Diseño e implementación del sensor de temperatura de humos para las calderas EK 17, EK 29, EK 45 y Biocalora series 0 y 1 (Ponast).

Sensor de temperatura de gases original.

El sensor de temperatura de humos original de cualquiera de estas calderas consta de una sonda PT1000 y una pequeña placa electrónica contenida dentro de un conector DB25. En mi caso la caldera me indicaba una temperatura de humos de 255 ºC y me instaba a que limpiara la caldera, la electrónica de la sonda se había averiado aunque la sonda PT1000 estaba en perfectas condiciones.

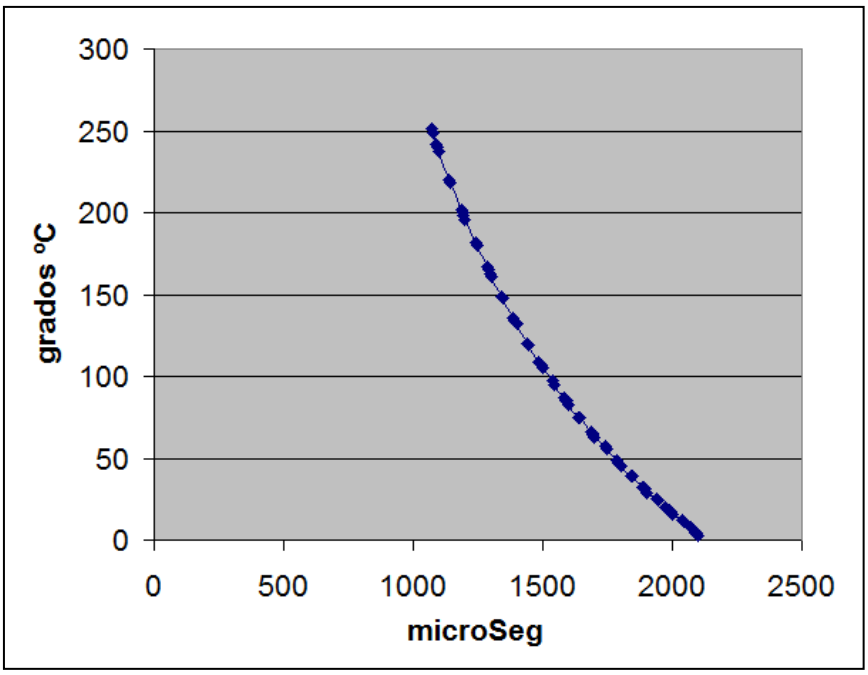
El objetivo del siguiente proyecto es el de montar un sistema electrónico que sirva de adaptador entre la sonda PT1000 y mi caldera EK29(Ponast KP21), la electrónica original (averiada y con un coste de mínimo ¡300€¡), que realiza dicha función nos da una señal cuadrada en función de la temperatura a la que se encuentra la sonda PT1000.

Mediante un generador de onda cuadrada (podemos modificar el periodo a voluntad), acoplado a la entrada de la caldera (donde se acopla el conector DB25 de la sonda) he obtenido los siguientes valores en la pantalla de la caldera:

Periodo microseg	Temperatura en Grados
2100	3
2095	4
2090	4
2085	6
2080	6
2075	7
2070	8
2045	11
2040	12
2000	16
1995	17
1990	18
1985	18
1980	19
1975	20
1970	20
1945	24
1940	25
1900	29
1895	31
1890	32
1885	32
1845	39
1840	39
1800	45
1795	46
1790	47
1785	49
1745	56
1740	57
1700	63
1695	64
1690	65
1685	66

Periodo microseg	Temperatura en Grados
1645	75
1640	75
1600	83
1595	85
1590	85
1585	87
1545	95
1540	97
1500	105
1385	136
1345	148
1340	149
1300	161
1295	163
1290	165
1285	167
1245	180
1240	182
1200	196
1195	198
1190	200
1185	202
1145	218
1140	220
1100	237
1095	240
1090	242
1075	249
1070	251

Con los datos obtenidos si los representamos gráficamente obtenemos el siguiente gráfico:

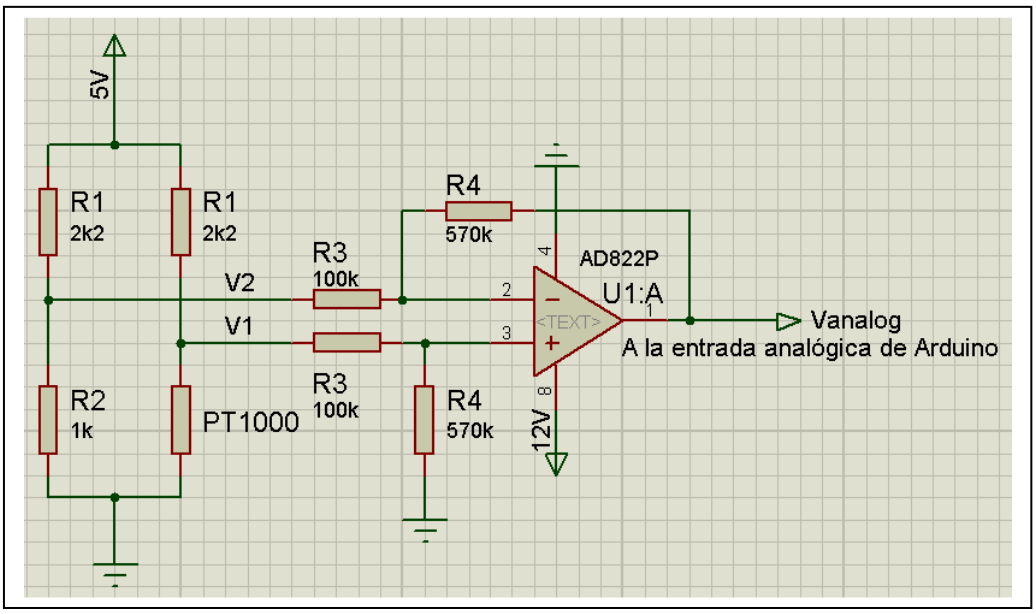


En la imagen anterior observamos que el dispositivo original, si es preciso, debe tener una función que no es lineal, el sistema que diseño debe tener este comportamiento.

Con estas premisas he optado por la siguiente configuración:

1. Un puente de wheatstone, donde ira la PT1000
2. Un amplificador diferencial, montado con un amplificador operacional AD822.
3. Una placa Arduino Uno, a la que se conectará en una entrada analógica, la salida del amplificador y que en una salida digital tendremos la señal que enviaremos a la caldera.

El esquema de los dos primeros bloques sería el siguiente:



Los 5V los sacamos de la patilla correspondiente de la placa de Arduino, los 12V y la tierra o GND los tenemos en el conector DB25 de la caldera, el tercer terminal de este conector es la señal de entrada de la sonda a la caldera.

El valor de la PT1000 en función de la temperatura se obtiene de la siguiente expresión:

$$R = 1000 + 3.9083 \cdot t - 0.0005775 \cdot t^2$$

El valor de Vanalog se halla de la siguiente formula:

$$Vanalog = \frac{400000 \cdot R1 \cdot V \cdot g}{231 \cdot t^2 - 1563320 \cdot t - 400000 \cdot (R1 + 1000)} + \frac{R1 \cdot V \cdot g}{R1 + R2}$$

Donde g es la ganancia del amplificador, $g = \frac{R4}{R3}$

R1, R2,R3 y R4 las resistencias del circuito anterior.

V es el valor real, que debemos medir en la salida 5V de la placa Arduino

Desarrollando la expresión anterior obtenemos:

$$231 \cdot t^2 - 1563320 \cdot t - 400000 \cdot R1 - 400000000 - \frac{400000 \cdot R1 \cdot V \cdot g}{Vanalog - \frac{g \cdot V \cdot R1}{R1 + R2}} = 0$$

De esta ecuación podemos obtener el valor de la temperatura que nos esta midiendo la sonda PT1000.

