Poznań University of Technology

Faculty of Control, Robotics
and Electrical Engineering

Institute of Robotics and Machine Intelligence

Division of Control and Industrial Electronics

Microprocessor-based control

Microprocessor systems

Laboratory report

Borys Ovsiyenko, 159150
borys.ovsiyenko@student.put.poznan.pl

Pavel Sashko, 159388
pavel.sashko@student.put.poznan.pl

Instructor:
Adrian Wójcik, M.Sc.
Adrian.Wojcik@put.poznan.pl

19-12-2025

Contents

Final task system description

1.1 Purpose and controlled process

The goal of the project is to design and implement a microcontroller-based closed-loop temperature control system for a resistive heating element. The controlled variable is the heater temperature T (process value), while the manipulated variable is the average electrical power delivered to the heater by PWM control of a MOSFET transistor. The system is intended to operate in a predefined safe temperature range and to allow set-point changes from a user interface.

1.2 System architecture

The control system consists of the following functional blocks:

- **Process (plant):** resistive heating element supplied from an external DC source (e.g., 12 V).
- **Actuator:** N-channel logic-level MOSFET used as a low-side switch, controlled by a PWM signal generated by the microcontroller.
- **Sensor:** NTC thermistor mounted close to the heating element. The thermistor is connected as a voltage divider.
- **Adjustable scaling:** a **100 kΩ potentiometer** is included in the divider network to tune the divider midpoint voltage so that the expected temperature range uses a larger portion of the ADC input range.
- **Signal conditioning and buffering:** the divider midpoint is buffered by an **op-amp configured as a unity-gain voltage follower** (voltage buffer) and additionally filtered by an RC low-pass element to reduce high-frequency noise and PWM-related disturbances. The voltage follower provides high input impedance (minimizing loading of the divider) and low output impedance (stable drive for the ADC).
- **Controller:** discrete-time PI (or PID) algorithm implemented in firmware with output saturation.
- **User interface and communication:** serial interface (UART) used to set the reference temperature and to read current values of measurement, reference, and control signals.

1.3 Electrical schematic

Figure 1 presents the electrical schematic prepared in KiCad. The design is divided into three main parts: microcontroller interface, temperature measurement circuit, and power stage.

1 Temperature measurement circuit

The temperature is measured using an NTC thermistor (nominally 10 kΩ at 25°C) connected as a voltage divider powered from 3.3 V. The divider includes a fixed resistor $R_1 = 10$ kΩ and a **100 kΩ potentiometer** (trimmer) used to adjust the effective divider ratio and set the operating point of the measurement node. This allows calibration/tuning so that the expected temperature interval produces a convenient voltage swing at the measurement node.
The divider midpoint (`NTC_ADC`) is connected to an **operational amplifier configured as a voltage follower (unity gain buffer)**. The op-amp output drives the ADC input (`ADC_TEMP`), isolating the divider from ADC sampling effects and preventing measurement errors due to loading. To reduce noise, the measurement node is filtered by a non-polarized capacitor $C_1 = 100$ nF to ground (analog low-pass filtering). A series resistor $R_3 = 1$ kΩ is placed between the buffer output and the ADC pin to improve robustness and limit input currents.
The measured ADC voltage is converted to thermistor resistance and then to temperature using the thermistor model (e.g., Beta equation or Steinhart–Hart approximation).

