SERQET

A 4th Year Engineering Project

By

Abdulrahim Kaddoura	100989743
Dalia Matar	101003740
Maria El-Hamad	101003010
Rami Daham	100995251

Supervisor: Dr. Peter X. Liu

Second Reader: Dr. Richard M. Dansereau

A report submitted in partial fulfillment of the requirements of SYSC-4907 Engineering Project

Department of Systems and Computer Engineering Faculty of Engineering

Carleton University

April 9, 2019

Abstract

The robotics industry is vastly improving on a daily basis. They build systems to improve the efficiency and ease of use of specific tasks and day to day activities. The Microcontroller acts as the main spinal cord of the system, it monitors and moves the scorpion according to specific instructions. The Raspberry Pi 3, RPi 3, acts as the brain of all operations, it communicates to all sub systems and computes all data. The mobile application is the main controller, used to send instructions and receive visual feedback. The modular structure of the system allows easy improvements and replacements for maintenance and upgrades.

Acknowledgements

On behalf of the whole team we would like to greatly thank Professor Peter Xiaoping Liu, the project supervisor, for giving us the opportunity to work on his proposed project. We would like to express our gratitude towards him and Professor Richard M. Dansereau for their advice and guidance throughout the duration of the project.

Furthermore, we would like to give our thanks to the Department of Systems and Computer Engineering for their role in planning the fourth-year project course and providing the labs and equipment necessary to accomplish most of the tasks related to the project, and thanks to Daren Russ for ensuring proper safety and guidance while using lab equipment.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

This report is a detailed overview of the fourth year Engineering Project, "SERQET", the bio-inspired scorpion robot. It involves Computer Systems Engineering students from Carleton University, and supervised by Dr. Peter X. Liu.

This document begins by discussing the project problem, its solution and the motivation behind the project deliverables. The solution and technical details are then discussed in detail to show the implemented sub systems and the full assembly. The report ends by listing the accomplishments and recommendations of the project while providing a brief conclusion.

Project Background

The Robotics industry is rapidly growing to design and implement robots capable of accomplishing specific industry tasks and day to day simple activities to improve efficiency and ease of use of said jobs. As a result, robotics has shown a gradual increase in its role in modern civilization with new wave of consumer robotics. The high exposure of these kind of robotics across platforms has expanded the scope of this industry to tackle multiple everyday issues starting from manufacturing robots, to industry specific robots, to cinematics and drones.

Biologically inspired robotics is a vast field of research that is allowing many robotic engineers to work alongside biologists. This partnership offers the opportunities to gain a better understanding of biologicals systems. The insight from a professional in nature helps engineers design and create systems that work better than those found in nature. Biological systems surround us in nature, it has always been a leading inspiration to a lot of robotic designs and functions. The study of bio-inspired robots is learning how different species work in terms of locomotion, behavior, intelligence and then being able to convert this information into a technological and mechanical representation.

This field is introducing researchers to different functions, structures and mechanisms that can be used to create and develop newly found systems, in addition, bio-inspired robots are used to solve scientific problems. For example, Daniel Goldman is a researcher that uses robots to study animals on a biological level, he is the investigator at CRAB Lab at the Georgia Institute of Technology. [1] In order to simplify locomotion, his team uses robots to explore and test hypothesis of real animals such as snakes, ants, and other small terrain-oriented species. [2]

Problem Motivation

With the vast expansion in the robotics industry. This project has given the team the opportunity to ride this new wave of robotics and expanding upon it. The technology of flying robotic drones and industry, task specific, robotics is already available in abundance. The team is planning to design and implement a system that would combine both sides of the spectrum and provide a bio-inspired robotic drone to help accomplish specific tasks.

Problem Statement

The goal of the project was to combine and improve multiple areas of robotics to produce a bio-inspired robot capable of moving through smaller rough caves and area, inaccessible by humans, while providing proper feedback.

Firstly, the team chose to design a scorpion inspired robot. This design provided the robot with six separate legs for navigation across slightly rougher terrain while providing two arms with grippers to handle object

Secondly, the project had to establish and use a stable and fast network to communicate with the robot, giving it the capability to receive instructions from a user and provide visual feedback from a camera.

Provided this information, the project tasks included:

- Leg structures able to mimic a scorpion's legs
- Claws capable to gripping objects and mimicking scorpion claws
- Establish and use a stable and fast communication Network
- Provide a mobile application to send instructions and receive visual feedback

With the main problem statements described. The next few sections of the document will describe the solution to accomplish said tasks.

Proposed Solution

To provide scorpion like limbs and movement, servo motors were used to provide degree specific movements over a 180-degree range. The structure provided stability of joints to reduce unwanted movements and slips.

We proposed establishing a built-in network instead of relying on an external network router for fast UDP communication by modifying the RPi to be an access point. We designed the RPi to be the brain of operations by communicating with all sub systems and computing data.

Finally, an android mobile application was implements to provide the user with the ability to control the robot and receive visual feedback from a camera on the robot.

Accomplishments

The design of the project was separated into subsystems and subtasks to ensure modularity and proper results. The goals accomplished in the duration of the project include:

- The structure of the chassis, legs, claws, and tail were designed to mimic a scorpion's shape while giving enough space for all legs and components to rest on it and function properly
- The RPi 3 was successfully configured to act as an access point to provide a network and become the main communication hub for all sub systems.
- The microcontroller interface was expanded by a PWM driver to support multiple servos for all the joints. It is also capable of applying all instructions and returning proper feedback.
- A visual Feedback was successfully implemented to the user using a camera on the robot.
- A mobile application was successfully designed and implemented to act as the user's UI to send instructions and access the camera's visual feedback.

Provided a brief list of the team's accomplishments towards the project, the following sections will provide a detailed explanation of the design, implementation, and development of all the project's subsystems and how it all works together to become SERQET. Furthermore, it details the challenges faced to achieve the overall goal.

Overview of Remainder of Report

The remainder of the report is structured as follows. The next section describes the technical solutions used to accomplish the goals listed in the above section. The subsections describe in detail all the tasks and solutions implemented to accomplish each subsystem and the overall project.

The following sections will describe the health and safety precautions taken by the team during the duration of the project to ensure no accident or injuries of any members. This is followed by details of the team's compliance with the code of ethics and the professionalism of the work done by the team.

This section is followed by a description of our project management plan, detailing how the project was structured and divided upon the team members. It includes the individual contributions of each team members and their suitability with their program and course work.

Finally, the document provides a section with our closing thoughts, recommendations for further development, and conclusion of the project.

Technical Solutions

Multiple components work together to create SERQET. This section will outline the technical design of creating SERQET. First creating the structure of the scorpion, base, legs, claws, tail, was an essential for this project. Then the motors and the movement of SERQET was implemented to imitate a real-life scorpion. Further on, creating an access point for the project, and a server to create the Raspberry pi communication hub and brain of operation. Lastly, the Android app was created to control SERQET's movement and to see the robot's point of view. Further explanation will occur in this section of all the components and how they all worked together to create SERQET.

System Architecture

1. Project System Overview

The robot was made up of multiple components, it was heavily focused on the body and structure of the animal while the hardware and software were used to design the scorpions' movements and functionalities. The robotic scorpion has six legs, equally distributed on each side, two frontal claws, a long tail and finally a double layered body to hold and manage these parts.

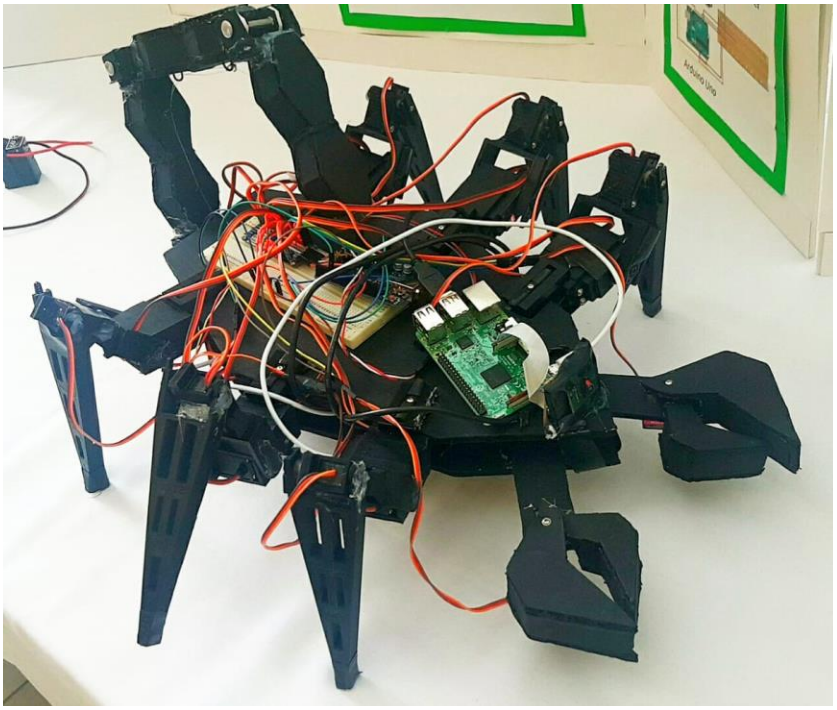

Figure 1: Project final product

The robot consists of multiple separate components, mainly a Raspberry Pi 3, a microcontroller and an android application developed using Android Studio. The Raspberry Pi is the central component of the robot, it was configured to work as the access point and brain of operations. A microcontroller is used to act as the scorpion's spinal cord, it controls the joints and movements of the robot. The android app served as a wireless controller, and a way to view the live video feed of the raspberry pi camera feedback.

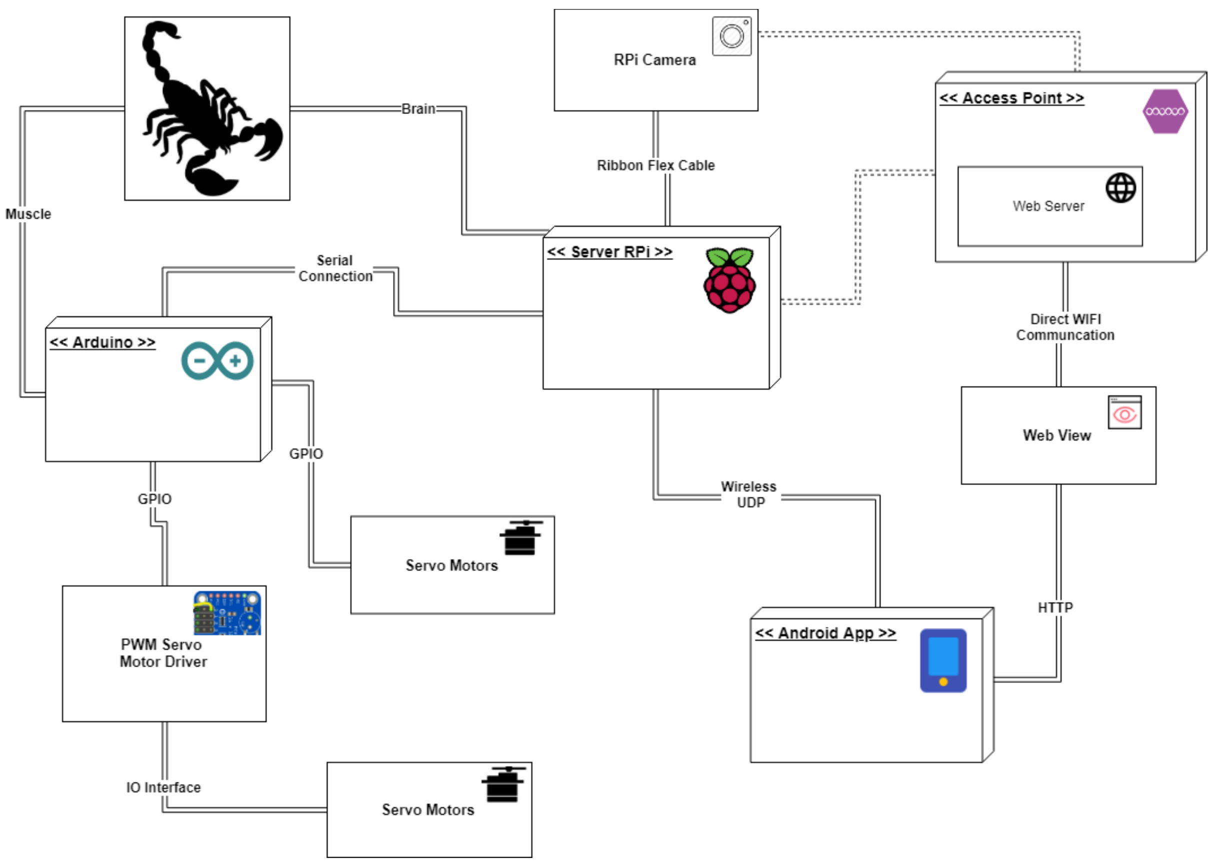

Figure 2: Full System Architecture Diagram

The microcontroller and android application are both connected to the main access point, via serial connection and wireless UDP respectively. In order to fully operate each individual joint of the scorpion, a PCA 9685 16-Channel Servo Motor Driver was connected to the microcontroller, it operated 16 of the servo motors which left the final 5 servo motors as digital pins on the microcontroller. The web view was implemented on the android application allowing the Raspberry Pi camera to provide live feed, and it was directly communicating with the access point via Wi-Fi.

