Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

Team 14: RF Triangulation
Bi-Weekly Update 3

Josh Broyles, Brandon Stokes,
Jack Parkinson, Kathleen Hutchinson

Sponsor: Max Lesser
TA: Souryendu Das

Project Summary

Biologists have struggled being able to collect data on wildlife's location, habitat use, and breeding patterns without getting directly involved with the animal they're tracking.

Radio Frequency Triangulation allows a user to track a **known frequency** (such as a previously tagged animal) within the **triangulated area of three antennas** by using a motor to successfully **pinpoint the strongest signal.**

Helpful to study both invasive and threatened wildlife, our RF Triangulation system will focus on being able to accurately **track a transmitter within a 150 meter radius with >10% error.**

Integrated System Diagram

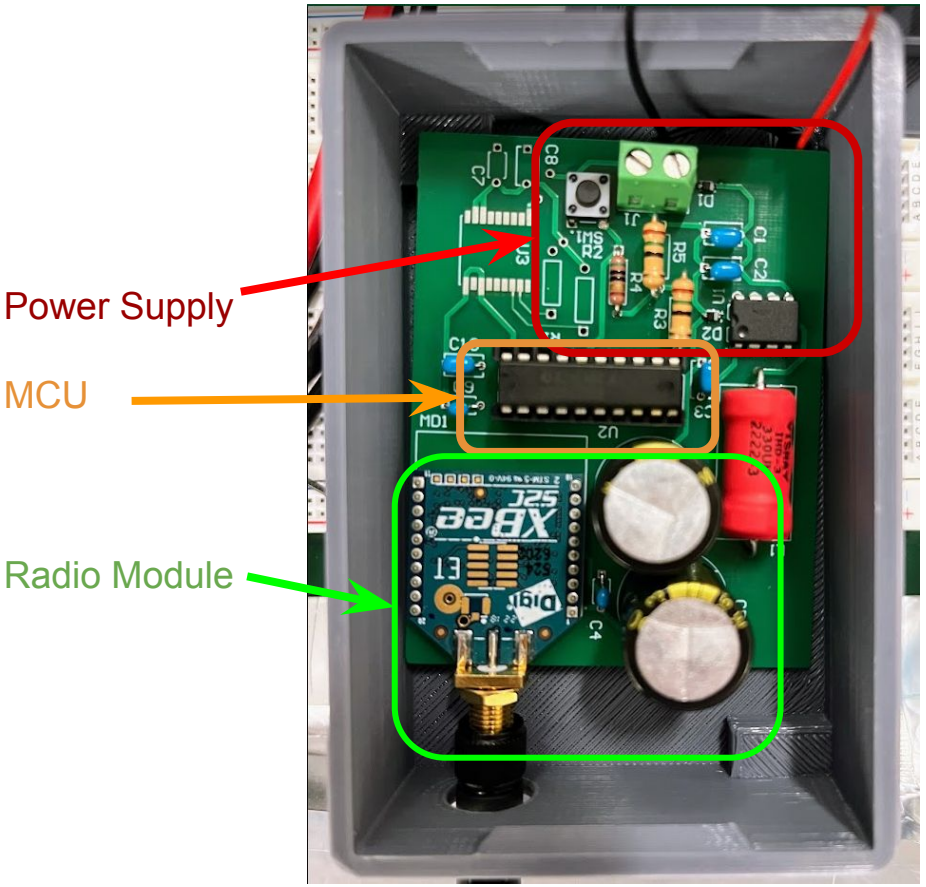
Ideally, our system will have one transmitter and three receivers.

The figure to the right shows a **completed transmitter**.

In the future, three receivers of *relatively the same form* will be created to connect to the transmitter.

The only differences between will be *three added modules* that are essential to the *stepper motor* and data transmission through *LTE*.

Due to power supply failure, the receiver is still in progress to be finished.



