Incubator eletronics

Kenneth Lausdahl
Your Company / University

The Other Dude
His Company / University

September 8, 2020

Abstract

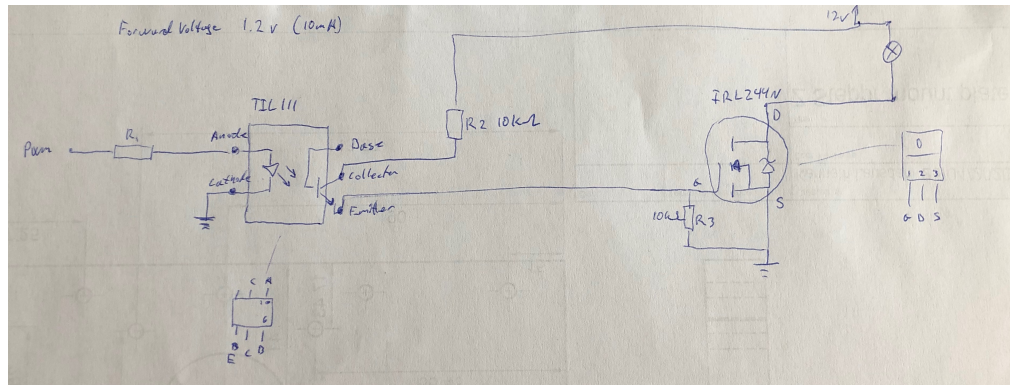Short introduction to subject of the paper ...

1 Design

Figure 1: Overall design

1.1 Calculation of optocoupker resistor

Spec for TIL111

1. Forward Voltage 1.2v (10mA)

$$R = \frac{V}{I}$$
$$R_1 = \frac{3.3v - 1.2v}{0.010A} = 210\Omega \tag{1}$$

This means that the input resistor between the PI pin and the anote of the optocoupler must be at least 210 Ω so 220 Ω is a good candidate

1

1.2 MOSFET

Spec for IRLZ44N (N-channel mostfet:

1. $R_{DS_{ON}} = 0.022\Omega \; V_{GS} = 10v \; I_D = 25A$

2. $V_{GS} = 16V$ max gate voltage

3. 175 °C - max operation temp

4. $V_{CS(th)} = 1v$

5. $R_{BJ}A = 62$ °C/W

Heat disipation assuming a heated bed plate with this spec is powered:

1. $R = 1.65\Omega$ for 12v

2. $P = 87w$

3. $A = 7.25A$

Calculate watts for when powering the heated bed

$$
\begin{aligned}
P &= R * I^2 \\
&= 22mA * 7.25^2 = 1156mW = 1.156W
\end{aligned}
\tag{2}
$$

Calculate watts the MOSFET can handle withtout cooling

$$
\begin{aligned}
P_D &= \frac{max(T_J) - T_A}{R_{\o JA}} \\
&= \frac{175°C - 25°C}{62} = 2.4W
\end{aligned}
\tag{3}
$$

So to use it without a heat sink we need $1.156W < 2.4W$ which it is so no heat sink required.

Calculating resistors between the optocoupker and mosfet. Two resistors are required. One to pull down the gate when not powered here a $10K\Omega$ or similar is fine the smaller the faster it turns off. The other resistor is required to make a voltage devider to protect the gate input voltage of the MOSFET which has a max input voltage of $V_{GS} = 16v$. This means if we power it by 12v then nothing is needed but for other reasons a resistor should be added, bla bla.. So we chose to design it to also allow for 24v and get both handled by the same design.

A Voltage desiver is defined as

$$
V_{out} = \frac{V_s * R_2}{R_1 + R_2}
\tag{4}
$$

So lets find a resistor

$$V_{out} = \frac{12v * 10K\Omega}{10K\Omega + 10K\Omega} = 6v$$
$$= \frac{24v * 10K\Omega}{10K\Omega + 10K\Omega} = 12v \tag{5}$$

So we see that both of these gives a voltage less than V_{gs} i.e. $12v < 16v$ so this resistor valye is fine. The value is also above $V_{CS(th)}$ so it is enough to turn it on

1.3 A first version of the schematics

Figure 2: Schematics

2 Assumptions and Limits

Claudio: Hao, Kenneth, please add the calculations and assumptions here.

