Physiballs

Handover Document

Authors : Abdelati Zelbane, Denis Tsvetkov, Eduardo Abend.

Date : 21-01-2013
Project : Physiballs Project

BEng Design Engineering:Mechatronics (year 2)
Middlesex University

1. Introduction

1.1 Purpose of the document

This document formally describes hand-over information on the Physiballs project. It is intended for review by the project manager at Red Loop design studio based in London, UK.

1.2. Scope of the document

This document describes construction of the product's open-frame prototype, specifying general known issues. This document proposes ideas for the final concept of a Physiball.

2. General Description

2.1 Device Overview

The current prototype is a fully functional, wireless board that can send, receive and represent data in real time. It is battery powered, allowing more flexibility when handled, or can be connected directly to a computer for analysis purposes. The system can also be integrated as an active sensor in other equipment. The current software model allows data output as either raw, calibrated data, or as yaw/pitch/roll values. The current data visualisation system creates graphs for all received values, while being fully customizable, e. g., size, number of graphs, or color.

A Jeenode module was chosen, because, as opposed to Bluetooth, once they have been programmed, they do not need to be paired, increasing the reliability of the co mmunication.

Testing the maximum range of the JeeNode SMD v1, the signal started to fail at a distance of approximately 30 metres in a busy indoors area with lots of phone and radio interference and trough walls.

2.2 Software documentation

With the given time constraints, it was decided that parts of another open-source firmware should be used. We chose the Razor AHRS developed by Peter Bartz and Sascha Spors. To this we have added modules for wireless radio communication and a different data structure. The data receiver and data representation modules were created especially for this project by the group members, with the help of Nick Weldin.

Real time data graphing - 3 axis and Magnitude

Code included:

- Razor AHRS(optimised for the current project);
- Data receiver module;
- Data representation module

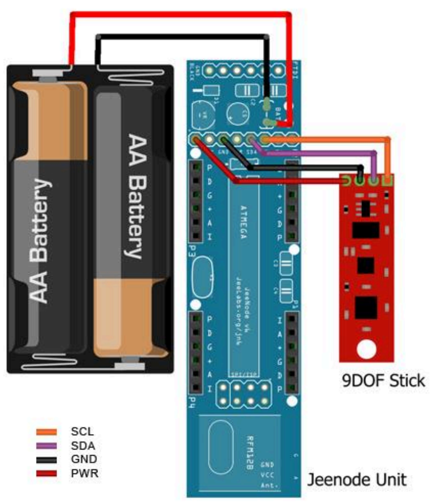
All code was written for Arduino using IDE version 1.0.3, and Processing 1.5.1(required for the data representation module).

