Práctica 2 - Señales
Interfaces y periféricos para robots

Roberto Cadena Vega

Objetivos

Familiarizarse con los diferentes tipos de señales a manejarse para el control de sistemas automáticos.

Conocimientos Previos

Señales digitales

Empezaremos esta práctica analizando la manera en que la tarjeta de desarrollo de Arduino transmite información a la computadora y viceversa.

Las mediciones que se muestran se hicieron conectando la tarjeta de desarrollo de Arduino a un osciloscopio casero como se muestra en la figura 1, sin embargo tu tienes la oportunidad de seguir estas mediciones con un osciloscopio profesional pidiendo este equipo al personal de laboratorio, por lo que te sugiero que empieces con estas mediciones, antes de seguir con la parte importante de la práctica.

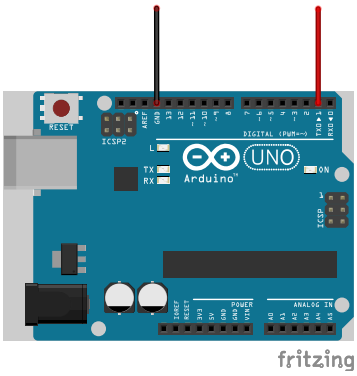

Figura 1: Conexion de Transmisión de datos del Arduino UNO

Se empieza con el siguiente código cargado en la tarjeta de desarrollo:

```
1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   Serial.println(1);
7   delay(2);
8 }
```

En este código se inicializa el puerto serial con una tasa de transmisión de $9600 baudios$, se envía el dato 1, seguido del terminador de linea (retorno de linea, enter) y espera $2ms$ para estabilizar el μC.

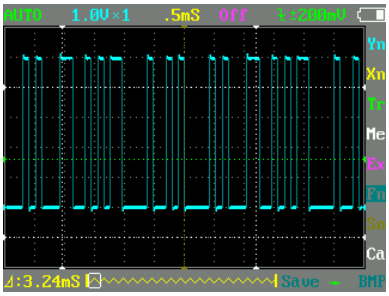

Figura 2: Medición de transmisión de datos con $2ms$ de retraso

Sin embargo es dificil leer en donde empieza y en donde termina cada dato como se puede ver en la figura 2, por lo que se modifica la ultima linea del código para que espere $10ms$ el μC como se ve en la figura 3.

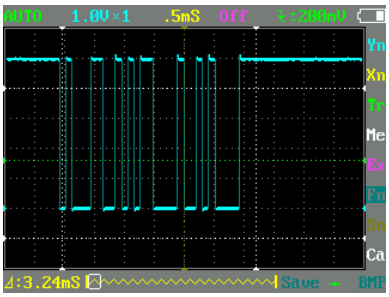

Figura 3: Medición de transmisión de datos con $10ms$ de retraso

Aún en esta figura estamos viendo mas bits de los que realmente queremos medir, por lo que cambiaremos la instrucción `println` por `print` para que solo envíe el dato que queremos, sin el terminador de linea, y podemos ver en la figura 4 que la transmisión en realidad consiste de 9 bits. Tambien podemos notar que el estado de descanso de la transmisión de datos es $5V$, por lo que si analizamos esta transmisión de datos desde el punto de vista de la computadora, si mandamos un dato que empiece con un 1 lógico, la computadora no tendría manera de saber que empezamos a mandar el dato; es necesario que mandemos un 0 lógico siempre que mandemos un dato, no importa cual sea el dato.

Tomando en cuenta esto, y que la transmisión de datos empieza con el bit menos significativo, tenemos que esta señal, la podemos decodificar como el siguiente byte:

$$\underbrace{0011}_{\text{Nibble mayor}}\overbrace{0001}^{\text{Nibble menor}} = 49_{10} \quad (1)$$

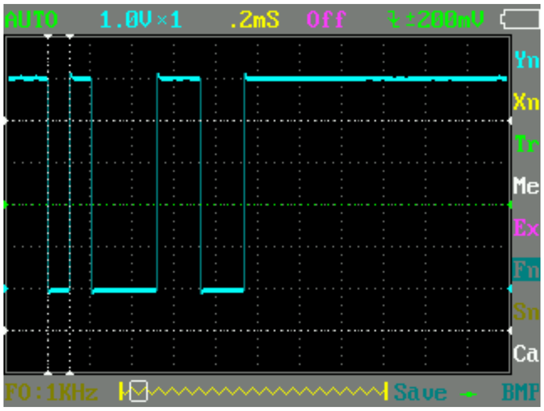

Figura 4: Transmisión de dato 1

Lo cual obviamente no es el dato 1, sin embargo, si nos fijamos en la tabla de caracteres ASCII, podemos ver que 49 corresponde al caracter "1".

Si ahora modificamos el código para enviar el dato "1", en lugar del dato 1, nos damos cuenta que en efecto, el μC esta convirtiendo 1 en caracter y despues enviandolo.

