

GRADO EN ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA

TRABAJO FIN DE GRADO

***DISEÑO E IMPLEMENTACIÓN DE
SISTEMA DE REFRIGERACIÓN
MEDIANTE CÉLULA PELTIER***

ANEXO 1 – CÓDIGO FUENTE ATMEGA328P

Alumno: Losa Blanco, Unai

Directora: Otaegi Aizpeolea, Aloña

Curso: 2019-2020

Fecha: 30/10/2019

```
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <FiniteStateMachine.h>
#include <Bounce2.h>
#include <PID_v1.h>
```

```
//HARDWARE
```

```
//Definimos el nombre de los pines de los botones para usarlos con mayor facilidad
en el código
```

```
#define UP_PIN 12
#define DOWN_PIN 8
#define ENTER_PIN 7
#define SSR_PIN 3
```

```
//PID
```

```
double Setpoint, Input, Output;
```

```
PID control(&Input, &Output, &Setpoint,40,5,1, DIRECT);
```

```
//SONDA
```

```
//Variables de coma flotante para el valor de voltaje y temperatura
```

```
#define SONDA_PIN A0
#define SHUNT_PIN A1
float voltaje=0;
float temperatura=0;
float tension=0;
float intensidad=0;
float potencia=0;
```

```
//BOTONES
```

```
//Añadimos antirebote por software a los botones
```

```
Bounce buttonUP = Bounce();
Bounce buttonDOWN = Bounce();
Bounce buttonENTER = Bounce();
```

```
//FSM
```

```
//Maquina de estado finito para la implementación de flujograma del proyecto
```

```
int actualState=0;
// nombre   funcion
```

```
State temp = State(func_temp);
State termo = State(func_termo);
State consum = State(func_consum);
//inicio de la FSM y su estado inicial, para que el menú empiece en el menú de
temperatura siempre
FSM menu = FSM(temp);
```

```
//LCD
LiquidCrystal_I2C lcd (0x27, 16, 2);
//Tiempo para el refresco de la variable temperatura
double lastMillis=0;
```

```
//TERMOSTATO
//Valor de setpoint para el termostato. Hará que el termostato muestre el 6 de
manera inicial
int termostato=6;
```

```
// FSM
void func_temp(){
  lcd.setCursor(3, 0);
  lcd.print("TEMPERATURA");

  if(millis()-lastMillis>500){
    lcd.setCursor(7,1);
    lcd.print(temperatura);
    lastMillis=millis();
  }
}
```

```
void func_termo(){

  lcd.setCursor(3,0);
  lcd.print("TERMOSTATO");
  lcd.setCursor(7,1);
  lcd.print(termostato);
}
```

```
void func_consum(){

  lcd.setCursor(5, 0);
  lcd.print("CONSUMO");
```

```
if(millis()-lastMillis>500){  
  lcd.setCursor(7,1);  
  lcd.print(potencia);  
  lastMillis=millis();  
}  
  
}
```

```
void setup() {
```

```
  //BOTONES
```

```
  pinMode(UP_PIN,INPUT_PULLUP);  
  pinMode(DOWN_PIN,INPUT_PULLUP);  
  pinMode(ENTER_PIN,INPUT_PULLUP);  
  pinMode(SSR_PIN,OUTPUT);
```

```
  buttonUP.attach(UP_PIN);  
  buttonUP.interval(25); // interval in ms  
  buttonDOWN.attach(DOWN_PIN);  
  buttonDOWN.interval(25); // interval in ms  
  buttonENTER.attach(ENTER_PIN);  
  buttonENTER.interval(25); // interval in ms
```

```
  //LCD
```

```
  Wire.begin();  
  lcd.begin(16, 2);  
  lcd.backlight();  
  //PID  
  Input=temperatura;  
  Setpoint=termostato;  
  control.SetMode(AUTOMATIC);  
  control.SetOutputLimits(0, 255);
```

```
}
```

```
void loop(){
```

```
  //CALCULO TEMPERATURA
```

```
  voltaje=((analogRead(SONDA_PIN))*(5/1023.0)-0.030);
```

```
temperatura=((voltaje*30)/0.5);  
temperatura=constrain(temperatura,0,30);
```

```
//CALCULO POTENCIA
```

```
tension=analogRead(SHUNT_PIN)*(5/1023.0);  
intensidad=(tension/0.005);  
potencia=tension*intensidad;
```

```
//PID
```

```
Input=temperatura;  
Setpoint=termostato;  
control.Compute();  
analogWrite(SSR_PIN, Output);
```

```
buttonUP.update();  
buttonDOWN.update();  
buttonENTER.update();
```

```
//Traducimos estado actual a un numero
```

```
//ESTADO MENU
```

```
if(menu.isInState(temp)){  
  actualState=0;  
}else if(menu.isInState(termo)){  
  actualState=1;  
}else if(menu.isInState(consum)){  
  actualState=2;  
}
```

```
//Switch en función del número del estado
```

```
switch (actualState) {  
  case 0:  
    if(buttonENTER.fell()){  
      lcd.clear();  
      menu.transitionTo(termo);  
    }  
    break;  
  case 1:  
    if(buttonUP.fell()){  
      termostato=termostato+1;
```

```
    termostato=constrain(termostato,3,9);
}else if(buttonDOWN.fell()){
    termostato=termostato-1;
    termostato=constrain(termostato,3,9);
}
if(buttonENTER.fell()){
    Setpoint=termostato;
    lcd.clear();
    menu.transitionTo(consum);
}
break;
case 2:
    if(buttonENTER.fell()){
        lcd.clear();
        menu.transitionTo(temp);
    }

    break;
}
menu.update();
}
```