DAEDALUS
Technologies

T. 1-778-990-7688
F. 1-778-385-0432
SUPPORT 1-778-388-3751

LARRY ZHAO
CHIEF EXECUTIVE OFFICER
LFZ2@SFU.CA

November 19, 2011

Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, B.C. V5A 1S6

Re: ENSC 440, Daedalus Technologies Design Specifications: Display Augmentation System

Dear Dr. Rawicz:

The attached document describes the design specifications related to our Display Augmentation System, a product of Daedalus Technologies. Our product is designed with both innovation and robustness in mind. We are aiming to provide end users with increased comfort and convenience in front of a visual display unit while helping reduce the likelihood of work related injuries.

The attached design specifications describes in detail how we plan to build the Display Augmentation System. These specifications were generated after careful analysis of various fields and experiences. The business and engineering teams at Daedalus Technologies are using this documentation to guide the design and development of our device.

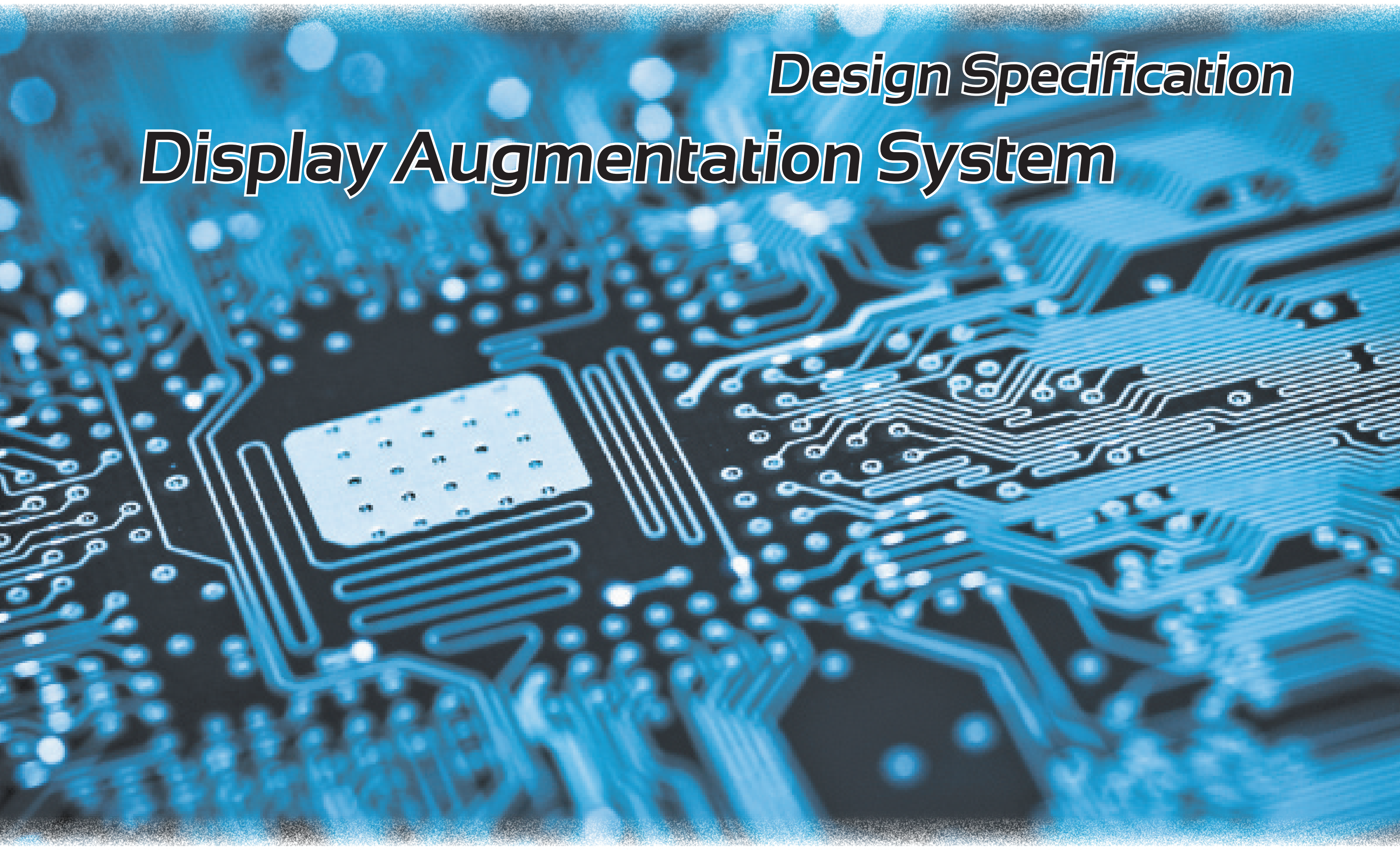
Daedalus Technologies is founded by fifth year Engineering Science students: Larry Zhao, Ian Brown, Calvin Ho, and Jordan Anguelov. Should you have any questions or concerns regarding our project proposal, please feel free to contact me by phone at (778) 990-7688, by e-mail at lfz2@sfu.ca or at Daedalus-tech@googlegroups.com.

Sincerely,

Larry Zhao

Chief Executive Officer
Daedalus Technologies

Enclosure: Daedalus Technologies Design Specification: Display Augmentation System



## Design Specification

# Display Augmentation System

DAEDALUS
Technologies

**Prepared for:**

| | |
|---|---|
| **Andrew Rawicz (ENSC 440)**<br>Simon Fraswer University<br>School of Engineering Science | **Mike Sjoerdsma (ENSC 305)**<br>Simon Fraswer University<br>School of Engineering Science |

**Prepared by:**

| | |
|---|---|
| **Larry Zhao**<br>Chief Executive Offier | **Jordan Anguelov**<br>VP Hardware |
| **Ian Brown**<br>VP Finance | **Calvin Ho**<br>VP Software |

**Date:** November 19, 2011
**Revision:** 1.0

# Executive Summary

The designs presented in this document will focus mainly on requirements related to the completion of a fully operational proof of concept model as stated in our Functional Specification document. This is because requirements with priority 1 are critical for core system functionality whereas most priority 2 and all priority 3 are geared towards a commercial model meant for mass production. As a result, the following document will pertain to a device designed for core functionality with an advanced user rather than customization by an end user.

Our Display Augmentation System, or simply DAS, will be the combination of a software application and custom hardware. Utilizing MATLAB, the software will consist of a Graphical User Interface(GUI) utilizing a webcam and face tracking algorithm. Ideally, any webcam with supported drivers will be usable by our application. This will allow users to use existing webcams or video capture devices rather than buying more proprietary hardware so they do not have to worry about adversely affecting image acquisition and face tracking. The results of our face tracking algorithm will be combined with other algorithms designed for the DAS by the engineers at Daedalus Technologies. Some additional processing is needed to help handle movement and multiple faces. Without additional processing, the data sent to the controller in the stand will most likely be unreliable and cause problems for end users unfamiliar with our product's inner workings. The controller that the computer will be communicating with for this proof of concept model will be an Arduino Mega 2560 with custom shield board housed in a custom stand designed by Daedalus Technologies. The stand itself will be Video Electronics Standards Association (VESA) compliant and crafted using the equipment available to members of Daedalus Technologies.

All of the components will be purchased from reputable dealers which adhere to standards mentioned in our functional specification documentation. By insuring that standards adhere to all stages of the development process, we can insure the highest quality product with the lowest chance of failure. This is very important to us here at Daedalus Technologies as we seek to implement a cost-effective and innovative product that appeals to a large audience.

# Table of Contents

# List of Figures

# List of Tables

# Glossary

AC - Alternating current
CPU - Central Processing Unit
DAS - Display Augmentation System
DC - Direct Current
DDR - Double Date Date
DVI - Digital Visual Interface; a video interface standard
EEPROM - Electrically Erasable Programmable Read-Only Memory
FPS - Frame Per Second
GB - Gigabyte
GPIO - General Programmable Input output
GPU - Graphical Processing Unit
GUI - Graphical User Interface
HDD - Hard Drive
IC- Integrated Circuit
ICSP - In-circuit serial programming
LED Light Emitting Diode
MATLAB - short for Matrix laboratory
Mbps - megabytes per second
MCR - MATLAB Compiler Runtime
MHz - Megahertz
OpenCV- Open Source Computer Vision Library
PCB - Printed Circuit Board
Port - an application-specific or process-specific software construct serving as a communications
PSU - Power Supply Unit
PWM - Pulse Width Modulation
RPM - Rotation Per Minute
RS232 -Recommended Standard 232
TBD - To Be Determined
UART - Universal asynchronous receiver/transmitter
UAT - User Acceptance Test
USB - Universal Serial Bus
VESA - Video Electronics Standards Association
VDU- Visual Display Unit.

# 1.0 Introduction

Daedalus Technologies' Display Augmentation System (DAS) is a revolutionary concept merging face tracking, automation and comfort that a modern computer user should have at their disposal. It is designed to be a scalable and adjustable platform to suit the daily needs of business professionals and casual users alike. Convenience and functionality are the two driving forces behind the products design. By allowing users to choose their viewing position and keep it that way, they can work with ease or enjoy a movie at home while not having to sacrifice comfort for the hustle and bustle of their activities.

## 1.1 Scope

This document outlines the design specifications to achieve a working proof of concept model as outlined by the functional specification document submitted previously. To achieve our goal, we will be implementing first and second priority requirements from our Functional Specification document.

## 1.2 Intented Audience

This document is intended for use by the engineers at Daedalus Technologies and all others who may become involved in constructing the DAS proof of concept model. This document will also include the necessary information on conduct thorough test cases on a DAS proof of concept model.

# 2.0 System Overview

An overview diagram outlining the various components and their interconnections for the DAS is shown in the figure 2.0.1 below. A more detailed explanation of the individual hardware and software components is explained more in the section below.

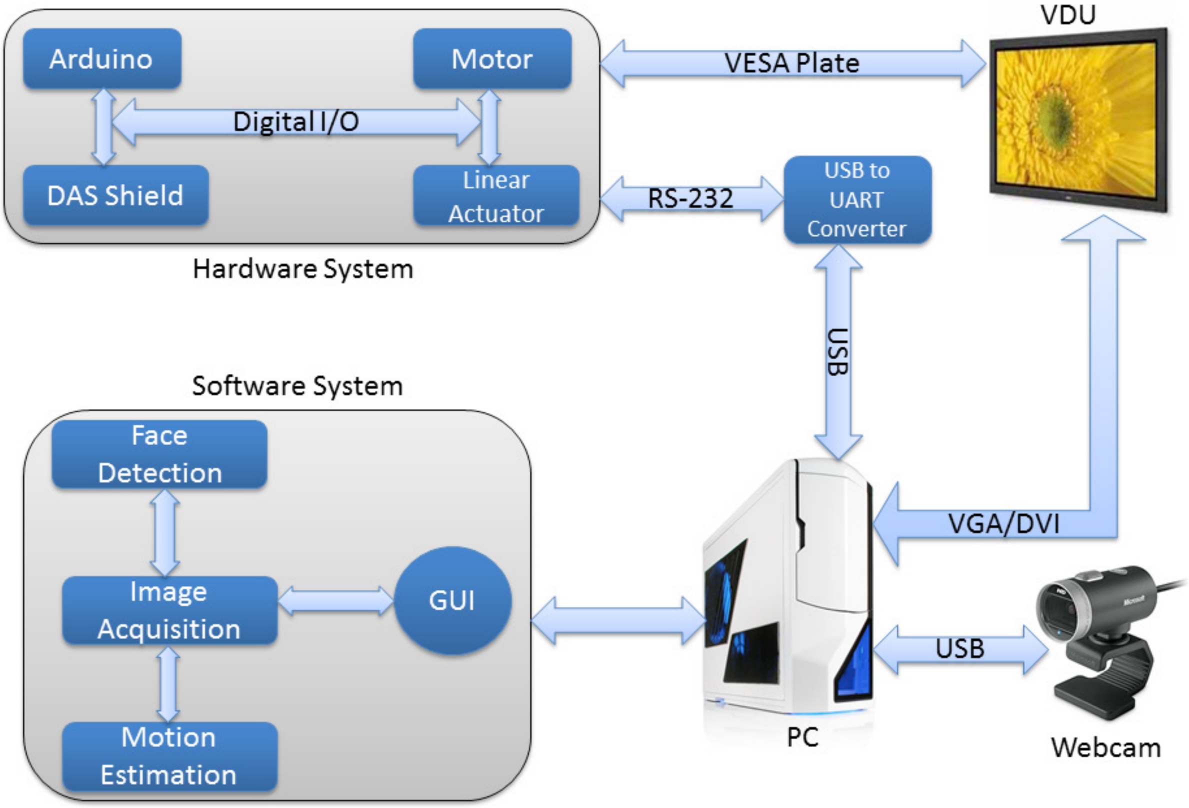


Figure 2.0.1 System Overiew

# 2.0 System Overview

The DAS concept model as shown in figure 2.0.1, is a fusion of software and hardware systems communicating via a USB to RS232 cable.

On the software side, a GUI manages the image acquisition, face detection and user interaction. In addition, it contains features such as a selectable timer to signal the user that a break is needed. The GUI also provides manual position control for the user to interact with. The image acquisition feature takes video feed from the Microsoft Lifecam Cinema webcam connected via Universal Serial Bus(USB) and periodically takes snapshots. These saved images will then be fed to the face detection algorithm which will draw a rectangular box around the user's face. The largest face will be returned with positional data to the main GUI program.  With this positional data, the motion estimation algorithm will take over to calculate the distance the user has moved with respect to the previous image. With the position value, the GUI then sends this data through the USB port via USB to RS232 converter into the Arduino microcontroller unit.

On the hardware side, the Arduino Mega, in conjunction with the DAS shield inside the DAS stand itself, will control a linear actuator, motor function and mechanical feedback. Communication between the Arduino and GUI operate using a strict protocol in development by the engineers at Daedalus Technologies. Power supplied to our DAS unit is through a 12V, 20 watts power supply unit connected to an electrical outlet.

A developed 3D model of DAS is shown below in Figure 2.0.2. The physical characteristics of the device will be presented and discussed in detail in the mechnical section of this document.

# 2.0 System Overview

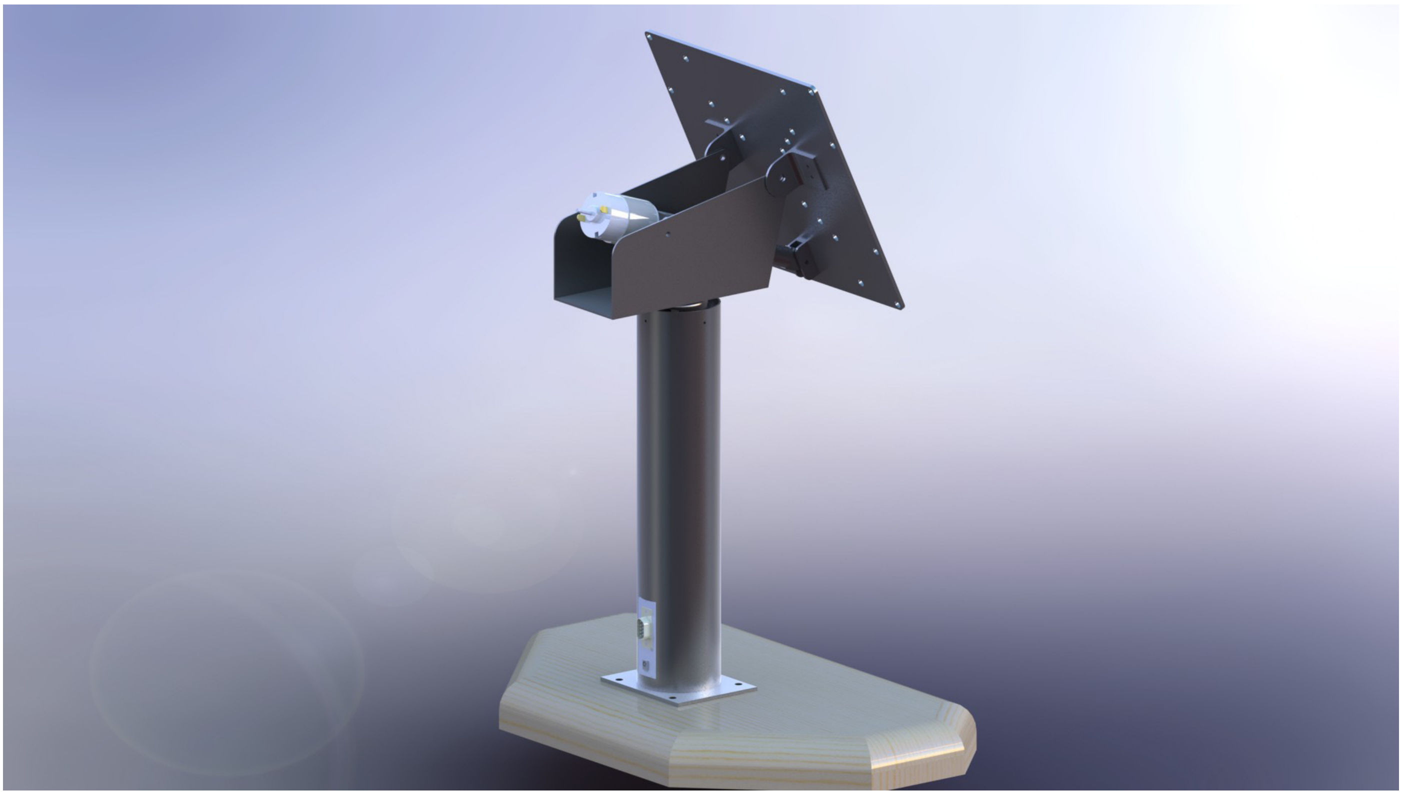


Figure 2.0.2 System Overeiw

# 3.0 Hardware

The hardware section provides detailed information on the exact design specification of the DAS's hardware requirements. This includes information about electrical components and peripheral device requirements to standardize development as agreed upon by the engineering team at Daedalus Technologies. Various features and specifications are given for each component along with a rationale for why these specific components were selected. Many of these parts were over specced and are made of high-grade components to test the limits of the DAS design.

