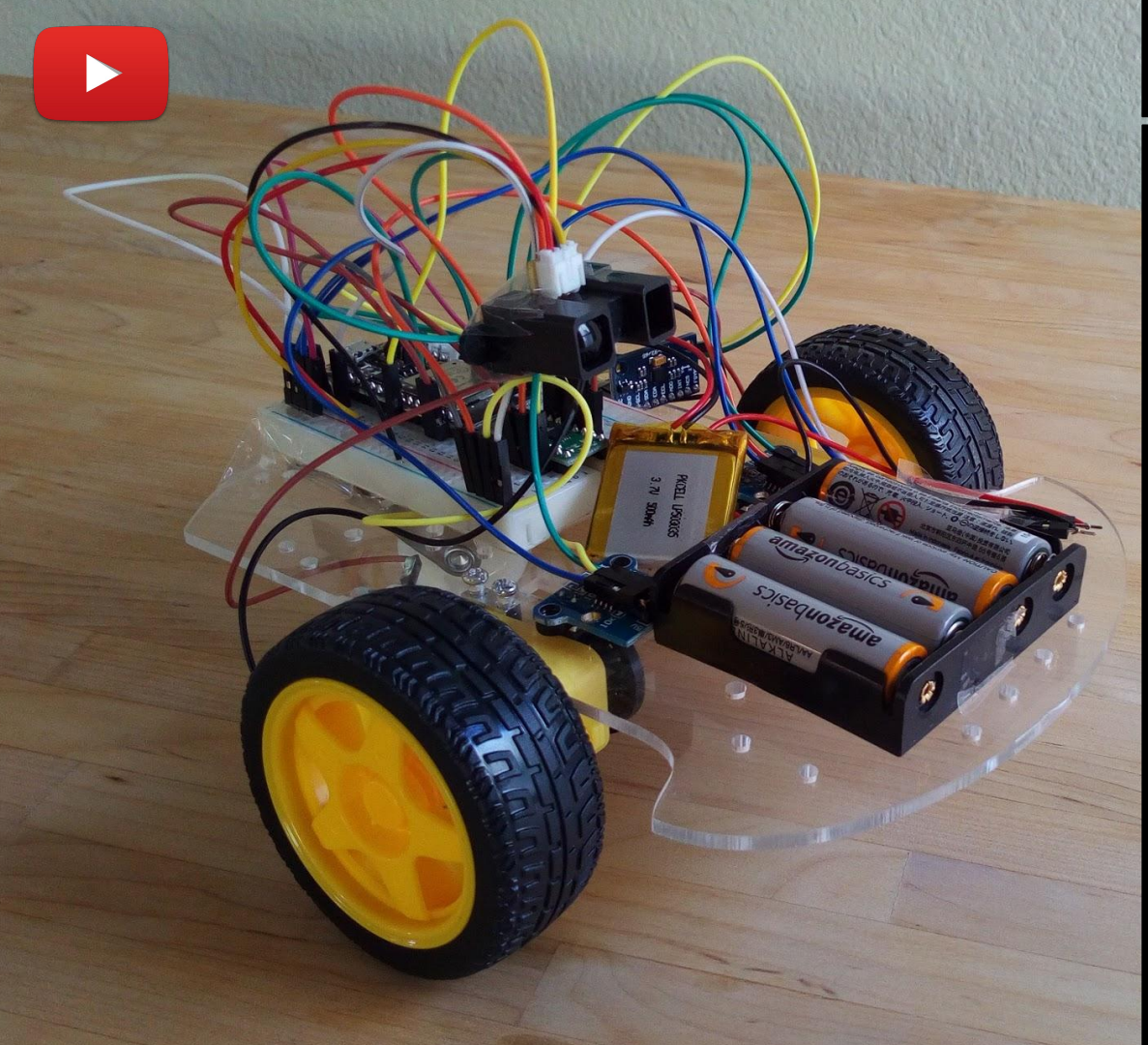
WALL-E 49

Andrew Shacker
Samantha Yang

EE49 S’18, Boser



Operation Flow

Connect to Internet, Broker

Initialize MPU Temperature, IR Distance Sensors

Turn robot 360° using PWM and encoder

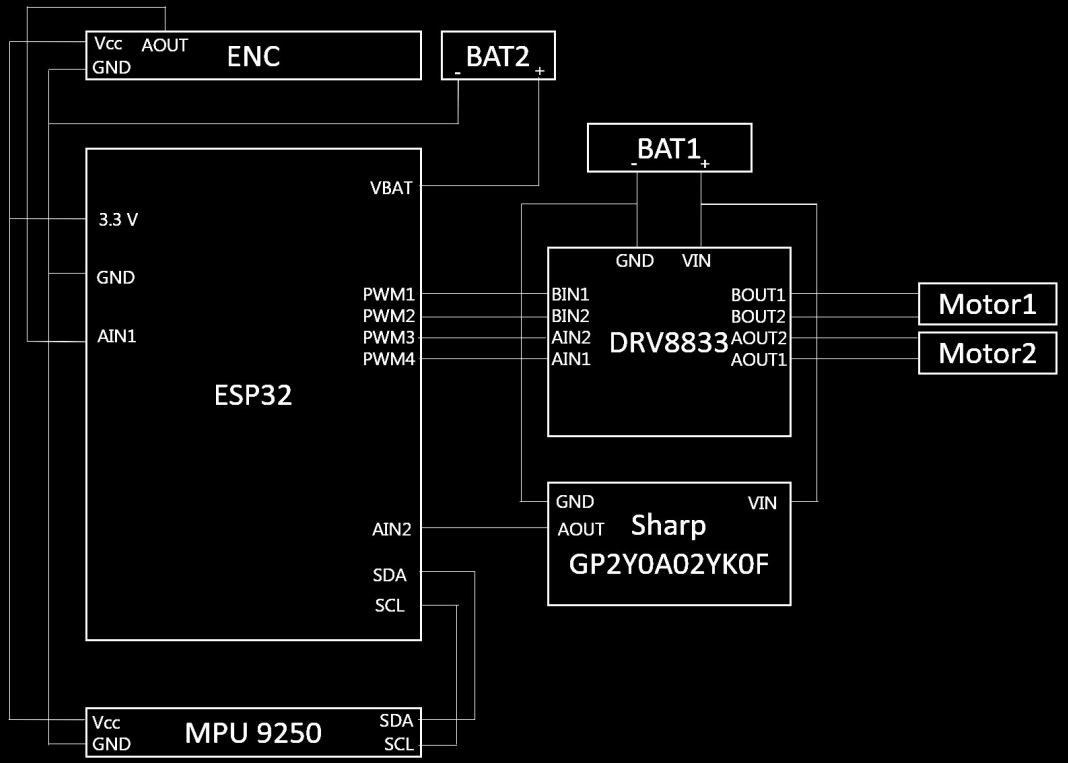
Obtain encoder count of largest obstacle-free distance and turn

Convert IR Sensor ADC code to distance (in.) and travel this length

Sample temperature using MPU and send to ThingSpeak via MQtT

Repeat robot movement using timer interrupt

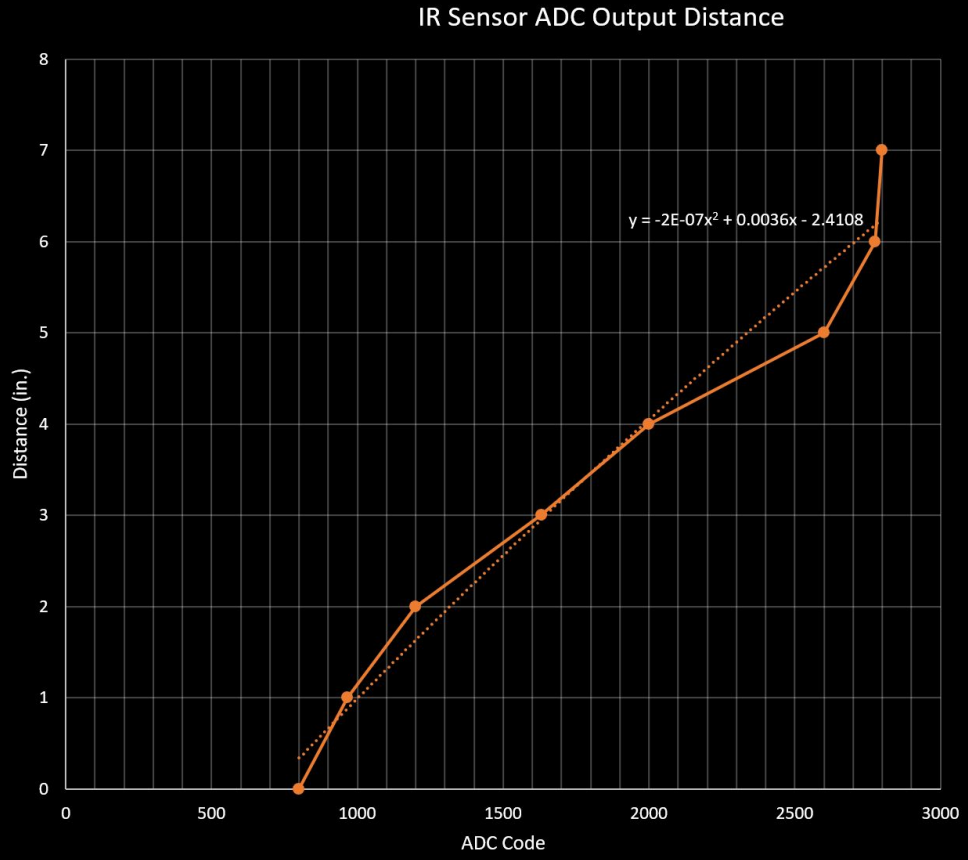
Circuit Diagram and Code

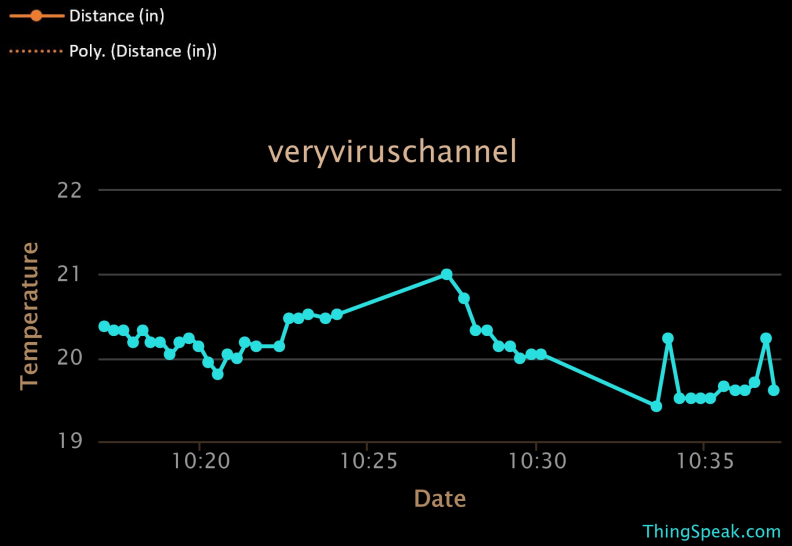



```python
from machine import Pin, I2C, ENC, Timer, ADC, PWM
from board import A10, ADC6
from board import A10, A5, A8, A6, A20, A21, SDA, SCL
from time import sleep
import time
from network import WLAN, STA_IF, mDNS
from mpu9250 import MPU9250
from plotclient import PlotClient
from mqttclient import MQTTClient

#Connect to WiFi
wlan = WLAN(STA_IF)
wlan.active(True)

wlan.connect('2WIRE030', '3922451880', 5000)

while not wlan.isconnected():
    print("Waiting for wlan connection")
    sleep(1)

print("WiFi connected at", wlan.ifconfig()[0])

# Advertise as 'hostname', alternative to IP address
try:
    hostname = 'veryviruswifi'
    mdns = mDNS(wlan)
    mdns.start(hostname, "MicroPython REPL")
    mdns.addService('_repl', '_tcp', 23, hostname)
    print("Advertised locally as {}.local".format(hostname))
except OSError:
    print("Failed starting mDNS server - already started?")

# start telnet server for remote login
from network import telnet

print("start telnet server")
telnet.start(user='veryviruswifi', password='1111')

# fetch NTP time
from machine import RTC

print("inquire RTC time")
rtc = RTC()
rtc.ntp_sync(server="pool.ntp.org")

timeout = 10
for _ in range(timeout):
    if rtc.synced():
        break
    print("Waiting for rtc time")
    sleep(1)
```

Code: github.com/astroshacker/Wall-E49

Distance Correlation and MQTT





Applications

Volcanic Research

Food Delivery

Medical Robots