PIC18F452 Oven Calibrator Using the MAX6674/MAX6675 Thermocouple-to-Digital Converter

Group: SMOKE
Duston Cline
Adam McCorkle

April 19, 2002

1 Project Requirements

This project required that the PIC microcontroller communicate with both the Maxim 6674 and 6675 thermocouple-to-digital converters. The project also required that the temperature measured by the thermocouple be displayed in degrees Fahrenheit and degrees Celsius. Another requirement was that the temperature measurements could be calibrated using one-point or two-point calibrations. The displaying of user-defined characters also had to be incorporated into the design. The chip being used, the unit of temperature, and the calibration data all had to be stored in EEPROM.

2 Thermocouple-to-Digital Converter Characteristics

Both the Maxim 6674 and 6675 convert the voltage of the thermocouple into a digital format. This digital format is transmitted from the chip via a SPI-compatible serial interface. For both chips, the data is sent in two bytes. The actual temperature bits have to be extracted from the two bytes of data. With the Chip Select (pin 6) forced low, the data is outputted to pin 7 (SO) on the falling edge of the clock input on pin 5(SCK). The data is read into the microcontroller on the SDI pin. The connections between the microcontroller and the either of the Maxim chips are shown in the following schematic.

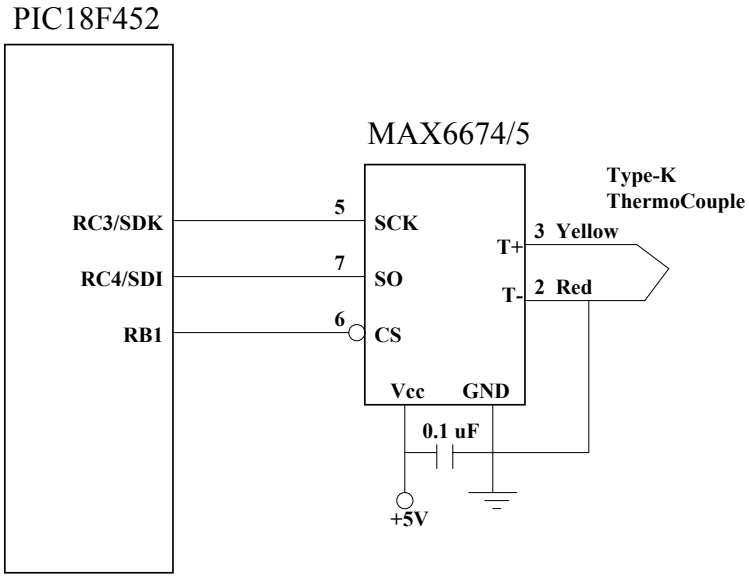

Figure 1. PIC Calibrator Schematic.

The Maxim 6674 outputs the measured temperature with a 0.125°C resolution. This is accomplished with a 10-bit binary number that ranges from 0 to 1023. This number can be converted to get a temperature reading ranging from 0°C to 127.875°C.

The Maxim 6675 outputs the measured temperature with a 0.25°C resolution. This is accomplished with a 12-bit binary number that ranges from 0 to 4095. This number can be converted to get a temperature reading ranging from 0°C to 1023.75°C.

3 Program Development

The SPI data is read on the falling edge of the clock output of the microcontroller. The phasing of the communication to the two MAXIM chips is configured with three control bits. In the SSPSTAT register, the CKE and SMP bits are initialized to zero. The CKP bit in the SSPCON1 register is also set to zero.

The program uses a byte defined as CALIRBATE during execution. This byte consists of eight bits that are used as Boolean flags for conditional branches. The bits of the CALIBRATE byte are defined in the table below.

Table I. CALIBRATE Register Flag Definitions.