2. Structural Design

One of the main motivations for the structure of the robot was that of a scorpion. Scorpions are commonly known for their unique skeleton and structure. They have three main body parts, the head, the central abdomen and lastly, the tail which is known as the metasoma. The head components contain the pedipalps and everything else that makes up a head in most animals. A scorpion's abdomen makes up its important organs and its eight legs which are clawed to help their maneuverability. The tail is segmented and is one of the distinguishable components of the animal, along with its telson at the very end, which may or may not be venomous depending on the species. [3]

The project structure was an assembly of the body, legs, claws and tail. Each one of their structures will be described in the following section.

2-1. Base Structure

The body of a scorpion can be described as long and thin, with the assembly of 4 legs angled towards its body, this inspired the design for the base as seen in figure 3. As mentioned previously in the project overview, the robot will have 3 legs on each side for a total of 6 legs.

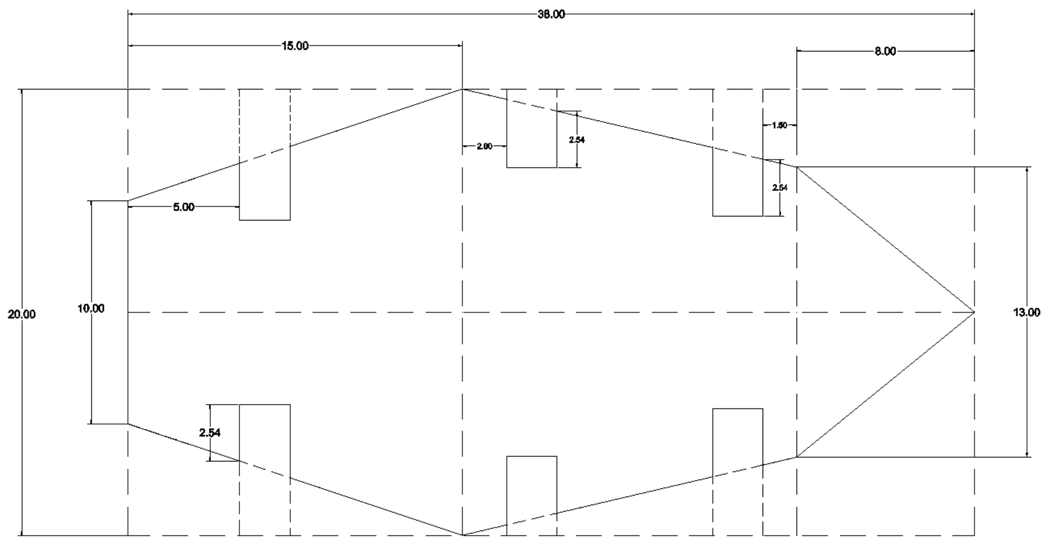

Figure 3: Base Structure

The body of the bio-robot was 38 cm by 20 cm, this wide space allowed for the hardware to comfortably sit tight, while also allowing room for the initial base joint which is described in section 2.2. The distributed strips around the body made room for the motor wings to fit and hold the base. It was double layered, with 3.5cm spacing to fit the battery packs on the bottom layer and the top would contain all the hardware components. This allowed for an organized layout and assembly; the top was easily accessible in the case of re-circuiting and maintenance. The top layer as shown in Figure 3 was designed to work as a cover, it was not permanently attached in place. This allowed for easy accessing the bottom layer and debugging the components if needed.

The body of the robot was made out of Styrofoam sheets with cardboard on both faces and 3 layers of acrylics paint to harden the sheets. This material was cheap and light, and so it was used to compensate for the weight of the sum of individual limbs. They resulted in a sturdy and easy to work with material, the shapes were easily cut out using a crafting knife.

2-2. Leg Structure and Components

The legs of the robot were one of the most important features, as one of the goals of the project was to successfully implement the animal's movement. The legs consist of 4 main components, the base bracket, shoulder bracket, elbow bracket and foot. They were designed on AutoCAD and 3D printed using PLA plastic filament. The use of 3d printing for the legs was to increase stability and to reinforce the motors, allowing for a steady and smooth motion.

Firstly, the base bracket in Figure 4 was designed to cage the servo motor inside of it, it fit the exact measurements of the motors and left rooms for screw reinforcements. Caging the motors into the brackets was necessary to avoid any motor jitter and big loss of energy. This bracket sat on the ends of the bottom body layer. It was screwed in tightly using 4-40x1/2-inch machine screws and reinforced with their respective nuts.

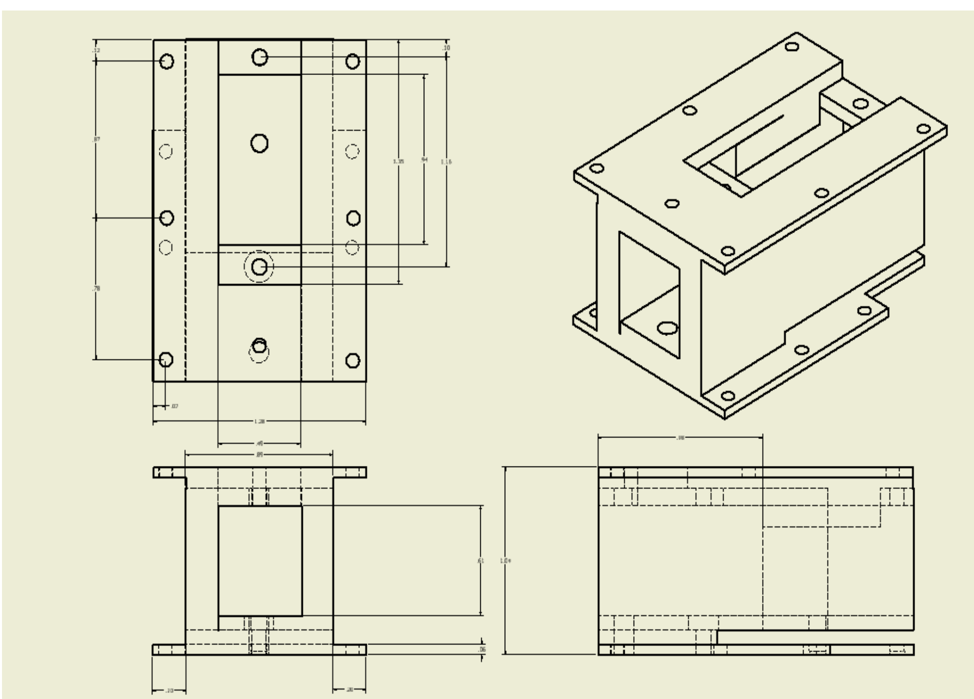

Figure 4: Base Bracket Technical drawing

The shoulder bracket was designed to both case the second motor and act as the wing of the motor on the base bracket. The respective design is illustrated in figure 5, it contains a semi-circular bottom which will slide into the bottom of the base bracket to further reinforce the parts by hooking it to both ends of the joint.

