USB Virtual Reality HID

by

Weston Taylor and Chris Budzynski

Advisor: Dr. Malinowski

Agenda

- Project Summary
- Analysis
- Block Diagram
- Hardware
- Inertial Sensors
- Position Calculation
- USB
- Results
- Questions

Project Summary

- **Purpose:** To create a USB HID that translates user movements into on-screen actions to provide a realistic interactive platform for PC games and other virtual environments.

- **Why?:** To show that low-cost MEMS inertial sensors capable of providing accurate position data for a complex interactive experience.

Previous Work

- MetaMersion Immersive Gaming System

Goals

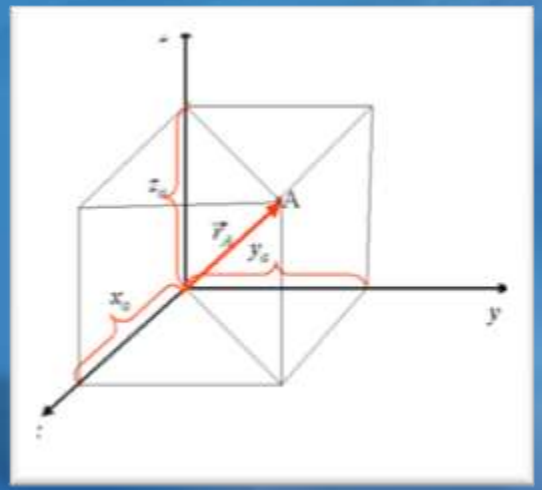
- **List of project goals:**
  - Translate sensor readings into a 3-D position
  - Send 3-D position to PC using USB (gamepad)
  - Use low-cost 8-bit embedded systems
  - Embedded programming = C language
  - Wireless communication between subsystems

Analysis

- Stationary INS (Inertial Navigation System)
  - No Linear Movement
  - Only Angular Movement (Pitch, Yaw)
  - Gyroscopes measure angular velocity

Analysis

- Gyroscopes
  - Measure °/sec
  - 1 integration to get position (absolute angle)
- Frames of Reference
  - Mathematical Transformations (Trig Functions)
  - Yaw → Dependent on Pitch
    - Cosine Lookup Table

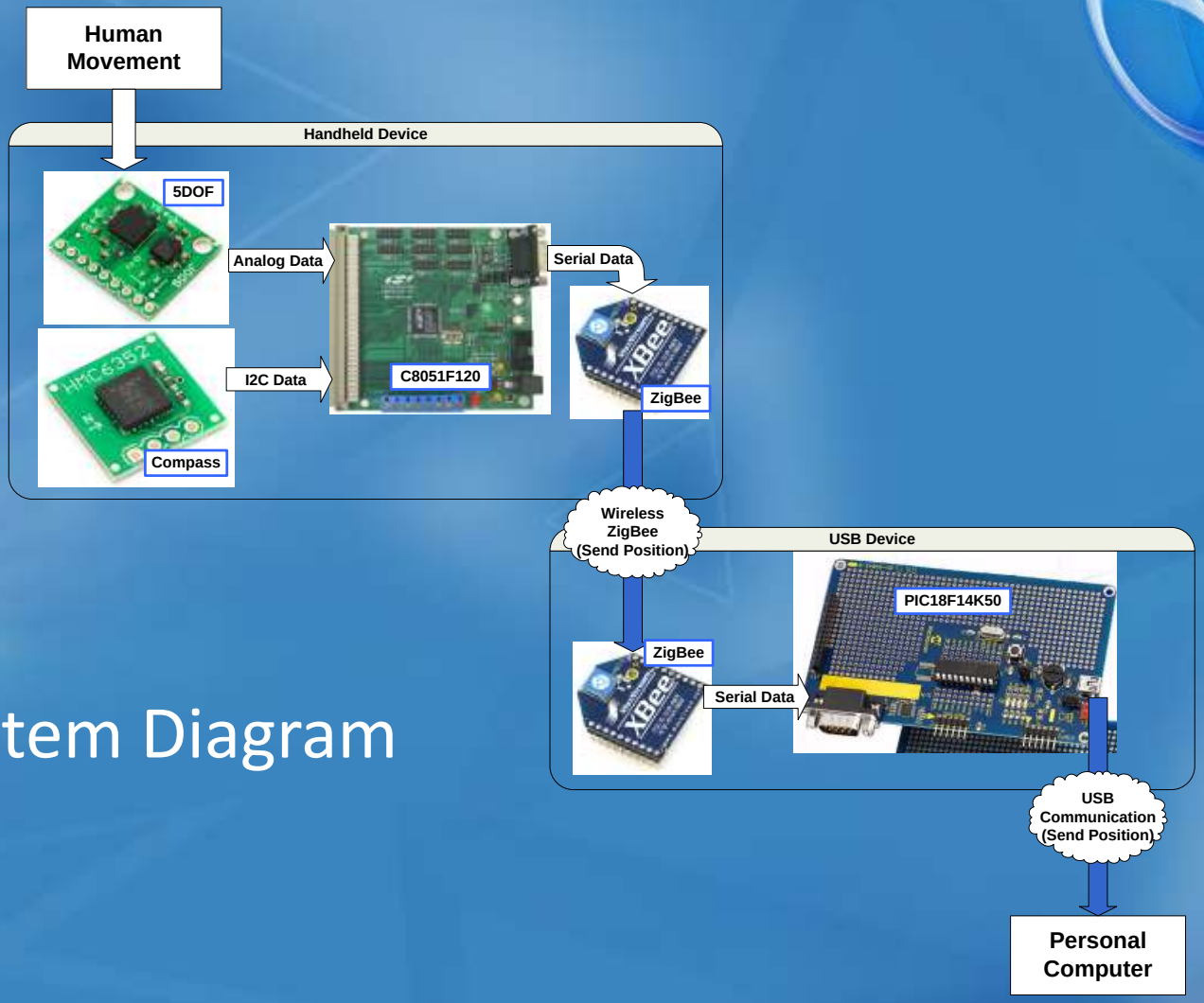


System Specs

- System Requirements
  - Hardware
    - 300-400 °/second → do not want saturation
    - Temperature Range → 0°C to 40°C
  - Software
    - Position calculation every 10ms
      - Human Reaction Time (100ms)
      - Monitor Refresh Rate = 60 Hz (16.7ms)

