

UNIVERSIDAD DE COSTA RICA
ESCUELA DE INGENIERÍA ELÉCTRICA
IE0624 - LABORATORIO DE
MICROCONTROLADORES

INFORME DEL LABORATORIO 1
GPIOs, TIMERS Y FSM

PROFESOR
MARCO VILLALTA FALLAS

JOSE DANIEL BLANCO SOLIS, B71137

III 2024

24 DE ENERO DE 2025

Índice

1. Introducción	3
2. Nota teórica	4
2.1. Microcontrolador ATtiny4313	4
2.2. BCD a 7 segmentos	4
2.3. Diseño del circuito	5
3. Desarrollo/Análisis	7
3.1. Diagrama de flujo	7
3.2. Código	8
3.3. Pruebas de funcionamiento	10
4. Conclusiones y Recomendaciones	15

Resumen

Este proyecto describe el diseño, desarrollo e implementación del sistema de control automatizado para una lavadora utilizando un microcontrolador *ATtiny4313*, el objetivo fue construir un sistema capaz de simular las distintas etapas de un ciclo de lavado, incluyendo suministro de agua, lavado, enjuague y centrifugado, se integraron múltiples componentes electrónicos como LEDs indicadores, display de 7 segmentos, botones de control y un motor DC, el diseño y la programación del sistema se basaron en conceptos fundamentales de control, lógica secuencial y manejo de interrupciones.

1. Introducción

Este proyecto tiene como objetivo simular el comportamiento de una lavadora, permitiendo al usuario interactuar con el sistema para seleccionar niveles de agua y controlar el ciclo de lavado, el sistema utiliza un microcontrolador *ATtiny4313* como unidad de control principal, el cual gestiona la secuencia de operaciones y se encarga de interactuar con los periféricos conectados, como LEDs, botones, display de 7 segmentos y un motor DC, además, se diseñaron y probaron tanto el circuito como el software del sistema, logrando una implementación adecuada entre los distintos componentes. Este reporte presenta los aspectos teóricos, técnicos y prácticos del proyecto, junto con un análisis detallado de cada etapa del desarrollo, así como, los resultados obtenidos y las conclusiones del trabajo realizado

2. Nota teórica

2.1. Microcontrolador ATtiny4313

El *ATtiny4313* es un microcontrolador de 8 *bits* basado en la arquitectura AVR, entre sus características relevantes para el uso en este proyecto se encuentran sus puertos configurables para interactuar con sensores, botones, LEDs y otros periféricos, estos puertos permiten manejar las señales digitales necesarias para controlar el sistema [**tiny**].

El microcontrolador dispone de temporizadores integrados que facilitan la ejecución de procesos que requieren un control exacto de la duración, como las diferentes etapas del ciclo de lavado, así mismo, este microcontrolador es compatible con el manejo de interrupciones, lo que permite responder de manera inmediata a eventos generados, como la activación de botones, en la figura 1 se aprecia el esquemático con sus pines etiquetados.

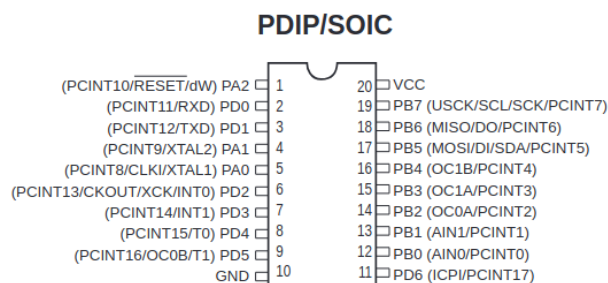


Figura 1: Esquema del *ATtiny4313*.

2.2. BCD a 7 segmentos

El display de 7 segmentos utilizado en este proyecto permite mostrar de manera visual el tiempo transcurrido en cada etapa del ciclo de lavado, este tipo es comúnmente empleado en sistemas electrónicos, en este caso está controlado mediante un convertidor BCD a 7 segmentos.

Este componente traduce las señales binarias enviadas por el microcontrolador a señales adecuadas para encender los segmentos correspondientes del display, dada esta integración, el sistema puede proporcionar información comprensible al usuario sobre el estado del proceso de lavado en tiempo real [**ti**], el esquema del BCD a 7 segmentos se puede ver en la figura 2

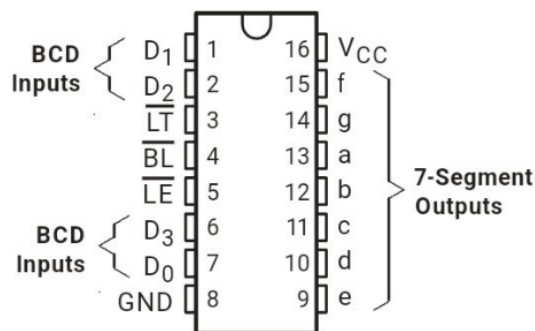


Figura 2: Esquema del BCD a 7 segmentos.

