Servo Electronics Controller Board

by Nicholas Antoniades

Prepared for Justin Pead
Dept. of Electrical and Electronics Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Bachelor of Science degree in Mechatronics Engineering.

January 29, 2019

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature: 

N. Antoniades

Date: January 29, 2019

Abstract

The purpose of this study is design a servomotor controller board that has all the functionality of the more expensive servomotor devices but costs much less.

The controller board allows for more advanced control of the servomotor, with serial communication hardware, and temperature and current sensors controlled by an on-board microcontroller. A master controls multiple slaves, communicating a change in position or a requested for information. The on-board power supply has a wide current range allowing for compatibility with a large range of servomotors.

All the components except the buck converter were successfully tested. It was found that the pad spacing between the IC pins did not have solder mask. Resulting in the designed PCB being unsuccessful as the pins would short when power and ground were connected. This issue can be avoided by ensuring solder mask is applied between pads.

The designed system cost substantially less than more common servomotor controller boards. When tested all modules of the system worked but a second PCB design needs to be created in order to build a functioning prototype.

Contents

List of Figures

List of Tables

1 | Introduction

1.1 Background to the study

With advancements in technology, the demand for a robotic solution when problem solving is increasing fast and is already in high demand. Unfortunately the current hardware available in the market for accurate motor control is expensive, resulting in the only consumers who have access to it, being the large companies and other organisations with big budgets. This limits those individuals who want to design and prototype robotic applications at home or at university for research projects.

These motor controller boards allow for more advanced control of the motor, not just direction control, they also allow for velocity, current and torque control with the aid of additional sensors. This helps give better performance and a higher accuracy of motor control. , allowing for more advanced robotic application.

There is a high demand for robotic software and hardware but until recent years it has been unaffordable for the general public. The aim of this study is to design and create an affordable servomotor controller board that will have all the functionality alike that of the more expensive controllers, but be much cheaper.

1.2 Objectives of this study

This project illustrates the design and production of a servomotor controller board. On-board sensors and microcontroller will be used to monitor temperature and current drawn as well as allow for motor control. The controller will be able designed to be cable of connecting to a wide range of servomotors as well as allowing multiple boards to be daisy chained together and be capable of serial communication. Advanced servomotor controllers are expensive. The designed board needs to have the abilities of the more expensive controllers, but be much cheaper.

1.3 Scope and Limitations

The design of the controller board was performed from an academic view-point. The main goal being to create a proof of concept, showing that it is possible for cheaper servomotor controllers to be designed with the same functionality of more expensive controller boards.

1.4 Plan of development

This report is organised in the following order. First the objectives of the study and the problem to be solved are outlined in the Introduction. Second, a review of available literature on the topic at hand. The required system is then designed and components are chosen. The components are tested and the method and results are discussed. The PCB is the designed and the component placement and testing is discussed. The methods for the implementation of the code is discussed and then finally comclusion are drawn and recommendations were made.

2 | Literature Review

2.1 Servomotors

Typically servomotors are rotary or linear actuators that are controlled using a microcontroller and an encoder. This allows for the precise angular position, velocity and/or acceleration control of the output shaft. Servomotors are commonly used in motoring applications such as 3D printing and CNC machines, or in robotic arms and assemblies which use forward and reverse kinematics in order control specific motor positions allowing for complex tasks to be completed accurately(1).

Figure 2.1: Servo controlled arm (2)

Figure 2.2: CNC machine (3)

A servomotors usually has 3 wires, Control, Power and Ground. The output shafts angle is controlled through adjusting the duty cycle of a Pulse Width Modulated (PWM) signal being supplied to its control wire. The on board microchip receives this signal and in turn outputs a voltage to a DC motor based on the duty cycle of the signal. This voltage rotates the motors shaft, which in turn rotates the output shaft to a specific angle, usually between 0°and 180°but there are also servos that have it outputs a voltage 360°of rotation. This position is tracked and controlled using an encoder such as a potentiometer in order to determine current position, but there are many other types of encoders which can give more accurate position, direction of rotation and velocity control. A typical servomotor and its components can be see in Figure 2.3 and Table 2.1.

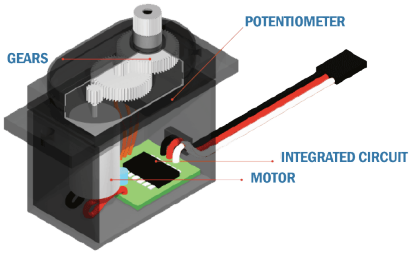


Figure 2.3: Servomotor components(4)

A servomotors components
- A Motor(AC/DC)
- Potentiometer as the encoder
- A microcontroller for control
- A Gear train

Table 2.1: Servomotor components

2.2 DC motors

A current carrying conductor within a magnetic field will experience a force. The direction of the force will be perpendicular to the plane containing the current carrying conductor and it's magnetic field which are also perpendicular to each other. This is the principle used in a simple brushed DC motor with permanent magnets used for poles and a slip ring commutator to alternate the flow of the current on the conductor. Power to the coils is supplied through fixed conductive brushes that make contact with the rotating commutator. The force rotates the shaft at a rate related to the voltage applied to the motor(5).

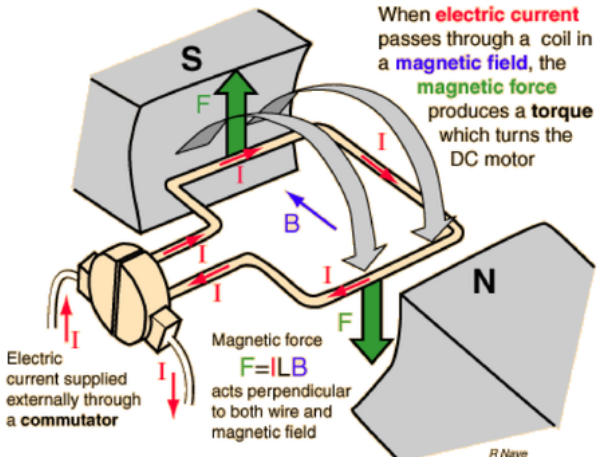


Figure 2.4: Current carrying conductor in a magnetic field(6)

With brushless DC motors the rotor is a permanent magnet, the rotor is then rotated by rotating the surrounding magnetic field. Brushless motors have a higher power to weight ratio and high speed control. They do not miss steps and are much more powerful and efficient, but cost a lot more than brushed DC motors (7)

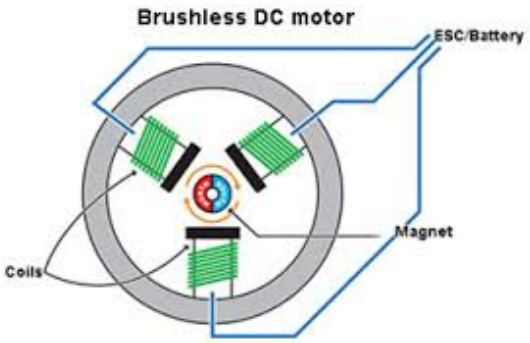


Figure 2.5: Brushless DC motor (7)

2.3 Microcontrollers

A microcontroller acts as the brain of the system, using different peripherals to interact with different devices and being able to make decisions in order to perform tasks. The hardware integrated with microcontroller and the quality of the software loaded determines how well it functions and tasks it can perform.

2.3.1 Flash Memory

Flash memory is an electronic non-volatile computer storage medium that can be electronically erased and reprogrammed. Flash memory contains its information even when the device is not powered. What is included in this is information required to start up the micro and be active during runtime.

The Flash memory on the STM32F051C6 has been organised into 32-bit wide memory cells that can be used for storing both code and data constants.

This information is divided into two parts :

1. **System memory** which is used to boot the device on start-up in System memory boot mode. This area is reserved for use by the manufacturers of the microcontroller, and contains the boot loader which is used to reprogram the Flash memory through the selected communication interface. It is programmed by ST when the device is manufactured, and protected against various write/erase operations.
2. **Optional bytes** that include those used when re-programming the device for specific applications.

2.3.2 DAC

Digital-to-analog converters are used to create an analog signal based on a digital value. The most basic type of electronic DAC uses pulse-width modulation and a low pass filter. A chosen stable current or voltage is achieved by adjusting the frequency of the PWM signal. The duty cycle of the PWM based on the value of digital input. This technique is often used for electric motor speed controllers and other applications (8).

2.3.3 ADC

An analogue-to-digital converter(ADC) allows for a voltage source to be sampled and read in to the microcontroller as digital code. The resolution of the ADC determines the number of bits used to specify a certain voltage range e.g. for an ADC with 2^8 bit having a resolution 255. If the voltage range to be sample is between 0V and 3.3V. Gives a resolution of 12.9mV per bit.

Successive Approximation Converters

Successive approximation ADCs are a type of analog-to-digital converter that reads in a continuous analog signal and outputs a digital binary value based on the input signal. This type of converter uses a binary search through all the possible quantization levels. The SAR outputs a binary value of the approximated ADC value, which is then converted back into an analog signal and compared to the input wave form using a comparator. If the signals are not the same the SAR increments its approximation until converging upon the final digital value to be output. Once this occurs an End Of Conversion (EOC) signal is sent to the microcontroller.

The STM32F051C6 uses a Successive Approximation ADC like the one seen below in Figure 2.6.

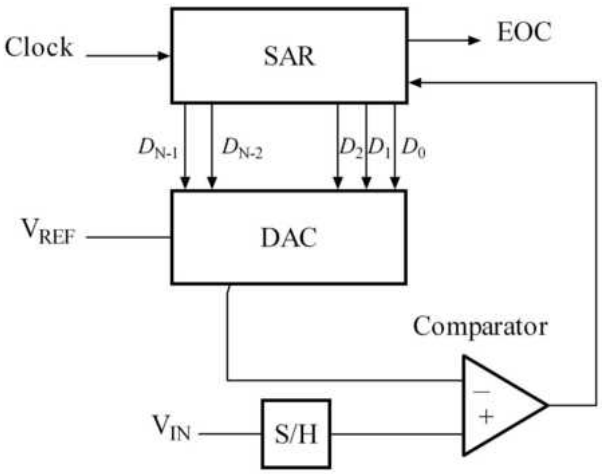


Figure 2.6: Successive Approximation Converters (9)

2.4 Sensors

Multiple sensors will be incorporated into the final design of the controller board. Sensors are important in the functioning and accurate control of the system as well as allowing for protection against overheating and excess current draws.

2.4.1 Current sensor

A current sensor is a device that detects electric current and generates a signal in proportion to that of the current, the output of the sensor can be analogue or digital (10).

Typical current sensors that are available:

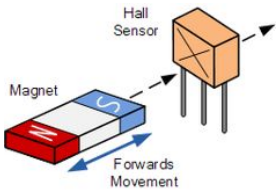
<table>
<tr>
<td> 

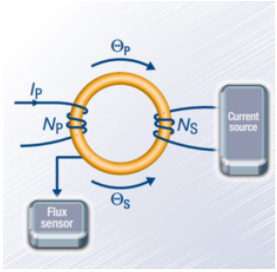
Figure 2.7: Hall sensor</td>
<td>Hall effect sensors, seen in Figure 2.7. can be used to measure the magnetic field made created by the current flowing through conductor and from that determine the actual current.</td>
</tr>
<tr>
<td>Shunt resistors, seen in Figure 2.8. Use Ohms law in order to calculate the current flowing. $V = I \times R$, with a known applied voltage and chosen resistor the current can be calculated.</td>
<td> 
Figure 2.8: Shunt resistor</td>
</tr>
<tr>
<td> 

Figure 2.9: Saturable inductor</td>
<td>Saturable inductor sensors, seen in Figure 2.9. Work in a similar manner to that of Hall effect sensors in that they measure the change in the magnetic field in order to determine the current flowing through a conductor.</td>
</tr>
</table>

Table 2.2: Types of current sensor

2.4.2 Encoders

Encoders are electromechanical devices that output an electrical signal proportional to that of the position of the input shaft.

Typical encoders that are available:

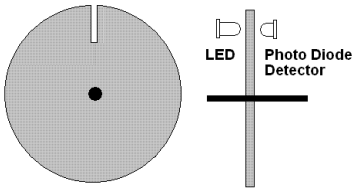
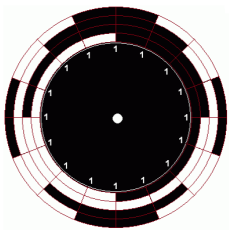
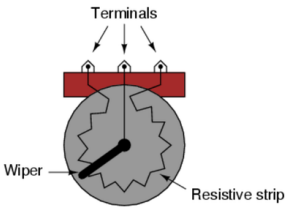
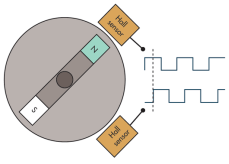
Optical encoders, which use pulses of light emitted by infrared diodes shining through slots. This can be seen in Figure 2.10 As the number of slots increases so does the encoders resolution(11).	 Figure 2.10: Optical encoder
 Figure 2.11: Binary encoder	**Binary encoders**, which uses a template that has segments based on a binary value to determine current position. This can be seen in Figure 2.11. multiple motor positions can be determined, the number of positions has a 2^N relationship with the number of segments in the encoder.
Potentiometers, are a mechanical encoder. This can be seen in Figure 2.12. Where the different resistance values and there associated voltage represent different position values of the wiper on the resistive strip (12).	 Figure 2.12: Potentiometer
 Figure 2.13: Hall effect encoder	**Hall effect sensors**, are transducers that have an output voltage that varies in response to a magnetic field. Seen in Figure 2.13. This us then used to determine position and velocity(13).

