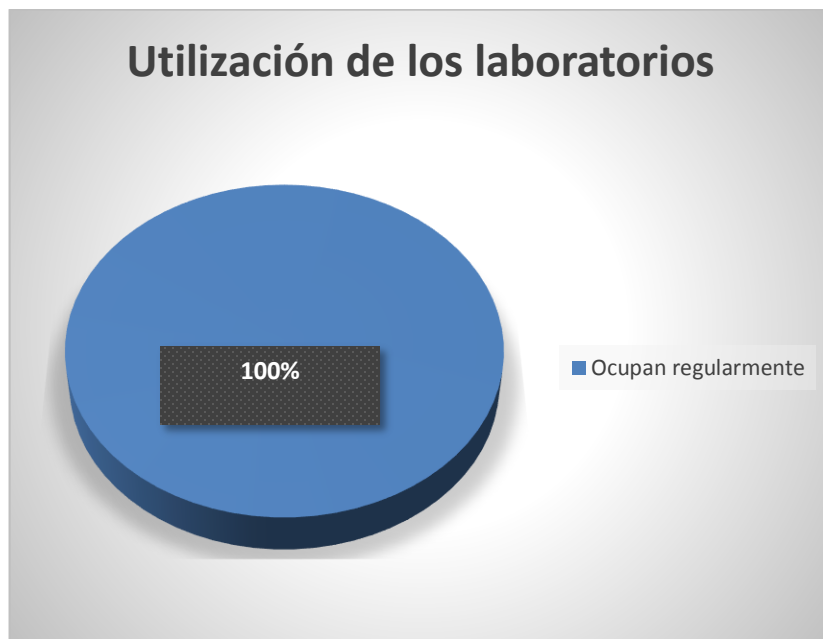


Figura 2.1 Gráfica de pastel referida a la entrevista (pregunta 1)

Fuente: (Elaboración propia)

Pregunta 2

- 80% de los estudiantes señalan que los laboratorios cuentan con el equipo necesario para realizar las prácticas.
- 20% de los estudiantes señalan que los laboratorios no cuentan con equipo necesario para realizar las prácticas.

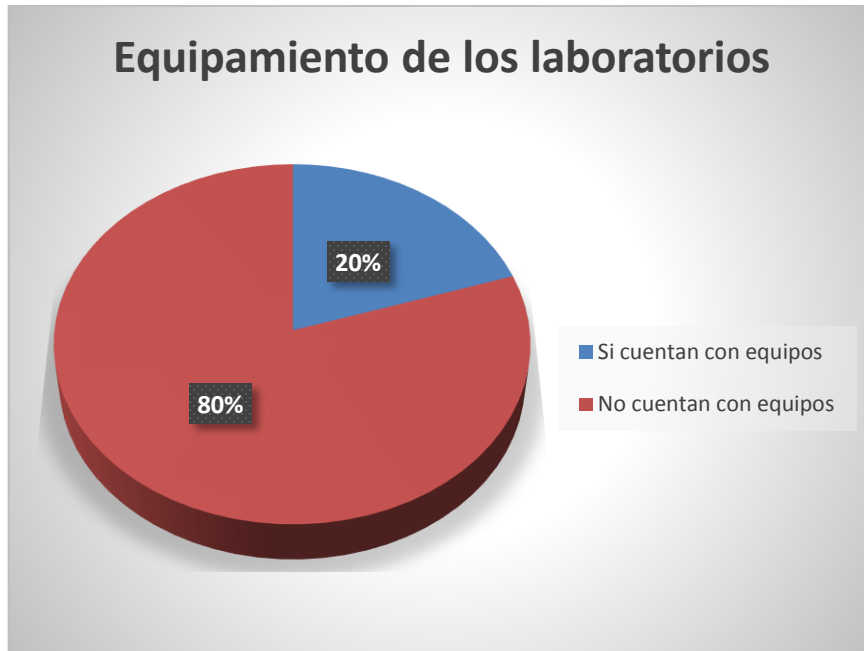


Figura 2.2 Gráfica de pastel referida a la entrevista (pregunta 2)

Fuente: (Elaboración propia)

Pregunta 3

Los estudiantes entrevistados afirman, que no existen proyectos reales que sirvan como guía de estudio en los laboratorios.

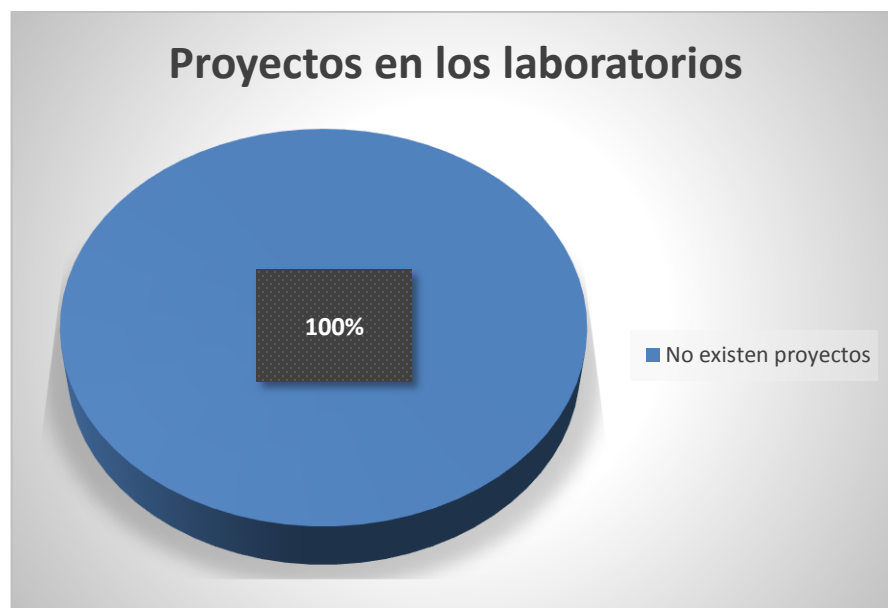


Figura 2.3 Gráfica de pastel referida a la entrevista (pregunta 3)

Fuente: (Elaboración propia)

Pregunta 4

Los 100 % estudiantes piensan que si es necesario la creación de nuevos proyectos de esta forma se impulsará el interés académico.



Figura 2.4 Gráfica de pastel referida a la entrevista (pregunta 4)

Fuente: (Elaboración propia)

2.6 Resultados Esperados

Con la implementación del seleccionador y transportador de placas coloreadas se espera, que sirva como guía de estudio para los estudiantes y también de alguna manera sea un incentivo para que poco a poco este tipo de proyectos aumente en los laboratorios de la Carrera de Electrónica.

CAPÍTULO III

PRESENTACIÓN DE RESULTADOS

3.1 Propuesta de Solución del Problema

Para la construcción del prototipo autónomo seleccionador y transportador de placas coloreadas se tomó en cuenta tres fases que facilitarán el progreso y desarrollo de este sistema: fase circuito de arranque, fase de control electrónico y fase electromecánica.

Este sistema está construido por partes para el fácil mantenimiento, utilizando la plataforma de Hardware libre Arduino que gestiona y controla toda la etapa de electromecánica, se realizará la programación en el microcontrolador Atmega328 y el tipo de lenguaje es en C, además se utilizará un shield Bluetooth para la comunicación serial con el dispositivo móvil Android.

3.2 Diseño del Prototipo

En la figura 3.1 se muestra las 3 fases que se utilizarán para el desarrollo del proyecto.

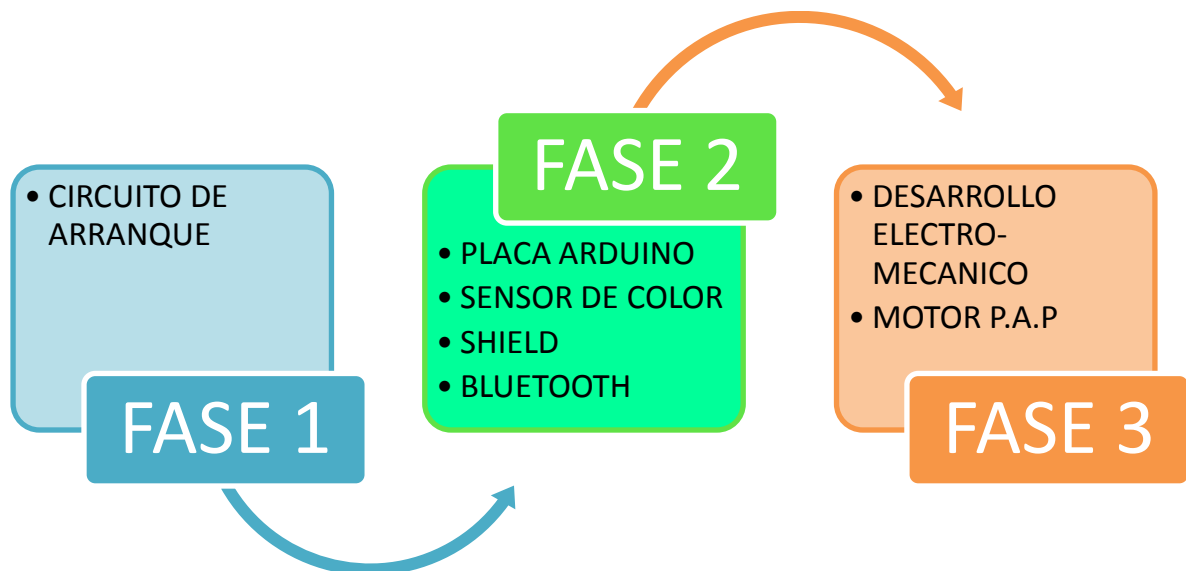


Figura 3.1 Diagrama General

Fuente: (Elaboración propia)

3.2.1 Fase 1

En esta fase están ubicados los pulsadores que permiten el arranque o inicio de todo el sistema, también se encuentran las fuentes encargadas de dar el voltaje necesario para la placa Arduino y el motor paso a paso, finalmente un led RGB que indicará el color con el que se trabajará.

Circuito de selección y arranque

En la figura 3.2 se muestra dos pulsadores de selección y de arranque conectados individualmente a una resistencia de 4k para que el voltaje que ingrese por cada uno ellos sea el adecuado permitiendo así conectar directamente al Arduino.

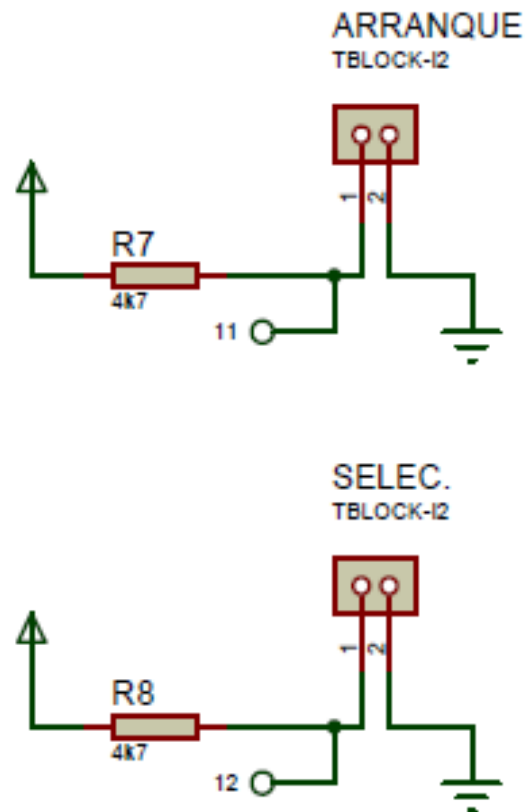


Figura 3.2 Fase de Selección y Arranque

Fuente: (Proteus – Investigador)

Circuito Regulador

A este circuito ingresa 24V los cuales no son adecuados para el funcionamiento del electroimán para ello se decidió por colocar un regulador LM7809 que permitirá reducir el voltaje y a su vez un disipador que permitirá extraer el calor que se provocará en el instante de arrancar el sistema.

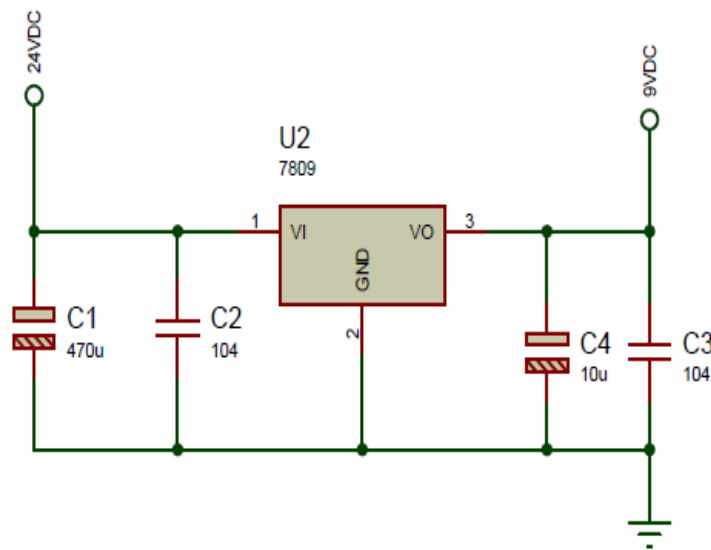


Figura 3.3 Regular de Voltaje
Fuente: (Proteus – Investigador)

3.2.2 Fase 2

En esta fase funcionará el módulo Arduino Uno, que es una placa con un microcontrolador basado en el chip Atmega328 que será programado de acuerdo con los requerimientos del proyecto, se trabajará con el TCS320 que es un sensor de color Arduino encargado de seleccionar el color deseado por el usuario, esta placa estará ubicada en la riel de la impresora cubierta de todo tipo de iluminación, se procederá a montar un Shield que tiene como finalidad monitorear la etapa de control del proyecto, siguiendo con el proceso ésta tendrá que estar montada a una siguiente placa Bluetooth que sirve para la comunicación y manipulación desde el dispositivo móvil.

Circuito Arduino

La figura 3.4 muestra la placa Arduino que se utilizará porque cumple con los requisitos para el funcionamiento del sistema, donde consta de un microcontrolador Atmega328 que será programado por medio del lenguaje C, para el arranque del sistema, tiene 14 entradas/salidas digitales , 6 entradas analógicas, conector USB, clavija hembra tipo Jack, conector ICSP y botón de reset.

En el anexo 2 se detalla el programa utilizado para el funcionamiento del selector y transportador de placas coloreadas.

