  
**Assignment Cover Sheet**

|  |  |
| --- | --- |
| **Subject Code:** | CSCI323 |
| **Subject Name:** | Artificial Intelligence |
| **Submission Type:** | Online Submission |
| **Assignment Title:** | Project |
| **Student Name:** | Humaid Khan, Fidel Lim, Muhammad Hassan Naseer |
| **Student Number:** | 6031444, 5847485, 5892806 |
| **Student Phone/Mobile No.** | 0564603915, 0555943154 |
| **Student E-mail:** | [fel585@uowmail.edu.au](mailto:fel585@uowmail.edu.au), [mhn971@uowmail.edu.au](mailto:mhn971@uowmail.edu.au) |
| **Lecturer Name:** | Dr. Farhad Oroumchian |
| **Due Date:** | Week 6 |
| **Date Submitted:** | Week 6 |

|  |  |
| --- | --- |
| **PLAGIARISM:** The penalty for deliberate plagiarism is FAILURE in the subject. Plagiarism is cheating by using the written ideas or submitted work of someone else. UOWD has a strong policy against plagiarism.  The University of Wollongong in Dubai also endorses a policy of non-discriminatory language practice and presentation.  **PLEASE NOTE:**STUDENTS MUST RETAIN A COPY OF ANY WORK SUBMITTED | **DECLARATION:** I/We certify that this is entirely my/our own work, except where I/we have given fully-documented references to the work of others, and that the material contained in this document has not previously been submitted for assessment in any formal course of study. I/we understand the definition and consequences of plagiarism.  **Signature of Student: Lim, Khan,Naseer** |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **Optional Marks:** | | **Comments:** | |

https://my.uowdubai.ac.ae/images/scissors.gif

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Lecturer Assignment Receipt**(To be filled in by student and retained by Lecturer upon return of assignment) | | | **Subject:** | **Assignment Title:** | | **Student Name:** | **Student Number:** | | **Due Date:** | **Date Submitted:** | | **Signature of Student:** | | |

https://my.uowdubai.ac.ae/images/scissors.gif

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Student Assignment Receipt** (To be filled in and retained by Student upon submission of assignment) | | | **Subject:** | **Assignment Title:** | | **Student Name:** | **Student Number:** | | **Due Date:** | **Date Submitted:** | | **Signature of Lecturer** | | |

**CSCI323**

Artificial Intelligence

Project

Chiku

Robot Spider moving using Q-Learning

**Lecturer’s Name:**

Dr. Farhad Oroumchian

**Students’ Name & Number:**

Fidel Lim - 5847485

Humaid Khan - 6031444

Muhammad Hassan Naseer - 5892806

**Table of Contents**

[**Member Breakdown of Tasks**](#_u0qcbnkrtoyn) **4**

[**Question:**](#_iq2bnxb7o3gc) **4**

[**Introduction**](#_3rts3kkbvqrl) **4**

[**Configuration (what pins are used for what)**](#_fw8r3u186y83) **5**

[Pictures of 3D Printing](#_2p2khcq9gn8l) 5

[Design](#_ilbnuv535zkb) 6

[**Description (How your system works)**](#_sw5bk813cwa8) **8**

[**Detail parameters (Max distance, Min distance, and other numerical values)**](#_rgltr1ixcrvf) **8**

[**Limitations (define your scope and what your project can not do)**](#_9kypjbu6sb45) **8**

[**Pictures of the robot in various stages of the work**](#_ey9owa18f0ge) **9**

[**Challenges**](#_w3ogc19hnnwp) **10**

[**Code**](#_dymazpv95v7h) **11**

# 

# Member Breakdown of Tasks

|  |  |  |
| --- | --- | --- |
| **Name** | **Student ID** | **Tasks** |
| Humaid Khan | 6031444 | * Coding (major) * Build the robot * Training and Testing robot |
| Fidel Lim | 5847485 | * Printed 3D printing materials * Build the robot * A little coding |
| Hassan Naseer | 5892806 | * Moral support * Coding chatbot(fail-safe) * Q-Learning research |
| Group work | | * Teamwork * Report |

# Question:

Robot Spider moving using Q-Learning

<https://www.hackster.io/yasaspeiris/raspberry-pi-powered-quadruped-bbb68b>

# Introduction

In this project, We built a four-legged robot and taught it to walk using Q-Learning. Q-Learning is a simple form of reinforcement learning that uses a Q-matrix to find the best actions. Q-Learning involves 3 main attributes: States, Actions, Rewards. The state is a unique configuration of the robot, the action is moving some components of the robot, and the reward is given based on how far the robot has travelled. For each state and action pairs, there was a value in the Q-matrix. The robot will perform random actions and update the Q-matrix based on the reward it achieves. By following the best actions in the current state, the robot learns to walk.

