UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

INGENIERIA ELECTRONICA EN CONTROL Y AUTOMATISMO

TEMA:

Uso de la plataforma Arduino y Simulink para desarrollo de aplicaciones prácticas para instrumentación virtual

AUTOR:

Zambrano Alcívar, Ángel David

Trabajo de Titulación previo a la obtención del título de

INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO

TUTOR:

Romero Paz, Manuel de Jesús

Guayaquil, Ecuador

14 de Septiembre del 2017

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

INGENIERIA ELECTRONICA EN CONTROL Y AUTOMATISMO

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por el Sr. **Zambrano Alcívar, Ángel David** como requerimiento para la obtención del título de **INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO.**

TUTOR

_____

Romero Paz, Manuel de Jesús

DIRECTOR DE CARRERA

_____

Heras Sánchez, Miguel Armando

Guayaquil, a los 14 días del mes de Septiembre del año 2017

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

INGENIERIA ELECTRONICA EN CONTROL Y AUTOMATISMO

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Zambrano Alcívar, Ángel David**

DECLARÓ QUE:

El trabajo de titulación "**Uso de la plataforma Arduino y Simulink para el desarrollo de aplicaciones prácticas para instrumentación virtual**" previo a la obtención del Título de **Ingeniero Electrónico en Control y Automatismo** ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 14 días del mes de Septiembre del año 2017

EL AUTOR

_____

Zambrano Alcívar, Ángel David

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
INGENIERIA ELECTRONICA EN CONTROL Y AUTOMATISMO

AUTORIZACIÓN

Yo, **Zambrano Alcívar, Ángel David**

Autorizó a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Titulación: **Uso de la plataforma Arduino y Simulink para el desarrollo de aplicaciones prácticas para instrumentación virtual**", cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

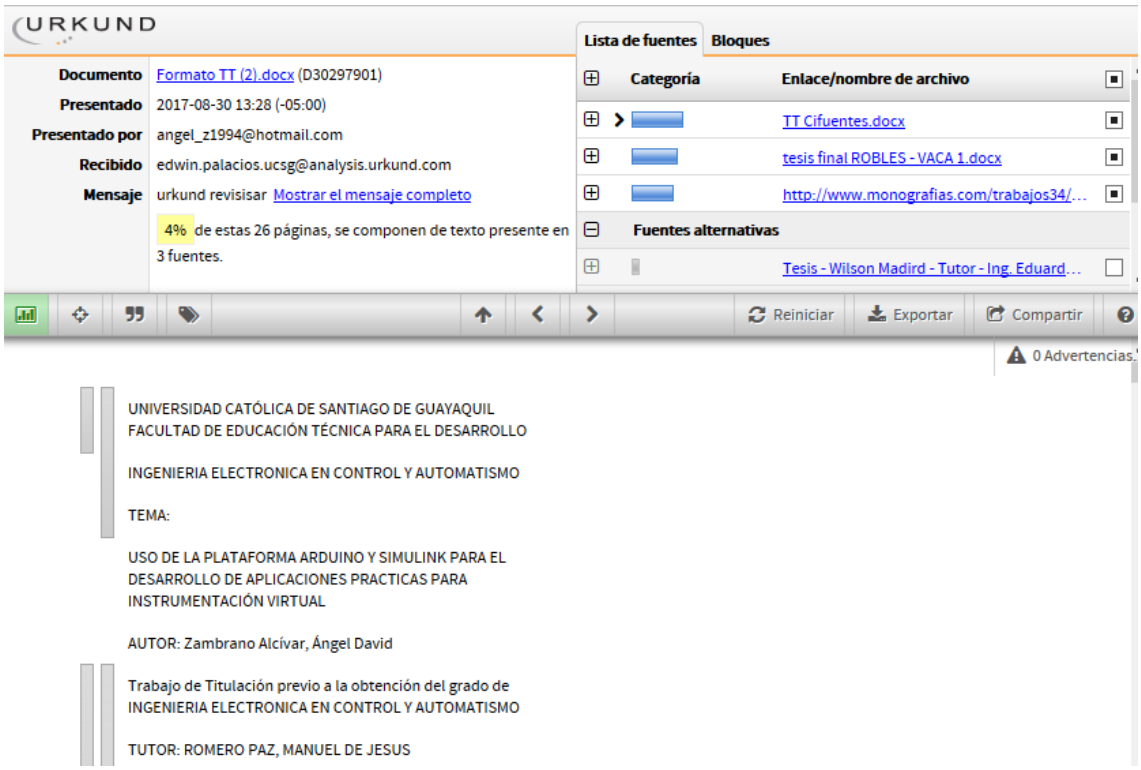
Guayaquil, a los 14 días del mes de Septiembre del año 2017

EL AUTOR

_____

Zambrano Alcívar, Ángel David

REPORTE DE URKUND



DEDICATORIA

Les dedico este trabajo a mis padres, a mi esposa e hijo porque las palabras que me brindaste no son suficiente para expresarte lo agradecido que estoy al tenerte a mi lado, me supiste demostrar que junto a las personas que están a mi lado me dan su apoyo de lucha para culminar mi carrera universitaria, la cual es un principio de mi vida a futura, agradezco tanto porque llegaste en el momento más indicado y te quedaste para siempre a mi lado.

EL AUTOR

ZAMBRANO ALCIVAR, ANGEL DAVID

AGRADECIMIENTO

En primer lugar agradezco a Dios por haberme dado la motivación y el bienestar que necesitaba para llegar a este punto de mi carrea. También agradezco a mis padres quienes me brindaron el apoyo económico y me ayudaron con consejos.

También debo agradecer a mi esposa Carolina y a mi pilar importante en mi vida que es mi hijo Ethan a los cuales me brindaron el apoyo para seguir mi carrera.

Debo también agradecer a la Coordinación Académica de la Facultad Técnica para el Desarrollo de la Educación de la UCSG por brindarme él apoya para seguir adelante en mi carrera.

EL AUTOR

ZAMBRANO ALCIVAR, ANGEL DAVID

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERIA ELECTRONICA EN CONTROL Y AUTOMATISMO

TRIBUNAL DE SUSTENTACIÓN

f. _____

ING. MIGUEL ARMANDO HERAS SANCHEZ, M. Sc.
DIRECTOR DE CARRERA

f. _____

ING. EDWIN FERNANDO PALACIOS MELÉNDEZ, M. Sc.
COORDINADOR DE ÁREA

f. _____

ING. LUIS SILVIO CÓRDOVA RIVADENEIRA, M. Sc.
OPONENTE

Índice General

Índice de Figuras

Capítulo 2

Capítulo 3

Resumen

El presente trabajo de titulación hace referencia al tema del USO DE LA PLATAFORMA ARDUINO Y SIMULINK PARA EL DESARROLLO DE APLICACIONES PRACTICAS PARA INSTRUMENTACIÓN VIRTUAL, para lo cual hemos trabajado con MatLab mediante Arduino Simulink basados en prácticas como por ejemplo Modulación PWM con entrada analógicas, encendido de led, lo que nos permite verificar la regulación del led mediante un pulso enviado desde el programa que vamos a utilizar que es Simulink.

En este proceso de titulación también hacemos referencia a los tipos de microcontroladores utilizados, el sistema que se desarrolló cuanta con los dispositivos esquemáticos para realizarlos mediantes prácticas en la UCSG.

En este trabajo de titulación a llegado experimentar muchos ámbitos y conocimientos prácticas y simulados. La metodología utilizada fue de fuentes bibliográficas y experimentales, las cuales nos permitió analizar los aspectos fundamentales y teóricos acerca al uso de la plataforma Arduino y Simulink para el desarrollo de aplicaciones de la instrumentación virtual. Se ha realizado una descripción general de los fundamentos básicos de la familia de los microcontroladores así también de la plataforma de desarrollo de Arduino IDE.

Posteriormente se desarrollaron seis aplicaciones prácticas en la que se integran el diseño de la instrumentación virtual, y su respectiva simulación.

Palabras claves: INSTRUMENTACION VIRTUAL, ARDUINO, MATLAB, SIMULINK, MICROCONTROLADORES, PIC

CAPÍTULO 1: INTRODUCCIÓN

1.1. Introducción.

La idea principal de estas actividades es practicar y aprender a controlar un Arduino con un sistema de programación diferente al usualmente usar para programar esta tarjeta. Para ellos implementaremos la comunicación del programa Simulink de MatLab que es un entorno de programación grafico que facilita un poco el proceso de ello.

En esta herramienta nos proporciona a interactuar con la materia prolongada de la instrumentación virtual Arduino y Simulink y la familia de los microcontroladores que se incorporan en la tarjeta Arduino. Este nos permite programar de una manera versátil su uso de las distintas librerías ya que es una librería de códigos abiertos.

La ingeniería con el conjunto de conocimientos científicos cuyo punto la tecnología con el fin de crear mucho más procesos a futuro para la producción, este proceso lleva a cabo el funcionamiento del control industrial por medio de la automatización y sus sistemas de productividad. También nos hace énfasis al requerimiento y aplicaciones de muchos más controladores para una practicas a futura.

1.2. Antecedentes.

En esta búsqueda de información de los proyectos se espera que se integren a los instrumentos y herramientas de simulación virtual. En los

trabajos de investigación se pudieron verificar otras fuentes en las cuales han trabajado mediante LabView, Isis Proteus, simplificando las señales de muchos microcontroladores de la familia de los PIC, y los microcontroladores ATMega. También se puede constatar en otros procesos encontrados la utilización de programas de altos niveles en los que introdujeron códigos abiertos para el desarrollo de la plataforma Arduino IDE.

1.3. Definición del Problema.

En la familia de los microcontroladores PIC, se pueden realizar prácticas asignada a la asignatura instrumentación virtual. Por lo tanto, en esto surge la necesidad de realizar aplicaciones prácticas que puedan integrar el cambio de instrumentación virtual mediante Simulink

1.4. Justificación del Problema.

En el desarrollo de los microcontroladores que se trabaja con una placa o hardware de código abierto Arduino, esto nos han permitido que el programa admita librerías de Arduino UNO a la cual contiene microcontroladores, el diseño del desarrollo nos permite la utilización para enviar información a través de la asignatura instrumentación virtual que ha sido diseñada mediante la plataforma Simulink.

1.5. Objetivos del Problema de Investigación.

1.5.1. Objetivo General.

Evaluar el uso de la plataforma Arduino y Simulink para el desarrollo de aplicaciones prácticas para instrumentación virtual.

1.5.2. Objetivos Específicos.

- Determinar los requerimientos y formar los procesos de las prácticas adquiridas mediante el simulador.
- Diseñar prototipos de tarjetas Arduino con respecto a Simulink, basado mediante los microcontroladores para la utilización de la automatización
- Evaluar los sistemas de prácticas de automatización en el uso de plataforma Arduino y Simulink, determinar el correcto funcionamiento los sistemas de control y sus posibilidades de fallo.

1.6. Hipótesis.

A través de las practicas requeridas se espera realizar las simulaciones virtuales para el programa, la cuales nos permita ver el desarrollo adecuado de las aplicaciones prácticas que orientan en la ayuda de los desarrollos de proyectos de integrados y nuevos proyectos de trabajos de titulación para la asignatura.

1.7. Metodología de Investigación.

En este proceso existen varios métodos investigativos a los cuales se pueden emplear, pero por lo tanto como cada método es diferente, esto depende de tema a tratar de investigación. En este caso este proyecto de titulación hace énfasis al método descriptivo y explicativo cuyo diseños son mostrado de forma virtual, ya que para este planteamiento se utiliza herramienta virtuales, Simulación y virtuales. En cambio en los proyectos de ingeniería los métodos más utilizados son los explicativos y exploratorios.

CAPÍTULO 2: DESCRIPCION DE ARDUINO Y MICROCONTOLADORES

2.1. Introducción a los microcontroladores.

Los microcontroladores tienen un impacto directo o indirecto en nuestra vida cotidiana. Normalmente, su presencia es desapercibida en la mayoría de los lugares como: supermercados (cajas registradoras), básculas, videojuegos, sistemas de seguridad, etc. También, en hogares tales como, hornos, lavadoras, despertadores, impresoras a inyección y laser, etc.

En otras aplicaciones, tales como, juguetes, teléfonos celulares, control climático, fax, instrumentos musicales, equipos estéreos, etc. Inclusive existen aplicaciones para oficinas, tales como, fotocopiadoras, elevadores, control de la transmisión, etc. Inclusive en procesos de automatización industrial, sistemas de seguridad, máquina de coser, videocámara, señales de tráfico, plataformas ferroviarias, entre otras aplicaciones. ¿Qué es lo que hace que estas máquinas sean inteligentes? La respuesta es el microcontrolador.

Crear aplicaciones para los microcontroladores es diferente a cualquier otro trabajo de desarrollo en electrónica e informática. Antes de Seleccionar un dispositivo en particular para una aplicación, es importante entender cuáles son las diferentes opciones y características y qué pueden significar con respecto al desarrollo de la aplicación. El microcontrolador es un componente muy común en sistemas electrónicos modernos. Su uso es tan extendido que es casi imposible trabajar en electrónica sin cruzarla.

Los microcontroladores se utilizan en un gran número de sistemas electrónicos tales como: Sistemas de gestión de motores en automóviles. Teclado de un PC. Instrumentos de medición electrónicos (como multímetros digitales, sintetizadores de frecuencia y osciloscopios) Impresoras. Teléfonos móviles. Televisores, radios, reproductores de CD, equipo de grabación de cinta. Audífonos. Sistemas de alarma de seguridad, sistemas de alarma contra incendios y sistemas de servicios de construcción. Uno podría seguir agregando a esta lista, pero el lector debe ahora tener la impresión de que hay un amplio margen para el uso de microcontroladores.

2.1.1. Que es un Microcontrolador

Un microcontrolador es una computadora pequeña y de bajo costo construida con el propósito de tratar con tareas específicas, como mostrar información sobre la visualización de siete segmentos en la plataforma ferroviaria o recibir información desde el mando a distancia de un televisor controlado. Los microcontroladores se utilizan principalmente en productos que requieren un grado de control para ser ejercido por el usuario.

Hoy varios tipos de microcontroladores están disponibles en el mercado con diferentes longitudes de palabras como 8 bits, 16 bits, 32 bits y microcontroladores. El microcontrolador es un microordenador comprimido fabricado para controlar las funciones de sistemas en máquinas de oficina, robots, electrodomésticos, vehículos de motor, y una serie de otros segmentos. Por lo tanto, en la actualidad el mundo tecnológico de las cosas está hecha con la ayuda del microcontrolador. Dependiendo de las

aplicaciones que tenemos que elegir determinados tipos de microcontroladores.

Un microprocesador incorpora las funciones de una CPU en un solo circuito integrado o unos pocos circuitos integrados. Es un procesador de computadora en un microchip y es un dispositivo programable multipropósito que utiliza datos digitales como entrada y proporciona resultados como una salida una vez que procesa la entrada según instrucciones almacenadas en su memoria. Los microprocesadores utilizan lógica digital secuencial ya que tienen memoria interna y operan sobre números y símbolos representados en el sistema numérico binario. Están diseñados para realizar operaciones aritméticas y lógicas que hacen uso de datos en el chip. Los microprocesadores de uso general en PC se utilizan para la visualización multimedia, computación, edición de texto y comunicación. Varios microprocesadores son parte de sistemas embebidos. Estos microprocesadores integrados proporcionan control digital a varios objetos incluyendo aparatos, automóviles, teléfonos móviles y control de procesos industriales.

En la figura 2.1 se muestra en detalle los componentes incorporados en la mayoría de microcontroladores, en este caso nos referimos a la familia de microcontroladores PIC. Se observa los componentes tales como, oscilador incorporado, convertidor A/D, microprocesador, memoria de programa y RAM.

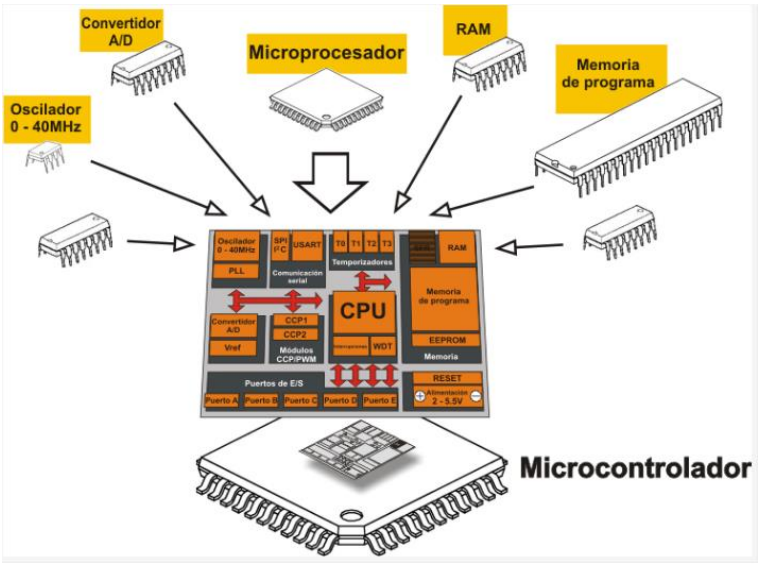


Figura 2. 1: Microcontroladores PIC.
Fuente: (Barrett & Pack, 2006)

2.1.2. Como está clasificado un microprocesador

Los microcontroladores se caracterizan por el ancho del bus, el conjunto de instrucciones y la estructura de memoria. Para la misma familia, pueden ser diferentes formas con diferentes fuentes. Este documento también va a describir algunos de los tipos básicos de la Microcontrolador que los usuarios más recientes no conocen.

- Clasificación según número de bits

a. Microcontrolador de 8 bits: Significa CPU o ALU puede procesar datos de 8 bits a la vez. Los ejemplos de microcontroladores de 8 bits son Intel 8031/8051. Estos se utilizan en el control de posición, aplicaciones de control de velocidad.

b. Microcontrolador de 16 bits: Realiza una mayor precisión y rendimiento en comparación con 8 bits. Estos se desarrollan para el propósito de aplicaciones de alta velocidad como el sistema de control de servo, robótica, etc. Algunos ejemplos de microcontrolador de 16

bits. Los microcontroladores Intel 8096 y Motorola MC68HC12 son familias de 16 bits extendidas.

c. Microcontrolador de 32 bits: Utiliza las instrucciones de 32 bits para realizar las operaciones aritméticas y lógicas. Se desarrollan con el fin de la aplicación de muy alta velocidad en procesamiento de imagen, telecomunicaciones, sistema de control inteligente, etc. Algunos ejemplos son la familia Intel/Atmel 251, PIC3x, ARM

2.1.3. Clasificación según dispositivos de memoria

1.) Microcontrolador de memoria incorporado: Cuando un sistema embebido tiene una unidad de microcontrolador que tiene todas las funciones Bloques disponibles en un chip se llama un microcontrolador integrado. Por ejemplo, 8051 que tiene memoria de programa y datos, E/S Puertos, comunicación en serie, contadores y temporizadores e interrupciones en el chip es un microcontrolador integrado.

2.) Microcontrolador de memoria externa: Cuando un sistema embebido tiene una unidad de microcontrolador que no tiene todas las funciones Bloques disponibles en un chip se llama un microcontrolador de memoria externa. Por ejemplo, 8031 no tiene memoria de programa en el Chip es un microcontrolador de memoria externa.

2.1.4. Clasificación según el conjunto de instrucciones

1.) Arquitectura CISC: CISC significa conjunto de instrucciones complejas, en la figura 2.2 nos permite al usuario aplicar 1 instrucción como alternativa a muchas instrucciones simples.

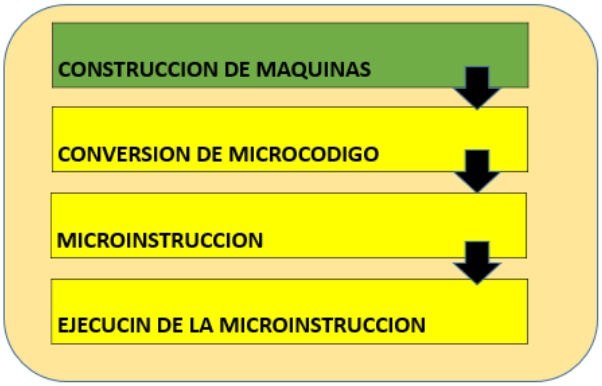


Figura 2. 2: Arquitectura CISC.
Fuente: (Khadse, Gawai, & Faruk, 2014)

2.) Arquitectura RISC: La arquitectura RISC significa a los Ordenadores de Conjunto de Instrucciones Reducidas. RISC reduce el tiempo de operación al acortar la Ciclo de reloj por instrucción. El RISC da una mejor ejecución que el CISC

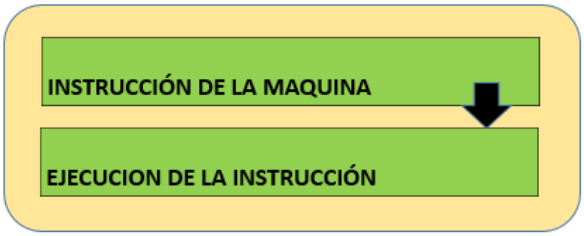


Figura 2. 3: Arquitectura RISC
Fuente: (Khadse et al., 2014)

2.1.5. Clasificación según arquitectura de memoria

1.) El punto cuando una unidad de microcontrolador tiene una dirección de memoria diferente espacio para el programa y memoria de datos, el microcontrolador tiene arquitectura de memoria Harvard en el procesador. El RISC da una mejor ejecución que la CISC.

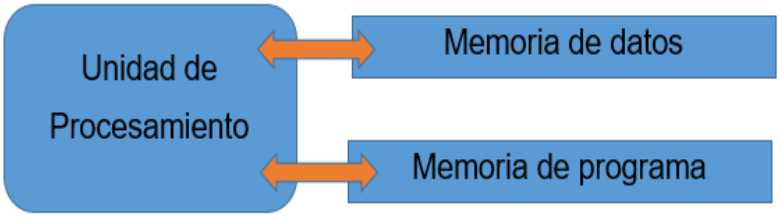


Figura 2. 4: Arquitectura hardware
Fuente: (Khadse et al., 2014)

2.) El punto cuando un microcontrolador tiene una dirección de memoria común para la memoria del programa y la memoria de datos, el microcontrolador tiene arquitectura de memoria primero en el proceso.

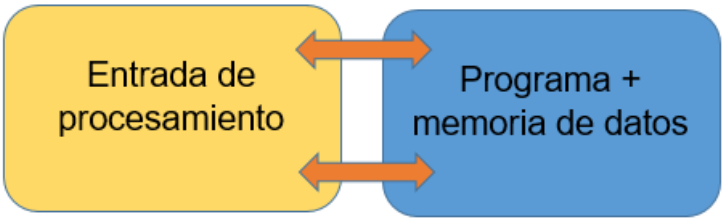


Figura 2. 5: Principio de la arquitectura.
Fuente: (Khadse et al., 2014)

2.2. Tipos de microcontroladores

2.2.1. Microcontroladores 8051.

El microcontrolador 8051 es un microcontrolador de ocho bits inventado en 1981 por Intel Corporación. Está disponible en DIP de 40 pines, como se muestra en la figura 2.4 Es un uC básico pero todavía muchas compañías lo siguen fabricando. Los tipos más antiguos de 8051 tienen 12 relojes por instrucción que lo hacen lento. Mientras que el 8051 reciente tiene 6 relojes por instrucción. El microcontrolador 8051 no tiene un bus de memoria incorporado y tampoco convertidores A/D y tales microcontroladores usan procesadores CISC, y la arquitectura Von Neuman.

Varias características del microcontrolador 8051 se dan como sigue.

1) CPU de 8 bits

2) Contador de programa de 16 bits

3) Palabra de estado de procesador de 8 bits (PSW)

4) Puntero de pila de 8 bits

5) ROM interna de 4K bytes (memoria de programa).

6) RAM interna de 128 bytes (memoria de datos).

7) Registros de función especiales (SFR) de 128 bytes

8) 32 pines de E / S dispuestos como cuatro puertos de 8 bits (P0 - P3)

9) Dos temporizadores / contadores de 16 bits: T0 y T1

10) Interrupciones vectorizadas

11) Una E / S serie full dúplex

Figura 2. 6: microcontrolador 8051.
Fuente: (Khadse et al., 2014)

2.2.2. Controlador de interfaz periférica.

Es una familia de microcontroladores de la tecnología de Microchip EE.UU. con la arquitectura de Havard. Originalmente se desarrolló como dispositivo de soporte para los ordenadores PDP (procesador de datos de programa) para soportar sus dispositivos periféricos y por lo tanto se denominó como PIC. Los microcontroladores PIC son procesadores RISC. En la figura 2.7 nos permite apreciar el microchip de la familia de los 16F877. Una cosa interesante sobre PIC es que su ciclo de máquina consta de sólo 4 pulsos de reloj en contraste con 12 pulsos de reloj en Intel 8051 Microcontrolador. PIC microcontroladores están encontrando su camino en

nuevas aplicaciones como teléfonos inteligentes, accesorios de audio, periféricos de videojuegos y dispositivos médicos avanzados.

Figura 2. 7: PIC 16F877
Fuente: (Khadse et al., 2014)

Varias características del microcontrolador PIC 16F877 se dan de la siguiente manera

- 1 CPU RISC de alto rendimiento
- Hasta 8K x 14 palabras de memoria de programa FLASH
- 35 instrucciones (longitud fija de codificación de 14 bits)
- Memoria de datos basada en RAM estática 368 × 8
- Hasta 256 x 8 bytes De memoria de datos de la EEPROM
- Capacidad de interrupción (hasta 14 fuentes)
- Tres modos de direccionamiento (directo, indirecto, relativo)
- Reinicio de encendido (POR)
- Memoria de arquitectura de Harvard
- Modo ahorro de energía SLEEP
- Voltaje: 2.0V a 5.5V
- Alta fuente del fregadero / fuente: 25mA
- Máquina basada acumulador

2.2.3. Microcontrolador ARM

ARM es un microcontrolador de 32 bits cuyo núcleo está diseñado por ARM Limited con arquitectura RISC. ARM tiene la arquitectura von Neumann

(programa y RAM en el mismo espacio). Los microcontroladores ARM son extremadamente utilizados en ahorro de energía y funcionan en muy bajo consumo de energía.

ARM Microcontrollers Widely utilizado en el teléfono moderno para las comunicaciones móviles. Estos también se utilizan en varios sistemas incorporados sistema le gusta iPod, unidad de juego de mano, controlador de disco y así sucesivamente. 8051 y PIC necesitan varios ciclos de reloj por instrucción. AVR y ARM ejecutan la mayoría de las instrucciones en un solo ciclo de reloj.

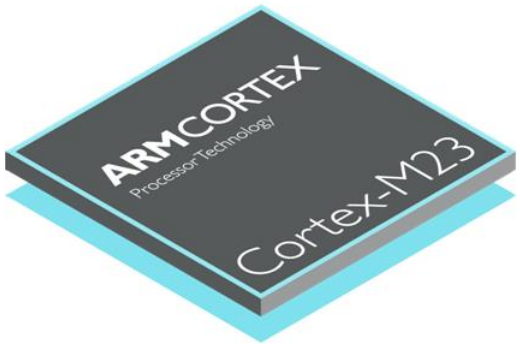


Figura 2. 8: Microcontrolador de la familia ARM.
Fuente: (Khadse et al., 2014)

Varias características del microcontrolador ARM se dan de la siguiente manera

1) Máximo funcionamiento de ciclo único

2) Archivo de registro de 16 x 32 bits constante.

3) Cargar o almacenar la arquitectura.

4) Ancho de instrucción preestablecido de 32 bits para simplificar el revestimiento de tuberías y la decodificación, a una densidad de código minimizada

5) Para el acceso desalineado de memoria no hay soporte

2.2.4. Microcontrolador AVR

El AVR es un modificado Harvard RISC arquitectura de 8 bits RISC chip único microcontrolador, que es desarrollado por Atmel en 1996. El AVR es para Alf-Egil Bogen y Vegard Wollan del procesador RISC. AVR toma sólo un reloj por instrucción. Los microcontroladores AVR se clasifican en tres tipos

A) TinyAVR - Menor memoria, tamaño pequeño, adecuado sólo para aplicaciones más sencillas.

B) MegaAVR - Estos son los más populares con buena cantidad de memoria (hasta 256 KB), mayor número de periféricos incorporados y aptos para aplicaciones de moderadas a complejas.

C) XmegaAVR - Se utiliza comercialmente para aplicaciones complejas, que requieren gran memoria de programa y alta velocidad

Figura 2. 9: Microcontrolador de la familia ARV ATMEGA 16.
Fuente: (Khadse et al., 2014)

Algunas de las características de Atmega16 son:

A) 16KB de memoria flash

b) 1KB de SRAM

c) 512 bytes de EEPROM

d) Disponible en DIP de 40 pines

e) ADC de 10 bits de 8 canales

f) Dos temporizadores / contadores de 8 bits

g) Un temporizador de 16 bits / Contador

h) 4 Canales PWM

i) En el Programador del Sistema (ISP)

j) USART en Serie

k) Interfaz SPI

l) Comparador Digital a Analógico

2.3. Microcontroladores «PIC».

Si sólo se pudiera determinar un modelo de micro controlador, éste debería tener muy potenciados todos sus recursos para poderse adaptar a las exigencias de las diferentes aplicaciones. Esta potenciación supondría en muchos casos un despilfarro. En la práctica cada fabricante de microcontroladores oferta un elevado número de modelos diferentes, desde los más sencillos hasta los más poderosos. Es posible seleccionar la capacidad de las memorias, el número de líneas de E/S, la cantidad y potencia de los elementos auxiliares, la velocidad de funcionamiento, etc. Por todo ello, un aspecto muy destacado del diseño es la selección del microcontrolador a utilizar.