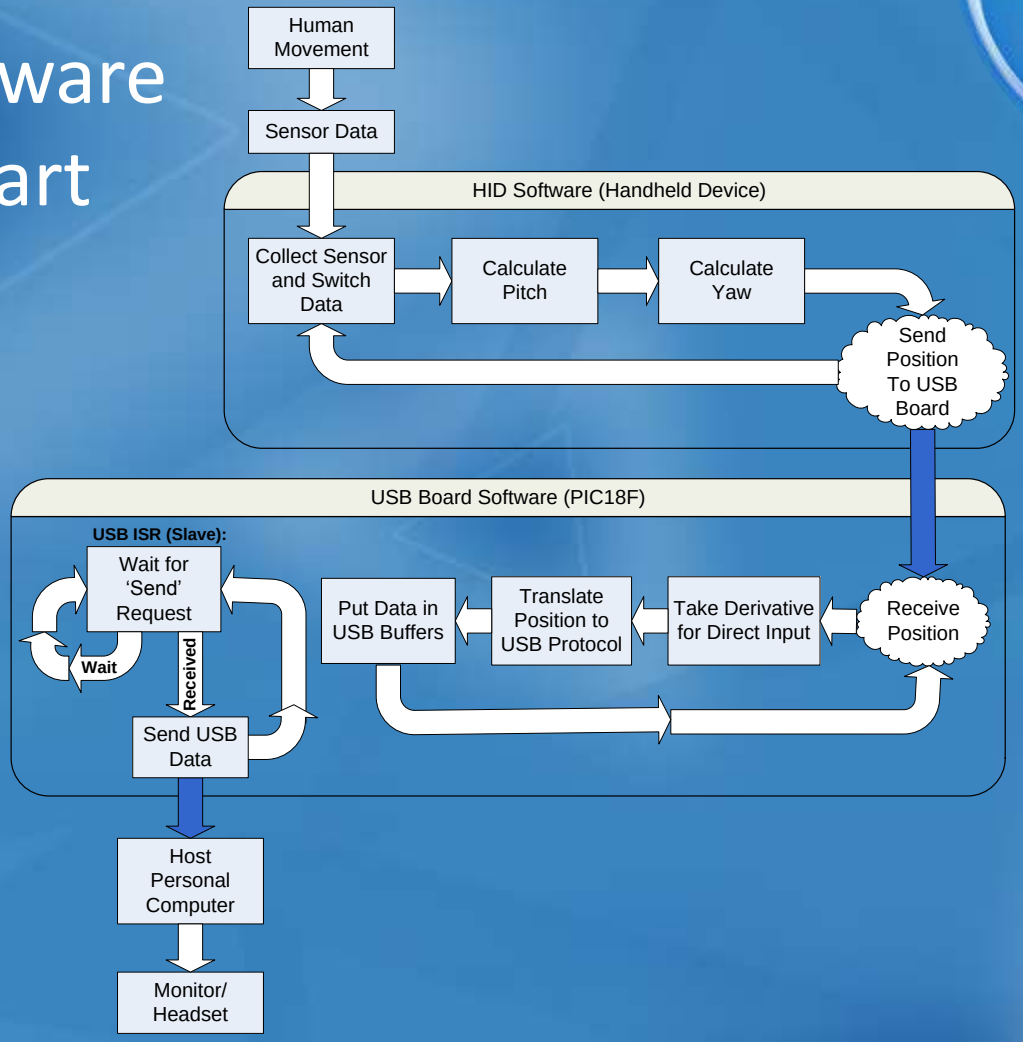
Hardware

- System Diagram



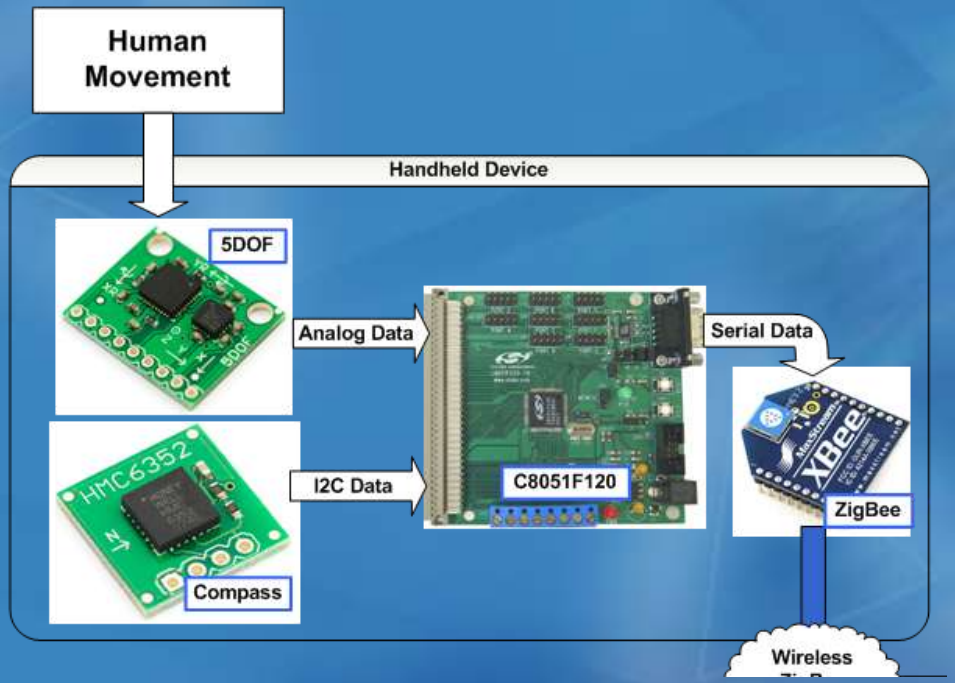
Software

- HL Software Flowchart



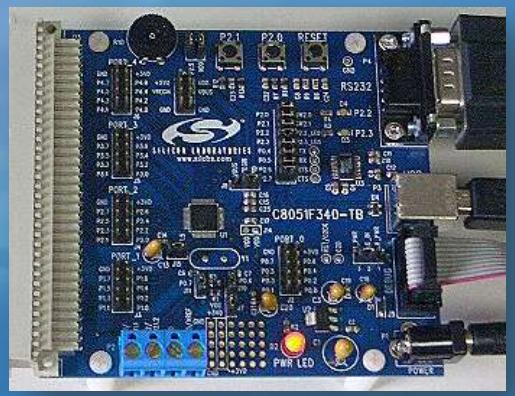
Hardware

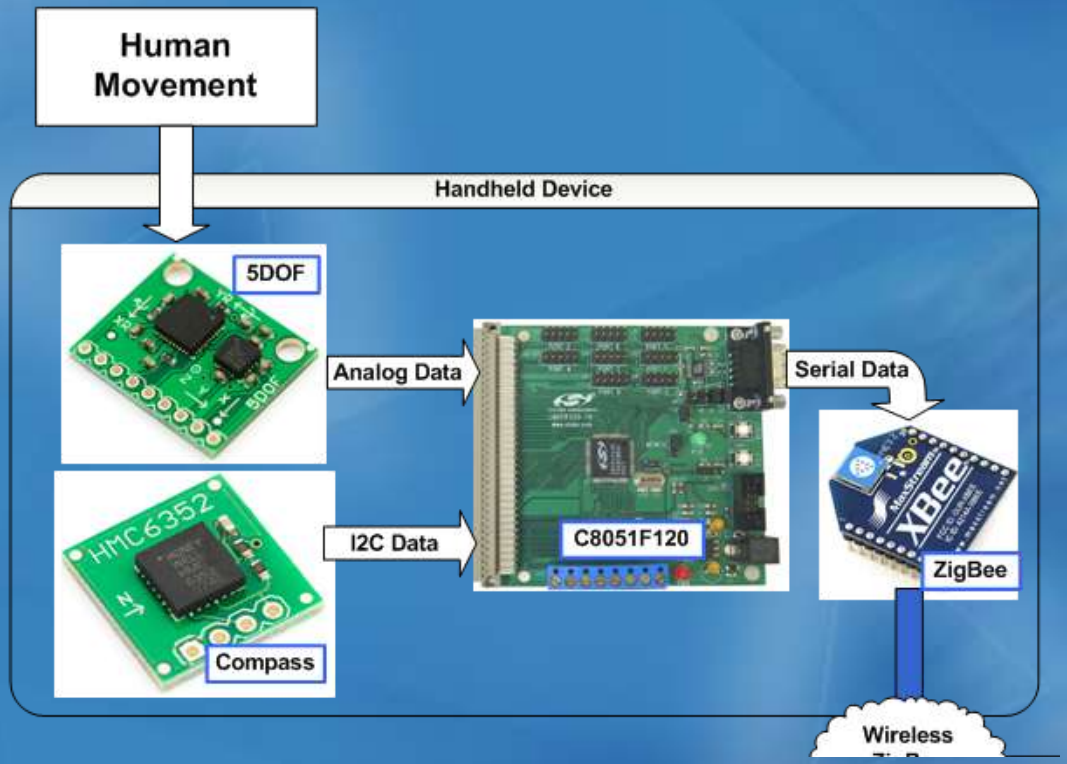
- IMU 5 DOF
  - 2-axis gyroscope = 500 °/sec
  - 2-axis accelerometer = ±3g
  - Analog Outputs
- Electronic Compass
  - HMC6352
  - I2C Interface



Hardware

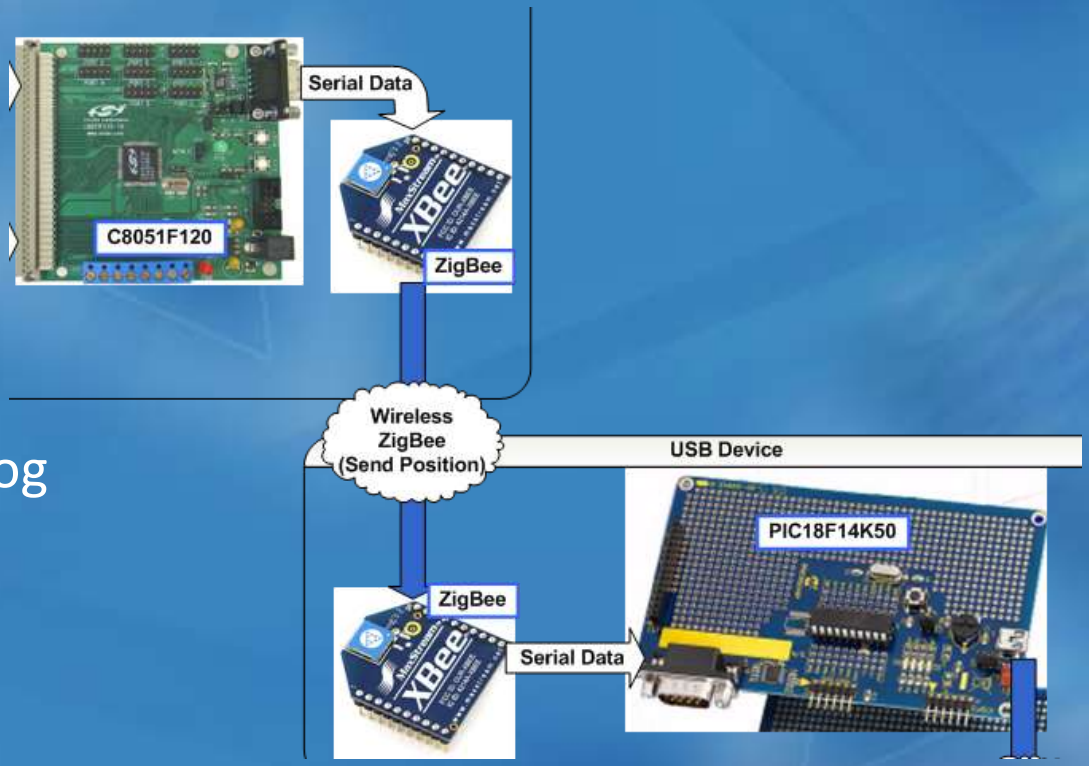
- Inertial Board - Silicon Labs C8051F120
  - 8-bit processor → 98MHz
  - 12-bit A/D





