

Laboratorio 2: GPIOs, Timers y FSM

Mauricio Rodríguez Obando/B96694, Katharina Alfaro Solís/B80251

Resumen—En el presente laboratorio se hace uso del microcontrolador ATtiny4313 por medio del uso de cinco interrupciones es usado para simular el comportamiento de una lavadora. Se utilizan además, estados para poder definir las etapas del ciclo de lavado por medio de una máquina de estados (FSM) y a la vez es controlado por un timer para poder de esta manera controlar los tiempos de cada uno de los estados de acuerdo al nivel de carga de la lavadora, sea alta, baja o media. Para poder realizar el mismo, se tiene que comprender bien el funcionamiento de los distintos puertos y funciones del microcontrolador, así como leer la debida documentación y hoja de datos del mismo.

I. NOTA TEÓRICA

I-A. Información del microcontrolador

EN este laboratorio se utilizará el microcontrolador ATtiny4313. Basándose en su hoja de datos [1] se tiene entre las principales características generales:

- De 8 bits con arquitectura RISC.
- Tiene 120 instrucciones, estas la mayoría ejecutadas en un solo ciclo del reloj.
- 2/4K bytes de flash autoprogramable en el sistema.
- EEPROM programable en el sistema de 128/256 bytes.
- SRAM interna de 128/256 bytes.
- 18 puertos I/O programables.
- Circuito de reinicio de encendido mejorado.
- Oscilador calibrado interno
- Tensión de operación entre 1.8 – 5.5V

Como principales características eléctricas (índices máximos absolutos) [1]:

- Temperatura de funcionamiento: -55 °C a +125 °C
- Temperatura de almacenamiento: -65 °C a +150 °C
- Tensión en cualquier Pin excepto RESET con respecto a Tierra: -0.5V a VCC+0.5V
- Tensión en RESET con respecto a Tierra: -0.5V a +13.0V
- Tensión de funcionamiento máximo: 6.0V
- Corriente CC por pin de I/O: 40.0 mA
- Pines VCC y GND de corriente CC: 200,0 mA

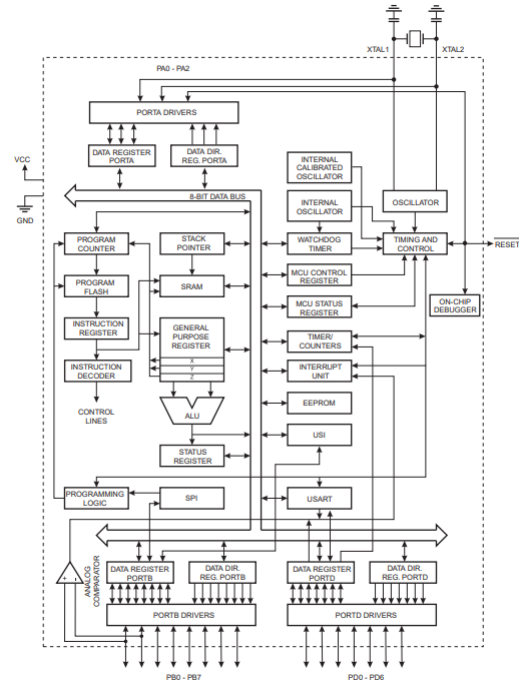


Figura 1. Diagrama de bloques del ATtiny4313. Tomado de [1].

Por otro lado en la figura 2 se muestra el diagrama de pines. En estos pines V_{DD} es el pin para la alimentación de la tensión digital, GND es tierra. Por otro lado, el puerto A (PA1, PA2 ...), es un puerto I/O bidireccional de 3 bits con resistencias pull-up internas. PA2 que tiene la capacidad RESET. Para usar el pin PA2 como pin de I/O, en lugar de Pin RESET, programe ("0") fusible RSTDISBL. Como entradas, los pines del puerto A que se bajan externamente generarán corriente si las resistencias pull-up están activadas. Los pines del puerto A están triestablecidos cuando una la condición de reinicio se activa, incluso si el reloj no está funcionando [1].

