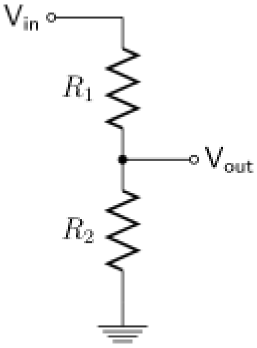
TERMORESISTENZE

PT100 increases its resistance as heat is applied. The temperature vs. resistance characteristic is described in pt100 resistance table



Arduino can read voltage on analog input. To get celsius degree readings we must:

1. read analog input as voltage
2. calculate resistance value (voltage divider)
3. lookup celsius degree from table based on resistance

Vin is 5 volt from arduino R1 is a resistance of known value in my program it is 220 Ohm actually R2 is the pt 100 Vout has to be connected to arduino analog input pin (A0 for instance)

R2 = R1 * 1 / (Vin / Vout - 1)

The circuit can be done based on the picture above it is fairly simple.

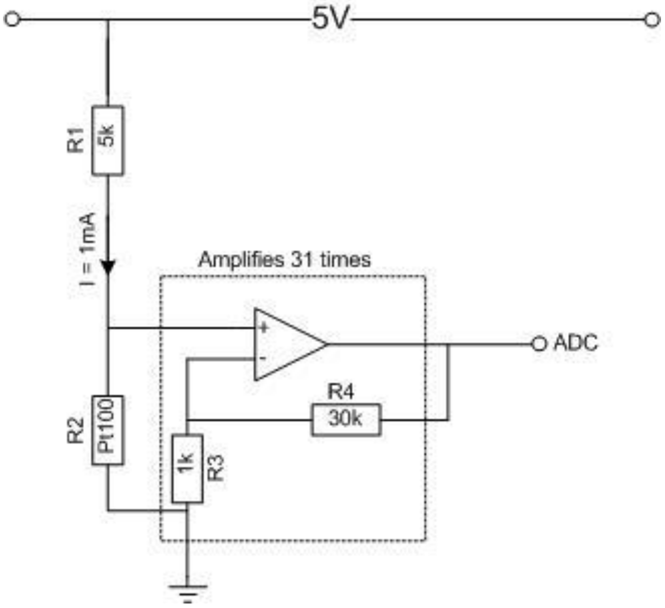
The sketch I wrote contains resistance data from 0C - 80C (can be extended easily) To get the degrees from resistance value I use **my version** of MultiMap function that uses one float array as resistance values and uses linear interpolation to calculate exact degrees

```
float in[] = { 100.00, 100.39, 100.78, 101.17, 101.56, 101.95, 102.34, 102.73, 103.12,
103.51,
               103.90, 104.29, 104.68, 105.07, 105.46, 105.85, 106.24, 106.63, 107.02,
107.40,
               107.79, 108.18, 108.57, 108.96, 109.35, 109.73, 110.12, 110.51, 110.90,
111.29,
               111.67, 112.06, 112.45, 112.83, 113.22, 113.61, 114.00, 114.38, 114.77,
115.15,
               115.54, 115.93, 116.31, 116.70, 117.08, 117.47, 117.86, 118.24, 118.63,
119.01,
               119.40, 119.78, 120.17, 120.55, 120.94, 121.32, 121.71, 122.09, 122.47,
122.86,
               123.24, 123.63, 124.01, 124.39, 124.78, 125.16, 125.54, 125.93, 126.31,
126.69,
               127.08, 127.46, 127.84, 128.22, 128.61, 128.99, 129.37, 129.75, 130.13,
130.52 };

// known resistance in voltage divider
int R1 = 217;

float MultiMap(float val, float* _in, uint8_t size)
{
  // calculate if value is out of range
  if (val < _in[0] ) return -99.99;
  if (val > _in[size-1] ) return 99.99;

  //  search for 'value' in _in array to get the position No.
  uint8_t pos = 0;
  while(val > _in[pos]) pos++;

  // handles the 'rare' equality case
  if (val == _in[pos]) return pos;

  float r1 = _in[pos-1];
  float r2 = _in[pos];
  int c1 = pos-1;
  int c2 = pos;
```

```
 return c1 + (val - r1) / (r2-r1) * (c2-c1);
}

void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);

}
void loop() {
  // put your main code here, to run repeatedly:
   int pt100 = analogRead(A0);


   float Vout = pt100 * (5.0 / 1023.0);
   float R2 = R1 * 1/(5.0/Vout - 1);

float c =  MultiMap(R2,in,80);

Serial.print("Resistance: ");
Serial.print(R2);
Serial.println(" Ohm");

Serial.print("Temperature: ");
Serial.print(c);
Serial.println(" C");


delay(400);
}
```

Come migliorare:

the 220 ohm pullup is too small. There is a noticable current running constantly through the pt100, which can interfere with the precision. A very minimalistic approach is to increase the pullup to reduce this current, and amplify the voltage on the divider


5V
R1
5k
I = 1mA
Amplifies 31 times
+
-
ADC
R4
30k
R2
Pt100
1k
R3