## 3.1 Printed Circuit Board

A key element of the electrical system is the custom shield board called: DAS shield, constructed for our product, the 3D rendering is shown in figure 3.1.4. The custom shield board will also allow for access to all Arduino pin outputs. The DAS shield houses various components, please refer to figure 3.1.5 and 3.1.6 for more detailed layouts and dimensions. Section 3.1.1 to 3.1.4 covers the main integrated circuit(IC) components featured on the DAS Shield.

### 3.1.1 H-Bridges

The most notable of the PCB shield components are the two Infineon BTM7752G H-Bridge motor drivers[1] see figure 3.1.6 labelled under the H-bridge section. The primary function of the H-Bridges is driving the two DC motors in both forward and reverse directions. While there are many H-bridge chips on the market we opted for the Infineon offering because it can handle up to 12A and up to 28V and has the following features[1]:

- Status flag diagnosis with current sense capability
- Overtemperature shut down with latch behaviour
- Overvoltage lock out
- Undervoltage shut down
- Switch-mode current limitation for reduced power dissipation in overcurrent situation
- Integrated dead time generation

We are confidant that these features are a good choice for the proof of concept model since it can help prevent engineering errors due to the various built in protection features.

## 3.1.2 Op - Amps

The DAS Shield makes use of two LM358 (SOIC8 surface mount package) general purpose operational amplifiers[2](see figure 3.1.6) to increase the gain of the IR proximity sensor to maximize the resolution of the ADC on the Arduino. The IR proximity sensor outputs analog voltages up to 2.7V with a +5V supply and the Arduino ADC reads voltages from 0-5V; therefore a non-inverting op-amp as shown below in figure 3.1.2 is utilized with a gain of approximately 1.6 . In our system, we used R2 of 4.75K Ohms and R1 of 8.25K Ohms. From the equation in figure 3.1.3 [3] our gain is 1.57.

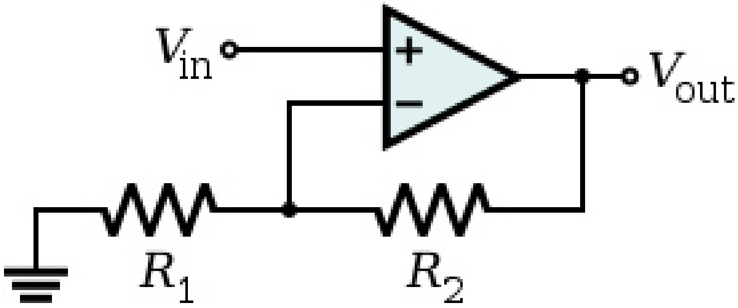


Figure 3.1.2 Op-Amp Diagram

$$V_{out} \approx \frac{V_{in}}{\beta} = \frac{V_{in}}{\frac{R_1}{R_1+R_2}} = V_{in}(1 + \frac{R_2}{R_1})$$

Figure 3.1.3 Op-Amp Equation

## 3.1.3 RS-232

While the Arduino has several serial ports, the DAS only makes use of Serial Port. The Arduino serial port is 0-5V logic; to communicate with other devices, voltages must be brought to RS-232 standard logic levels of 12V. To do this, a Texas Instruments MAX232 multi-channel RS-232 buffer and line driver chip is used [4] refer to figure 3.1.6 under section RS-232 serial port.

## 3.1.4 Buck Converter

The MAX232 chip and LM358 op amps need a 5V supply to operate. The Arduino has an on-board 5V Linear Drop Out (LDO) regulator to supply the ATmega2560 microcontroller; however, the LDO is not capable of supplying sufficient current. As a result, we opted to add a separate 5V supply. The buck converter is based around the Texas Instruments TPS54232, it takes from 3.5V to 28V input, 2A, 1MHz Step-Down Converter[5], which takes the 12V input and steps it down to 5V refer to figure 3.1.6 under section power input and buck converter.  The output voltage is programmed by a voltage divider.

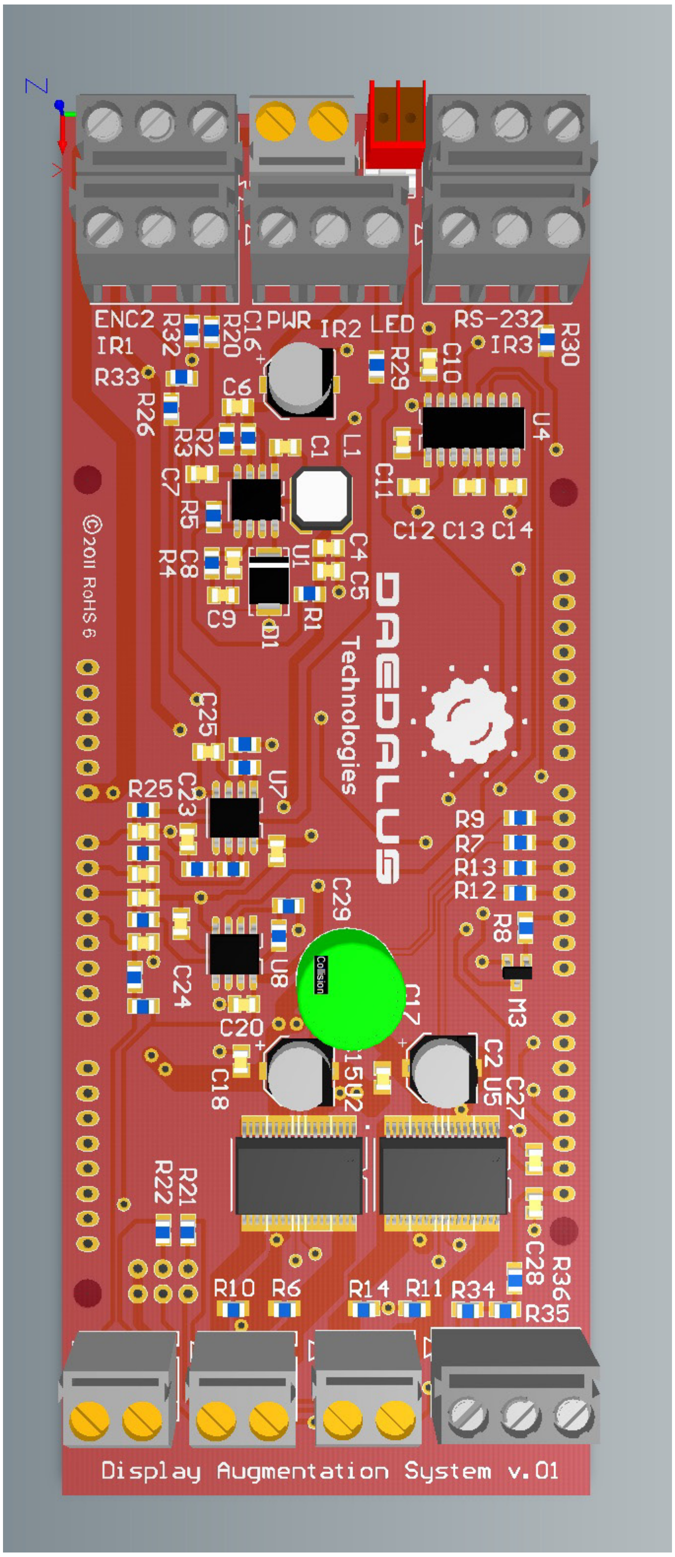


Figure 3.1.4 DAS Shield Rendering

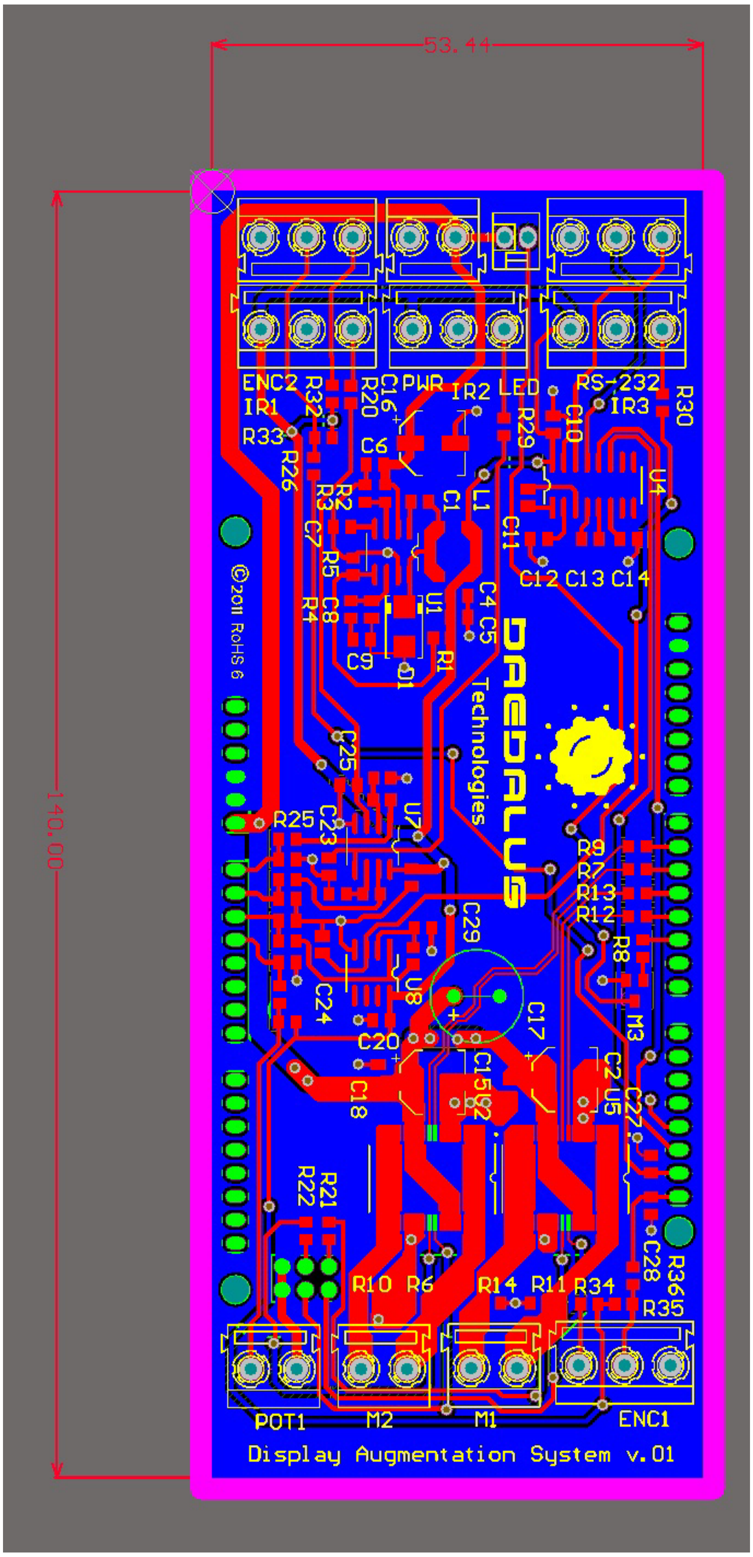


Figure 3.1.5 PCB Dimension(in mm) and Tracks

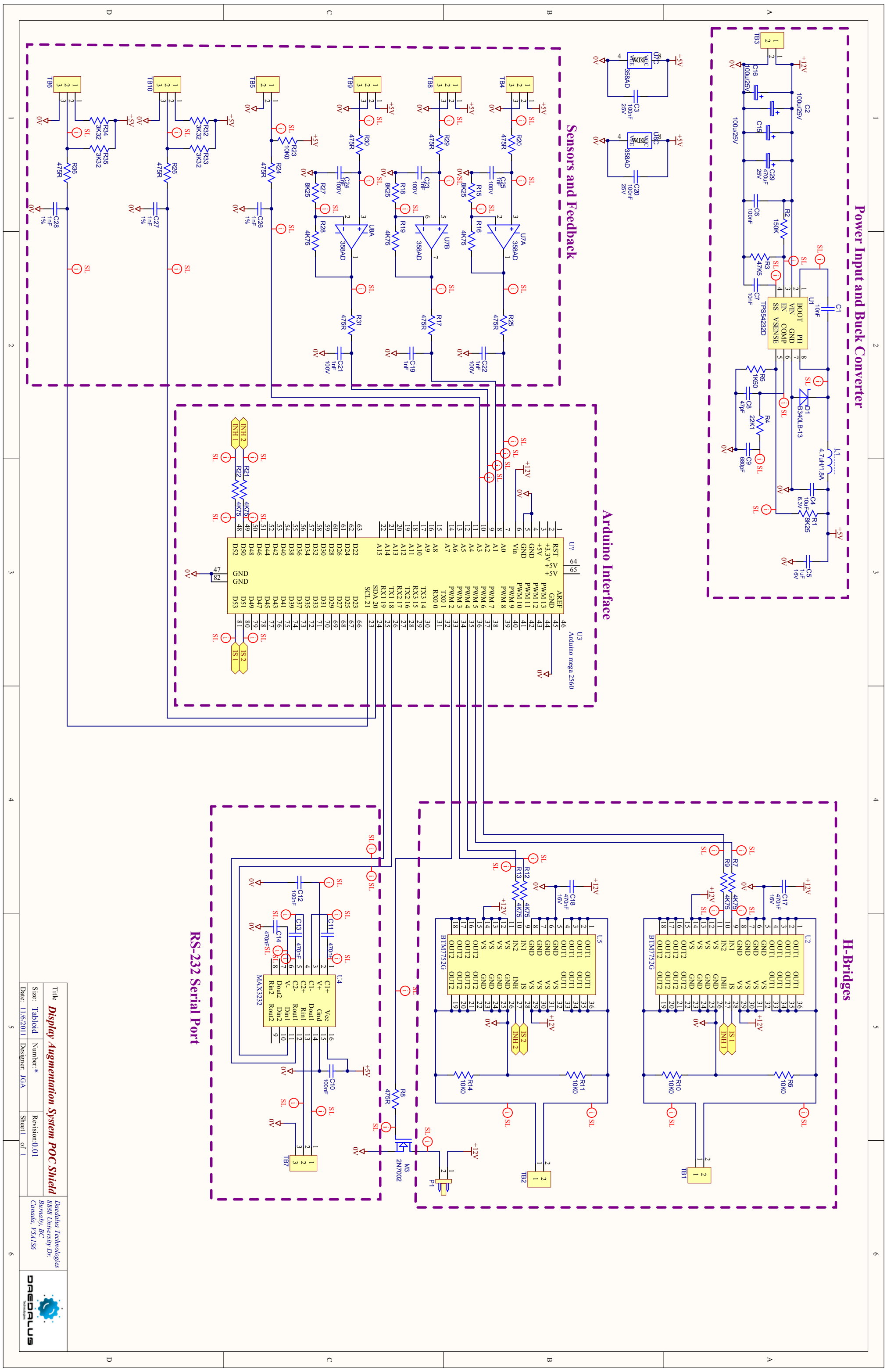


Figure 3.1.6 PCB Circuit Layout

## 3.2 Arduino

"Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments" [6]. The Arduino Mega 2560 uses the ATmega2650 microcontroller(see figure 3.2 below), which is a low-power 8-bit microcontroller made by the Atmel Corporation [7]. The Arduino platform uses a java-based IDE and a C/C++ like language to make prototyping easier as the engineers at Daedalus Technologies have experienced first hand. This is accomplished by having a wide variety of built-in functions at our disposal and many of the ATmega features already implemented in easily executable functions. For this proof of concept model, ease of implementation was favored over code efficiency and size which made the Arduino platform an ideal choice.