2.3. Diseño del circuito

Cada componente fue seleccionado para garantizar un desempeño óptimo y una integración lo mas simple y eficiente posible, los LEDs indicadores están conectados a los pines *D1*, *D3*, *D4* y *D5* del microcontrolador, permitiendo representar visualmente las diferentes etapas del ciclo de lavado, por otro lado, los botones utilizados para seleccionar el nivel de agua y el inicio del ciclo están conectados a los pines *B5*, *B6*, *B7* y *D2*, respectivamente, estas entradas digitales permiten al usuario interactuar con el sistema y configurar los parámetros deseados para el lavado.

En este caso se tomo en cuenta la reacción que se genera al activar un switch debido a que el contacto entre las piezas de metal no se da de forma instantánea, al considerar que se pueden generar rebotes se realizó un ajuste en la conexión de switch, el cual se puede apreciar en la figura 3, en este caso se planteo un valor de τ suficiente para el microcontrolador, pero que a la vista humana no fuese perceptible.

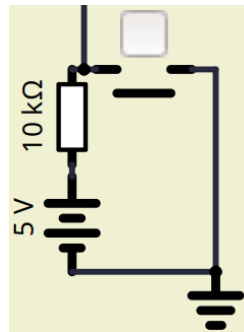


Figura 3: Circuito diseñado para control de rebotes.

Como se puede apreciar en la figura 4 el motor que simula el tambor de la lavadora, está controlado mediante un circuito con transistores conectados al pin *B0* del microcontrolador, asegurando que el mismo opere adecuadamente durante las etapas de lavado, enjuague y centrifugado, el proceso incluyó múltiples ajustes para garantizar que todas las conexiones fueran correctas y que los componentes funcionaran de manera coordinada.

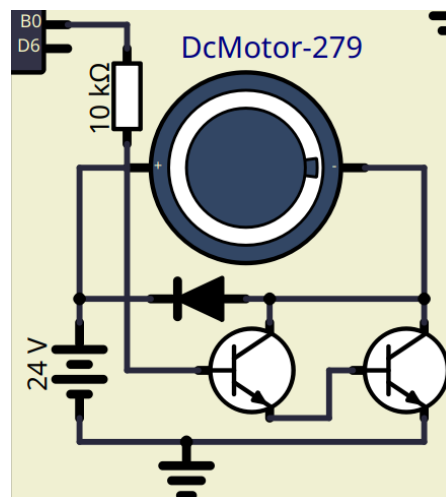


Figura 4: Circuito diseñado para control del motor.

Se utilizó el convertidor BCD a 7 segmentos para manejar el display de 7 segmentos, los pines de entrada de este convertidor $S0$, $S1$, $S2$, $S3$ están conectados a los pines $B1$, $B2$, $B3$ y $B4$ del microcontrolador, mientras que sus salidas están directamente vinculadas a los segmentos del display, esta configuración reduce la complejidad del código y del cableado necesario, entonces, cada vez que el microcontrolador reciba la señal del botón enviará por los pines los valores necesarios para conformar en BCD el número a mostrar, para lo cual el cero corresponde a 0 V y el 1 corresponde a 5 V , también se tomó en cuenta que es necesario conmutar, esto quiere decir que el valor en los pines estará cambiando a velocidad lo suficientemente rápida para que el displays no este visiblemente apagado.

