

UNIVERSIDAD DE COSTA RICA

ESCUELA DE INGENIERÍA ELÉCTRICA

MICROPROCESADORES

IE0623

Tarea 6

Autores:

Robin GONZÁLEZ RICZ B43011

Michelle GUTIERREZ B43195

Profesor:

Esteban BADILLA

Asistente:

Mariela HERNANDEZ

20 de febrero de 2021



Índice

| | |
|--|-----------|
| 1. Estructuras de datos | 1 |
| 2. Configuración de hardware | 1 |
| 2.1. Memorias de cálculo | 2 |
| 2.1.1. Real Time Interrupt | 2 |
| 2.1.2. Serial Communications Interface 1 | 2 |
| 2.1.3. ATD0 ISR | 3 |
| 3. Programa principal | 3 |
| 4. Subrutinas | 6 |
| 5. Calculo | 6 |
| 6. CONV_ASCII | 8 |
| 7. Subrutinas de interrupción | 9 |
| 7.1. RTI_ISR | 9 |
| 7.2. ATD0_ISR | 9 |
| 7.3. SC1_ISR | 10 |
| 8. Pruebas de funcionalidad | 11 |

Índice de figuras

| | | |
|-----|--|----|
| 1. | Diagrama de flujo del programa principal, parte 1. | 4 |
| 2. | Diagrama de flujo del programa principal, parte 2. | 5 |
| 3. | Diagrama de flujo de la subrutina Calculo. | 7 |
| 4. | Diagrama de flujo de la subrutina CONV_ASCII. | 8 |
| 5. | Diagrama de flujo de la subrutina de interrupción RTI. | 9 |
| 6. | Diagrama de flujo de la subrutina de interrupción ATD0. | 10 |
| 7. | Diagrama de flujo de la subrutina de interrupción SC1. | 11 |
| 8. | Captura del simulador en el estado de nivel bajo del tanque. | 12 |
| 9. | Captura del simulador en el estado: subiendo de nivel bajo tanque. . . | 13 |
| 10. | Captura del simulador en el estado de nivel alto del tanque. | 14 |
| 11. | Captura del simulador en el estado: bajando del nivel alto del tanque. . | 15 |

1. Estructuras de datos

EOM: byte para indicar fin de mensaje
CR: carry return
LF: line feed
SUB: substitue, borrado pantalla
P15: valor minimo de volumen al 15 %
P30: volumen al 30 %
P90: volumen al 90 %
P100: volumen maximo
estado: byte que define estado actual,;empezamos con vaciado,2:alarma,8:llenado

Nivel_PROM: promedio de lecturas, word
NIVEL: nivel del tanque
VOLUMEN: volumen del tanque
CONTADOR_DELAY: byte, contador
Puntero: word para cardar direcciones en X
encabezado: mensaje de texto
Volumen_ascii: mensaje de texto
Vaciado: mensaje de texto
Alarma: mensaje de texto
COMPLETADO: byte,contador

2. Configuración de hardware

```
;Configurar ATD 0
MOVB #$C2,ATD0CTL2 ;enciende ADC 0, al leer borra banderas, enciende INTs

MOVB #$30,ATD0CTL3 ;FIFO OFF, se haran 6 conversiones
MOVB #$01,ATD0CTL4 ;res a 10 bits, 2 ciclos de reloj, Prescalador en 17
MOVB #$80,ATD0CTL5 ;just a derecha, sin signo, controlada por software, sin
MUX

;Configurar interrupcion RTI
BSET CRGINT,$80 ;pone en 1 bit mas significativo
MOVB #$49, RTICTL ;carga valores para contar 10.24 ms segun formula
```

```

;leds
BSET DDRB,$01 ;habilitar leds
BSET DDRJ,$02
BCLR PTJ,$02

;config SCI 1
MOVW #39,SC1BDH ;poner SBR para baudrate
MOVB #0,SC1CR1
MOVB #$88,SC1CR2;habilitar interrupcion

```

2.1. Memorias de cálculo

2.1.1. Real Time Interrupt

Para obtener un periodo de 10 ms en la subrutina RTI:

$M = 4$

$N = 9$

En la dragon el Osc_CLK es 8 MHZ

$$T_{RTI} = \frac{(N + 1) \cdot 2^{(M+9)}}{Osc_CLK} = \frac{10x2^{13}}{8 \cdot 10^6} = 10,24 \cdot 10^{-3}s \quad (1)$$

Por lo cual hay un error debido a la baja granularidad en configuracion de los tiempos de interrupción de:

$$\left| \frac{V_{REAL} - V_{APROXIMADO}}{V_{REAL}} \right| * 100 = \left| \frac{0,01 - 0,01024}{0,01} \right| * 100 = 2,4\% \quad (2)$$