Puerto B (PB7..PB0), es un puerto I/O bidireccional de 8 bits con resistencias pull-up internas. Además, los pines de puerto D (PD6..PD0) son puertos I/O pero estos de 7 bits igual con resistencias pull-up internas. Estos dos poseen el mismo funcionamiento que los pines de puerto A, descrito anteriormente [1].

El pin RESET, un nivel bajo en este pin por más tiempo que la duración mínima del pulso generará un reinicio, incluso si el reloj no está funcionando y siempre que el pin de reinicio no haya sido desactivado. El la duración mínima del pulso se proporciona en la Tabla 22-3 en la página 201 en [1]. La entrada de reinicio es una función alternativa para PA2 y dW. Este también se puede usar como un pin de I/O (débil).

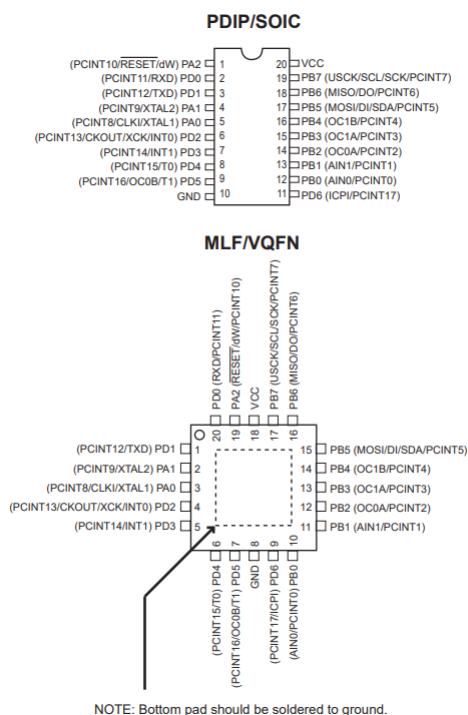


Figura 2. Diagrama de pines del ATtiny4313. Tomado de [1].

I-B. Periféricos utilizados

Para la configuración de los pines se utilizaron los siguientes registros:

- Registro DDRx - Data Directions Register: este registro se utiliza para configurar el modo de funcionamiento de los pines de entrada/salida del microcontrolador. Cada bit del registro DDR corresponde a un pin del microcontrolador, y si el bit se establece en 1, el pin se configura como una salida; si se establece en 0, el pin se configura como una entrada.
- Register PORTx - Data Register: se utiliza para leer o escribir el valor de los pines configurados como salidas mediante el registro DDR. Cada bit del registro PORT corresponde a un pin del microcontrolador, y si el bit se establece en 1, el pin se establece en un estado alto (VCC); si se establece en 0, el pin se establece en un estado bajo (GND). Si el registro se escribe con un 1 lógico cuando el pin esta configurado como entrada, la resistencia pull-up interna va a estar activa y para ser apagada el registro tiene que ser un 0 o tiene que ser configurado como salida.

Cabe mencionar que los pines están esta establecidos cuando la condición de reinicio se activa incluso si ningún timer esta funcionando.

Para hacer lectura de los botones y del temporizador del microcontrolador se utilizaron interrupciones; para las cuales se utilizaron los siguientes registros:

- Registro GIMSK - General Interrupt Mask Register: este registro se utiliza para habilitar o deshabilitar las interrupciones globales por cambio en los pines del

microcontrolador. Si los bits del 7 al 3 se establece en 1, se habilita la interrupción en el pin correspondiente; si se establece en 0, se deshabilita. Los bits del 0 al 2 están reservado por defecto y son leídos como 0.

- Registro MCUCR - MCU Control Register: se utiliza para controlar varias funciones del microcontrolador, como el modo de reposo y la configuración de los bordes de interrupción externa. Los bits correspondientes se establecen en función de la configuración deseada, de modo que sean activas por falnco creciente o decreciente.
- Registro TCCR0x - Timer/Counter Control Register x: se utiliza para controlar el temporizador/contador 0 del microcontrolador. Permite configurar la fuente de reloj, el modo de temporización y la prescaler.
- Registro TIMSK - Timer/Counter Interrupt Mask Register: se utiliza para habilitar o deshabilitar las interrupciones del temporizador/contador del microcontrolador. Si el bit correspondiente se establece en 1, se habilita la interrupción cuando se produce un overflow del contador; si se establece en 0, se deshabilita.

