



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

Grau en Enginyeria Mecànica

**DISSENY I MUNTATGE D'UN SISTEMA DE SEGUIMENT
SOLAR PER INSTAL·LACIONS DE BAIXA POTÈNCIA.**



Annex Codi

Autor: Roger Pérez Betoret
Director: Jordi Llumà Fuentes
Director: Manuel Andres Manzanares Brotons
Convocatòria: Setembre 2023

```
//*****LLIBRERIES*****
#include <RTCLib.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>
#include <Keypad.h>

//*****INICIALITZACIO LLIBRERIES*****
LiquidCrystal_I2C lcd(0x27, 16, 2);
RTC_DS3231 rtc;

//*****DEFINICIONS I VARIABLES GENERALS*****

Servo servoh;
Servo servov;
int posh;
int posv;

//*****VARIABLES TECLAT*****
const int files = 4;
const int columnes = 4;

char tecles[files][columnes] = {
  { '1', '2', '3', 'A' },
  { '4', '5', '6', 'B' },
  { '7', '8', '9', 'C' },
  { '*', '0', '#', 'D' }
};

byte entrades_files[files] = { 29, 28, 27, 26 };
byte entrades_columnes[columnes] = { 25, 24, 23, 22 };

Keypad teclat = Keypad(makeKeymap(tecles), entrades_files,
entrades_columnes, files, columnes);

String entrada_text;

//*****VARIABLES CANVI DE MODE*****
int Boto = 13;
int Comptadorboto = 0;
int Botoprevi = 0;
int Botoactual;
int Mode = 0;

//*****VARIABLES MODE MANUAL*****
```

```
int potenh = 4;
int potenv = 5;

//*****VARIABLES MODE RTC*****
int Diesenmesos[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
float Latitud = 0;
float Longitud = 0;
float phi;
float AzimuthCorregit;
float AlturaCorregit;
float Alturavella;
float incTGMT = 2;
//*****VARIABLESMODE LDR*****
int se = 0;
int sd = 1;
int ie = 2;
int id = 3;
int valor_sd = 0;
int valor_se = 0;
int valor_id = 0;
int valor_ie = 0;
int llindar =20;
int increment = 2;

//*****VARIABLES VENT*****

int anemometre = A7;

//*****VARIABLES PES*****

int FSR = A6;

//*****INICIALITZACIO*****

void setup() {

  lcd.init(); //Iniciar LCD
  lcd.backlight();

  servoh.attach(11); //Assignar pins a servos
  servov.attach(10);

  pinMode(Boto, INPUT); //Assignar pin a bóto
```

```

servoh.write(90); //Posicionar servos un posició inicial
servov.write(90);

rtc.begin(); //Iniciar RTC

//rtc.adjust(DateTime(2023,9,14,18,30,0));

DateTime now = rtc.now();
}

void SELECTOR() { //Programa de selecció de mode

    Botoactual = digitalRead(Boto); //Consulta de l'estat del boto de
seleccio,
    if (Botoprevi != Botoactual) { //si canvia d'estat
        if (Botoactual == HIGH) { //augmenta comptador Comptadorboto
            Comptadorboto++;
            lcd.clear();
        }
    }
    Botoprevi = Botoactual; //Guarda el nou estat
    if (Comptadorboto >= 3) { //si el comptador
        Comptadorboto = 0; //fa tota la volta torna a 0
        //entra al mode manual

        LDR();
        //RTC();
        MANUAL();
    }
    if (Comptadorboto == 0) { //Si val 0
        //entra al mode manual

        //RTC();
        //LDR();
        MANUAL();
    }
    if (Comptadorboto == 1) { //Si val 1
        //entra al mode RTC

        //LDR();
        //RTC();
        MANUAL();
    }
    if (Comptadorboto == 2) { //Si val 2
        //entra al mode LDR

        //MANUAL();

```

```

    LDR();
    //RTC();
}

delay(100);
}

void MANUAL() { //Programa de control manual

    posh = analogRead(potenh); //Llegir valor potenciometre horitzontal
    posh = map(posh, 0, 1023, 0, 180); //Relacionar limits analogics i
digital
    servoh.write(180-posh); //Actualitza la posicio

    posv = analogRead(potenv); //Llegir valor potenciometre vertical
    posv = map(posv, 0, 1023, 0, 180); //Relacionar limits analogics i
digital
    servov.write(posv); //Actualitza la posicio
    lcd.clear();
    lcd.print("Mode Manual "); //Al LCD es mostren les posicions
    lcd.setCursor(0, 1);
    if (posv <= 90) {
        lcd.print(posv);
    } else {
        lcd.print(180-posv);
    }
    lcd.print(" // ");
    lcd.print(posh);
    lcd.print(" ");
    delay(10);
}

void RTC() { //Programa de control RTC
    while (Latitud == 0) { //Al iniciar el Seguidor
        lcd.setCursor(0, 0); //Per primer cop
        lcd.print("COORDENADES:"); //Pregunta a traves del LCD
        lcd.setCursor(0, 1); //Les coordenades de
        lcd.print("LATITUD:"); //Longitut i Latitud
        char numero = teclat.getKey(); //Rep el premut al teclat
        if (numero) { //Si es numeric
            if (numero >= '0' && numero <= '9') { //afegeix al string
                entrada_text
                entrada_text += numero;
                lcd.print(entrada_text);
            } else if (numero == 'B') { //si es prem la tecla B
                if (entrada_text.length() > 0) { //i hi ha caracters