Table 2.3: Types of encoders

2.4.3 Temperature sensors

Temperature sensors are an important way to monitor and keep track of the state of the device during different operating conditions.

- Analogue temperature sensors output a voltage that depends on the temperature of the device and this can be read using the ADC of a microcontroller and then the devices temperature can be calculated.
- Digital temperature sensors will communicate the temperature at the sensor using I2C or another form of communication.

Typical temperature sensors that are available:

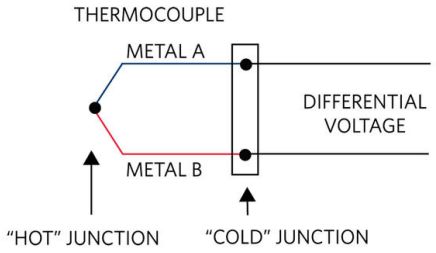
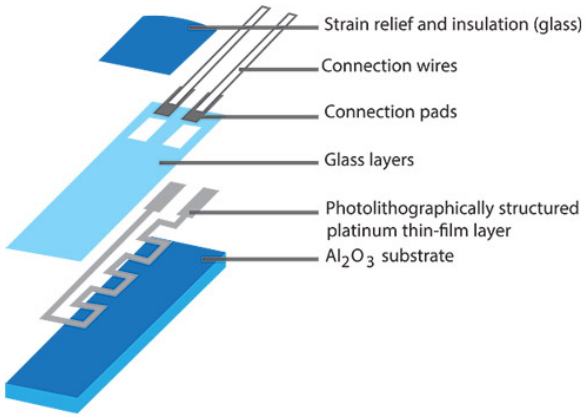
Thermocouples, use two different wires made of different metals connected at two points. The varying voltage between two points, reflecting proportional changes in temperature (14).	 Figure 2.14: Thermocouple setup
 Figure 2.15: RTD	**Resistance Temperature Detectors (RTDs)**, measure temperature changes by correlating the resistance of the device element and temperature. An RTD consists of a film for greater accuracy, a wire wrapped ceramic or a glass core (14).

Table 2.4: Types of temperature sensors

2.5 Communication

Traditionally when connecting multiple servomotors to a microcontroller, each servo must be individually connected to the micro, and each servo must individually be connected to the power supply. This can result in long masses of wires when multiple servomotors are used in one project, seen in Figure 2.16. This also limits the number of servomotors that can be controlled at a time to the number of GPIO pins available on the microcontroller.

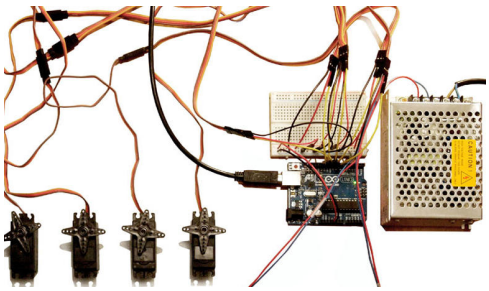


Figure 2.16: Multiple servomotors connected to one microcontroller(17)

The controller board that will be designed to enable that the servo motors to be connected in series and be able to communicate using the RS485 differential serial communication standard. This will allow for communication over a longer distances with a high noise rejection for increased accuracy. This enables the master to effectively communicate to each individual slave a certain position value and request information such as current drawn or temperature at the motor from it in return. The Master and Slave servo inter-connection can be seen below in Figure 2.17

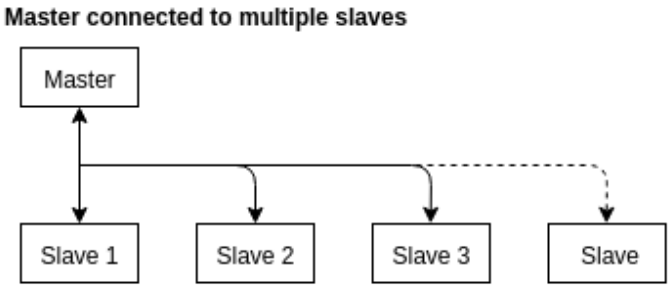


Figure 2.17: Servo inter-connection

The RS485 standard is a hardware protocol that will require that the microcontroller interacts with a transceiver which will be able to facilitate the communication.

2.5.1 Serial communication

Serial communication is the process of sending data one bit at a time, from one device to another sequentially over a communication channel or a computer bus (38). Communication is accomplished using 3 lines. (1) Ground (2) Transmit and (3)Receive.

Serial communication can be synchronous or asynchronous. What this means is that the communicating devices either run off their own clocks and before hand decide on a common speed at which to communicate, this is asynchronous communication. Or they will run off a common clock which will in turn give them a common communication speed, this is synchronous communication. Asynchronous communication allows for information to be transmitted and received simultaneously. This is know as Full Duplex communication. Where as synchronous communication is Half Duplex communication and only one device can transmit information at a time. Serial communication can be used over much greater distances in comparison to other forms of communication, such as parallel communication, making it ideal in applications when communicating between devices that are far apart .

Serial communication is made of four parts :

1. **Baud rate** is the speed at which the communication occurs and it indicates the bits the number of bits transferred per second. Baud rate can also be read as the frequency of the clock needed for sampling, for example 3000 baud would mean the clock is running at 3kHz.
2. **Data bits** are bits of information that needs to be transmitted.
3. **Stop bits** are used to indicate the end of the communication for a single packet.
4. **Parity** is used for error checking when using serial communication. There are four types of parity. Even, odd, marked, and spaced. When using even or odd parity the last bit after the data will be set to 0 or 1. Depending if it is even or odd parity. Then when a message is received its parity can be checked to check its accuracy. This can be used to detect noise and whether the communicating devices clocks are out of sync.

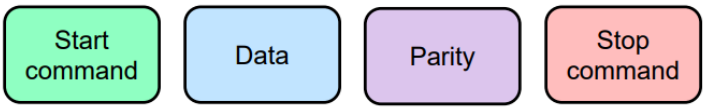


Figure 2.18: Basic structure of a serial message

2.5.2 Recommended Standards

When communicating using serial communication a Recommended Standard (RS) is typically used for easy connection between different devices. These standards use the Universal Asynchronous Receiver Transmitter (UART) hardware interface to facilitate the communication. Usually a UART is part of an integrated circuit (IC) and the signal level shifting needs to be handled by a driver circuit external of the UART. When communicating using a RS, a hardware interface is required in order to do the voltage level shifting. These components are known as level shifting transceivers.

A simplified diagram of the implementation of asynchronous serial communication between two devices can be seen below in Figure 2.19.

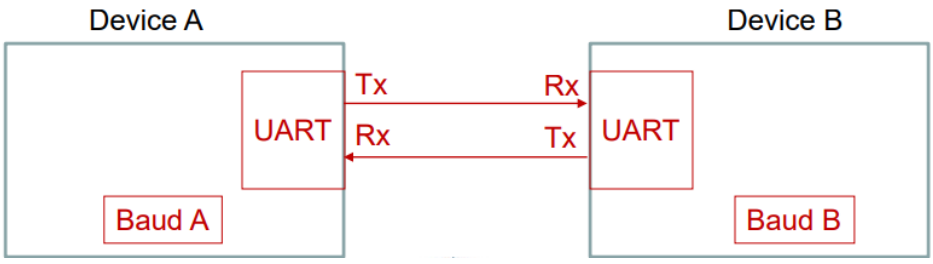


Figure 2.19: UART communication between two devices

Common Standards :

1. **RS232** Designed to work over a distance of up to 15 meters. The maximum baud rate is 19.2k. A logic level of -3v and less represents a 1 and 3v and more represents 0 (18).

2. **RS422** Introduced to enable higher data rates than what was possible with RS-232 using differential transmitters and receivers. Works over a distance of up to 15m (19).

3. **RS485** RS-485 uses differential serial communication. Differential communication transmits information using two complementary signals over a twisted pair of conductors. Any external electromagnetic interference will then effect both conductors in the same way. Since the receiving circuit detects only the difference between the pair of conductors, the signal can be communicated more clearly and over greater distances (36). allowing for there to be up to 32 devices connected to a single serial communication line (20).

2.6 Buck Converter

According to Joule's first law. The electrical current passing through a conductor will generate heat with power that is proportional to resistance of the conducter and square of the current, $P = I^2 \times R$. Therefore, in order to minimise power loss, the current drawn is reduced. In order to still transmit the same amount of power with the lower current, the voltage is increased as $P = V \times I$ (21).

That is, in order keep power the same, while transmitting lower current, the voltage must be increased. Thus in order to be able to efficiently supply power to multiple servomotor controller boards, all the boards will be connected to a supply voltage in the range of 5V to 40V. Each board will have its own on-board buck converter to step down the voltage and step up the current in order to drive the servomotor and other components.

The power supply required for the designed board must be able to power the servo motor as well as the other on board components. This means that the supply must be able to handle the total current requirement of the board while still supplying the 5V required by the servo motor. Any other voltage requirements in the circuit will require a voltage regulator for further regulation.

When considering the buck converter, the power demands of the servomotor was the greatest. The current drawn by the rest of the components was almost negligible in comparison. Table 2.5 was used to create an idea of a typical servomotors power consumption characteristics. This was used to help determine the current requirements of the designed buck converter.

Name	**Weight(g)**	**Stall Torque($\frac{kg}{cm}$)**	**Stall Current(A)**
MG90S	13.4	1.8	0.2
MG90	14	2	0.48
SG-5010	39	5.5	0.6
MG996R	55	9.4	2.5
MG946R	55	10.5	1.2
MG958	65	18	1.6

Table 2.5: Current requirements of various servomotors

It was determined that the buck converter needed to be able supply 5V while outputting up to 4A worth of current in order to be able to run a range of different servo motors at stall torque as well as sufficiently power the rest of the circuit.

2.6.1 Design Procedure

Buck converters are a type of switch mode power supply that step down a higher voltage to a lower voltage. This is achieved through high frequency switching of the circuit with the source voltage with the help of a few added necessary components.

The kind of switching required by the buck converter is typically achieved with a high speed MOSFET. A microcontroller supplies a PWM signal of the right frequency to the gate of the MOSFET, the duty cycle of the PWM signal determining the length of time of which the switch will stay open and closed. A basic buck converter circuit can be seen in Figure 2.20.

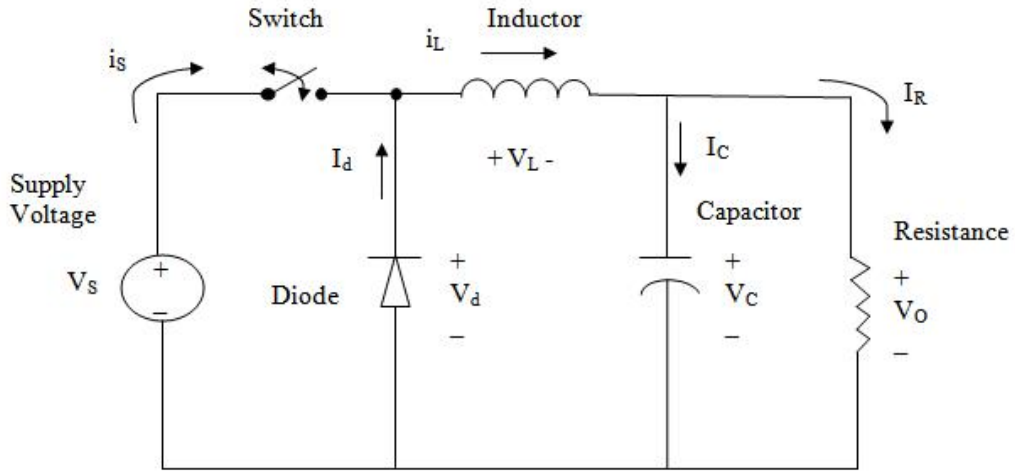


Figure 2.20: A basic buck converter circuit (33)

In order to sufficiently meet design requirements, the design procedure requires that the following parameters need to be known or calculated :

Parameters chosen	**Parameters needing to be calculated**
$V_{IN(max)}$ = maximum input voltage	D =duty cycle
V_{OUT} = desired output voltage	$\triangle_L = Inductor Ripple Current$
$I_{LIM(min)}$ = minimum current to the switch	I_{MAXOUT} = Maximum IC output current
f_s = minimum switching frequency of IC	L = calculated inductor value
μ = efficiency of the converter	$I_{SW(max)}$ = Maximum switch current

Table 2.6: Buck converter parameters

2.6.2 Calculations

Once the desired buck converter IC, and input output parameters have been chosen. The following calculations can be done in order to solve for the components required to build the buck converter circuit. The calculations used for the design process of the buck are based on those found in the application report "Basic Calculation of a Buck Converter's Power Stage", made by Texas Instruments (22).

