UNIVERSIDAD SERGIO ARBOLEDA

Ingeniería Electrónica
Informe de Laboratorio
Curso: Herramientas de Diseño Electrónico y Sistemas Digitales II. Proyecto de Aula. Grupo #:1

PROYECTO DE AULA

Cristian Andrés Montaño
cristian.montano@correo.usa.edu.co

Manuel Salgado Olmos
manuel.salgado@correo.usa.edu.co

Jhonatan Rodríguez Gamba
jhonatan.rodriguez@correo.usa.edu.co

Miguel Ángel Ruiz Torres
miguelan.ruiz@correo.usa.edu.co

Resumen: El proyecto de aula consiste en configurar un Arduino Due como maestro a través de Atmel Studio con protocolo UART y tres arduinos más(Uno,Mega) como esclavos por medio del entorno Arduino con el protocolo I2C, de tal forma que el maestro lea las variables de sus esclavos y los esclavos controlen actuadores y sensores, mostrandolos en una interfaz gráfica realizada en python.

I. MARCO TEÓRICO

ATMEL STUDIO

Studio 7 es la plataforma de desarrollo integrado (IDP) para desarrollar y depurar todas las aplicaciones de microcontroladores AVR® y SAM[1].RECURSOS UTILIZADOS

TRANSMISIÓN DIGITAL

La transmisión digital consiste en el envío de información a través de medios de comunicaciones físicos en forma de señales digitales. Por lo tanto, las señales analógicas deben ser digitalizadas antes de ser transmitidas[2].

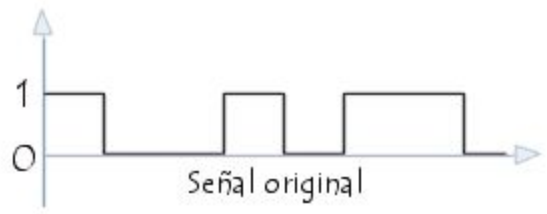


figura 1: Imagen de la transmisión digital.

UART

(Universal Asynchronous Receiver-Transmitter), Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del microcontrolador, el UART toma bytes de datos y transmite los bits individuales de forma secuencial[3].

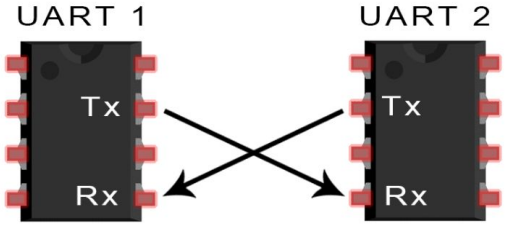


figura 2: Imagen de conexión del UART.

MODELO OSI

El modelo osi es una arquitectura para comunicaciones entre dispositivos desarrollada por la organización internacional de estandarización. Este modelo se compone de 7 capas las cuales se muestran en la siguiente imagen, cada una con su explicación[4].

figura 3: Capas del modelo osi

8N1

Es una abreviatura habitual de la configuración del puerto serie en modo asíncrono donde hay (8) bits de datos, ningún (N) bit de paridad, y un (1) bit de parada[5].

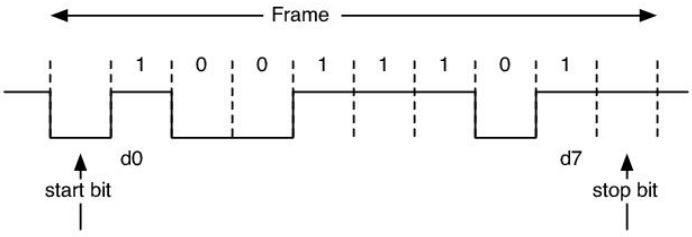


figura 4: Transmisión 8N1.

HyperTerminal

Es un programa que se puede utilizar para conectar con otros equipos, sitios Telnet, sistemas de boletines electrónicos (BBS, Bulletin Board Systems), servicios en línea y equipos host, mediante un módem, un cable de módem nulo o una conexión[6].

figura 5: Interfaz de HyperTerminal.

