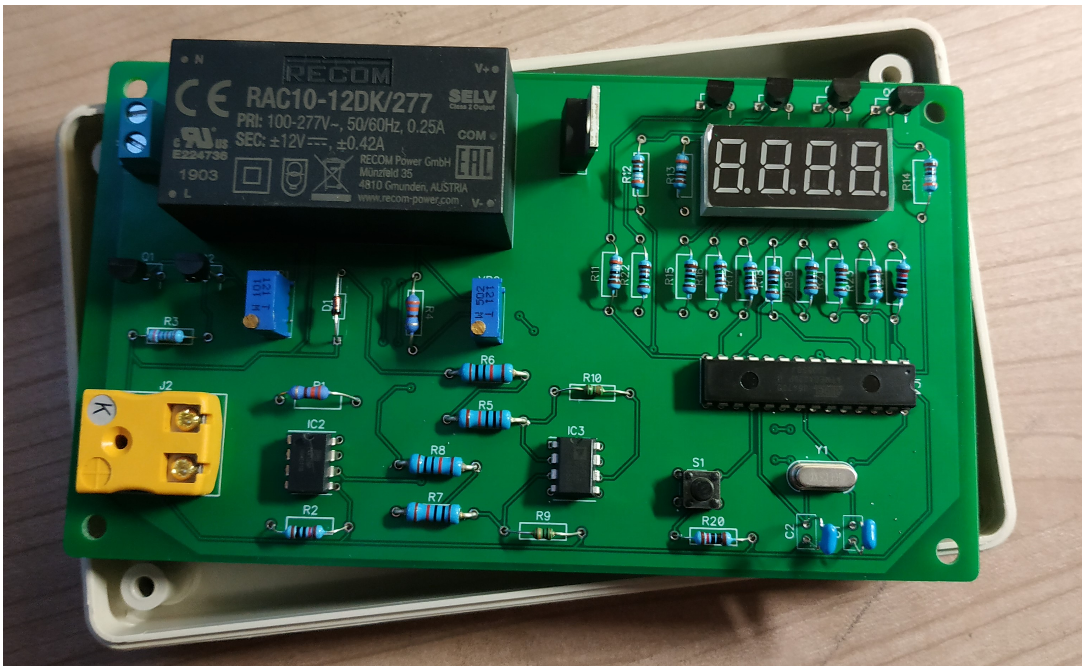
Digital Multi-Functional Thermometer

This thermometer provides instant digital readout of the temperature sensed by a K type thermocouple. It can measure temperatures precisely between 20 and 250 ºC (68 ºF and 482 ºF). This appliance is useful for both beginners and professional cookers to monitor the internal temperature of foods like steaks, chicken, fish and liquids like water or cooking oil.

It can also measure the ambience temperature through a diode, which is also used to make a temperature compensation in the thermocouple.

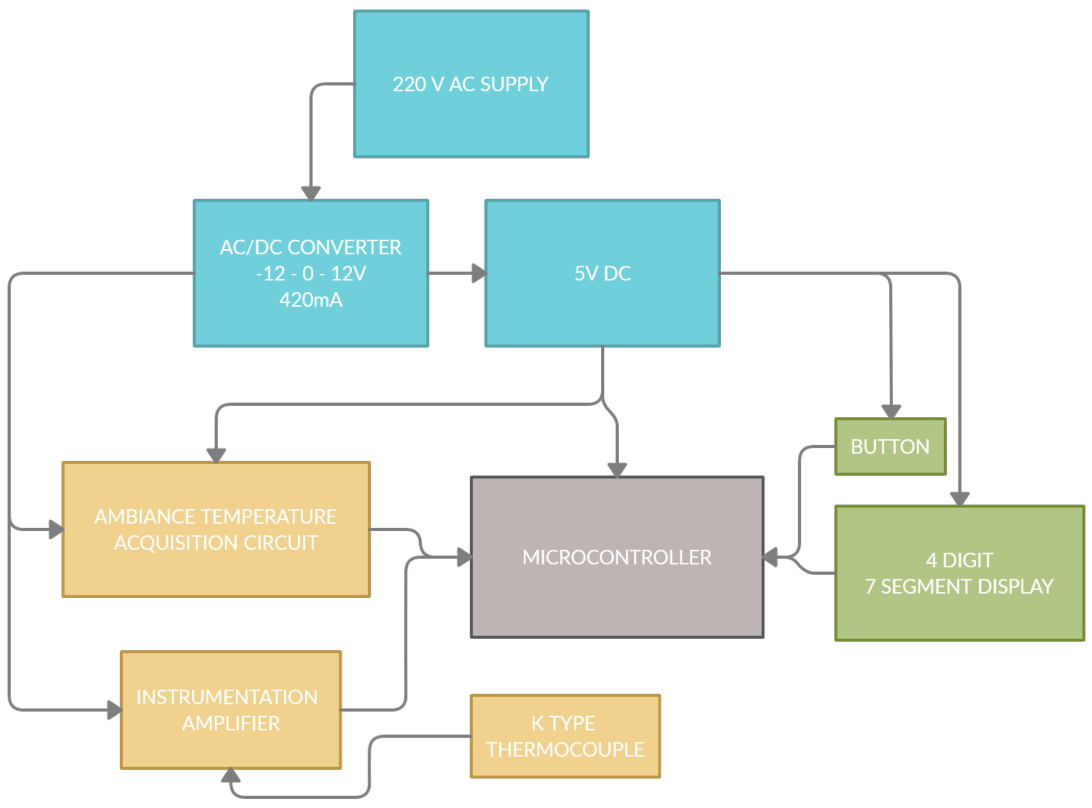
Specifications:

