

Diseño e implementación de un sistema de control de bombas peristálticas para dosificación de líquidos.

Mario Alejandro García Cabeiro¹ nene@rimed.cu

Guillermo Álvarez Bestard² guille@icmf.inf.cu

1 Instituto Superior Politécnico “José Antonio Echeverría”, MES, Cuba

2 Instituto de Cibernética, Matemática y Física, CITMA, Cuba

Resumen

El mismo muestra el diseño, la programación, la construcción y la instalación de una maqueta para la dosificación y la mezcla de líquidos. Para ello se emplea un panel de bombas peristálticas y una interfaz, compuesta por un microcontrolador, que gobierna las bombas siguiendo recetas preestablecidas. Esta a su vez se comunica con la computadora donde se diseñan las recetas o fórmulas y monitorea su ejecución a través de un software implementado para permitir la interacción del usuario con la maqueta.

Introducción

En la actualidad existen diversas formas para realizar la dosificación, siendo en muchos casos el elemento principal de estos sistemas sus bombas dosificadoras, las cuales pueden encontrarse de diversos principios de operación, por ejemplo, pueden ser bombas dosificadoras de membrana, de pistón, peristálticas, en fin una gran variedad, que acopladas a un sistema adecuado se pueden obtener muy buenos resultados, incluso en algunos sistemas se puede prescindir de válvulas y sensores de flujo. En numerosas aplicaciones el control de la dosificación con estos tipos de bombas se realiza a través de microcontroladores. En este trabajo el control de la dosificación se hará a través de bombas peristálticas, controladas precisamente por un microcontrolador. Su implementación se realizó en el Departamento de Control Automático del Instituto de Cibernética, Matemática y Física, en la Habana Cuba.

Descripción general de la maqueta

Esta maqueta está compuesta por un panel de 5 bombas peristálticas (nombradas como B0, B1, B2, B3, B4 de izquierda a derecha respectivamente), las cuales trabajan de forma independiente, pudiendo formarse un producto final a partir de hasta 5 líquidos primarios(1). A este panel se le agregó una interfaz con un microcontrolador que es capaz de controlar las bombas para dosificar las diferentes cantidades de productos primarios que se necesitan para formar el producto final.

La cantidad de líquido que pasa por cada bomba puede conocerse debido a que en estas bombas existe una relación entre las vueltas y el volumen dispensado (ml/rev). Las vueltas se miden mediante un sensor óptico que se colocó en la parte posterior del motor de cada

bomba. Para detectar las revoluciones de los motores fue necesario realizar varias modificaciones mecánicas dentro del panel.

El número de pulsos generados por cada sensor se almacena en el microcontrolador de la interfaz, donde se convierten a mililitros y se compara con una cantidad máxima asignada a cada bomba en la fórmula o dosis en ejecución. Cuando se igualan ambos valores se detiene la bomba correspondiente, en este caso el control se estará realizando desde el microcontrolador. Al terminar todas las bombas de dispensar las cantidades asignadas, se activa una señal sonora que indica el fin de la operación y por tanto el completamiento de la receta o fórmula.



Fig. 1 Panel de bombas peristálticas.

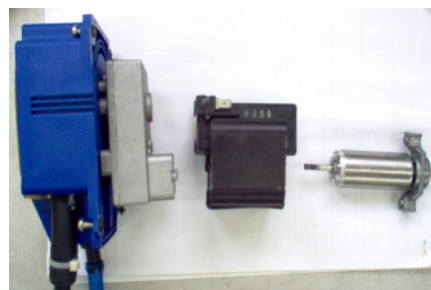


Fig. 2 Bomba (izquierda), estator (centro) y rotor (derecha) del motor.

Esta maqueta también contiene un agitador en el recipiente del producto final, que se puede activar o desactivar según convenga en cada fórmula a realizar, además se puede variar la velocidad de dicho agitador, lo cual se logrará variando el ciclo útil del PWM del microcontrolador utilizado.

La interfaz se comunica con una computadora mediante un puerto serie usando el protocolo RS-232. En la computadora debe estar instalado el software DOSIMAQ que será el encargado de interactuar con el usuario y enviarle a la interfaz las recetas o las operaciones a realizar.

Modificaciones realizadas al panel.

Al panel de bombas fue necesario agregarle un sensor (2) en cada motor para detectar las revoluciones y por lo tanto saber la cantidad de mililitros (ml) que circulan.



Fig. 3 Montaje del sensor en el motor.

También se agregó detrás del ventilador del motor una lámina plástica circular con ranuras, con el objetivo de atravesar el sensor y producir los pulsos que llegan al microcontrolador y permitirán conocer la cantidad de revoluciones del motor.

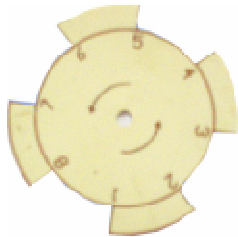


Fig. 4 Lámina agregada al ventilador del motor de B0.

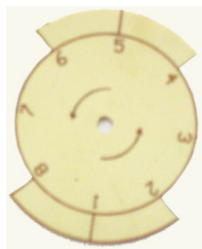


Fig. 5 Lámina agregada a los ventiladores de los motores de B1, B2, B3 y B4.



Fig. 6 Conjunto ventilador, lámina y sensor.

El número de ranuras en las láminas es diferente para poder detectar la misma cantidad de cambios por revolución en todas las bombas. Esto se debe a que los sensores de B1, B2, B3 y B4 están conectados a los pines RB4 a RB7 del puerto B, que generan una interrupción al detectar flancos ascendentes o descendentes, en cambio el sensor B0 está asociado al pin RB0 del mismo puerto, el cual genera una interrupción solo cuando detecta flancos ascendentes. De tal forma el sensor B0 debe generar el doble de pulsos que el resto para mantener la misma relación entre lo que detecta la interfaz y el volumen dispensado. Esto se ilustra en la siguiente figura, donde se muestra la respuesta de cada sensor cuando los motores se mueven a igual velocidad.

