

MICROART 2014-2015

ZYBO Sensor Board

Author: Paul Gerver

Introduction

The purpose of this document is to inform users about setting up the Zybo board for collecting data from the SparkFun MPU9150 sensor board. For setting up the Xilinx Tools to run the example, please see the master Xilinx Tools walkthrough.

The main focus of this task is to collect Accelerometer, Gyroscope, and Magnetometer data from the board and calculate the X and Y angles of the board (and eventually the quad) as well as positional information.

To communicate with the sensor board, the Zybo uses one of its I2C controllers to speak with the board at the speed of 400kHz. The board by default is in sleep mode, so the board is woken up by writing to a certain register. Please see the code documentation for all the details

XPS WARNING: Please follow the Xilinx Tools HowTo guide (located in “docs”) properly to get the I2C controller configure properly. Many hours have been put in to find out the correct configuration, and we don’t want you losing that time as well.

PREREQ: Read the Xilinx Tools HowTo

Step-by-step Walkthrough

Hardware

You’ll need:

- Zybo board (with USB cord)
- SparkFun MPU9150
- Pin connectors (Fly wires)

Please connect the MPU9150 pins GND, VCC, SDA, SCL to the **JF PMOD** (5,6,3,2 respectively)



Figure 16. Pmod diagram.

Software

1. Open XSDK and for the workspace select “tasks/zybo_sensor_board/sw”
2. The workspace should open with the files
 - a. If the project is blank, Import Application, BSP, and HW_platfrom
3. Ensure JTAG is properly configured, program the FPGA, launch the sensor_board program
4. In a COM terminal, view the data being sent back from the board with X (Pitch), Y (Roll) and Z (Yaw) angles calculated from the gyro and accelerometer.

Troubleshoot: When in doubt, re-export the XPS project to XSDK

FAQ

Q: How does this thing actually work?

A: Hardware-wise, through I2C. Software, now that’s a different story since this we created a utils file that operates the I2C controller by setting its registers properly (288 style). It can seem convoluted if you look at the code, but it does work.

Q: The angles are changing without me moving the board. Why?

A: Gyro’s are prone to drift and the Yaw angle, Z, is only a gyro reading, so it is very prone to drift. X and Y are also based on the gyro and accelerometer filtered together, so these shouldn’t change too much. When using different MPU9150 sensor boards, the gyro drift correction made in the software may need to be tweaked.

Q: I launched the program, and the data is not flowing. What’s wrong?

A: Something has gone terribly wrong. Ensure the bitfile being programmed to the board is configured as specified in the Xilinx Tools HowTo. Also check that the sensor board is plugged into the proper PMOD port (JF, next to the power switch), and pins (GND, VCC, SDA, SCL = 5,6,3,2).

Files

-sw	Directory containing the source code and XSDK workspace
-system	Directory containing the XPS project for the board