Custom single-channel RTD input module

Timothy Mutangila

May 2025

ELECTRICAL SCHEMATIC

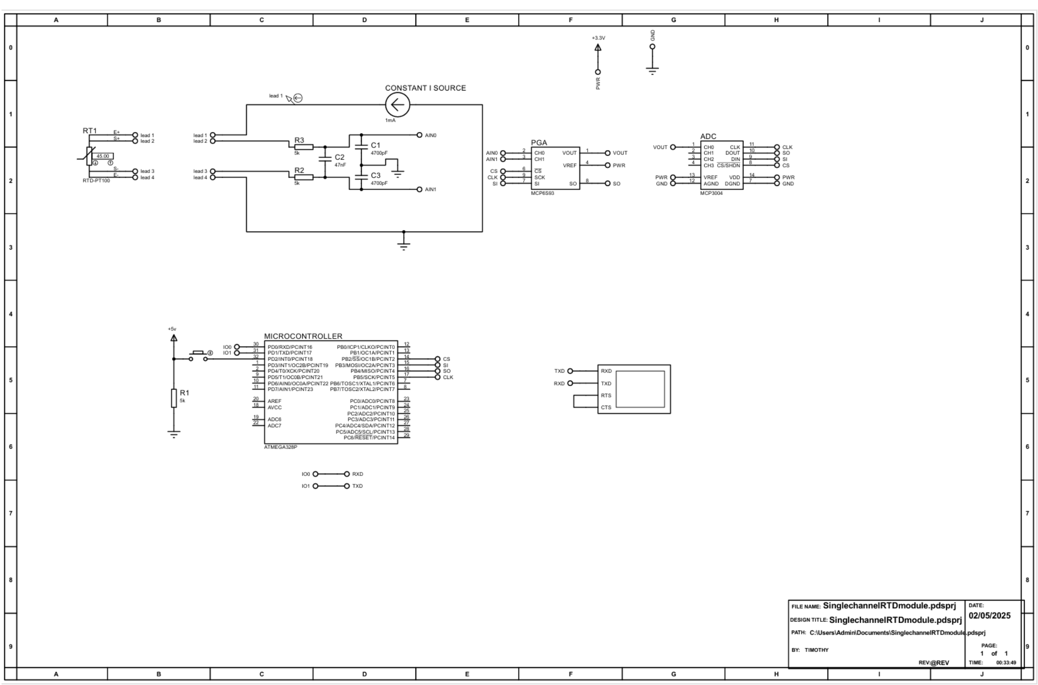

Figure 1: Final RTD Measurement Circuit Supporting PT100 and PT1000

DESIGN DESCRIPTION

The overall design is simple and straightforward. It starts with a port where the RTD lead wires are to be connected. This port has a constant current source that is to continuously excite the RTD that is plugged in and it also has a low-pass RC filter that filters out high frequency noise and switching spikes.
The now filtered voltage signal passes through a Programmable Gain Amplifier(PGA(MCP6S93)) that is connected to the Microcontroller(ATMEGA 328P) over SPI. The microcontroller adjusts the PGA gain according to the RTD type plugged in. The type of RTD plugged in is determined manually by a push button(selector switch) that is connected to a GPIO pin of the microcontroller.
The signal now being fully filtered is passed through the ADC(MCP3004) which is also connected to the microcontroller over SPI. The analog signal is converted to a string of binaries that is equivalent to the voltage.

This voltage being directly proportional to the resistance of the platinum coil is converted to the resistance which is then converted to the temperature being read using the Callendar–Van Dusen polynomial. The code contains a simplified Callendar–Van Dusen temperature conversion calculation as follows: The digital output from the ADC is first converted into voltage using the ADC resolution:

$$V_{\text{RTD}} = \frac{\text{ADC}_{\text{value}}}{2^n} \cdot V_{\text{REF}} \tag{1}$$

where:

- $\text{ADC}_{\text{value}}$ is the raw digital output,
- n is the ADC resolution (e.g., 10 bits for MCP3004, so $n = 10$),
- V_{REF} is the reference voltage.

Given a constant current source I_{EX}, the resistance of the RTD is calculated as:

$$R_{\text{RTD}} = \frac{V_{\text{RTD}}}{I_{\text{EX}}} \tag{2}$$

For PT100 or PT1000 RTDs, the resistance can then be converted into temperature using the simplified linear approximation:

$$T = \frac{R_{\mathrm{RTD}} - R_0}{\alpha \cdot R_0} \tag{3}$$

where:

- R_0 is the nominal resistance at 0°C (e.g., $100\,\Omega$ for PT100, $1000\,\Omega$ for PT1000),
- α is the temperature coefficient of resistance ($0.003\,85\,°\mathrm{C}^{-1}$ as the task required).

JUSTIFICATION

Accommodating Pt100 and Pt1000

To accommodate either the Pt100 or the Pt1000 RTD sensor, the design utilizes a physical selector switch that is manually pressed to change the mode of the input module.
Based on the state of this switch, the microcontroller:
-Adjusts the PGA gain (e.g., higher gain for PT100 due to lower resistance).
-Uses the appropriate scaling equation in software to convert ADC data to temperature for each RTD type.

The current source and front-end analog circuit are generic enough to handle the full resistance range of both sensor types

Noise Mitigation and Self Heating

Electrical noise mitigation is done by the low pass filter while the low constant current source of 1mA dissipates low power which causes negligible heating

Operating temperatures

The design meets the requirement of operating in temperatures of upto 101 degrees celcuis. This is because the ADC(MCP3004) and the PGA(MCP6S93) used both have operating ranges of -40 to 125 degrees celcius.

Digital temperature output over SPI

The system includes:
-A PGA connected to the RTD/filter that amplifies the signal.

-An ADC (MCP3004) that digitizes the analog voltage.

-The MCU reads the digital result via SPI.
-Processed temperature values are then sent to the virtual terminal (UART), and could just as easily be transmitted digitally over SPI to another system.

Electrical and thermal stability up to 101 degrees celcius

Passive components chosen are to be with low temperature coefficients.
By keeping power dissipation minimal using differential measurement, the circuit avoids instability even at elevated ambient conditions