

OREGON STATE UNIVERSITY

# Head-Up Display Alignment System

## Technology Review

*Senior Software Engineering Project (CS461)*  
*Fall 2016*

*Authors:*  
Krisna Irawan  
Jiongcheng Luo  
Drew Hamm

## Abstract

TODO (K)

Contents

I Introduction . . . . . 2

II Technologies . . . . . 2

    II-A Hardware selection of Microcontroller . . . . . 2

    II-B Hardware selection of Represented MEMS IMU . . . . . 3

    II-C Hardware selection of Communication Protocol Type . . . . . 4

III Conclusion . . . . . 5

References . . . . . 6

## I. INTRODUCTION

There are nine different main technologies that will be used or involved in this project. These technologies contain using of both software and hardware. For instance, for hardware, we will compare and discuss about some possible hardware options for building the demonstration system; for software, we will compare and discuss different algorithms or approaches for resolving the problem. Each of us takes an authorship (responsibility) for three technologies:

### *Jiongcheng (Roger):*

- Hardware selection of Microcontroller
- Hardware selection of Represented MEMS IMU
- Hardware selection of Communication Protocol Type

### *Krisna:*

- Hardware selection of Data display device
- Algorithm to find HUD offset using statistical correlation
- Programming Language

### *Drew:*

- Represent IRU: Simulation techniques
- Algorithm to convert raw MEMS IRU data output to Quaternion output
- Algorithm to get offset using both sensor outputs

## II. TECHNOLOGIES

### A. Hardware selection of Microcontroller

*a) Context:* It is critical to choose the most appropriate microcontroller for this system since there are limitations from both hardware and software perspectives. A microcontroller plays an important role in the system, which it will have following functionalities in the system:

- Process input from the IMUs and output to the computer/display
- Process the alignment algorithm
- Debugging and testing purpose

*b) Options:* We consider to use a board between *Arduino Pro Mini* [1], *Adafruit Pro Trinket* [2] and *InvenSense MPU-9250 CA-SDK Reference Board* [3]. Anyone of these board has its outstanding areas and lack, and we will choose one of them based on the following criteria.

#### *c) Criteria:*

- 1) **Support Language**, we expect to use a general programming language for the microcontroller since using familiar programming language will reduce our time on addition research and the save time on working on the algorithm itself.
- 2) **Clock Speed**, the alignment algorithm is required to within around 500 milliseconds time from taking the input from the IRUs to the output of aligned data, that require the board has a fast speed and running any processes.
- 3) **I2C Protocol**, this is necessary to have on the board since the MPU-9250 (our selected IMU) require I2C protocol to communicate.
- 4) **Connection with PC**, this is necessary to have on the board since we will use computer to any software program.
- 5) **Size**, size is less important since this system is for demonstration purpose.
- 6) **Cost**, this is relatively important but as long as the price is within the expected budget, that will be acceptable.

d) Table of Detailed Comparison:

Model	Arduino Pro Mini	Adafruit Pro Trinket	InvenSense MPU-9250 CA-SDK Reference Board
Core Chip	ATmega328	ATmega328	Texas Instrument MSP430
Support Language	C	C	C/C++
Clock Speed	8MHz (3.3V mode)	12MHz (3V mode)	16MHZ (1.8V 3.6V)
I2C Protocol/Number	Yes/1	Yes/1	Embedded Structure
Connection with PC	USB	FTDI/USB	USB/Bluetooth
Size	33mm X 18mm	38mm X 18mm X 2mm	Unknown
Cost (U.S Dollar)	\$9.95	\$9.95	\$440.00
Advantage	Small size	Faster clock speed	With embedded IMU
Shortage	Slower Clock Speed	Lack of resource of guidance/datasheet	High cost

e) *Overall Discussion:* By comparing these boards by the above criteria, *Arduino Pro Mini* and *Adafruit Pro Trinket* have similar performance and specification, they both have the advantages of small size and low cost. *InvenSense MPU-9250 CA-SDK Reference Board* is a special case, which its a board as well as the IRU itself since it has embedded on-board sensors. *MPU-9250 CA-SDK* is very powerful for multi-sensor system, other than an embedded MPU-9250 (accelerometer, gyroscope and compass), it also has pressure sensor, UV sensor, humidity and temperature sensor, light and proximity sensor. However, high cost is a noticeable shortage of *MPU-9250 CA-SDK*. In addition, and not all of its functionalities are necessary for this project.

