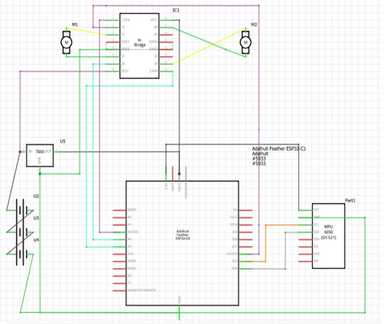
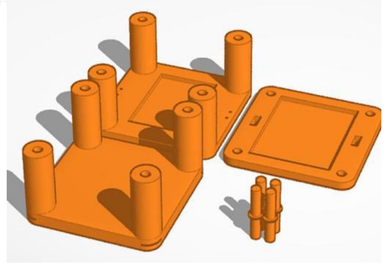
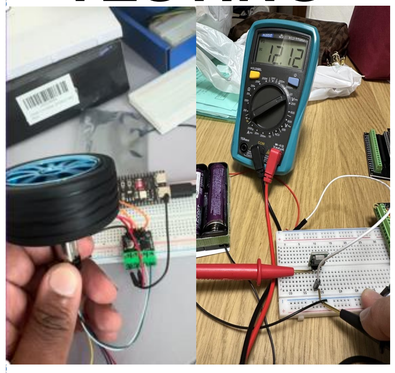
DESIGNING



-Early stages of the project were a testament to drawing upon individual team members strengths and available resources. Early circuit diagrams pulled from team members experience in Electronics Design Automation, much of which was honed by the ASU curriculum.
-CAD skills learned through both formal academia and life experience gave specter to brainstorming sessions. 3D printers owned by teammates gave these visions form



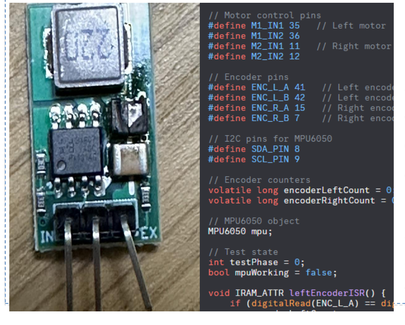
TESTING



-The next stages were characterized by “Dividing and Conquering”. As a team of students physically separated from one another, it was no mean feat to coordinate to construct a physical deliverable. From thousands of miles away, the team set about “eating the elephant”. Members provided proof of concept for voltage regulation, motor control programing, and Inertial Measurement Verification.

Predictive-Integral-Derivative (PID) programing is the function by which the bot can balance and compensate for real world conditions. As it is unique to each bot, and vital to its success, it was one of the most challenging and time-consuming obstacles to test.

TROUBLE-SHOOTING



```
// Motor control pins
#define M1_IN1 35   // Left motor
#define M1_IN2 36
#define M2_IN1 11   // Right motor
#define M2_IN2 12

// Encoder pins
#define ENC_L_A 41   // Left encod
#define ENC_L_B 42   // Left encod
#define ENC_R_A 15   // Right enco
#define ENC_R_B 7    // Right enco

// I2C pins for MPU6050
#define SDA_PIN 8
#define SCL_PIN 9

// Encoder counters
volatile long encoderLeftCount = 0
volatile long encoderRightCount =

// MPU6050 object
MPU6050 mpu;

// Test state
int testPhase = 0;
bool mpuWorking = false;

void IRAM_ATTR leftEncoderISR() {
    if (digitalRead(ENC_L_A) == di
```

-Lessons were learned in the transition from SPICE programs to tabletop assembly. H-bridges were burnt. ESP32s were bricked. Countless lines of Proportional-Integral-Derivative code was deleted and re-written.
-The team found many ways not to design a self-balancing robot.
-Throughout the arduous process, however, rigorous communication allowed the team to polish the components and coding entrusted to them into a workable finished product.

BUDGET & LOGISTICS

United States Build		Japan Build	
Components	Price	Components	Price
PCB Prototype Boards(x3)	$3.60	GA12-N20 Motors with Encoder (x2)	¥2,482
ESP32S3-DevKitC-1-N8R8(x2)	$12.20	DRV8833 Dual Motor Driver (x5)	¥750 (¥150)
Lipo Battery	$14.50	IMU MPU 6050 (x5)	¥1399 (¥280)
IMU MPU6050	$4.33	Freenove ESP32-WROOM (x2)	¥2,108
LM2596 Voltage Regulator	$1.60	18650 Battery (x2)	¥3000
DRV8871 Motor Driver(x2)	$8	Tamiya RC car wheels and coupling (x4)	¥1900 (¥950)
Pololu Motors with Encoder (x2)	$69.10	LM7805 voltage regulator (x5)	¥750 (¥150)
65MM Wheels w/ Couplers(x2)	$7.97	Breakout board (x2)	¥1,264
SPST On/Off Switch	$1.00	Misc Components/Wastage (resistors, capacitors, wires, solder, etc)	¥5,000
Misc Components(capacitors,wires,etc)	$10.00	TOTAL	¥18,653 ------ Batch Cost (assuming 80% Misc Cost) ¥14,384
TOTAL	$132.30	TOTAL (USD)	$119.57 -------- $92.13

The above shows some of the procurement challenges distinct to our project. Due to differing markets in Japan and the USA, our team had to prioritize locally available, but equivalently serviceable parts for an otherwise uniform build.

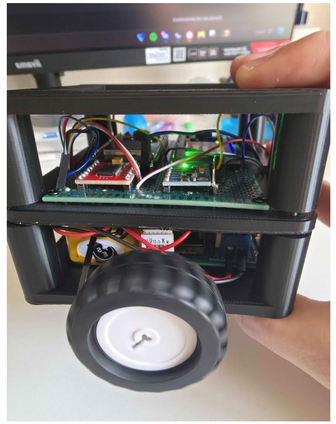
TEAM TWO’S COMPLEMENT SELF BALANCING SEGWAY STYLE ROBOT

Design challenges and solutions of creating a load carrying, autonomous bot. Design focused on creating an easily reproducible platform that can be assembled by STEM focused individuals. Secondary focus on "low-barrier-to-entry" price point and reproducibility for stakeholders or students in diverse regions.

Anterior View and **Lateral View**

Final Product

Architecture:
- Dual ESP32 boards
- MPU6050 IMU
- GA12-N20 DC motors with encoders
- TB6612FNG Dual Motor Driver
- 11.1 V 1500mAh LiPo battery
- Buck Voltage Converter

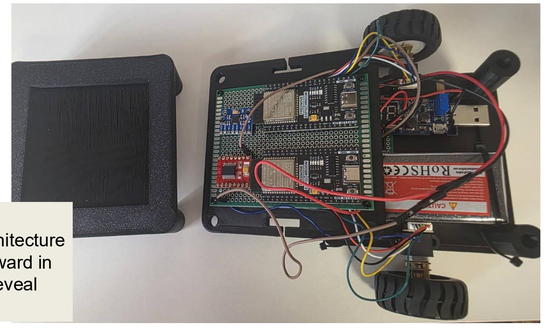


Key Features:
- Real-time balance control
- Dynamic load handling
- Low-cost, reproducible design

Performance and results

- Settling time: ~ 0.5 - 1.5 seconds
- Stable angle range: ± 2°
- Payload capacity: ~150 grams

Observations:
- Robot successfully maintains balance under small disturbances
- Aggressive PID tuning caused oscillations which caused the robot to behave erratically



With the enclosure sections separated, the three-tier architecture is clearly visible, fanned outward in a card-like arrangement to reveal each internal layer.

Ira A. Fulton Schools of Engineering
Arizona State University