Antweight Combat Robot

Electrical & Computer Engineering

Team 11: Ryan Middendorf and Teodor Tchalakov

Introduction

Objectives/Constraints

- 2lb weight limit
- Must fight in 6’x6’ area
- Made out of PET, PETG, ABS, or PLA/PLA+ filament, 3D printed
- Must use Bluetooth or Wifi communication
- Up to 2 minute fight time
- Controlled with custom pcb
- Easy manual shutdown and automatic shutdown for safety

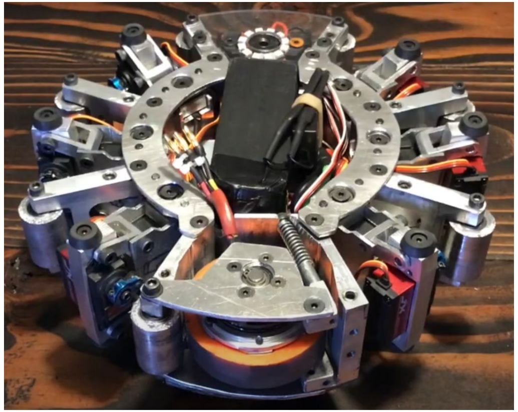
High Level Requirements

- The robot upon powering on should be able to pair with the controller in under 15 seconds
- The weapon should have a maximum tip speed of at least 100 mph and should be able to recover to that speed within 10 seconds after a collision
- The robot must be able to move controllably for a whole match (2 minutes) and be able to cross the arena in under 30 seconds

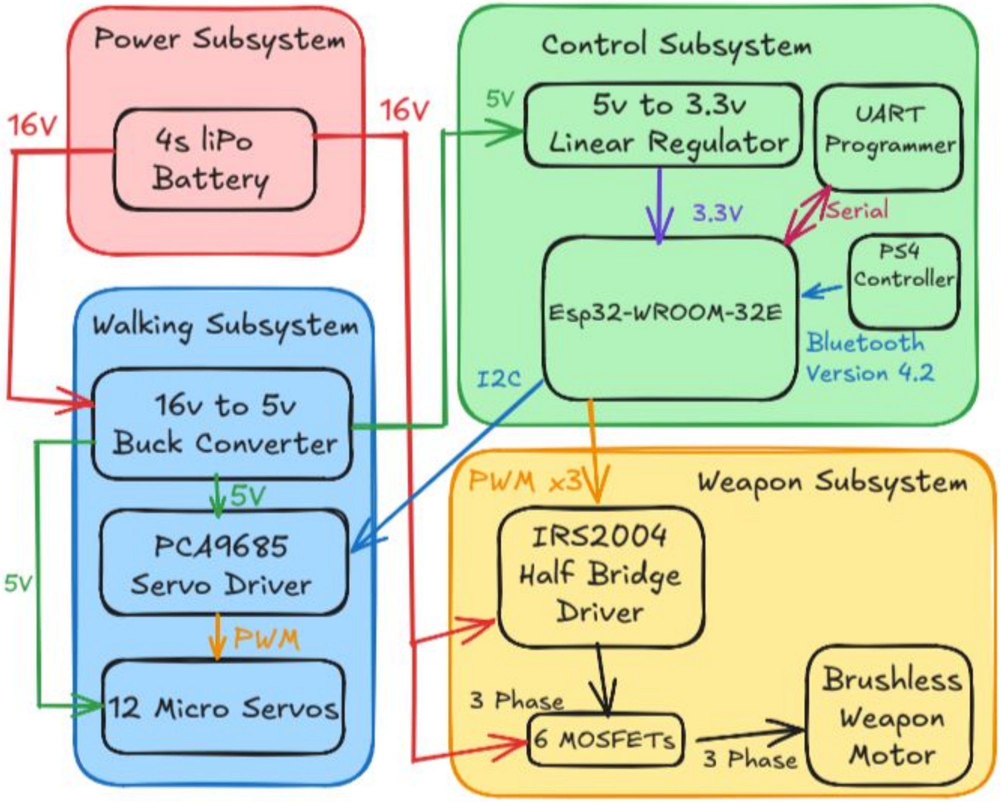
Our Idea

Big Picture

- Walking robot
- Big spinning shell
- Meet standard antweight combot rulesets
- 3D printed out of PLA+
- Connect wirelessly with a PS4 controller
- Controlled with custom pcb
- Win the competition