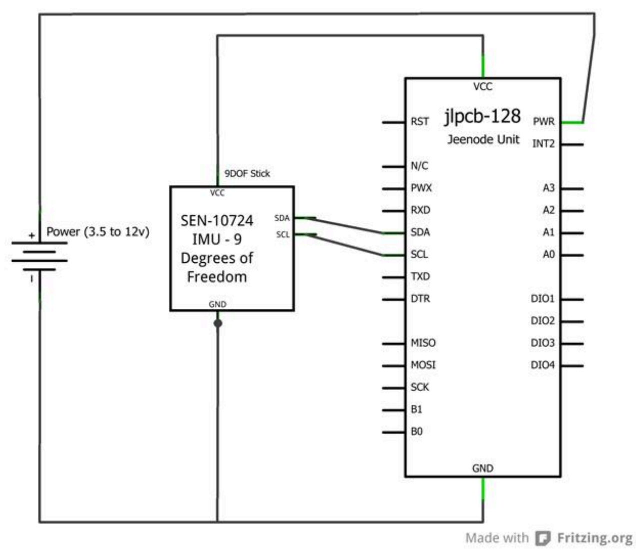
2.3 Hardware documentation

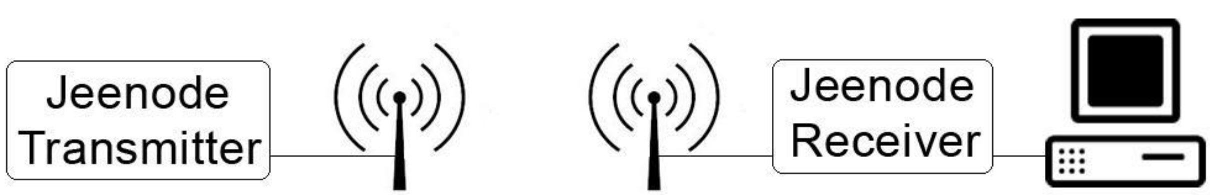
The current prototype uses the following hardware:

- Two Jeenode SMD v1 units;
- 9DOF sensor stick (Sparkfun);
- Battery pack (3.5-12v);
- USB to TTL Serial 3.3v cable;
- Host computer running the data representation module.

System diagrams:





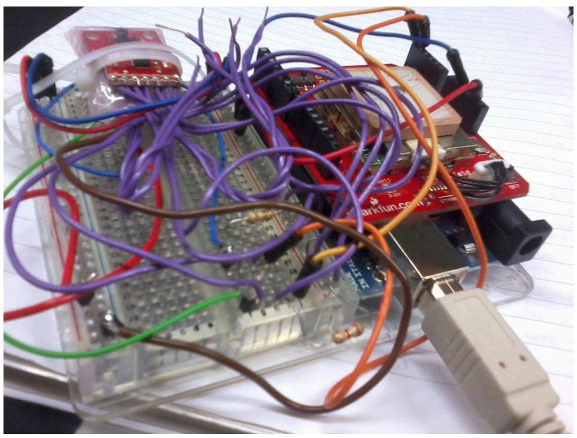


Links to all required software and documentation are included in the attachments section.

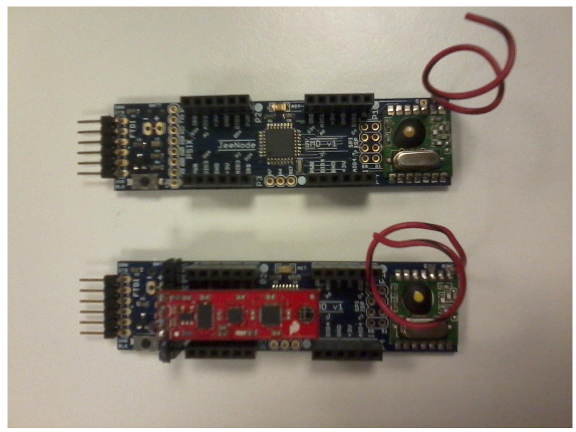
3. Known issues

3.1 Choice of hardware

Using three separate sensor boards (accelerometer ADXL345, gyroscope ITG3200 and magnetometer HMC5883L) seemed like a good choice, but several problems led us to using a single 9DOF unit.



Old open-frame model, sensors separate



New prototype, sensors together

Although three separate boards allow more flexibility with design and extra features the 9DOF lacks (e.g. access to the the interrupt pins) they also mean more than four times as much wiring will be required. Having that many connections creates a very error-prone system that is hard to troubleshoot. On the other side, the 9DOF stick has only four connections that (with a small tweak) plug directly into a JeeNode SMD v1.

There isn't a standard way of assigning axis directions on different sensors.This means that time must be taken to align, check and calibrate all three boards together every time the environment changes. The 9DOF stick, on the other side, is a unit with unified axis directions (printed on the board) and does not require re-calibrating once done, unless major changes are made to the board's environment, e.g., a metal case interfering with the magnetometer.

Before moving on to the 9DOF stick, we found some details worth mentioning about the individual sensors - the magnetometer does not work well on its own. It's easily affected by interference. Due to the way it's physically built, when it is tilted to more than thirty degrees, data received off it is very unreliable. A way to go around the issue is to connect an accelerometer to the system and use a tilt-compensation program in code. It was also found that using a higher Gauss scale resulted in slower, less accurate results, but overall the sensor was less affected by the environment. All problems are addressed and corrected when using the Razor AHRS calibration system (link in Attachments section).