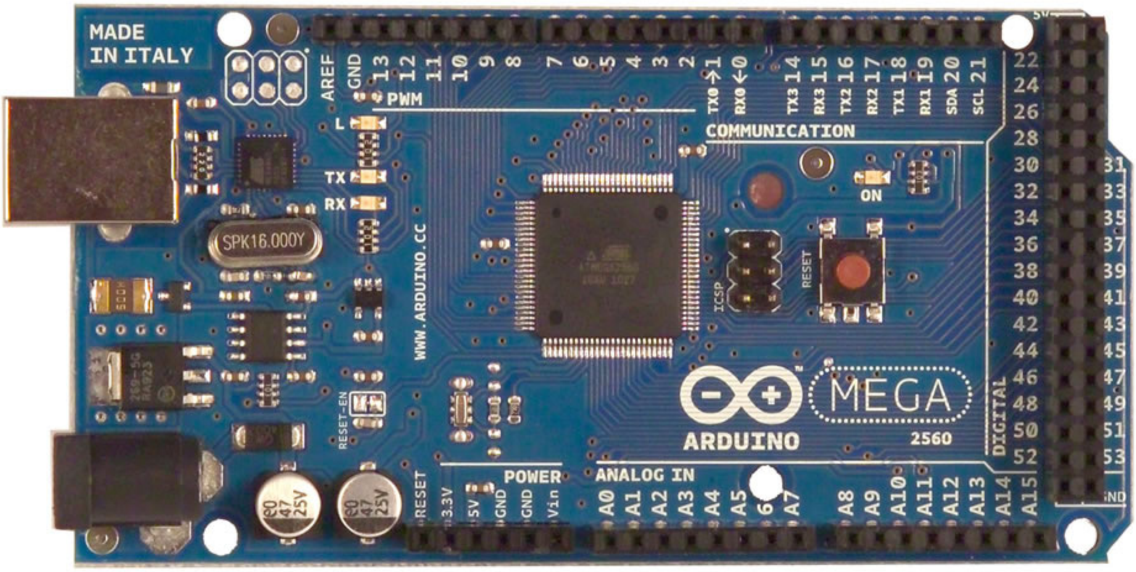


Figure 3.2 Arduino Mega 2560 [6]

The following table is a summary of the Arduino Mega 2560 specifications and features.

| **Microcontroller** | **ATmega2560** |
|---|---|
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limits) | 6-20V |
| Digital I/O Pins | 54 (of which 14 provide PWM output) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 256 KB of which 8 KB used by bootloader |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |
| Additional Accessories | USB Connection<br>5V Power Jack<br>ICSP Header<br>Reset Button<br>Overvoltage Protection |

Table 1 Arduino Mega 2560 Specifications [7]

The DAS's Arduino will be primarily used for motor control and mechanical feedback while communicating with a nearby computer via the custom shield board designed for this. For this first proof of concept model, DAS will be utilizing the ATmega2560's serial ports, PWM pins, and analog input pins via DAS shield.

## 3.3 Webcam

Any webcam with supported drivers will be usable by this system. For this proof of concept model, we are using a Microsoft Lifecam Cinema (see figure 3.3) because of its fast response time and wide range of capabilities. With a wide 73 diagonal degree field of view(FOV), 2 mega pixel HD sensor, auto focus, high speed USB 2.0 and 1280x720 pixels video with up to thirty frames per second (FPS), this webcam has provided excellent images for our proof of concept model[8]. It also comes with a unique mounting feature which fits almost all VDUs. The option to mod it with a wide angle lens to expand the FOV even greater[9], make this webcam an ideal choice for our proof of concept model.

Figure 3.3 Microsoft Lifecam Cinema [10]

## 3.4 USB to RS232

The RS232 cable will allow our GUI to communicate with our microcontroller which in turn will manage all other hardware functions. This converter will allow USB signals from the designated USB port to be converted into serial data which the microcontroller will be able to understand. The MAX 232 chip will transform out 0-5V logic to RS232.

For the proof of concept model, we chose the CP2110[11] USB to RS232 bridge (see figure 3.4 below) from Silabs as the converter is galvanically isolated (1500V) and has the following features:

- USB 2.0 compliant, full-speed (12 Mbps)
- No external crystal required
- Up to 1024 Bytes of EEPROM
- User-programmable custom Baud rates
- Supports all modem interface signals

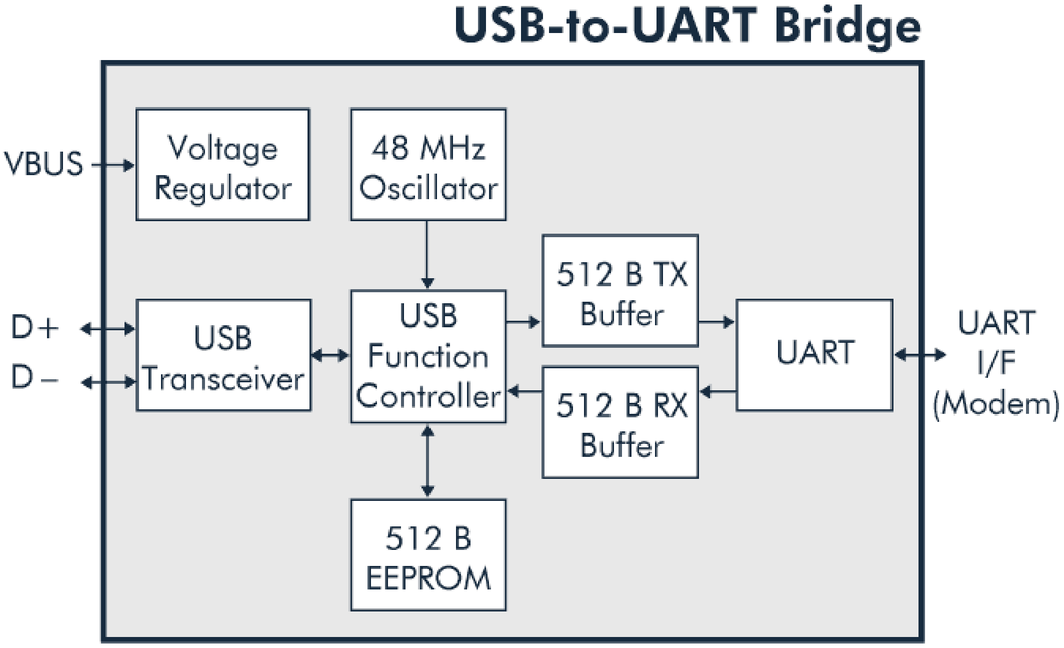


Figure 3.4 USB to RS232 Bridge [11]

## 3.5 Power Supply Unit

The power supply used is a 12V DC, 24W one like the one seen in the figure 3.5. From our functional specifications, we suggested that our system would not consume more than 10W ([R58-I]). Since this is a proof of concept model we opted for a slightly large power supply in case we needed more power. Because this selected power supply is UL/FCC/CE approved it must by default meet our stringent requirements ([R44-I], [R45-I] and [R60-I]).

Figure 3.5 Generic 12V DC, 24W PSU unit [12]

## 3.6 Proximity Sensor

Sharp GP2Y0A02YK0F is a distance measuring sensor unit composed of an integrated combination of position sensitive detector, infrared emitting diode and signal processing circuit as shown in the figure 3.6. This sensor has an analog output that varies from 2.8V at 15 centimeter to 0.4V at 150 centimeter with a supply voltage between 4.5 and 5.5 V[13]. The reason we choose this Infrared sensor from Sharp is due to price and distance detection specification, which fits our application. Ideally the user should be at arm's length away from the monitor[14], and if the user is no longer present in front of DAS, our Infrared sensor will also help us determine if self-shutdown protocol should be executed because the user is no longer present in front of DAS. It has the follow specifications:

- Distance measuring range : 20 to 150 cm
- Analog output type
- Package size : 29.5 13 21.6 mm
- Consumption current : Typ. 33 mA
- Supply voltage : 4.5 to 5.5 V

Agency approvals/Compliance
1. Compliant with RoHS directive (2002/95/EC)

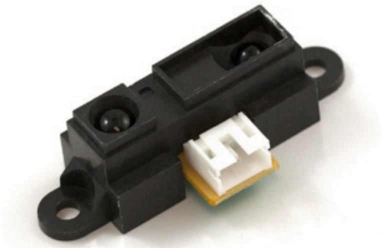


Figure 3.6 Sharp IR Sensor [13]

## 3.7 Computing Unit

A desktop or mobile processor is required for the GUI to run and execute image acquisition as well as running the face detection algorithm. For our proof of concept model, we decided to use IBM T60 laptop as shown in figure 13.

Figure 3.6 IBM T60 [15]

Test computer's specifications:
Processor: Intel Centrino Duo T2500
Memory/RAM: 2GB DDR2 RAM
Hard Drive: 100GB HDD
Graphics: ATI Mobility Radeon X1400

*When the GUI is running, it requires about 9% central processing unit (CPU) usage on a single core,  due to MATLAB being mostly a single threaded application[16].

After doing the necessary market research, we found that the cheapest consumer CPU offered by AMD and Intel was Sempron 145 and Celeron G440 respectively[17]. Both were released in 2010 and 2011 respectively[18]. Although they are not dual core by nature, their clock vs clock and core vs core speed is significantly faster[19]. This is due to the newer 32 nanometer fabrication process and more efficient and innovative architecture design (such as the newly released sandy bridge architecture) than the tested Centrino Duo T2500 processor, which was released back in 2006 with the "Yonah" architecture and 65 nanometer fabrication process[20]. Also, a graphic processing unit (GPU) is not required, due to MATLAB being 100% CPU dependent, because "most MATLAB graphics are rendered on the CPU, not the GPU, so you can use any recent graphics card with MATLAB. The performance improvement on a high-end graphics card is relatively modest when compared to an entry-level graphics card"[21].

# 3.0 Hardware

As a result, engineers at Daedalus Technologies are able to determine the minimal PC requirement to run DAS's control software package. These specifications are carefully chosen so that our program will not interfere with normal desktop computing usage.

DAS control center requires a PC that meets the requirements for and has installed one of these operating systems:

- Microsoft Windows® 7, Windows Vista®, and Windows XP Service Pack 2
- Intel or AMD 1.6GHz
- 1.0 GB hard drive space
- USB 2.0
- Broadband Internet access required, access fees may apply or CD-ROM for GUI installation.

MATLAB software package and toolboxes are not required to run our software system. We use MATLAB Complier (more details in section 5) to distribute our program as an executable or shared library. Support for other operating system is also possible, but we have not yet tested this on other platforms and it is not part of the functional requirement.

# 4.0 Mechnical

## 4.1 Mechnical Overview

The biggest challenge in the creation of DAS was the mechanical design. The design called for a moving support stand for VESA compatible monitors which must pan and tilt. This meant that the mechanical design must be strong, dependable yet attractive and cost effective. The design implemented for the proof of concept attempted to address these challenges.

## 4.2 Base Shaft

The base shaft forms the core of our proof of concept system. Its purpose is to support our system at a fixed height of 289 millimeters as shown in mechnical drawing figure 4.2, and also to house the main electronics system, and facilitate external connectors for power and communications.

## 4.3 Pan Bracket

The pan bracket consists of two major pieces (three if motor is included). Its purpose is to connect the rotating nacelle to the fixed base shaft and motor while facilitating a smooth frictionless rotation. The two major components were combined to achieve this task, the result of which can be seen in mechnical drawing shown in figure 4.3.

### 4.3.1 Shaft Plate

The shaft plate connects the shaft, motor and bracket together. It was custom made from 6.35 millimeters aluminum plate and rounded into a circular shape to fit inside the shaft. Then 43 millimeters holes were drilled in the 4 corners of side to attach the assembly and shaft. On the forward face a center hole was cut to accommodate the shaft of the motor as well as 6 mounting holes for the motor. In addition to the 4 additional holes were made for mounting the bracket (see mechnical drawing in figure 4.3).

### 4.3.2 Bracket

The bracket connects the shaft plate (and hence the shaft) to the rotating nacelle. The motor shaft must also be attached to the bracket such that there is movement with minimal friction. The idea and requirements were tough to design, and required creative thinking. To accomplish the task, the engineers at Daedalus Technologies took a “2 inch caster wheel‟ as it can be seen in figure 4.3.2, and modified it to fit in the mounting holes of the shaft plate(see mechanical drawing figure 4.3.1). The caster wheel had to also be drilled in the center to allow for the motor shaft to be securely installed. The caster wheel was a good choice because it offed the 4 mounting holes on the top for other objects to be bolted on and it offers very smooth frictionless rotation due to its ball bearing joints.

## 4.4 Nacelle

The nacelle houses the linear actuator, linear potentiometer and joints for the VESA plate. The nacelle was made from an old rectifier enclosure which was modified slightly to meet our needs. One of the major modifications included cutting the front face at an angle, so as to accommodate the movement of the VESA plate for the required angle of tilt see mechnical drawing in figure 4.4. Other modifications included drilling holes for mounting the VESA plate, linear actuator and rotating bracket. The overall nacelle design is sturdy and strong however as a result the weight is more than is needed and future designs may include a plastic nacelle.

## 4.5 Vesa Plate

The VESA plate was made to conform to the MIS-D/E/F standards set by VESA[22]. It is attached to brackets that are attached to the nacelle. It was made from a 2 millimeter thick aluminum sheet. Holes were then drilled in an exact pattern to meet the required standards. The two brackets were made from a right angled sheet of aluminum and were drilled in both sides and mounted to the VESA plate (see mechincal drawing in figure 4.5).

## 4.6 Linear Actuator

Perhaps the most challenging part of the mechanical design was the linear actuator, the design called for a powerful linear actuator, one that could push at least 15Kg. As can be seen from mechanical drawing figure 4.6.1, our design called for a 12V DC motor to be attached to a 6.35 millimeters threaded rod, two 6.35 millimeters nuts were secured into a 12.7 millimeters aluminum tube and the threaded rod was inserted inside. The theory was that as the motor spins the shaft would push or pull the aluminum tube. The aluminum tube was in turn attached to a small "U" shaped aluminum bracket which was bolted to the VESA plate. The problem with that design was that the motor and threaded rod had to be perfectly lined up and due to mechanical challenges the threaded rod was off center by a few millimeters. This in turn created a lot of friction and put a lot of stress on the motor. As a result, the the motor was unable to properly move the linear actuator. Also another problem was that the DC motor was spinning too fast with a 12V supply and would prove difficult to control. To address these problems, the engineers at Daedalus replaced the motor assembly with a more efficient motor[23]. This motor has a gearbox and offers more torque, less noise and because of its design it was more compact, which enables it to fit inside the closed nacelle(see figure 4.1).