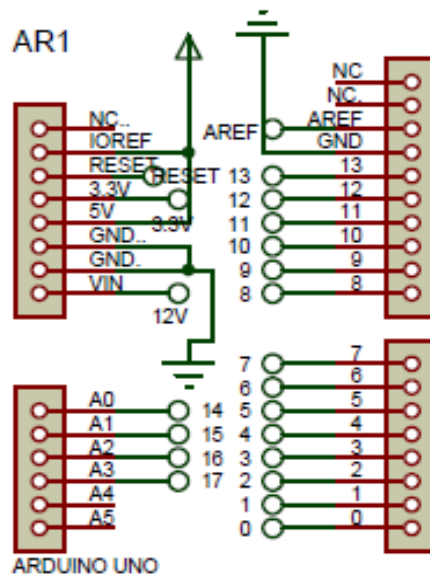


Figura 3.4 Circuito Arduino

Fuente: (Proteus – Investigador)

Circuito Sensor de color

El sensor TCS230 se utilizará para detección de color y operaciones de clasificación a diferencia de otros sensores de color que simplemente cuentan con salida de pulsos, este módulo permite además salida de voltaje analógico que permite un desarrollo e integración más sencillos y convenientes. Además, la función de calibración puede ayudar a la salida de datos más exactos.

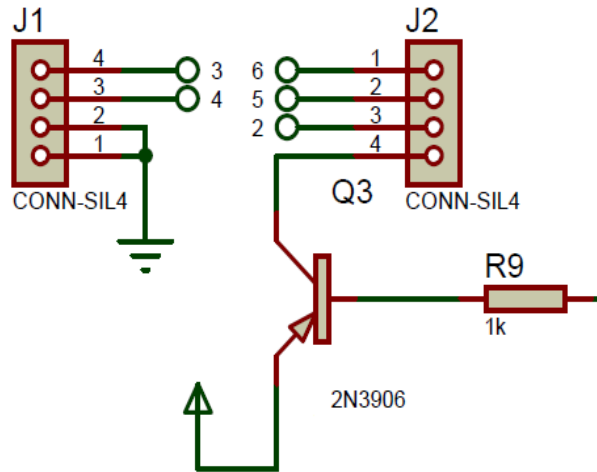


Figura 3.5 Circuito Sensor de Color
Fuente: (Proteus – Investigador)

Circuito Módulo Bluetooth

El módulo bluetooth tiene la opción de conectarse con cualquier dispositivo movil Android, la distribución de los pines 1 y 0 servirá para transmitir y recibir los datos, el pin 9 que es una señal de reloj indicará a la velocidad que trabaja el procesador y por último el pin 8 se utilizará para la comunicación serial.

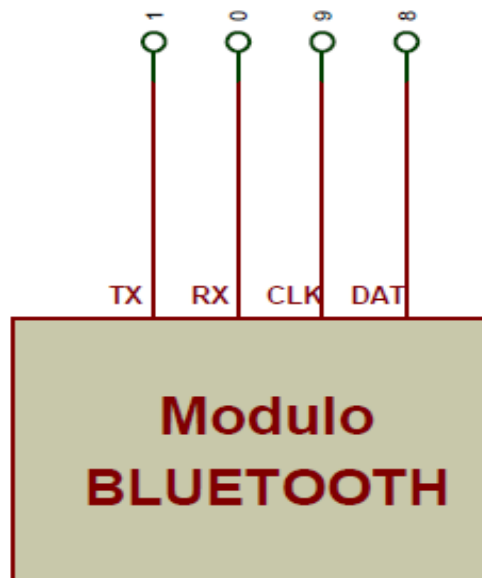


Figura 3.6 Circuito Módulo bluetooth
Fuente: (Proteus – Investigador)

3.2.3 Fase 3

En esta fase se encuentra una riel de impresora que servirá como base para montar la estructura donde irá colocada la placa del circuito con el sensor y el electroimán, este conjunto (sensor-electroimán) se desplazará a lo largo de cuatro posiciones, las primeras tres son donde el sensor buscará la placa con el color adecuado, una vez identificado, el electroimán se activará y llevará la placa a la cuarta posición, hecho esto el electroimán se apagará y dejará caer la placa en el lugar y regresa a la posición inicial.

Circuito de potencia

En la figura 3.7 se encuentra el circuito puente H que se encargará de mover y controlar la velocidad de giro al motor, para el funcionamiento se utilizará una fuente de 24V.

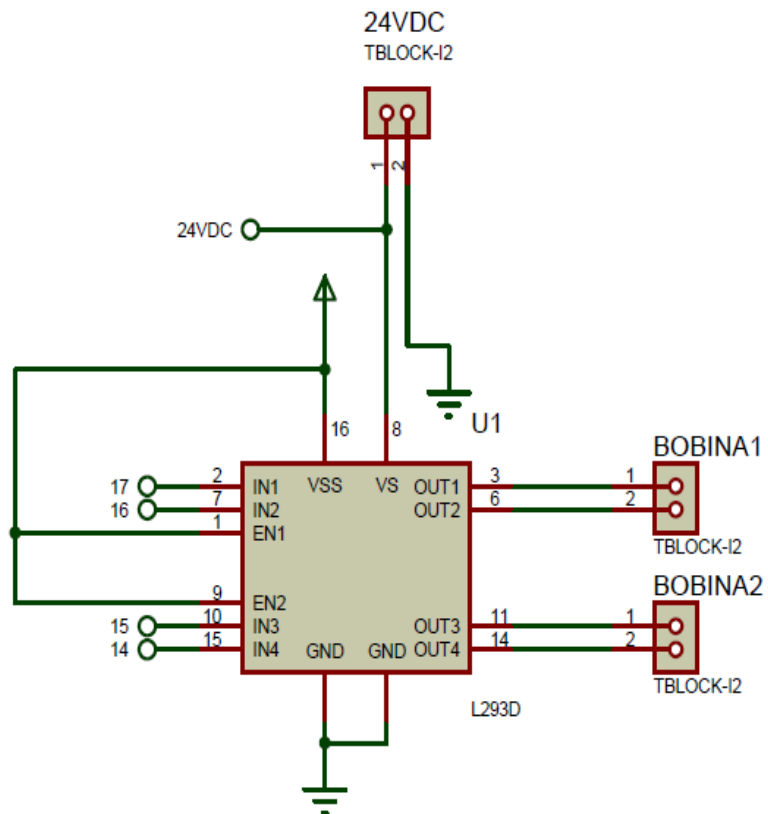


Figura 3.7 Circuito de Potencia – Puente H

Fuente: (Proteus – Investigador)

Circuito electroimán

Este circuito se activará mediante un voltaje de 9V al ingresar al circuito el transistor 2N3904 tiene la finalidad de polarizar el TIP122 y así poder alimentar el electroimán de esta forma se podrá coger y soltar la placa de color.

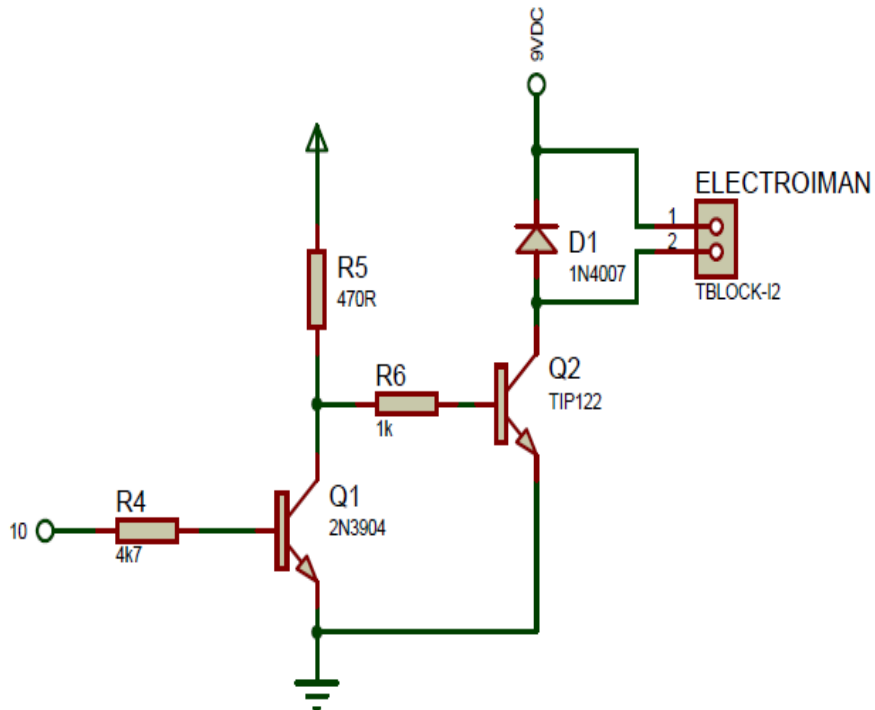


Figura 3.8 Circuito Electroimán
Fuente: (Proteus – Investigador)

3.2.4 Diagrama circuital general del proyecto

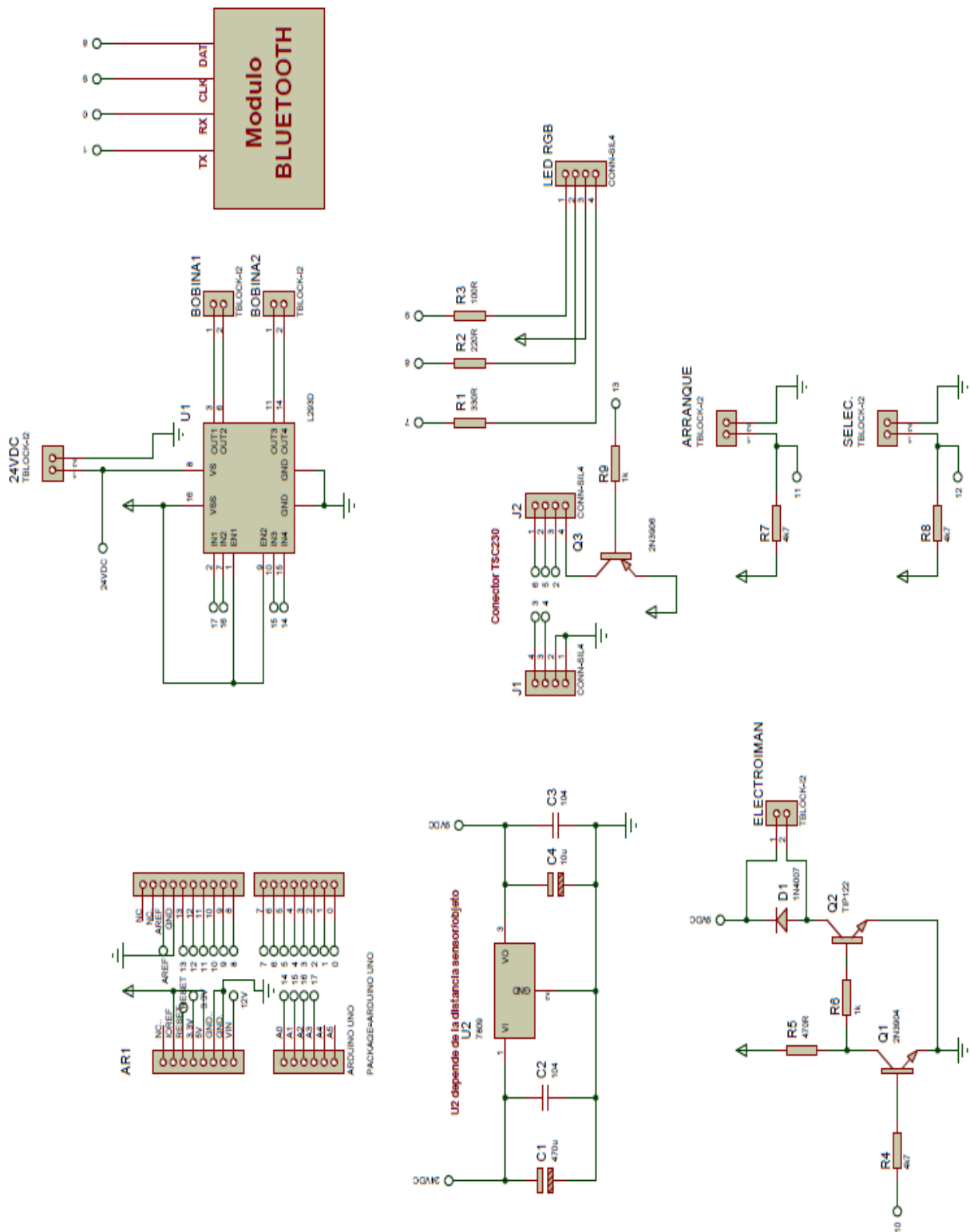


Figura 3.9 Diagrama General del Circuito

Fuente: (Proteus – Investigador)

3.2.5 Circuito impreso del proyecto

El circuito impreso se ejecutó en Proteus, un shield encargado de alimentar la parte de potencia, consta de pulsadores que permitirán la selección y arranque del sistema, tiene entradas y salidas para la conexión del motor y electroimán, además pines necesarios para ser montando a la placa Arduino y al Shield Bluetooth, la figura 3.10 muestra las pistas de conexión del shield complementario.

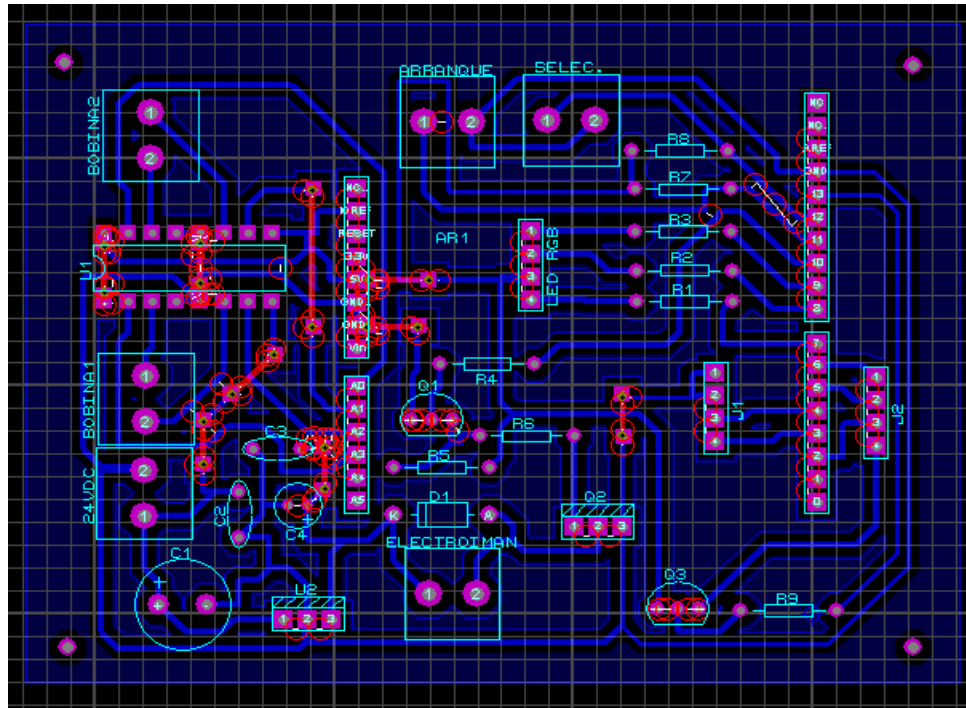


Figura 3.10 Circuito Impreso del proyecto

Fuente: (Proteus – Investigador)

La impresión del circuito es un shield complementario que fue necesario construir para que el sistema funcione de manera correcta.