2.1.2. Serial Communications Interface 1

Se debe configurar un baudrate de 38400, para esto hay que obtener el valor de SBR. Se tiene la ecuación para calcular SBR:

$$SBR = \frac{Bus_CLK}{16xDataRate} \quad (3)$$

$$SBR = \frac{24 \cdot 10^6}{16x38400} = 39,06 \quad (4)$$

Entonces $SBR = 39$. Como se puede ver hay un error:

$$\left| \frac{39 - 39,06}{39} \right| * 100 = 0,15 \% \quad (5)$$

2.1.3. ATD0 ISR

Para que el ATD funcione a una frecuencia de muestreo de 700 KHz se debe calcular el prescalador necesario dada la frecuencia del Bus_CLK de la Dragon12.

Usamos la siguiente ecuación para obtener el prescalador:

$$PRS = \frac{Bus_CLK}{2f_s} \quad (6)$$

$$\frac{24 \cdot 10^6}{2 \cdot 700000} = 16,1428 \quad (7)$$

Como se puede ver hay un error:

$$\left| \frac{16 - 16,1428}{16} \right| * 100 = 0,89 \% \quad (8)$$

3. Programa principal

Se diseñó este programa para representar una máquina de estados donde dependiendo del estado anterior y entradas actuales se encienden o apagan salidas. El programa primero determina en qué estado se encuentre y ejecuta las acciones correspondientes y determina su siguiente estado.

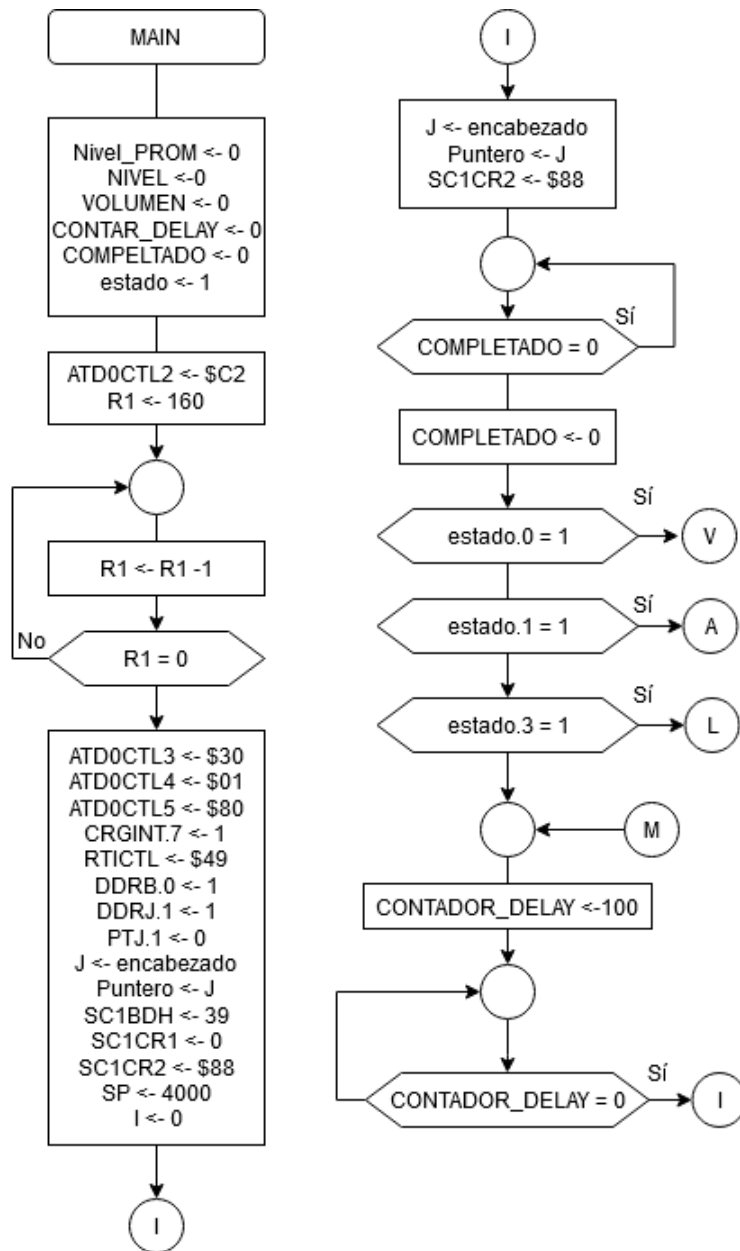


Figura 1: Diagrama de flujo del programa principal, parte 1.

