Activity: Integrate Temperature Sensor

In this activity, we replace the simulated temperature sensor with a physical temperature sensor. The steps in this lab are based on the Adafruit MCP9808 High Accuracy I2C Temperature Sensor Breakout Board. Use of other sensors may require adjustments to the steps.

# Assemble the Hardware

|  |  |  |
| --- | --- | --- |
| 1 | Solder the header pins to the breakout board. | Before:    After: |
| 2 | Power down the BeagleBone AI.  Carefully wire the MCP9808 breakout board to the BeagleBone AI. WARNING: THE BEAGLEBONE AI GPIO PINS ARE RATED FOR 3V3.  MCP<----- ---------> Bealgebone AI  Vdd 9.3 - VDD\_3V3  Gnd 9.1 - DGND  SCL 9.19 - I2C2\_SCL  SDA 9.20 - I2C2\_SDA  Power on the BeagleBone AI. |  |

# I2C Command Line

|  |  |  |
| --- | --- | --- |
| 1 | Use the command line to detect the MCP9808 breakout. | debian@beaglebone:~$ sudo i2cdetect -r 3  [sudo] password for debian:  WARNING! This program can confuse your I2C bus, cause data loss and worse!  I will probe file /dev/i2c-3 using read byte commands.  I will probe address range 0x03-0x77.  Continue? [Y/n]  0 1 2 3 4 5 6 7 8 9 a b c d e f  00: -- -- -- -- -- -- -- -- -- -- -- -- --  10: -- -- -- -- -- -- -- -- 18 -- -- -- -- -- -- --  20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --  70: -- -- -- -- -- -- -- -- |
| 2 | Read the current temperature from the command line. | debian@beaglebone:~$ i2cget 3 0x18 0x05 w  WARNING! This program can confuse your I2C bus, cause data loss and worse!  I will read from device file /dev/i2c-3, chip address 0x18, data address  0x05, using read word data.  Continue? [Y/n]  0x93c1  debian@beaglebone:~$ |
| 3 | Interpret the temperature.  On the BeagleBone AI, the bytes are swapped when reported by i2cget. So, the temperature value is actually 0xc193.  From the datasheet, the scaled temperature in two’s complement is found in bits 11 through 0.  Masking 0xc193 with 0x0fff gives the magnitude of 0x0193, 403 in decimal. This value is scaled to 2-4 (0.0625 in decimal) °C yielding a value of 25.1875 °C. Does that seem about right?  Our app is apparently using °F (I wonder why). Converting to °F yields a temperature of 77.3375 °F. |  |

# Simple C Program

|  |  |  |
| --- | --- | --- |
| 1 | We’ll use an available MIT licensed library to simplify the I2C access from a C program. | debian@beaglebone:~/cFS$ mkdir 3rdparty  debian@beaglebone:~/cFS$ cd 3rdparty  debian@beaglebone:~/cFS/3rdparty$ git clone https://github.com/METECS/serbus.git  <snip>  debian@beaglebone:~/cFS/3rdparty$ |
| 2 | Create a simple C program to read the temperature. | debian@beaglebone:~/cFS$ mkdir sandbox  debian@beaglebone:~/cFS$ cd sandbox  debian@beaglebone:~/cFS/sandbox$ mkdir readtemp  debian@beaglebone:~/cFS/sandbox$ cd readtemp  debian@beaglebone:~/cFS/sandbox/readtemp$  debian@beaglebone:~/cFS/sandbox/readtemp$ cat Makefile  all: readtemp  clean:  rm -f readtemp  rm -f \*.o  readtemp: readtemp.o  readtemp: i2cdriver.o  gcc -ggdb -Wall -Wextra -pedantic -o readtemp readtemp.o i2cdriver.o  readtemp.o: readtemp.c  readtemp.o: ../../3rdparty/serbus/include/i2cdriver.h  gcc -ggdb -Wall -Wextra -pedantic -c -I ~/cFS/3rdparty/serbus/include readtemp.c  i2cdriver.o: ../../3rdparty/serbus/src/i2cdriver.c  i2cdriver.o: ../../3rdparty/serbus/include/i2cdriver.h  gcc -ggdb -Wall -Wextra -pedantic -c -I ~/cFS/3rdparty/serbus/include ../../3rdparty/serbus/src/i2cdriver.c  debian@beaglebone:~/cFS/sandbox/readtemp$ cat readtemp.c  #include "i2cdriver.h"  #include <stdio.h>  #include <stdlib.h>  #define MCP9808\_BUS 3 // Connected to /dev/i2c-3  #define MCP9808\_ADDR 0x18 // MCP9808 slave address  #define MCP9808\_REG\_AMBIENT\_TEMP 0x05 // MCP9808 register for ambient temperature  /\*\*  \* @brief Reads and returns the current temperature from the MCP9808  \*  \* @param i2c\_fd I2C bus file descriptor  \*  \* @return the current temperature in Fahrenheit  \*/  float getTemp(int i2c\_fd) {  uint8\_t rx\_buffer[2] = {0};  int raw\_value = 0;  float converted\_value = 0.0f;  // Set the slave address:  if (I2C\_setSlaveAddress(i2c\_fd, MCP9808\_ADDR) < 0) {  printf("\*Could set slave address to %d\n", MCP9808\_ADDR);  exit(0);  }  // Read the 2 bytes of data:  I2C\_readTransaction(i2c\_fd, MCP9808\_REG\_AMBIENT\_TEMP, (void\*) rx\_buffer, 2);  // Process the raw temperature value  raw\_value = rx\_buffer[0] << 8 | rx\_buffer[1];  raw\_value &= 0x0fff;  converted\_value = raw\_value \* 0.0625;  converted\_value = (converted\_value \* 9.0f / 5.0f) + 32;  return converted\_value;  }  int main() {  int i2c\_fd;  float temp;  // Open the I2C device file:  i2c\_fd = I2C\_open(MCP9808\_BUS);  if (i2c\_fd < 0) {  printf("\*Could not open I2C bus %d\n", MCP9808\_BUS);  exit(0);  }  // Read and print the current temp  temp = getTemp(i2c\_fd);  printf("Temp : %5.2fF\n", temp);  // Close the I2C file descriptor:  I2C\_close(i2c\_fd);  return 0;  }  debian@beaglebone:~/cFS/sandbox/readtemp$ make  gcc -ggdb -Wall -Wextra -pedantic -c -I ~/cFS/3rdparty/serbus/include ../../3rdparty/serbus/src/i2cdriver.c  gcc -ggdb -Wall -Wextra -pedantic -c -I ~/cFS/3rdparty/serbus/include readtemp.c  gcc -ggdb -Wall -Wextra -pedantic -o readtemp readtemp.o i2cdriver.o |
| 3 | Run the program and observe the results. | debian@beaglebone:~/cFS/sandbox/readtemp$ ./readtemp  Temp : 77.11F |

# Modify Sample IO App

|  |  |  |
| --- | --- | --- |
| 1 | Modify the sample IO app that simulates reading a temperature to actually use the temperature sensor.  Good luck! |  |