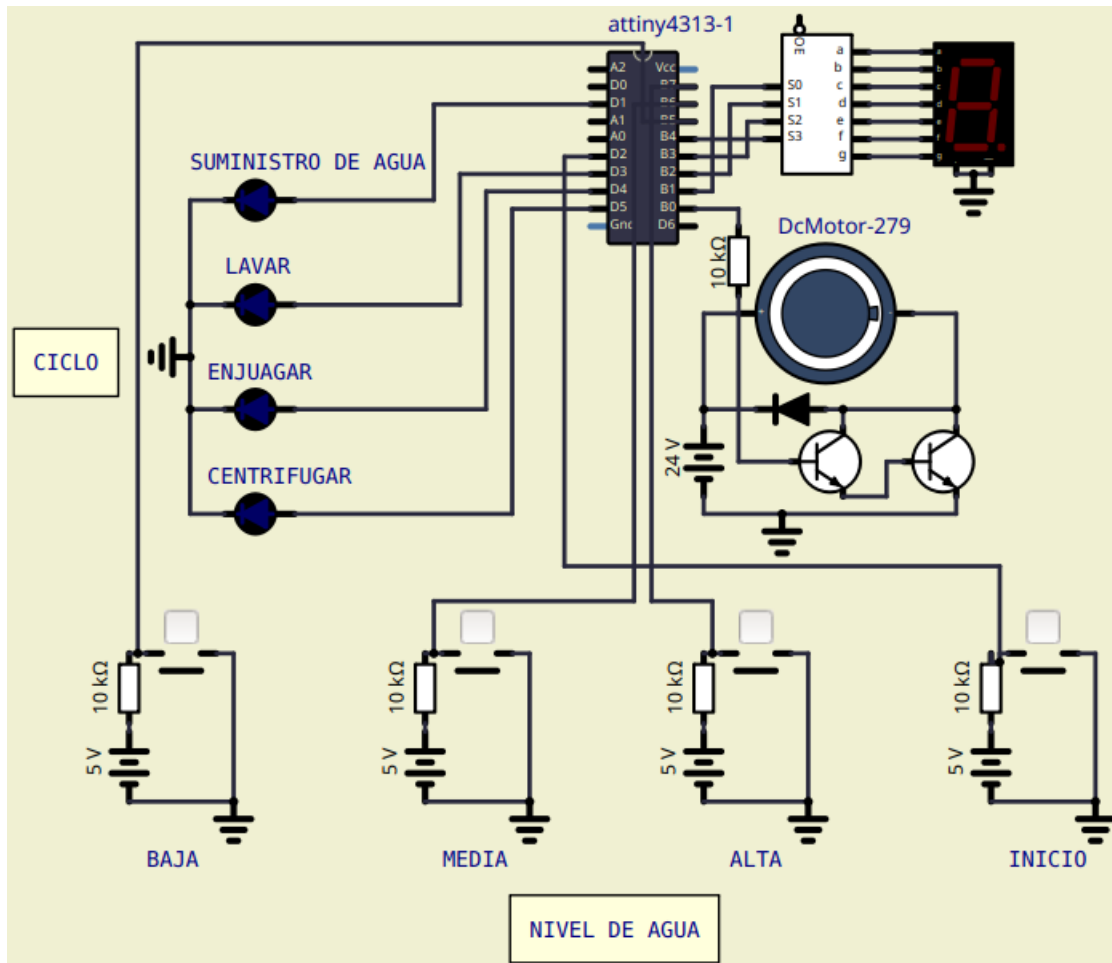


Figura 5: Circuito completo diseñado.

3. Desarrollo/Análisis

3.1. Diagrama de flujo

El ciclo de lavado se divide en las siguientes etapas principales: suministro de agua, lavado, enjuague y centrifugado, cada una está diseñada para replicar las operaciones que realiza una lavadora convencional, pero adaptadas al entorno controlado por el microcontrolador, en la etapa de suministro de agua, el sistema simula el llenado del tambor según el nivel seleccionado por el usuario, durante el lavado, el motor se activa para replicar la agitación del tambor, mientras que el LED correspondiente indica visualmente que esta etapa está en curso, el enjuague utiliza un proceso similar al lavado, pero con tiempos diferentes y el centrifugado actúa de la misma manera.