Project Timeline

The team will finalize and order parts/PCB

Finalize realistic goals with sponsor

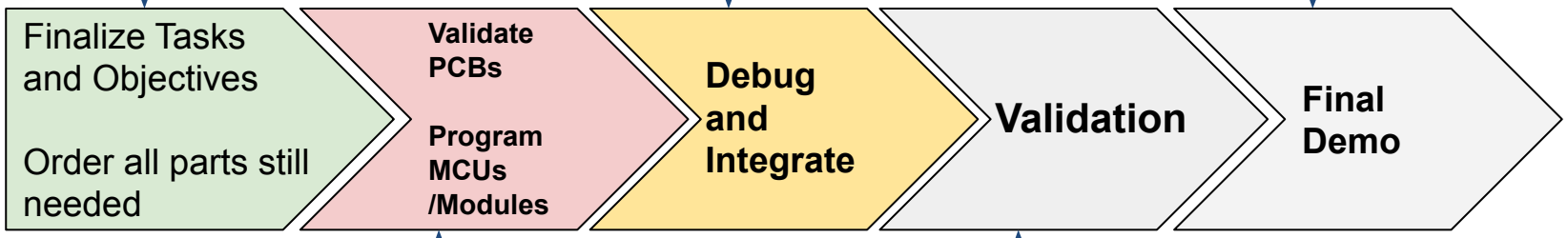
[Complete before February]

The team will debug any problems with code, power, etc. and begin integration

[March]

Once validation is complete with a working project, the team will complete a final demonstration

[Late April]



The team will finalize individual subsystems

[February]

Once fully integrated with all parts working, the team will begin validation testing.

[Late March through Mid April]

Completed

In Progress

Not Started

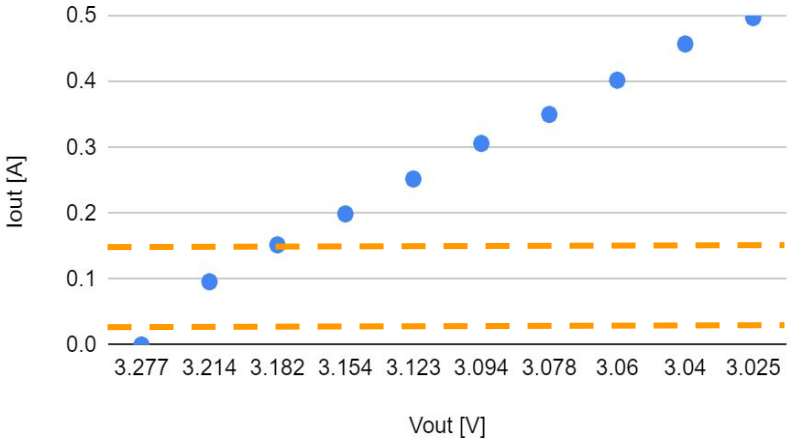
Transmitter

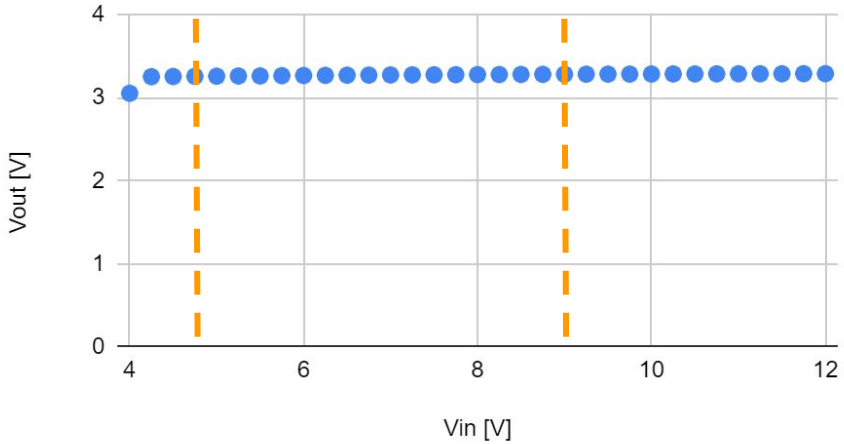
Josh Broyles

Accomplishments since last update 39 hrs of effort	Ongoing progress/problems and plans until the next presentation
Validated Transmitter Power Completed Radio Housing Working signal transmission to receiver side	N/A

Transmitter

Josh Broyles





Receiver: Antenna & Motor

Jack Parkinson

Accomplishments since last update 26 hrs of effort	Ongoing progress/problems and plans until the next presentation
Finalized Antenna Design Got Blinky code working on an offboard ESP32 Soldered on about half the PCB for testing	Write code for stepper motor based off of the blinky code Get red badge Buy parts for antenna Help Kat with the PCB problems and integration between our systems