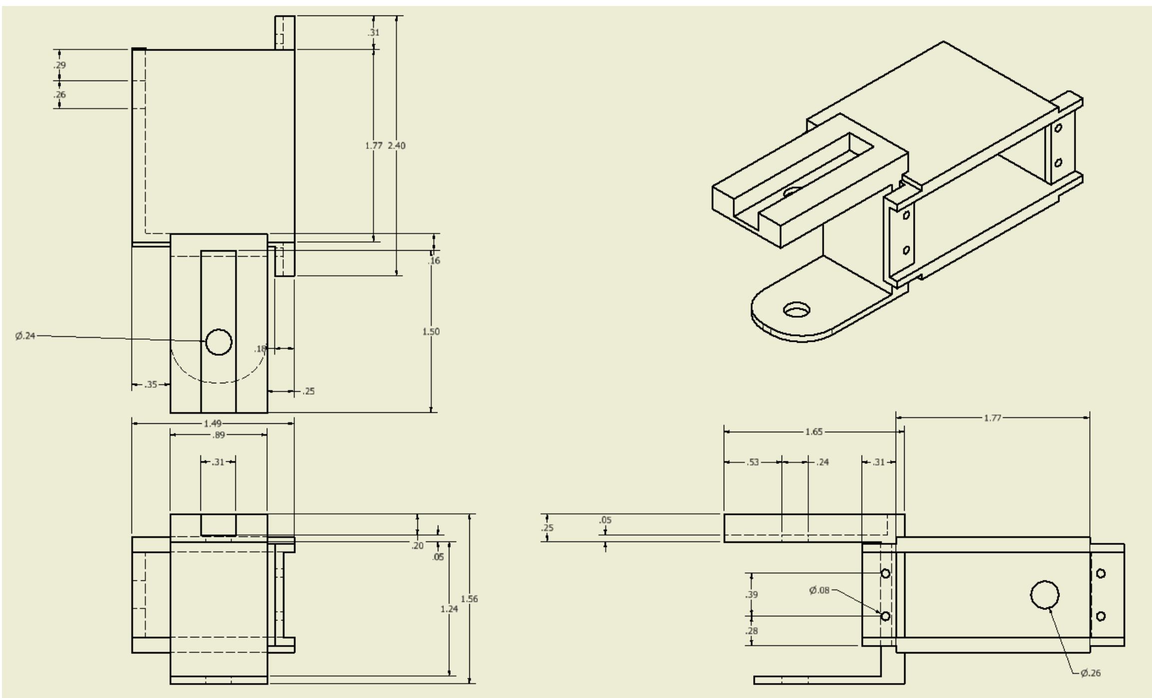

Figure 5: Shoulder Bracket Technical Drawing

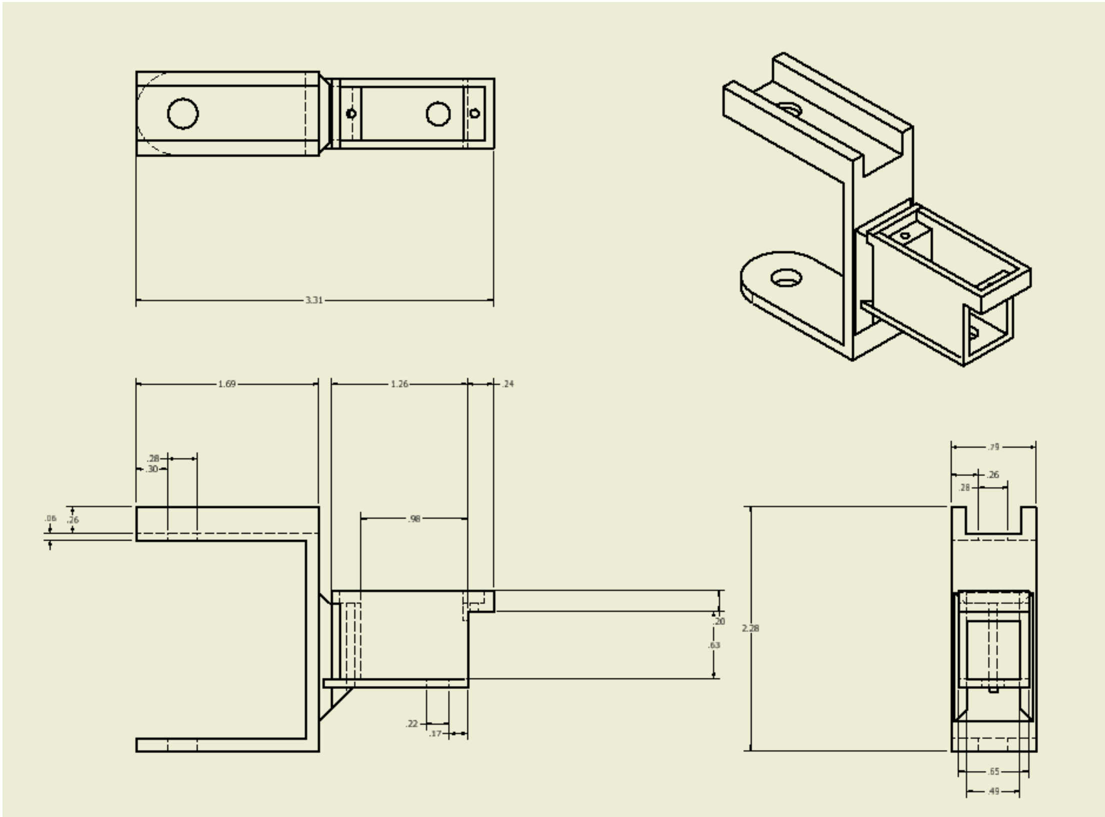

Figure 6: Elbow Bracket Technical Drawing

The last leg part that caged the final motor was the elbow bracket as shown in figure 6. This piece was similar to the previous. It acted as a motor cage and a wing for the last previous motor which would then be connected to the final leg component in figure 7.

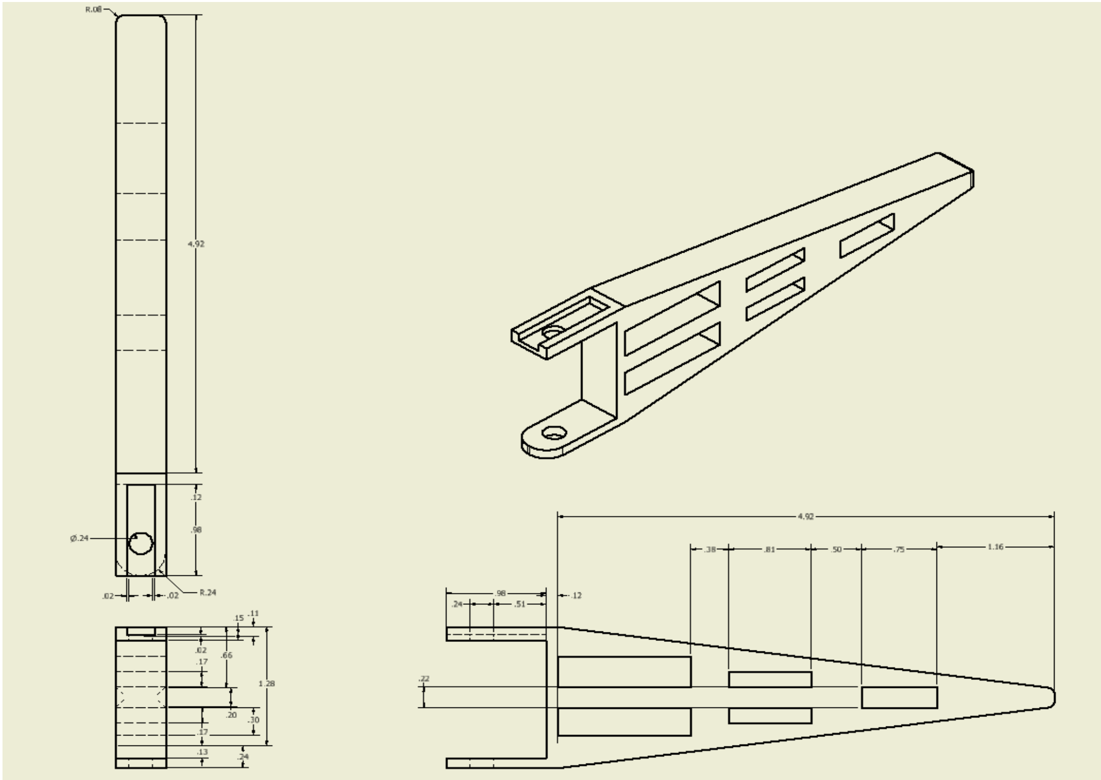

Figure 7: Foot Technical Drawing

The sizes of the designs worked with respect to the body, the weight of the legs kept the robot stable and firm. At the bottom of the final piece, a friction material was glued to help support it on its limbs and prevent slipping.

2-3. Claw Structure

A scorpion's claw is similar to many animals that have clawing mechanism. There are two sides of the clamp, of opposite sizes. The idea was to design a standard clamp and then scale it to be smaller and rotated. The smaller component as shown in figure 8 was fixed to the arm while the bigger clamp was attached to the wing of the motor. The motor was placed in the rectangular section of the arm and it would allow the bigger clamp to move forward and backward. Finally, to attach the claws to the chassis, the rectangular section was screwed into the front of the robot.

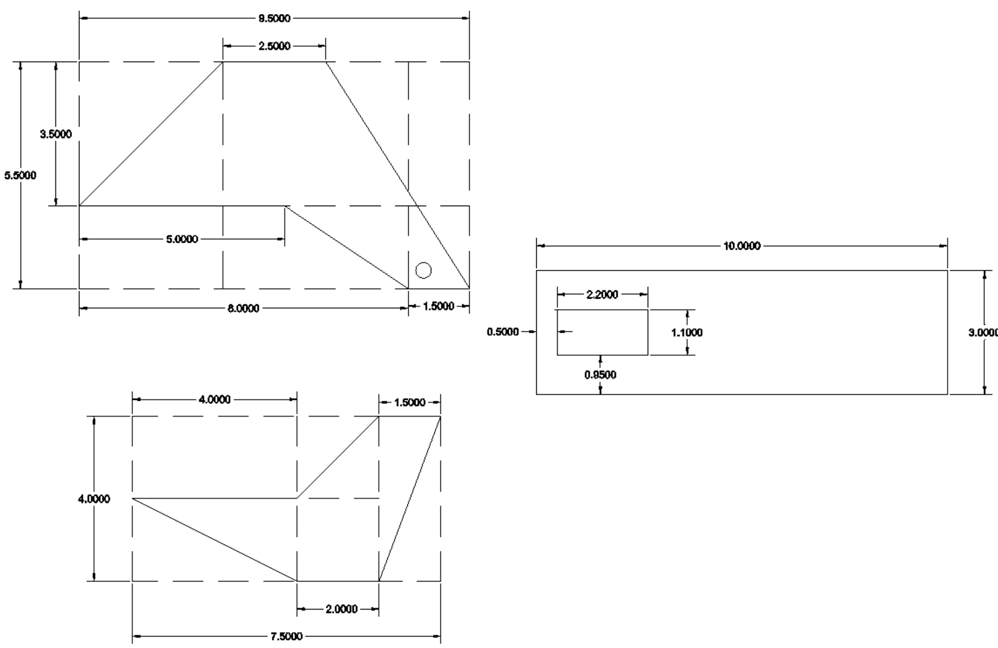

Figure 8: Claw Technical Drawing

The claws were made using the same materials as the base parts, although they were layered and made to be 3cm of thickness. The thickness supported a more realistic view of a claw and again, where reinforced and hardened using acrylic paint. Similarly, the tail was also designed the same way.

2-4. Tail Structure

Uniquely, scorpions have a segmented structure for tail, they consist of 5 segments and a final telson which is their splinter [3]. The design for the tail was simple to make 6 hexagons as shown in Figure 9 with a thickness of 3cm. Those hexagons were then glued together and hinged by Chicago Screws for every two pieces. This gave the tail the ability to move while maintaining the typical arched form. [3]

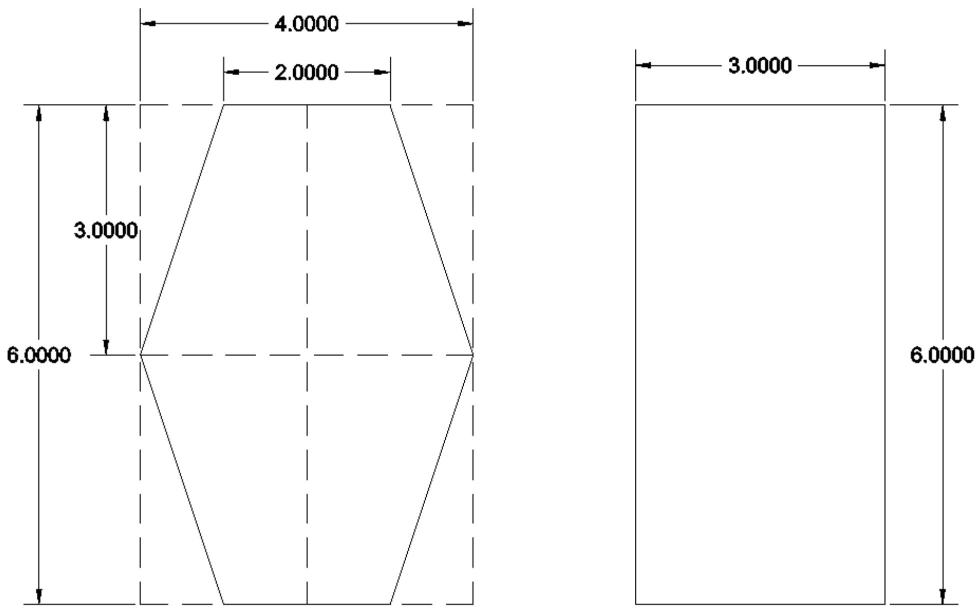

Figure 9: Tail Piece Technical Drawing

3. Motors and Joints

The structure of SERQET was to provide stable joints using motors while preventing any slips or wiggling.

