

Universidad de Costa Rica

Facultad de Ingeniería

Escuela de Ingeniería Eléctrica

IE0624 – Laboratorio II

III ciclo 2023

Título: GPIOs, Timers y FSM

Estudiantes:

Kevin Campos Campos

Josué Salmerón Córdoba

Grupo 1

Profesor: Marco Villalta

21 de enero de 2024

# Índice

1. Resumen	1
2. Nota teórica	2
3. Desarrollo/Análisis	14
4. Conclusiones y recomendaciones	21
5. Anexos	23

# Índice de figuras

1.	Pines del ATtiny4313. Tomado de [1]. . . . .	2
2.	Diagrama de bloques del ATtiny4313. Tomado de [1]. . . . .	4
3.	Diagrama de bloques del ATtiny4313. Tomado de [1]. . . . .	5
4.	Registro de dirección de datos. Tomado de [1]. . . . .	5
5.	Registro de datos. Tomado de [1]. . . . .	5
6.	Configuración del registro GIMSK. Tomado de [1]. . . . .	5
7.	Registro habilitación de interrupciones. Tomado de [1]. . . . .	6
8.	Registro de control de Timer/Counter. Tomado de [1]. . . . .	6
9.	Registro de comparación de salida A. Tomado de [1]. . . . .	6
10.	Conexión de los displays con el ATtiny4313. Tomado de [1]. . . . .	7
11.	Botones. Tomado de [?]. . . . .	7
12.	LEDs. Tomado de [?]. . . . .	7
13.	Demultiplexor. Tomado de [?]. . . . .	8
14.	Esquemático general . . . . .	9
15.	Encendido de un LED. . . . .	9
16.	Carga media . . . . .	10
17.	Indicador de carga media . . . . .	10
18.	Encendido de un LED con cuenta regresiva. . . . .	11
19.	Transición de estados de la lavadora. . . . .	12
20.	Diseño del circuito completo. . . . .	12
21.	Diagrama de flujo función led_carga. . . . .	14
22.	Máquina de estados transición de LEDs. . . . .	15
23.	Conituación máquina de estados transición de LEDs. . . . .	16
24.	Función display . . . . .	17
25.	Carga baja lavándose. . . . .	17
26.	Carga baja lavándose parte 2. . . . .	17
27.	Estado suministro, carga media. . . . .	18
28.	Estado lavar, carga media. . . . .	18
29.	Estado enjuagar, carga media. . . . .	19
30.	Estado centrifugar, carga media. . . . .	19
31.	Estado suministro, carga alta. . . . .	20
32.	Estado lavar, carga alta. . . . .	20
33.	Estado enjuague, carga alta. . . . .	20
34.	Estado centrifugar, carga alta. . . . .	21

# Índice de tablas

1.	Lista de equipos . . . . .	8
----	----------------------------	---

# 1. Resumen

En el presente trabajo, se presenta la simulación de una lavadora automática por medio de interrupciones, se definen 3 botones que se encargan de fijar la carga de ropa/artículo que se necesita limpieza, una vez hecho esto la lavadora inicia su labor por medio de un temporizador representado por dos displays de 7S, e indicando por medio de LEDs la carga asignada y el estado de lavado. El uso de interrupciones y una máquina de estados permitió realizar todo lo anterior, es decir, la lavadora funciona correctamente apenas reconoce la carga dada y muestra correctamente todos los pasos, excepto el botón de pausa, que por razones de tiempo no fue posible crearlo ya que no sobraron pines para su diseño, no obstante es un diseño que garantiza el funcionamiento apropiado de una lavadora automática.

## 2. Nota teórica

En esta sección se muestran las descripciones generales de todos los componentes usados para construir el circuito solicitado.

### Microcontrolador ATTiny4313

El ATTiny2313A/4313 es un microcontrolador CMOS de 8 bits de bajo consumo basado en el AVR mejorado. Arquitectura RISC. Al ejecutar poderosas instrucciones en un solo ciclo de reloj, el ATTiny2313A/4313 logra rendimientos cercanos a 1 MIPS por MHz, lo que permite al sistema diseñador para optimizar el consumo de energía frente a la velocidad de procesamiento [1]. Por lo que se requiere estudiar el diagrama de pines mostrado en la figura 1

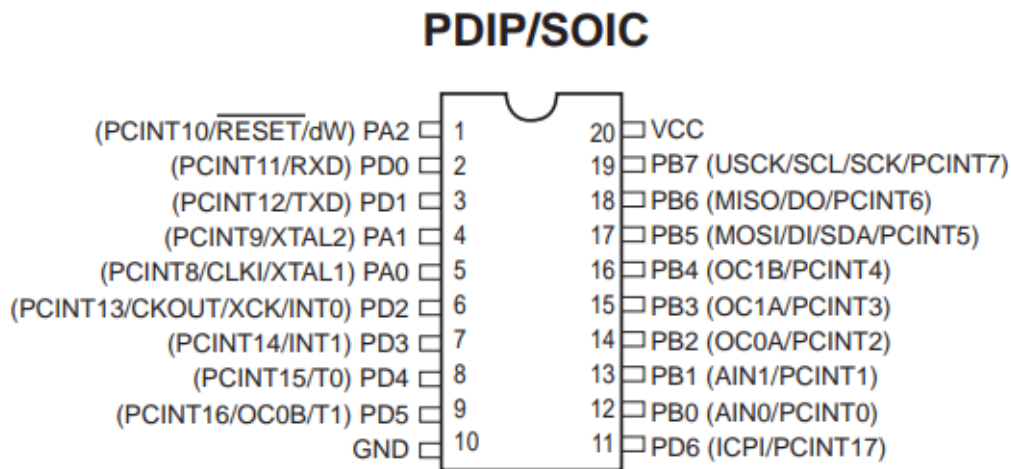


Figura 1: Pines del ATTiny4313. Tomado de [1].

### Características generales

A continuación, se describen las características de cada pin para tener un mejor entendimiento de lo que se puede hacer con cada uno de ellos [1].

1. PA2-RESET/dW/PCINT10: RESET: entrada activa en bajo y activada por un 1 el RSTDISBL fuse. El pullup se activa y el controlador de salida y la entrada digital se desactivan cuando se usa el pin de RESET. dW: Cuando el fusible debugWIRE Enable (DWEN) está programado y los bits de bloqueo están sin programar, se activa el sistema debugWIRE dentro del dispositivo de destino. El pin RESET del puerto está configurado como un pin de I/O bidireccional de cable Y (drenaje abierto) con pull-up habilitado y se convierte en la puerta de enlace de comunicación entre el objetivo y el emulador. Sirve como I/O y tiene la opción de RESET. Solo que debe ponerse un 0 para que funcione como I/O. Tiene una resistencia interna de pull up. PCINT10: Fuente de interrupción de cambio de pin. El pin PA2 sirve como una fuente de interrupción externa.
2. PD0-RXD/PCINT11. RXD: UART receptor de datos. PCINT11: El pin PD0 sirve como fuente de interrupción externa.
3. PD1-TXD/PCINT12. TXD: UART transmisor de datos. PCINT12: El pin PD0 sirve como fuente de interrupción externa.

4. PA1-XTAL2/PCINT9. XTAL2: chip de reloj oscilador. Se utiliza como pin de reloj para todas las fuentes de reloj de chip excepto oscilador RC interno calibrable y reloj externo. Cuando se utiliza como pin de reloj, el pin no se puede utilizar como pin de I/O. Cuando se utiliza un oscilador RC interno calibrable o un reloj externo como en las fuentes de reloj del chip, PA1 sirve como un pin de I/O normal. Este pin sirve como fuente de interrupción externa.
5. PA0-XTAL1/CLKI/PCINT8. XTAL1: oscilador de reloj de chip. Se utiliza para todas las fuentes de reloj de chip excepto oscilador RC interno calibrable. Cuando se usa como pin de reloj, el pin no se puede usar como pin de I/O. Cuando se utiliza un oscilador RC interno calibrable como fuente de reloj de chip, PA0 sirve como pin de I/O. Este pin sirve como fuente de interrupción externa.
6. PD2-INT0/XCK/CKOUT/PCINT13. INT0: este pin sirve como fuente de interrupción externa para el MCU. XCK: USART Reloj de transferencia utilizado sólo en el modo de transferencia síncrona. CKOUT: reloj sistema de salida.
7. PD3-INT1/PCINT14. Ambas descripciones representan la misma equivalencia, ya que su objetivo final es que el pin PD3 sirve como fuente de interrupción externa.
8. PD4-T0/PCINT15. T0: timer/counter0 la entrada del reloj contador externo se habilita configurando (uno) los bits CS02 y CS01 en el registro de control del timer/counter0 (TCCR0). PCINT15: este pin sirve como fuente de interrupción externa.
9. PD5-OC0B/T1/PCINT16. OC0B: Output Compare Match B. T1: La entrada del contador externo se habilita configurando (uno) los bits CS02 y CS01 en el registro de control del timer1/counter1 (TCCR1) PCINT16: este pin sirve como fuente de interrupción externa.
10. GND: nodo de tierra.
11. PD6-ICPI/PCINT17. ICPI: entrada de captura. Este pin puede actuar como un pin de captura de entrada para el timer/counter1. PCINT17: este pin sirve como fuente de interrupción externa.
12. PB0-AIN0/PCINT0. AIN0: es un comparador análogo de entrada positiva. Configure el pin del puerto como entrada con el pull-up interno apagado para evitar que la función del puerto digital interfiera con la función del comparador analógico PCINT0: este pin sirve como fuente de interrupción externa.
13. PB1-AIN1/PCINT1. AIN1: es un comparador análogo de entrada negativa. PCINT1: este pin sirve como fuente de interrupción externa.
14. PB2-OC0A/PCINT2. Output Compare Match A output. OC0A: el pin PB2 puede servir como salida externa para comparación de salida de timer/counter0 A. El pin debe configurarse como salida (configurando DDB2 igual 1) y cumplir esta función. PCINT2: este pin sirve como fuente de interrupción externa.
15. PB3-OC1A/PCINT3. OC1A: Output Compare Match A output. El pin PB3 puede servir como salida externa para comparación de salida de timer/counter1 A. El pin debe configurarse como salida (configurando DDB3 igual 1) y cumplir esta función. PCINT3: este pin sirve como fuente de interrupción externa.
16. PB4-OC1B/PCINT4. OC1B: Output Compare Match B output. Se puede configurar DDB4 igual a 1 para activar esta función. PCINT4: este pin sirve como fuente de interrupción externa.

17. PB5-DI/SDA/PCINT5.DI: Entrada de datos de interfaz serie universal en modo de tres cables. El modo de tres cables no anula las funciones normales del puerto, por lo que el pin debe configurarse como entrada. SDA: modo serial interfaz de datos de dos cables. PCINT5: este pin sirve como fuente de interrupción externa.
18. PB6-DO/PCINT6. D0: Salida de datos de interfaz serie universal en modo de tres cables. Modo de tres cables de salida de datos anula el valor de PORTB6 y se dirige al puerto cuando el bit de dirección de datos DDB6 es uno. Sin embargo, el bit PORTB6 aún controla el pull-up, lo que permite el pull-up si se ingresa la dirección y PORTB6 en 1. PCINT6: este pin sirve como fuente de interrupción externa.
19. PB7-USCK/SCL/PCINT7. USCK: interfaz serial de reloj modo de 3 cables. SCL: reloj serial para USI modo de dos cables. PCINT7: este pin sirve como fuente de interrupción externa.
20. VCC: fuente de alimentación.

En la figura 2 se muestra el diagrama de bloques de este microcontrolador.