```

```

        Latitud = entrada_text.toFloat(); //converteix a valor numeric
        entrada_text = ""; //borra el contingut del string
        lcd.clear();
    }
} else if (numero == 'A') { //si es prem la tecla A
    entrada_text = ""; //esborra l'escrit
    lcd.clear();
} else if (numero == '*') { /* és el separador decimal
    entrada_text += '.';
    lcd.print(entrada_text);
}
}
while (Longitud == 0) { //Al iniciar el seguidor
    lcd.setCursor(0, 0); //Latitud i Longitud
    lcd.print("COORDENADES:"); //Valen 0
    lcd.setCursor(0, 1); //El programa espera a
    lcd.print("LONGITUD:"); //Que l'usuari els defineixi
    char numero = teclat.getKey(); //Mateix procediment per a
    if (numero) { //Longitud
        if (numero >= '0' && numero <= '9') {
            entrada_text += numero;
            lcd.print(entrada_text);
        } else if (numero == 'B') {
            if (entrada_text.length() > 0) {
                Longitud = entrada_text.toFloat();
                entrada_text = "";
                lcd.clear();
            }
        } else if (numero == 'A') {
            entrada_text = "";
            lcd.clear();
        } else if (numero == '*') {
            entrada_text += '.';
            lcd.print(entrada_text);
        }
    }
}
}

if ((Latitud != 0) && (Longitud != 0)) { //Un cop definits el programa
pot
    //avançar passat aquest punt
    phi = Latitud * M_PI / 180.0000; //Convertir l'angle de Latitud
    //a radiants
    DateTime now = rtc.now(); //Obtenir la hora del RTC

```

```

float hora = now.hour();
float minut = now.minute() / 60.0000; //Convertir a format hora
float segon = now.second() / 3600.0000; //Convertir a format hora

float dia = (now.day()); //Obtenir el dia del RTC

float mes = (now.month()); //Obtenir el mes del RTC

int dda = 0; //Convertir el dia
for (int i = 0; i < mes - 1; i++) { //del mes en dia
    dda += Diesenmesos[i]; //de l'any
}
dda += dia;

float B = (360.000 / 365.000) * (dda - 81.000) * (M_PI /
180.000); //Calcular B
float delta = 23.45 * sin(B) * (M_PI /
180); //Calcular delta

float LT = hora + minut + segon; //Calcular LR

float LSTM = 15. * (incTGMT); //Calcular LSTM

float EoT = 9.87 * sin(2 * B) - 7.53 * cos(B) - 1.5 *
sin(B); //Calcular Eot
float TC = 4. * (Longitud - LSTM) +
EoT; //Calcular TC
float LST = LT + (TC /
60); //Calcular LST
float anglehorari = 15 * (LST - 12) * (M_PI /
180); //Calcular angle horari

float Altura = asin(sin(delta) * sin(phi) + cos(delta) * cos(phi) *
cos(anglehorari)) * (180 / M_PI); //Calcular Altrua

float alpha = Altura * (M_PI / 180); //Convertir Altura a radiants

float Azimuth; //Azimut és variable local per corregir
//Diferents calculs segons el moment del
dia
if (Alturavella > Altura) { //Si l'altura es decreixent (Passat migdia
solar)
    Azimuth = 360-(acos((sin(delta) * cos(phi) - cos(delta) * sin(phi) *
cos(anglehorari)) / cos(alpha)) * (180 / M_PI));
} else if (Alturavella < Altura) { //Si es creixent