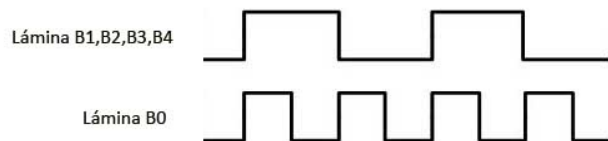


Fig. 7 Pulsos emitidos por los sensores de revoluciones.

Diseño, construcción y puesta a punto de la interfaz.

En este trabajo se utilizó el microcontrolador PIC 16F628A cuyas principales características se muestran en el anexo 2.

En este proyecto, se utilizó el lenguaje C para realizar la programación del PIC, para esto fue preciso utilizar un compilador de este lenguaje.

El compilador utilizado fue el PCW de la CCS Inc (3) y se integró al MPLAB, entorno de trabajo muy potente desarrollado por la Microchip (4) que incorpora todas las utilidades necesarias para la realización de un proyecto con microcontroladores. Este binomio permite editar el archivo fuente de nuestro proyecto y compilarlo, ensamblarlo y simularlo en pantalla, facilitando su ejecución en modo paso a paso, además permite ver cómo evolucionan sus registros internos, la memoria RAM y/o EEPROM, la memoria de programa (5).

Para la simulación del circuito eléctrico de la interfaz, se utilizó el simulador de circuitos electrónicos ISIS del sistema PROTEUS 7.4 SP3. En este simulador el acceso a los registros y su modificación es menos flexible que en el MPLAB. Por lo que también se integró a este ambiente, de modo que ejecutando el programa en el MPLAB se aprovechan todas las potencialidades de ambos sistemas. Por tanto se puede observar el comportamiento del circuito en el PROTEUS empleando la herramienta Proteus VSM MPALB Driver, y por tanto utilizar los instrumentos virtuales que este ofrece, sin perder las facilidades del MPLAB.

El firmware del microcontrolador consiste en un ciclo infinito de trabajo que se encarga de ejecutar acciones según los comandos recibidos desde la computadora. Este ciclo va a ser interrumpido cada vez que se genere alguna de las interrupciones habilitadas en el microcontrolador para este proyecto enumeradas a continuación.

1. Interrupción por recepción de un byte por el puerto serie.
2. Interrupción por cambio del estado de las entradas del puerto B (RB4 a RB7), utilizadas para detectar las revoluciones de los motores a través de los pulsos emitidos

por los sensores colocados en los motores de B1, B2, B3 y B4.

3. Interrupción por frente de subida en el PIN RB0, en donde se detectaran las revoluciones del motor de la B0 a través de los pulsos emitidos por el sensor agregado a dicha bomba.
4. Interrupción por desbordamiento del Timer0, utilizada para lograr la frecuencia sonora que se emitirá cuando se termine de realizar una fórmula.

No podemos dejar de señalar que una de las principales desventajas de usar el lenguaje C para la programación de los microcontroladores de esta familia es la poca eficiencia que se logra en la generación del código ensamblador. En este caso fue necesario recurrir a reprogramar algunas funciones y segmentos de código en ensamblador para no sobrepasar la capacidad de memoria de programa del microcontrolador.

El diseño del hardware de la interfaz se puede observar en el anexo 3, en el anexo 4 se observa la interfaz terminada con sus principales características y en el anexo 5 el sistema de medición montado en el panel.

EI DOSIMAQ.

El DOSIMAQ es un software programado en el lenguaje de alto nivel C# (6), el cual se ejecuta en una computadora permitiendo la interacción del usuario con el hardware de la maqueta de dosificación construida.

El mismo posibilita editar fórmulas, guardarlas y posteriormente cargarlas cuando se desee, el control para la realización de estas fórmulas podrá ser desde el microcontrolador (para fórmulas más simples donde todas las bombas se activarán desde el inicio y se irán desactivando cuando hayan dispensado la cantidad

definida por el usuario para cada bomba), o desde la PC para fórmulas más complejas en donde se podrán definir condiciones para que se ejecuten acciones (control secuencial). Además permite realizar un control manual de la fórmula que se esté realizando y también monitorear el estado de las fórmulas, conociendo la cantidad de ml que han pasado por cada bomba y el flujo que está pasando por cada bomba. La ventana para el monitoreo de fórmulas se ilustra en el Anexo 6.

Los resultados

A continuación se explicarán los principales resultados obtenidos mediante la realización de experimentos teóricos y prácticos.

Experimento 1

Este experimento consiste en realizar una fórmula controlada desde el microcontrolador con el fin de poder verificar en detalle el funcionamiento de la interfaz, ejecutándose el comando "D2,4,6,8,10" (ver detalles sobre este comando en Anexo). Para ello se utilizó el circuito mostrado en la anexo 7, a los pines del microcontrolador que corresponden a las bombas están conectados relés, los cuales al conmutar (cuando se acciona las bombas) permiten el paso de pulsos a través de él, simulando así el movimiento de los motores.

En el anexo 8 se ilustra un esquema que contiene las señales que activan las bombas y los pulsos que se reciben como simulación del movimiento de los motores. Como se observa, al pasar a nivel alto las señales de activar las bombas, aparece en los sensores los pulsos. Cuando se recibe la cantidad de pulsos equivalente al número de ml a bombear se desactivan las bombas.