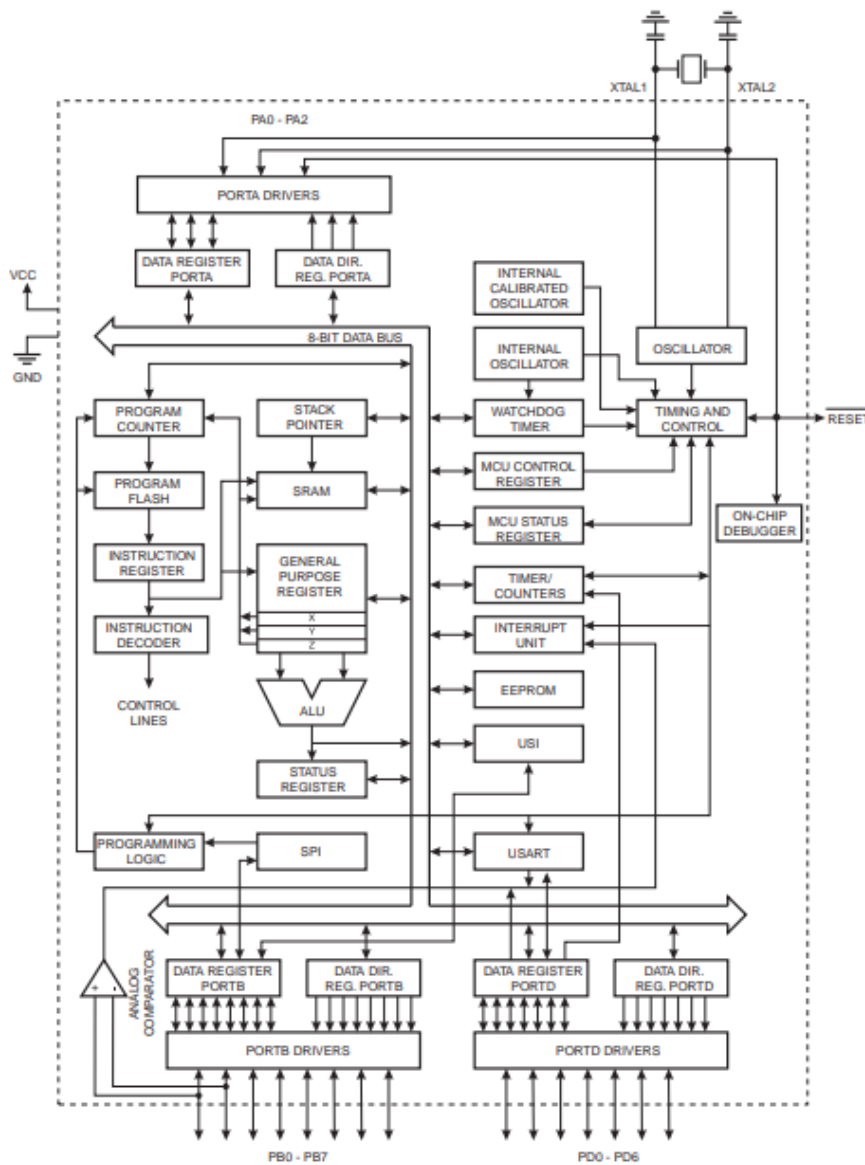


Figura 2: Diagrama de bloques del ATtiny4313. Tomado de [1].



Otro detalle importante es considerar los valores máximos con los que se puede trabajar el ATtiny4313.

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground.....	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground.....	-0.5V to +13.0V
Maximum Operating Voltage .....	6.0V
DC Current per I/O Pin .....	40.0 mA
DC Current $V_{CC}$ and GND Pins.....	200.0 mA

Figura 3: Diagrama de bloques del ATtiny4313. Tomado de [1].

## Periféricos

La descripción de los registros e instrucciones utilizados son los siguientes:

- **DDRxn**: Registro de dirección de datos. Encargado de determinar si los pines van a trabajar como entradas (0) o como salidas (1). El registro de dirección de datos se observa en la siguiente imagen:

### 10.3.9 DDRD – Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x11 (0x31)	–	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 4: Registro de dirección de datos. Tomado de [1].

- **PORTxn**: Registro de datos. Permite establecer los pines con un estado inicial, ya sea en alto o en bajo, esto para el uso de señales digitales.

### 10.3.8 PORTD – Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x12 (0x32)	–	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 5: Registro de datos. Tomado de [1].

- **GIMSK**: (General Interrupt Mask Register). Registro encargado de habilitar y deshabilitar interrupciones. Cuando se desea leer interrupciones es necesario configurar este registro.

### 9.3.2 GIMSK – General Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x3B (0x5B)	INT1	INT0	PCIE0	PCIE2	PCIE1	–	–	–	GIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Figura 6: Configuración del registro GIMSK. Tomado de [1].

- **PCMSKn**: Usado para habilitar o deshabilitar ciertos pines para que trabajen como pines de interrupción.

#### 9.3.4 PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	–	PCINT17	PCINT16	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCMSK2
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 7: Registro habilitación de interrupciones. Tomado de [1].

- **TCCR0x**: Este es el registro de control para el contador 0 (x puede ser A o B), además, permite establecer el prescaler y modos de comparación. Hay dos registros de control para el contador 0, ya sea el A o el B.

#### 11.9.1 TCCR0A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
0x30 (0x50)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 8: Registro de control de Timer/Counter. Tomado de [1].

- **OCR0A**: Registro que guarda el valor de comparación. Cuando el valor de OCR0A es igual al de TCNT0 la señal del comparador se pone en alto.

#### 11.9.4 OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x36 (0x56)	OCR0A[7:0]								OCR0A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Figura 9: Registro de comparación de salida A. Tomado de [1].

## Componentes electrónicos complementarios

### Decodificador BCD y display de 7 segmentos

BCD (Decimal Codificado en Binario) es un código que representa valores decimales en formato binario, para ello forma grupos de 4 bits para representar cada valor del 0 al 9. El 9 es el valor máximo que se puede representar en un dígito decimal [2]. Lo anterior justifica el hecho de usar dos display de 7 segmentos, al tener 2 de estos componentes es necesario usar dos convertidores ya que se necesita optimizar la cantidad de pines del MCU ATtiny4313.

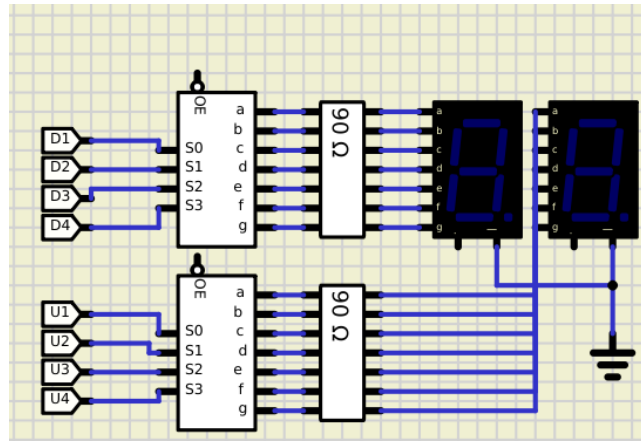


Figura 10: Conexión de los displays con el ATtiny4313. Tomado de [1].

## Switches

Los botones son elementos importantes para poder activar las interrupciones, no obstante, hay que tener en cuenta el efecto rebote para evitar las lecturas falsas en el circuito.

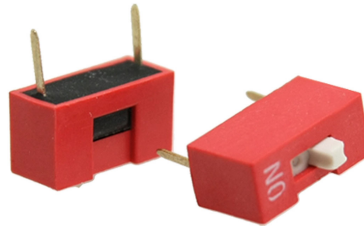


Figura 11: Botones. Tomado de [?].

## LEDs

Los Leds se encargarán de mostrar los estados de la lavadora, y para estos hay que ponerles una resistencia adecuada para no afectar este tipo de señales. Por este motivo, se procuro operar bajo tensiones eléctricas menores a 2,4 V.

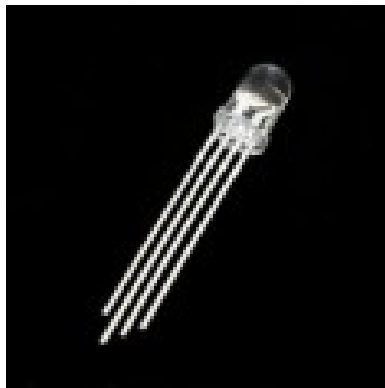


Figura 12: LEDs. Tomado de [?].

## demultiplexor

Es un decodificador/demultiplexor dual 1 de 4 de alta velocidad. El dispositivo tiene dos decodificadores independientes, cada uno de los cuales acepta dos entradas y proporcionando cuatro salidas LOW activas mutuamente excluyentes [?]. Se adaptó este componente para configurar la salida de los LEDs y así poderlos conectar al MCU.



Figura 13: Demultiplexor. Tomado de [?].

## Lista de componentes

En la tabla 1 resume el equipo electrónico a usar para este laboratorio.

**Nota:** Se encontró un kit donde vienen todas las magnitudes buscadas, 100  $\Omega$ , 100 k $\Omega$ , 10 k $\Omega$

Tabla 1: Lista de equipos

Componente	Cantidad	Precio
ATtiny4313	1	1.83\$
LED 7 segmentos	2	1.30\$
Resistencias	1	9.95\$
Capacitores 100 nF	4	0.76\$
Switches	4	2.6\$
LEDs	7	3.85\$
Demultiplexor	1	2.35\$
Convertidores Bcd-7S	2	2.60\$
<b>Total</b>		25.24\$

## Diseño del circuito

Al tratarse de una lavadora compuesta de máquina de estados, el diseño del circuito se fue realizando por etapas. No obstante, la lógica de la electrónica fue posible armarla con los conocimientos adquiridos del primer laboratorio. El esquemático base para la lavadora automática es el siguiente:

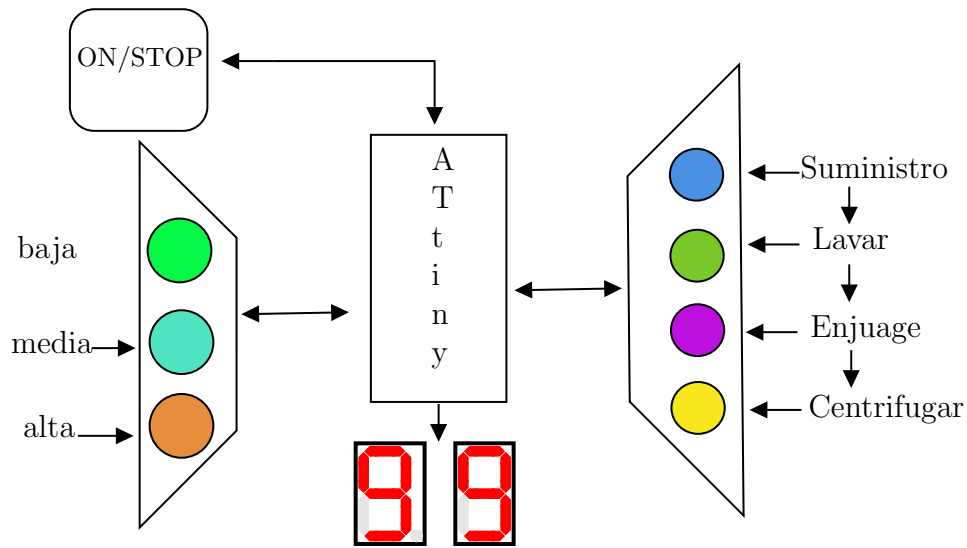


Figura 14: Esquemático general

Del esquemático 24, es claro notar que el circuito general se compone de varias etapas. Así, es importante saber el uso de las interrupciones para realizar una tarea tan simple como encender o apagar un LED, por ese motivo, hay que consultar la hoja del fabricante del MCU para tener más detalles de los pines para programar la interrupción, además habilitarlos para que sirvan como salida o entrada dependiendo de la conexión hecha previamente. El ejemplo de la figura 15,

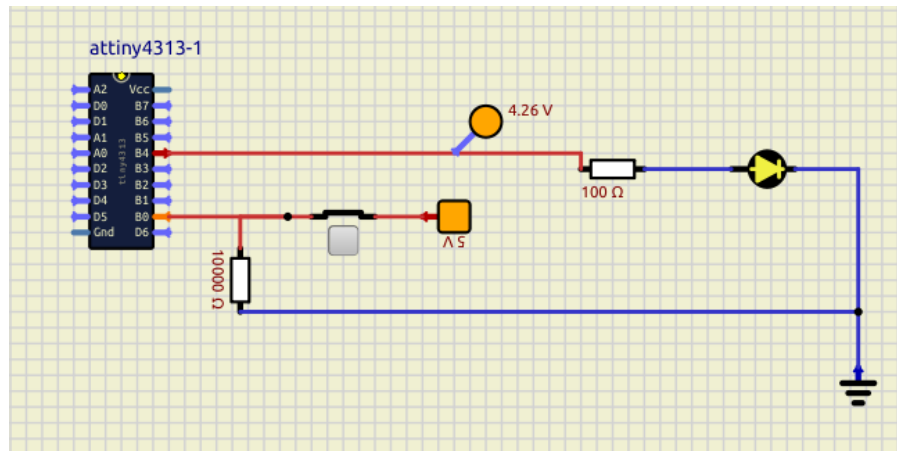


Figura 15: Encendido de un LED.

se muestra que el pin B0, es una entrada para este caso, y esto indica que ocupará habilitar la resistencia de pull-up con el enmáscarado `GIMSK`. Por su parte, el pin B4 se trata de una salida, entonces basta con hacer uso de `DDRB=0b00010000`, para que el programa entienda que es una salida. De esa manera, presionando el botón, es posible encender el led adecuadamente.