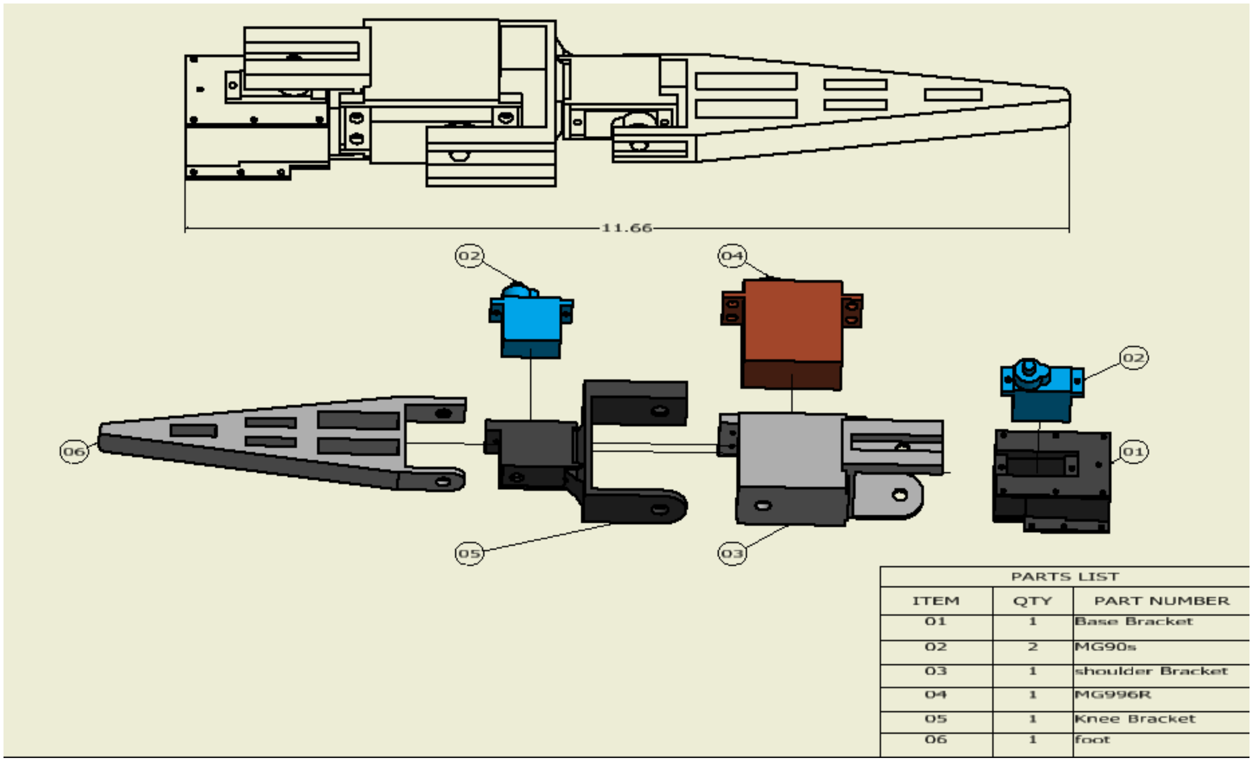

Figure 10: Overview of Leg Structure Assembly

Multiple, different, motors were used to establish all three leg joints with proper constraints, torques, and power to provide stable motion of all joints.

3-1. Base Joint

The motor, item 2, at the base joint, item 1, of the robot gave the legs the ability to move forwards and backwards relative to the main body of SERQET. An MG90s Servo motor, with a torque of 2 kg-cm at 6 volts and a 180-degree turning range, was used in this case as it does carry much weight. [4] The only stress applied on it is maintained by the structure of the base piece itself.

This joint was constrained by the presence of the main chassis of the robot on either side of the joint. The placement of the joint gave the joint enough room turn up to 120 degrees from 0, any further and the joint would collide with the chassis.

3-2. Shoulder Joint

For the Shoulder Joint, items 3 and 4, an MG996R servo motor, with a torque of 11.6 kg-cm at 6 Volts and 180 degrees in turning range, was used in this joint to maintain the high stress level. This joint is the main joint carrying most of the robot's weight. [5]

This joint was not constrained, due to the structure of the robot's joint piece, items 3 and 5, the servo was free to move from 0 degrees, pointing straight down, to 180 degrees, pointing straight up.

3-3. Elbow Joint

The Elbow joint, items 2 and 5, did not carry much stress other than having the shoulder joint rest on it. As a result, an MG90s was more than sufficient to carry the little amount of weight passed from the shoulder joint. [4]

Again, due to the design of the limbs and joints, the joint had no constraints and has a turning range of the full 180 degrees the servo provides.

3-4. Claw Joint

As previously mention in section 2.3, the claw contained a single motor which was used was used to move the bigger claw, it was attached to the wing of the motor. The smaller claw was fixed to the arm which simplified the design and mechanism of the claw.

3-5. Tail Joint

The tail had 3 separate joints hinged together, along with a fishing line that went through the hoops screwed into the base of the hexagons. When pressure is applied to the fishing line from the motor, it will move forward and when that pressure is released, the line relaxes and pulls back.

4. RPi Functionality

A Raspberry Pi 3 was used as the center of command for our system. It is used to maintain all communication with other devices across the system while computing algorithms for other system functionality. The RPi 3 was powered using a power bank, hidden inside the chassis, which provided the 2.4 A at 5 Volts needed by the board to function properly. [6]

4-1. Brain of Operations

The RPi is considered the brain of SERQET as it receives all data of the joints and mechanisms of the system and provides the appropriate output.

The RPi would establish a network with the controlling android mobile application which provided SERQET with the movement instructions. The RPi would take in the input signal and compute algorithms to output specific joint movement data to pass to perform the desired movement. The RPI would also read and maintain the data. The RPi was also responsible of the "eyes" of SERQET by maintaining the camera feedback, storing the data, and render it accessible.

4-2. Main Communication Hub

While being the brain of operations, the RPi also acted as the main communication hub between all the systems.

Figure 11: System Communication Overview

Firstly, it maintained a proper connection between SEQET itself and the android mobile controlling application to receive simple movement instructions from the user while providing visual feedback. This connection used the UDP datagram protocols to construct the movement signals while the visual feedback relied on a simple http connection. UDP datagram communication was used based on the nature of the system as being an embedded system generally. The system relies more on speed than accuracy of data with a default fallback status in case of an error signal.

To protect the RPi from user spamming a button the server checks for the most recent UDP packet received, saves it, flushes the socket and then handles the saved UDP packet.

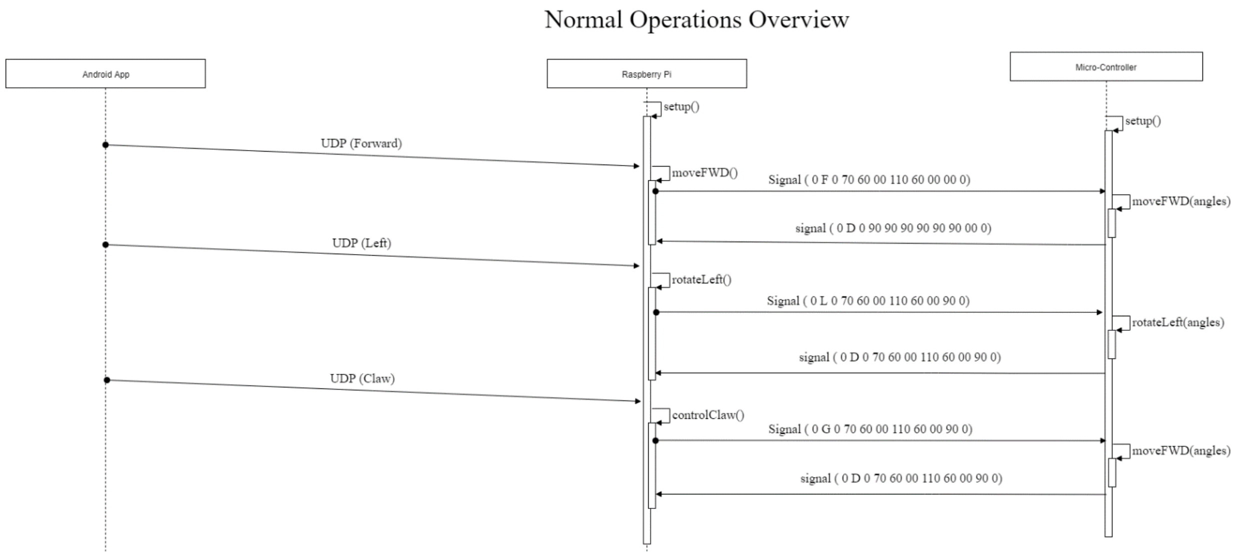

Figure 12: System Communication Sequence Diagram

Secondly, the RPi provided a direct serial connection to the microcontroller, "the spinal cord", to provide specific joint and leg instructions to perform based on the output it computes.

5. Wireless Communication and Access Point

Discussed in this section are the RPi Wi-Fi network standards, establishment of the access point, and configuring a proper network for system communication with external controlling device.

In the project, an RPi 3 is used as a standalone access point for the system to be able to communicate with the controlling mobile application wirelessly and provide proper live feedback to the user. By transforming the RPi 3 into a Wi-Fi access point, other devices are able to connect to SERQET directly and transmit control signals and feedback data between each other without relying on third party servers, routers, or switches to transmit the signal between the user and the robot.

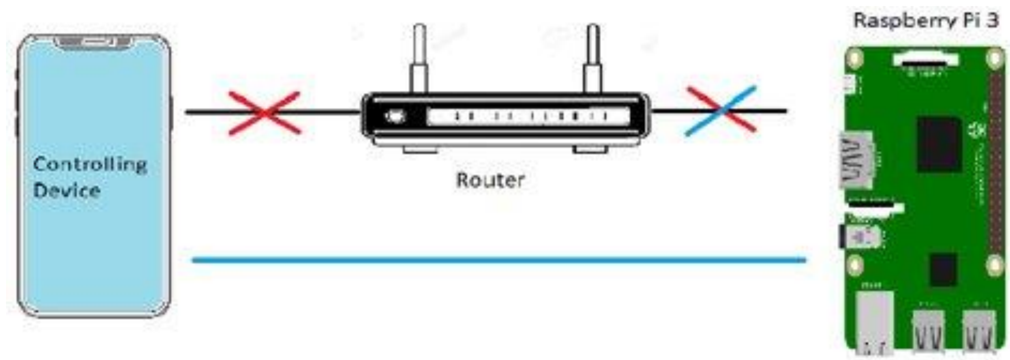

Figure 13: Wireless Communication Goal

By configuring the Linux OS on the RPi 3, the project cost was reduced significantly by removing the extra hardware needed to accomplish the wireless network.

5-1. RPi WiFi Standards

The RPi 3 comes with built in Wi-Fi capabilities already on the board and is able to connect to Wi-Fi access points just like most mobile devices and computers.

With a Linux OS running on the RPi 3, configuring the access point and allowing it to broadcast itself like a router and manage itself as a host and the connected devices to establish a stable network was efficient and easy. [7]

The RPi 3 wifi network capabilities run at 2.4 GHz and 802.11n wireless LAN, which provides a transmission speed of 450 Mbps. [8]

5-2. HostAPD

HostAPD, which stands for Host Access Point Daemon, is the software used to configure the regular RPi 3 Wi-Fi interface into a proper access point and provide host management and Authentication protocols.