Receiver: ESP32 & Modules

Kathleen Hutchinson

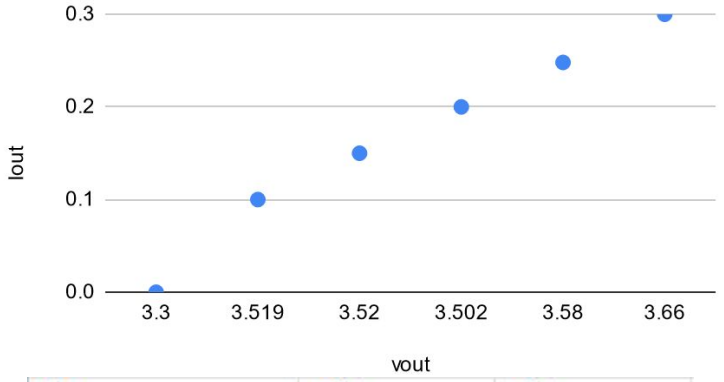
Accomplishments since last update 39 hrs of effort	Ongoing progress/problems and plans until the next presentation
Failed power validation Connected Josh's XBEE, to my XBEE, through ESP32, onto Pycom to Brandon	Buying parts PCB design part 2 Problems: buck converter inductor XBee output

Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Receiver: ESP32 & Modules

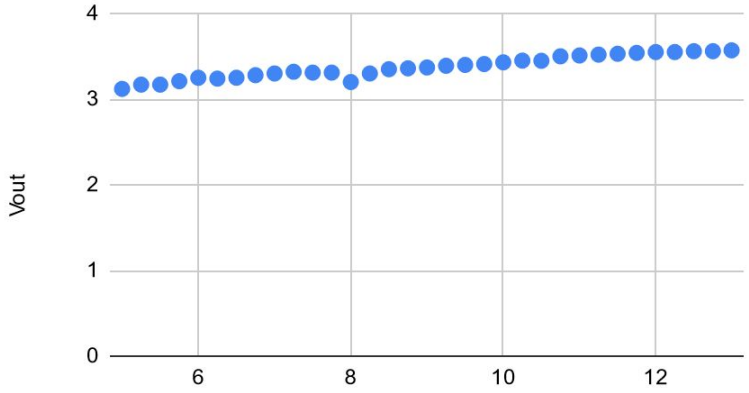
Kathleen Hutchinson

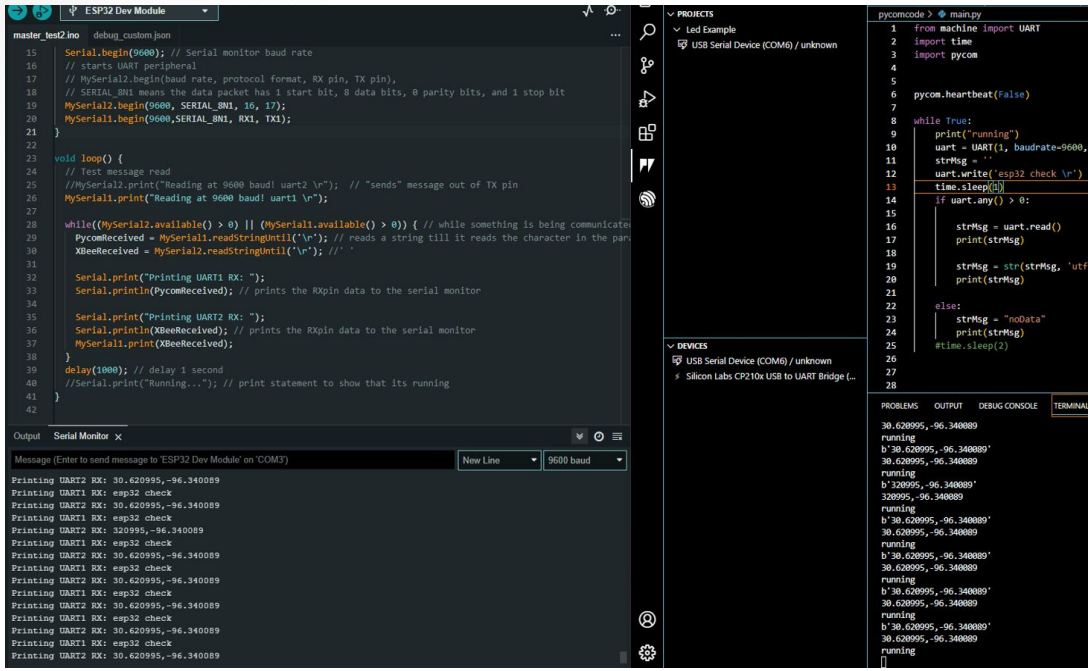
Receiver Load Regulation

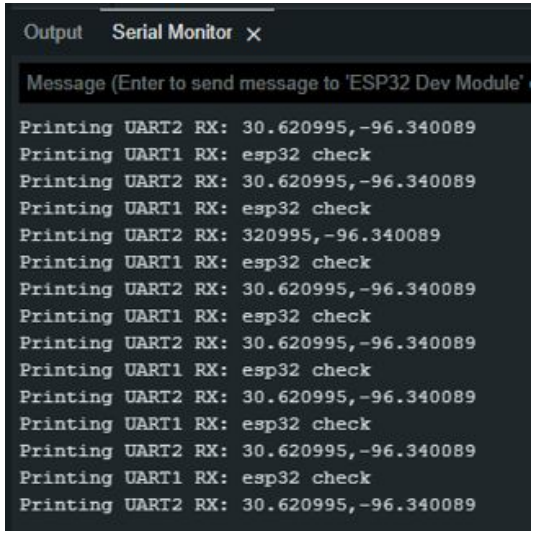


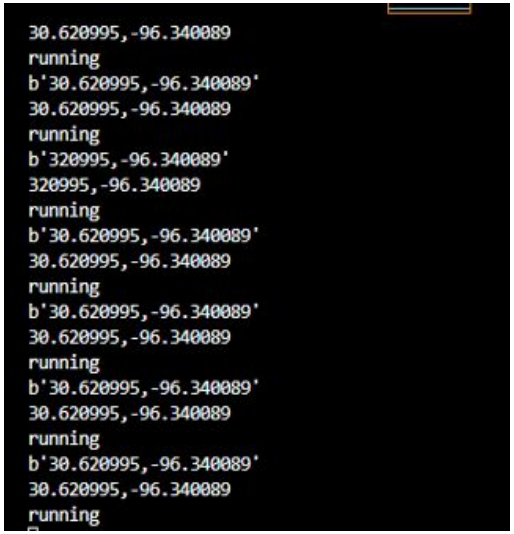
Vin	Vout	Iout
11.996	3.3	0
	3.519	0.1
	3.52	0.15
	3.502	0.2
	3.58	0.248
	3.66	0.3
	temp problem	0.35