- Power supply: 220V AC
- Temperature display units: ºC
- Resolution: 0.5 ºC
- Temperature range: 0ºC ~ 250ºC
- Temperature accuracy: ± 2 ºC



Index

System block diagram:



220 V AC SUPPLY
AC/DC CONVERTER -12 - 0 - 12V 420mA
5V DC
BUTTON
AMBIANCE TEMPERATURE ACQUISITION CIRCUIT
MICROCONTROLLER
4 DIGIT 7 SEGMENT DISPLAY
INSTRUMENTATION AMPLIFIER
K TYPE THERMOCOUPLE

The voltage that provides the k thermocouple is amplified with a gain of 243 to provide a signal of 10mv per degree to an adc channel in the microcontroller.

The ambience temperature it’s acquired with an 1N4148 diode and his temperature dependency, which is about -2mV/ºC the circuit uses an op-amp in differential mode to amplify this signal with a gain of five and then is sended to an adc channel in the microcontroller.

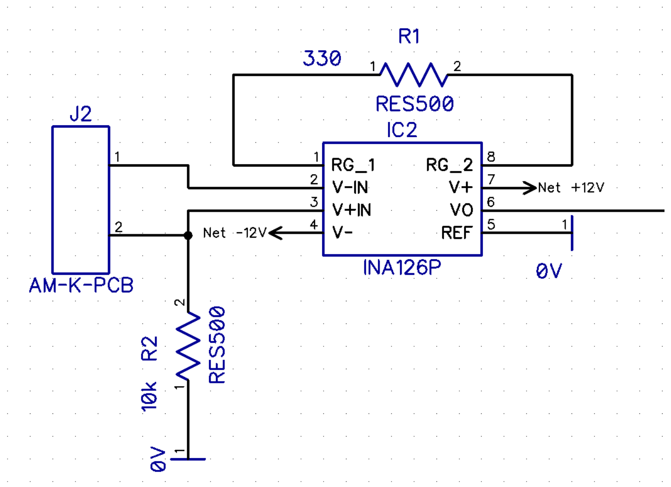
The four-digit seven segment display utilises a circuit to multiplex the displays, 12 digital pins of the microcontroller are used.

The voltage comes from and ac to dc converter which provides 12, -12 and 0 volts along with an LM7805 which provides 5 volts.

The microcontroller is an Atmega328P with an arduino IDE.

Selection of components.

Thermocouple.



The type k thermocouple provides a linear seebeck coefficient in the range between our range of application (0 – 250ºC) of about 41 µV/ºC, therefore.

$$Vin = 41 \cdot 10^{-6}(T - Tamb)$$

Our microcontroller has a resolution of 4.9 mV, so we need to amplify our signal to about 10mV/ºC

$$Vout = 10 \cdot 10^{-3}(T - Tamb)$$

To do that an ina126p instrumentation amplifier is used, it has low quiescent current, low offset voltage, low offset drift, low noise which is ideal for the application.

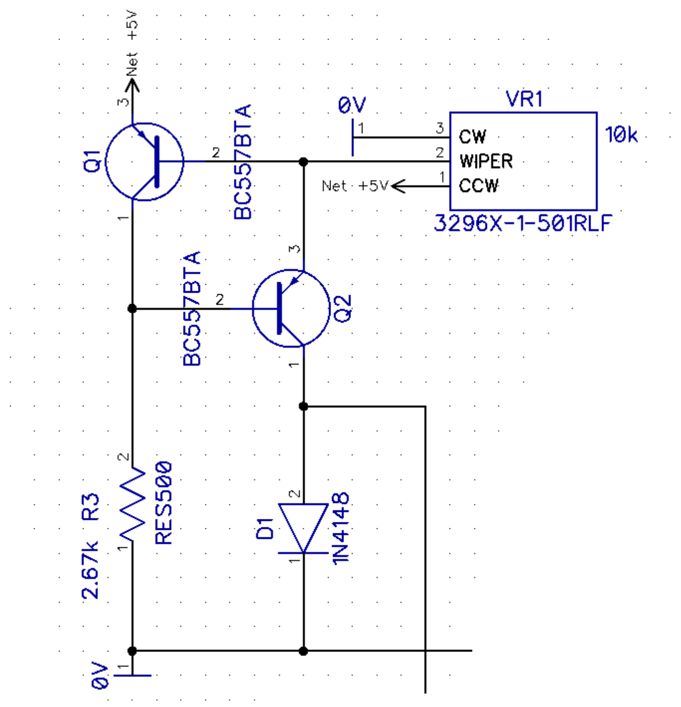
$$G = \frac{10^{-3}}{41 \cdot 10^{-6}} = 243$$

With the gain equation of the ina126p:

$$G = 5 + \frac{80K}{Rg} \rightarrow \boldsymbol{Rg = R1 = \frac{80k}{G - 5} = 330\Omega}$$

R2 = 10K -> To provide a path for the input bias current of both inputs.