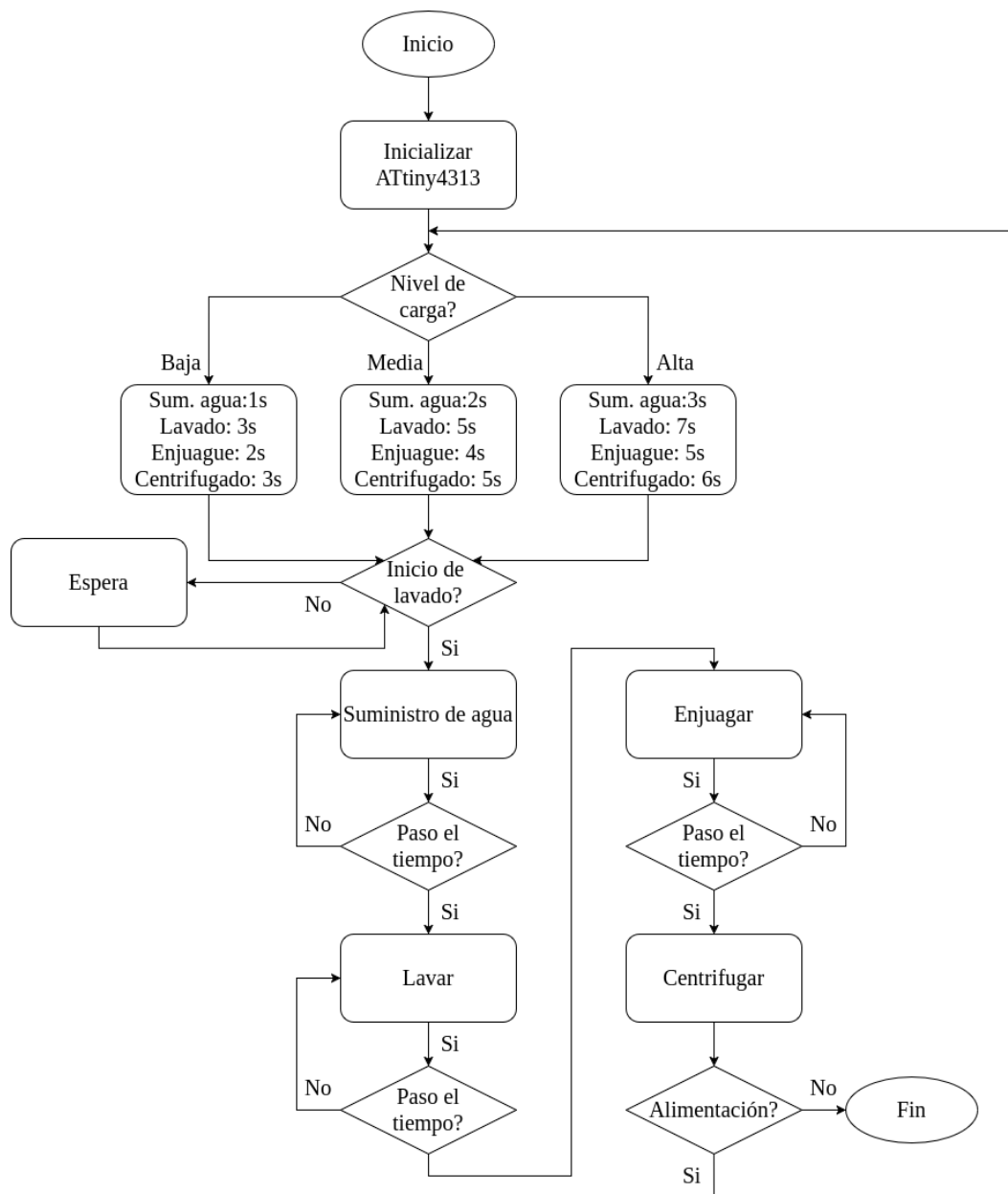


Figura 6: Diagrama de flujo del proceso.

3.2. Código

La programación del ATtiny4313 se realizó para manejar de manera eficiente las señales de entrada y salida, asegurando que cada componente funcionara según lo esperado, las interrupciones fueron configuradas para responder a los botones, lo que permite alternar entre los estados en tiempo real, además, se implementaron temporizadores para controlar la duración de cada etapa del ciclo de lavado, estos están integrados en el microcontrolador y permiten medir intervalos de tiempo, lo que es esencial para que las etapas se ejecuten correctamente, la actualización del display de 7 segmentos también se realizó mediante funciones específicas que traducen los valores de tiempo restantes en señales BCD adecuadas para el convertidor, a continuación se presenta el código diseñado.

```
1 #include <avr/io.h>
2 #include <avr/interrupt.h>
3 #include <util/delay.h>
4
5 enum Estado { espera, conf, cont, pausa, fin };
6
7 volatile enum Estado estadoActual = espera;
8 volatile uint8_t temporizadorActual = 0; // ndice del temporizador
   actual
9 volatile uint8_t contador[4] = {0, 0, 0, 0};
10 volatile uint8_t tiempos[4] = {0, 0, 0, 0};
11 const uint8_t pinesLED[4] = {PD1, PD3, PD4, PD5};
12 const uint8_t pinMotor = PB0;
13
14 void configuracion_inicial() {
15     // Configurar botones como entrada con pull-up
16     DDRD &= ~(1 << PD2); PORTD |= (1 << PD2);
17     DDRB &= ~((1 << PB5) | (1 << PB6) | (1 << PB7));
18     PORTB |= ((1 << PB5) | (1 << PB6) | (1 << PB7));
19
20     // Configurar pin del motor y LEDs como salida
21     DDRB |= (1 << pinMotor); PORTB &= ~(1 << pinMotor);
22     for (uint8_t i = 0; i < 4; i++) {
23         DDRD |= (1 << pinesLED[i]);
24         PORTD &= ~(1 << pinesLED[i]);
25     }
26
27     // Configurar pines BCD como salida
28     DDRB |= 0x1E;
29
30     // Configurar interrupci n externa INT0
31     MCUCR |= (1 << ISC01);
32     GIMSK |= (1 << INT0);
33     sei();
34 }
35
```



```
36 ISR(INT0_vect) {
37     _delay_ms(20);
38     estadoActual = (estadoActual == cont) ? pausa : cont;
39 }
40
41 // Leer botones para configurar tiempos
42 void configurarTiempos() {
43     if (!(PINB & (1 << PB5))) { _delay_ms(20); tiempos[0] = 1;
44         tiempos[1] = 3; tiempos[2] = 2; tiempos[3] = 3; }
45     else if (!(PINB & (1 << PB6))) { _delay_ms(20); tiempos[0] = 2;
