

SISTEMAS EMBEBIDOS

LABORATORIO 1

Lisbeth C. Alban-Checa

30 de noviembre de 2020

1. Introducción

Las entradas digitales nos permiten distinguir entre dos estados: activo(encendido)– inactivo(apagado). Arduino UNO(ATmega328p) consta de 14 puertos digitales que pueden funcionar como salidas y como entradas.

-B (pines digitales del 8 al 13)

-C (entradas analógicas)

-D (pines digitales del 0 al 7) Cuando actúan como entradas trabajan con dos estados de tensiones: 0V (estado de no activación) y 5V (estado de activación).

La comunicación serial entre dos dispositivos se reali-

za a través del intercambio de una secuencia de bits, donde se transmite bit a bit, uno por vez, donde, aunque es lenta la comunicación, tiene la ventaja de poder ser transmitida a mayores distancias y utilizar menos líneas de comunicación. La comunicación serial entre dos dispositivos únicamente utiliza 3 líneas las cuales son:

-Línea de recepción de datos (RX)

-Línea de transmisión de datos (TX)

-Línea común (GND)

2. Consulta

- Consultar sobre la rotación de datos dentro de la librería LiquidCrystal para realizar la rotación de palabras ingresadas por teclado matricial.
Para la programación en Arduino vamos a utilizar una librería que nos hará más fácil el desarrollo. Esta librería es LiquidCrystal.h. No hace falta instalarla en el entorno de desarrollo oficial ya que viene por defecto. Lo único que tenemos que hacer es añadirla como un include en nuestro programa o sketch.
- Presentar el funcionamiento mediante diagrama de bloques y flujo.

3. Ejercicio Propuesto

- Se debe diseñar un sistema de control de acceso por medio de una contraseña individual.
- Las contraseñas ya se encuentran establecidas para cada persona y están almacenadas en el sistema. Estas son: Carlos Arias cod:ca900813, Andres Juarez cod:aj881112 y Javier Andrada cod:ja890109. Estos son ingresados por comunicación serial.
- Cada usuario debe ingresar su código y en una LCD deberá aparecer por rotación de datos: "BIENVENIDO NOMBRE APELLIDO".
- Al presionar un botón, se deberá presentar como reporte por mensaje serial quien ya ingresó a la empresa y quien no lo ha hecho.
- El resto de restricciones son propuestas por cada estudiante.

4. Desarrollo

4.1. Simulación

5. Análisis de Resultados

Como podemos observar el resultado final ha sido satisfactorio ya que podemos observar lo que se nos fue pedido en el desarrollo de este laboratorio con sus respectivos ítems.

6. Conclusiones y Recomendaciones

6.1. Conclusiones

- La utilización de la comunicación serial es muy útil en cuanto al ingreso de claves e información que se desea adquirir de manera fácil.
- La utilización del sistema arduino y la simulación del programa en proteus es de gran ayuda para el desarrollo de esta materia.

6.2. Recomendaciones

- Revisar que las librerías sean las indicadas para la realización de los proyectos a ejecutarse.
- Saber representar la lógica de programación al sistema de simulación esto hará que se facilite los procesos al momento de crear nuestro proyecto.

Figura 1: DIAGRAMA DE FLUJO

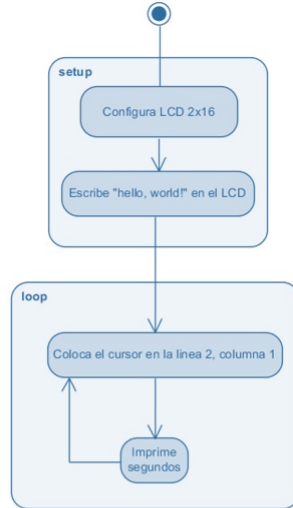


Figura 2: CODIGO ARDUINO

```

ALBANLISSETH_EMBEBIDOS_LAB_1 Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

ALBANLISSETH_EMBEBIDOS_LAB_1$
*LABORATORIO 1: PUERTOS DIGITALES Y COMUNICACION SERIAL
*Lisseth Alban
*Sistemas Embebidos
*Ing. Telecomunicaciones
*/
#include<LiquidCrystal.h>
// <libreria.h>
// "libreria.h"

LiquidCrystal lcd (13,12,11,10,9,8);
const char vector [11]= {'C','A','R','L','O','S',' ','A','R','I','A','S'};
int i;//variable para el vector

void setup() {
  // put your setup code here, to run once:
  lcd.begin(16,2);
}

void loop() {
  // put your main code here, to run repeatedly:
  lcd.setCursor(0,0); //ubicacion en lcd
  lcd.print("BIENVENIDO LISSETH ALBAN"); //envia mensaje
  delay(500);
  lcd.clear(); //borra la pantalla
  delay(500);
}

Guardar Cancelado.
El nombre del sketch ha sido modificado.
Los nombres de Sketch deben empezar con una letra o número, seguido por letras,
números, guiones, puntos y subrayados. La longitud máxima es de 63 caracteres.
25 Arduino Uno

```

Figura 3: SIMULACION

