



ESPE

**UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA**

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

**DEPARTAMENTO DE ELECTRÓNICA Y
COMPUTACIÓN.**

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN
Y AVIÓNICA.**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE TECNÓLOGO EN ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN Y AVIÓNICA.**

**TEMA: “IMPLEMENTACIÓN DE MATERIAL AUDIOVISUAL
UTILIZANDO ARDUINO PARA LA MEJORA DEL PROCESO
ENSEÑANZA-APRENDIZAJE DE NIÑOS DE EDADES
COMPRENDIDAS ENTRE 3 A 6 AÑOS”.**

AUTOR: CARLOS LUIS CAILLAGUA VELASCO

DIRECTOR: ING. CRISTIAN CHUCHICO

LATACUNGA

2017



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“IMPLEMENTACIÓN DE MATERIAL AUDIOVISUAL UTILIZANDO ARDUINO PARA LA MEJORA DEL PROCESO ENSEÑANZA-APRENDIZAJE DE NIÑOS DE EDADES COMPRENDIDAS ENTRE 3 A 6 AÑOS”**. Realizado por el señor **CAILLAGUA VELASCO CARLOS LUIS**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **CAILLAGUA VELASCO CARLOS LUIS**, para que lo sustente públicamente.

Latacunga, 03 de septiembre del 2017

Atentamente,

ING. CRISTIAN CHUCHICO
DIRECTOR



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA

AUTORÍA DE RESPONSABILIDAD

Yo, **CAILLAGUA VELASCO CARLOS LUIS**, con cédula de identidad N° 0503505984 declaro que este trabajo de titulación **“IMPLEMENTACIÓN DE MATERIAL AUDIOVISUAL UTILIZANDO ARDUINO PARA LA MEJORA DEL PROCESO ENSEÑANZA-APRENDIZAJE DE NIÑOS DE EDADES COMPRENDIDAS ENTRE 3 A 6 AÑOS”**, ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance científico de trabajo de grado en mención.

Latacunga, 03 de septiembre del 2017

Atentamente,

CAILLAGUA VELASCO CARLOS LUIS

C.I: 0503505984



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN
Y AVIÓNICA

AUTORIZACIÓN

Yo, **CARLOS LUIS CAILLAGUA VELASCO** autorizo a la Universidad de las Fuerzas Armadas – ESPE la publicación, en la biblioteca virtual de la Institución del trabajo **“IMPLEMENTACIÓN DE MATERIAL AUDIOVISUAL UTILIZANDO ARDUINO PARA LA MEJORA DEL PROCESO ENSEÑANZA-APRENDIZAJE DE NIÑOS DE EDADES COMPRENDIDAS ENTRE 3 A 6 AÑOS”**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad.

Latacunga, 03 de septiembre del 2017

Atentamente,

CAILLAGUA VELASCO CARLOS LUIS

C.I: 0503505984

DEDICATORIA

Este trabajo de grado va dedicado primeramente a Dios y mis padres Luis y Marisol. A dios por darme la fuerza para seguir adelante con mis proyectos, por dame salud y vida. A mis padres que con su amor y consejos me apoyaron en uno más de mis proyectos de vida y que con su paciencia han sabido apoyarme en todo.

También lo dedico a mis hermanos Katty, Andrés y Sebastián porque fueron ellos que también estuvieron conmigo en las buenas y en las malas que con su ejemplo de superación me dieron el aliento para poder superar cualquier obstáculo y así lograr mis metas.

CAILLAGUA VELASCO CARLOS LUIS

AGRADECIMIENTO

Por estar siempre a mi lado a Dios, a mi familia, amigos y en especial a mi tutor de proyecto Ing. Cristian Chuchico y mi director de Carrera Ing. Pablo Pilatasig, que con sus enseñanzas hicieron que me superara tanto en lo personal como en lo profesional, ayudándome así a lograr un objetivo más.

A mi familia que con su amor y su paciencia hicieron que pueda lograr una meta más en mi vida por darme una palabra de aliento en los momentos difíciles y por apoyarme siempre. A mis amigos que con sus locuras y con su forma de ser, han hecho que mi vida sea más emocionante, y en especial a mi mejor amigo Oscar Pullopasig quien con su amistad y paciencia me ha apoyado para superarme como profesional.

Gracias a cada una de las personas que me ayudaron a culminar mi carrera a todos ellos gracias.

CAILLAGUA VELASCO CARLOS LUIS

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	ii
AUTORÍA DE RESPONSABILIDAD	iii
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDOS	vii
ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS	xi
RESUMEN	xiii
ABSTRACT.....	xiv
CAPÍTULO I.....	1
PLANTEAMIENTO DEL PROBLEMA.....	1
ANTECEDENTES.	1
PLANTEAMIENTO DEL PROBLEMA.....	2
JUSTIFICACION.	2
OBJETIVOS.	2
1.1.1 Objetivo general.	2
1.1.2 Objetivos específicos.....	2
ALCANCE.....	3
CAPÍTULO II	4
MARCO TEÓRICO	4
2.1 Arduino	4
2.1.1 Definición de Arduino	4
2.1.2 Características de Arduino UNO Y Arduino MEGA 2560	5
2.1.3 ¿Para qué sirve Arduino?	6
2.2 Hardware Arduino	7
2.2.1 El HW de Arduino	7
2.2.2 Placas compatibles y no compatibles a Arduino	8
2.3 Software Arduino	9

2.3.1 EL SW de Arduino.....	9
2.3.2 IDE Arduino	10
2.3.3 Carga del Programa en Arduino.....	12
2.3.4 Configuración inicial de IDE	13
2.3.5 Gestor de tarjetas	14
2.4 Programación de Arduino.	15
2.4.1 Variables.	16
2.4.2 Variable global.....	16
2.4.3 Variable local.....	17
2.5 Datos y Variables.....	17
2.5.1 Tipos de variables	17
2.5.2 Operadores	17
2.5.3 Estructuras de control.....	18
2.5.4 Funciones definidas por el usuario.	18
2.6 Ventajas y desventajas de Arduino	19
2.6.1 Ventajas	19
2.6.2 Desventajas.....	20
2.7 Arduino MP3 shield board con TF card	20
2.7.1 Características de Arduino MP3 shield board con TF card	21
2.8 Circuito integrado 74HC573.....	21
2.8.1 Características circuito integrado 74HC573	22
2.9 Circuito integrado ULN2803.....	23
2.9.1 Características circuito integrado ULN2803	23
2.10 Diodo emisor de luz (LED)	25
2.10.1 Características de diodo led de alta luminosidad	26
2.11 Modulo teclado matricial.	26
2.11.1 Características	27
CAPÍTULO III.....	28
DESARROLLO DEL TEMA.....	28
3.1. PRELIMINARES	28
3.2. REQUERIMIENTOS MÍNIMOS DE HARDWARE.....	28
3.2.1. Leds de alto brillo	28
3.2.2. Cubo led y placa de control.....	29

3.2.3.	Funcionamiento del circuito integrado 74HC573.....	30
3.2.4.	Funcionamiento del circuito integrado ULN2803.....	31
3.2.5.	Arduino Mega.....	32
3.2.6.	Arduino UNO.....	34
3.2.7.	Arduino MP3 shield board con TF card.....	34
3.3.	REQUERIMIENTOS MÍNIMOS DEL SOFTWARE.....	35
3.3.1.	Líneas de programación para letras y números en el cubo led.....	36
3.3.2.	Líneas de programación para el audio MP3.....	39
3.3.3.	Carga y compilación del programa en la placa Arduino.....	41
3.4.	PRUEBAS DE FUNCIONAMIENTO.....	43
3.4.1.	Funcionamiento del proyecto.....	44
	CAPÍTULO IV.....	45
	CONCLUSIONES.....	45
	RECOMENDACIONES.....	46
	GLOSARIO DE TÉRMINOS.....	47
	REFERENCIAS BIBLIOGRÁFICAS.....	49
	ANEXOS.....	¡Error! Marcador no definido.
	¡Error! Marcador no definido.	

ÍNDICE DE TABLAS

Tabla 1 Conexión de la PCB con Arduino Mega.....	32
Tabla 2 <u>Conexión del teclado matricial con Arduino shield MP3.....</u>	<u>34</u>
Tabla 3 <u>Conexión para la comunicación con Arduino Uno y Mega.....</u>	<u>34</u>

ÍNDICE DE FIGURAS

Figura 1 Logo de Arduino	4
Figura 2 Universo Arduino	5
Figura 3 Placa y características de Arduino UNO	5
Figura 4 Placa y características de Arduino MEGA	6
Figura 5 Hardware Arduino.....	8
Figura 6 Placas compatibles y no compatibles para Arduino	9
Figura 7 Software Arduino	10
Figura 8 Entorno IDE de Arduino.....	11
Figura 9 Carga de programa en Arduino	12
Figura 10 Configuración inicial IDE de Arduino	13
Figura 11 Gestor de Tarjetas.....	14
Figura 12 Programación en Arduino.....	15
Figura 13 Tipos de Variables.....	16
Figura 14 Funciones definidas por el usuario.....	18
Figura 15 Arduino MP3 shield board con tarjeta TF sd.	21
Figura 16 Circuito Integrado 74HC573.....	22
Figura 17 Circuito Integrado ULN2803.....	23
Figura 18 Características de circuito integrado ULN2803.	24
Figura 19 Diodo led de alta Luminosidad.	25
Figura 20 Teclado Matricial.	26
Figura 21 Características del teclado matricial.....	27
Figura 22 Led de alto brillo.	29
Figura 23 Cubo led	29
Figura 24 PCB para el control del cubo led	30
Figura 25 Cubo led y PCB.....	30
Figura 26 Funcionamiento del circuito integrado 74HC573.....	31
Figura 27 Funcionamiento del circuito integrado ULN2803.....	32
Figura 28 Pines de conexión con la PCB y Arduino	33
Figura 29 Arduino Shield MP3.....	35
Figura 30 Grafica Tridimensional X, Y, Z.	35
Figura 31 Entorno de programación.....	36

Figura 32 Lera A mostrada en el cubo led.....	41
Figura 33 Entorno para abrir un programa	41
Figura 34 Selección del tipo de placa a utilizar.....	42
Figura 35 Selección del puerto COM.....	42
Figura 36 Boton subir para cargar el programa en la placa arduino.....	43
Figura 37 Primera prueba de funcionamiento.....	43
Figura 38 Segunda prueba de funcionamiento la tra T.....	44
Figura 39 Funcionamiento de la letra A y Shield MP3.....	44

RESUMEN

El proyecto técnico tiene como objetivo la implementación de material audiovisual para la mejora de enseñanza aprendizaje de niños de entre 3 y 6 años utilizando Arduino, para lo cual se utiliza leds de alto brillo para una mejor visibilidad y menor consumo de energía, además de la PCB de Arduino que se utiliza para el control del cubo led, ya que para su funcionamiento necesita una placa que contenga varias entradas y salidas digitales, así como de circuitos integrados para poder controlarlo mediante filas y columnas para la multiplexación del mismo. Para la creación de las figuras se utilizan puntos referenciales, o coordenadas X, Y, Z que se muestran en la pantalla 3D, Adicionalmente con implementación de audio en formato MP3, para escuchar el sonido cuando se pulse un botón del teclado matricial y se muestre la figura deseada.

También se utiliza el software IDE de Arduino que contiene un editor de texto para escribir el sketch, una consola de error, una área con los menús y los botones que realizan las funciones más comunes como son abrir sketch, guardar sketch, compilar y cargar el programa, con el cual se realiza las líneas de programación para el control del cubo led 3D y la reproducción de audio MP3.

PALABRAS CLAVES:

- **ARDUINO.**
- **SHIELD MP3**
- **LED.**
- **CIRCUITO INTEGRADO.**

ABSTRACT

The technical project has the main good to implement audiovisual material to improve the teaching and learning in process, children from 3 to 6 years old using Arduino. In which it is used high brightness leds a better visibility and lower energy consumption. In addition of the Arduino PCB that is used for the control of the led cube, It is used a plate that contains several outputs and digital inputs, as well as integrated circuits to get control by rows and columns for the multiplexation of it self. For the figures creation, it is used referencial points, or X, Y, Z coordinates which show on the 3D screen. And audio in MP3 implementation format, in order to hear the sound when a button is pressed on the matrix keyboard, and it shows the desire picture

Arduino IDE software is also used which contains a text editor to write the sketch, an error console, an area with menus and buttons that perform the most common functions such as opening the sketch, saving the sketch, compiling and loading the program, with which one the programming lines for control of the 3D led cube and the reproduction of audio MP3 are realized.

KEYWORDS:

- **ARDUINO.**
- **SHIELD MP3.**
- **LED.**
- **CIRCUIT INTEGRATED**

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA.

IMPLEMENTACIÓN DE MATERIAL AUDIOVISUAL UTILIZANDO ARDUINO PARA LA MEJORA DEL PROCESO ENSEÑANZA-APRENDIZAJE DE NIÑOS DE EDADES COMPRENDIDAS ENTRE 3 A 6 AÑOS.

ANTECEDENTES.

Los niños en edades de 3 a 6 años aproximadamente suelen ser curiosos y se interesan mucho en el aprendizaje, por tal motivo es importante buscar maneras prácticas y divertidas con diferentes actividades enfocadas en el reconocimiento de letras y números mediante formas y colores.