Maximum Switch Current

The first step to calculating the switch current is to determine the duty cycle, D, for the maximum input voltage:

$$\mathbf{D} = \frac{\mathbf{V_{OUT}}}{\mathbf{V_{IN(max)}} \times \mu}$$

The inductor ripple current is determined by chosen an inductor specified in the buck converters data sheet:

$$\triangle \mathbf{I_L} = \frac{(\mathbf{V_{IN(max)}} - \mathbf{V_{OUT}} \times \mathbf{D})}{\mathbf{f_s} \times \mathbf{L}}$$

It needs to be determined if the chosen buck converter IC can deliver the required maximum output current:

$$\mathbf{I_{MAXOUT}} = \mathbf{I_{LIM(min)}} - \frac{\triangle \mathbf{I_L}}{\mathbf{2}}$$

If the maximum output current is above the maximum output current required for the device, the maximum switch current of the system must be calculated in order to determine the peak current that the inductor, switch and external diode must be able to withstand:

$$\mathbf{I_{SW(max)}} = \frac{\triangle \mathbf{I_L}}{\mathbf{2}} + \mathbf{I_{OUT(MAX)}}$$

Inductor Selection

The buck converters data sheet will have a range of recommended inductor values, the higher the maximum output inductor, the higher the maximum output current due to reduced ripple. If no recommended inductor is given in the buck converters data sheet the following equation can be used:

$$\mathbf{L} = \frac{\mathbf{V_{OUT}} \times (\mathbf{V_{IN}} - \mathbf{V_{OUT}})}{\triangle \mathbf{I_L} \times \mathbf{f_s} \times \mathbf{V_{in}}}$$

Rectifier Diode

The chosen rectifier diode needs to be able to handle the fast switching and the current loads experienced during operation. The foreword current rating of the rectifier can be calculated using:

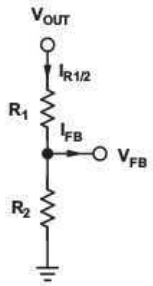
$$\mathbf{I_F = I_{OUT(MAX)} \times (1 - D)}$$

Schottky diodes can be used to reduce the losses experienced, due to there fast switching capabilities and the low voltage drop across the diode. The power that the chosen diode needs to be able to safely dissipate is calculated:

$$\mathbf{P_D = I_F \times V_F}$$

Output Voltage

Most buck converters set the output voltage with a resistive feedback divider network with resistors R_1 and R_2. This determines the feedback voltage V_{FB}. With V_{OUT} the desired output voltage and the feedback bias current, I_{FB}, the resistors for the voltage divider can be calculated:



$$\mathbf{R = \frac{V_{FB}}{I_{R\frac{1}{2}}}}$$

$$\mathbf{R_1 = R_2 \times (\frac{V_{OUT}}{V_{FB}} - 1)}$$

Figure 2.21: Resistive Divider

Input Capacitor

The minimum value of the input capacitor is normally given in buck converter ICs the data sheet. This is the minimum value necessary to stabilise the input voltage due to the peak current requirement of a switching power supply.

Output capacitor

The recommended value for the output capacitor is usually given in the buck converter ICs data sheet. The output capacitors value for a desired output voltage ripple is given by:

$$\mathbf{C_{OUT(min)}} = \frac{\mathbf{\triangle I_L}}{\mathbf{\times f_s \times \triangle I_L}}$$

If the output capacitors selection is based in transient response and not steady state ripple, the following formula can be used to calculate the output capacitance required for a desired minimum overshoot:

$$\mathbf{C_{OUT(min),OS}} = \frac{\mathbf{\triangle I_{OUT} \times L}}{\mathbf{2 \times f_s V_{OUT} \times V_{OS}}}$$

2.7 Voltage Regulator IC

The on board components will run at the same or lower voltage than that of the servo-motor and in order to step down the voltage an on board voltage regulator IC will be required.

Linear voltage regulators make use of a BJT or MOSFET device controlled by a high gain differential amplifier. The output voltage is compared to a reference voltage created by resistors R1 and R2 in Figure 2.22 The output voltage is constantly adjusted to match this reference voltage in order to maintain a regulated output.

Although the implementation of a linear regulator is simple, it is often a more inefficient method of regulation. The transistor or MOSFET acts as a resistor and any excess electrical energy is dissipated as heat(47) .

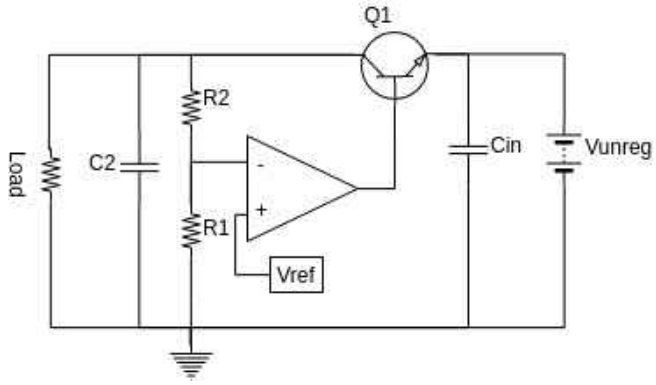


Figure 2.22: Typical voltage regulator IC

It is common to find fixed three terminal linear regulators that generate a fixed standard voltage needed by common microcontrollers and peripherals such as the 3.3V or 5V without requiring any external components.

2.8 Related projects

2.8.1 Dynamixel

The Dynamixel is an energy efficient high performance actuator that includes a controller, motor driver and network capabilities allowing multiple motors to be communicated to on a single channel.

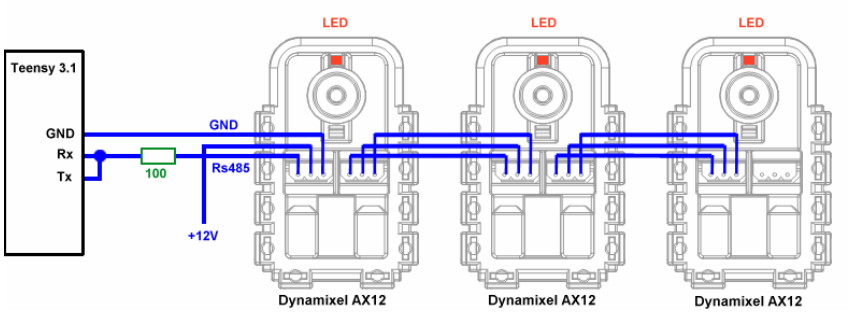


Figure 2.23: Dynamixel connectors (41)

The Dynamixel allows for position and velocity control, as well as current based torque control of a brushless DC motor. This allows for 360°rotation with six standard operating modes ,although it can be programmed specifically for each application, this allowing for wide range implementation. Temperature and current tracking allow for shutdown protection. Due to high cost only larger companies and well funded organisations tend to have access to these devices. A standard Dynamixel actuator can cost up to R4500.

Figure 2.24: Dynamixel based robotic arm (41)

2.8.2 Odrive

Odrive is a high performance servomotor controller board which uses a high resolution encoder with brushless DC motors in order to achieve high accuracy position and velocity control.

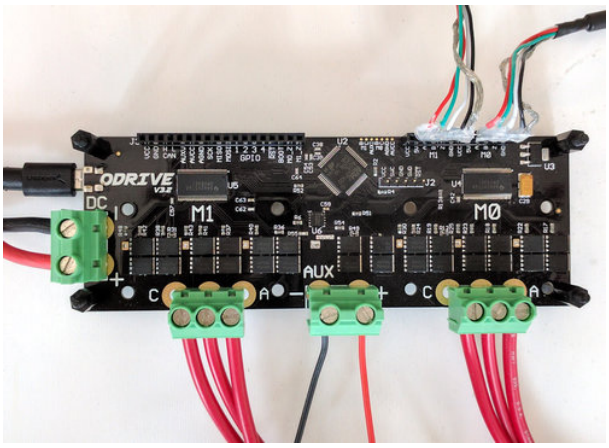


Figure 2.25: Odrive controller board(42)

The servomotor controller board can drive to motors. It allows for current, position and velocity control. Motor breaking can be achieved with a brake resistor and the board supports regenerative breaking. Communication with the board can be achieved with multiple interfaces, such as USB, UART, PWM and CAN. Although the Odrive is cheaper than the Dynamixel, it still costs R1800 for just the controller board.

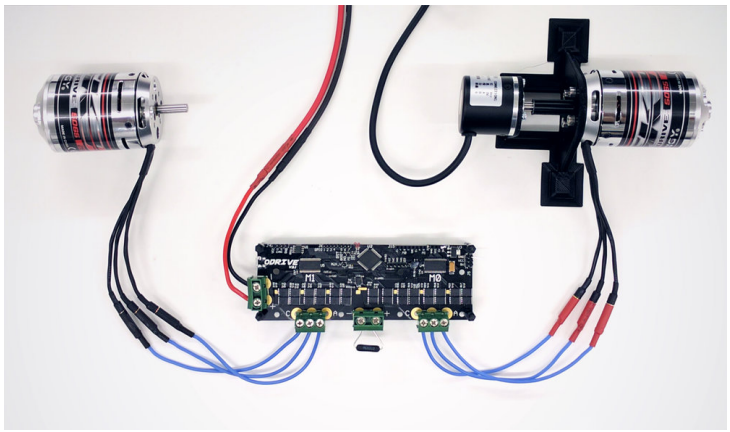


Figure 2.26: Odrive setup(42)

3 | System Design

The project was broken up into multiple stages, of which lead up to the design and manufacture of a PCB. This was used to create a list of goals that were required to be completed in order to successfully complete the project. This is also used to track the project progress in order to ensure that each stage of the project is accomplished in a timely manner.

The project stages were:

1. Requirement Specification
2. Design
3. Component Testing
4. Software
5. PCB Design
6. PCB Testing and Assembly
7. Project Review

3.1 Requirement specification

The designed servomotor control board will need to have the following capabilities in order for it to be able to achieve the same functionality that devices such as the Dynamixel and other actuators can achieve.

The following functionality is required:

Control	Communication	Sensors
Position	UART	Current
Current		Temperature

3.2 Hardware Design

An initial board layout based on the design specifications and requirements was created seen in Figure 3.1. This was used to determine all necessary components that would required. This is then also used when positioning the different components on the PCB.

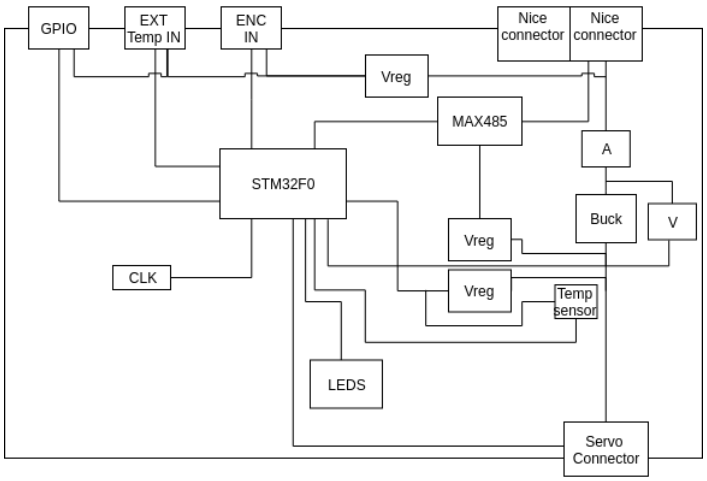


Figure 3.1: Basic board design

A flow diagram representing the two aspects of the system, power and communication can be seen in Figure 3.2.

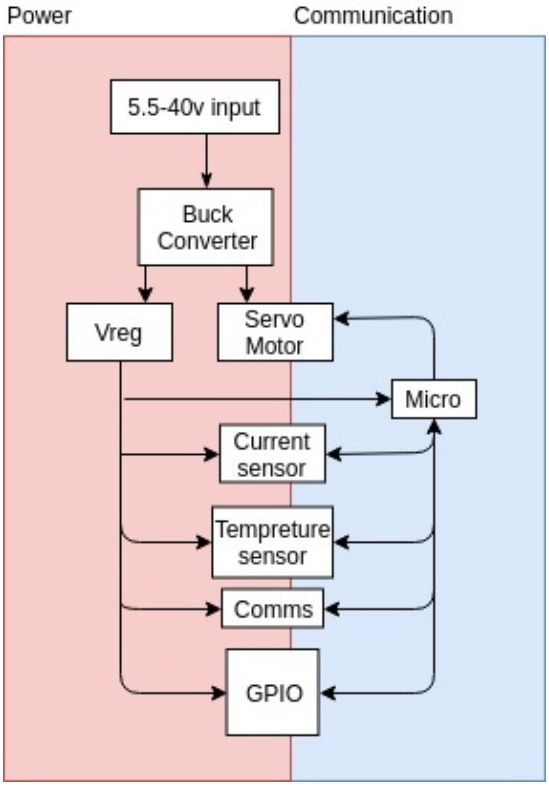


Figure 3.2: System flow diagram

3.3 Software Design

The Slave microcontrollers will be programmed to interact with all its peripherals and perform communication using standard protocols. The Master microcontroller will control each Slave and be able to request information such as temperature and current drawn. A basic diagram of the interaction between the microcontroller and the code running on them can be seen in Figure 3.3.

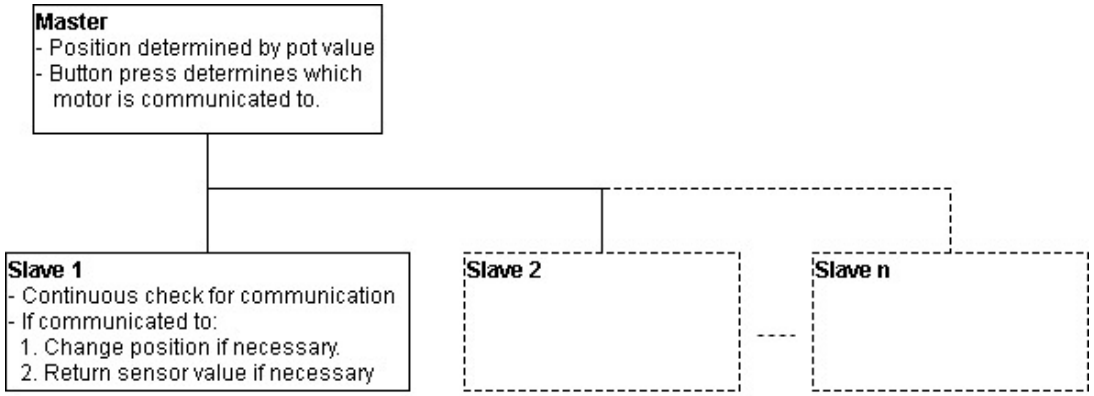


Figure 3.3: Master and Slave configuration

3.3.1 Master logic

The logic for the code running on master microcontroller can be seen in the diagram below in Figure 3.4.