I-C. Componentes electrónicos complementarios

I-C1. Decodificador BCD a 7 segmentos: Su función es codificar las señales recibidas del microcontrolador para poder mostrar el número aleatorio en las dos pantallas de siete segmentos. Para esto se utiliza el código BCD(Decimal Codificado en Binario) para ello forma grupos de 4 bits para representar cada valor del 0 al 9 [2]. El 9 es el valor máximo que se puede representar en un dígito decimal, por lo que este tipo de decodificador es ideal para mostrar números de 0-9 en un display de 7 segmentos.

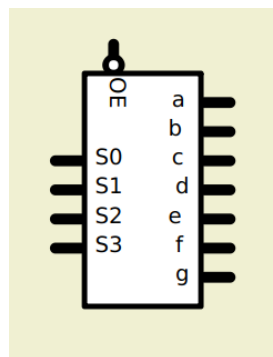


Figura 3. Decodificador BCD a 7 segmentos utilizado en SimulIDE

I-C2. Display de 7 segmentos doble: En este caso al ocupar valores numéricos del 00 al 99, se utilizó un display de 7 segmentos doble (para dos dígitos) con ánodo común tal como se muestra en la siguiente figura.

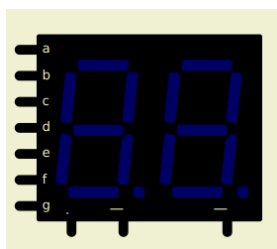


Figura 4. Display de 7 segmentos doble utilizado en SimulIDE

Para conectar correctamente estos display, al igual que cualquier LED, se debe colocar una resistencia en serie con cada LED para limitar la corriente que circula por cada uno de ellos, en total son 7 resistencias. Como se va a configurar un display de ánodo común, el pin común se coloca a (+) y se tendría que poner (-) en los cátodos de cada uno de los segmentos (a – g)] [2]. Se utiliza la hoja de datos [3] en la configuración de ánodo común y para mostrar un color rojo se ocupa una potencia de 62.5mW, 25mA por segmento y una tensión en reversa de 5V.

I-C3. Inversor: Corresponde a la compuerta NOT, de forma que genera una salida opuesta a la entrada de esta. Por lo que, si la entrada está en alto (1) entonces la salida será en bajo (0) y así el caso contrario.

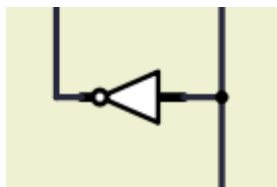


Figura 5. Inversor utilizado en SimulIDE

I-C4. LED: Se utilizan LEDs para indicar el nivel de carga de la lavadora y el estado de lavado en el que se encuentra. Estos consisten en diodos emisores de luz los cuales contienen en su interior un semiconductor que, al ser atravesado por una tensión continua, emiten luz.

I-D. Diseño del circuito

El circuito está diseñado para simular el funcionamiento de una lavadora de acuerdo a lo indicado, en donde se tienen 5 interrupciones: start/pause, stop o reset, lavado alto, medio y bajo. Además como salidas tenemos los LEDs que se van a encender de acuerdo a la carga de lavado, por otro lado los LEDs correspondientes a los estados de lavado dentro de un ciclo de lavado: alto, medio y bajo. También incluimos en nuestro diseño un LED que corresponde al botón de start/pause para poder observar como se comportaba esta interrupción. Podemos ver en la siguiente figura el diseño del mismo.