3.2.6 Circuito impreso Bluetooth FC ID:Z4T

La figura 3.11 muestra el circuito donde se puede apreciar las partes elementales que se ocuparán para la comunicación con la placa Arduino, el PIO que es un puerto de instrucciones del módulo Bluetooth puede ser leído por el puerto Arduino A1: bajo desconectado, alto conectado.

Rx: Entrada de datos del módulo Bluetooth.

TX: Salida de datos módulo Bluetooth.

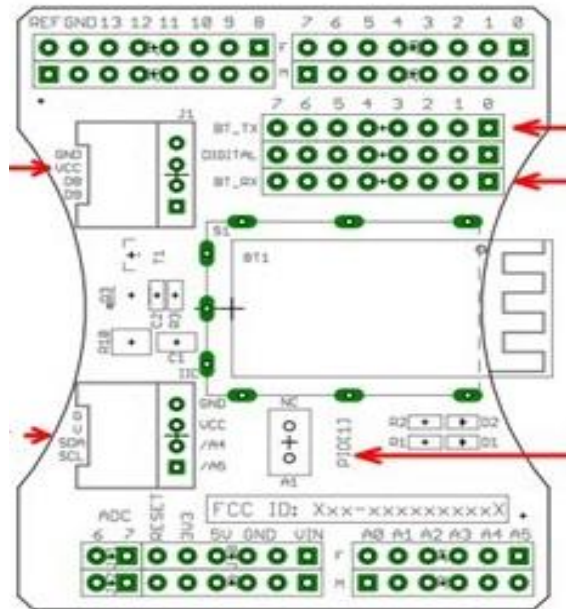


Figura 3.11 Circuito Impreso del Módulo Bluetooth

Fuente: (Investigador)

3.2.7 Elementos utilizados para la construcción del prototipo

| ITEM | DESCRIPCIÓN | CANTIDAD |
|------------------------|-------------|----------|
| Bornera 2 Pines | | 4 |
| C.I. L293d | Bridge | 1 |
| Cond. 470mf 35v | Capacitor | 1 |
| Zócalo 16 Pines | Conector | 1 |
| Cond. 0.1mf 50v | Capacitor | 2 |
| Cond. 10mf 16v | Capacitor | 1 |
| C.I. 7805cv | | 1 |
| Trans. 2n3906 Ecg159 | Transistor | 1 |
| Trans. 2n3904 Ecg123ap | Transistor | 1 |
| Rest. 4k Ohm 1/4w | Resistencia | 3 |
| Rest. 1k Ohm 1/4w | Resistencia | 2 |
| Rest. 330 Ohm 1/4w | Resistencia | 1 |

| | | |
|--------------------|----------------|---|
| Rest. 220 Ohm 1/4w | Resistencia | 1 |
| Rest. 100 Ohm 1/4w | Resistencia | 1 |
| Rest. 470 Ohm 1/4w | Resistencia | 1 |
| Diodo 1n4007 | Diodo | 1 |
| Trans. Tip122 | Transistor | 1 |
| Pulsador Tc11322mm | Pulsador | 3 |
| Led RGB 5mm 4pines | Indicador | 1 |
| Disipador | Ventilación | 1 |
| Placa Arduino Uno | Versión 3.0 | 1 |
| Sensor de Color | TCS230 | 1 |
| Bluetooth | FCC ID: Z4T-BT | 1 |

3.3 Implementación del Proyecto

En la figura 3.12 se puede apreciar la placa complementaria (Baquelita) que se construyó con las dimensiones adecuadas para que los pines de conexión coincidan con el Shield Arduino.

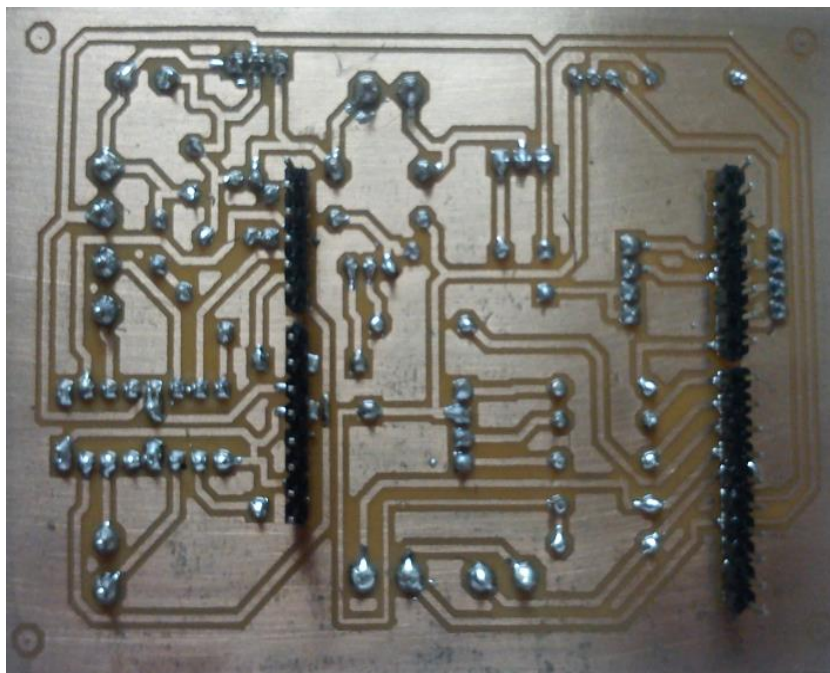


Figura 3.12 PCB Placa Complementaria

Fuente: (Elaboración propia)

En la figura 3.13 se observa los elementos que constan en la placa complementaria.

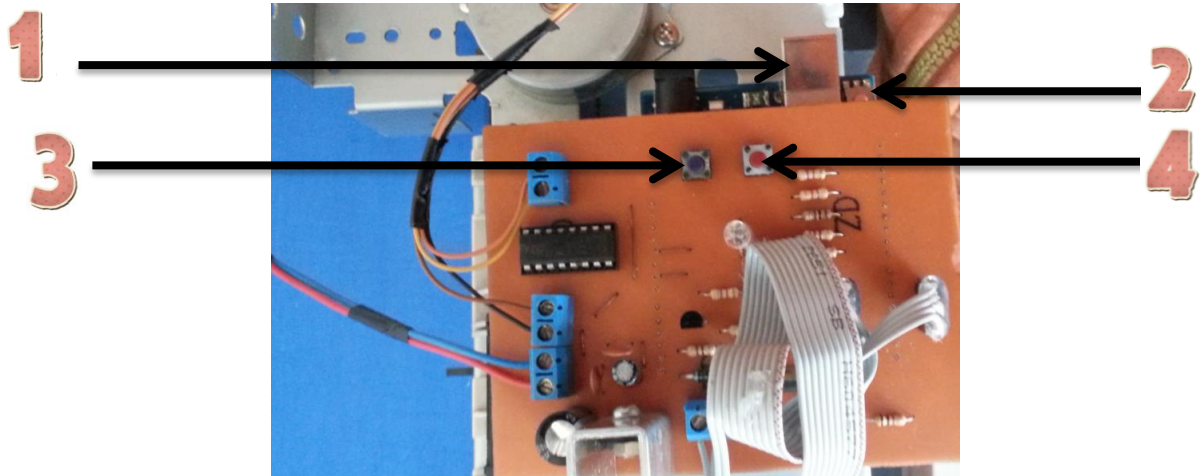


Figura 3.13 Elementos del Circuito Complementario

Fuente: (Elaboración propia)

Descripción:

- 1.- Conector USB
- 2.- Botón de reset
- 3.- Botón de selección de color
- 4.- Botón de arranque

La figura 3.14 muestra la placa Arduino con sus pines de conexión que se utilizan para montar el shield complementario y el módulo Bluetooth.

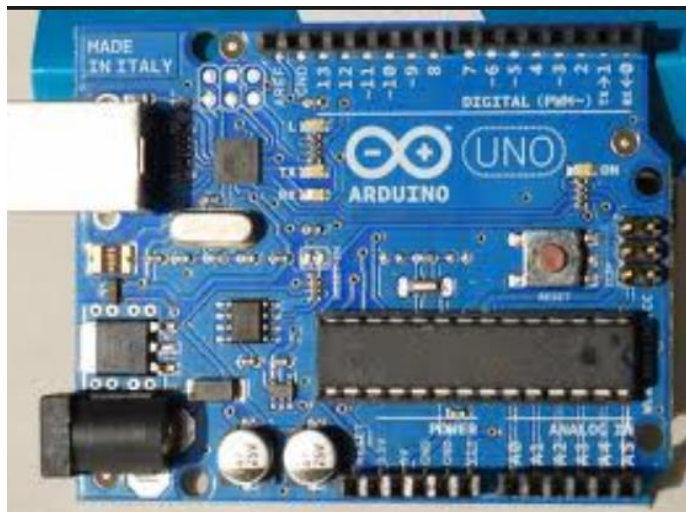


Figura 3.14 Placa Arduino

Fuente: (Arduino Uno, 2015)

La Figura 3.15 muestra el sensor de color TSC230, conectado al interior de la impresora evitando que la luz del exterior interfiera al momento de sensor el color deseado.



Figura 3.15 Sensor de Color TSC230

Fuente: (Sensor de color TCS230 electronilab, 2015)

La Figura 3.16 muestra la parte inferior del sistema, conectado a los respectivos elementos a un riel de impresora.



Figura 3.16 Parte Interna del Sistema

Fuente: (Elaboración propia)

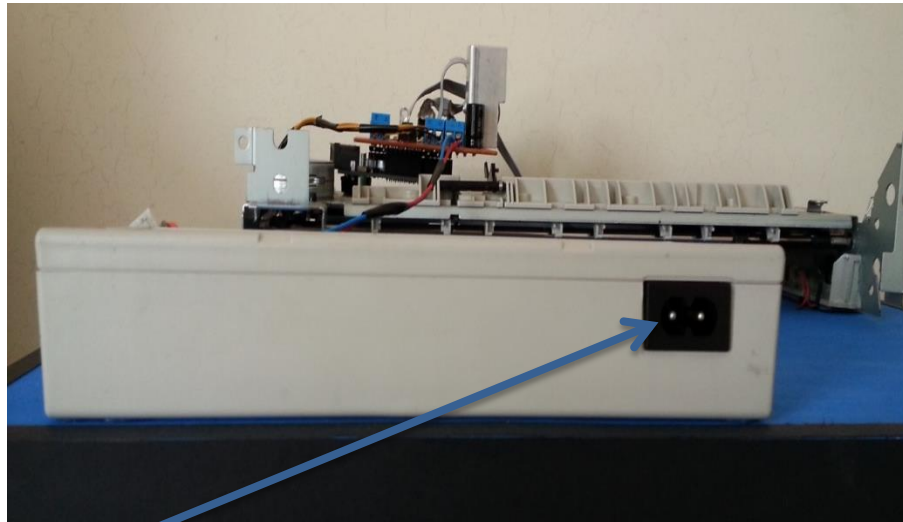
Descripción:

1.- Electroimán

2.- Sensor de color

3.- Riel

La figura 3.17 muestra la parte trasera del sistema, se utiliza una fuente proveniente de la impresora para el arranque del motor.



1

Figura 3.17 Vista Trasera del Sistema

Fuente: (Elaboración propia)

Descripción:

1.- Conector 110V

La figura 3.18 muestra la vista frontal del sistema y las partes principales que las conforman además se puede apreciar la conexión del Bluetooth y el shield complementario hacia la placa Arduino.

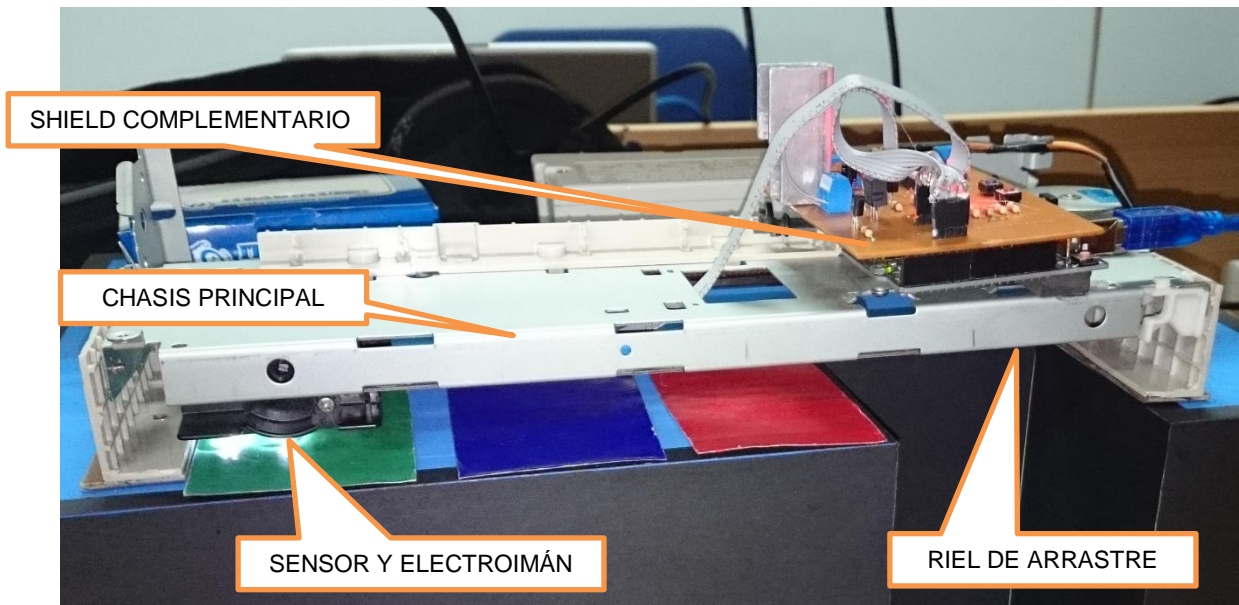


Figura 3.18 Vista Frontal del Sistema

Fuente: (Elaboración propia)

La figura 3.19 muestra la vista lateral del sistema conformado por un Led RGB con los tres colores que se eligen el instante de presionar el botón de selección.

Adicionalmente un servomotor de 4 hilos y la placa Arduino conectada al shield complementario necesario para el funcionamiento del prototipo.