3.2 Data representation

In theory, using the processed acceleration value and double-integrating it, yields relative distance. Using that and the yaw/pitch/roll values the data representation software creates a 3D object with real time movement and orientation in space. In practise though, the distance value calculated using this method contains a massive error, that is sometimes bigger than the actual distance travelled. This renders it unusable for either measurement or representation.

Visualisation of the prototype as a 3D object was designed on an old open-frame prototype using an arduino and completely wired connection. The way the code is written, in order for the data representation system to synchronize with the hardware, a two-way communication is needed. Using wireless communication, data can be represented in graphs for each axis and magnitude in real time.

3.3 Logging system and SD cards

In the course of this project, we implemented a data logging system using an SD card. While moving to a wireless platform, we encountered compatibility issues that made the task very difficult. Jeenode has its radio module connected via SPI to communicate with the onboard microcontroller.

Since the SD card uses the same connection, having both working simultaneously means that either one or the other will lose data, e.g., when information is written to the SD card, no data will be sent via radio until the operation is complete.We felt that, with the current set-up, losing data or having communication slowed down severely, defeats the purpose of real-time data representation. Therefore, a logging system is not implemented in the current prototype.

3.4 Miscellaneous

- Problems were encountered when using the latest(2.0b) version of Processing, to ensure full compatibility use the stable 1.5.1 version.
- When considering the use of a GPS unit, it is essential that the unit is used outdoors.
- In the early open-frame setup, problems with the arduino bootloader were encountered.This problem has been addressed by reprogramming the board’s bootloader firmware.
- Take care when connecting the 9DOF sensor to the Jeenode - The PWR pin provides the same voltage the battery does. Instead, use the provided 3V pin.
- Without the appropriate external casing, the system lacks extensive physical testing.
- Without calibrating the sensors in the environment they are to be used, you may encounter severe errors in the values received.

4. Future development

The next logical steps in the project would be to enhance its performance and usability by adding extra features such as a logging system for backup and a GPS unit for positioning.

4.1 Hardware changes

We would recommend an overall change in hardware as the current setup is barely sufficient for the required applications. An ideal processor choice should be able to connect and communicate with an sd card, gps unit, radio transceiver and the sensors simultaneously without the need of using serial emulation.

The creation of a pcb unit that houses all sensors with unified axis directions, a voltage regulator for the power supply, gps unit with external antenna, radio transceiver, indicators and switches would also be advised.

4.2 Data representation - position in space

Given the issues encountered with this representation system, a viable option overcoming them would be the addition of a GPS module.This does not eliminate the problem when indoors.

4.3 Data logging

To achieve a viable data logging system, we would recommend either a higher-grade data processor or a different type of wireless communication. Depending on the desired application, another option would be to completely stop transmission while logging data, or slow down communication to allow both systems to work consecutively.

4.4 Miscellaneous

- Wireless charging - the team envisioned the final prototype using inductive charging in a docking station, removing the need for cables and connectors.
- Considering data representation, a software package would be created that combines all visualisation methods in a single, customizable window with options for scientific data analysis.

5. Attachments

This section contains everything needed to further develop the current prototype, including code, documents and specifications.

5.1 Code

- Data processing and transmitter module(written for Jeenode SMD v1) Github
- Receiver module(written for Jeenode SMD v1) Github
- Representation module(written for Processing 1.5.1) Github

5.2 Datasheets

- Jeenode - Jeenode SMD
- 9DOF sensor stick - 9DOF Stick
- Accelerometer- ADXL345 (pdf)

- Gyroscope - ITG3200 (pdf)
- Magnetometer - HMC5883L (pdf)
- Razor AHRS documentation Razor AHRS Repository

5.3 Software

- Arduino Arduino Download page
- Arduino Jeenode library Jeelib for Arduino
- Processing (1.5.3!) Processing Download page