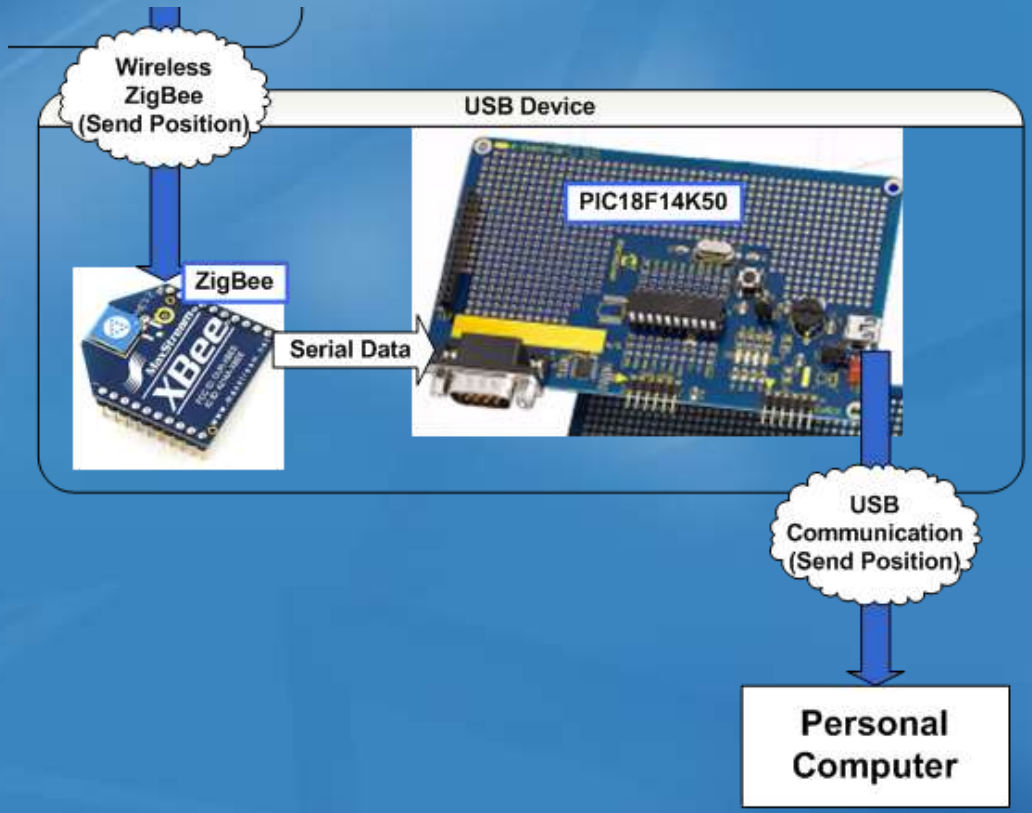
Hardware

- Wireless ZigBee - XBee 1mW Chip Antenna
  - UART to ZigBee Conversion
  - 2.4 GHz
  - Range = 300ft
  - Baud = 9600 bits/sec
  - Transparent Mode
- Problem
  - Interference with analog



Hardware

- USB Board – Microchip LPC USB Dev. Kit
  - PIC18F14K50 → 8-bit, 48MHz



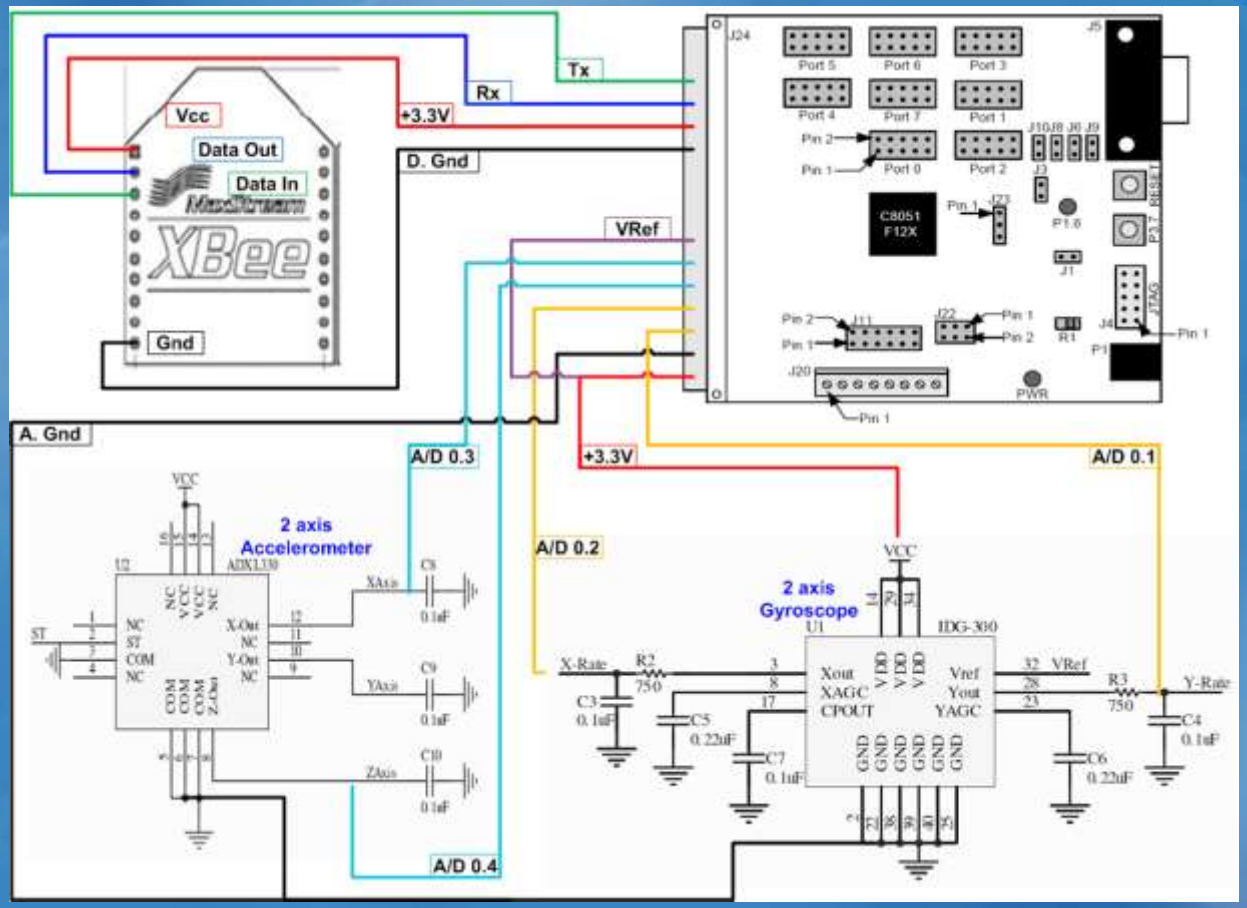
Parts List

- Parts List Table:

Equipment	Part Description	Quantity	≈ Cost	Supplier
Personal Computer	With USB and Half-Life 2	1	$0.00	Personal Laptop
USB Board	Microchip LPC USB Dev. Kit	1	$60.00	www.microchip.com
Main Board	SiLabs C8051F120 Dev. Kit	1	$99.00	In Lab
Gyro + Accelerometer	IMU 5 Degrees of Freedom SEN-00741	1	$100.00	www.sparkfun.com
Electronic Compass	Compass Module - HMC6352 SEN-07915	1	$60.00	www.sparkfun.com
Level Converter	Logic Level Converter BOB-08745	1	$2.00	www.sparkfun.com
Wireless / ZigBee	XBee 1mW Chip Antenna WRL-08664	2	$25.00	www.sparkfun.com
		Total Price:	$346.00	