# 4.0 Mechnical

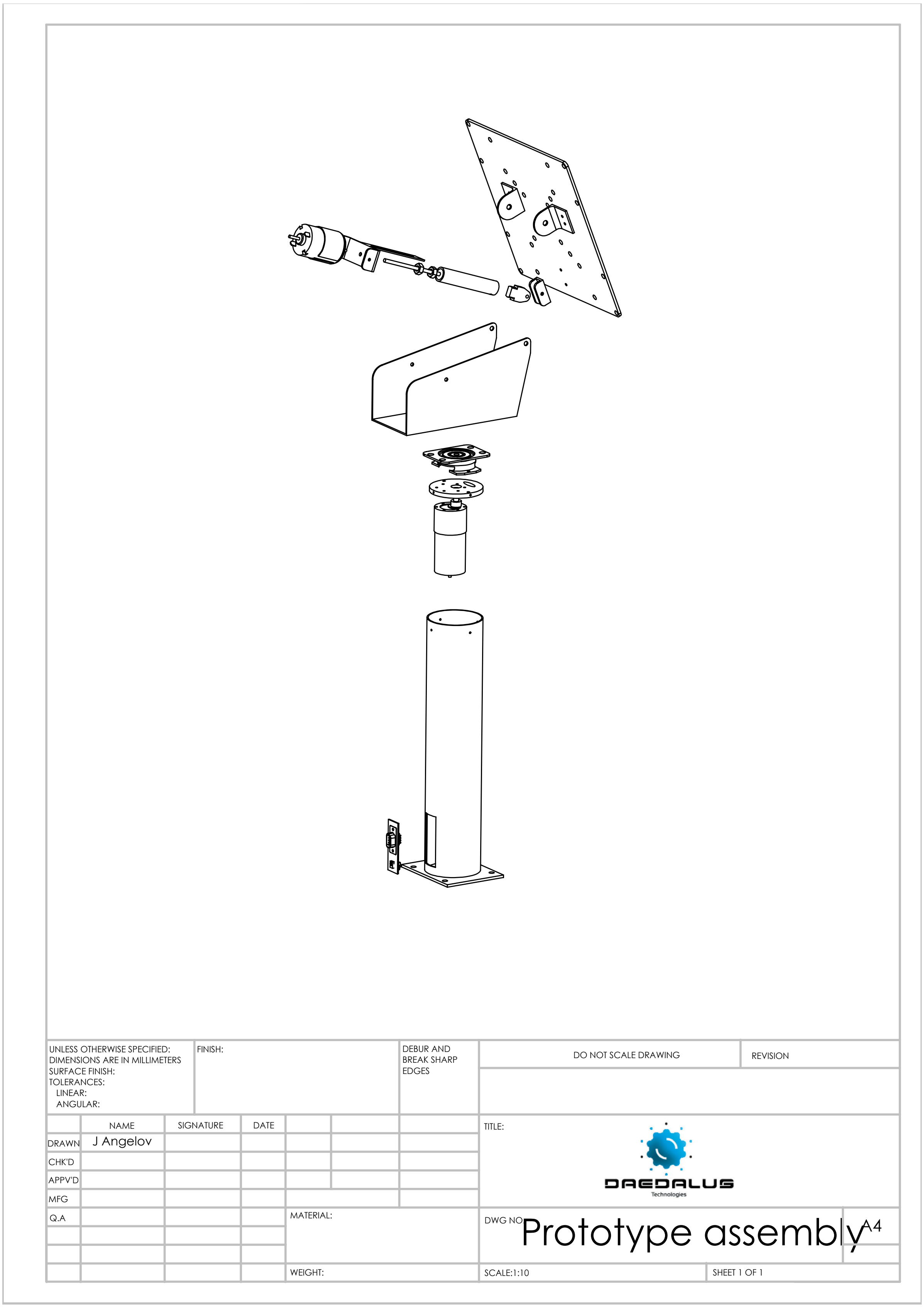


Figure 4.1 Mechnical System Assembly

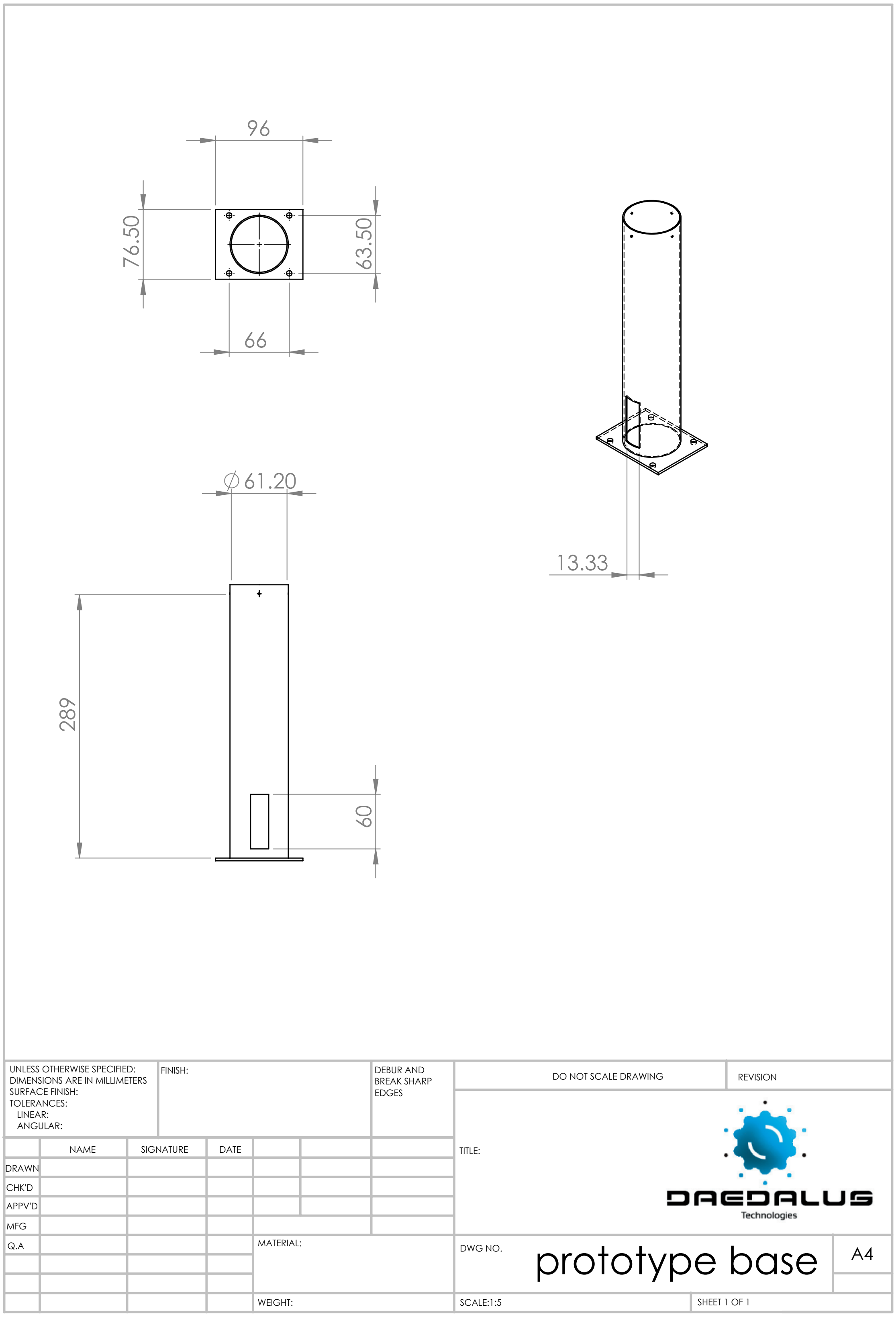
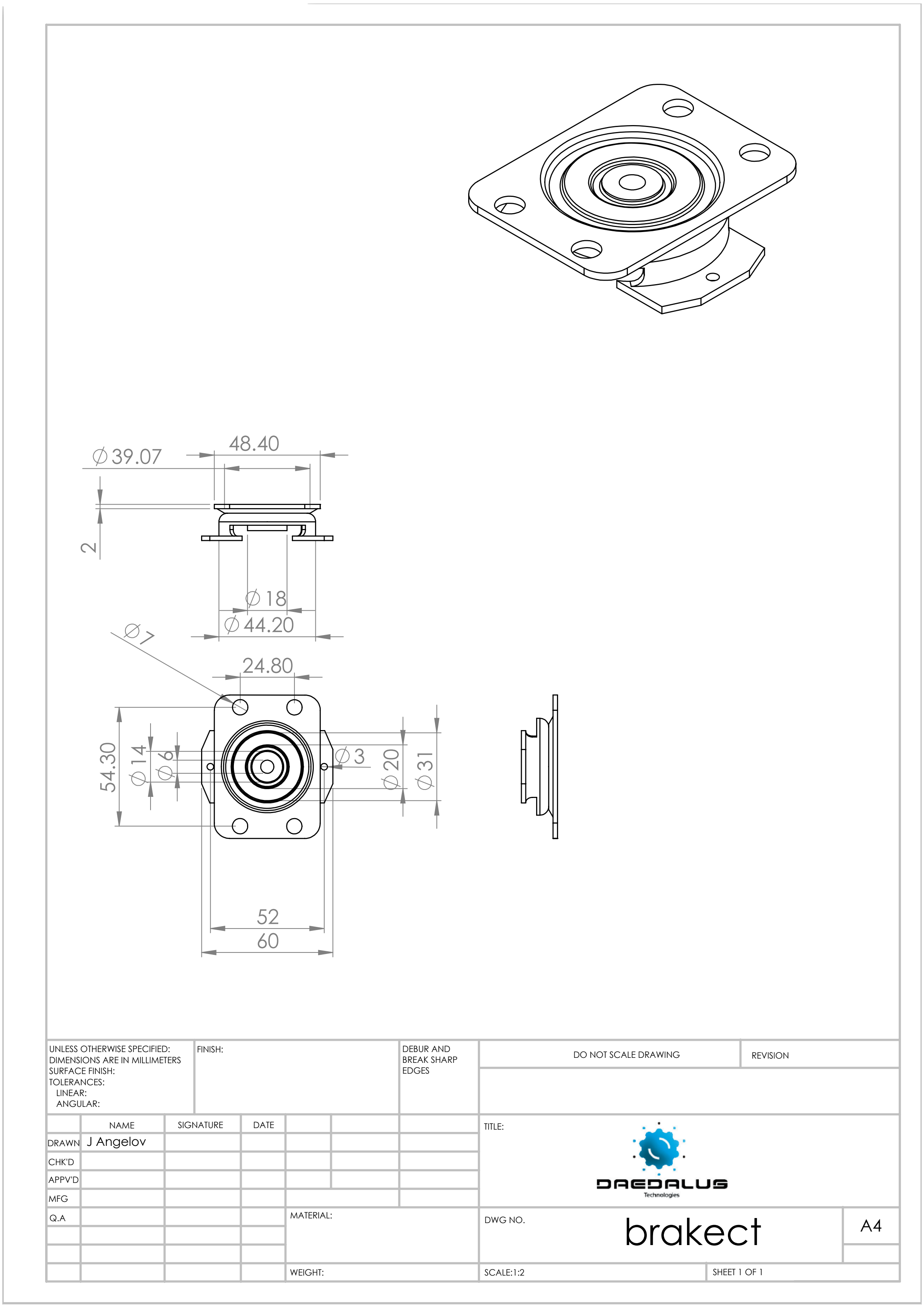


Figure 4.2 Stand



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND
BREAK SHARP
EDGES

DO NOT SCALE DRAWING

REVISION

| | NAME | SIGNATURE | DATE |
|---|---|---|---|
| DRAWN | J Angelov | | |
| CHK'D | | | |
| APPV'D | | | |
| MFG | | | |
| Q.A | | | |

TITLE:

MATERIAL:

WEIGHT:

DWG NO.

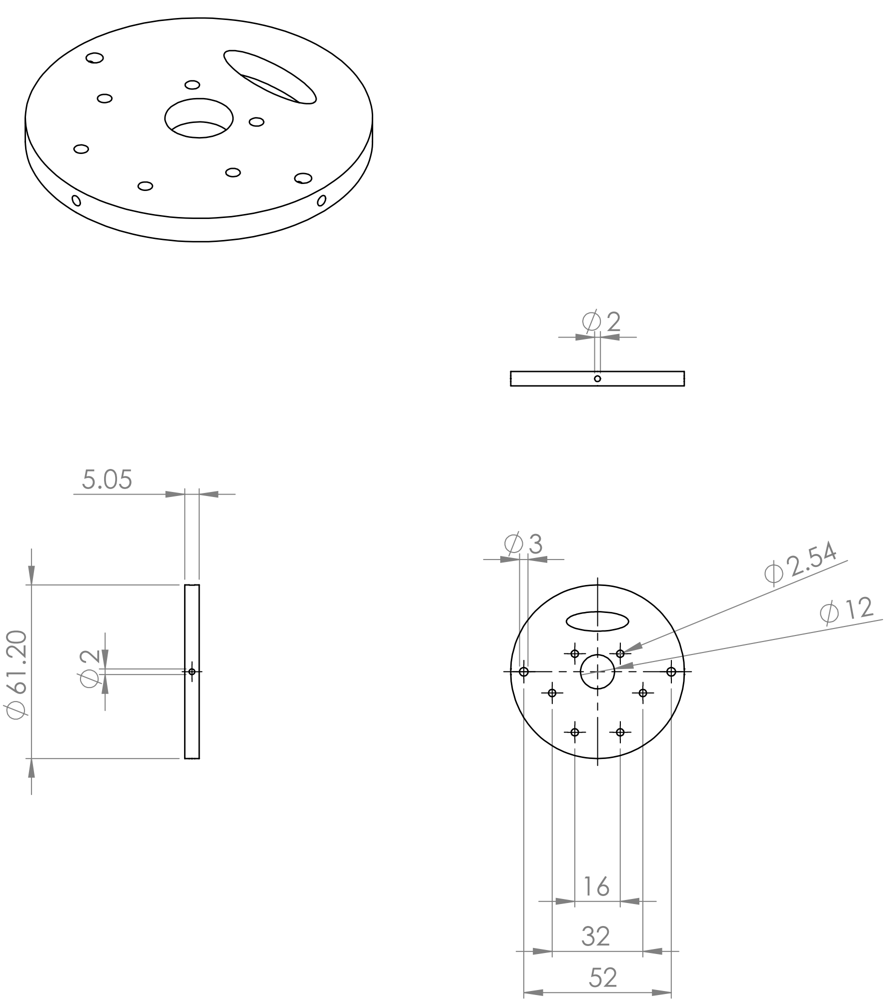
brakect

A4

SCALE:1:2

SHEET 1 OF 1

Figure 4.3 Pan Bracket



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REVISION

| | NAME | SIGNATURE | DATE |
|---|---|---|---|
| DRAWN | J Angelov | | |
| CHK'D | | | |
| APPV'D | | | |
| MFG | | | |
| Q.A | | | |

TITLE:

DAEDALUS Technologies

MATERIAL:

WEIGHT:

DWG NO.

Shaft Plate

A4

SCALE:1:2

SHEET 1 OF 1

Figure 4.3.1 Shaft Plate

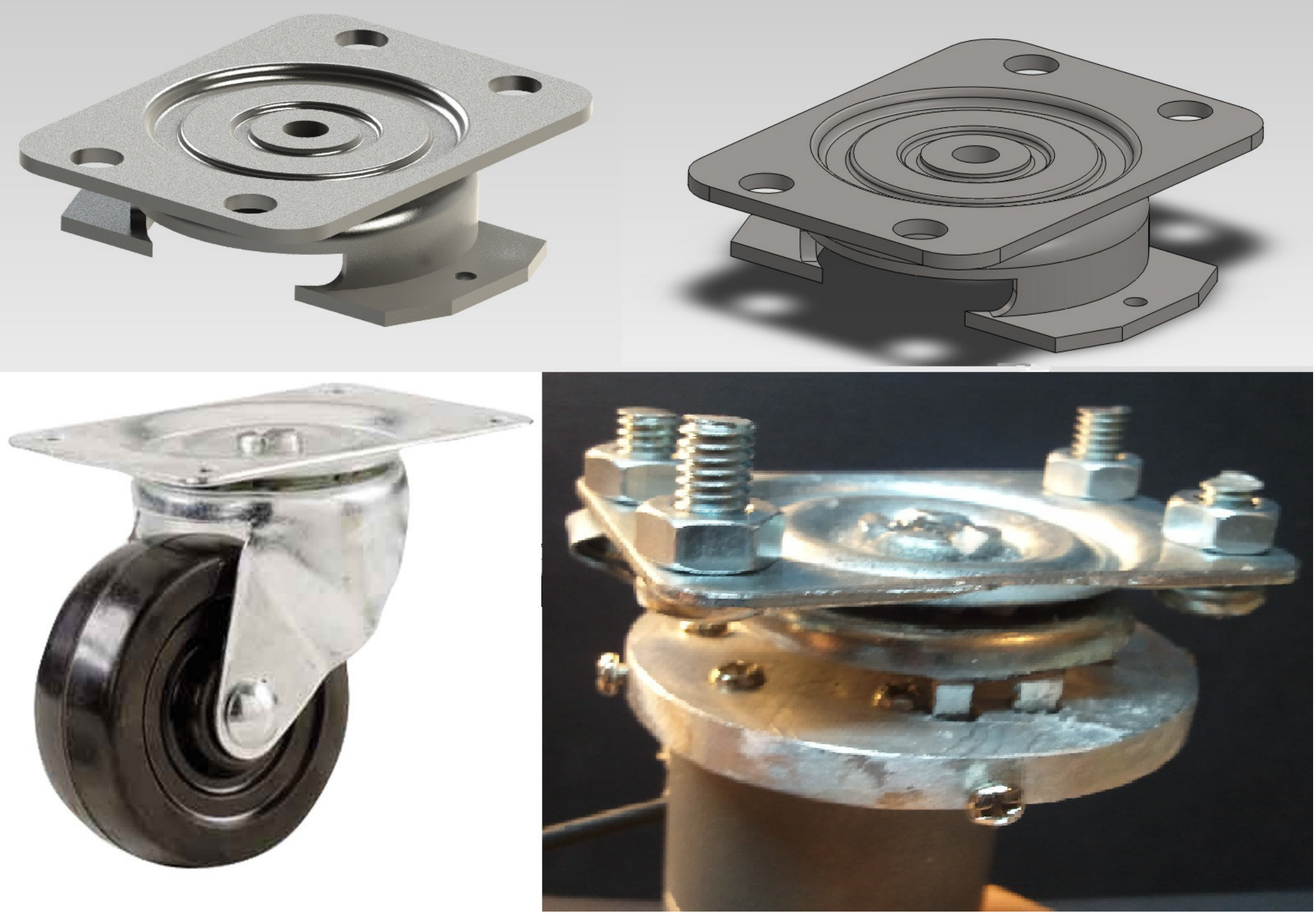
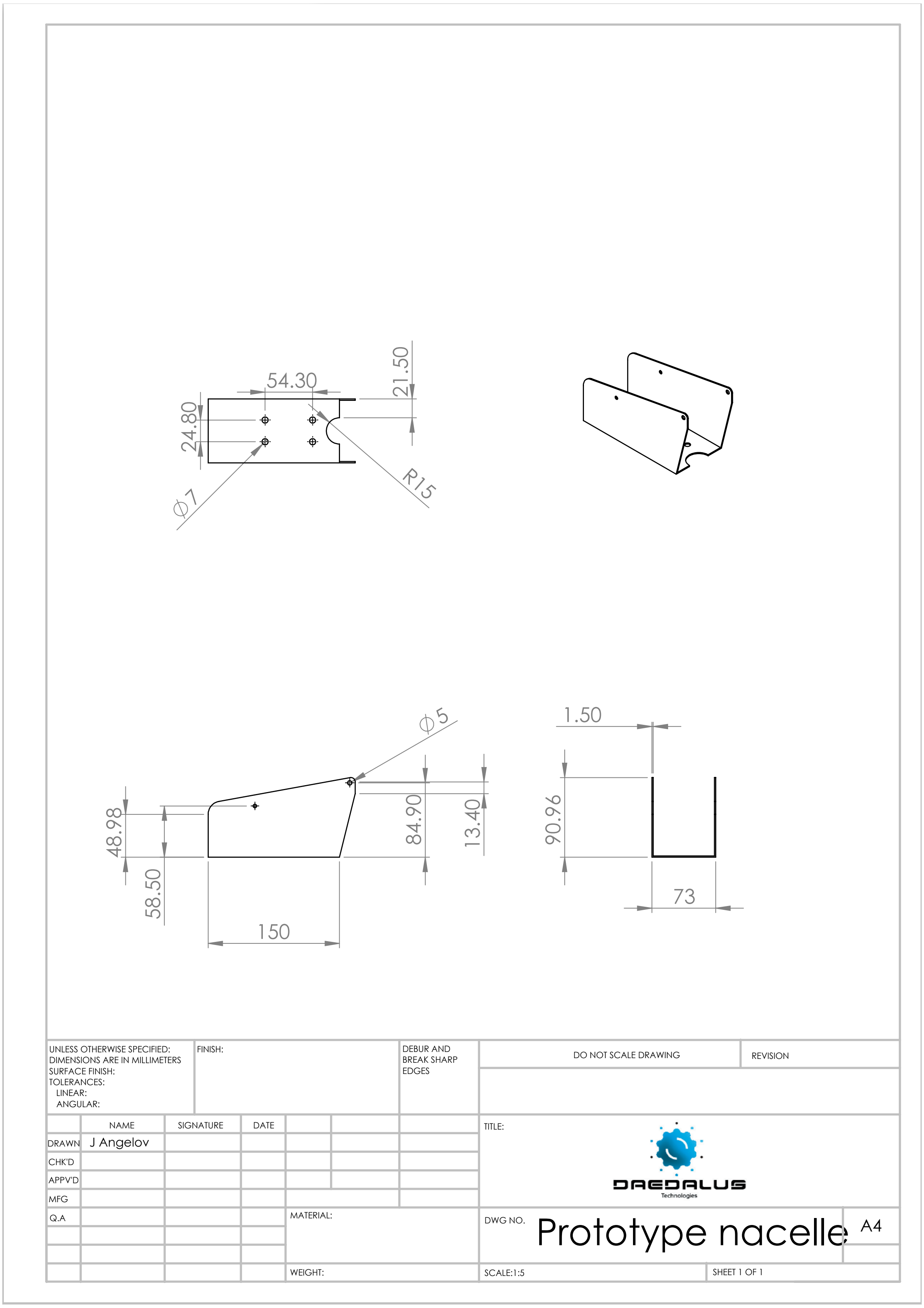