I2C

Es un protocolo serie desarrollado por Philips Semiconductors,es utilizado por muchos dispositivos integrados para comunicarse entre ellos, es un protocolo sencillo ya que para su funcionamiento solo requiere dos líneas, una de reloj (SCL) y otra de datos (SDA). Es un protocolo maestro esclavo en el que el maestro inicia o finaliza la comunicación y debe generar una señal de reloj, la línea de datos es bidireccional (el maestro puede mandar o recibir). Cada bit que se envía por la línea de datos lo acompaña un pulso de reloj por la línea SCL, el valor del bit se toma en la parte "alta" del reloj. [7][8]

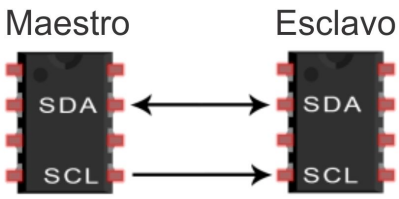


figura 6: Conexiones del protocolo I2C.

A. *Software:* Atmel Studio 7, Hyperterminal, Aduino.

B. *Componentes:* Protoboard, Cable para protoboard. Arduino DUE, Arduino UNO, Arduino MEGA, Resistencias, LEDs, pulsadores, potenciómetros, motor DC, sensor MQ.

C. *Equipos:* Computador.

II. PROCEDIMIENTO

A. *Planteamiento del Problema:*

Primero deberemos declarar los registros a usar en el programa ATMEL, los cuales a grandes rasgos serán:

Registros timer
Es la base para que nuestro proyecto funcione como una máquina de estado ya que habilita en que momento se deben recibir datos de los esclavos y en qué momento se deben enviar.

*t_ccr=0x5;// En el canal 1_0 se habilita el timer y un reinicio de inicializacion

*t_mrc=0x0;// Timer usa la frecuencia más grande o Timer Clock 1

Registros PIO

Deshabilita las entradas y salidas para volverlas periféricos.

*pio_pdra=0xFFFFFFFF; //Deshabilita el registro de salida para habilitarlo como periférico

*pio_pudra=0xFFFFFFFF;// Deshabilita el Pull Up

*pio_absra=0x00000000;//Selecciona el tipo de periférico A o B

Registros UART

*uart_cr=0x50;// UART control register, habilita o deshabilita la entrada o salida del dispositivo UART

*uart_mr=0x0800;// UART mode register, selecciona el tipo de paridad y el modo del canal

*uart_brgr=0x230;// Se configura el reloj del UART, a 9600 bds

Registros PMC

*pmc_pcer=0x10C00100; //Habilitamos los periféricos del UART, TWI y Timer en el PMC_0

Registros TWI

*twi_cwgr=0x70808;// Velocidad de bits a usarse por la comunicación I2C

B. *Diseño:*

Diseño para el esclavo 1 (Arduino uno)

Este esclavo recibirá datos de un sensor de gas MQ-7 el cual recolecta datos del ambiente que serán enviados por el puerto A0 y posteriormente al arduino DUE.

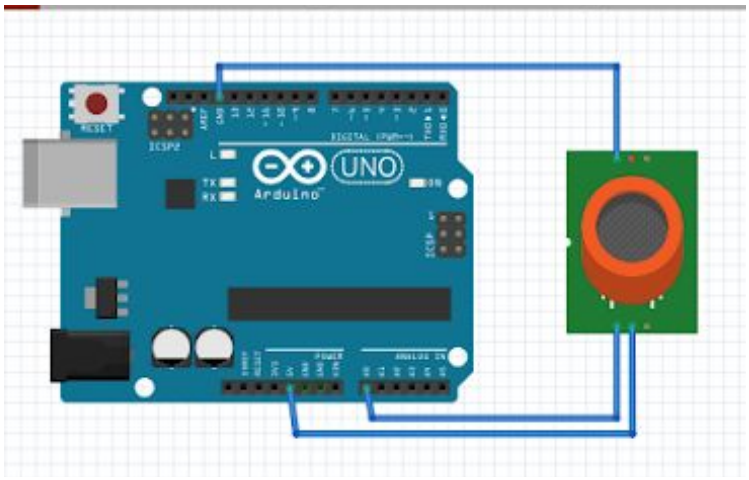


Figura 7: Conexiones del esclavo uno con el sensor MQ.