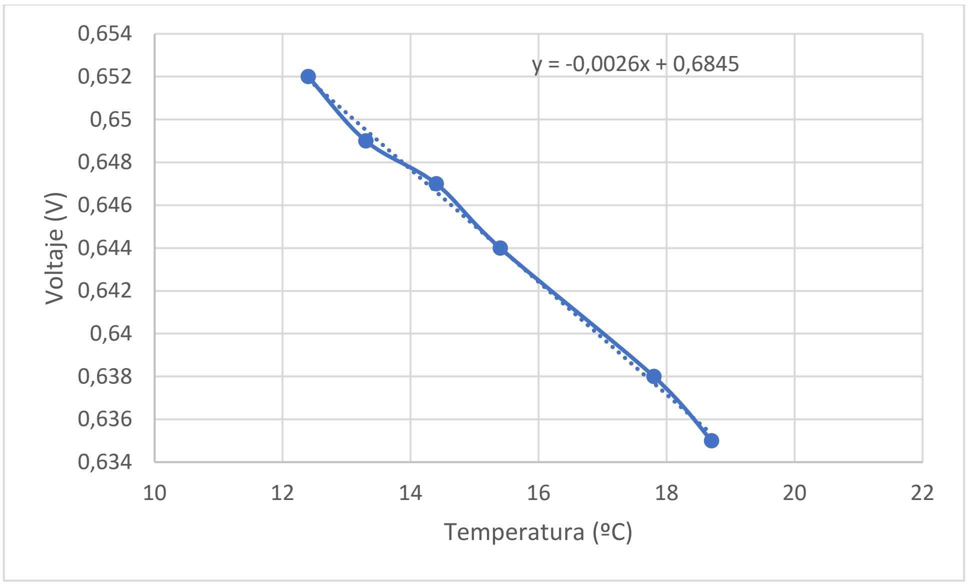
Temperature acquisition circuit.



First, we provide a constant current thru the diode to delete this dependency and obtain the following output. (For the tested diode)

$$Vout = 0.6849 - 0.0026 \cdot T$$

Which was obtained with a linear trendline.



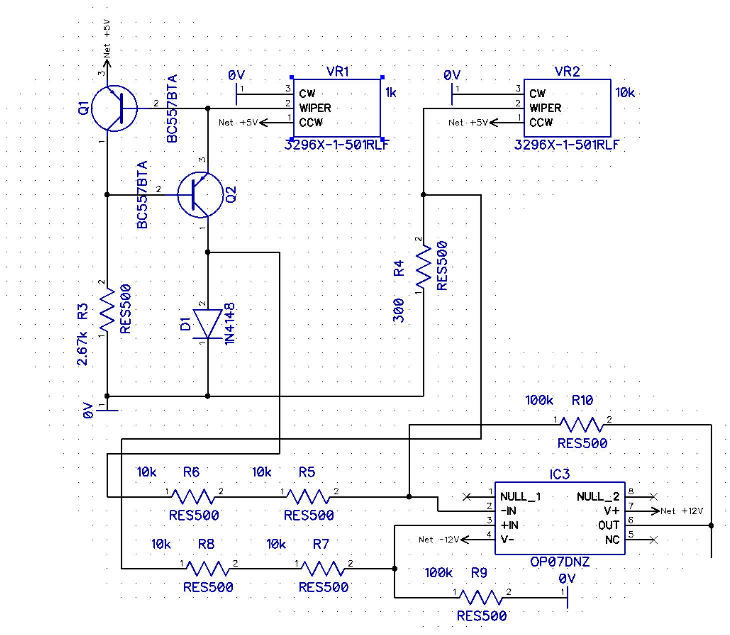
$$Ie,q2 = 2mA \rightarrow R2 = VR1 = \frac{Vbe,q1}{Ie,q2} \cong 350\Omega \rightarrow \boldsymbol{VR1\ potentiometer\ of\ 1K}$$

$$R3 = \frac{Vcc - Vbe,q1 - Vbe,q2}{Ie,q1} \rightarrow R3\min = 1800\Omega, R3\max = 1950\Omega$$

$\boldsymbol{R3 = 2.67K}\Omega$ to provide enough current to saturate transistors.

Temperature amplification circuit

A differential operational circuit will be used in this part, first to get rid of the 0.6849 V constant voltage of the diode and then amplify the temperature dependant voltage to get a 10 mV per degree signal.



First with VR2 and R4 we obtain 0.6849V, if we choose R4 = 330 Ω

$$Vpot = Vin \cdot \frac{330}{330 + Pot} \rightarrow Pot = \frac{330 \cdot Vin}{Vpot} - 330 = 2079\ \Omega$$