Los métodos anteriores para la enseñanza-aprendizaje se utilizaban carteles ubicados en el aula con letras y números en un tamaño grande, vistoso y con diferentes colores, estas son elaboradas en cartulina, fomix o cartón. Otro método es el rompecabezas con figuras de letras y números hechos en madera, espuma flex u otro material, este tipo de actividades les resultan demasiado interesante, mediante los cuales pueden crear “juegos de palabras”, de este modo irán afianzando poco a poco su conocimiento y aprenderán cada una de las letras y los números. (ParasuBebe.com, 2016)

En la actualidad la tecnología tiene la potencialidad de contribuir a transformar los sistemas escolares en un mecanismo mucho más flexible y eficaz, para lo cual se hizo una investigación de campo donde se pudo observar que un trabajo técnico realizado por el Sr. MARC GALLARDO RUIZ con el tema “Sistema electrónico para aplicación de juguetes infantiles basado en Arduino con capacidad de interacción multimedia”

Permitiendo a los niños de edades entre 3 a 6 años de educación inicial adaptarse con la implementación de material audiovisual, y de esta manera fortalecer el proceso de enseñanza-aprendizaje gracias a la interacción de Arduino.

PLANTEAMIENTO DEL PROBLEMA.

La educación inicial, como agente educativo que es, debe utilizar las tecnologías de Información y Comunicación para formar y preparar a sus estudiantes desde edades tempranas, para hallar las razones más sólidas que plantean la expectativa de tecnología en aulas y centros escolares.

En años anteriores se utilizaba métodos acordes a los recursos económicos y tecnológicos de la época, es por eso, que el desconocimiento de métodos actuales no permite a los estudiantes incrementar su aprendizaje en el ámbito tecnológico, esto puede ser una desventaja en la mejora del proceso enseñanza-aprendizaje debido a que la tecnología va avanzando cada día. (La Importancia del Uso de las TICS en la Educación Inicial, 2013)

JUSTIFICACIÓN.

La implementación del presente proyecto técnico tiene como objetivo establecer un modelo que integre metodologías y prácticas tecnológicas en los procesos de enseñanza-aprendizaje debido a la necesidad de integrar la tecnología en los estudiantes de educación inicial la cual permite un mayor rendimiento y comodidad de uso del proyecto a realizarse.

El proyecto permite a los estudiantes de edades comprendidas entre 3 y 6 años de educación inicial, aplicar tecnología mediante la implementación de material audiovisual en el proceso de enseñanza-aprendizaje. Siendo factible porque el Software tecnológico de aplicación es económico, sin restricciones, fácil de conseguir y no tiene ningún impedimento legal.

OBJETIVOS.

1.1.1 Objetivo general.

Implementar material audiovisual utilizando Arduino para la mejora del proceso enseñanza-aprendizaje de niños de edades comprendidas entre 3 y 6 años de edad.

1.1.2 Objetivos específicos.

- Indagar información referente a los dispositivos a utilizar en la implementación del material audiovisual mediante investigación técnica y

teórica para el funcionamiento, características del equipo y el impacto en proceso del aprendizaje infantil.

- Crear una programación mediante el software Arduino acorde a las especificaciones de los dispositivos y microcontroladores a ser implementados.
- Realizar pruebas y verificación de funcionamiento del material audiovisual.

ALCANCE.

El proyecto técnico correspondiente a la implementación de material audiovisual utilizando Arduino para la mejora del proceso enseñanza-aprendizaje de niños de edades comprendidas entre 3 y 6 años de edad, constituye una herramienta versátil que brinda a los docentes la posibilidad de captar la atención de los alumnos de mejor manera para un mejor desempeño de los mismos. Mediante esta pantalla led “cubo” el cual permite observar figuras “letras y números” en 3D, además de escuchar audio al momento en el que aparezca la figura deseada por el usuario.

CAPÍTULO II

MARCO TEÓRICO

2.1 Arduino

2.1.1 Definición de Arduino

Según (arduino.cl, 2017)

Arduino es una plataforma de prototipos de electrónica de código abierto (open-source) basada en hardware, software flexible y fácil de usar.

Está basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.”

Mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros dispositivos. (arduino.cl, 2017)



Figura 1 logo Arduino

Fuente: (arduino.cl, 2017)

El microcontrolador de la placa se programa usando el Arduino Programming Language “basado en Wiring” y el Arduino Development Environment “basado en Processing”. Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador, por ejemplo con Flash, Processing, MaxMSP, etc. (arduino.cl, 2017)

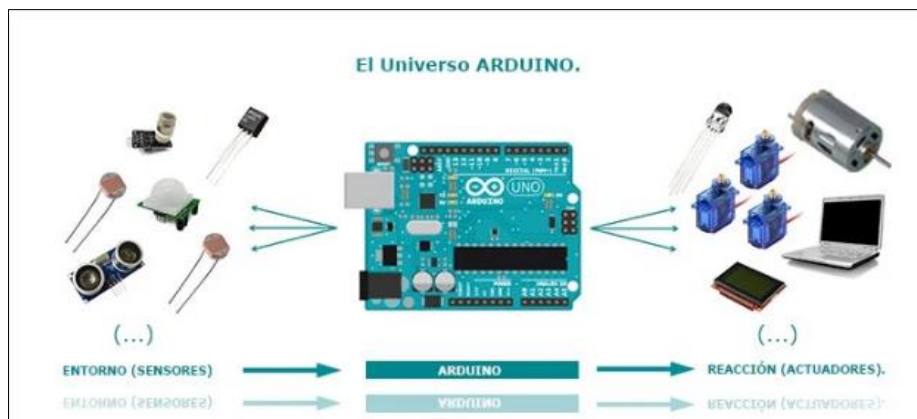


Figura 2 Universo Arduino
Fuente: (arduino.cl, 2017)

2.1.2 Características de Arduino UNO Y Arduino MEGA 2560

Según (arduino.cl, 2017) las características de Arduino son:

Arduino Uno utiliza el ATmega16U2 para el manejo de USB en lugar del 8U2 (o del FTDI encontrado en generaciones previas). Esto permite ratios de transferencia más rápidos y más memoria. No se necesitan drivers para Linux o Mac (el archivo inf para Windows es necesario y está incluido en el IDE de Arduino). (arduino.cl, 2017)

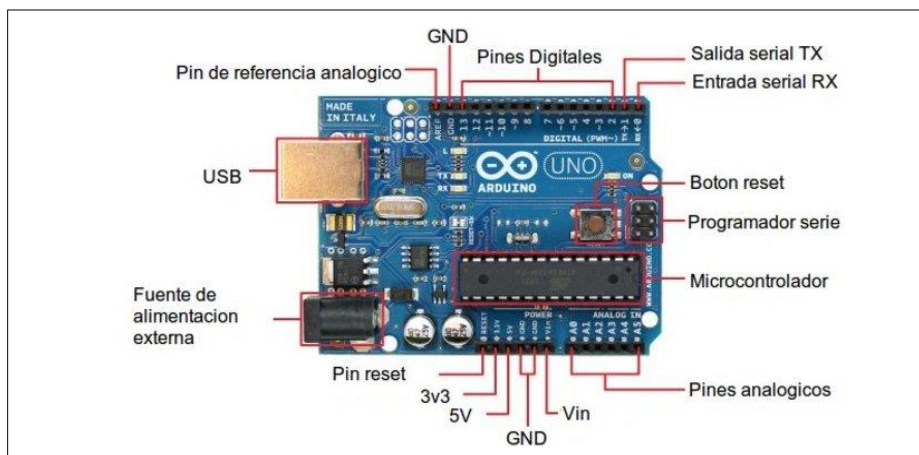


Figura 3 Placa y características de Arduino UNO
Fuente: (arduino.cl, 2017)

Arduino Mega está basado en el microcontrolador ATmega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas análogas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16 MHz, conexión USB, jack de alimentación,

conector ICSP y botón de reset (arduino.cl, 2017). Incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conectado a la PC por medio de un cable USB o con una fuente de alimentación externa. El Arduino Mega es compatible con la mayoría de los Shields diseñados para Arduino Duemilanove, diecimila o UNO. (arduino.cl, 2017)

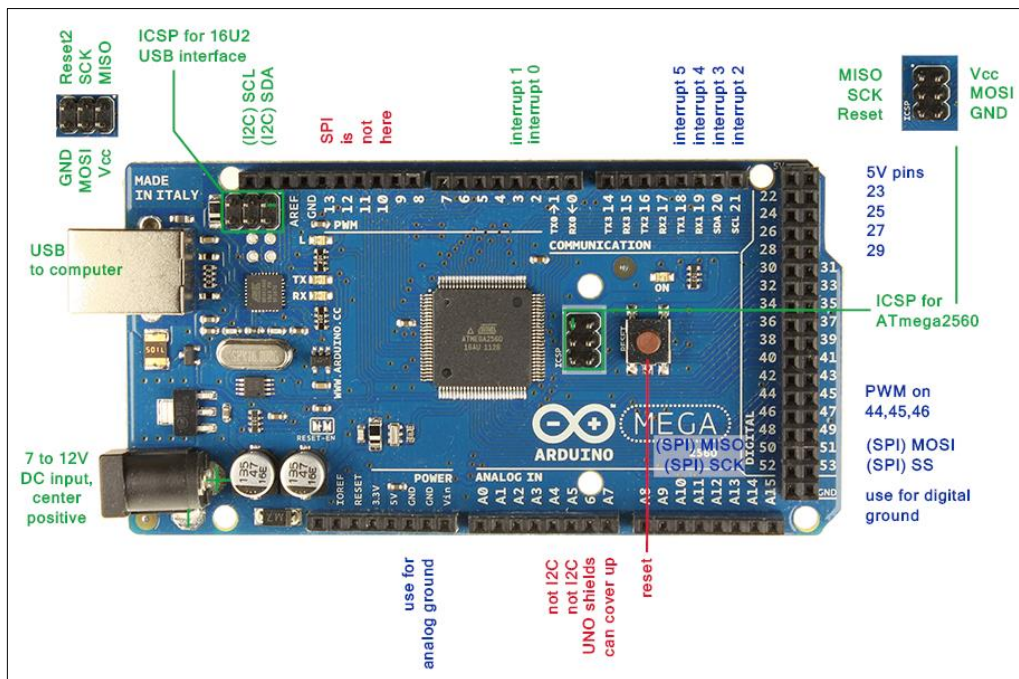


Figura 4 Placa y características de Arduino mega

Fuente: (arduino.cl, 2017)

2.1.3 ¿Para qué sirve Arduino?

Según (arduino.cl, 2017)

Arduino se puede utilizar para desarrollar elementos autónomos, conectándose a dispositivos e interactuar tanto con el hardware como con el software. Sirve tanto para controlar un elemento, el cual por ejemplo, un motor que suba o baje una persiana, basada en la luz existente es una habitación, gracias a un sensor de luz conectado al Arduino, o bien para leer la información de una fuente, como puede ser un teclado, y convertir la información en una acción como puede ser encender una luz y pasar por un display lo tecleado. (arduino.cl, 2017)

Hay otro factor importante en el éxito de Arduino, es la comunidad que apoya todo este desarrollo, comparte conocimiento, elabora librerías para facilitar el uso de Arduino y publica sus proyectos para que puedan ser replicados, mejorados o ser base para otro proyecto relacionado. (arduino.cl, 2017)

En resumen:

Arduino = HARDWARE + SOFTWARE + Comunidad

2.2 Hardware Arduino

2.2.1 El HW de Arduino

Según (arduino, 2016)

Es básicamente una placa con un microcontrolador. Un microcontrolador (abreviado μ C, UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. (arduino, 2016) Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida. (arduino, 2016) Características de un Microcontrolador:

- Velocidad del reloj u oscilador
- Tamaño de palabra
- Memoria: SRAM, Flash, EEPROM, ROM, etc...
- I/O Digitales
- Entradas Analógicas
- Salidas analógicas (PWM)
- DAC (Digital to Analog Converter)
- ADC (Analog to Digital Converter)
- Buses
- UART
- Otras comunicaciones.