```

```

    Azimuth = (acos((sin(delta) * cos(phi) - cos(delta) * sin(phi) *
cos(anglehorari)) / cos(alpha)) * (180 / M_PI));
  } else {
    Azimuth = Azimuth;
  }
  Alturavella = Altura;

  if ((Azimuth < 90) || (Azimuth > 359)) {           //Si l'azimut esta
entre els 90 i els 359
    AzimuthCorregit = 0;                             //Al seguidor es
mostrarà 0
  } else if ((Azimuth >= 270) && (Azimuth <= 359)) { //Si esta entre 270
i 259
    AzimuthCorregit = 180;                           //es mostrarà
l'altre limit, 180
  } else {
    AzimuthCorregit = Azimuth; //Ajusta la posició al rang del seguidor
  }

  if (Altura < 0) { //Només es mostren valors d'altura positius
    AlturaCorregit = 0;
  } else {
    AlturaCorregit = Altura;
  }

  posv = AlturaCorregit;
  posh = AzimuthCorregit;

  servoh.write(180-AzimuthCorregit); //Traduir els calculs
servov.write(AlturaCorregit); // en moviment als servos

  lcd.setCursor(0, 0); //Mostrar a
  lcd.print("MODE RTC "); //L'LCD
  if (now.hour() < 10) {
    lcd.print("0");
  }
  lcd.print(now.hour(), DEC); //la hora
  lcd.print(":"); // i la posició
  if (now.minute() < 10) {
    lcd.print("0");
  }
  lcd.print(now.minute(), DEC); // del seguidor
  lcd.setCursor(0, 1);
  lcd.print(AlturaCorregit, 0);
  lcd.print(" // ");
  lcd.print(AzimuthCorregit, 0);

```



```

    delay(100);
  }
}

void LDR() { //Programa control LDR

  valor_sd = analogRead(sd); //Lectura
  valor_se = analogRead(se); //dels 4
  valor_id = analogRead(id); //fotoresistors
  valor_ie = analogRead(ie);

  int msup = (valor_sd + valor_se) / 2; //mitjana
  int minf = (valor_id + valor_ie) / 2; //d'esquerra i dreta
  int mesq = (valor_se + valor_ie) / 2; //i de superior
  int mdre = (valor_sd + valor_id) / 2; // i inferior

  int difelv = msup - minf; //Calcul de la diferència
  int difazi = mesq - mdre; //entre mitjanes

  if (abs(difazi) >= llindar) { //Si la diferencia azimuthal és
    superior al llindar
      if (difazi > 0) { //Si és positiva
        if (servoh.read() < 180) { //I el servidor no ha arribat al
          final de cursa
            posh = servoh.read() + increment; //incrementa la posició
            servoh.write(posh); //Mou el servomotor
          }
        }
      if (difazi < 0) { //Si és negativa
        if (servoh.read() > 0) { //i no ha arribat a l'altre final
          posh = servoh.read() - increment; //disminueix
          servoh.write(posh);
        }
      }
    }

  if (abs(difelv) >= llindar) { //mateix procediment
    if (difelv > 0) { //per l'elevació
      if (servov.read() < 180) {
        posv = servov.read() + increment;
        servov.write(posv);
      }
    }
    if (difelv < 0) {
      if (servov.read() > 0) {

```

```

        posv = servov.read() - increment;
        servov.write(posv);
    }
}

//Al LCD es mostren les posicions
lcd.clear();
lcd.print("Mode LDR      ");
lcd.setCursor(0, 1);
if (posv <= 90) {
    lcd.print(posv);
} else {
    lcd.print(180-posv);
}
lcd.print(" // ");
lcd.print(posh);
lcd.print(" ");
delay(50);
}

void loop() { //Programa repetitiu
    while (analogRead(anemometre) <= 100 && analogRead(FSR) < 500) {

        SELECTOR(); //Si no hi ha vent ni objectes estranys
    } //El programa s'executa a Selector

    if (analogRead(anemometre) >= 100 && Comptadorboto != 1) {
        //Si hi ha vent i no esta en mode manual
        if (posv < 90) { //Porta progrssivament el seguidor
            posv = posv + 1;
        } else if (posv > 90) { //a la posició de bandera mñes propera
            posv = posv - 1;
        }
        servov.write(posv);

        lcd.clear();
        lcd.print("ATENCIO VENT "); //Avis de vent
        lcd.setCursor(0, 1);
        if (posv <= 90) {
            lcd.print(posv);
        } else {
            lcd.print(180-posv);
        }
    }
}

```

```
    lcd.print("    //    ");
    lcd.print(posh);
    lcd.print("    ");
    delay(100);
}

if (analogRead(FSR) > 500 && Comptadorboto != 1) { //Si hi ha un cos
extrany
    if (posv >= 90 && posv < 180) { //Porta
progressivament el seguidor
        posv = posv + 1;
    } else if (posv > 0 && posv <= 90) { //A una posició vertical
        posv = posv - 1;
    }
    servov.write(posv);

    lcd.clear();
    lcd.print("ATENCIO OBJECTE "); //Avis d'objecte
    lcd.setCursor(0, 1);
    if (posv <= 90) {
        lcd.print(posv);
    } else {
        lcd.print(180-posv);
    }
    lcd.print("    //    ");
    lcd.print(posh);
    lcd.print("    ");
    delay(100);
}
}
```