Configuring HostAPD parameters allowed us to provide the various details needed as the host such as the Wi-Fi ID, encryption type, network type, Network Password. After configuring the HostAPD parameters, we tell the system where these parameters are by referencing the file.

```
interface=wlan0
driver=nl80211
ssid=Sysc4907CU
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=4thYearProject
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Figure 14: HostAPD Configuration Parameters and Values [9]

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Figure 15: Referencing the HostAPD Configuration Files [9]

The system successfully created a network access point with networkID “Sysc4907CU”, passphrase “4thYearProject” and WPA encryption. The network has successfully been initialized with HostAPD.

5-3. Dnsmasq

DNSmasq is an easy lightweight software that provides the infrastructure to the network including router bootup, network advertisement, and Dynamic Host Configuration Protocol, dhcp. [10]

```
interface wlan0
static ip_address=192.168.0.10/24
```

Figure 16: Applying the Network Host IPv4 Address

The main network infrastructure was initialized by giving the network a static IP address to use as the host (IP = 192.168.0.10/24). This is the IP address the network will use as the router. [9]

```
interface=wlan0
usually wlan0
  dhcp-range=192.168.4.2,192.168.4.2,255.255.255.0,24h
```

Figure 17: Applying Range of IPv4 Addresses Used by Connected Devices

The wireless network interface was given a range of IP addresses to use for devices willing to connect to it. In this case, the range was configured to allow for only one device to connect at a time given they know the password to the network. [9]

This was accomplished by editing the main dnsmasq configuration file and adding a dhcp range parameter. With this, the network has been given a range of IPv4 address to be used by external devices connected to the network and it is also given a broadcast signal for data distribution among the system.

5-4. Access Point Configuration

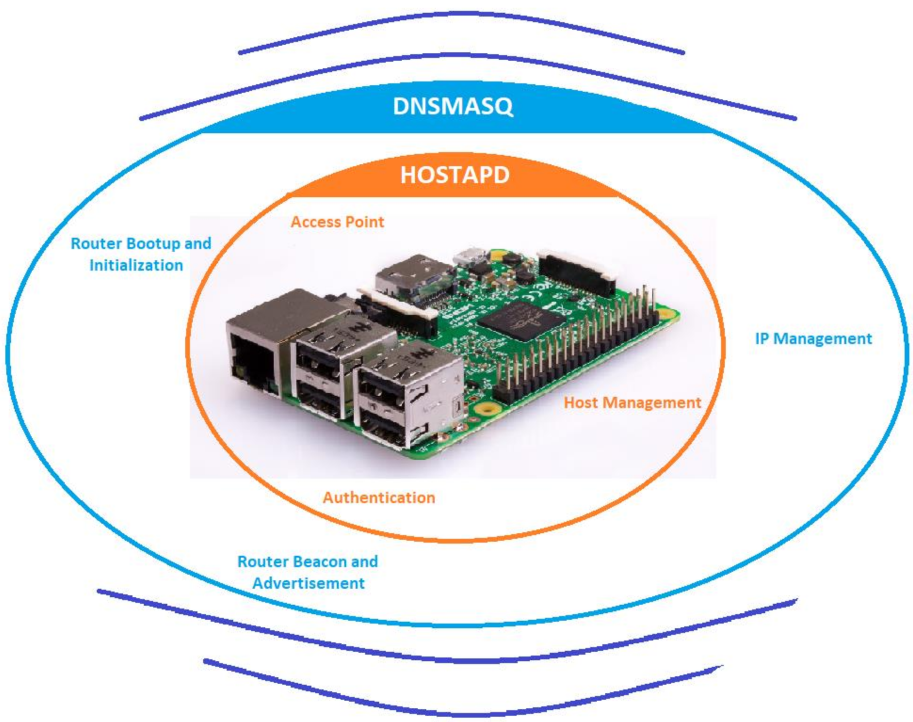

Figure 18: Wireless Network Overview

The figure above gives a simple overview of the wireless network successfully implemented in the system to cover wireless communication and feedback.

6. Live Video Feedback

SERQET was designed to provide the user with a live video feed from the robot itself. Using a camera as the “eyes” of the robot, and broadcasting its input.

6-1. Camera Components

The RPi camera was used to capture the video feedback. The camera was connected to the RPi directly using a ribbon cable attached to the proper port on the RPi board. The camera provided a maximum of 1080p video resolution and 8MP stills.

Figure 19: RPi 3 Camera Module

By using the RPi3 camera instead of a generic web cam, the cost was slightly reduced. Furthermore, the camera was made to work directly with the RPi without having to download any external software or driver other than its API used by the RPi to control it. Finally, the RPi camera uses less power than other cameras. [11]

6-2. Video Feedback

The RPi camera is continuously turned on, taking video frames in mjpeg format and updating the video buffer containing the frames and pictures taken.

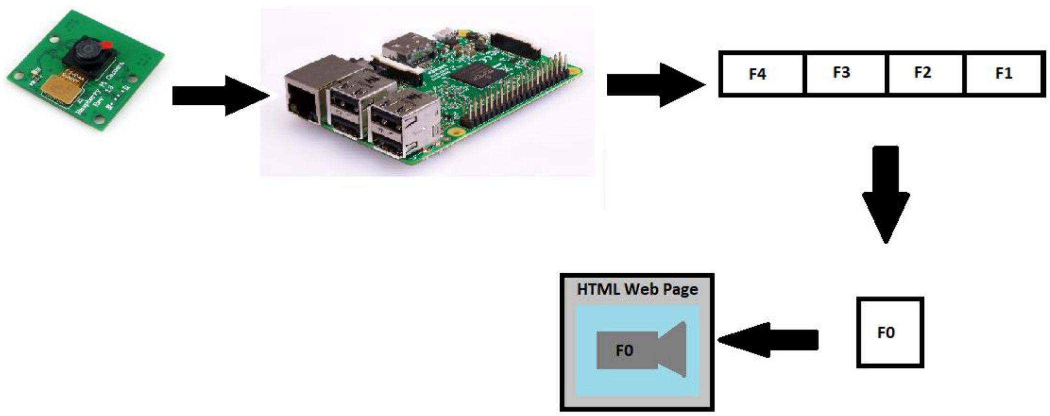

Figure 20: Overview of Live Video Feedback

A webpage was created dynamically using python code. The webpage would continuously retrieve frames from the video buffer and display it on the index webpage. The webpage is then broadcasted to the network on port 8000.

The live video feed suffered from high delay and lag due to overflow in the video buffer. However, to resolve the issue, the video buffer was regularly flushed to provide space for newer frames, reducing the delay.

This provided the system with a live video feed that can be accessed by connecting to the RPi's network and accessing the data output from the RPi's host IP at port 8000 through the mobile application.

7. Microcontroller and PWM Driver

7-1. Circuit Design and Serial Communication

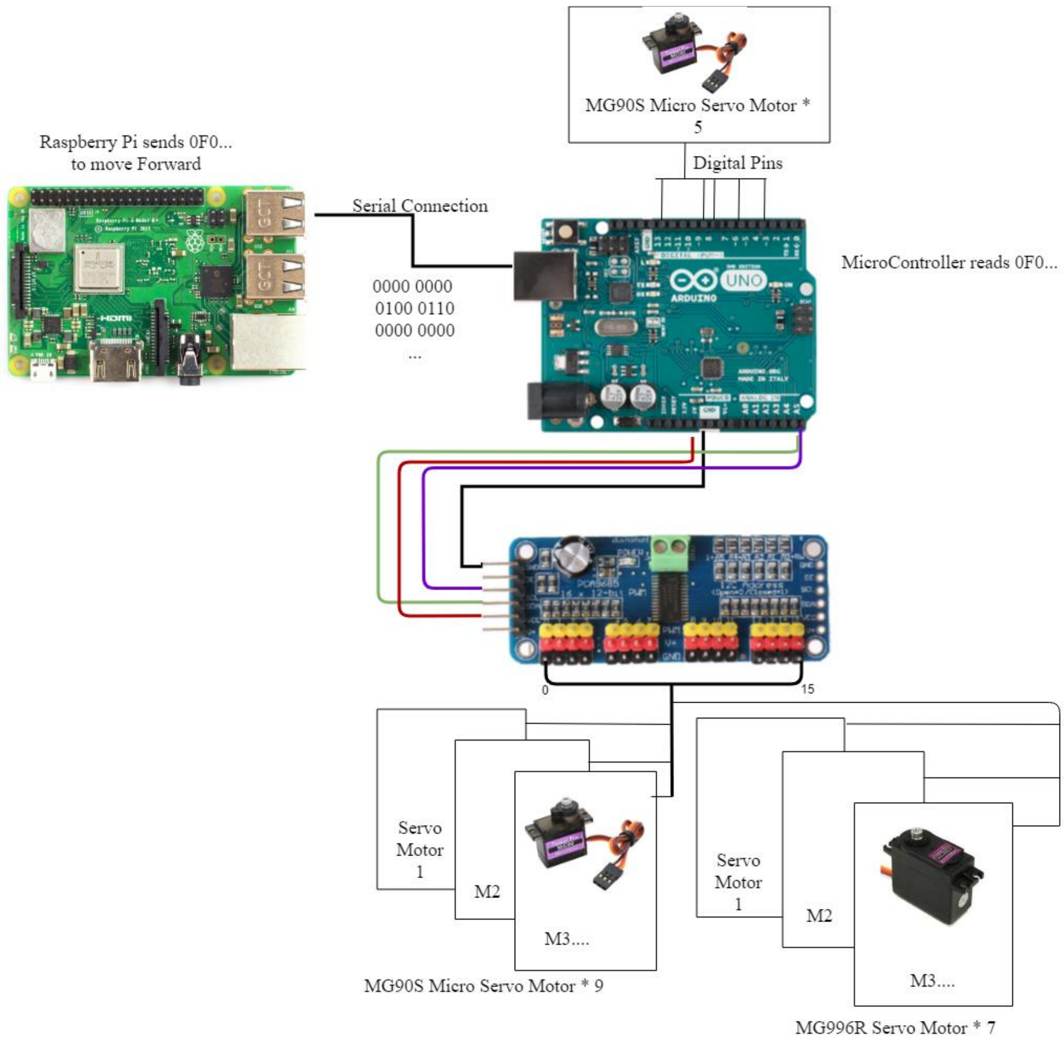

Figure 21: Circuit Diagram and Overview

As shown in the above figure, the communication between the RPi and the micro-controller is done through a USB cord. This establishes a Serial connection between the "brain" and the "spinal cord" of the system. To read and interpret the signals being sent from one another, the serial communication interface and its API calls were used as the basis.

7-2. Communication Protocol and signals

The following section will introduce, define and illustrate how the communication signals function.

Table 1: Variable Representation and Values

Command/Degree Bytes	Definition
F	Scorpion will be moving forward
B	Scorpion will be moving backward
L	Scorpion will be turning left
R	Scorpion will be turning right
D	Done moving return signal
G	Control claw gripper
X, Y, Z, A, B, C, D	Final servo angles positions returned
x, y, z, a, b, c, d	Servo angles to move to
0	Reset to default

Table 2: Control and Feedback Signals

Signal Sent Content examples	Action
0 F 0 xx yy zz aa bb cc 00 0	Move scorpion forward
0 B 0 xx yy zz aa bb cc 00 0	Move scorpion backward
0 L 0 xx yy zz aa bb cc 00 0	Turn the scorpion left
0 R 0 xx yy zz aa bb cc 00 0	Turn the scorpion right
0 G 0 00 00 00 00 00 00 dd 0	Open/Close Claws
0 D 0 XX YY ZZ AA BB CC DD 0	ACK
0 0 0 00 00 00 00 00 00 00 0	Default standing position

The scorpion's movement is handled by signals sent between the micro-controller and the RPi. The signal's format is show in table 2 and examples are shown in table 3 and 4, and sequence diagrams 22, 23, and 24. The signals, as shown in table 2 are 18 bytes long, 3 bytes are reserved for delimiters, 1 byte for the movement command and 14 bytes for angles.

The micro-controller is constantly waiting for a signal from the RPi by monitoring the "serial connection" and setting a flag when data is available to be read. The RPi sends a string which is decomposed into an array of characters. Characters are then encoded into binary using ASCII, then each character is sent one at a time through the serial USB cord. The micro-controller will detect the signal and read it one byte at a time and decode it using ACII to characters.

Table 3: Signal Format Sent by RPi to Micro-controller

1 byte	1 byte	1 byte	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	1 byte
0	F	0	xx	yy	zz	aa	bb	cc	dd	0

Table 4: Signal Format Sent as Feedback to RPi from Micro-controller

1 byte	1 byte	1 byte	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	2 bytes	1 byte
0	D	0	XX	YY	ZZ	AA	BB	CC	DD	0

Test cases are provided, as shown in the sequence diagram in figure 22, during normal operations, the signal sent contains the movement command as well as angles for both left and right-side base, shoulder and elbow servos. Once the movement is done, an acknowledgement signal is sent with the servo's new angle positions.

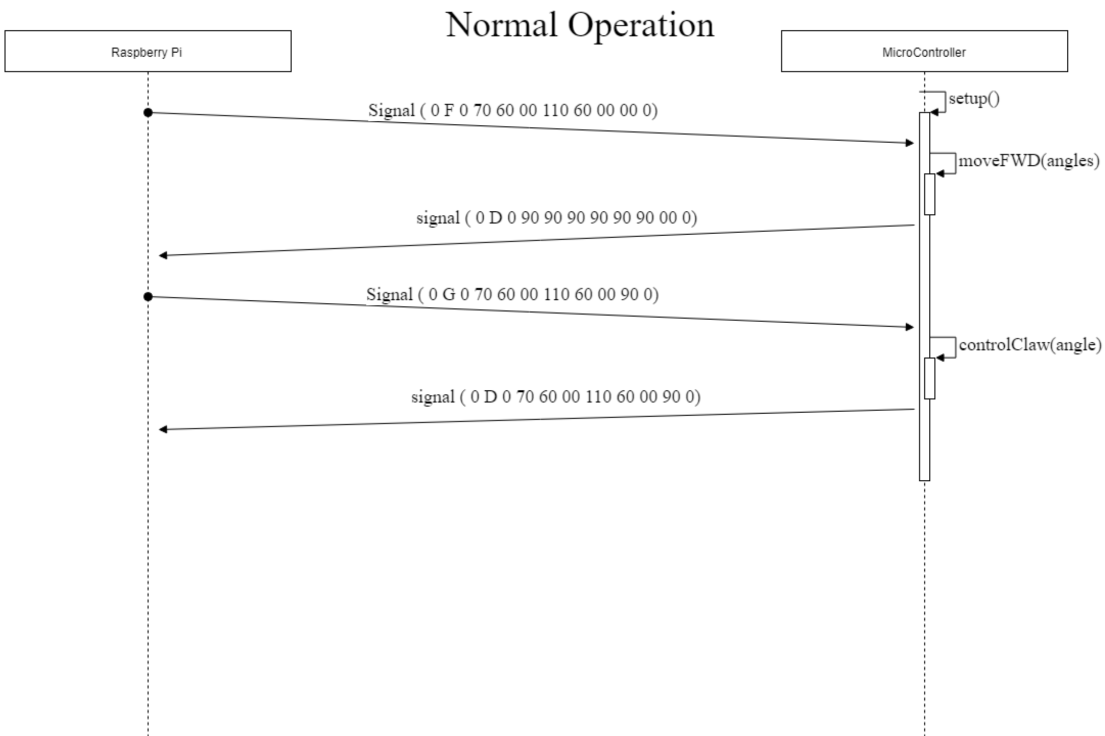

Figure 22: Sequence Diagram Representing Normal Communication

As shown in sequence diagram 23, if a signal contains the wrong movement command, or an invalid angle value or does not follow the predefined format set for signals, the micro-controller is going to reset the servos' angles to standing position and send a signal back to the RPi.

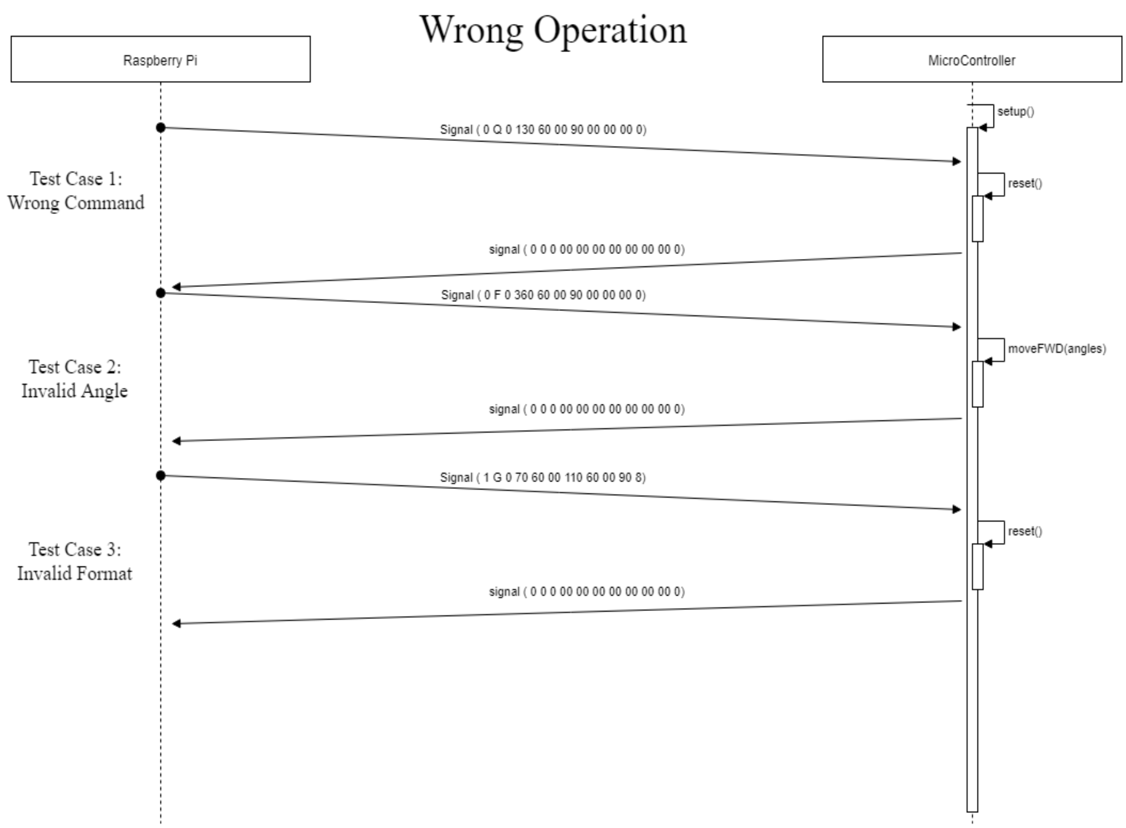

Figure 23: Sequence Diagram Representing Communication with Unknown/Wrong Values