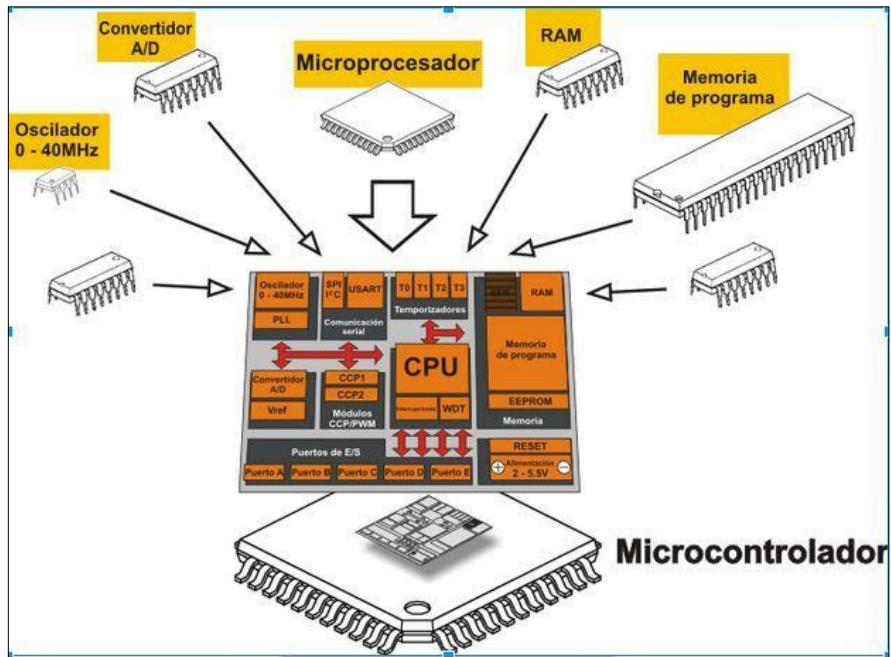


Figura 5 Hardware Arduino

Fuente: (arduino, 2016)

2.2.2 Placas compatibles y no compatibles a Arduino

Según (arduino, 2016)

El HW Arduino no solo se queda en las placas oficiales, sino que en los últimos años han aparecido muchas placas de prototipo basadas en los mismos o diferentes microcontroladores que bien por acuerdos con Arduino, por los propios fabricantes de los microcontroladores o por la propia comunidad de usuarios, estas placas tienen soporte del IDE de Arduino y es posible programarlas como el resto de Arduino oficiales con el mismo lenguaje de programación. (arduino, 2016)

Placas no oficiales Arduino con soporte para el IDE de Arduino son:

- ESP8266
- ESP32:

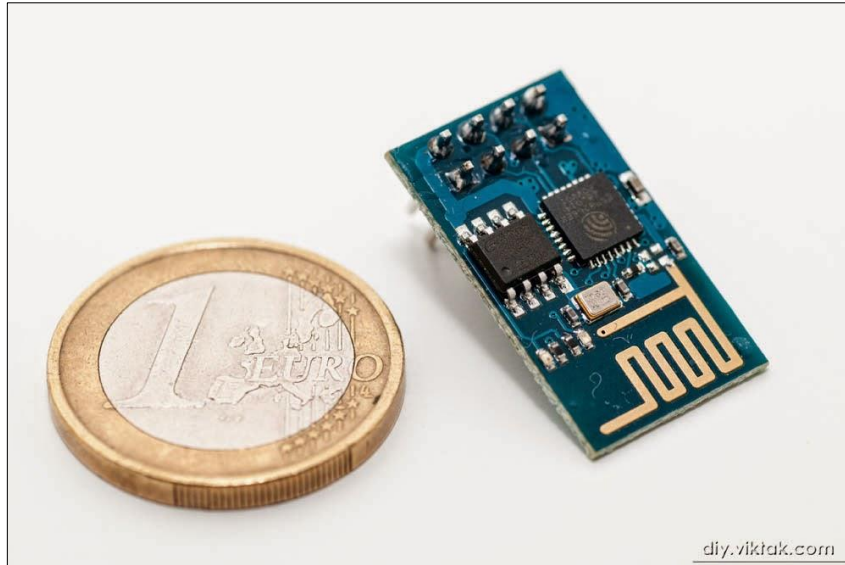


Figura 6 Placas compatibles y no compatibles para ARDUINO

Fuente: (arduino, 2016)

2.3 Software Arduino

2.3.1 EL SW de Arduino

Según (arduino, 2016)

El software de Arduino es un IDE, entorno de desarrollo integrado (siglas en inglés de Integrated Development Environment). Es un programa informático compuesto por un conjunto de herramientas de programación.

El IDE de Arduino es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware. (arduino, 2016)

Es destacable desde la aparición de la versión 1.6.2 la incorporación de la gestión de librerías y la gestión de placas muy mejoradas respecto a la versión anterior y los avisos de actualización de versiones y librerías.

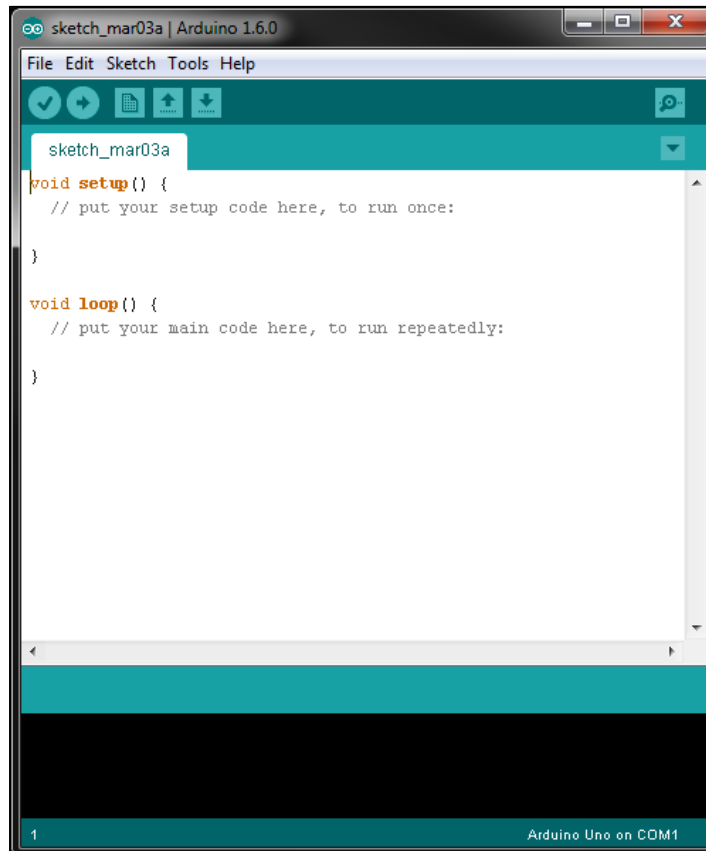


Figura 7 Software Arduino

Fuente: (arduino, 2016)

2.3.2 IDE Arduino

Según (arduino, 2016)

El IDE de Arduino va a ser la herramienta de trabajo con Arduino y habrá que conocer su funcionamiento. Los programas de Arduino están compuestos por un solo fichero con extensión “ino”, aunque es posible organizarlo en varios ficheros. El fichero principal siempre debe estar en una carpeta con el mismo nombre que el fichero. (arduino, 2016)

Anteriormente a la versión 1.x, de Arduino se usaba la extensión “pde”. Cuando se pasó a la versión 1.6 hubo grandes cambios, que deben tenerse en cuenta si se usa el código antiguo al pasarlo al nuevo. (arduino, 2016)

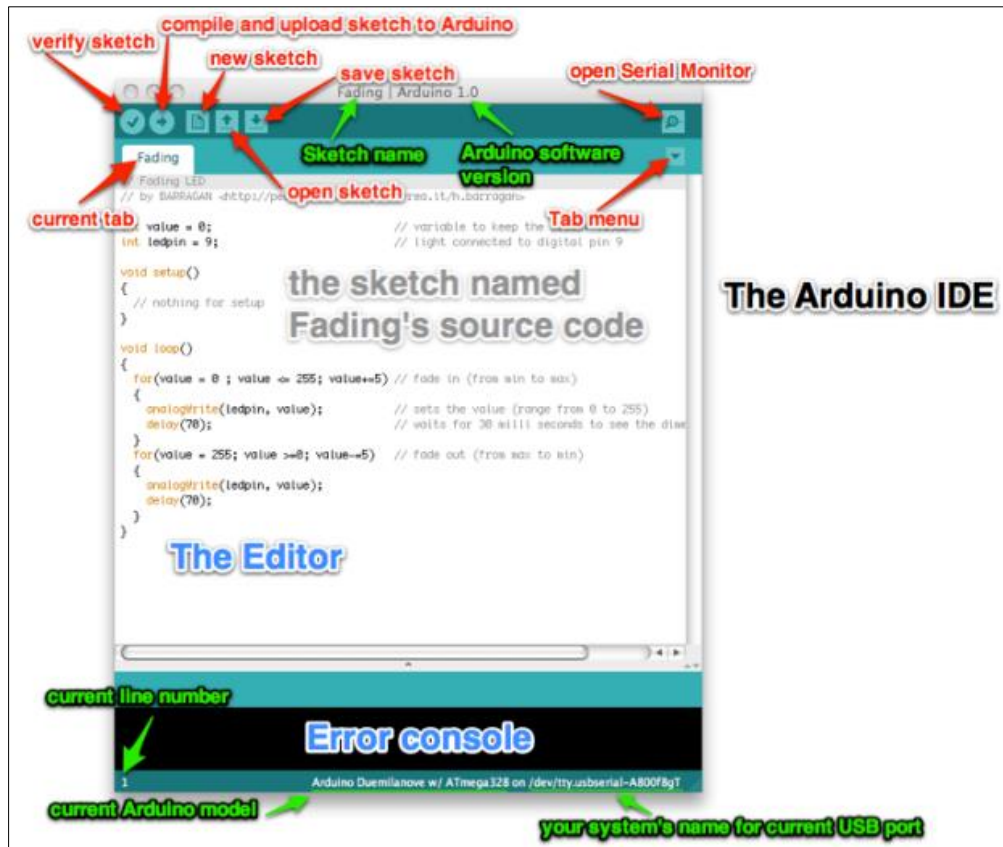


Figura 8 Ventana IDE de Arduino

Fuente: (arduino, 2016)

Cada uno de los menús y opciones que tiene, los más importantes son:

- Botones de Verificar y Subir
- Botón Monitor Serie
- Consola de Error
- Menú herramientas Placa y Puerto
- Puerto y placa seleccionada
- Menú preferencias
- Proyecto/Sketch

Resaltado de palabras clave Las novedades de la nueva versión 1.6 del IDE:

- Soporte multiplataforma de Arduino
- Detección automática de la placa conectada
- Muestra memoria Flash y SRAM ocupada por un sketch o proyecto
- Autoguardado al compilar y cargar sketch

- Carga de sketch vía red Wifi o Ethernet) para Arduino Yun.

Según (arduino, 2016)

El IDE de Arduino es una multiplataforma y en caso de instalar el IDE Arduino en otros sistemas operativos estas son las instrucciones para:

- Windows.
- Mac OSX.
- Linux.

2.3.3 Carga del Programa en Arduino.

Según (arduino, 2016)

El IDE de Arduino contiene un editor de texto para escribir el sketch, una consola de error, un área con los menús y los botones que realizan las funciones más comunes como son abrir sketch, guardar sketch, compilar y cargar programa. (arduino, 2016)

Cuando cargamos un programa en Arduino, se está usando el bootloader de Arduino, que es un pequeño programa cargado en el microcontrolador que permite subir el código sin usar hardware adicional. El bootloader está activo unos segundos cuando se resetea la placa, después comienza el programa que tenga cargado el Arduino en su memoria Flash. El led integrado en la placa (pin 13) parpadea cuando el bootloader se ejecuta. (arduino, 2016)

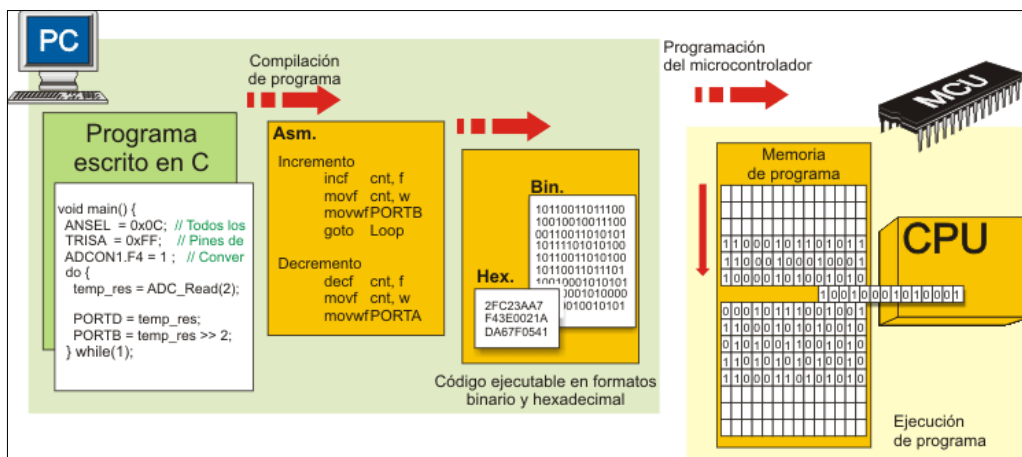


Figura 9 Entorno IDE de Arduino

Fuente: (arduino, 2016)

2.3.4 Configuración inicial de IDE

Según (arduino, 2016)

Después de la instalación, lo primero que se realiza es configurar el IDE para facilitar la edición de los programas, donde se muestra la información de compilación como errores,... etc para teóricamente cargar el programa realizado en Arduino. Cuanta más información se tenga, más fácil será localizar un problema. (arduino, 2016)

Para ello, se ingresa en el menú Archivo → preferencias y activar:

- Números de Línea.
- Mostrar salida detallada en la compilación y al subir un sketch.
- Configurar la ruta de nuestro workspace.
- Advertencia del compilador: Todos.
- Asociar extensión .ino a nuestro IDE.
- Habilitar plegado de código.
- Verificar el código después de subir.