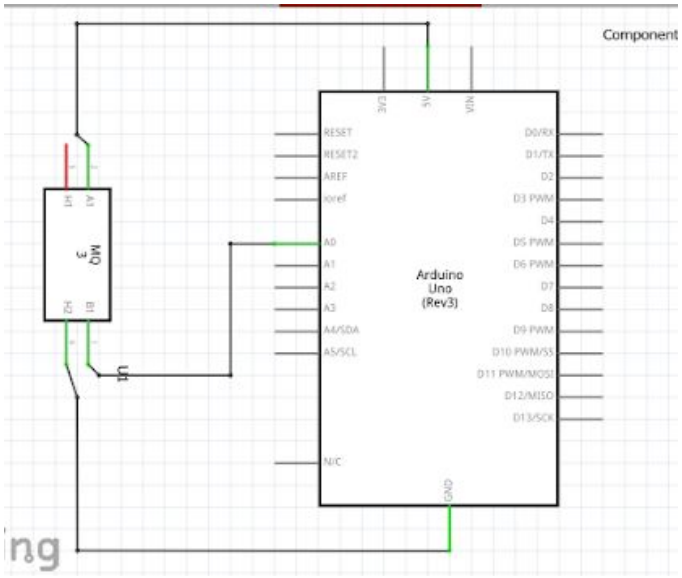


Figura 8: Esquemático de las conexiones del esclavo 1.

Diseño para el esclavo 2 (Arduino uno)

Este esclavo recibirá datos de un potenciómetro el cual recolecta datos que serán enviados por el puerto A0 y posteriormente al arduino DUE.

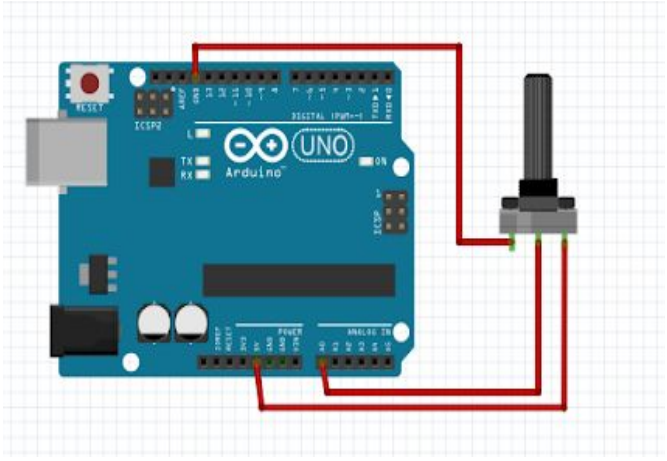


Figura 9: Conexiones del esclavo 2 con el potenciómetro.

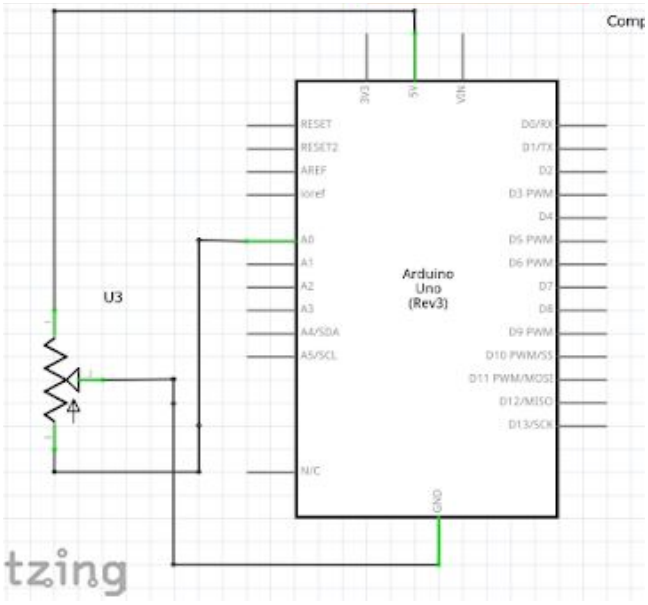


Figura 10: Esquemático del esclavo 2.

Diseño para esclavo 3 (Arduino mega)

Este arduino tendrá dos salidas una que lleva a 3 leds que se activan por medio del arduino DUE y la otra lleva a un motor el cual se controla por pwm igualmente a través del arduino maestro (DUE), finalmente tendrá una entrada perteneciente a un pulsador que activará un cuadro en la ventana gráfica de python.