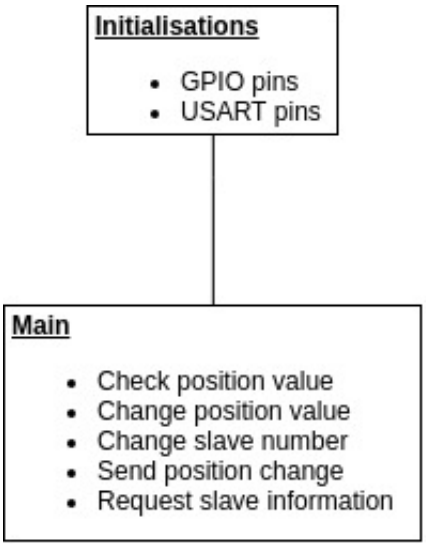


Figure 3.4: Master code flow

3.3.2 Slave logic

The logic for the code running on the slave microcontrollers can be seen in the flow diagram below in Figure 3.5.

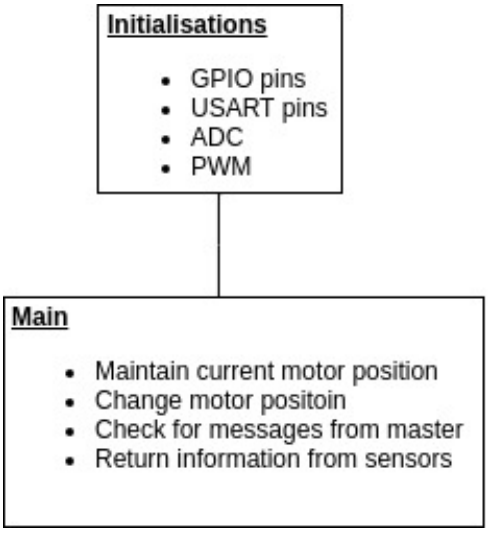


Figure 3.5: Slave code flow

3.4 Component Selection

After the system requirements have been established, and a basic system design has been created. The necessary components can be chosen.

The components that were chosen for the board had to be small enough to make the board compact, but still big enough so that it could be soldered down by hand accurately. Ideally a pick and place machine could be used to do the soldering, allowing for smaller components to be chosen, but for the sake of producing a functioning prototype it was only practical to choose slightly bigger components.

3.4.1 Microcontroller

The chosen micro controller needed to have the ability to interact with all the peripherals on the board as well as complete necessary computations in order for it to be able to control the servo motor board. As well as meeting the technical specifications, the chosen micro needed to be well documented in order to simplify the coding and debugging process.

The STM32F051C6 has an ARM Cortex-32-bit RISC core operating at up to 48MHz. This microcontroller was chosen as it met all the requirements and includes the following peripherals and functionality:

Features	**Types of communication**
64Kbytes of Flash	I2C
12 bit ADC	SPI
12-bit DAC	I2S
six 16-bit timers	HDMI
32-bit timer	USART
Advanced control timer	
Temperature sensor	

The STM32F051C6 supports hardware controlled RS485 USART communication which will allow for the communication between the micro controllers using the MAX485 chip. Its 12-bit ADC is a successive approximation analogue-to-digital converter.

It has up to 16 channels which can be used to read the output voltage from the current and temperature sensors as well as from other GPIO pins. A 16-bit timer will be used to output the PWM signal the will be used to control the servomotors position.With multiple low power modes minimising the overall power consumption, optimising performance and consumption which is very important for applications that are run off batteries.

In Figure 3.6, the necessary net labels were appended to the pins that are required for the microcontroller to function, as well as the pins that will be used to interact with the different peripherals.

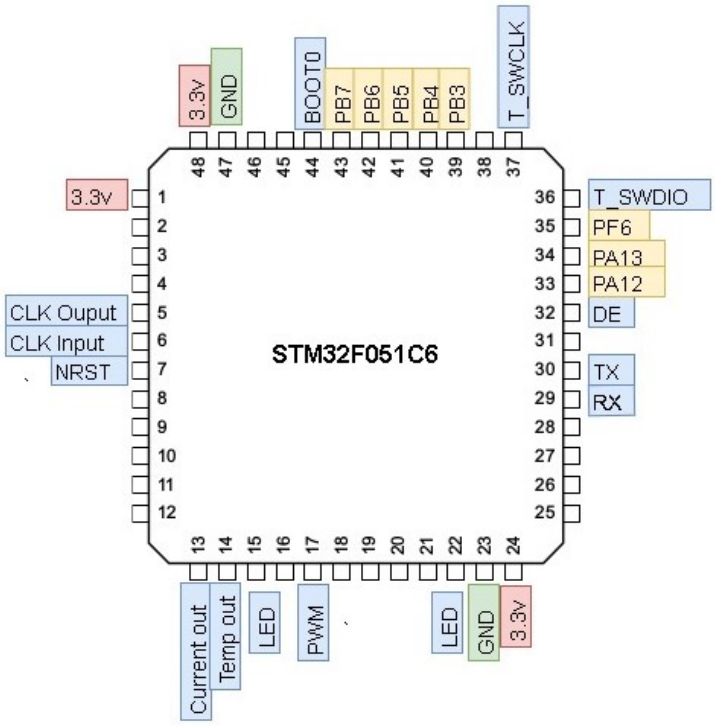


Figure 3.6: STM32F051 LQFP48 package(24)

The STM will be programmed using the Atollic TrueSTUDIO IDE. The code was then compiled to hardware code and uploaded to the microcontroller using the ST-link debugger supported by Atollic.

3.4.2 Temperature sensor

An analogue temperature sensor was chosen, the voltage output will be connected to one of the microcontrollers ADC. The MCP9700T has been designed for general purpose temperature monitoring and is typically used in home appliance and office equipment as well as hard disks and other PC peripherals.

Figure 3.7: MCP9700T chip (43)

The MCP9700 has the following features:
- Input voltage range of 2.3V to 5.5V
- Operates between -40°C to 125°C
- 2°C accuracy between 0°C to 70°C
- 10.0mV/°C resolution

Table 3.1: MCP9700 features

Unlike resistive sensors, the Linear Active Thermistor IC does not require any signal conditioning and the Vout pin should be directly connected to the input of a microcontroller. The temperature coefficients are scaled to provide a 1°C/bit resolution for an 8-bit ADC with a reference voltage of 2.5V and 5V respectively. This device is immune to the effects of parasitic capacitances which provides flexibility with the PCB layout as the device can be remotely located from the micro controller.

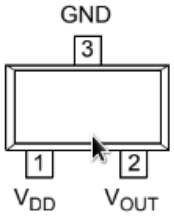


Figure 3.8: MCP9700 typical setup

3.4.3 Current sensor

The chosen current sensor was the INA169 high side current shunt monitor. It is an analogue current sensor that produces and output current relative to that of a differential input voltage. The input voltage is read from either side of a current sensing resistor RS. The output current is then converted to a voltage across a load resistor RL. The voltage gain depends on the size of the Rl and thus can be controlled to have an output voltage of the same range as the input to the microcontrollers ADC. As a result the output can be connect directly to the input of the microcontroller ADC.

This type of current sensor is typically used in automotive, computer and cellphone power management amongst other things.

Figure 3.9: INA169 chip (26)

The INA169 has the following features:
- Input voltage range of 2.7V -60 V
- Functions normally from -40°C to 85°C
- A single resistor sets the gain.

Table 3.2: INA169 features

The typical current sensor configuration can be seen below in Figure 3.10 with $RL = 10k\Omega$ and $RS = 0.1\Omega$. This gives the current sensor the ability to accurately sense voltage in the range of 350mA-3.5mA which is what can be required from the servo motor at higher loads. A 0.1uF capacitor is recommended between the V+ input and ground. The INA169 can be set up with different values of Rs in order to measure different current ranges. The different resistances associated with the different current ranges can be seen in the table below (27).

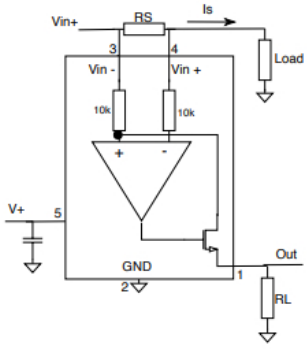


Figure 3.10: INA169 configuration

Rs	**Range**
10Ω	3.5mA-35mA
1Ω	35mA-350mA
0.1Ω	350mA-3.5A

Table 3.3: Current sense range.

3.4.4 MAX485

The MAX485 is a low power level shifting RS485 transceiver chip which operates from a single 5V supply. The IC contains one driver and one receiver. The receiver has an input has a fail-safe feature that guarantees a logic-high output if the input is open circuit.

Figure 3.11: MAX485 chip (44)

The MAX485 has the following features:
- Common-Mode input voltage range from -7V to 12V
- Functions normally from 0°C to 70°
- Allows for up to 32 transceivers on the bus
- Has a data rate of 2.5Mbps
- Current-limiting shutdown protection
- Thermal shutdown protection

The data sheet supplies all the information required to run the MAX485 chip, this was used to create the pin setup seen below in Figure 3.12. Which was used when creating the testing methods.

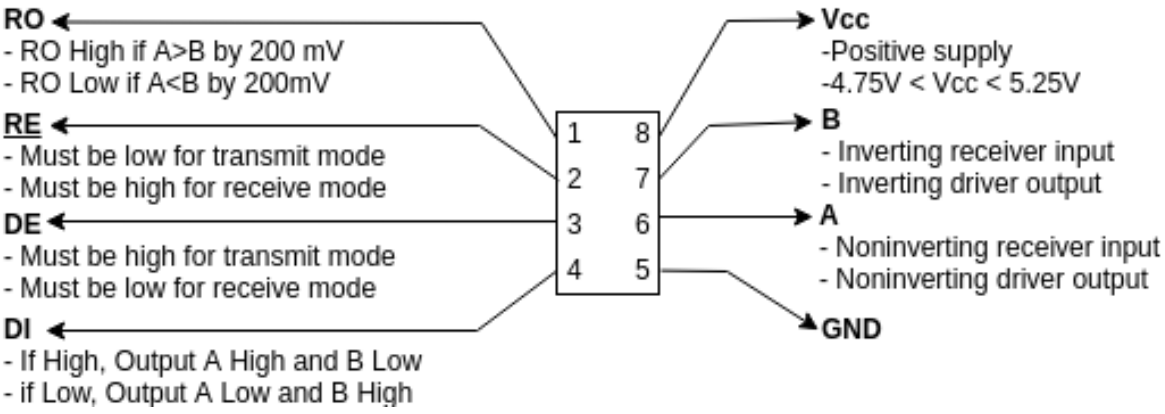


Figure 3.12: MAX485 pin setup

3.4.5 Communication lines

The connector cable used for the communication will be constructed from two wires that twist together over the length of the connection and terminated with termination resistors between the two cables at the beginning and end of each connection. The twisting of the wires is done for the purpose of cancelling out any electromagnetic interference from any external source as well as any noise produced from switch mode power supplies nearby(39).

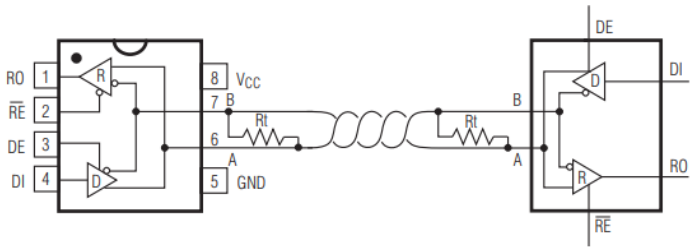


Figure 3.13: Differential communication over the twisted pair (29)

Termination resistors, Rt, are used to match the impedance of a transmission line to the hardware impedance of the interface it is connected to. This allows the the signals to be received with maximum power. Unterminated, or termination with some value unequal to the impedance of the cable, leads to mismatch that creates reflections in the network. Reflections are part of the energy of the signal that literally returns back up the line. Reflections can constructively or destructively interfere with the next bits propagating down the bus(30).

3.4.6 Connector

The chosen connectors for the servo board had different requirements. The connectors from the buck converter to the servo motor needed to be able the high currents that the motor could draw while the rest of the pins will be used for low current applications. For this, the Molex through hole male pin connectors were chosen which can handle up to 4A.

Figure 3.14: Molex though hole male pin connector(31)

3.4.7 Buck Converter

The buck converter chosen for this design needed to output 5V at up to 4 A, with an input voltage range of 5.5-40 Volts. While being small enough to minimise space but big so that it can be still soldered by hand. Using Texas Instruments online power supply designer, Webench. Webench generates the required circuit component values when choosing a buck converter. There were four options where available, seen below in Table 3.4.

1. LM5088 HTSSOP (16 pin) 5.00 mm × 4.00 mm package
2. TPS54561 WSON (10 pin) a 4.00 mm × 4.00 mm package
3. TPS54560 HSOP (8 pin) 4.89 mm x 3.9 mm package
4. LM5145RGYR VQFN (20 pin) 3.50 mm × 4.50 mm package

Table 3.4: Available buck converters

The buck converter chosen was The LM5088, a high voltage non-synchronous buck controller which features all the necessary functions to implement an efficient high voltage buck converter using a minimum number of external components. This Buck controller is typically used for motor driver and USB powered applications.

