MARO
OptiFit

March 9, 2016

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6, Canada

Re: ENSC 305W/440W Design Specification for OptiFit: A Powerlifting Monitor & Warning System

Dear Dr. Rawicz:

The attached document, *Design Specification for OptiFit: A Powerlifting Monitor & Warning System*, outlines the specific design information for our Capstone Engineering Science project. The team at Omaro is designing and implementing OptiFit, a system that monitors physiological data during powerlifting, in addition to providing the user immediate feedback on their form.

The purpose of this design specification is to provide system-level design as well as separate breakdowns of the hardware, software, mechanical and embedded components. Moreover, this document provides design references for the proof-of-concept version of OptiFit only; details about future iterations will be discussed to help guide future development as well.

Omaro consists of four final-year students: Amid Sedghi, Chris Esterer, Henry Hein, and Jarid Warren. If you have any questions or concerns about our submission, please feel free to contact me via email at jaridw@sfu.ca.

Sincerely,

Jarid Warren
CEO
Omaro GP

Enclosure: *Design Specification for OptiFit: A Powerlifting Monitor & Warning System*

Design Specification

for

OptiFit: A Powerlifting Monitor & Warning System

Project Team: *Amid Sedghi*
Chris Esterer
Henry Hein
Jarid Warren

Contact Person: *Jarid Warren*
jaridw@sfu.ca

Submitted to: *Dr. Andrew Rawicz – ENSC 440W*
Steve Whitmore – ENSC 305W
School of Engineering Science
Simon Fraser University

Issue Date: *March 9, 2016*

Revision: *1.0*

Acknowledgement

Omaro would like to thank the DUST networks team from Linear Technology Corporation (LTC) for graciously providing us with a development kit of their high-end industrial networking solution. In particular, we would like to mention SFU alumnus Gordon Charles, Dr. Brett Warneke, Ross Yu, and Dr. Jonathan Simon.

Abstract

Powerlifting and bodybuilding have seen an explosion of research in the past decade as more and more North Americans are prioritizing their health, fitness, and performance.

Unfortunately, the advanced tools used to measure the physical performance of athletes are lacking in both the commercial and educational sectors. Research and education tools are broad in terms of raw data, but narrow in terms of analysis and user feedback. Products available directly to consumers meanwhile are over-priced, bulky and usually only give the user one biometric piece of data.

OptiFit, from Omaro, will give users a large array of data in addition to automated analysis that can warn the user should their form be dangerous or incorrect. To give analysis during a user's lift, our software package will take advantage of three sensors: an electromyography (EMG) to measure muscle activity, a force pad to calculate force delivered to a barbell, and a Microsoft Kinect to track body mechanics.

This document includes high and low-level design specifications for the implementation of OptiFit, including the hardware, Win32 software, embedded software, and mechanical components that comprise the system. It also includes future considerations for the commercial version, design justifications and references functional requirements for the system. A high-level test plan is also included in Appendix A to help quantify the performance of the product at the release of the proof-of-concept.

Table of Contents

List of Figures & Tables

1. Introduction

OptiFit is a complete physiological monitoring system designed to bridge the gap found in current research tools and commercial tools by offering more insight and safety for the user.

Currently on the market, researchers have the ability to track parameters such as muscle activation, force generation, and mechanics using infrared (IR) sensors; however, these pieces of data are all collected separately and analyzed after the movement has been performed. Despite the fact that the data across sensors is intimately related, research tools have no way of automating this process. Commercially, on the other hand, the tools are either limited in capability, marketed towards consumers or include obtrusive wires that impede movement.

OptiFit, as shown in Figure 1, combines the data from muscle activation, force exerted, and skeletal mechanics to give the user contextual information about their movement and issue real-time feedback should their technique be poor.

Figure 1: Mock-up of OptiFit in use.

1.1 Scope

This document outlines the design specifications that OptiFit will adhere to. It contains detailed information about the implementation of hardware, software, embedded systems, and mechanical components while justifying design choices over other alternatives. For the proof-of-concept to perform as expected, it will follow the exact design outlined here. Unless otherwise noted, this document pertains to the proof-of-concept version only. The Design specification also includes a test plan to help evaluate the performance of the working system.

1.2 Intended Audience

This document is intended to be referenced by members of Omaro throughout development to serve as a guideline for implementation. It also serves the purpose to bring future engineers, testers, and investors up to date as far as the details of the project are concerned.

2. System Overview

OptiFit is designed to be used in a research lab or physical therapy office. The user will wear the Electromyograph (EMG, to measure muscle activity) around their chest with a Velcro strap and the operator will attach the electrodes. The operator will then start the software while the user is standing on the force pad (force plate and force sensor are also used and are interchangeable in this document). After the operator has pressed start, the user will perform their movement and will only be interrupted should the operator stop OptiFit or if their form is poor; in that case, they will receive audio feedback.

It should be noted that the proof-of-concept version will monitor the deadlift alone whereas future versions will expand their scope to other movements via software. To monitor muscle activity with the deadlift the major area of focus will be the lower back or more specifically, the erector spinae.

The EMG sensor and force pad will talk wirelessly to the operator's personal computer (PC) via a DUST wireless network manager connected to the USB2/COM port. A Microsoft Kinect will be aimed at the user and recording their mechanics during their movement. The Kinect will in turn communicate with the PC via a USB3 port. A Win32 program running on the PC will collect the data from hardware, analyze it, display the information via a Graphical User Interface (GUI), and issue appropriate warnings. The block diagram of OptiFit can be seen below in Figure 2.

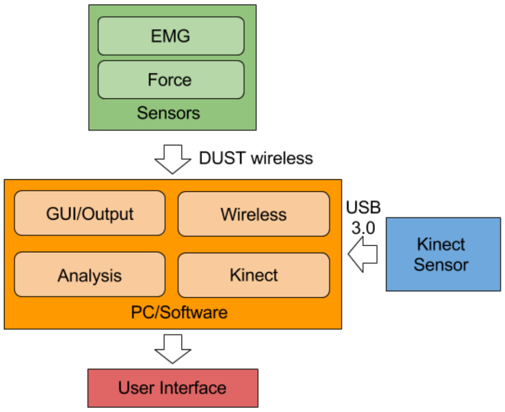


Figure 2: Block diagram of OptiFit.

OptiFit is designed to run on the operator's own PC by plugging in the Kinect and DUST wireless manager before downloading our software online. This was an important design decision for Omaro as we originally envisioned a stand-alone product that included its own battery-powered computer. Considering that the Kinect requires a significant amount of processing power, it made a lot more sense to take advantage of the fact that most of the population already owned their own machine. This also makes the system less expensive, more sustainable and easier to implement as a proof-of-concept.

3. Hardware Design

OptiFit uses three sensors: the Microsoft Kinect, an EMG, and a force sensor to get a holistic point of view of the user's exercise. Although a large number of sensors can expand the functionality of OptiFit's analysis, it does present challenges concerned with data transmission, processing and system integration.

The Microsoft Kinect talks to the PC via USB3 and can be accessed directly from the Microsoft Virtual Studio IDE in Windows. The Force sensor and EMG meanwhile, communicate to the PC's COM port via the DUST wireless network freeing the user of any obtrusions during their exercise.

The mother program of OptiFit will use the sensor data and in turn display the information via a GUI, and warn the user should their parameters seem dangerous. Unlike the software side, the prototype will include most of the hardware features found in the final commercial version.

3.1 Microsoft Kinect

The Microsoft Kinect, shown in Figure 3, is the heart and namesake of OptiFit. It enables the system to track skeletal positioning safely during lifts and in turn allows OptiFit the ability to visualize the exercise.

Originally, the project was intended to be implemented using sensors that tracked the position of the spine in space over time. Although this could have worked, the solution lacks in two major areas: design would take an incredible amount of calibration, and the hardware would be limited in scope.

What the Kinect allows us to do is visualize mechanics rather than relying on data alone. The software tools that Microsoft provides are substantial in that they allow for much faster development, and the technology lends itself to showing the user a more intuitive, graphical display of their movements. In addition, the hardware includes a myriad of features that OptiFit takes advantage of. The Kinect also has an infrared camera for temperature monitoring and voice control that will be used in future versions of OptiFit. A full list of the Kinect's features satisfying functional specifications `[R3.1.X]` is listed in detail in Table 1.

Figure 3: Microsoft Kinect for PC v2. [3]

Table 1: Microsoft Kinect features and benefits [4]

Feature	Benefits
Improved body tracking	The enhanced fidelity of the depth camera, combined with improvements in the software, have led to a number of body tracking developments. The latest sensor tracks as many as six complete skeletons (compared to two with the original sensor), and 25 joints per person (compared to 20 with the original sensor). The tracked positions are more anatomically correct and stable and the range of tracking is broader.
Depth sensing 512 x 424, 30 Hz One mode: 0.5–4.5m	With higher depth fidelity and a significantly improved noise floor, the sensor gives you improved 3D visualization, improved ability to see smaller objects and all objects more clearly, and improves the stability of body tracking.
1080p color camera 30 Hz (15 Hz in low light)	The color camera captures full, beautiful 1080p video that can be displayed in the same resolution as the viewing screen, allowing for a broad range of powerful scenarios. In addition to improving video communications and video analytics applications, this provides a stable input on which to build high quality, interactive applications.
Active Infrared 512 x 424 30 Hz	In addition to allowing the sensor to see in the dark, the new IR capabilities produce a lighting-independent view—and you can now use IR and color at the same time.
Wider/expanded field of view	The expanded field of view enables a larger area of a scene to be captured by the camera. As a result, users can be closer to the camera and still in view, and the camera is effective over a larger total area.