Experimento 2

En este experimento también se realiza una fórmula controlada desde el microcontrolador, usando el mismo esquema mostrado en el anexo 8. Pero a diferencia del experimento 1, se usará el DOSIMAQ para manipular la interfaz, por lo que fue necesario crear dos puertos series virtuales para poder conectar el circuito con el software antes mencionado, para lo que se utilizó el "Virtual Serial Port Driver".

La fórmula a realizar corresponde con el comando "D1000,2000,3000,4000,5000", la Fig. 8 ilustra cuando se carga la fórmula en el DOSIMAQ, antes de comenzar su elaboración.

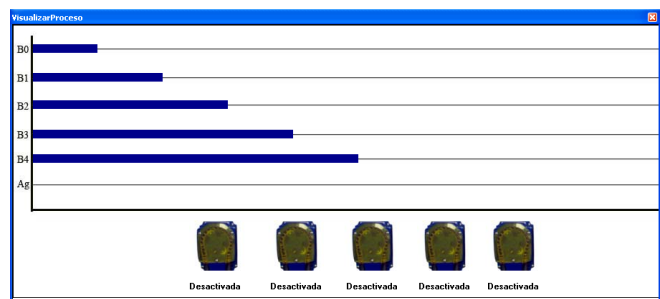


Fig. 8 Fórmula a realizar

Las barras azul oscuro representan los ml que se deben bombear, y a medida que esto vaya ocurriendo las barras azul claro crecerán encima de las oscuras, indicando los ml bombeados.

La Fig. 9, que se encuentra en el anexo 9, se ilustra el resultado final de los pulsos recibidos, los cuales coinciden con los ml bombeados para B1, B2, B3 y B4. En el caso de B0 se muestra el doble, pues está asociada a RB0, donde solamente se detectan los frentes ascendentes de los pulsos, y fue necesario duplicarles las orejas a la lámina agregada al ventilador del motor de esta bomba, para obtener la cantidad de cambios necesarios por revolución en RB0.

En Fig. 10 se muestra el arreglo pulsosB que almacena la cantidad de cambios detectados en los pines asociados a los sensores.

Durante el desarrollo de este experimento se pudo comprobar que la exactitud del control desde el microcontrolador es muy buena, pues no se comenten errores en la dosificación.

Experimento 3

Este experimento consiste en realizar la misma fórmula del experimento 2 pero el control será desde la computadora. En el anexo 10 se muestran los valores transferidos por las bombas.

Si se comparan estos resultados con los obtenidos mediante el control desde el microcontrolador, se observa que esta variante es menos exacta, debido en su mayor parte al tiempo que demora la comunicación entre la computadora y la interfaz. Se comete un error máximo de aproximadamente de 25 ml por bomba, lo que representa para cada bomba el 2.5%, 1.2%, 0.83%, 0.62% y el 0.5% respectivamente. O sea se comete un error máximo de un 2.5% con respecto al valor total para esta fórmula. Este error disminuye a medida que aumentan los ml bombeados. Para obtener errores menores de 1% se deben bombear mínimo 2500ml por cada bomba.

Experimentos de validación.

Para la comprobación de los resultados y total validación de la maqueta se realizan en la práctica los experimentos teóricos anteriores, obteniéndose resultados similares a los obtenidos en las simulaciones. Con lo que se resalta la importancia de la simulación y de la selección de las herramientas adecuadas para la misma.

Se comprobó además que las bombas entregan la cantidad de líquido que indica la interfaz, midiendo con probetas y recipientes graduados, el volumen de líquido dispensado por cada bomba, detectándose un aumento en el error del líquido bombeado por cada bomba producto al tiempo de la demora de conmutación del relé, esto ocurre para ambas formas de control (desde el microcontrolador y desde el software de alto nivel). El error aumentó entre 7 y 10 ml por bombas.

La interfaz mantuvo un comportamiento estable y no se detectaron errores en la comunicación serie.

Con la realización de estos experimentos se da por terminada la construcción de la maqueta para dosificaciones con bombas peristáltica, solo falta incorporarle el recipiente y el agitador, que como se mencionó antes son las únicas partes de la maqueta que al término de este trabajo no están listas.

Conclusiones

Los objetivos de este trabajo fueron cumplidos satisfactoriamente porque se logró construir una maqueta para la dosificación y mezcla de líquidos, la cual puede ser utilizada para ilustrar dicho proceso a estudiantes vinculados con este tema, ya que a pesar de que el experimento esté enfocado a escala de laboratorio, puede ser extendido a cualquier ámbito de los procesos industriales reales.

El empleo de herramientas de simulación como MPLAB y Proteus permitió validar la mayor parte del circuito de la interfaz. Su uso durante la puesta a punto del sistema permitió detectar errores en el diseño y corregirlos antes de fabricar la interfaz. La fortaleza de estas herramientas garantizó una simulación interactiva y un fácil acceso a los registros y periféricos del

microcontrolador. También mediante la simulación se pudo comprobar el funcionamiento del software de alto nivel, lo cual facilitó su puesta a punto y la del sistema en general.

El presente trabajo constituye un ejercicio integral, ya que no solo se desarrolló la parte teórica del proyecto, sino también la parte práctica, lo que resulta una experiencia muy provechosa pues obliga a enfrentar problemas reales que no existen durante el desarrollo teórico del trabajo.

