Sensor Nodes Laboratory

Milestone Report 2

Communication Interface

Sebastian Thekkekara and Nitish Nagesh, Department of Electrical and Computer Engineering,
Technical University of Munich, Arcisstr. 21, 80333 Munich, Germany

Introduction:

In the previous report, the read-out front-end circuit was discussed. Further, while implementing the hardware, we did encounter a few challenges. We were not completely familiar with the Inter-Integrated-Circuit (I2C) communication protocol. We are also needed more time to understand and apply CC1350 Launchpad by Texas Instruments. Instead of the resistance temperature detector (RTD), a potentiometer was used to vary the resistance and obtain the desired voltage. In this report, we will show the changes made to the readout circuit. We will also discuss the setup of the communication interface between the read-out circuit and communication board.

Read-out circuit:

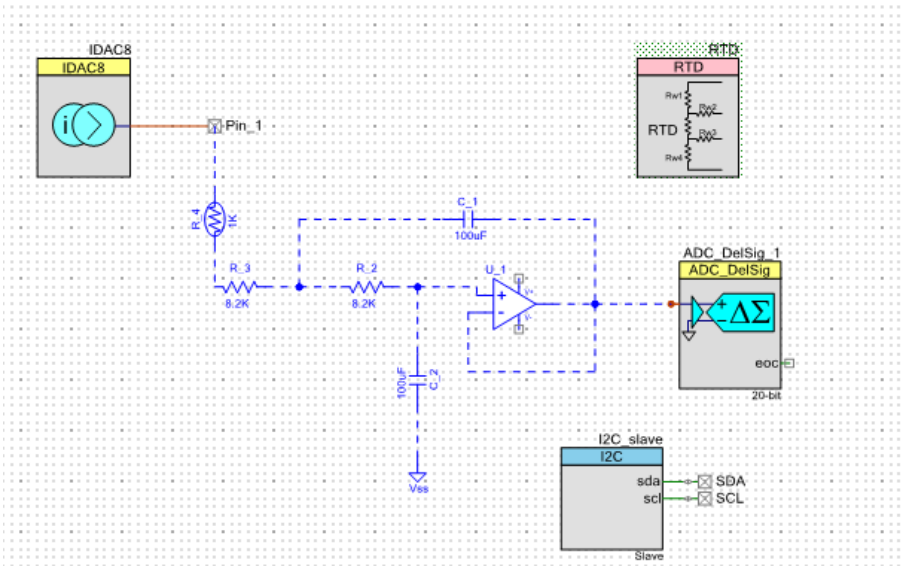

Fig. 1. Redesigned read-out circuit schematic

Fig. 1. shows the read-out circuit which we redesigned. In this iteration, we applied a modular approach to debug the errors and achieve a valid output. We discarded the reference and measured the voltage directly across the Pt1000 resistor. Later we designed a filter to supply a filtered output to the ADC. For this purpose, a low pass filter was used with quality factor Q = 1/2. The temperature data was read every 2 seconds which corresponds to 0.5 Hz. According to the Nyquist criteria, the filter was designed such that its maximum cut-off frequency of the filter is the ADC sampling frequency/2 i.e. 0.25 Hz. To maintain anti-aliasing effect, we chose the cut-off frequency as 0.2 Hz. A second order butterworth low pass filter was implemented using sallen-key topology as it is an active filter topology. Using [1], the two resistance values were chosen as 8.2 kilo-ohm and the two capacitance values were chosen as 100 micro-farad. While testing the filter, one operational amplifier (Op-amp) became extremely hot and caused burns on one of our fingers! On investigating,

it was inferred that soldering of the pins was incorrect. We replaced it with a standard op-amp IC741 and then everything worked smoothly.

Hardware Setup:

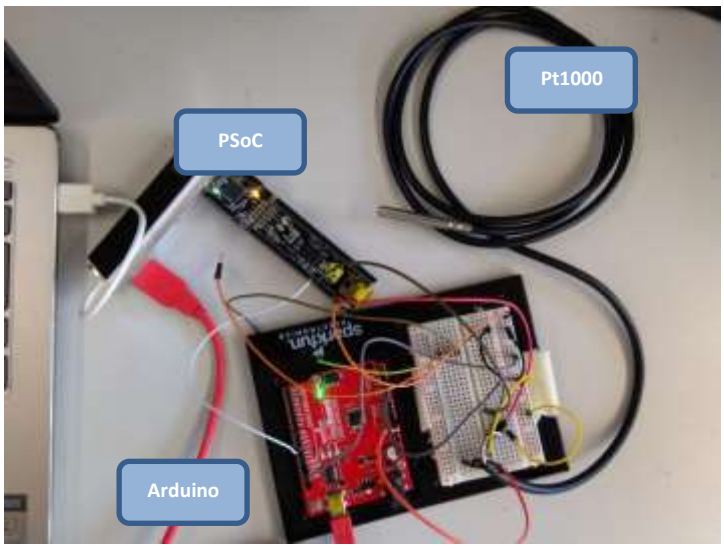

Fig. 2. Communication interface

Fig. 2. shows the preliminary hardware setup for realising a wireless sensor node. The PSoC is connected to the laptop. A 970 ohm resistor was used initially to test the circuit and was connected to the Arduino board containing the Analog-to-Digital Converter (ADC). Later it was replaced with the Pt1000 resistor. Instead of CC1350 Launchpad, an Arduino was used as the microcontroller to read values of current and voltage for the sake of simplicity. The communication between the PSoC and Arduino happens through Universal Asynchronous Receiver Transmitter (UART) protocol. This was chosen again for the sake of simplicity and familiarity. In the final stages, the accuracy and multi-functional capability is envisioned to be improved by using TI Launchpad and I2C communication. I2C allows multiple masters and slaves to be connected without disturbing the communication pathway. This is possible because I2C uses a unique address transmitted before sending the data and also has a shared bus allowing many devices to be connected at a time. As a result, I2C uses lesser number of wires and allows data transmission on a single while enabling lower pin count [2].

Results and Discussion:

In this stage, we successfully completed the filter design and were able to test the voltage output first across the 970 ohm resistor and finally across the Pt1000 resistor. In the next step, the idea is to implement I2C communication as opposed to UART implemented in this stage. Arduino was used to enable easy implementation which will eventually be replaced by CC1350 Launchpad according to the project requirements.

References:

[1] http://sim.okawa-denshi.jp/en/OPstool.php

[2] https://maker.pro/arduino/tutorial/common-communication-peripherals-on-the-arduino-uart-i2c-and-spi