3.2 EMG Sensor

OptiFit takes advantage of an EMG sensor to monitor muscular activation during exercise. The EMG signal will be used with the system as a parameter to see if the user is doing the exercise correctly. OptiFit's EMG has been made from the ground up and consists of an instrumentation amplifier, filtering in the range of 10 – 3000Hz as specified by `[R2.1.4]`, rectification and amplification before being sent into the analog-to-digital converter (ADC) for wireless transmission. Although we could have purchased a pre-made EMG, our team wanted a project with a substantial hardware component to display our full range of skills.

To pick up an EMG signal, a differential amplifier with a high Common-Mode Rejection Ratio (CMRR) is required at the input—ie. an instrumentation amplifier. The differential amplifier is used to pick up a small difference in voltage between two spots on the same muscle, rejecting the "global" signal obtained. It is also desirable to have low input current (`[R2.1.5]`) at the input, to ensure user safety and to prevent degradation. A reasonably priced instrumentation amplifier that fit our requirement `[R2.1.6]` was the INA826, which has a minimum CMRR of 100dB and maximum input current of 8mA [5].

Next, the output of the INA826 is fed into a high-pass filter (HPF) to remove remaining DC offsets and low frequency noise. The majority of the spectral density of an EMG signal reaches down to about 10Hz; to calculate this cut-off frequency we used the following from *Design with Operational Amplifiers* by Sergio Franco (DWOA):

$$f_o = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \tag{1}$$

As we already had access to 100nF capacitors, we determined standard values for R_1 and R_2 to be 39kΩ and 82kΩ respectively. Using a quad-package TL074CN Operational Amplifier (OpAmp) [6] we implemented the HPF circuit as shown below in Figure 4. An active filter was used here as the quad-package OpAmp that was required for subsequent stages had an additional amp available. In addition, we wanted to get a sharp cut-off frequency for non-DC low-frequency noise.

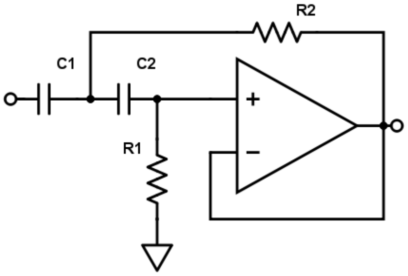


Figure 4: Active HPF circuit for OptiFit.

The HPF is then used as the input into the full-wave precision rectifier (FWPR)/LPF combination to implement an "envelope detector," that is, a running average of the signal. We included the envelope detector so that for the next stage of the project (wireless transmission), the output will only report smoother, positive values rather than being so erratic. This architecture satisfies `[R2.1.2]`.

The first stage of the FWPR is an inverting amplifier that when the input is positive, produces a negative output. When the input is negative, the feedback resistor is turned off and the output reaches 0.65V from the feedback diode voltage drop.

The second stage is an inverting amplifier that flips the signal back to its original sign. It also works as a summation of two signals taken from the first amp and the input: when the original input is negative, a half-wave negative signal and a mixed amplitude full-wave positive signal are summed leaving a full-wave positive signal. A positive signal has a similar effect except the signs are interchanged.

The FWPR was implemented as shown in Figure 5 from DWOA using the same TL047CN OpAmp as in the HPF.

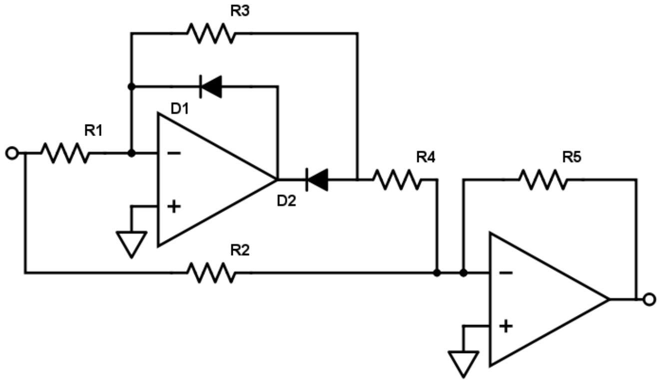


Figure 5: FWPR circuit used in OptiFit's EMG.

To determine the resistor values we started by observing the gain of the system, as the signal becomes positive or negative using equations from DWOA:

$$V_O = A_P V_I \;\; for\; V_I > 0 \tag{2}$$

$$V_O = -A_N V_I \;\; for\; V_I < 0 \tag{3}$$

Where V_o and V_I are the output and input voltage of the FWPR respectively; while A_N and A_P are the negative and positive voltage gains respectively. From inspection, the following can be seen:

$$A_N = \frac{R_5}{R_2} \tag{4}$$

$$A_P = \frac{R_2 R_5}{R_1 R_3} - A_N \tag{5}$$

To achieve the desired effect, we ideally want both gains to be equal ($A_N = A_P$) so that the summed signal represents true rectification. The trivial solution is to make $2R_1 = R_2 = 2R_3 = 2R_4 = R_5$ so that equations (4) and (5) are equal. We chose 2kΩ and 1kΩ to realize the FWPR with standard resistor values.

The FWPR output is then ported into a passive RC LPF to remove the high frequency component of the signal obtained. We opted for a passive filter as the additional area required to add one OpAmp was not necessary for the marginal increases in performance. To determine the resistor value for a 47nF capacitor and cut-off frequency of 3.33kHz we used:

$$f_c = \frac{1}{2\pi RC} \tag{6}$$

Calculating for a standard value, R was chosen as 1kΩ.

Lastly, the signal is amplified by a non-inverting OpAmp configuration to get sufficient amplitude before the ADC. The gain of the topology is determined by the ratio of the two resistors shown on the right hand side of Figure 6:

$$Gain = V_{LPF}\left\{1+\frac{R_2}{R_1}\right\} \tag{7}$$

As the input of the ADC on our microcontroller can only take up to 1.8V (`[R2.1.3]`), and V_{LPF} (the peak voltage out of the LPF) is around 120mV, we chose R_2 and R_1 to be 5.6kΩ and 470Ω respectively for a modest gain of 22dB.

The entire EMG circuit can be seen below in Figure 6 as a block level representation. A detailed schematic with labeled connections is included in Appendix B. Please note, the RC values labeled in this schematic are different than the individual blocks described in this section for simplicity. The labels shown on the PCB in Figure 7 are consistent with Appendix B.

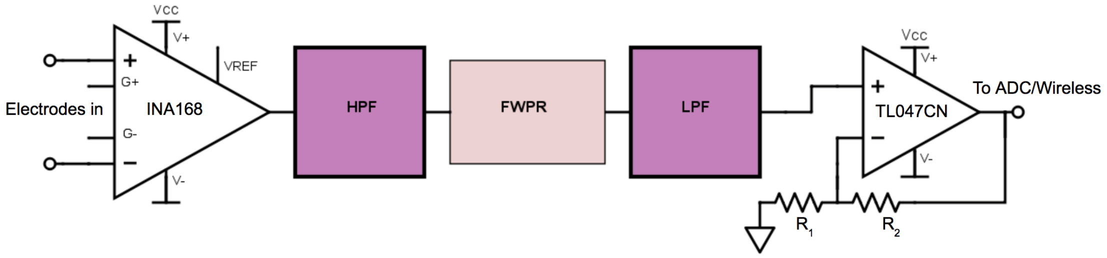


Figure 6: Block level diagram of EMG.

3.2.1 PCB

The EMG will be implemented on a PCB to provide reliable performance in addition to a small form factor. We decided to use through-hole technology rather than Surface Mount Technology (SMT) to make manufacturing and debugging easier. With that said, we were limited to a 5cm x 10cm dimension to fit all of the necessary components to realize the EMG.

To make the EMG fully wireless, we had to include enough power for the OpAmp's voltage rails for a reasonable amount of usage. Another design problem was providing dual power supplies to have the amplifiers operate below and above the reference voltage. To solve this problem, we decided to use a rail splitter, the TLE2426CP, to push the reference voltage from ground to $V_{BATTERY}/2$. By doing this, we save excessive weight and bulk from having two separate batteries. A 9V battery was chosen according to `[R2.1.10]` to provide enough voltage swing for the circuit. With the maximum total input current of the active components at 4.7mA, the maximum total output power of 42.3mW would last around 60 hours with a 9V Duracell battery rated at approximately 300mAh.

The PCB shown in Figure 7 satisfies the functionality specifications `[R4.1.X]`. The PCB also includes holes so that it can be fixed to the case, as outlined in §6.1. Future versions of the PCB will shift to a full surface-mount design as specified by `[R4.1.11]`.