Referencias Bibliográficas

1. Wetmatic. *Manual de instalación y funcionamiento.*
2. **Electronics, N E C.** *Photo interrupters PS4001.*
3. **CCS.** *Ayuda de CCS C Compiler.*
4. Microchip. [En línea] 2009. <http://www.microchip.com>.
5. **Remiro Domínguez, Fernando.** *El MPLAB.*
6. **González, J.** *El lenguaje de programación C#.*
7. *Bombas Peristálticas: Una Solución Simple para Medir los Químicos más Corrosivos.*
Truetel, Chuck. PetroQuiMex, págs. 50-54.
8. **Microsoft.** *Documentación de Microsoft Visual Studio 2005.*
9. **MAXIM.** MAX 220-249.
10. **López Pérez, Eduardo.** *Protocolo RS-232. Ingeniería en Microcontroladores.*
11. **Gardner, N.** *An introduction to programming The Microchip PIC in CCS C.*
12. **Campos, M F, Castañeda, R y Contreras, A C.** *Implementación de un sistema de desarrollo utilizando microcontroladores PIC MICROCHIP TECHNOLOGY.*
13. **Álvarez, Guillermo y otros.** *Simulación de una interfaz de supervisión y control para proyectores planetarios ZKP-2.*
14. Principio de funcionamiento de las bombas peristálticas. [En línea] http://www.quiminet.com/ar9/ar_K%01H%C6%11m%F4%C3.htm.
15. Principales Características de las bombas peristálticas. [En línea] http://www.quiminet.com/ar8/ar_%D2%0E%26%3C%E8%F3%17q.htm.
16. PICMANIA. [En línea] http://picmania.garcia-cuervo.com/Electronica_Basica.htm#TRANS1.
17. MAXIM. [En línea] <http://www.maxim-ic.com/support>.
18. **datos, MICROSHIP Hoja de. PIC16F627A/628A/648A.**
19. **Mendez, J. A.** Infomed.MONITORES DE HEMODIÁLISIS. [En línea] www.infomed.cu.
20. **Microchip.** PIC16F627A/628A/648A.

Anexo 1: Lista de Comandos.

	Comando	Respuesta	Descripción
1	Da,b,c,d,e	<	Comienza a realizar fórmula de tipo 1.
2	W#=a,b,c,d,e	<	Guarda fórmula de tipo 1 en memoria EEPROM.
3	R#	Fa _R ,b _R ,c _R ,d _R ,e _R	Lee fórmula de tipo 1 de la memoria EEPROM.
4	S#	<	Comienza a realizar fórmula guardada en memoria EEPROM.
5	E	Ta _E ,b _E ,c _E ,d _E ,e _E ,n,E _E oE _{E1} E _{E2} E _{E3} E _{E4}	Encuesta al microcontrolador para saber la cantidad de ml bombeados, el estado del nivel y el estado de las bombas.
6	IE ₀ E ₁ E ₂ E ₃ E ₄	<	Asigna estados a las bombas, utilizado en el control manual y para el control de fórmulas de tipo 2.
7	Av	<	Asignarle velocidad al agitador.
8	C	<	Inicializa los contadores que almacenan la cantidad de revoluciones, además inicializa la bandera trabajando.
9	Bz	<	Apaga o enciende el zumbador.

Variable	Valor	Descripción.
a,b,c,d,e	0 a 10000 ml	Cantidad de ml a bombear por cada bomba cuando se ejecute la fórmula.
<		Significa que se ejecutó satisfactoriamente el comando.
#	0 a 9	Número de fórmula en memoria EEPROM.
E _i	0 o 1	Indica el estado (0: desactivada, 1: activada) que tomará la i-ésima bomba.
v	0% a 100%	Velocidad del agitador representada en porcentaje.
z	0 o 1	Indica el estado que tomará el zumbador (0: desactivado, 1: activado).
a _R ,b _R ,c _R ,d _R ,e _R	0 a 10000 ml	Cantidad de ml a bombear por cada bomba cuando se ejecute la fórmula leída.
a _E ,b _E ,c _E ,d _E ,e _E	0 a 10000 ml	Cantidad de ml bombeados por cada bomba.
n	0 o 1	Indica el estado del nivel (0 nivel normal , 1 nivel alto)
E _{Ei}	0 o 1	Indica el estado actual (0: desactivada, 1: activada) de la i-ésima bomba.