# Configuration (what pins are used for what)

## Pictures of 3D Printing

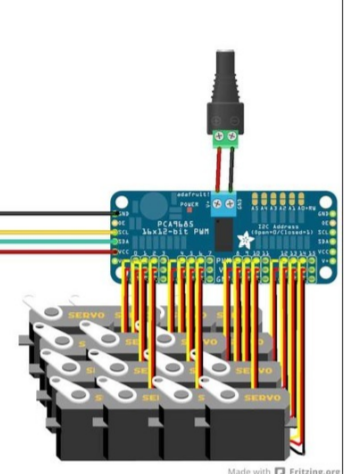
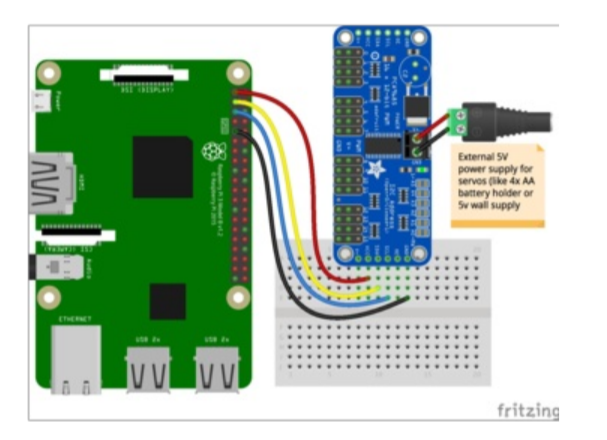
|  |  |
| --- | --- |
| Joint 1 (4x) | |
| Top (1x) | Base (1x) |
| Joint 2 (4x) | Servo Back Pivot (12x) |
| Leg (4x) | Switch Holder (1x) |