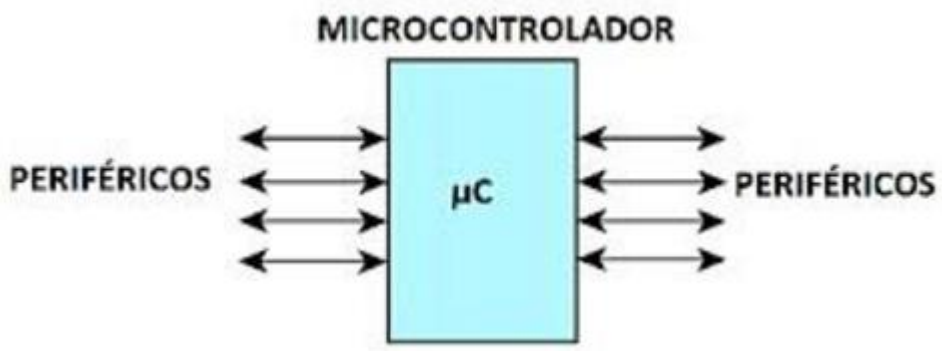


Figura 2. 10: Aplicación de micro controlador
Fuente:(Angulo usategui & Angulo Martinez, 2003)

2.3.1. Arquitectura Interna

Un micro controlador posee todos los componentes de un computador, pero con unas características fijas que no pueden alterarse. Las partes principales de un microcontrolador son

Procesador:

Memoria no volátil para contener el programa

Memoria de lectura y escritura para guardar los datos

Líneas de EIS para los controladores de periféricos:

a) Comunicación paralelo

b) Comunicación serie

c) Diversas puertas de comunicación (bus l2ºC, USB, etc.)

Recursos auxiliares:

a) Circuito de reloj

b) Temporizadores

c) Perro Guardián («watchdog»)

d) Conversores AD y DA

e) Comparadores analógicos

f) Protección ante fallos de la alimentación

g) Estado de reposo o de bajo consumo

2.3.2. Gama baja o básica: PIC16C5X con instrucciones de 12 bits

Se trata de una serie de PIC de recursos limitados, pero con una de las mejores relaciones coste/prestaciones. Sus versiones están encapsuladas con 18 y 28 pines o puertos y pueden alimentarse a partir de una tensión de 2,5 V lo que les hace ideales en las aplicaciones que funcionan con pilas.

Tienen un repertorio de 33 instrucciones cuyo formato consta de 12 bits. No admiten ningún tipo de interrupción y la pila sólo dispone de dos niveles.

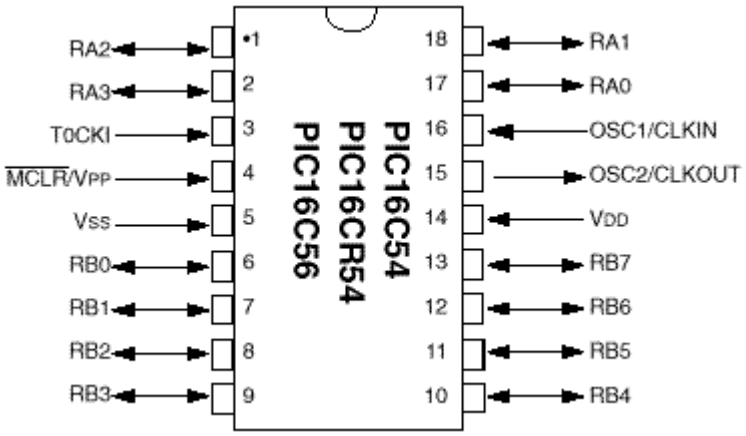


Figura 2. 11: Diagrama de pines o puertos del uC PIC de 12 bits
Fuente: (Angulo usategui & Angulo Martinez, 2003)

2.3.3. Gama media: PIC16CXXX con instrucciones de 14 bits

Es la gama más variada y completa de los PIC. Abarca modelos con encapsulado desde 18 patitas hasta 68, cubriendo varias opciones que integran abundantes periféricos. Dentro de esta gama se halla el «fabuloso PIC 16F84». El repertorio de instrucciones es de 35 a 14 bits cada una y compatible con el de la gama baja. Sus distintos modelos contienen todos los recursos que se precisan en las aplicaciones de los microcontroladores de 8 bits. También dispone de interrupciones y una Pila de 8 niveles que permite el anidamiento de subrutinas. La gama media puede clasificarse en las siguientes subfamilias:

- Gama media estándar (PIC16C55X)
- Gama media con comparador analógico (PIC16C62X/64X/66X)
- Gama media con módulo de captura (CCP), modulación de anchura de impulsos (PWM) y puerta serie (PIC16C6X)
- Gama media con CAD de 8 bits (PIC16C7X)

- Gama media con CAD de precisión (PIC14000)
- Gama media con memoria Flash y EEPROM (PIC16F87X y PIC16X8X)
- Gama media con driver LCD (PIC16C92X

Dentro de la gama media también se halla la versión PIC14C000, que soporta el diseño de controladores inteligentes para cargadores de baterías, pilas pequeñas, fuentes de alimentación ininterrumpidas y cualquier sistema de adquisición y procesamiento de señales que requiera gestión de la energía de alimentación.

2.3.4. Gama alta: PIC17CXXX con instrucciones de 16 bits

Se alcanzan las 58 instrucciones de 16 bits en el repertorio y sus modelos disponen de un sistema de gestión de interrupciones vectores muy potente. También incluyen variados controladores de periféricos, puertas de comunicación serie y paralelo con elementos externos y un multiplicador hardware de gran velocidad. Quizás la característica más destacable de los componentes de esta gama es su arquitectura abierta, que consiste en la posibilidad de ampliación del microcontrolador con elementos externos. Para este fin, las patitas sacan al exterior las líneas de los buses de datos, direcciones y

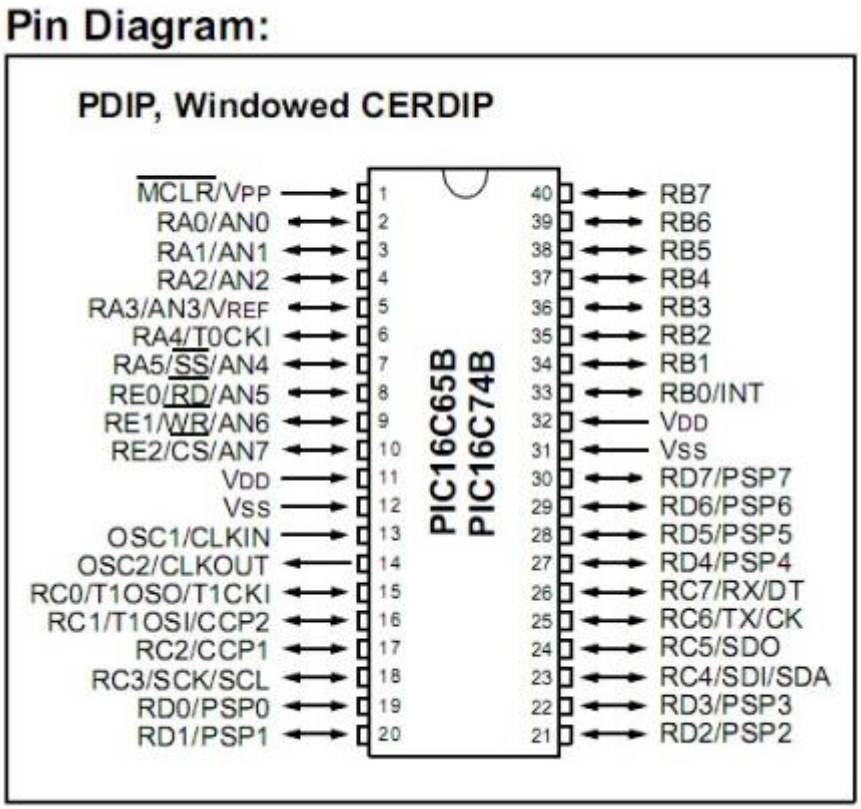


Figura 2. 12: Diagrama de pines o puertos del uC PIC de 16 bits
Fuente:(Angulo usategui & Angulo Martinez, 2003)

Control, a las que se conectan memorias o controladores de periféricos. Esta filosofía de construcción del sistema es la que se empleaba en los microprocesadores y no suele ser una práctica habitual cuando se emplean micro controlador.

2.4 Que es Arduino

Arduino, según Massimo Banzi, uno de sus creadores, es una plataforma de computación física de código abierto basada en un simple tablero de E/S y un entorno de desarrollo que implementa el lenguaje de procesamiento. Personalmente, se da por aceptada la posición de Richard Stallman y Free Software Fundación sobre cómo nombrar un software que respeta las libertades del usuario para ejecutar, estudiar, cambiar y distribuir

el programa de software original y modificado por el usuario. Así que por lo general prefiero referirme a dicho software, en lugar de usar la denominación de Software Libre Aburrido, con el término Software Libre o Software Libre para que el lector pueda comprender claramente que se da importancia a las libertades, no sólo al acceso del código fuente.

Teniendo en cuenta las consideraciones hechas para el software y las trasladar al mundo del hardware, prefiero usar el término libre hardware en lugar de Open Source Hardware para que quede claro que estamos más preocupados por las libertades dadas al usuario usando libre hardware en lugar del acceso abierto a los diseños de hardware.

También me gustaría señalar el hecho de que la mayor parte del éxito de Arduino se debe a una emocionante comunidad de desarrolladores, hackers, aficionados que aportan código, documentos guías en el arduino.cc y otros sitios web. Por lo tanto, en mi opinión, una mejor definición de Arduino sería: un hardware físico libre basada en una simple tarjeta de entrada / salida (E / S), un desarrollo en

Que implementa el lenguaje de procesamiento y una comunidad de usuarios que comparten sus conocimientos y conocimientos en sus proyectos basados en Arduino.

2.4.1. ¿Por qué Arduino

Hay muchas plataformas de prototipos de hardware disponibles pero Arduino es una buena opción como

- Es un proyecto de hardware y software libre, por lo que tanto el software como el hardware son extremadamente accesibles y muy flexibles y pueden personalizarse fácilmente y extendido
- Es flexible, ofrece varias entradas digitales y analógicas, SPI, I2C, una interfaz serie y salidas digitales y PWM
- Es fácil de usar, se conecta a un ordenador a través de USB y se comunica utilizando el protocolo serie estándar, se ejecuta en modo autónomo y como una interfaz conectada a computadoras PC / Macintosh
- Es barato, menos de 30 euros por tabla y viene con un entorno de desarrollo libre
- Está respaldado por una creciente comunidad en línea, un montón de código fuente ya está disponible y listo para ser utilizado

2.4.2. Que podemos hacer con Arduino

Arduino es una gran herramienta para desarrollar objetos interactivos, tomando entradas de una variedad de interruptores o sensores y controlando una variedad de luces, motores y otras salidas. Los proyectos de Arduino pueden ser independientes o pueden conectarse a una computadora mediante USB. El Arduino será visto por el ordenador como una interfaz serie estándar existen API de comunicación en serie en la mayoría de los lenguajes de programación, por lo que la interfaz de Arduino con un programa de software que se ejecuta en el equipo es bastante sencillo.

2.4.3. Arduino Ardware.

El tablero de Arduino es un tablero del micro controlador, que es un pequeño circuito (el tablero) que contiene una computadora entera en un pequeño microprocesador (el micro controlador). Existen diferentes versiones de la placa Arduino: son diferentes en componentes, puntería y tamaño, etc. Algunos ejemplos de placas Arduino son: Arduino Duemilanove / UNO, Arduino Mega, Arduino Nano, Arduino Mini. Los diagramas esquemáticos de Arduino se distribuyen utilizando una licencia abierta para que cualquiera pueda crear su propia tarjeta compatible con Arduino. El nombre de Arduino es una marca registrada por lo que no es posible llamar a una tarjeta clonada Arduino: por eso es muy común encontrar referencias en tablas duino como Seeeduino, FreeDuino, Japónico, Zigduino, iDuino, etc.

Figura 2. 13: Componentes de arduinos hardware.
Fuente: (Varesano, 2011)

2.4.4. Arduino Shields

- Arduino pueden ampliarse usando escudos, PCB diseñados ad hoc
- Teniendo el mismo diseño de pines de Arduino, que puede ser apilado por encima de la adición de ad
- Funcionalidades tradicionales. La figura muestra un ejemplo bastante extremo de escudo Arduino

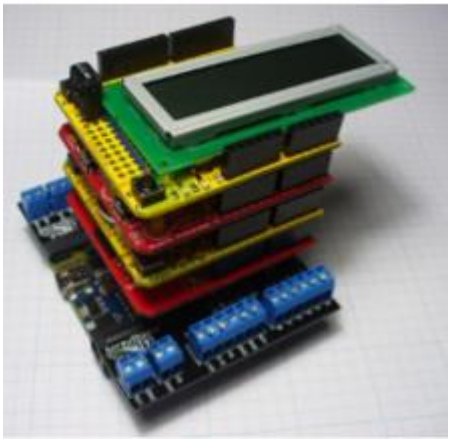


Figura 2. 14: Escudos o Shields de la familia Arduino.
Fuente: (Varesano, 2011)

Hay una gran cantidad de escudos disponibles, cada uno de ellos especialmente diseñado para una aplicación. Algunos están siendo desarrollados por el equipo de Arduino, mientras que la mayoría de ellos han sido desarrollados por terceras compañías o profesionales aficionados. Existen escudos para el control de motores DC y AC, comunicación Ethernet, reproducción MP3, salida de vídeo analógica, pantallas LCD y LED, etc. La idea es que usando escudos de Arduino, es posible agregar una característica específica a Arduino sin la molestia de desarrollar un circuito ad-hoc o PCB tratando de implementar tal característica. Por otra parte, algunos escudos son fáciles de utilizar debido a las bibliotecas disponibles y que permiten el desarrollo de aplicaciones sencillas.

2.4.5. Arduino Duemilanove

Por lo tanto el tablero de Arduino Duemilanove que es proporcional, según los desarrolladores de Arduino, "el más simple de usar y el mejor para aprender.

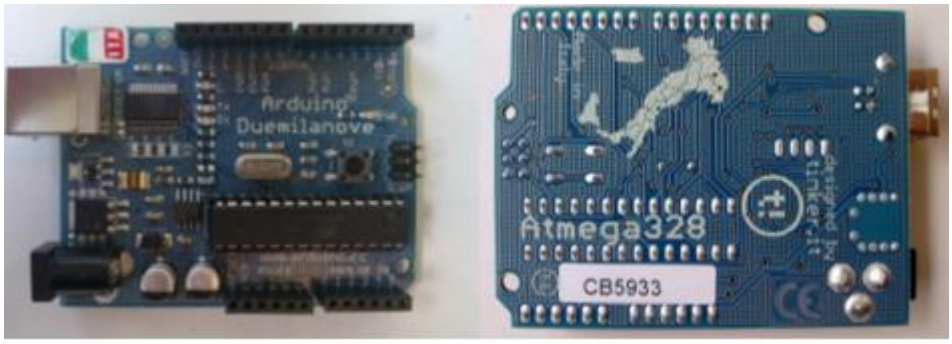


Figura 2. 15: Placa microncontroladora de Arduino Duemilanove.
Fuente: (Varesano, 2011)

2.4.5.1. Arduino Duemilanove componentes internos

Echemos un vistazo a lo que hay dentro de un Arduino Duemilanove. En la figura 2.16 se enumeran los componentes internos más importantes del tablero y que se describen a continuación:

Figura 2. 16: Componentes principales de la tarjeta Arduino Duemilanove.
Fuente:(Varesano, 2011)

1. Chip FTDI: este es el componente que permite al Arduino comunicarse con el ordenador a través de USB. El micro controlador Arduino sólo es capaz de comunicación serie. El chip FTDI convierte las señales serie en USB y viceversa. También tiene un regulador de voltaje interno que convierte la potencia de 5 V procedente del USB a 3.3 V

2. LED de estado. Está conectado al pin 13 con una resistencia de 1K Ω. Cada vez que se aplica un voltaje por el microcontrolador al pin 13, el LED se encenderá
3. TX serie y LED RX. Sirven como indicadores de una comunicación con el PC u otro dispositivo serie
4. Cristal de 16 MHz. Este es el componente que actúa como fuente de reloj para el micro controlador. Básicamente genera una señal On-O ff que el microcontrolador usa para cambiar su estado.
5. Botón de reinicio. Una vez presionado, el microcontrolador se restablecerá.
6. Power LED (PWR) que se encenderá cuando el Arduino esté conectado a cualquier fuente de alimentación que indique que el microcontrolador está funcionando.
7. Este es el microcontrolador, el corazón de Arduino. El Duemilanove utiliza el ATMEGA 328p de Atmel, un microcontrolador RISC basado en AVR de 8 bits que combina memoria de flash de 32 KB ISP con capacidades de lectura a escritura, EEPROM de 1 KB, SRAM de 2 KB, 23 líneas de E / Tres temporizadores / contadores flexibles con modos de comparación, interrupciones internas y externas, USART programable en serie, una interfaz serie de 2 hilos orientada por byte, puerto serie SPI, un convertidor A / D de 6 canales de 10 bits capaz de funcionar Hasta 200 KHz, temporizador de vigilancia programable con oscilador interno y cinco modos de ahorro de energía seleccionables por software. El ATMEGA 328p funciona entre 2,7-5,5 voltios.

8. Varios componentes: condensadores, diodos y reguladores de tensión. Se utilizan para estabilizar la fuente de alimentación, convertirla a la tensión correcta necesaria para el Arduino y evitar daños en los pantalones cortos.

2.4.5.2. conectores Arduino Duemilanove

Un aspecto clave de la placa Arduino es la cantidad de conectores disponibles. Estos son los componentes que permiten conectar las placas Arduino a otros componentes (sensores, resistencias, botones, etc.) para que puedan interactuar con ellos: lectura, escritura, movimiento, etc.

AREF: Pin de referencia analógica La tensión en este pin determina la tensión a la que los convertidores analógico-digitales (ADC) informarán el valor decimal 1023, que es su salida de nivel más alto. Esto significa que con este pin usted podrá cambiar el valor máximo legible por los pines Analog In: esta es una forma de cambiar la escala del analógico en los pines.

GND: tierra digital utilizada como tierra para entradas / salidas digitales

Pines TX / RX 0-1: Entrada / salida serie Estos pines se pueden utilizar para E / S digitales igual que los pines DIGITAL 2-13, pero no se pueden utilizar si se utiliza comunicación serie. Si su proyecto utiliza la comunicación en serie, es posible que desee utilizar los de comunicación serie en lugar de utilizar la interfaz USB a serie.

RESET Al poner esta línea LOW es posible reiniciar el tablero: también hay un botón para hacerlo en el tablero pero, como escudos

adicionales podrían hacer que el botón inaccesible, esto se puede utilizar para restablecer el tablero.

USB Se utiliza para cargar bocetos (programas binarios de Arduino) en la placa y para la comunicación en serie entre la placa y el ordenador. Arduino puede ser alimentado desde el puerto USB.

2.4.5.3. Arduino Base Workshop KIT

El tablero de Arduino sí mismo es bastante inútil a menos que lo tapemos a otros componentes eléctricos. Por lo general, junto con un tablero Arduino, tiendas también venden kits Arduino que contienen muchos componentes útiles para el desarrollo de circuitos basados en Arduino.

El KIT está compuesto por los siguientes componentes:

1. 1 x Junta Arduino Duemilanove
2. 1 x cable USB
3. 1 x conectores de cabezal de cabeza simple recto 2,54 40x1
4. 5 x 10K Ω Resistores 1 / 4W (marrón, negro, naranja, dorado)
5. 5 2.2K Ω Resistor 1/4 W (rojo, rojo, rojo, dorado)
6. 10 x 220 Ω Resistencias 1 / 4W (rojo, rojo, marrón, dorado)
7. 5 x 330K Ω Resistores 1 / 4W (naranja, naranja, amarillo, dorado)
8. condensador de condensador de 5 x 100nF

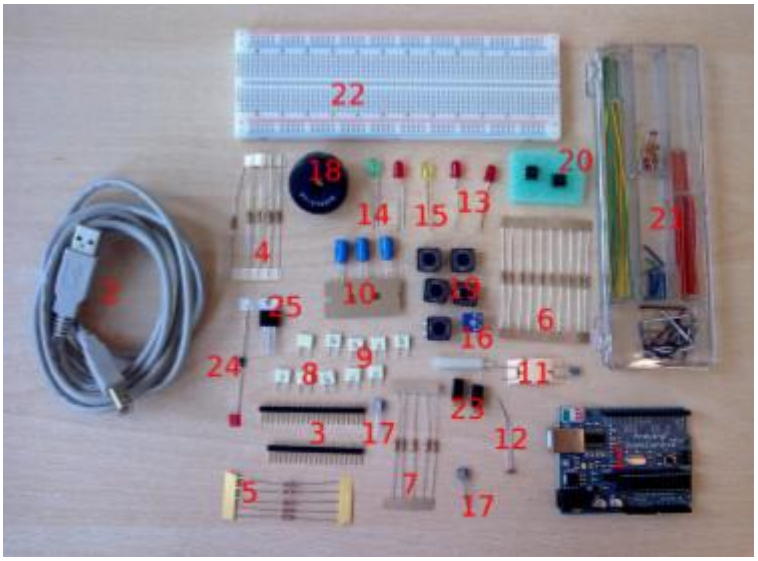


Figura 2. 17: Arduino Base Workshop KIT
Fuente: (Varesano, 2011)

9. 5 x 10nF condensador de poliéster

10. 3 x 100uF condensador electrolítico 25Vdc

11. 1 x 4,7K Ω Termistor

12. 1 x 10..40K Ω LDR VT90N2

13. LED rojo de 3 x 5 mm

14. LED VERDE de 1 x 5mm

15. 1 x 5mm LED AMARILLO

16. Potenciómetro lineal 1 x 10K Ω, terminales PCB

17. 2 x BC547 Transistor en Paquete TO92

18. 1 x Piezo zumbador

19. Botón de 5 x PCB, tamaño 12x12mm

20. 2 x 4N35 Optoacoplador DIL-6 paquete

21. 1 x Juego de 70 cables de puente

22. 1 x tablero de pan, 840 puntos de empate

23. 2 x Sensor de inclinación

24. 1 x diodo 1n4007 25. 1 x MOS Irf540

2.3.4 Arduino software

El otro componente de la plataforma Arduino es el IDE Arduino. Contiene todo el software que ejecutará una computadora para programar y comunicarse con una placa Arduino

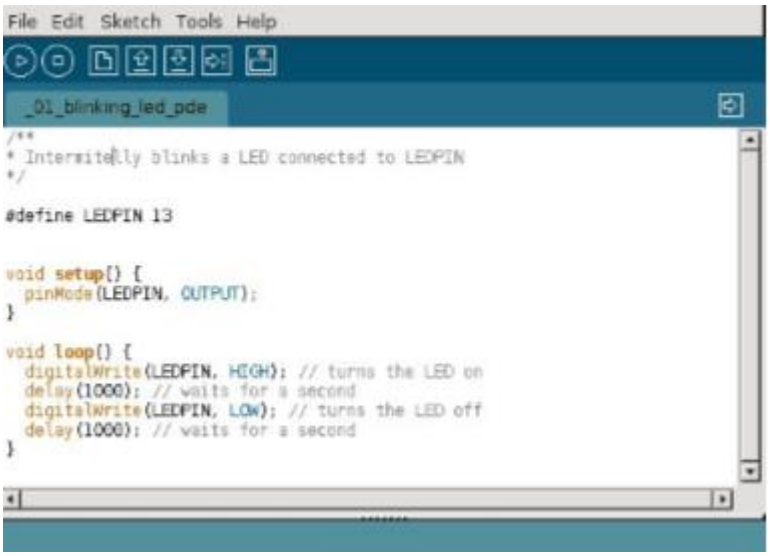


Figura 2. 18: Arduino IDE
Fuente: (Varesano, 2011)

El IDE de Arduino contiene un editor que podemos usar para escribir bocetos (ese es el nombre de los programas Arduino) en un lenguaje de programación simple modelado después del procesamiento. Utilizando el IDE, el programa que escribimos se convierte en un programa C y luego se compila Utilizando avr-gcc, un compilador libre, libre y de código abierto basado en el Gnu C Compiler (gcc) Especialmente diseñado para micro controladores AVR. Este proceso produce código binario que el microcontrolador en el tablero de Arduino podrá entender y ejecutar.

2.3.4.1 comunidades Arduino

Como muchos otros proyectos de software libre y hardware, lo que hace que Arduino sea La comunidad que lo rodea. El número de usuarios

que cotidianamente colaboran y comparten a través del Arduino. cc sitio web principal es enorme. El sitio que contiene Arduino es un editor públicamente, donde la gente puede pedir ayuda en sus proyectos o discutir sobre cualquier cosa relacionada a Arduino y prototipos de electrónica.

2.3.4.2 Criticas a Arduino

La plataforma de Arduino ha sido criticada en algunos aspectos y creo que vale la pena señalar lo que esos críticos son en aras de la transparencia debido a que hay un espaciado adicional de 0.06 "entre los conectores de clavijas digitales, no es posible conectar el Arduino directamente a una placa de procesamiento que tenga conectores espaciados de 0.1". Por la misma razón, no es posible utilizar prototipos estándar de perfboards con Arduino.

Los desarrolladores de Arduino justificaron esto como un diseño sencillo que afectó a las primeras versiones de Arduino. Sin embargo, como ya había escudos disponibles para ello, decidieron mantener el error de diseño para la compatibilidad con versiones anteriores sin embargo, el enorme éxito de Arduino y los grandes proyectos que la gente está haciendo con él, demuestran que, incluso con sus limitaciones, Arduino puede ser una plataforma de prototipos muy buena.

CAPÍTULO 3: SIMULACION Y RESULTADOS OBTENIDOS

3.1. Aplicación de practica #1; Encendiendo el Led

Objetivos

- Familiarización con el software Simulink y con nuestra placa Arduino
- Aprender conexiones con Arduino y control de salida de una señal de pulso por un pin digital

Materiales

- Led
- Resistencia de 1kohm
- Cables
- Arduino mega

Desarrollo

La práctica consta en el encendido de un led por medio de una señal de pulsos enviada desde el programa creado en Simulink

El diagrama de conexiones es el siguiente:

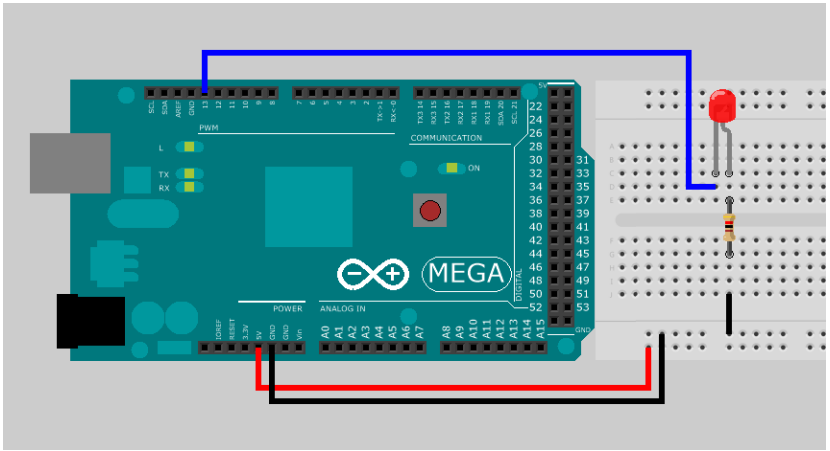


Figura 3. 1: Encendido de LED
Elaborado por: Autor.

Utilizaremos la alimentación de 5V y gnd de nuestra placa esto realizaremos en la mayoría de prácticas que se realizaran, el led se lo conectara al pin 13 esta es una salida digital en nuestra placa.

Programación

Procedemos a abrir Simulink en MatLab: (véase la figura 3.2).

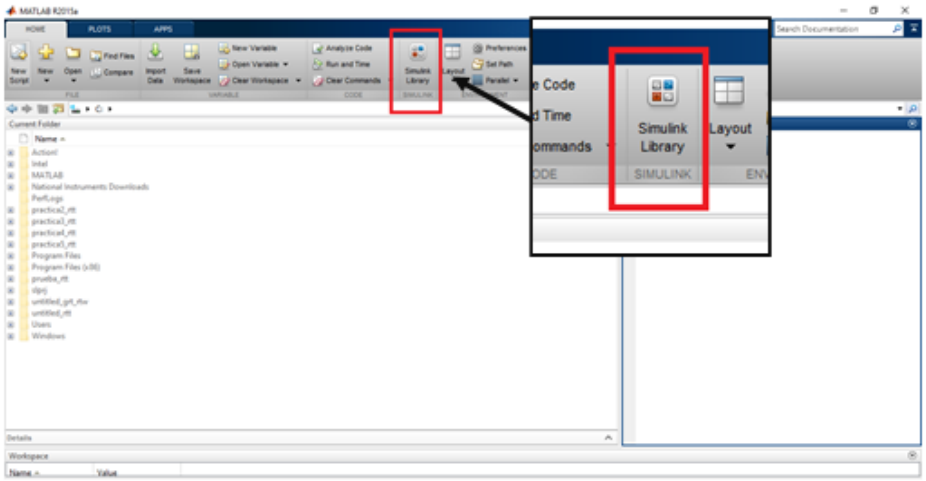


Figura 3. 2: Programacion mediante MATLAB
Elaborado por: Autor.

Ya abierta la librería de Simulink creamos un nuevo modelo. (Véase la figura 3.3).