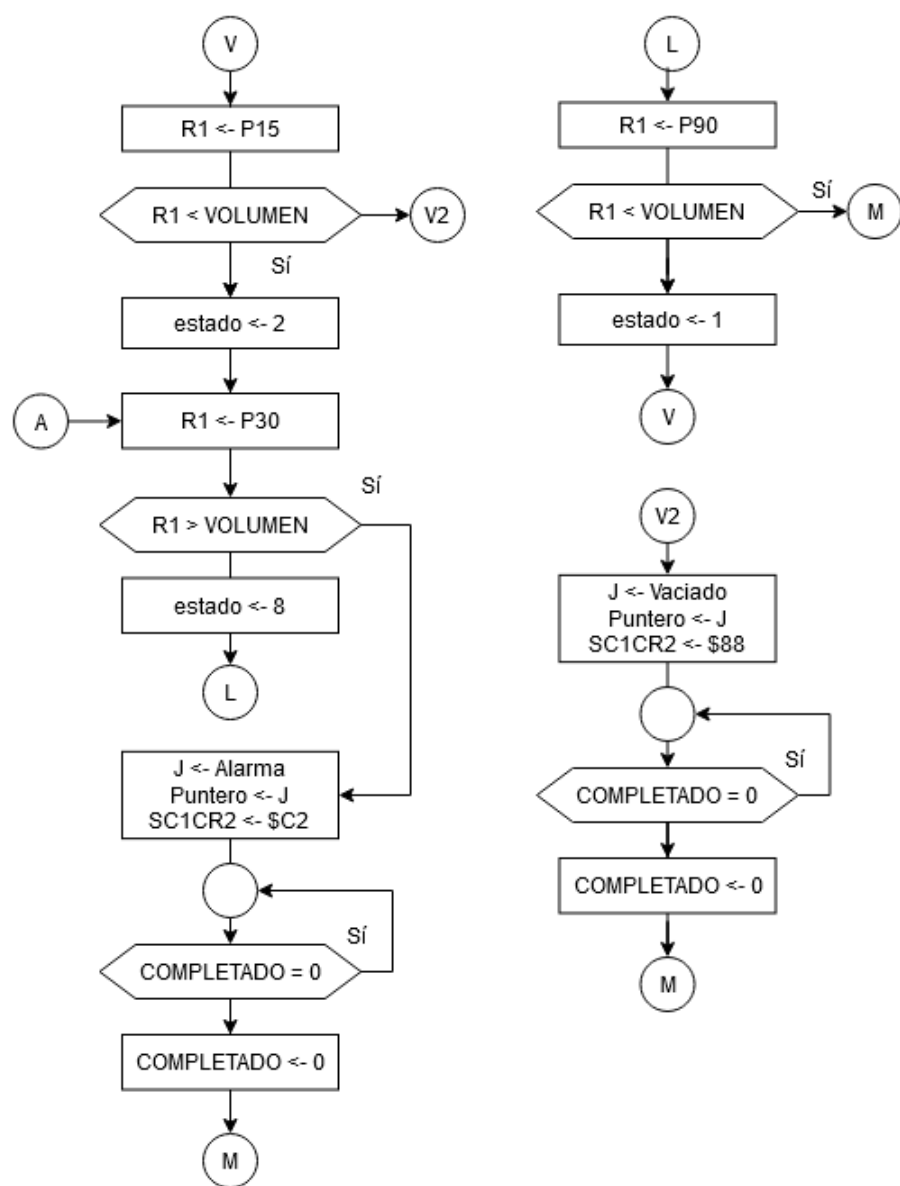


Figura 2: Diagrama de flujo del programa principal, parte 2.

4. Subrutinas

5. Calculo

Esta subrutina se encarga de calcular el volumen correspondiente, si este es mayor al valor máximo, asigna siempre 106, luego si el valor del volumen se encuentra por debajo del 15 % y una vez encendida, si el volumen sobrepasa el 90 % del valor máximo se apaga.

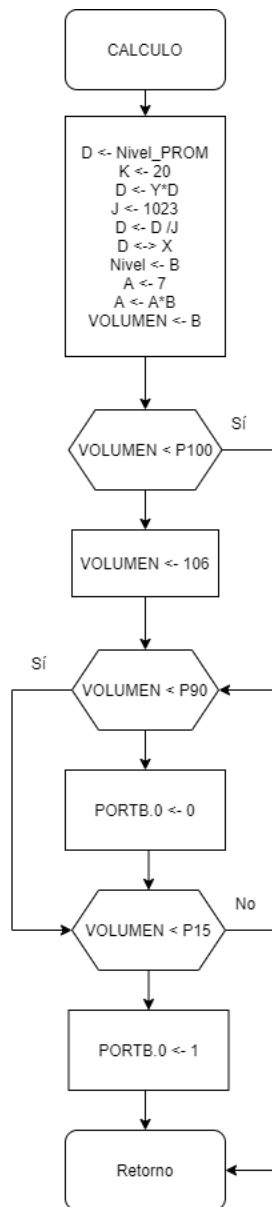


Figura 3: Diagrama de flujo de la subrutina Calculo.

