# Digital Temperature Monitor

- Author: Priyansu Sahoo and Saroj Rout
- Description: This projects reads 8-bit temperature data using SPI from a LM70 sensor.
- Language: Verilog

## Objective

This is an educational project for undergraduate engineering students with the objective of exposing them to real-world product design. In the process, the students learn a wide variety of engineering principles including product design, digital system design, mixed-signal modeling, digital design using Verilog, design verification, ASIC design flow, FPGA design flow, and documentation using gitHub.

## Project Brief

This project implements a digital temperature monitor by connecting a temperature sensor (LM70 [`docs/datasheet-LM70-TI-tempSensor.pdf`]) and a three-segment display to measure and display a range of $0 - 99°C$ or $0 - 99°F$ with an accuracy of $\pm 2°C$.

## How it Works

### Mode of Operation

| MODE | `ui_in[0]` | `ui_in[1]` | `ui_in[2]` | DESCRIPTION |
|---|---|---|---|---|
| 1 | 0 | X | 0 | External Display in deg-C |
| 2 | 0 | X | 1 | External Display in deg-F |
| 3 | 1 | 0 | X | MSB Onboard Display in deg-C |
| 4 | 1 | 1 | X | LSB Onboard Display in deg-C |

Figure 1 shows the general block diagram of the complete system. The temperature sensor (Texas Instrument LM70) has a dynamic range of 11 bits with a resolution of $\pm 0.25°C$. In this project, we will only use MSB 8 bits with a resolution of $\pm 2°C$. As shown in the timing diagram in the top right corner of the Figure 1, the LM 70