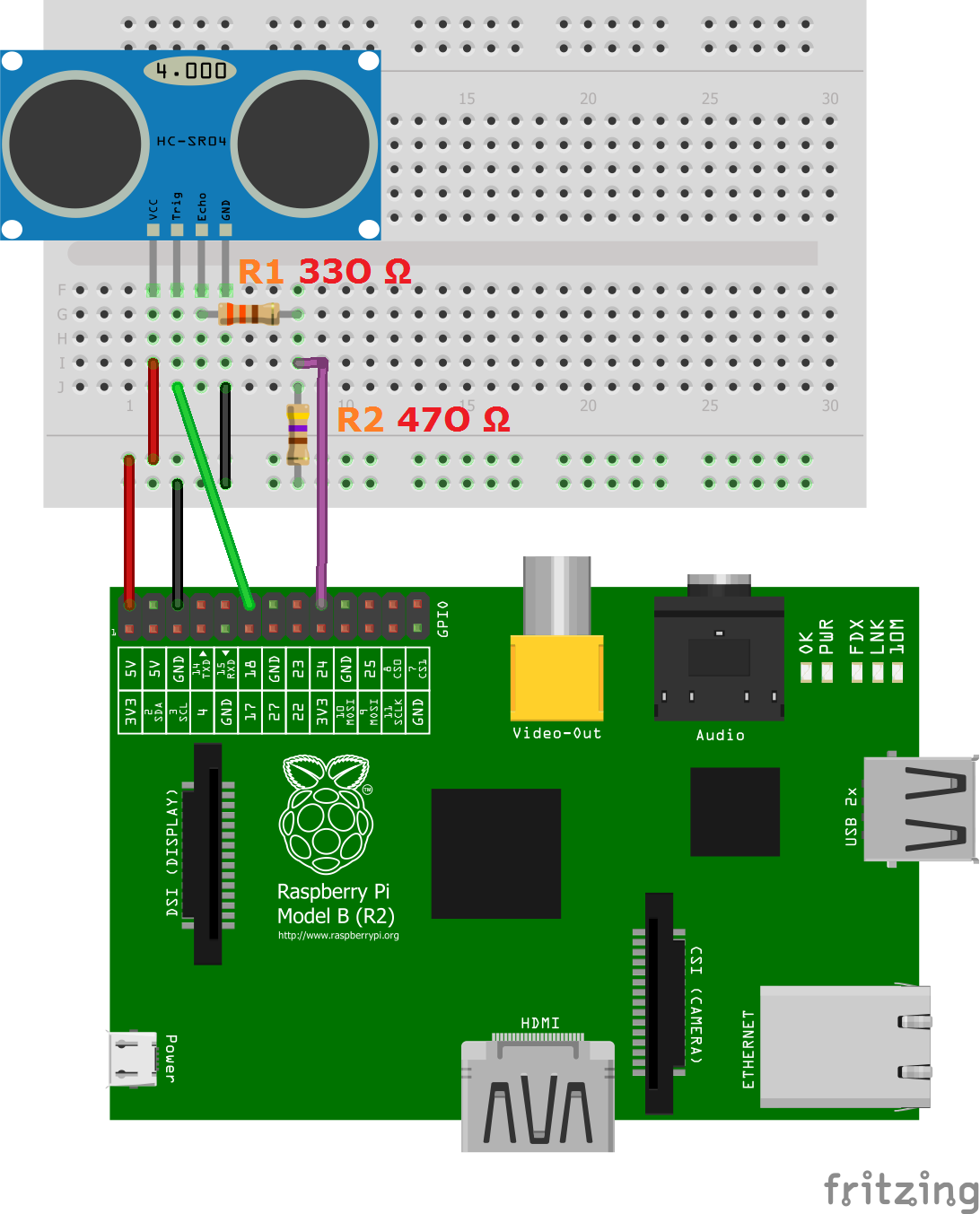
## Design

The first step in building the robot was 3D printing the required parts. By following the tutorial, we printed the following parts:

Next, we need to assemble the robot, this took a lot of time as it was a tight fit that required opening the motors.

Following this, we had to wire the PCA9685, this was used to power the motors as the Raspberry Pi did not have any pwa pins.



Then, we hooked the ultrasonic sensor to the Raspberry Pi by connecting the following circuit

The last step in the hardware was connecting the power to both the PCA9685(SERVOS) and the Raspberry Pi. The servos required a separate 5v, 10A power supply(each motor needed about 1 A).



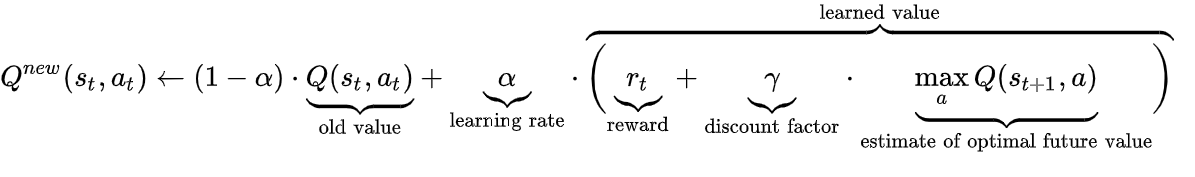
The Raspberry Pi used a power adapter .



# **Description (How your system works)**

Our project follows 4 simple steps.

1. Pick best Q-matrix action
2. Get reward
3. Update Q-matrix
4. Repeat for many iterations



For training the Q learning matrix, we initialized the Q matrix to 0s. Next we use the epsilon greedy method to select either a random action or the action with the highest Q matrix value. We get the distance from the Ultrasonic sensor as means to measure the amount of reward. We use this reward to update the Q matrix for 300 iterations with 15 epochs.

# **Detail parameters (Max distance, Min distance, and other numerical values)**

Starting Angles= [70, 100, 140, 90, 80, 150, 90, 110, 140, 150, 130, 130]

Q learning requires 3 attributes:

1. Rewards= amount the distance has increased
2. States = a configuration of the motors.We have discretized the motor angles to [-0.5, 0, 0.5]. These are then reassigned based on the initial angles of each motor using a factor of 60. E.g for motor 0, the if the state = 0.5, the angle = (0.5\*60) + 70 = 100
3. Actions = movement of motors to a certain angle, We have discretized the motions to [-0.5, 0, 0.5]. These are actions are reassigned using an angle of 60. E.g if the action is -0.5, the motor will move -0.5 \* 60 degrees = -30 degrees

The Q matrix is a combination of 8 motors with these discrete values

Leading to 8^12 by 8^12 matrix

# **Limitations (define your scope and what your project can not do)**

Scope:

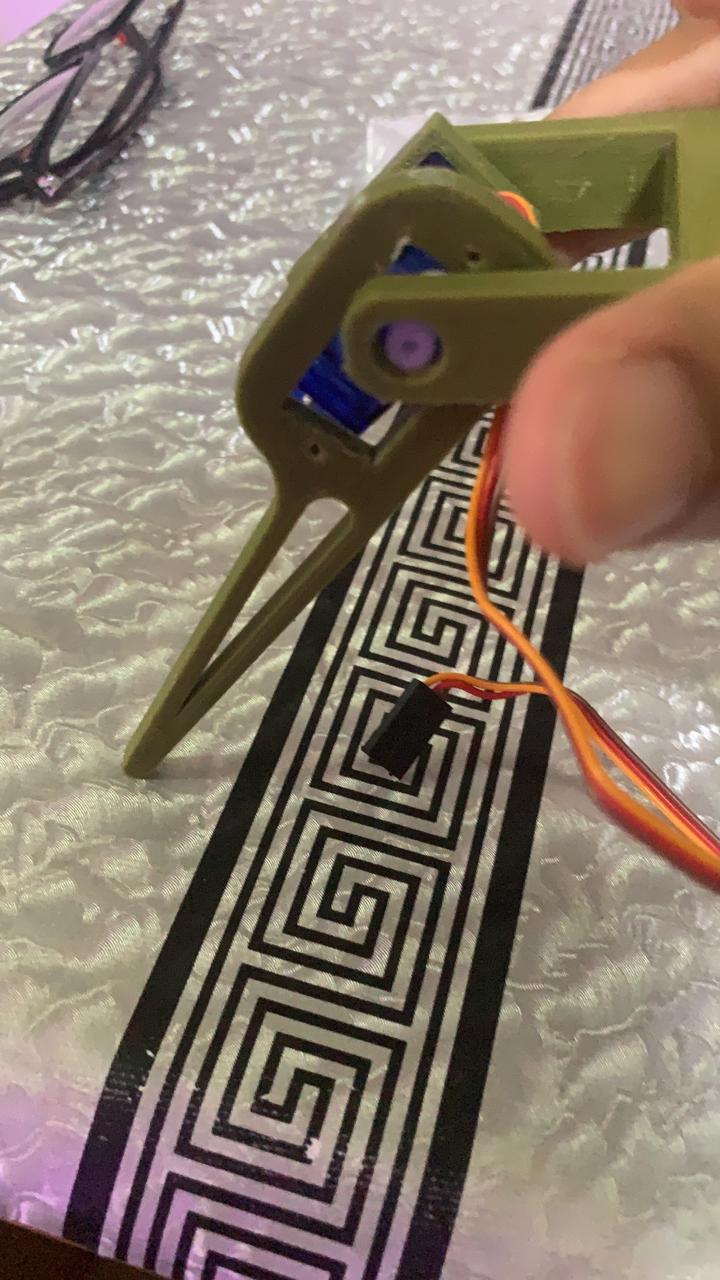
Hardware:

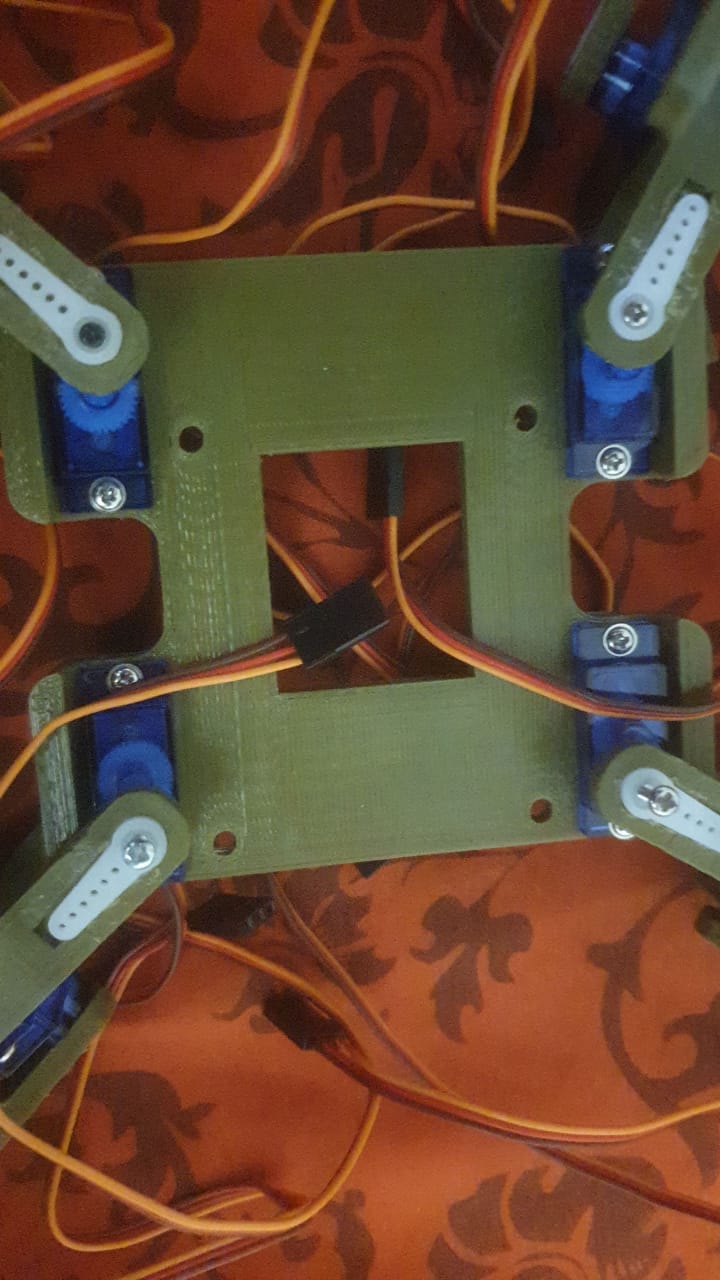
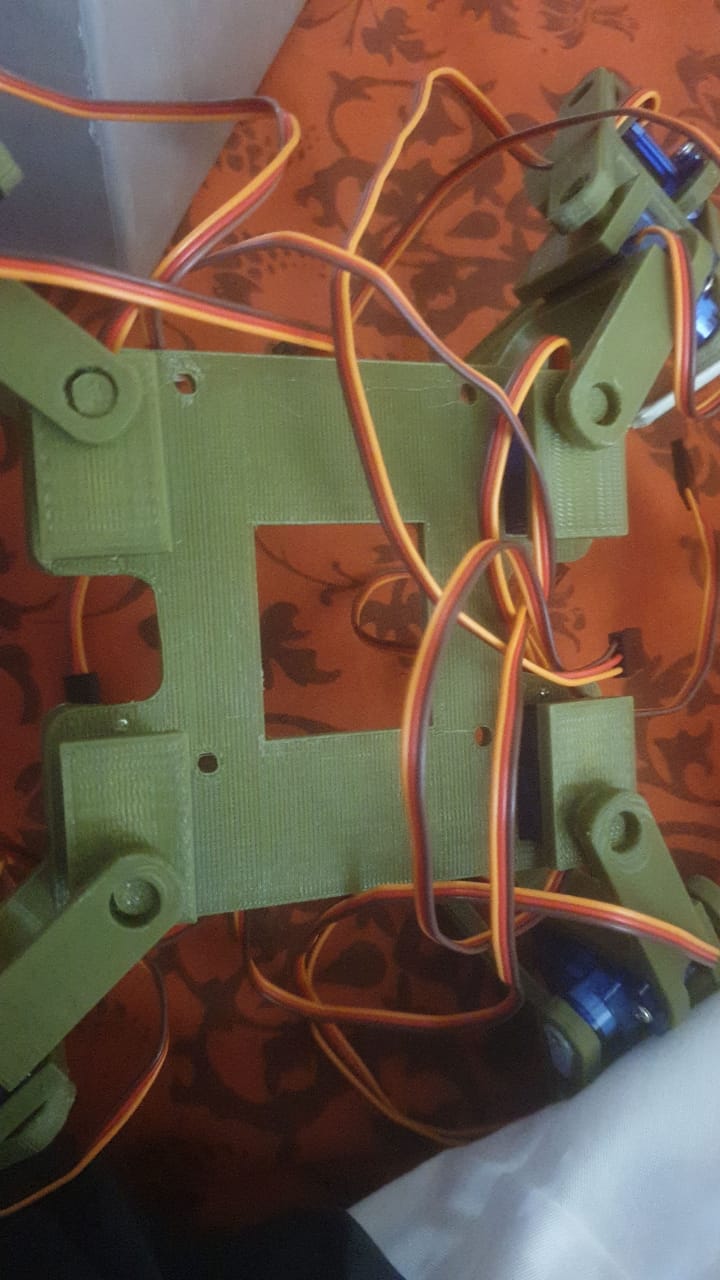
* 3d print materials
* Assemble robots
  + Fix motors using screws✔
  + Solder components✔
  + Glue any loose ends✔
* Wire circuits
  + Connect power pins✔
  + Solder circuit for Ultrasonic✔
  + Connect Ultrasonic✔
  + Connect PCA9685✔
  + Connect Servos✔