6. CONV_ASCII

Esta subrutina convierte valores a ASCII haciendo el ajuste necesario según la posición de los números decimales en el código ASCII.

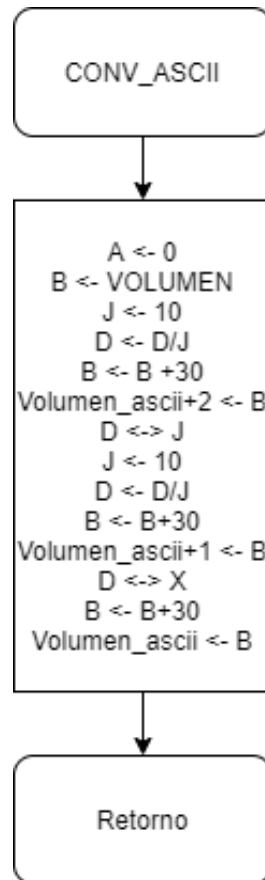


Figura 4: Diagrama de flujo de la subrutina CONV_ASCII.

7. Subrutinas de interrupción

7.1. RTI_ISR

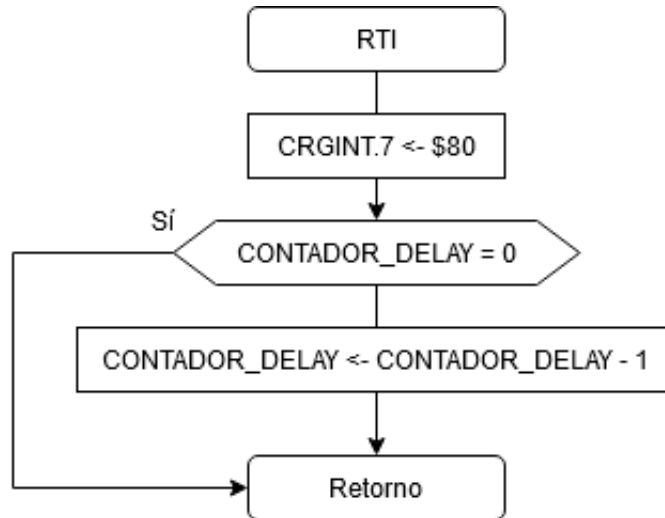


Figura 5: Diagrama de flujo de la subrutina de interrupción RTI.

7.2. ATD0_ISR

Se encarga de leer los resultados de la conversión digital/análogica en los registros correspondientes, saca el promedio de los datos, luego reinicia el ciclo de conversión y llama a las subrutinas Cálculo y Conv_Ascii.

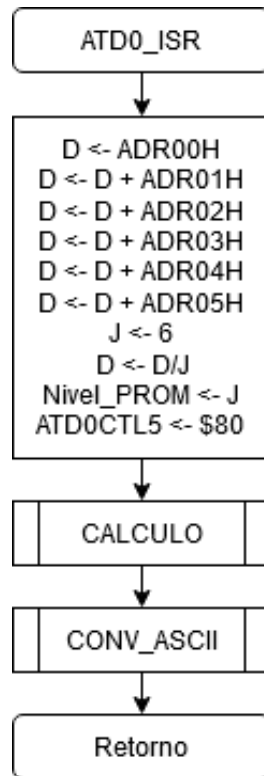


Figura 6: Diagrama de flujo de la subrutina de interrupción ATD0.

7.3. SC1_ISR

Se encarga de recorrer el arreglo de caracteres y enviar byte por byte, para imprimirlos en la terminal, hasta encontrar el indicador de fin de mensaje.