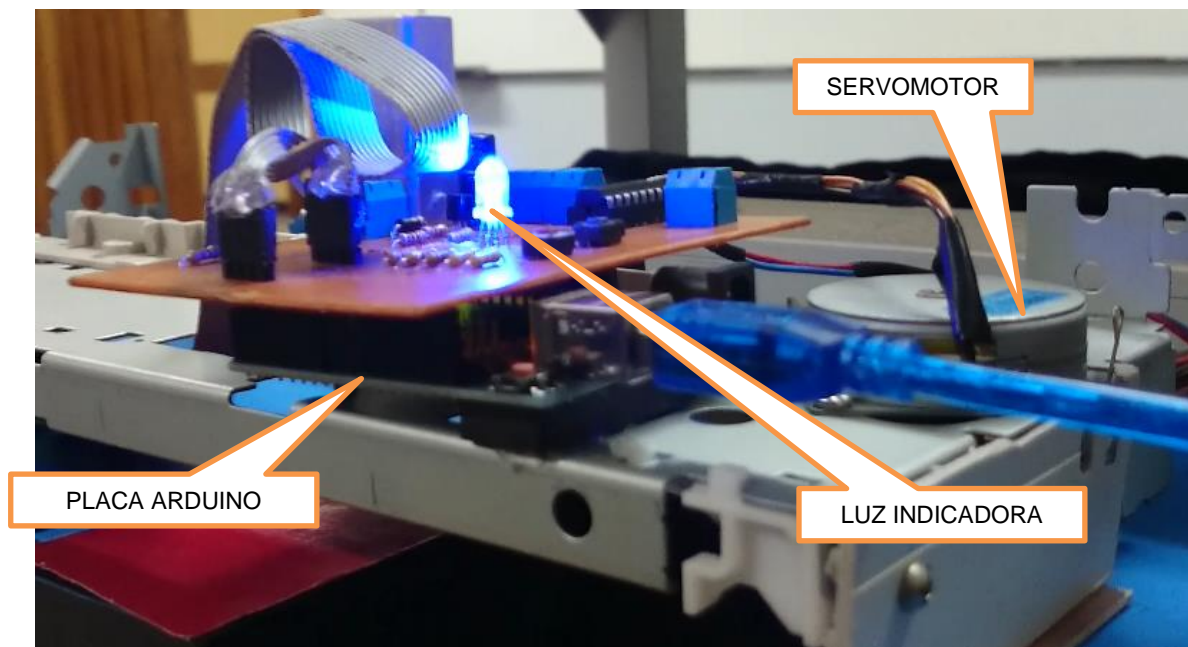


Figura 3.19 Vista Lateral del Sistema

Fuente: (Elaboración propia)

La figura 3.20 muestra el proyecto terminado y las conexiones de los shields.

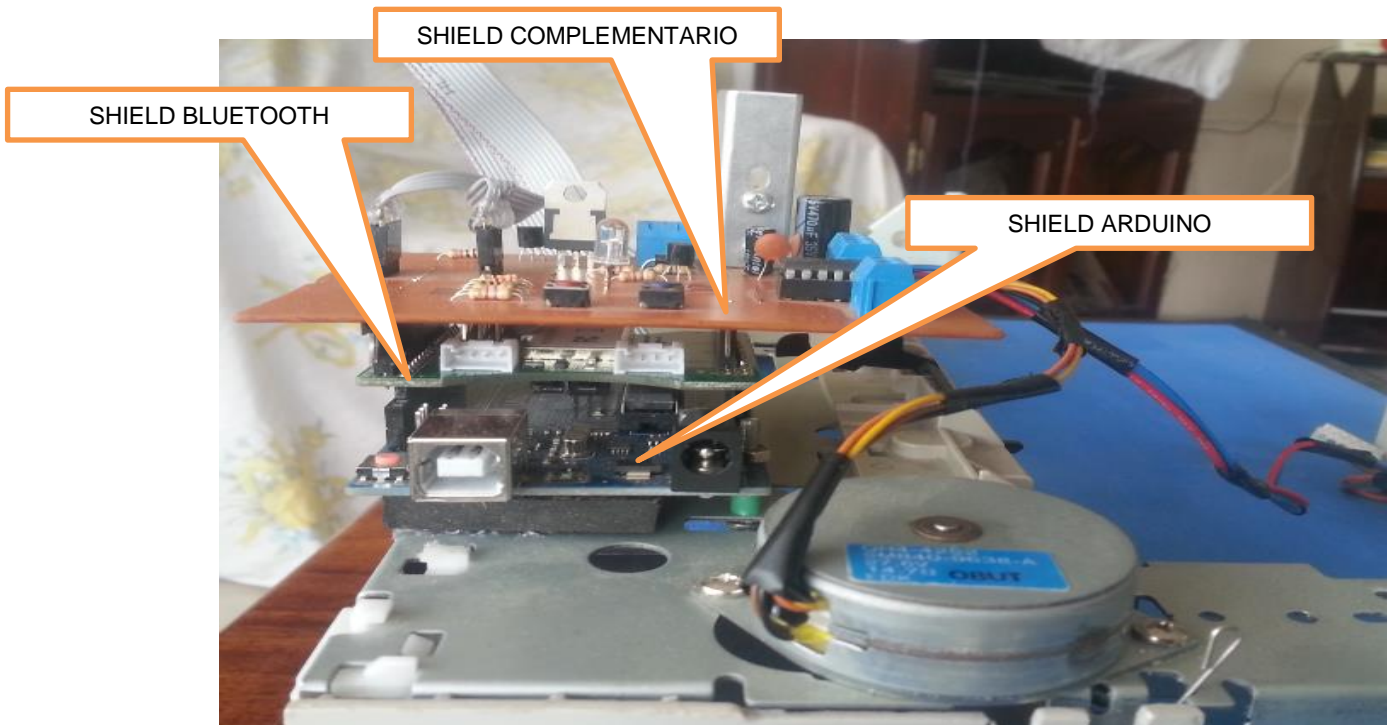


Figura 3.20 Vista Lateral de las Placas

Fuente: (Elaboración propia)

La figura 3.21 muestra el producto final.



Figura 3.21 Producto final

Fuente: (Elaboración propia)

3.4 Evaluación Técnica

La evaluación técnica realizada al prototipo seleccionador y transportar de placas coloreadas se las hizo con la finalidad de no tener ningún problema el instante de arrancar el sistema ya sea manualmente o por medio de un dispositivo móvil.

En la tabla 3.1 se detalla la evaluación técnica realizadas a cada una de las partes que conforman el proyecto.

Tabla 3.1 Evaluación Técnica

| ÍTEM | DESCRIPCIÓN/STATUS | |
|-------------------------------|-----------------------------|--------------|
| Fuente Principal Motor | VOLTAJE | CALIFICACIÓN |
| | 24 V | 100% |
| Fuente Placa Arduino Uno | ENCENDIDO | CALIFICACIÓN |
| | CORRECTO | 100% |
| Luces Indicadoras | ENCENDIDO/APAGADO | CALIFICACIÓN |
| | CORRECTO | 100% |
| Panel De Control (Pulsadores) | ON/OFF | CALIFICACIÓN |
| | CORRECTO | 100% |
| Controlador Del Servomotor | VOLTAJES Y CORRIENTES | CALIFICACIÓN |
| | VALORES NORMALES | 100% |
| Microcontrolador Arduino Uno | ENCENDIDO Y PROGRAMACIÓN | CALIFICACIÓN |
| | CORRECTO | 100% |
| Servomotor | BOBINAS Y GIRO DE ROTOR | CALIFICACIÓN |
| | CORRECTO | 100% |
| Sensor De Color | ENCENDIDO Y SALIDA DE DATOS | CALIFICACIÓN |
| | CORRECTO | 100% |
| Electroimán | ENCENDIDO Y FUNCIONANDO | CALIFICACIÓN |
| | CORRECTO | 100% |
| Puertos De Conexión | FIJOS Y CONECTADOS | CALIFICACIÓN |
| | CORRECTO | 100% |
| Riel De Desplazamiento | DESPLAZAMIENTO | CALIFICACIÓN |
| | SIN OBSTÁCULOS | 100% |
| Chasis Principal | DEFORMACIONES | CALIFICACIÓN |
| | NINGUNA | 100% |
| Módulo Bluetooth | ENCENDIDO Y SALIDA DE DATOS | CALIFICACIÓN |
| | CORRECTO | 100% |

Fuente: (Elaboración propia)

3.5 Pruebas de funcionamiento

Para la validación del funcionamiento del prototipo se realizaron las siguientes pruebas.

3.5.1 Prueba 1: Identificación y selección de placas coloreadas

Tabla 3.2 Identificación y selección de placas coloreadas

| PLACA | SENSOR DE COLOR | | | POSICIÓN 4 (Depósito) |
|-------|-----------------|------------|------------|--------------------------|
| | POSICIÓN 1 | POSICIÓN 2 | POSICIÓN 3 | |
| | ROJO | VERDE | AZUL | ✓ |
| ROJO | AZUL | ROJO | VERDE | ✓ |
| | VERDE | AZUL | ROJO | ✓ |
| | ROJO | VERDE | AZUL | ✓ |
| VERDE | AZUL | ROJO | VERDE | ✓ |
| | VERDE | AZUL | ROJO | ✓ |
| | ROJO | VERDE | AZUL | ✓ |
| AZUL | AZUL | ROJO | VERDE | ✓ |
| | VERDE | AZUL | ROJO | ✓ |

Fuente: (Elaboración propia)

3.5.1 Prueba 2: Distancia entre las placas y el electroimán

Tabla 3.3 Distancia entre las placas y el electroimán

| ELECTROIMÁN | |
|-------------|--------|
| DISTANCIA | ESTADO |
| 1 cm | activo |
| 2cm | pasivo |
| 1cm | activo |
| 1,5 cm | pasivo |
| 3 cm | pasivo |
| 5 cm | pasivo |
| 1 cm | activo |
| 6 cm | pasivo |
| 2 cm | pasivo |
| 1,5 cm | pasivo |

Fuente: (Elaboración propia)

3.6 Análisis de Resultados

3.6.1 Análisis de Resultados de la Prueba 1

Se realizaron 3 diferentes tipos de pruebas con cada placa de color, donde se comprobó que sin importar la ubicación de las placas el sensor identificó la placa de color seleccionada y la trasladó a su posición final.

3.6.2 Análisis de Resultados de la Prueba 2

Se realizaron 10 pruebas para conocer la distancia adecuada de trabajo del electroimán a las placas coloreadas, de las cuales se pudo apreciar que la distancia adecuada es de 1cm, pues sólo a esa distancia el electroimán se activa y sujeta la placa, caso contrario si se aleja tan sólo 1mm su trabajo deja de ser eficiente y no se garantiza que sujete la placa.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Se construyó el prototipo seleccionador y transportador de placas coloreadas para el laboratorio de Electrónica de la Universidad Tecnológica Israel con el fin de que sirva como guía de estudio para los estudiantes.
- El uso de partes recicladas además de ayudar el medio ambiente facilita la construcción del proyecto.
- Al elaborar estos proyectos únicamente se debe considerar los requerimientos del sistema (en este caso un proyecto electrónico) porque al incorporar más características al prototipo esto puede entorpecer el correcto funcionamiento para el que fue concebido.
- La utilización de hardware prefabricado optimiza la ejecución de este tipo de proyectos ahorrando tiempo y recursos.
- La ubicación de las placas coloreadas debe ser exacta, pues de no ser así causarían errores al momento de funcionar el sistema.
- Tomar en cuenta que para la activación con la aplicación Android el teléfono debe tener visible el bluetooth, caso contrario no podrá realizarse la conexión.

Recomendaciones

- Si se requiere disponer de más posiciones de selección de placas coloreadas se puede modificar la programación teniendo en cuenta el tamaño del objeto a escoger.
- Si se estropea la placa microcontroladora ARDUINO no necesariamente tiene que ser reemplazada por una similar, hay una familia de Chips compatibles.
- Si se desea cambiar los códigos de selección para la comunicación Bluetooth se lo puede hacer desde el programa interno.
- Si se desea aumentar un control vía SMS se debería aumentar un modem GSM Arduino.
- El electroimán podría ser reemplazado por un brazo mecánico, en caso de que se requiera una distancia más grande entre las placas coloreadas y el sensor.
- Para que el sistema se conserve y su ejecución sea el adecuado realizar un mantenimiento anual.

BIBLIOGRAFÍA

- Arduino Uno.* (02 de enero de 2015). Obtenido de <http://arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardUno>
- Arduino Uno.* (02 de enero de 2015). Obtenido de https://www.google.com.ec/search?q=arduino+uno&biw=1366&bih=657&tbm=isch&tbo=u&source=univ&sa=X&ei=8KDTVKLLIYzsggTEIiHYCQ&ved=0CDsQsAQ#imgdii=_&imgrc=Cbj4eEjlm_kj3M%253A%3BmC1VLEIMmTMXPM%3Bhttp%253A%252F%252Farduino.cc%252Fen%252Fuploads%252FMain%252FArdu:
<http://arduino.cc/en/Main/arduinoBoardUno>
- Arduino Uno.* (02 de enero de 2015). Obtenido de www.arduino.cc/es/pmwiki.php?n=:
<http://www.arduino.cc/es/pmwiki.php?n=:>
- Atmega 328.* (05 de enero de 2015). Obtenido de https://www.google.com.ec/search?q=tabla+caracteristica+atmega328+arduino&biw=1366&bih=657&tbm=isch&tbo=u&source=univ&sa=X&ei=nZPTVKzYAZKKgwT_oICoBw&ved=0CBsQsAQ&dpr=1#imgdii=_&imgrc=3XMAHgLTGfC2M%253A%3B9ByhuvlwAfynaM%3Bhttp%253A%252F%252Fwww.vb-mundo.c:
http://mindstormsarduino.blogspot.com/2010_11_01_archive.html
- BlueTerm.* (02 de enero de 2015). Obtenido de <https://play.google.com/store/apps/details?id=es.pymasde.blueterm.:>
https://play.google.com/store/apps/details?id=es.pymasde.blueterm&hl=es_419
- Bluetooth.* (02 de enero de 2015). Obtenido de https://www.google.com.ec/search?q=Bluetooth&biw=1366&bih=657&source=Inms&tbm=isch&sa=X&ei=_pjTVKH1K4GngwT9oYCIBQ&ved=0CAYQ_AUoAQ#tbn=isch&q=Bluetooth+seeed+studio&spell=1&imgdii=_&imgrc=INUTYQnff2OIM%253A%3BhUq5Ge7L2nQJ7M%3Bhttp%253A%252F%252Fwww.seeeds:
http://www.seeedstudio.com/wiki/Bluetooth_Shield
- Colímetro TCS230.* (29 de noviembre de 2011). Obtenido de www.bitacoradealfon.es/TCS230_arduino.php:
http://www.bitacoradealfon.es/TCS230_arduino.php
- Electroiman.* (05 de enero de 2015). Obtenido de <https://explorable.com/es/construye-un-electroiman:>
<https://explorable.com/es/construye-un-electroiman>
- IDE Arduino.* (22 de julio de 2014). Obtenido de https://www.google.com.ec/search?q=Bluetooth&biw=1366&bih=657&source=Inms&tbm=isch&sa=X&ei=_pjTVKH1K4GngwT9oYCIBQ&ved=0CAYQ_AUoAQ#tbn=isch&q=Plataforma+de+Programaci%C3%B3n+Arduino&imgdii=_&imgrc=yVcgx98duP-PcM%253A%3BUXx2PAWPOXYauM%3Bhttp%253A%252F%252F:
<http://estudiomiranda.com/?p=27>
- Sensor de color TCS230 electronilab.* (02 de enero de 2015). Obtenido de [electronilab.co/tienda/sensor-de-color-tcs230/:](http://electronilab.co/tienda/sensor-de-color-tcs230/) <http://electronilab.co/tienda/sensor-de-color-tcs230/>
- Servomotores.* (18 de julio de 2009). Obtenido de [www.neoteo.com/servomotores-el-primer-paso-hacia-tu-robot/:](http://www.neoteo.com/servomotores-el-primer-paso-hacia-tu-robot/) <http://www.neoteo.com/servomotores-el-primer-paso-hacia-tu-robot/>
- Shields Arduino.* (07 de diciembre de 2010). Obtenido de [ayudaelectronica.com/%C2%BFque-son-los-arduino-shields/:](http://ayudaelectronica.com/%C2%BFque-son-los-arduino-shields/) <http://ayudaelectronica.com/%C2%BFque-son-los-arduino-shields/>