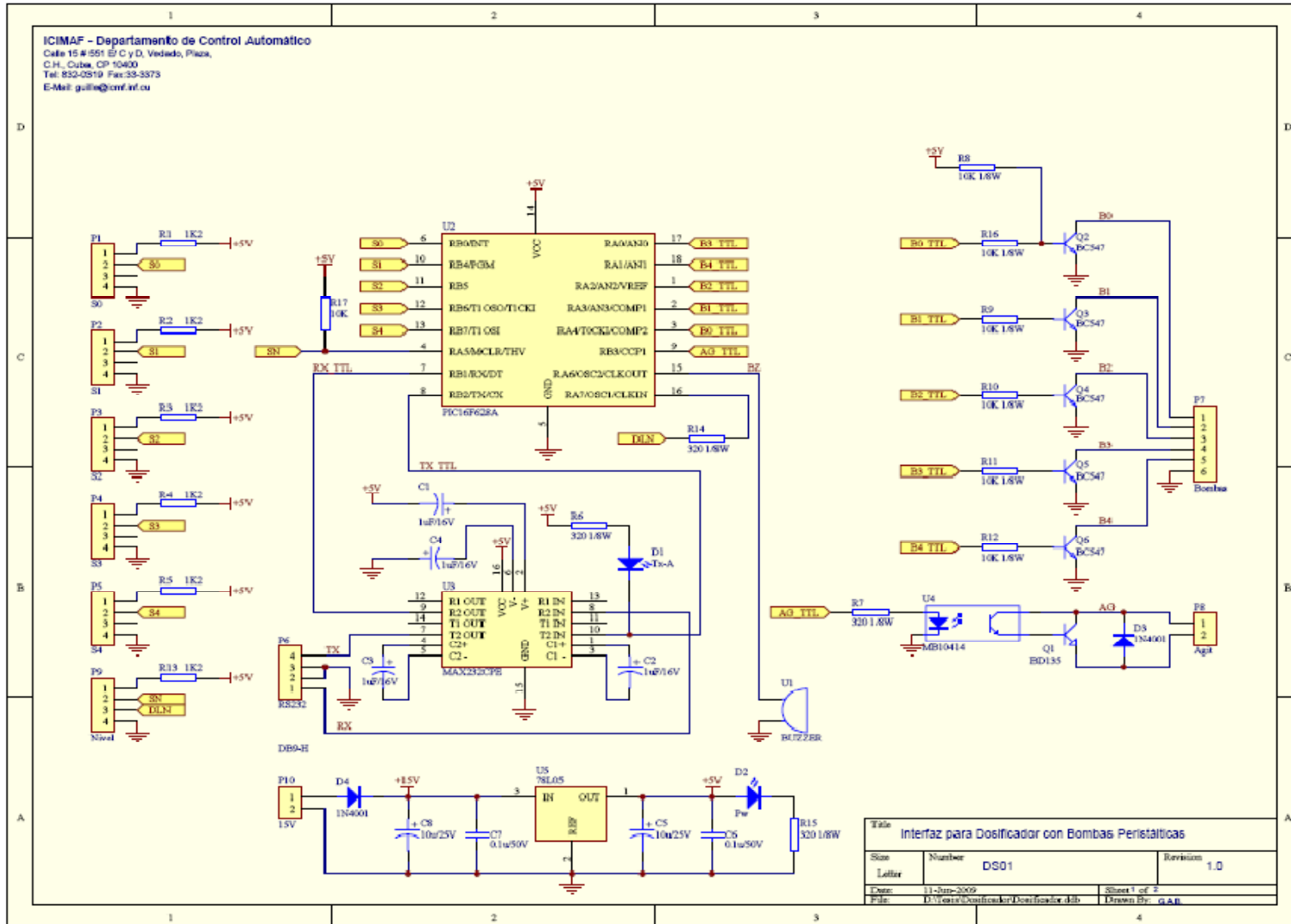
Nota1: en todos los casos "a" corresponde a B0, "b" a B1, "c" a B2, "d" a B3 y "e" a B4.

Nota2: Todos los comandos se encuentran antecedido por inicio de trama "<" y precedidos por fin de trama retorno de carro "\r" (el ASCII 13 en decimal ó D en hexadecimal).

Anexo 2: Principales características del PIC 16F628A.

Memoria de programa	Memoria de datos (SRAM)	Memoria de datos (EEPROM)	I/O	CCP (PWM)	USART	Comparadores	Timers (8/16 bits)
2048 (palabras)	224 (bytes)	128 (bytes)	16	1	1	2	2/1

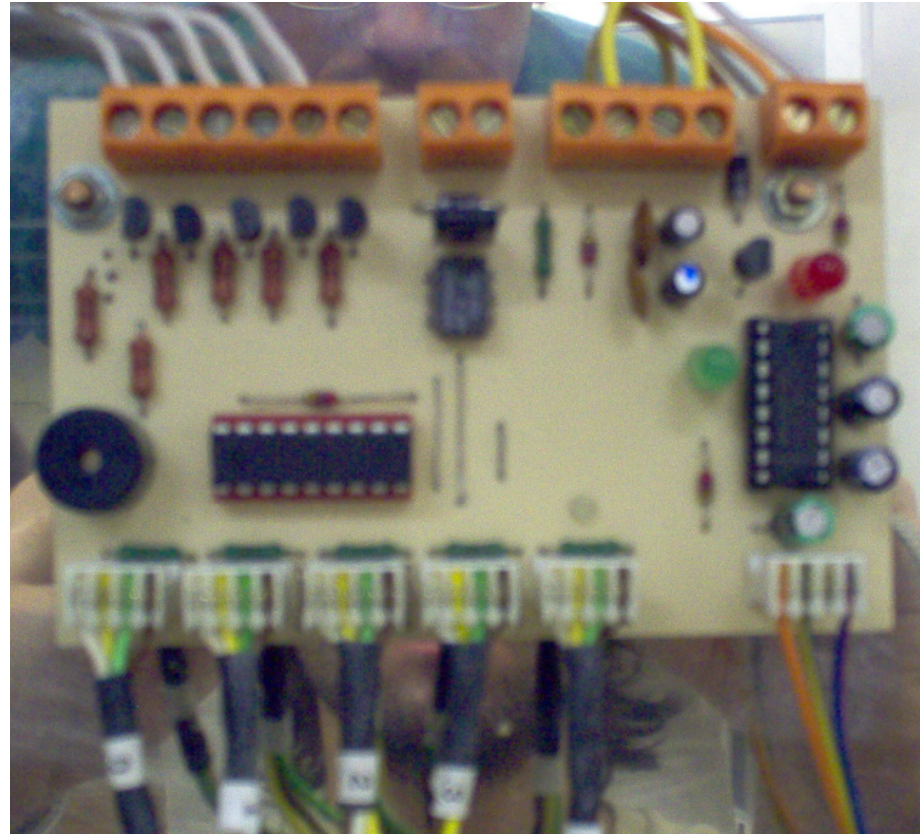
Anexo 3: Esquema que ilustra el diseño del hardware de la interfaz.