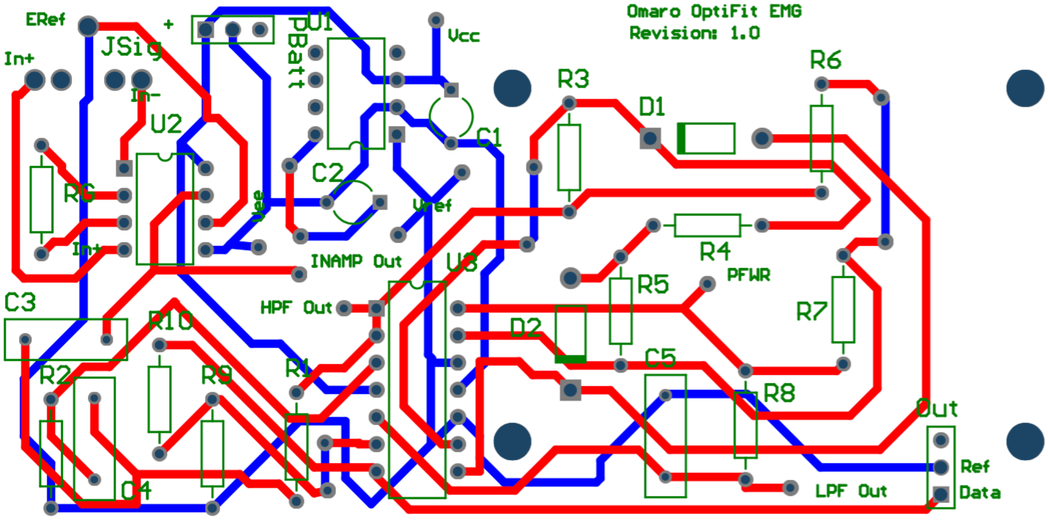


Figure 7: EMG PCB layout for OptiFit. Blue lines are reserved for power/jumps.

All of the components outlined in the previous section have been compiled in Table 2, labeled in accordance with the full schematic in Appendix B and the PCB layout in Figure 7. R_G is a nominal value for the gain of the instrumentation amplifier that would be modified depending on the muscle measured.

Table 2: EMG PCB's component values.

Component	Value	Component	Value
R_1	39kΩ	C_1	1uF
R_2	82kΩ	C_2	100uF
R_3	1kΩ	C_3	100nF
R_4	1kΩ	C_4	100nF
R_5	1kΩ	C_5	47nF
R_6	2kΩ	D_1	N/A
R_7	2kΩ	D_2	N/A
R_8	1kΩ	U1	TLE2426CP
R_9	470Ω	U2	INA826
R10	5.6kΩ	U3	TL074CN
R_G	Tentative		

3.3 Force Sensor

With the power of software, the Force Sensor will provide OptiFit with the data necessary to compute the force exerted on the barbell. We opted to design the force sensor as a pad or plate, to make set-up easier for the user. Initial design solutions included insoles with force sensitive resistors; however, these ideas were scrapped as we concluded that the logistics of different sized feet, size constraints, and difficult set-up would not be appropriate for a product that is being marketed for business-to-business (B2B) sales (such as rehabilitation clinics).

We quickly realized that there already exists a proven safe, simple solution to measure force—bathroom scales. Upon tearing down a scale we discovered that the most economical and feasible solution would be strain gauges. As pictured in Figure 8, the 45kg strain gauge includes three outputs: plus, minus, and ground.

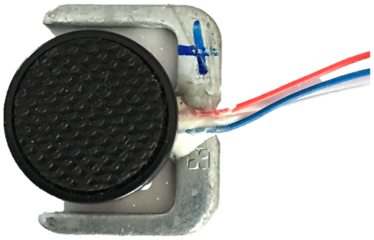


Figure 8: Strain gauge taken from bathroom scale for OptiFit's force pad.

Like a bathroom scale, each channel of the force pad will have four strain gauges. By having two separate channels with 4 x 45kg strain gauges each, we are not only able to measure if the user is shifting left or right, but also we satisfy our force requirement `[R2.2.1]`. A Wheatstone bridge configuration, as shown on the left side of Figure 9, is used to couple the current generated from each of the four strain gauges of the channel into the ADC. The Sparkfun combinator board, on the right hand side of the same figure, is specifically made for strain gauges/load cells and will be used to realize the Wheatstone bridge.

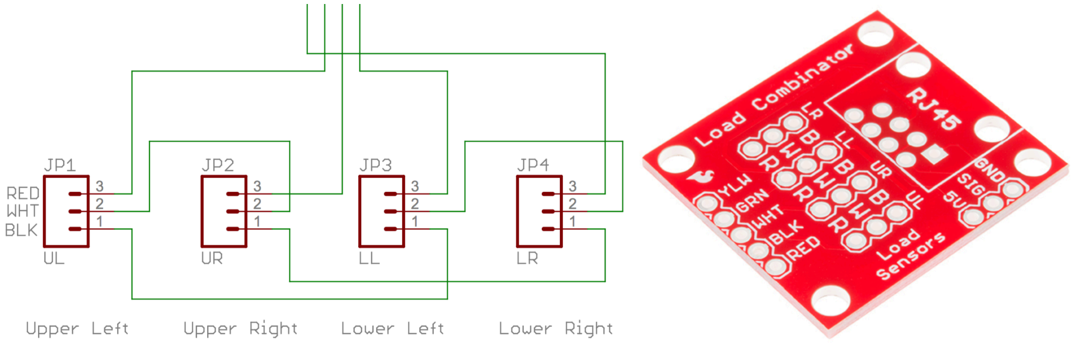


Figure 9: Wheatstone bridge configuration of four load cells (left). Implemented with the Sparkfun combo board (right).

A Sparkfun load cell amplifier (Figure 10) designed for our exact application will be used to condition the signal. The board includes the ADC and will be used to step up the signal before it is sent digitally to the DUST wireless mote.

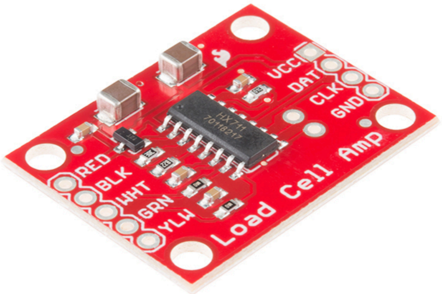


Figure 10: Sparkfun load cell amplifier with built in ADC.

What the sensor will measure and send as far as total force will be as follows:

$$F = m_{user}g \qquad for\ t = 0 \tag{8}$$

$$F = m_{user}g + m_{barbell}(g + a_{barbell}) \quad for\ t_{lift} > t > 0 \tag{9}$$

$$F = g(m_{user} + m_{barbell}) \quad for\ t_{lift} = t \tag{10}$$

Where g is the gravity constant on Earth and t_{lift} is the time to complete one deadlift repetition. To determine the force that the user exerts on the barbell:

$$F_{barbell} = m_{barbell}a_{barbell} \tag{11}$$

Using equations (8) and (9) as well as an input in the GUI for $m_{barbell}$, OptiFit will be able to display the user's force exerted throughout the lift. A force diagram of demonstrating the equations above is shown below in Figure 11.

Figure 11: Freebody diagram of deadlift for OptiFit analysis, source: bodybuilding.com.

4. Embedded Design

Embedded Design concerns the transfer of the data (voltage levels specifically) from the EMG or force pad sensors to the DUST wireless mote, referred to as DUST from hereon. The DUST network is comprised of a mote and a network manager as shown in Figure 12. One mote will be dedicated to the EMG sensor and a second one will be attached to the force pad sensor.

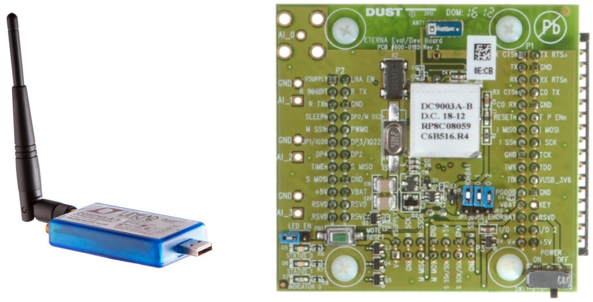


Figure 12: DUST SmartMesh IP USB Network Manager (left) and Eval/Dev Mote Module (right).

The mote is the piece that will be attached to the EMG circuitry described in §3.2 to retrieve muscle activation data. The voltage levels provided by the EMG sensor requires processing from the moment it enters the analog input of the mote, to the digital output of its ADC, and finally to the USB manager and PC. It is a different scenario for the mote on the force pad sensor since the circuitry of the sensor provides a digital output that does not require further processing on the mote. Therefore, the data will be transferred directly to the mote's digital input.

In this section, the design of sending/receiving of data will be broken down into two sub-sections: the wireless mote and the network manager. Each section will dig deeper into the specifics of respective functionalities and data handling. The flowchart in Figure 13 below provides an overall picture of the embedded design. The flow of data starts from the input of the mote and continues with data processing performed by the mote's SoC. It continues on to the network manager where serial communication provides a bridge between the manager and the PC. Finally, the mother program and the GUI outlined in §5 perform their duties to provide a graphical output of the sensor data against time. The graph at the end of the flowchart is an output of our prototype EMG displayed on an oscilloscope.

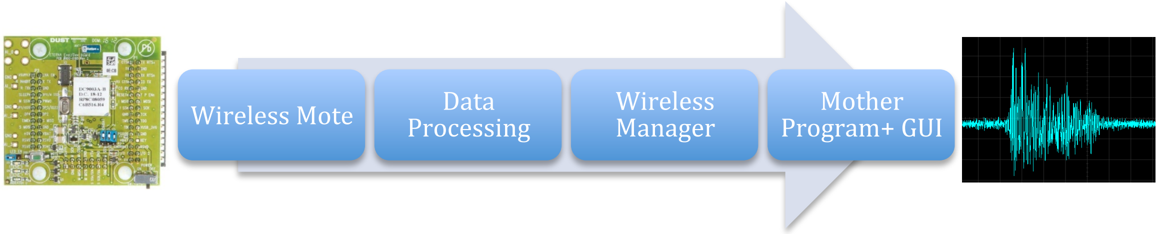


Figure 13: Embedded software flowchart.