Shields Arduino Uno. (05 de enero de 2015). Obtenido de https://www.google.com.ec/search?q=shields+arduino+uno&biw=1366&bih=657&tbm=isch&imgil=OkTG9CnOt8VjuM%253A%253B84pW9_H1aec8bM%253Bhttp%25253A%25252F%25252Fwww.freertronics.com.au%25252Fpages%25252Fstacking-arduino-shields&source=iu&pf=m&fir=OkTG9CnOt8VjuM%253A%253B84pW9_H1aec8bM%253Bhttp%25253A%25252F%25252Fwww.freertronics.com.au%25252Fpages%25252Fstacking-arduino-shields#:
<http://www.freertronics.com.au/pages/stacking-arduino-shields#.VNOVjS6-PNk>

Ventajas Arduino. (02 de enero de 2015). Obtenido de [www.todoelectronica.com/¿por-que-arduino-ventajas-sobre-otros-microc...:](http://www.todoelectronica.com/¿por-que-arduino-ventajas-sobre-otros-microc...)
<http://www.todoelectronica.com/%C2%BFpor-que-arduino-ventajas-sobre-otros-microcontroladores-p-13507.html>

ANEXOS

ANEXO1

**Encuesta a los estudiantes
de la “Universidad
Tecnológica Israel”**

UNIVERSIDAD TECNOLÓGICA ISRAEL
CARRERA DE ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

ENTREVISTA A LOS ESTUDIANTES DE LA UNIVERSIDAD TECNOLÓGICA ISRAEL

Nombre: Fernando Saura

Cargo: Estudiante

Fecha: 19/01/2015

1.- ¿Con que regularidad utiliza usted los laboratorios de la Universidad Tecnológica Israel?

Siempre

Regularmente

Nunca

2.- ¿Piensa que los laboratorios cuentan con el equipamiento necesario para realizar las prácticas?

SI

NO

3.- ¿Los laboratorios disponen de proyectos reales que sirvan como guía de estudio?

SI

NO

4.- ¿Cree que este tipo de proyectos ayudarán a impulsar el interés académico?

SI

NO

UNIVERSIDAD TECNOLÓGICA ISRAEL
CARRERA DE ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

ENTREVISTA A LOS ESTUDIANTES DE LA UNIVERSIDAD TECNOLÓGICA ISRAEL

Nombre: Mayra Yaguana

Cargo: Estudiante

Fecha: 19 de Enero del 2015

1.- ¿Con que regularidad utiliza usted los laboratorios de la Universidad Tecnológica Israel?

Siempre

Regularmente

Nunca

2.- ¿Piensa que los laboratorios cuentan con el equipamiento necesario para realizar las prácticas?

SI

NO

3.- ¿Los laboratorios disponen de proyectos reales que sirvan como guía de estudio?

SI

NO

4.- ¿Cree que este tipo de proyectos ayudarán a impulsar el interés académico?

SI

NO

UNIVERSIDAD TECNOLÓGICA ISRAEL
CARRERA DE ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

ENTREVISTA A LOS ESTUDIANTES DE LA UNIVERSIDAD TECNOLÓGICA ISRAEL

Nombre: Jose Miguel Robel.....

Cargo: Estudiante.....

Fecha: 19/01/2015.....

1.- ¿Con que regularidad utiliza usted los laboratorios de la Universidad Tecnológica Israel?

Siempre

Regularmente

Nunca

2.- ¿Piensa que los laboratorios cuentan con el equipamiento necesario para realizar las prácticas?

SI

NO

3.- ¿Los laboratorios disponen de proyectos reales que sirvan como guía de estudio?

SI

NO

4.- ¿Cree que este tipo de proyectos ayudarán a impulsar el interés académico?

SI

NO

UNIVERSIDAD TECNOLÓGICA ISRAEL
CARRERA DE ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

ENTREVISTA A LOS ESTUDIANTES DE LA UNIVERSIDAD TECNOLÓGICA ISRAEL

Nombre: *Carlos Vasquez*

Cargo: *Estudiante*

Fecha: *19/01/2015*

1.- ¿Con que regularidad utiliza usted los laboratorios de la Universidad Tecnológica Israel?

Siempre

Regularmente

Nunca

2.- ¿Piensa que los laboratorios cuentan con el equipamiento necesario para realizar las prácticas?

SI

NO

3.- ¿Los laboratorios disponen de proyectos reales que sirvan como guía de estudio?

SI

NO

4.- ¿Cree que este tipo de proyectos ayudarán a impulsar el interés académico?

SI

NO

UNIVERSIDAD TECNOLÓGICA ISRAEL
CARRERA DE ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

ENTREVISTA A LOS ESTUDIANTES DE LA UNIVERSIDAD TECNOLÓGICA ISRAEL

Nombre: Aracely Chicarja

Cargo: Estudiante

Fecha: 19-01-2015

1.- ¿Con que regularidad utiliza usted los laboratorios de la Universidad Tecnológica Israel?

Siempre

Regularmente

Nunca

2.- ¿Piensa que los laboratorios cuentan con el equipamiento necesario para realizar las prácticas?

SI

NO

3.- ¿Los laboratorios disponen de proyectos reales que sirvan como guía de estudio?

SI

NO

4.- ¿Cree que este tipo de proyectos ayudarán a impulsar el interés académico?

SI

NO

ANEXO 2

Programa Arduino


```
pinMode(Clkpin, OUTPUT); // Setup the RGB LED Clock pin
```

```
//$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
pinMode(ledr, OUTPUT);
```

```
pinMode(ledg, OUTPUT);
```

```
pinMode(ledb, OUTPUT);
```

```
pinMode(A, OUTPUT);
```

```
pinMode(B, OUTPUT);
```

```
pinMode(C, OUTPUT);
```

```
pinMode(D, OUTPUT);
```

```
pinMode(trans, OUTPUT);
```

```
pinMode(su1, INPUT);
```

```
pinMode(su2, INPUT);
```

```
pinMode(electroiman, OUTPUT);
```

```
Serial.begin(115200);
```

```
pinMode(s0,OUTPUT);
```

```
pinMode(s1,OUTPUT);
```

```
pinMode(s2,OUTPUT);
```

```
pinMode(s3,OUTPUT);
```

```
digitalWrite(ledr, HIGH);
```

```
digitalWrite(ledg, HIGH);
```

```
digitalWrite(ledb, HIGH);
```

```
digitalWrite(electroiman, HIGH);
```

```
for (int x = 0;x<200; x++)
```

```
{
```

```
delay(20);
```

```
dato1 = digitalRead(su1); //almacenamos en dato el valor
```

```
        ;//del suiche

if (dato1 == LOW ){//si dato es alto
    delay(200);
    colorled=colorled+1;
}
else
{
if (colorled==1){
    digitalWrite(ledr, LOW);
    digitalWrite(ledg, HIGH);
    digitalWrite(ledb, HIGH);
    bandera=0;
    }

if (colorled==2){
digitalWrite(ledr, HIGH);
digitalWrite(ledg, LOW);
digitalWrite(ledb, HIGH);
bandera=0;
    }

if (colorled==3){
digitalWrite(ledr, HIGH);
digitalWrite(ledg, HIGH);
digitalWrite(ledb, LOW);
bandera=0;
    }

if (colorled==4){
digitalWrite(ledr, HIGH);
digitalWrite(ledg, HIGH);
digitalWrite(ledb, HIGH);
```

```
bandera=1;
    }

    if (colorled>4){

        colorled=1;
        bandera=0;
        }

    }

}
```

```
    for (int x = 0;x<3; x++){
        digitalWrite(ledr, LOW);
        delay (100);
        digitalWrite(ledr, HIGH);
        delay (100);
    }
```

```
    for (int x = 0;x<3; x++){
        digitalWrite(ledg, LOW);
        delay (100);
        digitalWrite(ledg, HIGH);
        delay (100);
    }
```

```
    for (int x = 0;x<3; x++){
        digitalWrite(ledb, LOW);
        delay (100);
        digitalWrite(ledb, HIGH);
```

```

        delay (100);

        digitalWrite(trans,HIGH);
        seleccolor();
    }

}

```

```

void TCS()
{
    digitalWrite(s1,HIGH);
    digitalWrite(s0,LOW);
    flag=0;
    attachInterrupt(0, ISR_INT0, CHANGE);
    timer2_init();
}

void ISR_INT0()
{
    counter++;
}

void timer2_init(void)
{
    TCCR2A=0x00;
    TCCR2B=0x07; //the clock frequency source 1024 points
    TCNT2= 100; //10 ms overflow again
    TIMSK2 = 0x01; //allow interrupt
}

int i=0;
ISR(TIMER2_OVF_vect)//the timer 2, 10ms interrupt overflow again. Internal overflow
interrupt executive function
{
    TCNT2=100;
    flag++;
}

```



```
if(flag==1)
{
    counter=0;
}
else if(flag==2)
{
    digitalWrite(s2,LOW);
    digitalWrite(s3,LOW);
    countR=counter/1.051;
    //Serial.print("red=");
    //Serial.println(countR,DEC);
    digitalWrite(s2,HIGH);
    digitalWrite(s3,HIGH);
}
else if(flag==3)
{
    countG=counter/1.0157;
    // Serial.print("green=");
    //Serial.println(countG,DEC);
    digitalWrite(s2,LOW);
    digitalWrite(s3,HIGH);
}
else if(flag==4)
{
    countB=counter/1.114;
    //Serial.print("blue=");
    // Serial.println(countB,DEC);
    digitalWrite(s2,LOW);
    digitalWrite(s3,LOW);
}
else
{
    flag=0;
    TIMSK2 = 0x00;
```



```
digitalWrite(ledr, LOW);
digitalWrite(ledg, HIGH);
digitalWrite(ledb, HIGH);
```

```
delay(100); // This delay is required.
}
```

```
//If the character received = 'g' , then change the RGB led to display a GREEN colour
```

```
if(recvChar=='v' && bandera==1){
Send32Zero(); // begin
DataDealWithAndSend(0, 255, 0); // first node data
Send32Zero(); // send to update data
colorled=2;
digitalWrite(ledr, HIGH);
digitalWrite(ledg, LOW);
digitalWrite(ledb, HIGH);
```

```
delay(100); // This delay is required.
}
```

```
if(recvChar=='a' && bandera==1){
Send32Zero(); // begin
DataDealWithAndSend(0, 0, 255); // first node data
Send32Zero(); // send to update data
colorled=3;
digitalWrite(ledr, HIGH);
digitalWrite(ledg, HIGH);
digitalWrite(ledb, LOW);
```

```
delay(100); // This delay is required.
}
```



```
{

digitalWrite(trans,LOW);
tomar();
digitalWrite(trans,HIGH);
analis();
//Serial.println(colorobjeto,DEC);
if (colorled==colorobjeto)
{
destello();
seleccolor();
tomarpwm();
avance3();
delay(1000);
digitalWrite(electroiman,HIGH);
arranque=0;
retroceso3();
}

if (arranque==1)
{
avance1();
}
////////////////////////////////////
if (arranque==1)
{
digitalWrite(trans,LOW);
tomar();
digitalWrite(trans,HIGH);
analis();
//Serial.println(colorobjeto,DEC);
if (colorled==colorobjeto)
{
destello();
```

```
seleccolor();
tomarpwm();
avance2();
delay(1000);
digitalWrite(electroiman,HIGH);
arranque=0;
retroceso3();
}
}
```

```
if (arranque==1)
{
avance1();
}
```

```
////////////////////////////////////
```

```
if (arranque==1)
{
digitalWrite(trans,LOW);
tomar();
digitalWrite(trans,HIGH);
analisis();
//Serial.println(colorobjeto,DEC);
if (colorled==colorobjeto)
{
destello();
seleccolor();
tomarpwm();
avance1();
delay(1000);
digitalWrite(electroiman,HIGH);
arranque=0;
retroceso3();
```

```
}  
}
```

```
if (arranque==1)
```

```
{  
  retroceso2();  
  arranque=0;  
}
```

```
////////////////////////////////////
```

```
}
```

```
}
```

```
void sensor()
```

```
{
```

```
  TCS();
```

```
  if((countR>10)||((countG>10)||((countB>10))
```

```
  {
```

```
    if((countR>countG)&&(countR>countB))
```

```
    {
```

```
      // Serial.print("red");
```

```
      // Serial.print("\n");
```

```
      delay(1000);
```

```
    }
```

```
    else if((countG>=countR)&&(countG>countB))
```

```
    {
```

```
      // Serial.print("green");
```

```
// Serial.print("\n");
    delay(1000);
}
else if((countB>countG)&&(countB>countR))
{
    // Serial.print("blue");
    // Serial.print("\n");

    delay(1000);

}
}
else
{

    delay(1000);

}
}
```

```
void horario()
{
    digitalWrite(A, LOW);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
    digitalWrite(D, LOW);
    delay(4);
    digitalWrite(A, HIGH);
    digitalWrite(B, LOW);
    digitalWrite(C, HIGH);
```



```
digitalWrite(D, LOW);  
delay(4);  
digitalWrite(A, HIGH);  
digitalWrite(B, LOW);  
digitalWrite(C, LOW);  
digitalWrite(D, HIGH);  
delay(4);  
digitalWrite(A, LOW);  
digitalWrite(B, HIGH);  
digitalWrite(C, LOW);  
digitalWrite(D, HIGH);  
delay(4);
```

```
digitalWrite(A, LOW);  
digitalWrite(B, LOW);  
digitalWrite(C, LOW);  
digitalWrite(D, LOW);
```

```
}
```

```
void antihorario()  
{  
    digitalWrite(A, LOW);  
    digitalWrite(B, HIGH);  
    digitalWrite(C, LOW);  
    digitalWrite(D, HIGH);  
    delay(4);  
    digitalWrite(A, HIGH);  
    digitalWrite(B, LOW);  
    digitalWrite(C, LOW);  
    digitalWrite(D, HIGH);  
    delay(4);
```

```
digitalWrite(A, HIGH);  
digitalWrite(B, LOW);  
digitalWrite(C, HIGH);  
digitalWrite(D, LOW);  
delay(4);  
digitalWrite(A, LOW);  
digitalWrite(B, HIGH);  
digitalWrite(C, HIGH);  
digitalWrite(D, LOW);  
delay(4);
```

```
digitalWrite(A, LOW);  
digitalWrite(B, LOW);  
digitalWrite(C, LOW);  
digitalWrite(D, LOW);
```

```
}
```

```
void avance1()  
{  
  for (int x = 0;x<42; x++){  
    digitalWrite(A, LOW);  
    digitalWrite(B, HIGH);  
    digitalWrite(C, HIGH);  
    digitalWrite(D, LOW);  
    delay(4);  
    digitalWrite(A, HIGH);  
    digitalWrite(B, LOW);  
    digitalWrite(C, HIGH);  
    digitalWrite(D, LOW);  
    delay(4);
```

```
digitalWrite(A, HIGH);
digitalWrite(B, LOW);
digitalWrite(C, LOW);
digitalWrite(D, HIGH);
delay(4);
digitalWrite(A, LOW);
digitalWrite(B, HIGH);
digitalWrite(C, LOW);
digitalWrite(D, HIGH);
delay(4);
}
```

```
digitalWrite(A, LOW);
digitalWrite(B, LOW);
digitalWrite(C, LOW);
digitalWrite(D, LOW);

}
```

```
void avance2()
{
  for (int x = 0;x<84; x++){
    digitalWrite(A, LOW);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
    digitalWrite(D, LOW);
    delay(4);
    digitalWrite(A, HIGH);
    digitalWrite(B, LOW);
    digitalWrite(C, HIGH);
    digitalWrite(D, LOW);
    delay(4);
    digitalWrite(A, HIGH);
```

```
digitalWrite(B, LOW);
digitalWrite(C, LOW);
digitalWrite(D, HIGH);
delay(4);
digitalWrite(A, LOW);
digitalWrite(B, HIGH);
digitalWrite(C, LOW);
digitalWrite(D, HIGH);
delay(4);
}
```

```
digitalWrite(A, LOW);
digitalWrite(B, LOW);
digitalWrite(C, LOW);
digitalWrite(D, LOW);
```

```
}
```

```
void avance3()
```

```
{
  for (int x = 0;x<126; x++){
    digitalWrite(A, LOW);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
    digitalWrite(D, LOW);
    delay(4);
    digitalWrite(A, HIGH);
    digitalWrite(B, LOW);
    digitalWrite(C, HIGH);
    digitalWrite(D, LOW);
    delay(4);
    digitalWrite(A, HIGH);
    digitalWrite(B, LOW);
```

```
digitalWrite(C, LOW);  
digitalWrite(D, HIGH);  
delay(4);  
digitalWrite(A, LOW);  
digitalWrite(B, HIGH);  
digitalWrite(C, LOW);  
digitalWrite(D, HIGH);  
delay(4);  
}
```