46         tiempos[1] = 5; tiempos[2] = 4; tiempos[3] = 5; }
47     else if (!(PINB & (1 << PB7))) { _delay_ms(20); tiempos[0] = 3;
48         tiempos[1] = 7; tiempos[2] = 5; tiempos[3] = 6; }
49 }
50
51 // Actualizar salida BCD
52 void actualizarBCD(uint8_t valor) {
53     PORTB = (PORTB & 0xE1) | (valor << 1);
54 }
55
56 // Controlar LEDs y motor
57 void controlarHardware(uint8_t led, uint8_t motor) {
58     for (uint8_t i = 0; i < 4; i++) PORTD &= ~(1 << pinesLED[i]); //
59         Apagar todos los LEDs
60     if (led < 4) PORTD |= (1 << pinesLED[led]); // Encender LED
61         correspondiente
62     if (motor) PORTB |= (1 << pinMotor); else PORTB &= ~(1 <<
63         pinMotor); // Controlar motor
64 }
65
66 // Manejar el temporizador actual
67 void manejarTemporizador() {
68     if (contador[temporizadorActual] < tiempos[temporizadorActual]) {
69         _delay_ms(1000);
70         contador[temporizadorActual]++;
71         actualizarBCD(contador[temporizadorActual]);
72     } else {
73         contador[temporizadorActual] = 0; // Reiniciar contador
74         actualizarBCD(0);
75         temporizadorActual++; // Pasar al siguiente temporizador
76         if (temporizadorActual >= 4) estadoActual = fin;
77     }
78 }
79
80 int main(void) {
```

```
75     configuracion_inicial();
76
77     while (1) {
78         if (estadoActual == espera) {
79             configurarTiempos();
80             if (tiempos[0] > 0) estadoActual = conf;
81         }
82
83         switch (estadoActual) {
84             case conf:
85                 actualizarBCD(0);
86                 break;
87
88             case cont:
89                 controlarHardware(temporizadorActual, (
90                     temporizadorActual > 0)); // Motor activo en
91                     temporizadores 2, 3 y 4
92                 manejarTemporizador();
93                 break;
94
95             case pausa:
96                 controlarHardware(4, 0); // Apagar LEDs y motor
97                 break;
98
99             case fin:
100                 controlarHardware(4, 0);
101                 _delay_ms(1000);
102                 for (uint8_t i = 0; i < 4; i++) contador[i] = tiempos
103                     [i] = 0; // Reiniciar variables
104                 temporizadorActual = 0;
105                 estadoActual = espera;
106                 break;
107             default:
108                 break;
109         }
110     }
111     return 0;
112 }
```