La anterior fue de gran importancia para entender el comportamiento de las interrupciones y todo lo necesario para hacer que éstas trabajen adecuadamente. Por lo que una de las primeras etapas realizadas para este laboratorio fue encender LEDs por medio de botones haciendo uso de interrupciones. No sin antes mencionar que al trabajar con botones es importante diseñarlos de tal manera que el efecto rebote cause falsas lecturas en el circuito, por tanto, se diseñó un factor de  $\tau$  igual para todos los botones.

$$\tau = RC \Rightarrow \tau = 100 \text{ k}\Omega \cdot 100 \text{ nF} = 0,01 \quad (1)$$

Lo que es equivalente a frecuencia de 100 Hz.

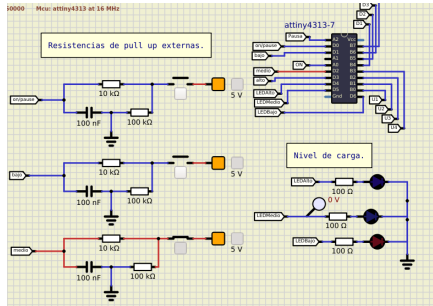


Figura 16: Carga media

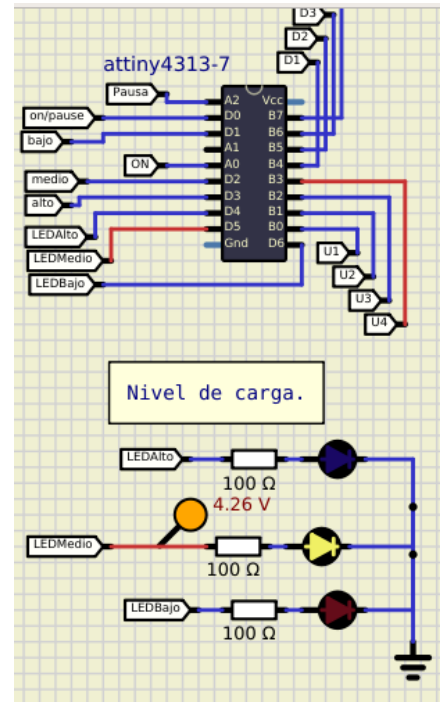


Figura 17: Indicador de carga media

Primero se presiona el botón, donde es claro notar el flujo de corriente esperado que por medio de una interrupción le llegará al LED de carga media, y de manera inmediata es posible ver que el LED se enciende tal como se esperaba. Además, note que le está llegando una alimentación de 5 V, lo que es una tensión eléctrica adecuada para que este componente no se dañe. Esta prueba se realizó con todos los demás LEDs.

Ahora, la segunda etapa consistió en mostrar en los displays de segmentos de la figura 10 con los tiempos totales de cada carga que se le vaya asignar a la lavadora. Esto consiste en programar una cuenta regresiva haciendo uso de los pines B, los cuales están configurados como salida y así mostrar el respectivo número definido previamente en el código. Entonces, al presionar cualquiera de los 3 botones los display mostrarán el total del tiempo para cada carga tal como se muestra a continuación:

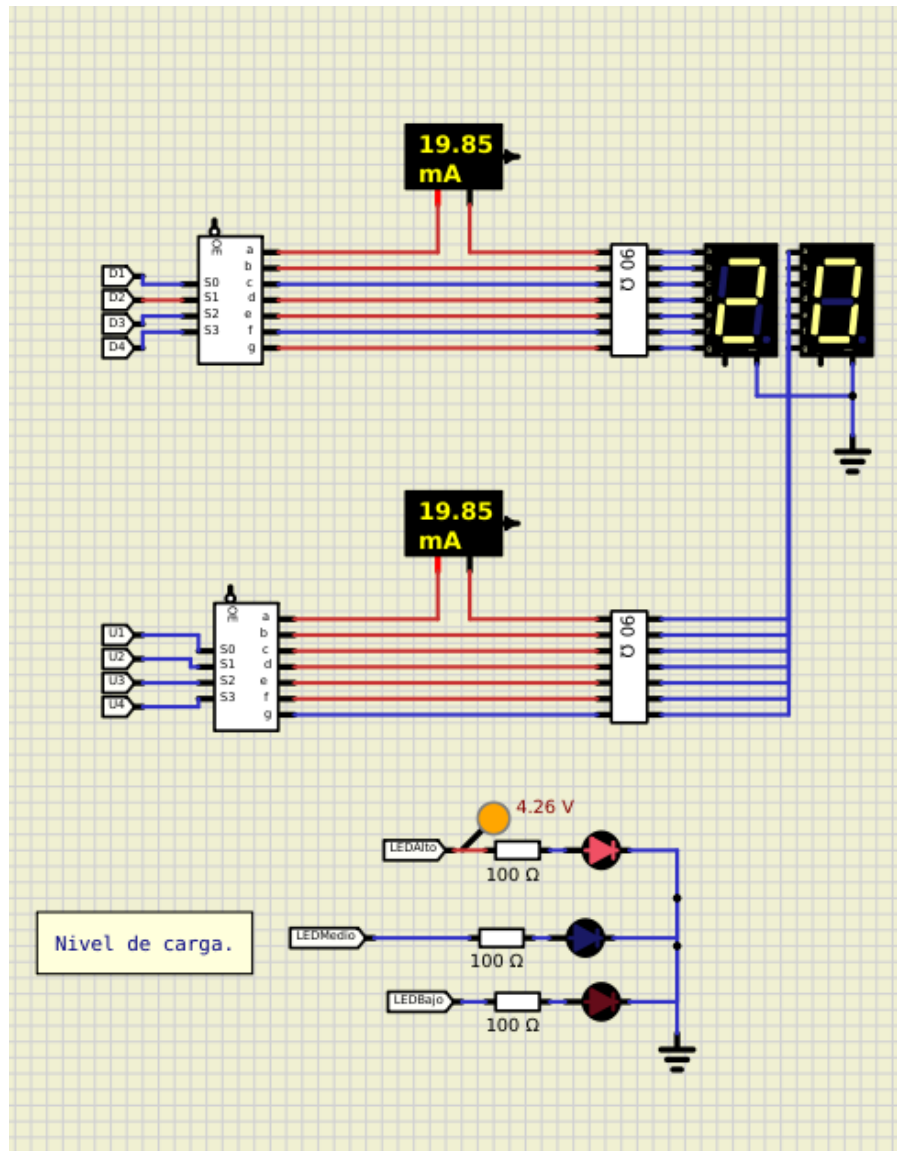


Figura 18: Encendido de un LED con cuenta regresiva.

Lo primero a resaltar es que la corriente eléctrica que le está llegando a los displays es la adecuada, ya que no supera los 0,02 A, así como al LED. Además, note que a los displays hay conectada una resistencia de  $90\ \Omega$  para brindar protección a cada LED, esto justifica el hecho de no ver un parpadeo en cada segmento. Luego, el uso de las interrupciones para este caso (y los demás) presentan el resultado esperado a la hora de presionar el botón con carga alta.

Ahora, la siguiente prueba que se realizó fue la sincronización de los tiempos con los LEDs correspondientes para las 4 etapas de la lavadora. Esto por medio de una máquina de estados para realizar cada transición tal como se muestra en la figura 19.

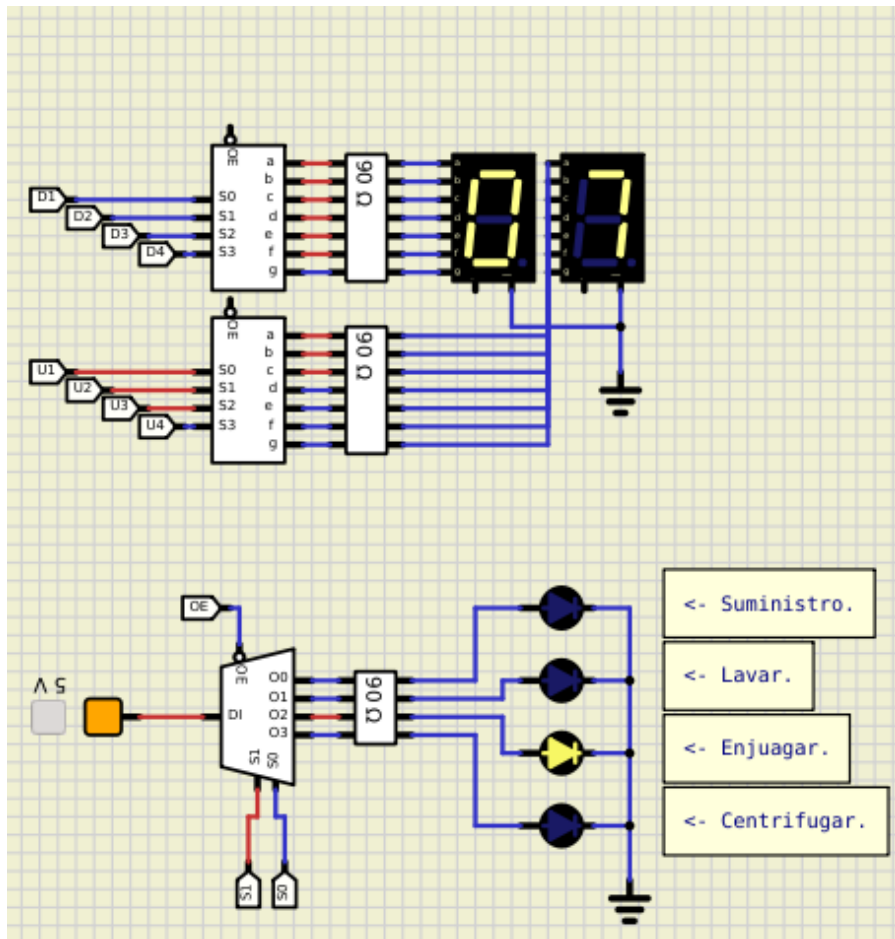


Figura 19: Transición de estados de la lavadora.

Hecho lo anterior, es posible ya mostrar el diseño completo del circuito que simula el comportamiento de una lavadora.

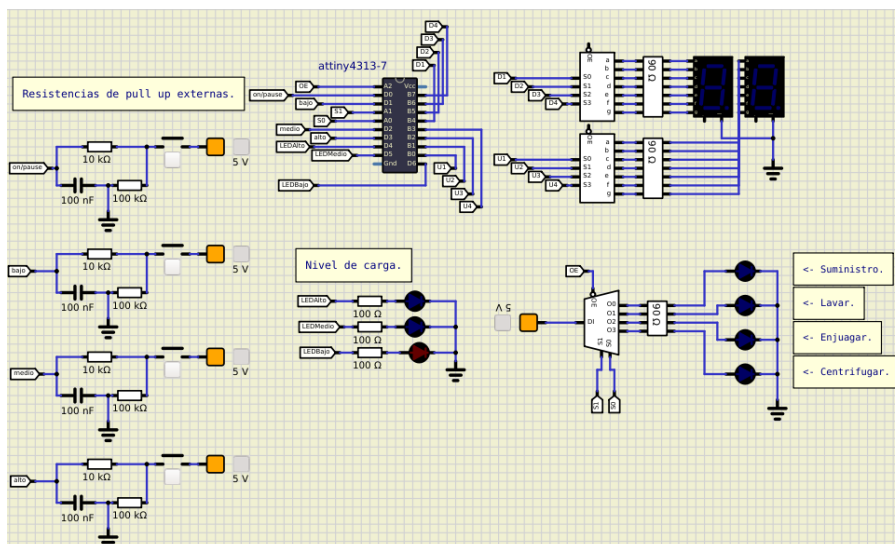


Figura 20: Diseño del circuito completo.

La programación de las funciones que hicieron este comportamiento posible se explicarán en la sección de resultados por medio de diagramas e imágenes. Por razones de tiempo, la última etapa del botón de pausa no se realizó, porque ya no hay más pines disponibles y se tendría que



colocar un demultiplexor en alguna parte del circuito para lograr esto, eso implicaría muchos cambios en el código ya realizado.