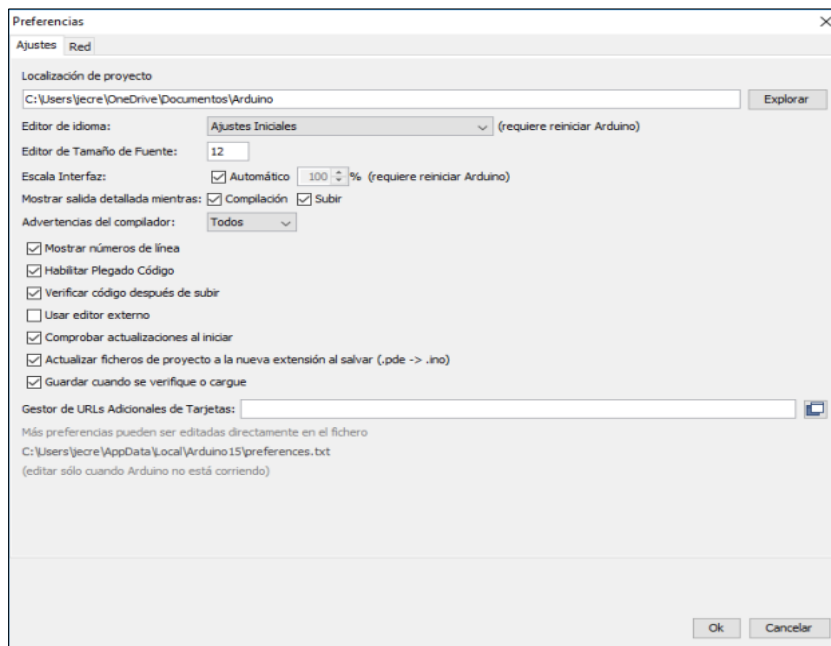


Figura 10 Configuración inicial de IDE

Fuente: (arduino, 2016)

Según (arduino, 2016)

Desde esta pantalla se configura donde se guardan las preferencias, sketches y librerías, de forma que al instalar una actualización se mantengan todos los datos o si se instala varios IDEs van a compartir estos datos.

Las preferencias se guardan en el directorio:

- C:\Users\nombre_usuario\AppData\Local\Arduino15\, así como el listado de librerías y placas disponibles desde el gestor de librerías y tarjetas.

Los sketches y librerías se guardan en:

- C:\Users\nombre_usuario\Documentos\Arduino. (arduino, 2016)

2.3.5 Gestor de tarjetas

Según (arduino, 2016)

El gestor de tarjetas disponible desde el menú herramientas → Placa → Gestor de tarjetas, se muestra a qué tipo de placas se tiene soporte para programarlas y además se puede instalar el soporte para otro tipo de placas que no son Arduino. (arduino, 2016)

Por defecto se tiene instalado el soporte a las placas Arduino AVR que son la mayoría, pero que permite instalar el soporte para los Arduino con MCU ARM de 32 bits como el Arduino MKR1000 o las Intel como el Arduino 101. (arduino, 2016)

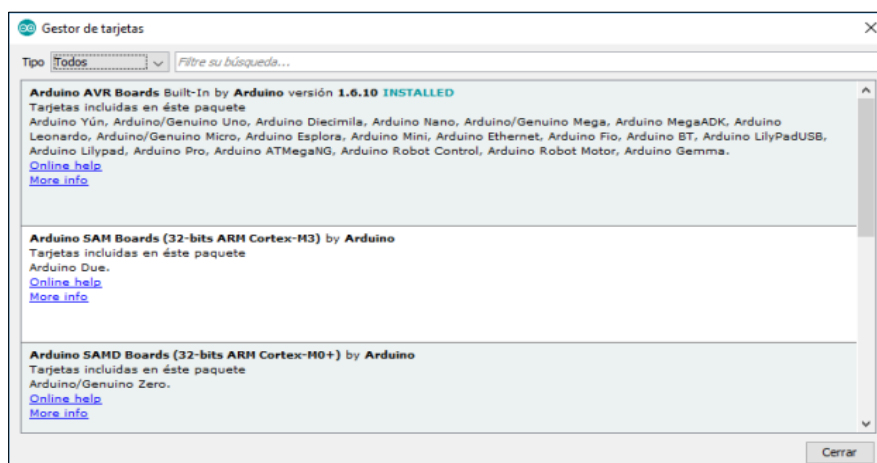


Figura 11 Gestor de tarjetas

Fuente: (arduino, 2016)

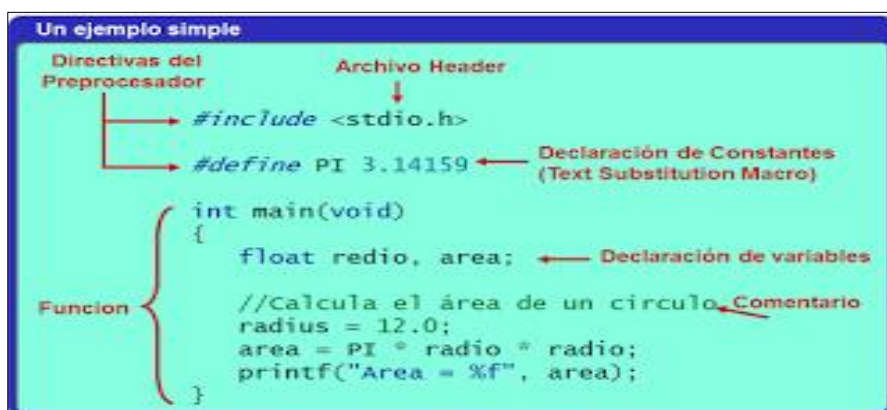
2.4 Programación de Arduino.

Según (arduino, 2016)

El lenguaje de programación de Arduino no es un C++ puro sino que es una adaptación que proviene de avr-libc que provee de una librería de C de alta calidad para usar con GCC (compilador de C y C++) en los microcontroladores AVR de Atmel y muchas utilidades opensource específicas para las MCU AVR de Atmel como avr-dude. (arduino, 2016)

Las herramientas necesarias para programar los microcontroladores AVR de Atmel son avr-binutils, avr-gcc y avr-libc y ya están incluidas en el IDE de Arduino, pero cuando se compila y se carga un sketch se está usando estas herramientas. (arduino, 2016)

Aunque se hable de que hay un lenguaje propio de programación de Arduino, no es cierto, la programación se hace en C++ pero Arduino ofrece unas librerías, también llamado core, que facilitan la programación de los pines de entrada y salida y de los puertos de comunicación. (arduino, 2016) Así como otras librerías para operaciones específicas. El propio IDE ya incluye estas librerías de forma automática y no es necesario declararlas expresamente. Otra diferencia frente a C++ standard es la estructura del programa ya que no usa la función main(), sino que usa las funciones setup() y loop().(arduino, 2016)



```
Un ejemplo simple
Directivas del Preprocesador      Archivo Header
↓
#include <stdio.h>
#define PI 3.14159 ← Declaración de Constantes (Text Substitution Macro)

int main(void)
{
    float radio, area; ← Declaración de variables
    //Calcula el área de un círculo ← Comentario
    radius = 12.0;
    area = PI * radio * radio;
    printf("Area = %f", area);
}
```

Figura 12 Programación en Arduino

Fuente: (arduino, 2016)

Según (arduino, 2016)

En el 99% de los casos se puede hacer un proyecto de Arduino de cierta complejidad con la librería que ofrece el core de Arduino y no es necesario añadir más instrucciones ni tipos de datos que los que hay en el core. Es posible usar comandos estándar de C++ en la programación de Arduino siempre que estén incluidos en el avr. (arduino, 2016)

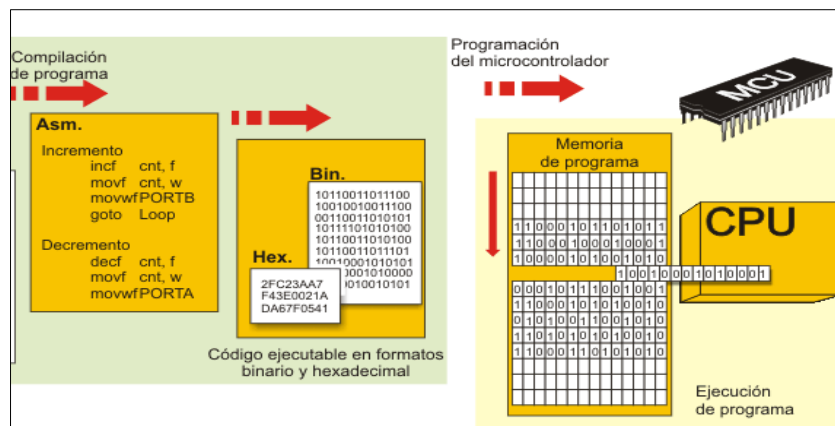


Figura 13 Carga y compilación en el programa

Fuente:(arduino, 2016)

2.4.1 Variables.

Según (arduino, 2016)

Una variable puede ser declarada al inicio del programa antes de la parte de configuración `setup()`, a nivel local dentro de las funciones, y, a veces, dentro de un bloque, como para los bucles del tipo `if... for...`, etc. En función del lugar de declaración de la variable así se determinará el ámbito de aplicación, o la capacidad de ciertas partes de un programa para hacer uso de ella. (arduino, 2016)

2.4.2 Variable global

Según (arduino, 2016)

Es aquella que puede ser vista y utilizada por cualquier función y estamento de un programa. Esta variable se declara al comienzo del programa, antes de `setup()`. Y se recuerda que al declarar una variable global, está ocupa un espacio en la memoria permanente en la zona del `static data` y el abuso de variables globales supone un uso ineficiente de la memoria. (arduino, 2016)

2.4.3 Variable local

Según (arduino, 2016)

Es aquella que se define dentro de una función o como parte de un bucle. Sólo es visible y sólo puede utilizarse dentro de la función en la que se declaró. Por lo tanto, es posible tener dos o más variables del mismo nombre en diferentes partes del mismo programa que pueden contener valores diferentes, pero no es una práctica aconsejable porque complica la lectura de código. (arduino, 2016)

El modificador de variable `static`, es utilizado para crear variables que solo son visibles dentro de una función, sin embargo, al contrario que las variables locales que se crean y destruyen cada vez que se llama a la función, las variables estáticas mantienen sus valores entre las llamadas a las funciones. (arduino, 2016)

2.5 Datos y Variables

2.5.1 Tipos de variables

Según (arduino, 2016)

Además de usar este tipo de datos que son los que aparecen en el “reference” de Arduino, es posible usar cualquier tipo de variable de C++ estándar con las limitaciones propias de cada microcontrolador. (arduino, 2016)

2.5.2 Operadores

Según (arduino, 2016)

El core de Arduino ofrece una serie de operadores según su reference:

- Aritméticos:

➤ `+, -, *, /`

- Compuestos:

➤ `++, --, +=, -=, *=, /=`

- Comparación:

➤ `==, !=, <, >, <=, >=`

2.5.3 Estructuras de control.

Según (arduino, 2016)

Las estructuras de control en Arduino según el referente son:

- Estructuras de decisión
 - If, else, switch case.
- Estructuras de repetición
 - for, while, do..., while.

2.5.4 Funciones definidas por el usuario.

Según (arduino, 2016)

Una función es un bloque de código que tiene un nombre y un conjunto de instrucciones que son ejecutadas cuando se llama a la función. Son funciones `setup()` y `loop()`.(arduino, 2016)

Las funciones de usuario pueden ser escritas para realizar tareas repetitivas y para reducir el tamaño de un programa. Segmentar el código en funciones que permite crear piezas de código que hacen una determinada tarea y volver al área del código desde la que han sido llamadas. (arduino, 2016)

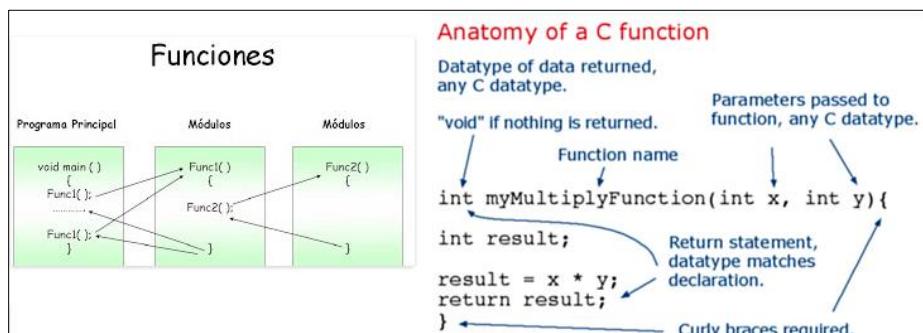


Figura 14 Funciones definidas

Fuente: (arduino, 2016)

2.6 Ventajas y desventajas de Arduino

2.6.1 Ventajas

Según (Acamica, 2016)

- Simplifica. El proceso de trabajar con microcontroladores.
- Bajos costos. Las placas Arduino son más accesibles comparadas con otras plataformas de microcontroladores. Los módulos más caros de Arduino pueden ser montadas a mano bajando sus costos. (Acamica, 2016)
- Multi-Plataforma. El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux; mientras que la mayoría de entornos para microcontroladores están únicamente limitados a Windows. (Acamica, 2016)
- Entorno de programación simple y directa. El entorno de programación de Arduino es fácil de usar para principiantes y flexible para los usuarios avanzados. Además, Arduino está basado en el entorno de programación de Processing, con lo que los estudiantes que aprendan a programar en este entorno se sentirán familiarizados con el entorno de desarrollo Arduino. (Acamica, 2016)
- Software ampliable y de código abierto. El software Arduino es de distribución de licencia libre y preparada para ser adaptado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y en caso de querer profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado. (Acamica, 2016) De igual modo se puede añadir directamente código en AVR C en los programas de los usuarios, si es que así lo desean. (Acamica, 2016)
- Hardware ampliable y de Código abierto. Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de

desarrollo para entender cómo funciona y ahorro de dinero. (Acamica, 2016).