3.3. Pruebas de funcionamiento

Las pruebas realizadas al sistema incluyeron tanto la verificación de las conexiones físicas del circuito como del código implementado, se revisaron todas las conexiones para garantizar que no hubiera errores en el cableado y posteriormente se cargó el programa en el microcontrolador, los LEDs respondieron correctamente a las señales del microcontrolador, encendiéndose y apagándose según

la etapa activa, suministro de agua (figura 7), lavado (figura 8), enjuague (figura 9) y centrifugado (figura 10), el motor también funcionó de manera adecuada, simulando el movimiento del tambor durante las etapas adecuadas.

Uno de los aspectos destacados fue la correcta visualización del tiempo transcurrido en el display de 7 segmentos, durante las pruebas se observó que el display actualizaba los números en sincronía con las etapas del ciclo, dando una representación clara del progreso del ciclo actual.

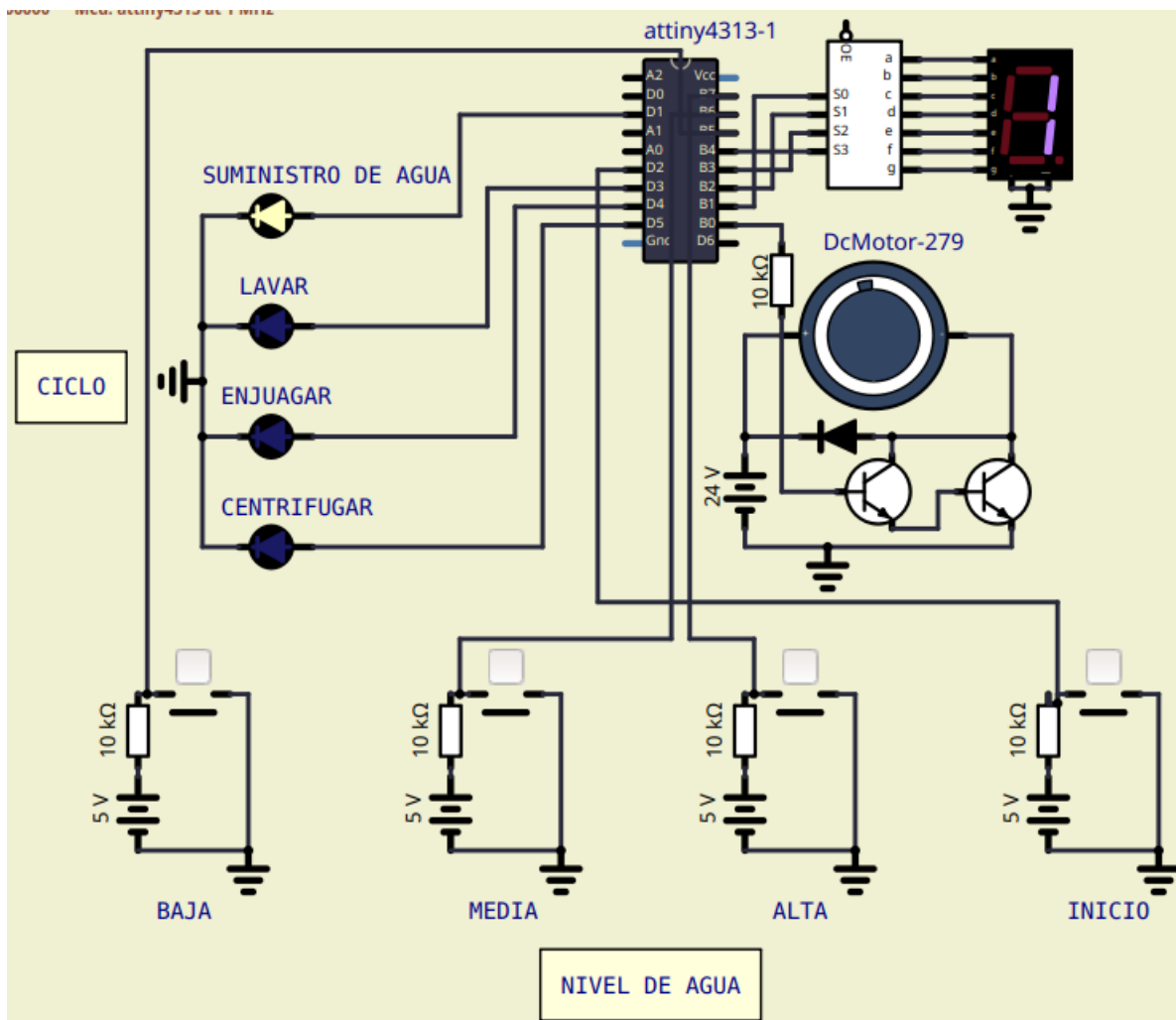


Figura 7: Ejecución en ciclo de suministro de agua.