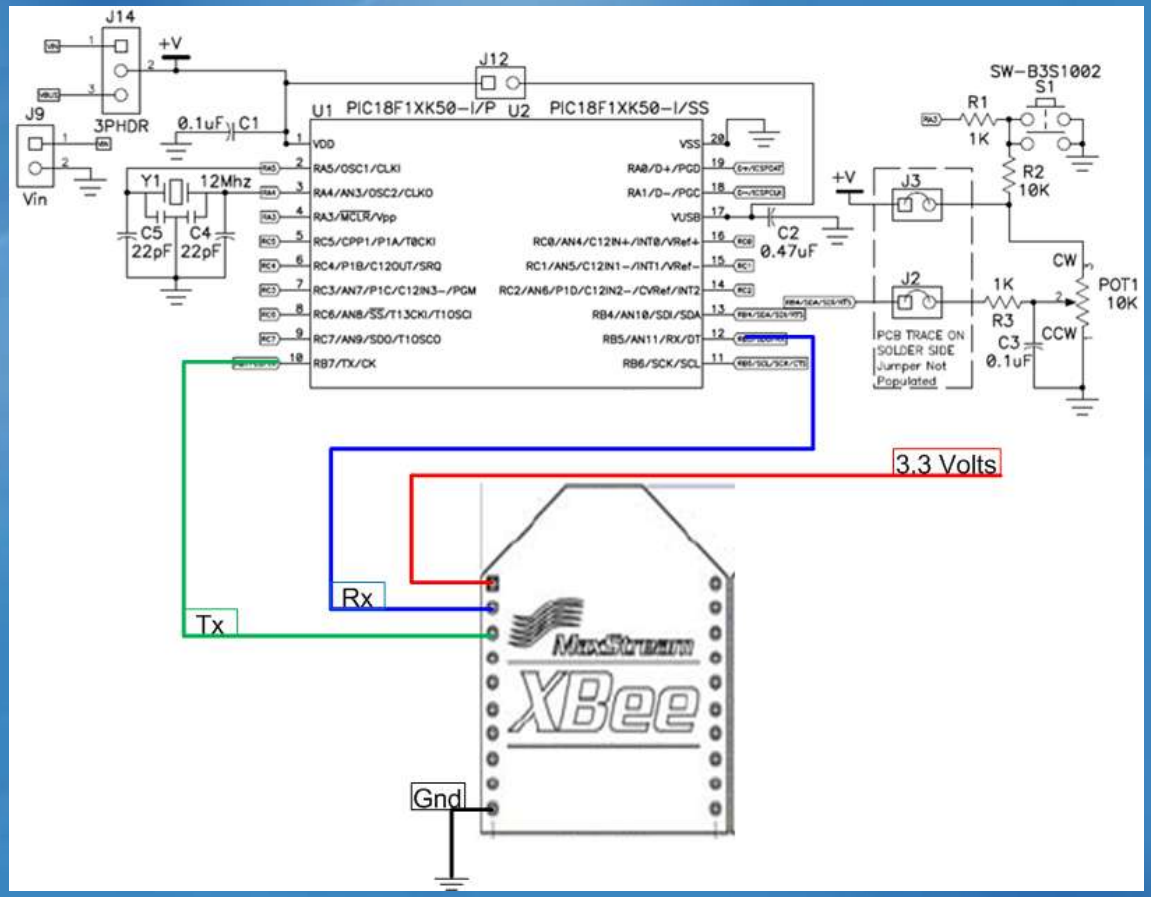
Schematic

- Inertial Board Schematic



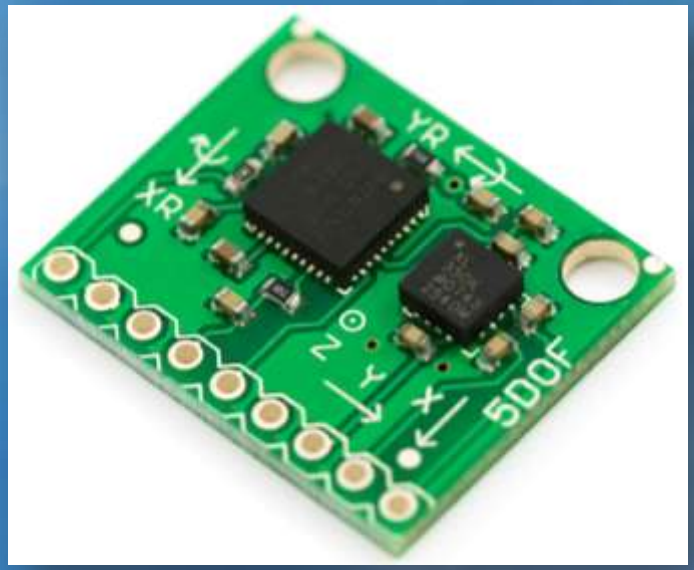
Schematic

- USB Board Schematic



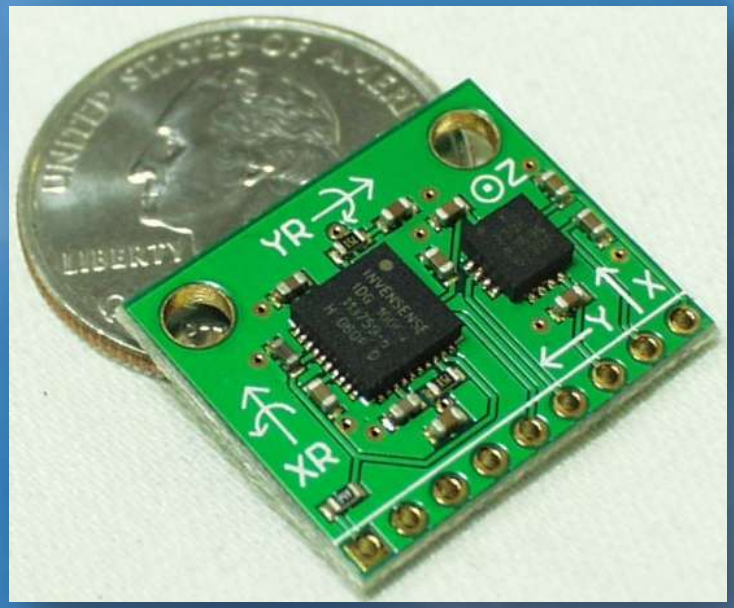
Inertial Sensors

- Inertial Sensor Inherent Problems
  - High Frequency Noise
  - Scale Factor Variations
  - Offset → causes Drift (significant)



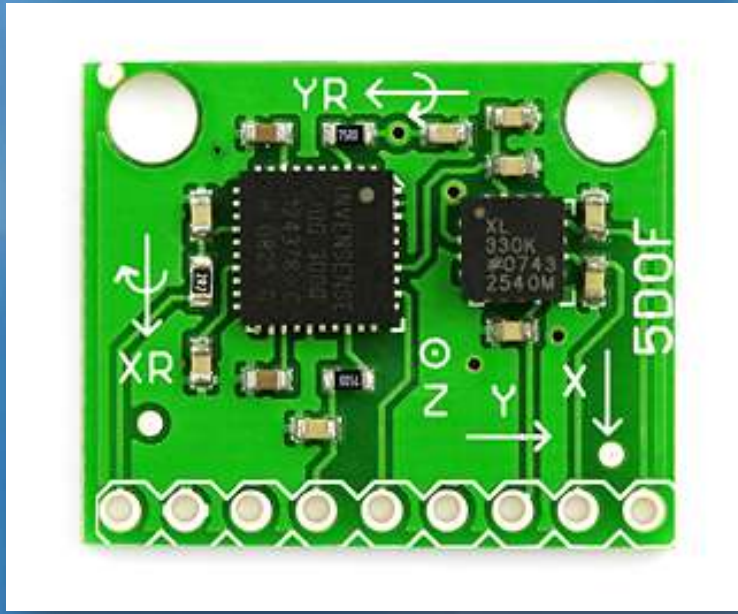
Noise

- Problem → High Frequency Noise
- Solution → LPF
  - Gyroscopes → 80Hz
  - Accelerometers → 500Hz



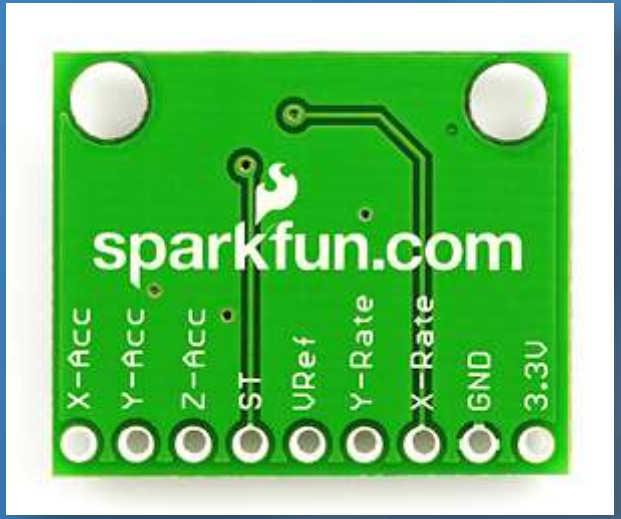
Scale Factors

- Problem → Scale Factor Variations
  - Datasheet = not exact
  - + and – directions different
  - Accumulated error = drift
- Solution → Feedback



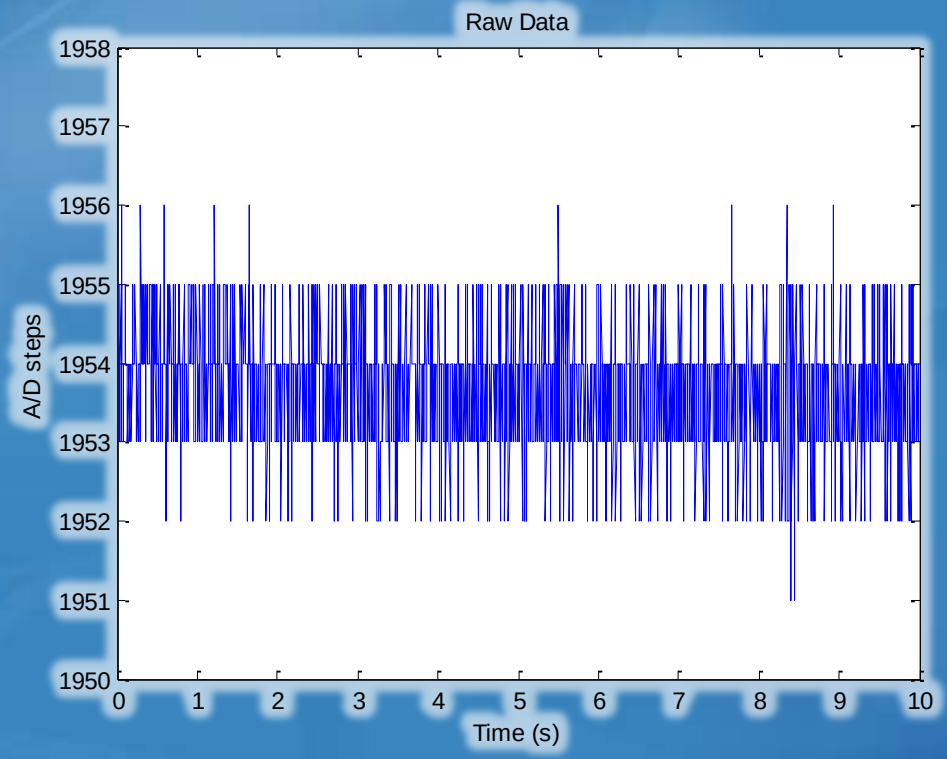
Offset

- Problem → Offset, zero-rate voltage
  - Causes drift from integration
- Solution →
  - Calibration routine = 1024 sample average
  - Adaptive zero-rate voltage
  - Window filter
  - Feedback



