emStart Sprint 2 Status Report

Members: Ivan Borra, Matthew Gasper, Matthew Grabasch, TJ Scherer, and Matthew Selph

Introduction Recap

Sprint 2 Activity

Remaining Tasks

Project Timeline

Introduction Recap

This is an emulating system for an existing small radio telescope (SRT). The SRT hardware and software are both open-source, developed by MIT Haystack Observatory.

The current system works, but can be improved. To improve it, we need to be able to receive signals at any given time. Unfortunately, this is not possible for the real SRT system as the signal comes from space and is not available on demand.

In order to make these signals available on demand, we need to do the following:

- Emulate the relative angle between an observer on Earth and an object in space
- Receive radio signals transmitted from the emulated object in space

Introduction Recap

Sprint 2 Activity

Remaining Tasks

Project Timeline

Earth Emulator Software

Python program capable of reading from a configuration file to load the data required for emulation

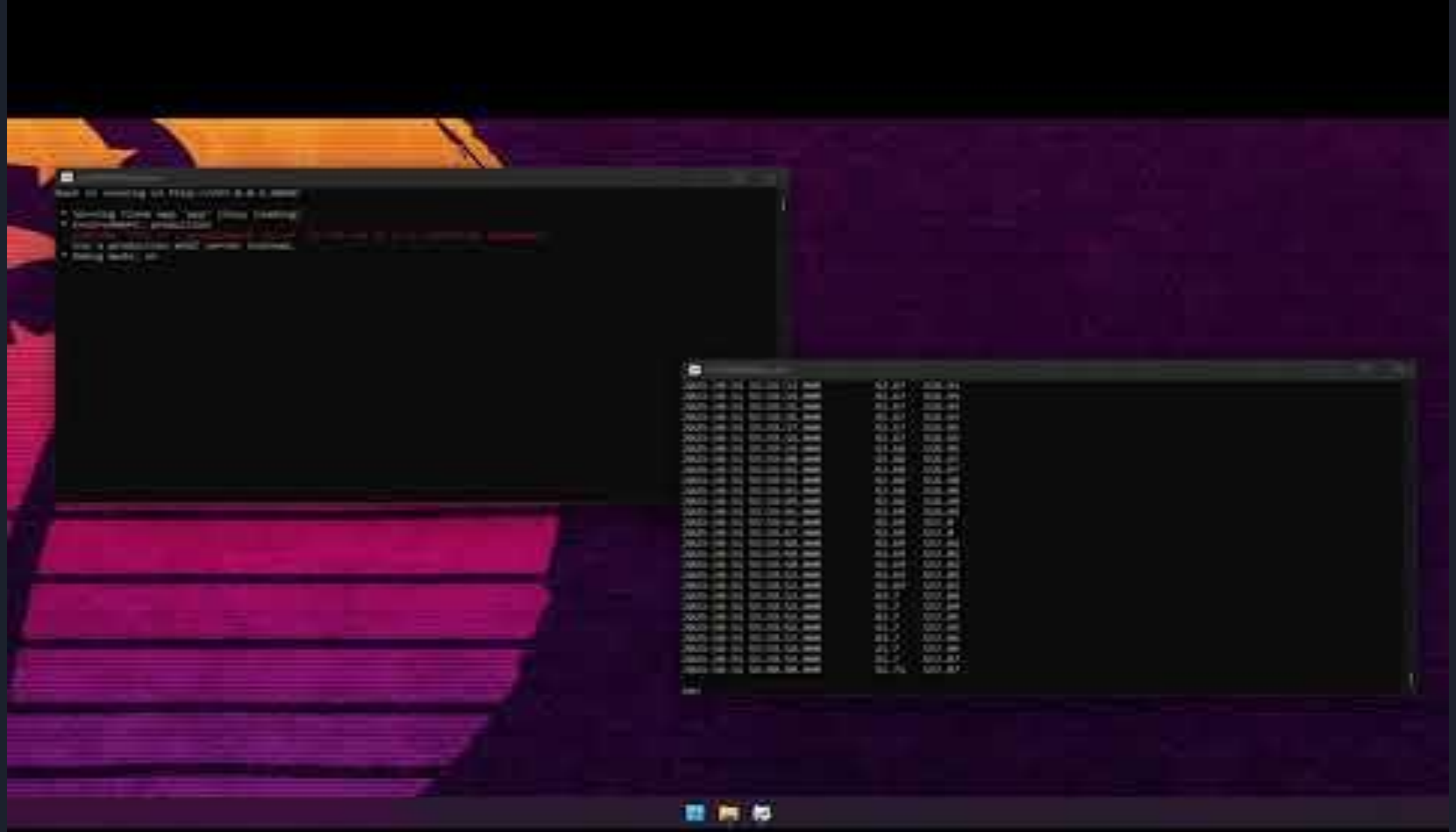
Once the emulation begins, the user has full control of time

