Linux y Microcontrolador

Javier Montiel González C.U.159216, Andrea González Cardoso C.U. 157961,
Diego Villalvazo Suzer C.U. 155844

Resumen—En esta práctica se utilizó GitHub y Arduino con el objetivo de familiarizarse con cada uno ellos. Se buscó aprender el lenguaje de programación para realizar dos tareas.

1 Introducción

Los sistemas embebidos se utilizan en diversas áreas de la Ingeniería por lo que su manejo es de vital importancia. La cantidad de aplicaciones que permiten implementar los microcontroladores han permitido su uso en dispositivos que utilizamos en nuestro día a día.

Como ingenieros buscamos facilitar la realización de las tareas que nos resultan tediosas, los microcontroladores son un recurso de gran ayuda al momento de diseñar una solución a dicho problema.

3 Marco Teórico

Linux es un Sistema Operativo (SO) tipo Unix diseñado para aprovechar al máximo las capacidades de las computadoras. Es un SO con capacidades de multiprocesamiento, multitarea y multiusuario.

El Microcontrolador es un circuito integrado que es el componente principal de una aplicación embebida. Este incluye sistemas para controlar elementos de entrada/salida. También incluye a un procesador y una memoria en la que puede guardar el programa sus variables (flash y RAM). Su función es la de automatizar procesos y procesar información.

El microcontrolador se aplica en toda clase de inventos y productos donde se requiere seguir un proceso automático dependiendo de las condiciones de distintas entradas..

4 Desarrollo

Parte 1: El objetivo fue realizar un programa que simulara un alumbrado el cual debía reaccionar a variaciones en la intensidad luminosa del ambiente. Para llevar esto a cabo se utilizó una resistencia y se programó el microcontrolador; el código se muestra a continuación:

```
int pinLed = 4;
int pinLDR = A0;
int valorLDR;

void setup() {
  pinMode(pinLed, OUTPUT);
  Serial.begin(9600);
}
```

En esta sección del código se declararon las variables e inició el programa.

```
void loop() {
  digitalWrite(pinLed, LOW);
  valorLDR = analogRead(pinLDR);
  Serial.println(valorLDR);
  if(valorLDR > 995){
       digitalWrite(pinLed, HIGH);
  }
  delay(200);
}
```

En esta sección del código se escribió el código que va a estar corriendo de forma constante y se determinó el valor mínimo para que se prenda el alumbrado (led).

Parte 2: El objetivo fue realizar un programa que al recibir un número del teclado (en hexadecimal), se desplegara éste en un display de 7 segmentos. El código utilizado se muestra a continuación:

```
int num_array[17][7] =
       {{ 1,1,1,1,1,1,0 }, //0
       { 0,1,1,0,0,0,0 },  //1
       { 1,1,0,1,1,0,1 },  //2
       { 1,1,1,1,0,0,1 },  //3
       { 0,1,1,0,0,1,1 },  //4
       { 1,0,1,1,0,1,1 },  //5
       { 1,0,1,1,1,1,1 },  //6
       { 1,1,1,0,0,0,0 },  //7
       { 1,1,1,1,1,1,1 },  //8
       { 1,1,1,1,0,1,1 },  //9
       { 1,1,1,0,1,1,1 },  //A
       { 0,0,1,1,1,1,1 },  //b
       { 1,0,0,1,1,1,0 },  //C
       { 0,1,1,1,1,0,1 },  //d
       { 1,0,0,1,1,1,1 },  //E
       { 1,0,0,0,1,1,1 },  //F
       { 0,0,0,0,0,0,1 }}; //-
```

En esta sección del código se determinaron los segmentos que debían prenderse dependiendo el valor ingresado.

```
void Num_Write(int);

void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);

  Serial.begin(9600);
}

void loop()
{

  // check for incoming serial data:
  if (Serial.available() > 0) {
        int num = Serial.read();2

        switch(num){
        case 48:
        Num_Write(0);
        break;
        case 49:
        Num_Write(1);
        break;
        case 50:
        Num_Write(2);
        break;
        case 51:
        Num_Write(3);
        break;
        case 52:
        Num_Write(4);
        break;
        case 53:
        Num_Write(5);
        break;
        case 54:
        Num_Write(6);
        break;
        case 55:
        Num_Write(7);
        break;
        case 56:
        Num_Write(8);
        break;
        case 57:
        Num_Write(9);
        break;
        case 97:
        Num_Write(10);
        break;
        case 65:
        Num_Write(10);
        break;
        case 98:
        Num_Write(11);
        break;
        case 66:
        Num_Write(11);
        break;
        case 99:
        Num_Write(12);
        break;
        case 67:
        Num_Write(12);
        break;
        case 100:
        Num_Write(13);
        break;
        case 68:
        Num_Write(13);
        break;
        case 101:
        Num_Write(14);
        break;
        case 69:
        Num_Write(14);
        break;
        case 102:
        Num_Write(15);
        break;
        case 70:
        Num_Write(15);
        break;
        default:
        Num_Write(16);
        break;
        }
  }
}
```

En esta sección del código se leen y validan los datos ingresados en el teclado. Al mismo tiempo se convierte el valor input a decimal (ya que se lee en código ASCII) para poder desplegarlo en el display.

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

Esta última sección recibe el número que debe desplegarse en el display y va recorriendo el elemento del arreglo correspondiente para lograrlo.