2.6.2 Desventajas.

Según (Acamica, 2016)

Dado que la programación no se realiza en Ensamblar, el precio a pagar por el uso de las librerías es un retraso en la ejecución de las instrucciones, algunos microsegundos que en el caso de dispositivos de uso cotidiano son irrelevantes, pero significativos a la hora de hacer adquisición de datos. (Acamica, 2016)

El hecho de que la plataforma venga ya ensamblada le quita flexibilidad a los proyectos, así por ejemplo se estaría obligatoriamente a usar un espacio y forma acorde con el PCB del Arduino, para superar esto, se debe trabajar con un microcontrolador diferente al de la plataforma y diseñar las PCB desde cero como con los PICs. (Acamica, 2016)

2.7 Arduino MP3 shield board con TF card

Según (geeetech, 2014)

Para la mejor experiencia de audio MP3, el VS1053 incluye procesamiento especial de EarSpeaker con altavoces estéreo. Además de ser capaz de decodificar todos los formatos principales, VS1053 es capaz de grabar en el archivo Ogg Vorbis. (geeetech, 2014)

Con la ranura para tarjetas TF en el otro lado de la placa, puede reproducir archivos mp3 de la tarjeta TF utilizando este breakout. Usando Arduino u otro microcontrolador. (geeetech, 2014)

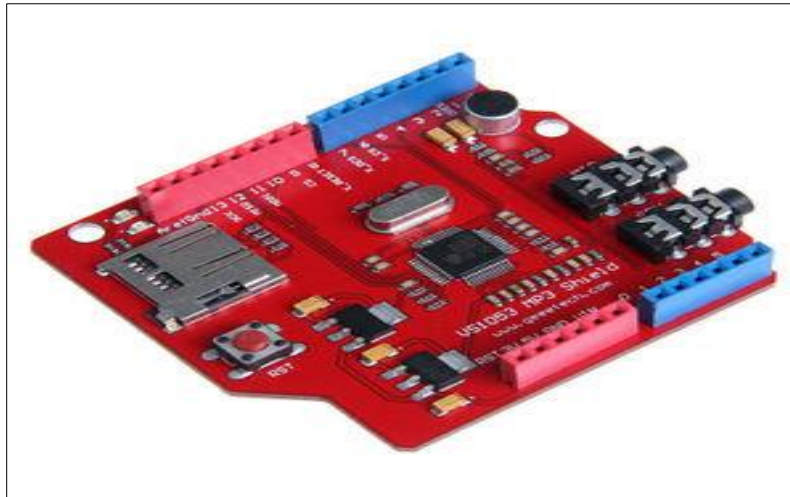


Figura 15 Arduino MP3 shield board con TF card

Fuente: (geeetech, 2014)

2.7.1 Características de Arduino MP3 shield board con TF card

Según (geeetech, 2014) las características de Arduino MP3 shield son:

- Puede reproducir una variedad de formatos de música, el apoyo a la codificación OGG en tiempo real de grabación
- SPI, las líneas de señal de control son conducidas hacia fuera
- Auriculares y salida estéreo
- Un micrófono para la grabación
- Una interfaz de entrada line_in
- Indicador de encendido
- 3.3V y 2.8V del chip LDO AMS-1117 a bordo, proporciona hasta 800mA de corriente
- Una sola fuente de alimentación: +5 VDC
- 12.288 MHz de cristal
- Ranura para tarjetas TF. (geeetech, 2014)

2.8 Circuito integrado 74HC573

Según (Fairchild semiconductor TM, 2017)

El CI 74HC573 octal de tipo-D de alta velocidad utiliza avanzada tecnología CMOS. Poseen una alta inmunidad al ruido y un bajo consumo de energía de

los circuitos integrados CMOS estándar, estos dispositivos son ideales para la interconexión con líneas de datos en un sistema organizado en un bus de datos. (Fairchild semiconductor TM, 2017)

Cuando la entrada LATCH ENABLE (LE) está en ALTA, las salidas Q salen de las entradas D. Cuando LATCH ENABLE se pone LOW, los datos en las entradas D se conservarán en las salidas hasta que LATCH ENABLE vuelva a ser HIGH nuevamente (Fairchild semiconductor TM, 2017). Cuando se aplica un nivel lógico ALTO a la entrada OUTPUT CONTROL OC, todas las salidas pasan a un estado de impedancia ALTA, independientemente de qué señales están presentes en las otras entradas y el estado de los elementos de almacenamiento (Fairchild semiconductor TM, 2017). La familia lógica 74HC es la velocidad, la función y el pinout compatibles con la familia lógica 74LS estándar. Todas las entradas están protegidas de daños debidos a descargas estáticas por medio de pinzas internas de diodo a VCC y tierra. (Fairchild semiconductor TM, 2017)

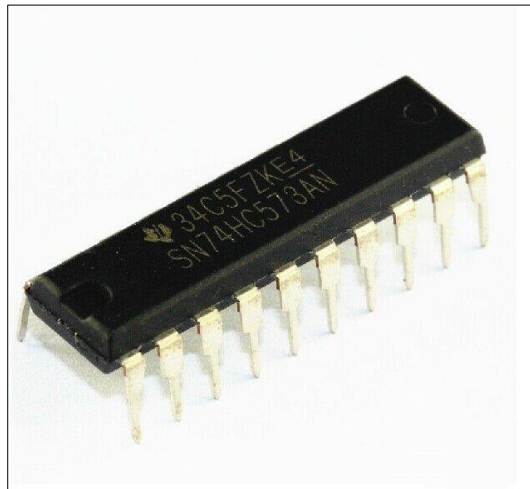


Figura 16 Circuito integrado 74HC573

Fuente: (Fairchild semiconductor TM, 2017)

2.8.1 Características circuito integrado 74HC573

Según (Fairchild semiconductor TM, 2017) las características son:

- Retardo de propagación típico: 18 ns.
- Amplio rango de voltaje de funcionamiento: 2 a 6 voltios.
- Corriente de entrada baja: 1 μ A máximo.

- Corriente de reposo baja: 80 μ A máxima (serie 74HC).
- Compatible con sistemas orientados al bus.
- Capacidad de salida: 15 cargas LS-TTL.

2.9 Circuito integrado ULN2803.

Según (Texas Instruments, 2017)

El dispositivo ULN2803 es un dispositivo Darlington de 50 V, 500 mA matriz de transistores. El dispositivo consta de ocho NPN Darlington que cuentan con salidas de alto voltaje con diodos de abrazadera de cátodo común para conmutación de cargas inductivas. (Texas Instruments, 2017) . La calificación de colector-corriente de cada Darlington es de 500 mA. Los pares Darlington pueden en estar en paralelo para una mayor capacidad de corriente. Las aplicaciones incluyen controladores de relé, controladores de martillo, controladores de lámparas, controladores de pantalla (LED y descarga), controladores de línea y buffers lógicos. (Texas Instruments, 2017). El dispositivo ULN2803A tiene una resistencia de base de la serie 2.7 k Ω para cada par de Darlington opera directamente con TTL o 5V dispositivos CMOS. (Texas Instruments, 2017)



Figura 17 Circuito integrado 74HC573

Fuente: (Texas Instruments, 2017)

2.9.1 Características circuito integrado ULN2803

Según (kitelectronica, 2017) las características son:

- Voltaje de alimentación: Típicamente el ULN2803 se puede utilizar con voltajes de 5, 9 y 12 voltios, pero también puede soportar voltajes hasta de 50 voltios. (kitelectronica, 2017)
- El “voltaje de alimentación” se hace por el pin 10 (COM), que más bien se trata de un punto común para todos los cátodos de los diodos clamp. La conexión a tierra (GND) la encontramos en el pin 9. (kitelectronica, 2017)
- Corriente: La corriente máxima que podemos obtener por salida del ULN2803 es de 500mA, pero la misma se puede incrementar si utilizamos las salidas en paralelo. (kitelectronica, 2017)
- Entradas: Responden a niveles lógicos C-MOS y TTL; soportan un voltaje máximo de 30 voltios. Las entradas se encuentran desde el pin 1 al 8.
- Salidas: Se encuentran desde el pin 11 al 18. Se pueden conectar en paralelo para aumentar la corriente máxima de 500mA.
- Diodos clamp: El ULN2803 posee un diodo polarizado inversamente entre cada salida y el “voltaje de alimentación” (pin 10 (COM)) con el objetivo de eliminar las corrientes inversas, que se producen cuando se conectan cargas inductivas como: relays, solenoides, motores y transformadores. Esto evita que los transistores se dañen. (kitelectronica, 2017)

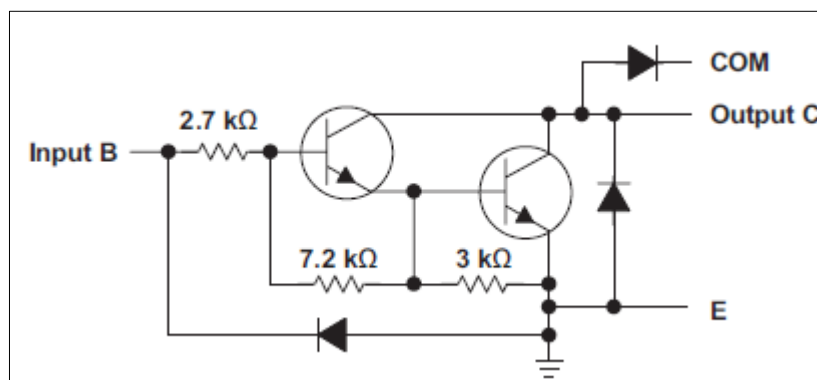


Figura 18 Circuito integrado ULN2803

Fuente: (kitelectronica, 2017)

Según (kitelectronica, 2017)

El circuito integrado ULN2803 posee un par de transistores NPN en configuración Darlington y componentes asociados, lo que le da una alta ganancia de corriente a la salida con poca corriente de entrada. Nótese la

inclusión de la resistencia de 2.7 k Ω en la figura 18, a la entrada del circuito; haciendo que no sea necesario incluirla externamente. (kitelectronica, 2017)

2.10 Diodo emisor de luz (LED)

Según (iluminet, 2017)

El LED (Light-Emitting Diode: Diodo Emisor de Luz) son elementos de estado sólido (semiconductores) que emiten energía luminosa al ser alimentados directamente por una energía eléctrica, los cuales dependiendo de su operación pueden ser de baja o alta potencia. (iluminet, 2017) Los Leds de alta potencia más utilizados son los de potencias de 1W, aunque actualmente existen avanzados diseños en potencias de 3, 5, 10, 20 y 30 W. Los Leds de baja potencia son diseños sencillos, que no incluyen ningún tipo de óptica de control del flujo luminoso y son de potencias de hasta 0.12 W; este tipo de leds se utilizan principalmente para aplicaciones de señalización o indicación. (iluminet, 2017)

Los leds de alto brillo son diseños más completos que incluyen diversas alternativas de ópticas de control del flujo luminoso; este tipo de leds se utilizan principalmente para iluminación concentrada en aplicaciones, permitiendo generar amplias posibilidades creativas de diseño y efectos de color. (iluminet, 2017)



Figura 19 Diodo led de alta luminosidad

Fuente: (iluminet, 2017)

2.10.1 Características de diodo led de alta luminosidad

Según (iluminet, 2017) estas son las características:

Los diodos emisores de luz, de alta luminosidad tienen lo siguiente.

- Vida promedio de 50,000 horas.
- Flujo luminoso de 55 lúmenes.
- Eficacia de 55 lm/W.
- Voltaje de operación de 3-4 voltios de corriente directa.
- Corriente de operación de 350 mA.
- Bajas pérdidas por disipación de calor.
- Colores blanco, azul y verde fabricados de Nitruro de Galio e Indio.
- Ámbar y rojo fabricados con Fosforo de Galio, indio y Aluminio.