Una vez conocida la temperatura solo hace falta determinar el tiempo que debe durar el periodo de la onda cuadrada de la señal que debemos aplicar a la caldera.

Con objeto de seguir lo más fielmente la gráfica del principio, he optado por usar la técnica del polinomio interpolador de Lagrange, en concreto con cinco puntos, dos los extremos de la gráfica y los otros tres intermedios, resultando la siguiente formula:

$$p = -4.460445533 \cdot 10^{-8} \cdot t^4 - 6.150560338 \cdot 10^{-6} \cdot t^3 + 0.01850352205 \cdot t^2 - 7.747068439 \cdot t + 2123.074843$$

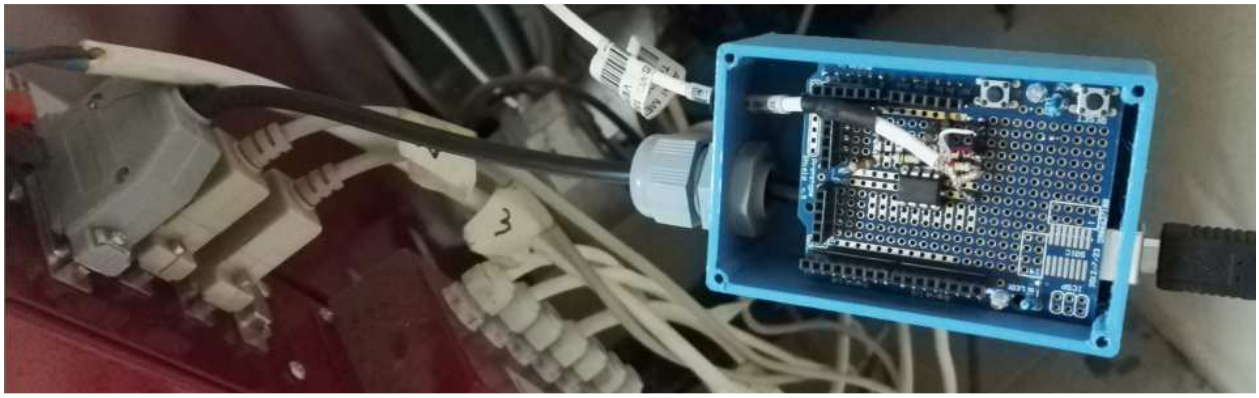
En la que “t” es la temperatura hallada previamente, y “p” el tiempo del periodo que necesitamos.

Todos los cálculos se hacen leyendo el valor de Vanalog, mediante el muestreo de la entrada analógica de la placa Arduino a la que se conecta A0 en mi caso, y a partir de ahí realizando los cálculos matemáticos necesarios. La velocidad y potencia de cálculo de una placa Arduino Uno han sido más que suficientes.

El material necesario es el siguiente:

1. Una placa Arduino Uno (Un clon chino en España cuesta 5€).
2. Un ProtoShield para Arduino Uno, necesario para montar el amplificador, 2€.
3. Amplificador operacional AD822 en encapsulado DIP 6€
4. Conector macho DB25 2€
5. Sonda PT1000 original..
6. Una caja de tamaño suficiente para alojar todo
7. Soldador, estaño, polímetro y un poquito de maña con la electrónica.

Las siguientes imágenes muestran el dispositivo montado, donde puede observarse el ProtoShield donde se monta el amplificador, debajo de este se encuentra la placa Arduino.



A continuación pondré el código de Arduino, pero antes algunas consideraciones.

Una vez que la placa esté programada y conectada a la caldera, es preciso conectar el monitor serie en el IDE Arduino, para interactuar con Arduino la velocidad del mismo debe ser 115200 baudios.