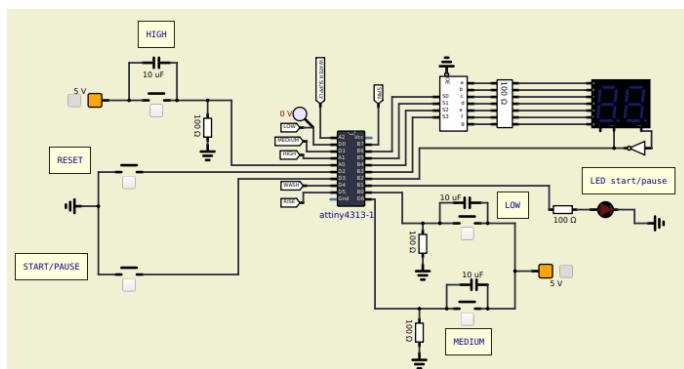


Figura 6. Circuito completo

Como se observa en la figura anterior se hace uso de los tunnel para pasar la señal a los LEDs y se ven de la siguiente manera en el simulador:



Figura 7. Salidas del circuito correspondientes a los LEDs

En el proceso del diseño, primeramente se probó el funcionamiento de las interrupciones de la siguiente manera:

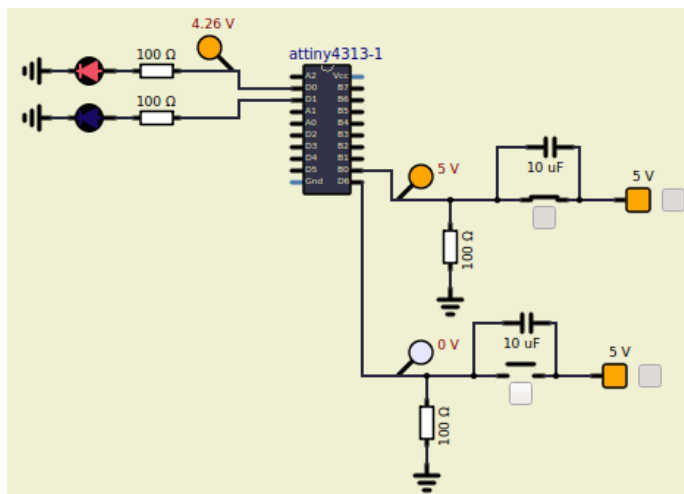


Figura 8. Prueba de las primeras interrupciones

En la figura anterior se muestra cómo al presionar el botón de más arriba, se prende el LED en color rojo que está conectado al puerto DO, esto por medio del uso de interrupciones.

Para los botones/interrupciones se utilizó un capacitor y un resistor tal como se muestra en la figura anterior, esto con el fin de generar un filtro RC, que filtra los posibles rebotes hacia la entrada del microcontrolador. Por efectos de diseño se escogió un capacitor de $10\mu F$ de manera que con un resistor de 100Ω se tiene una frecuencia de corte de 159.2 Hz, por

que el filtro atenuará las señales con frecuencias superiores a 159.2 Hz.

$$f_c = \frac{1}{2 \times 100\Omega \times 10\mu F} = 159,2Hz \quad (1)$$

Además se utiliza un decodificador BCD de binario a decimal para de esta manera poder desplegar los números en el display de 7 segmentos. También es importante aclarar que antes de colocar el display de 7 segmentos se utiliza un resistor de 100Ω para proteger el mismo. Luego el inversor se utiliza para hacer que ambos displays tengan un funcionamiento correcto.

I-E. Lista de componentes y costo

Cuadro I
COSTO PROMEDIO DE LOS COMPONENTES UTILIZADOS EN EL DISEÑO

Componente	Cantidad	Costo por unidad promedio
ATtiny4323	1	\$1.232
Resistencia de 100Ω	11	\$5.41
Capacitor de 10uF	3	\$0.30
LED de 7 segmentos	2	\$0.40
Convertidor BCD a 7 seg	1	\$0.25
Inversor	1	\$0.25
Botón	5	\$1.00
Total		\$8.842

II. DESARROLLO/ANÁLISIS

En la sección del diseño del circuito se muestra la figura donde se realiza todas las conexiones al microcontrolador ATtiny4313. Podemos observar que utilizamos los puertos D2, D3, A0, D6 y B0 como las entradas de las interrupciones. Estas en orden corresponden a INT0, INT1, PCINT8, PCINT17 y PCINT0. Cada una de estas es configurada en una ISR aparte, utilizando INT0_vect para disparar INT0, INT1_vect para disparar INT1, PCINT0_VECT para disparar PCIE0 cuando cambia el puerto PCINT0, PCINT1_VECT para disparar PCIE1 cuando cambia el puerto PCINT8 y PCINT2_VECT para disparar PCIE2 cuando cambia el puerto PCINT17.