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

```
void retroceso2()  
{  
  for (int x = 0; x < 84; x++){  
    digitalWrite(A, LOW);  
    digitalWrite(B, HIGH);  
    digitalWrite(C, LOW);  
    digitalWrite(D, HIGH);  
    delay(4);  
    digitalWrite(A, HIGH);  
    digitalWrite(B, LOW);  
    digitalWrite(C, LOW);  
    digitalWrite(D, HIGH);  
    delay(4);  
    digitalWrite(A, HIGH);  
  }  
}
```

```
digitalWrite(B, LOW);  
digitalWrite(C, HIGH);  
digitalWrite(D, LOW);  
delay(4);  
digitalWrite(A, LOW);  
digitalWrite(B, HIGH);  
digitalWrite(C, HIGH);  
digitalWrite(D, LOW);  
delay(4);
```

```
digitalWrite(A, LOW);  
digitalWrite(B, LOW);  
digitalWrite(C, LOW);  
digitalWrite(D, LOW);
```

```
}  
}
```

```
void retroceso3()  
{  
  for (int x = 0;x<126; x++){  
    digitalWrite(A, LOW);  
    digitalWrite(B, HIGH);  
    digitalWrite(C, LOW);  
    digitalWrite(D, HIGH);  
    delay(4);  
    digitalWrite(A, HIGH);  
    digitalWrite(B, LOW);  
    digitalWrite(C, LOW);
```

```
digitalWrite(D, HIGH);  
delay(4);  
digitalWrite(A, HIGH);  
digitalWrite(B, LOW);  
digitalWrite(C, HIGH);  
digitalWrite(D, LOW);  
delay(4);  
digitalWrite(A, LOW);  
digitalWrite(B, HIGH);  
digitalWrite(C, HIGH);  
digitalWrite(D, LOW);  
delay(4);
```

```
digitalWrite(A, LOW);  
digitalWrite(B, LOW);  
digitalWrite(C, LOW);  
digitalWrite(D, LOW);
```

```
}
```

```
}
```

```
void seleccolor()
```

```
{
```

```
if (colorled==1){
```

```
    digitalWrite(ledr, LOW);  
    digitalWrite(ledg, HIGH);  
    digitalWrite(ledb, HIGH);  
}
```

```
if (colorled==2){
```

```
    digitalWrite(ledr, HIGH);  
    digitalWrite(ledg, LOW);
```

```

    digitalWrite(ledb, HIGH);
        }

    if (colorled==3){
    digitalWrite(ledr, HIGH);
    digitalWrite(ledg, HIGH);
    digitalWrite(ledb, LOW);

        }

    if (colorled==4){
    digitalWrite(ledr, HIGH);
    digitalWrite(ledg, HIGH);
    digitalWrite(ledb, HIGH);

        }

}

void analisis()
{
    colorobjeto=0;

    if((countR>18)&&(countR<25))
    {
        if((countG>2)&&(countG<8))
        {
            if ((countB>4)&&(countB<8))
            {
                colorobjeto=1;//rojo
            }
        }
    }
}

```



```
}
```

```
if((countR>7)&&(countR<12))  
{  
  if((countG>8)&&(countG<13))  
  {  
    if ((countB>5)&&(countB<10))  
    {  
      colorobjeto=2; //verde  
    }  
  }  
}
```

```
if((countR>4)&&(countR<10))  
{  
  if((countG>2)&&(countG<7))  
  {  
    if ((countB>6)&&(countB<10))  
    {  
      colorobjeto=3; //azul  
    }  
  }  
}
```

```
}
```

```
void destello()
```

```
{
```

```
    for (int x = 0;x<3; x++){  
        digitalWrite(ledr, LOW);  
        delay (100);  
        digitalWrite(ledr, HIGH);  
        delay (100);  
    }
```

```
    for (int x = 0;x<3; x++){  
        digitalWrite(ledg, LOW);  
        delay (100);  
        digitalWrite(ledg, HIGH);  
        delay (100);  
    }
```

```
    for (int x = 0;x<3; x++){  
        digitalWrite(ledb, LOW);  
        delay (100);  
        digitalWrite(ledb, HIGH);  
        delay (100);  
    }
```

```
}
```

```
    digitalWrite(ledr, HIGH);  
    digitalWrite(ledg, HIGH);  
    digitalWrite(ledb, HIGH);
```

```
}
```



```
}
```

//The following code snippets are used update the colour of the RGB LED-----copy and paste-----

```
void ClkProduce(void){  
    digitalWrite(Clkpin, LOW);  
    delayMicroseconds(20);  
    digitalWrite(Clkpin, HIGH);  
    delayMicroseconds(20);  
}
```

```
void Send32Zero(void){  
    unsigned char i;  
    for (i=0; i<32; i++){  
        digitalWrite(Datapin, LOW);  
        ClkProduce();  
    }  
}
```

```
uint8 TakeAntiCode(uint8 dat){  
    uint8 tmp = 0;  
    if ((dat & 0x80) == 0){  
        tmp |= 0x02;  
    }  
  
    if ((dat & 0x40) == 0){  
        tmp |= 0x01;  
    }  
  
    return tmp;  
}
```

```
// gray data
```


ANEXO 3

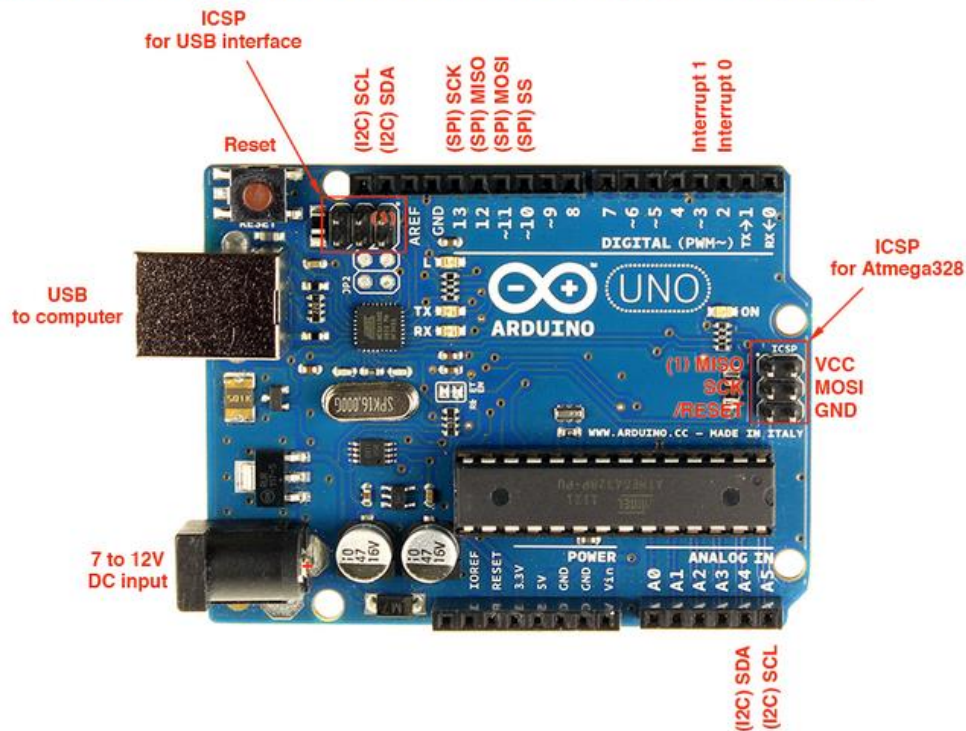
Costos del proyecto

| DESCRIPCIÓN | VAL. UNITARIO | TOTAL |
|-------------------------|---------------|---------------|
| Borneras | 1,48 | 5,92 |
| C.I. L293d | 3,95 | 3,95 |
| Cond. 470mf 35v | 0,80 | 0,80 |
| Zócalo 16 Pines | 3,40 | 3,40 |
| Cond. 0.1mf 50v | 0,70 | 1,40 |
| Cond. 10mf 16v | 0,88 | 0,88 |
| C.I. 7805cv | 1,10 | 1,10 |
| Trans. 2n3906 Ecg159 | 0,45 | 0,45 |
| Trans. 2n3904 Ecg123ap | 0,45 | 0,45 |
| Rest. 4k Ohm 1/4w | 0,03 | 0,09 |
| Rest. 1k Ohm 1/4w | 0,03 | 0,06 |
| Rest. 330 Ohm 1/4w | 0,03 | 0,03 |
| Rest. 220 Ohm 1/4w | 0,03 | 0,03 |
| Rest. 100 Ohm 1/4w | 0,03 | 0,03 |
| Rest. 470 Ohm 1/4w | 0,03 | 0,03 |
| Diodo 1n4007 | 0,76 | 0,76 |
| Trans. Tip 122 | 0,95 | 0,95 |
| Pulsador Tc11322mm | 0,98 | 2,94 |
| Led RGB 5mm 4pines | 2 | 2 |
| Electroimán | 4,20 | 4,20 |
| Disipador | 2,90 | 2,90 |
| Baquelita | 3 | 3 |
| Servo motor 5 hilos9 | 11 | 11 |
| Fuente 110v | 9,50 | 9,50 |
| Placa Arduino Uno | 37 | 37 |
| Sensor De Color | 28 | 28 |
| Bluetooth FCC ID:Z4T-BT | 35 | 35 |
| Material didáctico | 96 | 90 |
| Diseño e impresión | 85 | 98 |
| Total | USD | 343,87 |

ANEXO 4

Datasheet Arduino

ARDUINO UNO BOARD LAYOUT: From Nick Gammon (Original Info [HERE](#))



Arduino Duemilanove / YourDuino328 / YourDuinoRobo1 :

Input and Output

Each of the 14 digital pins can be used as an input or output, using [pinMode\(\)](#)[Ⓐ], [digitalWrite\(\)](#)[Ⓐ], and [digitalRead\(\)](#)[Ⓐ] functions. They operate at 5 volts. Each pin can provide (Source) or receive (Sink) a maximum of 40 mA (20 mA is recommended maximum for normal designs) and has an internal pull-up resistor (that can be activated when desired) of 20-50 kOhms.

NOTE: The [YourDuinoRobo1](#)[Ⓐ] has the same pins, but they are additionally connected to 3-pin connectors with Ground-Voltage-Signal. See Pins and Cables [HERE](#):

Specialized Pin Functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These are used for downloading of software sketches and should be reserved for that in most designs. These pins are connected to the corresponding pins of the ATmega8U2 (UNO versions) or FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#)[Ⓐ] function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#)[Ⓐ] function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the [SPI library](#)[Ⓐ].
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

There are **6 analog inputs**, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#)[Ⓐ] function.

NOTE: The Analog input pins can also be used as Digital Input or Output pins.

