

MATERIA DE SISTEMAS EMBEBIDOS-LABORATORIO 2

PUERTOS DIGITALES Y COMUNICACION SERIAL

Edwin A. Solano Arciniegas

8 de diciembre de 2020

1. Introducción

Se diseña en Arduino un sistema de ingreso el cual consta de una simulación en proteus con Arduino uno, Se realiza un programa en Arduino uno en el cual con el uso de interrupciones se puede seleccionar diferentes modos, como lo son reloj, configuración de hora de manera análoga a digital, un juego de luces y detección de sensores, para el uso de esta se incluye librerías Ms-Timer2 y LiquidCrystal, este sistema consta de inicio del sistema y apagado con un botón y el otro modo consta del sistema que se requiere ejecutar nombrados anteriormente.

Interrupción de power on, de tal manera que el modo siguiente es fijar la hora de manera análoga y convertirla a digital de este modo se puede configurar la hora actual, el siguiente modo inicia un juego de leds que está cargado preliminarmente, duarte 20 minutos en este caso 20 segundos dado el sistema en que los minutos son segundos por cuestiones de simulación, el tercer modo es detección de sensores activos que manda un mensaje serial cuando uno de estos o varios se hayan activado, con otra interrupción power off, se apaga todo el sistema.

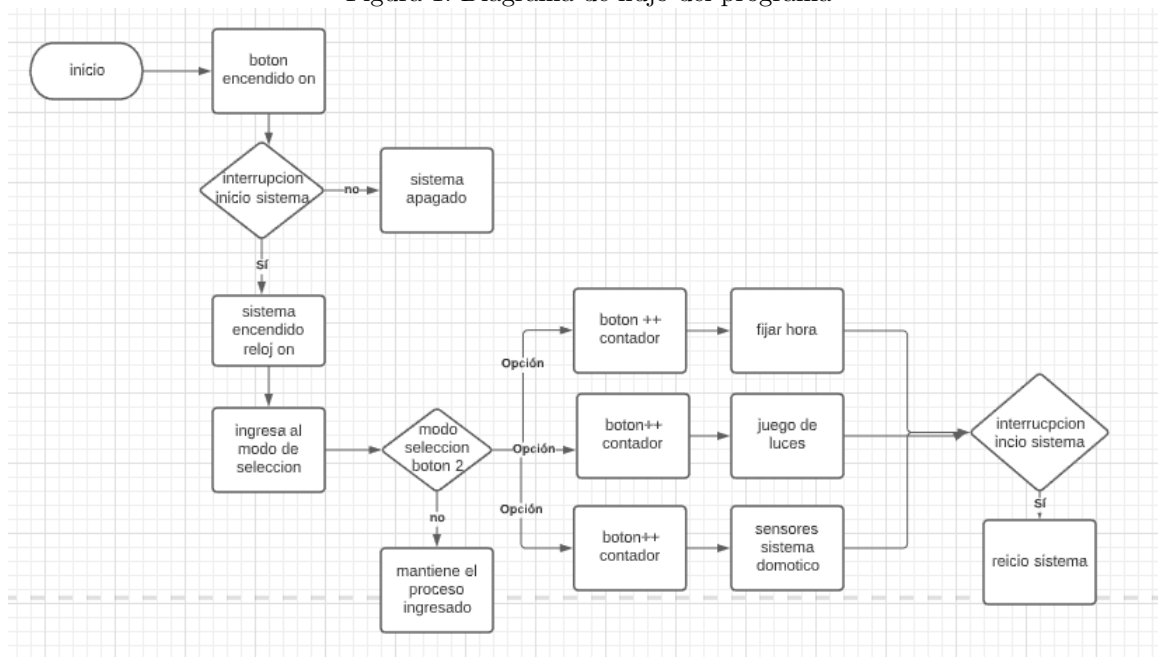
El reloj arranca cuando se inicia la primera in-

2. Diseño del Sistema

2.1. Diagrama de Flujo

El diagrama de flujo explica cómo se comporta el sistema con las interrupciones y las maneras de elegir los diferentes modos:

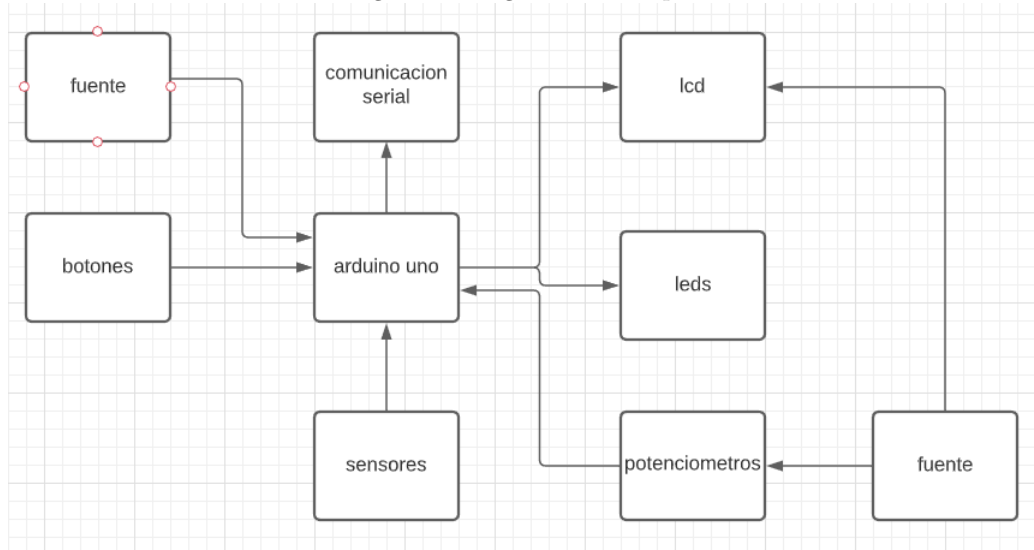
Figura 1: Diagrama de flujo del programa



2.2. Diagrama de Bloques

En el diagrama de bloques es posible identificar que componentes están conectados al Arduino que sería el sistema principal, de tal manera que se identifica los componentes de entrada y los componentes de salida para los diferentes modos

Figura 2: Diagrama de bloques



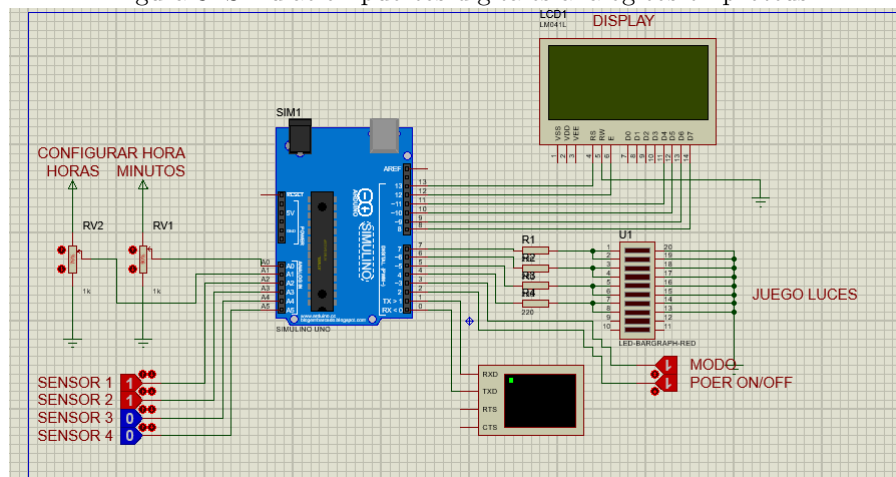
3. Desarrollo

3.1. Simulación

3.1.1. Simulación Proteus

Se realiza la simulación en proteus así cargar el programa en el sistema y verificar el funcionamiento de este sea correcto de tal manera que es posible corregir errores y depurar el sistema antes de armarlo completamente.