Las interrupciones de mayor prioridad corresponden a INT0 e INT1 que corresponden a los botones star/pause y reset. Por otro lado la interrupción para carga baja corresponde a la del PCIE0, la de carga media la del PCIE2 y la de carga alta la del PCIE1.

Además se hace uso de otra interrupción para hacer uso del timer0, este es el contador de 8 bits de ATtiny4313, para habilitar este se utilizaron los registros TCCR0A, TCCR0B, OCR0A, OCR0B, TIMSK y TIFR para poder operarlo, esto se hizo desde la función setup_timer0. Además se configuro el modo de operacion CTC como modo de comparación, de igual manera se utilizó la interrupción global sei (). Para esta interrupción se utilizó el vector TIMER0_COMPA_vect, dentro de la ISR se define la rutina de interrupción la cual se basa en un contador el cual va aumentando la cuenta cada vez que pasa un segundo, en este caso se establecio un prescaler en el setup_timer0 de 64 y tiene un reloj de 1MHz por lo que un segundo es cada 15625 ciclos del timer.

Para el desarrollo del programa se realizó el siguiente diagrama de flujo:

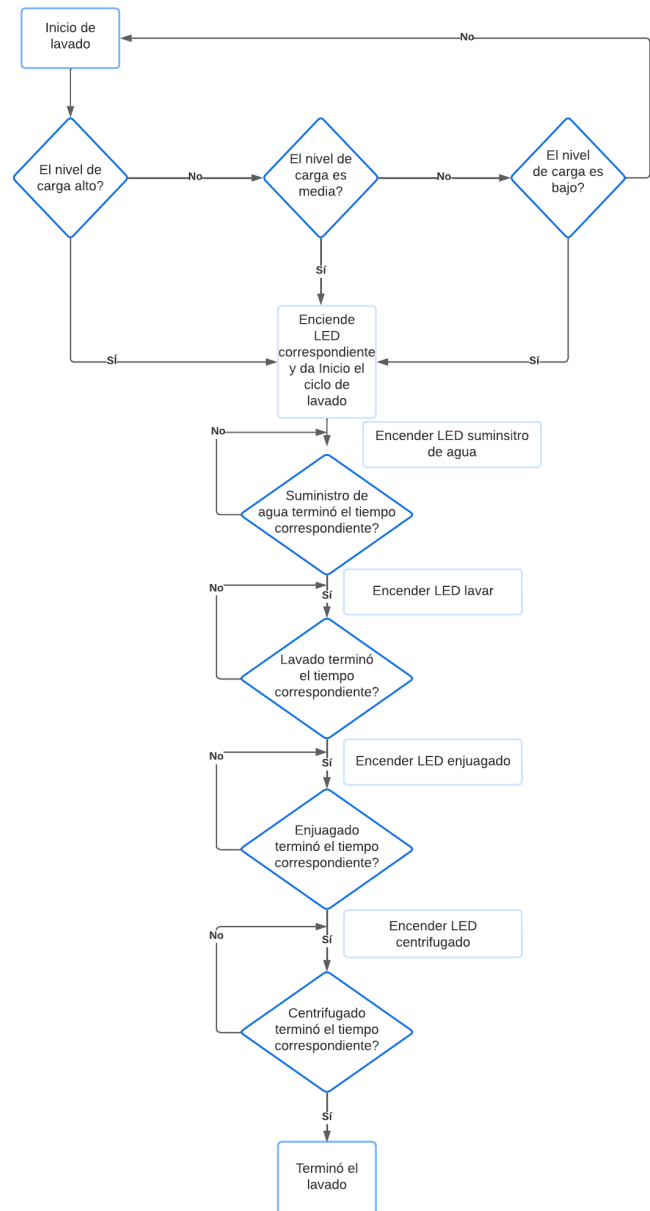


Figura 9. Diagrama de flujo Lavadora

EL diagrama de flujo anterior se utilizó como una base para realizar la máquina de estados de acuerdo lo requerido en un ciclo de lavado. De manera que los estados en la FSM corresponden a: STATE_IDEL, STATE_WATER_SUPPLY, STATE_WASH, STATE_RISE, STATE_SPIN y STATE_FINISH. EL estado inicial (STATE_IDEL) pasa de una vez al estado inicial de lavado STATE_WATER_SUPPLY dentro de este al igual que todos los demás estados de lavado del ciclo se llama una función en la cual se define los puertos que son la salida de este, además del comportamiento durante este estado cuando la carga es baja, media o alta. Esto debido a que dependiendo del nivel de carga así es el tiempo en cada estado de lavado.