Final Design



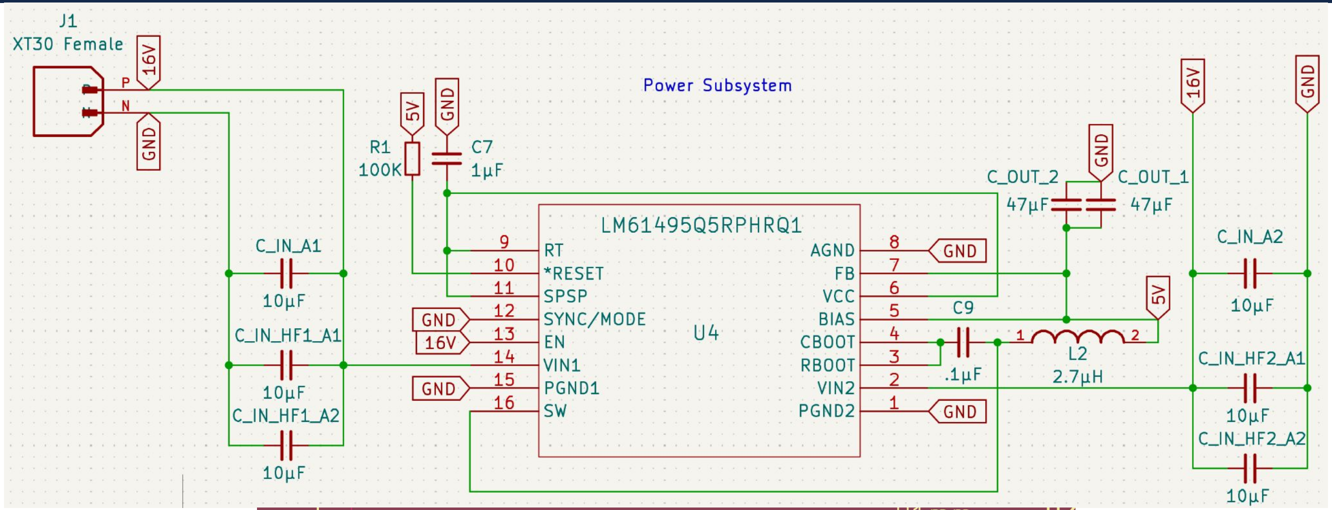
Power Subsystem

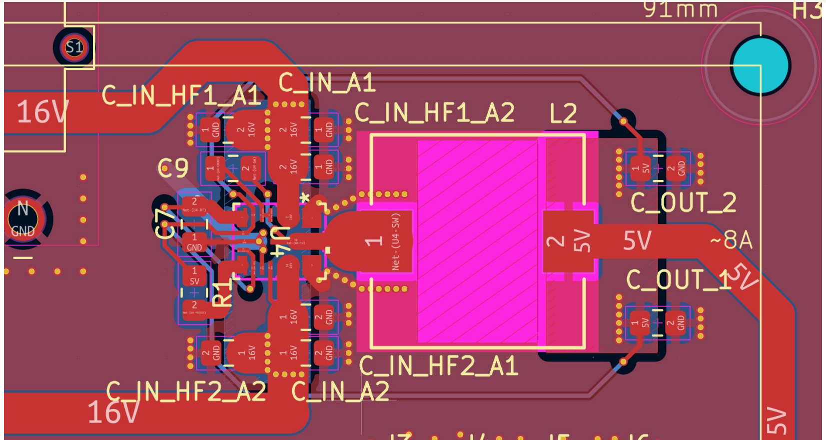
Subsystem Overview

- Provide stable 3.3V power to ESP32 MCU
- Power all 12 servos and weapon motor in active use for 2 minute battle
- Handle peak current draws effectively