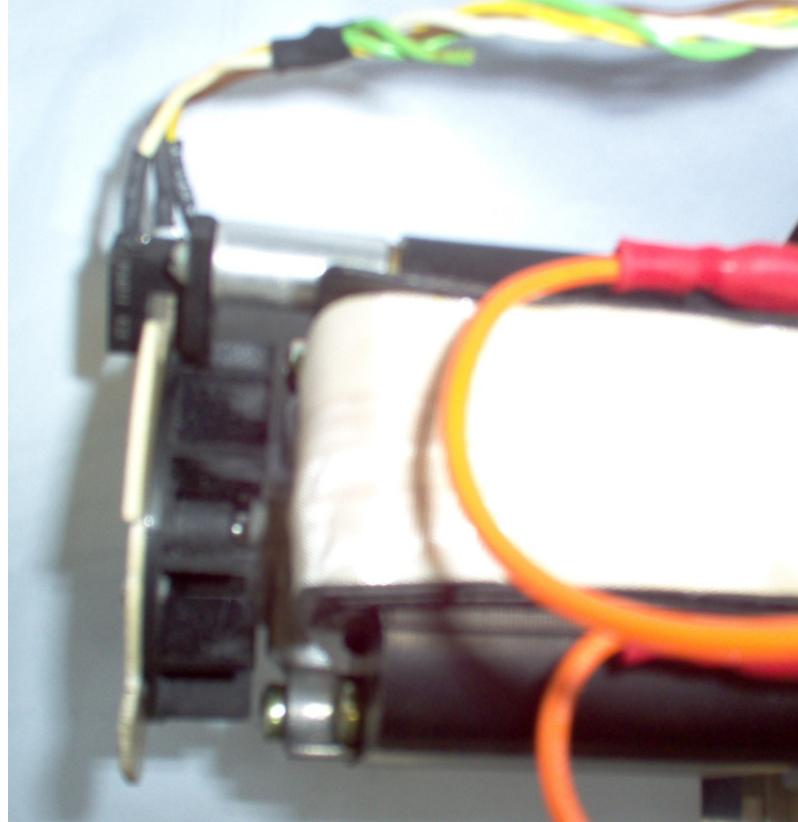
García, M. A.; Alvarez, G.

“Diseño e implementación de un sistema de control de bombas peristálticas para dosificación de líquidos.”

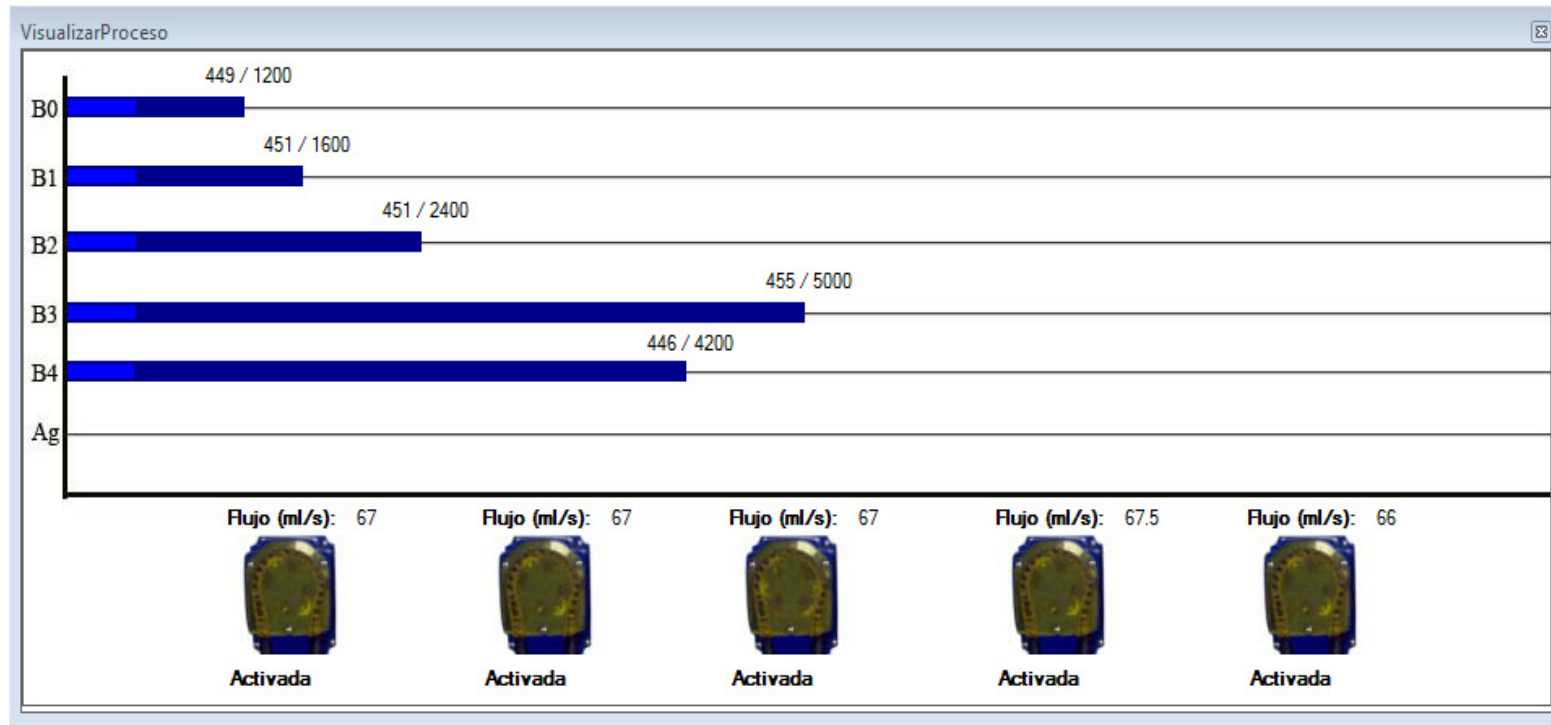
Anexo 4: Interfaz terminada.