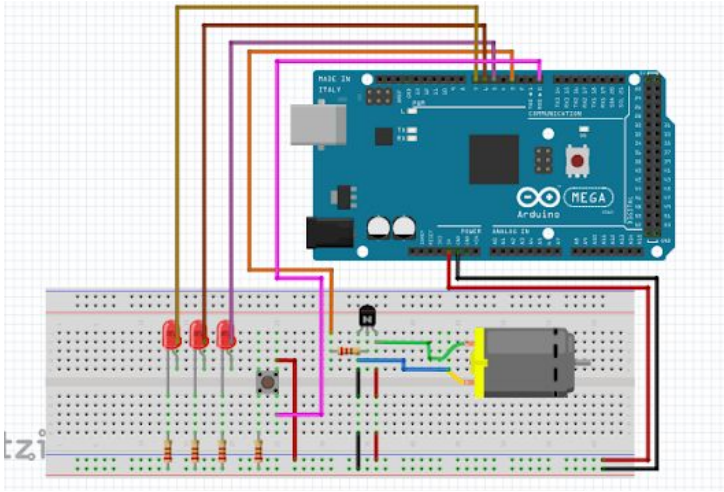


Figura 11: Conexiones del esclavo 3 con el motor dc, pulsador y leds.

La interfaz gráfica se realizó en Python y muestra las gráficas de los dos dispositivos análogos (Potenciómetro y sensor de gas MQ), también se muestra un apartado para ingresar texto donde se controla la velocidad del motor por medio de PWM y permite encender LEDS externos, el cuadro verde es la representación de un led que se pondrá de ese color si se oprime un pulsador anclado al arduino mega, si no se oprime pasa a gris. La interfaz gráfica se muestra en el ANEXO 2, el código para esta interfaz se presenta a continuación: CODIGO

A continuación se adjunta el código perteneciente al esclavo arduino mega: CODIGO. en este código el apartado de "salida digital" nos indica que leds se deben prender dependiendo de qué número se le envíe por medio de la interfaz gráfica, también se encuentra el código para controlar el pwm, cuya salida es la número 3. "Entrada digital" nos recibe un pulso del pulsador (1 o 0) para ser representado luego en la interfaz gráfica.

Para el funcionamiento del esclavo arduino uno se utilizó el siguiente código: CODIGO para este solo se utilizo el apartado de entrada digital, que en realidad es una entrada análoga y aquí va conectado un potenciómetro.

En el esclavo tres se utilizó el mismo procedimiento del esclavo 2 pero para un sensor de gas MQ: CODIGO en este código se puede omitir la parte de salida digital.

C. *Montaje*:

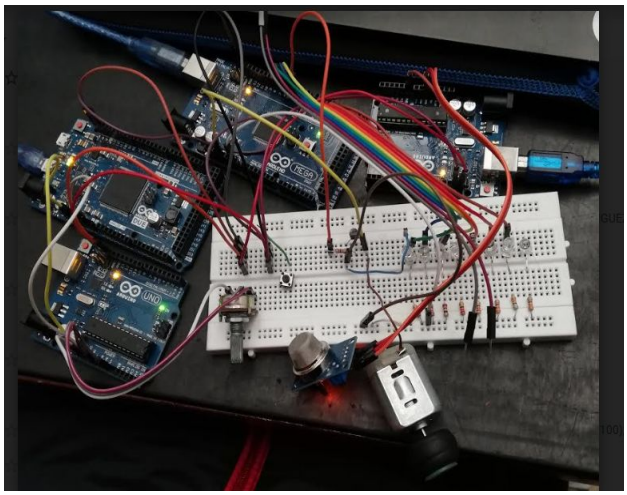


figura 12: Montaje del circuito.

D. Análisis de Resultados

Luego de recibir los datos provenientes de los esclavos,el maestro mediante la UART(en hyperterminal) interpreta estos datos como lenguaje ASCII, de la siguiente forma:

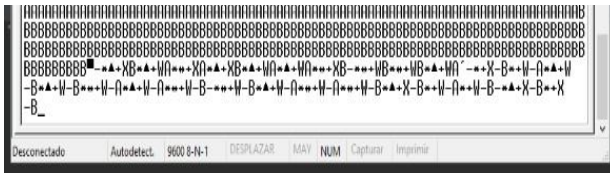


figura 13: Datos recibidos de los esclavos, mostrados en Hyperterminal.

Para interpretar de forma correcta y por separado dichos datos y poder mostrarlos de alguna forma en la interfaz de python, es necesario enviar los datos con unos respectivos indicadores, en nuestro caso usaremos como indicadores los símbolos : “+”, “-” y “*”.

Ya teniendo estos valores llegando a python por medio del puerto notamos que estos se dan en Hexa, así que es necesario convertirlos en un valor entero en python para poder interpretar los valores y generar las respectivas gráficas de las entradas análogas.