Figure 4.3.2 Caster Wheel



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND
BREAK SHARP
EDGES

DO NOT SCALE DRAWING

REVISION

| | NAME | SIGNATURE | DATE |
|---|---|---|---|
| DRAWN | J Angelov | | |
| CHK'D | | | |
| APPV'D | | | |
| MFG | | | |
| Q.A | | | |

TITLE:

MATERIAL:

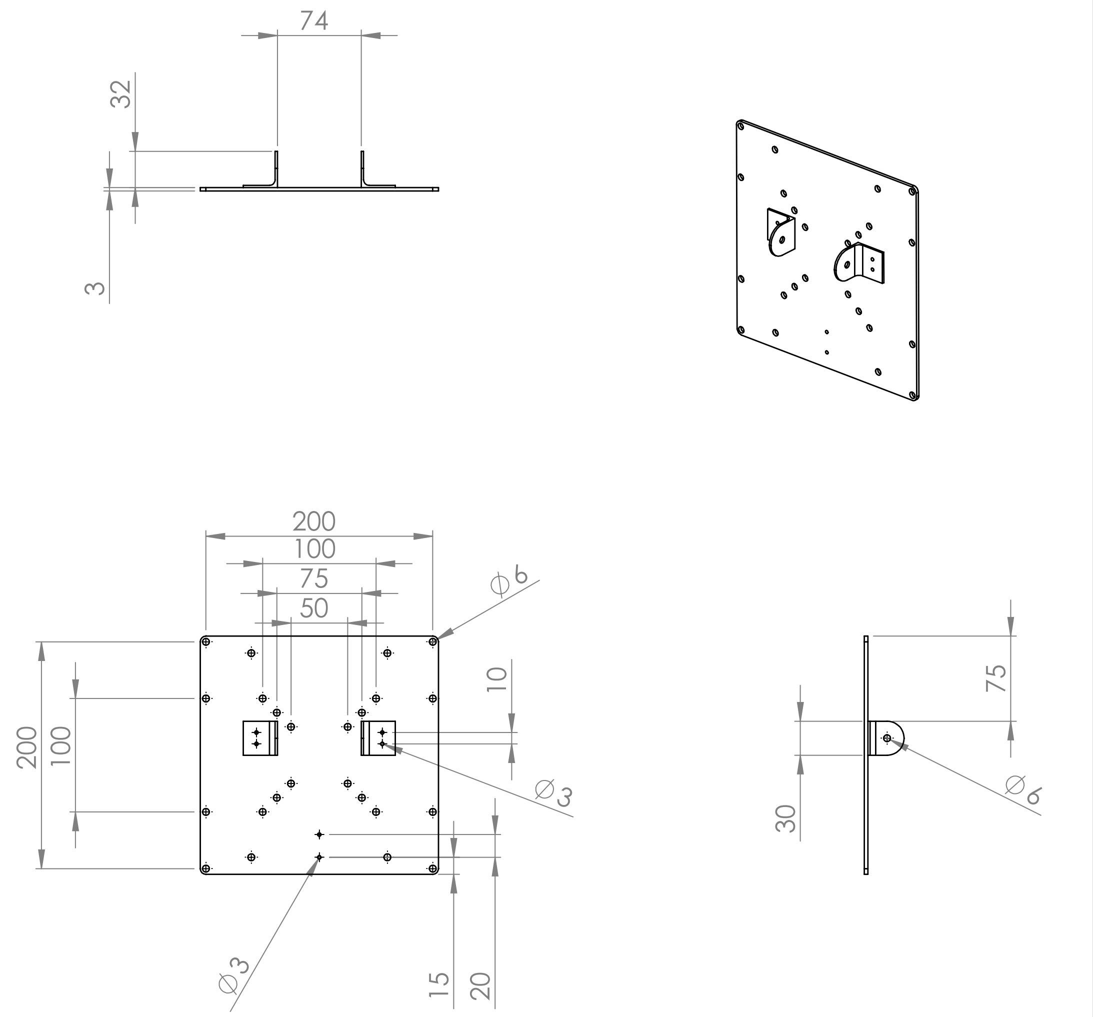
WEIGHT:

DWG NO. Prototype nacelle A4

SCALE:1:5

SHEET 1 OF 1

Figure 4.4 Nacelle



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND
BREAK SHARP
EDGES

DO NOT SCALE DRAWING

REVISION

| | NAME | SIGNATURE | DATE |
|---|---|---|---|
| DRAWN | J Angelov | | |
| CHK'D | | | |
| APPV'D | | | |
| MFG | | | |
| Q.A | | | |

TITLE:

MATERIAL:

DWG NO. VESA PLATE

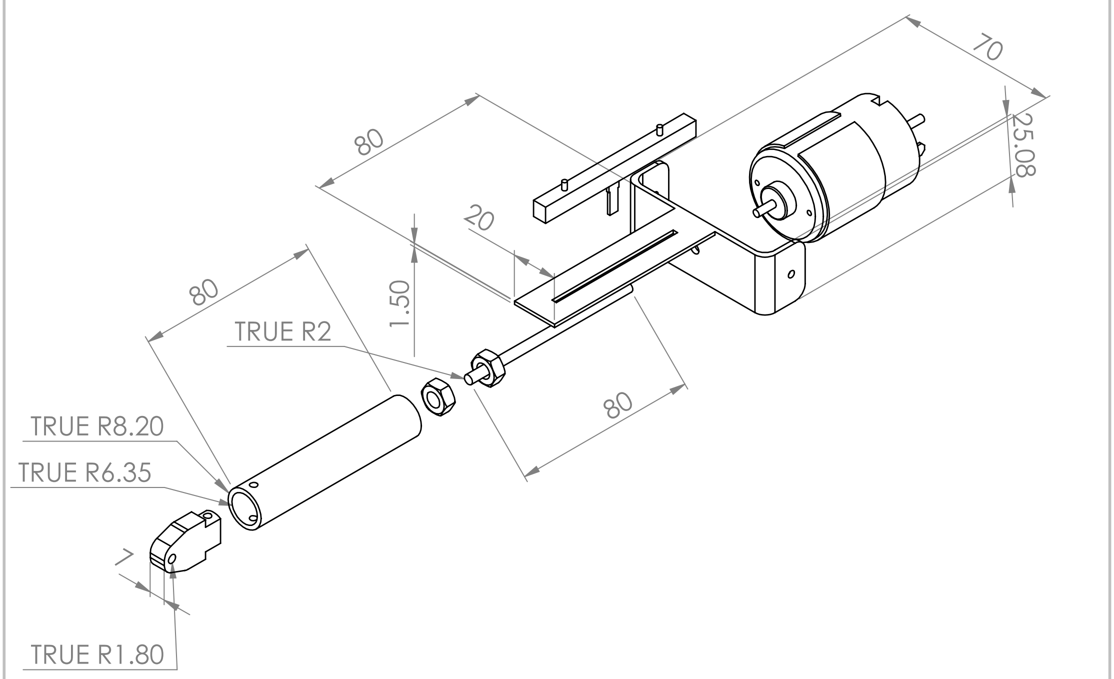
A4

WEIGHT:

SCALE:1:5

SHEET 1 OF 1

Figure 4.5 VESA Plate



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:

FINISH:

DEBUR AND
BREAK SHARP
EDGES

DO NOT SCALE DRAWING

REVISION

| | NAME | SIGNATURE | DATE |
|---|---|---|---|
| DRAWN | J Angelov | | |
| CHK'D | | | |
| APPV'D | | | |
| MFG | | | |
| Q.A | | | |

TITLE:

MATERIAL:

WEIGHT:

DWG NO. motor assem

A4

SCALE:1:5

SHEET 1 OF 1

Figure 4.6.1 Linear Actuator Assembly

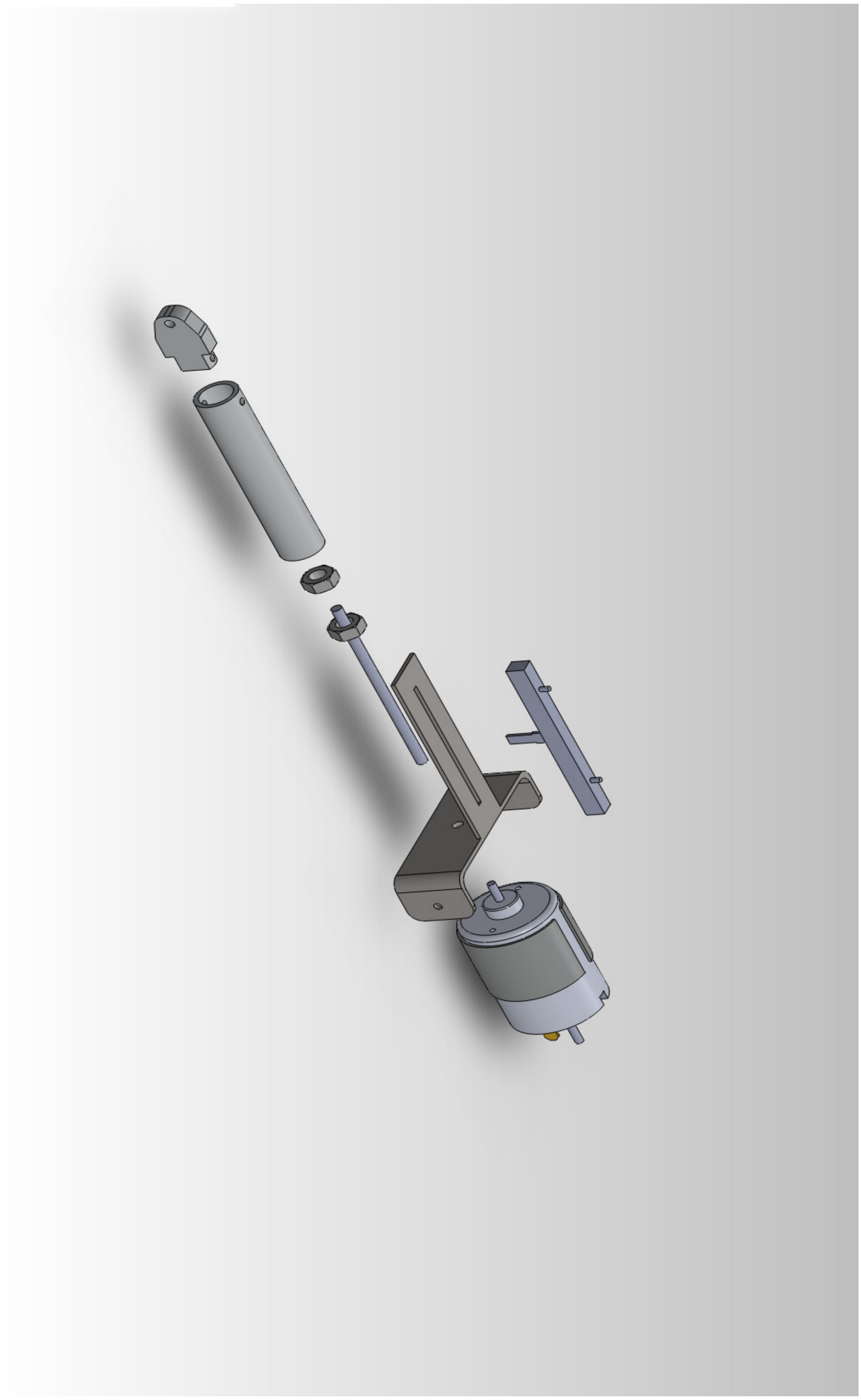


Figure 4.6.2 Linear Acuator Rendering

## 4.7 Actuators Operation

In this proof of concept model, there will be a total of two degrees of freedom. One degree of freedom will be the ability to augment the angle of the monitor to achieve an optimal viewing angle and the other will allow the monitor to pan laterally and to track users as they move around the surrounding area. Only two degrees of freedom were included in this proof of concept model as the implementation for three or more degrees of freedom would warrant a more advanced design with sufficient testing beyond the scope of this first stage.

### 4.7.1 Tilt Motion

The First degree of freedom allows the attached screen to tilt up and down to achieve the ideal viewing angle. A motor and worm drive will vary the length of a rod connected to the bottom of the mounting plate. A sliding potentiometer mounted on the rod will provide angle feedback to be processed by the Arduino code.

This worm drive in figure 4.6.2 will move a threaded rod 80 millimeters long in and out of the rodsocket. The end of this rodsocket will be attached from the bottom of the VESA mounting plate. This allows the DAS to achieve it's functional specification goal tilting up 40 degrees and down 20 degrees.

The linear actuator needed to have a 12V DC motor which favored torque over rotational speed and fit within the area allowed for by the Solidworks models in figure 4.6.2. These were the only design constraints imposed by the electrical and mechanical components because the most important fact here is the linear potentiometer feedback, the motor only has to fulfill these basic requirements. The motor we chose was a GM Steering Column Tilt Wheel Actuator Motor[23], its specification is unknown, due to it is a part of an automobile made in 2002. However, through testing we found the results very favorable to what we desired.

The resulting linear actuator manufactured by Deadalus Technologies has the following specification:

- 12V DC
- 1.2 Amp
- 60mm in 3 Seconds
- Built in limit switch
- 10kg push/pull force*
- 15kg max static/holding force*

*Tested with 15kg weight, it could possibly move much more weight, but for the purpose of the proof of concept model its sufficient and meets our functional requirement.

In addiction to the above mechanism, the linear actuator for the tilt mechanism will also include a feedback system. Figure 4.7.2 shows the circuit used to encode the position of the linear actuator. It is a voltage divider with sliding potentiometer and low-pass filter attached. Based on the potentiometer position, we will know the rod length and the resulting VESA plate angle. The VESA plate angle will be calculated based on physical measurements and data within the Arduino program.

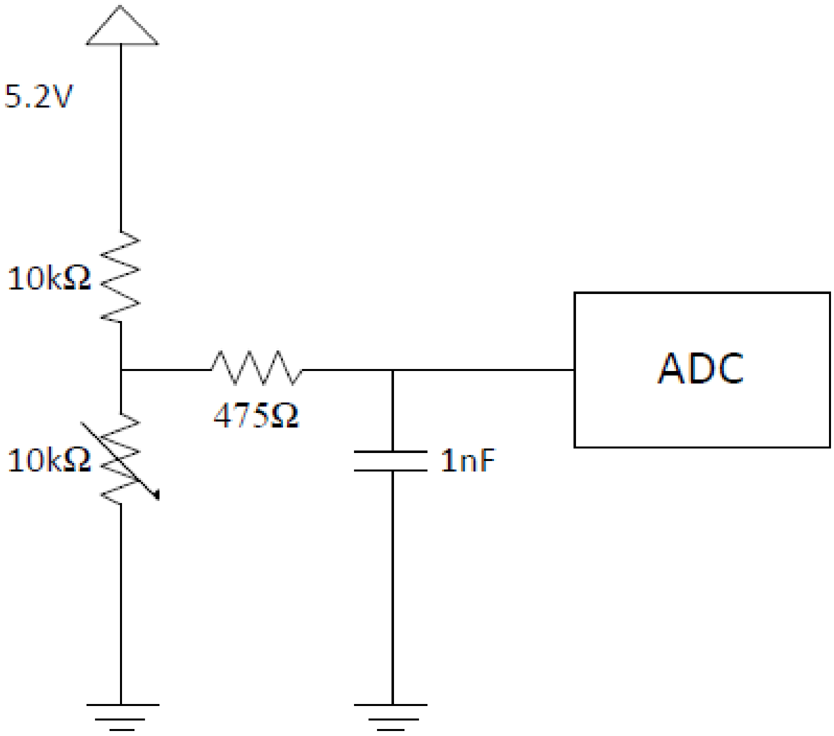


Figure 4.7.2 Tilt Control System

## 4.7.2 Pan Motion

The second degree of freedom allows a motor with attached gearbox and shaft to rotate the nacelle at the top of the stand to rotate in the horizontal plane. This setup allows us to achieve the 180 degrees of freedom listed as one of our functional specification requirements. The Arduino will also be responsible for keeping track of the position but information will be largely determined by the webcam. Absolute position will be important when moving to home or other preset positions if needed. However, moving to compensate for user movements will be done on a relative position basis.