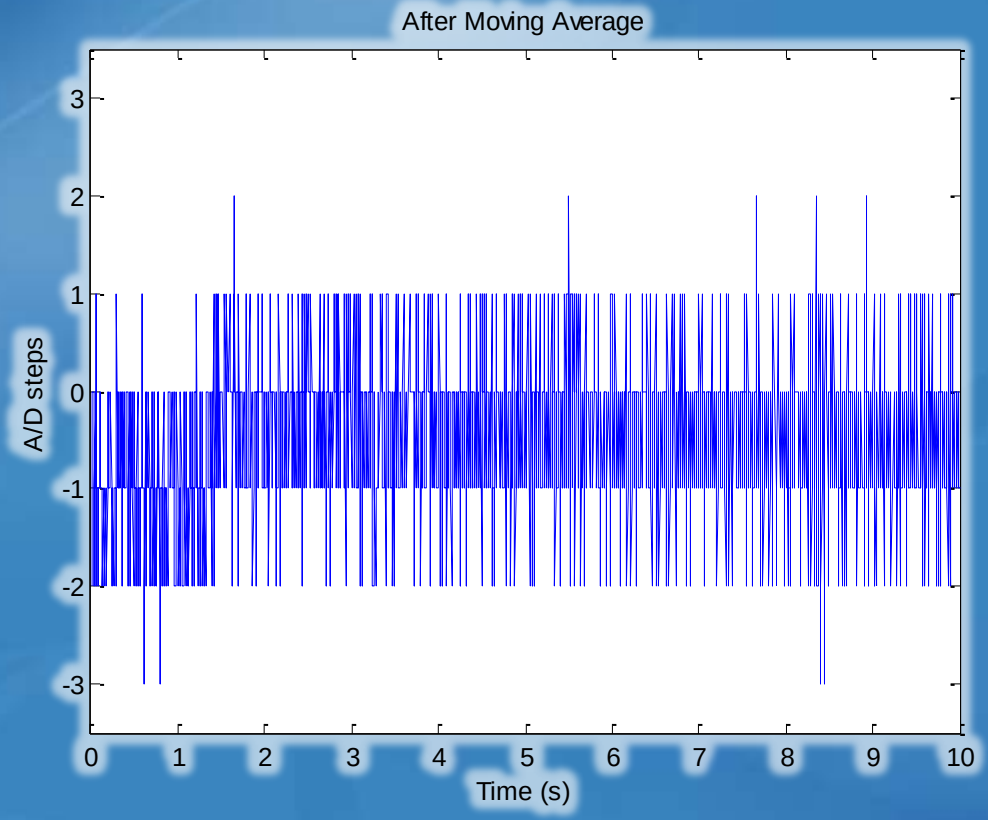
Sensor Data

- HyperTerminal (.txt) → MATLAB
- Raw A/D Data



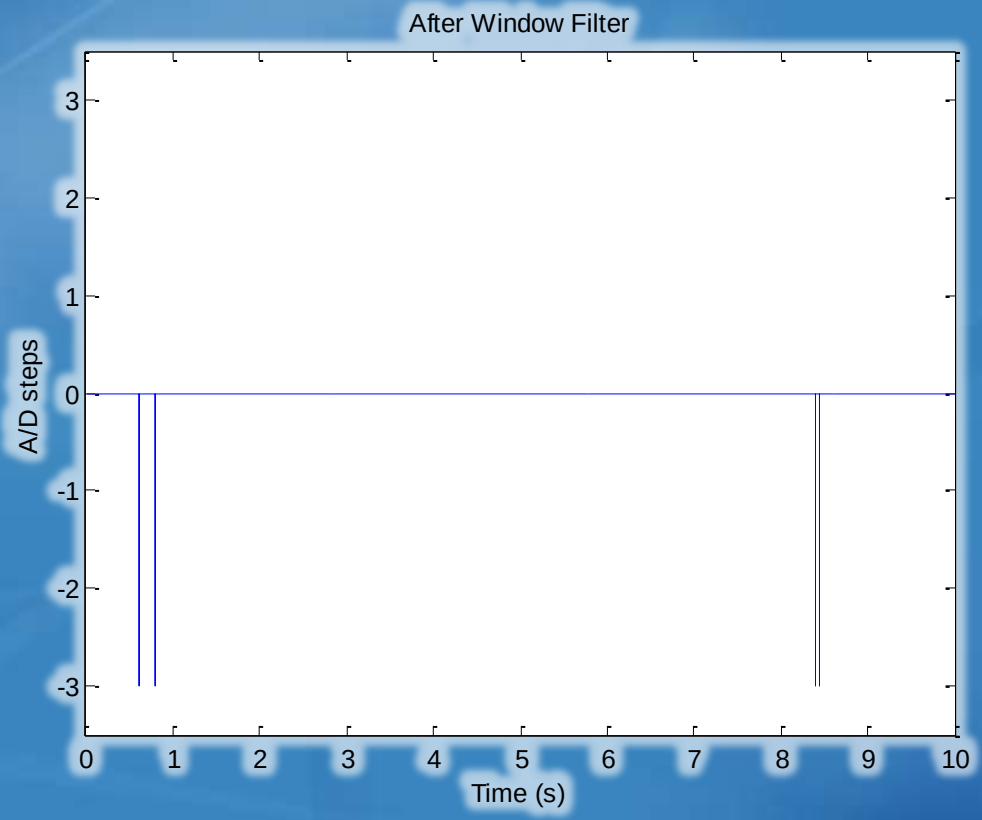
Sensor Data

- Data after calibration and adaptive zero-rate



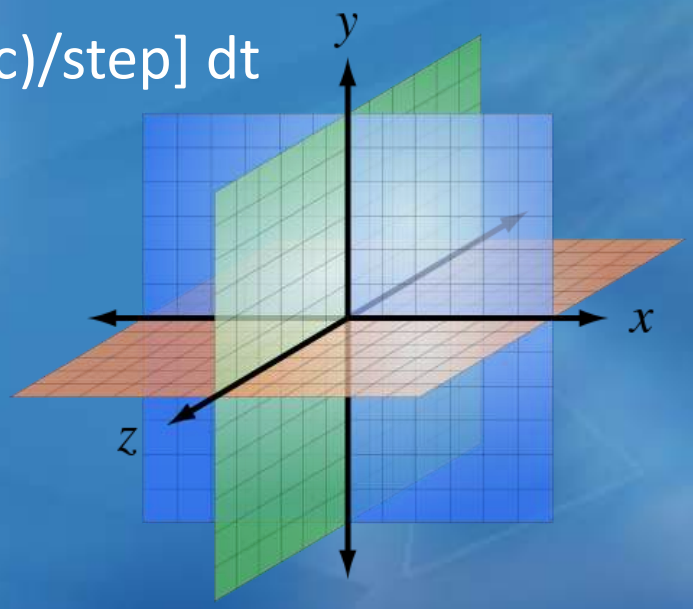
Sensor Data

- Data after window filter → no drift at rest
  - 2 A/D steps = ±1.6mV → ±0.8°/sec



Position

- Basics
  - Gyroscopes measure °/sec
  - A/D reads → steps
  - Find conversion factor → (°/sec)/step
  - Want an absolute angle in degrees (°)
- Calculation
  - ∠ Position = ∫ [A/D steps] [(°/sec)/step] dt
  - ∠ Position = ∫ [°/sec] dt
  - ∠ Position = #°



Position

- Software – Conversion Factors / Constants
  - Conversion from A/D Value to °/s:

$$\frac{3.3\text{ V}}{4096\text{ steps}} \text{X} \frac{1°/\text{s}}{2.0\text{ mV}} = \frac{1500°/\text{s}}{4096\text{ steps}} = .402832$$

  - Summation As Approximation of Integration

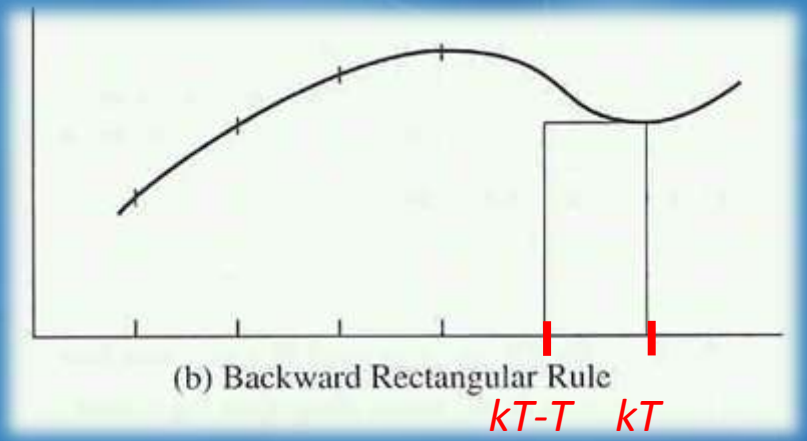
$$\int .4028(\text{ADC})dt \approx \sum .4028(\text{ADC})(\Delta t) = \frac{\text{ADC}}{496.485}$$

