5555

**UNIVERSIDAD NACIONAL DE CORDOBA**

ELECTRÓNICA DIGITAL 2

Trabajo final

**DOCENTES**

- Del Barco, Martín

- Sánchez, Julio

**ALUMNOS**

- Vega Cuevas, Silvia Jimena 40248013

- Klincovitzky, Sebastián 41158451

**2020**



Contenidos:

**Objetivo** **3**

**Desarrollo3**

Diagrama de flujo 6

Circuito en Proteus 6

Cálculos 7

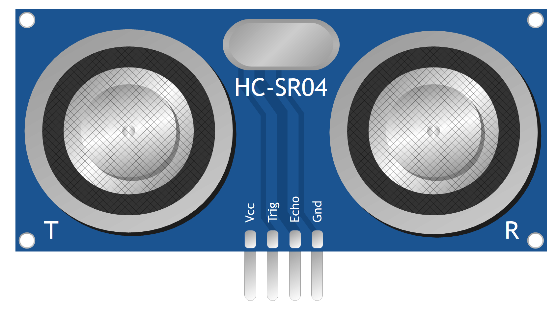
**Conclusión** **9**

Objetivo

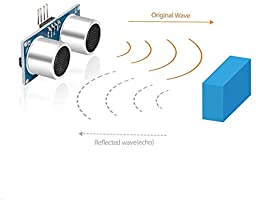
Realizar un programa que utilice al menos 3 interrupciones de distinta naturaleza, y utilice el puerto serie. El programa debe realizarse con un programa escrito en Assembler y usando un simulador como Proteus para armar el circuito utilizando un microcontrolador PIC16f887.

Desarrollo

Para nuestro trabajo elegimos trabajar con el sensor de distancia HC-SR04:



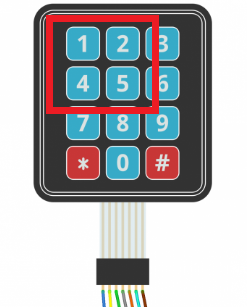
La forma en la que trabaja este sensor es que al recibir en su pin “Trigger” un pulso durante un tiempo mayor a 10µs, este envía una señal de 40 KHz, y espera su rebote con el objeto que tiene enfrente, teniendo la limitación de 2cm mínimo, y 400 cm máximo.



Una vez enviada la señal, recibe un 1 lógico por su pin “Echo”. Sabiendo esto, para conocer la distancia se presenta la siguiente ecuación:

Sabiendo que la velocidad del sonido es 340 m/s, si queremos saber cuánto debe ser el ancho del pulso de 1cm, podemos calcular que, cada aproximadamente 59µs el sensor a medido 1cm.

Por una decisión de diseño, hemos decidido que, para empezar a medir, se debe hacer uso de un teclado matricial 4x4, donde se utilizarán solo 4 teclas(este entonces será 2x2). En el siguiente apartado se procederá a explicar las funcionalidades del programa en base a los botones del teclado matricial.



Funcionalidades de botones:

**Botón 1:**

El programa identifica que este fue el botón apretado, y procede a enviar el pulso necesario al “Trigger” que se encuentra conectado al Puerto B, en su pin RB5.

Por su parte, cuando llega la señal proveniente del pin “Echo” del sensor, se produce una interrupción en RB4.

El programa, por su parte y haciendo uso del Timer0, cuenta el tiempo que pasa entre señal y señal, identificando de esta forma la distancia medida.

Una vez que se apaga “Echo” esto vuelve a interrumpir en RB4, esta vez parando el timer, y realizando los cálculos necesarios para poder hacer uso de la cuenta realizada anteriormente.

El primer centímetro medido es internamente medido con un valor distinto de Timer0 que el realizado por interrupciones posteriormente, debido a que existen distintas operaciones al medio del programa. Los cálculos para ambos valores iniciales son:

* Usando un clock interno de 4MHz, por lo que el tiempo de instrucción del PIC es de 1µs.
* Usando un Prescaler de 1:2

**Botón 2:**

Se encuentran conectados 3 displays al PIC, que en un momento inicial mostrarán ‘0’ en su pantalla, realizando durante todo el tiempo que programa este prendido, una multiplexación con el Timer1, donde cada display se encuentra prendido durante aproximadamente 7ms (considerando una frecuencia de 50 Hz para el multiplexado).

Se decidió que los valores que muestren los displays permanezcan fijos, hasta que se les indique que se actualicen con el ultimo valor medido haciendo uso del segundo botón del teclado. Los cálculos para el Timer1 son:

Donde se utilizó un Prescaler de 1:1 y el resultado final se guarda en los dos registros TMR1L y TMR1H.

**Botón 3** (4 en la imagen):

Se cuenta con un LED rojo conectado al pin RC5, que, en caso de indicárselo con el tercer botón del teclado, empezará a titilar si la distancia medida es menor a 10.

En caso de volver a apretar el botón, el LED dejara de titilar.

Cabe mencionar que el LED utiliza todo el tiempo el ultimo valor medido, por lo que, aunque en los displays no se aprecie, si la distancia medida es menor a 10, empezará a titilar.

Para el parpadeo se decidió que cambie su estado en intervalos de medio segundo. Para ello se aprovechó de la multiplexación de los displays, ya que estos permutan cada 7ms, cada 71 veces que esto suceda, habrá pasado aproximadamente 500ms.

