|  |
| --- |
| /\*  \* Displays text sent over the serial port (e.g. from the Serial Monitor) on  \* an attached LCD.  \* YWROBOT  \*Compatible with the Arduino IDE 1.0  \*Library version:1.1  \*/    #define \_OPEN\_SYS\_ITOA\_EXT  #include <stdio.h>  #include <stdlib.h>  #include <math.h>  #include <Wire.h>  #include <LiquidCrystal\_I2C.h>  //funciones  void parpadeo();  void parar();  void izquierda();  void derecha();  void retroceder();  void avanzar();  void movimiento(char ind);  void enviar\_lcd(unsigned long d);  unsigned long distancia (const char trig, const char echo);  void sensar();  //pines  //el sentido de asignacion de pines es de izquierda a derecha  //pines de sensores  const char echo1 = 53;  const char trig1 = 52;  const char echo2 = 51;  const char trig2 = 50;  const char echo3 = 49;  const char trig3 = 48;  //pines de los motores  const char motAPWM = 02;//PIN DEL PWM DEL MOTOR B //ENABLE B DEL MODULO DE POTENCIA  const char motBPWM = 03;//PIN DEL PWM DEL MOTOR A //ENABLE A DEL MODULO DE POTENCIA  const char motA\_P1 = 41;//IN4 DEL MODULO DE POTENCIA PARA EL MOTOR A  const char motA\_P2 = 40;//IN3 DEL MODULO DE POTENCIA PARA EL MOTOR A  const char motB\_P1 = 39;//IN2 DEL MODULO DE POTENCIA PARA EL MOTOR B  const char motB\_P2 = 38;//IN1 DEL MODULO DE POTENCIA PARA EL MOTOR B  const char motAENC = 30;//PIN DEL ENCODER DEL MOTOR A  const char motBENC = 31;//PIN DEL ENCODER DEL MOTOR B  //pines de los leds  const char led\_on = 13;//led del sistema/led de marcha y parada del carro  const char led\_ade = 14;  const char led\_ret = 16;  const char led\_der = 17;  const char led\_izq = 18;  /\* otros pines:  \* I2C0: 20 y 21 para el lcd  \* UART0: 00 y 01 para la pc  \* UART1: 19 y 18 para el modulo de voz  \* UART2: 17 y 16 para el modulo bluetooh  \*/  //variables  char PWMA=0;//PWM DEL MOTOR A (0-255)  char PWMB=0;//PWM DEL MOTOR B (0-255)  char DIR\_I2C=0x27;//direccion del adress del I2C  char mov=0;  String inputString = ""; // a String to hold incoming data  bool stringComplete = false; // whether the string is complete  // set the LCD address to 0x27 for a 16 chars and 2 line display  LiquidCrystal\_I2C lcd(DIR\_I2C,20,4);  void setup()  {  //inicializar el lcd  lcd.init();  lcd.backlight();  lcd.clear();  lcd.print(" Silla de Ruedas ");  lcd.setCursor(0,1);  lcd.print(" UTP ");    //inicializar puerto serial usart  Serial.begin(9600);  inputString.reserve(30);  Serial1.begin(9600);  Serial2.begin(9600);    //inicializar pines  //inicializar pines del sistema  pinMode(led\_on , OUTPUT); //pin como salida  pinMode(led\_ade, OUTPUT); //pin como salida  pinMode(led\_ret, OUTPUT); //pin como salida  pinMode(led\_izq, OUTPUT); //pin como salida  pinMode(led\_der, OUTPUT); //pin como salida  digitalWrite(led\_on , LOW);//Inicializamos el pin con 0  digitalWrite(led\_ade, LOW);//Inicializamos el pin con 0  digitalWrite(led\_ret, LOW);//Inicializamos el pin con 0  digitalWrite(led\_izq, LOW);//Inicializamos el pin con 0  digitalWrite(led\_der, LOW);//Inicializamos el pin con 0  //inicializar pines del sensor  pinMode(trig1, OUTPUT); //pin