### 3. Desarrollo/Análisis

En esta sección, se hablará sobre las conexiones realizadas en el mcu y los diagramas de las principales funciones que componen el circuito de la lavadora.

Lo más sencillo fue declarar todos los pines B como salidas, porque es donde van conectados los Bcds en cascada con resistencias y los displays. Para declarar esos pines como salidas basta con escribir `DDRB = 0xff`. Ahora, con respecto a los botones, son partes del circuito que se encargarán de causar las interrupciones para hacer funcionar el circuito, en cuanto a la carga baja, media y alta, se conectaron a los pines D1, D2 y D3 respectivamente. A cada una de ellas fue necesario habilitar su comando de interrupción adecuado. Revisando la hoja del fabricante, para la carga baja, se debe habilitar el `PCMSK2` con el macro `PCINT12` y el `GIMSK` con el macro `PCIE2` para surtir efecto en este pin. En la carga media y alta, por medio de `GIMSK` se debe habilitar la interrupción con la macro `INT0` y `INT1`, esto hace que los botones trabajen apropiadamente. Ahora, en cuanto al `ISR` para cada uno de estos pines corresponden a las macros `PCINT2_vect`, `INT0_vect` y `INT1_vect`. Dentro de estas interrupciones, hay una variable con el nombre: `tiempo_inicio` que se iguala al tiempo total para cada carga, donde previamente se realizó un `#define` con dichas magnitudes, además, se hace un llamado a la función `led_carga` que se representa por el siguiente diagrama:

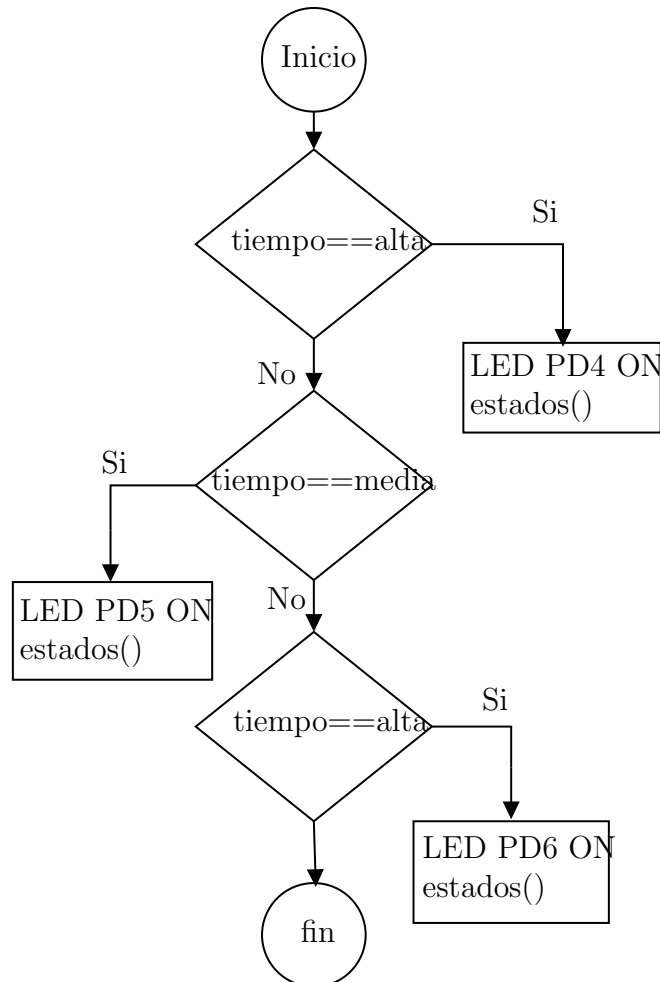


Figura 21: Diagrama de flujo función `led_carga`.

El diagrama anterior permite encender los LEDs con respecto al tiempo ya asignado en el enunciado del laboratorio, note que dentro de esta función hay un llamado a otra función: `estados`, esta es la máquina de estados para realizar la transición del encendido de los LEDs

dependiendo si está en suministro, lavar, enjuague o centrifugar. El siguiente diagrama de flujo resume el comportamiento de esta función.

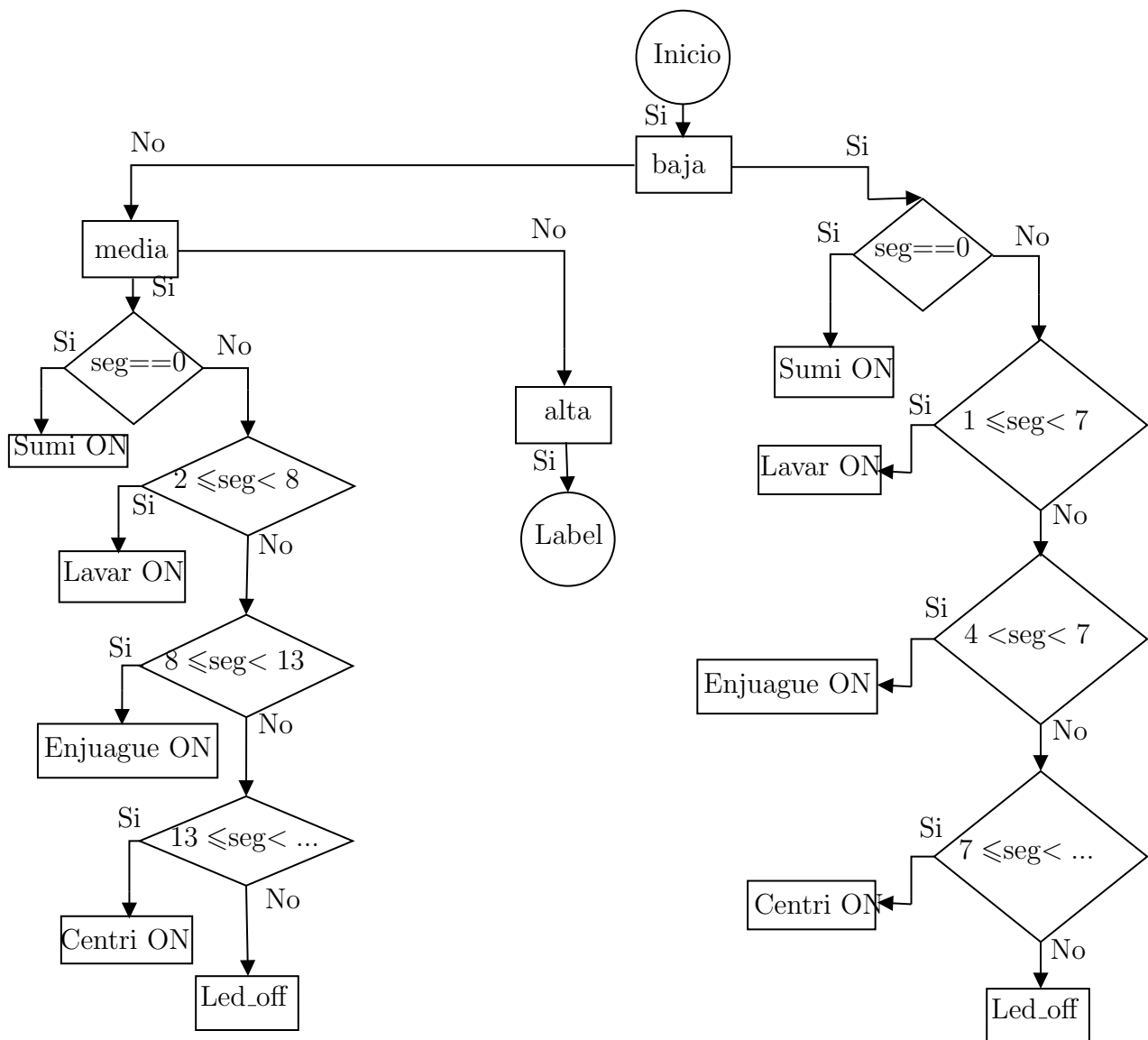


Figura 22: Máquina de estados transición de LEDs.

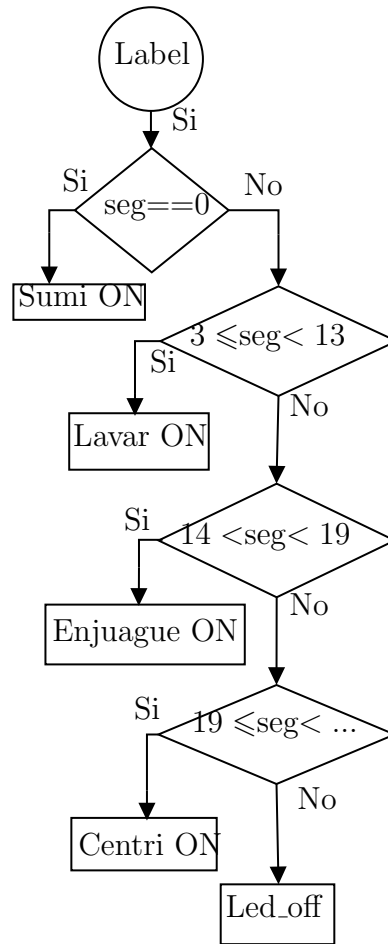


Figura 23: Conituación máquina de estados transición de LEDs.

Esta máquina de estados se encarga de administrar los tiempos para cada carga, por lo que se define un intervalo de tiempo para cada etapa, esto con base al tiempo ya establecido. Entonces, una vez que se cumple cada desigualdad, se apaga el LED actual y se enciende el siguiente LED con el estado correspondiente. Esto para cada situación, dependiendo de la carga que se le asigne a la lavadora. Ahora, hay dos detalles en esta máquina de estados, lo primero es que para referescar los LEDs se hace uso de la interrupción `TIMER0_COMPA_vect` y esto con base al tiempo asignado, esto permite también llevar una cuenta de los segundos que pasan. Claramente hay que hacer las configuraciones para lograr esto, lo primero es con `TCCR0A` y la macro `WGM12`, luego la variable `OCR1A` para obtener un 1 segundo, el prescaler de 256 y habilitar la interrupción. Esta interrupción se compone de una función: `display`, la cual es encargada de mostrar la cuenta regresiva de los números mostrados en los displays, el diagrama de flujo de esta función se muestra a continuación,

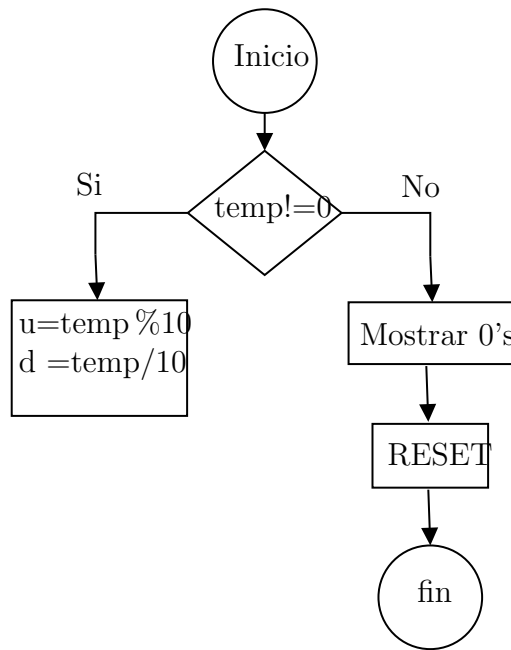


Figura 24: Función display

de donde primero se define la variable como la resta de esta misma variable con segundos, luego con ayuda de condicional se comprueba que el tiempo de carga no sea 0 para poder separar unidades y decenas del número, así, se toman los 4 bits LSB y se colocan en las unidades, mientras que los 4 bits MSB en las decenas, se muestran los ceros y finalmente se reinician las variables a 0. Lo segundo es que se muestra la función: `led_off` que se encarga de limpiar los puertos donde están conectados las cargas y los estados, en otras palabras apaga los LEDs que indican el comportamiento actual de la lavadora.

Ya con todo lo realizado al momento, es pertinente ver el funcionamiento general de la lavadora. Se mostrará el funcionamiento de ésta a partir de los 3 puntos de carga, y observar la transición de estados por medio de la cuenta regresiva de los displays.

Para la carga baja, al igual que las otras se conectaron algunos medidores de voltaje y osciloscopios para comprobar que la corriente eléctrica es la adecuada, y demostrar que los LEDs se comportan tal como se esperaba.