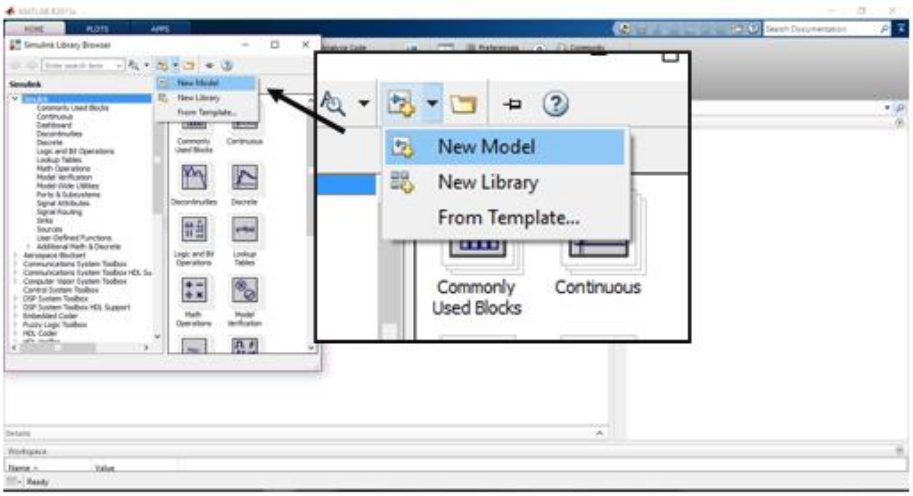


Figura 3. 3: Muestra de programa de Simulink
Elaborado por: Autor.

Creado nuestro nuevo elemento agregamos a este los bloques que utilizaremos:

- De la librería de Arduino agregamos un digital output pin. (véase la figura 3.4).

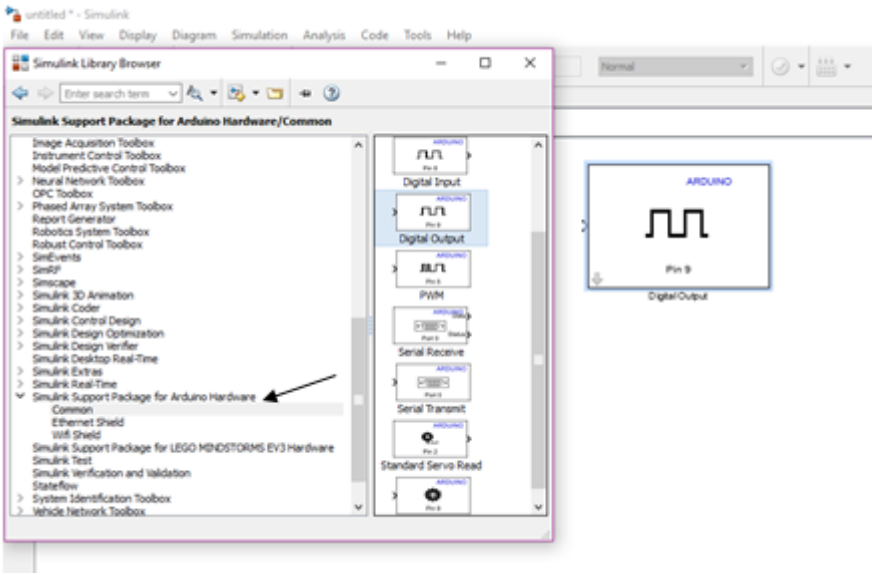


Figura 3. 4: Funcion de digital output
Elaborado por: Autor.

- De la librería sources agregamos un bloque de pulse generador (véase la figura 3.5).

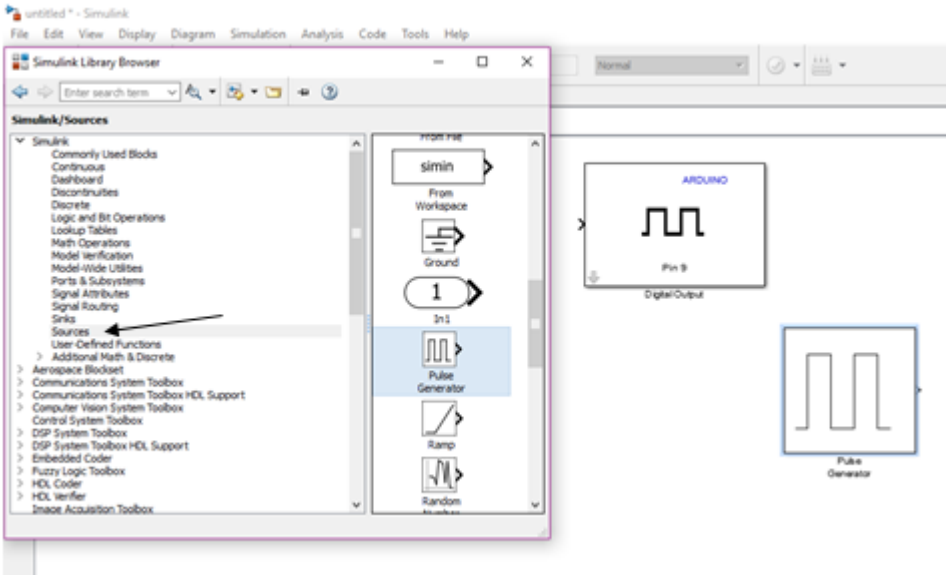


Figura 3. 5: Pulsor de bloque generator
Elaborado por: Autor.

- De la librería commonly used block agregamos un scope. (véase la figura 3.6).

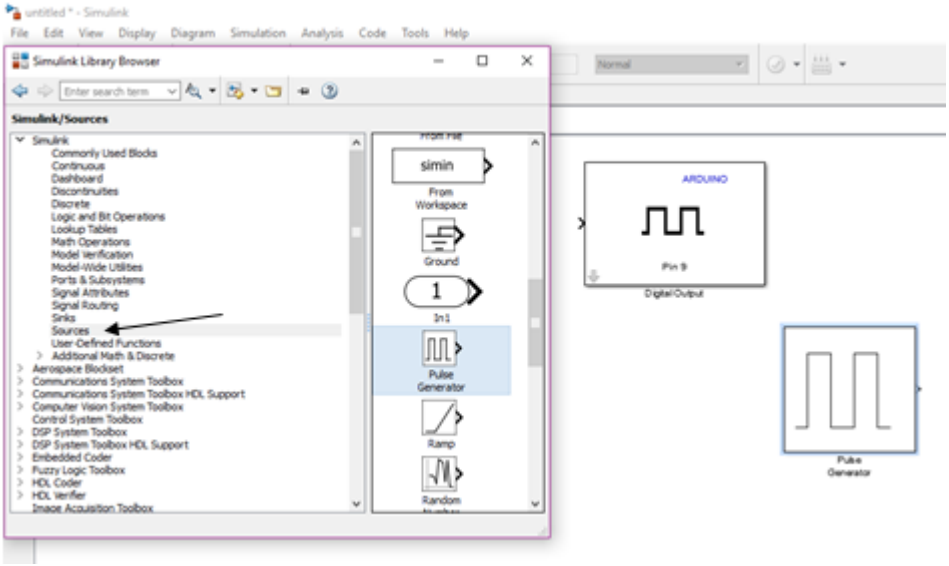


Figura 3. 6: Librería commonly used
Elaborado por: Autor.

- De la librería commonly used block agregamos un scope. (véase la figura 3.7).

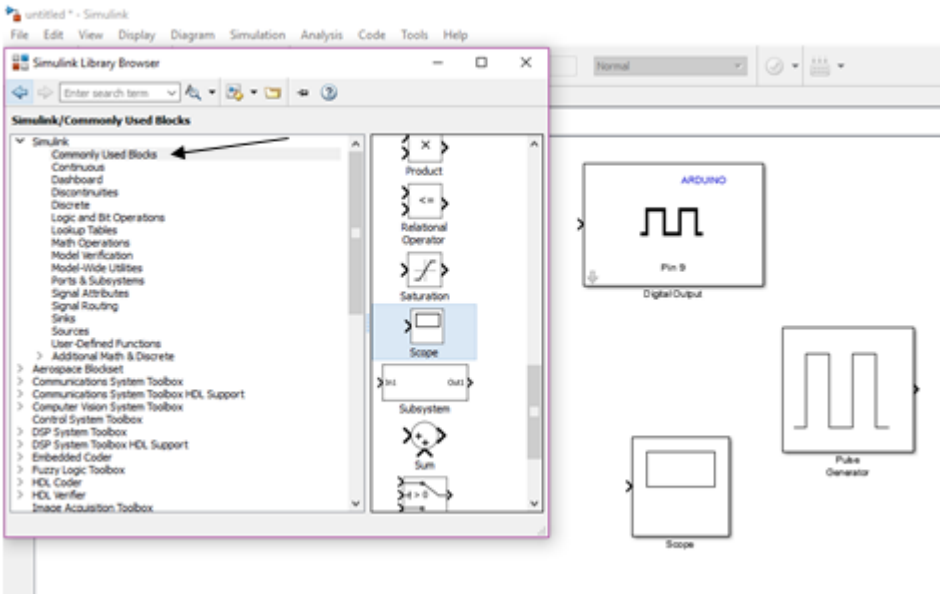


Figura 3. 7: Aplicación de scope
Elaborado por: Autor

- Ya agregados los bloques se procede a hacer las conexiones. (véase la figura 3.8).

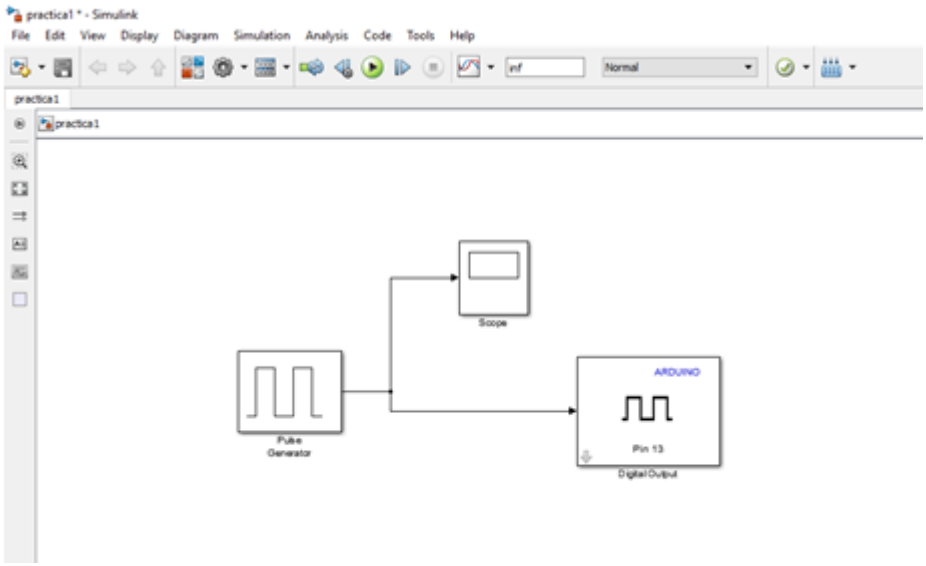


Figura 3. 8: Aplicación de conexión
Elaborado por: Autor.

Ahora configuraremos la fuente y la salida digital en esta se colocará el número del puerto en la placa del Arduino que se utilizará para observar la señal de salida en este case se tomará el pin 13

Configuración de la fuente de pulso. (Véase la figura 3.9).

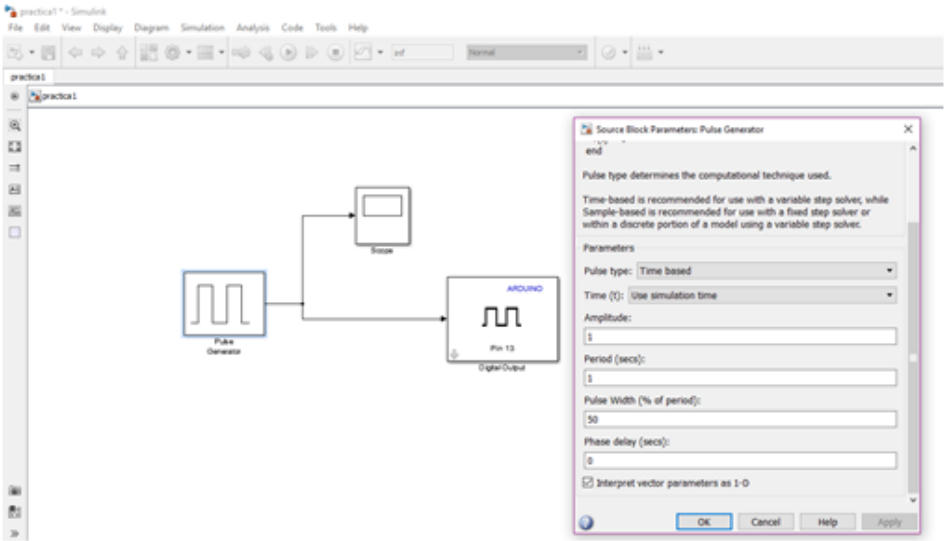


Figura 3. 9: Configuracion de pulso
Elaborado por: Autor

Configuración de pin de salida digital. (véase la figura 3.10).

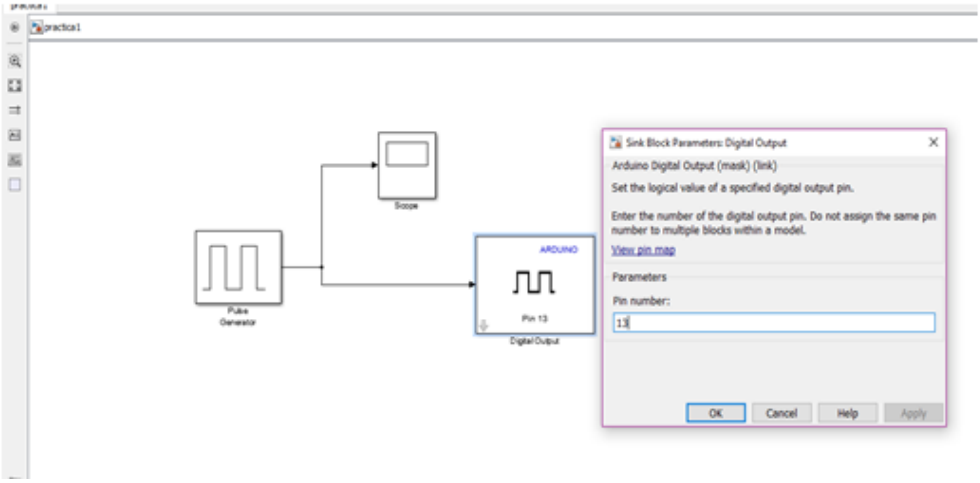


Figura 3. 10: Configuraciones de pin de salida
Elaborado por: Autor.

Una vez concluida con la conexión y configuración se comienza la configuración del programa para poder simular y cargar el programa en nuestro Arduino.

Configuración para simulación y compilación en nuestro Arduino

- Vamos a tools> run on the target hardware> opción. (véase la figura 3.11).

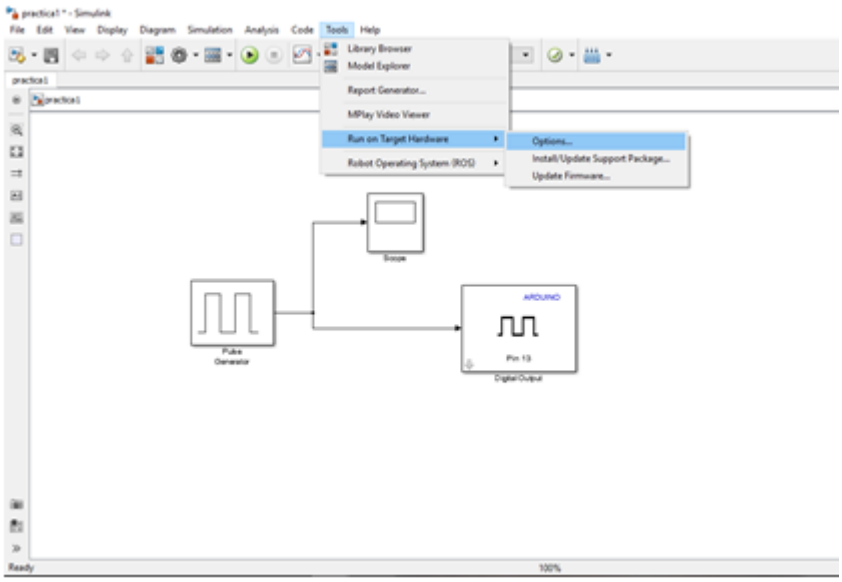


Figura 3. 11: Realizacion de opciones
Elaborado por: Autor.

Aquí encontraremos una ventana donde nos dirá en target hardware selection, debemos elegir el que diga Arduino Mega 2560, luego de esto nos aparecerán algunas especificaciones de la tarjeta y observaremos un cuadro de dialogo que dice <<set host Com port>> lo cambiaremos a Manually (véase la figura 3.12) y escribiremos el número de puerto al que se conecta el Arduino mega a la computadora en este caso se conectó al puerto com10 (véase la figura 3.13). (Véase la figura 3.14).

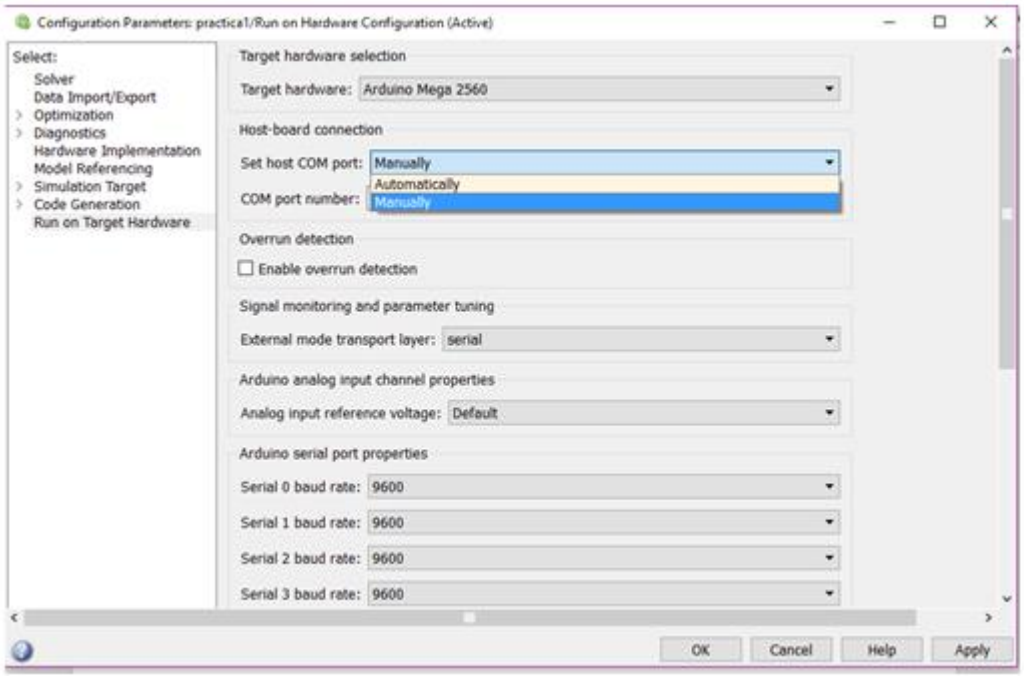


Figura 3. 12: Configuraciones de comunicación del puerto COM manual.
Elaborado por: Autor.

Para verificar el número de puesto al que se conectó buscamos administrador de dispositivos ya con la ventana abierta buscaremos en la lista la opción puertos y la desglosamos nos aparecerán los puertos existentes de nuestra computadora que se están usando y en este caso nos indica que el usb-serial está conectado al (com10) este sería nuestro Arduino. (Véase la figura 3.14).

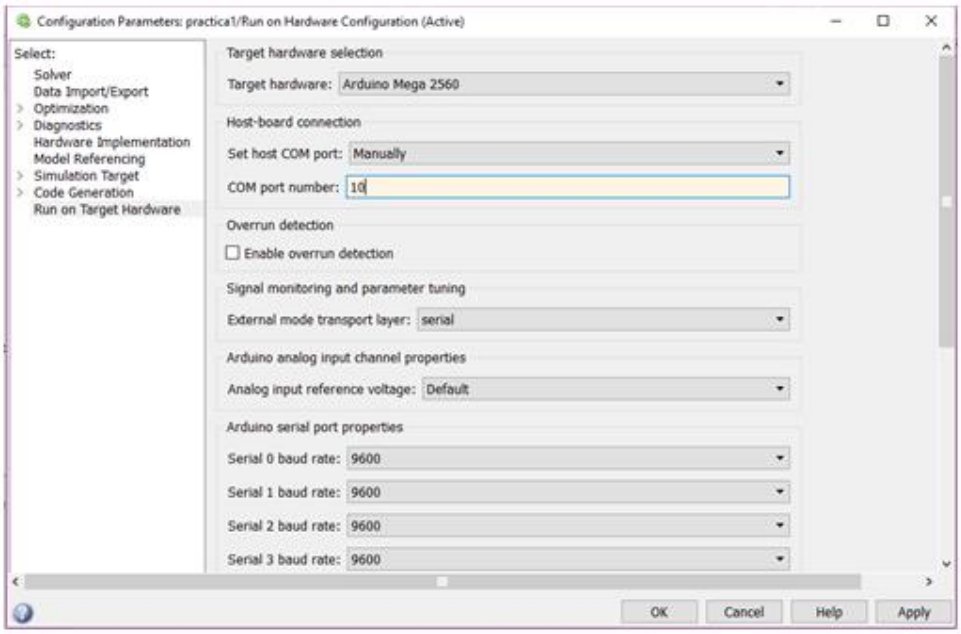


Figura 3. 13: Selección del número de puerto COM.
Elaborado por: Autor.

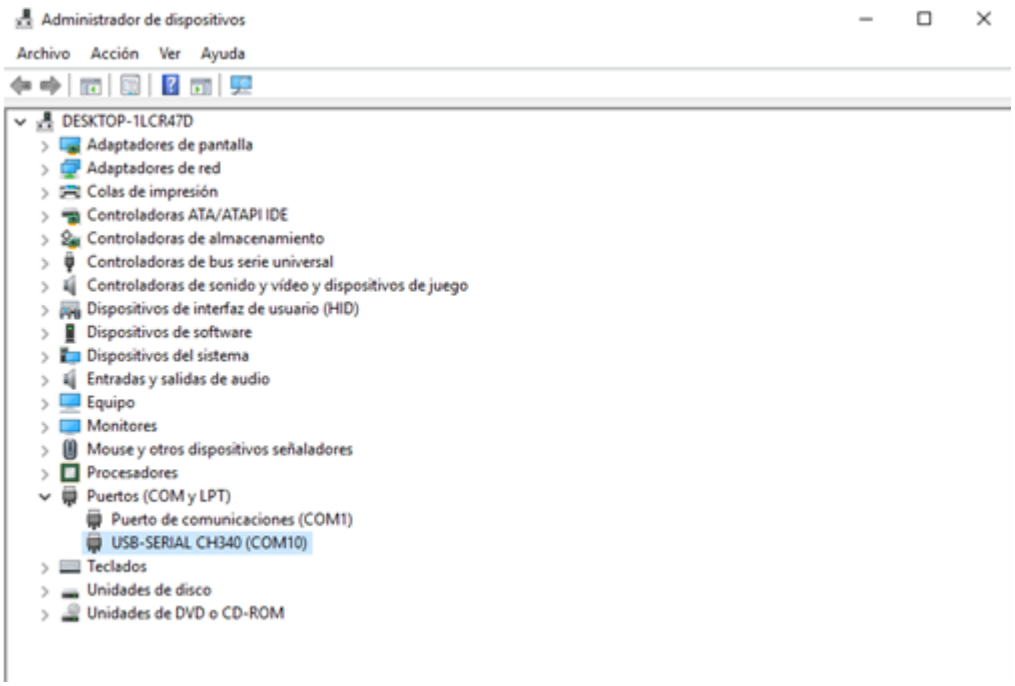


Figura 3. 14: Busqueda de opciones de puerto USB serial.
Elaborado por: Autor.

Cuando acabemos esto en la parte superior de nuestra ventana encontraremos un recuadro que dice 10 se debe cambiar a inf para que

nuestra simulación no se detenga y alado habrá un menú popup donde debemos selección la opción eternal para poder nosotros interactuar desde el programa con el Arduino y viceversa. (Véase la figura 3.15). Y (Véase la figura 3.16).

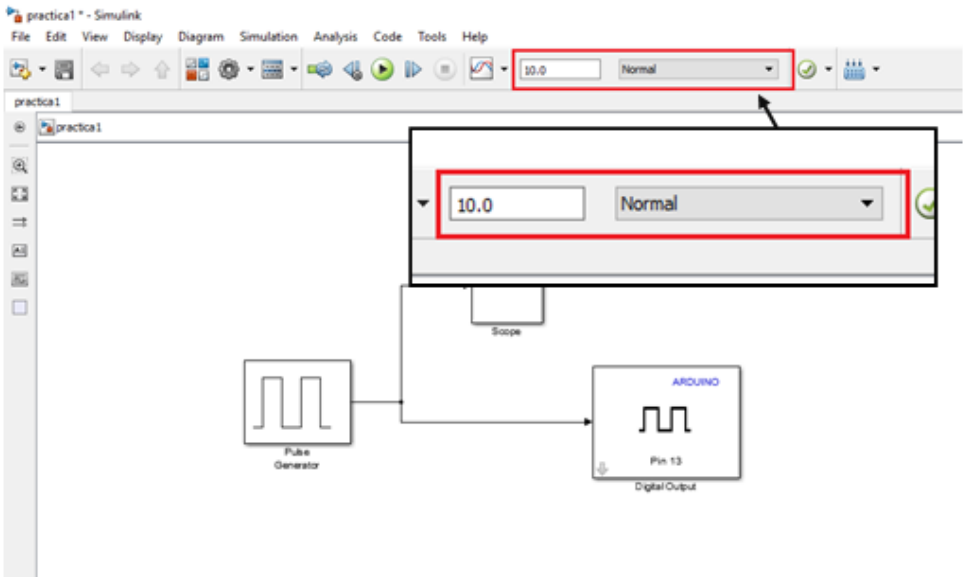


Figura 3. 15: Selección de opciones a INF
Elaborado por: Autor.

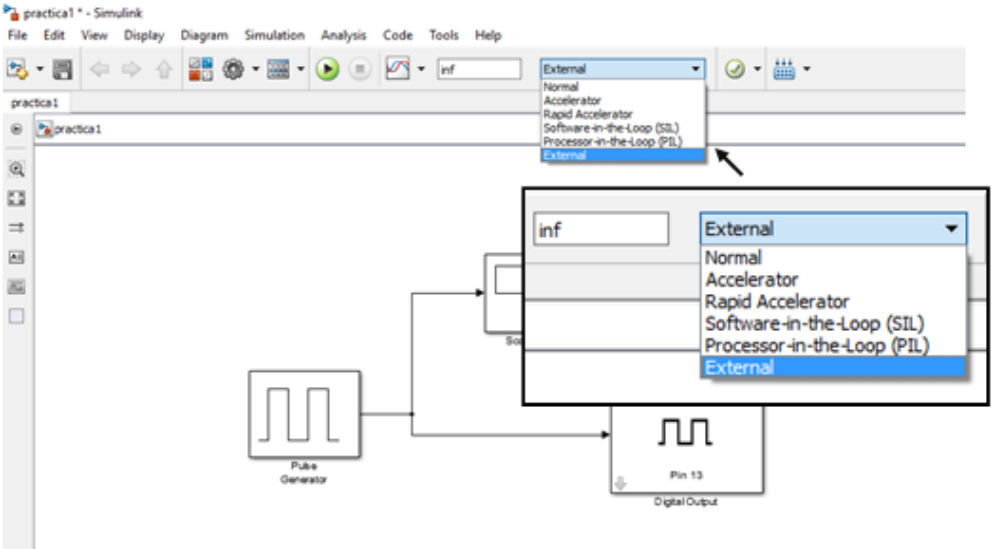


Figura 3. 16: Selección en la opcion external
Elaborado por: Autor.

Ya con todas estas configuraciones finalmente podremos proceder a compilar, cargar y simular nuestra práctica, cabe recalcar que este procedimiento se lleva a cabo en cada modelo nuevo que se cree desde el

configurar en que tarjeta cargaras tu programa hasta el cambiar el tiempo por default de 10.0 a inf y cambiar la pestaña de modo en external.

Para compilar usaremos deploy to hardware le damos clic y el programa compilara y cargara a la tarjeta. (Véase la figura 3.17).

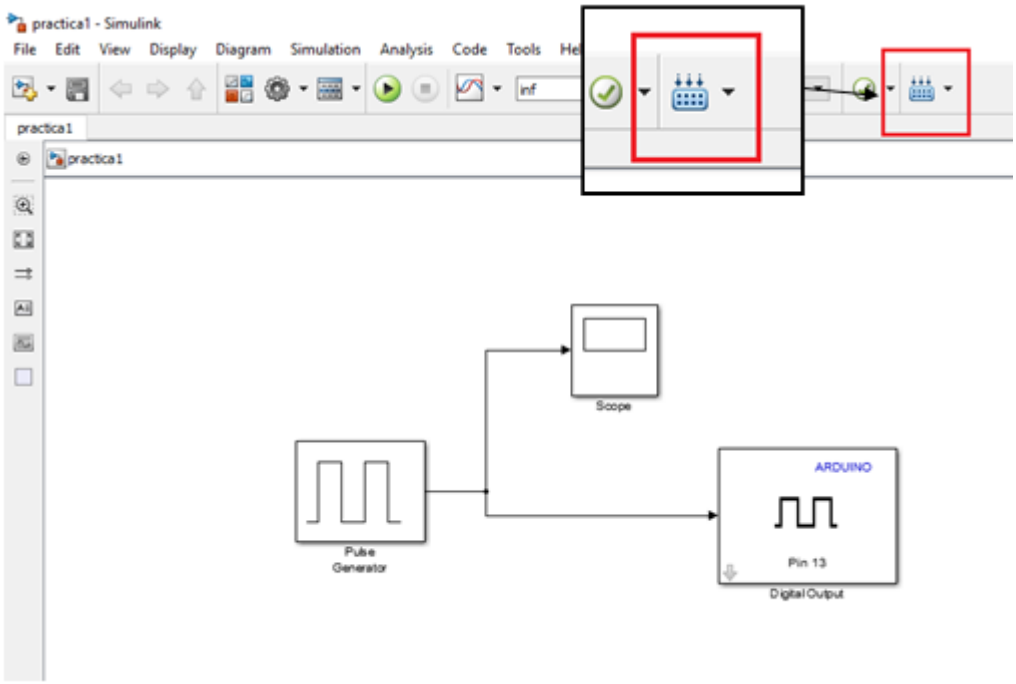


Figura 3. 17: Esperamos compilacion de tarjeta
Elaborado por: Autor.