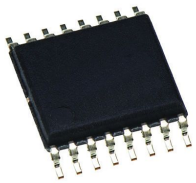


Figure 3.15: LM5088 (32)

The LM5088 has the following features:
- Operates between -40°C to 125°C
- Input voltage range of 4.5V to 75V
- Thermal shutdown protection
- Adjustable output voltage from 1.205V

Table 3.5: LM5088 features

A simplified schematic of the typical setup of the LM5088 with all its needed external components can be seen below in Figure 3.16.

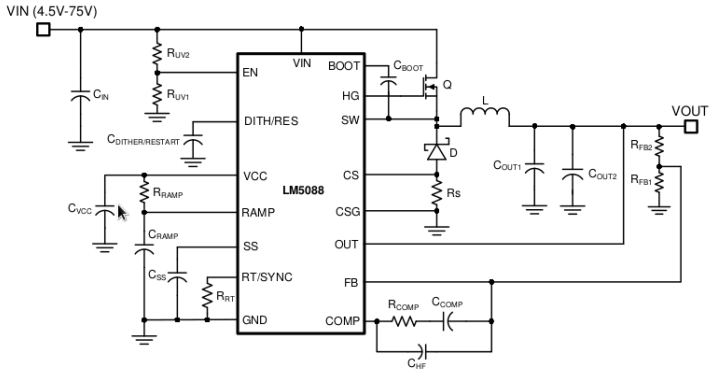


Figure 3.16: LM5088 typical setup

3.4.8 Voltage regulator

The voltage regulator chosen needed to be able to regulate the 5V output from the buck converter to the 3.3V needed by the microcontroller to run. The MCP1702 chip is a low power compact chip that only requires an input and output capacitor to run.

Figure 3.17: MCP1702 chip

The MCP1702 has the following features:
- Input voltage range of 2.7V to 13.2V
- Output 3.3V with a 0.4% tolerance
- Short circuit protection
- Thermal shutdown protection

Table 3.6: MCP1702 features

The MCP1702 is a low dropout (LDO) voltage regulator. Typically used in battery powered applications, smoke detectors as well as digital cameras amongst other devices and applications. A simplified schematic of the typical setup of the MCP1702:

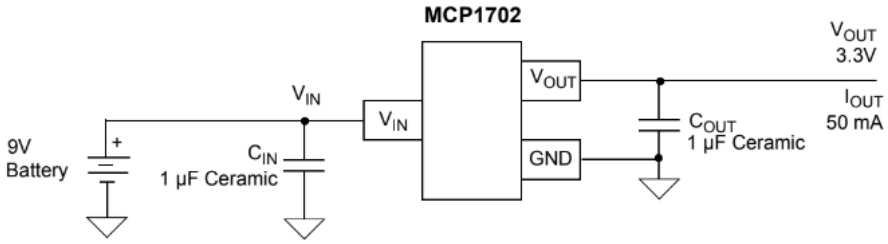


Figure 3.18: MCP1702 typical setup

4 | Component Testing

The components chosen where all Surface-Mount Devices (SMD) in order to minimise the size of the designed PCB. In order to test these components, they were soldered onto pieces of Vero board with male headers so the the component can be connected to a bread board for testing. Any code that was used has been included in the testing procedure and schematics are available in the GitHub repository.

4.1 Microcontroller

To test the basic functioning of the micro controller, a schematic was designed with the minimum components required to run the microcontroller and allow for a simple test. The microcontroller will simply turn an LED on and off every 100 mili seconds. This was then implemented on a bread board.

The designed schematic included the necessary components required to run the micro controller. This included an 8MHz external clock, a reset push button, GPIO and debugger pins as well as an LED indicator. As well as the USB debugger setup.

Implemented code:

Initialisations:	**Main:**
1. GPIO pins for LED's	1. LED on 2. 100ms delay 3. LED off 4. Item 100ms delay

Table 4.1: Microcontroller code implementation

4.2 Temperature Sensor

In order to test the temperature sensor. A test would need to be created where the sensors output voltage can be measured and related to the temperature. This test requires that the temperature sensor used for the reference has been calibrated. The reference sensor and the sensor being tested would then be exposed to various temperatures and the results will be tabulated and the sensed temperature can then be calculated using the fact that the sensor output 500mV at 0°C, found in the datas heet .Thus at 25°C, the Temperature= $\frac{\mathbf{Vout-500mV}}{\mathbf{gain}}$.

An alcohol thermometer was used as the reference temperature sensor. The thermometer was calibrated by submerging it in an ice bath and reading the temperature after five minutes. The thermometer read 0°C, the temperature of ice melting, which meant it was calibrated.

The device was connected following the schematic, but no other components are necessary to run the device. It is suggested in the data sheet to connected a 0.1uF capacitor between V_D and Ground and the gain was set to 10 by using RS = 10Ω. The chip was then exposed to multiple temperatures while recording Vout using an oscilloscope. Vout was then measured and the actual and sensed temperatures were tabulated below in Table 4.2.

Actual Temperature (°C)	**Vout (V)**	**Sensed Temperature(°C)**
25	0.738	23.8
28	0.775	27.5
50	0.984	48.4
62	1.098	59.8
68	1.11	61.0
87	1.25	75.0

Table 4.2: Temperature sensor results

The temperature sensor worked with expected accuracy. The MCP1700 data sheet specifies a 2°C error for temperatures less than 55°C. This accuracy can be seen between 25°C-50°C. For the results above 55°C it can be seen that as the variance increases as the temperature increases.

4.3 Current Sensor

In order to create a stable current source for the current sensor, A resitor bank is used for the load. This will allow for the resistance to be changed incrementally by connecting at different points along the resistor bank. Along with a known resistance, applying a constant 10V across the output load will allow for the current through Rs to be known and easily controllable. The INA169 was in its standard configuration. RL was chosen to be 10kΩfor a gain of 10. For testing purpose a current range of 1mA - 30mA would be tested for and an Rs of 10Ω. This will allow for sensing at this current range. A lower current range was chosen for testing as low currents are safer and is cheaper to work with. When creating the resistive load bank, it had to be taken note of that the max differential voltage that can be experienced between Vin+ and Vin- is 500mV which outputs an Iout of 500uA. This required that the load needed to have a resistance greater than at least 10 times more than that of the sensing resistor RS.

The actual drawn current was simply calculated using $V = IR$. The current measured by the sensor was calculated using $Is = \frac{Vout*1k\Omega}{10*10k\Omega}$. The output current, voltage and sensed current equations were found in the data sheet.

$$\mathbf{Iout = gm(Vin+ - Vin-)}, \qquad \mathbf{gm = 1000\frac{uA}{V}}$$

$$\mathbf{Vout = \frac{Is*Rs*Rl}{1k\Omega}} \qquad \mathbf{Is = \frac{Vout*1k\Omega}{Rs*Rl}}$$

Table 4.3 shows the input and output, as well as the actual and calculated current values that were recorded during the testing. The load resistance in the table refers to the output load being used to create a specific current, not to be confused with RL the load resistor that determines the gain for Vout.

Load resistance (Ω)	**Iactual (mA)**	**Vout (V)**	**Is (mA)**
900	10	0.98	9.8
800	12	1.12	11.2
700	14	1.28	12.8
600	16	1.47	14.7
500	18	1.74	17.4
400	24	2.2	22
300	30	2.8	28

Table 4.3: Current sensor results

The current sensor was able to sense and calculate the output current with an error of about $+ - 2mA$. It was found that at resistances of less than 400Ωthe fraction of the voltage drop across Rs relative to the voltage drop across the load becomes big enough for it to effect the sensed current Is, resulting in a less accurate Vout voltage.

4.4 MAX485

Termination resistors of RT = 120Ωfor RS485 communication when impedance matching and a 1uF capacitor between V_{cc} and ground for to stabilising the voltage ripple. The MAX485 chip can be tested in its two states, transmitting and receiving.

Transmitting:	**Receiving:**
1. Connect chip to ground and power	1. Connect power and ground
2. Connect RE to ground	2. Write data to USART TDR
3. Connect DE to 5V	3. Connect RE to 5V
4. Supply DI with a PWM signal	4. Supply DI with a PWM signal
5. Use oscilloscope to probe pins A and B	5. Differential signal is applied to pins A and B
	6. Use oscilloscope to probe pins RO

Table 4.4: Transmission and reception tests

4.4.1 Implementation

When testing the MAX485 the transmission and reception test were both successful indicating a correct setup of the transceiver. The oscilloscope readings of both the transmission and reception tests can be seen below in Figures 4.1-4.4.

Transmission test

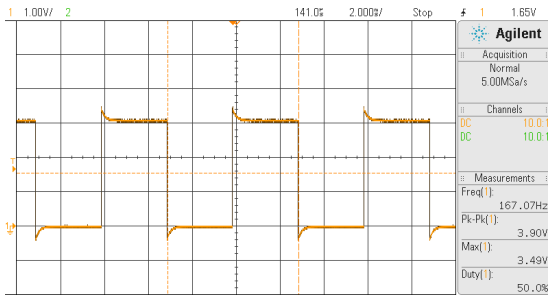


Figure 4.1: Signal to be transmitted

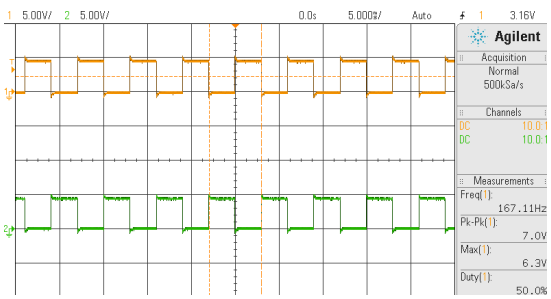


Figure 4.2: A and B output transmitted

Reception test

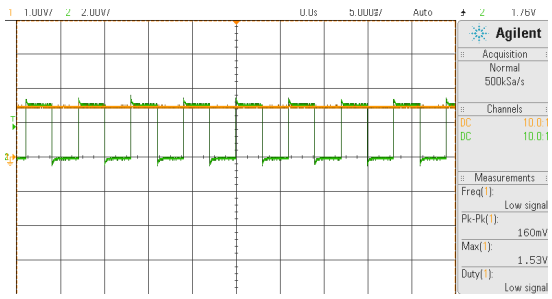


Figure 4.3: Signal being received

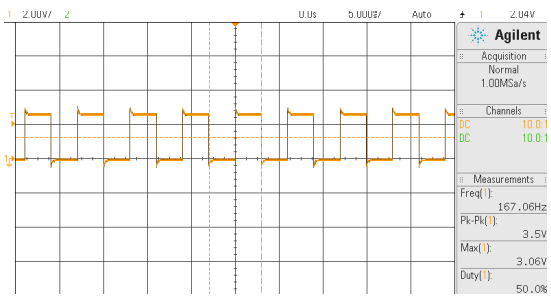


Figure 4.4: RO received signal

Table 4.5: Caption

4.5 Buck Converter

The data sheet for the LM5088 includes a typical setup for this particular buck converter. It goes through each step in the design procedure as well as all the equations needed to solve for all the necessary resistor and capacitor values as well as any other needed components that are required to run the buck converter.

Texas Instruments online power supply designer, WeBench was used to choose the buck converter chip and components required to produce the required output voltage.

The chosen buck converter design consisted of only surface mount components. The buck converter chip has a thermal pad at its base and with 16 pins it made it impossible to assemble the buck converter for testing.

Though it can be assumed that the designed buck converter works as Texas Instruments is reliable supplier of electronic equipment and components.

4.6 Voltage Regulator

The MCP9700 chip was connected to 5V and Ground. The output voltage was then measured at the output pin.

The regulator stability was then tested for various current draws using a resistor bank as a load. The results can be seen below in Table 4.6.

Load resistance (Ω)	I_{out} **(mA)**	**Vout (V)**
1kΩ	3.3	3.28
500Ω	6.6	3.32
250Ω	13.2	3.35
200Ω	16.5	3.33
125Ω	26.4	3.37
100Ω	33	3.38

Table 4.6: Voltage load stability

Various input voltages were applied in order to test the stability of V_{OUT} with a constant load. The results can be seen below in Table 4.7.

V_{IN}	**Vout (V)**
2	2.2
3	2.8
4	3.35
5	3.33
6	3.37
7	3.32
8	3.28

Table 4.7: Voltage stability

It was found that the voltage regulator functioned well for voltages above 2.5V and maintained a stable output voltage regardless of the change in input voltage.

5 | PCB Design

Once the components were chosen and there schematics drawn up. Altium Designer was then used to create the PCB design of the servo motor controller that could then be converted into a format that the PCB manufacturers, TraX, can then use to create the PCB. The board was designed in several stages.

5.1 Schematic

The first step in designing the PCB on was to create each of the designed schematics in Altium. All the necessary pins, net labels were connected up for each component. The schematics can be seen in the following pages.