4.1 Wireless Mote

In the initial step, the EMG sensor will be providing data to the mote module through the analog input and ground pin AI_1 and GND pin respectively on the board. The pins on the mote are processed with an IEEE 802.15.4e equipped System-on-Chip (SoC). What is beneficial about DUST's motes compared to other available solutions is that their SoC's are configured for extremely low power.

Furthermore, the SoC has an auto-forming mesh technology that self-heals and self-sustains the network, taking extra steps of configuration and maintenance out of our hands while meeting requirement `[R3.2.1]`. There is also the zero collision packet exchange technology that ensures no data is lost due to bad timing of a packet exchange. Given all the features, the DUST was our justified choice compared to available alternatives such as Bluetooth, 802.11 WiFi and even ZigBee since they individually did not provide the same combination of powerful features as those found in DUST's technology. The network's features also satisfy functional specifications `[R2.5.X].`

Data processing will be conducted by writing two custom C programs for each mote: one for the analog (EMG) input and another for the two digital (force left and right) inputs into the mote. The custom applications will be built by modifying example programs provided in DUST's on-chip SDK folder. LTC recommends that the voltage on the analog signal lies within 0 – 1.8 V_{PK-PK} to avoid damaging the input pins. The example program for extracting an analog input includes variable on-chip parameters that are user programmable. These parameters include sampling rate, pin number, voltage offset, and Variable-Gain Amplification (VGA) [12]. The voltage-offset parameter changes the reference voltage level of the signal while the VGA parameter zooms the signal into a user defined voltage range. We will be using these parameters to obtain better resolution of our 400 mV EMG signal instead of the default 1.8 V swing. The analog data will then enter the ADC and finally be sent for transmission. Since the force sensor signal will enter the mote as a digital signal, the same process will apply with the exception of the ADC.

The next procedure is to send the data over to the manager module. The transmission of data is done after establishing a communication flow between the mote and manager. This step also requires us to write an application. Once the data is transferred, the manager's serial API will receive a data notification. These stages of software development are quite complicated; however, LTC has provided their consumers with sample applications written in Python code (`[R2.5.1]`). We will be looking at an application named "UpStream" to create our own model of data transmission. The same procedure to supply packets to the manager will be applicable to the force pad sensor as well.

4.2 Network Manager

The USB Network Manager is a module that manages network performance/security and exchanges data with a host application. There are two ways to access the manager: the Command Line Interface (CLI), and the Application Programming Interface (API). In the case of OptiFit, the API communicates to the manager by talking to the corresponding serial port assigned for API usage. The manager will transfer data at a rate of 115200 baud to meet both performance and functional requirements, namely `[R3.2.5]`. We will be tapping into readily available resources from LTC, namely the Python developer Software Development Kit (SDK) to create the host application.

On a high level, our design needs to ensure that the manager establishes a communication channel with the client (i.e. the host application on the mote). The communication is done via message passing of commands or notifications. To do this, LTC's messaging system employs a client/server handshake topology to accomplish tasks. The commands are in the form of *task queries* sent by the mote to the manager. In our case, we will use commands to see if the manager is ready to establish a connection and if the manager is ready to receive data from the mote. Notifications, on the other hand, are *responses* from the manager concerning the task queries sent by the mote.

For OptiFit, the manager will send notifications to the mote regardless of the mote's status, whether that be establishing a connection or receiving data. In the case where the sensor data is received, the notification will include source (mote) Medium Access Control address, source port number, destination (manager) port number, and a timestamp of when the data was received. To initialize communication and transmit data, we will modify a simple packet exchange example provided in the Python Developer SDK folder called the "hello" packet exchange.

Once the manager successfully receives data, we are to design a c-plus plus (C++) serial communication port application (wireless.cpp). The purpose of this application is to grab the data sitting in the manager wireless module through the serial port in our PC. The data is then incorporated into the mother program to perform analysis on the sensor data, display the data on the GUI, and issue a sound should the form is improper. The mother program and the software package design are discussed thoroughly in the next section.

4.3 Future Design

Future data transfer will be provided with network analysis software that can identify and troubleshoot any network issues between sensors and the manager. The software will be able to further keep track of rate of exchange of packets and ensure that syncing is done properly and whether the system will need calibration.

5. Software Design

The design of software is categorized into five sections. These five sections are further explored in Table 3 below.

Table 3: Software Design Breakdown.

Application	File Name	Functionality
Mother Program	Build0.0.cpp	brings all applications together
Data Transfer	Wireless.py	transfers data from DUST manager
Kinect Program	BodyBasics.cpp	grabs joint data from Kinect sensor
Analysis Class	OptiBody.cpp	performs analysis on data from all sensors
GUI	GUI.cpp	graphs data in real-time

5.1 Mother Program

The mother program is the main program that brings the result of all of our applications together; it is a Win32 application, written in C++, that uses Windows API function calls. These function calls perform windowing for the application and message handling for the top menu. We have edited the message handling function code to allow intractability with the program. The main interface is the top menu, where the Kinect and GUI application can be started. This edited code creates processes that would initiate the Kinect and start monitoring skeletal position (`[R3.5.1]`) in separate threads. This program's purpose, in summary, is to interact with all applications and handle their threading.

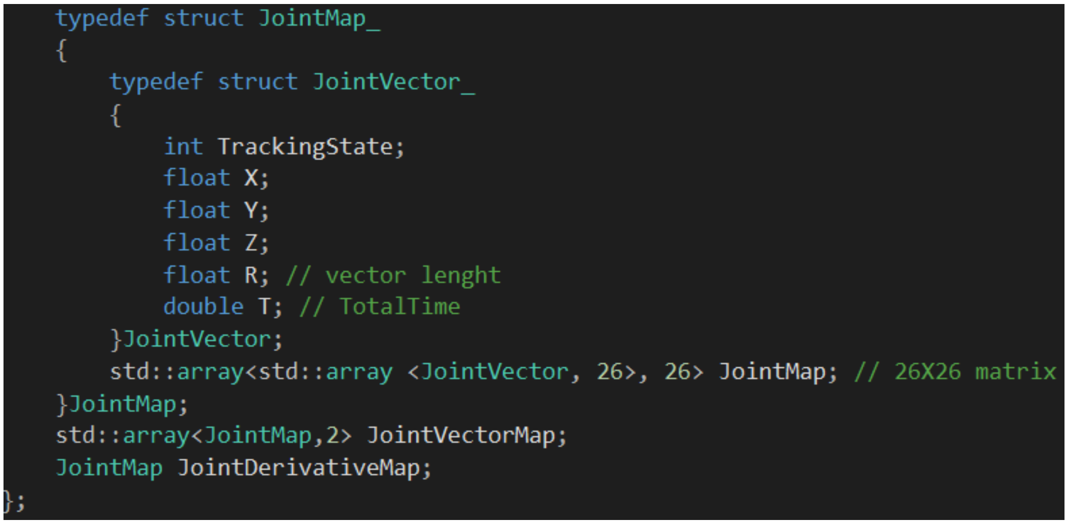
5.2 Data Transfer Program

The data transfer program was explored in §4.2. Currently, the Python SDK is available to us; however, depending on various factors we may or may not change the data transfer application to C++.

5.3 Kinect Program

The Kinect program is responsible for tracking skeletal data. That includes the positioning of the joints and their rotational movement. The first action of this application is to initialize Kinect sensors and start pulling information from them (`[R3.1.3]`). The Kinect records 3D space points for 26 joints of the user that are adjoined to make joint vectors. Along with the vector data, global time is also saved to display the value as a function of time. This indicates that our software needs to perform analytical calculations on arrays of size 26 x 26 and record the results properly. These calculations refer to the conversion of data from skeletal space to depth space requirement `[R3.1.4]`.

The sample code shown in Figure 14 below is the data structure the joint data is stored in. A “map” in software terms refers to a matrix that holds values or structures in a coordinate system. The JointVectorMap and JointDerivativeMap, use the data structure below and use the JointType as mapping variables. The variables X,Y, and Z refer to the three dimensional positions of the joint. R refers to the angular dimension of the joint, while the TrackingState variable specifies whether the Kinect sensor can see the joint or not. Lastly, the variable T is used to a hold a time stamp to synchronize data.



```
    typedef struct JointMap_
    {
        typedef struct JointVector_
        {
            int TrackingState;
            float X;
            float Y;
            float Z;
            float R; // vector lenght
            double T; // TotalTime
        }JointVector;
        std::array<std::array <JointVector, 26>, 26> JointMap; // 26X26 matrix
    }JointMap;
    std::array<JointMap,2> JointVectorMap;
    JointMap JointDerivativeMap;
};
```

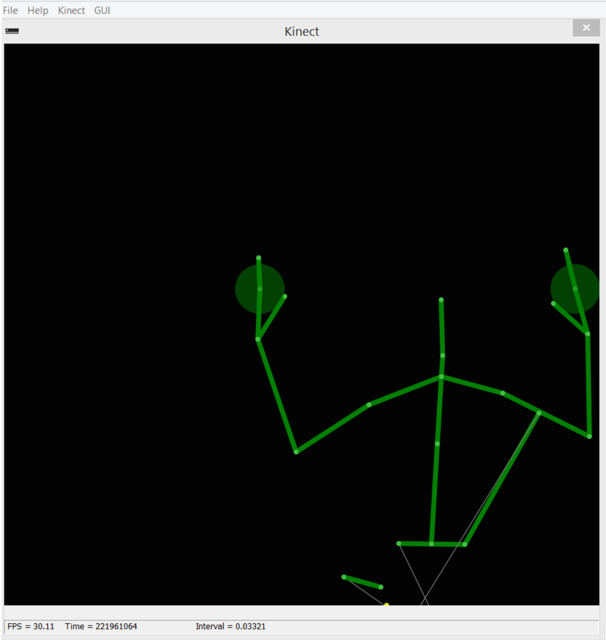
Figure 14: JointMap data structure of the analysis class

5.4 Analysis Class

The analysis class, “OptiBody,” handles the computation of joint and sensor data. The joint data is sent from the Kinect program and stored in a sparse matrix of size 26 x 26. Using the matrices, specific code is written to analyze the vector position of the data with respect to one another (`[R3.3.1]`) and more importantly with respect to time. This code will also provide derivatives and satisfy the functional specification `[R3.3.2]`. A specific example is the rate of change of movement of the wrist joints: since our proof-of-concept will focus on the performance of the deadlift the wrist joints will move with the barbell. We will be using the acceleration of the wrist joints to compare the barbell acceleration to the measured value on the force pad to determine whether the exercise is being executed correctly.