Cuando se inicie la simulación damos clic en play para poder interactuar con el Arduino y podremos darle doble clic al scope y veremos cuál es la señal en la entrada que tiene nuestro pin 13 del Arduino.

Podremos observar físicamente que el led se enciende y se apaga según como se halla configurado a la señal de salida. (Véase la figura 3.18).

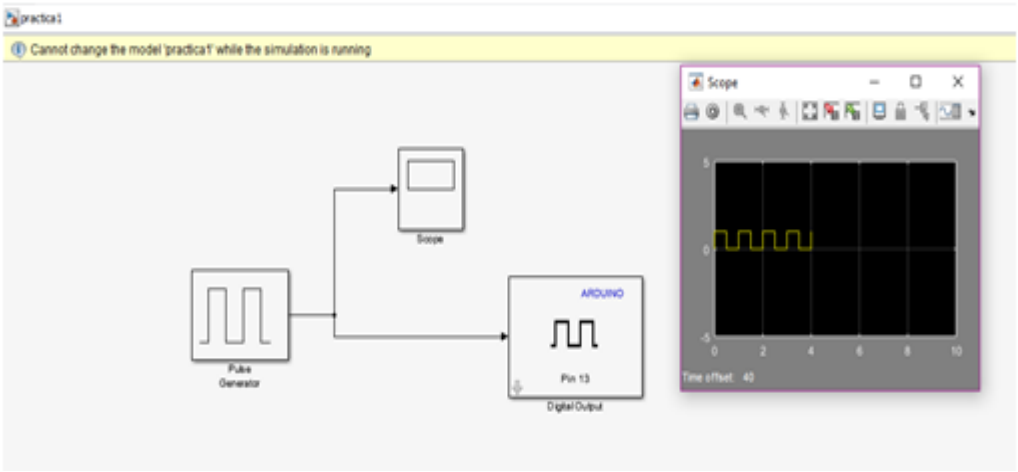


Figura 3. 18: Muestra de señal de salida
Elaborado por: Autor.

3.2. Aplicación de práctica #2; modulación pwm con una entrada analógica

Objetivos

- Aprender a usar la entrada analógica
- Aprender a usar las salidas de pwm

Materiales

- Led
- Resistencia de 1kohm
- Cables
- Arduino mega

Desarrollo

La práctica consta en el encendido de un led por medio de una señal de pwm enviada desde un potenciómetro conectado a una entrada analógica de nuestra tarjeta

El diagrama de conexiones es el siguiente:

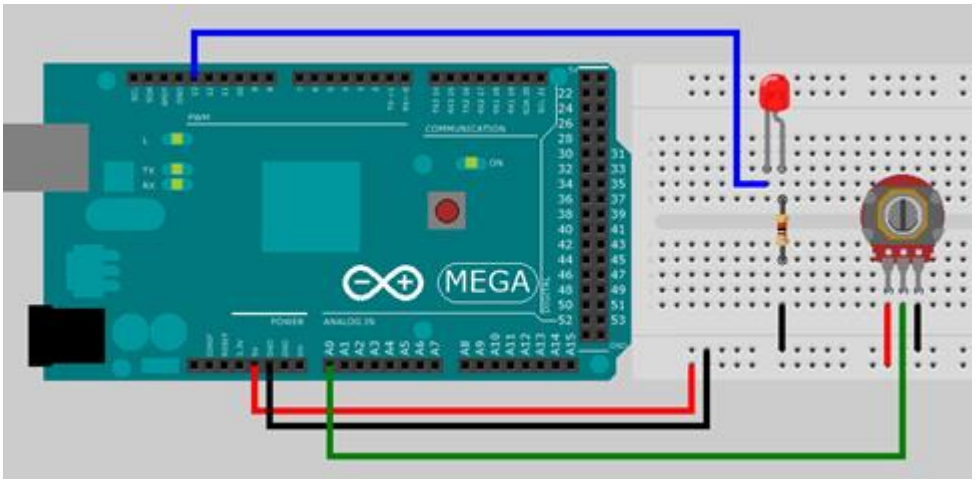


Figura 3. 19: Molulacion PWM
Elaborado por: Autor.

Programación

- ✓ Agregamos todos los elementos que se usaran en el programa:

  Una entrada analógica del Arduino y una salida pwm

  Un bloque data type conversión

  Un bloque gain

  Un scope

  Y un mux

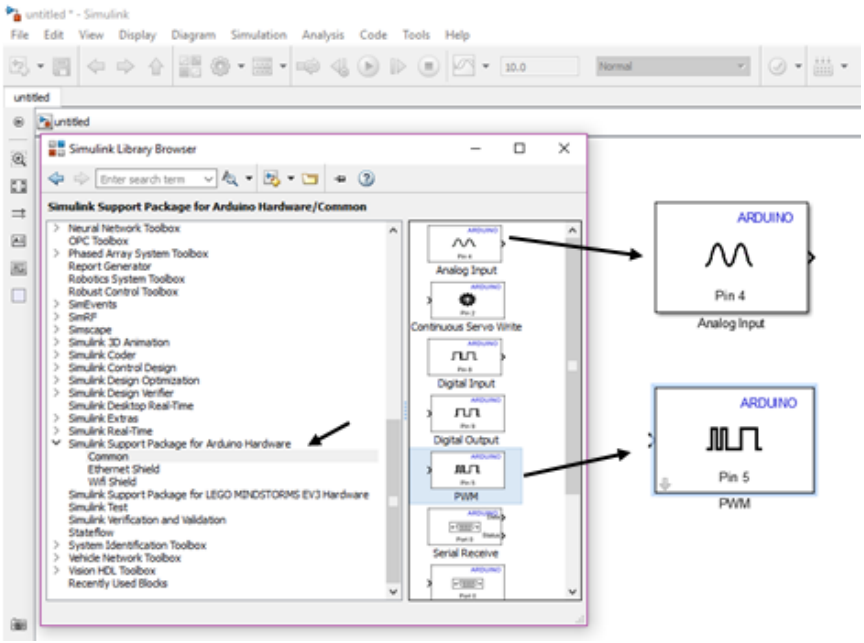


Figura 3. 20: Cargar nuevo programa
Elaborado por: Autor.

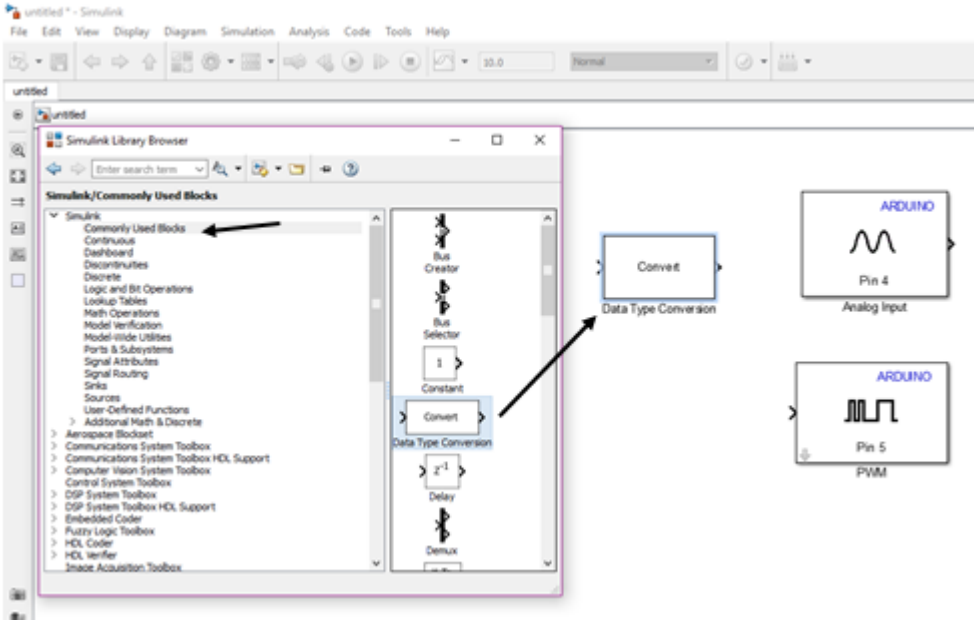


Figura 3. 21: Agregacion de bloque a la conexión
Elaborado por: Autor.

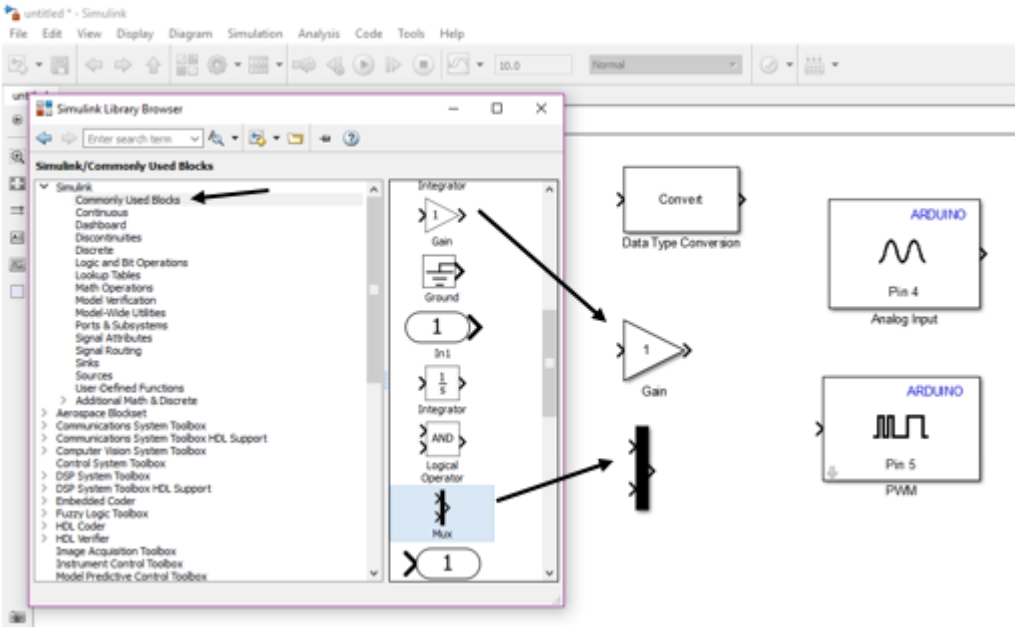


Figura 3. 22: Selección de bloque a usar.
Elaborado por: Autor.

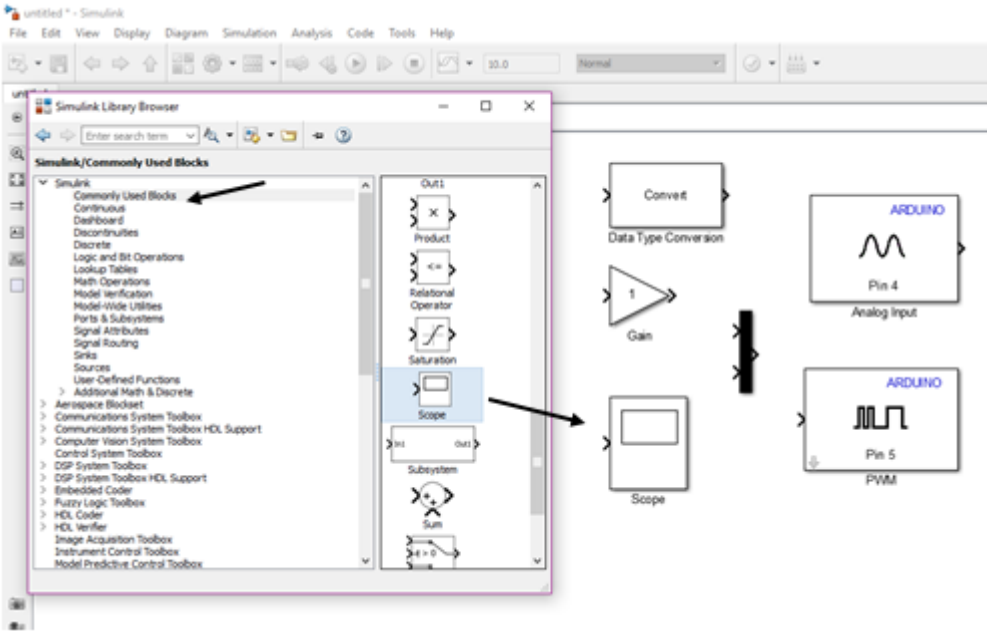


Figura 3. 23: Selección de aplicación scope.
Elaborado por: Autor.

- Luego de agregar los bloques realizamos las conexiones tal y como se muestra en la figura. (Véase la figura 3.24)

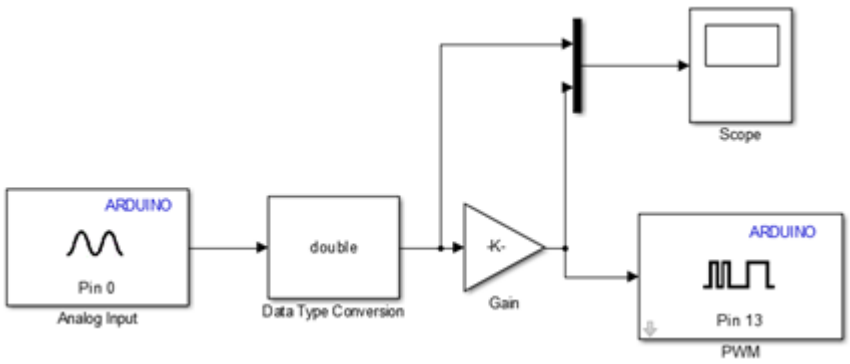


Figura 3. 24: Esquema de bloques a la figuras
Elaborado por: Autor.

Hora procedemos a configurar los pines de entrada analógica y salida pwm que usaremos en la placa en este caso hemos cogido el pin0 de entrada analógica y el pin13 de pwm. (Véase la figura 3.25), (Véase la figura 3.26).

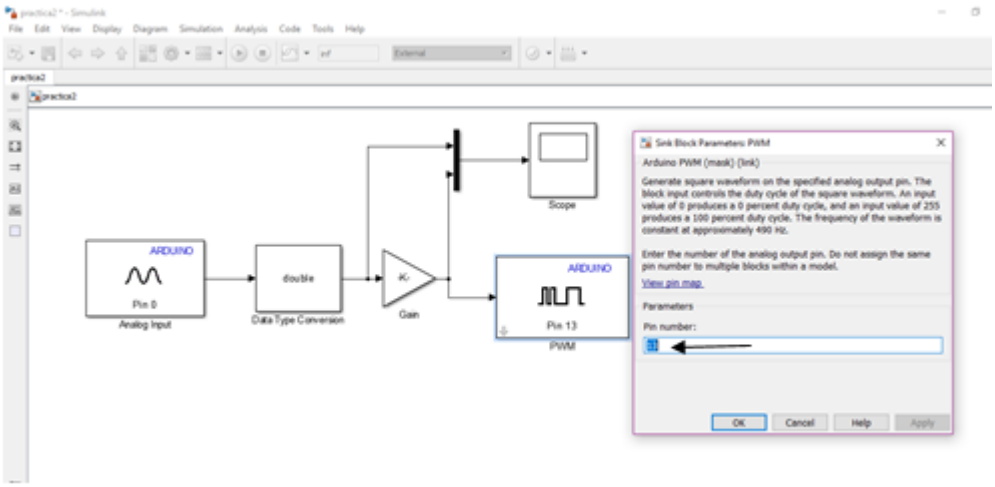


Figura 3. 25: Configuracion pin13 pwm.
Elaborado por: Autor.

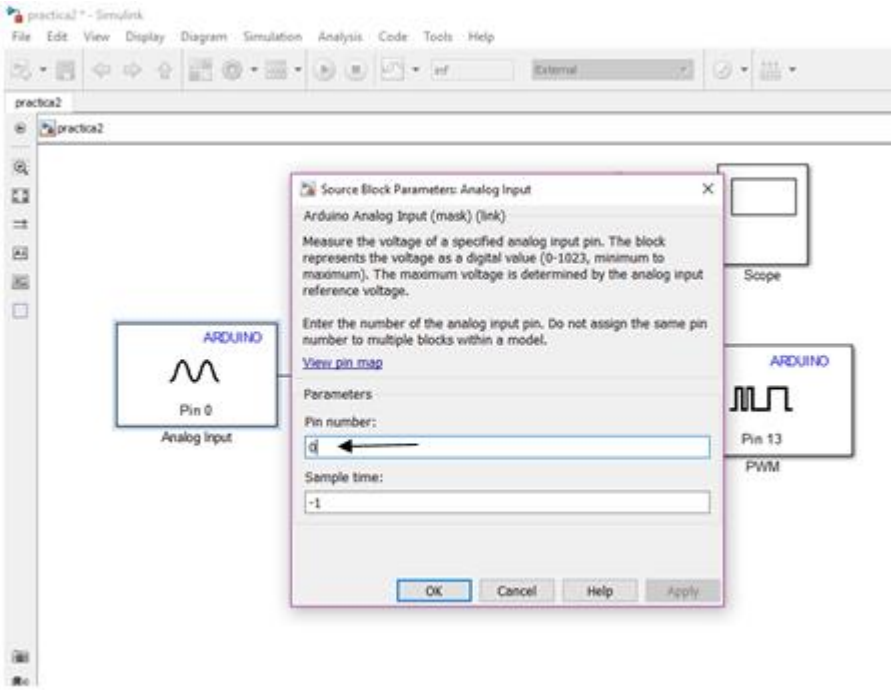


Figura 3. 26: Configuracion de pines analogicos
Elaborado por: Autor.

Ahora configuramos nuestro bloque de data type conversión y donde dice

Output data type seleccionamos doublé. (Véase la figura 3.27).

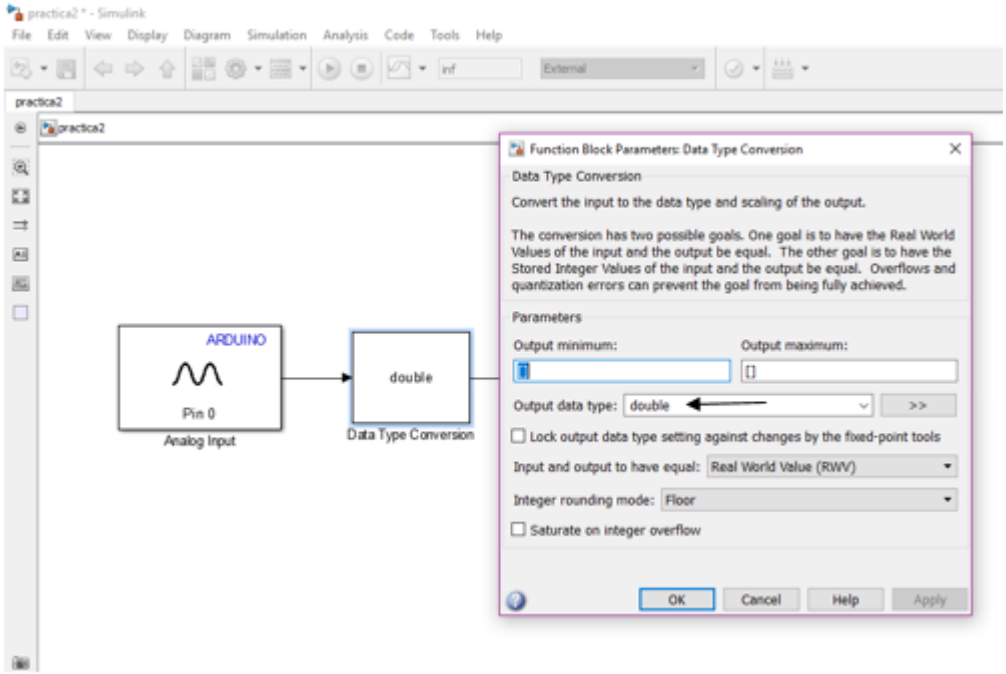


Figura 3. 27: Configuracion data type.
Elaborado por: Autor.

Finalmente configuramos el bloque gain esto se hace porque nuestro pwm solo trabajo con valores hasta 255 y nuestra entrada analógica trabaja con valores hasta 1023, entonces si hacemos la división 255 /1023 tendremos que cuando halla 1023 en la salida del bloque double tendremos 255 en la entrada de nuestro bloque de pwm. (Véase la figura 3.28). (Véase la figura 3.29).

Este es un ejemplo para comprender como funciona o porque se coloca el bloque de ganancia

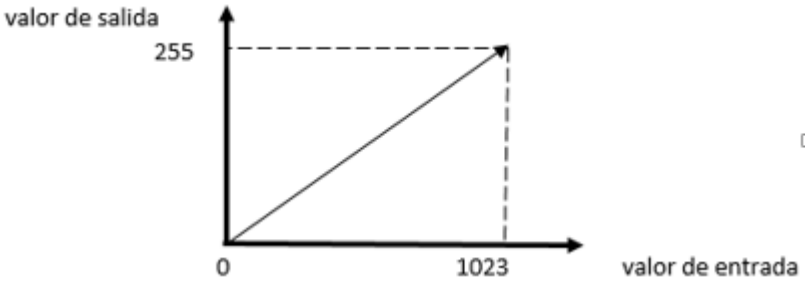


Figura 3. 28: Muestreo de bloque de ganancia
Elaborado por: Autor.

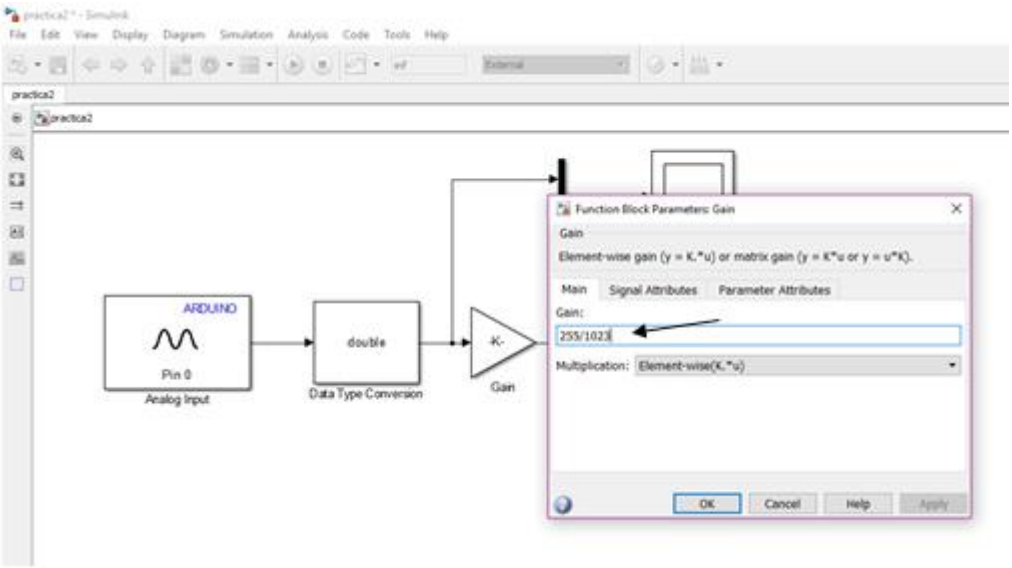


Figura 3. 29: Configuracion de pin
Elaborado por: Autor.

Listo ahora podremos simular, tomar en cuenta que para simular se debe hacer las configuraciones que se indican en la primera práctica en “Configuración para simulación y compilación en Arduino” (Véase la figura 3.30)

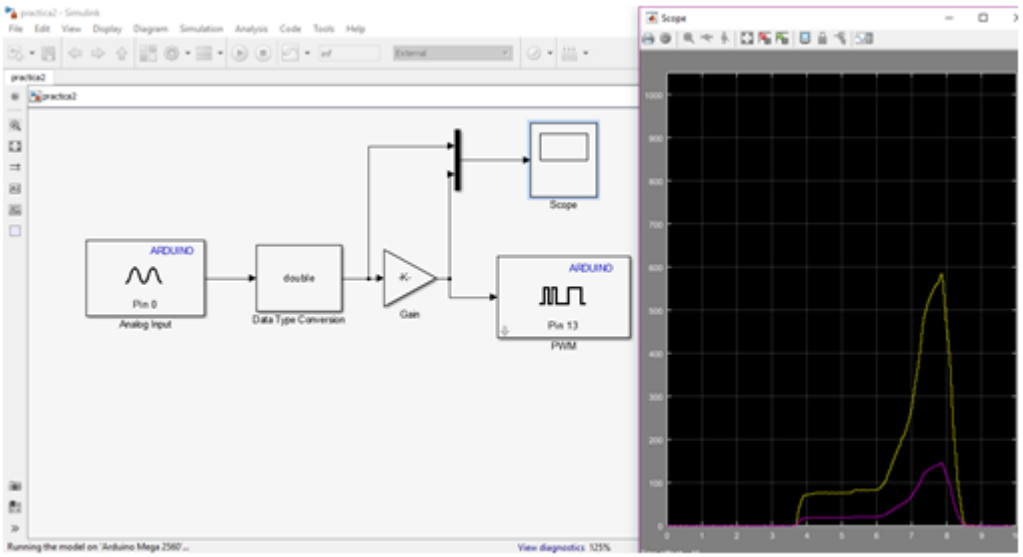


Figura 3. 30: Simulacion y compilacion de Arduino
Elaborado por: Autor.

La grafica que se observa la señal amarilla es la señal directa de la entrada analógica y la morada es la señal escalada que sale del bloque gain al bloque PWM esto se da por la operación que se configuro en el bloque gain

3.3. Aplicación de práctica#3; Encendido y apagado de un motor DC

Objetivos

- Aprender a controlar en tiempo real el Arduino desde el Simulink
- Conocer el controlador lm293d para motor de dc de poco amperaje

Materiales

- Integrado lm293d
- Motor de 5v
- Cables
- Ardruino mega

Desarrollo

La práctica consiste en el encendido y apagado de un motor por medio de una señal booleana enviada desde un switch utilizado en el programa Simulink

El diagrama de conexiones es el siguiente:

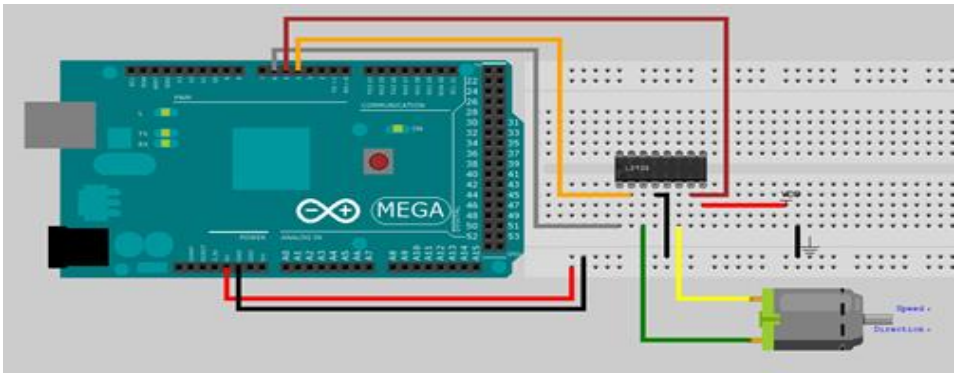


Figura 3. 31: Encendido y apagado de motor DC
Elaborado por: Autor.

Integrado lm293

El integrado L293D incluye cuatro circuitos para manejar cargas de potencia media, en especial pequeños motores y cargas inductivas, con la capacidad de controlar corriente hasta 600 mA en cada circuito y una tensión entre 4,5 V a 36 V. Los circuitos individuales se pueden usar de manera independiente para controlar cargas de todo tipo y, en el caso de ser motores, manejar un único sentido de giro. Pero, además, cualquiera de estos cuatro circuitos sirve para configurar la mitad de un puente H.

El integrado permite formar, entonces, dos puentes H completos, con los que se puede realizar el manejo de dos motores. En este caso el manejo será bidireccional, con frenado rápido y con posibilidad de implementar fácilmente el control de velocidad

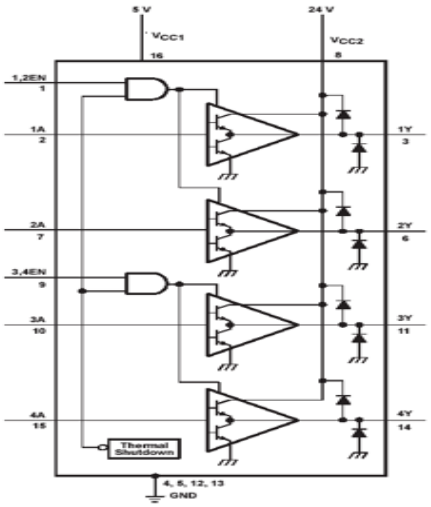


Figura 3. 32: Integrado, formacion de puente H
Elaborado por: Autor.

Diagrama de conexiones

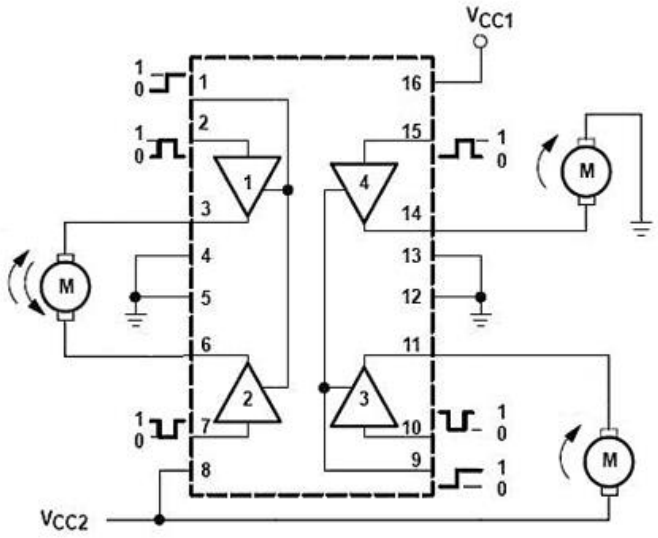


Figura 3. 33: Diagrama de conexión
Elaborado por: Autor.