  - $\Delta T = 1/200\text{Hz} = 0.005$ so Position Approx. is:
    - Angular Position = Previous Position + (Current ADC/496)

Code

- Integration Summation
  - Backward Rectangular Rule

```
238     // Integration
239     pitch_integration = pitch_integration + pitch;
```



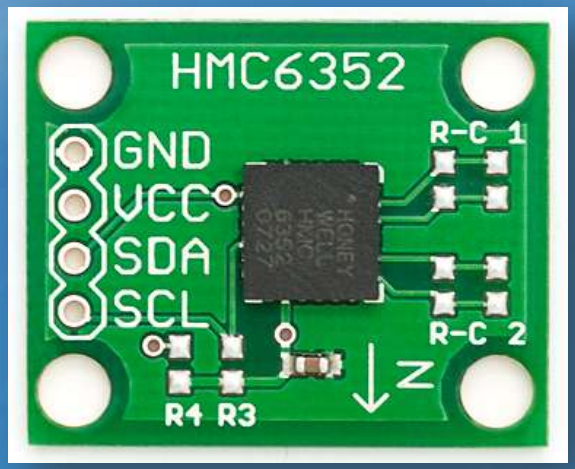
Code

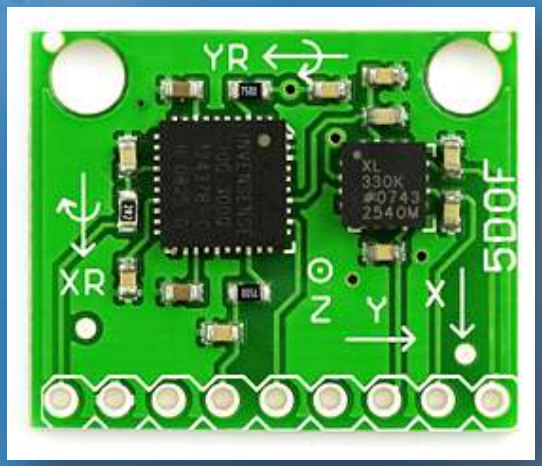
- Window Filter
- Adaptive Zero-Rate Offset
  - Only runs inside Window Filter

```
232     // Window Filter - gets rid of worst noise causing drift
233     if(pitch < 3 && pitch > -3) {
234         pitch = 0;
235
236         pitch_zrc_sum += pitch;
237         pitch_zrc = pitch_zrc_sum >> 10;
238     }
```

Feedback

- Integration of error → Drift
  - Open Loop System
- Feedback
  - Pitch Drift Correction (accelerometer)
  - Yaw Drift Correction (electronic compass)



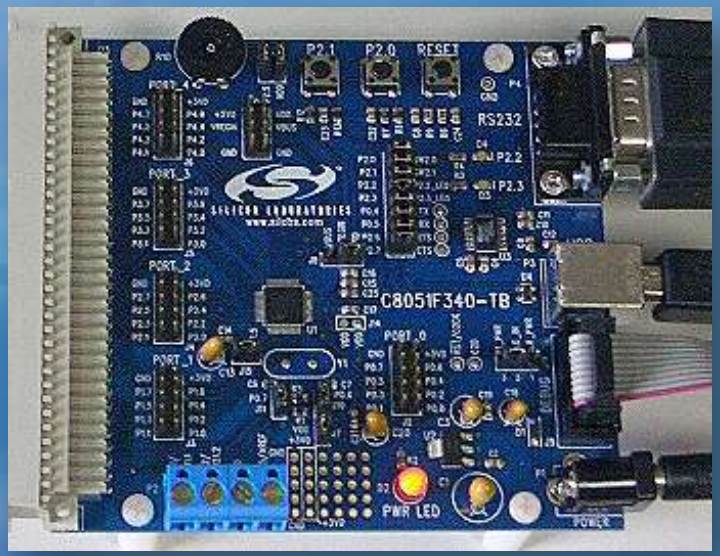


USB

- USB Basics
  - Master / Slave → Polling (choose interval)
  - Human Interface Device (HID)
    - Mouse + Keyboard or Gamepad
    - Flexible → Report Descriptor
  - Our Report Descriptor
    - Rx – Rotation around x-axis → Pitch
    - Ry – Rotation around y-axis → Yaw
    - Buttons (6)

USB

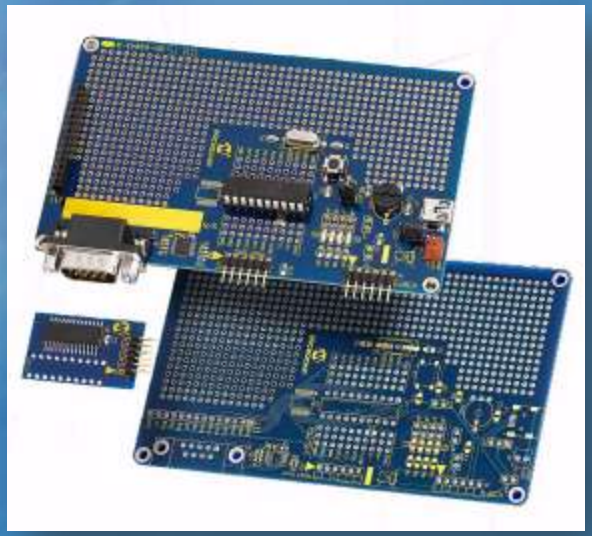
- Silicon Labs C8051F340 = USB Dev. Kit
  - Mouse and Keyboard Working
  - Gamepad → Error Code 10

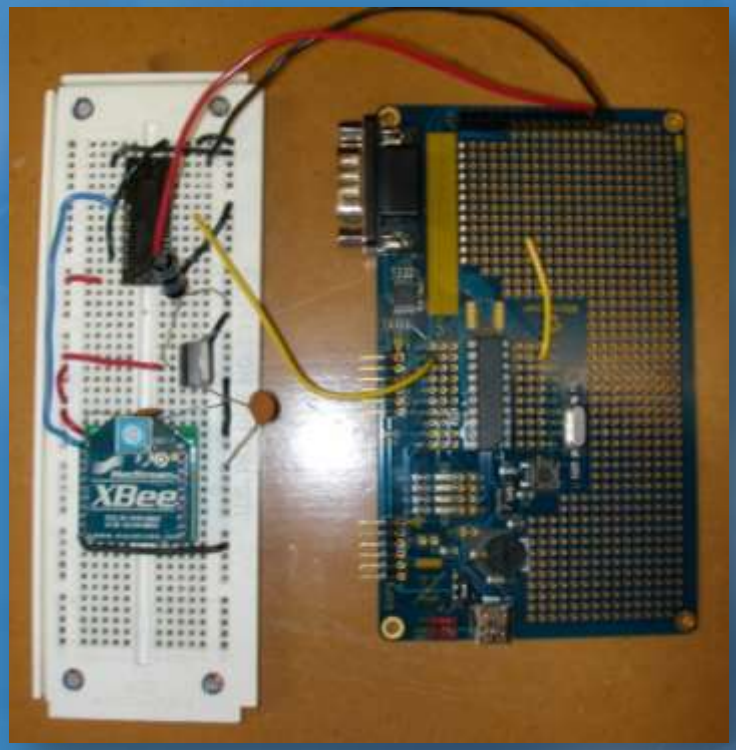


CERTIFIED USB

USB

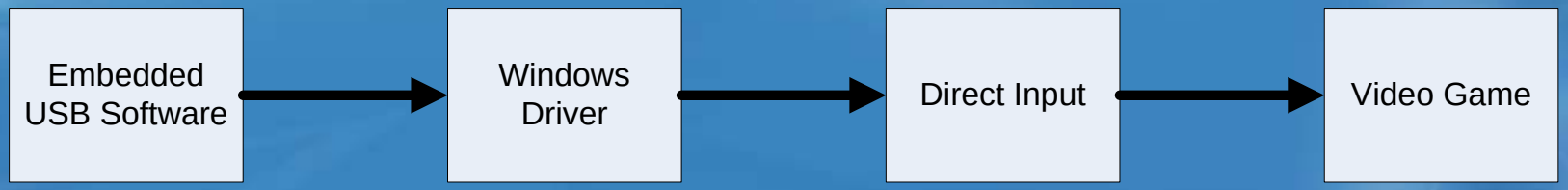
- Microchip LPC USB Dev. Kit
  - Gamepad working in one week