A 10K potentiometer is used for VR2

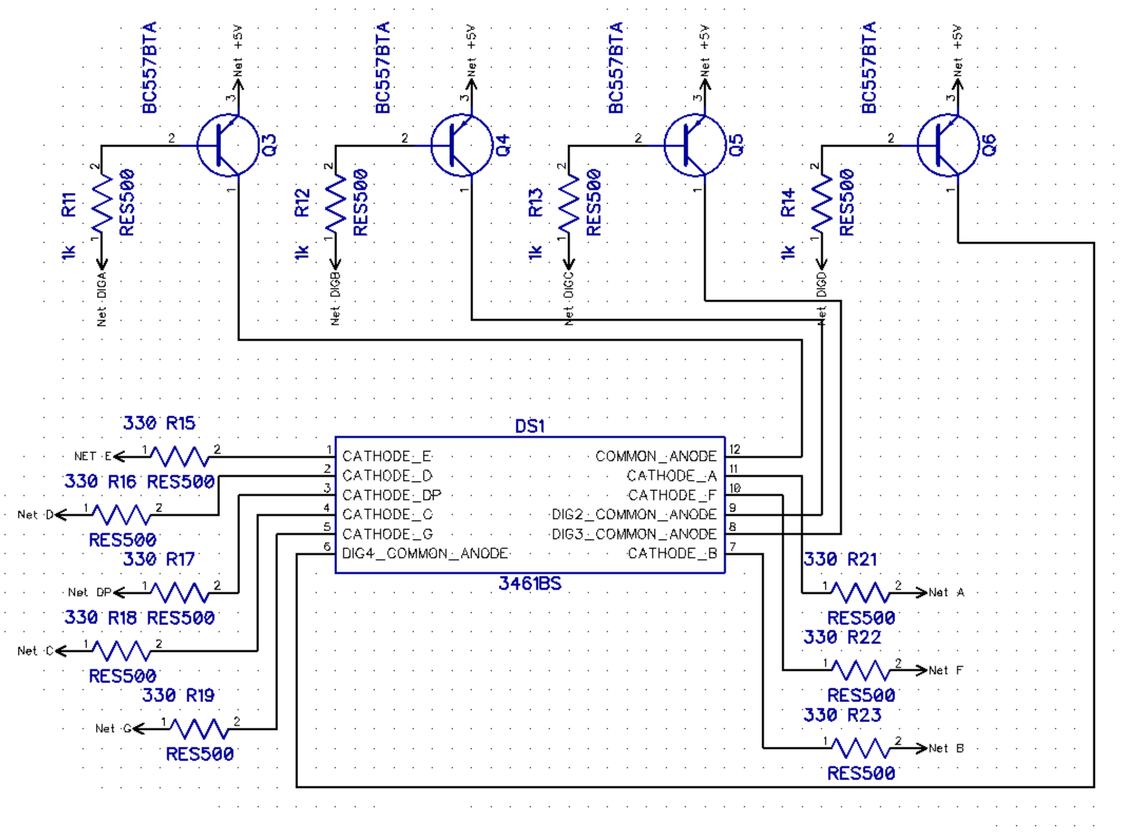
Then an OP07 is used in differential mode to obtain the output signal dependant with the temperature.

$$Vtemp = \frac{R10}{(R6 + R5)} \cdot (Vpot - Vdiode) = 5 \cdot (0.6849 - 0.6849 + 0.0023 \cdot T) = 0.0115 \cdot T$$

R10 = 100k (<5 Tolerance); R5=R6=R8=R7 = 10K

4 Digit 7 segment display driver.

To control which segment will be in it's on state and to supply the same current to each segment (diode) and to reduce the current sink of the microcontroller four transistor's working as switches will be used.



6.5 mA will be enough to light each diode, therefore **R15 to R19 and R21 to R23 = 330 Ω (Vr=Vcc-Vce|sat - Vdiode),** now we need to provide the enough current to the base to saturate the transistors in the worst condition. (All leds on)

$$Ic, \max = 6.5 \cdot 8 = 52\, mA$$

$$Ib > \frac{Ic}{\beta} > 0.52\, mA$$

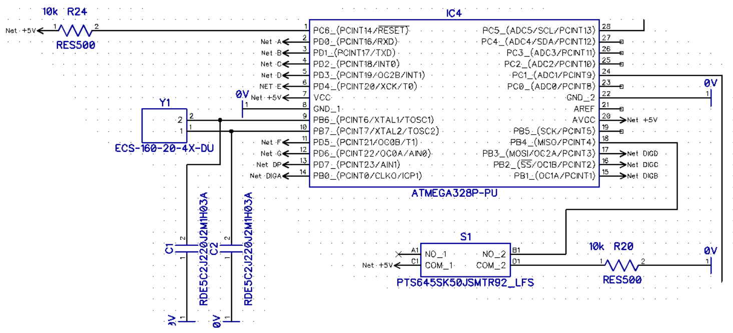
$$Rb > \frac{Vuc - (Vcc - Vbe|sat)}{Ib} > 576\Omega$$

R11 = R12 = R13 = R14 = 1KΩ, to ensure saturation.

When the digital pin out its zero then the transistor is in cut off, Ic = 0

Microcontroller

The microcontroller is an Atmega328P, the operating voltage is 5V. To work with the Arduino IDE a 16Mhz crystal is needed along with two 22pF ceramic capacitors, also a 10K resistor is used to bypass the reset pin.

