HW4
CMPE364: Microprocessor Based Design
Spring 2017

Imagine that you are part of a team designing a robot that can navigate a maze. As part of this design, you have been tasked with determining how to interface to an I2C ultrasonic range sensor. Your team leader informs you that you will be using the MB1202 I2CXL-MaxSonar-EZ0.

Your job is to a write a simple mbed program that reads and prints the range values from the sensor once every second.

What to submit:
You will submit a mini-report containing:

1. A half page write-up that summarizes the most important things to know about interfacing to this sensor. (You should include things like the device address, how to configure it, how to read data from it, what format the data is in, etc.)
2. The source code of your mbed program that reads and prints the value of the range sensor once every second. Your code should be heavily commented to discuss exactly what each line does and why.

Solution

Write-Up

The MB1202 I2CXL-MaxSonar-EZ0 is an ultrasonic distance sensor with an I2C interface. It measures the distance, in centimeters, and item is from the sensor. Here is the most important information needed for interfacing to this device:

- The 7-bit device address (in binary) is 1110000. When using this with the mbed, an 8-bit value is expected and 0xE0 should be used.
- Taking a reading is a three step process. First, you send a command to take a range reading. Then you wait at least 100ms. Then, you read the two byte response.
    - The command to take a range reading is 01010001 (81).
    - After requesting a range reading, initiate a read and the device will immediately send you two bytes. There is no need to send a command during the read.
- The two bytes come back in MSB to LSB order.

Commented Source Code

```
#include "mbed.h"

I2C i2c(p28, p27)

// The address of the device
const int addr = 0xE0;

int main() {
     char cmd[2];
     int range = 0;

     while(1) {
          // Command to request the sensor take a range reading
          cmd[0] = 81;
          i2c.write(addr, cmd, 1);

          wait(0.5);

          // Read the values.  This reads two bytes.
          i2c.read(addr, cmd, 2);

          // Convert the two bytes into the range
          range = (cmd[0]<<8) | cmd[1];

          // Print the range
          printf("The range is %d cm\n", range);

          wait(0.5);
     }
}
```