Receiver Line Regulation









Database & GUI

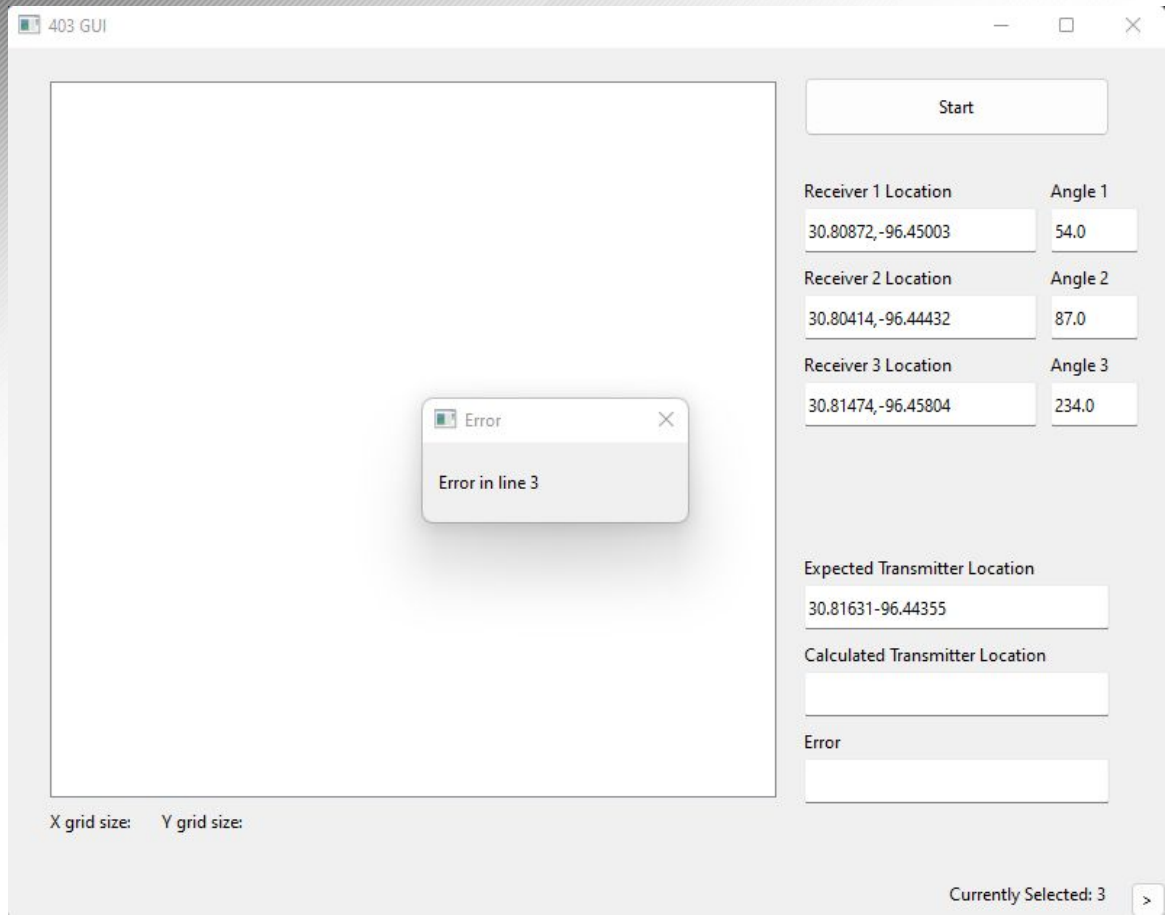
Brandon Stokes

Accomplishments since last update 21 hrs of effort	Ongoing progress/problems and plans until the next presentation