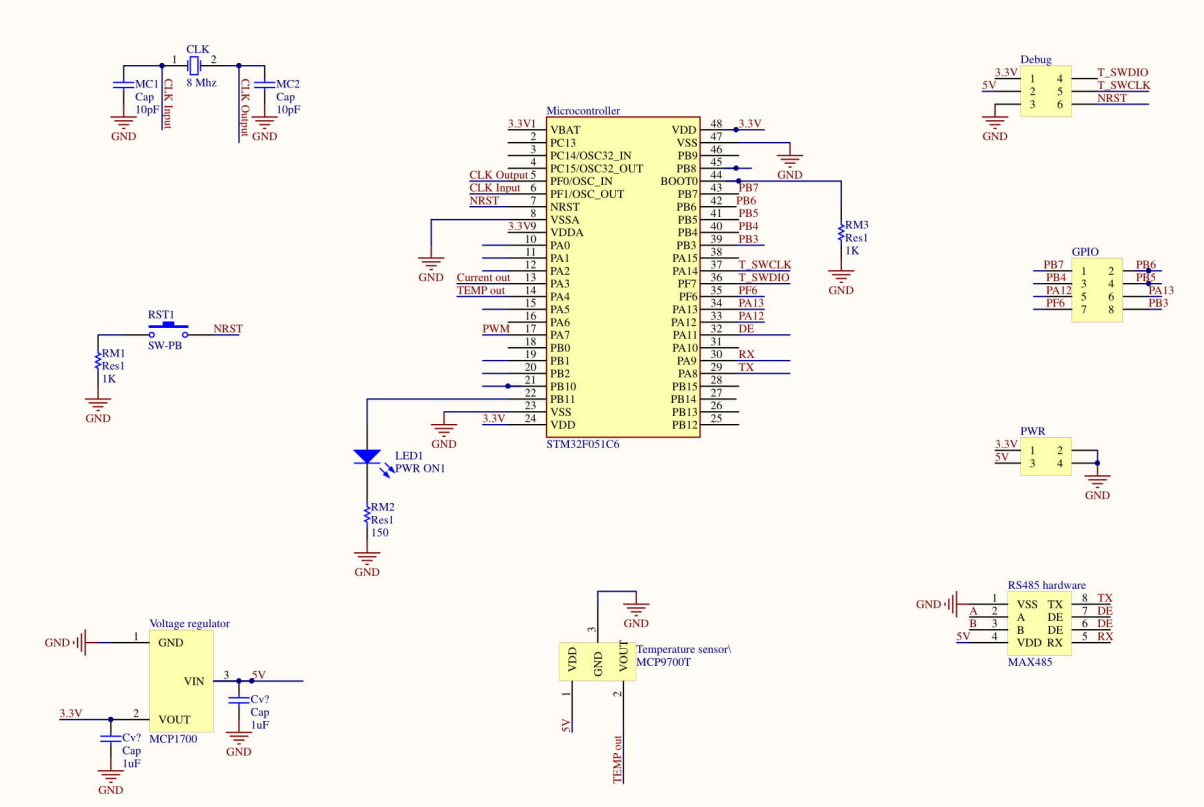


Figure 5.1: Microcontroller and components schematic

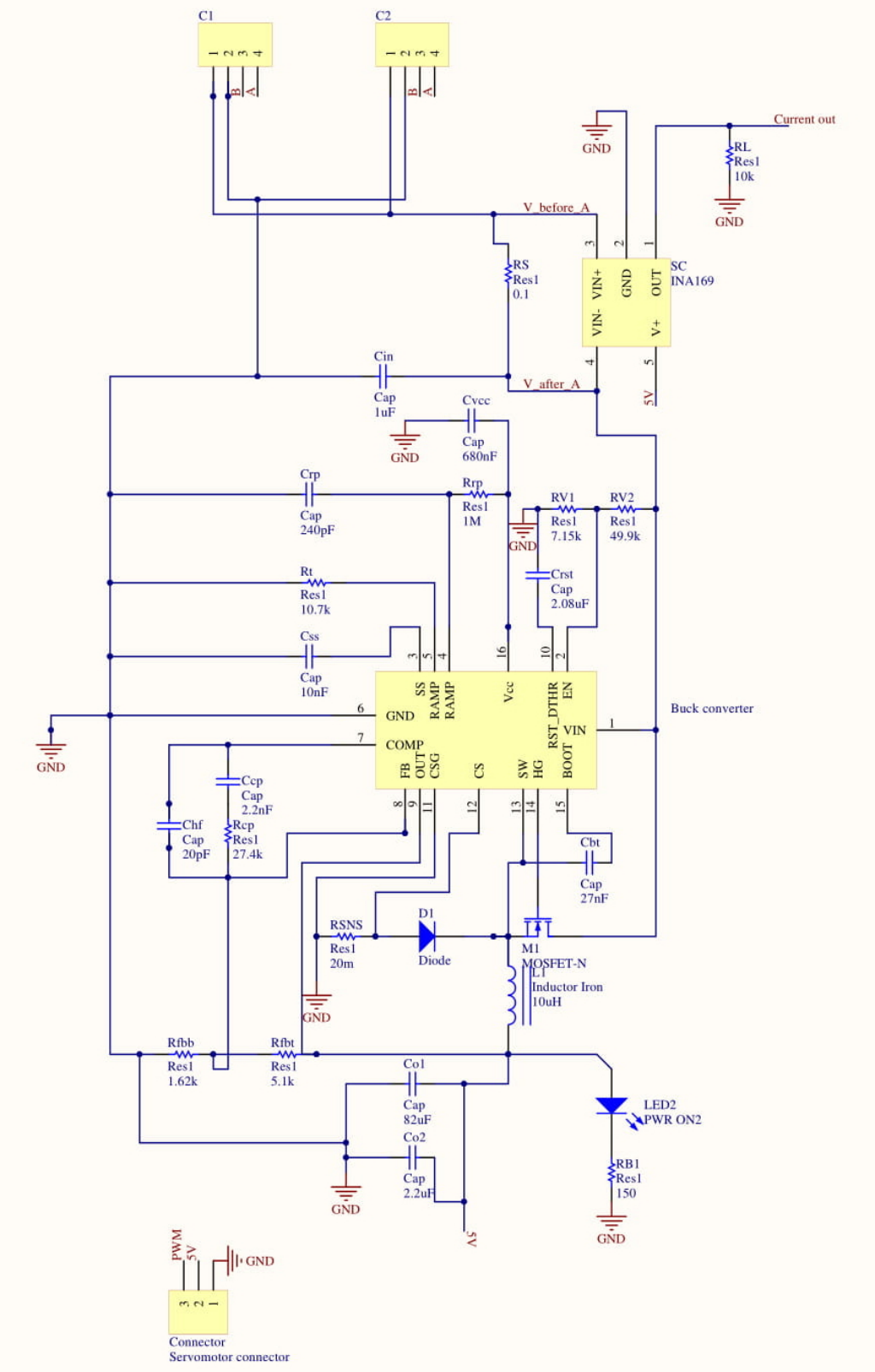


Figure 5.2: Buck converter schematic

5.2 Parts and footprints

Each component chosen will have an associated PCB footprint, where the solder pads and through connectors may be soldered. Most resistors, capacitors and inductors come in various standard package sizes, but other components such as the MOSFET or buck converter will have their own unique footprint. For these components the parts and their footprints had to be specifically made using measurements from the associated data sheet.

First each component needed to be designed in an Altium library. This is where the pin numbers and outputs are allocated.

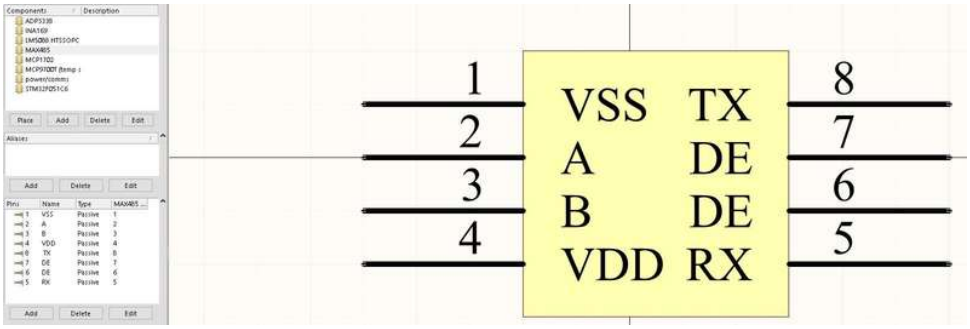


Figure 5.3: Creating a component

Each components footprint was also then created. This is where the actual solder pad for each pin is created.

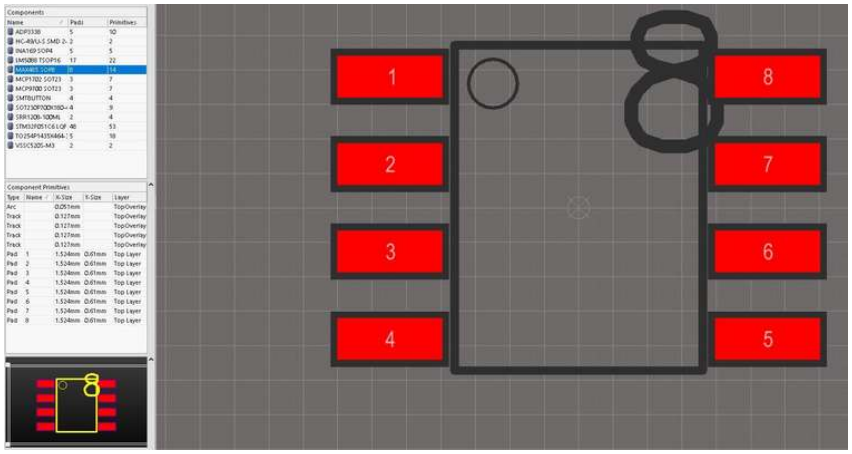


Figure 5.4: Creating a footprint

5.3 Board optimisation

Once the schematic was designed, and all the components and their footprints were either found in the free Altium libraries (Altium Vault) or created, a model of the PCB can be created. At this point the components can be placed and any physical specifications of the PCB can be defined.

In an initial design, the placement of components and the routing of the tracks had been done in a messy and inefficient way. The tracks created between components looped around other components or obstacles when attempting to take the shortest route. Laying tracks like this can lead to stray inductances, more importantly, the tracks between all the components were the same width and did not take current requirements into account which can lead to the tracks breaking and causing damage to the PCB. No ground plain had been used and this added to additional ground tracks running across the PCB.

5.3.1 Track parameters

The buck converter will be supplying all the current to the servo motor as well as to the rest of the components on the board board. The standard track width automatically set on Altium Designer is 0.25mm, which is much too thin to carry up the 4 Amps that the buck converter was designed to output. Using the following formula the necessary trace width was calculated (45).

$$Width = \frac{(Current[Amps]/(k*(Temprise[^{\circ}C]^{b})^{(}1/c)))}{(Thickness[oz]*1.378[mils/oz])}$$

Where :

k = 0.024
b = 0.044
c = 0.725
Current = 4 A
Temprise = 10 °C
Thickness = 0.725 mm

it was calculated that the track width would need to be roughly 2.8 mm, so a safe width of 4 mm was chosen for all the tracks that would be carrying high current. The lower current pins were given a track width of 0.8 mm.

5.3.2 Component positioning

The buck converters positioning relative to the rest of the components, especially the microcontroller, is an important factor. Switch mode supplies tend to generate a switching noise at the switching frequency which is generally between 500kHz and 3MHz (40). This was solved in two ways. First was to separate the buck converter from the microcontroller by putting the two devices on opposite sides of the board. The other was to create a ground plane on the top and bottom of the board. The ground plane would would then ground any stray EM signals that will be created. This would also neaten up the board by reducing the number of ground tracks that would be otherwise required. This was achieved using Altium Designers Polygon pours option setting.

Once the buck converter had been placed, the next few parts that needed to be positioned were the current and temperature sensors, the 3.3v regulator, and the MAX485. All of these were relatively straight forward, keeping in mind that the MAX485 needed to be as close to the micro as possible as to reduce any possible noise. Any input and output pins as well as the reset button and any LED's were then positioned in such a way as to optimise the functioning of the board as well as to keep it as compact as possible.

Optimised layout

The layout for the optimised PCB can be seen in Figure 3.18 with each individual components being highlighted.

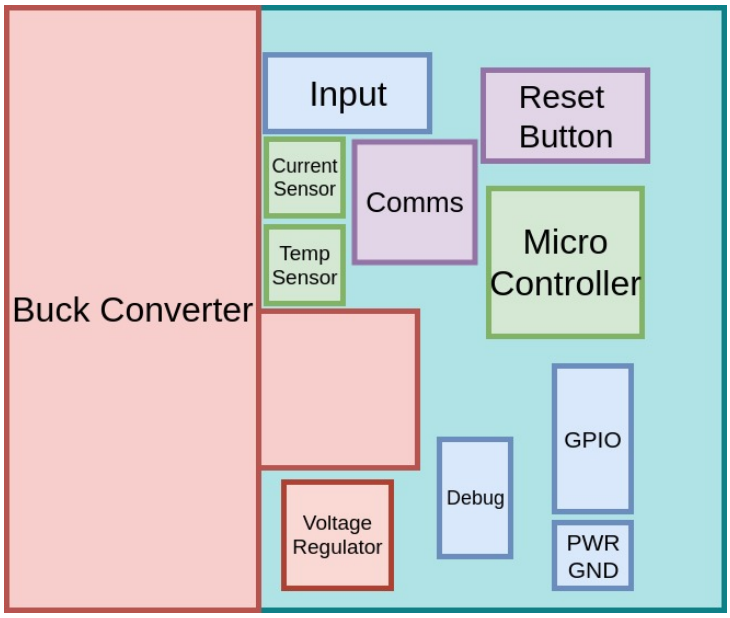


Figure 5.5: Optimised PCB layout

5.3.3 Final Design

The final schematic for the PCB design can be seen below in Figure 5.6. The top(red) and bottom(blue) layer tracks and pads can be seen. The top and bottom ground planes were removed for this schematic to make the tracks more visible.