Now, we only look at *Adafruit Pro Trinket* and *Arduino Pro Mini*. Based on the comparison and the criteria, *Adafruit Pro Trinket* will be more preferable. Because this module has a faster clock speed compares to the *Arduino Pro Mini*, although it has a larger size but this doesnt affect to its actual performance based on our project requirement, other than that, all other concerned criteria are same as the *Arduino Pro Mini*.

## B. Hardware selection of Represented MEMS IMU

a) *Context:* An Inertial Measurement Unit (IMU) is the most significant part of the hardware components of this system. An MEMS IMU will be used to measure the acceleration, velocity position of the aircraft and output data for alignment algorithm. Therefore, we are looking for a model of IMU that is highly accurate, well performed as well as with affordable expense for the demonstration of this project, following chart compare these three model with a list of comparison.

b) *Options:* We are looking at three models that are considered to be the most likely options for representing MEMS IRU. *9DoF Sensor Stick* [4], *MPU-9250 IMU* [5] and *9DoF IMU* [6] are three models of IMU. All these three IMUs are with 9 degrees of freedom/9 axis sensor, that indicates all of these IMUs contain MEMS sensors of accelerometer, gyroscope and magnetometer(compass), which is a basic requirement for choosing the IMU in this project. Following are some specific criteria that we look and compare for choosing the most proper MEMS IRU.

c) *Criteria:*

- 1) **Operating voltage range**, this is determines based on the model we use for the microcontroller, which an operable voltage range of an IMU should be less than the output voltage of its microcontroller.
- 2) **Support I2C protocol**, this is significant since I2C could be the only protocol for an IMU to communicate with the microcontroller.
- 3) **Accelerometer output resolution**, this will determine the accuracy of the acceleration data output.
- 4) **Gyroscope output resolution**, this will determine the accuracy of the angular velocity data output.
- 5) **Magnetometer output resolution**, this will determine the accuracy of the orientation data output.
- 6) **Cost**, this is relatively important but as long as the price is within the expected budget, that will be acceptable.

d) Table of Detailed Comparison:

Model	9DoF Sensor Stick	MPU-9250 IMU	9DoF IMU
<b>MEMS Sensors</b>	Accel.: ADXL345 Gyro: ITG-3200 Magn.: HMC5883L	MPU-9250	LSM9DS1
<b>Operating Voltage Range</b>	2.1 3.6V	2.4 3.6V	1.9 3.6V
<b>Output Type</b>	Unknown	16 bits ADC	16 bits ADC
<b>Support I2C Protocol</b>	2g, 4g, 8g, 16g	2g, 4g, 8g, 16g	2g, 4g, 8g, 16g
<b>Gyroscope Output Resolution (Angular Velocity)</b>	Full scale = 2000 degree/s	250, 500, 1000, 2000 degree/s	250, 500, 1000, 2000 degree/s
<b>Magnetometer Output Resolution (gauss)</b>	8 gauss	48 gauss	4, 8, 12, 16 gauss
<b>Size</b>	22.22 X 18.48 mm	Unknown	14.3 mm X 20.5 mm
<b>Cost (U.S. Dollar)</b>	\$49.95	\$14.95	\$24.95
<b>Advantage</b>	Higher output resolution	Least expensive	Wide spectrum of sensor range
<b>Shortage</b>	High expense	Weaker performance	A little expensive

e) *Overall Discussion:* In overall, these three models have similar performance and specification of out resolution, however, *9DoF Sensor Stick* will be the least preferred option since it has the highest expense and it lacks of resources of its datasheet the relevant guidance. By comparing between *MPU-9250 IMU* and *9DoF IMU*, they both have specific datasheet that can be found, and their prices are both within the acceptable range, so by looking at more specific detail of their hardware performance, *9DoF IMU* has wider sensor range to choose for its output resolution, which is more functional than the *MPU-9250 IMU*. Therefore, the *9DoF IMU* model would be the first preference for selecting the represent MEMS IMU.