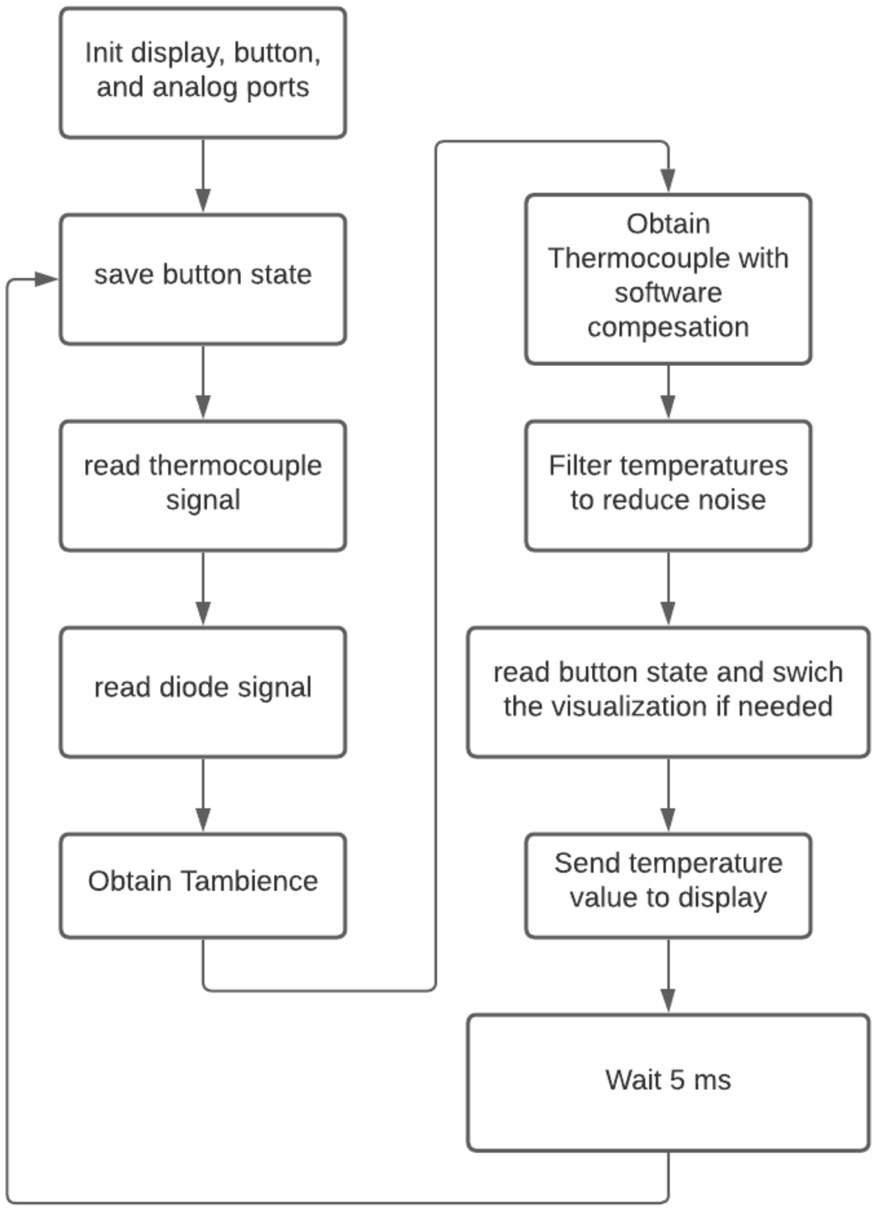


A button (S1) is added so the user can change between the thermocouple temperature and the ambiance temperature.

Software

In this section the code is presented. It contains the display driver, the function to obtain both temperatures (thermocouple, and diode) from de A/D converters and the cold-junction software compensation.

Flow diagram:


Init display, button, and analog ports
save button state
read thermocouple signal
read diode signal
Obtain Tambience
Obtain Thermocouple with software compesation
Filter temperatures to reduce noise
read button state and swich the visualization if needed
Send temperature value to display
Wait 5 ms

Code:

Main:

```
#include "four7Seg.h"

//VARIABLES
float value = 32;
const long interval = 5;
const int  buttonPin = 12;
unsigned long counter = 0;
int a,b,c,d = 0;  //To select values to num7seg
int anDiode = A1;
int anK = A5;
int anValue = 0;
float Temperatureamb = 0, FilterTempamb = 0, SumTempamb = 0;
float Temperaturek = 0, FilterTempk = 0, SumTempk = 0;
float T_Thermocouple = 0;
int n = 0;
int buttonState = 0;         // current state of the button
int lastButtonState = 0;     // previous state of the button
int i = 0;                   // to change between T_k, T_amb;

void setup() {
  initDisplay();
  pinMode(buttonPin, INPUT); //Set digital pin 13 as input
}

void loop() {
  buttonState = digitalRead(buttonPin);
  anValue = analogRead(anK);
  Temperaturek = Ttermocouple(anValue);
  anValue = analogRead(anDiode);
  Temperatureamb = Tambience(anValue);
  SumTempk = SumTempk + Temperaturek;
  SumTempamb = SumTempamb + Temperatureamb;
  T_Thermocouple = FilterTempk + FilterTempamb;
  //Filter temperature over one second.
  if(n >= 200){
    FilterTempk = SumTempk/n;
    FilterTempamb = SumTempamb/n;
    SumTempamb = 0;
    SumTempk = 0;
    n = -1;
  }

  // compare the buttonState
  if (buttonState != lastButtonState) {
    // if the state has changed, increment the counter
    if (buttonState == HIGH) {
       i = i + 1;
    }
  }
  n = n + 1;
  if (i == 1)
  {
    splitnumber(T_Thermocouple, &a, &b, &c, &d);
    multiplexDisplay(T_Thermocouple, a, b, c, 0);
  }
  else
  {
    splitnumber(Temperatureamb, &a, &b, &c, &d);
    multiplexDisplay(Temperatureamb, a, b, c, d);
    i = 0;
  }
  lastButtonState = buttonState;
  delay_milis(interval);
}

float Tambience(int analogValue)
{
  //Obtains the temperature from a 1N4148 diode
  //The equation for the one is used: Vdiode = 5*0.0026*T
  //Aux variables
  float vdiode = 0;
  vdiode = (5 * (float)analogValue)/1024;
  return vdiode*76.923 + 4; //where 4 is the offset
}

float Ttermocouple(int analogValue)
{
  //Obtains the temperature from a 1N4148 diode
  //The equation for the one is used: Vdiode = 5*0.0026*T
  //Aux variables
  float vdiode = 0;
  vdiode = (5 * (float)analogValue)/1024;
  return vdiode*100 + 2.5; //(T-Tamb) offset error
}

void delay_milis(unsigned long period)
{
  counter = millis() + period;
  while(millis() < counter)
  {
  }
}
```