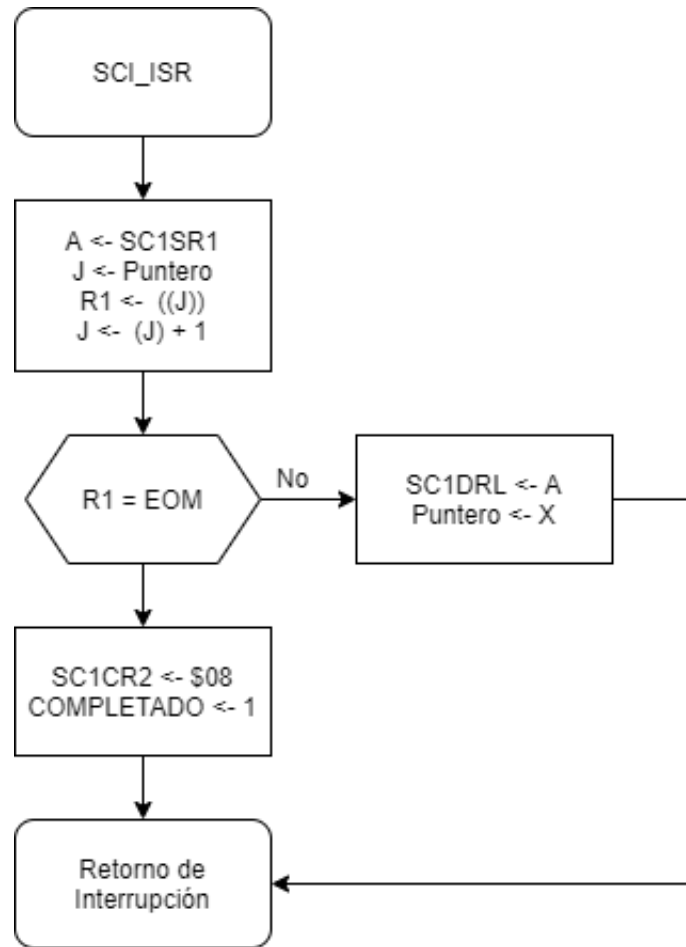


Figura 7: Diagrama de flujo de la subrutina de interrupción SC1.

8. Pruebas de funcionalidad

Se probó en el simulador que el led responde al volumen de agua en el tanque. También que los mensajes de alarma o apagado siguen la secuencia adecuada y funcionan de manera perfecta. Para esto se muestran imágenes con diferentes estados del sistema de control del tanque.

En la figura 10 se observa un volumen de $7m^3$, por lo cual se enciende la alarma de nivel bajo y se enciende la bomba.