5.5 GUI Program

Finally, the GUI program will be graphing data from all three sensors. Its first task is to open a window for each sensor data when menu options are selected. To open data from the Kinect, for example, the GUI is sent a pointer from the mother program that points to the OptiBody class, grabbing data that the user specifically demands with the GUI. The required data from the GUI is held in a buffer deque (double-ended vector). In addition, the GUI maps the data to a pixel value to make a XY point (not to be confused with position points in Kinect program) and then draws a line using the Windows D2D1 library. These lines will add up to form a graph to represent the skeletal positioning of the user (`[R3.4.1]`) as shown in Figure 15.



```
FPS = 30.11    Time = 221961064          Interval = 0.03321
```

Figure 15: Example of Kinect GUI showing user's skeleton data.

The GUI program is also to satisfy requirements `[R3.4.2]` and `[R3.4.4]` to display force/EMG sensor data against time. This will be accomplished by retrieval of data from files created by the Data Transfer application.

The most critical duty of the software, however, is to ensure the user is alerted should they perform a deadlift improperly. To accomplish this, we will focus on warning the user if there is too much force on the back. In this case, the weight is fixed in a position (zero acceleration, and no change in force exerted) while the EMG sensors are indicating large amount of muscle activation in the lower back. Also, the software will look for large barbell acceleration and for nominal EMG activity, which could indicate improper form. A complete overview of the software can be seen in Figure 16.

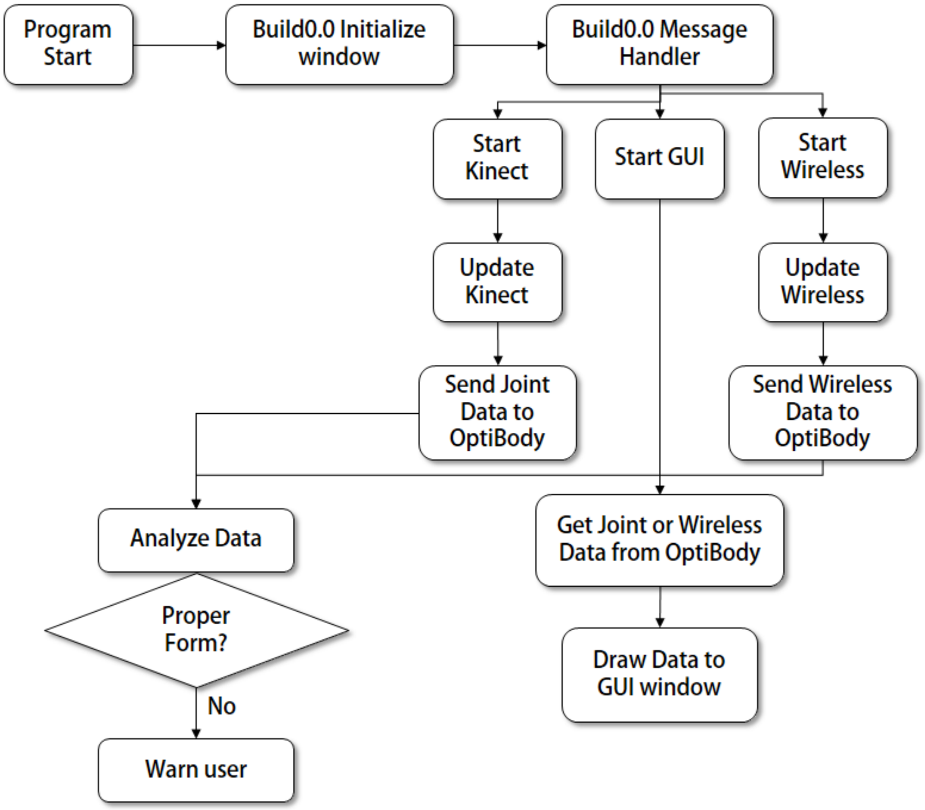


Figure 16: Complete flowchart of OptiFit's software.

5.6 Future Design

Future software design will include a wide variety of analysis that indicates various alerts. There are several ways of failing a powerlift; therefore, there is a huge amount of potential to expand the software to include every type of improper form. The analysis will further expand into various exercises as well as movements associated with specific sports. Another future software feature will be the inclusion of voice activated "start" and "stop" commands so that an operator will no longer be required. This feature takes advantage of Kinect's speech recognition, specified in functional specifications `[R3.1.5]`, `[R3.1.6]`, and `[R3.4.7]`.

6. Mechanical Design

6.1 EMG Casing

The EMG case is important in that it is how OptiFit directly interacts with the user's physiology. Omaro wanted to ensure the case prioritized user safety, sustainable materials, easy use, and accessibility for development. A Computer Automated Design (CAD) picture of the EMG case is shown below in Figure 17.

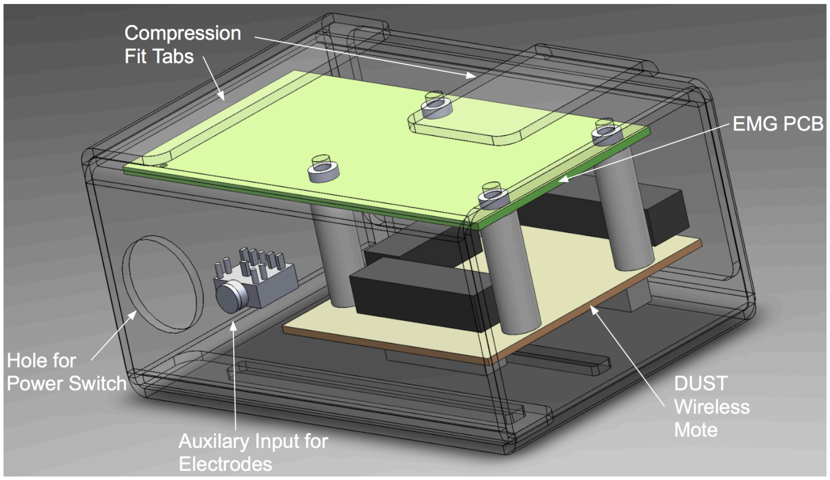


Figure 17: EMG case with PCB (green) and DUST mote (orange) that satisfies `[R4.2.X]`.

The laser-cut acrylic design is just large enough at 11cm x 7.6cm x 5.4cm to enclose all of the necessary circuitry, including the PCB, DUST wireless mote, auxiliary input (for electrodes), and 9V battery to fit `[R4.2.1]`. We opted for a custom case as the flexibility of laser cutting allowed us full customization compared to stock boxes.

Two vertical slots are included so that the case can sit length-wise on the middle of the user's back to reduce bulk and interference. A Velcro and elastic strap will be used for easy and comfortable fastening. The case itself as shown in the figure is actually two separate compression-fit pieces so that the user can attach one side themselves, while an operator can slide in the circuitry and fix the electrodes in place as outline by `[R1.6.2]`. We preferred the two-piece design as it made for easy troubleshooting during development. Full dimensions are in Appendix C.

The downside of this design however, is that it is based off of idealized bends in the acrylic. Small errors in these bends may make the compression fit sloppy. What is feasible to work around this issue is laser-cutting multiple cases and using trial-and-error. As the acrylic is highly recyclable and reusable, this still fits our sustainability design specification. Future versions for the commercial product will iron out the intricacies of manufacture and be of smaller size as the PCB shifts to SMT.

6.2 Force Pad

Like the EMG case, the force pad was an area where Omaro wanted to first take into account the user's well being during their lift. As the demands of powerlifting in particular are so strenuous, we required a sensor that would stand up to continuous strain and abuse as we outlined in `[R1.5.5]`, `[R1.8.6]`, and `[R2.2.1]`. We also valued a sensor that did not interfere with the user's ability to perform their powerlift. The last thing we wanted to include, as outlined in the previous section, was sustainable construction materials that considered the cradle-to-cradle design process. Our CAD depiction of the force plate can be seen below in Figure 18.

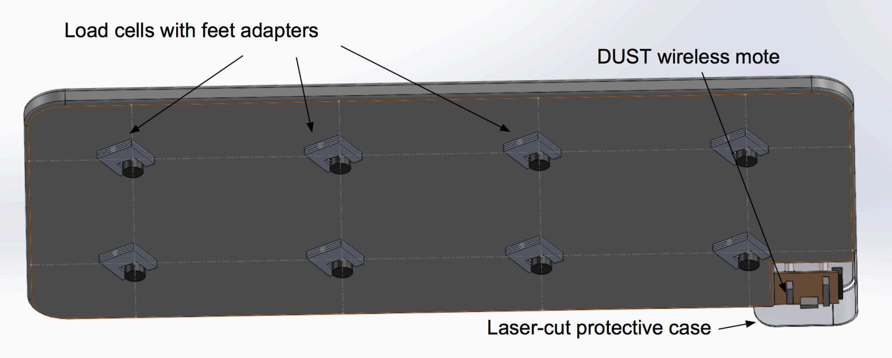


Figure 18: Force sensor design meeting safety and performance requirements.