Podemos hacer una prueba más, cambiando "1" por "a", con lo que obtendremos la transmisión de la figura 5.

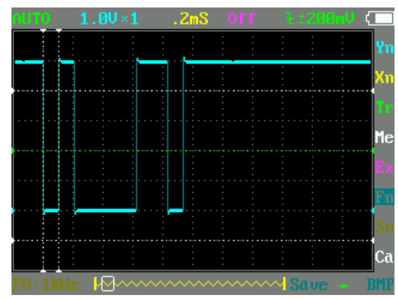

Figura 5: Transmisión de dato "a"

Notemos que estos datos que hemos enviado, no solo se pueden interceptar por medio del pin Tx, mas bien, estos datos tienen la intención de llegar a la computadora por medio de la conversión de datos del chip FTDI o bien CH340G y al puerto serial de la computadora.

Podemos darnos cuenta de esto si abrimos el monitor serial del IDE de Arduino (Menú principal, Herramientas, Monitor Serie), podremos ver una pantalla como la de la figura 6.

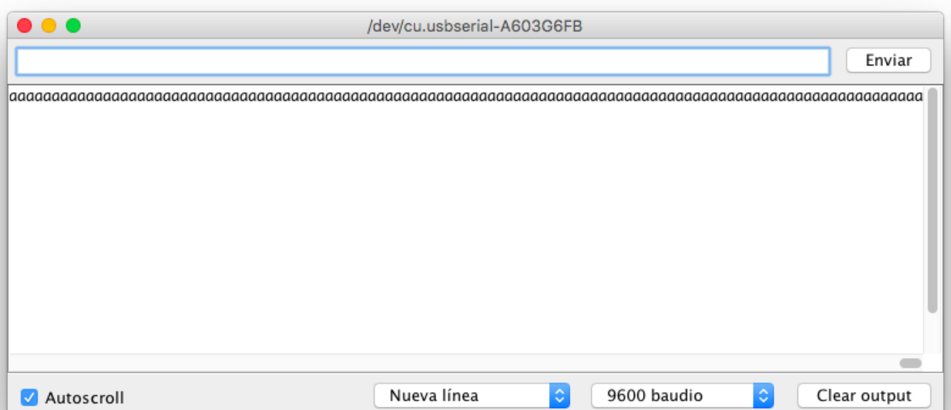

Figura 6: Monitor serial del IDE de Arduino

Equipo

El siguiente equipo será proporcionado por el laboratorio, siempre y cuando lleguen en los primeros 15 minutos de la práctica, y hagan el vale conteniendo el siguiente equipo (exceptuando las pinzas).

- Osciloscopio
- Cables BNC - Caimán
- Cable de alimentación
- Multímetro
- Pinzas

Materiales

- Protoboard
- Potenciometro ($1k\Omega$, $10k\Omega$, etc.)
- Cables

Desarrollo

Para el desarrollo de esta práctica es necesario que cargues el siguiente programa a tu tarjeta de desarrollo Arduino:

```
1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   Serial.println(analogRead(A0));
7   delay(2);
8 }
```

Diseña un circuito de tal manera que el arduino lea la resistencia variable de un potenciometro.

Conclusiones

El alumno deberá describir sus conclusiones al final de su reporte de práctica.

Hoja de Anotaciones

1. ¿Cuál es el tiempo de estabilización del voltaje de salida cuando cambia de $0V$ a $5V$?

2. ¿Cuál es el tiempo de estabilización del voltaje de salida cuando cambia de $5V$ a $0V$?

3. Tomando en cuenta que el Arduino esta configurado para $9600 baudios$ ¿Que tiempo teórico debe tomar la transmisión de un bit de información?

4. Tomando en cuenta que el Arduino esta configurado para $9600 baudios$ ¿Que tiempo real toma la transmisión de un bit de información?

5. Tomando en cuenta que el Arduino esta configurado para $9600 baudios$ ¿Que tiempo real toma la transmisión de un caracter de información?

6. ¿Cuantos bits son transmitidos, si se envía el dato 1? ¿Porqué?

7. ¿Cuantos bits son transmitidos, si se envía el dato 1.0? ¿Porqué?

8. Diseña el código de Arduino que envie los caracteres "bola" a la computadora, cada que la computadora le envie los caracteres "hola".

9. ¿Que tiempo toma la transmisión de estos datos en teoría? ¿En realidad?

10. Diseña el circuito necesario para que el código descrito en el desarrollo lea la posición de un potenciometro. Utiliza la parte posterior de esta hoja.

11. Utiliza el multimetro, el osciloscopio y la grafica serial del IDE de Arduino para medir los niveles de voltaje en el potenciometro; caracteriza la señal obtenida por el multimetro, por el osciloscopio y la grafica del IDE en una grafica en la parte posterior de esta hoja.

12. Carga el ejemplo Fading que se utilizó en la práctica 1, utiliza un multimetro y un osciloscopio para caracterizar la señal dada al LED.

Integrantes del equipo: Revisó:

_____

_____ _____