[Upper body exercise pose labeling system for injury reduction]

HARDWARE DESIGN DOCUMENT

AUTHORS – [PAOLO NARDI, JOHAN HÅKANSSON]

REVISION DATE: [2018-08-15]

REVISION: [02]

Revision History

Revision Number	Date	Comment
1.0	2018-05-28	First version
2.0	2018-08-15	Second version

Table of Contents

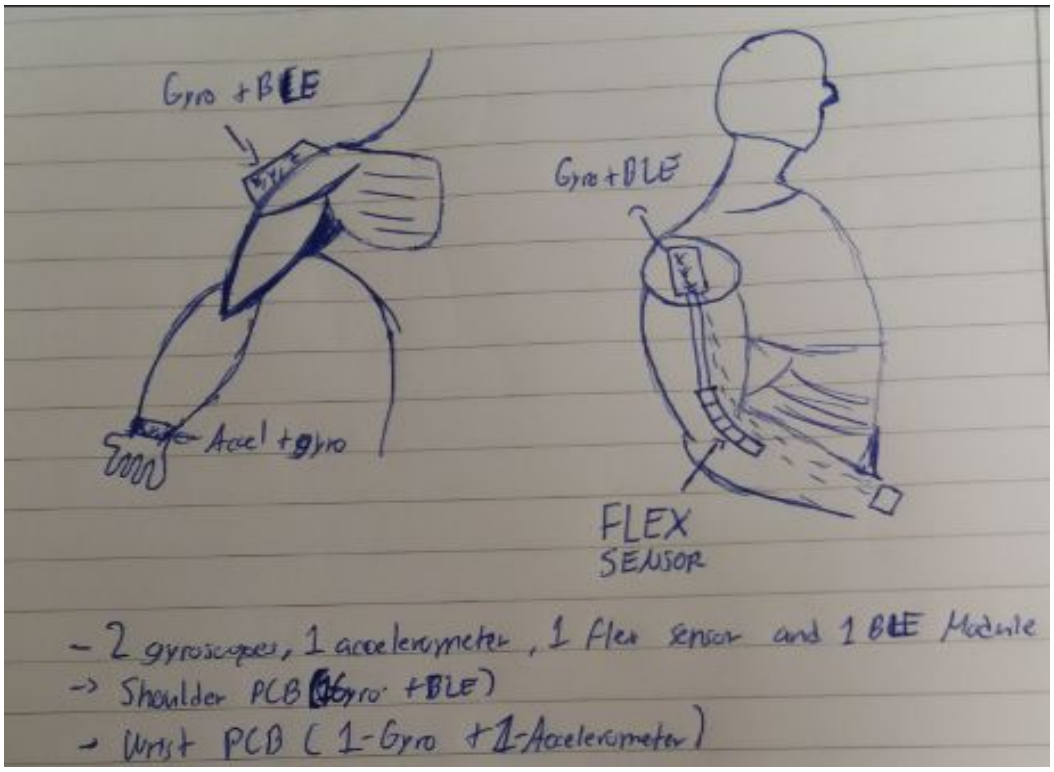
Document Overview

The product being proposed here is an upper body exercise pose labeling system. The system aims to allow researchers to gather and label how well an exercise was performed for different upper body exercises in order to reduce training injuries

By data labeling unit, we mean a data acquisition unit alongside a mobile application that allows the user to label different upper body exercises performed in a session. For example, after a user records a bench press exercise, he will be able to label the data according to the exercise performed and the “correctness” of the pose with his smartphone. The idea behind this is to have a real time pose labeling system, to allow users to reduce the likelihood of injuries while training. From the obtained labeled data, the prototype can be worked further to create a classifier which can give real-time feedback to the user upon the type of exercise and the “correctness” of the pose being performed.

This document will focus mainly the hardware needed to build the data acquisition unit and the implementation. Meaning we will discuss the schematic design and the creation of the PCBs and components used in this product, in order to allow researchers to recreate our system. The document will be broken down into the following sections: Hardware Overview, Hardware Design, Hardware Implementation and Performance Overview

HARDWARE OVERVIEW



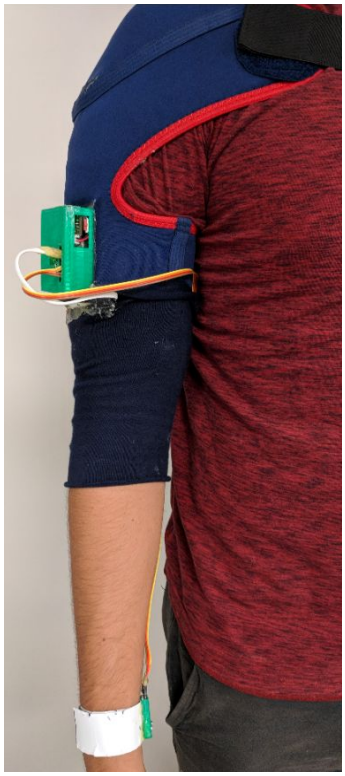


Figure 1 and 2: Data acquisition unit sensors positions schematic and prototype

The data acquisition unit proposed here acquires data from 1 accelerometer sensor located in the wrist, 2 gyroscope sensors, 1 located in the wrist and another in the shoulder and one flex sensor located in the elbow to detect the arm's flexing angle. To acquire data from this sensors we are using the Bluegiga BLE 113 Smart Module, which contains a 8051 microcontroller that can be programmed using BGSCRIPT. The BLE module is mounted on a Vulkan shoulder strap that is placed on the shoulder together with a gyroscope, in a special PCB. On the wrist the user is wearing a Pololu AltIMU-10 v4 gyro and accelerometer module, nested into a special case. This module is connected with cables to the shoulder PCB where the BLE chip is placed. The data is then sent via the BLE module to a smartphone to be uploaded to the cloud or to be graphed in real time in MATLAB. The whole system is driven by a 3.7V, 1A lithium battery.

Description

While lifting weights, can be a fun and intuitive process, many lifters (specially begginers) tend to ignore correct lifting postures. This can lead to unnecessary injuries and long term health effects. The purpose of this system is to allow researchers to gather data for correct lifting postures using different upper body exercises. With the acquired labeled data, one can then classify different exercises and categorize them depending on the correctness of the posture. This can help people who do not want to spend thousands of dollars on private trainers to get real-time feedback of their workouts. This device could also be further used for people who have recently suffered training injuries and need to perform rehab exercises.

Hardware Architecture

A Vulcan shoulder strap 3092 is used in this project to nest the shoulder PCB and sensors for monitoring the shoulder rotation. The strap is a medical posture piece that supports and relief muscles and joints without limiting movement. It also relieves pain, reduces swelling and accelerates healing. A breadboard was used to create a prototype to test if the components used in this system could allow us to obtain reliable data. On the prototype breadboard we put all the component, such as the BLE113 module, accelerometer and gyroscope. Here we tested the system, before the PCB was made to help us understand how the component worked together. The accelerometer and the gyroscope are connected with a common power, Vdd that outputs 3.3 volts, and a common ground. Both the accelerometer and the 2 gyroscopes communicate through a common I^2C line. There are two signals associated with the I2C bus: the serial clock line (SCL) and the serial data line (SDA). The latter is a bi-directional line used for sending and receiving the data to/from the interface.The slave address (SAD) associated to the accelerometer (LSM303D) is 00111xxb, whereas the xx bits are modified by the SDO/SA0 pin in order to modify the device address. If the SDO/SA0 pin is set to high, the address is 0011101b, otherwise, if the SDO/SA0 pin is connected to ground, the address is 0011110b. This solution permits the connection and addressing of two different accelerometers to the same I2C line. Over the elbow there is a Flex sensor that measures the flexing angel of the arm. This sensor simply acts as a variable resistor,

whose resistance changes as it's being bent. The flex sensor alongside a constant resistor, work together to create a voltage divider, whose output is read by the microcontroller ADC converter.

Hardware Design

Hardware Components

The hardware design criteria for this project, states that the PCB should be as small as possible, and that the positioning of the sensors is as close to the body as possible. Having the sensors close to the body, allows the readings from the sensors to be more accurate. The placement of the sensors is critical, as they need to be steady and at fixed position.