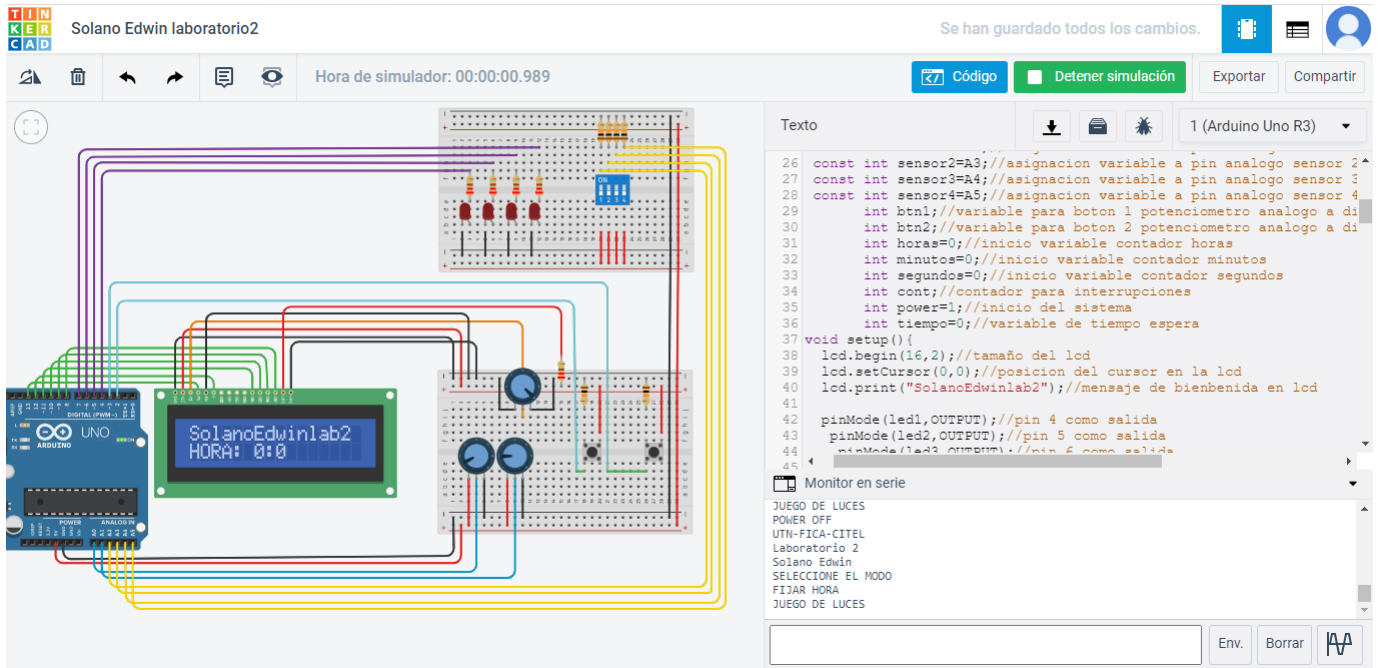
Figura 3: Simulación puertos digitales analógicos en proteus



3.1.2. Simulación Tinkercad

Se realiza la simulación el programa en línea Tinkercad el cual es posible hacer una simulación casi real del como funcionaria el sistema de tal manera que se lo puede adaptar, aunque este programa no es muy completo ayuda a guiarse de manera didáctica de las conexiones de este.

Figura 4: Simulacion en Tinkercad



4. Análisis de Resultados

4.1. programa completo

https://github.com/edwinsolano/SolanoEdwin_laboratorio2.git

4.2. codigo arduino

Ingreso de librerías, variables a usar, contadores, pines de entrada:

Figura 5: ingreso de variables

```

#include <MsTimer2.h>//llamar a libreria MsTimer dos
#include<LiquidCrystal.h>//llamar a libreria de lcd
LiquidCrystal lcd (13,12,11,10,9,8);//configuracion de pines lcd
const int led1=4;//asignacion variable led 1 pin 4
const int led2=5;//asignacion variable led 2 pin 5
const int led3=6;//asignacion variable led 3 pin 6
const int led4=7;//asignacion variable led 4 pin 7
const int sensor1=A2;//asignacion variable a pin analogo sensor 1
const int sensor2=A3;//asignacion variable a pin analogo sensor 2
const int sensor3=A4;//asignacion variable a pin analogo sensor 3
const int sensor4=A5;//asignacion variable a pin analogo sensor 4
int btn1;//variable para boton 1 potencio metro analogo a digital horas
int btn2;//variable para boton 2 potencio metro analogo a digital minutos
int horas=0;//inicio variable contador horas
int minutos=0;//inicio variable contador minutos
int segundos=0;//inicio variable contador segundos
int cont;//contador para interrupciones
int power=0;//inicio del sistema
int tiempo=0;//variable de tiempo espera
void setup() {

```

codigo para las interrupciones de cada boton, para cmabiar de encendido a apagado y cambiar de modo

Figura 6: interrupciones

```
}
void loop() {
  if(power==1){//encendido del sistema
    reloj();//llama al metodo reloj enciende
  }
  if(cont==1){//interrupcion
    andigHoras();//llama al metodo configurar hora
  }
  if(cont==2){//interrupcion
    juegoLuces();//llama al metodo con juego de luces
  }
  if(cont==3){//interrupcion
    actSensor();//llama al metodo de sensores
  }
}

void activacion(){//interrupcion activacion del sistema
  switch(power){//encendido
    case 0://primer caso mensaje de bienvenida en CX y LCD
      Serial.println("UTN-FICA-CITEL");//mensaje en CX
      Serial.println("Laboratorio 2");//mensaje en CX
      Serial.println("Solano Edwin");//mensaje en CX
      Serial.println("SELECCIONE EL MODO");//mensaje en CX
      lcd.clear();
    }
  }
```