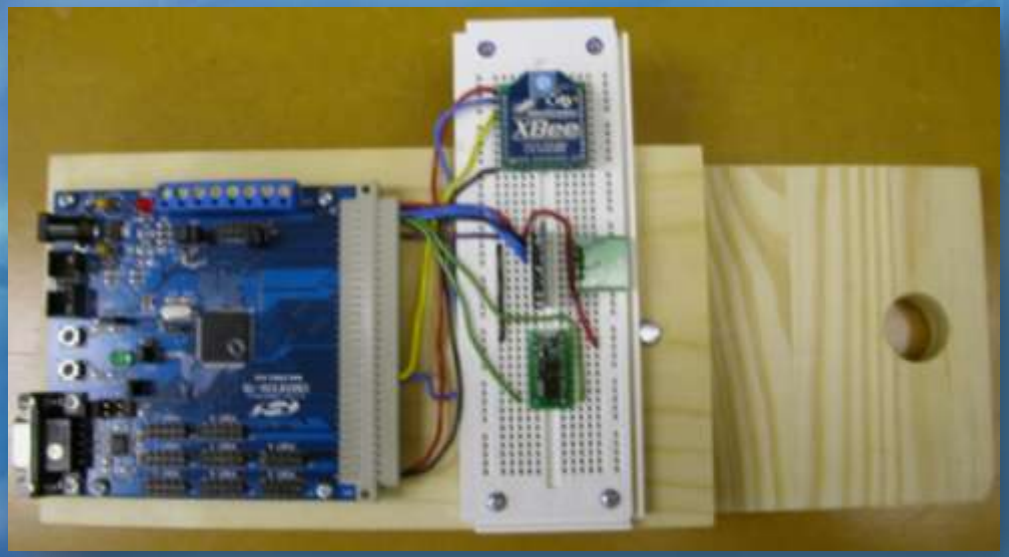
USB

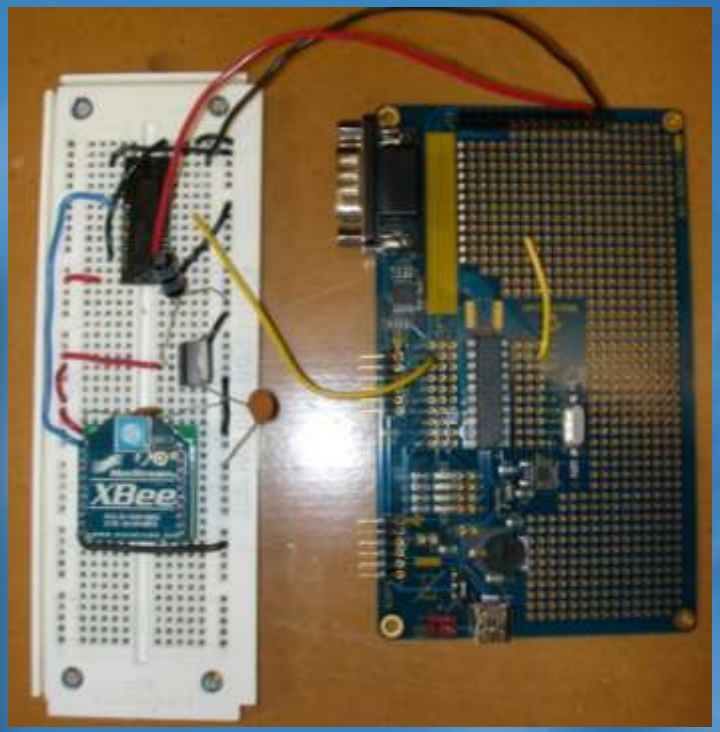
- Problems
  - Direct Input
    - Cannot actually send absolute position
    - Absolute # of units / axis → Translate to angle
    - Needs derivative = send Δ position
  - Multiple software layers / interpretations



Results

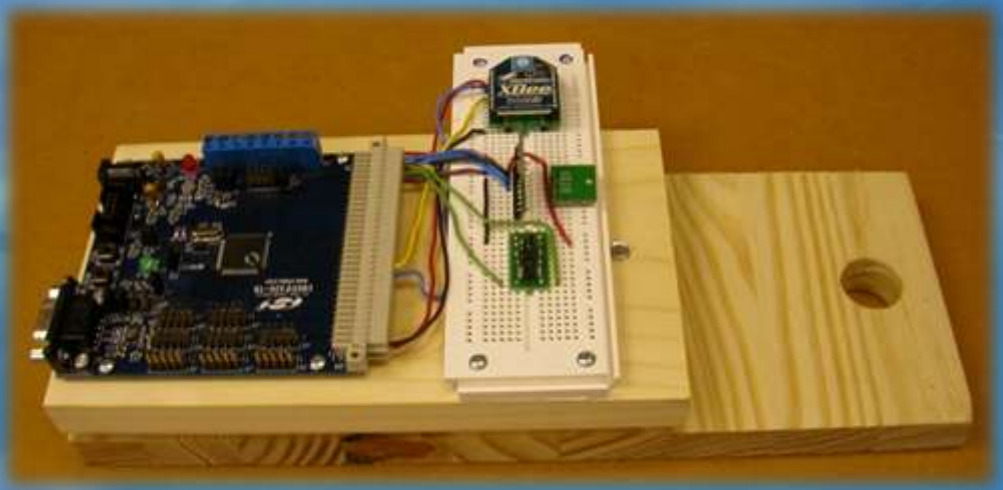
- MEMS = good / accurate position
- Software limitations
  - No absolute position
  - No software feedback