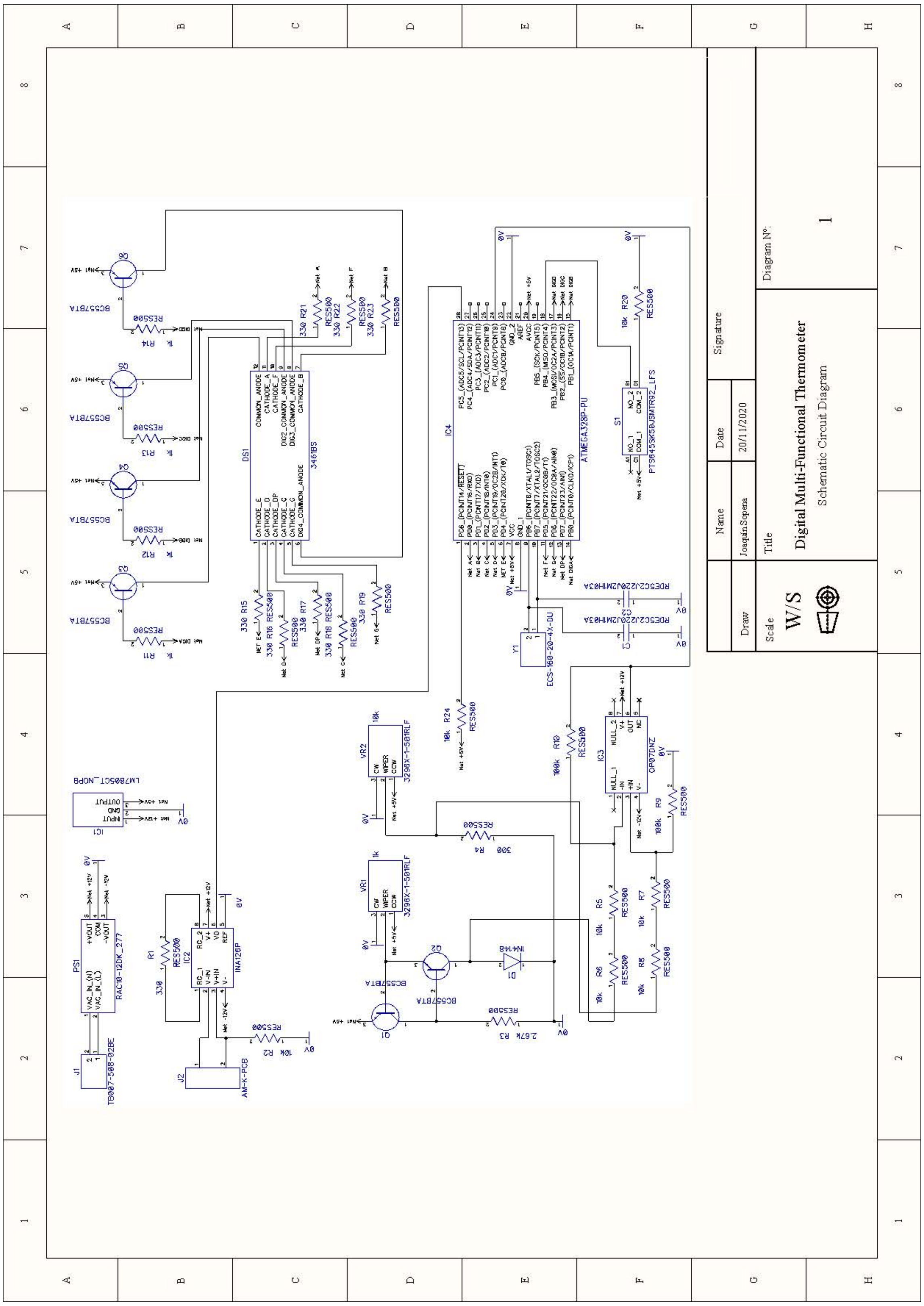
7 segment Driver

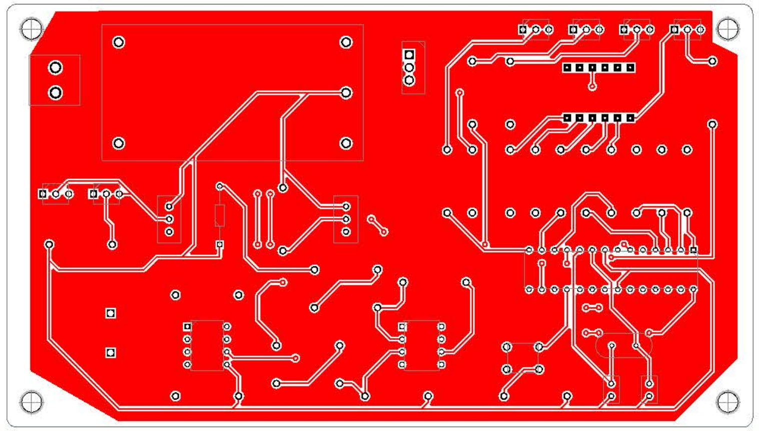
```
/*****************************************************************

/* file name : four7Seg.c

/*****************************************************************

/* description    : Driver for four seven segment displays

/*                common anode and multiplexed.

/* programmer     : Joaquin Sopena

/* lenguage       : ANSI C

/* date           : October 25, 2020

/*****************************************************************/


#include "Arduino.h"


/**************************************************************************
**/

/*                    Local variables                                */

/**************************************************************************
**/


int num7seg[] = { B11000000, B11111001, B10100100, B10110000,

               B10011001, B10010010, B10000010, B11111000,

               B10000000, B10011000};

//Display is connected to pins 0 to 7 of the Arduino, this array contains

//the combinations for numbers 0 to 9.


int count_seg = 4;         //Value to manage multiplexing



/**************************************************************************
**/

/*                   Functions                                   */