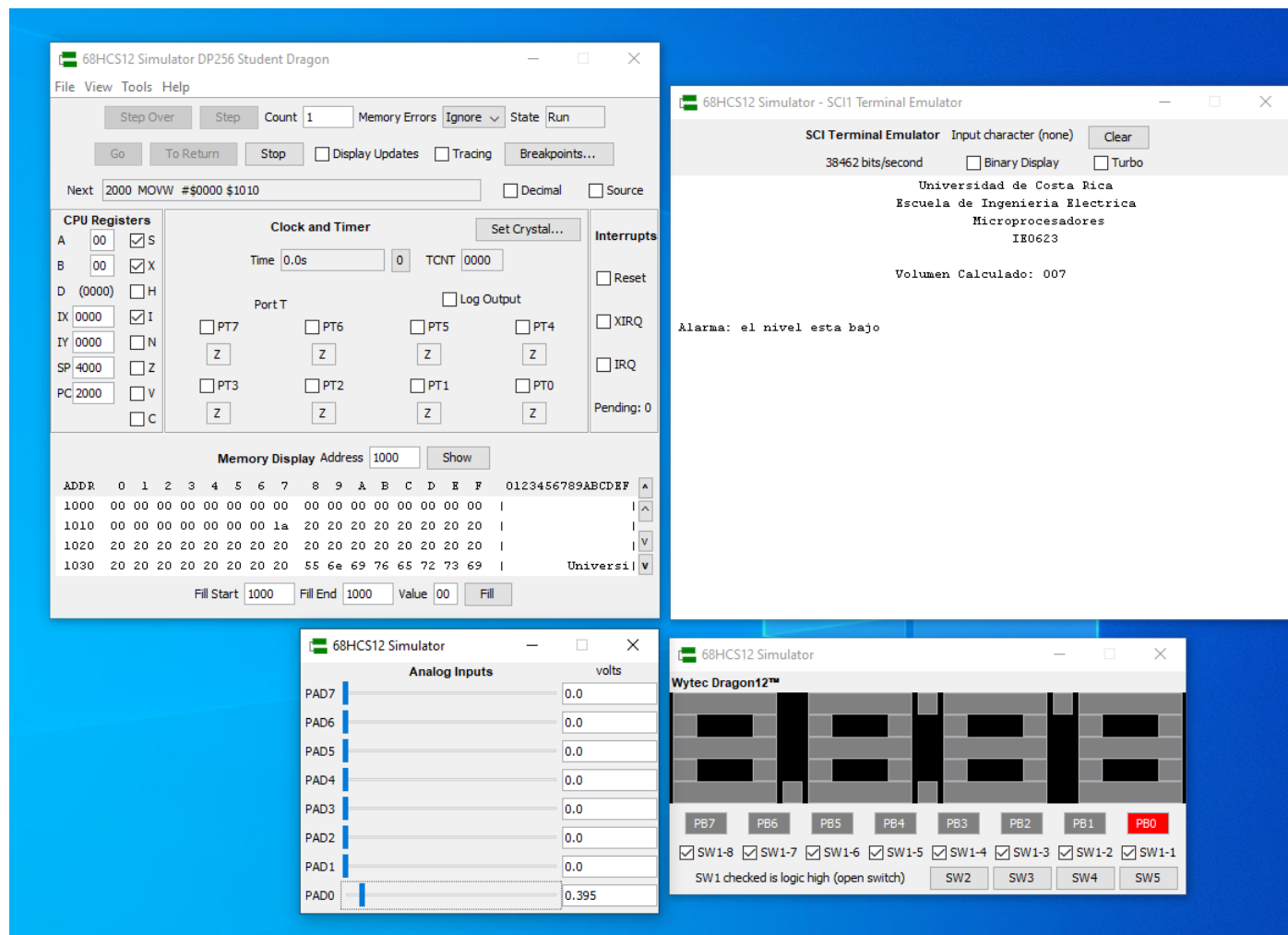


Figura 8: Captura del simulador en el estado de nivel bajo del tanque.

En la figura 9 se observa un volumen de $63m^3$, por lo cual se apaga la alarma de nivel bajo y el tanque se mantiene encendido pues aún no se llega al nivel alto.