5 Resultados

Cada apartado de la práctica se realizó con éxito. Los circuitos que se elaboraron para la realizar las pruebas en Arduinos se presentan a continuación:

Parte 1

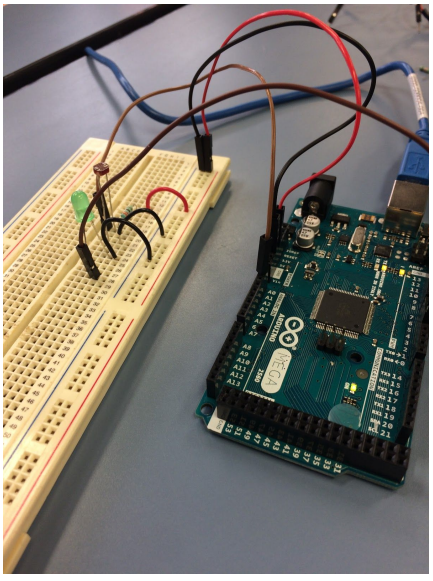


Imagen 1: Con luz

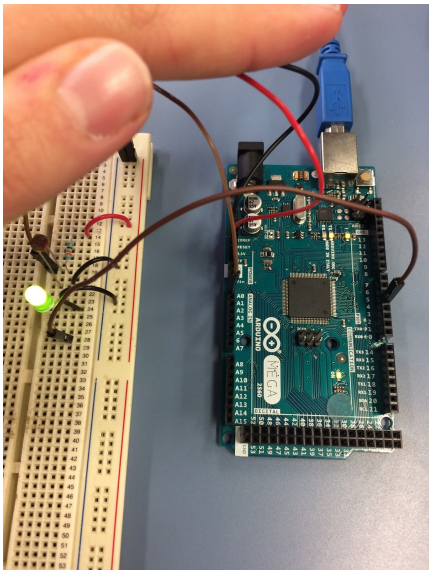
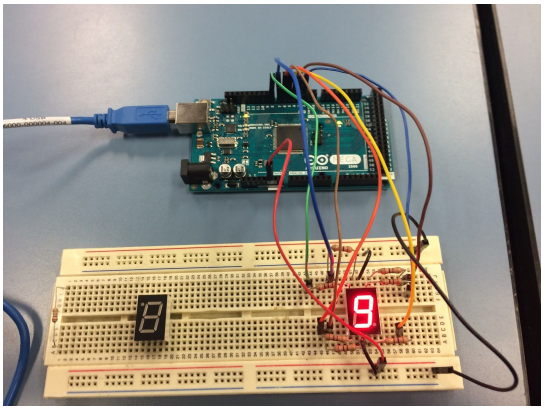


Imagen 2: Sin luz

Parte 2



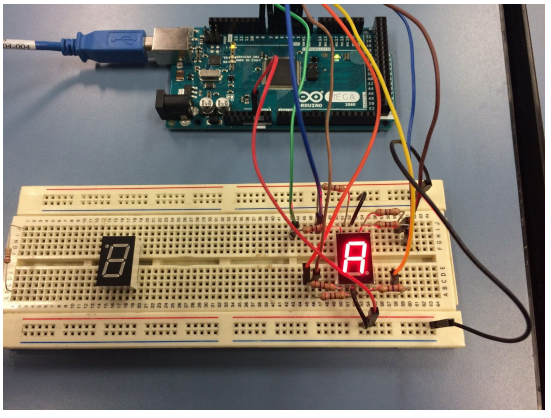


Imagen 3 y 4: Se escribió un número hexadecimal en el teclado de la computadora

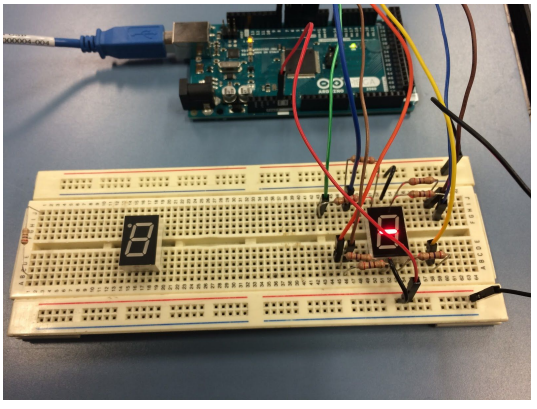


Imagen 5: Se escribió algún caracter que no fuera un número hexadecimal

6 Conclusiones

6.1 Figures and Tables

Andrea: Con esta práctica pudimos empezar a entender el funcionamiento básico de un Arduino, al igual que los principios para poder programar en este. Además de entender y practicar algunos comandos básicos de Linux.

Creo que fue una buena práctica introductoria que nos ayudó a familiarizarnos con las herramientas de trabajo que estaremos utilizando en el resto del curso. Se llegó a los objetivos deseados gracias a un buen trabajo en equipo.

Javier: El realizar estos programas fue de gran ayuda para familiarizarnos con la interfaz de programación de Arduino. También el trabajar con la terminal de Linux nos mostró la manera de guardar programas en el repositorio de Github.

Diego: En la práctica aprendimos lo básico de Github, Linux y Arduino. En realidad fue una práctica sencilla, pero creo que su propósito fue que nos familiarizaremos con las interfaces de los nuevos programas y del nuevo lenguaje de programación. Creo que como equipo logramos los objetivos que nos propusimos e hicimos un buen trabajo en general

Bibliografía

[1] Linux. Consultado el 31 de enero de 2018. Disponible en http://xml.cie.unam.mx/xml/Linux/glinux-2.html

[2] Microcontrolador. Consultado el 31 de enero de 2019. Disponible en https://hetpro-store.com/TUTORIALES/microcontrolador/