Software:

* Test servos (learn to control servos)✔
* Test ultrasonic sensor✔
* Build reward function using Ultrasonic values✔
* Build performAction function using servos✔
* Create Q matrix✔
* Update Q matrix on actions✔

# **Pictures of the robot in various stages of the work**



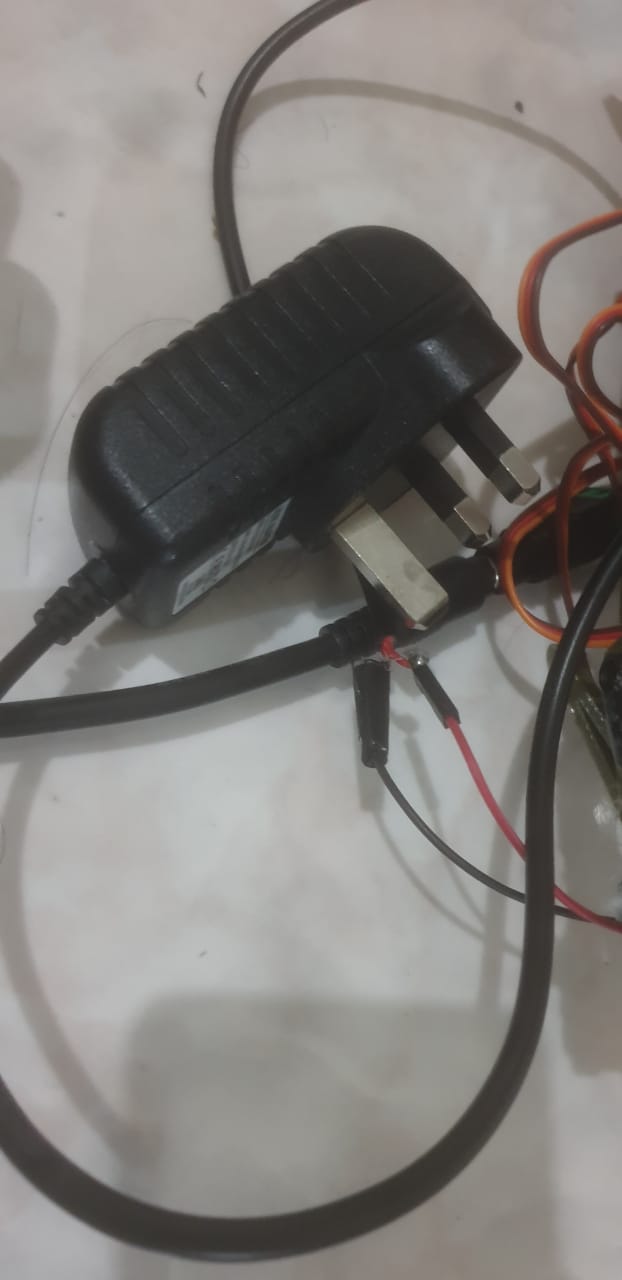


# Challenges

1. One of the challenges we have encountered during this project is the availability of the materials. Some of the materials needed to build the robot is quite hard to find online. We did ask Mr. Majid for some guidance but luck was never with us. Eventually, since we are running out of time, we came up with an alternative solution to use some available materials we have and use it for our project.
2. The next biggest issue we faced was providing power to the rpi as the port was blocked by the motors.



The solution for this was stripping the power adapter and connecting the pins directly to the GPIO pins.



1. Another issue that we had was with the motors. After running the robot several times, we noticed that the robot is operating in a weirdly manner. Focusing on the motors led us to believe that the motors were continuous, since we’re applying Q learning with a matrix, we require discrete angles. Thus, we changed all the servos to half rotating servos(ones with discrete angles).
2. Since we had 12 motors, each acquiring 3 possible actions and 3 possible states, we would need to have 3^12 possible states and 3^12 actions. This led to a long processing time, we solved this issue by holding the position of 4 motors fixed. This reduced the states, actions to 3^8 which helped gain much more speed in training and processing.
3. Training the robot takes time. Due to the problems we encountered (the ones mentioned before this) before, we had to keep training the robot a lot of times until it walks properly.

# Code

Please see attached file for code.