The final sequence diagram, in figure 24, includes a test case for timeouts, if the RPi does not receive an acknowledgment and timeout occurs, the signal is not rebroadcasted, if the following signal does not receive an acknowledgment and timeout then the RPi will reset the connection. During the reset, if the RPi fails to detect the micro-controller, then the micro-controller either not receiving power or the cord linking the RPi to the micro-controller is no longer working. The user would have to apply some maintenance.

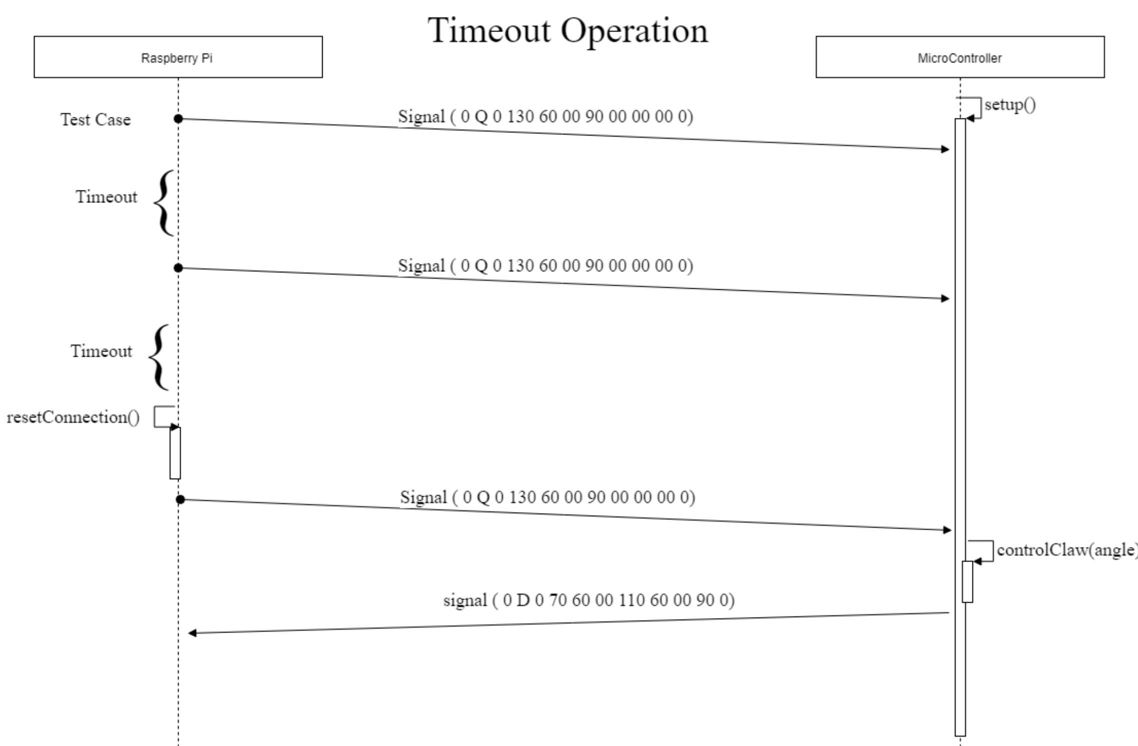

Figure 24: Sequence Diagram Representing Communication in Case of Signal Timeout

7-3. PWM Driver

Figure 21 shows an overview of the connections between the PWM and microcontroller. The PWM 16-bit channel was connected to the microcontroller by their ground pins, analog pin 5 in the microcontroller was connected to SCL in the PWM Servo driver, analog pin 4 was connected to SDA, and the 5v on the microcontroller is connected to the VCC of the servo driver. As a result, this configuration allowed the microcontroller and the 16-bit servo driver to work together and move the motors.

Each channel in the PWM Servo Driver had the servo connect to the ground, the power, and the signal. The PWM Signal distribution has low delay which allows the signal to send the angle to the motor right away and a significantly fast. The control signal in the PWM Driver allowed the motor to move to specific angles to allow our project, “SERQET”, to move forward, backward, left, to open and close the claws, and to move the tail.

To program and use the 16-bit channel PWM Servo Driver the project needed to include the Adafruit library. Adafruit library has many functions that are useful for this project for example the PWM Servo Driver works with pulse length and with Adafruit it can convert degree to pulse length and vice versa.

The PWM Servo Driver uses analog servo motors therefore, the project uses Adafruit. Adafruit only uses pulse length therefore, a function was created to convert degree to pulse length. This function allowed the project to use degrees and then convert it into pulse length to give us an exact angle. In the code the frequency for PWM was set to a constant of 60 Hz which allows the analog servo to run at 60 Hz, and this determined how many full pulses are generated. [12]

7-4. Signal and Power Distribution

The PWM Servo Driver is powered by the Micro-Controller. The servo motors connected to the driver are powered by an external 2200 mah 7.4-9.8 volts NiMH battery pack. The servo driver is an i2c controlled driver that has a built-in clock of up to 1.6 KHz [13].

This model was used because it allows for the microcontroller to run other tasks and not continuously sending signals to the servo controller while allowing us to demultiplex the servo control signal from a couple of pins to the 16 outputs. It has a 12-bit resolution for each output allowing the servos to update at a rate of 60Hz. The driver has a 5V compliant, which allowed the micro-controller to power it safely. There is a terminal block for power input which is connected to the battery pack and used to power the 16 servo motors on the PWM driver. [13]

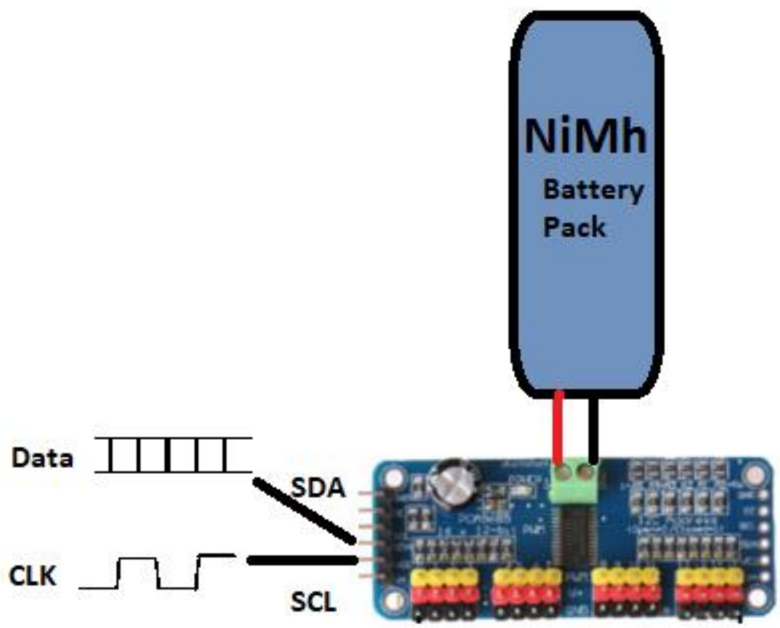

Figure 25: PWM Power and Signal Distribution Overview

The two lines to send and receive data are from the I2C protocol as shown in figure 25 above. [14] The control pins on the driver are the SCL which is an I2C clock pin, and SDA which is I2C data pin, this allows data to be sent between the two devices. [15] As the clock line changes from low too high a single bit of information is transferred from the board to the PWM driver over the SDA line using serial I2C protocol. After this information is sent the device executes the request, and in the project's case it will transmit new data back as feedback. [14]

7-5. Joint Control and Feedback Loop

A two-dimensional 5 by 3 array was created because there are 3 motors in each leg which means that there are 3 channels being used for one leg. The 2-dimensional array used the 16 channels on the PWM servo driver, as shown below:

- Leg 1 used: Channel 0,1,2
- Leg 2: Channel 3,4,5
- Leg 3: Channel 6,7,8
- Leg 4: Channel 9,10,11
- Leg 5: Channel 12,13,14
- Leg 6 used: Channel 15 and the rest were digital pins from the microcontroller

There is an additional one array created for the digital pins, which are 2, 4, 6, 8, and 13. For leg 6 it used the digital pins 4, and 13. The left gripper in on pin 8, and the right gripper is pin 2. Lastly, the tail used digital pin 6.

Through a feedback loop, the RPi is able to store in an array the values of the right and left side base, shoulder, and elbow servo angle positions. Doing so, the RPi will have better control over the servo rotational speed and its position. It handles all the angle calculations since it has more processing power. Through this the raspberry pi can control and apply the constraints needed for leg movements. Our system is closed loop in which the Raspberry Pi is the control system, the feedback system allows to maintain and adjust positions according to constraints. As a result, the Scorpion has better coordination between movements.

7-6. Movement Gait

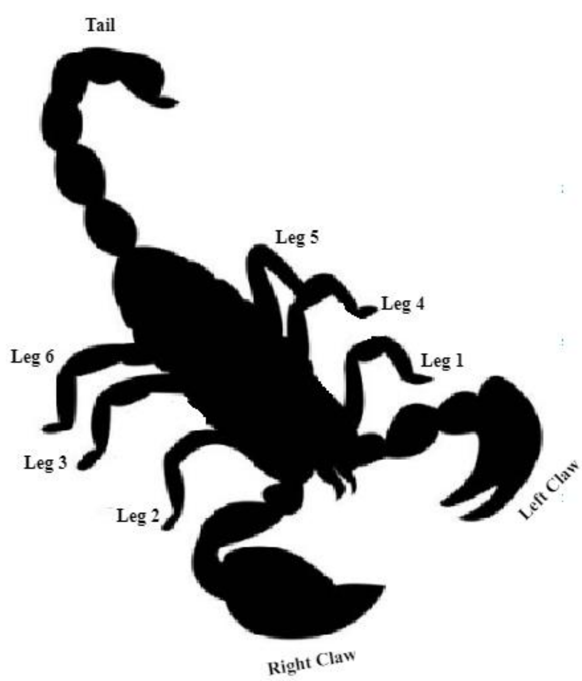

Figure 26: SERQET Limbs Definition [16]

The Scorpion's body movement consists of smaller leg movements. Each movement has its own leg movement sequence. The forward sequence moves leg 3, and 4 forward first for better balance as most of the weight is forward. Legs 1 and 2 are then moved forward then following legs 6, and 5. Finally all the legs rotate back into the initial position giving the Scorpion a push forward.

For the backward movement, legs 3, and 4 moves backwards for balancing purposes. Followed by legs 6 and 5, then legs 1 and 2. That allows the Scorpion to get a push backwards when all servos return to initial position.

To allow the Scorpion to turn left legs 2, 3 and 6 are moved forward and the remaining legs move backwards. After the left side legs return to their initial angle that allows the scorpion to rotate to the left side. The same concept is used for turning right, however the sequence is done backwards with different angles.

Further on, the claw movement is handled by one servo for each claw. Through the feedback the RPi knows if the claw is open or closed, and so when a signal is sent it will contain the proper angle to either open or close. The tail's movement is based on a motor that pulls on a string and either retracts or relaxes the tail.

SERQET is able to mimic a scorpion's movement while moving forward, backwards and to the side. A scorpion has multiple different movement patterns and the one that has been chosen is the one that focuses on balance over speed. SERQET is able to pinch like a scorpion and move its tail similarly to what a real scorpion can.

8. Android Mobile Application

8-1. Main App Functionality

The main functionality of the android app is to control the scorpion's movement and display the live video feed that the RPi broadcasts. It is used as an external device to connect directly to SERQET as a replacement for a generic controller while providing the user with proper feedback. Nowadays, everyone carries a smartphone and carrying around a generic controller is inconvenient. Smartphones all contain a WI-FI module as well as have a screen. The generic control does not come with a screen and getting one would increase the budget and getting all the components to have the screen display the live video feed would increase the weight and the overall size of the controller, rendering it inefficient and inconvenient compared to a much thinner, smaller and lighter option.