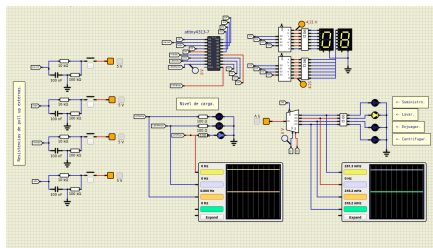


Figura 25: Carga baja lavándose.

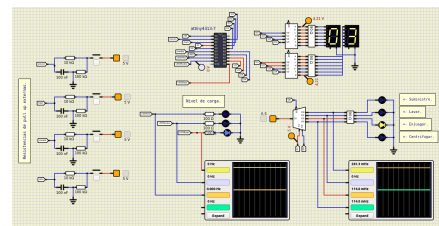


Figura 26: Carga baja lavándose parte 2.

Al presionar el botón bajo, note que el fluido de corriente se va para el pin D6, y las combinaciones de los displays se envía a las etiquetas S0 y S1, que cambiarán conforme avancen los estados. También, en el osciloscopio conectado en la sección de nivel de carga, note que la señal de LEDBajo se mantiene en alto, mientras que las demás en bajo. Luego, en el siguiente osciloscopio, se muestra la cuenta regresiva del estado de suministro, la cual es de 1 segundo, de manera inmediata se tiene el estado de lavar que toma 3 segundos, para el estado de lavar, luego el siguiente estado de enjuagar que toma 2 segundos, y esto es consistente con el tiempo que le pertenece al estado de centrifugar, o sea, 3 segundos y con esto concluye el comportamiento de

la carga baja. El otro detalle es que las magnitudes de las tensiones no superaron los valores máximos de los componentes electrónicos, por tanto, están seguros y no tendrán ningún daño.

Para cuando se presiona el botón de carga media, se obtiene el mismo comportamiento anterior, con la diferencia en los tiempos de cada estado de la lavadora. Se inicia con el estado de suministro, que debe durar 2 segundos, tanto el LED como la señal del osciloscopio se mantienen en alto y tensión eléctrica presente en el pin D5 es la adecuada para un buen funcionamiento, además, las interrupciones para que este evento sea acertado fueron ya establecidas.

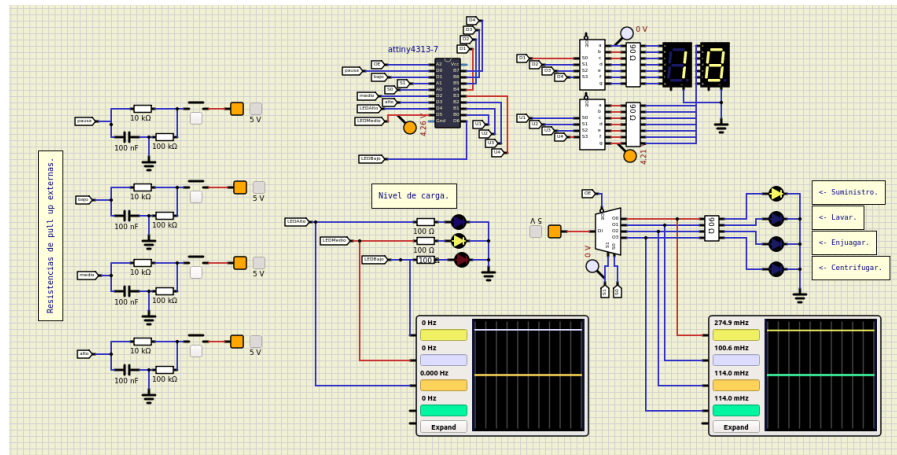


Figura 27: Estado suministro, carga media.

Ahora, gracias a la máquina de estados programada es posible observar el siguiente estado:  
lavar,

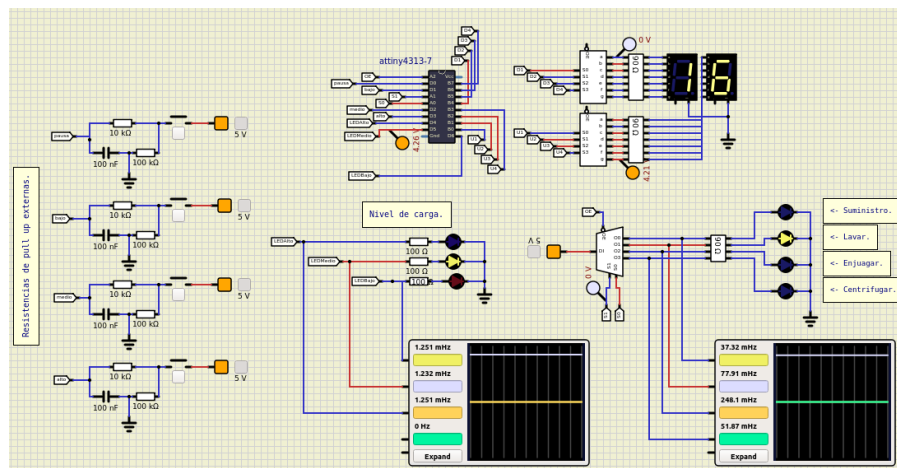


Figura 28: Estado lavar, carga media.

dicha máquina es la que decide si pasar o no a un estado al otro por medio de un intervalo de tiempo ya establecido , así para cada uno de los estados de lavado. Para la etapa de enjuague se tiene la cuenta correctamente, es decir, dura los 4 segundos y las tensiones eléctricas funcionan apropiadamente.

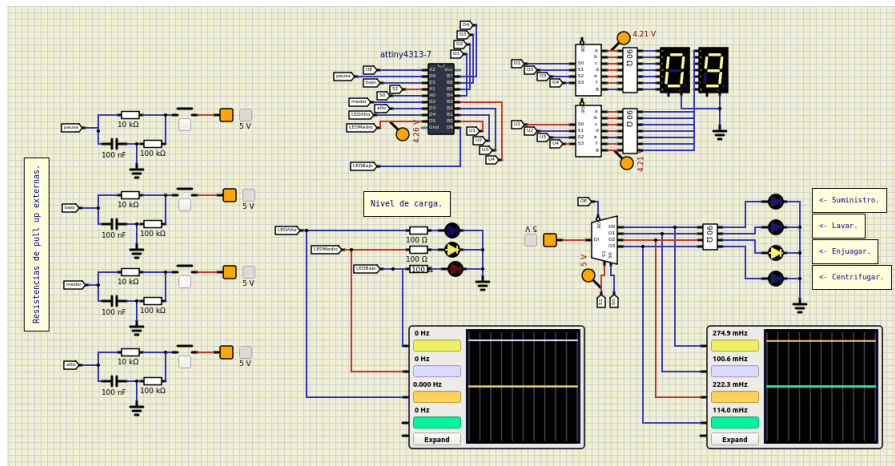


Figura 29: Estado enjuagar, carga media.

Por último, el manejo de la rutina por medio de `ISR(TIMER1_COMPA_vect)` y `ISR(TIMER0_COMPA_vect)` fue posible tener la actualización de los LEDs conforme avanza el tiempo, de lo contrario, los intervalos de tiempo ya establecidos no harían nada. Gracias a esto se obtiene el estado de centrifugar.

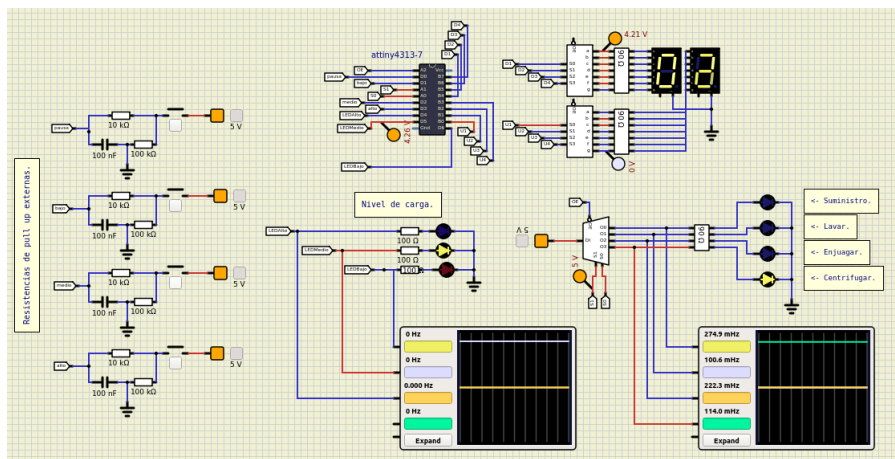


Figura 30: Estado centrifugar, carga media.

Finalmente, cuando se presiona el botón de carga alta, la señal de interrupción se envía correctamente, las etiquetas de los LEDs para esta carga todos están activos, las señales del osciloscopio se respetan, así como la transición de los estados. La tensión eléctrica presente en los displays es de 4,21 V es la adecuada porque éstos no parpadean, esto quiere decir que nos les está llegando corriente muy alta. Por lo que se concluye que el programa es capaz de reconocer cuando se le coloca una carga alta, tal como se muestra en la siguiente transición de los estados.

- Estado suministro:

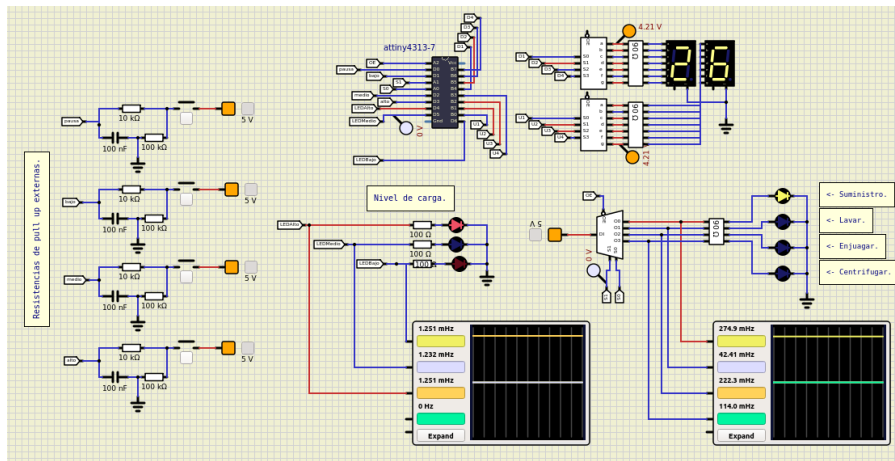


Figura 31: Estado suministro, carga alta.

- Estado lavar:

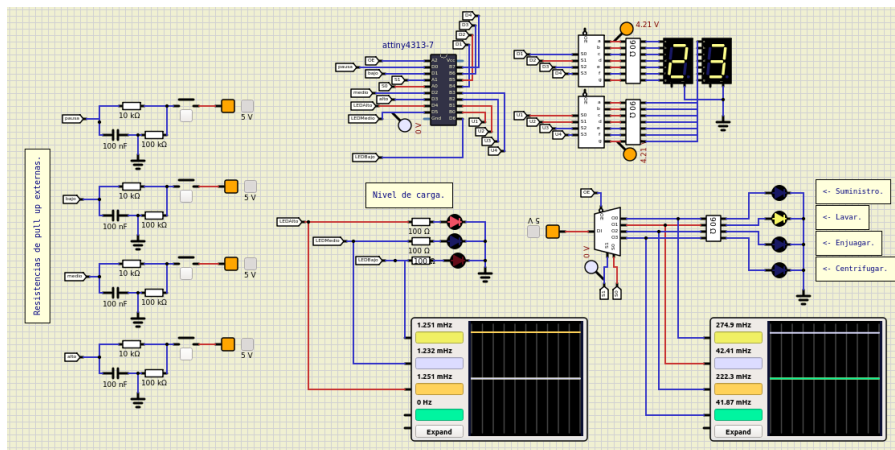


Figura 32: Estado lavar, carga alta.