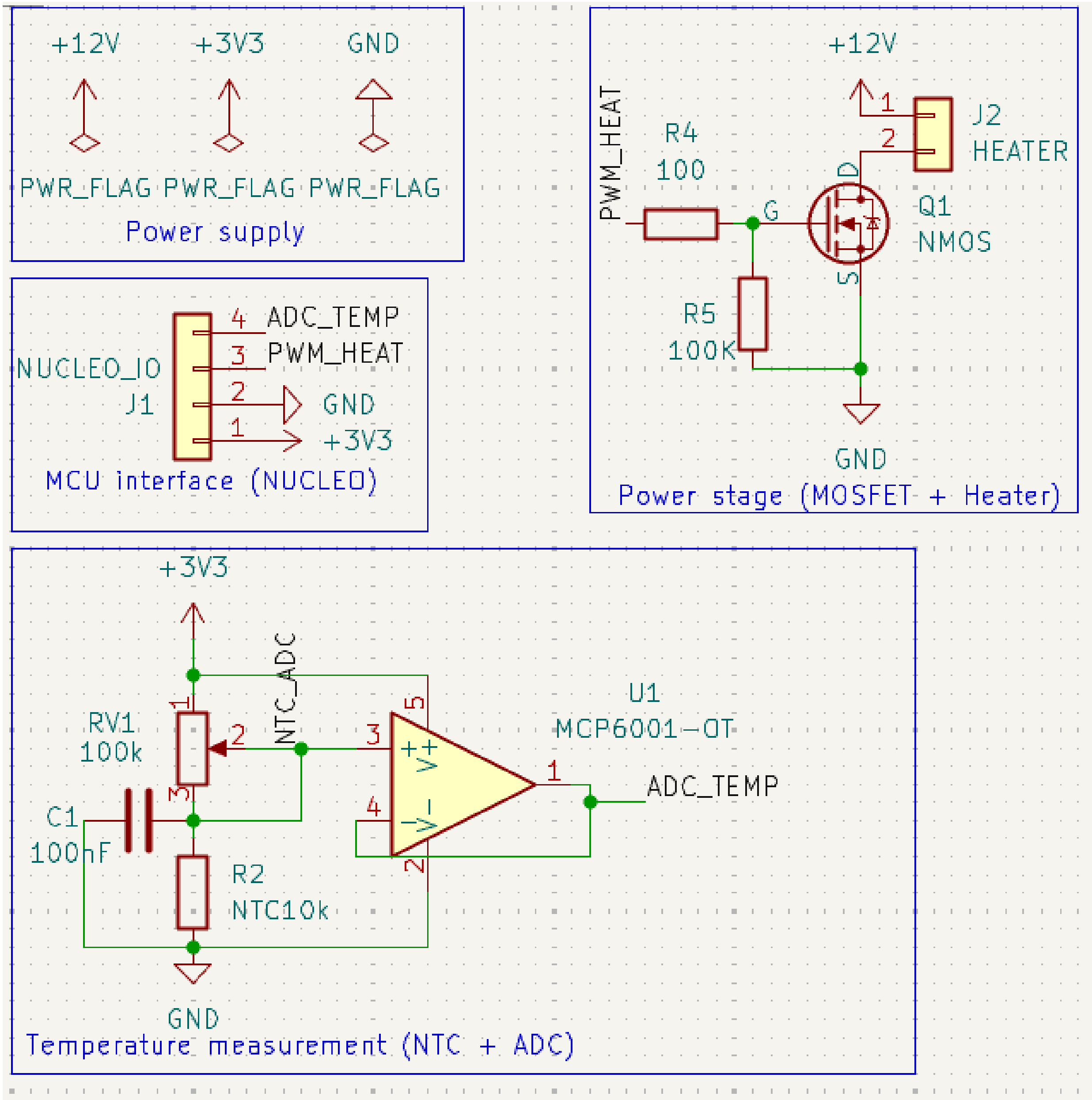

Fig. 1. Electrical schematic of the temperature control system (KiCad).

2 Power stage (heater driver)

The heater is supplied from an external DC source (e.g., +12 V) and switched on the low side by an N-channel MOSFET. The MOSFET gate is driven by the microcontroller PWM output (`PWM_HEAT`) through a gate resistor $R_4 = 100\ \Omega$ to limit switching transients and reduce EMI. A pull-down resistor $R_5 = 100\ \text{k}\Omega$ ensures the MOSFET remains off during reset or when the PWM pin is floating. The grounds of the microcontroller and the power stage are common to provide a correct reference for both ADC measurement and gate drive.

1.4 Firmware operation

The firmware executes a periodic control loop with constant sampling time T_s. Each cycle consists of:

1. Starting ADC conversion and obtaining the current temperature measurement T_{meas}.
2. Computing the control error $e(k) = T_{\text{ref}} - T_{\text{meas}}$.

3. Updating the PI/PID controller and obtaining the control output $u(k)$.
4. Applying output saturation and writing the PWM duty cycle to the timer compare register.

To avoid unsafe operation, the reference temperature T_{ref} is limited to a predefined safe range and the duty cycle is constrained to $0\% \ldots 100\%$.

1.5 Serial interface

The serial interface provides parameterization and monitoring of the embedded control system:

- **Set-point command:** the user sends a command containing the desired temperature reference (e.g., `T035.0` for 35.0°C).
- **Telemetry:** the system returns current values of measurement, reference, and control signals, for example in a JSON-like format: `{"T_meas":34.7,"T_ref":35.0,"PWM":62.1}`.

1.6 Expected performance

The controller parameters are tuned to achieve a small steady-state error across the entire control range. For a PI controller, the integral term eliminates steady-state error for constant disturbances, provided the system is not saturated. The final tuning aims to keep the steady-state error below the required threshold (e.g., 5% of the control range for a passing grade).