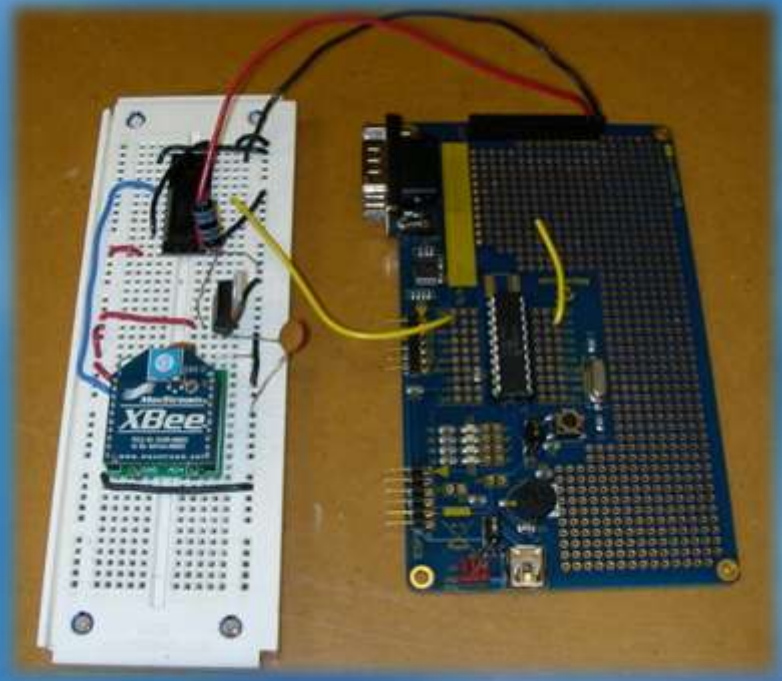




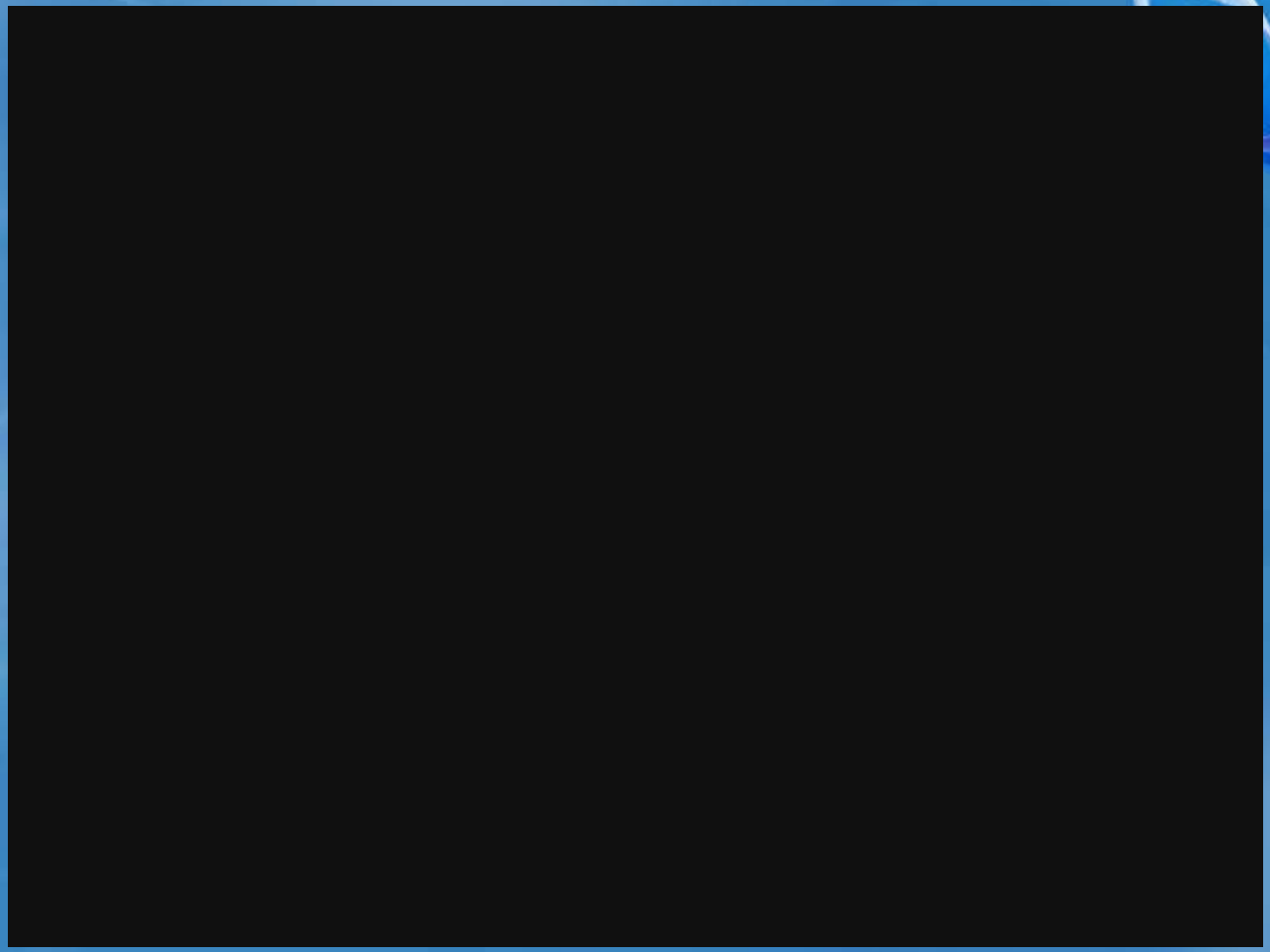
Results

- Future Suggestions
  - Write own Windows driver
  - Patch to game → Direct Input
  - Step pad / Buttons for spatial movement





Video



References

[1] Silicon Labs, C8051F34x Data Sheet,
https://www.silabs.com/products/mcu/usb/Pages/C8051F34x.aspx

[2] Wikipedia, Inertial Navigation System,
http://en.wikipedia.org/wiki/Inertial_navigation_system

[3] GeneSys Engineering Department, Inertial Sensors and Systems An Introduction,
http://www.genesys-offenburg.de/genesyse.htm

[4] D. Schertz, EE565 Fall 07 Lectures Notes 20 – 24, Bradley University

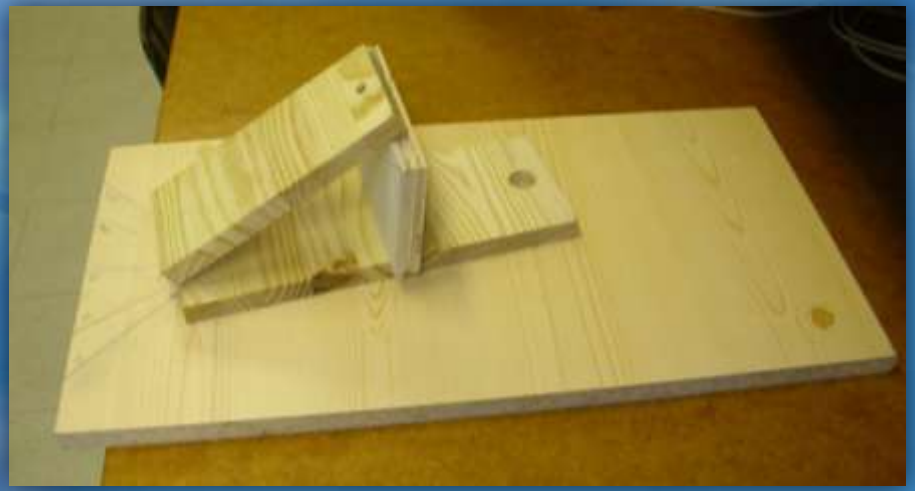
[5] Device Class Definition for Human Interface Device (HID) Version 1.11
http://www.usb.org/developers/hidpage/

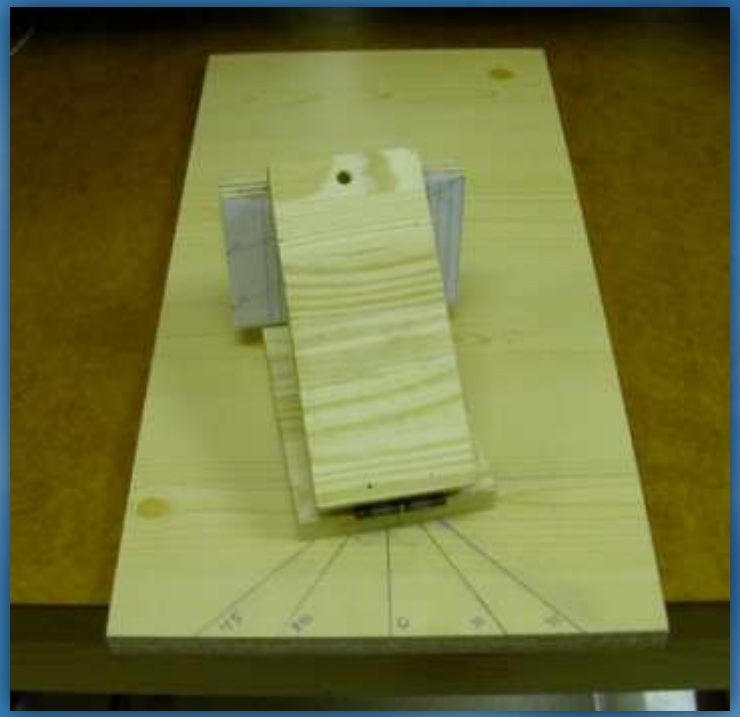
[6] HID Usage Tables Version 1.12
http://www.usb.org/developers/hidpage/

Questions

Test Platform

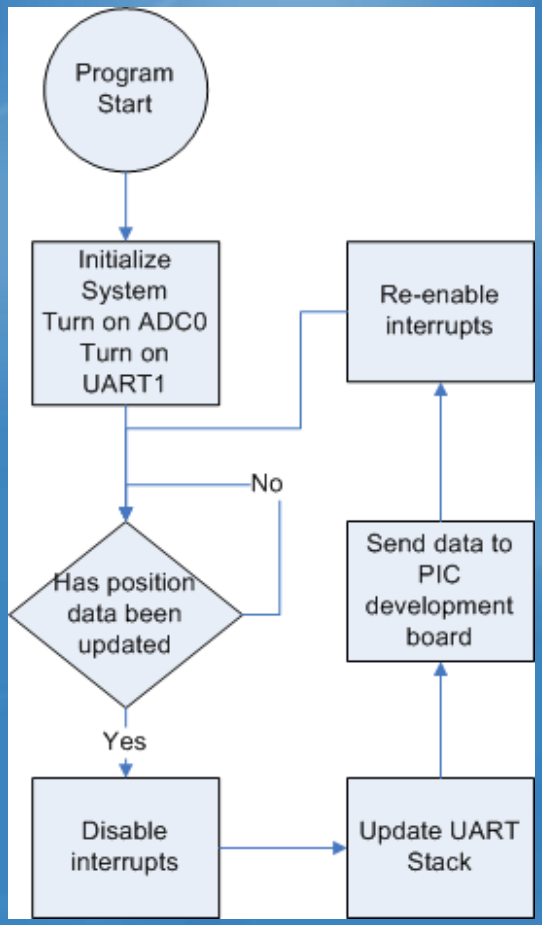
- Yaw = 0° – 360° Rotation
- Pitch = 0° – 180° Rotation





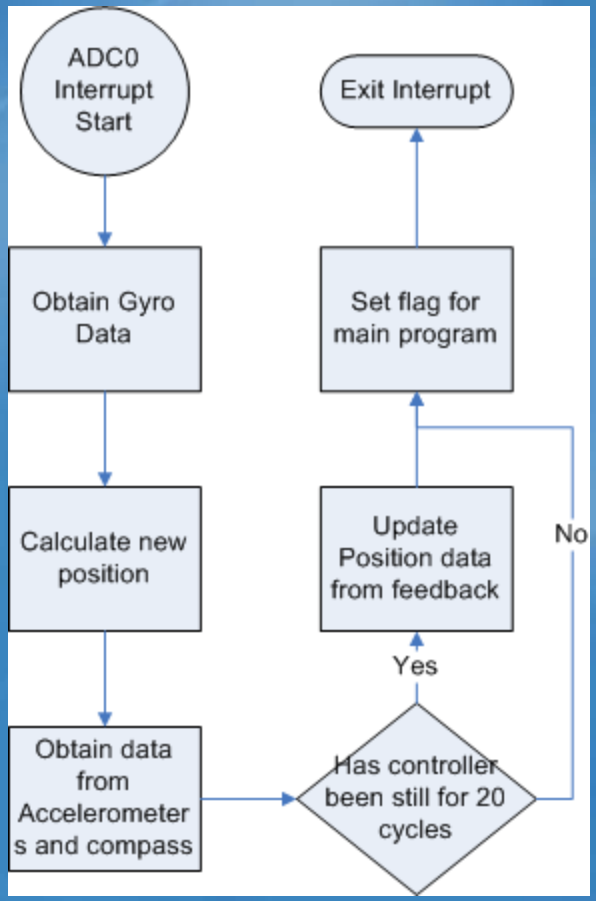
SW Flowcharts

- void main(void)



SW Flowcharts

- ADC0 Interrupt



SW Flowcharts

- USB Board