P542 Hardware System Design II

Lab Report 4

Enrique Areyan

Description of solution

In this lab I follow a similar patterns to that of previous labs. In this case I first develop a driver for the I2C communication interface and then two drives that leverage the I2C interface to communicate with the accelerometer and magnetometer.

Note that the state machine for this lab is exactly the same as for lab 3 with three states, one per axis. In the same manner as lab 3, all of the state machine is wrapped around a `while(1)` and implemented as a `switch` statement. We poll data from the gyroscope and accelerometer at each iteration right after the while statement and regardless of the state of the machine. In this case we avoid any potential buffer issues like those faced in lab2.

To get the angle of the board after obtaining the data from the accelerometer, I had to normalize each component dividing by the sum of the data read from the three axis, then taking the *arcsin* (which makes sense since after normalizing the numbers which range between -1 and 1), and then converting to degrees by multiplying by 180/pi. This operation takes place only for the user-selected axis.

Description of issues

Fortunately, there were no major issues in this lab. By following the same design pattern that proved to be successful in previous labs I avoided common pitfalls and was able to finish the assignment in a reasonable amount of time. The only issue I encountered while setting the I2C communication was that of forgetting to change the port letter to B in the GPIO_Init function, a common error due to code copy and paste. This goes to show the level of attention to detail that one has to have while coding drivers for this type of embedded systems.

Teammate for the project **Sajith Sasidharan**.

Project Ideas

1. **Remote control car for weather reporting**: build a remote controlled car that can take periodic room readings such as: temperature, light and humidity; and report back via Wi-Fi, or another wireless interface, to the board where a primitive, terminal interface can show a “map” of the readings for a fixed *NxM* grid.
2. **Remote glove controlled car**: build a remote controlled car where the remote control primitives are built into a glove. The user can press a certain combination of fingers to move the car in certain direction (e.g., press the index finger to your hand to move the car forward, press index and middle finger to move the car backward).