- Pause
- Fast forward
- Rewind
- Skip

The emulator constantly broadcasts information to synchronize the other subsystems

The emulator also has a user interface which uses sockets to receive data and send commands

The user interface is hosted through Dash, a web based dashboard



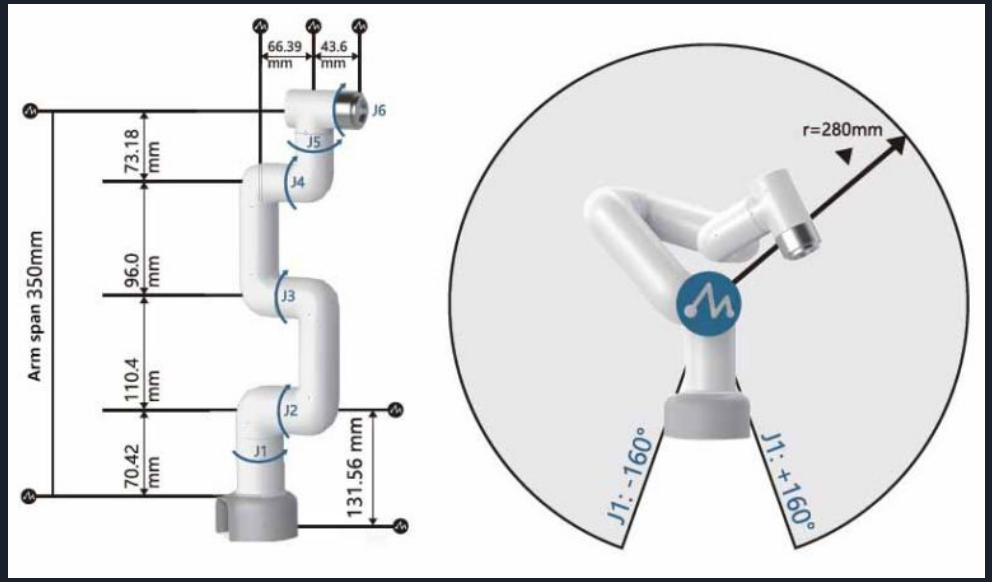
Earth Emulator Skeleton Code

Test plan goal:

- Create a coordinate system.

Test plan code:

- Set arm to default position
- Turn joint two 90 degrees
- Rotate arm on joint one 360 degrees
- Record arm coordinate and angle data



Ground Station Communication/Servo Movement

- Arduino and Raspberry Pi communicate through UART over a USB cable
- RPI sends Azimuth and Angle to arduino to implement
- Arduino checks for safe values and rotates ground station
- Feedback servo used to check for proper alignment

Future Plan

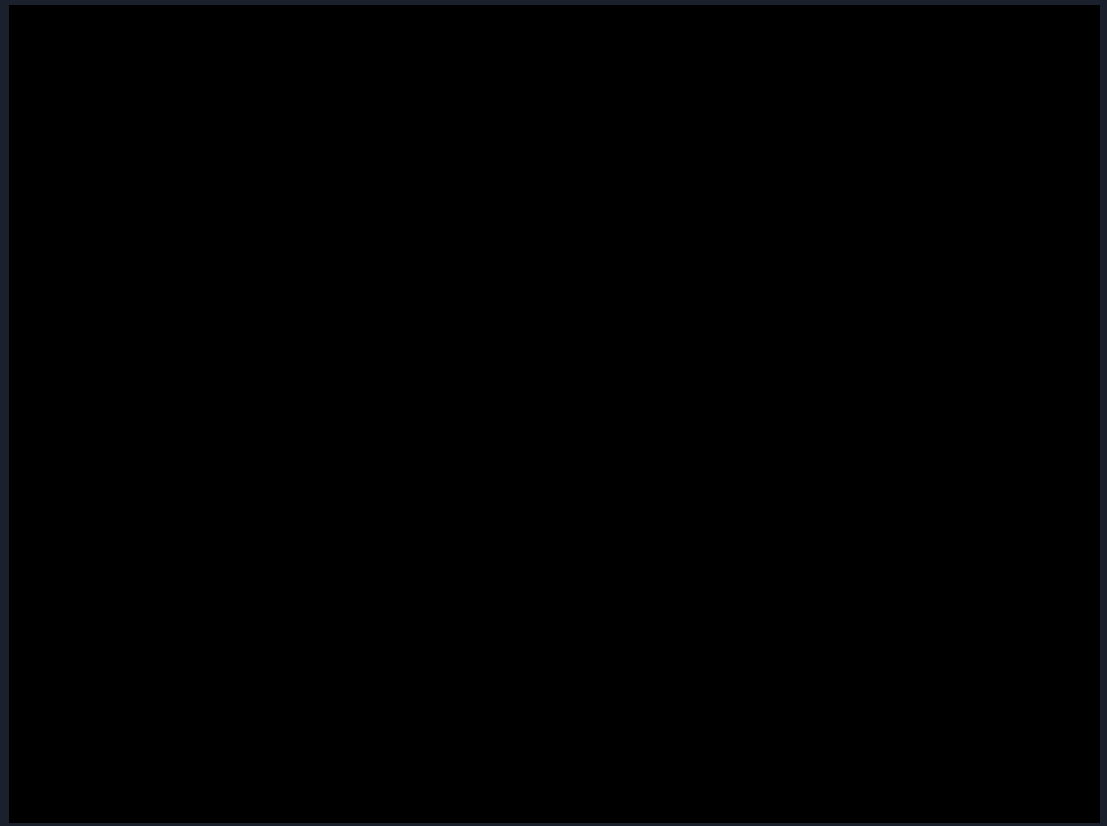
- Implement Hardware (Arrived Yesterday)
- Verify Software works properly
- Integrate RPI Module into overall software model