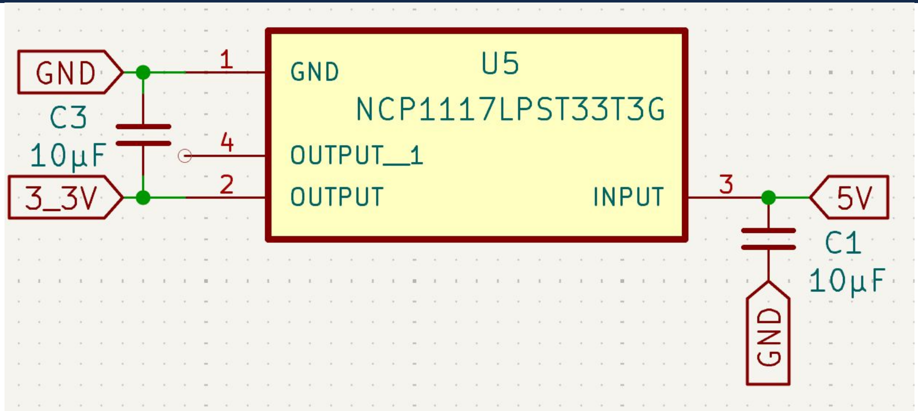
Subsystem Overview

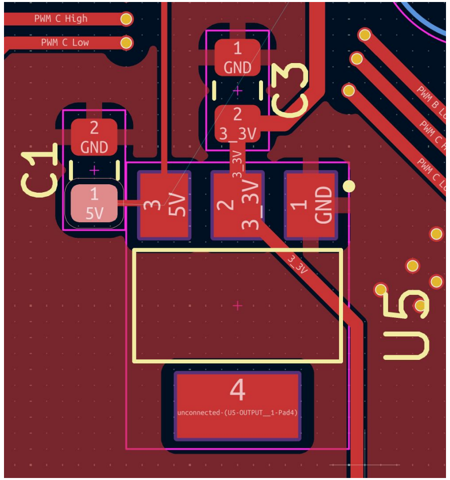
Switching Regulator
5V
10A



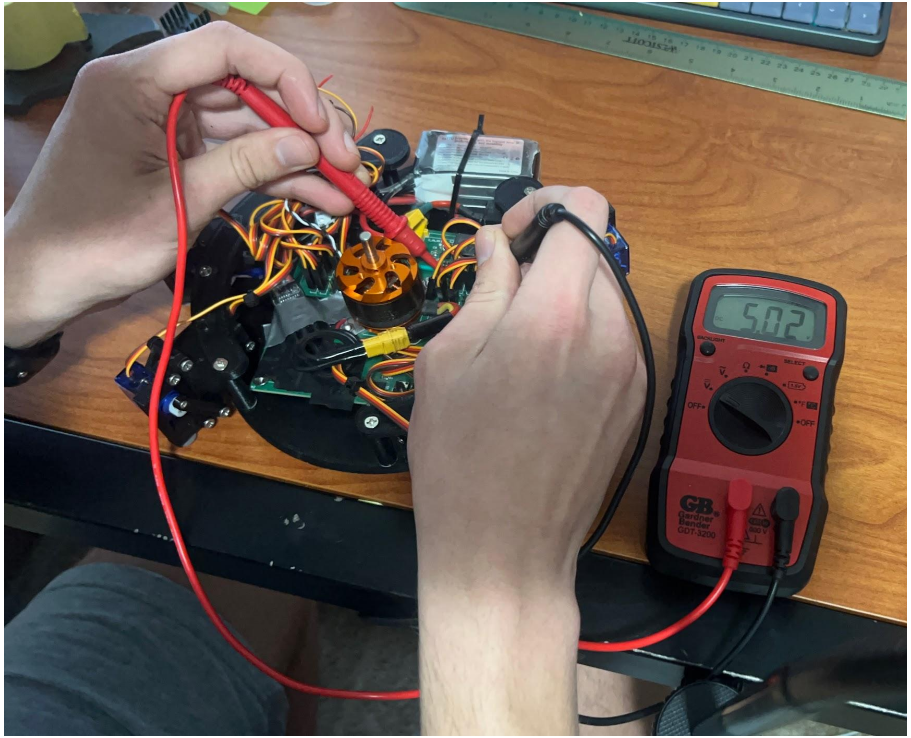


Linear
Regulator
3.3V
1A





Voltage Regulation



16V to 5V Buck Converter

5V to 3.3V Linear Regulator

Battery Calculations

- The battery needs to be able to supply 47.5A if everything draws their max current simultaneously
  - 35A for weapon motor
  - 12A for all 12 servos
  - 0.5A for the ESP32 and PCA9685 chips
- Peak current draw the battery can handle:

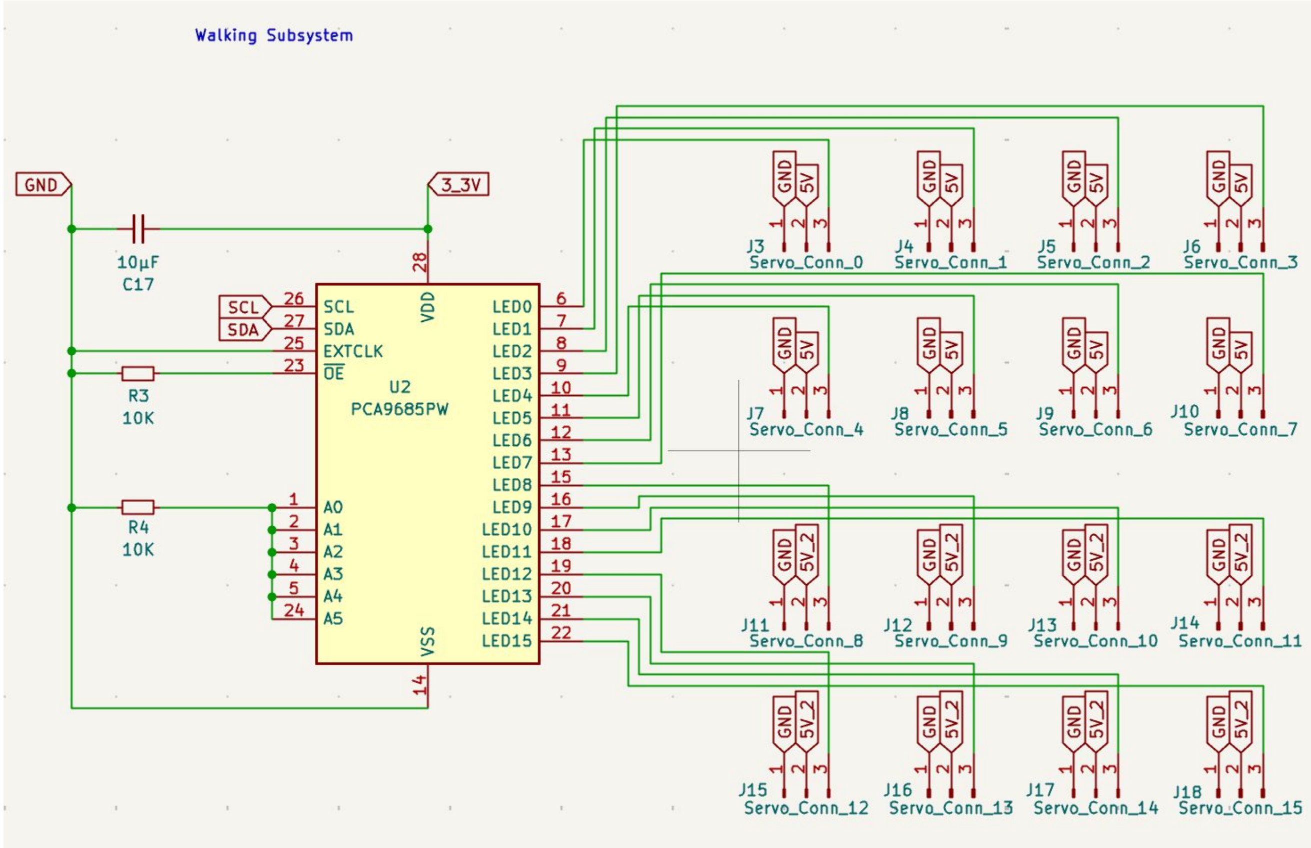
$$850mAh * 75C = 63.75A$$

- Only needs to supply 18.5A during normal operation
  - 15A for the weapon motor
  - 3A for the servos
  - 0.5A for the ESP32 and PCA9685 chips
- Average current the battery can supply for 2 minutes