### C. Hardware selection of Communication Protocol Type

a) *Context:* There are many different types of protocol for hardware component devices communicating with each other. We chose three different protocol types that are the most common types to use, as well as doable in our hardware system.

b) *Options:* *I2C (Inter-Integrated Circuit)*, *SPI (Serial Peripheral Interface)* and *UART (Universal Asynchronous Receiver/Transmitter)* are three possible communication protocol types for our project, each of them has its own advantages and lack. In regards to this project, we will consider the following criteria for choosing the most appropriate protocol type for our hardware devices communication [7].

c) *Criteria:*

- 1) **High Transmission Speed**, this is one of the most critical criteria to be consider, since our hardware system requires multi-devices to process near synchronously, a fast communication speed between hardware devices is necessary.
- 2) **Transmission Distance**, this criterion is less important within this system since we consider all of the hardware devices will be implemented as a whole device, that means transmission distance of the protocol type wont be a necessary criterion.
- 3) **Multi-Devices Support**, since we may use more than 1 IMU to work together in order to gathering more accurate data, the protocol we choose to use must support multi-devices communication.
- 4) **Number of wire needed to connect with microcontroller**, this criterion is less important but less number of wires between hardware device provides higher portability of the system, and it also reduces the difficulties on the hardware set-up process.

d) Table of Detailed Comparison:

Protocol Type	<i>I2C/IIC</i>	<i>SPI</i>	<i>UART</i>
<b>Transmission Speed (Standard Speed Mode)</b>	>1 Mbit/s	~10 Mbit/s	0.3 Kbit/s ~ 1 M bit/s
<b>Transmission Distance</b>	Short (Within integrated circuit components)	Short (Within integrated circuit components)	Long (wireless)
<b>Multi-Devices Support</b>	Yes	Yes	Yes
<b>Number of Wire Needed</b>	2	3 + 1 for each signal line	none
<b>Advantage</b>	Easy to implement	Fast transmission speed	can be implemented in wireless environment
<b>Shortage</b>	Slow transmission speed	More number of wires needed	Hard to implement

e) *Overall Discussion:* By comparing the above criteria, we first may eliminate UART because it has the slowest transmission speed within these three types, so its not expected for using in our system. SPI has significant fast speed however since we will use multiple IMUs connecting together in the system, SPI doesnt provide a good portability that it requires 1 more physical wire for each additional devices. Therefore, I2C is the most preferable option, it has relatively high transmission speed, and its easy to implemented for a complicated system.

### III. CONCLUSION

TODO(D)

## REFERENCES

- [1] "Arduino pro mini 328 - 3.3v/8mhz," Sparkfun Electronics. [Online]. Available: <https://www.sparkfun.com/products/11114>
- [2] "Adafruit pro trinket - 3v 12mhz," Adafruit. [Online]. Available: <https://www.adafruit.com/product/2010>
- [3] "Invensesense mpu-9250 ca-sdk reference board user guide," InvenSense Inc, 2012. [Online]. Available: <https://store.invensense.com/datasheets/invensesense/MPU-9250CA-SDK.pdf>
- [4] "<https://www.sparkfun.com/products/10724>," Sparkfun Electronics. [Online]. Available: SparkFun9DegreesofFreedom-SensorStick
- [5] "Sparkfun imu breakout - mpu-9250," Sparkfun Electronics. [Online]. Available: <https://www.sparkfun.com/products/13762>
- [6] "Sparkfun 9dof imu breakout - lsm9ds1," Sparkfun Electronics. [Online]. Available: <https://www.sparkfun.com/products/13284>
- [7] "Introduction to ic and spi protocols." [Online]. Available: <http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/>