La función que se llama en el estado STATE_FINISH se escribió de manera que en esta todas las luces, de los estados de lavado se encienden cuando llega al final del lavado, tal y como se muestra en la siguiente figura.

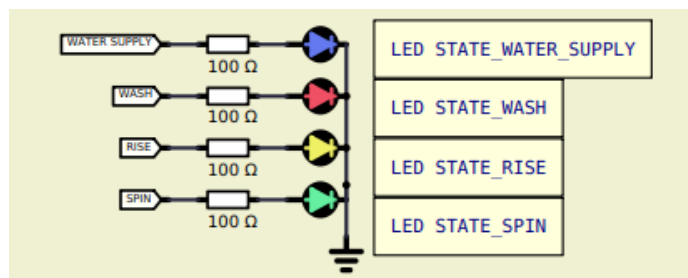


Figura 10. Leds al finalizar el lavado

Dentro de las funciones de cada uno de estos estados de lavado, se define el compare_value que es el que se compara con la interrupción del timer0 definido y además este es el valor que se espera ir observando en los display de 7 segmentos, ya que se espera ir observando la cuenta regresiva de la duración del ciclo de lavado. En esta parte se conectó el display correctamente como se observa en la figura del circuito sin embargo, esta parte no funcionó de la manera esperada, de manera que el error puede ser por no lograr la conexión correcta entre los display, el timer y la interrupción de esta en el código. Antes de llegar al estado FINISH en donde se prenden todos los leds, se obtiene los siguientes resultados intermedios que complementan un poco mas la verificación del funcionamiento funcionamiento del diseno:

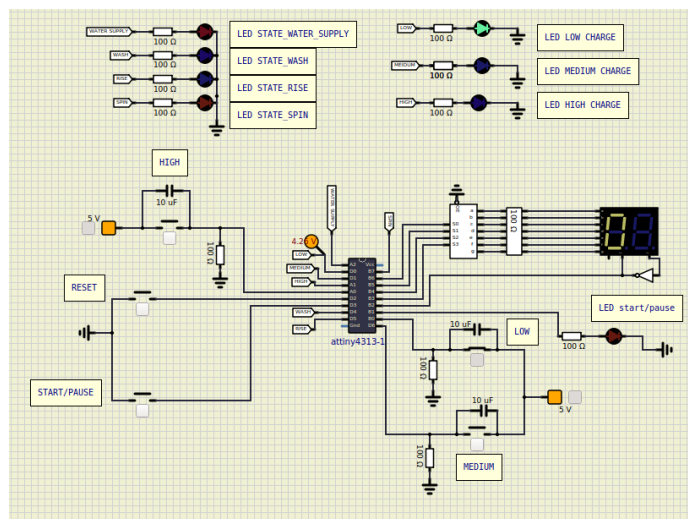


Figura 11. Selección de carga

De esta figura se puede observar como tras presionar el botón de carga baja se prende el led correspondiente por lo que ya es seleccionado esta condición necesaria para el funcionamiento. Luego también se procede a presionar el botón de START/PAUSE, en donde al ser presionado y el temporizar estar detenido se prende el led y comienza el ciclo

de lavado; seteado los tiempos de para cada ciclo según la carga seleccionada. como se observa en la siguiente figura:

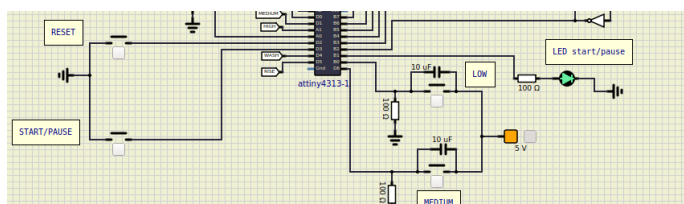


Figura 12. Utilización del botón START/PAUSE

Así inmediatamente después presionar el botón de start comienza el ciclo de lavado por el suministro de agua por lo que se prende su led correspondiente, durante el tiempo necesario según la carga seleccionada para así posteriormente avanzar al siguiente ciclo hasta obtener el encendido de todos los leds indicando que se termino el ciclo.

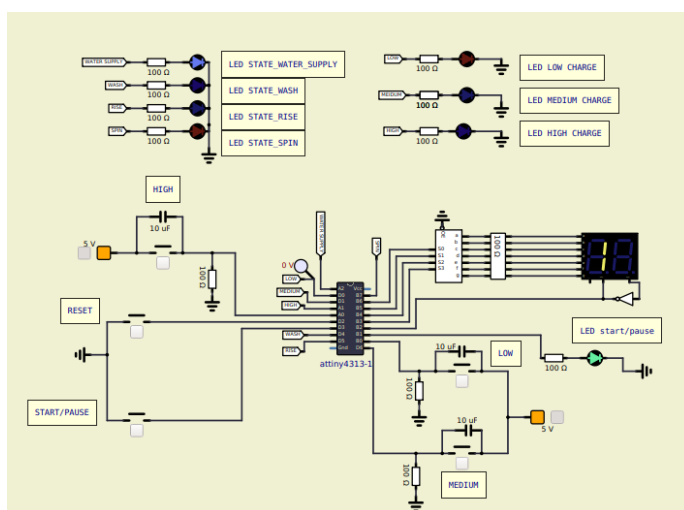


Figura 13. Comienzo del ciclo de lavado

Para el compartamiento del display de 7 segmentos se realizó una función llamada display_led_with_timer donde se recibe el tiempo que dura cada estado del lavado de acuerdo a la carga seleccionada. De forma que se muestra cuando se encuentra en este estado de forma descendente el tiempo que dura en el mismo, a continuación se muestra como ejemplo para la carga baja cuando se encuentra en el estado de lavado.

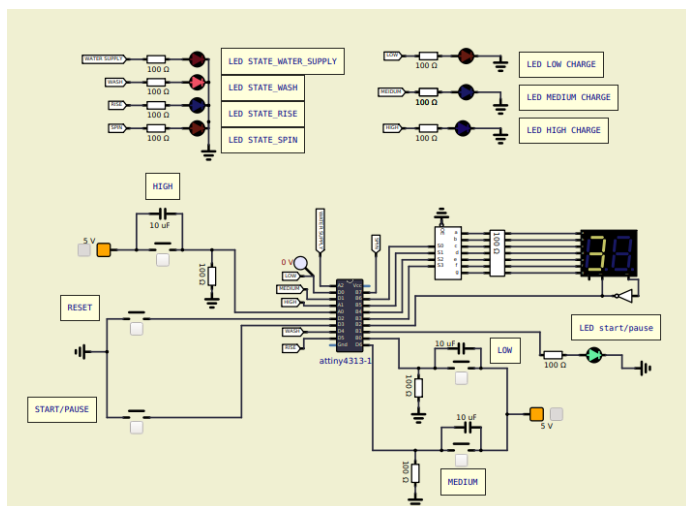


Figura 14. Comienzo del estado lavado en la carga baja, tiempo restante 3 segundos