$$\frac{60\frac{C}{min}}{2min} = 30C \qquad 850mAh * 30C = 25.5A$$

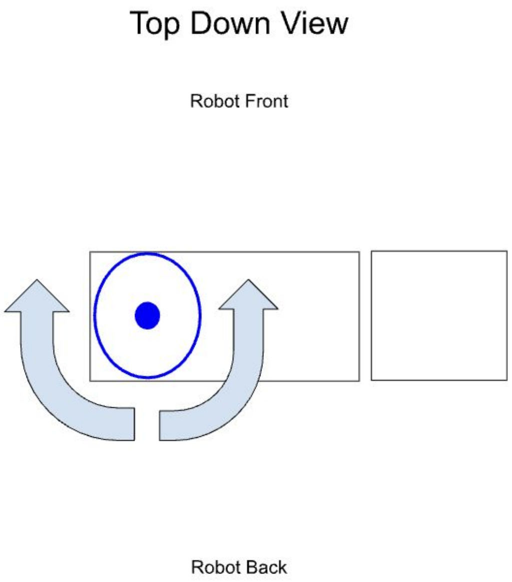
Walking Subsystem

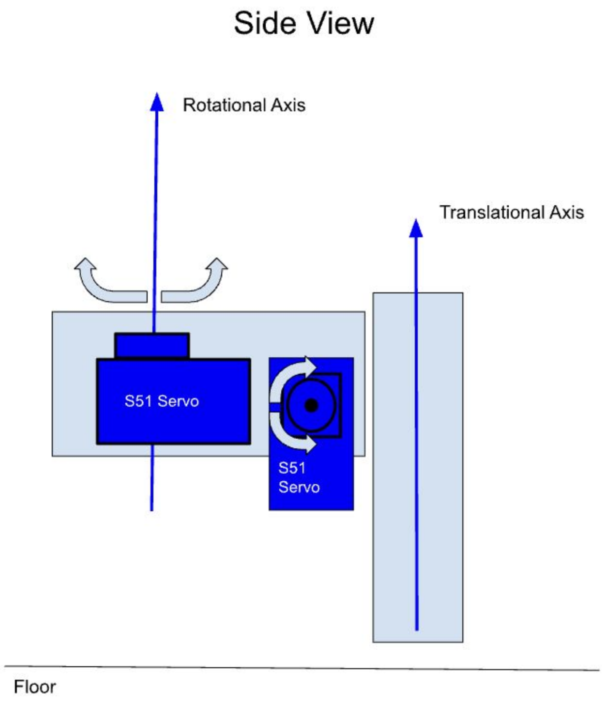
Subsystem Overview

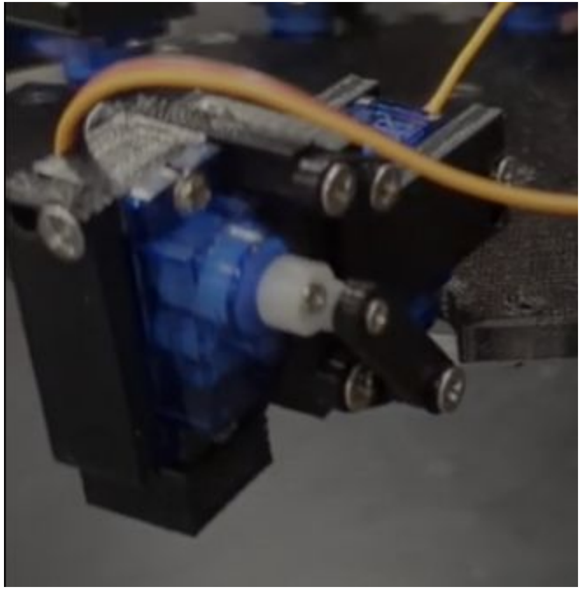


- PCA9685 chip to generate up to 16 PWM signals
- 12 9g micro servos to control all 6 legs
- Controlled over I2C to ESP32