Figure 4.7.4 is implemented twice as part of the encoding scheme and allow for a very high resolution encoder. The TCST5250 - Transmissive Optical Sensor with Phototransistor Output[24] module shown in Figure 4.7.3 uses a beam of light to keep track of motor revolutions. When a predetermined point is reached, the beam of light is interrupted and causes a software interrupt inside the Arduino program.

Figure 4.7.3 TCST5250 Optical Sensor [24]

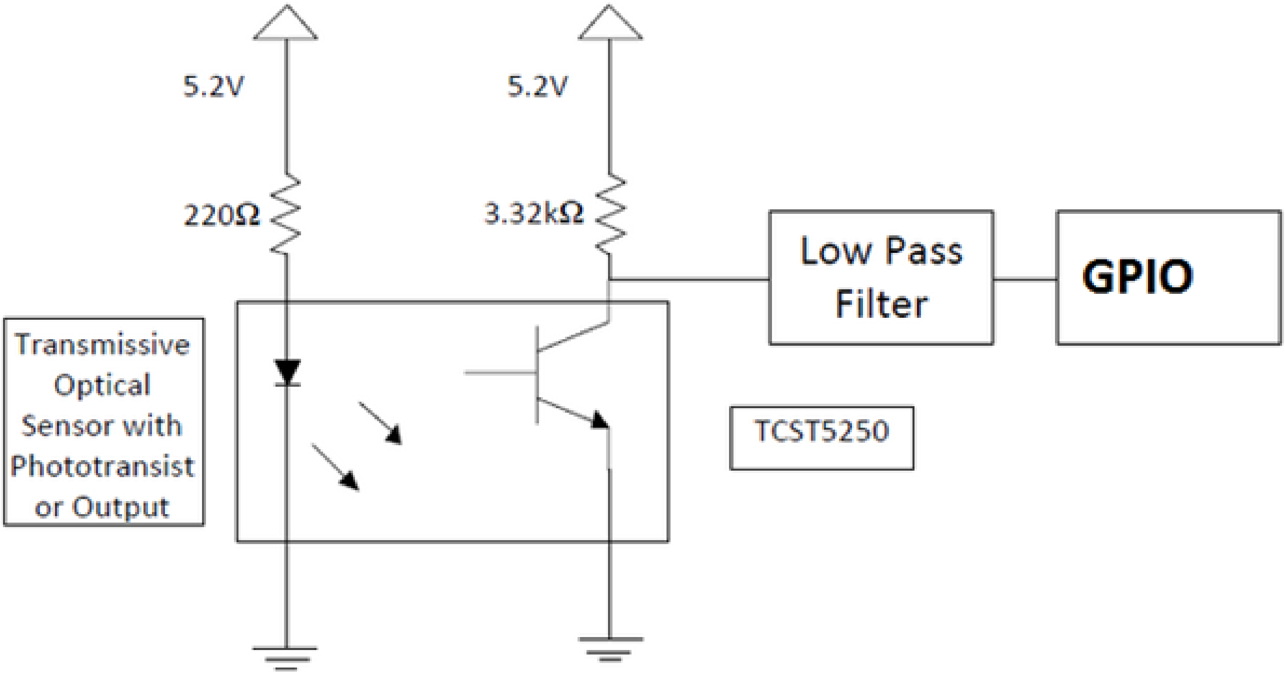


Figure 4.7.4 Pan Motor Control

One of these modules is implemented to encode nacelle position "zero" which is the far left side of the rotation. When the nacelle reaches this "zero" position, the beam will be interrupted and the Arduino will know that the left limit was reached.

The other implementation is to encode the motor's movements. The second optocoupled module is attached to the bottom of the shaft at the bottom of the motor. Once per revolution, the beam is broken and a software interrupt is triggered. As shown below the gearbox operates at 4 RPM, and the 48 revolutions per second speed was recorded with an oscilloscope attached to the output of the optical sensor circuit in figure 4.7.4. From this point, controlling the nacelle position correctly can be done with a high degree of accuracy from the arduino software.

Motor used for panning see figure 4.7.5, has the follow specifiations[25]:

- Operating Range: 3-12VDC
- Torque @ Max Efficiency: 205 oz-in. (12V)
- Torque @ Stall: 992 oz-in.@12VDC
- Gear train damage can occur if stalled (locked)
- .240" (6mm) Diameter Shaft
- No load current: 45mA
- No load speed: 4 RPM
- Gear ratio: 900:1
- Motor size: 1.30"Dia. x 1.015"L
- DC reversible motors
- High torque construction
- Oil bearing design for long service life

The reason we chose this motor is because of the high torque and low RPM specification in addition to its low price.

Figure 4.7.5 Motor for Panning [25]

# 5.0 Software

## 5.1 Requirements

- Sudden and drastic changes in face location will be ignored.
- When multiple faces are detected, the monitor will be centered until user defined cases are identified.
- The GUI will display the currently processing image.
- The GUI can calibrate target face position.
- The GUI has start and stop buttons and buttons to move monitor position.
- The GUI and image shown will be a fixed size.
- Software will work with select webcams initially.
- Algorithm will detect any face in view.

## 5.2 Software Overview

With the uses of the MATLAB Complier, we are able to make our products, made in MATLAB, available to use in all Windows operating system platforms. It allows us to package our application as an executable or a shared library. Executables and libraries created with the MATLAB Compiler use a run-time engine called the MATLAB Compiler Runtime (MCR). The MCR can be packaged with our application for distribution and can be deployed royalty free [26].

The GUI, or DAS Control Center, is designed to act as a communication portal to transmit information between our peripheral devices and Arduino as well as provide the user with different flexibilities such as the option to manual move our mechanical system to an optional timer that notifies the user when a break is needed.

## 5.3 User Interface

The graphic user interface shown in figure 5.3.1 is the primary user interface to our DAS system. It provides all the necessary feedback for our DAS hardware in addition to some calibration features. The GUI designed by the engineering team is subject to change. The GUI will also change because it is designed from our engineering team's perspective and therefore may not be suitable to potential end users. Our team at Daedalus Technologies will enforce the necessary changes as we approach our user acceptance test (UAT) and will change our GUI layout as we get more feedback from the potential end users.

The GUI is composed of 5 main sections: a window in the middle to display the current processing image; optional break timer; system control; webcam control and manual position control panels.

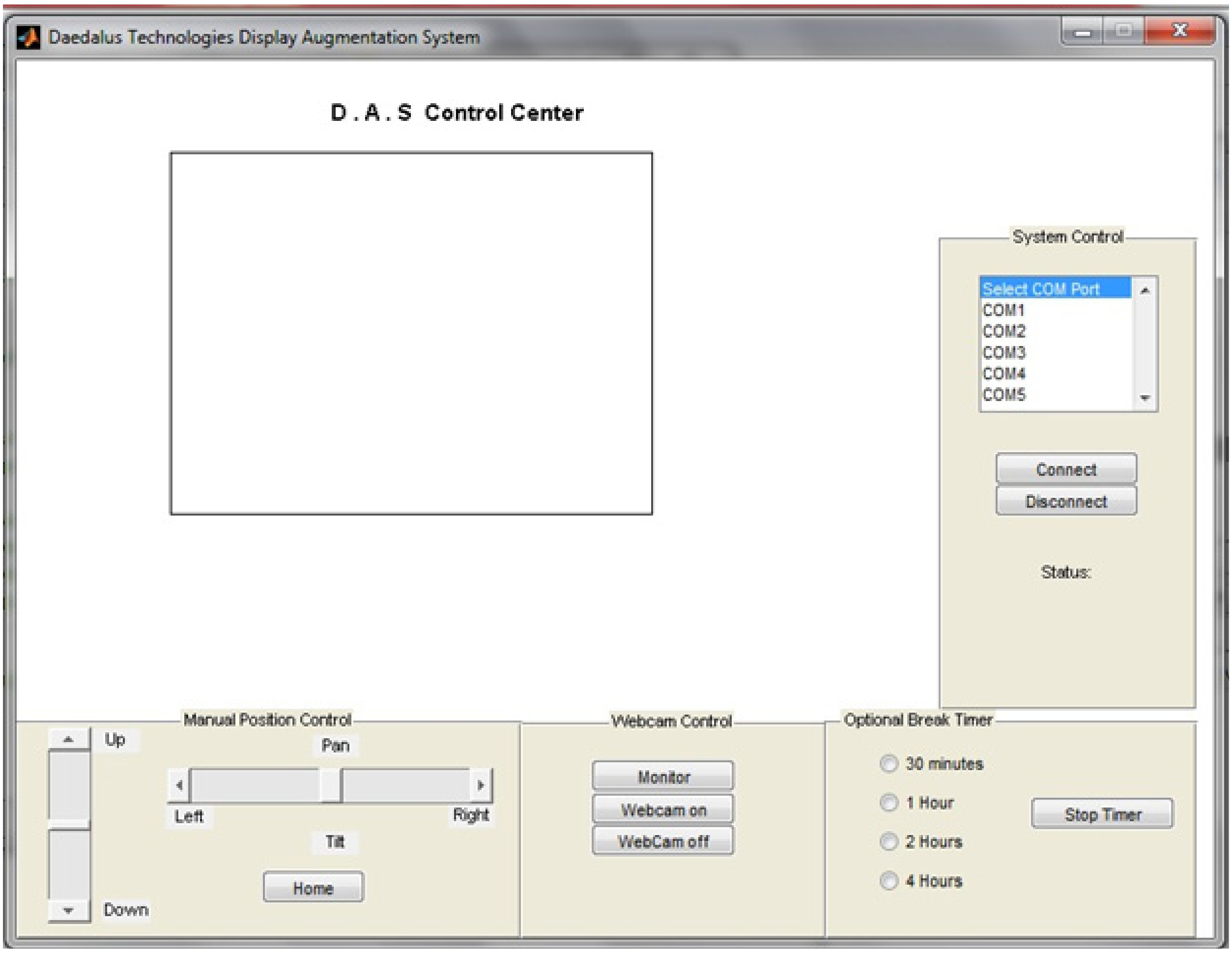


Figure 5.3.1 User Interface

The optional break timer panel allows the user to select a timer from the 4 possible choices, which act as a reminder that a break is need when spending significantly amount of time in front of a monitor. The manual position control panel enables the user to manually move the DAS unit. The Webcam control panel can turn the webcam on and off in addition to starting the image capture and face detection algorithm to enable face tracking from our system. Finally the system control panel allows the user to select a port to establish a valid communication with DAS, or disconnect to shut down the system. Figure 5.3.2 shows our the the general algorithm for our GUI.

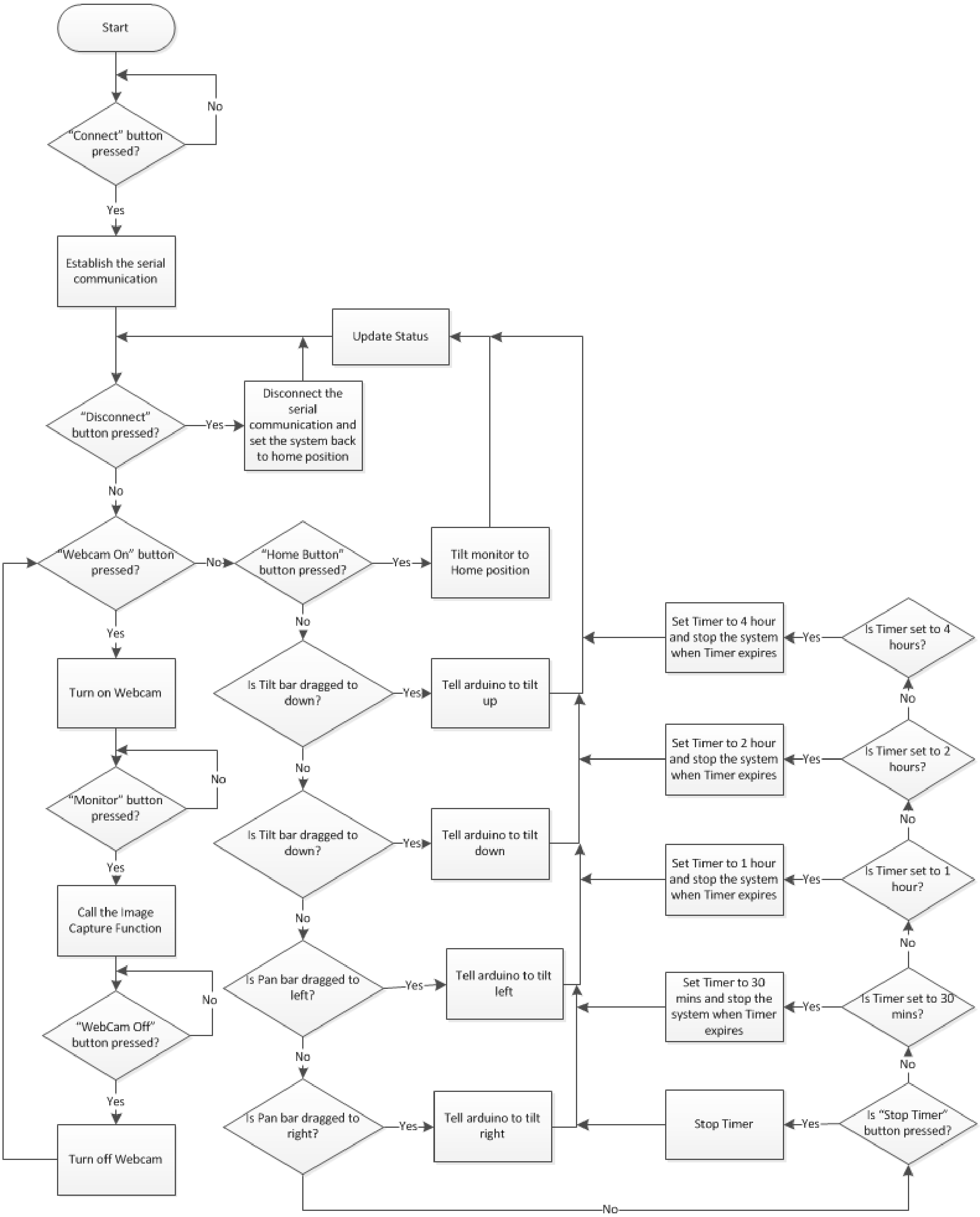


Figure 5.3.2 GUI Flowchart

## 5.4 Image Processing

The DAS image processing will consist of 3 parts: acquisition, face detection, and motion estimation. These three algorithms will be contained within a single timer interrupt triggered by the GUI when the “Monitor” button is pressed. It will occur on regular intervals of one second. This number is the result of weeks of testing and debugging. It is important to note that the face detection algorithm requires the webcam to be on at all times. Through our testing, we discovered that turning the webcam on and off was the largest delay to the face detection algorithm. As a result, the acquisition stage now assumes that the webcam is available for use.

Acquisition will occur via the attached webcam in the “YUY2_640x480” colour space - a MATLAB Image Processing Toolbox option determined upon the creation of the video capture function we are using to stream data to the GUI. These built-in MATLAB functions communicate via USB with the webcam and determine input resolution and color parameters. As shown in figure 5.4.1, the purpose of this timer interrupts is to obtain the newest picture for our face detection.

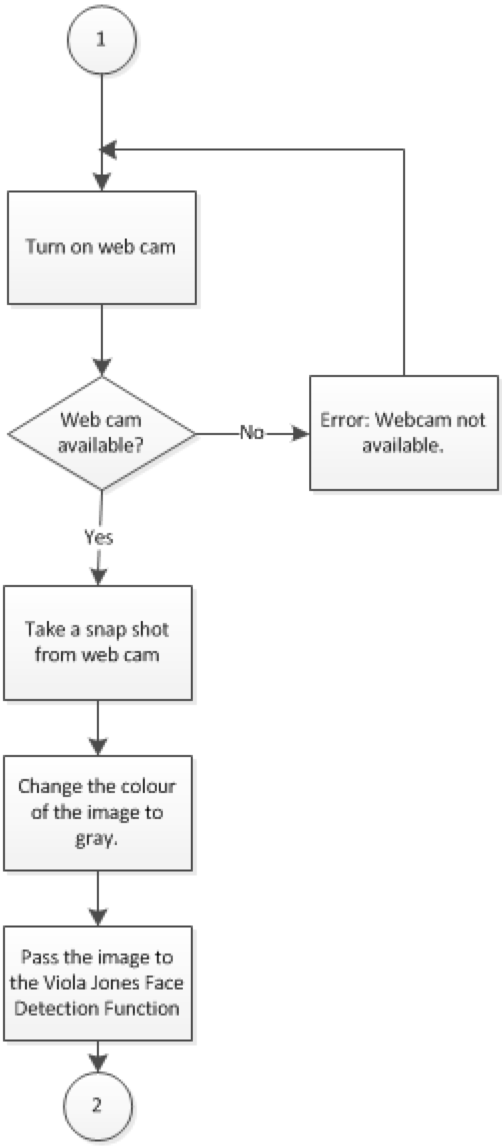


Figure 5.4.1 Face Detection Flowchart

Analysis of the image will occur via OpenCV 's Viola-Jones face detection method. The Viola Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones[27]. "The Open Source Computer Vision Libraries or OpenCV is a library of programming functions aimed at real-time computing vision with a variety of applications including face detection"[28]. For our project, we implemented this method with MATLAB by using OpenCV and Haar like features[29]. The Viola- Jones can be thought of as pixel intensity set evaluations where the sum of the luminance of the pixels in the white region of the feature is subtracted from the sum of the luminance in the remaining gray section. This difference in value is used as the feature value, and can be combined to form a weak hypothesis on regions of the image. Four of the Haar-like features are chosen for this implementation: one horizontal division, one vertical division, one that has two vertical divisions and the last one containing both the horizontal and vertical division see figure 5.4.2.