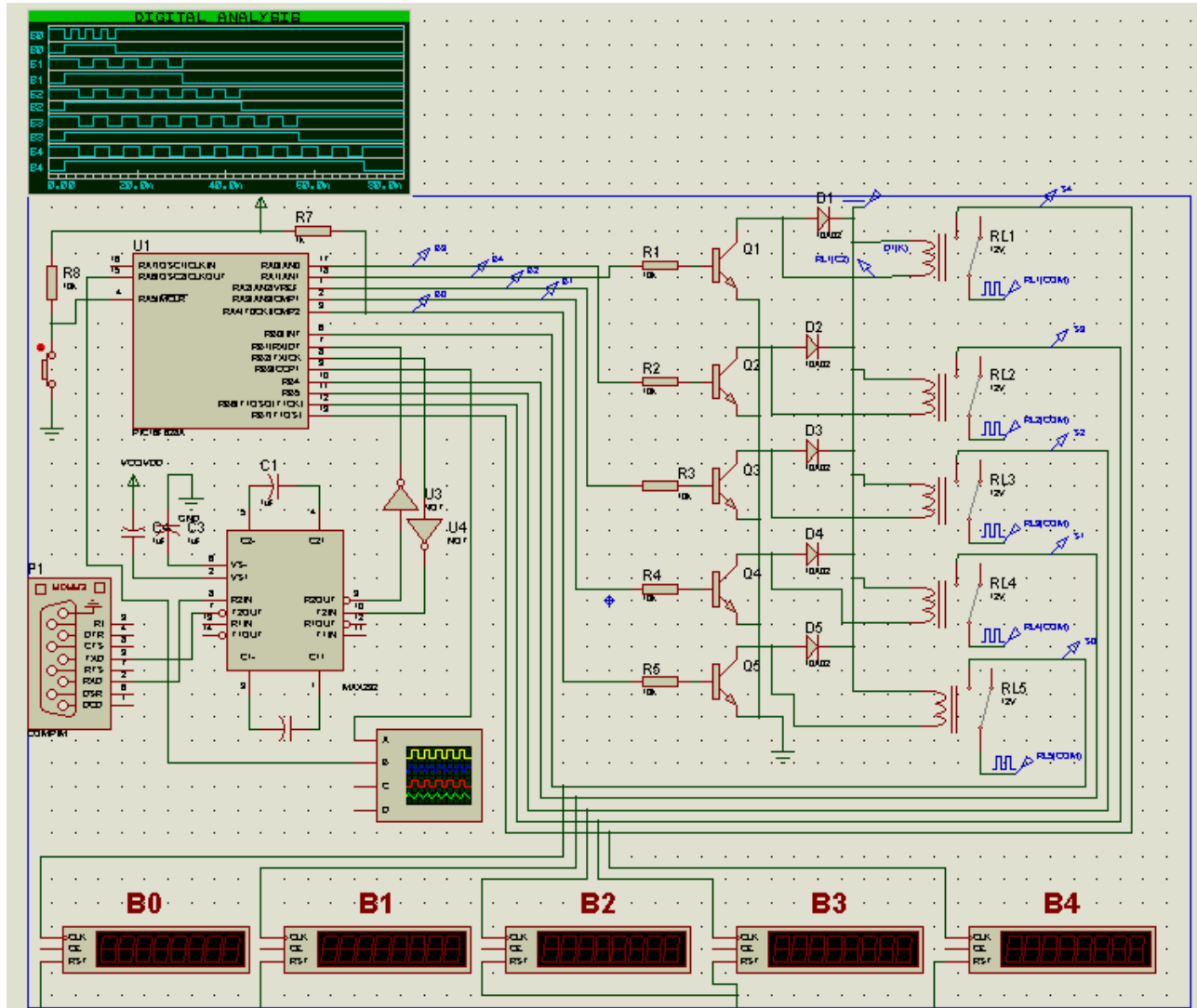
Anexo 5: Sistema de medición montado.