- Estado enjuague:

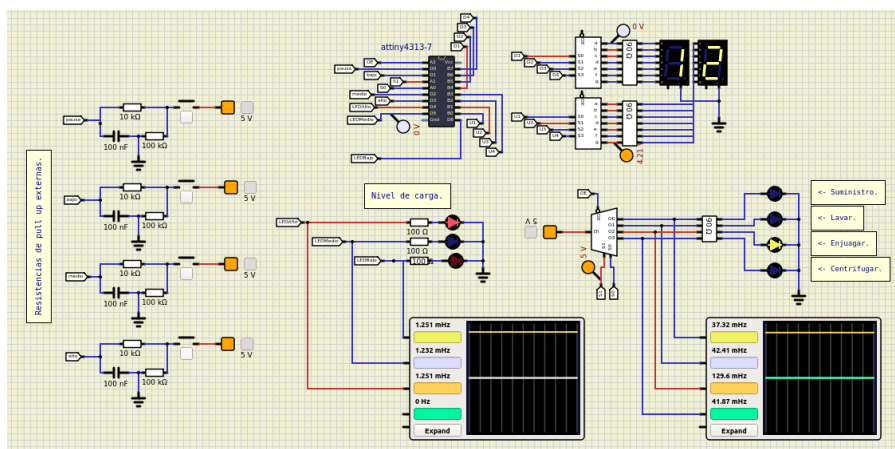


Figura 33: Estado enjuague, carga alta.

- Estado centrifugar:



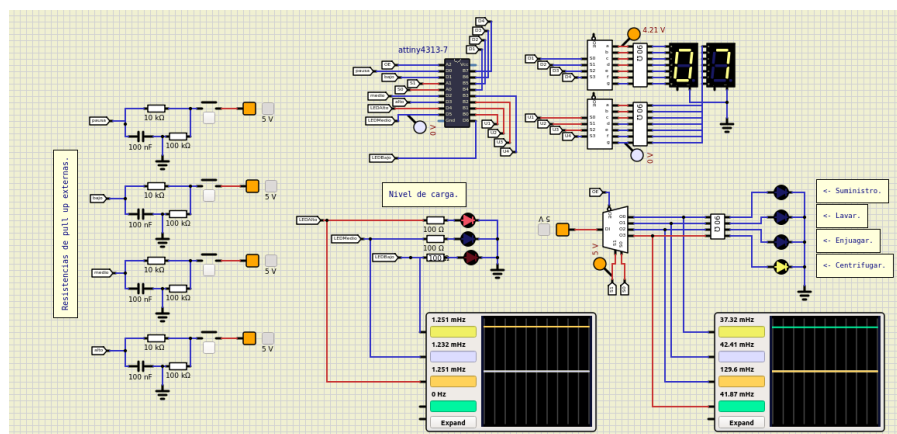


Figura 34: Estado centrifugar, carga alta.

En resumen, el haber configurado los temporizadores: `Timer/Counter0` y `Timer/Counter1` para hacer uso de CTC, los registros `TCCR04`, `WGM01` junto con un valor de comparación en las macros `OCR0A` y `OCR1A`, así como el uso de pre-escaler con sus respectivos registros, esto para reducir la señal eléctrica de alta frecuencia a una frecuencia menor por medio de una división [1], y también el uso de las interrupciones para cada timer. Esto en conjunto, con las máquinas de estado y las demás funciones permiten el buen funcionamiento de la lavadora automática.

## 4. Conclusiones y recomendaciones

Para concluir, en este trabajo fue posible entender el comportamiento de un MCU por medio del uso de interrupciones, lo que es una manera avanzada para controlar comportamientos de otros dispositivos electrónicos, por ejemplo: LEDs y displays de 7 segmentos. El haber realizado una lectura detenida de la hoja del fabricante del MCU y luego realizar pruebas excesivas para comprender el funcionamiento de cada pin fue algo muy importante para darle sentido a lo solicitado en este trabajo. La falta de literatura disponible hizo que se realizaran muchos experimentos de acuerdo a la necesidad de este laboratorio, no se escatimó el tiempo dedicado a las muchas depuraciones que se realizaron, no obstante, fueron valiosas y poder tener una simulación adecuada para esta lavadora, pues por falta de tiempo no se logró la optimización esperado en el uso de los pines, es decir, hicieron falta para poder implementar el botón de inicio y pausa, por lo que la lavadora automática está programada de tal que seleccionar la carga ésta inicia su trabajo y se detiene hasta cuando termine el tiempo asignado para cada etapa de lavado. Lo anterior se pudo haber arreglado implementando otro demultiplexor para generar las salidas de los LEDs de las cargas, ya que con las 3 salidas de los pines D4, D5 y D6 se pudo tener  $2^3 = 8$  salidas o inclusive solamente trabajar con 2 salidas similar al caso de los estados, sin embargo no se implementó porque habría que cambiar el código implementado y como se mencionó anteriormente hubo una limitación de tiempo. Mucho del tiempo perdido fue debido a que se intentó hacer que los números aparecieran con solamente 5 bits en lugar de los 8 utilizados, uno de las señales de los 5 bits se invertía cada vez que se llamaba el temporizador de los displays de esta manera solo se activaba uno en un instante dado, sin embargo, esta implementación no se logró porque el número de las decenas tomaba el de las unidades.

Aún así, el trabajo realizado es satisfactorio en cuanto a funcionamiento porque las interrupciones funcionaron correctamente, es decir, los LEDs responden a ellas, los temporizadores respetan los tiempos de ejecución, en ese sentido, el trabajo fue impecable. Como recomendaciones para este trabajo:

- Hacer todas las depuraciones posibles para entender el funcionamiento de lo que se está realizando.
- Revisar las configuraciones de las interrupciones, porque es posible que se olvidé realizar alguna habilitación y esto hace que se obtenga el resultado no esperado, y el compilador tampoco se lo va a decir. En ese sentido, revisar muy bien que botones son los que se usarán para interrupciones y cuales serán simplemente salidas.
- La siguiente página fue de mucha utilidad para el cálculo de los tiempos de los temporizadores:  
<https://eleccelerator.com/avr-timer-calculator/>

## Referencias

- [1] Atmel. Features attiny2313a/attiny4313. Atmel, <https://ww1.microchip.com/downloads/en/DeviceDoc/doc8246.pdf>, Agosto 2017. Accedido en enero de 2024.
- [2] Mobile education and engineer. Decodificadores de bcd a 7 segmentos. Robot, <https://wp.7robot.net/decodificadores-de-bcd-a-7-segmentos/>, abril 2017. Accedido en enero de 2024.

## 5. Anexos

Aquí van las hojas del fabricante de los componentes usados para este laboratorio.

## Features

- High Performance, Low Power AVR<sup>®</sup> 8-Bit Microcontroller
- Advanced RISC Architecture
  - 120 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
- Data and Non-volatile Program and Data Memories
  - 2/4K Bytes of In-System Self Programmable Flash
    - Endurance 10,000 Write/Erase Cycles
  - 128/256 Bytes In-System Programmable EEPROM
    - Endurance: 100,000 Write/Erase Cycles
  - 128/256 Bytes Internal SRAM
  - Programming Lock for Flash Program and EEPROM Data Security
- Peripheral Features
  - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Modes
  - Four PWM Channels
  - On-chip Analog Comparator
  - Programmable Watchdog Timer with On-chip Oscillator
  - USI – Universal Serial Interface
  - Full Duplex USART
- Special Microcontroller Features
  - debugWIRE On-chip Debugging
  - In-System Programmable via SPI Port
  - External and Internal Interrupt Sources
  - Low-power Idle, Power-down, and Standby Modes
  - Enhanced Power-on Reset Circuit
  - Programmable Brown-out Detection Circuit
  - Internal Calibrated Oscillator
- I/O and Packages
  - 18 Programmable I/O Lines
  - 20-pin PDIP, 20-pin SOIC, 20-pad MLF/VQFN
- Operating Voltage
  - 1.8 – 5.5V
- Speed Grades
  - 0 – 4 MHz @ 1.8 – 5.5V
  - 0 – 10 MHz @ 2.7 – 5.5V
  - 0 – 20 MHz @ 4.5 – 5.5V
- Industrial Temperature Range: -40°C to +85°C
- Low Power Consumption
  - Active Mode
    - 190 µA at 1.8V and 1MHz
  - Idle Mode
    - 24 µA at 1.8V and 1MHz
  - Power-down Mode
    - 0.1 µA at 1.8V and +25°C



**8-bit AVR<sup>®</sup>  
Microcontroller  
with 2/4K Bytes  
In-System  
Programmable  
Flash**

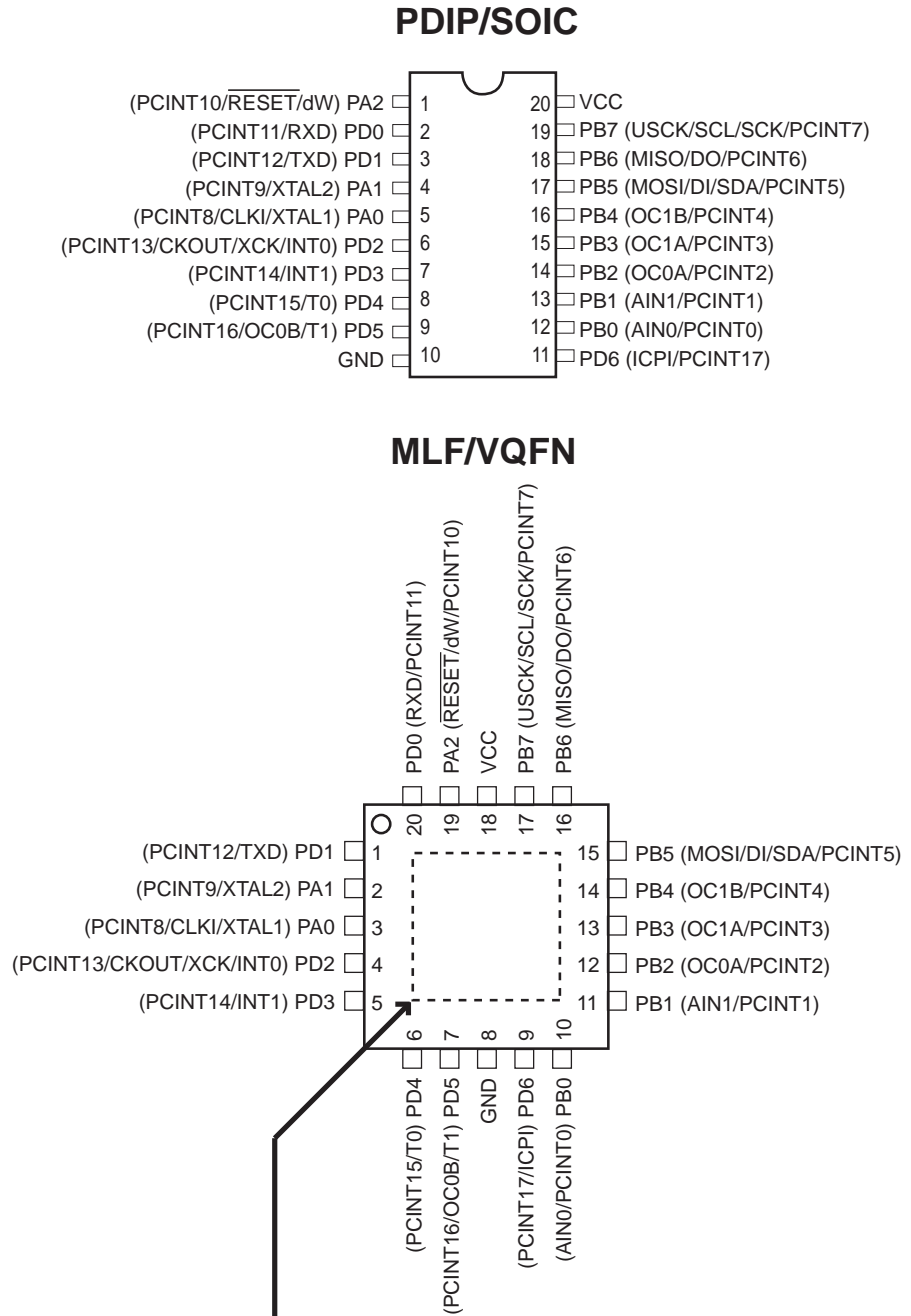
**ATtiny2313A  
ATtiny4313**

Rev. 8246B-AVR-09/11



## 1. Pin Configurations

Figure 1-1. Pinout ATtiny2313A/4313



## 1.1 Pin Descriptions

### 1.1.1 VCC

Digital supply voltage.

### 1.1.2 GND

Ground.

### 1.1.3 Port A (PA2..PA0)

Port A is a 3-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability, except PA2 which has the  $\overline{\text{RESET}}$  capability. To use pin PA2 as I/O pin, instead of RESET pin, program ("0") RSTDISBL fuse. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATtiny2313A/4313 as listed on [page 62](#).