• Completed integration with Kathleen and the communication between our systems • Tested app error handling for receiver off angle and location	• Continue to work with transitioning from WiFi to LTE

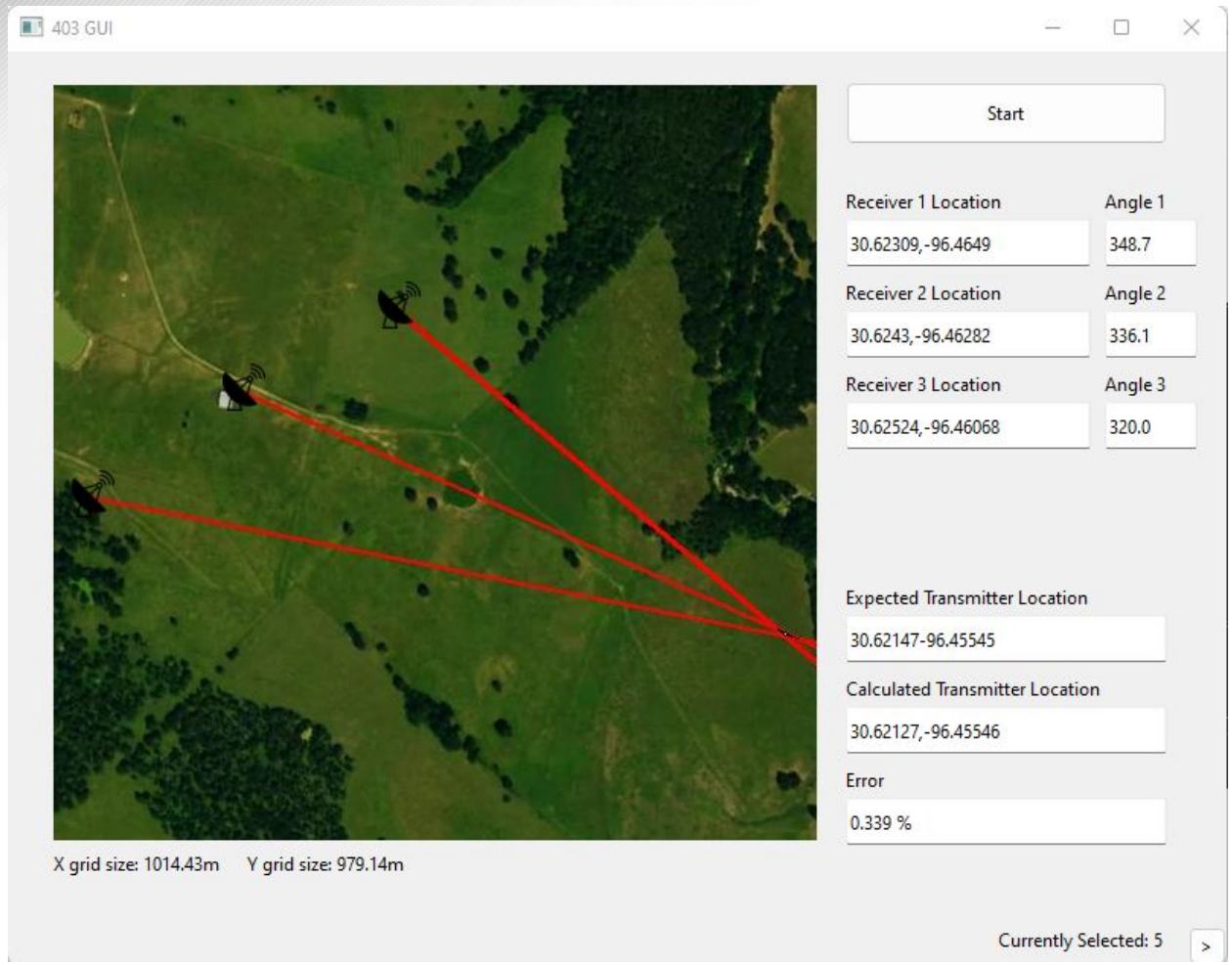
Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