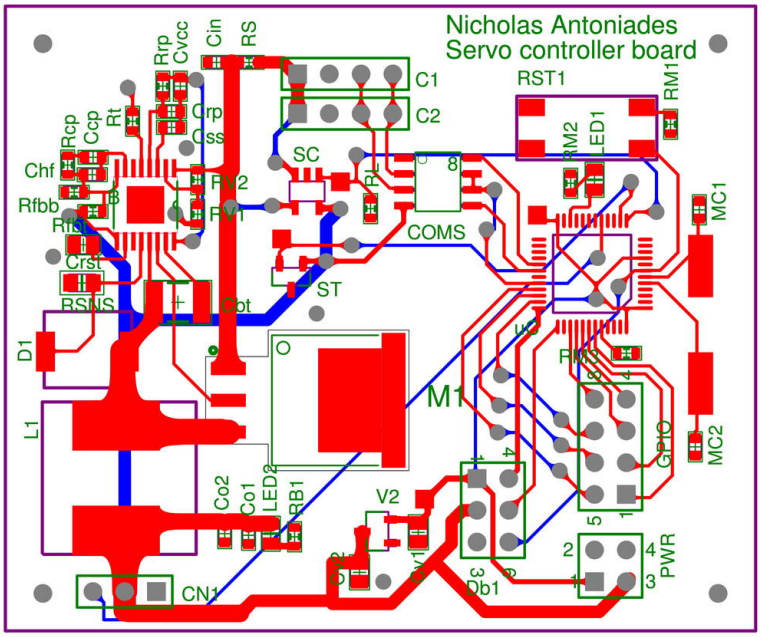


Figure 5.6: Final PCB design

Once the final design was completed it was exported to the Gerber format and sent of to Trax, where the actual PCB was manufactured.

5.3.4 Component List

Buck converter	R101.82	**Microcontroller**	R65.66	**Current sensor**	R38.88
R_{rp} = 1MΩ	0.63	R_{M1} = 1kΩ	0.62	R_{RL} = 10kΩ	0.63
R_t = 10.7kΩ	0.64	R_{M2} = 150Ω	0.66	R_{RS} = 0.1Ω	4.85
R_{V1} = 7.15kΩ	0.66	R_{M3} = 1kΩ	0.67	SC = INA169	33.40
R_{V2} = 49.9kΩ	0.73				
R_{cp} = 27.4kΩ	0..85	C_{MC1} = 10pF	2.35	**Temperature sensor**	R3.61
R_{RSNS} = 20MΩ	0.68	C_{MC2} = 10pF	2.35	ST = MCP9700T	3.61
R_{fbt} = 5.1kΩ	0.68				
R_{B1} = 150Ω	0.62	uC = STM32F051C6	35.17	**RS485**	R72.11
		8mHz crystal	4.83	COMS = MAX485	72.11
C_{hf} = 20pF	0.56	LED	5.23		
C_{cp} = 2.2nF	0.32	SMD push button	13.78	**Voltage regulator**	R8.28
C_{bt} = 27nF	0.61			V2 = MCP1700T	8.28
C_{o1} = 82uF	5.23				
C_{o2} = 2.2uF	1.53				
D1 = VSSC520S-M3	8.45				
M1 = IPD50N10S3l16	14.67				
L1 = 10uH	10.12				
B1 = LM5088	54.52				
Total cost: R290.36					

Table 5.1: Component list

6 | PCB Assembly and Testing

The PCB would be assembled in sections. All the components required to get a certain aspect of the PCB functioning would be soldered down and then that aspect will be tested.

Components to be assembled :

1. Buck converter
2. Voltage regulator
3. Temperature sensor
4. Current sensor
5. Microcontroller
6. MAX48

The voltage regulator, MAX485, temperature and current sensor chips all had simple configurations and when tested worked as expected as seen in chapter 4, Component Testing. The microcontroller and buck converter required a more complicated setup. The two chips and the components necessary for their functioning can be see soldered down in Figure 6.1.

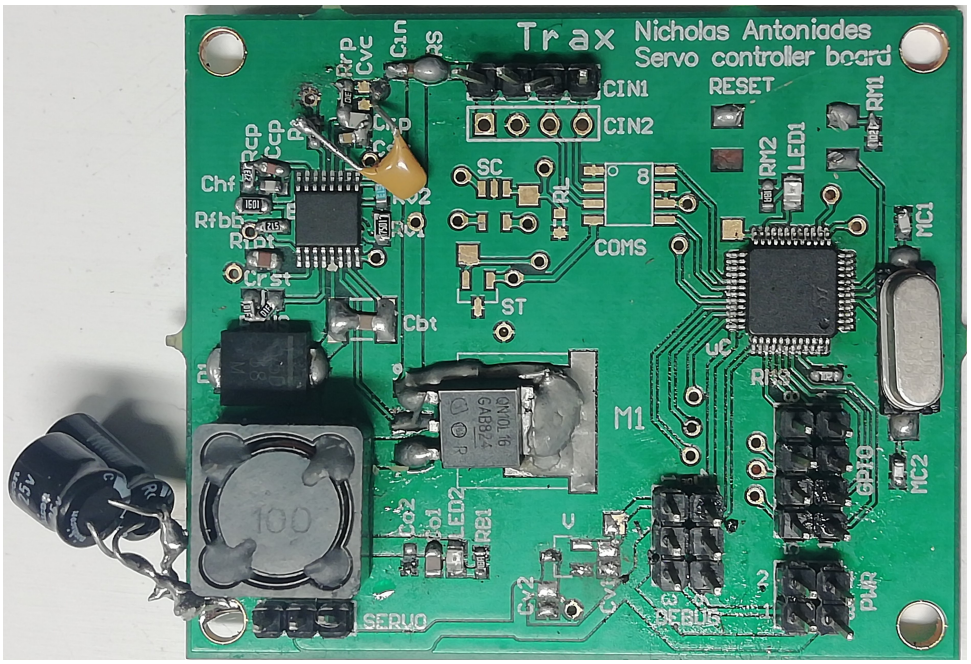


Figure 6.1: Microcontroller and buck converter on PCB

6.1 Buck Converter

The buck converter chip and all necessary components required for it to function were soldered on to the board. Headers were soldered down enabling the power, ground and output cables to be to connected allowing testing. A couple issues came up when soldering the buck converter to the PCB.

1. The capacitor values designed for had were only available in the electrolytic packages. The through hole capacitors were soldered to the surface mount pads.

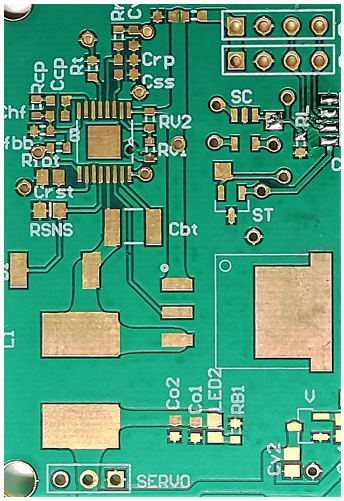


Figure 6.2: Buck converter footprints

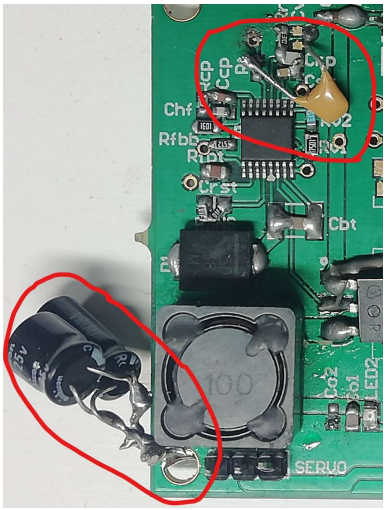


Figure 6.3: LM5088 soldered down

2. The MOSFET footprint that was designed had been chosen for a different MOSFET that was not available from suppliers. The pins of the new MOSFET were in a different order and a wire had to be soldered across the pad so that the in order to allow the MOSFET to be connected correctly. This can be seen in Figures 6.4 and 6.5.

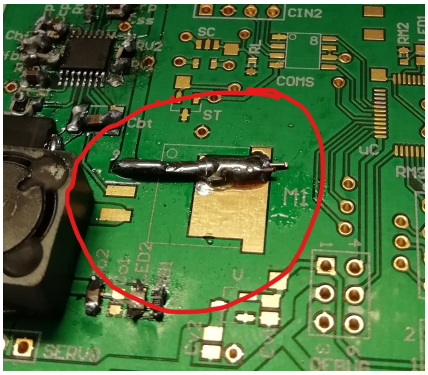


Figure 6.4: Correcting MOSFET

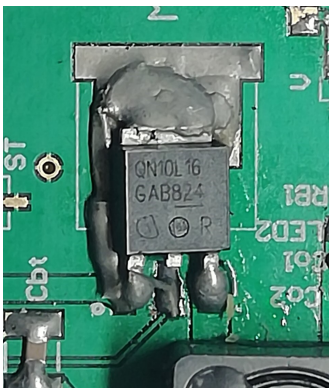


Figure 6.5: MOSFET soldered down

3. When designing the footprint, no solder mask was inserted between the pins of the buck chip. The result of this was when soldering, solder leaked over creating shorts, as well as letting pins getting close enough to arc. Seen in Figures 6.6 and 6.7.

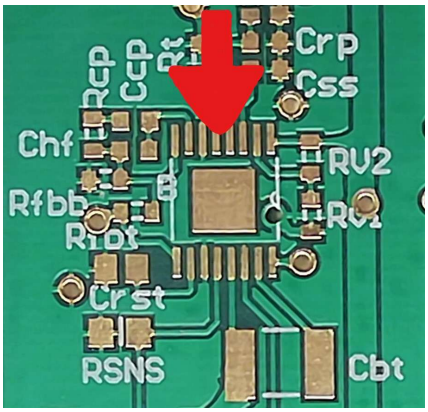


Figure 6.6: LM5088 footprint

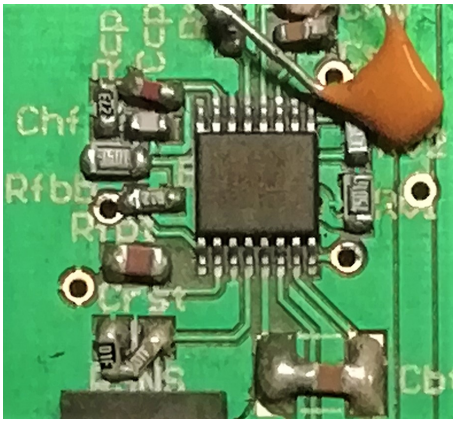


Figure 6.7: LM5088 soldered down

6.2 Microcontroller

The STM chip, clock, an indicator LED and necessary resistors and capacitors were placed on the board. Headers were soldered down enabling the power, ground and debugging cables to be to connected.

The same mistake that had been made in the buck converter footprint design was made for the microcontroller. When connected to power and ground, the pins would short stopping the circuit from working. This can be seen in Figures 6.8 and 6.9.

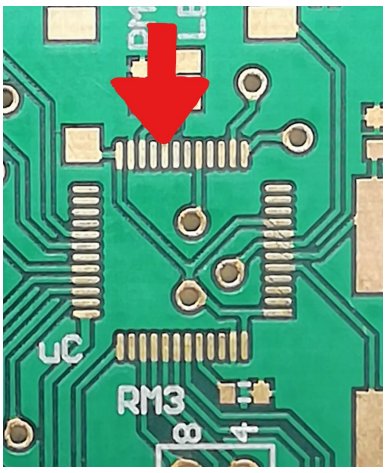


Figure 6.8: STM footprint

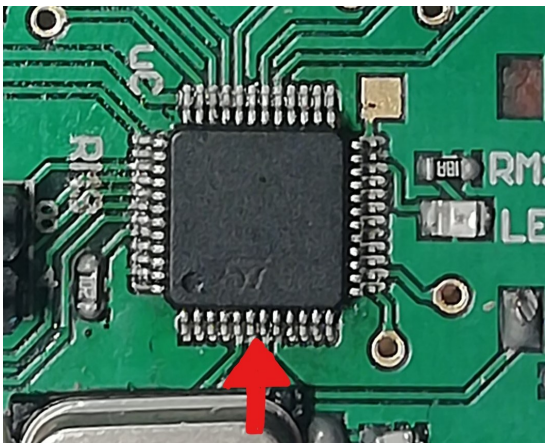


Figure 6.9: STM soldered down

7 | Software

The STM microcontroller has an ARM Cortex-32-bit RISC core that will be programmed in C using Atollic TrueSTUDIO. The IDE will compile and upload the code to the STM using the ST-link debugger and toolchain. All code can be found in the GitHub repository. **https://github.com/Marshmellowfellow/Thesis-Servocontroller-Board.git**

7.1 ADC

Initialisations:	**Main:**
1. Configure GPIO pins	1. Select channel to sample
2. Enable ADC clock	2. Activate ADC to be read
3. Set ADC parameters	3. Start conversion
	4. Read in ADC value

Table 7.1: ADC code implementation

7.1.1 Testing

In order to test the implementation of the ADC code a voltage in the range 0V-3V will be applied to the ADC pin of the micro controller. The ADC will sample with an 8-bit resolution. This will give $\mathbf{2^8 - 1 = 255}$ bits, giving a resolution of $\frac{\mathbf{3}}{\mathbf{255}} \frac{\mathbf{V}}{\mathbf{Steps}} = \mathbf{11.76} \frac{\mathbf{mV}}{\mathbf{step}}$.