8-2. Communication Protocol

For the communication protocol between the RPi and the android app, UDP was chosen due to the nature of the system as being generally an embedded system. Embedded systems are known to being time dependent and so using the faster option, UDP, was the top choice in SERQET’s case. The system, in this project, relied more on the speed of the data communicated between different devices more than the integrity of the data transmitted as every device has a default status to return to in case of an error and continue working normally.

The table shown below contains the UDP datagram content as well as how it will be interpreted by the RPi:

Table 5: UDP Signals and Definitions

Valid Commands	Interpretation
Forward	Move the scorpion forward (send forward signal to Micro-controller)
Backward	Move the scorpion backward (send backward signal to Micro-controller)
Left	Rotate the scorpion left (send rotate left signal to Micro-controller)
Right	Rotate the scorpion right (send rotate right signal to Micro-controller)
Claw	Open or close gripper (send gripper signal to Micro-controller)

Shown below are sequence diagrams representing test cases.

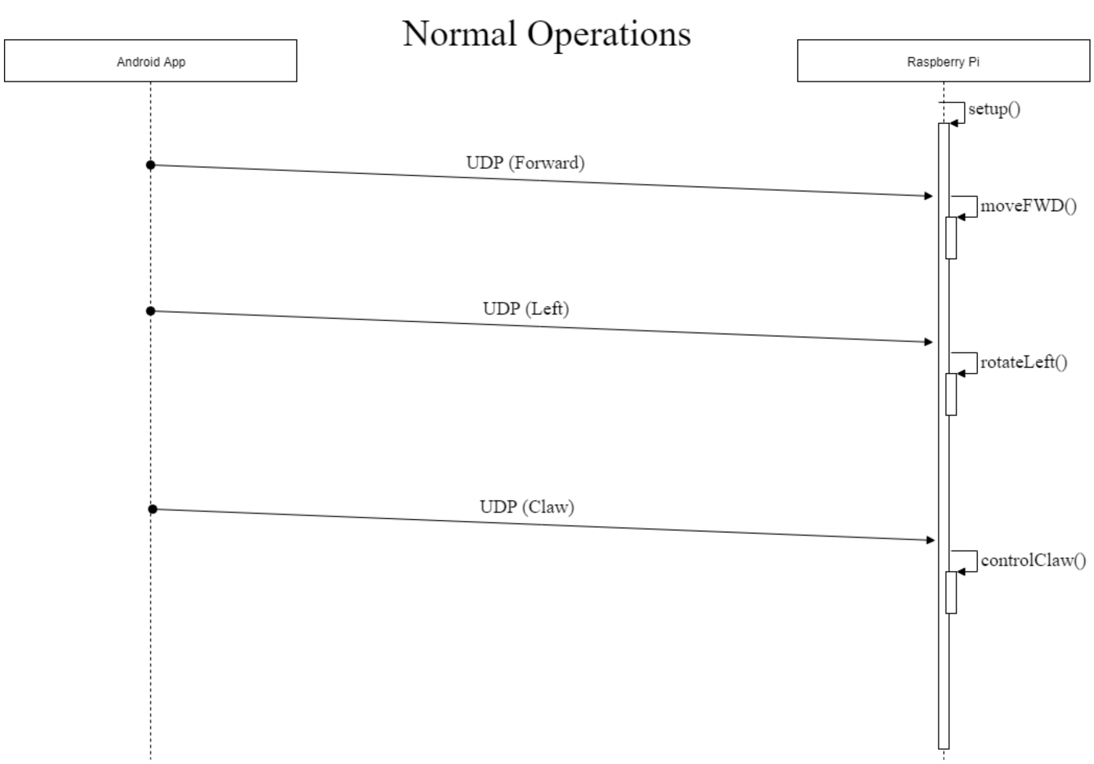

Figure 27: Wireless UDP - Normal Operation

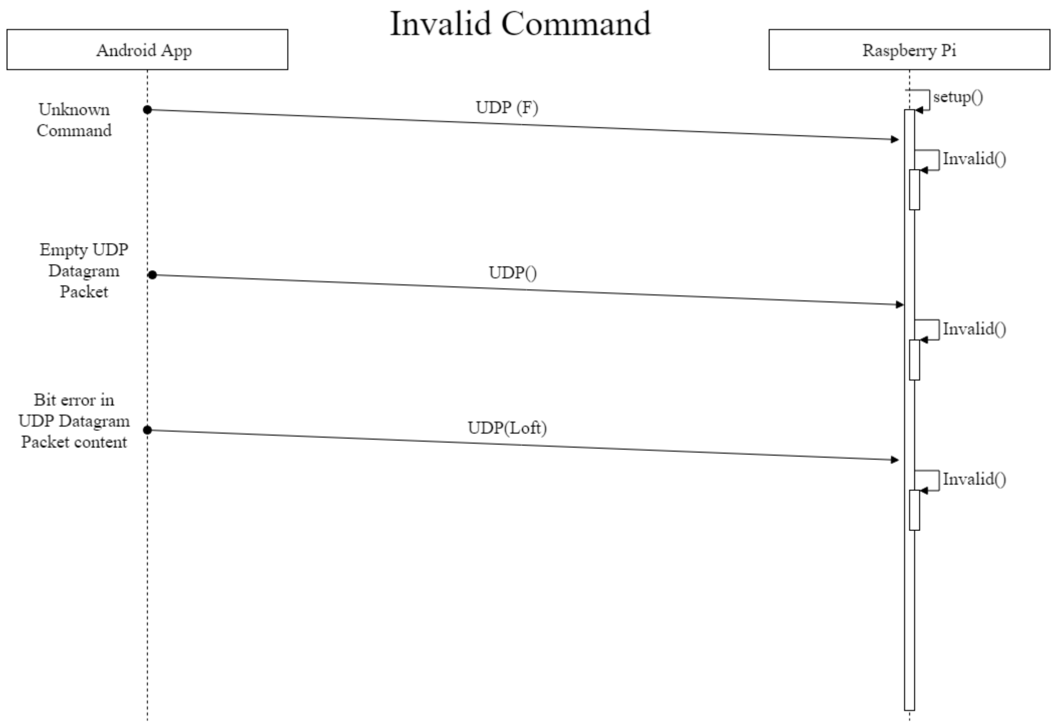

Figure 28: Wireless UDP - Invalid Command

As shown in sequence Figure 27 during normal operations a UDP datagram packet containing valid data is sent to the RPi. A corresponding signal is then sent to the micro-controller.

The android application does not rely on acknowledgments to know whether or not the connection is still on or if the scorpion is executing the movement selected by the user. It instead relies on a visual feedback, if the scorpion does not execute the buttons corresponding movement repeatedly then the connection is down and requires a manual reboot of the raspberry pi. If the video stream does not update, then the connection to the RPi is down and it will require a manual reboot of the RPi.

8-3. Controller Solution

Figure 29: Android Mobile Controller Application

The app is made of two parts, the WebView container and the buttons container. The WebView container displays the live video feed broadcasted by the RPi by loading the following URL: http://RPi_IP_Address:8000 where the RPi_IP_Adress is an integer containing the RPi's IP address and 8000 is the port ID where the video feed is being broadcasted. The buttons containers contain all of the buttons responsible for the scorpion's body movement as well as gripper control. Each button is linked to a different function that is triggered once the button is clicked. The function sends a UDP datagram packet containing data corresponding to the movement the user pressed.

Project Conclusion

1. Final Project Status

As a conclusion, all sub systems work together as detailed in the sections above which creates a biologically inspired robotic scorpion. The scorpion has an external controller as an android mobile application able to send instructions to the robot through UDP datagram packets and receive visual feedback through a camera set at the front of the scorpion. The scorpion successfully became the main brain of operations by providing a communication hub for all sub systems and computing all the data required by the project to function properly. The microcontroller is successfully able to read instructions and apply all joint movements accordingly and return updated information about the scorpion.

2. Future Considerations and Upgrades

2-1. Step size

While the raspberry pi takes in an instruction from the user, computes the joint positions and passes on the information to the microcontroller. A good consideration is to allow the user to manually set the step size of the joint movements.

2-2. Improve tail functionality

While the tail somewhat moves like an actual scorpion, a valid update was to attach the feedback camera to the tip to allow moving the camera angles for a better view.

2-3. Accommodate rough terrains

While the robot is somewhat able to move across uneven terrain. Adding force sensors on the tips of the legs could provide additional ability for the robot to move across rough terrain.

2-4. Mobile application

The developed mobile application controller was android based. To further improve this system. Additional applications for various OSs could be developed to allow different brand devices to connect and control the robot.

3. Closing Statement

The project goal was to design and implement a biologically inspired robot able to move across smaller places, unreachable by humans, and provide visual feedback. As a team we believe we have successfully combined the robotics designed for search and rescue and drones to create a rover like scorpion able to achieve its designed goal and provide visual feedback about areas inaccessible by humans. However, looking back, the team had plans and designs for various improvements and upgrades that could have been implemented given more time and budget.

Chapter 2: The Engineering Project

2.1 Health and Safety

The project consisted of both hardware and software components, and as a result there were multiple health and safety risks present to the team members. Most of the project work was done in System and Computer engineering Department's computer labs which follow Carleton University's health and safety policy. A lot of hardware safety risks included soldering, power management, circuitry and cutting out the body panels of the robot.

Throughout the project, soldering was done in the provided labs: the advanced electronic lab, ME 4135 and the Technical Support Staff room, ME 4224. The labs were equipped with the proper safety equipment such as the safety goggles and the fume extractor to protect against solder fumes and additional help from professionals was provided to ensure the teams safety. Additionally, team members made sure to have proper attire being no lose clothing nor any lose or untied hair which may pose as safety risk.

In terms of circuitry, the team kept their workspace clean of electrical hazards such as drinks and/or any liquids near the hardware components and power supplies. The environment was grounded, and any form of power supply was never left running, it was always unplugged before making changes to the circuits. Finally, cutting out the body panels of the robot required the use of professional crafting blades, this was done carefully, and team members made sure not to cause any distractions to the one cutting out the panels.

There were not many software health and safety risks present in the project, however long hours on the computers required the team members to take breaks to move their body and constantly adjust their working positions to prevent any body exhaustion and injuries.

2.2 Engineering Professionalism

Professional practice is an important course about the important roles and responsibilities that professional engineers have towards society and their peers, the integrity of their work and finally to ensure that their work always meets the highest standards while complying with the law and Code of Ethics. The team demonstrated the importance of the workplace's safety by following the Carleton University's health and safety policy, as mentioned in section 2.1, everything was handled professionally with safety being the number one priority of the team members and the people around them. To ensure the public's safety, the team has positioned all electric components and high-power components inside the robot chassis to prevent accidental touch and damage to anyone.