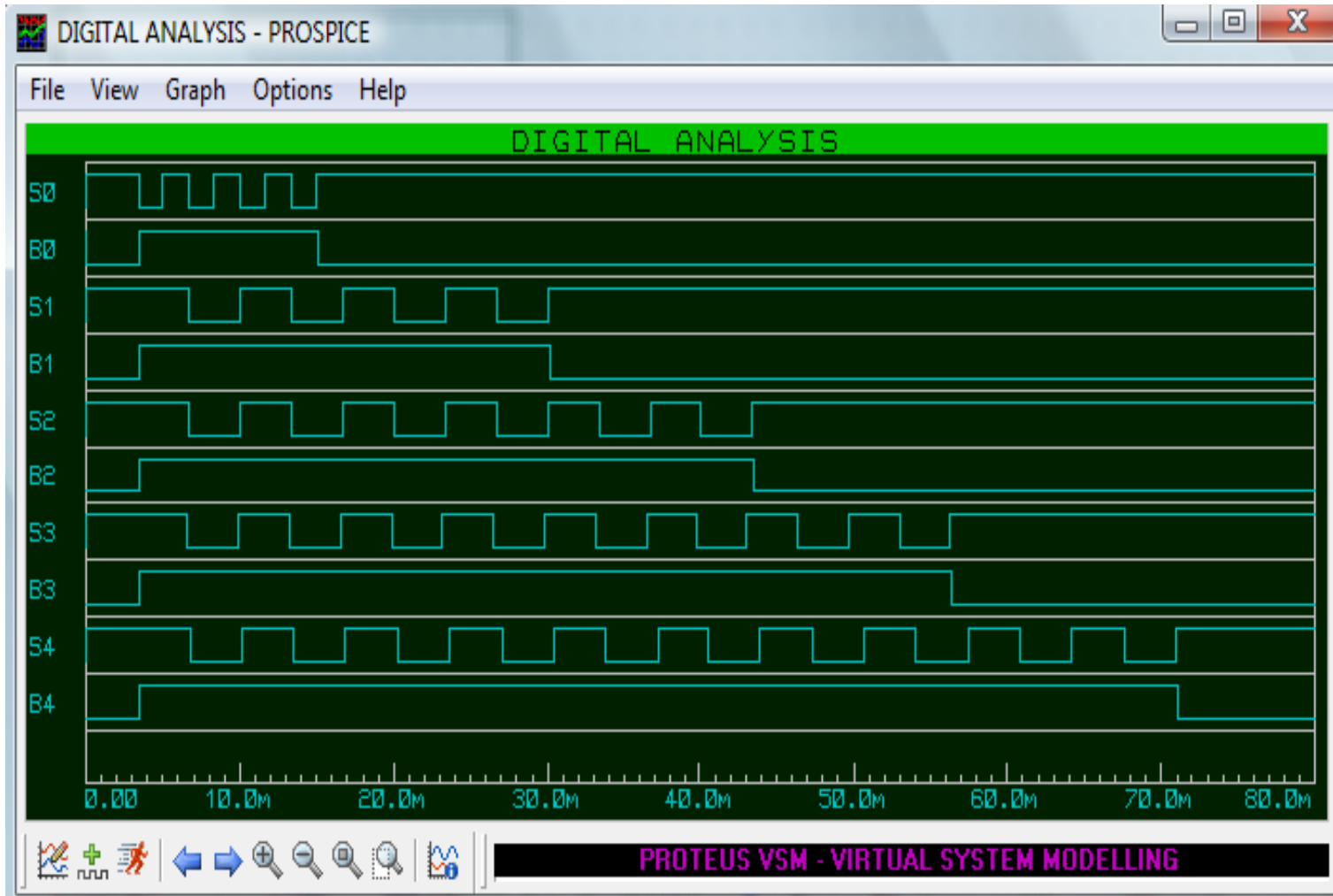
Anexo 6: Ventana para monitoreo del proceso en el DOSIMAQ.



Anexo 7: Circuito utilizado para la simulación de la interfaz.



Anexo 8: Resultados de la simulación del experimento 1.



Anexo 9: Resultados del experimento 2.

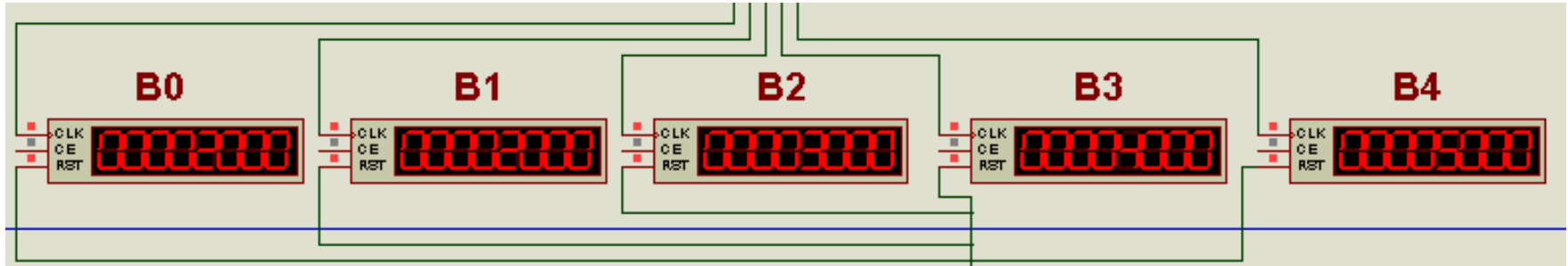


Fig. 9 Contadores que ilustran pulsos recibidos.

Address	Symbol Name	Value	Hex	Decimal	Char
065	pulsosB				
065	[0]	0x07D0	0x07D0	2000	..
067	[1]	0x0FA0	0x0FA0	4000	..
069	[2]	0x1770	0x1770	6000	.p
06B	[3]	0x1F40	0x1F40	8000	.@
06D	[4]	0x2710	0x2710	10000	'.

Fig. 10 Vista en MPLAB de la cantidad de pulsos almacenados.

Anexo 10: Resultados del experimento 2.

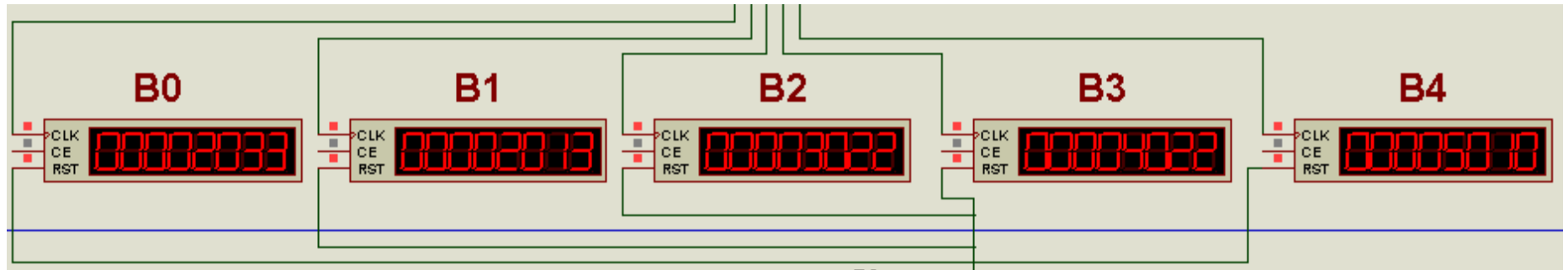


Fig. 11 Contadores que ilustran pulsos recibidos.