Datasheet del integrado

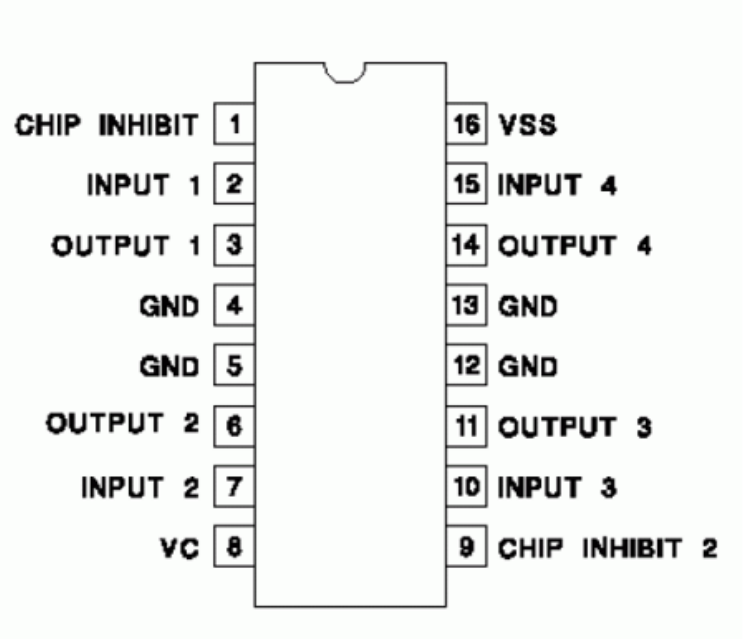


Figura 3. 34: Datasheet del integrado
Elaborado por: Autor.

Programación

- Iniciaremos agregando los bloques que utilizaremos para hacer nuestro programa
    - Bloques de salida digital del Arduino
    - Switch
    - Compuerta and
    - Constantes

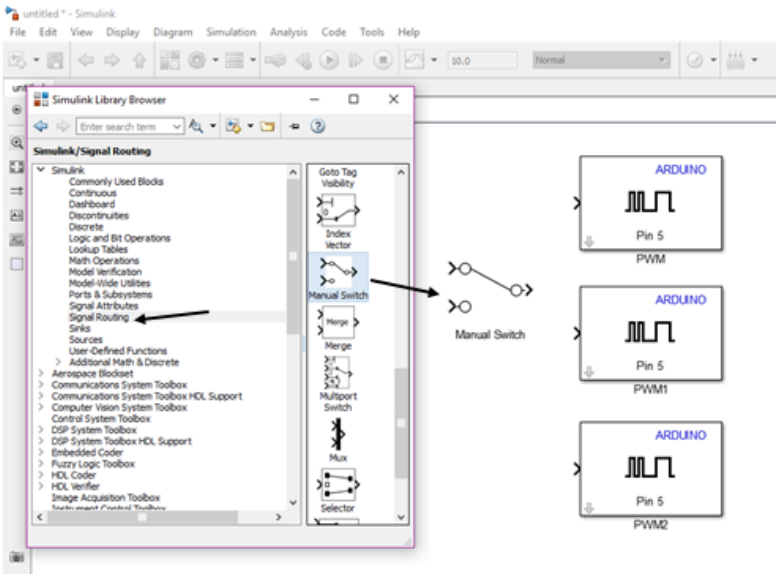


Figura 3. 35: Aplicación de slida digital
Elaborado por: Autor.

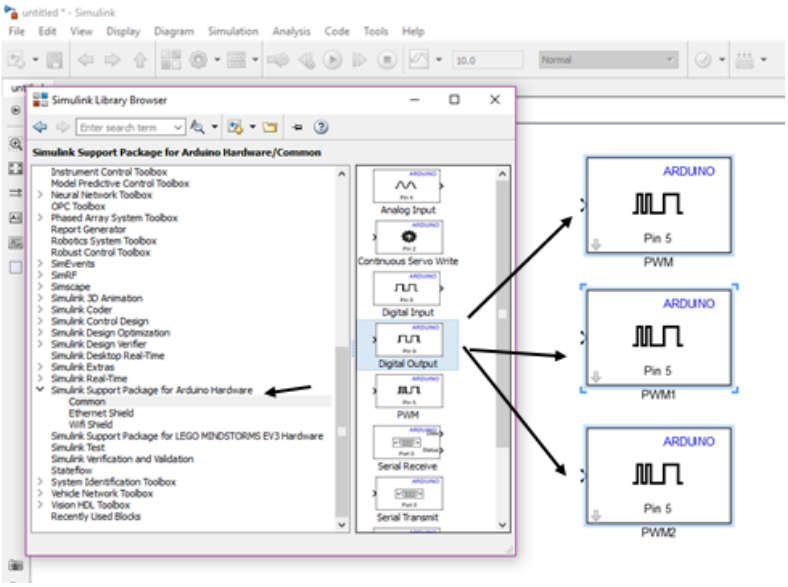


Figura 3. 36: Aplicación de swicth
Elaborado por: Autor.

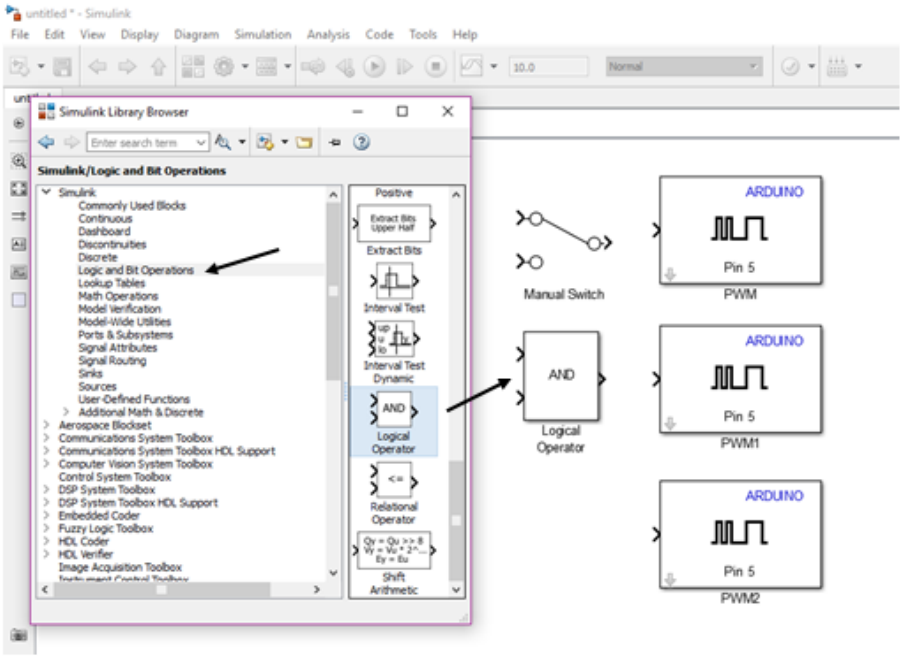


Figura 3. 37: Aplicación devariables constantes.
Elaborado por: Autor.

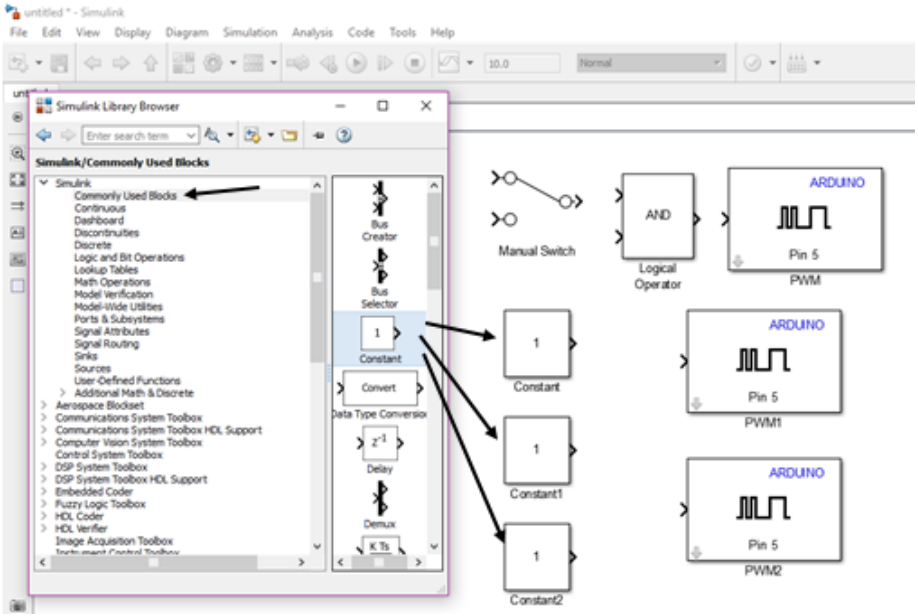


Figura 3. 38: Aplicación de bloques
Elaborado por: Autor.

✓ Continuamos con las conexiones de los bloques tal y como se muestra en la figura. (Véase la figura 3.39).

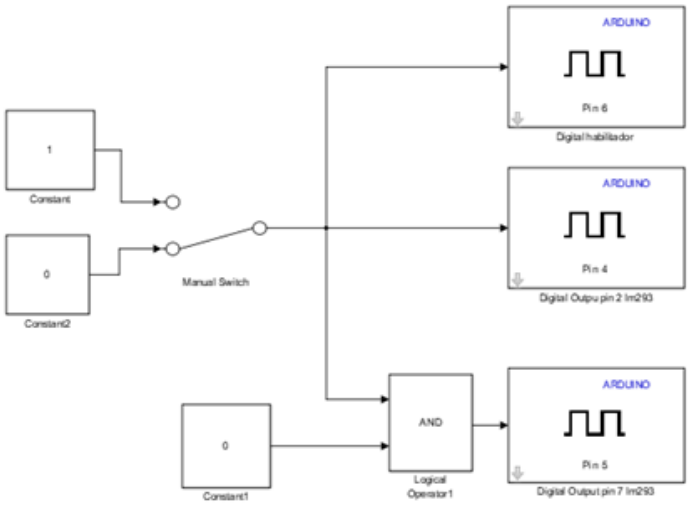


Figura 3. 39: conexión de bloques
Elaborado por: Autor.

- ✓ Procedemos a configurar los bloques de salida analógica según los pines que vamos a utilizar en este caso serán pin 4, 5 y 6.

Pin 6. Configuraciones de las salidas analógicas. (Véase la figura 3.40)

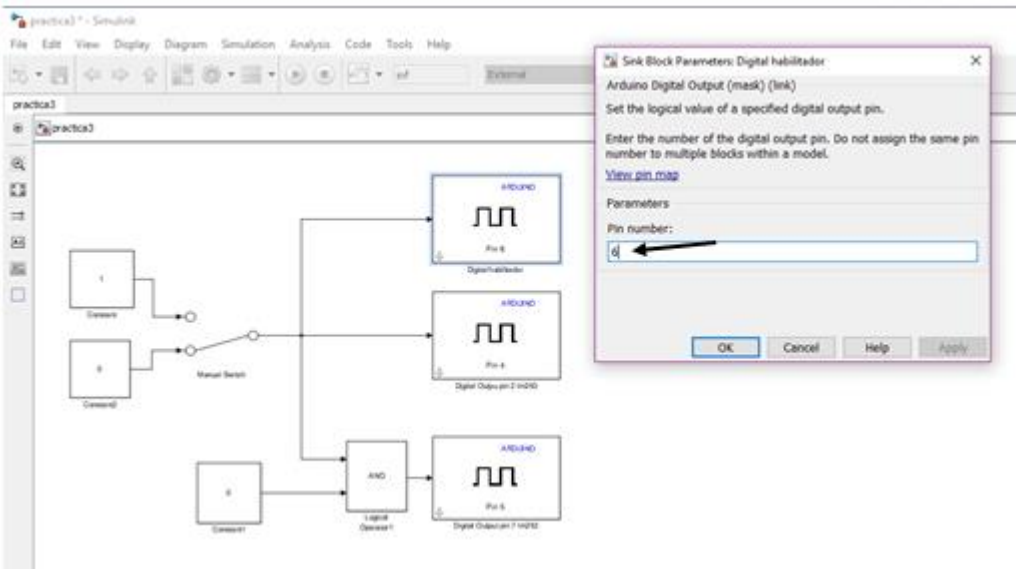


Figura 3. 40: Configuracion de salida analogica pin6.
Elaborado por: Autor.

Pin 4. Configuración de bloque de salida (Véase la figura 3.41)

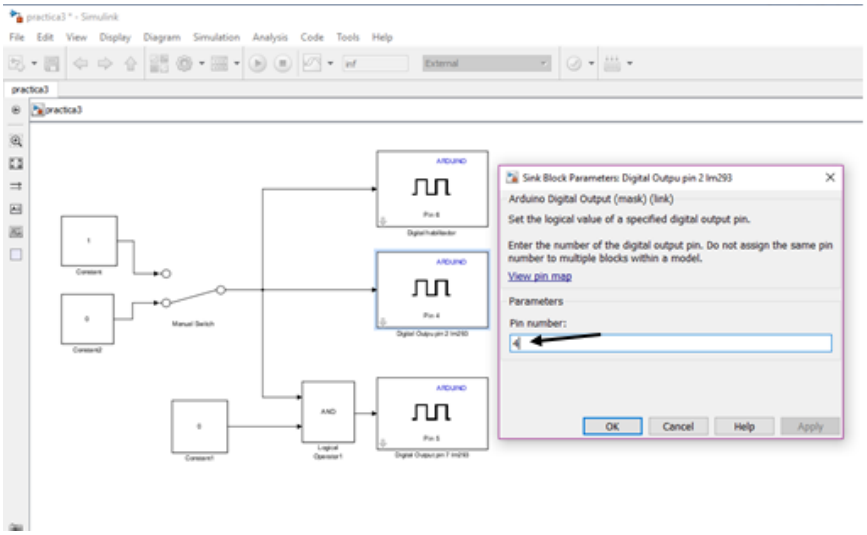


Figura 3. 41: Configuracion de salida analogica pin4.
Elaborado por: Autor.

Pin 5. Configuración de salidas analógicas. (Véase la figura 3.42)

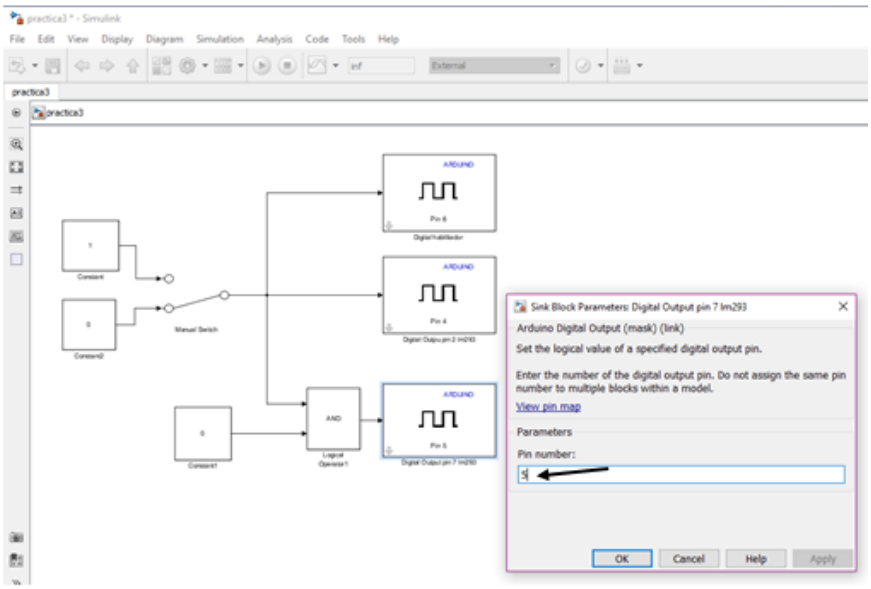


Figura 3. 42: Configuracion de salida analogica pin5
Elaborado por: Autor.

- ✓ Ahora configuramos los valores de las constantes.

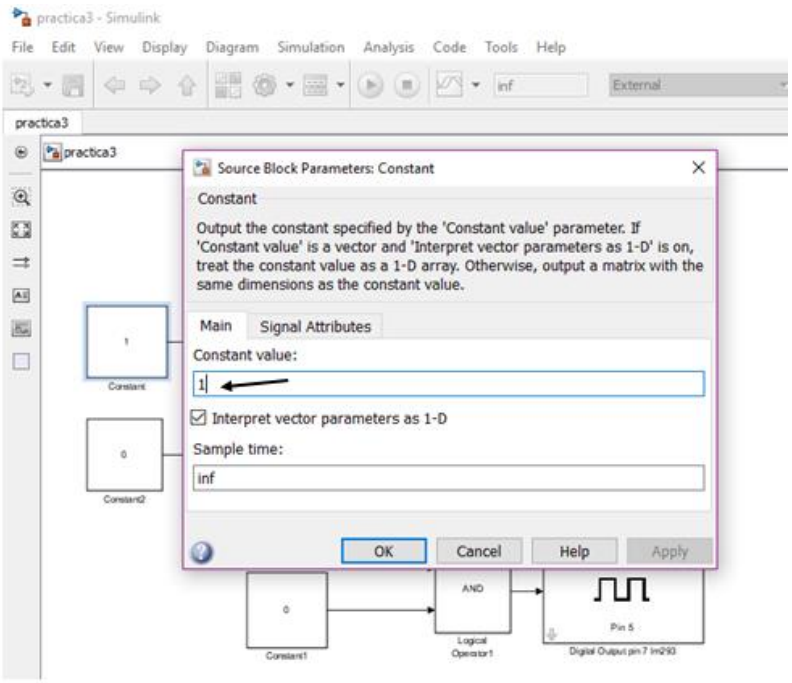


Figura 3. 43: Configuracion de valores de las constantes 1.
Elaborado por: Autor.

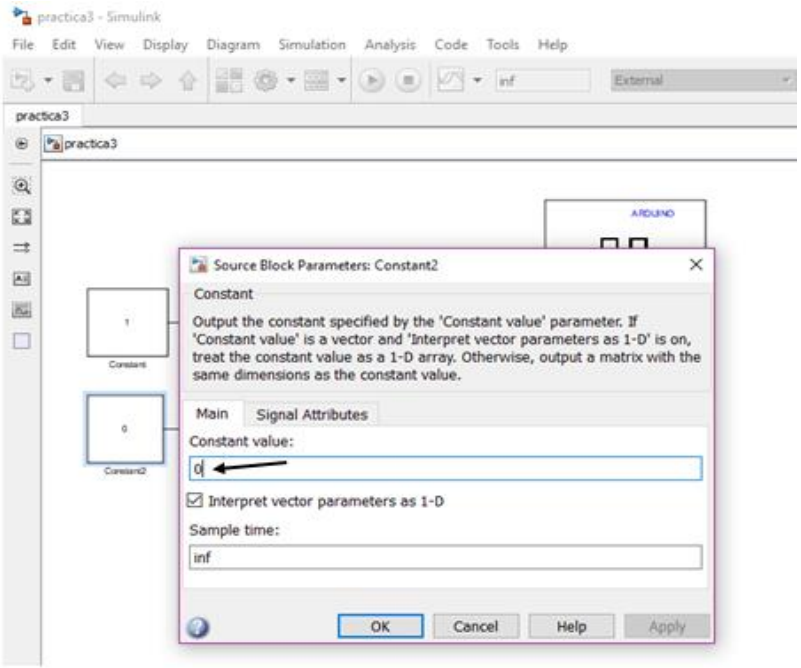


Figura 3. 44: Configuracion de valores de las constantes 0.
Elaborado por: Autor.

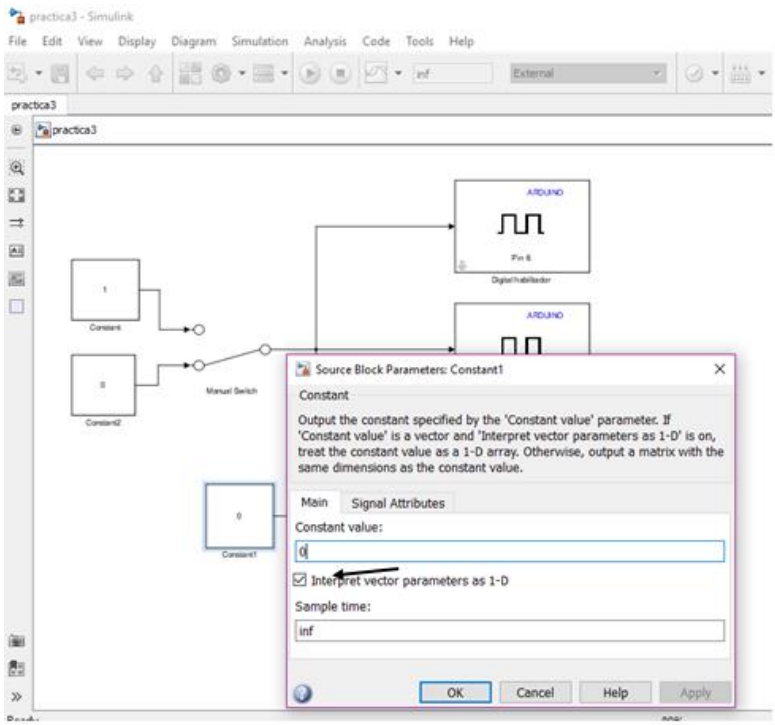


Figura 3. 45: Configuracion de valores de las constantes
Elaborado por: Autor.

✓ Ya con las configuraciones terminadas procedemos a simular.

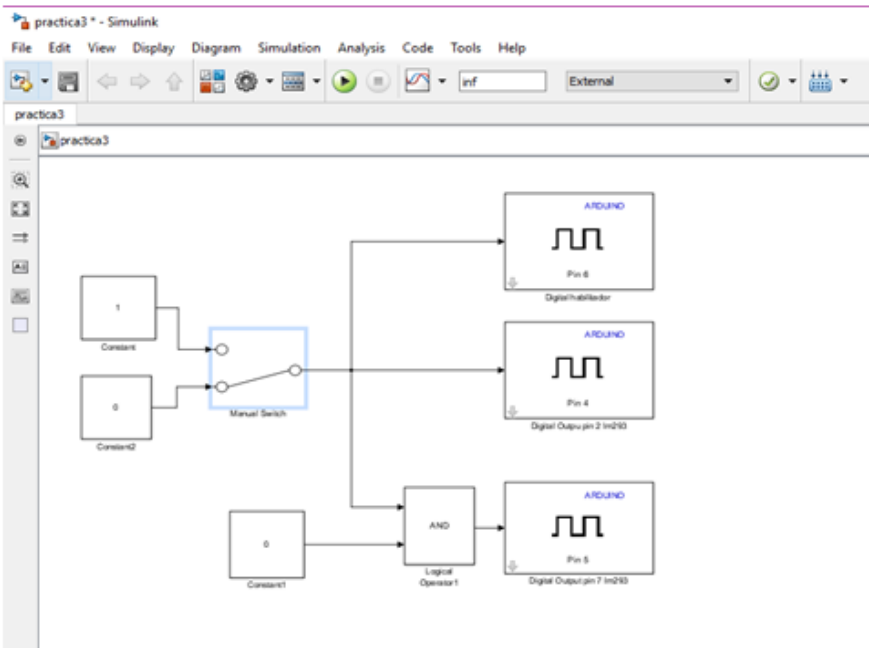


Figura 3. 46: Muestreo de Simulacion d eencendido
Elaborado por: Autor.

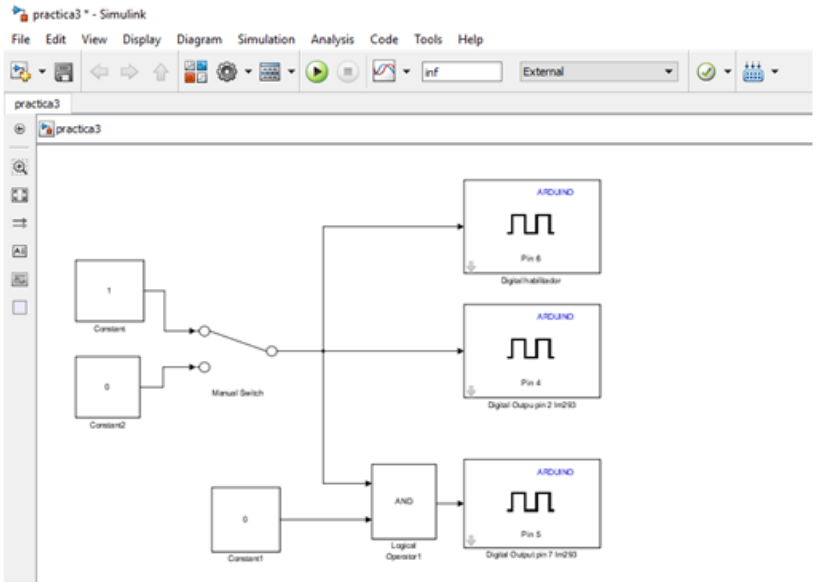


Figura 3. 47: Muestreo de Simulacion de apagado
Elaborado por: Autor.

En el momento de la simulación podremos observar que al dar doble clic sobre el switch cambiara de estado y esto hará que nuestro motor encienda y al darle doble clic nuevamente se apagara el motor esto, esto sucede también gracias a la conexión que se tiene físicamente porque con nuestro programa deshabilitamos nuestro motor en el pin 1 del integrado con el haremos que nuestro motor deje de funcionar. (Véase la figura 3.47).

3.4. Aplicación de práctica#4 cambio de giro de motor de velocidad y cambio de giro automático

Objetivos

- Aprender a controlar en tiempo real el Arduino desde el Simulink.
- Conocer el controlador lm293d para motor de dc de poco amperaje.
- Aprender a controlar la velocidad del motor con una entrada analógica.

Materiales

- Integrado lm293d
- Motor de 5v
- Cables
- Arduino mega

Desarrollo

La práctica consiste en cambiar el sentido de giro de un motor por medio de una señal booleana enviada desde un switch utilizado en el programa Simulink y el control de la velocidad de este por medio de una señal pwm modulada con un potenciómetro.

El diagrama de conexiones es el siguiente:

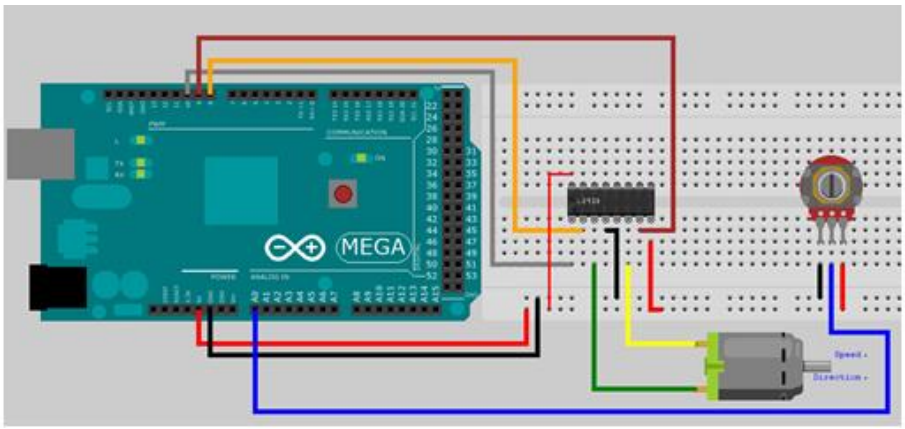


Figura 3. 48: Cambio de giro de motor de velocidad
Elaborado por: Autor.

- Se procede a agregar todos los bloques que utilizaremos en nuestro programa.

✓ Salida pwm, salida digital y entrada analógica del Arduino
✓ Switch
✓ Convertidor
✓ Bloque gain
✓ Generador de señal de pulso
✓ Compuerta lógica
✓ Mux

- ✓ Scope
- ✓ Contantes
- ✓ Entrada analógica, salida digital, salida pwm. (Véase la figura 3.49).

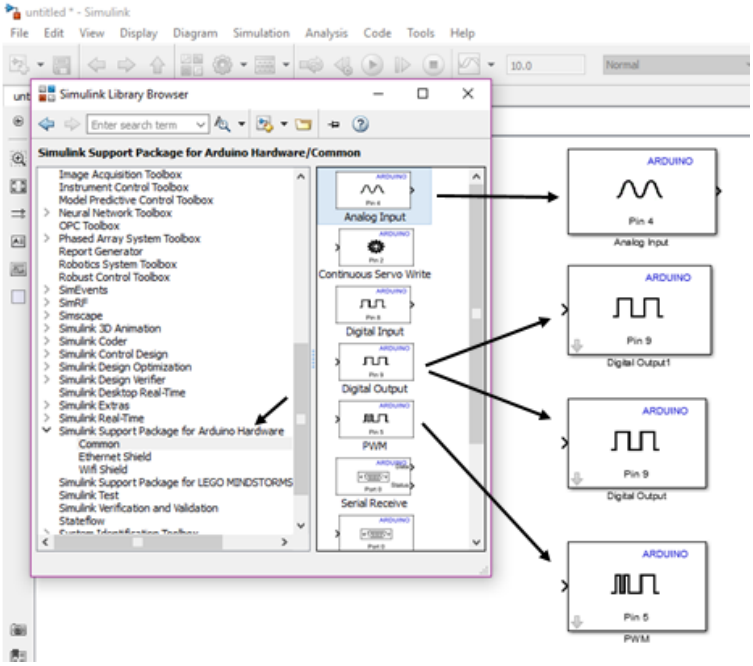


Figura 3. 49: Agregacion de bloques support
Elaborado por: Autor.

Bloque de constante, convertidor, gain. (Véase la figura 3.50).