Bit	Functionality
0	0 = Calibrate, 1 = No Calibration
1	0 = 1 Point Calibration, 1 = 2 Point Calibration
2	1= Calibrate Freeze
3	1= Calibrate Boil
4	1 = Calibration In Progress
5	1 = Temperature Conversion Done
6	0 = Fahrenheit, 1 = Celsius
7	0 = MAX6674, 1 = MAX6675

After a one-point calibration, the displayed temperature is calculated by adding an offset value to the measured temperature. The offset value is store in the CALIBRATIONOFFSET register in the program. The following equation displays the formation of the displayed temperature.

$$TEMP_{displayed} = TEMP_{measured} - OFFSET$$

After a two-point calibration, the displayed temperature is calculated by adding an offset value to the measured temperature, multiplying this value by 100, and dividing by the measured temperature range. This temperature range is defined as the measured boiling point minus the measured freezing point. This range is stored in the

CALIBRATIONRANGE register in the program. The following equation displays the formation of the displayed temperature.

$$TEMP_{displayed} = \left(TEMP_{measured} - OFFSET\right) \times \frac{100}{TEMP_{range}}$$

Four bytes of data are stored in the EEPROM Memory. The first byte (stored at 0x00) is a one-time write byte that holds a unique identifier to distinguish if the program has been ran before. If the program has been run before, the three following bytes can be retrieved from the EEPROM memory. Otherwise the default values for the program are used.

The other three bytes stored in EEPROM are the CALIBRATE, CALIBRATIONOFFSET, and CALIBRATIONRANGE bytes. Whenever any of these three bytes change during program execution, the new value is stored in the EEPROM memory.

4 Theory of Operation

Below is the listing of each of the menus displayed and the functionality presented with each display.

TEMP
123°C (°F)

This is the menu that appears upon initial power up. The temperature is displayed in either degrees Celsius or degrees Fahrenheit. A short press of the switch toggles the temperature unit, and a long press advances to the next menu.

CHIP SEL
6674(6675)

This menu displays the current Maxim chip being used. A short press of the switch toggles the chip selection, and a long press advances to the next menu.

CALIBRAT
YES(NO)

This menu allows the user to calibrate the temperature readings. A short press of the switch toggles the calibration option, and a long press advances to the next menu. If the user chooses not to calibrate, the program returns to the initial menu of the application.

CALIBRAT
1 POINT (2 POINT)

This menu displays the type of calibration to perform. A short press of the switch toggles between a one-point and a two-point calibration, and a long press advances to the next menu.

CALIBRAT
FREZ BGN (---)(END)

This menu displays the status of the freezing point calibration. A short press of the switch begins the calibration. If a long press of the switch is initiated during the calibration, the calibration is aborted and the calibration constants are reset to their default values. Otherwise, a long press of the switch when the status is displayed as "END" advances to the next menu.

CALIBRAT
BOIL BGN (---)(END)

This menu displays the status of the boiling point calibration. A short press of the switch begins the calibration. If a long press of the switch is initiated during the calibration, the calibration is aborted and the calibration constants are reset to their default values. Otherwise, a long press of the switch when the status is displayed as "END" advances to the next menu.

5 Lessons Learned

One characteristic of the thermocouple that was discovered during initial testing was the fact that the device has polarity. If the thermocouple is connected to the circuit with the leads reversed, a rising temperature will be interpreted as a falling temperature and vice-versa. This effect did not hinder the development of this project, however, future groups may want to make note of this characteristic.

During the development of this project, a problem arose such that the temperature measurement was never updating. Viewing the data coming from the MAX6674, it was seen that the converter was never updating the data being sent to the microcontroller. After investigating the problem, it was found that the converter needed between 150 to 180 milliseconds to convert the thermocouple voltage. The microcontroller program was polling the converter at a rate much faster than this, therefore only one conversion was occurring and the temperature measurement would never update. A 200-millisecond delay was placed in between successive polls to the Maxim chip to correct this problem.

Another problem arose when the custom characters were being displayed to the screen. The program initially relied on successive increments of the table pointer to display the individual characters. This worked correctly, provided that no other functions manipulated the values in the table pointer registers during the advancements of the

progress bar. This, however, was not the case during program development because the multiply and divide macros included in the MATH18 files did manipulate the table pointer registers and did not return these registers to their initial values. This problem was corrected in the final release of the program.

6 Conclusions

The overall project was a success. The functionality of the application was achieved and the end result passed several intensive tests. These tests involved realistic one-point and two-point calibrations, as well as experimental tests that verified the programs computations and conversions. Due to the unavailability of the MAX6675, the functionality of the interface with this chip could not be tested.