The design above is made of 0.75" thick plywood: a stiff, inexpensive and recycled material that is easy to cut and screw into. Wood quickly became the most feasible material because of these reasons over other alternatives. The eight 45kg strain gauges are spread evenly under the board's center of mass to prevent deformation and support lifts up to 700lbs. They are mounted to the underside using laser-cut adaptors that screw into the main platform. The measurements of the force pad and the strain gauge adaptors can be found in Appendix C.

In the corner of the force pad, as shown in the figure, is a 8cm x 8cm cutout that houses the DUST wireless mote. This cut out was necessary as it made room for the ~1" tall mote while keeping the overall clearance of the pad low enough (4.2cm) to avoid interference during the lift. Like the rest of our design, the housing is made with laser-cut acrylic that wraps around the mote for protection.

For the commercial version of OptiFit, Omaro would like to create a pad that is large enough for the entire lift to be performed on. The proof-of concept version is portable and easier to implement at 80cm x 40cm, but it can only accommodate the user's feet, not the barbell itself. A larger commercial product would reduce interference with the lift due to the clearance of the force pad and enable room for more sensors and circuitry. Since OptiFit is designed as a B2B product, it makes sense to sell a pad that will be installed into the entire area of a clinic or lab.

7. Conclusion

The proposed design specifications emphasizing the proof-of-concept version of OptiFit will meet the functional requirements outlined by Omaro such as: safety, sustainability, performance, reliability, and usability. Together with the sub-sections of each component and the attached appendixes, this document outlines the full design and test plan of OptiFit. However, as the design process will continue to evolve until the public demonstration there may be some slight modifications that will be included in future revisions.

When completed, OptiFit will excel where other research and commercial products have failed. Its ability to use data received from muscle activation, force generation, and skeletal mechanics will allow the user to receive more detailed analysis in addition to real time audio feedback.

The proof-of-concept version of OptiFit will debut on April 18, 2016.

Glossary of Acronyms

ADC	Analog-to-Digital Converter
API	Application Programming Interface
B2B	Business-to-business
CLI	Command Line Interface
COM	Communications Port
DC	Direct Current
DUST	DUST wireless network from Linear Technology
DWOA	*Design with Operational Amplifiers*
EMG	Electromyograph
FWPR	Full-Wave Precision Rectifier
GND	Ground/0V
GUI	Graphical User Interface
HPF	High-Pass Filter
IEEE	Institute for Electrical & Electronics Engineers
IDE	Integrated Development Environment
LPF	Low-Pass Filter
LTC	Linear Technology Corporation
PC	Personal Computer
PCB	Printed Circuit Board
RC	Resistor-Capacitor
SDK	Software Development Kit
SMT	Surface Mount Technology
SoC	System on Chip
USB3	Universal Serial Bus 3

Sources & References

[1] https://learn.sparkfun.com/tutorials/getting-started-with-load-cells?_ga=1.11284980.1511930005.1450850059

[2] https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide?_ga=1.11284980.1511930005.1450850059

[3] http://gamerant.com/kinect-2-pc-release-date-price/

[4] https://dev.windows.com/en-us/kinect/hardware

[5] http://www.ti.com/lit/ds/symlink/ina826.pdf

[6] http://www.ti.com/lit/ds/symlink/tl071.pdf

[7] Franco, S. "Design with Operational Amplifiers" 3rd ed.

[8] http://cds.linear.com/docs/en/design-note/SmartMesh_IP_Mote_Serial_API_Guide.pdf

[9] http://cds.linear.com/docs/en/user-guide/SmartMesh_IP_User_s_Guide.pdf

[10] http://cds.linear.com/docs/en/software-and-simulation/SmartMesh_IP_Tools_Guide.pdf

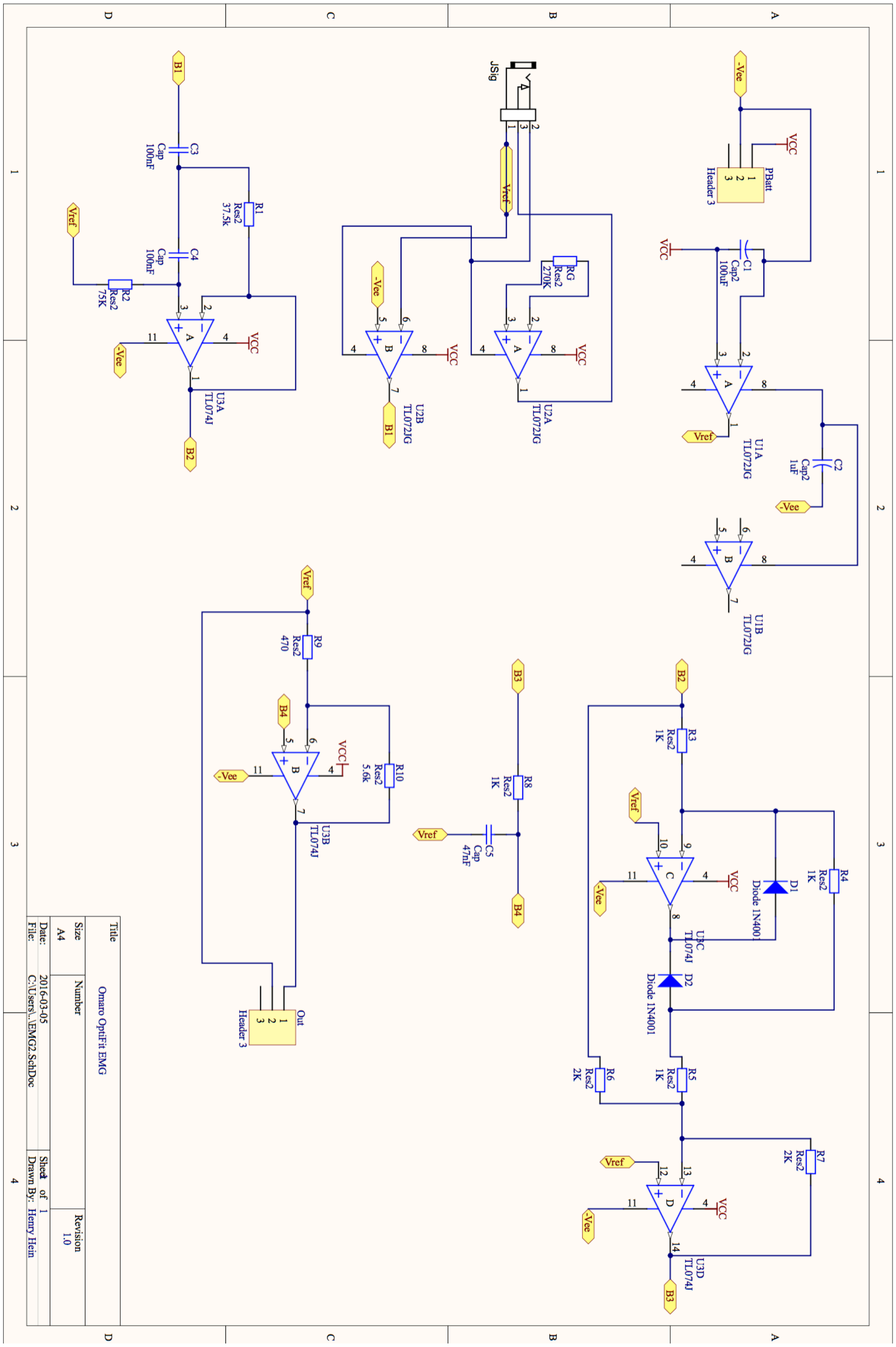
[11] https://msdn.microsoft.com/en-us/library/hh973085.aspx