como salida  pinMode(echo1, INPUT ); //pin como entrada  pinMode(trig2, OUTPUT); //pin como salida  pinMode(echo2, INPUT ); //pin como entrada  pinMode(trig3, OUTPUT); //pin como salida  pinMode(echo3, INPUT ); //pin como entrada  digitalWrite(trig1, LOW);//Inicializamos el pin con 0  digitalWrite(trig2, LOW);//Inicializamos el pin con 0  digitalWrite(trig3, LOW);//Inicializamos el pin con 0  //inicializar pines del motor  pinMode(motAPWM, OUTPUT); //pin como salida  pinMode(motA\_P1, OUTPUT); //pin como salida  pinMode(motA\_P2, OUTPUT); //pin como salida  pinMode(motAENC, INPUT ); //pin como entrada  pinMode(motBPWM, OUTPUT); //pin como salida  pinMode(motB\_P1, OUTPUT); //pin como salida  pinMode(motB\_P2, OUTPUT); //pin como salida  pinMode(motBENC, INPUT ); //pin como entrada  digitalWrite(motAPWM, LOW);//Inicializamos el pin con 0  digitalWrite(motA\_P1, LOW);//Inicializamos el pin con 0  digitalWrite(motA\_P2, LOW);//Inicializamos el pin con 0  digitalWrite(motBPWM, LOW);//Inicializamos el pin con 0  digitalWrite(motB\_P1, LOW);//Inicializamos el pin con 0  digitalWrite(motB\_P2, LOW);//Inicializamos el pin con 0  parar();  parpadeo();  delay(2000);  }  char PWM\_0=80;  void loop()  {  lcd.clear();  lcd.setCursor(0,0);  lcd.print("modulo");  sensar();  movimiento(mov);  delay(200);  if (stringComplete) {  Serial.println(inputString);  if(inputString=="00\n"){mov=0;}  if(inputString=="01\n"){mov=1;}  if(inputString=="02\n"){mov=2;}  if(inputString=="03\n"){mov=3;}  if(inputString=="04\n"){mov=4;}  Serial.println(mov,10);  // clear the string:  inputString = "";  stringComplete = false;  }    // Serial.print("pwm:");  // Serial.print(PWMA,10);  // Serial.print(" y ");  // Serial.print(PWMB,10);  // Serial.println();  // delay(2000);  }  void serialEvent() {  while (Serial.available()) {  // get the new byte:  char inChar = (char)Serial.read();  // add it to the inputString:  inputString += inChar;  // if the incoming character is a newline, set a flag so the main loop can  // do something about it:  if (inChar == '\n') {  stringComplete = true;  }  }  }  void sensar(void){  unsigned long d=0;  unsigned long temp\_d=99;  unsigned long count=0;  // do{  d=distancia(trig1,echo1);  if(d<20){temp\_d=d;}  if(d<10){count=count+1;}  while(d!=0){d=0;}  d=distancia(trig2,echo2);  if(d<20){if(temp\_d>d){temp\_d=d;}}  if(d<10){count=count+2;}  while(d!=0){d=0;}  // d=distancia(trig3,echo3);  // if(d<20){if(temp\_d>d){temp\_d=d;}}  // if(d<10){count=count+2;}  // while(d!=0){d=0;}    if(temp\_d!=99){enviar\_lcd(temp\_d);}  temp\_d=99;  if(count!=0){mov=0;}  count=0;  // }while(count!=0);  return;}  unsigned long distancia (const char trig, const char echo){  unsigned long t=0;  unsigned long d=0;  digitalWrite(trig, HIGH);  digitalWrite(trig, HIGH);  delayMicroseconds(12); //Enviamos un pulso de 10us  digitalWrite(trig, LOW);  digitalWrite(trig, LOW);    t = pulseInLong(echo, HIGH);  delayMicroseconds(200); //8 pulsos que se necesitan para que se pueda volver a activar el trigger  d=(t)\*(346.13)/20000;  t=0;  return d;  }  void enviar\_lcd(unsigned long d){ //envio de distancia a l lcd  unsigned long temp\_d=0;  temp\_d=d;  lcd.clear();  lcd.setCursor(0,0);  lcd.print("modulo");  lcd.setCursor(0,1);// quitar para mandar una posicion    char buf [6]="";  ultoa(temp\_d,buf,10);  for(char i=0;i<6;i++){  if((buf[i]>=0x30) & (buf[i]<=0x39)){lcd.write(buf[i]);  }}  lcd.print("cm.");  while(temp\_d!