The Viola- Jones also used the integral image for intermediate image representation[30]. The sum of the pixels in a rectangular window can be computed easily using the intermediate representation. The integral image value at pixel location (x,y) in an image is defined as the sum of all pixels to the left and above the pixel (x,y). By expressing the same relationship as a pair of recurrences, it is possible to compute the integral image with just one pass over the image.

For our project, we first compiled the .cpp file provided by the OpenCV software, then we convert the[31] file into a MEX (MATLAB Executable[32])( file using the command "mex FaceDetect.cpp -I../Include/ ../lib/*.lib -outdir ../bin/" in MATLAB. This allows us to implement the Viola Jones algorithm with a mex implementation. After the file is compiled, the webcam captures images to be passed through the Viola-Jones algorithm using MATLAB. This function will first gray the image and detect the face area and draw a square around it. With this, we can use the coordinate to track the location of the user. The flowchart of our implementation is show in figure 5.4.3.

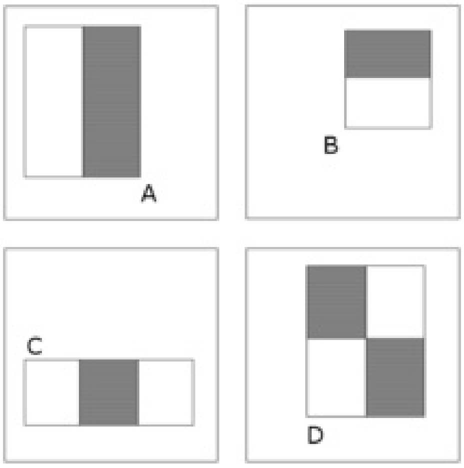


Figure 5.4.2 Haar like features [29]

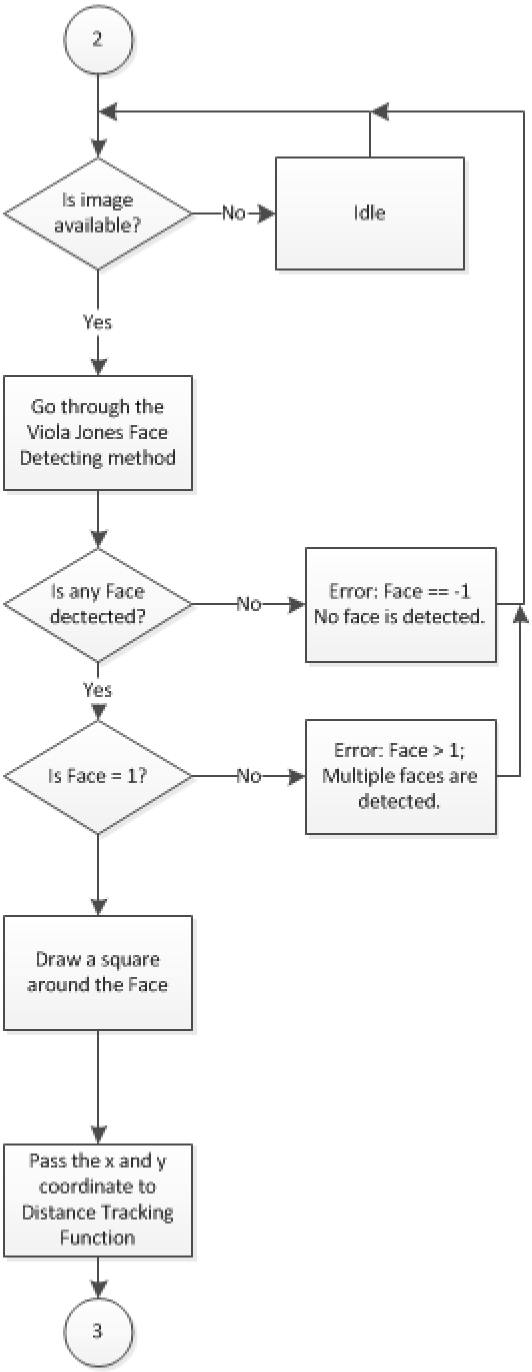


Figure 5.4.3 Face Detection Flowchart

Interpolating the data obtained from the OpenCV and Viola-Jones analysis is the final step before information is transmitted to the Arduino for further processing. Once a valid coordinate is confirmed, the information is passed through the motion estimation algorithm and transformed into a message appropriate for transport to the Arduino.

The motion estimating function will keep checking if the x-y coordinate data are available. Once the data is available, the function will estimate the position of the user by calculating the location of the user on the image, and send out a set of x-y location data to the Arduino, which will handle the physical motion adjusting. See flowchart below in figure 5.4.4.

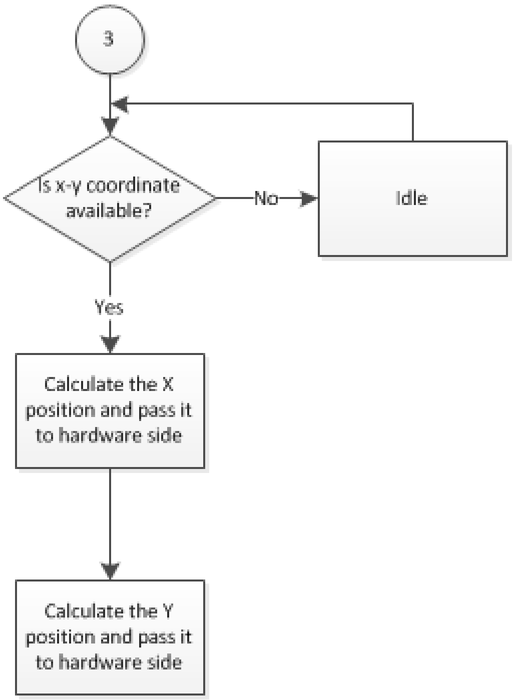


Figure 5.4.4 Motion Estimation Flowchart

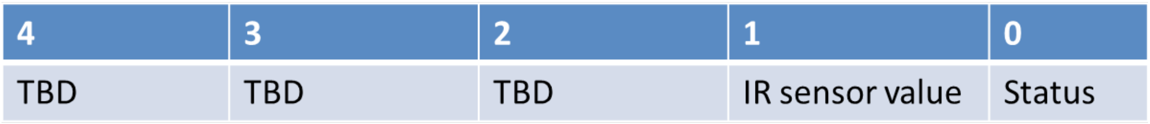
## 5.5 Arduino Software

The Arduino software will have several basic functions which will involve communicating with the GUI via serial, controlling the motors, and managing peripherals. The Arduino will be responsible for a minimal amount of work because for the purposes of this proof of concept model, the Arduino will act as a slave to the computer and manage hardware functions only.

The Arduino will communicate via the serial port to the computer running the GUI using an RS232 cable. This will allow serial data to be transmitted as USB data for the computer to understand. As the flow of information is bi-directional even in this proof of concept stage, a basic protocol has been established. An 8-byte message protocol has been established to send positional data from the face detection algorithm to the Arduino and vice versa. Figure 5.4.1 to 5.4.3 shows the basic structure of our protocol:

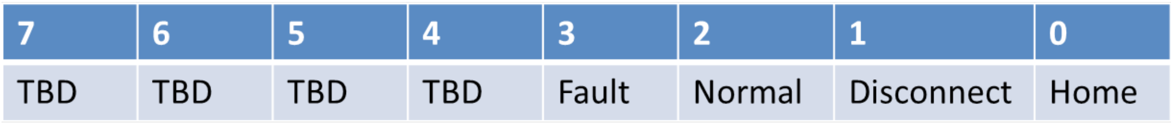
| 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| TBD | TBD | Y position (Tilt) | X position (Pan) | Status |

Figure 5.4.1 Data sent by GUI to Arduino



| 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| TBD | TBD | TBD | IR sensor value | Status |

Figure 5.4.2 Data sent by Arduino to GUI



| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TBD | TBD | TBD | TBD | Fault | Normal | Disconnect | Home |

Figure 5.4.3 Status Bits Definitions

The Arduino will control two 12V DC motors in the proof of concept model, the two motors both have different control methods; however controlling the two H-bridges will be done with the same method. In each case the two motors move in 2 directions (left or right and up or down). Because of the way the H-Bridges operate, the two PWM signals must not be on at the same time. The easiest method of control is to enable one switch pair at a time and keep the other one turned off. In addition to the PWM signals to the gates, the H-Bridge used includes an enable pin, which ensures the motors are not turned on unless the enable pin is set. Using the enable pin means we can safley alternate between the two PWM lines as shown in figure 5.4.4 below.

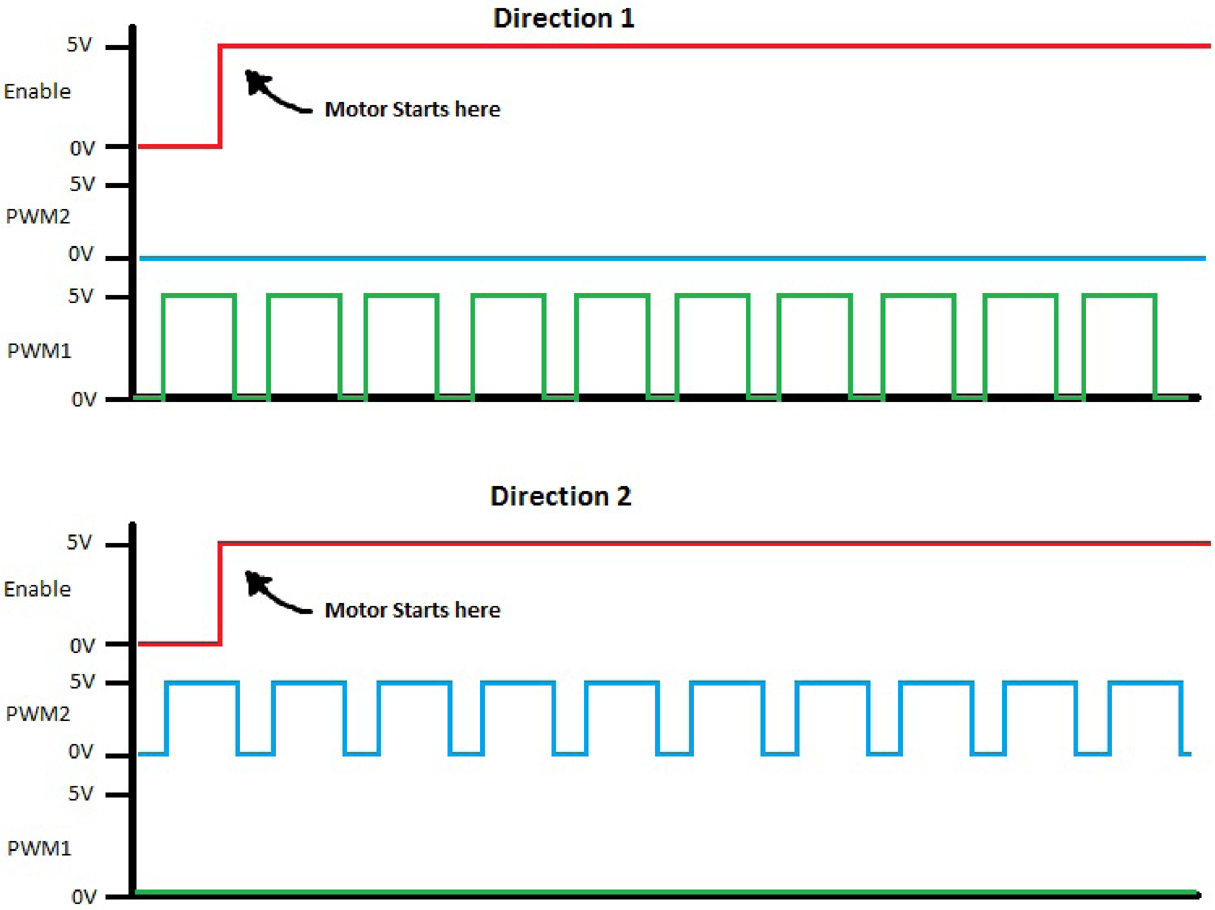


Figure 5.4.4 PWM Motor Control

The general state machine of the Arduino software is shown below in figure 5.4.5(note the Rxarray correspond to the array setup similar to figure 5.4.2). The structure of the software consists of a state machine complemented by 3 interrupts.

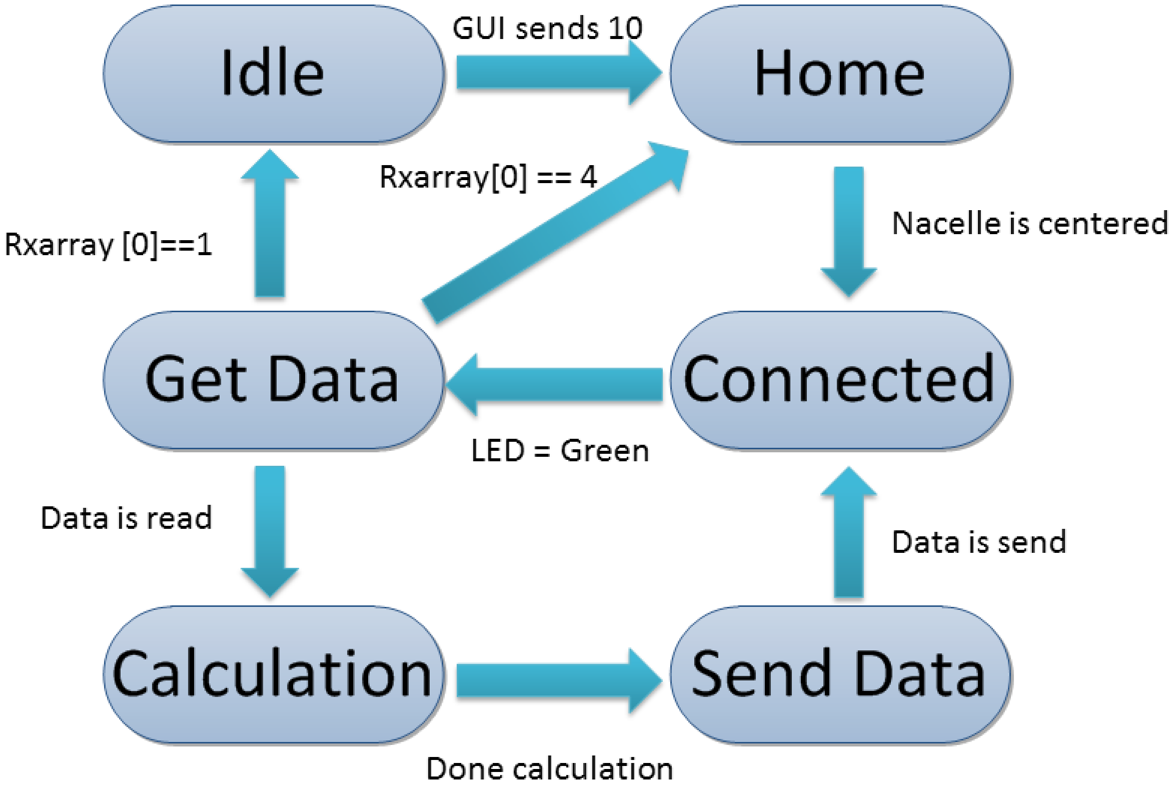


Figure 5.4.5 State Machine

The main interrupt is Timer 1 which is set to trigger every 1 miliseconds and is responsible for updating the status and direction of the two motors and reading the linear potentiometer voltage divider output voltage. The two other interrupts are external interrupts used to control the position of the pan motor. The first interrupt is used to home the pan motor. Once the home button is pressed, the pan motor begins to turn to the left. Once it reaches the predefined maximum left position, the photo micro-sensor (see section 4.7) triggers the external interrupt. The interrupt service routine then shuts off the pan motor, resets the revolution counter, changes direction and begins turning the motor to the right. Knowing the the RPM of the motor at 12V (approximately 48 revolutions per second as mentioned in section 4.7) and the RPM of the output of the gearbox is approximately 4 RPM, we can do some simple calculation, to find out how many turns are needed for a quarter of a revolution to bring the DAS from the very left to the center . First we take 60 seconds divide by 4 RPM and multiply by quarter of revolution giving us 3.75 seconds. Then take 3.75 seconds and mutilply by 48 revolution per second equals to 180 revolutions. The revolution count is monitored in the timer interrupt and once it is over 180 the pan motor is turned off.