Database & GUI

Brandon Stokes

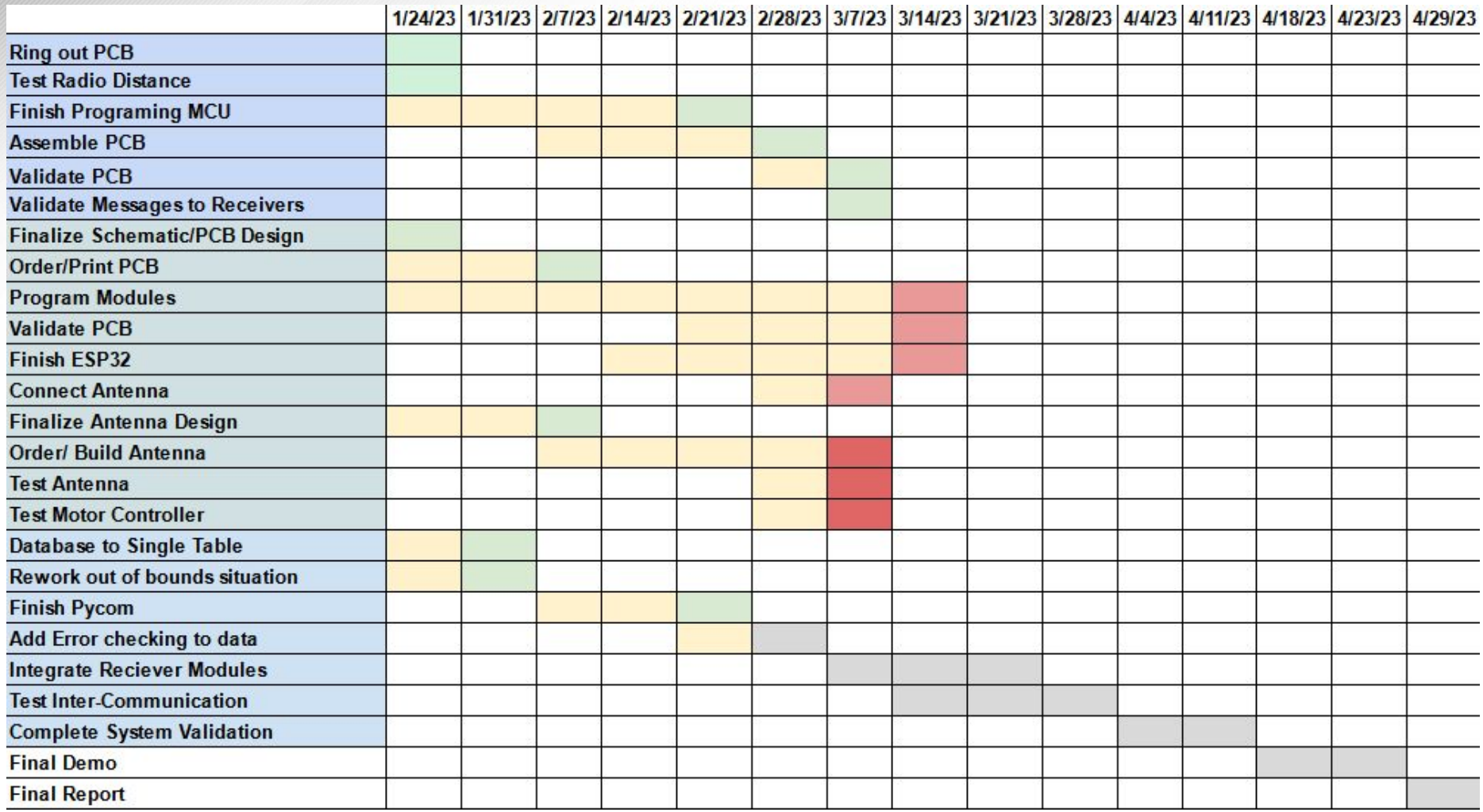



```python
from machine import UART
import time
import pycom


pycom.heartbeat(False)

while True:
    print("running")
    uart = UART(1, baudrate=9600, pins=('P21','P20')) # voor esp
    strMsg = ''
    uart.write('esp32 check \n')
    time.sleep(1)
    if uart.any() > 0:

        strMsg = uart.read()
        print(strMsg)

        #strMsg = str(strMsg, 'utf-8')
        #print(strMsg)

    else:
        strMsg = "noData"
        print(strMsg)
    #time.sleep(2)
```




```
Time taken for dataset 5  is  2.518
```