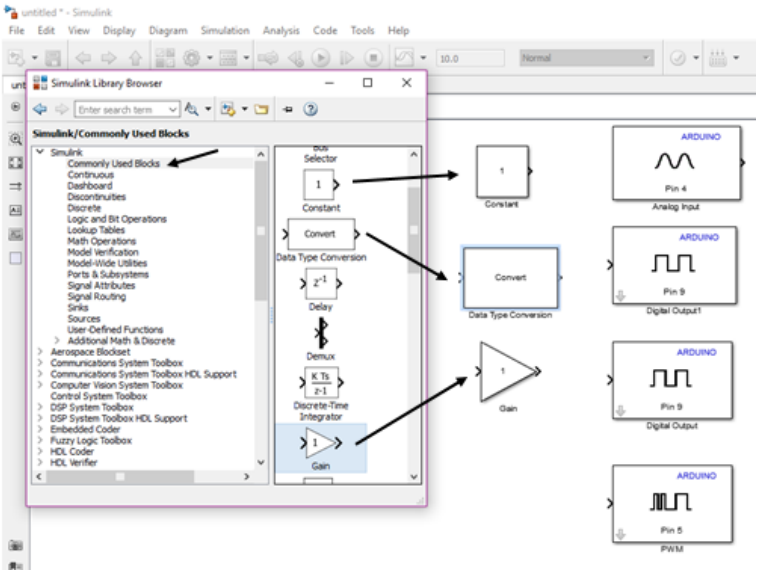


Figura 3. 50: Bloques de constante convertidor.
Elaborado por: Autor.

Mux, scope.Configuracion de aplicaciones (Véase la figura 3.51).

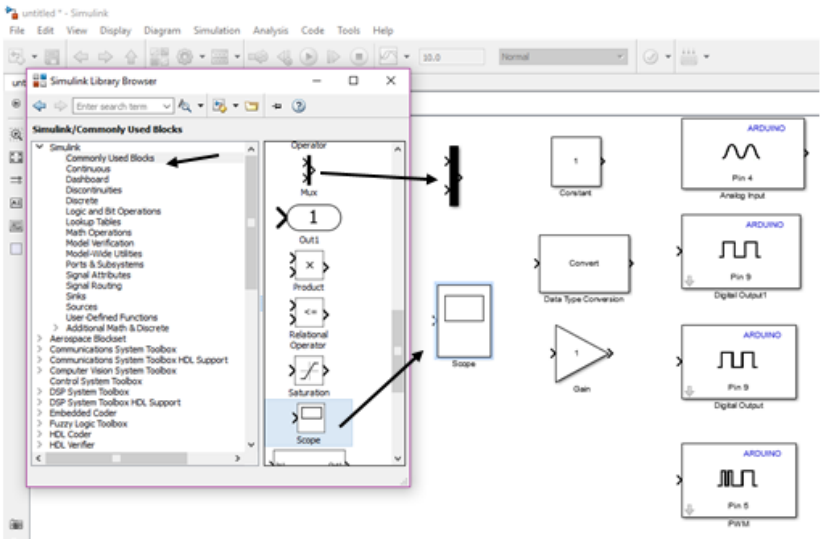


Figura 3. 51: Aplicación de scope
Elaborado por: Autor.

Siwtch, aplicación, configuración (Véase la figura 3.52).

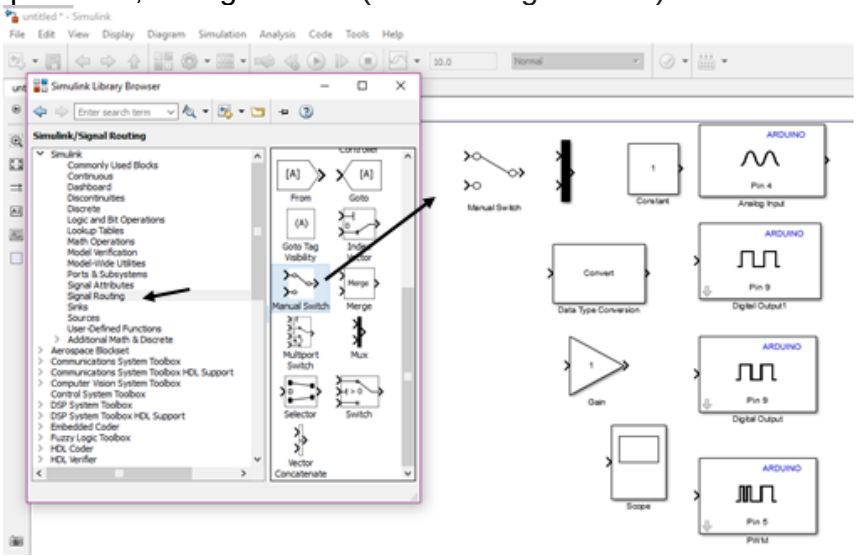


Figura 3. 52: Aplicación de siwtch
Elaborado por: Autor.

Generador de señal de pulso. (Véase la figura 3.53).

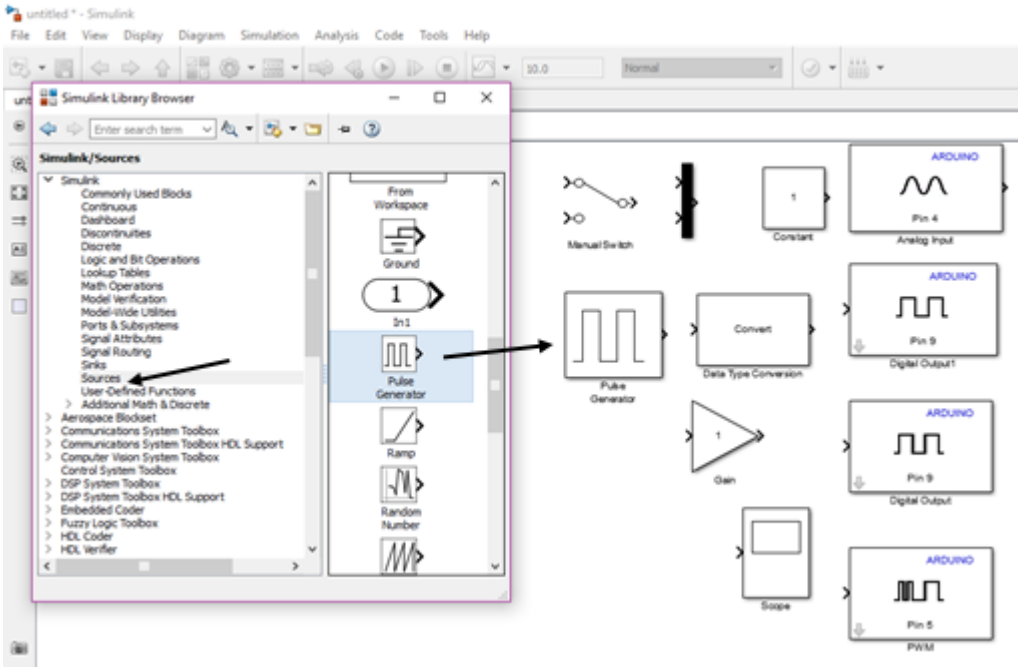


Figura 3. 53: Genarador de señal de pulso.
Elaborado por: Autor.

Compuerta lógica. (Véase la figura 3.54).

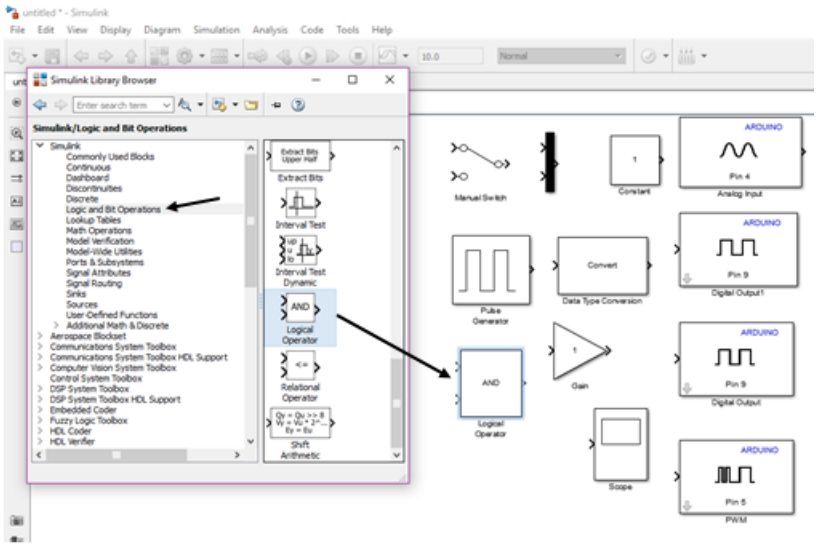


Figura 3. 54: Compuertas logicas AND.
Elaborado por: Autor.

- Una vez agregados todos los elementos procedemos a hacer las conexiones. (Véase la figura 3.55).

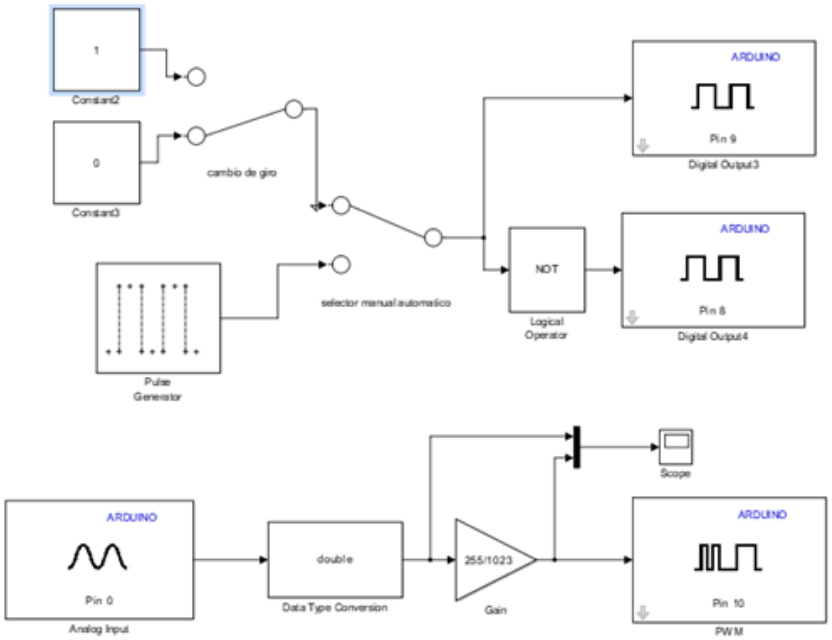


Figura 3. 55: Conexión de los elementos.
Elaborado por: Autor.

- Luego de conectar comenzamos con configurar los bloques de salida digital, pwm y entrada analógica del Arduino. (Véase la figura 3.56).

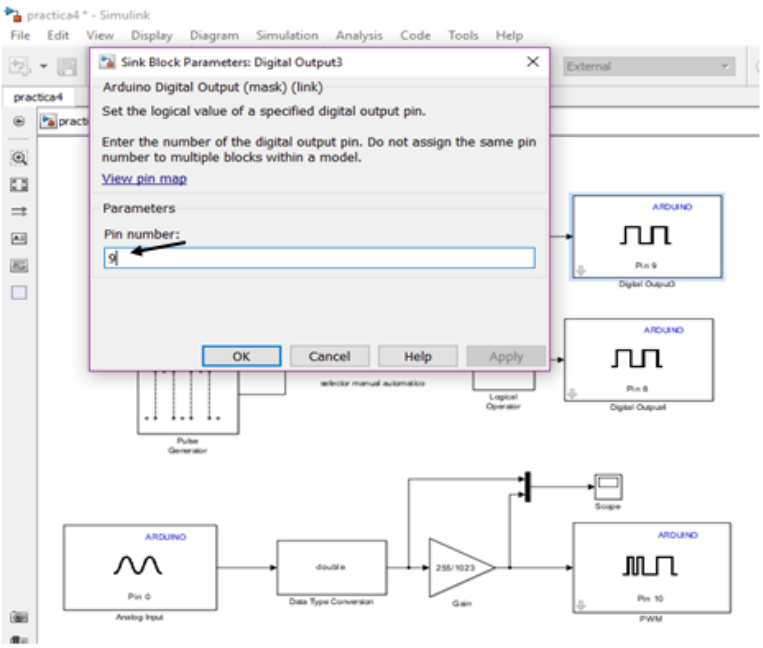


Figura 3. 56: Configuracion de bloques de salidas
Elaborado por: Autor.

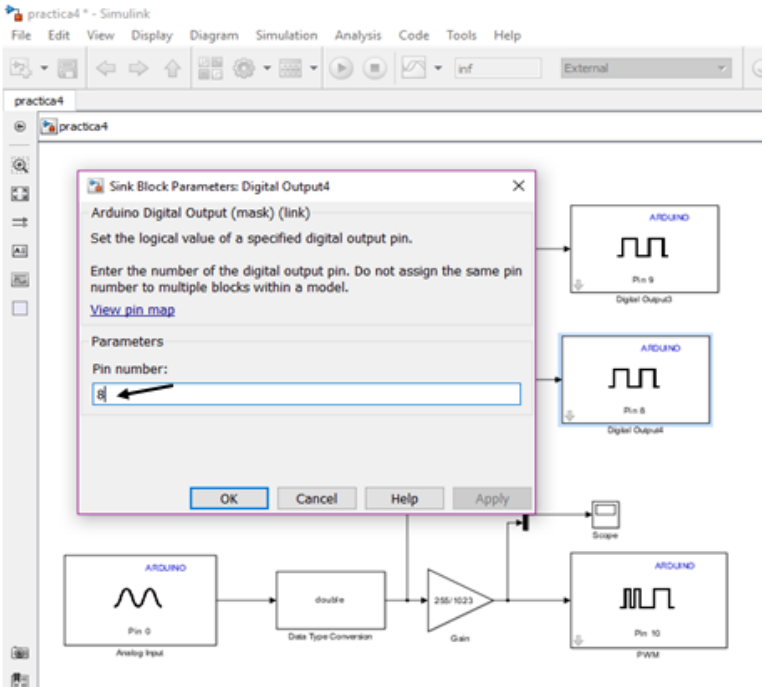


Figura 3. 57: Configuracion de bloques.
Elaborado por: Autor.

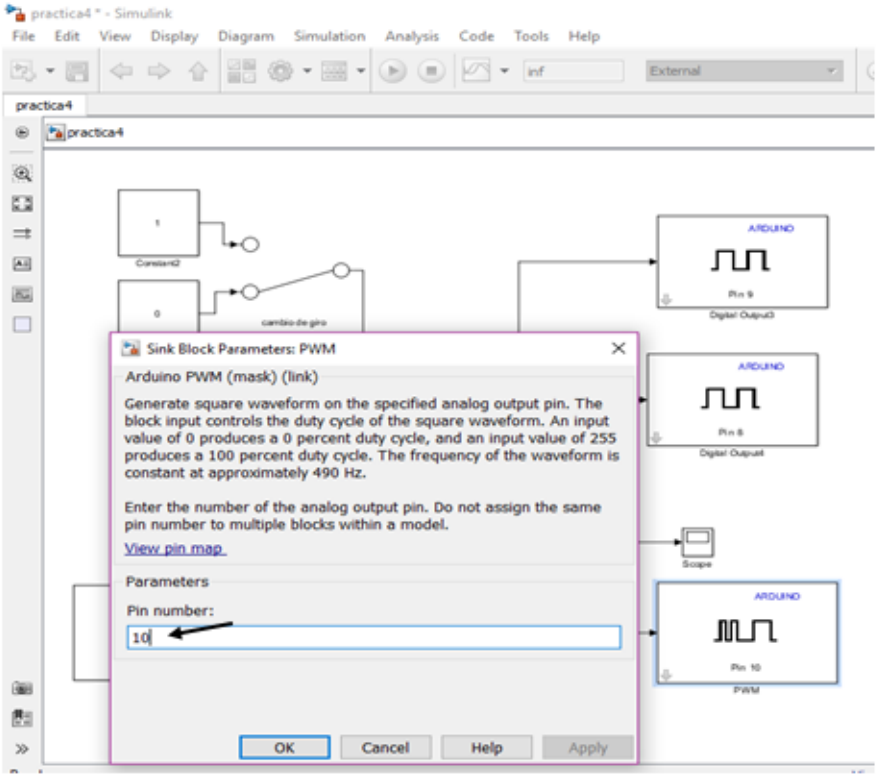


Figura 3. 58: Configuracion de bloques pwm
Elaborado por: Autor.

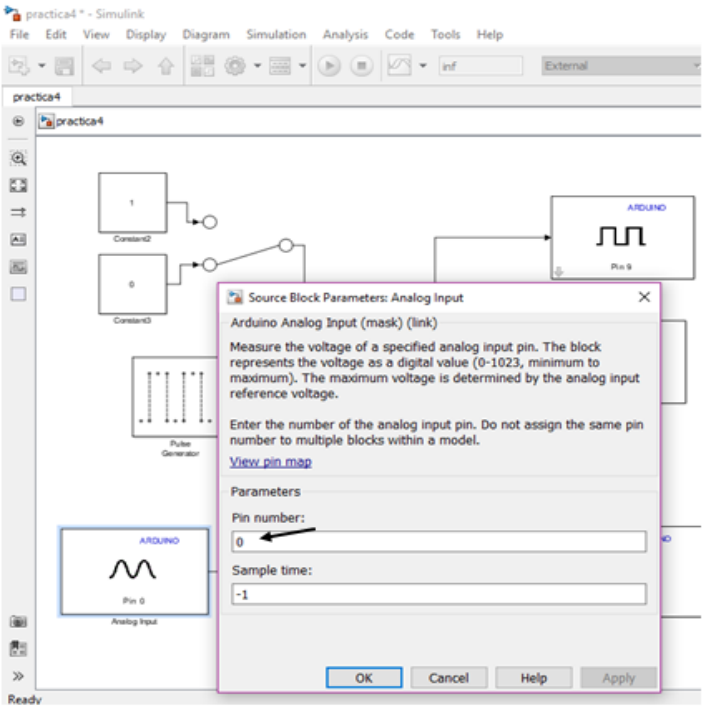


Figura 3. 59: Configuracion de bloque señal de salida.
Elaborado por: Autor.

- Ahora configuramos nuestro generador de señal de pulso. (Véase la figura 3.60).

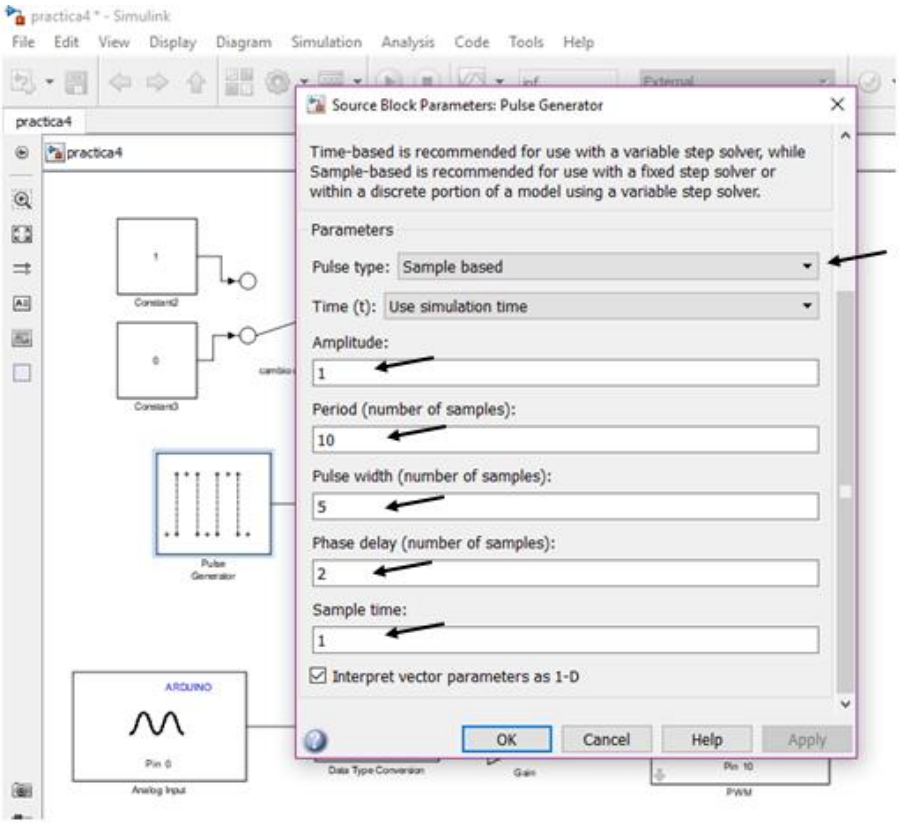


Figura 3. 60: Configuracion de generador de señal
Elaborado por: Autor.

- Continuamos con la configuración de la compuerta lógica. (Véase la figura 3.61).

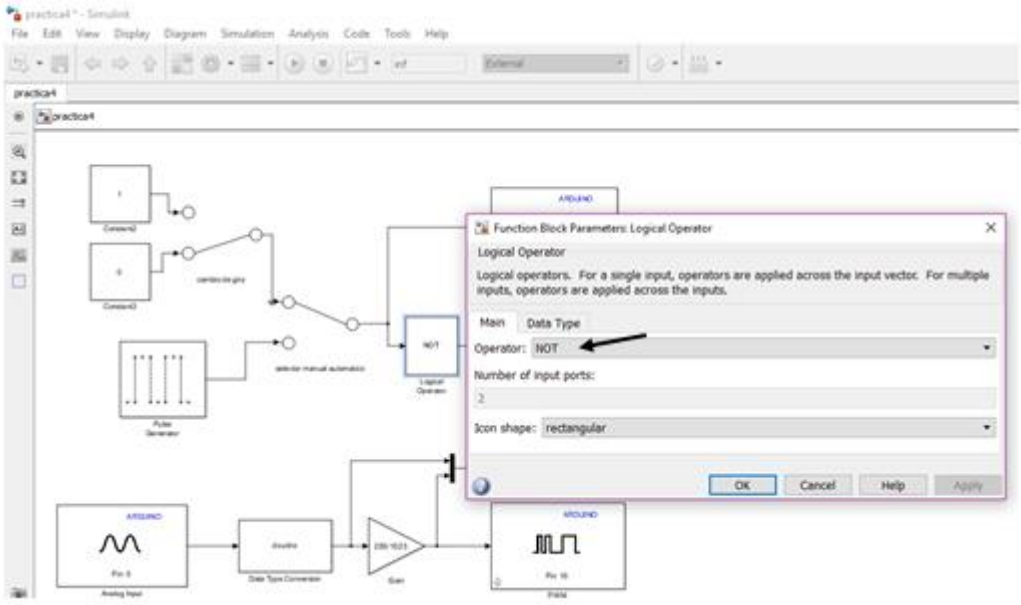


Figura 3. 61: Configuracion de puertas logicas.
Elaborado por: Autor.

- Ahora configuramos nuestro bloque de conversión. (Véase la figura 3.62).

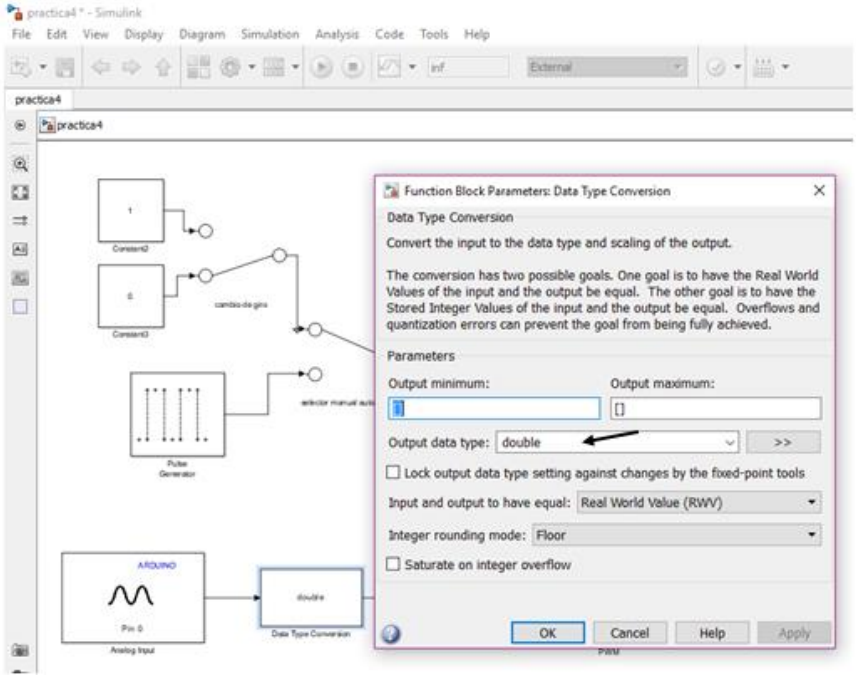


Figura 3. 62: Configuraciones de dato de conversion.
Elaborado por: Autor.

- Se procede a configurar nuestro bloque gain. (Véase la figura 3.63).

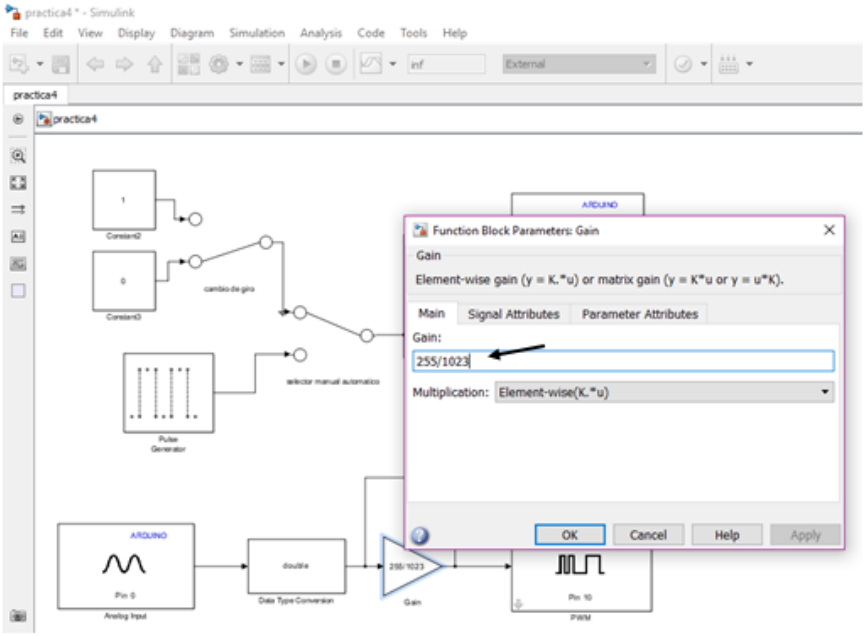


Figura 3. 63: Configuracion de bloque Gain.
Elaborado por: Autor.

- Y finalmente configuramos nuestras constantes. (Véase la figura 3.64).

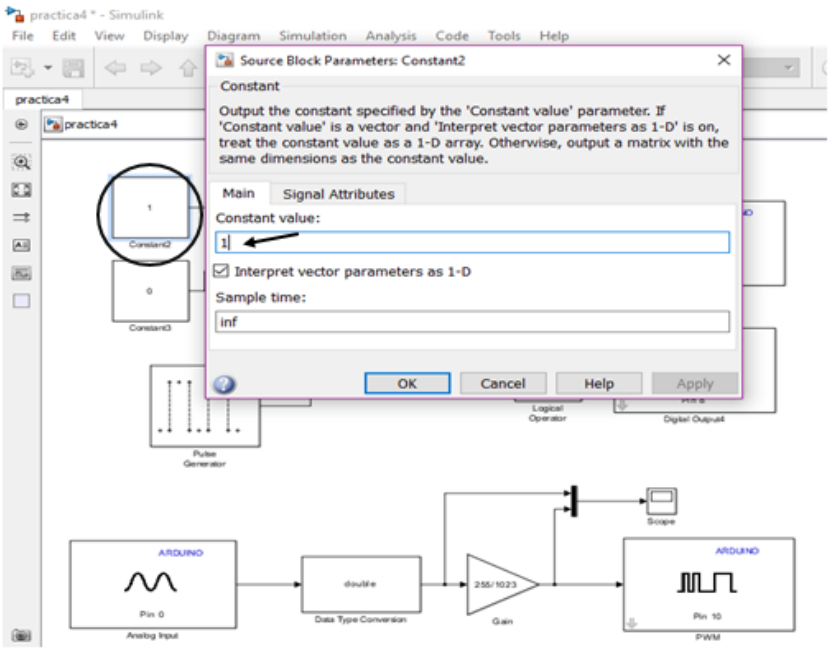


Figura 3. 64: Configuracion de constante1.
Elaborado por: Autor.

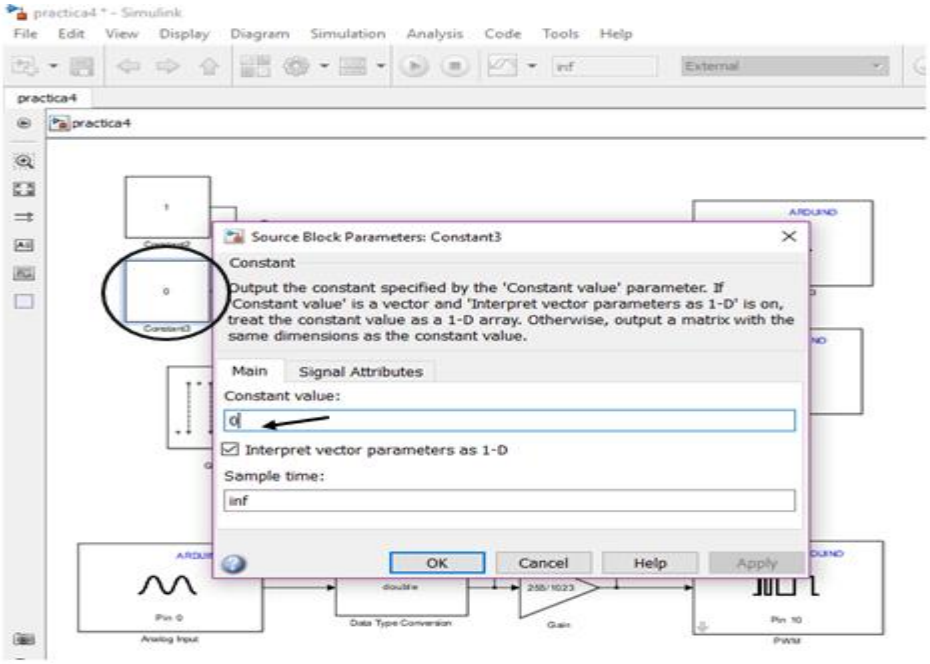


Figura 3. 65: Configuracion de constante 0.
Elaborado por: Autor.