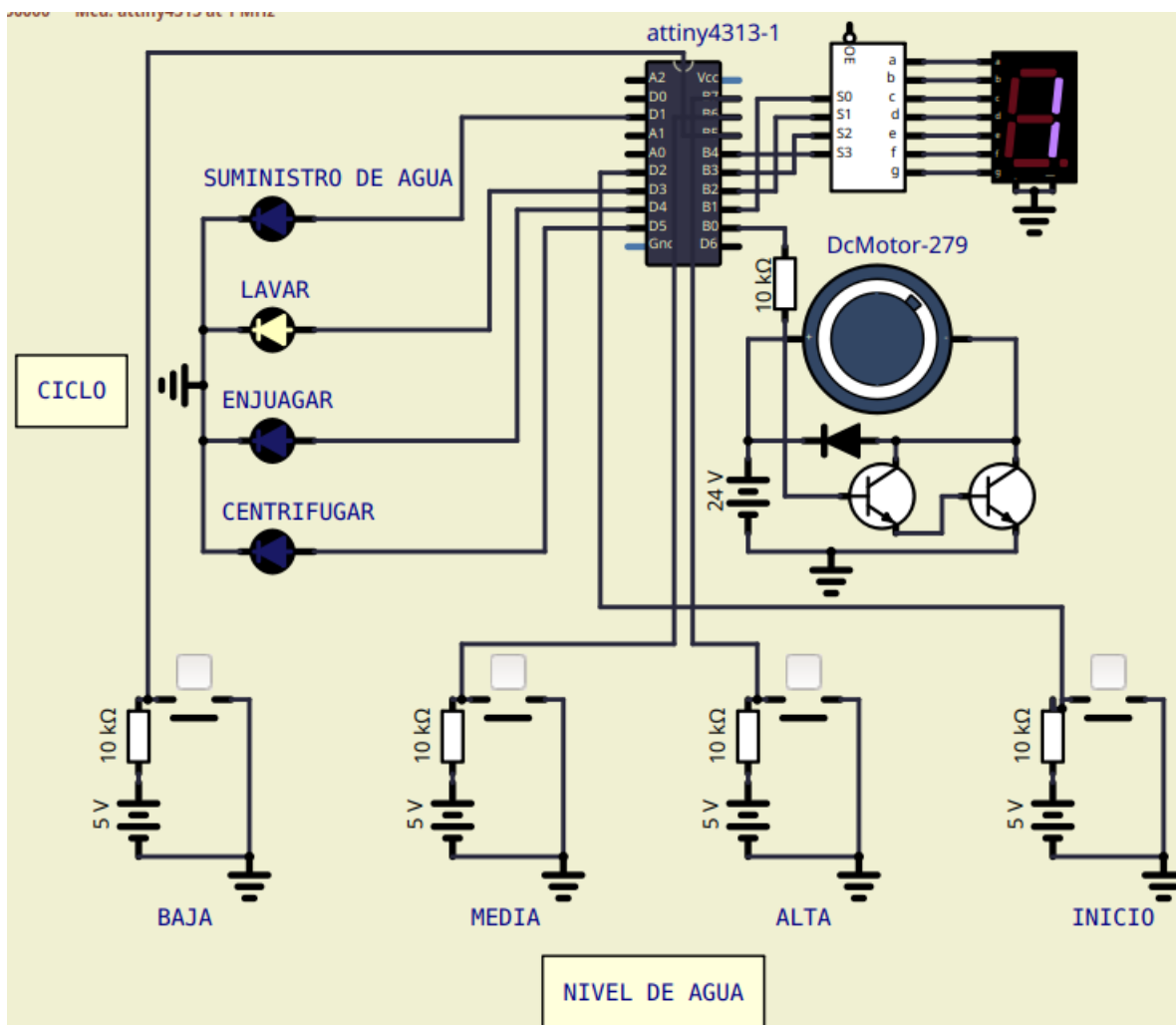


Figura 8: Ejecución en ciclo de suministro de agua.

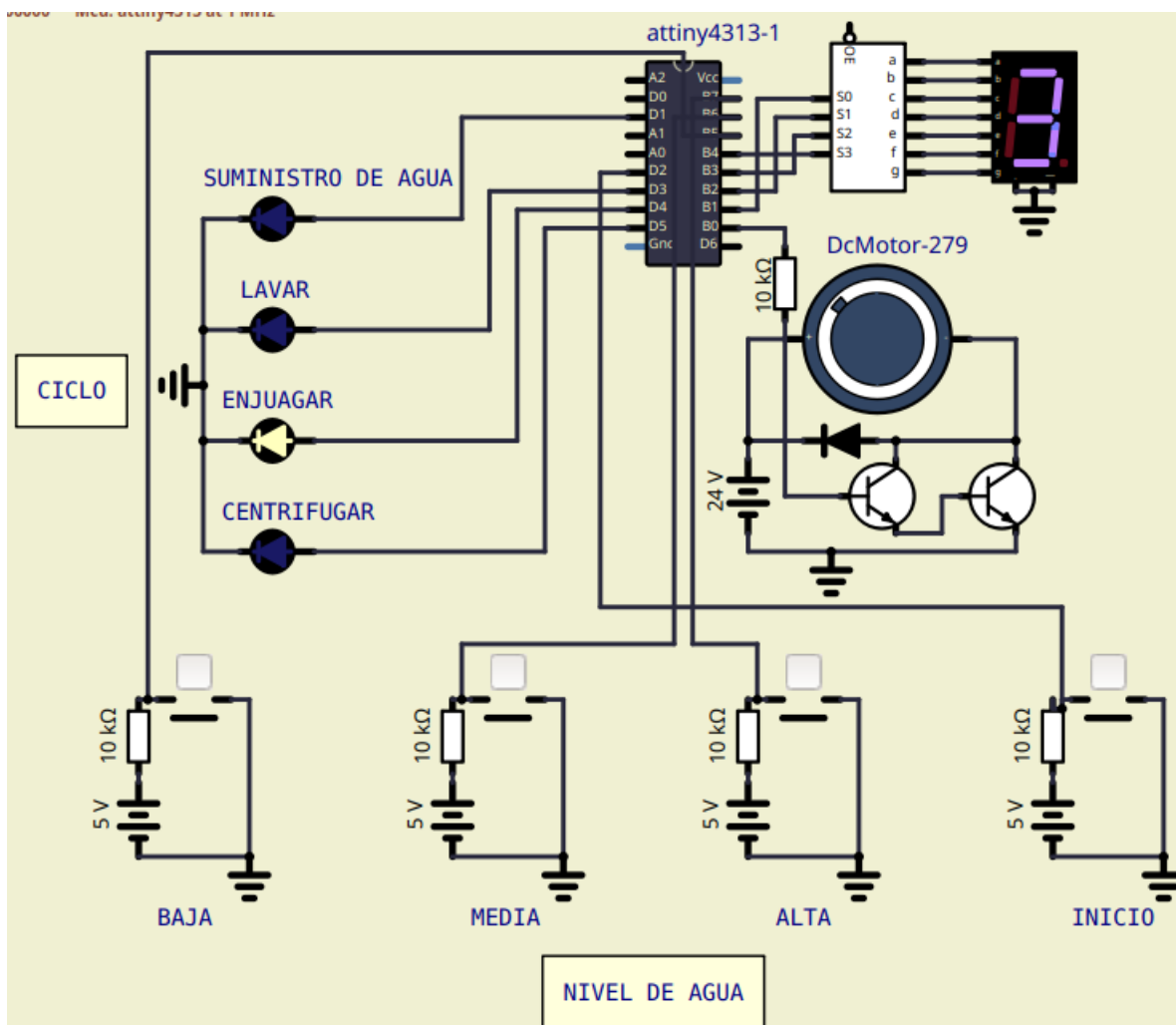


Figura 9: Ejecución en ciclo de suministro de agua.