### 1.1.4 Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATtiny2313A/4313 as listed on [page 63](#).

### 1.1.5 Port D (PD6..PD0)

Port D is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATtiny2313A/4313 as listed on [page 67](#).

### 1.1.6 $\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running and provided that the reset pin has not been disabled. The minimum pulse length is given in [Table 22-3 on page 201](#). Shorter pulses are not guaranteed to generate a reset. The Reset Input is an alternate function for PA2 and dW.

The reset pin can also be used as a (weak) I/O pin.

### 1.1.7 XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit. XTAL1 is an alternate function for PA0.



#### 1.1.8 XTAL2

Output from the inverting Oscillator amplifier. XTAL2 is an alternate function for PA1.

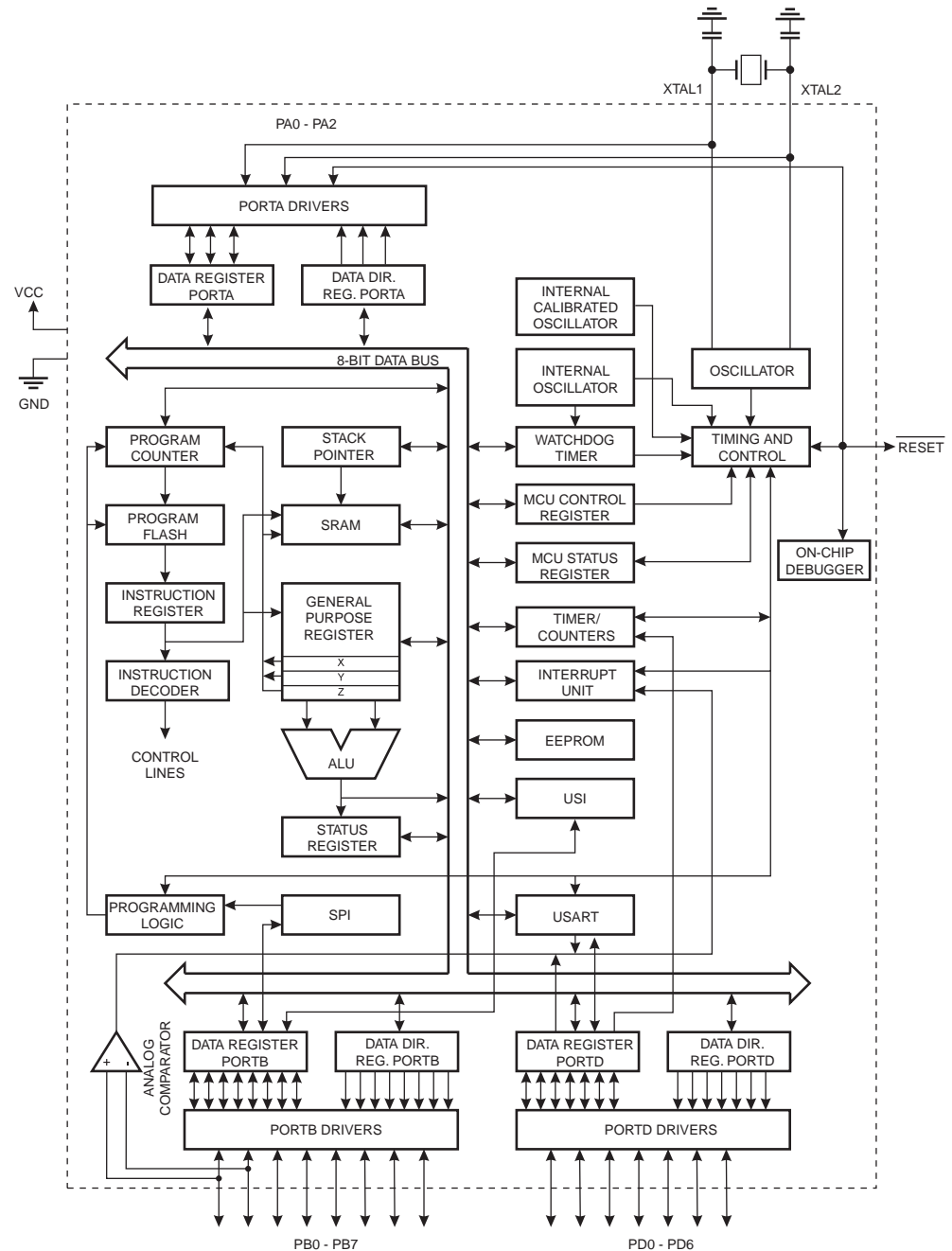


## 2. Overview

The ATtiny2313A/4313 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATtiny2313A/4313 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

### 2.1 Block Diagram

Figure 2-1. Block Diagram





### 10.2.1 Alternate Functions of Port A

The Port A pins with alternate function are shown in [Table 10-3](#).

**Table 10-3.** Port A Pins Alternate Functions

Port Pin	Alternate Function
PA0	XTAL1: Crystal Oscillator Input CLKI: External Clock Input PCINT8: Pin Change Interrupt 1, Source 8
PA1	XTAL2: Crystal Oscillator Output PCINT9: Pin Change Interrupt 1, Source 9
PA2	$\overline{\text{RESET}}$ : Reset pin dW: debugWire I/O PCINT10: Pin Change Interrupt 1, Source 10

- **Port A, Bit 0 – XTAL1/CLKI/PCINT8**

- XTAL1: Chip Clock Oscillator pin 1. Used for all chip clock sources except internal calibratable RC oscillator. When used as a clock pin, the pin can not be used as an I/O pin. When using internal calibratable RC Oscillator as a chip clock source, PA0 serves as an ordinary I/O pin.
- CLKI: Clock Input from an external clock source, see [“External Clock” on page 27](#).
- PCINT8: Pin Change Interrupt source 8. The PA0 pin can serve as an external interrupt source for pin change interrupt 1.

- **Port A, Bit 1 – XTAL2/PCINT9**

- XTAL2: Chip Clock Oscillator pin 2. Used as clock pin for all chip clock sources except internal calibratable RC Oscillator and external clock. When used as a clock pin, the pin can not be used as an I/O pin. When using internal calibratable RC Oscillator or External clock as a Chip clock sources, PA1 serves as an ordinary I/O pin.
- PCINT9: Pin Change Interrupt source 9. The PA1 pin can serve as an external interrupt source for pin change interrupt 1.

- **Port A, Bit 2 –  $\overline{\text{RESET}}$ /dW/PCINT10**

- $\overline{\text{RESET}}$ : External  $\overline{\text{Reset}}$  input is active low and enabled by unprogramming (“1”) the RSTDISBL Fuse. Pullup is activated and output driver and digital input are deactivated when the pin is used as the  $\overline{\text{RESET}}$  pin.
- dW: When the debugWIRE Enable (DWEN) Fuse is programmed and Lock bits are unprogrammed, the debugWIRE system within the target device is activated. The RESET port pin is configured as a wire-AND (open-drain) bi-directional I/O pin with pull-up enabled and becomes the communication gateway between target and emulator.
- PCINT10: Pin Change Interrupt source 10. The PA2 pin can serve as an external interrupt source for pin change interrupt 1.

Table 10-4 relates the alternate functions of Port A to the overriding signals shown in Figure 10-5 on page 60.

**Table 10-4.** Overriding Signals for Alternate Functions in PA2..PA0

Signal Name	PA2/RESET/dW/PCINT10	PA1/XTAL2/PCINT9	PA0/XTAL1/PCINT8
PUOE	$\overline{\text{RSTDISBL}}^{(1)} + \text{DEBUGWIRE\_ENABLE}^{(2)}$	EXT_OSC <sup>(3)</sup>	EXT_CLOCK <sup>(4)</sup> + EXT_OSC <sup>(3)</sup>
PUOV	1	0	0
DDOE	$\overline{\text{RSTDISBL}}^{(1)} + \text{DEBUGWIRE\_ENABLE}^{(2)}$	EXT_OSC <sup>(3)</sup>	EXT_CLOCK <sup>(4)</sup> + EXT_OSC <sup>(3)</sup>
DDOV	DEBUGWIRE_ENABLE <sup>(2)</sup> • debugWire Transmit	0	0
PVOE	$\overline{\text{RSTDISBL}}^{(1)} + \text{DEBUGWIRE\_ENABLE}^{(2)}$	EXT_OSC <sup>(3)</sup>	EXT_CLOCK <sup>(4)</sup> + EXT_OSC <sup>(3)</sup>
PVOV	0	0	0
PTOE	0	0	0
DIEOE	$\overline{\text{RSTDISBL}}^{(1)} + \text{DEBUGWIRE\_ENABLE}^{(2)} + \text{PCINT10} \cdot \text{PCIE1}$	EXT_OSC <sup>(3)</sup> + PCINT9 • PCIE1	EXT_CLOCK <sup>(4)</sup> + EXT_OSC <sup>(3)</sup> + (PCINT8 • PCIE1)
DIEOV	DEBUGWIRE_ENABLE <sup>(2)</sup> + ( $\overline{\text{RSTDISBL}}^{(1)} \cdot \text{PCINT10}$ • PCIE1)	$\overline{\text{EXT\_OSC}}^{(3)} + \text{PCINT9}$ • PCIE1	(EXT_CLOCK <sup>(4)</sup> • $\overline{\text{PWR\_DOWN}}$ ) + (EXT_CLOCK <sup>(4)</sup> • EXT_OSC <sup>(3)</sup> • PCINT8 • PCIE1)
DI	dW/PCINT10 Input	PCINT9 Input	CLKI/PCINT8 Input
AIO		XTAL2	XTAL1

- Notes:
1. RSTDISBL is 1 when the fuse is "0" (Programmed).
  2. DebugWIRE is enabled when DWEN Fuse is programmed and Lock bits are unprogrammed.
  3. EXT\_OSC = crystal oscillator or low frequency crystal oscillator is selected as system clock.
  4. EXT\_CLOCK = external clock is selected as system clock.

## 10.2.2 Alternate Functions of Port B

The Port B pins with alternate function are shown in Table 10-5.

**Table 10-5.** Port B Pins Alternate Functions

Port Pin	Alternate Function
PB0	AIN0: Analog Comparator, Positive Input PCINT0: Pin Change Interrupt 0, Source 0
PB1	AIN1: Analog Comparator, Negative Input PCINT1: Pin Change Interrupt 0, Source 1
PB2	OC0A: Timer/Counter0 Compare Match A Output PCINT2: Pin Change Interrupt 0, Source 2
PB3	OC1A: Timer/Counter1 Compare Match A Output PCINT3: Pin Change Interrupt 0, Source 3



**Table 10-5.** Port B Pins Alternate Functions

Port Pin	Alternate Function
PB4	OC1B: Timer/Counter1 Compare Match B Output PCINT4: Pin Change Interrupt 0, Source 4
PB5	DI: USI Data Input (Three Wire Mode) SDA: USI Data Input (Two Wire Mode) PCINT5: Pin Change Interrupt 0, Source 5
PB6	DO: USI Data Output (Three Wire Mode) PCINT6: Pin Change Interrupt 0, Source 6
PB7	USCK: USI Clock (Three Wire Mode) SCL : USI Clock (Two Wire Mode) PCINT7: Pin Change Interrupt 0, Source 7

- **Port B, Bit 0 – AIN0/PCINT0**

- AIN0: Analog Comparator Positive input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the Analog Comparator.
- PCINT0: Pin Change Interrupt Source 0. The PB0 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 1 – AIN1/PCINT1**

- AIN1: Analog Comparator Negative input. Configure the port pin as input with the internal pull-up switched off to avoid the digital port function from interfering with the function of the analog comparator.
- PCINT1: Pin Change Interrupt Source 1. The PB1 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 2 – OC0A/PCINT2**

- OC0A: Output Compare Match A output. The PB2 pin can serve as an external output for the Timer/Counter0 Output Compare A. The pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC0A pin is also the output pin for the PWM mode timer function.
- PCINT2: Pin Change Interrupt Source 2. The PB2 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 3 – OC1A/PCINT3**

- OC1A: Output Compare Match A output: The PB3 pin can serve as an external output for the Timer/Counter1 Output Compare A. The pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.
- PCINT3: Pin Change Interrupt Source 3: The PB3 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 4 – OC1B/PCINT4**

- OC1B: Output Compare Match B output: The PB4 pin can serve as an external output for the Timer/Counter1 Output Compare B. The pin has to be configured as an output (DDB4 set

(one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

- PCINT4: Pin Change Interrupt Source 4. The PB4 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 5 – DI/SDA/PCINT5**

- DI: Three-wire mode Universal Serial Interface Data input. Three-wire mode does not override normal port functions, so pin must be configured as an input. SDA: Two-wire mode Serial Interface Data.
- PCINT5: Pin Change Interrupt Source 5. The PB5 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 6 – DO/PCINT6**

- DO: Three-wire mode Universal Serial Interface Data output. Three-wire mode Data output overrides PORTB6 value and it is driven to the port when data direction bit DDB6 is set (one). However the PORTB6 bit still controls the pull-up enabling pull-up, if direction is input and PORTB6 is set (one).
- PCINT6: Pin Change Interrupt Source 6. The PB6 pin can serve as an external interrupt source for pin change interrupt 0.