La primera vez que funcione hay que introducir unos cuantos datos, para ello abrimos el monitor serie y cuando tengamos datos en pantalla pulsamos la tecla “p” o “P”

V → valor real de la salida de 5V de la placa Arduino
R1 y R2 →Valor medido con un polímetro, en ohmios.
g → Es el resultado de dividir el valor medido de R4 entre R3, podemos poner hasta 5 decimales.

```
//*****************************************************************
#include <TimerOne.h>
#include <EEPROM.h>
union Float_Byte
{
float datoF;
byte datoB[4];
} unionFB;
union Integer_Byte
{
int datoI;
byte datoB[2];
} unionIB;
int Sonda=2;
int pulsos;
char cadena[30]; //Creamos un array que almacenará los caracteres que escribiremos
//en la consola del PC. Le asignamos un tope de caracteres, en este caso 30
byte posicion=0; //Variable para cambiar la posición de los caracteres del array
float valor,V,Vanalog,c,g;
int R1,R2,periodo;
//unsigned int frecuencia;
//long x,y;
void setup(){
Timer1.initialize(2000000); // Dispara cada 2 s
Timer1.attachInterrupt(Muestrea); // Activa la interrupcion y la asocia a
//Muestrea
Serial.begin(115200);
//Set up the Chip Select pin to be an output from the Arduino.
pinMode(Sonda, OUTPUT);
// V=5;
// R1=2200;
Leer_V();
Leer_R1();
Leer_R2();
Leer_g();
```

```
// Vanalog=4.4834;
Serial.print("Valor de V=");Serial.println((float)V,4);
Serial.print("Valor de R1=");Serial.println(R1);
Serial.print("Valor de R2=");Serial.println(R2);
Serial.print("Valor de g=");Serial.println((float)g,4);
Serial.println("Para modificar parametros pulse P");
}
void loop(){
leer_texto();
if ((cadena[0] == 'P') || (cadena[0] == 'p')) {
cadena[0]=0;
menu();//vamos a cambiar parametros
}
// tone(Sonda,frecuencia);
digitalWrite(Sonda,HIGH);
delayMicroseconds(periodo);
digitalWrite(Sonda,LOW);
delayMicroseconds(periodo);
}
void leer_texto(){
if(Serial.available()) //Nos dice si hay datos dentro del buffer
{
memset(cadena, 0,sizeof(cadena));//memset borra el contenido del array
//"cadena" desde la posición 0 hasta el final sizeof
while(Serial.available()>0) //Mientras haya datos en el buffer ejecuta la función
{
delay(5); //Poner un pequeño delay para mejorar la recepción de datos
cadena[posicion]=Serial.read();//Lee un carácter del string "cadena" de la
//"posicion", luego lee el siguiente carácter con "posicion++"
posicion++;
}
valor=atof(cadena);//Convertimos la cadena de caracteres en enteros
//Serial.println(valor);
posicion=0;//Ponemos la posicion a 0
}
}
void espero_texto(){
while(!Serial.available()) //Esperamos datos dentro del buffer
{}
memset(cadena, 0,sizeof(cadena));//memset borra el contenido del array
//"cadena" desde la posición 0 hasta el final sizeof
while(Serial.available()>0) //Mientras haya datos en el buffer ejecuta la función
{
delay(5); //Poner un pequeño delay para mejorar la recepción de datos
cadena[posicion]=Serial.read();//Lee un carácter del string "cadena" de la
//"posicion", luego lee el siguiente carácter con "posicion++"
posicion++;
}
valor=atof(cadena);//Convertimos la cadena de caracteres en enteros
//Serial.println(valor);
```

```
posicion=0;//Ponemos la posicion a 0
}
void Guarda_V()
{
// guardar un float en la EEPROM en las posiciones 0-3
unionFB.datoF = V;
EEPROM.write(0, unionFB.datoB[0]);
EEPROM.write(1, unionFB.datoB[1]);
EEPROM.write(2, unionFB.datoB[2]);