/**************************************************************************
**/
```

```
//Converts a decimal number to its equivalent for seven segment displays
//Values from 0 to 999 with one decimal.

void splitnumber(float value, int *h, int *t, int *u, int *th)
{
  int tens, units, tenths;
  int temp = (int)value;
  float tempth;
  if(value < 1000) //If not, error.
  {
    *h = temp / 100;
    *th = (int)((value - temp)*10);
    temp = temp % 100;
    *t = temp / 10;
    *u = temp % 10;
  }
}

void multiplexDisplay(float value, int a, int b, int c, int d)
{
  if(count_seg == 4 & value > 99){
    PORTB = B11111110;
    PORTD = num7seg[a];
  }
  if(count_seg == 3 & value > 9){
    PORTB = B11111101;
    PORTD = num7seg[b];
  }
  if(count_seg == 2){
    PORTB = B11111011;
```

```
        PORTD = num7seg[c] & B01111111;
    }
    if(count_seg == 1){
        PORTB = B11110111;
        PORTD = num7seg[d];
        count_seg = 5;
    }
    count_seg -=1;
}

void initDisplay()
{
    DDRD = B11111111;   //Set Arduino digital pins 0 to 7 as output
    DDRB = B00001111;   //Set Arduino digital pin 8-11 as output
}
```

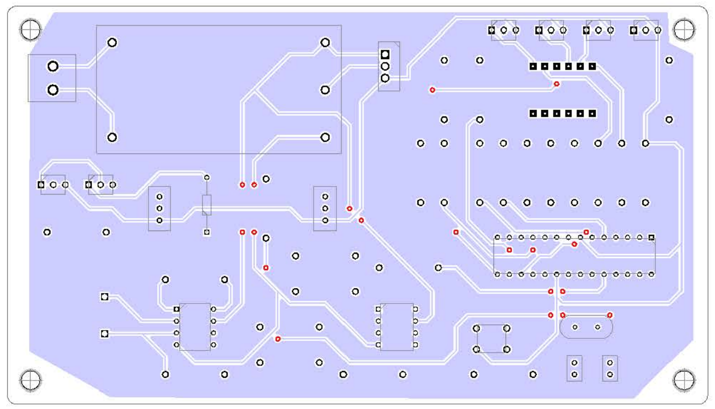
PCB and circuit diagrams:





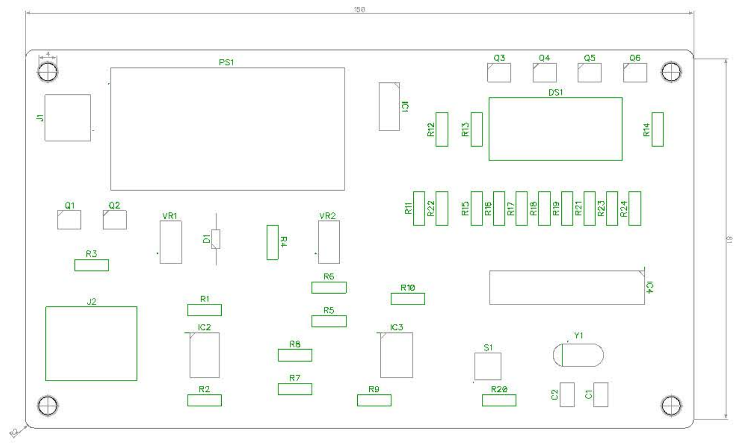
	Name	Date	Signature
Draw	Joaquin Sopena	20/11/2020	