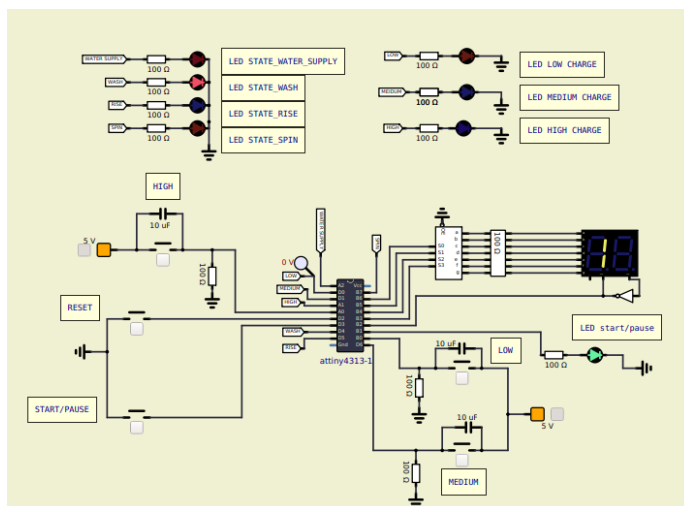


Figura 16. Comienzo del estado lavado en la carga baja, tiempo restante 1 segundo

III. CONCLUSIONES

El uso de las interrupciones forma parte crítica de este laboratorio, así como el entendimiento de estas y realizar de forma adecuada sus conexiones. De manera que, podemos concluir que en este laboratorio estas funcionan adecuadamente, de forma que el estado del programa se guarda cuando una de estas es disparada.

Se hace uso del timer0 que corresponde al contador de 8 bits en el microcontrolador utilizado, ATTINY4313. Este es utilizado por medio de interrupciones en este caso.

Para la realización de este laboratorio se utilizó una máquina de estados, por lo cual, es importante definir bien los estados de este y el flujo de trabajo. Se recomienda ir por partes y también visualizar todo el funcionamiento del la FSM para no dejar escapar algún detalle.

IV. LINK DEL REPOSITORIO DE GITHUB

Link al repositorio de github:

https://github.com/kathalsol/Laboratorio_Microcontroladores/tree/main/Labo2

REFERENCIAS

- [1] 8-Bit, ATtiny2313A ATtiny4313, Atmel, 2011. [Online]. Available: <https://www.tme.eu/Document/8d5d175b16ee510fd73f212b6eed1278/ATTINY2313A-4313-DTE.pdf>
- [2] M. JPM, "74ls48 decodificador bcd 7 segmentos [cd4511]," Disponible en línea, accesado el 29 de Marzo de 2023. [Online]. Available: <https://www.microjpm.com/products/a74ls48/>
- [3] A. Technologies, "14.2 mm (0.56 inch) general purpose two digit seven segment displays," Disponible en línea, accesado el 26 de Marzo de 2023. [Online]. Available: <https://www.farnell.com/datasheets/95204.pdf>

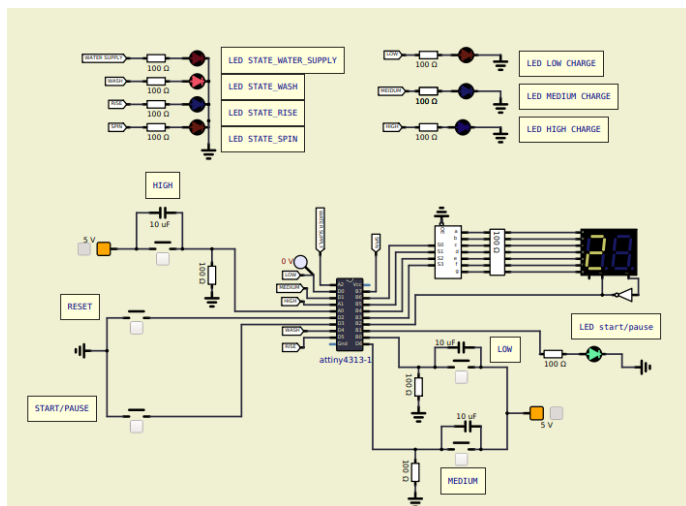


Figura 15. Comienzo del estado lavado en la carga baja, tiempo restante 2 segundos