**Botón 4** (5 en la imagen):

Finalmente se hace uso del Puerto Serie de forma asincrónica, tanto transmitiendo como recibiendo. Cuando se aprieta el cuarto y último botón del teclado, se transmitirá por el puerto serie los últimos valores medidos por el sensor. Para la utilización del transmisor no se utilizaron interrupciones, sino que se optó por polling.

**Funcionalidades adicionales:**

Cuando se reciban datos, estos interrumpirán al PIC, y en caso de haber recibido una “M” empezara una nueva medición, por su parte si recibe una “D” mostrara nuevamente los datos medidos. Haciendo uso del rx del puerto serie.

Diagrama de flujo

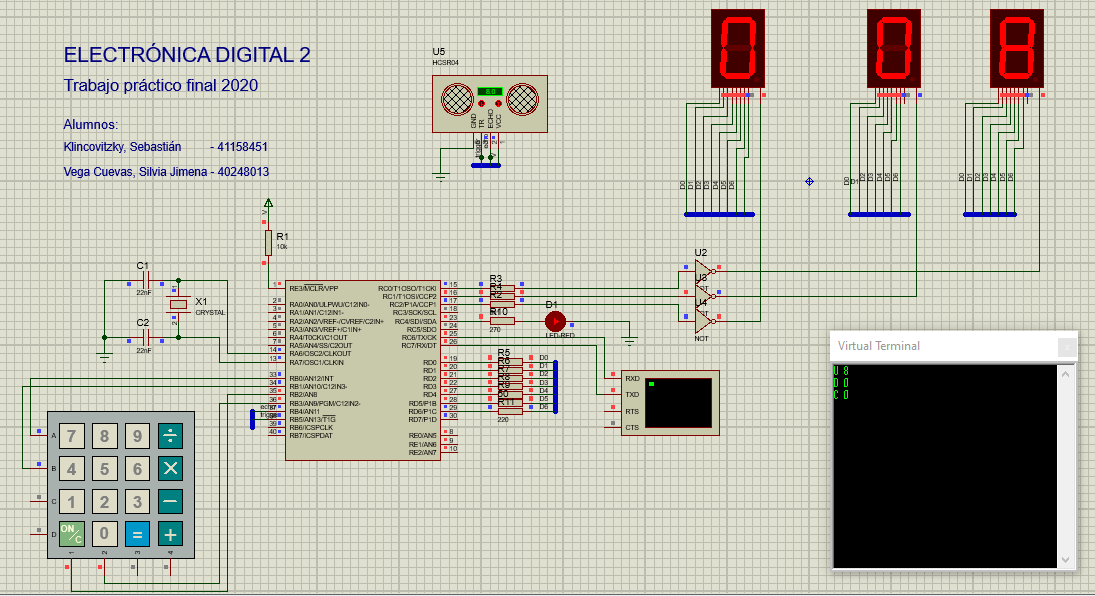
Debido a su tamaño, para que se pueda entender mejor, se adjuntara el diagrama de flujo aparte en la imagen “TPElectronicaDigital.png”, para poder verlo en la máxima calidad posible.

Desarrollo del código

Se hace entrega del código para ser revisado.

Sin embargo, se hace una aclaración. En ciertas partes del programa se hace una lectura del puerto B, pero luego no se hace nada con ello. Esto es para evitar posibles bugs del programa.

Implementación del circuito:



**Cálculos de las resistencias utilizadas**

Los displays utilizados son cátodo común y se consideran los siguientes valores:

* Corriente entregada a un segmento:
* Tensión alta a la salida del puerto:
* Tensión led rojo:
* Tensión de saturación:
* Beta del transistor NPN\* BC-337 :
* Tensión base emisor de transistor NPN\*:

\*transistor: en el esquemático utilizado en Proteus se pueden observar unas NOTs utilizadas, esto es debido a que los displays utilizados no reaccionan bien con ellos en la simulación. Sin embargo, en la implementación del circuito, se pretende utilizar transistores NPN BC337 para la multiplexación.

1. Resistencias de displays :

Considerando valores comerciales:

1. Resistencias que van al transistor de multiplexado

Considerando valores comerciales:

1. Resistencia de led rojo

Considerando valores comerciales:

Conclusiones

El microcontrolador demuestra ser una herramienta poderosa y versátil. Con este trabajo hemos podido ver como un solo chip con un programa en assembler puede interactuar con tantos componentes físicos de forma correcta, incluso de forma simultánea en algunos casos, permitiendo resolver distintos problemas, concentrados en un mismo lugar.

Por otro lado, se aprendió a administrar las múltiples interrupciones existentes para que confluyeran en algo de utilidad sin que haya conflictos entre ellas. También en la opinión de los alumnos, realizar este trabajo, aunque agobiante por el lenguaje utilizado, permitió comprender de una forma más profunda los registros que utiliza el PIC.

Además, con el uso del puerto serie, se puede comprender como funciona el aún mejor ya que la posterior comunicación con otro servidor, permite no solo recibir los datos realizados, ya no solo medidos, si no también procesados, e incluso interactuar a cierta distancia con el sistema. También dio pie a los alumnos a investigar acerca de protocolos de comunicación existentes.

Finalmente, también pudimos apreciar la utilidad de los sistemas Huésped-Objetivo, donde los sistemas embebidos tienen la oportunidad de demostrar que tan útiles pueden llegar a ser. Sobre todo, teniendo en cuenta que el trabajo está basado en simuladores como lo es el Proteus, demostrando ser una poderosa herramienta del diseño de hardware, y además haciendo uso de software externo.