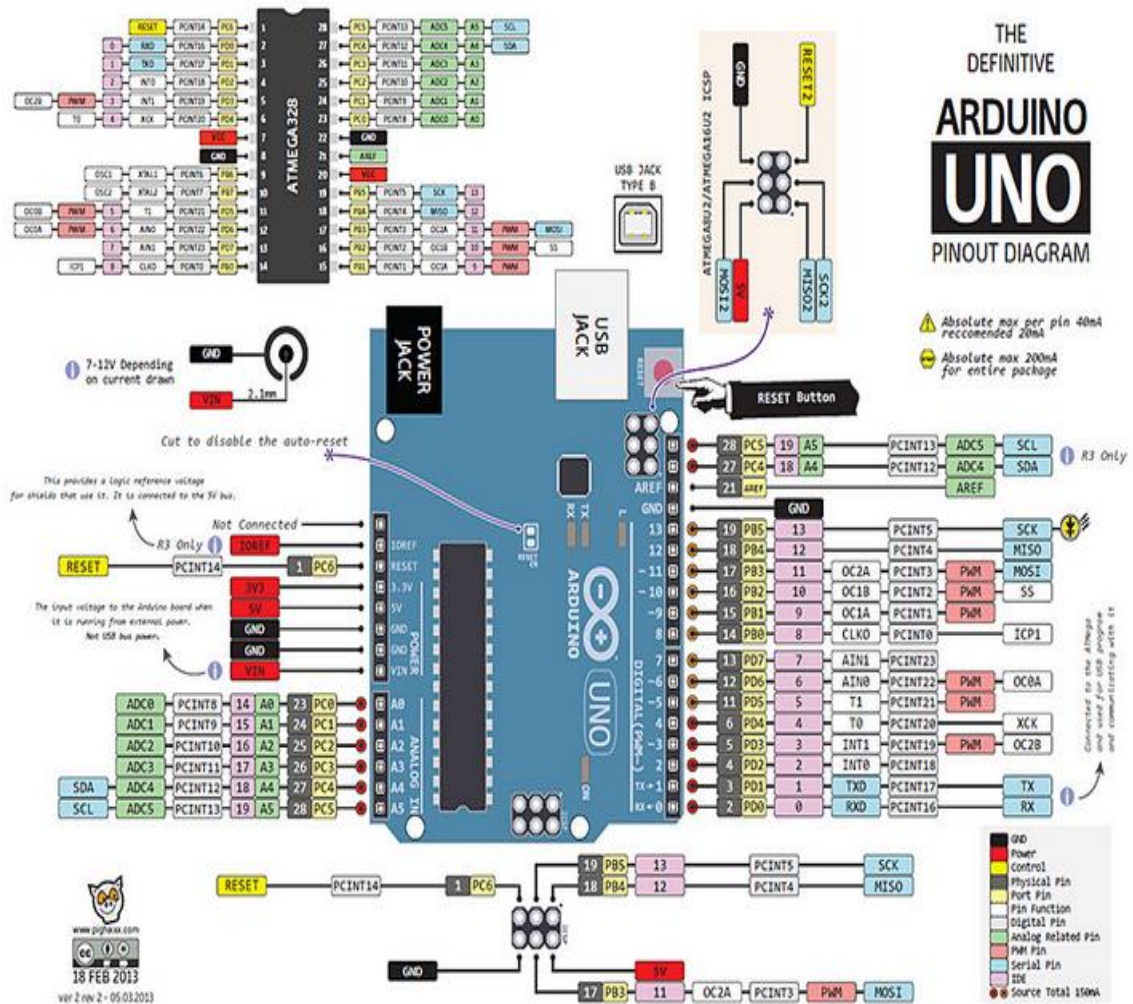
Additionally, some pins have specialized functionality:

- **I2C aka TWI: (A4 or SDA pin) and (A5 or SCL pin).** Support I2C -TWI (Two Wire Interface) communication using the [Wire library](#)².

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference](#)²).
- **Reset.** Bring this line to Ground (LOW) to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

ARDUINO UNO BOARD LAYOUT DIAGRAM: THANKS! to Alberto Piganti ([Original PDF HERE](#))³



Power Considerations: (See "Detailed Power Considerations", below)

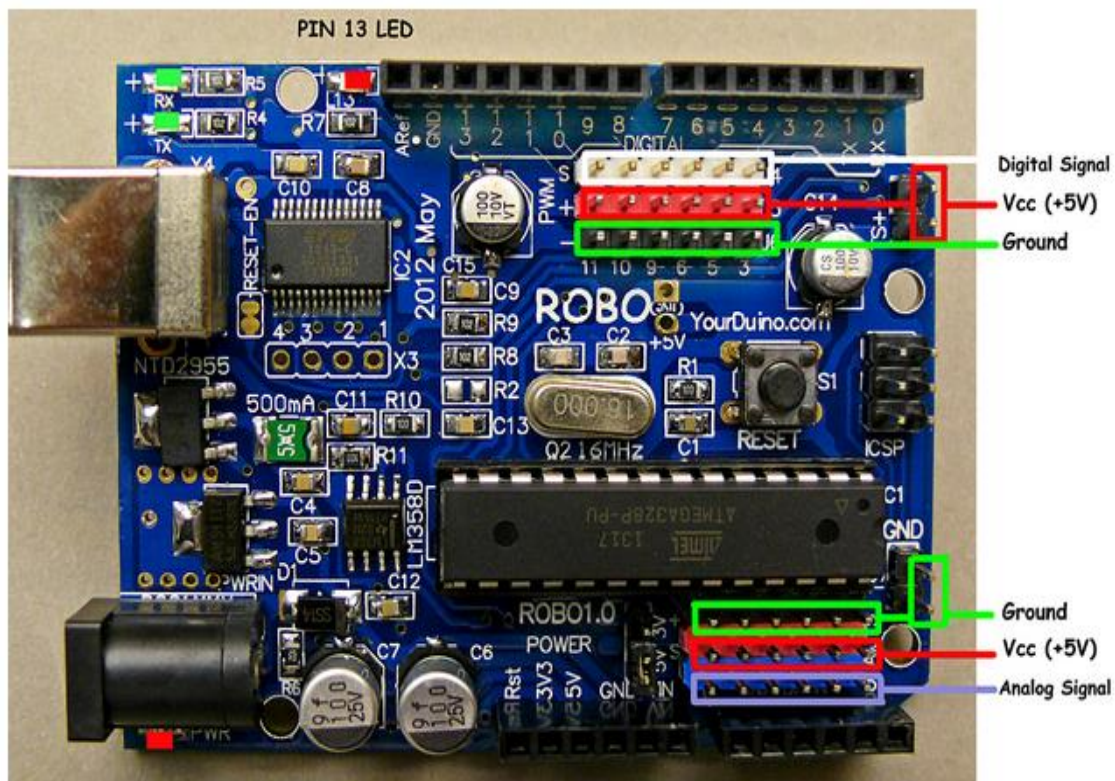
The power pins are as follows:

- **VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). This is connected to the center terminal of the Coaxial "Power" connector. You can supply a voltage through this pin, or, if supplying voltage via the power connector, access it through this pin.
- **5V**. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN/Power Connector via an on-board regulator, or be supplied by USB or another regulated 5V supply.
 - **LIMITATIONS**: If an external power supply is connected to VIN, the current available from 5V depends on the power dissipated in the onboard regulator, which is a maximum of about 1.5W. (Ref)^aAt 7V VIN this is about 700mA, at 9V VIN it's about 350 mA, at 12V it's about 200 mA. Calculate: $(VIN - 5) / 1.5$
 - **PRACTICAL LIMITS**: The ATMEGA328 and other chips use about 120mA, so what's LEFT for your LED's, and other external devices is the limitations above minus about 150 mA, so now at VIN of: 7V it's about 550mA, 9V about 200 mA, and 12V 50 mA.
- **3V3**. A 3.3 volt supply generated by the on-board regulator or FTDI chip. Maximum current draw is 50 mA.
 - NOTE: The YourDuinoRobo1 has a separate 3.3V power regulator that provides up to 350mA at 3.3V
- **GND**. Ground pins.

Memory:

The ATmega328 has 32 KB of Flash Memory for user's Software Sketches (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)^a).

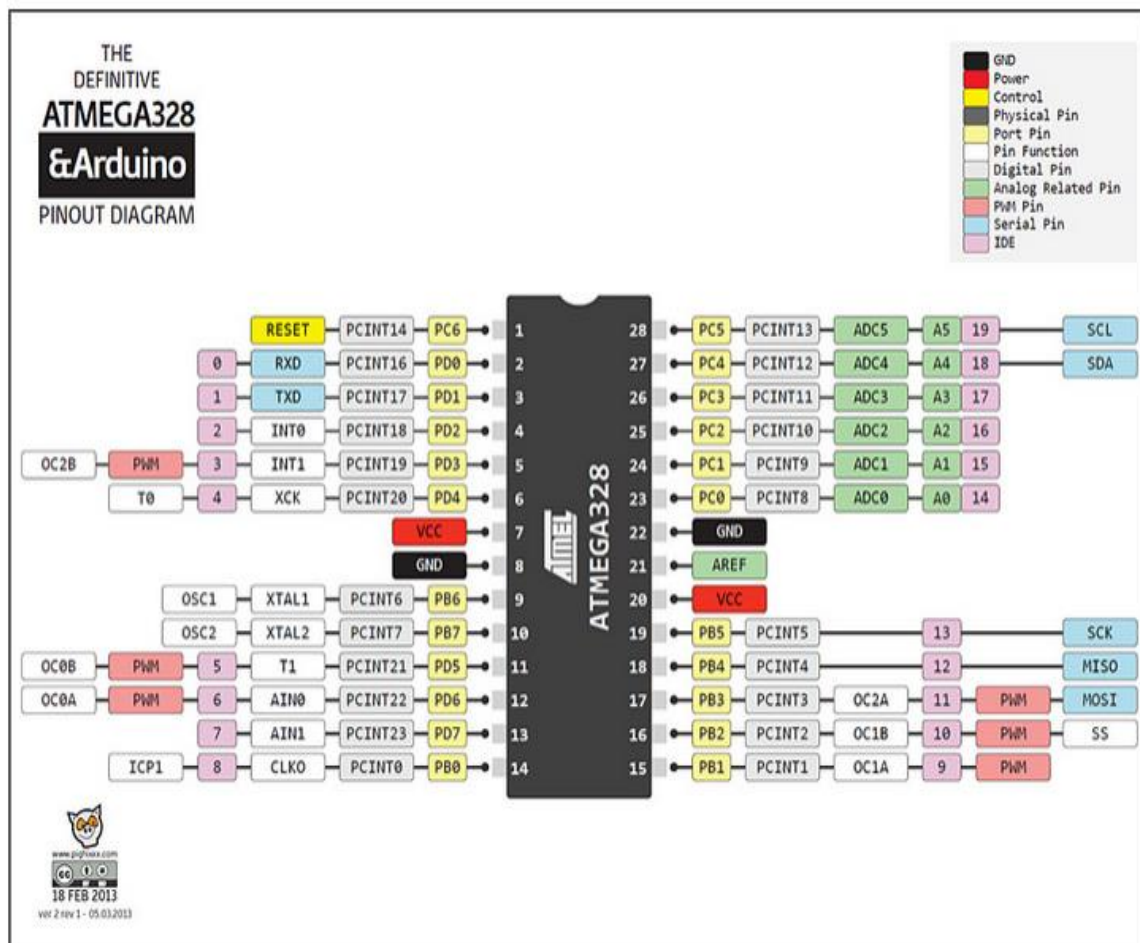
YourDuinoROBO1 BOARD LAYOUT DIAGRAM:



ATMEGA328 (UNO, Duemilanove) CHIP PINOUT DIAGRAM:

Summary:

| | |
|-----------------------------|--|
| Microcontroller | ATmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |
| Clock Speed | 16 MHz |



Powering an Arduino

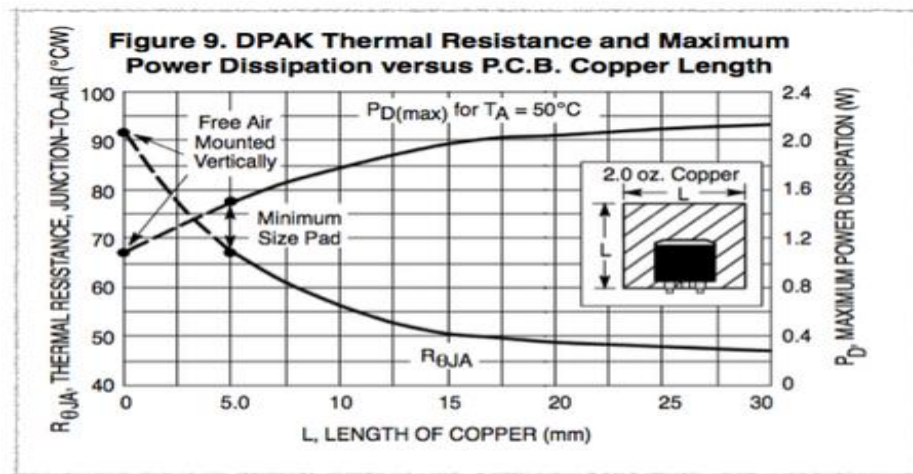
The Arduino Diecimila/Duemilanove/YourDuinoRobo1 has an on board regulator for use when powering from an external power supply. It uses the MC33269 series regulator chip. A question often asked is "can I run it off a 12V supply". A look at the schematics show that C5 and C6 will be exposed to the full input voltage so the first thing is to see that those capacitors are rated at at least 12V, in this case they are with a 35V rating. So no trouble there. Next see if the regulator can stand that much, again looking at the data sheet shows this to be able to stand 20V. So next see if it can handle the power.

The regulator drops the difference between the input voltage and the output voltage. Also there is a series diode in front of the regulator so that drops 0.7V. This gives us a total voltage drop across the regulator of $12 - 5 - 0.7 = 6.3V$.

The power burnt off by the regulator is a function of the current it is taking. Now the board takes about 120mA when it is running, add a bit extra for interface current LEDs and so on and let's call that 200mA.

So now we can calculate the power, it is simply $6.3 * 0.2 = 1.26$ Watts.

Is that too much? Well the data sheet for that device is a little more helpful than most. It includes this graph which shows what effect the copper on the PCB has on the thermal resistance.



If you look at the Arduino you will see that there is only the minimum pad, that is there is no extra copper surrounding the regulator. So in this case reading off the graph we have a maximum power of about 1.5W for a 50 °C ambient or a thermal resistance of 68 °C / W. So that means for our example we can only draw $1.5 / 6.3 = 0.238$ A or 238mA. So it looks like we will be just about OK with 200mA. The data sheet also says there is a thermal cut out that operates at 170 °C so you can use that if you need in your calculations. However, it warns that the thermal cut out is not a substitute for an adequate heat sink.

ANEXO 5

Datasheet sensor de color

Tcs230

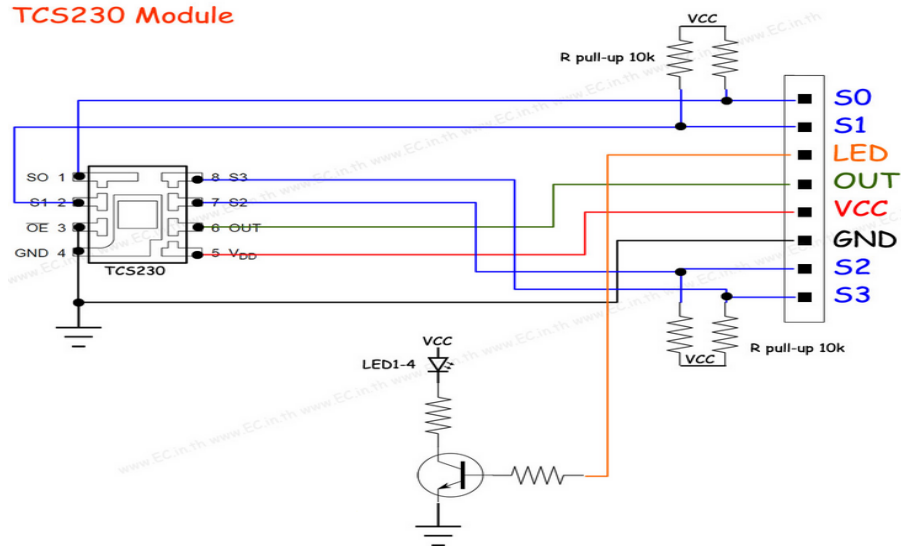


Description

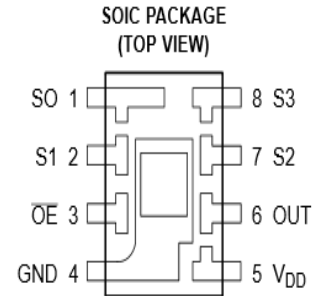
The TCS230 programmable color light-to-frequency converter combines configurable silicon photodiodes and a current-to-frequency converter on single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance). The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable (OE) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

The light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters. The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All 16 photodiodes of the same color are connected in parallel and which type of photodiode the device uses during operation is pin-selectable. Photodiodes are 120 μm x 120 μm in size and are on 144- μm centers.