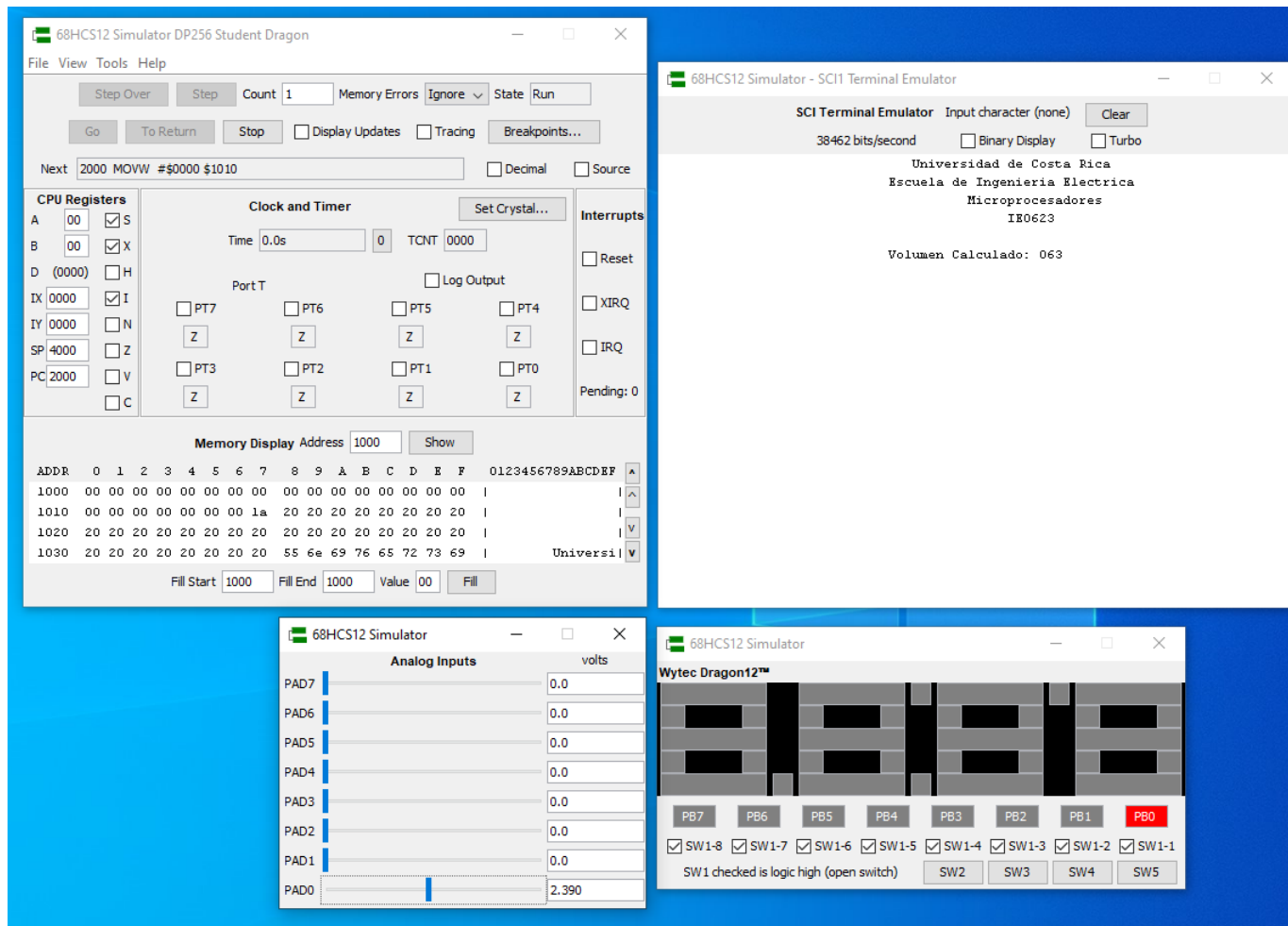


Figura 9: Captura del simulador en el estado: subiendo de nivel bajo tanque.

En la figura 10 se observa un volumen de $105m^3$, por lo cual se mantiene apagada la alarma de nivel bajo y ya se ha apagado la bomba pues ya sobrepasó el nivel de volumen del 90 %.

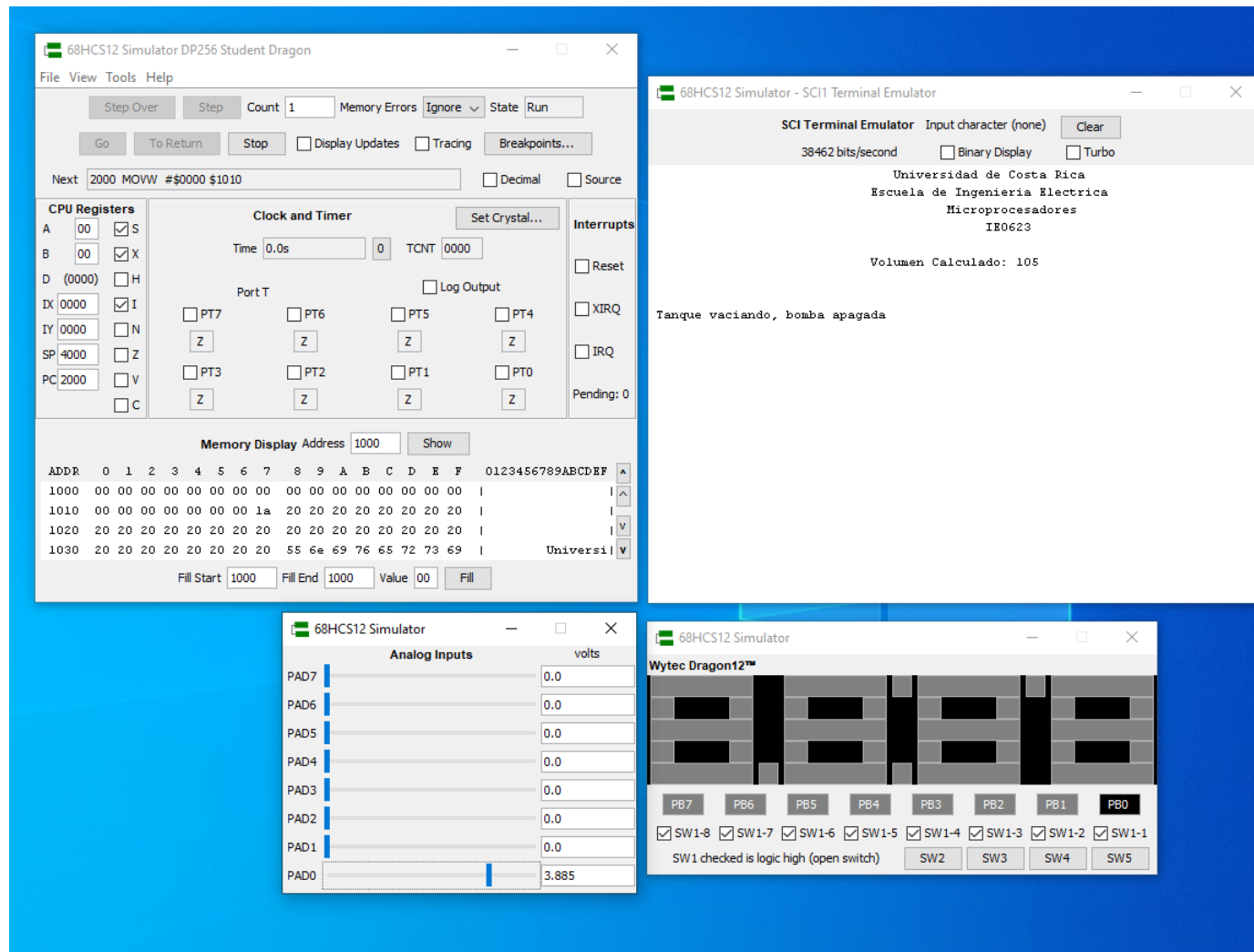


Figura 10: Captura del simulador en el estado de nivel alto del tanque.