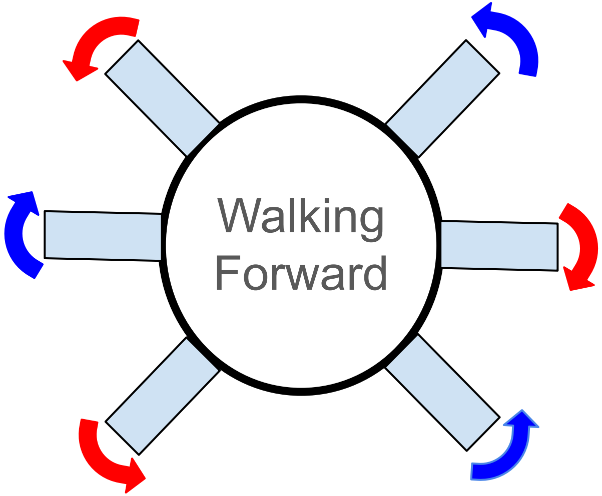
Walking Motion Diagrams



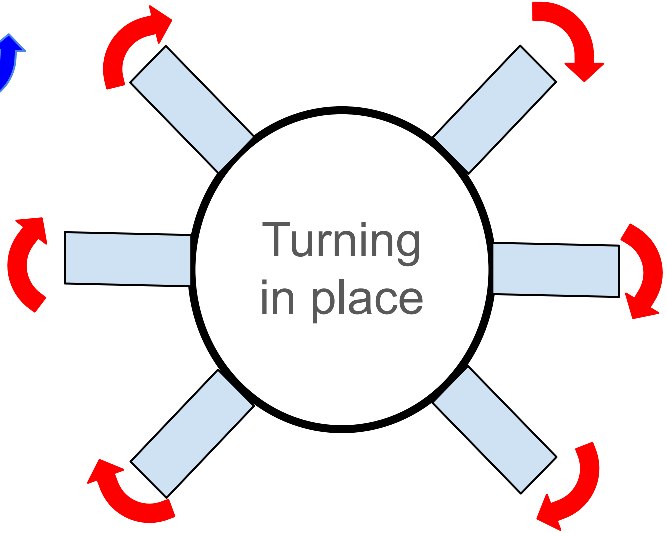


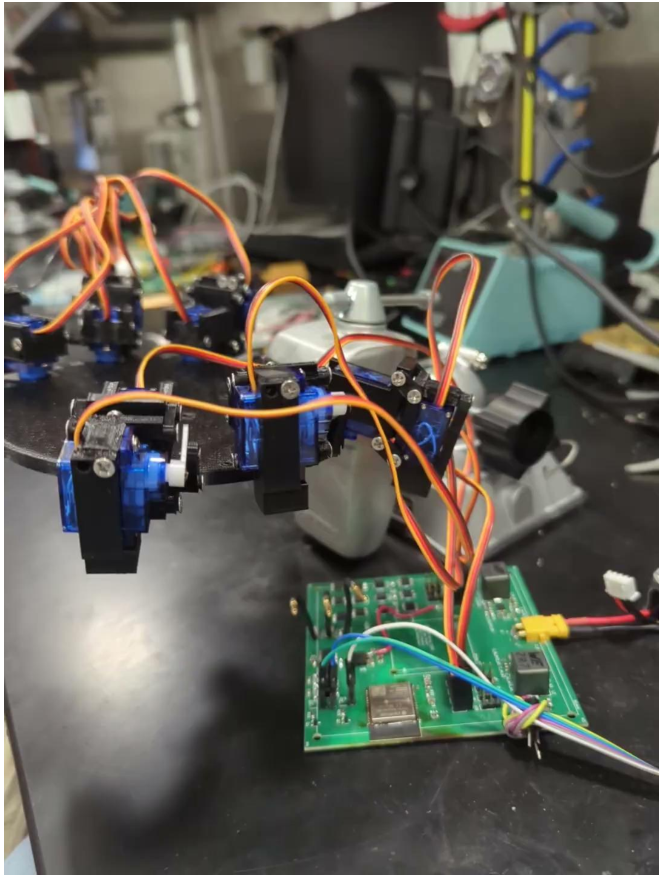


Walking Motion Diagrams

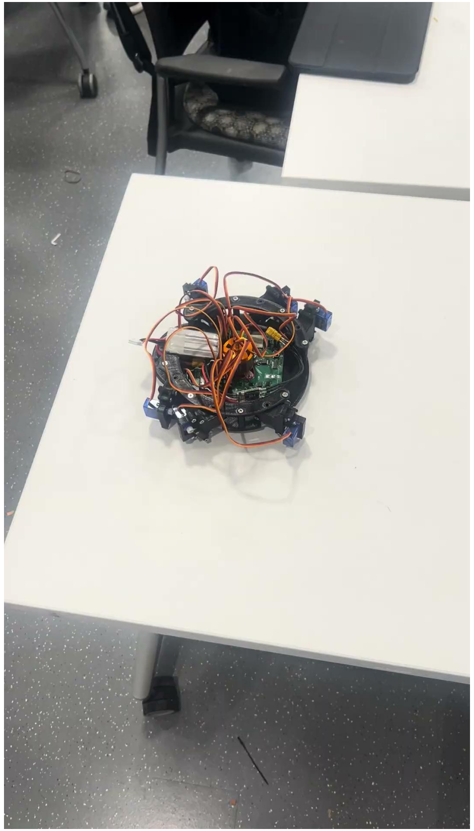


Red means leg is up
Blue means leg is down





Walking Tests



Requirements

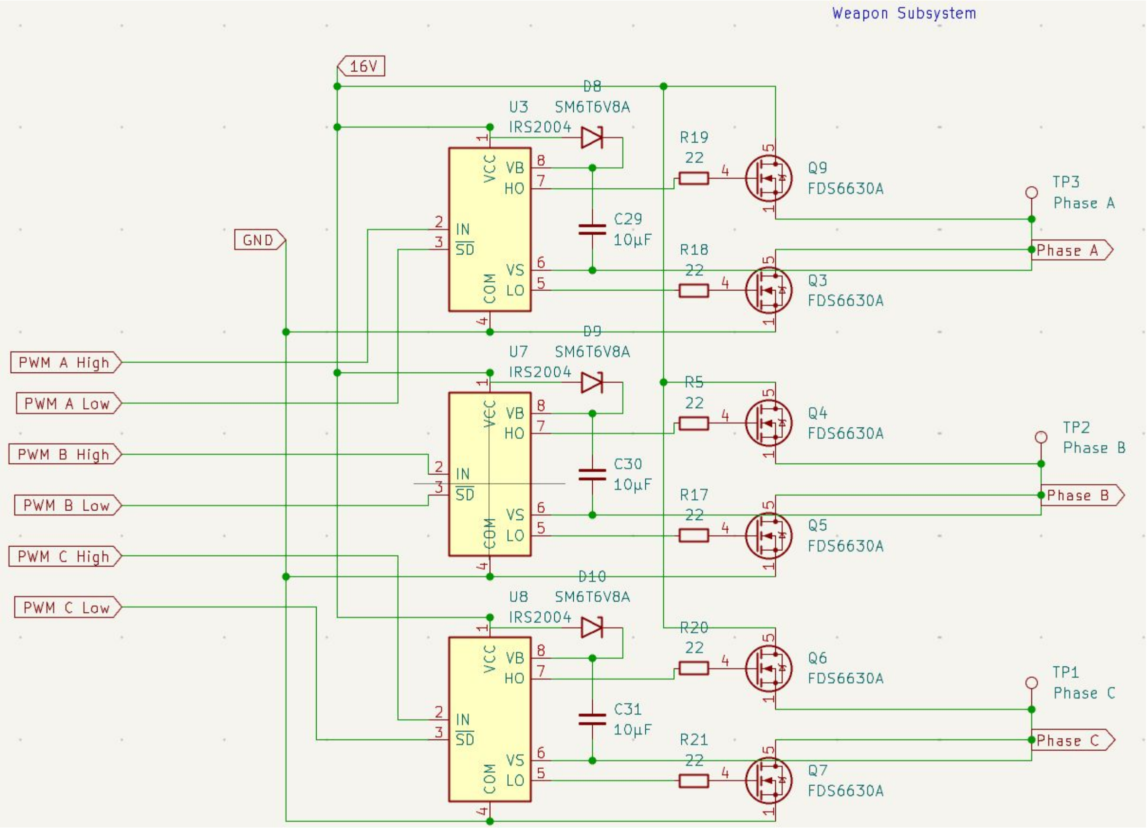
- Each servo must be individually controlled
- Must walk 6 ft in under 30 seconds
- Must be intuitively controllable

Solution

- ESP32 communicates with a PCA8695 chip over I2C
- PCA8695 sends 12 individual PWM signals to all 12 servos
- Each leg iterates through 4 end-effector state positions resulting in a walking motion
- Walked 6 ft in 27.4 seconds