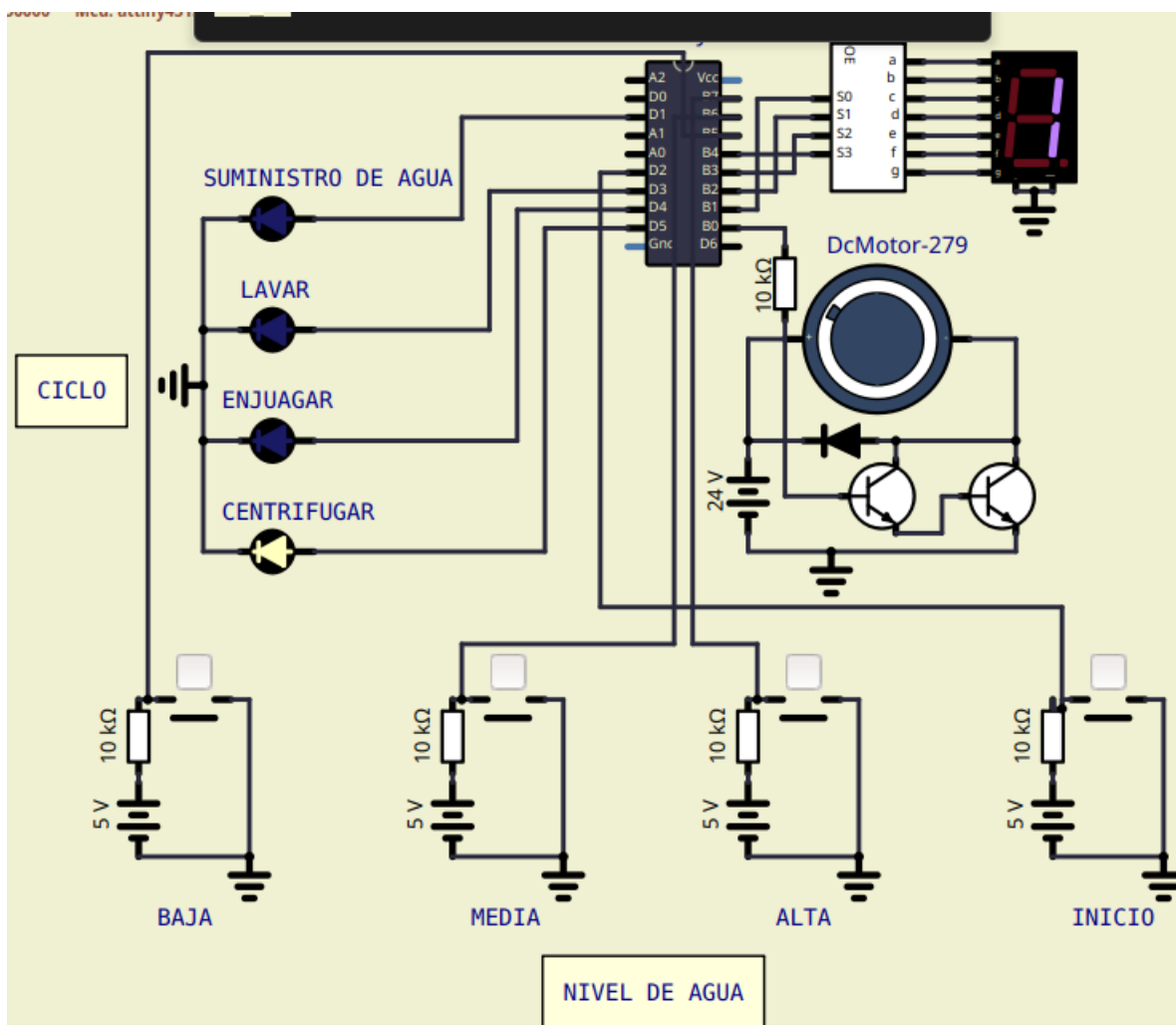


Figura 10: Ejecución en ciclo de suministro de agua.

4. Conclusiones y Recomendaciones

Se logró implementar con éxito el sistema funcional de control automatizado para una lavadora utilizando el microcontrolador ATtiny4313, el cual resultó muy versátil en cuanto a su capacidad para manejar entradas, salidas, temporizadores e interrupciones, la integración de componentes y la programación adecuada según las indicaciones permitieron cumplir los objetivos planteados, las pruebas repetitivas fueron esenciales para corregir errores y garantizar un funcionamiento del sistema.

Referencias

- [1] Atmel Corporation. (2011). *8-bit AVR Microcontroller with 2/4K Bytes In-System Programmable Flash*. Recuperado de: <https://www.alldatasheet.com/datasheet-pdf/pdf/392236/ATMEL/ATTINY4313.html>.
- [2] Texas Instruments. (2000). *DM7447A BCD to 7-Segment Decoder/Driver with Open-Collector Outputs*. Recuperado de: <https://www.ti.com/lit/ds/symlink/dm7447a.pdf>.