(The full component list can be found further down in the document. There you can find the components measures and manufactures information)

The main components for this project are the accelerometer (LSM303D), gyroscope (L3GD20H), the SparkFun 4.5' Flex Sensor and the Bluegiga 113 BLE smart module. The accelerometer measures the acceleration in x,y and z directions. The gyroscope measures the rotational motion in the x,y and z direction. The BLE is placed on the shoulder alongside a gyroscope.The BLE device is the central data acquisition unit, which connects to the smartphone (Android phone), to send the acquired data. On the elbow there is a flex sensor that measures the angle of the arm.

All of the components were first created in EDWinXP CAD program, where the dimensions of each component are inserted, to create correct pads, so that the components can be soldier without a problem. EDWin is a critical tool, where one must be sure that the dimensions of the components are entered appropriately, else the system will not work.

Computer Systems

For this project we used a Windows computer, to use the following programs: EDWinXP (for the creation of the PCB), NotePad++ (To program the BLE) and Free CAD (To design the 3D boxes to contain the components). To program the BLE113 chip we have used bluegiga script to flash the code into the chip. Bluegiga BGScript is a simple programming language that allows end-user applications to be embedded to the Bluegiga Smart modules, the coding is done in notepad++. The benefit of using BGScript is that one can create fully Bluetooth standalone Smart devices without the need of an external MCU and this enables further size, cost Bluetooth and power consumption reductions. Although being a simple and easy-to-learn programming language BGScript does provide features and functions to create powerful applications and it provides the necessary APIs for managing Bluetooth connections, security, data transfer and various hardware interfaces such as USB, SPI, I2C, and ADC. BGScript is a full event based programming language and code execution is started when events such as system start-up, connection,

I/O interrupt etc. occur. The Bluetooth Smart SDK comes with all the necessary tools for code editing and compilation and also the needed tools for installing the complied firmware binary to the Bluegiga Bluetooth Smart modules.

We have been using MATLAB to plot data in a graph in real time, to help us monitor the sensor readings.

The mobile applications used for this project were create using MIT app inventor. This is a very handy tool and a good complement to android studio. Android studio demands a high-end computer to work properly. After a short test with app inventor, we decided to use this tool instead of the android studio because of its easy way of building the app and it does not require a high-end computer. For more detail information regarding the software used, read the software documentation document.

Hardware Realization

Component List:

The PCB used in this system, which is placed in the shoulder, contains the following components:

- 1 Bluegiga BLE113 from Silicon Labs, with dimension 9.850mm x 16.4295mm. Link: https://www.silabs.com/products/wireless/bluetooth/bluetooth-low-energy-modules/ble113-bluetooth-smart-module
- 1 Push button to RESET the BLE113 chip. This unit has the dimension 7.747mm x 11.200mm.
- 1 ST L3GD20H Gyroscope, with dimensions 2.290mm x 3.300mm. This part is connected with 1 CC0603 10 nF capacitor with dimensions 2.252 x 1,327mm and 4 RES 0603 4.7K ohm pull-up resistors whose dimensions are 2.252mm x 1.327mm each, for proper sensor functionality.
  Link: https://www.st.com/en/mems-and-sensors/l3gd20h.html
- 1 CC Debugger unit, with dimensions 6.477mm x 7.747mm, which contains 1 RES 0603 10K Ohm pull-up resistor with dimensions 2.252mm x 1.327mm., for proper functionality
- 1 MCP1824 5V to 3.3V Linear Regulator with dimensions 7.027 mm x 7.075mm
  Link: https://www.microchip.com/wwwproducts/en/MCP1824
- 1 ON/OFF switch, to turn the device on and off, with dimension 10.127mm x 7.127mm.
- 1 MINI USB device,for charging the battery, with dimension 11.277mm x 10.277mm.
- 1 MCP73833,Stand-Alone Linear Li-Ion/Li-Polymer Charge Management Controller. This device has Included integrated pass transistor, integrated current sensing, and reverse discharge protection. It also includes safety features such as cell temperature monitoring, on-chip thermal regulation and safety timers, and the dimension is, 5.800mm x 3.277mm.

- 1 FUSE_1206Y is a safety fuse for protection of the battery, and the dimensions are, 4.6532mm x 2.2032mm.
- 1 NTC 10K Ohm RESISTOR, NTC thermistors are resistors with a negative temperature coefficient, which means that the resistance decreases with increasing temperature. They are primarily used as resistive temperature sensors and current-limiting devices, and the dimension is, 5.4102mm x 2.8702mm.
- 3 DIODE connected to the MCP73833,, which display the capacity of the battery. Each DIODE has dimensions 4.2672mm x 2.4892mm.
- 1 RES 0603 2.2 kOhm connected to the MCP7833, with dimensions 2.252mm x 1.327mm.
- 2 C_T491, 10 uF discharge capacitor, connected to the MCP78333 charging station as seen in the hardware schematics below, with dimension 5.666mm x 3.4695mm each.
- 1 3.3V, 1000 mAh Battery LiPo that supplies the circuit with power, with dimensions 6.700mm x 9,00mm
- 2 4 input List, to connect the Flex Sensor and Pololu AltIMU v4 module to our main PCB, each list with dimensions 10.49mm x 2.87mm each.
- 1 RES 0603 26K Ohm resistor connected between the input voltage VDD and the list which connects with the Flex sensor, with dimensions 2.252mm x 1.327mm.

-Alongside the main PCB, we also have 1 4.5 inches Flex Sensor from SparkFun. This sensor is located in the users elbow and is connected to 1 of the shoulder PCB's List as mentioned above. The Flex Sensor patented technology is based on resistive carbon elements. As a variable printed resistor, the Flex Sensor achieves great form-factor on a thin flexible substrate. When the substrate is bent, the sensor produces a resistance output correlated to the bend radius, the smaller the radius, the higher the resistance value.This device needs the Vdd and a ground connection to work. A resistor RES 0603 with 26K Ohm's resistance is connected in series with the flex sensor, to create a voltage divider, in accordance to the datasheet of the component. For more information regarding this sensor, look at the following link: https://www.sparkfun.com/products/8606

-The final component is the Pololu AltIMU v4 module gyro and accelerometer module, which the user wears on its wrist. This module contains 1 L3GD20H gyroscope and 1 LSM303D accelerometer. The module dimensions are 25 mm × 13 mm × 3 mm. For more information regarding the schematics and datasheets associated with this module head to the following link: https://www.pololu.com/product/2470

SCHEMATICS

-Pololu AltIMU v4 module:

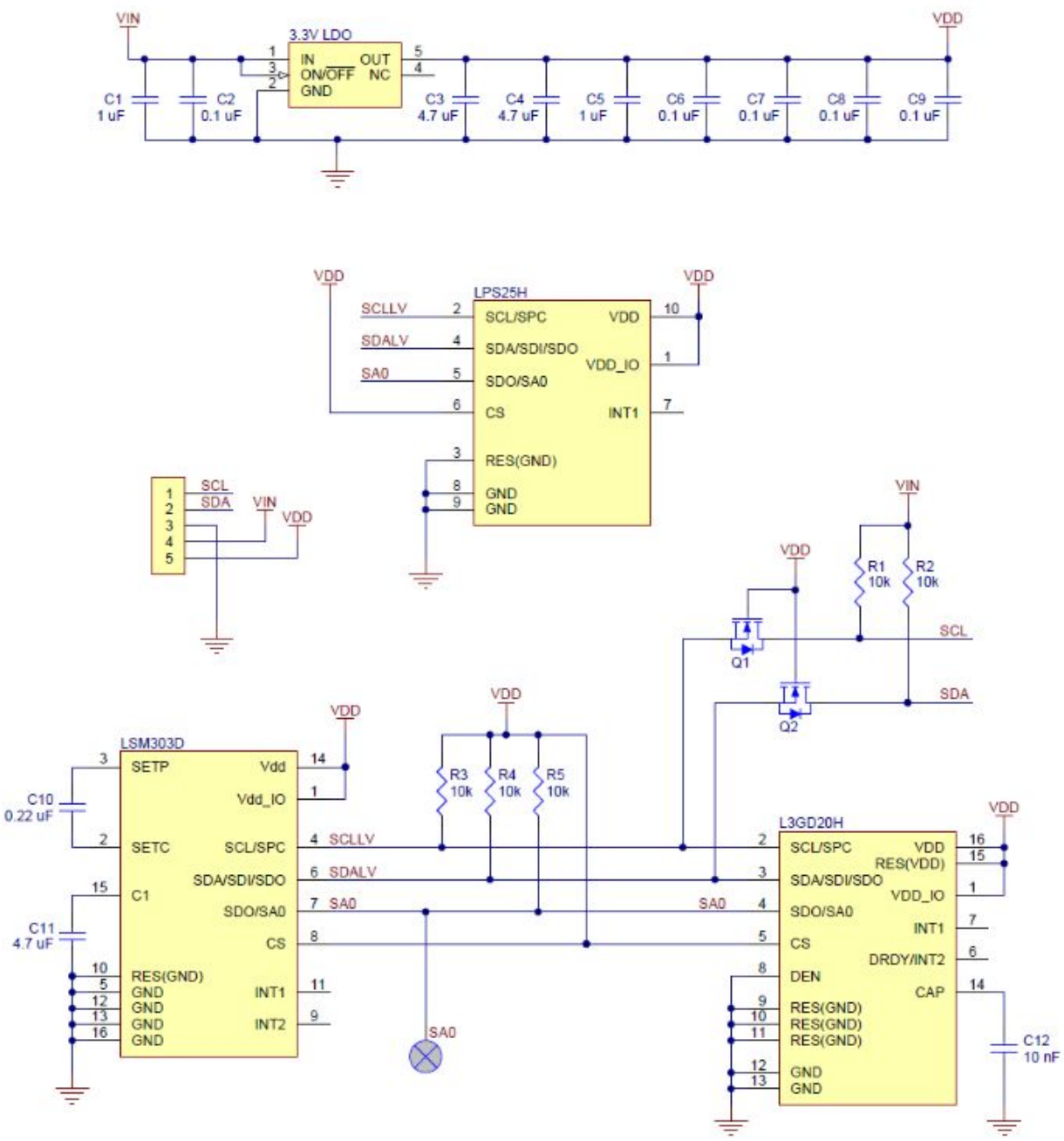
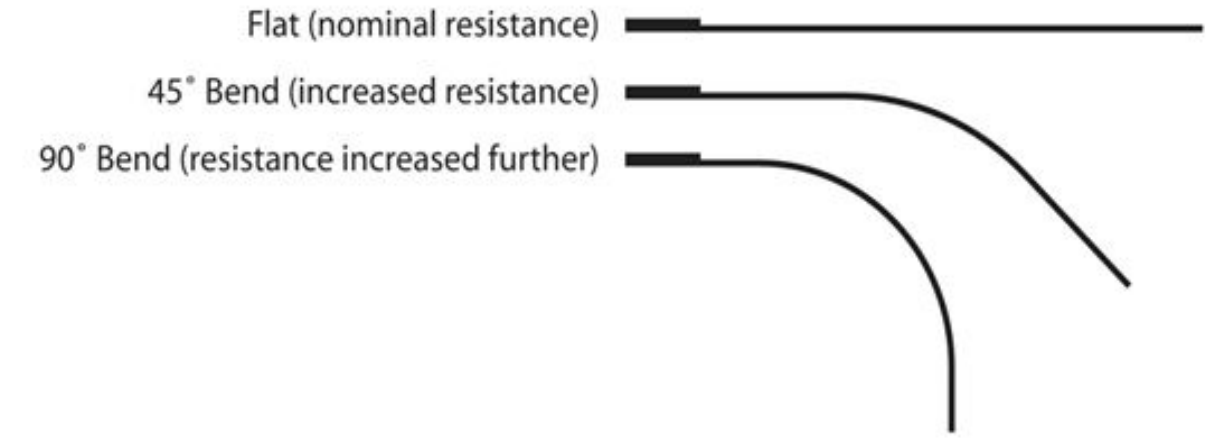


FIGURE 3. Schematic of the Pololu AltIMU v4 Gyro Module

NOTE: One can create a customized PCB for the wrist, which would contain a L3GD20H gyroscope and a LSM303D accelerometer, by following the above schematics. This can greatly reduce the cost of the product.

Flex Sensor functionality and schematics



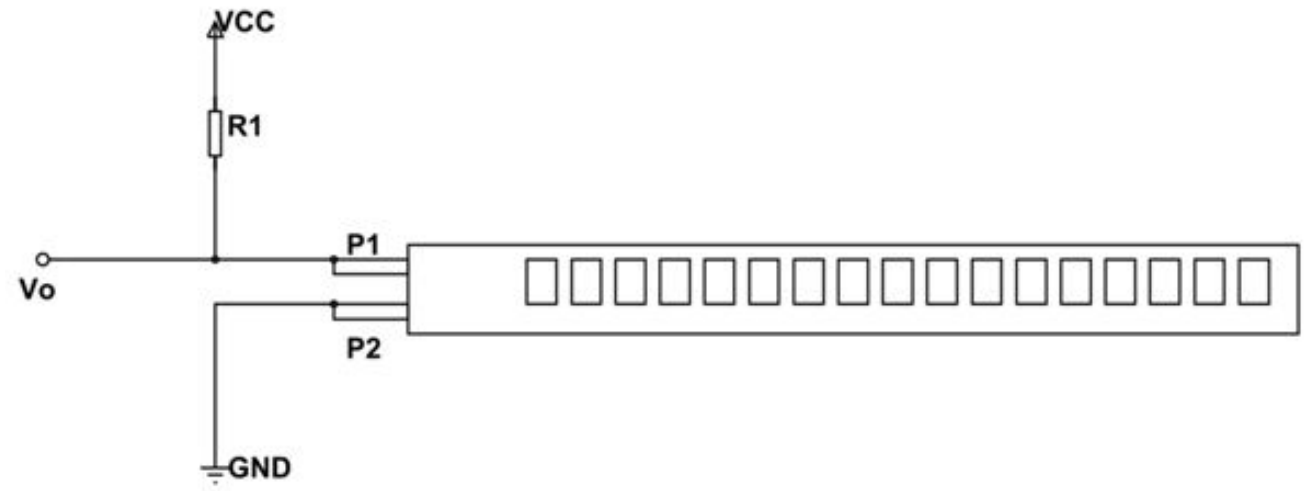


FIGURE 4 and 5. The functionality and schematic of the Flex Sensor.

NOTE: In our device, R1 is equal to 26K ohm. The value for R1 needs to be calculated for each sensor, depending on the maximum 90 degree nominal resistance of the flex sensor

Shoulder PCB schematics:

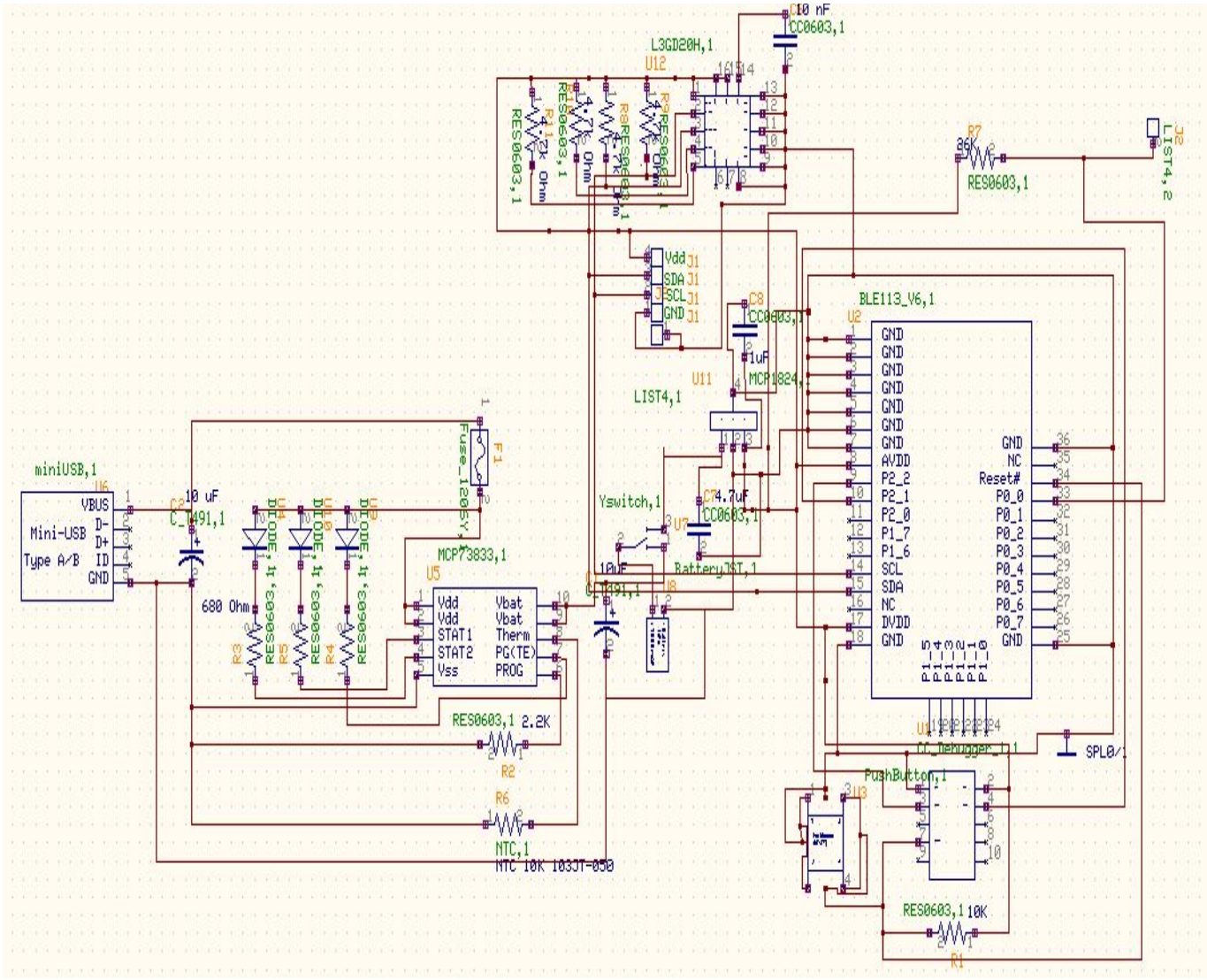


FIGURE 6. Schematic over the PCB on the shoulder.

PCB Design

Shoulder PCB Design:

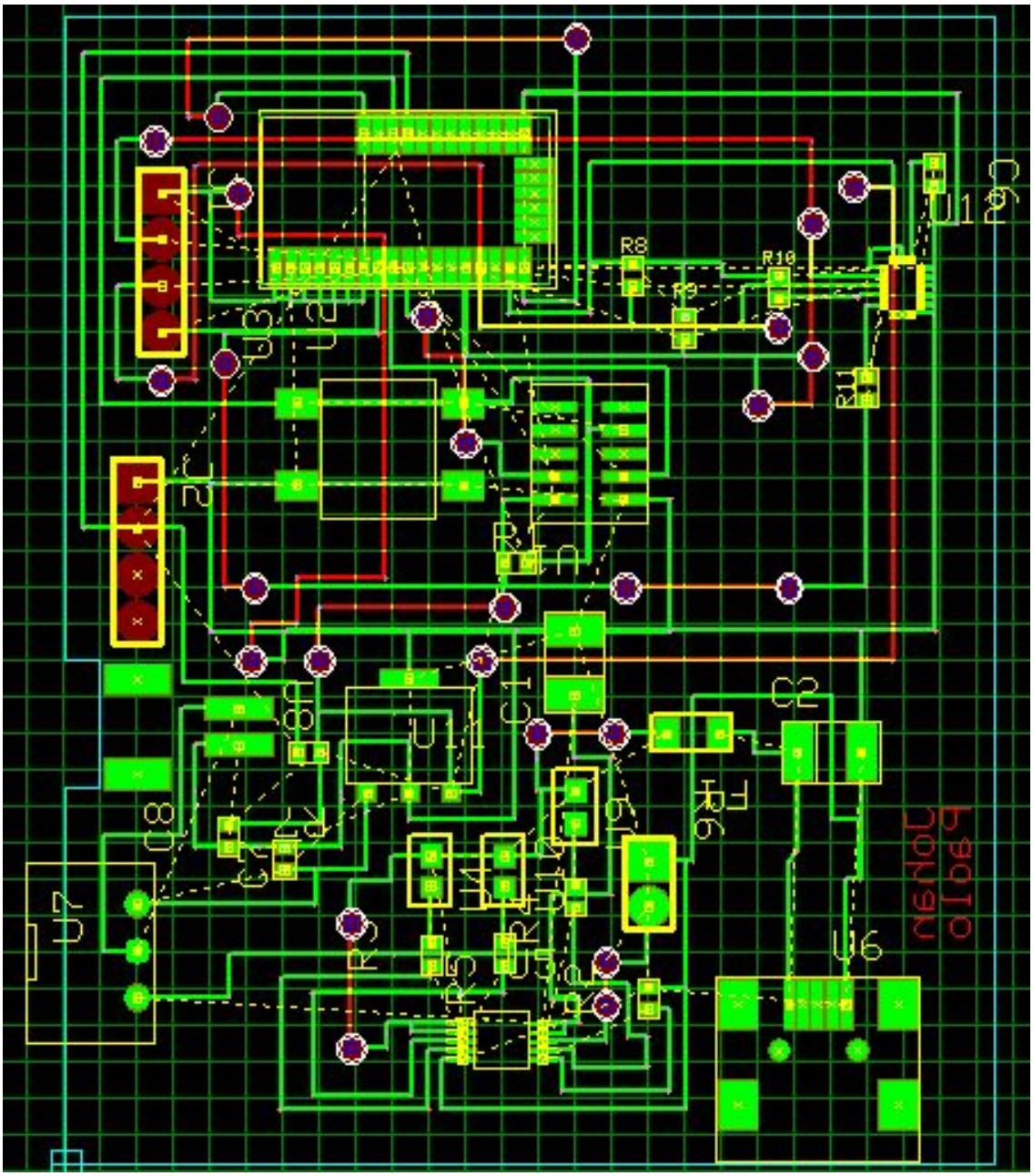


FIGURE 7: Layout of the shoulder PCB.

CASING

To protect and nest our PCB and battery, 1 case was created for the Shoulder PCB, as seen below:

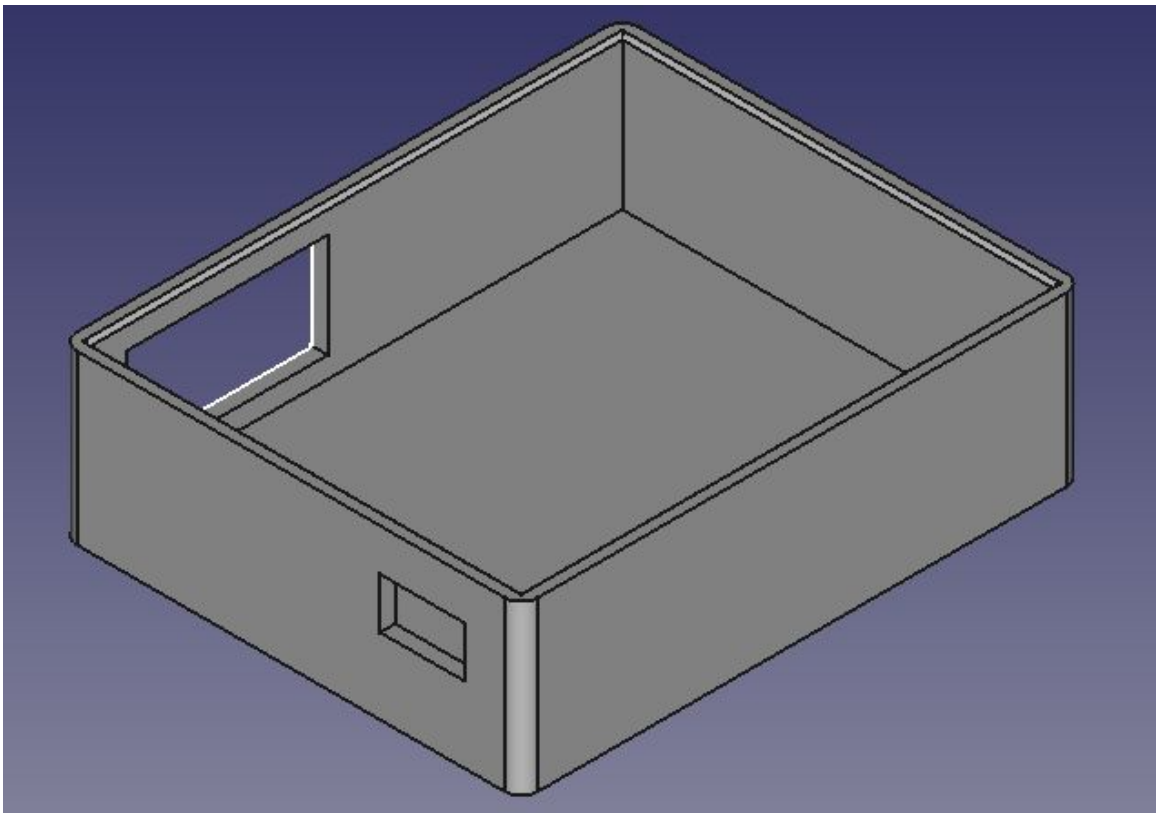


FIGURE 8: Case for the Shoulder PCB

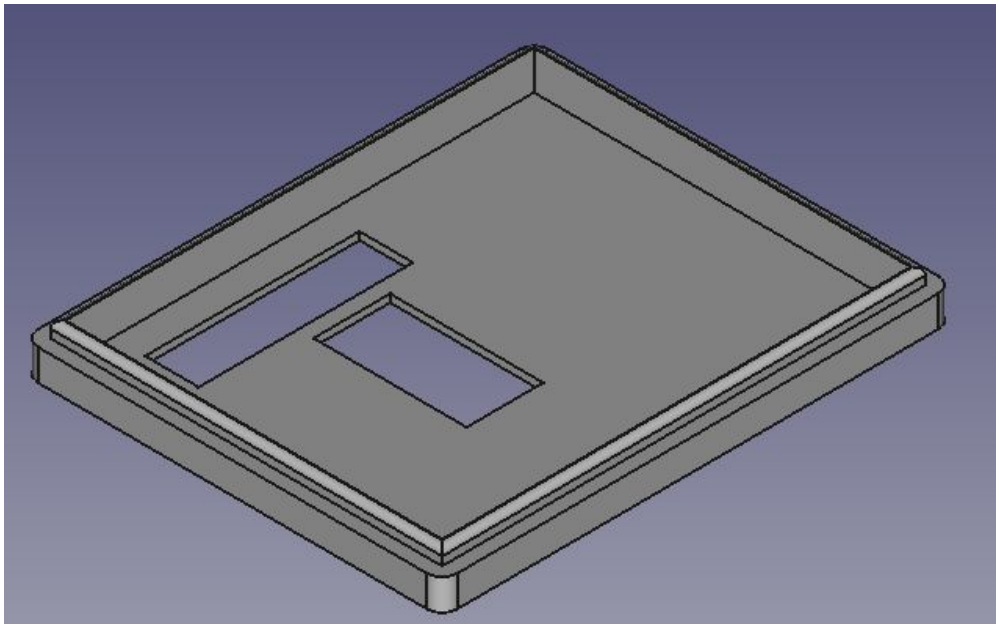


FIGURE 9. Cover for the Shoulder PCB

To protect the Pololu AltIMU module a case was created as seen below:

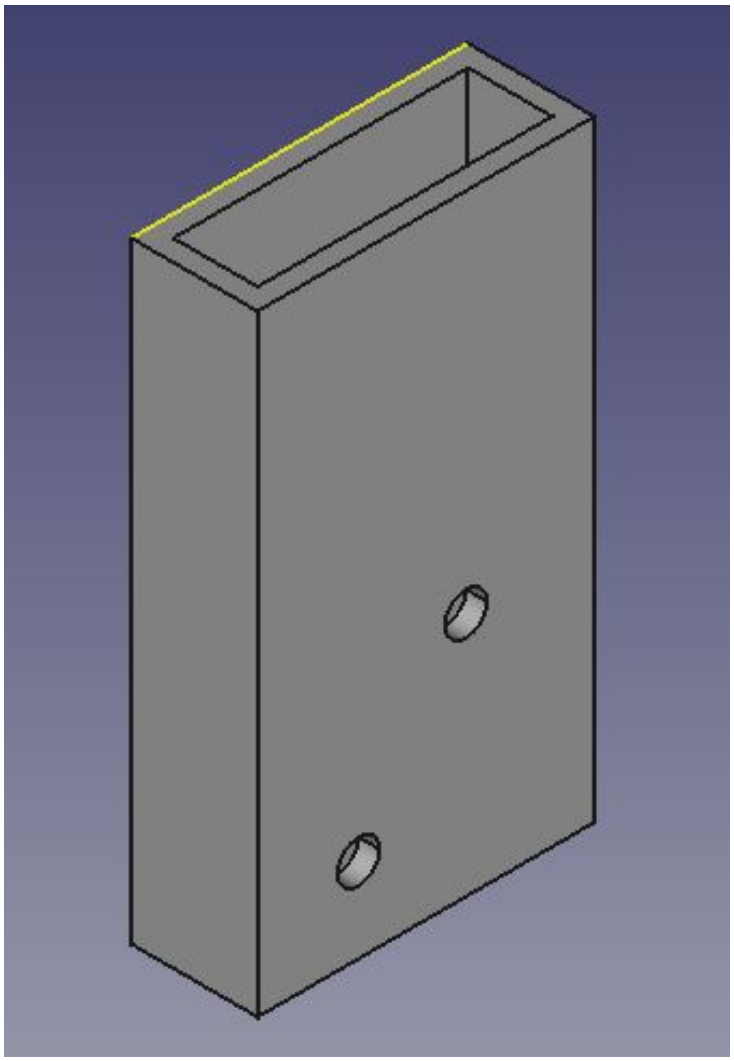


FIGURE 10: Case for the Pololu AltIMU v4

Finalized Hardware

Once the PCB was created and the components were soldered, the PCB was put into its case as seen below:

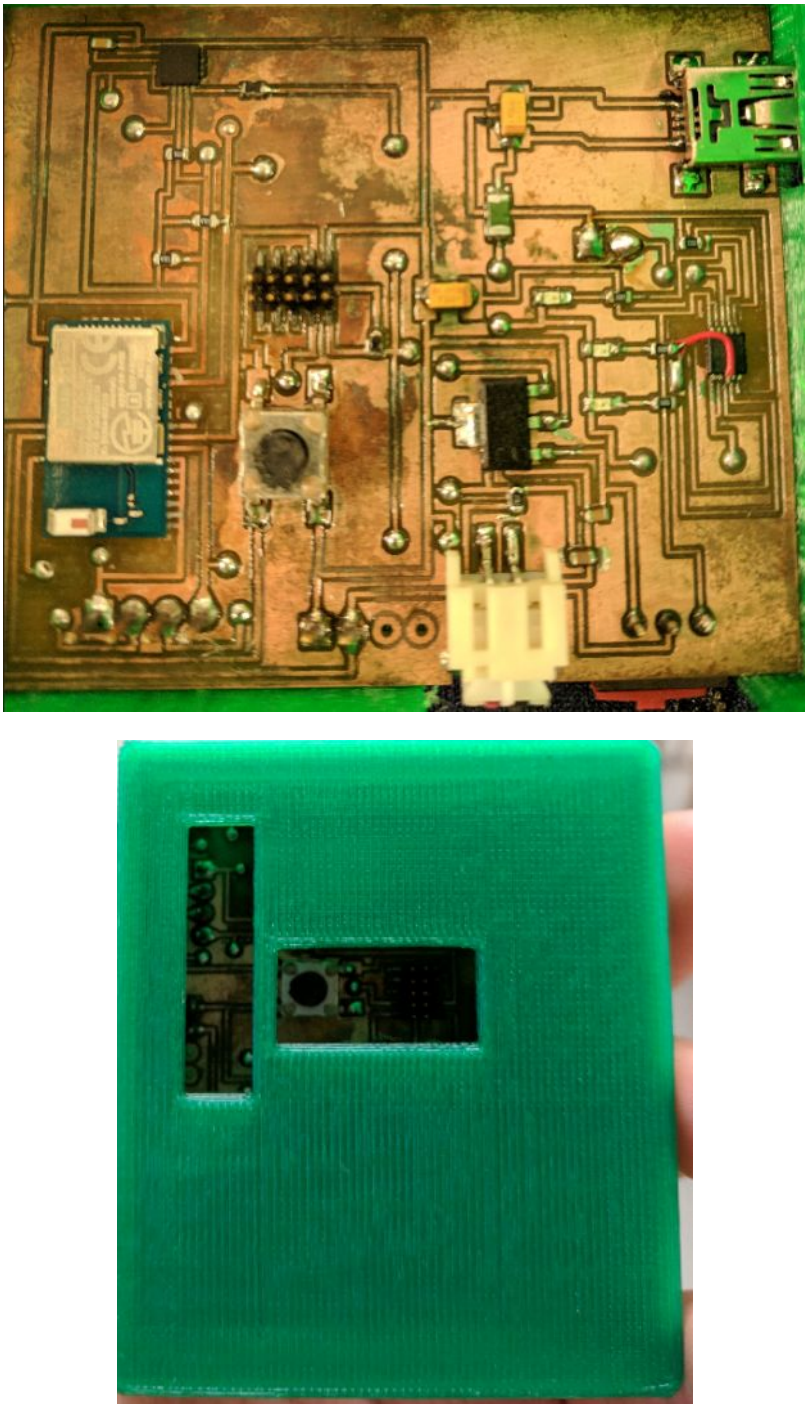


FIGURE 11-12: Data Acquisition unit PCB and casing integration

Then the Pololu AltIMU module was put into its case and glued into a wrist strap, as seen below:

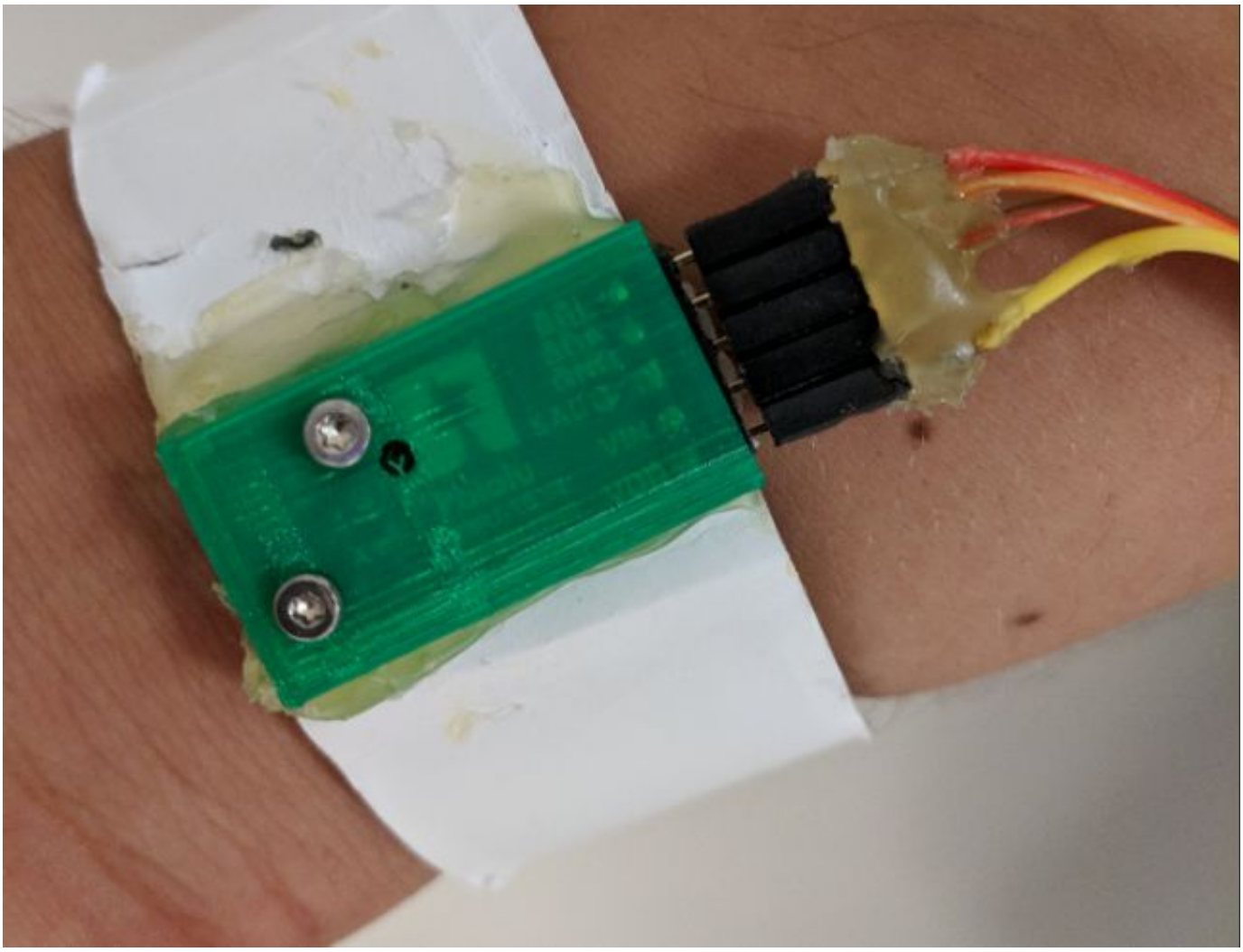


FIGURE 13: Pololu Module finalized.

NOTE: In order for the device to work properly, connect the yellow cable part from the 5 input list located alongside the wire that connects the PCB and the Pololu Module, into the Pololu Module VDD input as seen in the picture above!

The next step was to connect all the parts together with wires, with the Shoulder PCB List's connections as seen below:

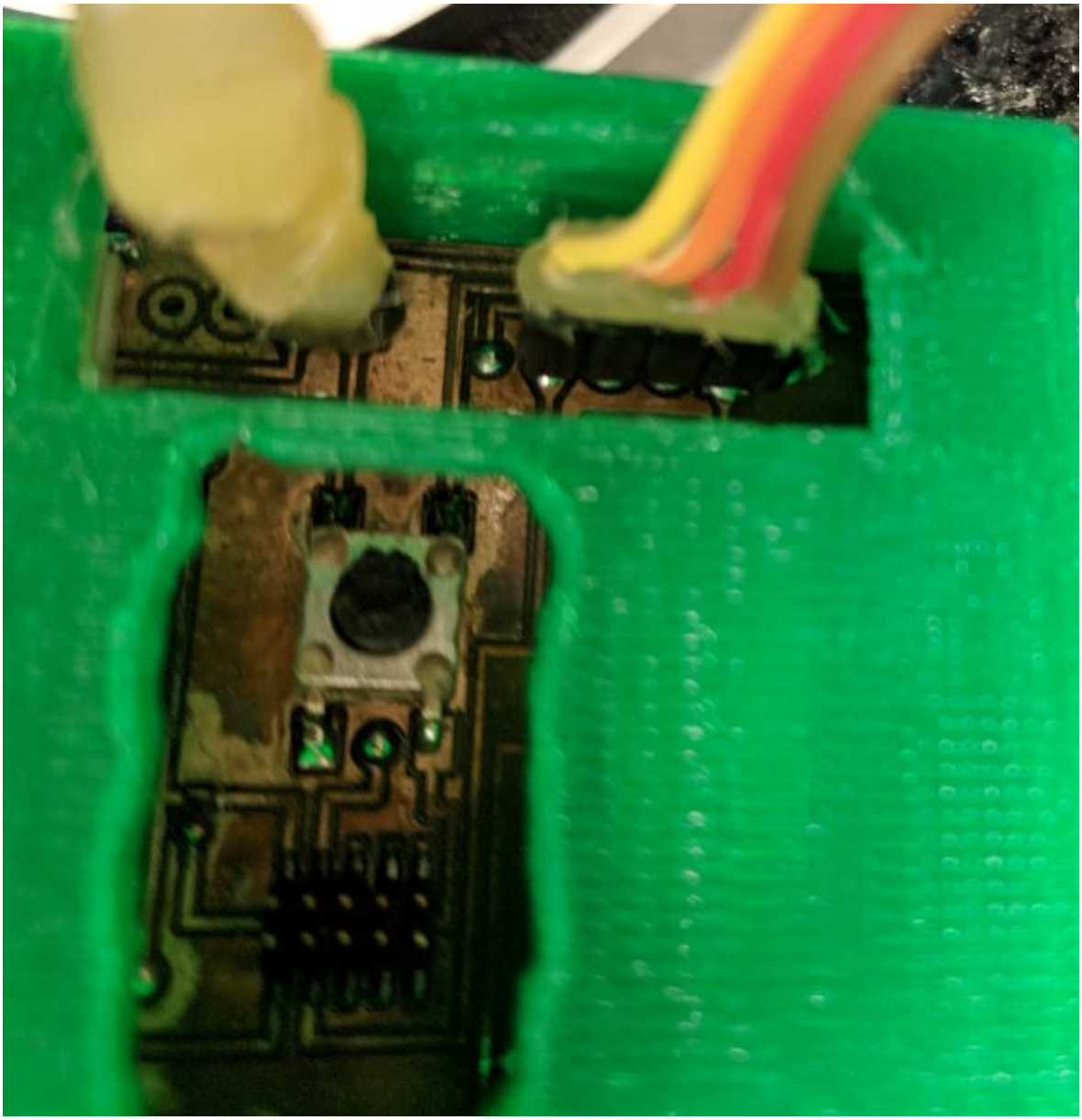


FIGURE 14: Connections to the main PCB

NOTE: In order for the device to work properly, connect the yellow cable part from the 4 input list located alongside the wire that connects the PCB and the Pololu Module, into the VDD input of the PCB as seen in the picture above!

The last step was to assemble everything together, to have the finalized prototype as seen below:

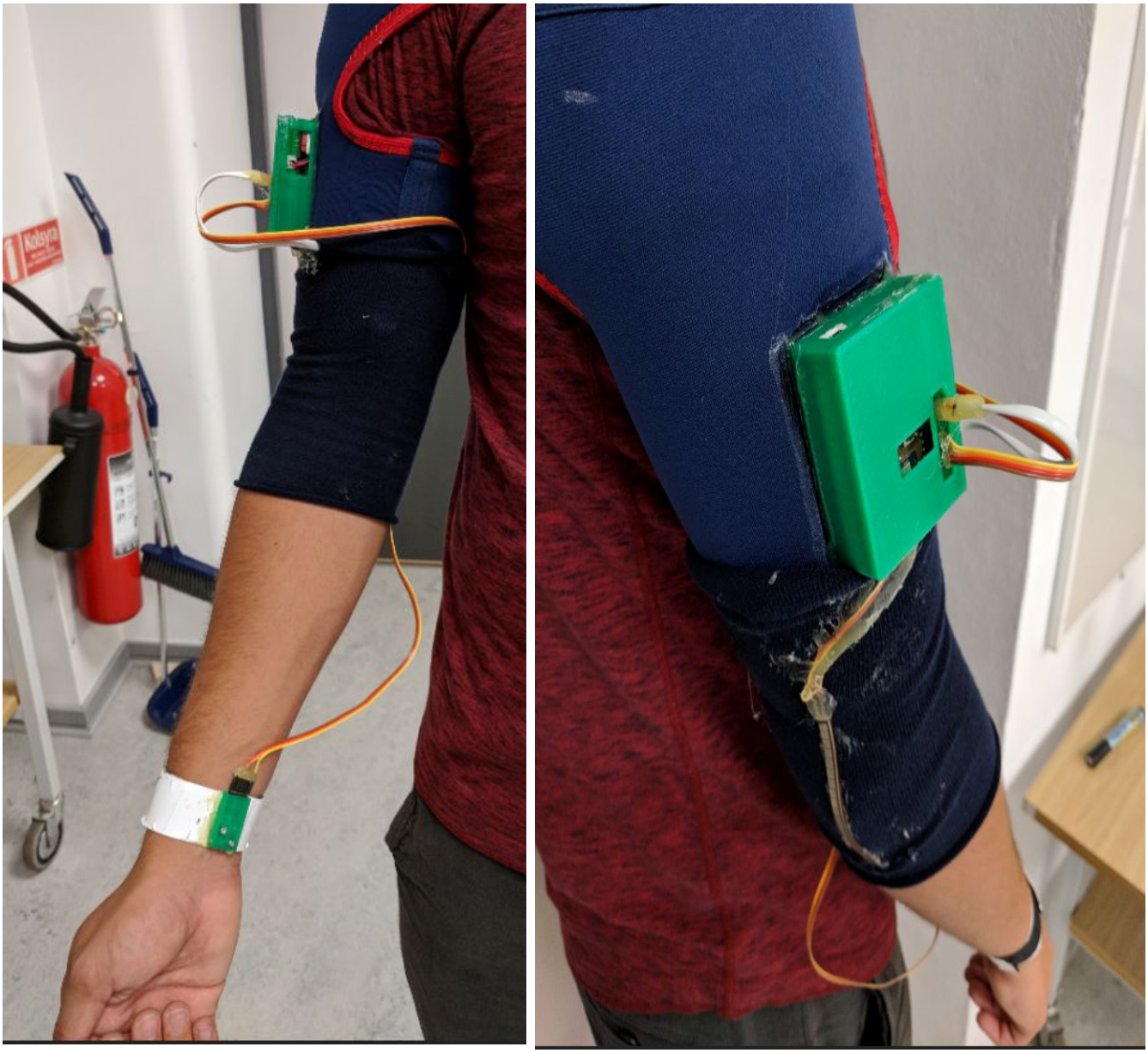


FIGURE 15-16: Final Hardware implementation

HARDWARE FUNCTIONALITY

Device power switch, off/on position:

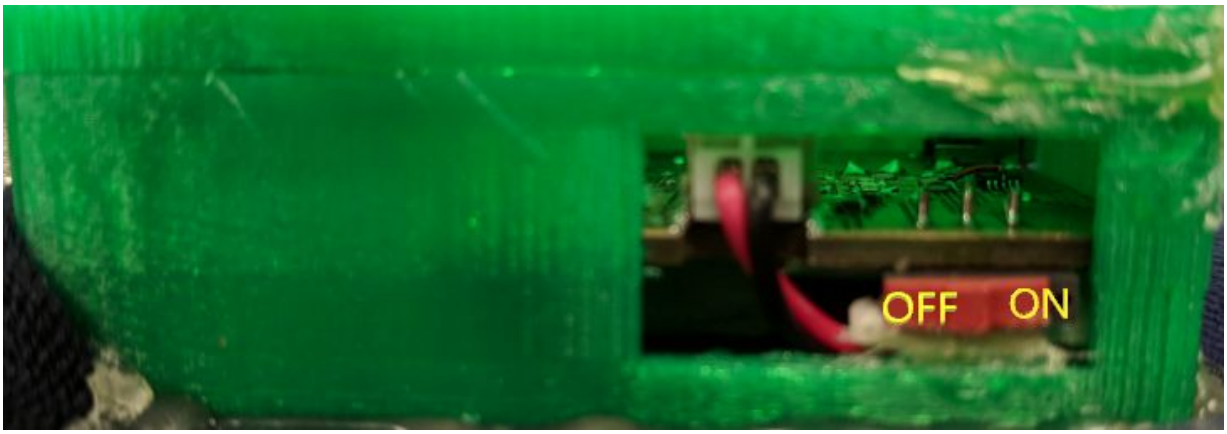


FIGURE 17: OFF/ON Switch Placement

When trying to upload code to the device, please ensure the debugger is connected as seen below:

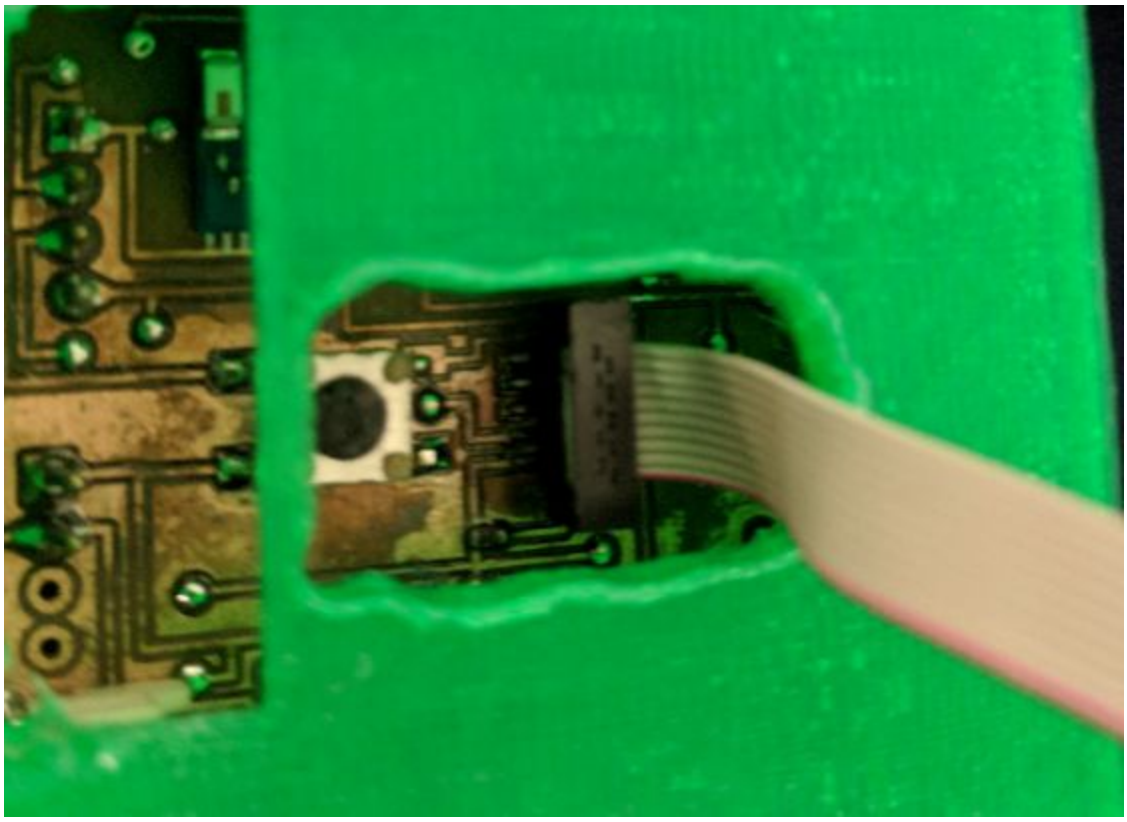


FIGURE 18:CC Debugger cable positon

For the debugger to work properly, ensure that the device is powered on and that debugger has been connected properly. One can flash new code into the device, when the debugger's light is set to green, as seen below:

FIGURE 19:CC Debugger ready to flush code

If the device is powered off, or there's an error in the connection, the Debuggers light will be red, as seen below:

FIGURE 19:CC Debugger not ready to flush code

Once the connections have been fixed, one can press the reset button. If the light turns green, the debugger can flash code into the device, else check all connections.

PERFORMANCE

POWER CONSUMPTION

This is the power consumption table over the components that has been used in the project.

Component:	Supply Current:
1,“4.5” FlexSensor	0.50 Watts continuous -Works as a voltage divider 100k resistor+flat= 33µA and voltage = 1.14V (Tested 100k resistor+90° bend = 33µA and voltage = 1.8 V (Tested)
2,Gyroscopes (L3GD20H) 1 Accelerometer (LSM303D)	6mA each -> 18 mA
1, Bluegiga BLE113 Bluetooth® Smart Module	TX Peak: 18.2mA RX Peak: 14.3mA Sleep: 0.4uA

TABLE 1. Component power consumption.

Bluegiga BLE113 Bluetooth® Smart Module

-TX peak current: 18.2 mA

-TX peak current: 18.2 mA (with DC/DC) -RX peak current: 14.3 mA

-Sleep mode current: 0.4uA

-Operating voltage: 2-3.6 V

-Power per 3 hours at 3.3V-> (18.2 mA+14.3 mA+0.4 µA)*3.3 = 0.10725W*3h = 0.3217Wh

Gyroscope, Accelerometer, (L3GD20H, LSM303D.

-Operating voltage: 2.2 to 3.6 V

-Supply current: 6mA

Power needed for 3 hours at 3.3V -> (6mA)*3.3=0.020 W*3h = 0.06Wh

Flex sensor

-Operating voltage: 3.3V

-Supply current: 33uA

-Power needed for 3 hours at 3.3V -> (33uA)*3.3= 0.189mW*3h= 0.98mWh

The device is meant to operate in sleep-mode for the majority of the time.

When the user wants to start recording a pose, the uC shall turn on, gather the data for as long as the user wants to record a specific pose, then send the data and then go back to sleep mode again.

Since this is a pose labeling system, a normal pose recording session will go anywhere from 1 second to 1 minute. The reasoning behind this is, to acquire liable data of the pose and be able to label it's "correctness". Meaning a labeler would not want to capture a whole set of an exercise, but around 2-4 reps.

A normal upper body workout session goes for around 1-2 hours, which approximately 30 to 50 minutes are spend in actual exercises and the rest is spend taking small breaks. With this being in mind, the worst case scenario of battery usage takes into account approximately 180 minutes or 3 hours of full recording time.

For this project we are using a 1A lithium battery with a nominal voltage output of 3.7V -> 1000*3.7/1000 = 3.7 Wh -> Our battery capacity

For our total of 3 hours of full recording time a week:

We have approximately -> 0.3217+0.06+0.04 + 0.98x10^-3 + (0.0604x2)

Approximately = 0.54Wh

3.7/0.54 = 6 -> Meaning we can approximately record 6 sessions of 3 hours.

In overall we will save approximately (Power per 3 hours at 3.3V-> (4 mA+0.1µA+0.75µA)* 3.3V = 0.0132W *3h = 0.04Wh))

AVAILABILITY

The device is meant to operate in sleep-mode for the majority of the time, except when recording data. When the user wants to start recording an exercise, the microcontroller shall turn on, gather the data for as long as the user wants to record a specific pose, send the data to the mobile application and then go back to sleep. Since this is a pose labeling system at the moment, a normal pose recording session will go anywhere from 1 second to 1 minute. The reasoning behind is, is to acquire liable data of the pose and be able to label it's "correctness". Meaning a labeler would not want to capture a whole set of an exercise, but around 2-4 reps. A normal upper body workout session goes for around 1-2 hours, which approximately 30 to 50 minutes are spend in actual exercises and the rest is spent taking small breaks. For this project we use a 1A lithium battery with a nominal voltage output of 3.7V -> 1000*3.7/1000 = 3.7 Wh.

We is using the sample rate of 25Hz. To gather as much data as possible.