[12] https://dustcloud.atlassian.net/wiki/display/OCSDK/ADC

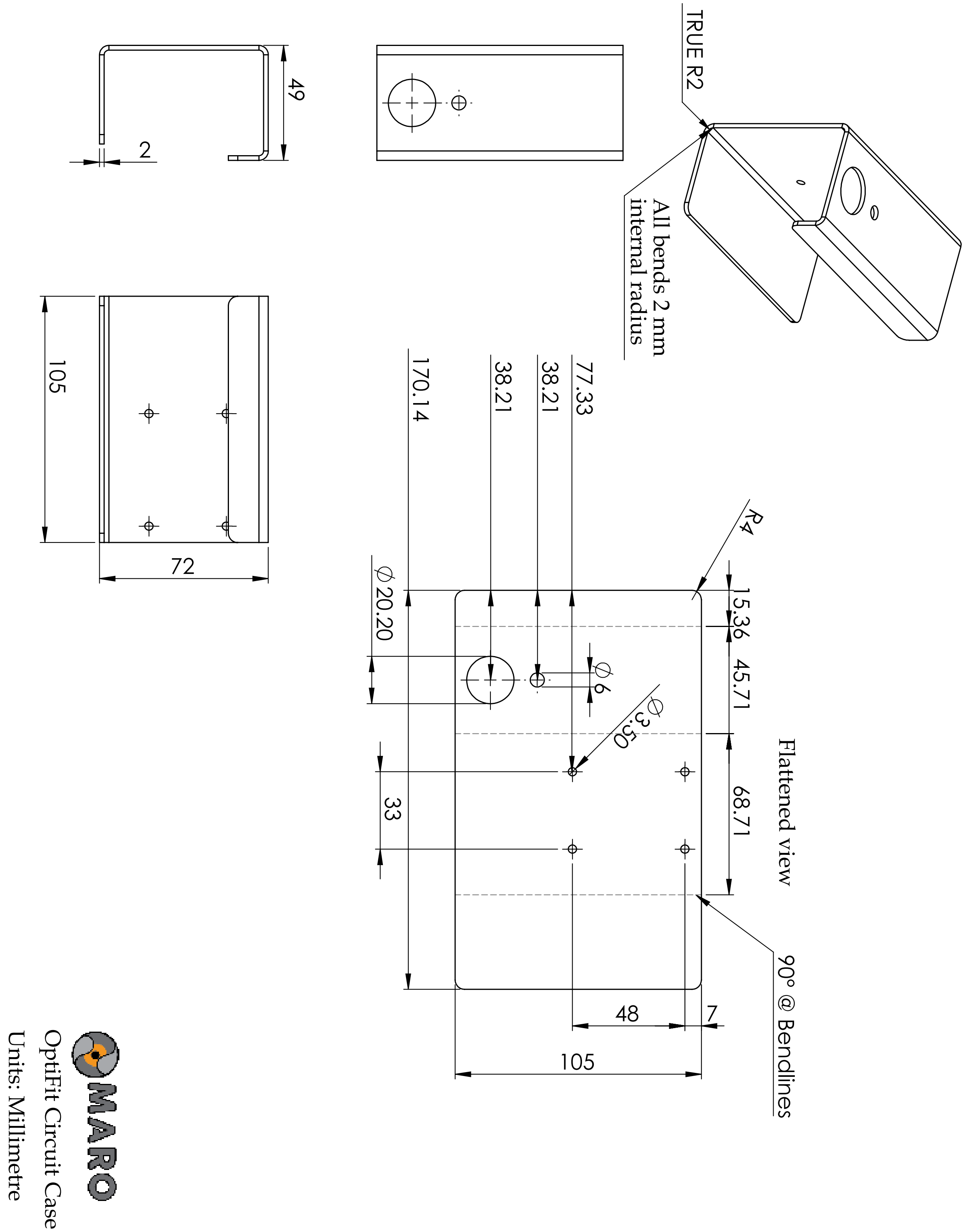
Appendix A – System Level Test Plan

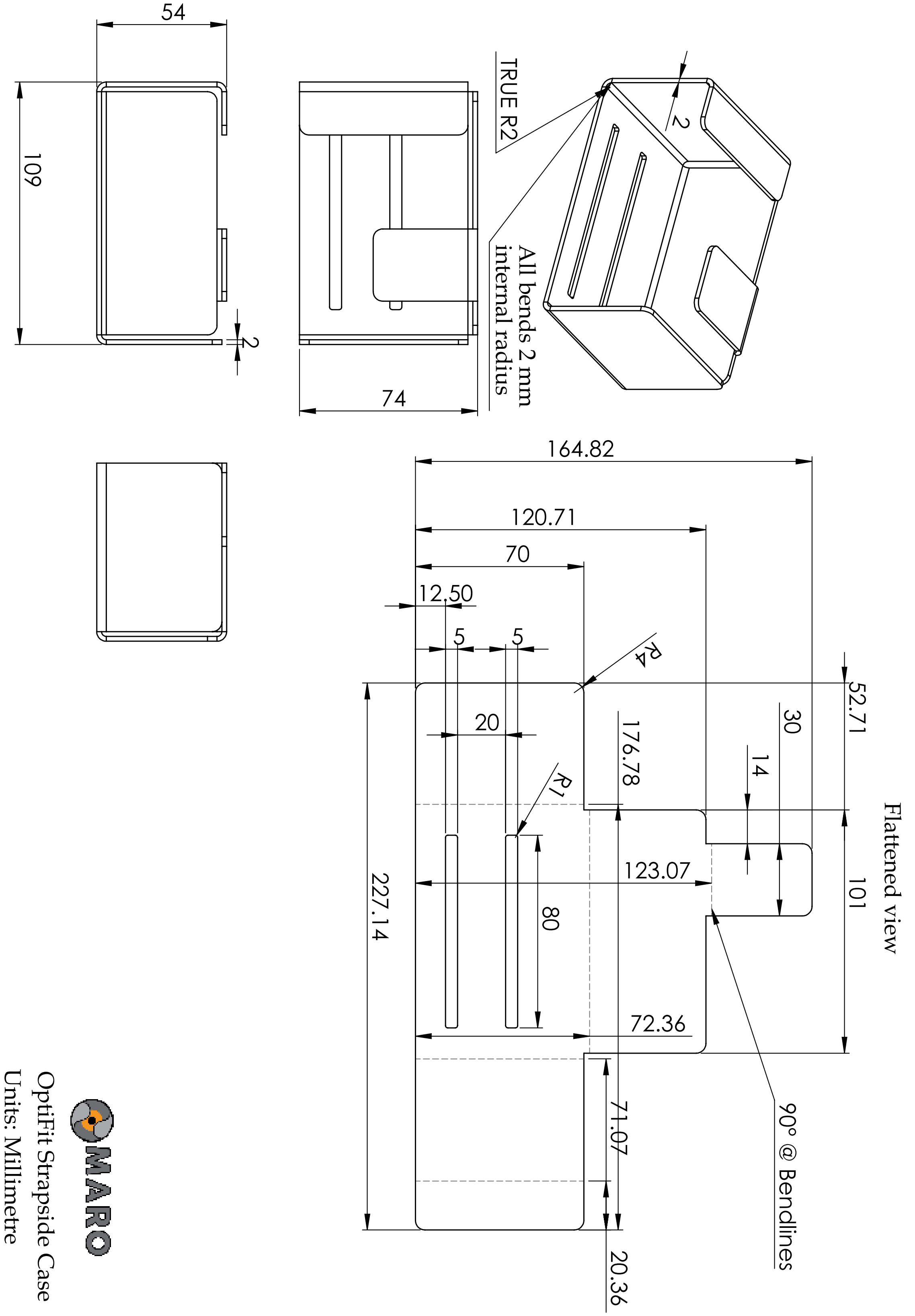
Test	Key Points	Pass? (✓/ x)
Hardware Start-up	- LED of DUST motes are on. - Wireless network forms within 2 min. - LED of EMG is on.	
Software Start-up	- Menu opens; this menu includes options for starting the Kinect, connecting to wireless, and additional sensor graphs. - Menu also allows for user to enter the mass of the deadlift.	
Kinect Initialization	- "Start Kinect" from menu opens window and the user's skeleton appears after 45 seconds.	
GUI Initialization	- After the Kinect is initialized, GUI windows for sensor data can be opened from the top menu.	
EMG Sensor	- EMG signal for the lower back displays on the GUI when attached to the user. - Noticeable peaks of voltage are present when the muscle is activated.	
Force Sensor	- Force signal displays on the GUI. - Noticeable changes in data are present throughout the lift.	
Functionality	- Lifts that are performed correctly are not interrupted by OptiFit. - Lifts that are deemed incorrect by OptiFit receive audio feedback loud enough to be heard by the user.	