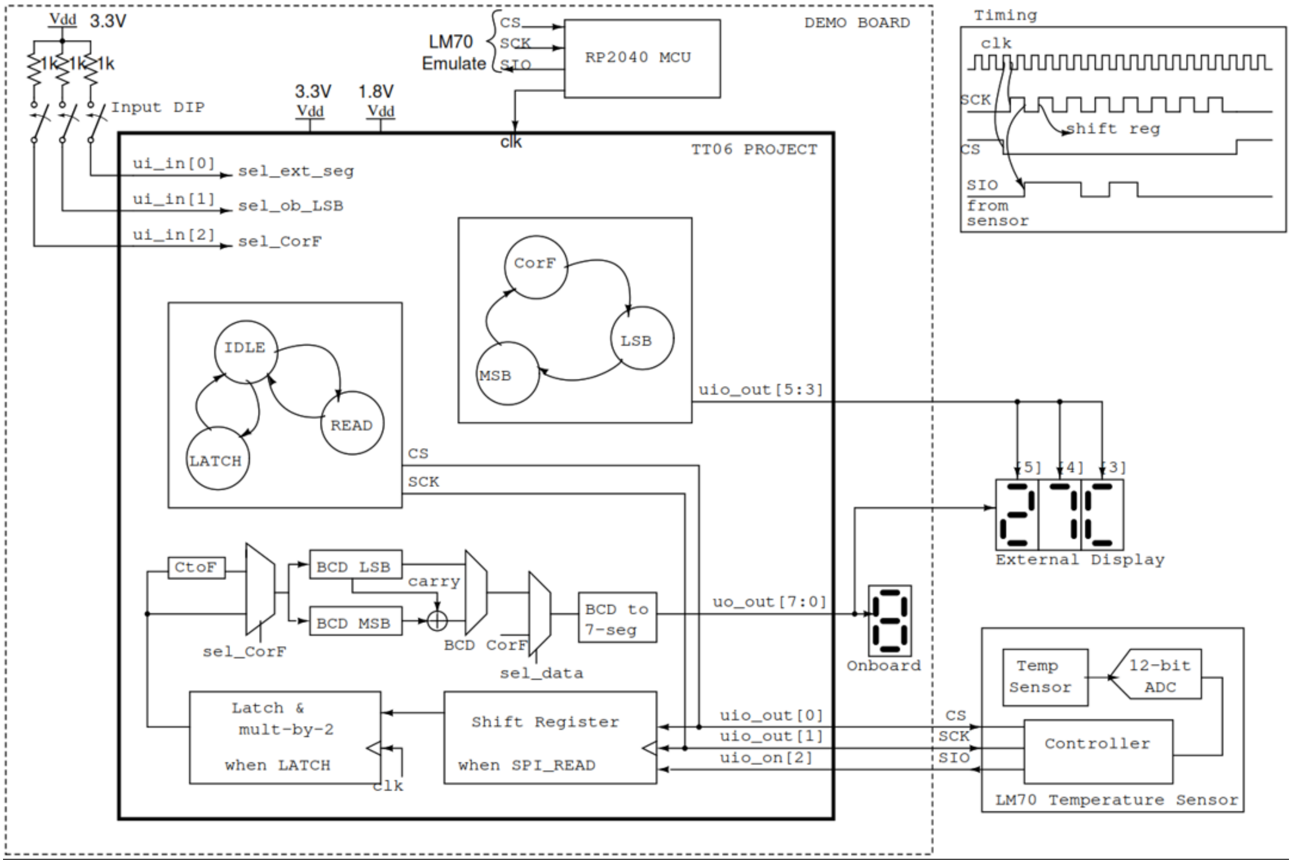

Figure 1: Block diagram of the complete system.

is configured as an SPI *peripheral*, with communication initiated by choosing the chip (CS) low. While CS is low, the data is clocked out of the sensor every *negative edge* of the SPI clock (SCK) and the design reads those data at the following *positive edge*. The design provides eight SCK clock pulses, and then the CS is pulled high to stop the communication.

The serial 8-bit data are captured in a shift register, and the data is *latched* after 8 SCK clock pulses. Before the data are latched, it is multiplied by 2 (left shift by 1). This multiplication captures the fact that the LSB of the data is $2^{\circ}\ C$.

he exact equation to convert temperature in centigrade to Fahrenheit is $T_F = T_C 9/5 + 32$ . To keep the hardware simple, the implemented equation is approximated to $T_F = T_C 2 + 32$. By approximating $9/5$ by $2$, the hardware is simply a left shift by 1. But this approximation results in an error in the output that is a function of temperature: $0.62\%$ error at $0^{\circ}C$ and $9.43\%$ error at $100^{\circ}C$. Based on the input `ui_in[2]`, a MUX selects the temperature in Celsius or Fahrenheit.

The data are then converted to binary coded decimal (BCD) decimal for the two temperature digits to be displayed. The BCD data are then converted to 8-bit 7-segment display format to drive an external display. To save output pins, the 7-segment for all three displays are connected to the ports `uo_out[7:0]` and the displays are *time multiplexed* using the select lines `uio_out[5:3]`. If the displays are switched fast enough (but not too fast), all three displays appear steady without any appearance of flicker.

Since the demo PCB board has one 7-segment display, a provision in the design is made for test purpose where the temperature can be displayed on the onboard display, LSB and MSB one at a time. Kindly refer to the aforementioned 'Mode of Operation' table for further elucidation.

## How to test

This project is designed with testability in mind so it can be tested with barebone PCB without any external hardware. The table below suggests different test modes for testing the design without any external hardware.

| TestNo. | Mode | uio_in[2] | Ext. H/W | RP2040 | 7-seg Ouput |
|---|---|---|---|---|---|
| 1 | 3 | 0 | None | clk~10kHz | 0 |
| 2 | 4 | 0 | None | clk~10kHz | 0 |
| 3 | 3 | `SIO` from RP2040 | None | clk~10kHz and SIO | MSB of data sent by RI |
| 4 | 4 | `SIO` from RP2040 | None | clk~10kHz and SIO | LSB of data sent by RP |

For the first two tests, the uio_in[2] port is grounded and a clock frequency of approximately 10 kHz is provided to the design from the RP2040 as shown in Figure 1. And when the inputs (ui_in[2:0]) are configure in Mode 3 or 4, the single 7-segment display should display 0 in both modes.

Test 3 and 4 in the table above will use the RP2040 as a SPI peripheral and micro-python code will be written to emulate the temperature sensor LM70. This will allow us to test the entire design without connecting the external temperature sensor or display.

## External hardware

Needs a LM07 interfaced on the PCB. Detail hardware plan will be updated when we get close to receiveing the PCB.

## Pinout

| # | Input | Output | Bidirectional |
|---|---|---|---|
| 0 | Ext/Int | 7seg-A | CS (O) |
| 1 | Int-LSB | 7seg-B | SCK (O) |
| 2 | Ext-CorF | 7seg-C | SIO (I) |
| 3 | | 7seg-D | 7seg-sel0 (O) |
| 4 | | 7seg-E | 7seg-sel1 (O) |
| 5 | | 7seg-F | 7seg-sel2 (O) |
| 6 | | 7seg-G | |
| 7 | | 7seg-DP | |