The ADC will store an 8-bit value representing the sampled input into the ADC Data Register(DR). This value will then be written to the General Purpose Input Output(GPIO) Output Data Register(ODR). GPIO pins 0-7 will be set in output mode and connected to an LED. The binary value of the sample can then be output to the 8-bit LED output so that it can be read and compared to the actual voltage at the ADC input pin.

The binary value output will be a number between 0-255. The voltage calculated using $V_{calculated} = (ADC_{output}) \times (resolution)$. The applied voltages and sampled values can be seen below in ..table.. and were calculates using the resolution of $11.76 \frac{mV}{step}$

$\mathbf{V_{Applied}}$ **(V)**	**Output 8-bit value**	$VCalculated$ **(V)**
0.66	00110000	564.48
1.21	01011000	1.03
1.81	10000100	1.55
2.36	10101100	2.02
3.30	11111110	2.99

Table 7.2: ADC testing applied voltages

It can be seen that with some error, the ADC was cable of sampling a voltage. Thus the microcontroller can read an output voltage from the current and temperature sensors.

7.2 PWM

The following configurations were implemented in order to output a PWM signal at a GPIO pin.

Initialisations:	
1. Enable the GPIO clock	6. Set timer to PWM mode 1
2. Enable the timer	7. Set signal frequency
3. Set GPIO pins to AF	8. Set duty cycle
4. Map GPIO pins to the timer	9. Enable output compare
5. Map GPIO pins to the timer	

Table 7.3: PWM initialisations

7.2.1 Testing

The PWM configuration was tested by outputting a PWM signal and connecting to an oscilloscope.

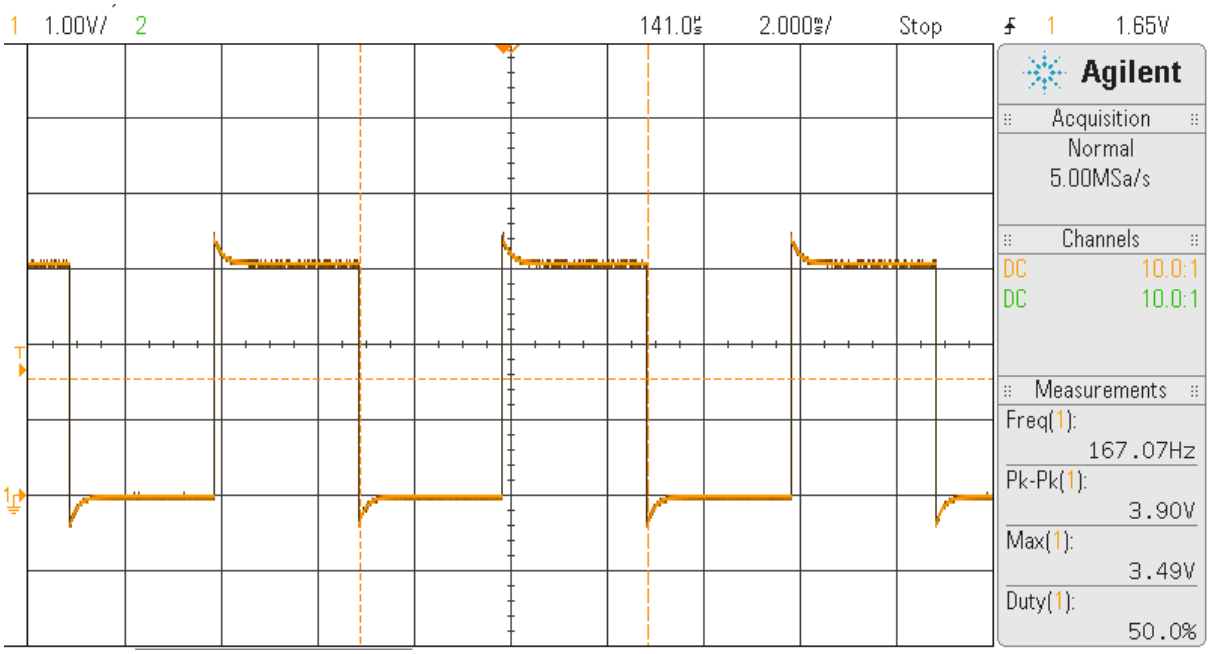


Figure 7.1: Oscilloscope output

It can be seen that a PWM signal with a 50% duty cycle was successfully implemented and is cable of control the servomotors position through varying the Duty Cycle of the signal.

7.3 USART

Initialisations:	**Implementation:**
1. Enable the UART clock	**Transmit**
2. Enable the GPIO clock	1. Write data to USART TDR
3. Set GPIO pins to AF	2. Store message into a variable
4. Clear USART control register	3. Clear USART ICR
5. Define clock speed	**Receive**
6. Set baud rate and parity	1. Wait for Message
7. Enable USART TX and RX	2. Store message into a variable
	3. Clear USART ICR

Table 7.4: USART initialisations and implementation

7.3.1 Testing

Transmit

The microcontroller was programmed to transmit the three messages "0000000", "1101011", "0000000" every 10 mili seconds using the serial protocol. The TX pin was probed with an oscilloscope and the transmitted message can be seen below in Figure 7.2 The "1's" represented by 3.3V and "0's" by 0V.

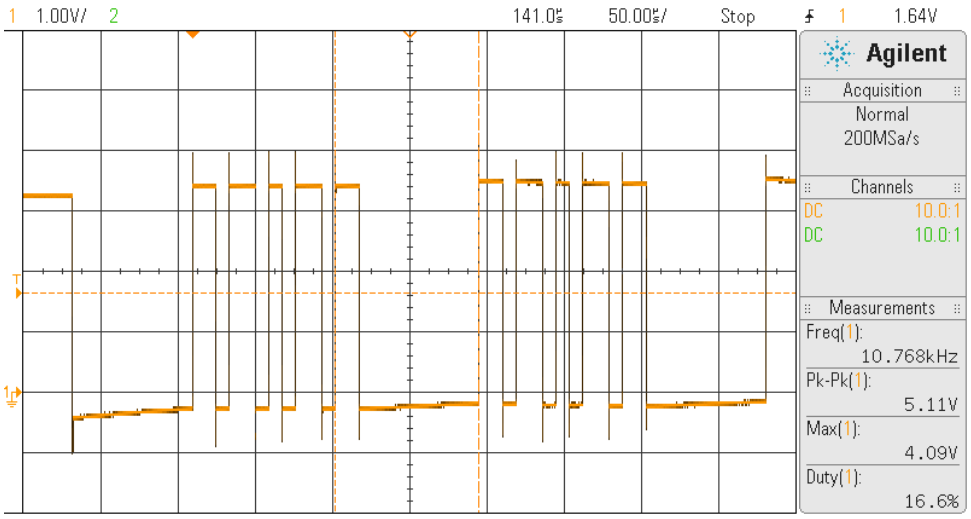


Figure 7.2: Oscilloscope output

Receive

A master microcontroller was programmed to transmit the three messages "0000000", "1101011", "0000000" every 100 mili seconds using the serial protocol. A slave was then made to receive the message and display the 8-bit output on 8 LED's to be compared to the 8-bit message sent. The implementation worked as designed.

7.4 Final code implementation

Initialisations:	**Initialisations:**
1. GPIO pins for LED's and switches	1. GPIO pins for LED's and switches
2. USART	2. USART
3. PWM	3. PWM
4. ADC	4. ADC
Main:	**Main:**
1. Check for button press	1. Check for button press
2. Read in voltage from potentiometer	2. Transmit message to master with USART
3. Transmit message to slave with USART	3. Wait for USART message from master
4. Wait for USART message from slave	4. Output PWM signal to servomotor
5. Control status LED	5. Control status LED
	6. Read in current and temperature sensors

Table 7.5: Final initialisations and implementation

8 | Conclusions

The design of the servomotor controller board was considered successful. Although the designed PCB for the system did not function, each aspect of the system, except for the buck converter, was individually tested providing a proof of concept. The components on the designed PCB had poorly design footprints and the result was shorting between the pins and the PCB not functioning.

The designed implementation of the buck converter could not be properly tested due to the fact that the components were surface mount making assembly and testing difficult. Despite this, it can be accepted that the deigned buck converter works if implemented correctly based on the fact that the design was obtained by Texas Instruments a reliable supplier.

The final design cost was R290.36, this did not include the cost of manufacturing the PCB or the cost that would be required to pay for automated component placement for accurate results. From the results of the testing and from the cost of the designed PCB, it can be seen that the system design was successful and can offer the advanced functionality of the more expensive controllers at a lower price.

9 | Recommendations

When selecting components for a design, it is important to ensure that the components chosen are available in the package being designed for before continuing with the design process.

When designing or finding a components footprint, ensure that the dimensions and pad placement matches that of the components pins by checking in its associated data sheet.

It is important to check that the pads for the pins of an IC component have enough space between them with solder mask is used in order to avoid accidentally creating a short during soldering and arcing due to difference in potential.

When choosing component placement, check that the components are not too close to each other. This causes difficulties in the process of actually soldering the component down when there are many other components are already on the board making it hard to access small pads with a soldering iron.

All the necessary components should have indicator LED's to indicate different system states for testing and debugging purposes. Test pads should also be included enabling easier testing and debugging of the PCb and must be included in the design process.

Bibliography

[1] https://en.wikipedia.org/wiki/Servomotor

[2] https://www.ebay.com/itm/Robot-Arm-Arduino-6-Axis-Servo-Control-Palletizing-Arduino-Mega2560-Robotics/121753587385

[3] https://lagunatools.com/cnc/iq-series/iq-24-36-cnc/

[4] https://myframe.co/servo-motor-wire-color-code/3/

[5] https://www.renesas.com/us/en/support/technical-resources/engineer-school/brushless-dc-motor-01-overview.html

[6] http://www.physicsinstuff.com/physics-in-dc-motor/

[7] https://en.wikipedia.org/wiki/Brushless DC electric motor

[8] https://en.wikipedia.org/wiki/Digital-to-analog converter

[9] https://en.wikipedia.org/wiki/Successive approximation ADC

[10] https://en.wikipedia.org/wiki/Current sensor

[11] http://www.optical-encoders.eu/optical-encoder.html

[12] https://www.globalspec.com/learnmore/sensors transducers detectors/rotary

[13] https://www.electronicdesign.com/components/understanding-resolution-optical-and-magnetic-encoders

[14] https://www.ametherm.com/blog/thermistors/temperature-sensor-types

[15] https://reviseomatic.org/help/2-control/Position Sensing.php

[16] https://botscene.net/2012/10/18/make-a-low-cost-absolute-encoder/ position sensing/rotary position sensors

[17] file:///home/marshmewllow/Desktop/Thesis/Figures/toomanywires.jpg.html

[18] https://www.electronics-notes.com/articles/connectivity/serial-data-communications/rs232-v24-basics-tutorial.php

[19] https://www.electronics-notes.com/articles/connectivity/serial-data-communications/rs422-basics-tutorial.php

[20] https://www.electronics-notes.com/articles/connectivity/serial-data-communications/rs485-introduction-basics.php

[21] http://www.betaengineering.com/high-voltage-industry-blog/transmitting-electricity-at-high-voltages

[22] http://www.ti.com/lit/an/slva477b/slva477b.pdf

[23] https://en.wikipedia.org/wiki/Linear regulator

[24] https://www.st.com/resource/en/datasheet/dm00039193.pdf

[25] https://za.rs-online.com/web/p/low-dropout-voltage-regulators/6989044/

[26] https://www.distrelec.nl/en/current-sense-amplifier-ic-sot23-ina169-texas-instruments-ina169na-250/p/30022559

[27] https://learn.sparkfun.com/tutorials/ina169-breakout-board-hookup-guide/all

[28] http://www.ni.com/white-paper/11390/en/

[29] https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf

[30] https://e2e.ti.com/blogs /b/analogwire/archive/2016/07/28/rs-485-basics-when-termination-is-necessary-and-how-to-do-it-properly

[31] https://za.rs-online.com/web/p/pcb-pin-socket-strips/8967620

[32] https://za.rs-online.com/web/p/dc-dc-controllers/7615283/

[33] http://www.powere.dynamictopway.com/dc2.htm

[34] M. S. Tsoeu and M. Braae, "Control Systems," *IEEE*, **vol. 34(3)**, pp. 123-129, 2011.

[35] J. C. Tapson, *Instrumentation*, UCT Press, Cape Town, 2010.

[36] https://www.wikipedia.org/wiki/Differential signaling/

[37] https://en.wikipedia.org/wiki/Serial communication

[38] https://en.wikipedia.org/wiki/Serial communication

[39] https://www.quora.com/What-is-the-purpose-of-a-twisted-pair-cablem

[40] https://www.digikey.com/en/articles/techzone/2016/may/minimizing-noise-generated-by-switched-mode-power-supplies

[41] https://www.generationrobots.com/en/401075-dynamixel-ax-12-a-actuator-robotis.html

[42] https://odriverobotics.com/odrive

[43] https://www.newark.com/microchip/mcp9700t-e-tt/temperature-sensor-2c-3-sot-23/dp/17M0678?CMP=AFC-ECIA

[44] https://www.ebay.com/itm/Brand-New-10pcs-MAX485CSA-MAX485-MAX485ESA-RS485-Transceiver-Chip-SOP8-SMD-/262951878549

[45] https://www.4pcb.com/trace-width-calculator.html

[46] https://www.paceworldwide.com/pacenter/soldering/lead-free-vs.-leaded-solder

[47] http://www.analog.com/en/technical-articles/how-voltage-regulator-works.html