THERMAL CONTROL

October 4, 2020

The general schematic of the temperature control system can be found in the file with the name "schematic". Oh!

We can split it into three main subparts

1. SENSOR
2. COOLER - HEATER
3. TRANSFER of DATA

SENSOR

The main idea for this subpart is to use a voltage divider which, is characterized by the following equation:

$$V_o = V_i \frac{R}{R + R_t.} \tag{1}$$

For this subpart, we are using:

- A 150Ω resistor
- A PtA 100

We know that the PtA is charcterised by the relation:

$$R = R_0 \left(1 + aT\right) \tag{2}$$

where R_0 the temperature at 0 Celcius and $a = 0.0039K^{-1}$ a constant that characterizes the material.

Using these equations we can create the map $V_o \to T$. Thus

$$T = \frac{1}{a}\left(\frac{R}{R_0}\left(\frac{V_i}{V_o} - 1\right) - 1\right) \tag{3}$$

Since V_i, V_o are connected to Arduino Due we use the digital values. Hence, $V_i = 1023$. Also the output reading, limits our precision of Voltage at $\Delta V_o = 1$. Another source of error is the intrinsic variance of the resistance of the temperature at different temperature which can reach the relative error of 1%.

From physical point of view, we can note the variance of voltage powered by Arduino and EM signals which may produced inside the circuit. However, comparing these two sources of noise with the inability of Arduino to detect smaller variances in output voltage, we can safely ignore them.

To show this we can use the transfer of error method in eq(3)

$$\Delta T = -\frac{1}{a}\frac{R}{R_0}\frac{V_i}{V_o^2}\Delta V_0 \tag{4}$$

which for values we get at room temperatures lead to

$$\Delta T \approx 1.K$$

COOLER-HEATER

The main idea for this subsection is to use a Peltier element. The disadvantage of this, is, its power consumption. In our case, we have adapted and used a small one consuming 20 Watts when heating or cooling is needed.

The advantage, on the other hand, is that the same element can be used both for heating and cooling, depending from the current direction. In order to control this system we are using four MOSFETS as depicted (Fig.2), in a ”H” configuration. This allows us to connect the ground of the Peltier element to the ground of Arduino and thus making the conceptual part of the circuit easier to grasp.

In this part we dont care about EM or other fluctuation, because the only purpose is to produce or transfer heat. However, since the Peltier element induces a current of up to 5 Amp, it is a great source of EM waves.

Finally, controlling this part is easy. What we’ve done so far is to merely create a Voltage that is pinned to the gate of two Mosfets. When the voltage is over the threshold, which in our case it is so (3.3 V ¿ 2 V) Mosfets are fully opened and the current flows to Peltier. An idea though is the following; Arduino via a PWD port sends an analog voltage to the gate of the appropriate MOSFET and lets the current flow through it. Here, we should note that Arduino, as an electronically device doesn’t have an actual analog output voltage. The way it manifests to do so is via a rapidly alternating orthogonal pulse. The seemingly analog voltage is adjusted by the working time of this pulse. The reason we make this reference is because this function cannot let us use a capacitor or avoid EM wave production. However using an analog output in this configuraton, let us regulate the current that goes through the MOSFETS and hence adjust the power, Peltier consumes.

TRANSFER of DATA

Under request of the software team, we implemented a transfer function of data to the master Arduino. This is done by a wire connection between the two Arduino’s. Ours sends an analog signal that corresponds to the temperature. It is not the V output, that was read initially, because it would be needed to be translatted again by the master Arduino. In order to create a simple function we multiply the temperature by 10 and pass that as analog voltage. For example, if $T = 25.$ then $V_{pass} = 250$

That means a temperature of 0C can not be passed. But...

EMERGENCY

If temperature is below or above an emergency temperature the system sends an emergency signal and goes to sleep for an hour.

The emergency signal is a simple high voltage, at the wire connecting two Arduino’s.