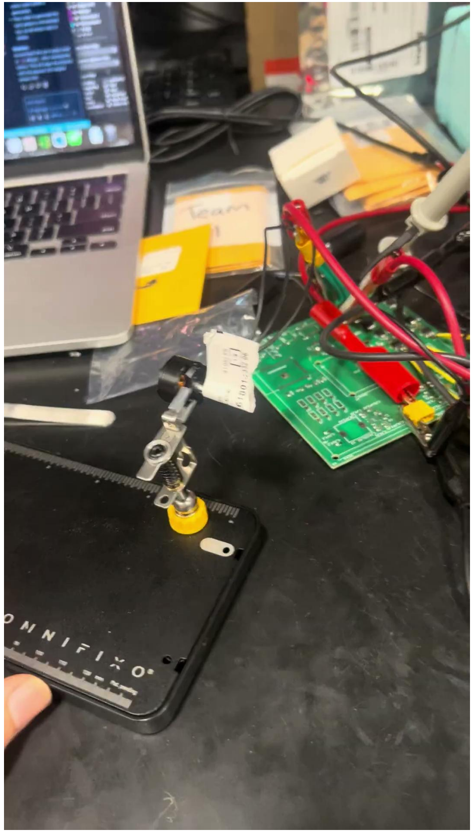
Weapon Subsystem

Subsystem Overview



- Take PWM inputs from the ESP32 to generate a rotating 3-phase output for the brushless motor
- Adjusting the signals from the ESP32 should control the speed of the motor

ESC Testing



Limitations

- ESP32 generates a 6-step trapezoidal signal to commutate the motor but has no idea what position the motor is in
- Unfortunately couldn't figure out how to tune it to smoothly spin the motor
- Only worked on a smaller motor due to current limitations

Future Fixes

- Using a sensored motor
- Replacing the MOSFETs with a version that supports higher current draw

Tip Speed Testing

Requirements

- To meet our high level requirement, we must have a tip speed above 100 mph and recover from hits in less than 10 seconds

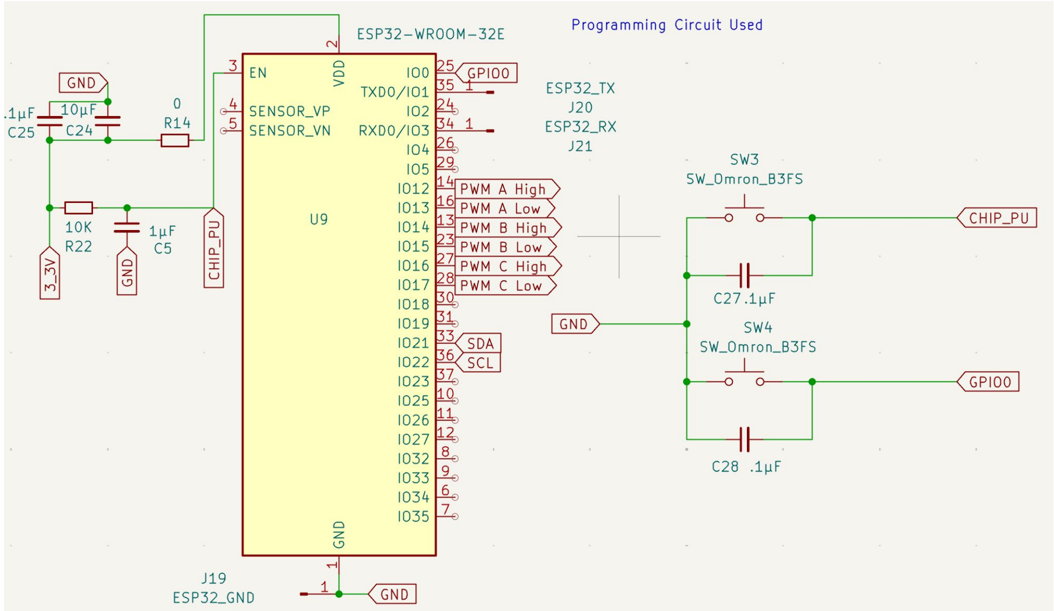
Calculations

- In a test box our shell reached 3172 RPM in 6.52 seconds
- The shell has diameter of 12 inches

$$12\ in * \pi * 3172\ RPM = 119581.6\ \frac{in}{min} * \frac{60\ \frac{min}{hour}}{63360\ \frac{in}{mile}} = 113\ MPH$$

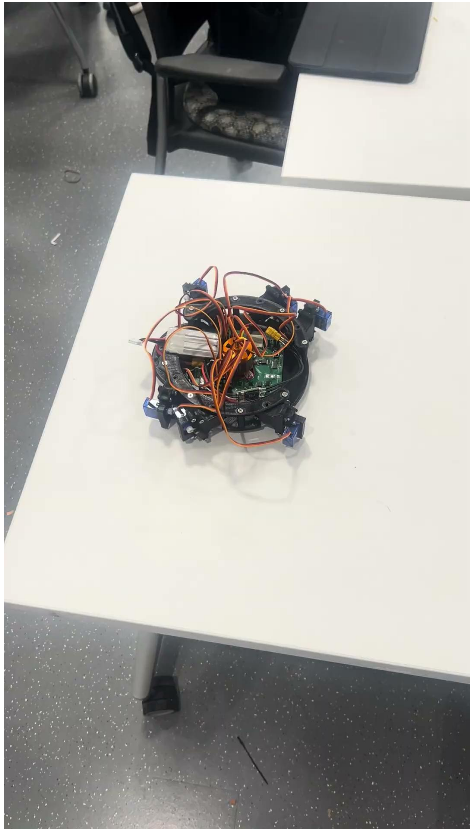
Controller Subsystem

Subsystem Overview



- ESP32-WROOM-32E Module
- Connects to PS4 controller over Bluetooth V4.2 (Bluetooth Classic)
- Controls walking and weapon subsystems
- Has a manual and automatic safety system