III. CONCLUSIONES

- Al no sincronizar adecuadamente el timer que usamos para escritura con las operaciones de lectura en el UART, nos presentaba pérdida de datos al momento de leer.
- El contador del timer es reiniciado cada que cumple un periodo de 9 segundos, ya que el conteo de este registro puede alcanzar hasta los 100 segundos según la frecuencia implementada.

Personales:

- Hacer uso adecuado del tiempo de clase para no dejar los trabajos hasta último momento y tener muy claras las fechas de presentación.

IV. REFERENCIAS

[1] Microchip. (2018, junio). Atmel Studio. [Online] Available: https://www.microchip.com/mplab/avr-support/atmel-studio-7

[2] CCM. (2017, diciembre, 18) Transmisión digital. [Online] Available:https://es.ccm.net/contents/690-transmision-de-datos-transmision-digital-de-datos

[3] rinconingenieril. (2017.octubre,31). UART. [Online] Available:https://www.rinconingenieril.es/funciona-puerto-serie-la-uart/

[4] W. Stallings, Comunicaciones y redes de computadores, 6th ed. Pearson Educación de México, S.A. de C.V., 2011, pp. 18,19.

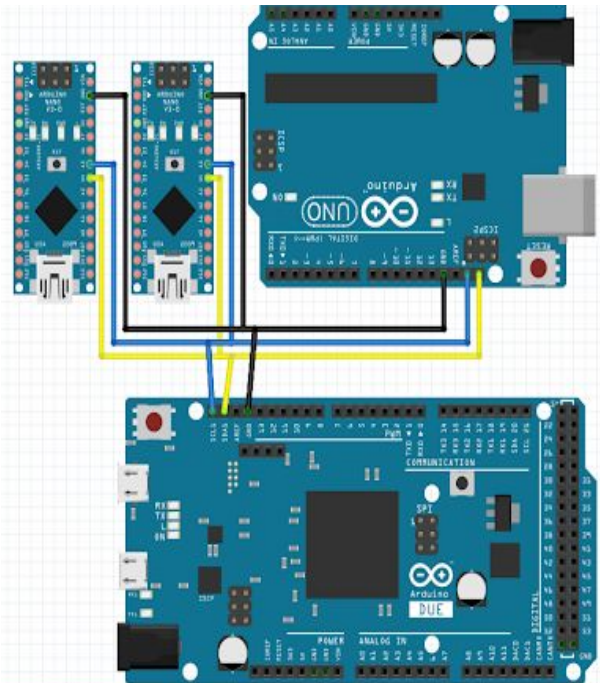
[5] Wikipedia. (). 8N1. [Online] Available:https://es.wikipedia.org/wiki/8N1

[6] redesparacomputadores. (2010,octubre,29). Hyperterminal. [Online] Available:http://redesparacomputadores-lubian.blogspot.com/2010/10/hyperterminal.html

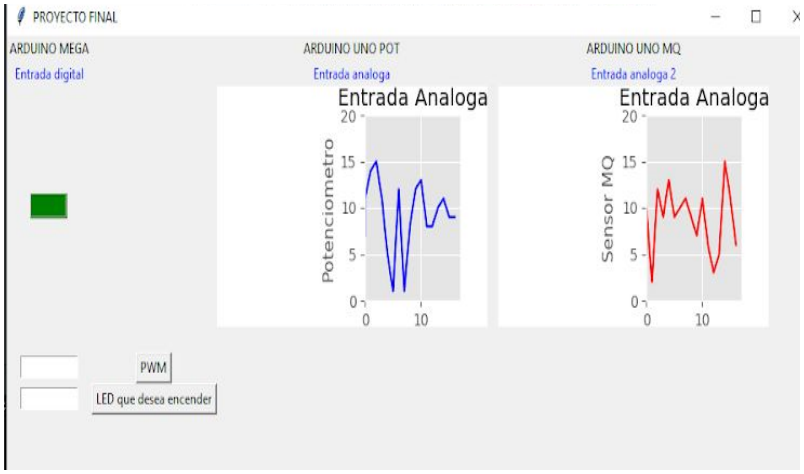
[7] Philips Semiconductors, "APPLICATION NOTE I2C BUS", San Jose, CA, 2003.

[8] K. Hemmanur, "Inter-Integrated Circuit (I2C)", 2009.

V. ANEXOS



Anexo 1. Diagrama de conexiones I2C



Anexo 2. Interfaz gráfica python