2.11 Modulo teclado matricial.

Según (disca.upv, 2017)

Hay circuitos especializados en esta tarea, pero es fácil hacer que un microcontrolador lea estos teclados matriciales (los ordenadores compatibles PC incluyen un pequeño microcontrolador que hace esta tarea y envía las teclas pulsadas a la unidad central). La disposición en matriz de los teclados responde a la necesidad de leer gran cantidad de conmutadores con pocas líneas de entrada. (disca.upv, 2017)

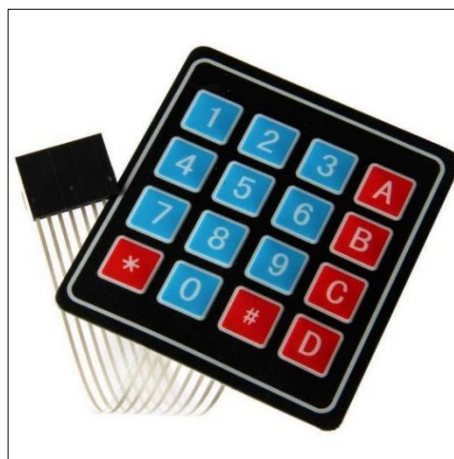


Figura 20 teclado matricial

Fuente: (disca.upv, 2017)

2.11.1 Características

Según (disca.upv, 2017)

Las características del teclado matricial se aprecian en lo siguiente:

- 16 botones con organización matricial (4 filas x 4 columnas)
- Teclado tipo membrana
- Mayor resistencia al agua y al polvo
- Auto adhesivo en la parte de atrás
- Tiempo de rebote (Bounce time): ≤ 5 ms
- Máximo voltaje operativo: 24 V DC
- Máxima corriente operativa: 30 mA.
- Resistencia de aislamiento: 100 M Ω (a 100 V)
- Voltaje que soporta el dieléctrico: 250 VRMS (@ 60Hz, por 1 min)
- Expectativa de vida: 1.000.000 de operaciones
- Dimensiones del pad: 6.9 x 7.7 cm aprox.
- Cable de cinta plana de 8.3 cm de largo aprox. (incluido el conector)
- Conector tipo DuPont hembra de una fila y 8 contactos con separación estándar 0.1" (2.54mm)
- Temperatura de operación: 0 a 50 °C

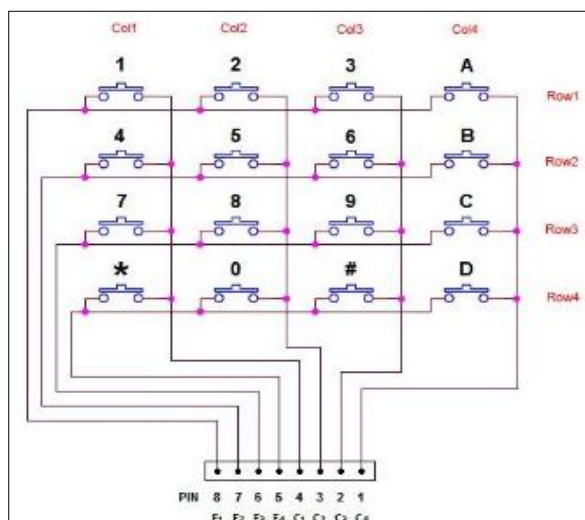


Figura 21 Esquema teclado matricial interno

Fuente: (disca.upv, 2017)

CAPÍTULO III

DESARROLLO DEL TEMA

3.1. PRELIMINARES

En este capítulo se explica de manera clara y concisa paso a paso la implementación del material audio visual, utilizando Arduino Uno, Arduino mega y Arduino shield MP3 además de los requerimientos mínimos de software para la programación del mismo.

3.2. REQUERIMIENTOS MÍNIMOS DE HARDWARE

Los componentes utilizados para la implementación de material audiovisual fueron los siguientes:

- Leds de alto brillo
- Circuito integrado 74HC573
- Circuito integrado ULN2803
- Arduino Mega
- Arduino MP3 shield board con TF card
- Arduino Uno
- Teclado Matricial 4x4

3.2.1. Leds de alto brillo

Los leds de alto brillo son utilizados para una mejor visibilidad, ya que brindan una mayor luminosidad a diferencia de los led comunes, son la manera más eficiente de convertir la corriente eléctrica en iluminación. Es un dispositivo electrónico excepcional para una gran variedad de aplicaciones de óptica con múltiples flujos luminosos de luz y temperatura, los cuales brindan un menor consumo de energía y alta durabilidad.

En este proyecto de utilizaron leds de alto brillo de 3mm color azul para hacer el cubo led, como se muestra en la siguiente figura.



Figura 22 Diodo led de alta luminosidad

3.2.2. Cubo led y placa de control

Lo siguiente fue realizar el montaje del cubo de leds sobre la PCB o placa de control, el cual permitirá controlar los leds para formar las letras y números.

Una vez que se tienen los leds de alta luminosidad se procede a soldarlos para formar el cubo led.

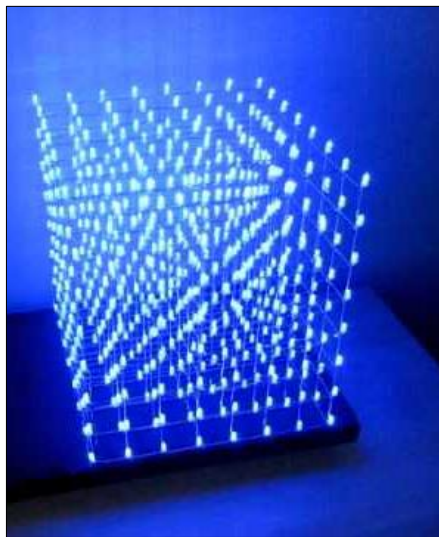


Figura 23 Cubo led

Formado el cubo, se arma la placa de control con los diferentes circuitos integrados.

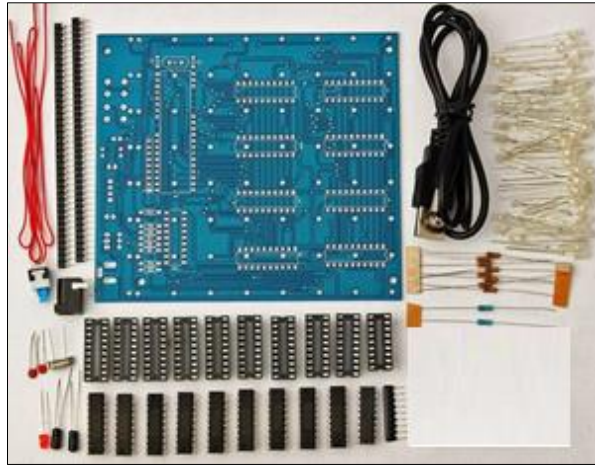


Figura 24 PCB para controlar en cubo led.

La placa armada, se conecta con el cubo led para ser controlado, con los circuitos integrados que se mencionan a continuación.

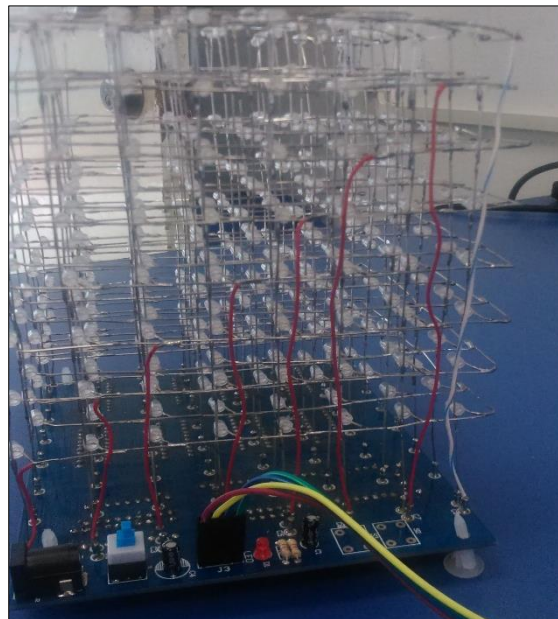


Figura 25 Cubo led y PCB

3.2.3. Funcionamiento del circuito integrado 74HC573

Ya armada y montada la PCB con el cubo led, se integran los demás circuitos de control, lo que realiza el CI 74HC573 es que cuando la entrada LATCH ENABLE (LE) está en HIGH, las salidas Q salen de las entradas D, y cuando

LATCH ENABLE se pone LOW, los datos en las entradas D se conservarán en las salidas hasta que LATCH ENABLE vuelva a ser HIGH nuevamente.

Se aplica un nivel lógico HIGH a la entrada OUTPUT CONTROL OC, todas las salidas pasan a un estado de impedancia HIGH, independientemente de qué señales están presentes en las otras entradas y el estado de los elementos de almacenamiento.

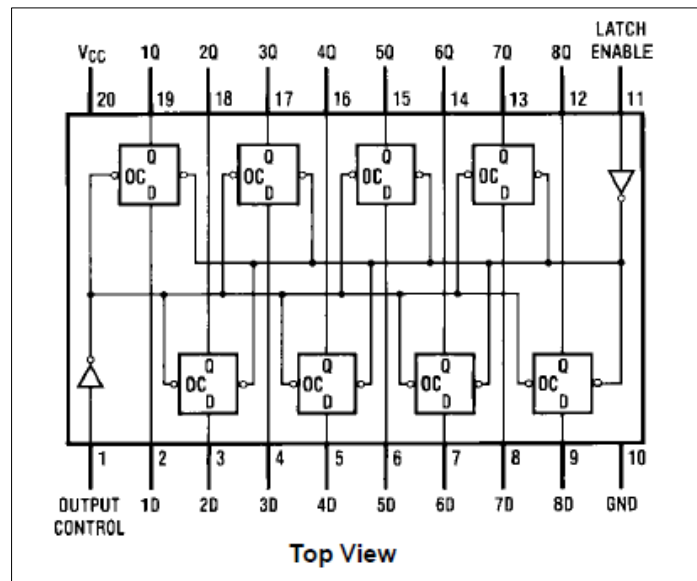


Figura 26 Diagrama interno del circuito integrado 74HC573

Fuente.- (Fairchild semiconductor TM, 2017)

En este caso se utilizaron 8 circuitos integrados 74HC573, los cuales controlan a los 64 ánodos (positivos) de los leds, Todas las entradas están protegidas de daños debido a descargas estáticas por medio de pinzas internas de diodo a VCC y tierra.

3.2.4. Funcionamiento del circuito integrado ULN2803

El circuito integrado ULN2803 es un dispositivo Darlington de 50 V, con una corriente de 500 mA, con una matriz de transistores. El dispositivo consta de ocho transistores NPN Darlington que cuentan con salidas de alto voltaje con diodos de cátodo común.

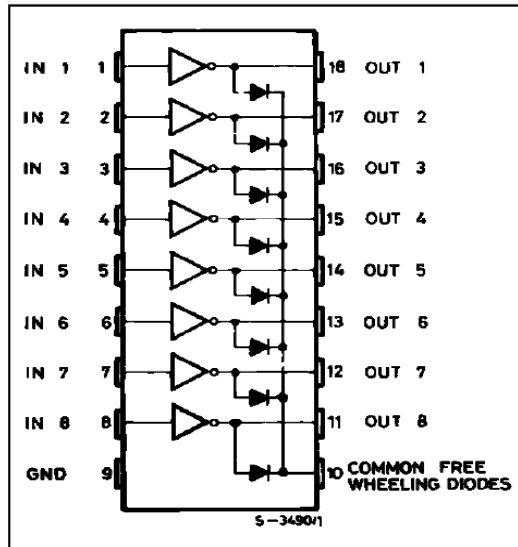


Figura 27 Diagrama interno de circuito integrado ULN2803

Fuente.- (Texas Instruments, 2017)

Este circuito integrado permite que las entradas respondan a niveles lógicos C-MOS y TTL los cuales permiten que se puedan controlar, con las 8 segmentos negativos del cubo led.

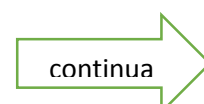
3.2.5. Arduino Mega

Arduino Mega está basado en el microcontrolador ATmega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas análogas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16 MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset.

Tabla 1

Conexión de la PCB con Arduino Mega

ARDUINO MEGA	PCB
Pin 2	P00
Pin 3	P01
Pin 4	P02
Pin 5	P03
Pin 6	P04
Pin 7	P05
Pin 8	P06



Pin 9	P07
Pin 30	P10
Pin 31	P11
Pin 32	P12
Pin 33	P13
Pin 34	P14
Pin 35	P15
Pin 36	P16
Pin 37	P17
Pin 40	P20
Pin 41	P21
Pin 42	P22
Pin 43	P23
Pin 44	P24
Pin 45	P25
Pin 46	P26
Pin 47	P27

Fuente: Carlos Caillagua

Para el funcionamiento del cubo led se conecta las salidas del Arduino mega con las entradas de la PCB hacia los pines ya mencionados en la tabla anterior.

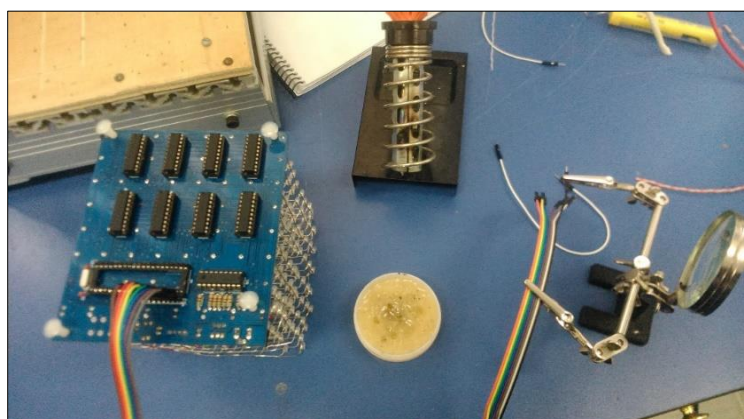


Figura 28 pines de conexión con la PCB y Arduino

3.2.6. Arduino UNO

En este proyecto se utiliza la placa Arduino UNO, esta placa tiene un microcontrolador ATmega 328P-PU, de 8bits de la familia de los AVR. La cual consta de 14 pines digitales de E / S, 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación y un botón de reinicio.

Tabla 2

Conexión del teclado matricial con Arduino shield MP3

Teclado matricial	Arduino Shield MP3
In 5	A0
In 6	A1
In 7	A2
In 8	A3
In 1	A4
In 2	A5
In 3	P6
In 4	P5

Fuente: Carlos Caillagua

Tabla 3

Conexión para la comunicación con Arduino Uno y Mega

Arduino Uno	Arduino Mega
RX0	TX1
GND	GND

Fuente: Carlos Caillagua

3.2.7. Arduino MP3 shield board con TF card

Esta tarjeta shield MP3 está diseñado con base en VS1053B de VLSI. Usando este shield se puede conectar directamente con Arduino UNO enganchándose con los pines del mismo, VS1053 es un procesador versátil de códec MP3 que

es capaz de decodificar una variedad de formatos de música, incluyendo Ogg Vorbis / MP3 / AAC / WMA / audio MIDI. Con este módulo se realiza la reproducción de audio en formato MP3.