Controller Testing



Requirements

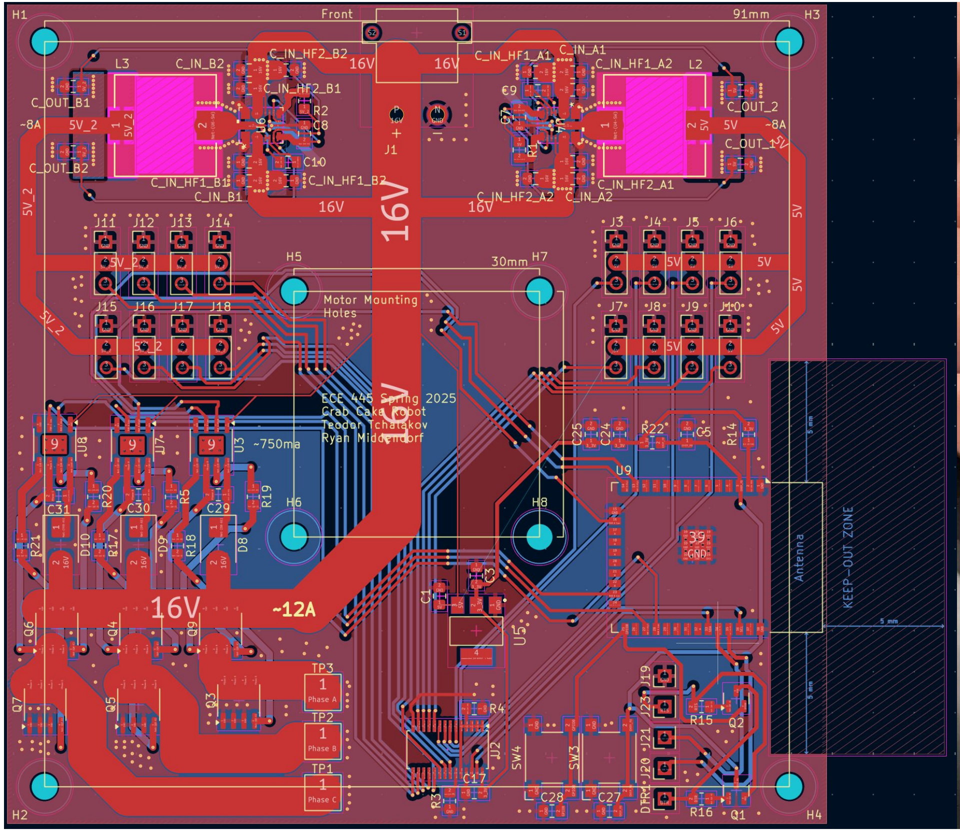
- Each servo must be individually controlled
- Must walk 6 ft in under 30 seconds
- Robot will automatically disable all motors when the Bluetooth connection is lost or manual safety enabled

Solution

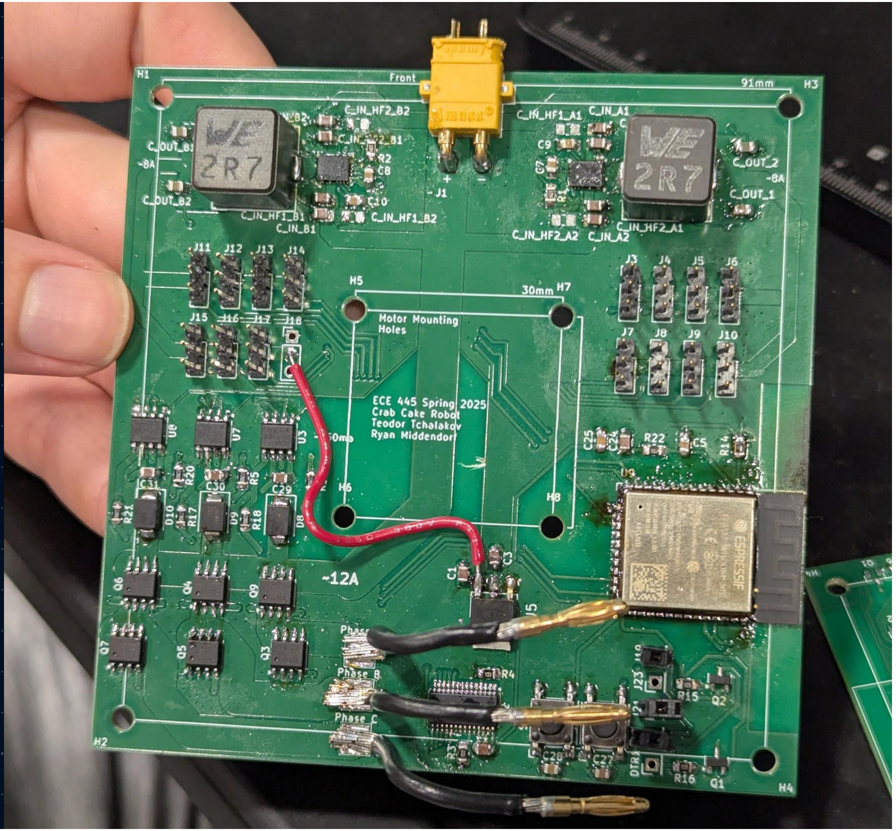
- ESP32 communicates with a PCA8695 chip over I2C
- PCA8695 sends 12 individual PWM signals to all 12 servos
- Each leg iterates through a 4 state state machine resulting in a walking motion
- Walked 6 ft in 27.4 seconds
- Detects Bluetooth connection lost and stops weapon motor

Battle Testing

PCB



Front
91mm
16V
~12A
Motor Mounting Holes
30mm
ECE 445 Spring 2025
Crab Cake Robot
Teodor Tchalakov
Ryan Middendorf
~750ma
Antenna
KEEP-OUT ZONE
Phase A
Phase B
Phase C