Appendix B – Full EMG Schematic



Appendix C – CAD Files

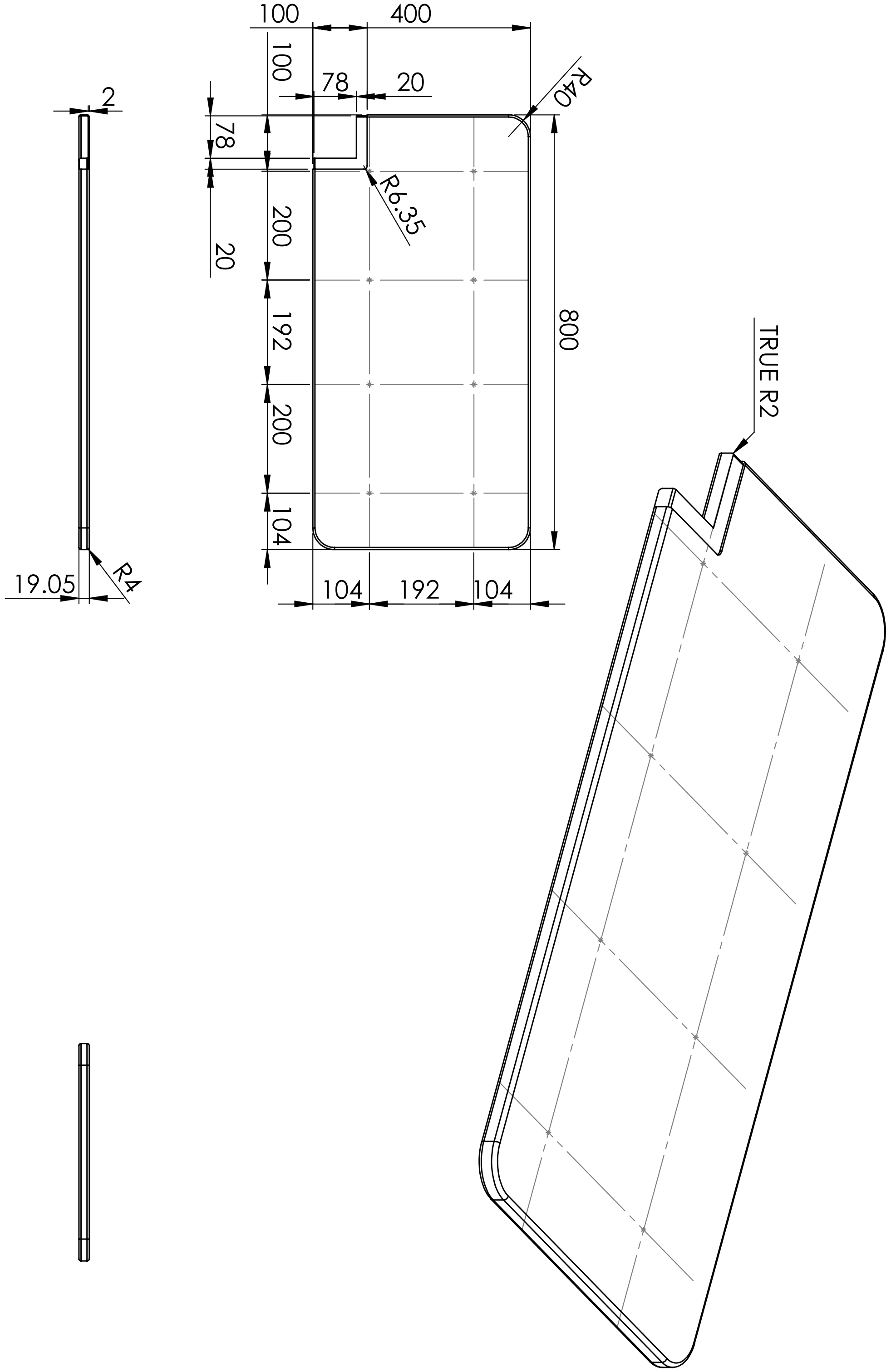




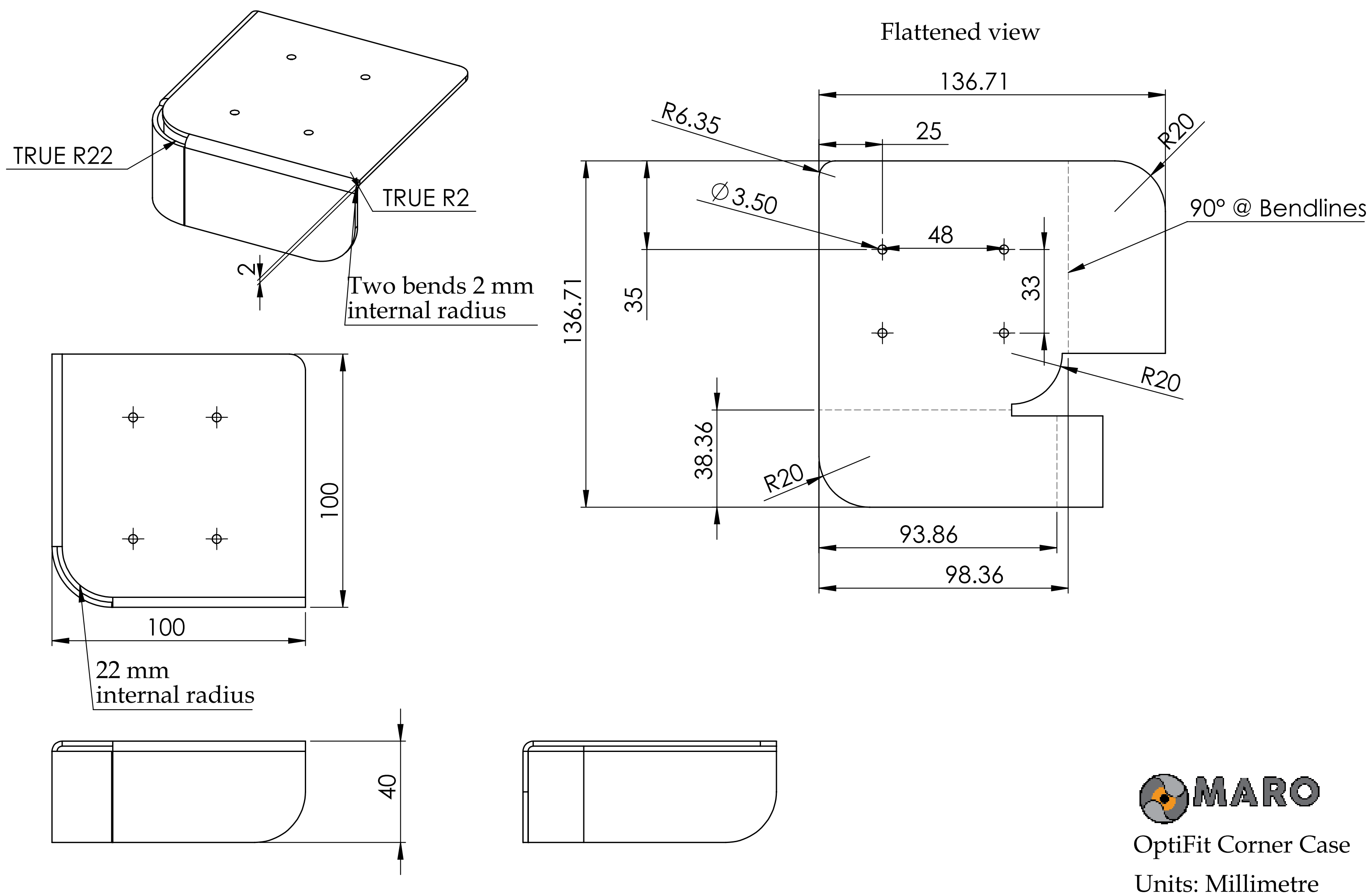
OMARO

OptiFit Strapside Case

Units: Millimetre

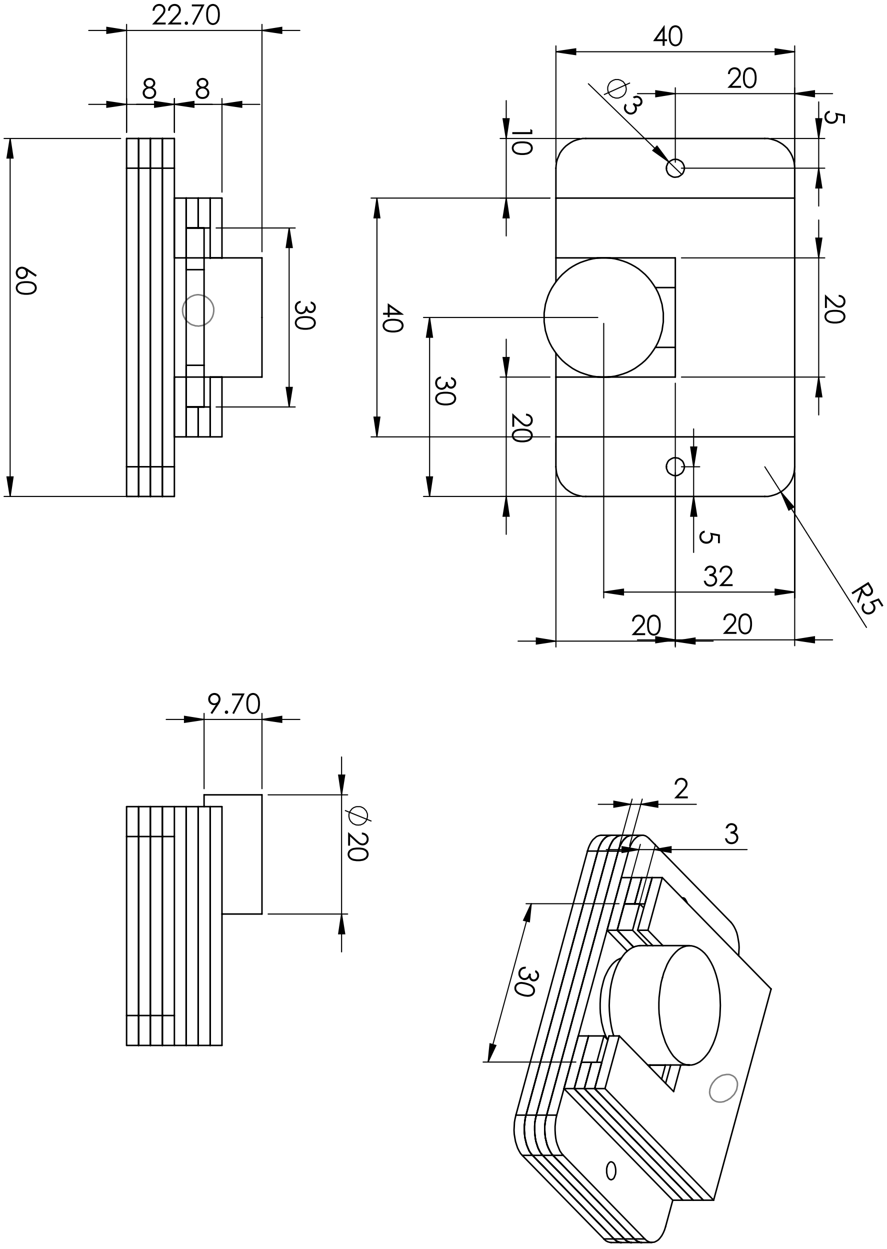


OptiFit Force Pad
Units: Millimetre



OptiFit Corner Case

Units: Millimetre



OMARO

OptiFit Force Sensor

Units: Millimetre