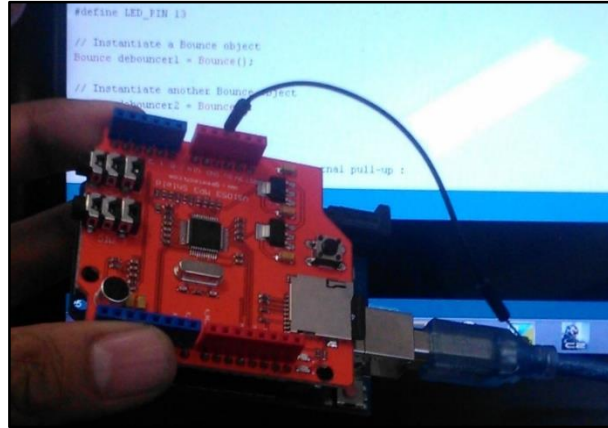


Figura 29 Arduino shield mp3

3.3. REQUERIMIENTOS MÍNIMOS DEL SOFTWARE

En el software de Arduino se realiza la programación para poder controlar el cubo led mediante coordenadas X, Y, Z; ya que se trata de un cubo en tres dimensiones, se utilizan estos puntos como referencia para realizar animaciones o en este caso letras y números.

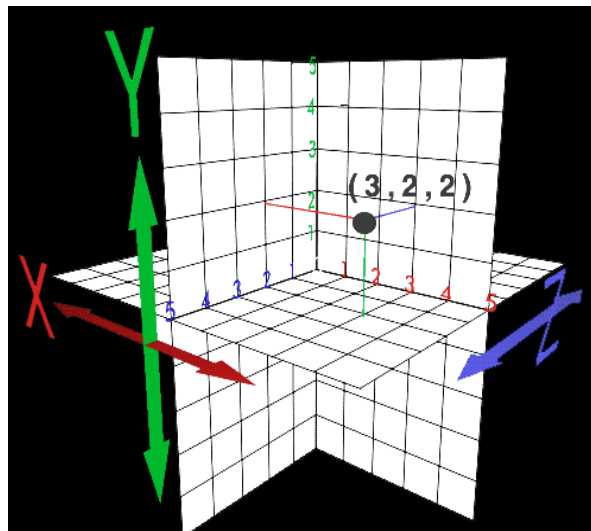


Figura 30 grafica tridimensional con los ejes X, Y, Z
Fuente.- (infor.uva, 2017).

El software a utilizar para la programación es Arduino 1.6.11.



Figura 31 Entorno de programación.

3.3.1. Líneas de programación para letras y números en el cubo led

Para el primer punto de programación es importante incluir las librerías, las cuales activaran los comandos para que los dispositivos externos, puedan ser compatibles con el entorno de programación de Arduino.

Programación para la letra A.

word tiempo=1; variable para un numero natural(entero positivo) y tiempo valor en uno

void setup() // Pone el código del arreglo (algoritmo), para correr una vez

{ // Llave para abrir la programación.

Después de asigna los pines del Arduino como salidas digitales los que se vaya a utilizar.

pinMode(2,OUTPUT); asignación del pin 2 de Arduino como salida digital

pinMode(3,OUTPUT); asignación del pin 3 de Arduino como salida digital

pinMode(4,OUTPUT); asignación del pin 4 de Arduino como salida digital

pinMode(5,OUTPUT); asignación del pin 5 de Arduino como salida digital

//DATOS

digitalWrite(2,LOW);//P00 escribe un valor LOW en el pin digital 2

digitalWrite(3,LOW);//P01 escribe un valor LOW en el pin digital 2

//FILAS EJE Z

digitalWrite(30,LOW);//P10 escribe un valor LOW en el pin digital 30

digitalWrite(31,LOW);//P11 escribe un valor LOW en el pin digital 31

digitalWrite(32,LOW);//P12 escribe un valor LOW en el pin digital 32

//ENABLED LATCH

digitalWrite(40,HIGH);//P20 escribe un valor HIGH en el pin digital 40

digitalWrite(41,HIGH);//P21 escribe un valor HIGH en el pin digital 41

//ENABLED LATCH

digitalWrite(40,LOW);//P20 escribe un valor LOW en el pin digital 2

digitalWrite(41,LOW);//P21 escribe un valor LOW en el pin digital 2

}

void loop() declaración de funciones sin valor de retorno

{

En este paso se escribe un valor ya sea en alto (HIGH) o en bajo (LOW) para activar el led de la coordenada asignada en X, Y, Z para formar las letras y los números.

//PASO 1ª

//DATOS EJE X (74HC573)

digitalWrite(2,LOW);//P00 X0// 39

digitalWrite(3,HIGH);//P01 X1// 38

digitalWrite(4,HIGH);//P02 X2// 37

```
digitalWrite(5,LOW);//P03 X3// 36
digitalWrite(6,LOW);//P04 X4// 35
digitalWrite(7,HIGH);//P05 X5// 34
digitalWrite(8,HIGH);//P06 X6// 33
digitalWrite(9,LOW);//P07 X7// 32//0
```

```
//ENABLED LATCH EJE Y (74HC573)
```

```
digitalWrite(40,LOW);//P20 Y0//21 pins conectados al cubo
digitalWrite(41,HIGH);//P21 Y1//22
digitalWrite(42,LOW);//P22 Y2//23
digitalWrite(43,LOW);//P23 Y3//24
```

```
//FILAS EJE Z (ULN2803)
```

```
digitalWrite(30,HIGH);//P10 Z0 //1//0 pins conectados al cubo
digitalWrite(31,HIGH);//P11 Z1// 2
digitalWrite(32,HIGH);//P12 Z2// 3
delay(tiempo);
```

En este paso (borrar) el Arduino borra el dato ya enviado, y espera a que sea enviado otro bit para poderse visualizar en el cubo led.

```
//borrar
```

```
digitalWrite(2,LOW);//P00
digitalWrite(3,LOW);//P01
digitalWrite(4,LOW);//P02
//FILAS EJE Z
digitalWrite(30,LOW);//P10
digitalWrite(31,LOW);//P11
digitalWrite(32,LOW);//P12
```

```
//ENABLED LATCH
```

```
digitalWrite(40,HIGH);//P20
digitalWrite(41,HIGH);//P21
digitalWrite(42,HIGH);//P22
```

```
}
```

Programación resumida, para la programación completa ver el Anexo A

3.3.2. Líneas de programación para el audio MP3

Para el segundo punto de programación también es importante incluir las librerías, las cuales activaran los comandos para que los dispositivos externos de Arduino puedan ser compatibles con el entorno de programación. Para utilizar una librería es importante añadir al principio de la programación `#include`, en este caso incluye las siguientes librerías.

Programación para el audio MP3.

```
#include <Keypad.h> //librería para el uso de teclado matricial
#include <SPI.h> //librería para el bus de comunicación y transmisión con los
circuitos integrados.
```

```
#include <SdFat.h> // librería para el uso de la tarjeta Memory sd card
#include <SFEMP3Shield.h> // librería para el uso y habilitación del shield mp3.
```

```
SdFat sd; //inicializa tarjeta sd card
```

```
SFEMP3Shield MP3player; // inicializa shield MP3
```

```
int8_t track_actual = 0; // inserta un numero entero entre 32,767 y -32,768
```

En esta parte se asigna las líneas de programación para las filas y columnas además de los pines asignados para conectarlos con el teclado matricial.

```
const byte Filas = 4; //Cuatro filas
```

```
const byte Cols = 4; //Cuatro columnas
```

```
byte Pins_Filas[] = {5, 4, A5, A4}; //Pines Arduino a los que contamos las filas.
```

```
byte Pins_Cols[] = { A3, A2, A1, A0}; // Pines Arduino a los que contamos las
columnas.
```

```
char Teclas [ Filas ][ Cols ] = //carácter ASCII almacenado en 8 bits
```

```
}
```

```
void loop()
```

```
{
```

```
#if defined(USE_MP3_REFILL_MEANS) \
```

```
&& ( (USE_MP3_REFILL_MEANS == USE_MP3_SimpleTimer) \
```

```

|| (USE_MP3_REFILL_MEANS == USE_MP3_Polled) )

MP3player.available();
#endif

char pulsacion = Teclado1.getKey() ; // carácter ASCII almacenado en 8 bits
if (pulsacion != 0) // Si el valor es 0 es que no se ha pulsado ninguna tecla
switch(pulsacion)
{
case '*': // control del flujo de programas
MP3player.stopTrack(); // hacer algo cuando (*) es igual a 1
break; // rompe el flujo y termina la sentencia.
case '0': control del flujo de programas
MP3player.stopTrack();
MP3player.playTrack(1); // hacer algo cuando (1) es igual a 1 // y así para todos
los casos
break; rompe el flujo y termina la sentencia. // y así para todos los casos
case '1':
MP3player.stopTrack();
MP3player.playTrack(2);
break;

} //cierra el programa.

```

Programación resumida, para la programación completa ver el Anexo B

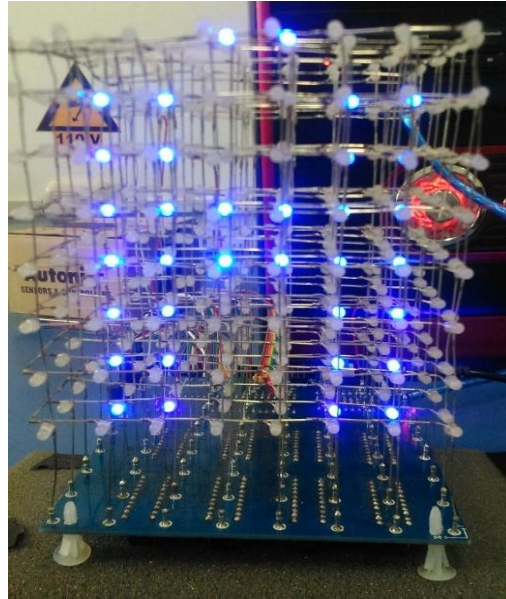


Figura 32 Letra A mostrada en el cubo led

3.3.3. Carga y compilación del programa en la placa Arduino

Al finalizar la programación se compila y verifica que no existan errores para posteriormente grabar el programa en la placa Arduino, hay que dar clic en: Archivo/abrir reciente/prueba1Arduino

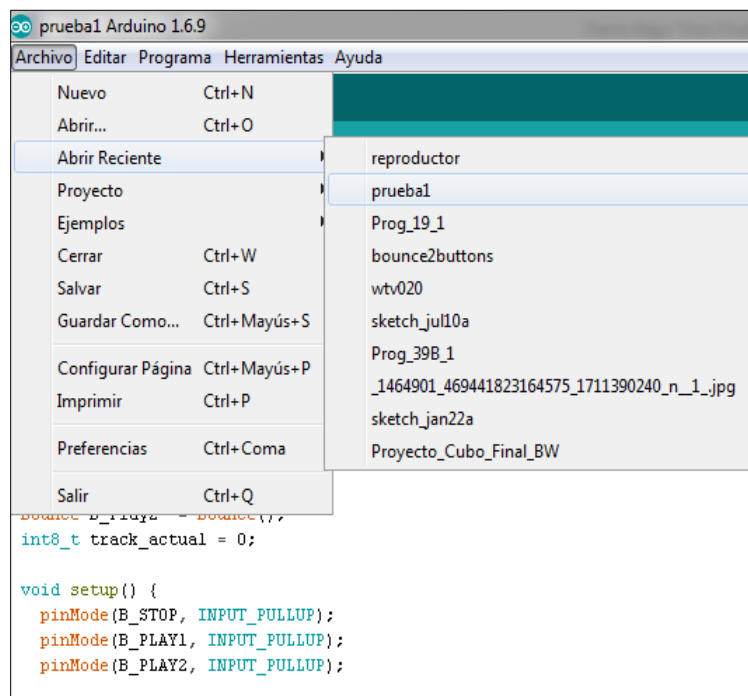


Figura 33 Entorno para abrir el programa

En la barra de herramientas en la parte superior se selecciona el tipo de placa se ingresa a: programar los puertos de programación. En este caso Herramientas/Placa/Arduino Genuino Mega 2560.

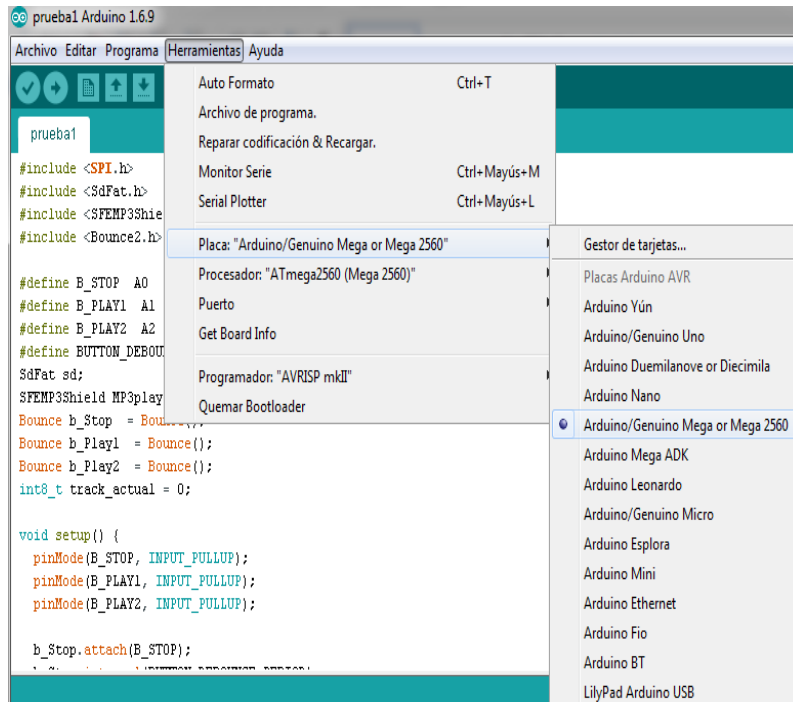


Figura 34 Selección del tipo de placa a utilizar.