- Una vez configurados todos los bloques de manera correcta se procede a hacer la simulación. (Véase la figura 3.66).

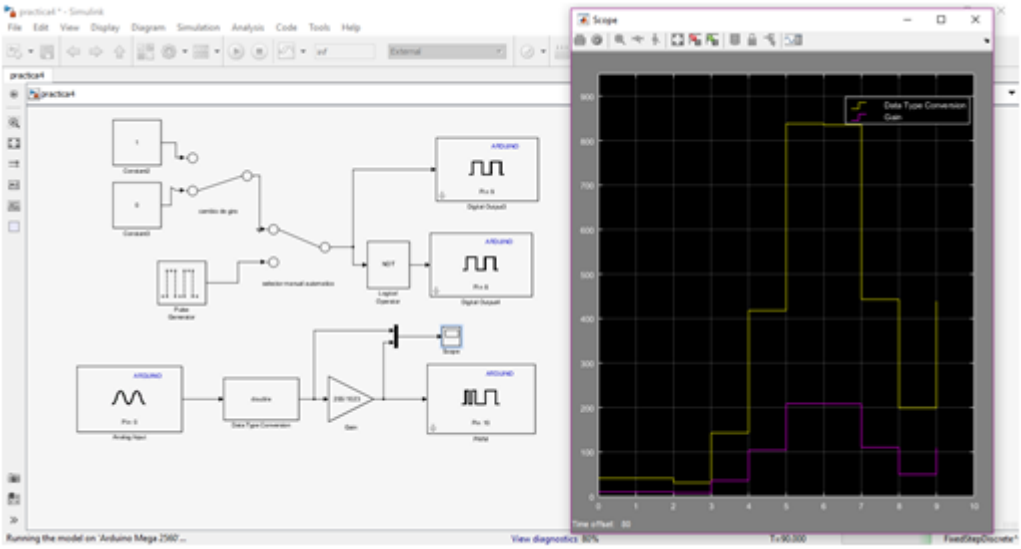


Figura 3. 66: Simulacion requerida.
Elaborado por: Autor.

Cuando giramos nuestro potenciometro podremos observar en la grafica que en la salida pwm que esta conectada al enable del integrado lm293d varia esto indica que el valor de voltaje que le ingresa hace que la

velocidad del motor aumente o disminuya. La otra funcion es el cambiar de guiro del motor , esto se logra dando doble click al switch conectado a las constantes cuando este en conectado a cero girara hacia un sentido y cuando este conectado a uno girara en el sentido contrario.

Y la ultima funcion es el cambio de giro automatico , esto se logra dando doble click sobre el segundo switch el cual nos permite seleccionar si deseamos controlar el cambio de giro manualmente con el primer switch o que cambie de giro según la señal que se configuro en el bloque de generador de señal de pulso.

3.5. Aplicación de práctica#5: control de led con la intensidad de luz

Objetivos

- Aprender a utilizar una foto resistencia.
- Crear circuitos que reaccionen al ambiente o señales físicas en este caso la luz.

Materiales

- foto resistencia
- 3 leds
- Cables
- Resistencia 1k
- Ardruino mega

Desarrollo

La práctica nos permite controlar el encendido de un juego de leds se el número que sea de leds dependiendo de la intensidad de luz que recepta nuestra foto resistencia dependiendo de cuál es el punto mínimo de luz que se configure para el encendido de los leds.

El diagrama de conexiones es el siguiente:

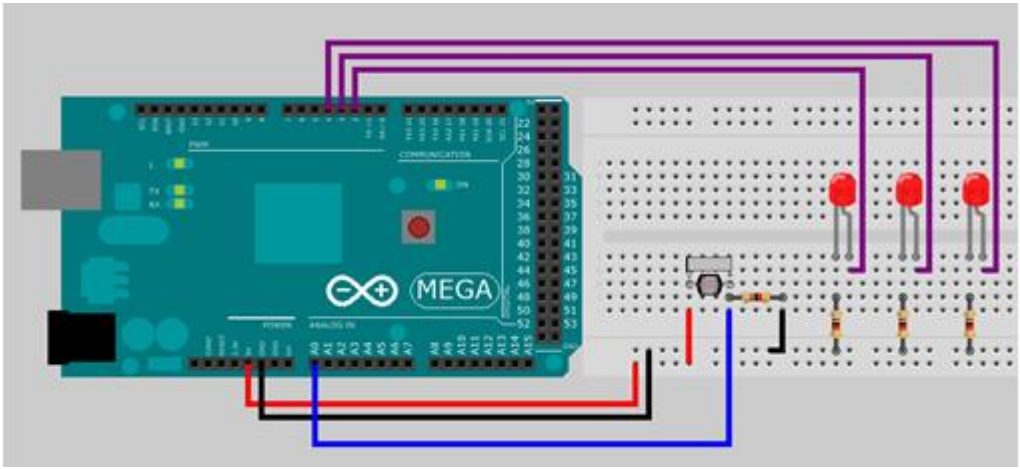


Figura 3. 67: Control de led
Elaborado por: Autor.

Programación

- Comenzamos agregando todos los bloques que utilizaremos

  Entrada analogica y salida digital del ardruino

  Bloque relational operator

  Constante. (Véase la figura 3.68).

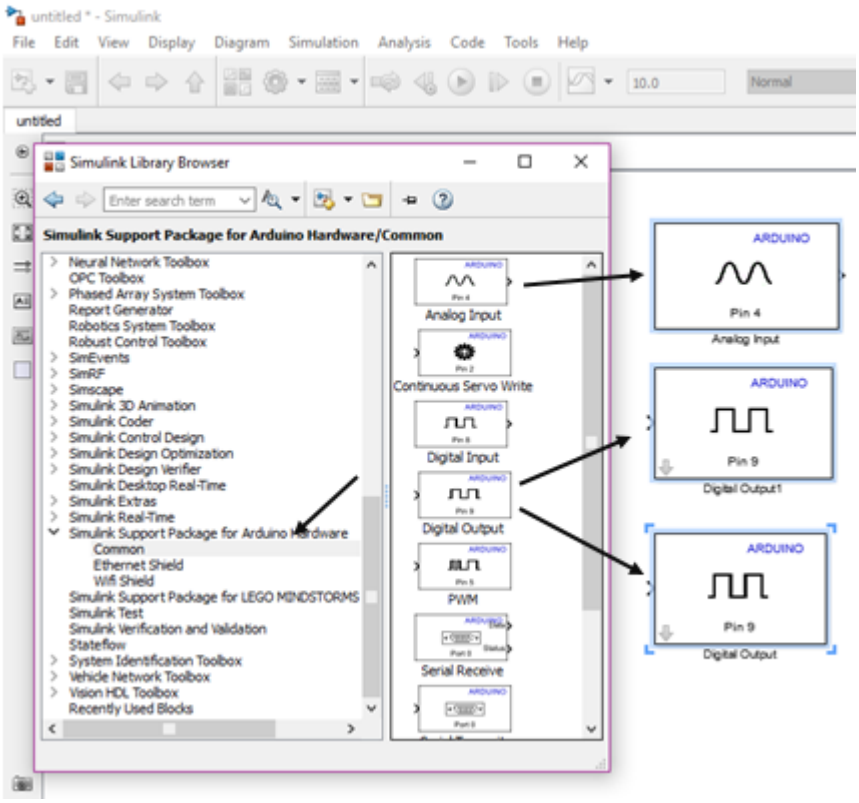


Figura 3. 68: Aplicación de bloque a utilizar pin.
Elaborado por: Autor.

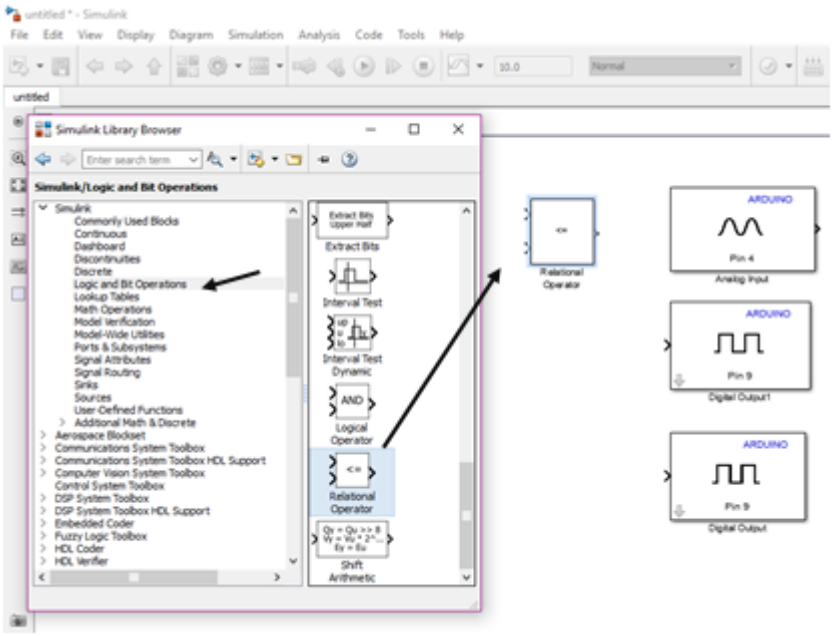


Figura 3. 69: Aplicación de bloque a utilizar.
Elaborado por: Autor.

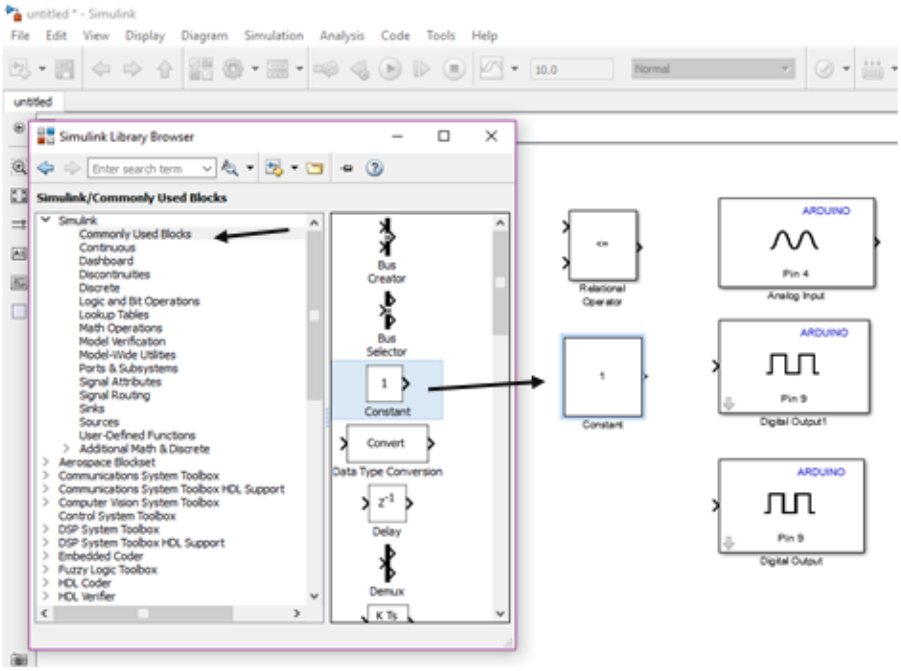


Figura 3. 70: Aplicación de bloque constante .
Elaborado por: Autor.

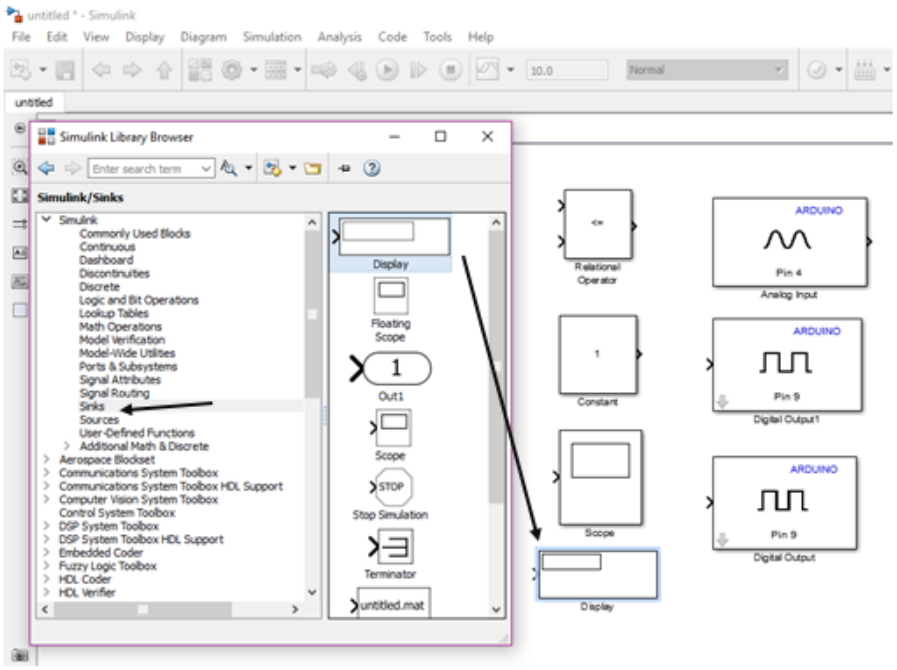


Figura 3. 71: Aplicación de bloque display.
Elaborado por: Autor.

- Continuamos con la conexión de nuestros bloques. (Véase la figura 3.72).

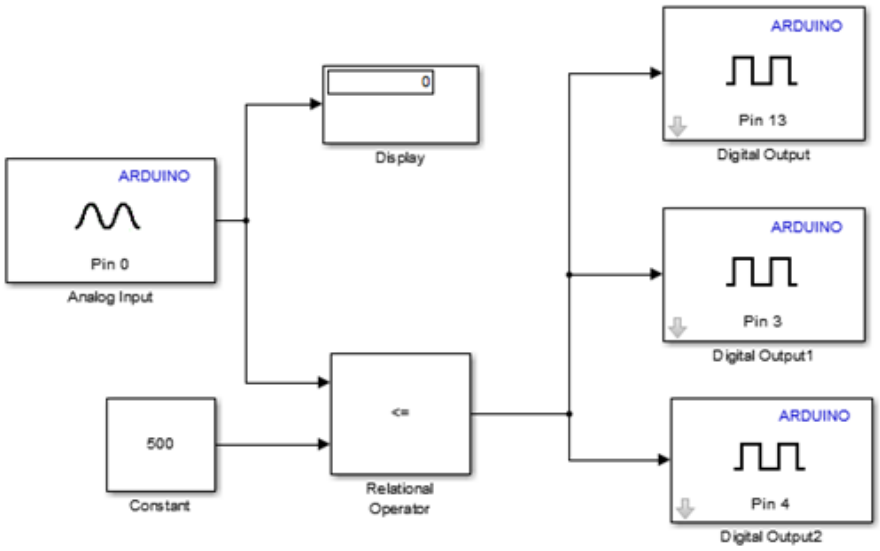


Figura 3. 72: Conexcion de bloques.
Elaborado por: Autor.

- Seguimos con la configuración de nuestros bloques
- Configuramos los bloques de salida y entrada del Arduino

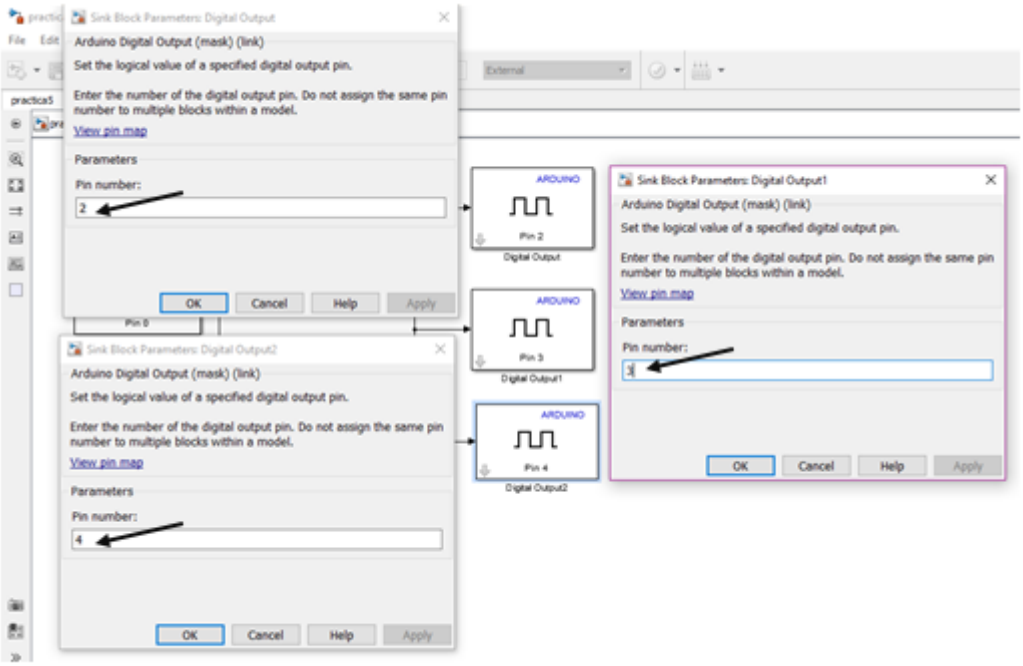


Figura 3. 73: Configuracion de bloque de salida y entrada.
Elaborado por: Autor.

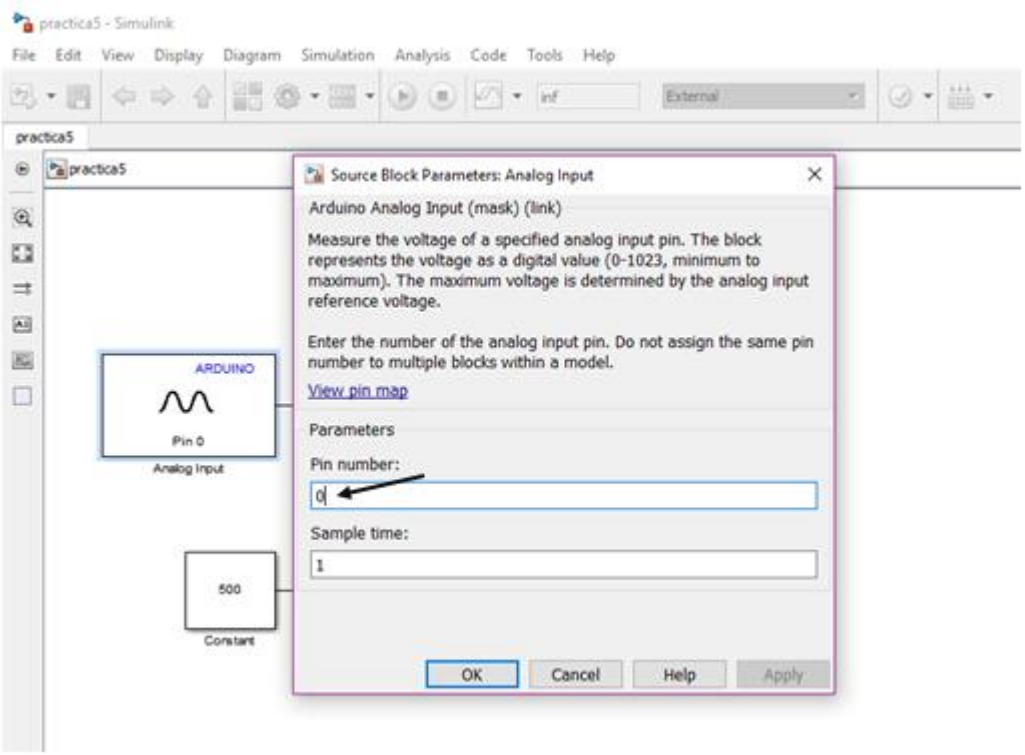


Figura 3. 74: Configuracion de bloques analogico.
Elaborado por: Autor.

Ahora configuramos nuestro bloque de constante en 500 este será el valor con el que se comparará la señal analógica recibida. (Véase la figura 3.75).

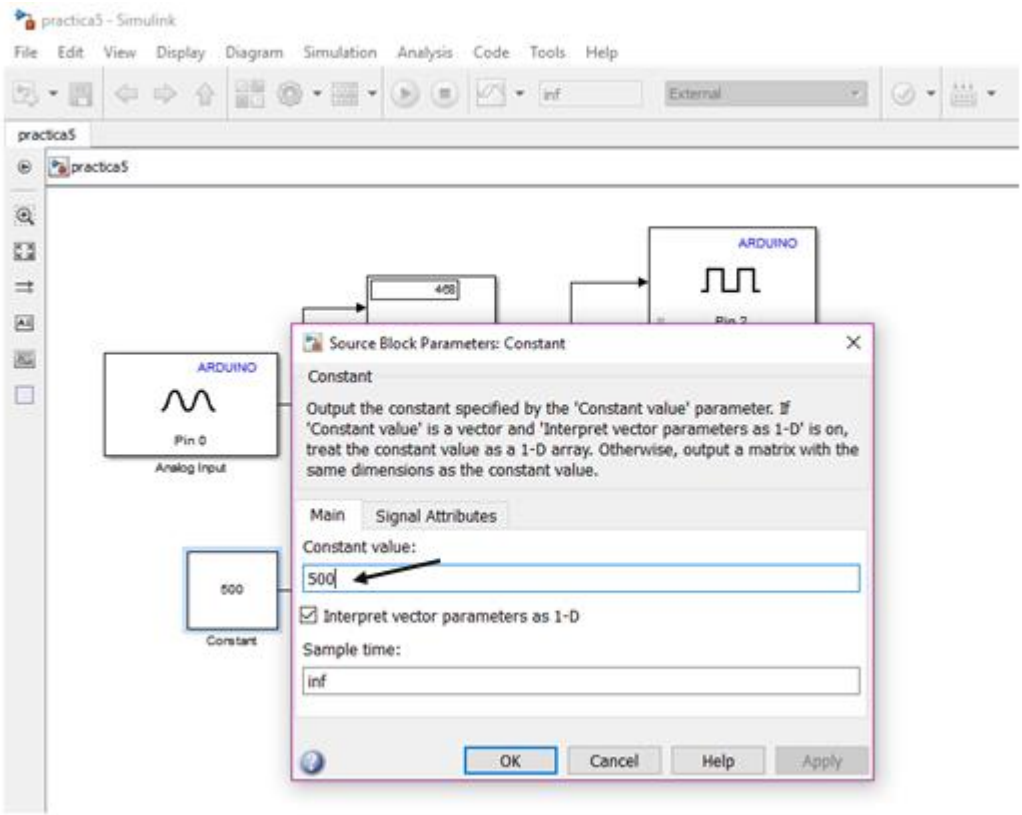


Figura 3. 75: Configuracion de bloques constante.
Elaborado por: Autor.

Luego configuramos nuestro bloque relational operator este hará la comparación de la señal de entrada con el valor constante antes definido, si el valor de la señal es menor o igual a la constante de 500 este bloque enviara una señal de uno lógico lo que nos indicara que se encenderán nuestros leds por medio de los bloques de salida digital y si la señal analógica es mayor a 500 enviara un cero lógico que apagara los leds.(el bloque se configura en la opción menor igual). (Véase la figura 3.76).

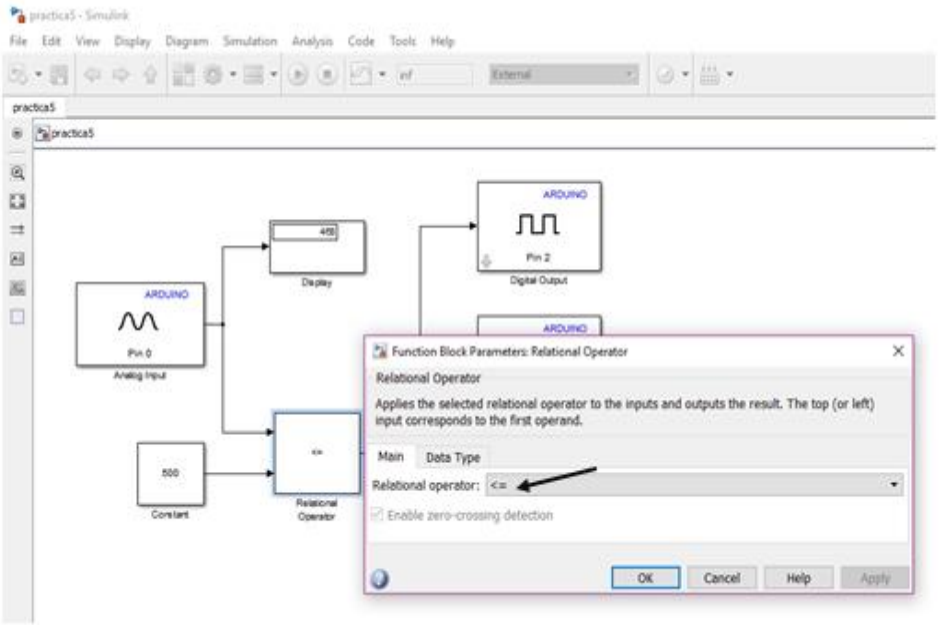


Figura 3. 76: Configuracion de bloque relational operator.
Elaborado por: Autor.

- Ahora podremos simular nuestro programa. (Véase la figura 3.77).

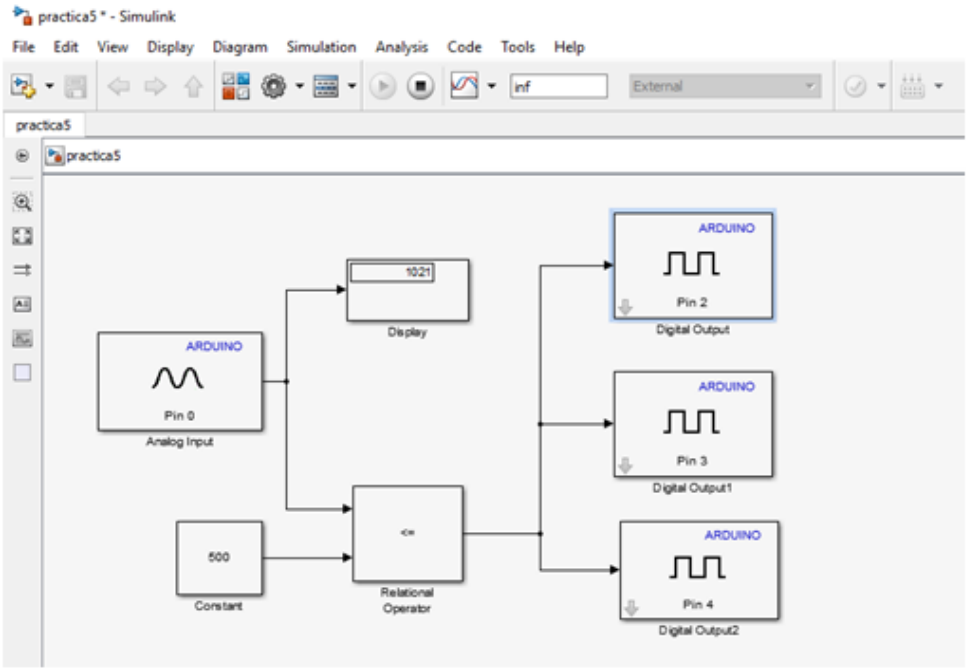


Figura 3. 77: Conexión de nuestro programa.
Elaborado por: Autor.

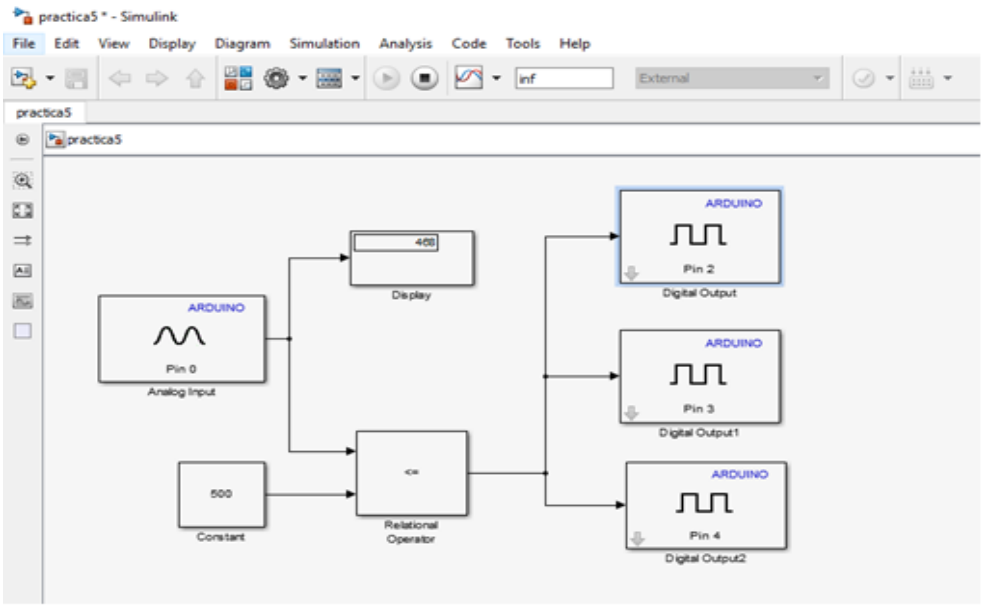


Figura 3. 78: Simulacion de programa.
Elaborado por: Autor.

En los displays podremos observar el valor que nos envía la foto resistencia, este valor será comparado con la constante de 500.