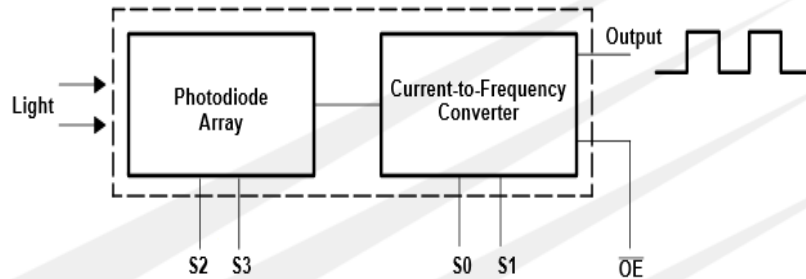
TCS230 Module



- High-Resolution Conversion of Light Intensity to Frequency
- Programmable Color and Full-Scale Output Frequency
- Communicates Directly With a Microcontroller
- Single-Supply Operation (2.7 V to 5.5 V)
- Power Down Feature
- Nonlinearity Error Typically 0.2% at 50 kHz
- Stable 200 ppm/°C Temperature Coefficient
- Low-Profile Surface-Mount Package



Functional Block Diagram



TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER
 TAOS046 - FEBRUARY 2003

Terminal Functions

| TERMINAL NAME | NO. | I/O | DESCRIPTION |
|-----------------|------|-----|--|
| GND | 4 | | Power supply ground. All voltages are referenced to GND. |
| OE | 3 | I | Enable for f_o (active low). |
| OUT | 6 | O | Output frequency (f_o). |
| S0, S1 | 1, 2 | I | Output frequency scaling selection inputs. |
| S2, S3 | 7, 8 | I | Photodiode type selection inputs. |
| V _{DD} | 5 | | Supply voltage |

Table 1. Selectable Options

| S0 | S1 | OUTPUT FREQUENCY SCALING (f_o) | S2 | S3 | PHOTODIODE TYPE |
|----|----|------------------------------------|----|----|-------------------|
| L | L | Power down | L | L | Red |
| L | H | 2% | L | H | Blue |
| H | L | 20% | H | L | Clear (no filter) |
| H | H | 100% | H | H | Green |

Available Options

| DEVICE | T _A | PACKAGE - LEADS | PACKAGE DESIGNATOR | ORDERING NUMBER |
|--------|----------------|-----------------|--------------------|-----------------|
| TCS230 | -25°C to 85°C | SOIC-8 | D | TCS230D |

Absolute Maximum Ratings over operating free-air temperature range (unless otherwise noted)†

| | |
|--|-----------------------------------|
| Supply voltage, V _{DD} (see Note 1) | 6 V |
| Input voltage range, all inputs, V _I | -0.3 V to V _{DD} + 0.3 V |
| Operating free-air temperature range, T _A | 0°C to 70°C |
| Storage temperature range | -25°C to 85°C |
| Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds | 260°C |

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to GND.

Recommended Operating Conditions

| | MIN | NOM | MAX | UNIT |
|--|----------------------------------|-----|-----|-------------------|
| Supply voltage, V _{DD} | 2.7 | 5 | 5.5 | V |
| High-level input voltage, V _{IH} | V _{DD} = 2.7 V to 5.5 V | | 2 | V _{DD} V |
| Low-level input voltage, V _{IL} | V _{DD} = 2.7 V to 5.5 V | | 0 | 0.8 V |
| Operating free-air temperature range, T _A | 0 | 70 | | °C |

ANEXO 6
Datasheet Bluetooth FCC
ID: Z4T-BT



Description

Comments **95**

Distributors

The Bluetooth Shield integrates a Serial Bluetooth module. It can be easily used with Arduino/Seeedstudio for transparent wireless serial communication. You can choose two pins from Arduino D0 to D7 as Software Serial Ports to communicate with Bluetooth Shield (D0 and D1 is Hardware Serial Port). The shield also has two Grove connectors (one is Digital, the other is Analog) for you to install Grove modules.

Features

- Seeeduino/Arduino compatible
- Up to 10m communication distance in house without obstacle
- UART interface (TTL) with programmable baud rate (SPP firmware installed)
- Default baud rate: 38400, data bits: 8, stop bit: 1, Parity: No parity
- Default PINCODE: "0000"
- A full set of configuration commands
- On board PCB antenna
- FCC ID certificated

Documents

Please visit our [wiki page](#) for more info about this product. It will be appreciated if you can help us improve the documents, add more demo code or tutorials. For technical support, please post your questions to our [forum](#).



Introduction

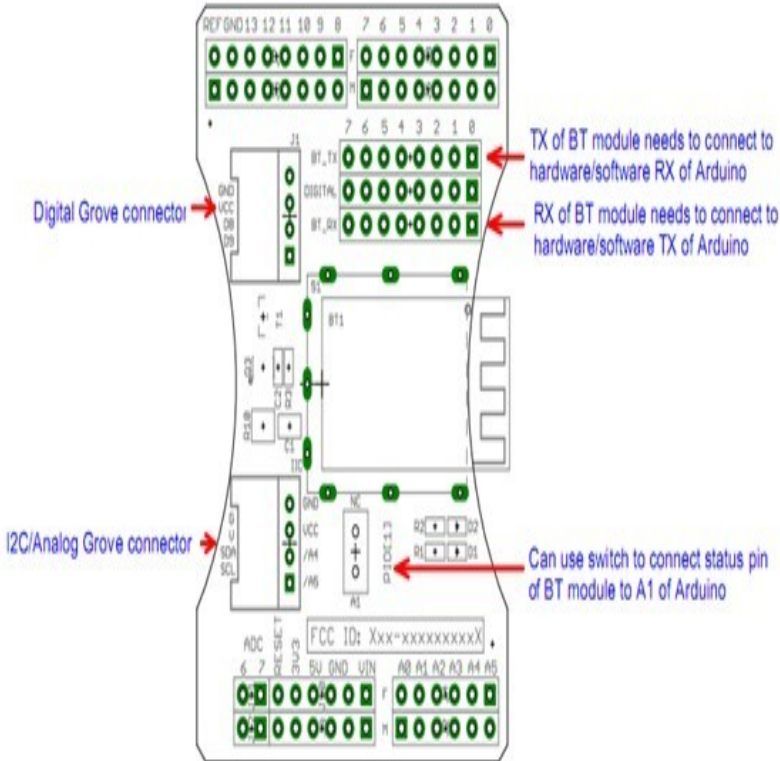
The Bluetooth Shield integrates a Serial Bluetooth module. It can be easily used with Arduino/Seeedstudio for transparent wireless serial communication. You can choose two pins from Arduino D0 to D7 as Software Serial Ports to communicate with Bluetooth Shield (D0 and D1 is Hardware Serial Port). The shield also has two Grove connectors (one is Digital, the other is Analog) for you to install Grove modules.

Model: [SLD63030P](#)

Features

- Input Voltage: 3.3V
- Baudrate: 9600, 19200, 38400, 57600, 115200, 230400, 460800
- Seeeduino/Arduino compatible
- Up to 10m communication distance in house without obstacle
- UART interface (TTL) with programmable baud rate (SPP firmware installed)
- Default Baud rate: 38400, Data bits: 8, Stop bit: 1, Parity: No parity
- Default PINCODE: "0000"
- A full set of configuration commands
- On board PCB Antenna
- FCC Part 15 Certificated

Interface funcion



| Pad Type | Description |
|----------------------|---|
| PIO1 | Status instruction port of Bluetooth module can be read by Arduino A1 port: low-disconnected, high-connected. |
| BT_RX | UART Data input of Bluetooth module. |
| BT_TX | UART Data output Bluetooth module. |
| Two Grove connectors | One is Digital (D8 and D9), the other is I2C/Analog (A4 and A5). |

Demonstration 1: Two Bluetooth Shield Connect

This demo will show you how to connect two Bluetooth shield.

You need two piece of [Seeeduino V3.0](#), a [Grove - Button](#), and a [Grove - LED](#).

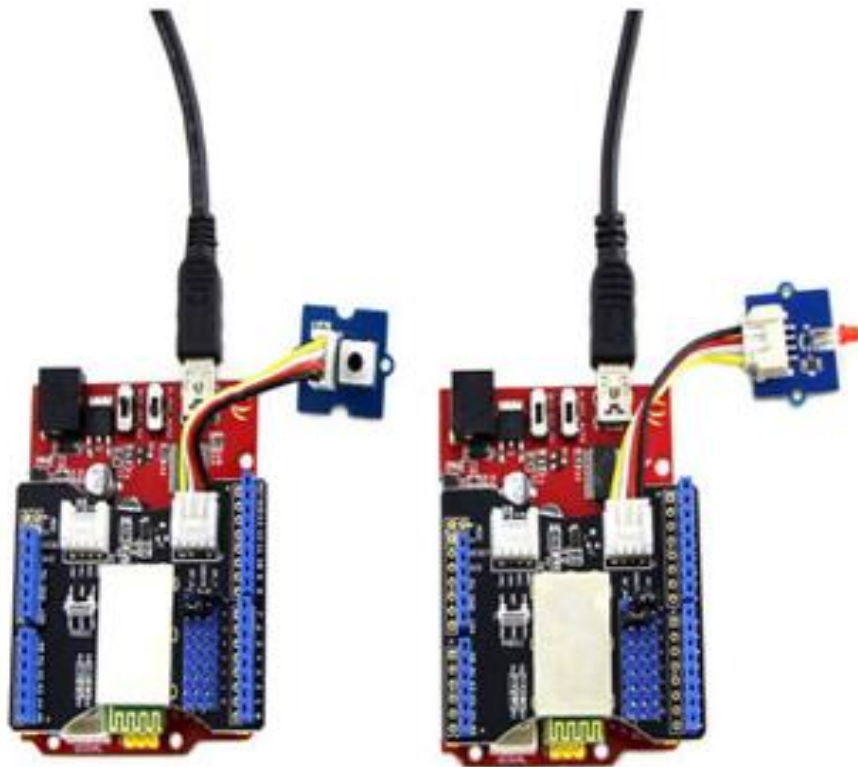
One Bluetooth Shield as Master while the other as Slave, and there's a Button connect to the Master, a Led connect to Slave.

When the button press up, the led in Slave will change.



Hardware Installation

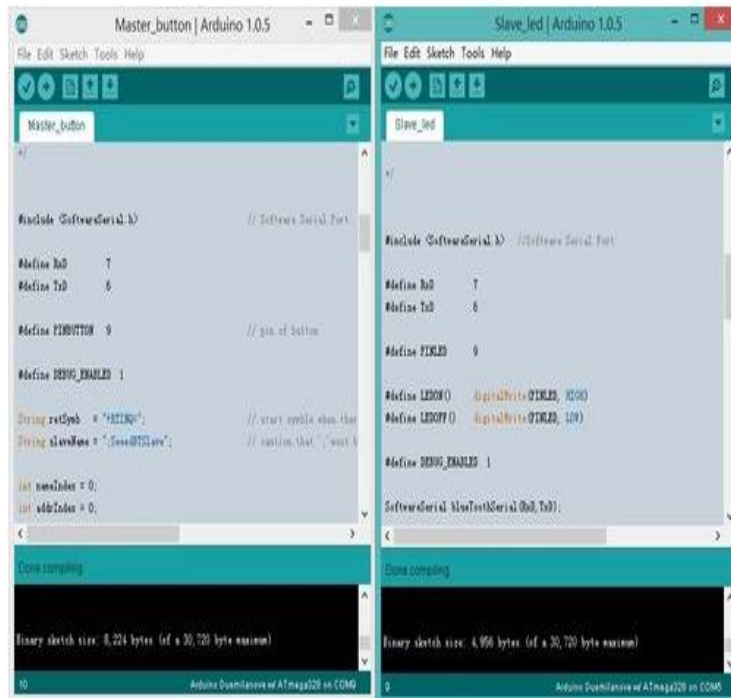
Firstly, we choose a Bluetooth Shield as Master, and plug Grove - Button to D8,D9 of this Bluetooth Shield.

For the Slave, we connect Grove - Led to its D8,D9 Grove. And D7 as TX, D6 as RX, as following:



Download Code and Upload

1. You can download the code in github, click [here](#) , then extract it to libraries folder of Arduino.
2. Open Arduino IDE, open File -> Examples -> Bluetooth_Shield_Demo_Code -> Master_Button, then you can open the code of Master
3. Open Arduino IDE, open File -> Examples -> Bluetooth_Shield_Demo_Code -> Slave_led, then you can open the code of Slave
4. Click Upload to Upload the code, if you have any problem about how to start Arduino, please click [here](#)  for some help.



Check The Result

1. After finish Uploading the code to both Master and Slave, reset the two devices meanwhile
2. You can see the red and green led blink, indicate that devices was initializing and connecting.
3. After about several seconds, only green led blink, indicate that Master and Slave had connected.
4. Now you can press up the button, then the led will change its state.

Note: If the phenomenon is not observed above, try unplugging the power and re-plug in again.

Demonstration 2: Connect to Smart Phone

This demo will show you how to connect Bluetooth Shield to a Smart Phone.

We need a Seeeduino V3.0, a Grove - Temperature Sensor, and what's more, a Smart Phone that with Bluetooth function.

Via a Bluetooth SPP App, we send a 't' to Bluetooth Shield, then it'll return the temperature.

Hardware Installation


Plug Grove - Temperature Sensor to A4,A5 Grove of Bluetooth Shield.

TX connect to D7 while RX connect to D6. as folowing:



Download Code and Upload

1. You can download the code in github, click [here](#), then extract it to libraries folder of Arduino.
2. Open Arduino IDE, File -> Examples -> Bluetooth_Shield_Demo_Code -> Slave_Temperature, then you can open the code
3. Click Upload to Upload the code, if you have any problem about how to start Arduino, please click [here](#) for some help.



```
Slave_temperature | Arduino 1.0.5
File Edit Sketch Tools Help

/* Upload this sketch into Sceduino and press reset*/

#include <SoftwareSerial.h> //Software Serial Port
#define RxD 7
#define TxD 6

#define DEBUG_ENABLED 1

#define PIN_TEMP AS

Done Saving
Binary sketch size: 7,024 bytes (of a 30,720 byte maximum)
33 Arduino Duemilanove w/ ATmega328 on COM9
```

Get Temperature

After installing an SPP app, try connecting it to SeeedBTSlave, pin code is: "0000"



When connect is ok, send 't' to SeeedBTSlave, and you can get the temperature now:

