

COMPETITION

PREPARATION DAY

Debug, Build, Test!

Thursday, February 5, 2026



Today's Schedule

SESSION 1: 9:00 - 10:50 AM - Skills Building



9:00 - 9:20 • Python Review & Setup

Quick review, load code to Picos



9:20 - 10:10 • Debug Challenge

Fix 3 broken code snippets



10:10 - 10:50 • Programming Behaviors

5 movement challenges



BREAK: 10:50 - 11:10 AM

SESSION 2: 11:10 AM - 1:00 PM - Competition Prep

Python Quick Review

Remember from Tuesday:

Variables

Store values

```
speed = 50
```

Functions

Reusable code

```
def move_forward():
```

If Statements

Make decisions

```
if distance < 20:
```

Loops

Repeat actions

```
for i in range(4):
```

DEBUG CHALLENGE!

Fix 3 broken code snippets

Work in pairs • Test on your robot • Ask for help!

1 Challenge 1: Motor Test (EASY)

This code should spin all 4 wheels forward for 2 seconds. Find the bugs!

```
from machine import Pin
import time

# Front wheels
front_left_1 = Pin(8, Pin.OUT)
front_left_2 = Pin(9, Pin.OUT)
front_right_1 = Pin(11, Pin.OUT)
front_right_2 = Pin(12, Pin.OUT)

# Rear wheels
rear_left_1 = Pin(5, Pin.OUT)
rear_left_2 = Pin(6, Pin.OUT)
rear_right_1 = Pin(2, Pin.OUT)
rear_right_2 = Pin(3, Pin.OUT)

# Spin all wheels forward
def spin_forward()
front_left_1.high(); front_left_2.low()
front_right_1.high(); front_right_2.low()
rear_left_1.high(); rear_left_2.low()
rear_right_1.high(); rear_right_2.low()

spin_forward()
time.sleep(2)
```



Hint: Look for missing punctuation and indentation!

2 Challenge 2: Sensor Reading (MEDIUM)

This code should read distance every second. Find the bugs!

```
from machine import Pin, time_pulse_us
import time

trig = Pin(14, Pin.OUT)
echo = Pin(15, Pin.OUT)

def get_distance():
    trig.value(0)
    time.sleep_us(2)
    trig.value(1)
    time.sleep_us(10)
    trig.value(0)

    pulse_time = time_pulse_us(echo, 1, 30000)
    distance = (pulse_time * 2) / 29.1
    return distance

while True:
    dist = get_distance()
    print("Distance:", distance, "cm")
    time.sleep(1)
```



Hint: Check pin modes, math formulas, and variable names!

3 Challenge 3: Square Pattern (HARD)

This code should make the robot drive in a square. Multiple bugs!

```
from machine import Pin
import time

# Setup pins (abbreviated for space)
f11 = Pin(8, Pin.OUT); f12 = Pin(9, Pin.OUT)
fr1 = Pin(11, Pin.OUT); fr2 = Pin(12, Pin.OUT)
r11 = Pin(5, Pin.OUT); r12 = Pin(6, Pin.OUT)
rr1 = Pin(2, Pin.OUT); rr2 = Pin(3, Pin.OUT)

def forward():
    f11.high(); f12.low(); fr1.high(); fr2.low()
    r11.high(); r12.low(); rr1.high(); rr2.low()

def turn_right():
    f11.high(); f12.low(); fr1.low(); fr2.high()
    r11.high(); r12.low(); rr1.low(); rr2.high()

# Drive in square
for i in range(3):
    forward()
    time.sleep(2)
    turn_right
    time.sleep(0.5)
```



Hint: How many sides in a square? Check function calls!

Programming Behaviors

Complete some of these challenges (until 10:50):



Square Pattern (EASY - Guided)

Drive in a perfect square (4 sides, 90° turns)



Spin in Place (EASY)

Rotate 360° without moving forward



Move Until Obstacle (MEDIUM)

Go forward, stop when object < 15cm detected



Figure-8 (MEDIUM)

Program smooth figure-8 movement



Simple Avoidance (HARD)

Drive forward, backup & turn when obstacle hit



Test each behavior on your robot!

 **Break Time!** 

20 minutes - back at 11:10!

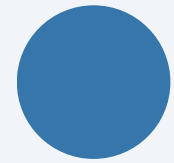
After break: Competition Preparation! 



Understanding the Full Robot Code

Walk through main.py - the complete obstacle avoidance system:

⚠ Motors are DIRECTION-ONLY (full speed when ON) - no PWM speed control



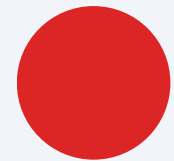
Motor Control → `set_motors_direct()` - Controls all 4 wheels



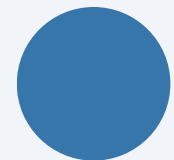
Sensor Reading → `get_distance()` - Measures obstacle distance



Obstacle Detection → `check_obstacles()` - Monitors while driving



Avoidance Sequence → `avoid_obstacle()` - Stop, backup, turn 180°



WiFi Control → Receives commands from Freenove app



All these pieces work together to make autonomous navigation!



Download Freenove App

 **Only ONE person per team needs to download this app!**

How to Download:



iPhone/iPad

1. Open App Store
2. Search: **Freenove**
3. Download the app with the ORANGE icon

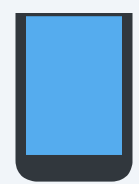


Android

1. Open Google Play Store
2. Search: **Freenove**
3. Download the app with the ORANGE icon

Your Robot Car Model:

4WD Car Kit for Raspberry Pi Pico(w)



WiFi Setup & Robot Control

Set up your phone hotspot and connect:

1. Create Phone Hotspot

Use a simple name (e.g., 'Team1', 'Robot2') and password

2. Update Code

Edit lines 379-380 in main.py with your hotspot name & password

3. Upload & Run

Upload to Pico - watch Thonny console for IP address

4. Connect Freenove App

Enter robot's IP address and port 5000

5. Drive!

Use joysticks - obstacle avoidance runs automatically!

```
wlan.connect("YourHotspotName", "YourPassword")
```



Customize Your Robot

Modify these settings in main.py:

Line 51

`OBSTACLE_DETECT_CM = 20`

→ Try 15 (risky) or 30 (safe) - detection distance

Line 52

`AVOID_BACKUP_MS = 600`

→ Try 400 (short) or 1000 (long) - backup time

Line 53

`AVOID_TURN_MS = 1200`

→ 600 (90°), 900 (135°), 1200 (180°), 1800 (270°)

Line 252

`random.choice(['LEFT', 'RIGHT'])`

→ Change to just 'LEFT' or 'RIGHT' - consistent turns

 Edit, save, upload to Pico, test on course!



Team Strategy Session

Choose your competition approach (examples):

Aggressive Strategy

Detect at 15cm, quick turns (90°), fast reactions

✓ *Fast course time* • ⚠ *Higher risk of collisions*

Conservative Strategy

Detect at 30-40cm, full 180° turns, safe backup

✓ *Very reliable avoidance* • ⚠ *Slower course time*

Balanced Strategy

Detect at 20-25cm, 135° turns, medium backup

✓ *Good balance of speed & safety* • ⚠ *May need fine-tuning*



Discuss with your team and customize your code accordingly!



PRACTICE RUNS!

Test on the actual Friday course

Debug • Refine • Perfect your strategy!



Practice Run Guidelines

Make the most of practice time:

1

Run 1: Baseline Test

Use your current settings, see what happens

2

Run 2: Adjust & Test

Fix issues from Run 1, test new settings

3

Run 3: Strategy Test

Test your competition strategy

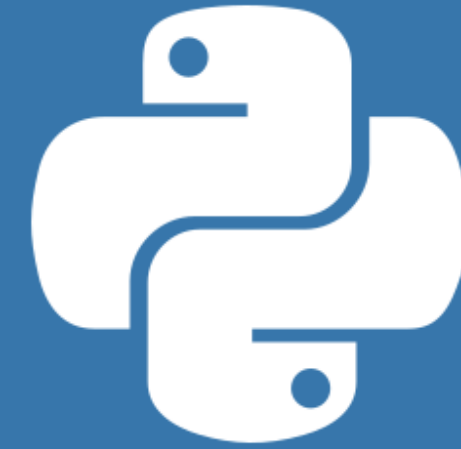
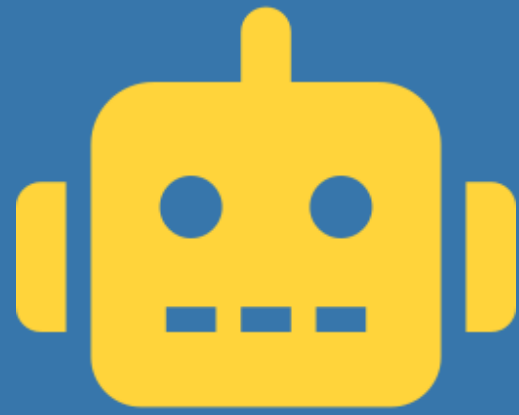
4

Run 4+: Refinement

Fine-tune timing, test edge cases



Goal: Have working, tested code ready for Friday!



Excellent Work Today!

Tomorrow:

- IBM/RedHat AI Session with James Mernin
- COMPETITION TIME! 🏆
- PIZZAS!

See you tomorrow - bring your A-game!