

Introducción a Arduino

Daniela Arely Morales Hernández
Ana Carolina Sandoval Mejía 152808
Stephanie Lizeth Malvaes Diaz

Abstract — Como primer acercamiento al laboratorio de Principios de Mecatrónica realizamos algunos experimentos que sirven para refrescar el conocimiento adquirido en cursos anteriores respecto al armado de circuitos eléctricos, fórmulas y cálculos teóricos, así como comparación con datos obtenidos prácticamente. Logramos exitosamente realizar dos programas con el Arduino: uno que indicará mediante un led cuando un sensor recibe muy poca luz y otro que desplegará en un display de 7 segmentos un dígito hexadecimal previamente introducido mediante el teclado.

Index Terms — Métodos, pushBoton, led, low, high.

I. Introducción.

El objetivo de la práctica es familiarizarse con el micro-controlador utilizando la plataforma Arduino para desarrollar sistemas interactivos. Haciendo uso de los puertos analógicos y digitales capaces de enviar y recibir información para procesarla e implementarla en nuestro sistema..

II. Conceptos.

Arduino está compuesto por una placa AVR de la familia de micro-controladores RISC de la compañía de semiconductores ATMEL. La forma en el que el sistema trabaja es mediante la recepción de datos a través de sus pines analógicos y digitales, los cuales pueden ser conectados a una gran cantidad de

sensores, motores u otros actuadores para enviar y recibir información.

III. Desarrollo

A. Linux

La primera parte del laboratorio consistió en realizar diferentes operaciones en la terminal de Linux para familiarizarse con los conceptos y comandos que se utilizan para poder realizar distintos procedimientos. Al acceder a GitHub creamos y clonamos un repositorio en el que se creó un repositorio, donde eliminamos, movimos y cambiamos de nombre directorios y archivos. Al final logramos ver los cambios ejecutados.

B. Blink.

En la primera parte ejecutamos el código del blink el cual prendía intermitentemente el LED conectado al cuerpo especificado. El siguiente inciso consistía en lograr que un led de la placa AVR se prendiera y apagara al sobrepasar una frecuencia determinada usando como sensor una fotoresistencia. En esta parte logramos entender la manera en que se inician variables y la manera en que son empleadas cada una en el código. Para su conexión usamos un fotoresistor, un led y dos resistencias diferentes para cada elemento. Se conectaron a los pines del Arduino y con el comando `digitalAnalog()` se leía la señal del fotoresistor.

Agregamos un botón que se conectó a una resistencia de 1000 ohms, para que cambiara el estado del LED sin importar la intensidad de la luz recibida por el fotoresistor. El botón se declaró como una entrada de tipo INPUT_PULLUP y se usó la funcionalidad digitalRead() para cambiar el estado del botón y digitalWrite() para que esté ejecutando la acción. El código funciona según cuatro casos, del siguiente modo:

- Si el *IdStatus*, el estado del fotoresistor, es menor o igual a 50 el led se enciende;
- Al mantener presionado el botón, el led se apaga durante el tiempo que esté presionado el botón.
- Si *IdStatus* es mayor a 50, entonces el led se mantiene apagado,
- Si el botón se mantiene presionado el botón el led estará prendido.

Esto es apreciable en la siguiente parte del código:

```
if (IdStatus <= 50) {
    digitalWrite(ledPin, HIGH);
    if(digitalRead(botPin)==LOW){

        digitalWrite(ledPin, LOW);
    }
}
else{
    digitalWrite(ledPin, LOW);
    if(digitalRead(botPin)==LOW){
        digitalWrite(ledPin, HIGH);
    }
}
```

C. Números hexadecimales.

El tercer inciso consistía en realizar programa capaz de recibir un dígito en hexadecimal (0-F) por medio del teclado y lo desplegará en un display de siete

segmentos. Para lo cual primero declaramos un arreglo de enteros de 17*7 llamado *num_array*. Cada fila del arreglo representa un dígito en hexadecimal además de un caso extra el cual imprime una raya horizontal si se introduce cualquier otro carácter. Las celdas de cada fila corresponden a un segmento del display y éstos se ordenaron alfabéticamente a lo largo de las columnas. De esta manera, cada fila (o dígito) contiene en sus celdas un 0 ó 1 para indicar si ese segmento se mantiene apagado o encendido respectivamente para así poder desplegar correctamente los dígitos en el display. Posteriormente dentro del *loop* declaramos casos para condicionar la salida del display dependiendo de la entrada recibida, *Num_Write* era la que se encargaba de traducir la instrucción y mandarlo al display con ayuda del arreglo *num_array* y la función *digitalWrite*. A continuación se muestra el método *Num_Write*:

```
//Num_Write
void Num_Write(int number)
{
    int pin= 2;
    for (int j=0; j < 7; j++) {
        digitalWrite(pin, num_array[number][j]);
        pin++;
    }
}
```

IV. Resultados

Se consiguió de manera efectiva cada paso de la práctica; las ramas de las carpetas se configuraron conforme el diagrama, y se hicieron todas las modificaciones solicitadas; el led se encendió y apagó según las especificaciones sobre luminiscencia que recibía el fotoresistor; el botón modificó el

estado del led al mantenerse presionado; el display de 7 segmentos encendió de manera adecuada representando los números hexadecimales ingresados por el teclado .

V. Conclusiones

Arely: Con esta práctica pudimos investigar y aprender sobre la forma de programar, los comandos y el lenguaje de Arduino. Es una buena introducción para aprender a usarlo. Durante el proceso tuvimos varios tropiezos con errores que al principio no entendíamos y poco a poco los fuimos investigando y solucionando.

Ana Carolina: Con esta práctica pudimos tener una introducción a los comandos que se usan en la terminal de Linux para manipular archivos y realizar distintas tareas. Además tuvimos una introducción al lenguaje de programación de Arduino, el cual es relativamente sencillo y parecido al de Quartus. También construimos circuitos como los realizados en Circuitos Lógicos lo que nos permitió practicar el armado, pues la práctica no tenía diagrama alguno de los circuitos y nosotras tuvimos que pensar cómo armar el circuito correctamente. Pude notar que el Arduino tiene mucho potencial para crear y concebir diversos sistemas embebidos.

Stephanie: Esta práctica esclareció el tema de los sistemas embebidos, al realizar funciones dedicadas, viendo como se comporta un sistema en tiempo real, usando señales tanto analógicas como digitales, usando un microcontrolador, y recordando los circuitos electrónicos. Además la práctica introdujo el uso de los comandos de linux, y el uso de la terminal, lo que es de suma utilidad. Todo esto me permitió ver la gran

utilidad de realizar sistemas de menor complejidad, pero que requieren de sensores y actuadores y que sean usables en la vida diaria. Mi rol en esta práctica fue la programación y el armado de circuitos, en conjunto con mis compañeras.

Referencias

1.<http://lacajamakerspace.org/index.php/electronica/arduino/tutoriales?download=1:manual-de-programacion-de-arduino>