2.1 Reading Frequency of the sensors

From the sensor manual, The calculation of the frequency can be calculated as follows: Assume recovery time between read slots is 3 us, and the minimum time of reading a bit is 62us, we need to read 8 bits, so the approximate reading time is 62*8=496 about 500us. or maybe we need to read 16 bits, then the time

ed data. In addition, the master can generate read-time slots after issuing Convert T [44h] or Recall E^2 [B8h] commands to find out the status of the operation as explained in the *DS18S20 Function Commands* section.

All read-time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between slots. A read-time slot is initiated by the master device pulling the 1-Wire bus low for a minimum of 1µs and then releasing the bus (see Figure 13). After the master initiates the

Figure 3: Reading time slot

```
08:08:28: Temp Sensor 1 is: (21.062, 69.9116)
08:08:28: Temp Sensor 1 is: (21.062, 69.9116)
08:08:29: Temp Sensor 1 is: (21.062, 69.9116)
08:08:30: Temp Sensor 1 is: (21.062, 69.9116)
08:08:31: Temp Sensor 1 is: (21.062, 69.9116)
08:08:32: Temp Sensor 1 is: (21.062, 69.9116)
08:08:33: Temp Sensor 1 is: (21.062, 69.9116)
08:08:34: Temp Sensor 1 is: (21.0, 69.8)
08:08:35: Temp Sensor 1 is: (21.062, 69.9116)
08:08:35: Temp Sensor 1 is: (21.0, 69.8)
08:08:36: Temp Sensor 1 is: (21.062, 69.9116)
08:08:37: Temp Sensor 1 is: (21.0, 69.8)
08:08:38: Temp Sensor 1 is: (21.062, 69.9116)
08:08:39: Temp Sensor 1 is: (21.062, 69.9116)
08:08:40: Temp Sensor 1 is: (21.062, 69.9116)
```

Figure 4: Reading time sequence

is approximate 1ms. If nothing is wrong I guess the frequency is larger than 10Hz.

Example code of reading and writing the sensors can be found at: Example code

The python file for reading sensor data is in HOME folder in the Raspberry named read-test.py (Using python3 read-test.py to run the code, the default will only read one sensor, it could be snesor 1, I haven't confirmed). The frequence for reading is approximate 1 Hz which is shown in figure 4

A problem about this is that when reading three sensors' data, the frequency is much slower, I guess it is approximate 0.3Hz. This code read many raw data, the raw data can be seen by figure 5.

I don't know if we reduce the length of the raw data we read then would it be fater?

The sencond way is try to use C code. However we don't know if it worth doing so.

Third way is heating slower i.e. heat for a while and read some data and then heat for a while agian and repeat.

line read: ['2a 00 4b 46 ff ff 0e 10 84 : crc=84 YES\n', '2a 00 4b 46 ff ff 0e 10 84 t=20875\n']
line read: ['29 00 4b 46 ff ff 06 10 37 : crc=37 YES\n', '29 00 4b 46 ff ff 06 10 37 t=20375\n']
line read: ['2a 00 4b 46 ff ff 0e 10 84 : crc=84 YES\n', '2a 00 4b 46 ff ff 0e 10 84 t=20875\n']
Temp Sensor 1 is: (20.875, 69.575)
Temp Sensor 2 is: (20.375, 68.675)
Temp Sensor 3 is: (20.875, 69.575)

Figure 5: Sensor Data

3 Heating Flows

The heat comes from the electrical power which is $U * I$, most of the energy goes into the air to increasing the temprature of the Air. In the mean while the air loses energy in two ways:

1. **Thermal Convection** the major way losing energy which can be calculated by $\Phi = A * h * (T_2 - T_1)$ where A is the inner area of the box, and h is a parameter which can be queried.

2. **Thermal Radiation** this way is less important. This losed energy can be calculated by $Q = \varepsilon * \sigma * T^4$