Después de elegir en cual placa se va a cargar el programa, se va a Herramientas/Puerto/COM; saldrá una lista de puertos en donde está asignado el puerto de comunicación con la PC y se escoge el que se asigna por default.

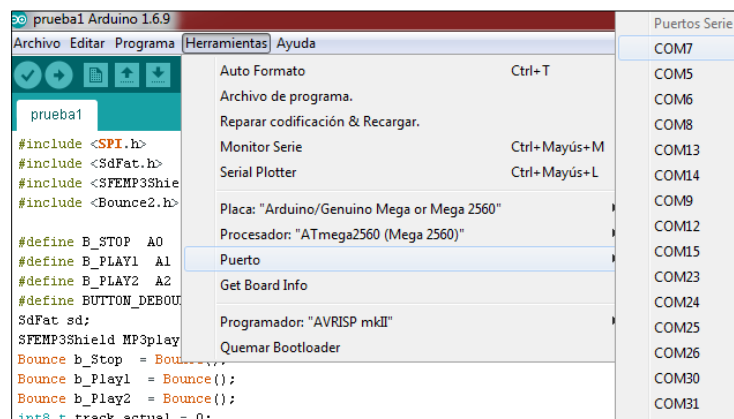


Figura 35 Selección del puerto COM

Una vez configurado el puerto COM seleccionado, se pulsa el botón “SUBIR”

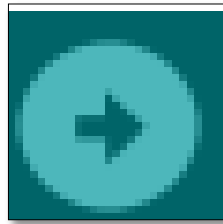


Figura 36 Botón “SUBIR” para cargar el programa en la placa Arduino

En la parte inferior de la ventana, aparecerá el mensaje “CARGA TERMINADA”. Un tiempo después de la subida del programa concluya, el LED ámbar de la placa empezara a parpadear, si lo hace, el Arduino está cargado y ejecutado. Para cargar en la placa Arduino uno es el mismo procedimiento.

3.4. PRUEBAS DE FUNCIONAMIENTO

Para las pruebas de funcionamiento se realizó la primera animación con el encendido de la parte superior e inferior de cubo, formando un cuadrado de fondo que va de abajo hacia arriba como se puede observar en la siguiente figura.

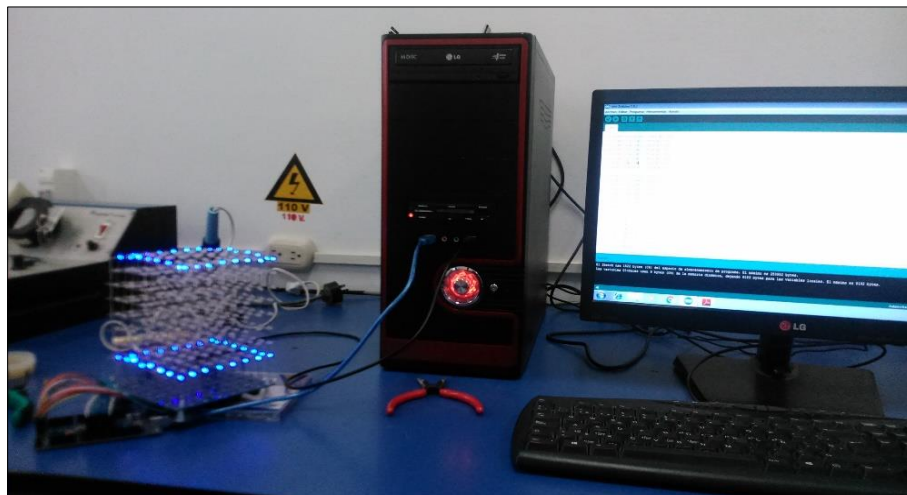


Figura 37 Primera prueba de funcionamiento

Para la alimentación del Arduino y del cubo led se conecta con la pc que entrega un voltaje de alimentación de 5V para cada uno de los dos casos la cual es suficiente para no dañar la placa de Arduino y la PCB del cubo.

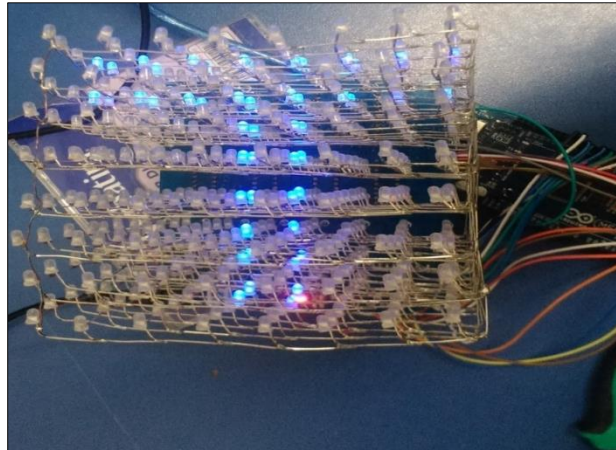


Figura 38 segunda prueba de funcionamiento letra T.

Para la segunda prueba se realiza la animación de letras mediante las coordenadas X, Y, Z como se puede ver en la figura anterior.

3.4.1. Funcionamiento del proyecto

La última prueba se realiza en conjunto con el Arduino shield MP3 para verificar si existe salida de audio y la visualización de la letra A en el cubo led.

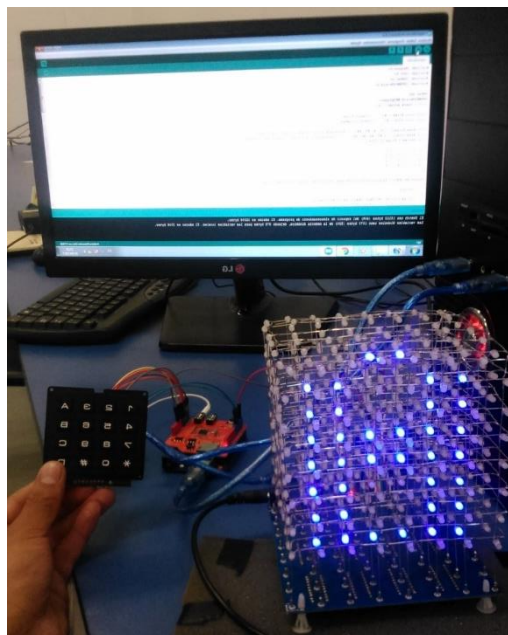


Figura 39 funcionamiento letra A y shield de audio MP3.

Lo que hace es enviar un dato mediante el teclado matricial hacia el Arduino mega y a la PCB del cubo, y mientras el dato es recibido se realiza una comunicación hacia Arduino uno, el cual envía y activa el shield MP3 para la reproducción del audio mientras se muestra la letra.

CAPÍTULO IV

CONCLUSIONES

- Se implementó en este proyecto material didáctico interactivo mediante Arduino y diferentes tipos de dispositivos electrónicos para la mejora de la enseñanza aprendizaje de niños comprendidos entre 3 y 6 años.
- Se indago información referente a los Leds de alto brillo los cuales ofrecen una mayor visibilidad para poder apreciar las animaciones mucho mejor, que con leds comunes, por medio de una pantalla led “cubo”, esto hace de la aplicación práctica de visualización de figuras y animaciones en 3D, mas entretenidas, programado en la interfaz de Arduino y Processing el cual es un entorno de software libre que ofrece esta plataforma.
- Se creó un programa de control con líneas de código abierto, los cuales pueden ser programados mediante el software IDE de Arduino, que es un programa informático compuesto por un conjunto de herramientas de programación, un editor de código, un compilador y un depurador.
- Se realizó pruebas de funcionamiento para la cual se utiliza la PCB de Arduino UNO, el cual tiene incluido un microcontrolador ATmega 328P-PU- 8bits AVR, necesita un cristal de 16MHz externo y una alimentación de 5V, como también la PCB de Arduino MEGA 2560, tiene incluido un microcontrolador ATmega2560, tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), cristal oscilador de 16 MHz y una alimentación de 5v. En el circuito de control, esencial para realizar este proyecto se utiliza CI's existentes en el mercado, componentes convencionales de bajo costo, los cuales funcionan independientemente para cada columna de cátodos y niveles de ánodos, lo que hay que seleccionar e implementar, es el numero primordial de estos componentes que se necesita para poder elegir el tamaño deseado en el cubo final. Así mismo con la implementación de Arduino MP3 Shield para la salida audio, que se controla por interfaz serial “Rx D0, Tx D1 y con un teclado matricial.

RECOMENDACIONES

- Seguir las normas y parámetros establecidos de alimentación que posee cada dispositivo en el proyecto, de manera que no ocasione daños internos y externos, los cuales se encuentran en el datasheet de los mismos.
- Verificar el estado y funcionamiento de los leds ya que algunos se pueden quemar, cuando se los energiza con voltajes superiores a lo establecido por los fabricantes.
- Entender el entorno de programación, y conocer su funcionamiento, así como también las características de la PCB Arduino UNO y la PCB de Arduino MEGA2560, para evitar riesgos, que se puedan provocar en el proceso de utilización.

GLOSARIO DE TÉRMINOS

A

AVR: En inglés, Automatic Voltage Regulator (Regulador Automático de Voltaje), dispositivo de hardware empleado para mantener un voltaje específico en dispositivos electrónicos.

B

BIT: En informática y otras disciplinas, unidad mínima de información, que puede tener solo dos valores (0 o 1) Unidad de medida de la velocidad de transmisión de los datos.

D

DAC: (del inglés digital to analogue converter) es un dispositivo para convertir señales digitales con datos binarios en señales de corriente o de tensión analógica.

F

Flash: realizar el almacenamiento de grandes cantidades de datos en un espacio reducido, permitiendo la lectura y escritura de múltiples posiciones de memoria en la misma operación.

G

GUI: (del inglés graphical user interface), es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

I

IDE: Significa "Integrated Development Environment" ("Entorno de Desarrollo Integrado"). Ya que el Arduino es como un pequeño ordenador que ejecuta una serie de códigos que previamente le hemos introducido, necesitaremos un programa para poder meter estos códigos a la propia placa.

L

LATCH ENABLE: (latch memory en inglés) es un circuito electrónico biestable asíncrono usado para almacenar información en sistemas lógicos digitales. Los latches a diferencia de los flip-flops no necesitan una señal de reloj para su funcionamiento.

M

MICROCONTROLADOR: Es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos.

P

PCB: En electrónica, “placa de circuito impreso” (del inglés: Printed Circuit Board,), es la superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora.

O

OPEN SOURCE: (Código abierto) es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.

T

TTL: trabaja con 5V. Los niveles lógicos vienen definidos por el rango de tensión comprendida entre 0,0V y 0,8V para el estado L (bajo) y los 2,2V y Vcc para el estado H (alto).

REFERENCIAS BIBLIOGRÁFICAS

Acamica. (29 de febrero de 2016). Acamica. Recuperado el 17 de mayo de 2017, de <https://edgardosilvi.wordpress.com/2016/02/29/acamica-ventajas-y-desventajas-de-arduino/>

arduino. (12 de septiembre de 2016). arduino. Recuperado el 17 de mayo de 2017, de <https://aprendiendoarduino.wordpress.com/category/software/>

arduino.cl. (6 de mayo de 2017). arduino.cl. Obtenido de <http://arduino.cl/que-es-arduino/>

disca.upv. (abril de 2017). disca.upv. Recuperado el 19 de mayo de 2017, de <http://www.disca.upv.es/aperles/web51/modulos/teclado/>

Fairchild semiconductor TM. (mayo de 2017). Fairchild semiconductor TM. Obtenido de <http://Fairchild semiconductor TM.com>

geeetech. (septiembre de 2014). geeetech.com. Recuperado el 18 de mayo de 2017, de http://www.geeetech.com/wiki/index.php/Arduino_MP3_shield_board_with_TF_card

iluminet. (abril de 2017). iluminet. Recuperado el 20 de mayo de 2016, de <http://www.iluminet.com/diodos-emisores-de-luz-de-alta-potencia-de-1w/>

infor.uva. (MAYO de 2017). infor.uva. Obtenido de <https://www.infor.uva.es/~descuder/proyectos/animacion/Ejes.htm>

kitelectronica. (mayo de 2017). kitelectronica. Obtenido de <http://www.kitelectronica.com/2016/02/tutorial-circuito-integrado-uln2803.html>

La Importancia del Uso de las TICS en la Educación Inicial. (1 de junio de 2013). Obtenido de <http://tecnologiasaplicadaslaurimar.blogspot.com>

ParasuBebe.com. (24 de noviembre de 2016). 11 Métodos para Enseñar el Abecedario. Obtenido de <http://parasubebe.com/ensenar-el-abecedario/>

Texas Instruments. (mayo de 2017). Texas Instruments. Obtenido de <http://www.ti.com/lit/ds/symlink/uln2803a.pdf>

