CSCE 236 Embedded Systems, Spring 2012
Lab 6

In Class: Thursday, March 29, 2012

Names of Group Members:

1 Instructions

This is a group assignment to work on during class. You only need to hand in one copy of this, but make sure that the names of all of your group members are on this sheet to receive credit. Complete all of the sections below and make sure to get the instructor or TA to sign off where required.

2 I^{2}C Inter-Robot Communication

In this section we will modify the sample I^{2}C code from the Arduino library to enable bi-directional communication between two Arduinos connected together. The Arduino calls their I^{2}C library *Wire*[1]. Look at the Arduino example code for the Wire library, in particular the `master_writer` and `slave_receiver`.

Now, connect your Arduinos' I^{2}C buses together. First, you must connect the grounds of the two boards together (to give them the same voltage reference). Then connect the SCL and SDA pins together. SCL is on pin A5 and SDA is pin A4 (they are also available above the AREF pin if you prefer). Test the `master_writer` and `slave_receiver` to verify that they function together correctly.
Checkoff: *Show the sample Arduino code working on your Arduinos.*

It turns out that with I^{2}C a device can be both a master and a slave. Now, write code so that when one of the buttons is pressed, the light on the other Arduino will turn on. To do this, you simply need take the `slave_receiver` code and in the main loop add a transmission (same as from `master_writer`) whenever the button is pressed or released. You can use identical code on each Arduino, just make sure you switch the addresses. In addition to turning on the other's LED, print over the serial port "my button pressed" and "other button pressed."
Checkoff: *Show the code turning on and off the other Arduino's LED and the serial printing. Note that it should work from board A to B and from B to A.*

3 Compass and Accelerometer

In this section we will start to use the compass and accelerometer, which are I^{2}C devices. On the course webpage there is a link to the datasheet for the ST LSM303DLM 3-axis compass and accelerometer chip that we are using[2]. Connect this to your breadboard and wire it appropriately to the power and I^{2}C lines. **Make sure to do this without power applied to your board. Before powering your board make sure everyone in your group verifies the connections to avoid burning out the chip.**

On the course website, there is sample code for this lab for reading the magnetometer and accelerometer values from the chip. Download this code, review it, and compile it. If you have connected the board correctly, you should be able to view the serial output, which will be all zeros. If any error messages are

[1] I^{2}C is generically known as a "two-*wire* interface" (TWI) because using the I^{2}C name used to require paying a licensing fee. SMBus is a stricter subset of the I^{2}C protocol and is often used on computer motherboards to communicate with low-speed peripherals (e.g. a temperature sensor on the motherboard). In other words, there are three names (I^{2}C, TWI, SMBus) that all refer to basically the same interface and protocol.

[2] The module we are using is from Pololu: `http://www.pololu.com/catalog/product/1273`.

printed, there is something wrong with your connections. Ask an instructor for help if error messages are printed.

Your job is to fill in the code that takes the series of bytes received from the sensors and puts them into the global variables to make it available in the rest of the code. Look for the sections labeled `STUDENT CODE` and fill them in appropriately by referring to table 15 in the LSM303DLM datasheet. The sensor reading for each axis are sent as two separate bytes that you must combine (bit shifting) to get the actual reading. The order that these are sent is specified in the datahsheet table. You can read the next received byte by using the command `Wire.read()`.

Checkoff: *Show the output of the accelerometer and magnetometer.*

Checkoff: *Experimentally verify the orientation of all of the axes of both sensors. This serves to verify that you correctly parsed the received data.*

Getting a compass heading requires a little more work. Fortunately, we can take advantage of an existing library to do this. Go to: `https://github.com/pololu/LSM303` and download the source (there is a **zip** icon towards the top of the page that lets you get a zip file of the source.). Place the LSM303 directory in your own sketch directory (where things you save in arduino go) under a folder called `libraries` (create this folder if it doesn't exist). Then restart the Arduino program and this code should show up under the libraries examples in the menu as LSM303. Follow the instructions on the webpage for calibrating the compass and obtaining a heading. This will be needed for the final project, which has a checkoff for this part.

More info on compasses, if you want to know: Generally, to obtain a compass heading you need to project the 3D magnetic field vector (this points towards the strongest magnetic field, which is north if you aren't around any magnets or other magnetic material) onto the ground plane determined from the 3D "down" vector from the accelerometer. This will then tell you the direction of magnetic north. Doing it this way lets you get a compass heading regardless of how you orient the device and is why these two sensors are combined on a single chip. Our robot stays relatively level, so you don't absolutely need the accelerometer. However, for handheld compasses (e.g. smart phones) or other robots (e.g. one that goes up and down hills) it is important to do this full projection approach[3]. In practice, the magnetic field sensors are not completely linear, especially when near any metal, so rather extensive calibration (much more than setting maxes and mins as the above library does) is needed if you want truly accurate heading information.

[3]It is also fun to think back to the "old" magnetized balanced pin type of compass, which will only work when it is relatively level. When used on ocean going ships, they had to create complicated gimbals to keep it level as the ship pitched or rolled. In essence, the magnetometer replaces the magnetized pin and the accelerometer replaces the gimbal.