Execution Plan



Validation Plan

Paragraph	Test Name	Success Criteria	Methodology	Status	Responsible Engineer
3.2.1.1	LTE Stability	The LTE does not drop more than 1 time per 5 minutes and shall reconnect within 20s.	System is put into default operational state (tracking transmitter) and left to run for 30 minutes while Pycom tracks LTE connection	UNTESTED	Brandon Stokes Kathleen Hutchinson
3.2.1.2	Antenna Characterization	Physical antenna has a gain of at least 7 in the direction of the antenna.	Physical antenna will be made and tested inside a characterization chamber with the help of Professor Nowka.	UNTESTED	Jack Parkinson
3.2.1.3	Motor accuracy	The motor can turn with speed and precision while the carrying the weight of the reciever PCB and antenna.	After connecting the system to the motor, it will spin with vairying speeds and steps and be stopped to check accuracy and time.	UNTESTED	Jack Parkinson
3.2.1.4	System Connection	The time it takes to connect and transmit data between the GUI, transmitter, and receiver shall be 30s.	System runs start signal from application to receivers and back to application with data and timed using timer function into code.	UNTESTED	Brandon Stokes Kathleen Hutchinson
3.2.1.5	Operation Time	System operates continuously on battery power for 30 minutes.	System transmitter and receivers are put into default operational state and left to run for 30 minutes.	UNTESTED	Full Team
3.2.1.6	Detection Range	The detection range shall be an 150m radius from transmitter to a single receiver.	Receiver antenna will be place 150 meters from transmitter and be able to detect and step towards the signal transmitted.	UNTESTED	Josh Broyles Jack Parkinson Kathleen Hutchinson
3.2.2.1	Mass	The weight of the system shall be at max 27lbs.	Measure receiver unit with digital scale.	UNTESTED	Jack Parkinson
3.2.3.1.1	Input Voltage	The input voltage level for the ESP32 and MSP430 shall be 3.3V	Use E-Load to validate line and load regulation	TESTED UNTESTED	Josh Broyles Kathleen Hutchinson
3.2.3.1.2	Input Voltage (Motor)	The input voltage level for the DRV811PWPR (motor driver) shall be 12V.	Use multimeter to validate input voltage levels	UNTESTED	Jack Parkinson
N/A	Full System Demo	A user of system is able to accurately track the transmitter in an open space with a positional error of less than <10%.	System runs start signal to receivers which tracktransmitter's strongest signal with the motor, both stationary and moving, which sends data to GUI that outputs an accurate map with an error of <10% calculated through GPS points.	UNTESTED	Full Team

Thank you for your attention!

Feel free to ask us questions