The second external interrupt is triggered every time the motor makes a revolution and based on the direction of the motor the counter value is either incremented if panning right and decrementing if panning left. See figure 5.4.6 for futher details.

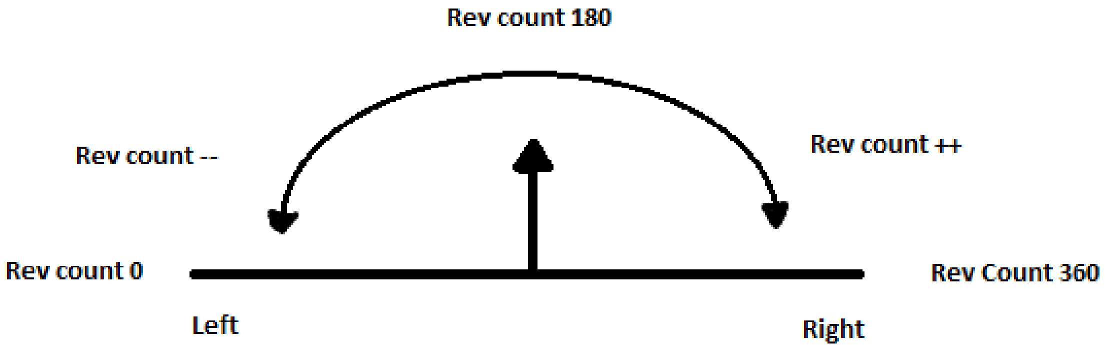


Figure 5.4.6 Position Control

# 6.0 Test Plan

Here at Daedalus technologies, our test plans are carefully devised to ensure the safety, functionality, and reliability of our product: Display Augmentation System (DAS).

## 6.1 GUI Unit Testing

In order to ensure that our GUI software that is error free, we will first test each individual modules of the GUI program, to ensure it performs its tasks correctly.

**Module: Face Detection Algorithm**
Test Steps: Load random frontal face images taken with webcam and from web
Expected Result: The system will detect all the faces and draw a square around it

Test Steps: Racial test, test subject including all racial color
Expected Result: Any racial user will be detected

Test Steps: Load side face, half face, covered face, tilted face
Expected Result: Face detection failed or invalid

**Module: Image Acquisition**
Test Steps: Click on webcam button
Expected Result: Webcam will turn on

Test Steps: Run Image Capturing Function
Expected Result: Image will be captured and send to face detecting function

**Module: Motion Estimation**
Test Steps: Load images with user standing in different position
Expected Result: Position coordinate will be send to Arduino

**Module: Connect Button**
Test Steps: Press the Connect Button on the GUI
Expected Result: Serial communication will established, and system is ready to be used

**Module: Setting Optional Timer**
Test Steps: Press the timer option and wait for timer to expire
Expected Result: When timer expired, the system will stop and a timer expired message will be displayed

**Module: Manual Position Control**

Test Steps: 1. Set up dummy protocol with Arduino
2. Manually adjust the system by dragging the Pan/Tilt bar to left, right, up, and down

Expected Result: The Arduino's build in LED should light up everytime it recieves a command from the manual position slider bars

**Module: Monitor On Button**

Test Steps: Click Monitor Button when Webcam is attached

Expected Result: The GUI will automatically track the position of the user a

**Module: Home Button**

Test Steps: 1. Set up dummy protocal with Arduino
2. Click Home Button Button

Expected Result: The Arduino's build in LED should light up indicating it has received command to reset to home position

## 6.1 Electrical Unit Tests

Electrical tests are executed to ensure that are we are not exceeding the recommanded voltage levels imposed by the various hardware manufacturers.

**Module: Check Buck Converter Output**

Test Steps: 1. Connect a multimeter probe with reference to ground to output of Buck converter
2. Apply 12V supply voltage to the DAS shield and observe the output voltage

Expected Result: The observed output should be just above 5V

**Module: Check Op-Amp Gain**

Test Steps: 1. Power up the DAS shield with a 12V supply
2. Apply a 2.7V input signal to the input of the op amp
3. With a mulitmeter referenced to ground, observe the output of the op-amp

Expected Result: The observed output cannot be greater than 5V as it may damage the Arduino

## 6.2 Requirement Testing

All requirements are based on Daedalus Technologies' functional specification documentation. For more details on a specific requirement please refer to DAS's functional specification document.

### Mechanical

Test Name: Mechanical requirement

Requirements: [R2-I] [R33-I] [R34-I] [R39-II] [R41-I] [R61-II]

Purpose: Confirm that the system supports all models of monitors in the market, and is able to hold it. User should not be seeing internal circuitry and PCB.

Procedure:

1. Try different models of monitors with the device.
2. Observe if the system can support it without falling.
3. Observe if there are any internal circuitry showing to the user.

Post Conditions: The system will support all VESA monitors in the market and be able to hold them. User will not see any external circuitry.

### Arduino and DAS Shield

Test Name: Check RS-232 serial port

Requirements: [R49-I]

Purpose: To verify the RS-232 level shifter outputs the correct voltage and signals.

Procedure:

1. Connect oscilloscope probes to both RX and TX lines referenced to ground.
2. Send serial data between Arduino and Matlab GUI and observe signals on scope.

Post Conditions: The voltage of the signals observed should be in the +/- 12V range.

Test Name: Check H-Bridges

Requirements: [R46-I] [R47-I]

Purpose: To verify H-Bridges are working

Procedure:

1. Ensure the bench power supply is set to 12V and current limited to 1A maximum.
2. Connect a 12 DC motor to the DAS shield M1 or M2 terminal blocks.
3. Apply a PWM signal to one side of the chosen H-Bridge ensuring the other is off.
4. Set the enable pin HIGH to enable the H-Bridge.
5. Reverse polarity of the motor by turning on the other side and turning off the previous pair.

Post Conditions: The motor should begin turning in one direction and when one pair of gates is turned on and should turn the other way when the other pair is turned on.

### GUI

Test Name: Graphical User Interface Intuitiveness
Requirements: [R4-II]
Purpose: Confirm that the GUI is user friendly with a well-constructed layout. Error messages are correctly shown and buttons are functional.

Procedure:
1. Ask user to run the GUI and observe if there are any difficulties.
2. Ask user to test out all the buttons.
3. Ask user to produce some error scenario.

Post Conditions: There should not be any difficulties for the user, all the buttons are functional and all error messages are displaying correctly.

### Image Capture

Test Name: Image Capture Functionality
Requirements: [R25-II] [R98-I] [R99-I] [R100-II]
Purpose: Confirm that face detection will detect any person using the system. An error message will show when the system failed to detect any user, andstay idle when multiple users is detected.

Procedure:
1. Ask user to start the system and observe if face detection is functional.
2. Ask user to move away from the webcam.
3. Ask multiple users to stand in front of the system.

Post Conditions: When image quality is too low, an error message will be displayed.

### Face Detection

Test Name: Face Detection Functionality
Requirements: [R14-I] [R77-I] [R78-II] [R79-II]
Purpose: Confirm that images are capturing and sending to the face detecting function.

Procedure:
1. Ask user to start the system and observe if image is capturing and sending to the face detecting function.
2. Ask user to use the system in a dark room.
3. Ask user to use the system in front of a bright background.

Post Conditions: Any face should be detected when there is only one user. Error messages will be displayed and system should stay idle when no user or multiple users are detected.

**System Basic Connectivity**

Test Name: Overall System Functionality

Requirements: [R15-I] [R17-I] [R27-I ] [R30-II] [R35-I] [R36-I][R76-I] [R102-II]

Purpose: Confirm that the overall system is functional. The system will track user and adjust the monitor to follow.

Procedure:
1. Ask user to start the system and the monitor will begin at centre location.
2. Ask user to move for a distance and observe if monitor is following.
3. Ask user to tile his head with one any displacement.

Post Conditions: System will start at centre location and always track the user. The system will always keep the flat panel monitor centered on the user's face and able to perform up and down tilt of +40 to -20 degrees and left and right rotation of +/- 90 degrees.

**User covertness**

Test Name: User Comfort Experience

Requirements: [R6-I] [R7-I] [R13-I] [R28-I] [R29-I] [R63-I] [R64-I] [R62-II]

Purpose: Confirm that the overall system is comfortable and safe for users. Users will not get injured while using the system and system is properly rooted when a VDU is mounted.

Procedure:
1. Ask user to use the system and observe for any possible hazards.
2. Check if the noise is noticeable while the system is being used.

Post Conditions: User will not be injured while using the system, system will not be able to be tipped on its side. The system's noise will not exceed 50dB while operating.

**Power**

Test Name: Power Test

Requirements: [R18-I] [R19-I] [R20-I][R42-I] [R43-I]

Purpose: Confirm the system will work with external power supply. User should see status LED light up once power switch as been flipped.

Procedure:
1. User connect the power supply to DAS, and flip the power switch to On.
2. Check if LED status light is red to indicate on.
3. User will notice the Display will tilt to the home position (flush with user).

Post Conditions: User will see LED in red when the system is powered on, Blue when the system is Homing and Green when system is connected and ready to use.

## 6.3 System Integration Test

In order to integrate the separate modules of our system efficiently and effectively, we must carefully perform unit tests as we proceed to overall system integration. Since the testing will involve having to integrate many modules, we will integrate them as necessary to perform the necessary tests. System integration will consist of two major components: software and hardware. Each component will consist of a series of smaller modules. We perform unit tests on each smaller modules before we do component integration to ensure overall system integration will be efficient and trouble free.

Tests performed on face detection algorithm will consist of testing many static images taken from the World Wide Web to webcam captures images. This way we can check the effectiveness and limits of our face detection algorithm, making sure we have a good overall understanding so we can approach the component integration with increased clarity. Image capture will be tested on how fast the program can capture images and the clarity of the images. The first smaller modular software integration will occur when image capture and face detection has passed its respective unit tests. This integration will consist of combining face detection and image capture into one program to test if the image captured from the webcam will get detected by the face detection algorithm. The second software modular integration will consist of integrating the GUI with face detection plus image capturing. The final software integration test consists of checking the robustness and functionality of the integrated software system, such as image capture and face detection error handling and constructing a GUI that is user friendly yet robust for our product DAS.

Unit tests performed on hardware modules will consist of writing smaller programs to test out individual section of the systems such as pan motor control, tilt motor control, IR sensor detection, serial port communication and LED state changing. Each system will also endure calibration testing to ensure the values we pass to the GUI is accurate. The smaller programs written for the unit tests will be modular in the sense that they can be integrated with ease into the final program, i.e the pan motor control test code is the same as in the final code). This will make it easier to write and integrate the final program.

The final integration test will consist of integrating the hardware and software component into one functional unit. Test cases are implemented based on the DAS's functional specification document to ensure we meet the necessary requirements.

# Conclusion

This document specifies the design solutions to the Daedalus Technologies DAS. Detailed explanations of the hardware and software components are included, as well as the reasoning for choosing the hardware components. A working prototype of our innovative device is expected to be available on December 8th. In addition to our design solutions, a test plan is also outlined to ensure that Daedalus Technologies product meets all the necessary functional specifications.

Design specifications are meant to be framework of what our engineers have accomplished with regards to functional specification. This is however, not a final document and it is subject to change as we continuously revise and update our design methodologies to meet the final prototype.

# References

[1] Infineon. (2011, November) Integrated High current Motor Drivers (NovalithIC™ / Trilith IC) [Online]. Available: http://www.infineon.com/cms/en/product/automotive-ics/motor-drivers-automotive/integrated-high-current-motor-drivers-novalithic-tm/trilith-ic/channel.html?channel=db3a304312bae05f0112bdd4f0640048&pt=db3a304412fcb7bb01131b678b261bd6

[2] Texas Instruments. ( 2010, July). LM358 [Online]. Available: http://www.ti.com/product/lm358

[3] Wikipedia, the free encyclopedia (2011, November 20). Operational Amplifier [Online]. Available: http://en.wikipedia.org/wiki/Operational_amplifier

[4] Texas Instruments. (2004, January) Interface RS-232 [Online]. Available: http://www.ti.com/product/max3232

[5] Texas Instruments. (2011, February ) DC/DC Converter [Online]. Available: http://www.ti.com/product/tps54232

[6] Arduino. (2011, November) Arduino.cc [Online] Available: http://www.arduino.cc/

[7] Atmel corporation. (2011, November). Atmega2560 [Online]. Available: http://www.atmel.com/dyn/products/product_card.asp?part_id=3632

[8] Microsoft. (2011, November). Microsoft LifeCam Cinema Technical Data Sheet [Online]. Available: http://download.microsoft.com/download/7/8/6/78695FFE-D786-46A1-BA24-964E97858CC0/TDS_LifeCamCinema.pdf

[9] Hanselman, Scott. (2010, February 17). DIY: Making a Very Wide Angle Webcam on the Cheap [Online]. Available:
http://www.hanselman.com/blog/DIYMakingAVeryWideAngleWebcamOnTheCheap.aspx

[10] Microsofe Hardware (2011) LifeCam Cinema [Online]. Available: http://www.microsoft.com/hardware/en-ca/p/lifecam-cinema

[11] Silicon Labs. (2011, November). USB Connectivity Bridges [Online]. Available:
http://www.silabs.com/products/interface/usbtouart/pages/default.aspx

[12] Mini-Box AC-DC power adapter with power cord. [Online]. Available:
http://www.mini-box.com/24w-12v-2A-AC-DC-Power-Adapter

[13] SHARP Corporation (Dec 2006) Distance Measuring Sensor Unit [Online]. Available: http://www.sharpsma.com/webfm_send/1487

[14] Healthy Computing. (2011, November). Monitor: Setup & Usage [Online]. Available: http://www.healthycomputing.com/office/setup/monitor/

[15] Thinkwiki. ThinkPad T60. [Online]. Available: http://www.thinkwiki.org/images/8/8e/ThinkPadT60.jpg

[16] MathWorks. (2011, November). Why is MATLAB only able to use 25% of my CPU? [Online]. Available: http://www.mathworks.com/support/solutions/en/data/1-WNXI8/index.html?solution=1-WNXI8

[17] Celeron + sempron price] Netlink Computer Inc. (2011, November). Processors (CPUs) [Online]. Available: http://ncix.com/products/?minorcatid=106&po=0&ps=2

[18] CPU World. (2011, November 14). AMD Sempron 145 - SDX145HBK13GM (SDX145HBGMBOX) [Online]. Available: http://www.cpu-world.com/CPUs/K10/AMD-Sempron%20145%20-%20SDX145HBK13GM%20%28SDX145HBGMBOX%29.html

[19] PassMark Software. (2011. November 19). CPU Benchmarks [Online]. Available: http://www.cpubenchmark.net/midlow_range_cpus.html

[20] Wikipedia, the free encyclopedia. (2011, August 21). Yonah (microprocessor) [Online]. Available: http://en.wikipedia.org/wiki/Yonah_%28microprocessor%29

[21] MathWorks. (2011, November). Choosing a Computer to Run MATLAB and Simulink Products [Online]. Available: http://www.mathworks.com/products/matlab/choosing_hardware.html#_Graphics_Processing_Unit

[22] Wikipedia, the free encyclopedia.( 2011, August). Flat Display Mounting Interface [Online]. Available: http://en.wikipedia.org/wiki/Flat_Display_Mounting_Interface

[23] ebay user “accuratediesel.“ (2011, November). GM Steering Column Tilt Wheel Actuator Motor NEW [Online] Available: http://www.ebay.com/itm/GM-Steering-Column-Tilt-Wheel-Actuator-Motor-NEW-/270730695532?item=270730695532&vxp=mtr

[24] Vishay Semiconductors. (2011, March 11). Transmissive Optical Sensor with Phototransistor Output [Online]. Available: ]http://www.vishay.com/docs/83787/tcst5250.pdf

# References

[25] Servo City. (2011, November). 4 RPM Gear Motor [Online]. Available: http://servocity.com/html/4_rpm_gear_motor.html

[26] MathWorks. (2011, November). Matlab Compiler [Online]. Available: http://www.mathworks.com/products/compiler/

[27] Wikipedia, the free encyclopedia. (2011, September). Viola-Jones object detection framework [Online]. Available: http://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework

[28] Wikipedia, the free encyclopedia. (2011, November). Open Source Computer Vision Library [Online]. Available: http://en.wikipedia.org/wiki/OpenCV

[29] Wikipedia, the free encyclopedia. (2011, September). Haar like features [Online]. Available: http://en.wikipedia.org/wiki/Haar-like_features

[30] Wikipedia, the free encyclopedia. (2011, August). Summed area table (integral image) [Online]. Available: http://en.wikipedia.org/wiki/Summed_area_table

[31] Krishna, Sreekar. (2008, May 21). Open CV Viola-Jones Face Detection in Matlab [Online]. Available http://www.mathworks.com/matlabcentral/fileexchange/19912-open-cv-viola-jones-face-detection-in-matlab

[32] Instructables user "leevonk." (2011, November). Making a Matlab MEX file [Online]. Available: http://www.instructables.com/id/Making-a-Matlab-MEX-file/