Conclusion

What We Learned

- PCB design
- ESP32 BLE communication
- Walking state machine code

What We Would Change

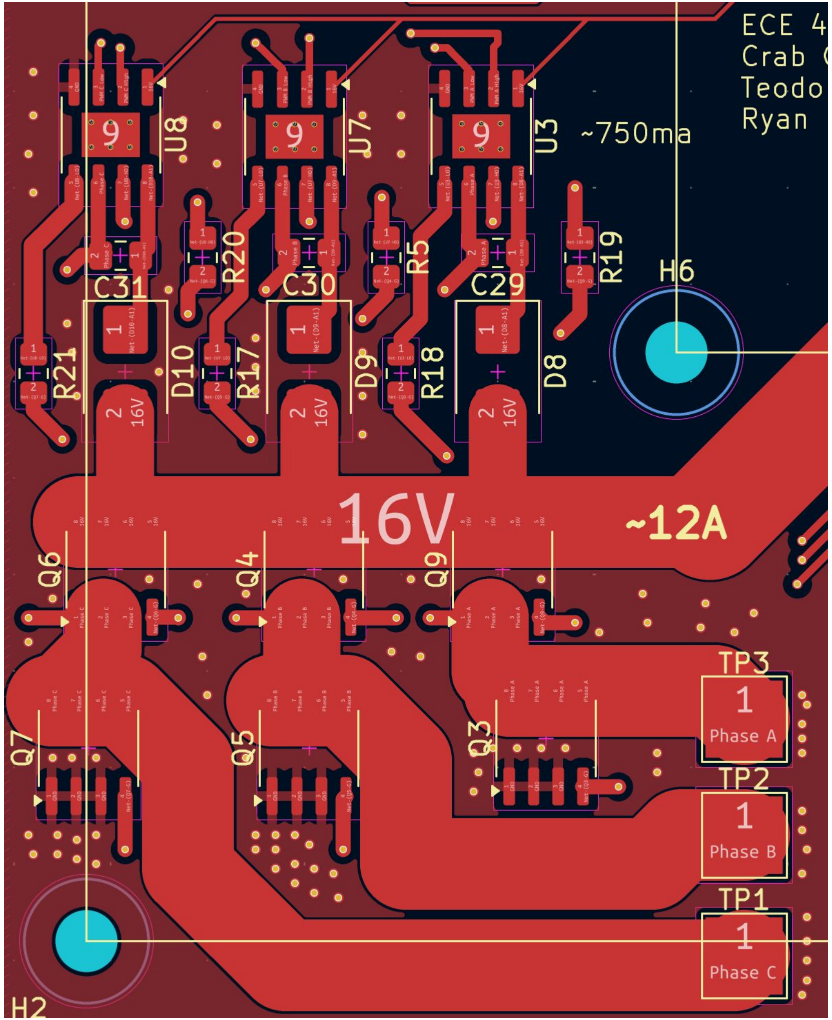
- Improve the ESC MOSFETs current rating
- Use simpler motor controller chip that requires less tuning

What Are The Next Steps

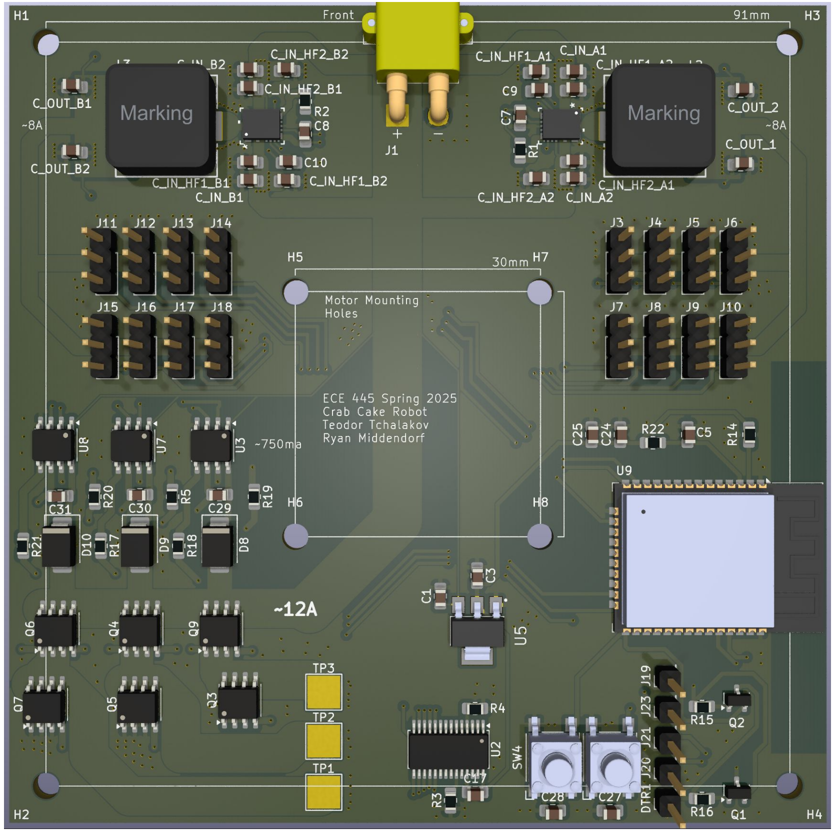
- Increase walking speed by optimizing leg travel
- Reduce overall size and weight
- Improve armor and impact protection

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN



ECE 4
Crab
Teodo
Ryan
~750ma
U8
U7
U3
R20
R5
R19
C31
C30
C29
H6
R21
D10
R17
D9
R18
D8
16V
16V
~12A
Q6
Q4
Q9
Q7
Q5
Q3
TP3
Phase A
TP2
Phase B
TP1
Phase C
H2


Front
91mm
Marking
Marking
30mm
Motor Mounting
Holes
ECE 445 Spring 2025
Crab Cake Robot
Teodor Tchalakov
Ryan Middendorf
~750ma
~12A
~8A
~8A