The course puts a strong empathizes on team work, as it's an important aspect in the field of engineering. Projects are mostly done in groups of people with all kinds of educational background and this offers a wider scope of ideas and skills. Throughout the project, the team work was done professionally, every member had their respective tasks to work on but worked together regularly. The team would constantly troubleshoot issues together and offer advice to the others on areas of the project they excelled in. Finally, the team has referenced all outside materials to preserve the integrity of the project.

2.3 Project Management

To control the project in its time constraint weekly checkups occurred between the group members. The following tools that were used to communicate with each other is through WhatsApp and it consisted of checking up on how everyone is doing with their parts. The team used One Drive to finish the deliverables that were part of the project. Trello allows team members to share, communicate ideas and to-do lists, and organizes them in a proper project format. During the start of the term the team used Trello to have a to-do list and it included the items needed throughout the project and the price of each item and as the team progressed through the term things were checked off the to-do list and extra items were placed.

2.4 Justification of Suitability for Degree Program

This section will discuss how the university engineering program studied by the team members is used and applied in the project SERQET.

The team consists of four Systems and computer engineering student. The project system implementation is a mix of many various angles of the program studied.

Firstly, the robot structure was to be designed in a CAD software to be 3D printed or manufactured. This was practiced in the ECOR 1010 course in the reverse engineering project, where students were to reverse engineer and redesign a product using intelliCAD. Furthermore, this was accomplished while maintaining the mechanical structure and stability of the robot we learned in ECOR 1101: Mechanics.

Secondly, even though power management and distribution were a minor aspect in our project, it was only made successful through our understanding of circuits from ELEC 2501.

Thirdly, Microcontroller and the peripheral chips were understood and implemented using our completion and understanding of SYSC3601. Furthermore, SYSC 2003 gave us all the basics of understanding and programming microcontrollers to begin with.

Fourthly, for the implementation of the main communication hub, the team used their knowledge and understanding of SYSC4602, SYSC 3303, and SYSC3010. Each of these courses gave further information on networking, UDP protocol, and signal processing.

Finally, through the various different software and programming courses taken, the team was able to use the knowledge gained to program the different aspects of the projects with ease.

In general, the team's work during the duration of the project and the program courses and the knowledge provided, the team believes that we have used many various aspects of this knowledge to implement the project successfully.

2.5 Individual Contributions

All the team members fully contributed in every aspect of the project. This section itemizes the individual contributions of each team member.

2.5.1 Project Contributions

The following table states what each group member contributed to the project.

Table 6: Group members contribution to the project

Group Members	Project Contribution
Abdulrahim Kaddoura	RPi Communication, Live Video Feed
Dalia Matar	Structure of Scorpion, RPi Communication
Maria El-Hamad	Arduino, Power Management
Rami Daham	Android Application, RPi Functionality

2.5.2 Report Contributions

This section will show the team's report contribution.

Abdulrahim Kaddoura:

- Problem Motivation
- Problem Statement
- Proposed Solution
- Accomplishments
- Wireless Communication and Access Point
- Live Video Feedback
- Final Project Status
- Closing Statement

Dalia Matar:

- Introduction
- Project Background
- Project System Overview
- Structural Design
- Motor and Joints
- Health and Safety
- Engineering Professionalism

Maria El-Hamad:

- Abstract
- Acknowledgement
- Microcontroller and PWM Driver
- Future Considerations and Upgrades
- Project Management
- Individual Contributions

Rami Daham:

- Overview of Remainder of Report
- Technical Solution
- Raspberry Pi Functionality
- Android Mobile Application
- Future Considerations and Upgrades
- Justification of Suitability for Degree Program

The overall editing was done together as a team.

References

[1]"Daniel Goldman | School of Physics", 19-Feb.-2015. [Online]. Available: https://www.physics.gatech.edu/user/daniel-goldman. [Accessed: 08-Apr.-2019].

[2]"Research," *Robophysics*. [Online]. Available: http://crablab.gatech.edu/pages/research/projects/robophysics.html. [Accessed: 08-Apr-2019].

[3]"Scorpion Facts & Information for Kids and Researchers," *Scorpion Sweepers Pest Control*, 27-Mar-2019. [Online]. Available: https://www.scorpsweep.com/facts/. [Accessed: 08-Apr-2019].

[4]"Adafruit Micro Servo - MG90S High Torque Metal Gear [ADA1143]," *The Pi Hut*. [Online]. Available: https://thepihut.com/products/adafruit-micro-servo-mg90s-high-torque-metal-gear. [Accessed: 08-Apr-2019].

[5]"TowerPro MG996R Servo," *TowerPro MG996R Servo Specifications and Reviews*. [Online]. Available: https://servodatabase.com/servo/towerpro/mg996r. [Accessed: 08-Apr-2019].

[6]"Power Supply," *Power Supply - Raspberry Pi Documentation*. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md. [Accessed: 08-Apr-2019].

[7] M. Rundle, “Raspberry Pi 3 is the first with built-in Wi-Fi,” *WIRED*, 04-Oct-2017. [Online]. Available: https://www.wired.co.uk/article/raspberry-pi-three-wifi-bluetooth-release-price-cost. [Accessed: 08-Apr-2019].

[8] G. Kelly, “802.11ac vs 802.11n WiFi: What's The Difference?,” *Forbes*, 12-May-2015. [Online]. Available: https://www.forbes.com/sites/gordonkelly/2014/12/30/802-11ac-vs-802-11n-wifi-whats-the-difference/#5b1544153957. [Accessed: 08-Apr-2019].

[9]“Setting up a Raspberry Pi as an access point in a standalone network (NAT).” [Online]. Available: https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md. [Accessed: 08-Apr-2019].

[10]“Dnsmasq,” *Dnsmasq - network services for small networks.* [Online]. Available: http://www.thekelleys.org.uk/dnsmasq/doc.html. [Accessed: 08-Apr-2019].

[11]“Camera Module V2,” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.org/products/camera-module-v2/. [Accessed: 08-Apr-2019].

[12]“Adafruit PCA9685 16-Channel Servo Driver,” *Library Reference | Adafruit PCA9685 16-Channel Servo Driver | Adafruit Learning System*. [Online]. Available: https://learn.adafruit.com/16-channel-pwm-servo-driver/library-reference. [Accessed: 08-Apr-2019].

[13]Adafruit Industries, “Adafruit 16-Channel 12-bit PWM/Servo Driver - I2C interface,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/815. [Accessed: 08-Apr-2019].

[14]“MasterReader,” *Arduino*. [Online]. Available: https://www.arduino.cc/en/Tutorial/MasterReader. [Accessed: 08-Apr-2019].

[15]“Adafruit PCA9685 16-Channel Servo Driver,” *Overview | Adafruit PCA9685 16-Channel Servo Driver | Adafruit Learning System*. [Online]. Available: https://learn.adafruit.com/16-channel-pwm-servo-driver?view=all. [Accessed: 08-Apr-2019].

[16]“scorpion silhouette vector,” *iStock*. [Online]. Available: https://www.istockphoto.com/ca/vector/scorpion-silhouette-vector-gm877366178-244841681. [Accessed: 08-Apr-2019].