=0){temp\_d=0;}  delay(100);  return;}  void movimiento(char ind){  switch(ind){  case 0:  parar();  digitalWrite(led\_on , LOW );  digitalWrite(led\_ade, LOW );  digitalWrite(led\_ret, LOW );  digitalWrite(led\_izq, LOW );  digitalWrite(led\_der, LOW );  break;    case 1:  avanzar();  digitalWrite(led\_on , HIGH);  digitalWrite(led\_ade, HIGH);  digitalWrite(led\_ret, LOW );  digitalWrite(led\_izq, LOW );  digitalWrite(led\_der, LOW );  break;    case 2:  izquierda();  digitalWrite(led\_on , HIGH);  digitalWrite(led\_ade, LOW );  digitalWrite(led\_ret, LOW );  digitalWrite(led\_izq, HIGH);  digitalWrite(led\_der, LOW );  break;    case 3:  derecha();  digitalWrite(led\_on , HIGH);  digitalWrite(led\_ade, LOW );  digitalWrite(led\_ret, LOW );  digitalWrite(led\_izq, LOW );  digitalWrite(led\_der, HIGH);  break;    case 4:  retroceder();  digitalWrite(led\_on , HIGH);  digitalWrite(led\_ade, LOW );  digitalWrite(led\_ret, HIGH);  digitalWrite(led\_izq, LOW );  digitalWrite(led\_der, LOW );  break;    default:  parar();  digitalWrite(led\_on , LOW );  digitalWrite(led\_ade, LOW );  digitalWrite(led\_ret, LOW );  digitalWrite(led\_izq, LOW );  digitalWrite(led\_der, LOW );  break;  }  return;}  void avanzar(void){  PWMA=200;  PWMB=200;  analogWrite(motAPWM, PWMA);  analogWrite(motBPWM, PWMB);  digitalWrite(motA\_P1, LOW );  digitalWrite(motA\_P2, HIGH);  digitalWrite(motB\_P1, LOW );  digitalWrite(motB\_P2, HIGH);  return;}  void retroceder(void){  PWMA=185;  PWMB=185;  analogWrite(motAPWM, PWMA);  analogWrite(motBPWM, PWMB);  digitalWrite(motA\_P1, HIGH);  digitalWrite(motA\_P2, LOW );  digitalWrite(motB\_P1, HIGH);  digitalWrite(motB\_P2, LOW );  return;}  void izquierda(void){  PWMA=185;  PWMB=185;  analogWrite(motAPWM, PWMA);  analogWrite(motBPWM, PWMB);  digitalWrite(motA\_P1, LOW );  digitalWrite(motA\_P2, LOW );  digitalWrite(motB\_P1, LOW );//ACTIVAR MOTOR DERECHO PARA IR A LA IZQUIERDA  digitalWrite(motB\_P2, HIGH );  return;}  void derecha(void){  PWMA=185;  PWMB=185;  analogWrite(motAPWM, PWMA);  analogWrite(motBPWM, PWMB);  digitalWrite(motA\_P1, LOW );//ACTIVAR MOTOR IZQUIERDA PARA IR A LA DERECHO  digitalWrite(motA\_P2, HIGH);  digitalWrite(motB\_P1, LOW );  digitalWrite(motB\_P2, LOW );  return;}  void parar(void){  PWMA=0;  PWMB=0;  analogWrite(motAPWM, PWMA);  analogWrite(motBPWM, PWMB);  digitalWrite(motA\_P1, LOW);//Inicializamos el pin con 0  digitalWrite(motA\_P2, LOW);//Inicializamos el pin con 0  digitalWrite(motB\_P1, LOW);//Inicializamos el pin con 0  digitalWrite(motB\_P2, LOW);//Inicializamos el pin con 0  return;}  void parpadeo (void){  for(char i=0;i<10;i++){  digitalWrite(led\_on, HIGH);  delay(50);  digitalWrite(led\_on, LOW );  delay(50);  }  return;} |