EEPROM.write(3, unionFB.datoB[3]);
}
void Guarda_R1()
{
// guardar un int en la EEPROM en las posiciones 4-5
unionIB.datoI =R1;
EEPROM.write(4, unionIB.datoB[0]);
EEPROM.write(5, unionIB.datoB[1]);
}
void Guarda_R2()
{
// guardar un int en la EEPROM en las posiciones 4-5
unionIB.datoI =R2;
EEPROM.write(6, unionIB.datoB[0]);
EEPROM.write(7, unionIB.datoB[1]);
}
void Guarda_g()
{
// guardar un float en la EEPROM en las posiciones 0-3
unionFB.datoF = g;
EEPROM.write(8, unionFB.datoB[0]);
EEPROM.write(9, unionFB.datoB[1]);
EEPROM.write(10, unionFB.datoB[2]);
EEPROM.write(11, unionFB.datoB[3]);
}
void Leer_V()
{
// reconstuir el float leyendo la EEPROM
unionFB.datoF = 0.0 ;
unionFB.datoB[0] = EEPROM.read(0);
unionFB.datoB[1] = EEPROM.read(1);
unionFB.datoB[2] = EEPROM.read(2);
unionFB.datoB[3] = EEPROM.read(3);
V=unionFB.datoF;
}
void Leer_R1()
{
// reconstuir el int leyendo la EEPROM
unionIB.datoI = 0 ;
unionIB.datoB[0] = EEPROM.read(4);
unionIB.datoB[1] = EEPROM.read(5);
```

```
R1=unionIB.datoI;
}
void Leer_R2()
{
// reconstuir el int leyendo la EEPROM
unionIB.datoI = 0 ;
unionIB.datoB[0] = EEPROM.read(6);
unionIB.datoB[1] = EEPROM.read(7);
R2=unionIB.datoI;
}
void Leer_g()
{
// reconstuir el float leyendo la EEPROM
unionFB.datoF = 0.0 ;
unionFB.datoB[0] = EEPROM.read(8);
unionFB.datoB[1] = EEPROM.read(9);
unionFB.datoB[2] = EEPROM.read(10);
unionFB.datoB[3] = EEPROM.read(11);
g=unionFB.datoF;
}
void Calcula_Temp()
{
// Vanalog=2.2392;
c=-400000*R1-400000000-400000*R1*V*g/(Vanalog-g*V*R1/(R1+R2));
c=3383.809524-sqrt(11450166.89-c/231);
Serial.print((float)c,7);Serial.println(" grados C");
//valor=2025.666666-4.612612612*c;
//Serial.print((float)valor,7);Serial.println(" microSeg");
valor=-pow(c,4)*4.460445533/100000000;
valor=valor-pow(c,3)*6.150560338/1000000;
valor=valor+pow(c,2)*1850.352205/100000;
valor=valor-c*7747.068439/1000;
valor=valor+2123.074843;
//frecuencia=round(1000000/valor);
//Serial.print(frecuencia);Serial.println(" herzios");
Serial.print((float)valor,7);Serial.println(" microSeg");
//Serial.println();
Serial.println("******");
periodo=round(valor/2);
//Serial.println(periodo);
}
void Muestrea()
{
Vanalog= analogRead(0)*V/1023;
Serial.println("******");
Serial.print(analogRead(0));Serial.println(" muestreo 0-1023");
Serial.print((float)Vanalog,7);Serial.println(" V analogica muestreada");
Calcula_Temp();
}
void menu()
```

```
{
Timer1.detachInterrupt(); // Desactiva la interrupcion
Serial.println("Teclea los parametros, una letra para no modificar");
Serial.print("Valor de V=");Serial.println(V);
espero_texto();
if(valor!=0) //se ha tecleado un valor
{
V=valor;Guarda_V();
}
Serial.println(V);
Serial.print("Valor de R1=");Serial.println(R1);
espero_texto();
if(valor!=0) //se ha tecleado un valor
{
R1=valor;Guarda_R1();
}
Serial.println(R1);
Serial.print("Valor de R2=");Serial.println(R2);
espero_texto();
if(valor!=0) //se ha tecleado un valor
{
R2=valor;Guarda_R2();
}
Serial.println(R2);
Serial.print("Valor de g=");Serial.println(g,5);
espero_texto();
if(valor!=0) //se ha tecleado un valor
{
g=valor;Guarda_g();
}
Serial.println(g);
Serial.println("Fin de introduccion de parametros") ;
Timer1.attachInterrupt(Muestrea); // Activa la interrupcion y la asocia a Muestrea
}
//*****************************************************************
```