Cuando nos indica valores mayores a 500 en este caso 1021 podremos apreciar en nuestro circuito que los led se apagan, esto indica que hay suficiente luz para que no se enciendan los led, pero si nosotros obstruimos la luz que llega a la fotorresistencia el valor decaerá y si da igual o menor a 500 los led encenderán. (Véase la figura 3.78).

3.6. Aplicación de práctica#6; secuencia de encendido de leds

Objetivos

- Aprender a realizar secuencias de leds
- Aprender a utilizar flip flops en el programa Simulink

Materiales

- Cables
- 3 leds
- 3 resistencias 1k
- Ardruino mega

Desarrollo

La práctica consiste en realizar una secuencia de encendido de tres leds, con una configuración de tres flip flops, estos nos darán el tiempo de encendido de cada led.

El diagrama de conexiones es el siguiente:

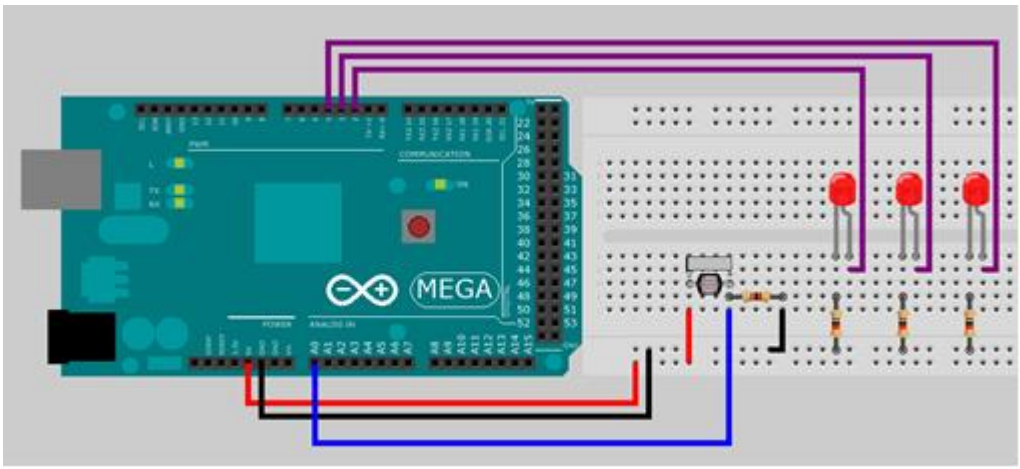


Figura 3. 79: Secuencia de encendido de led
Elaborado por: Autor.

- Comenzamos a agregar los bloques que usaremos
  - Salidas digitales de Arduino
  - Compuertas lógicas
  - Flip flops
  - Señal de reloj para los flip flops
  - Scope
  - Mux

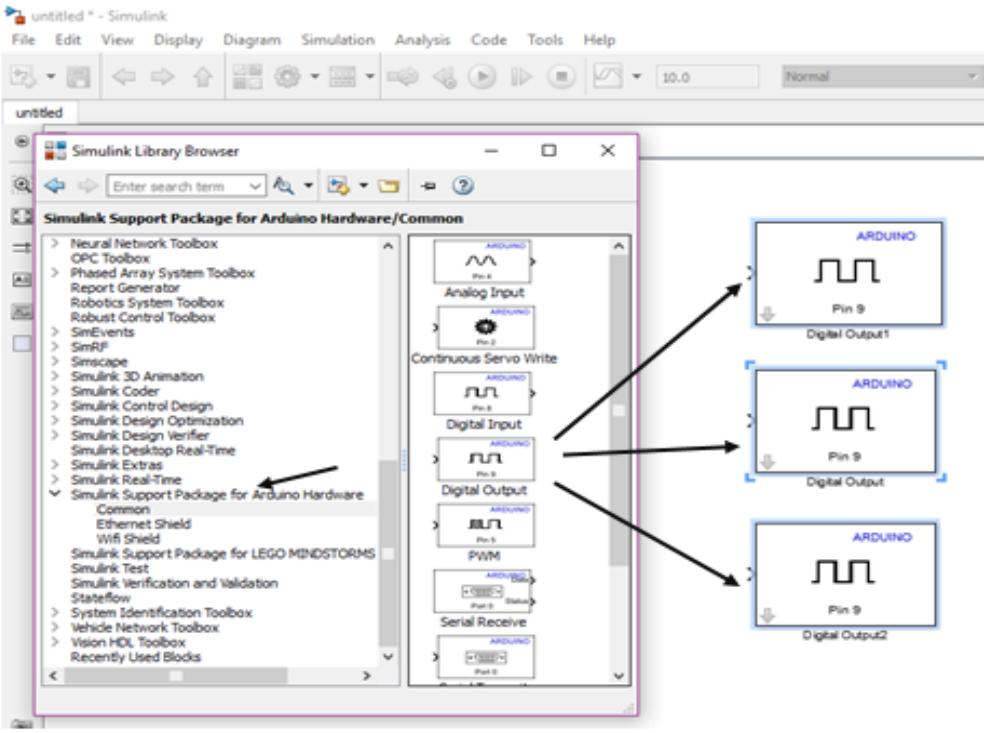


Figura 3. 80: Aplicación de bloques pines .
Elaborado por: Autor.

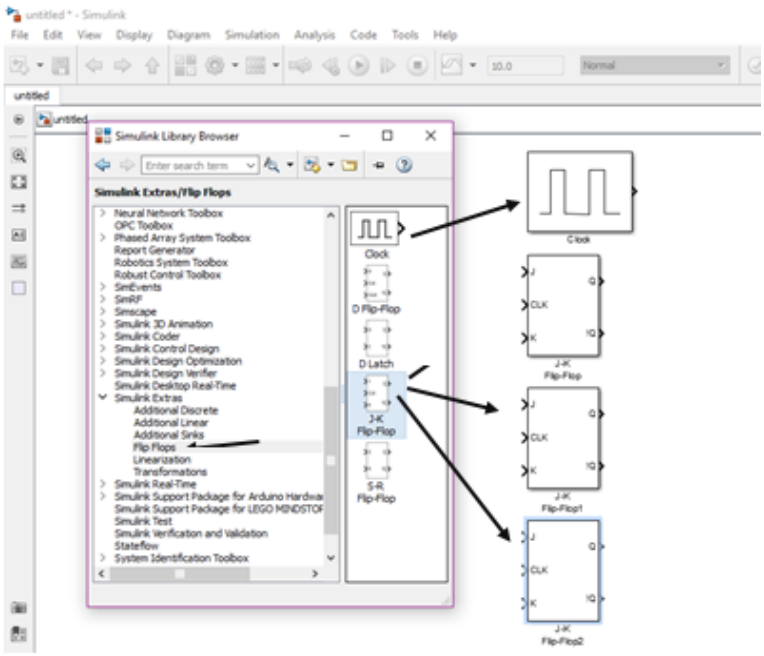


Figura 3. 81: Aplicación de bloques
Elaborado por: Autor.

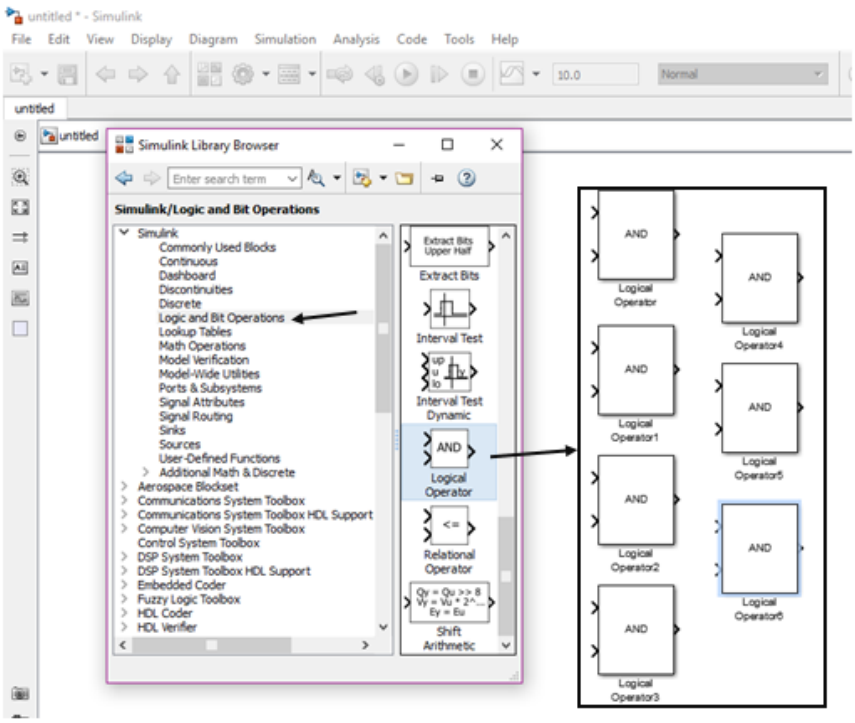


Figura 3. 82: Aplicación de bloques AND.
Elaborado por: Autor.

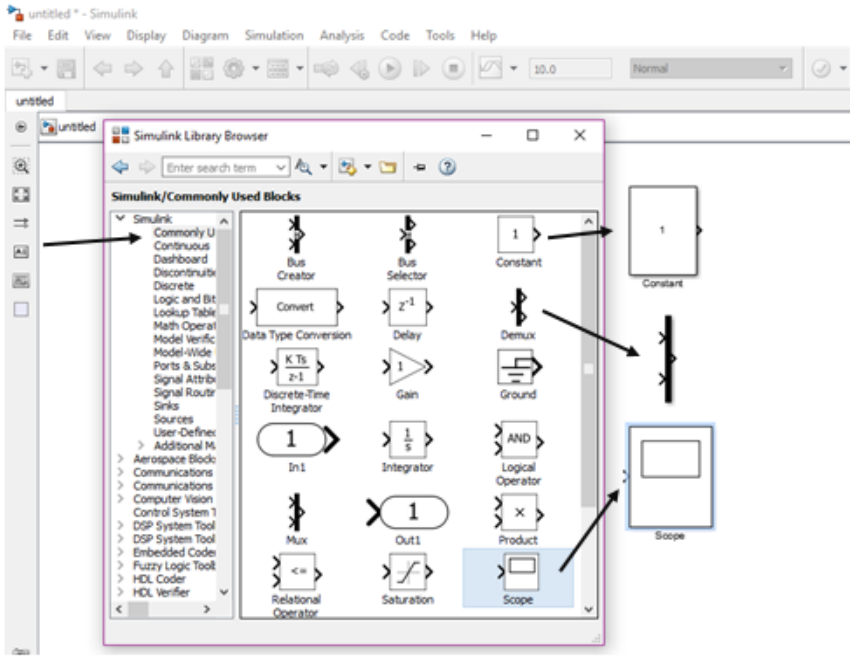


Figura 3. 83: Uso de aplicaciones de bloque scope.
Elaborado por: Autor.

- Ahora haremos nuestras conexiones. (Véase la figura 3.84).

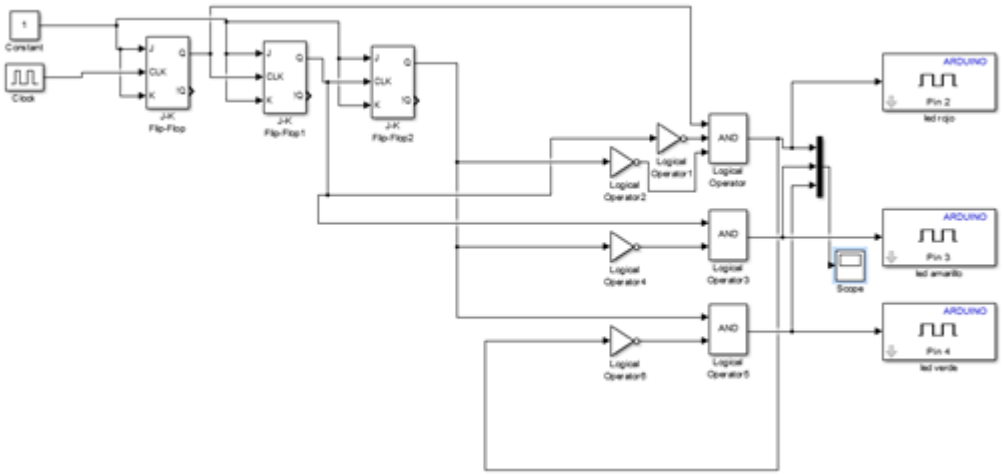


Figura 3. 84: Realizamos conexión de aplicaciones.
Elaborado por: Autor.

- Seguimos con la configuración de los bloques primero se configura el número de pin de las salidas analógicas. (Véase la figura 3.85).

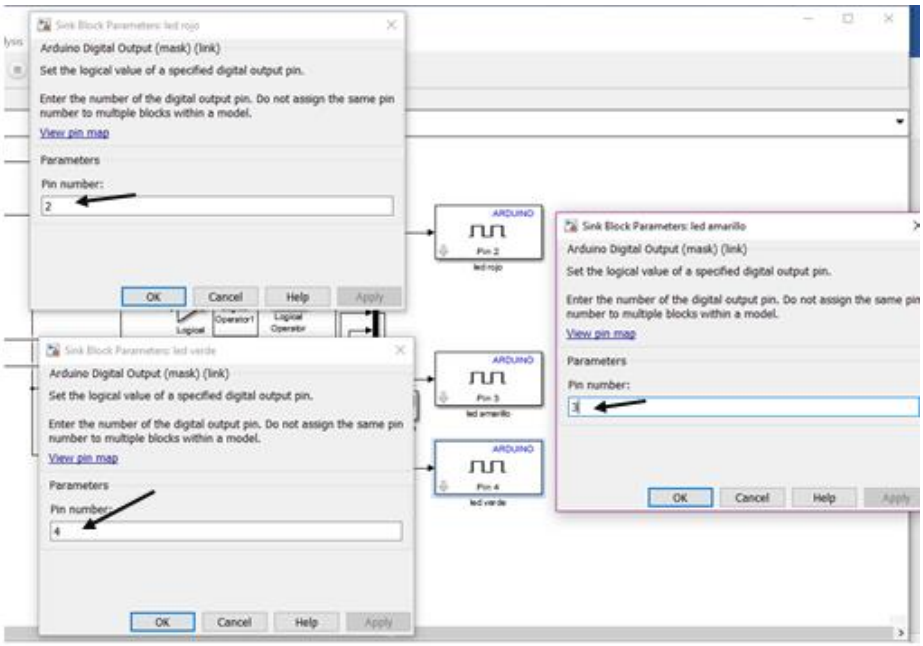


Figura 3. 85: Configuracion de salidas analogicas.
Elaborado por: Autor.

- Continuamos con la configuración de las compuertas lógicas cuatro compuertas se configurarán como not. (Véase la figura 3.86).

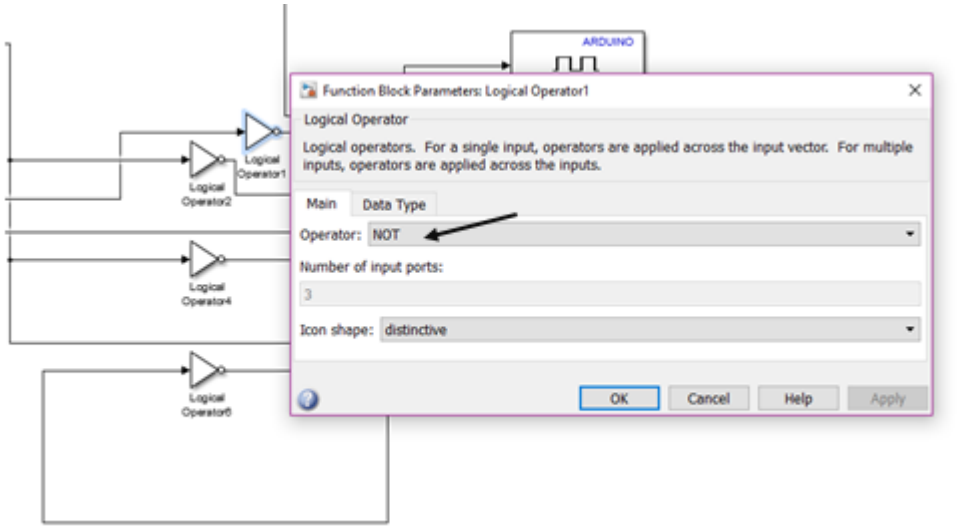


Figura 3. 86: Configuracion de la puerta NOT.
Elaborado por: Autor.

- Luego configuramos las compuertas and la primera se le configurarán 3 entradas las dos siguientes se configurarán con dos entradas. (Véase la figura 3.87).

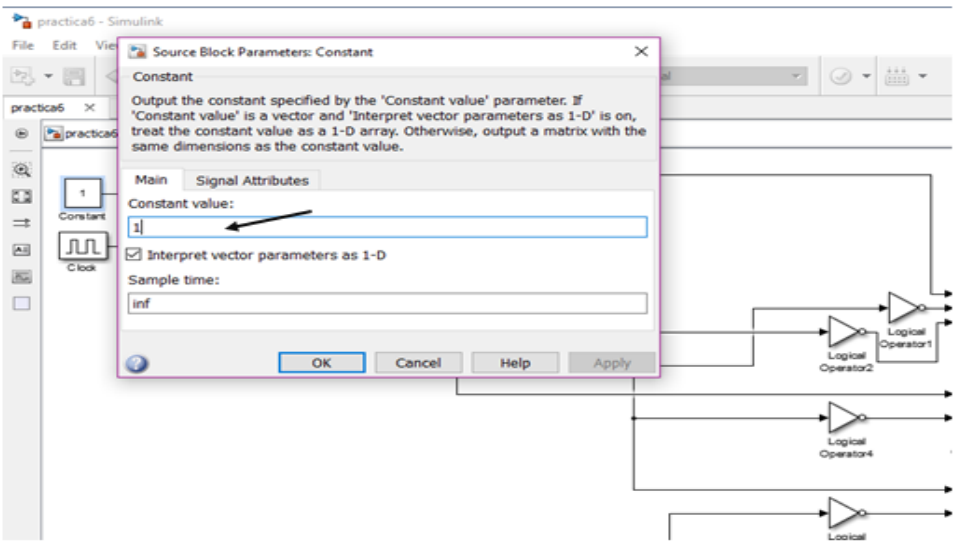


Figura 3. 87: Configuracion de constante.
Elaborado por: Autor.

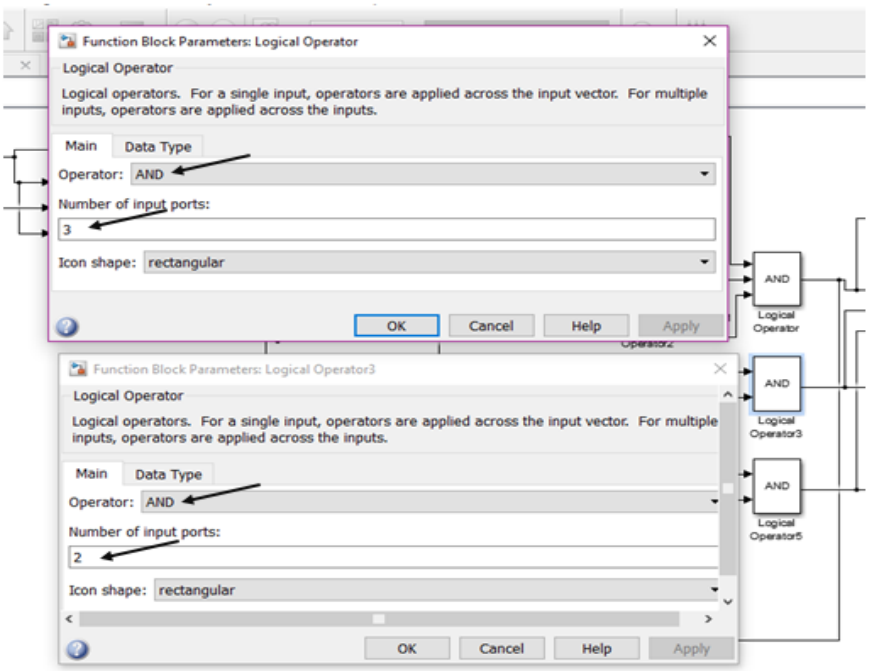


Figura 3. 88: Configuracion de puertas AND.
Elaborado por: Autor.

- Por ultimo configuraremos nuestra constante
- una vez hecha la configuración correcta de nuestros bloques procedemos a realizar la simulación

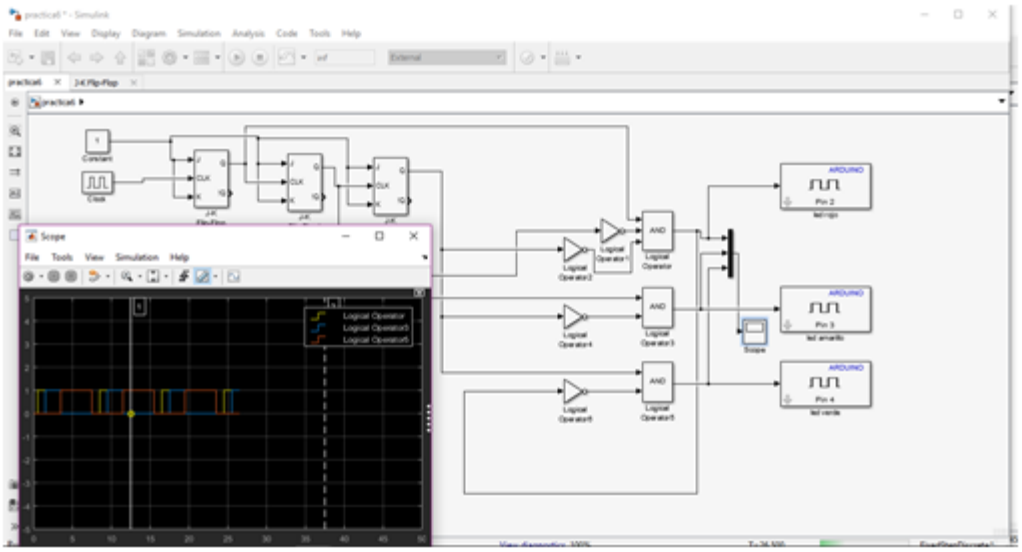


Figura 3. 89: Simulacion requerida.
Elaborado por: Autor.

Podremos observar que la primera salida tiene una amplitud pequeña, la segunda salida tiene el doble de la primera, y la tercera el doble de la segunda.

Estas señales son las que harán que se enciendan y apaguen los leds según el tiempo o la amplitud que tiene cada señal que sale de cada uno de los flip flops. (Véase la figura 3.89).

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.

4.1. Conclusiones.

Al término del presente trabajo de titulación y después de haber analizado y discutido los alcances obtenidos podemos llegar a las siguientes conclusiones:

- A través del uso de los microcontroladores podemos seguir proporcionando ejemplos aún más claros y concretos en el área de instrumentación virtual.
- Los procesos que se presenten en el área son muy específicamente a los de la materia instrumentación virtual, con procesos que se presentan dentro de la asignatura.
- Esta herramienta es una muy buena utilidad a futuro, ya que es una herramienta de diseño y que proveen a los alumnos y puede ser implementada en sus casas por lo que se genera un bajo costo.
- Durante las pruebas de ejecución en tiempo real de las aplicaciones prácticas virtuales se pudo constatar el correcto uso y funcionamiento de cada aplicaciones desarrolladas

4.2. Recomendaciones.

- Respecto a la herramienta plataforma de Arduino ya que genera un bajo costo, tras generar algunas limitaciones en su tarjeta de Arduino, y con su adquisición de datos permite generar una gran velocidad en el programa.

- Desarrollar nuevas propuestas de trabajos de titulación a la cuales nos permita utilizar sistema de Scada sobre la plataforma de instrumentación virtual.
- Impulsar licencia para Simulink para la información de profesionales de la carrera de ingeniería Electrónica en Control y Automatismo.

REFERENCIAS BIBLIOGRÁFICAS

Angulo usategui, jose. M., & Angulo Martinez, ignacio. (2003). pic diseño practico de aplicaciones, 355.

Barrett, S. F., & Pack, D. J. (2006). *Microcontrollers fundamentals for engineers and scientists* (1. ed). San Rafael, Calif.: Morgan & Claypool.

Khadse, R., Gawai, N., & Faruk, B. (2014). Overview and Comparative Study of Different Microcontrollers. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, *2*(12), 311–315.

Varesano, fabio. (2011). Using Arduino for Tangible Human Computer Interaction. *ABRIL*, *2*, 171.

Pérez, F. E. V., &Areny, R. P. (2007). Microcontroladores: fundamentos y aplicaciones con PIC. Marcombo.

Breijo, E. G. (2012). Compilador C CCS y simulador Proteus para microcontroladores PIC. Marcombo. Recuperado a partir de https://books.google.es/books?hl=en&lr=&id=k8vMlKuRAyUC&oi=fnd&pg=P
P14&dq=microcontrolador&ots=JgCccQWBAq&sig=ugLe62ddyMdeJBlJTKH uqUOSfZw

Verle, M. (s/f). 1.1 Introducción al Mundo de los microcontroladores | Microcontroladores PIC – Programación en BASIC. Recuperado a partir de http://learn.mikroe.com/ebooks/microcontroladorespicbasic/chapter/introduccio n-al-mundo-de-los-microcontroladores/

Microchip Technology Inc. (2016). Home | Microchip Technology Inc. Recuperado el 19 de noviembre de 2016, a partir de http://www.microchip.com/

Los microcontroladores de hoy en día. (2012, diciembre 2). Recuperado a partir de https://microcontroladoressesv.wordpress.com/los-microcontroladores-de-hoy- en-dia/

MacKenzie, I. S., & Phan, R. C. W. (1999). The 8051 microcontroller (Vol. 3). Prentice Hall.

Koutroulis, E., Kalaitzakis, K., & Voulgaris, N. C. (2001). Development of a microcontroller-based, photovoltaic maximum power point tracking control system. IEEE Transactions on power electronics, 16(1), 46-54.

Iovine, J. (2004). PIC microcontroller project book. McGraw-Hill, Inc..

Nemecek, C. (2005). U.S. Patent No. 6,922,821. Washington, DC: U.S. Patent and Trademark Office.

Pahuja, R., & Kumar, N. (2014). Android Mobile Phone Controlled Bluetooth Robot Using 8051 Microcontroller. International Journal of Scientific Engineering and Research, 2(7).

Mon, Y. (2015). The Bluetooth based LED control for Arduino test platform by using mobile APP. International journal of scientific & technology research, 4(6), 330-332.

Schubert, T. W., D'Ausilio, A., & Canto, R. (2013). Using Arduino microcontroller boards to measure response latencies. Behavior research methods, 45(4), 1332-1346.

Monk, S. (2013). *30 Arduino projects for the evil genius*. McGraw-Hill Professional.

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Zambrano Alcívar, Ángel David** con C.C: # 0931441430 autor del Trabajo de Titulación: **USO DE LA PLATAFORMA ARDUINO Y SIMULINK PARA EL DESARROLLO DE APLICACIONES PRACTICAS PARA INSTRUMENTACION VIRTUAL.** Previo a la obtención del título de **INGENIERO ELECTRONICA EN CONTROL Y AUTOMATISMO** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 14 de Septiembre de 2017

f. _____

Nombre: Zambrano Alcívar, Ángel David

C.C: 0931441430

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN			
TÍTULO Y SUBTÍTULO:	**USO DE LA PLATAFORMA ARDUINO Y SIMULINK PARA EL DESARROLLO DE APLICACIONES PRACTICAS PARA INSTRUMENTACIÓN VIRTUAL**		
AUTOR(ES)	Zambrano Alcívar, Ángel David		
REVISOR(ES)/TUTOR(ES)	M. Sc. Romero Paz, Manuel De Jesús		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería electrónica en control y automatismo		
TITULO OBTENIDO:	Ingeniería electrónica en control y automatismo		
FECHA DE PUBLICACIÓN:	14 de Septiembre de 2017	**No. DE PÁGINAS:**	88
ÁREAS TEMÁTICAS:	Instrumentación Virtual		
PALABRAS CLAVES/ KEYWORDS:	INSTRUMENTACION VIRTUAL, ARDUINO, SIMULINK, MICROCONTROLADORES,		

RESUMEN/ABSTRACT (150-250 palabras):

El presente trabajo de titulación hace referencia al tema del Uso del plataforma y Simulink para el desarrollo de aplicaciones prácticas para instrumentación virtual, para lo cual hemos trabajado con MatLab mediante Arduino Simulink basados en prácticas como por ejemplo Modulación PWE con entrada analógicas, encendido de led, lo que nos permite verificar la regulación del led mediante un pulso enviado desde el programa que vamos a utilizar que es simulink .En este proceso de titulación también hacemos referencia a los tipos de microcontroladores utilizados, el sistema que se desarrolló cuanta con los dispositivos esquemáticos para realizarlos mediantes prácticas en la UCSG En este trabajo de titulación ha llegado experimentar muchos ámbitos y conocimientos prácticas y simulados. La metodología utilizada fue de fuentes bibliográficas y experimentales, las cuales nos permitió analizar los aspectos fundamentales y teóricos acerca al uso de la plataforma Arduino y simulink para el desarrollo de aplicaciones de la instrumentación virtual. Se ha realizado una descripción general de los fundamentos básicos de la familia de los microcontroladores así también de la plataforma de desarrollo de Arduino IDE.Posteriormente se desarrollaron seis aplicaciones prácticas en la que se integran el diseño de la instrumentación virtual, y su respectiva simulación.

ADJUNTO PDF:	☒ SI	☐ NO
CONTACTO CON AUTOR/ES:	**Teléfono:** +593-99-770-3626	E-mail: angel_z1994@hotmail.com
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	**Nombre: ROMERO PAZ, MANUEL DE JESUS**	
	Teléfono: +593-99-460-6932	
	E-mail:	

SECCIÓN PARA USO DE BIBLIOTECA	
Nº. DE REGISTRO (en base a datos):	
Nº. DE CLASIFICACIÓN:	
DIRECCIÓN URL (tesis en la web):	