Metodo convertir datos analogos a digitales

Figura 7: codigo para convertit de analogo a digital

```
void andigHoras(){//convertor de analogo digital para configurar hora

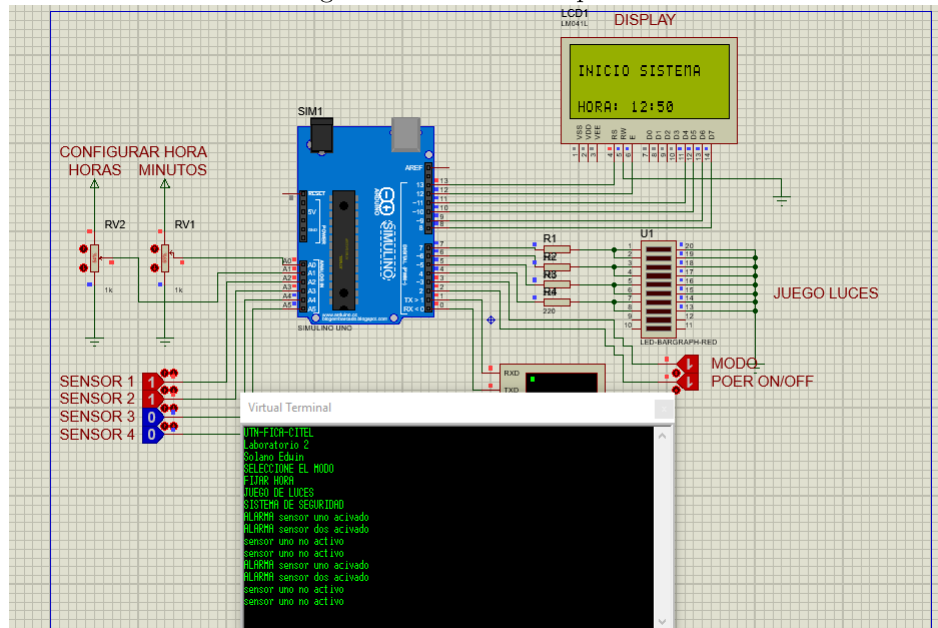
  btn1=analogRead(0);//leer el dato minutos
  btn1=map(btn1,0,1023,0,60);//conversion
  minutos=btn1;//asignacion de dato
  btn2=analogRead(1);//ler segundo dato horas
  btn2=map(btn2,0,1023,0,24);//conversion
  horas=btn2;//asignacion de dato
}
```

4.3. simulacion

Ejecución del programa en proteus:

- Se inicia el sistema con la primera interrupción el cual da un mensaje de bienvenida y comienza a contar el reloj, este comienza en 0 horas y 0 minutos.
- Se presiona el botón de modos con cada interrupción es posible cambiar un diferente juego, el primero es configurar la hora del reloj, el segundo indica un juego de leds, y el tercero entra a un sistema domótico que detecta si se ha activado algún sensor.
- Para poder configurar la hora esta se lo hace por medio de potenciómetros los cuales, al variarlos este varia la hora y los minutos, es un convertor de análogo a digital.
- Se muestra en una lcd y en mensaje serial todos los modos a configurar.
- Los sensores en este caso se uso logicstate para poder simular si esta en estado activo o desactivo.
- si se presiona el mismo botón de inicio el sistema se detiene y se reinicia todo.

Figura 8: simulación en proteus



5. Conclusiones y Recomendaciones

5.1. Conclusiones

Se obtiene un programa el cual puede interactuar en él, simplificado en un sistema domótico que aplicándolo en condiciones reales sería posible conectar a una casa he encender luces al azar y tener control de los sensores.

Se hace el uso de las librerías MsTimer2 para el funcionamiento del reloj, con esta librería es mucho mas sencillo hacer funcionar un reloj ya que se utiliza el contador que tiene internamente Arduino.

Se tiene resultados tanto por comunicación serial y por LCD que indica cuando se ha iniciado el sistema y cuando se apaga, además de contar con un armado en Tinkercad que proporciona de manera didáctica como sería el prototipo.

5.2. Recomendaciones

Se debe tener iniciado algunas variables en 0 para poder contar caso contrario no detectara las interrupciones así también el uso y la asignación de variables debe estar bien especificada como, por ejemplo, string, char, int, boolean, float.

Se identifica en Arduino que pines son analógicos y digitales y cuáles son los de las interrupciones, para así poder usar las librerías y que el código funcione en el hardware de manera correcta.

Es recomendable crear el código con métodos para cada situación ya que así cualquier error o cualquier depuración que se le desee hacer es más fácil ubicarse y poder solucionar el inconveniente.