En la figura 11 se observa un volumen de $91m^3$, por lo cual se mantiene pagada la alarma de nivel bajo y la bomba está apagada pues nos encontramos en el estado de vaciado.

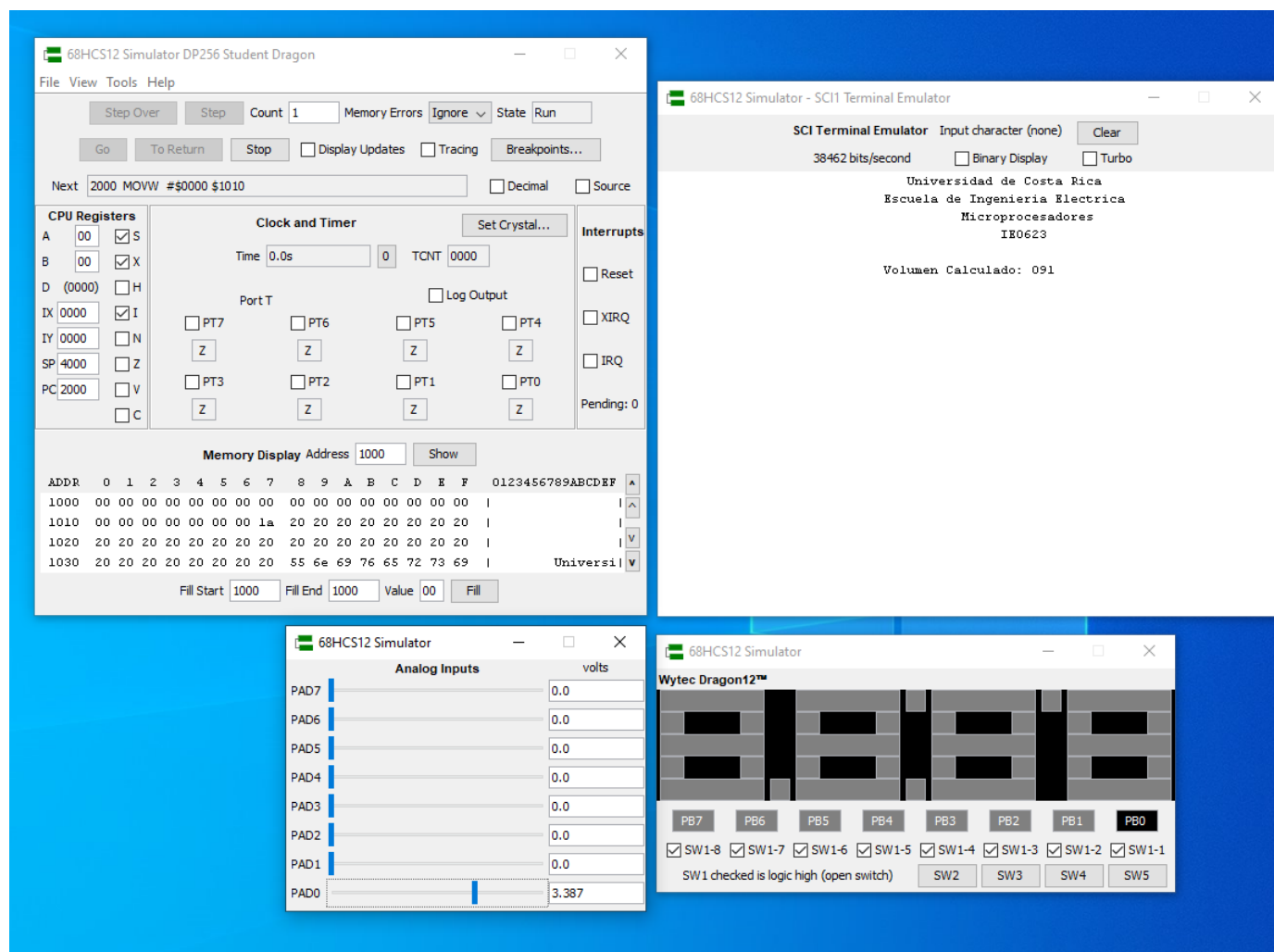


Figura 11: Captura del simulador en el estado: bajando del nivel alto del tanque.