- **Port B, Bit 7 – USCK/SCL/PCINT7**

- USCK: Three-wire mode Universal Serial Interface Clock.
- SCL: Two-wire mode Serial Clock for USI Two-wire mode.
- PCINT7: Pin Change Interrupt source 7. The PB7 pin can serve as an external interrupt source for pin change interrupt 0.

Table 10-6 and Table 10-7 relate the alternate functions of Port B to the overriding signals shown in Figure 10-5 on page 60. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

**Table 10-6.** Overriding Signals for Alternate Functions in PB7..PB4

Signal Name	PB7/USCK/ SCL/PCINT7	PB6/DO/PCINT6	PB5/SDA/ DI/PCINT5	PB4/OC1B/ PCINT4
PUOE	USI_TWO_WIRE	0	0	0
PUOV	0	0	0	0
DDOE	USI_TWO_WIRE	0	USI_TWO_WIRE	0
DDOV	$(\overline{\text{USI\_SCL\_HOLD}} + \overline{\text{PORTB7}}) \cdot \text{DDB7}$	0	$(\overline{\text{SDA}} + \overline{\text{PORTB5}}) \cdot \text{DDB5}$	0
PVOE	$\text{USI\_TWO\_WIRE} \cdot \text{DDB7}$	USI_THREE_WIRE	$\text{USI\_TWO\_WIRE} \cdot \text{DDB5}$	OC1B_PVOE
PVOV	0	DO	0	0OC1B_PVOV
PTOE	USI_PTOE	0	0	0
DIEOE	$(\text{PCINT7} \cdot \text{PCIE}) + \text{USISIE}$	$(\text{PCINT6} \cdot \text{PCIE})$	$(\text{PCINT5} \cdot \text{PCIE}) + \text{USISIE}$	$(\text{PCINT4} \cdot \text{PCIE})$
DIEOV	1	1	1	1
DI	PCINT7 Input USCK Input SCL Input	PCINT6 Input	PCINT5 Input SDA Input DI Input	PCINT4 Input
AIO	—	—	—	—

**Table 10-7.** Overriding Signals for Alternate Functions in PB3..PB0

Signal Name	PB3/OC1A/ PCINT3	PB2/OC0A/ PCINT2	PB1/AIN1/ PCINT1	PB0/AIN0/ PCINT0
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC1A_PVOE	OC0A_PVOE	0	0
PVOV	OC1A_PVOV	OC0A_PVOV	0	0
PTOE	0	0	0	0
DIEOE	$(\text{PCINT3} \cdot \text{PCIE})$	$(\text{PCINT2} \cdot \text{PCIE})$	$(\text{PCINT1} \cdot \text{PCIE})$	$(\text{PCINT0} \cdot \text{PCIE})$
DIEOV	1	1	1	1
DI	PCINT7 Input	PCINT6 Input	PCINT5 Input	PCINT4 Input
AIO	—	—	AIN1	AIN0

## 10.2.3 Alternate Functions of Port D

The Port D pins with alternate functions are shown in [Table 10-8](#).

**Table 10-8.** Port D Pins Alternate Functions

Port Pin	Alternate Function
PD0	RXD: UART Data Receiver PCINT11:Pin Change Interrupt 2, Source 11
PD1	TXD: UART Data Transmitter PCINT12:Pin Change Interrupt 2, Source 12
PD2	INT0: External Interrupt 0 Input XCK: USART Transfer Clock CKOUT: System Clock Output PCINT13:Pin Change Interrupt 2, Source 13
PD3	INT1: External Interrupt 1 Input PCINT14:Pin Change Interrupt 2, Source 14
PD4	T0: Timer/Counter0 Clock Source PCINT15:Pin Change Interrupt 2, Source 15
PD5	OC0B: Timer/Counter0 Compare Match B output T1: Timer/Counter1 Clock Source PCINT16:Pin Change Interrupt 2, Source 16
PD6	ICPI: Timer/Counter1 Input Capture Pin PCINT17:Pin Change Interrupt 2, Source 17

- **Port D, Bit 0 – RXD/PCINT11**
  - RXD: UART Data Receiver.
  - PCINT11: Pin Change Interrupt Source 11. The PD0 pin can serve as an external interrupt source for pin change interrupt 2.
- **Port D, Bit 1 – TXD/PCINT12**
  - TXD: UART Data Transmitter.
  - PCINT12: Pin Change Interrupt Source 12. The PD1 pin can serve as an external interrupt source for pin change interrupt 2.
- **Port D, Bit 2 – INT0/XCK/CKOUT/PCINT13**
  - INT0: External Interrupt Source 0. The PD2 pin can serve as an external interrupt source to the MCU.
  - XCK: USART Transfer Clock used only by Synchronous Transfer mode.
  - CKOUT: System Clock Output.
  - PCINT13: Pin Change Interrupt Source 13. The PD2 pin can serve as an external interrupt source for pin change interrupt 2.
- **Port D, Bit 3 – INT1/PCINT14**
  - INT1: External Interrupt Source 1. The PD3 pin can serve as an external interrupt source to the MCU.
  - PCINT14: Pin Change Interrupt Source 14. The PD3 pin can serve as an external interrupt source for pin change interrupt 2.

- **Port D, Bit 4 – T0/PCINT15**

- T0: Timer/Counter0 External Counter Clock input is enabled by setting (one) the bits CS02 and CS01 in the Timer/Counter0 Control Register (TCCR0).
- PCINT15: Pin Change Interrupt Source 15. The PD4 pin can serve as an external interrupt source for pin change interrupt 2.

- **Port D, Bit 5 – OC0B/T1/PCINT16**

- OC0B: Output Compare Match B output: The PD5 pin can serve as an external output for the Timer/Counter0 Output Compare B. The pin has to be configured as an output (DDD5 set (one)) to serve this function. The OC0B pin is also the output pin for the PWM mode timer function.
- T1: Timer/Counter1 External Counter Clock input is enabled by setting (one) the bits CS02 and CS01 in the Timer/Counter1 Control Register (TCCR1).
- PCINT16: Pin Change Interrupt Source 16. The PD5 pin can serve as an external interrupt source for pin change interrupt 2.

- **Port D, Bit 6 – ICPI/PCINT17**

- ICPI: Timer/Counter1 Input Capture Pin. The PD6 pin can act as an Input Capture pin for Timer/Counter1.
- PCINT17: Pin Change Interrupt Source 17. The PD6 pin can serve as an external interrupt source for pin change interrupt 2.

Table 10-9 and Table 10-10 relates the alternate functions of Port D to the overriding signals shown in Figure 10-5 on page 60.

**Table 10-9.** Overriding Signals for Alternate Functions PD6..PD4

Signal Name	PD6/ICPI/PCINT17	PD5/OC1B/T1/PCINT16	PD4/T0/PCINT15
PUOE	0	0	0
PUOV	0	0	0
DDOE	0	0	0
DDOV	0	0	0
PVOE	0	OC1B_PVOE	0
PVOV	0	OC1B_PVOV	0
PTOE	0	0	0
DIEOE	ICPI Enable + PCINT17	T1 Enable + PCINT16	T0 Enable + PCINT15
DIEOV	PCINT17	PCINT16	PCINT15
DI	ICPI Input/PCINT17	T1 Input/PCINT16	T0 Input/PCINT15
AIO	–	–	AIN1



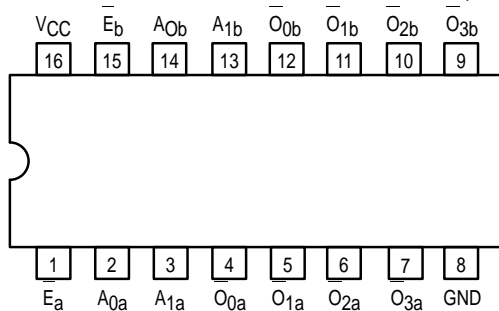


# DUAL 1-OF-4 DECODER/ DEMULTIPLEXER

The LSTTL/MSI SN54/74LS139 is a high speed Dual 1-of-4 Decoder/Demultiplexer. The device has two independent decoders, each accepting two inputs and providing four mutually exclusive active LOW Outputs. Each decoder has an active LOW Enable input which can be used as a data input for a 4-output demultiplexer. Each half of the LS139 can be used as a function generator providing all four minterms of two variables. The LS139 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

- Schottky Process for High Speed
- Multifunction Capability
- Two Completely Independent 1-of-4 Decoders
- Active Low Mutually Exclusive Outputs
- Input Clamp Diodes Limit High Speed Termination Effects
- ESD > 3500 Volts

CONNECTION DIAGRAM DIP (TOP VIEW)



NOTE:  
The Flatpak version  
has the same pinouts  
(Connection Diagram) as  
the Dual In-Line Package.

## PIN NAMES

A<sub>0</sub>, A<sub>1</sub> Address Inputs  
E Enable (Active LOW) Input  
O<sub>0</sub>–O<sub>3</sub> Active LOW Outputs (Note b)

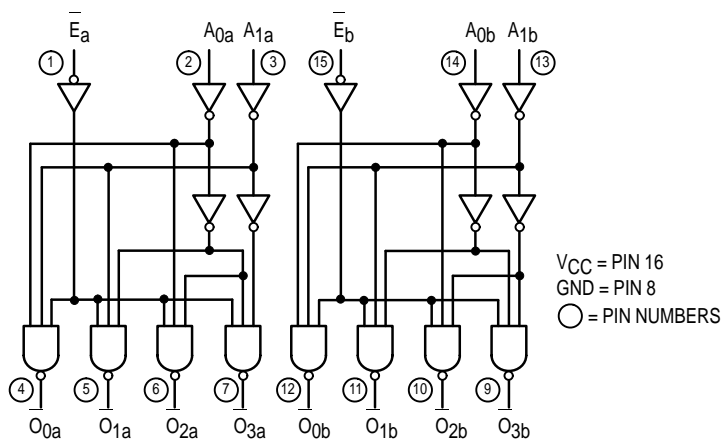
## LOADING (Note a)

HIGH	LOW
0.5 U.L.	0.25 U.L.
0.5 U.L.	0.25 U.L.
10 U.L.	5 (2.5) U.L.

## NOTES:

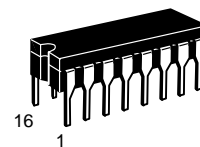
- a) 1 TTL Unit Load (U.L.) = 40  $\mu$ A HIGH/1.6 mA LOW.  
b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

## LOGIC DIAGRAM

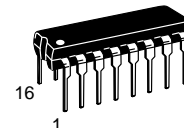


## SN54/74LS139

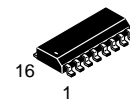
## DUAL 1-OF-4 DECODER/ DEMULTIPLEXER LOW POWER SCHOTTKY



J SUFFIX  
CERAMIC  
CASE 620-09



N SUFFIX  
PLASTIC  
CASE 648-08

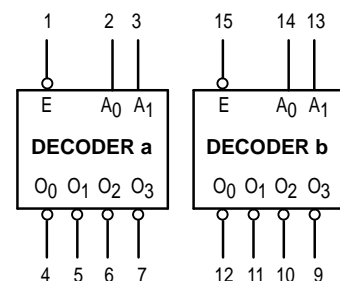


D SUFFIX  
SOIC  
CASE 751B-03

## ORDERING INFORMATION

SN54LSXXXJ Ceramic  
SN74LSXXXN Plastic  
SN74LSXXXD SOIC

## LOGIC SYMBOL



V<sub>CC</sub> = PIN 16  
GND = PIN 8