Scale	Title	Diagram Nº:
1/1	Digital Multi-Functional Thermometer PCB - TOP	2



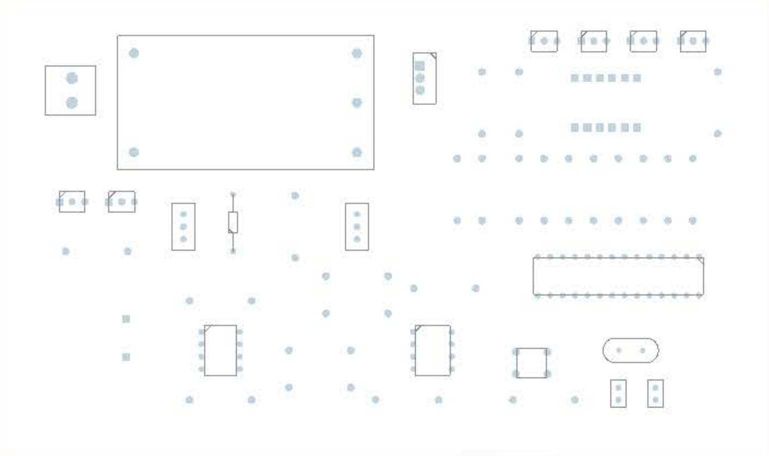
	Name	Date	Signature
Draw	Joaquin Sopena	20/11/2020	

Scale	Title	Diagram Nº:
1/1	Digital Multi-Functional Thermometer PCB - BOTTOM	3



	Name	Date	Signature
Draw	Joaquin Sopena	20/11/2020	

Scale	Title	Diagram Nº:
1/1	Digital Multi-Functional Thermometer PCB - TOP SILK	4



	Name	Date	Signature
Draw	Joaquín Sopena	20/11/2020	

Scale	Title	Diagram Nº
1/1	Digital Multi-Functional Thermometer PCB - TOP MASK	5

Quantity	RefDes	Name	Pattern
1	C1	RDE5C2J220J2M1H03A	RCE5C1H104J2A2H03B
1	C2	RDE5C2J220J2M1H03A	RCE5C1H104J2A2H03B
1	D1	1N4148	DIOAD1165W56L425D185
1	DS1	3461BS	3461BS
1	IC1	LM7805CT_NOPB	TO254P470X1028X1955-3P
1	IC2	INA126P	DIP794W53P254L959H508Q8N
1	IC3	OP07DNZ	DIP794W56P254L959H533Q8N
1	IC4	ATMEGA328P-PU	DIP794W53P254L3467H457Q28N
1	J1	TB007-508-02BE	TB00750802BE
1	J2	AM-K-PCB	Thermocouple PCB Mounting Socket
1	PS1	RAC10-12DK_277	RAC1005SK277
1	Q1	BC557BTA	TO260P419X476X877-3P
1	Q2	BC557BTA	TO260P419X476X877-3P
1	Q3	BC557BTA	TO260P419X476X877-3P
1	Q4	BC557BTA	TO260P419X476X877-3P
1	Q5	BC557BTA	TO260P419X476X877-3P
1	Q6	BC557BTA	TO260P419X476X877-3P
1	R1	RES500	RES-12.7/7.6x2.5
1	R2	RES500	RES-12.7/7.6x2.5
1	R3	RES500	RES-12.7/7.6x2.5
1	R4	RES500	RES-12.7/7.6x2.5
1	R5	RES500	RES-12.7/7.6x2.5
1	R6	RES500	RES-12.7/7.6x2.5
1	R7	RES500	RES-12.7/7.6x2.5
1	R8	RES500	RES-12.7/7.6x2.5
1	R9	RES500	RES-12.7/7.6x2.5
1	R10	RES500	RES-12.7/7.6x2.5
1	R11	RES500	RES-12.7/7.6x2.5
1	R12	RES500	RES-12.7/7.6x2.5
1	R13	RES500	RES-12.7/7.6x2.5
1	R14	RES500	RES-12.7/7.6x2.5
1	R15	RES500	RES-12.7/7.6x2.5
1	R16	RES500	RES-12.7/7.6x2.5
1	R17	RES500	RES-12.7/7.6x2.5
1	R18	RES500	RES-12.7/7.6x2.5
1	R19	RES500	RES-12.7/7.6x2.5
1	R20	RES500	RES-12.7/7.6x2.5
1	R21	RES500	RES-12.7/7.6x2.5
1	R22	RES500	RES-12.7/7.6x2.5
1	R23	RES500	RES-12.7/7.6x2.5
1	R24	RES500	RES-12.7/7.6x2.5
1	S1	PTS645SK50JSMTR92_LFS	PTS645SK50JSMTR92LFS
1	VR1	3296X-1-501RLF	3296X1223LF
1	VR2	3296X-1-501RLF	3296X1223LF
1	Y1	ECS-160-20-4X-DU	HC-49USX

	Name	Date	Signature
Draw	Joaquín Sopena	20/11/2020	

Scale	Title	Diagram Nº:
W/S	**Digital Multi-Functional Thermometer** PCB - Bill of Materials	6