Arduino Code

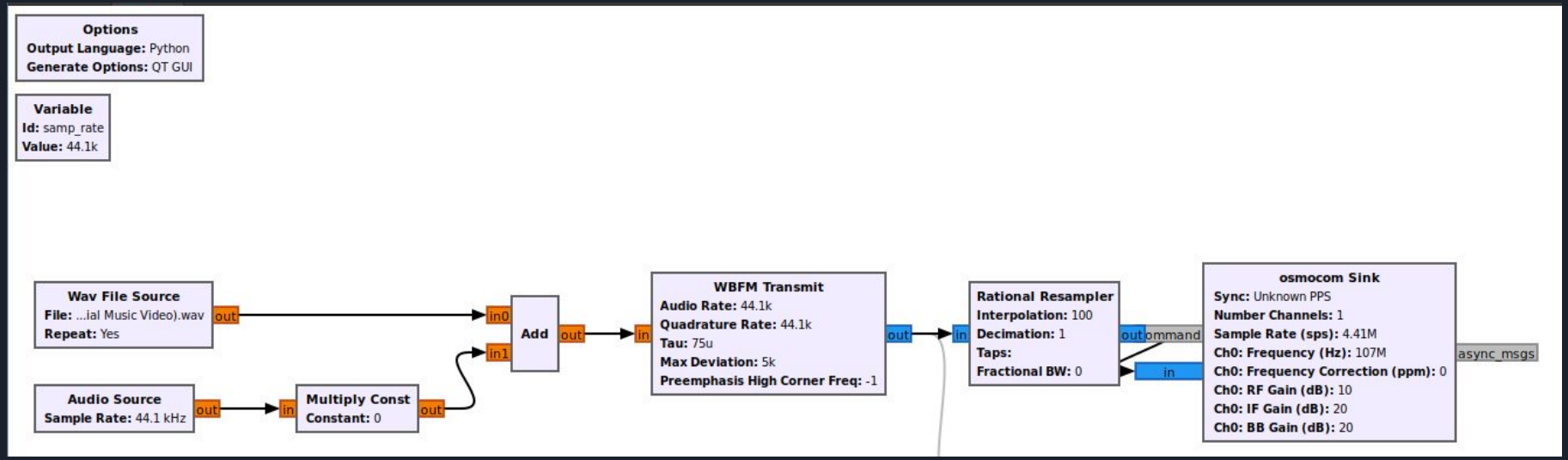
```
void setup() {
    // set servo control pin number
    azimuth_servo.setServoControl(AZIMUTH_SERVO_PIN);

    // set Kp to proportional controller
    azimuth_servo.setKp(1.0);

    //atatch servo to control pin
    angle_servo.attach(ANGLE_SERVO_PIN);

    Serial.begin(115200);
}

void loop() {
  // put your main code here, to run repeatedly:
  String az_ser = "";
  String an_ser = "";
  int run_cnt = 0;
  if (Serial.available() > 0) {
    az_ser = Serial.readStringUntil('z');
    //Serial.print("azimuth: ");
    //Serial.println(az_ser);
    an_ser = Serial.readStringUntil('n');
    //Serial.print("angle: ");
    //Serial.println(an_ser);

     //convert vaues from Serial port to int
    azimuth_set = az_ser.toInt();
    angle_set = an_ser.toInt();

    Serial.print("Azimuth: ");
    Serial.print(azimuth_set);
    Serial.print("\n");
    Serial.print("Angle: ");
    Serial.print(angle_set);
    Serial.print("\n");
    run_cnt++;
  }
```

```
  if (run_cnt >= 3)
  {
    if (((azimuth_set >= 0)||(azimuth_set <= 360))&&((angle_set >= 0)||(angle_set <= 90)))
    {
      //set_coord(azimuth_set,angle_set);
    }
  }

}

void set_coord(float azimuth, float angle)
{
  //assumed that North is 0 Degrees
  while((azimuth_servo.Angle() - OFFSET_FROM_NORTH) != azimuth)
  {
    azimuth_servo.rotate((azimuth - OFFSET_FROM_NORTH),1);
  }
  angle_servo.write(angle);

}
```

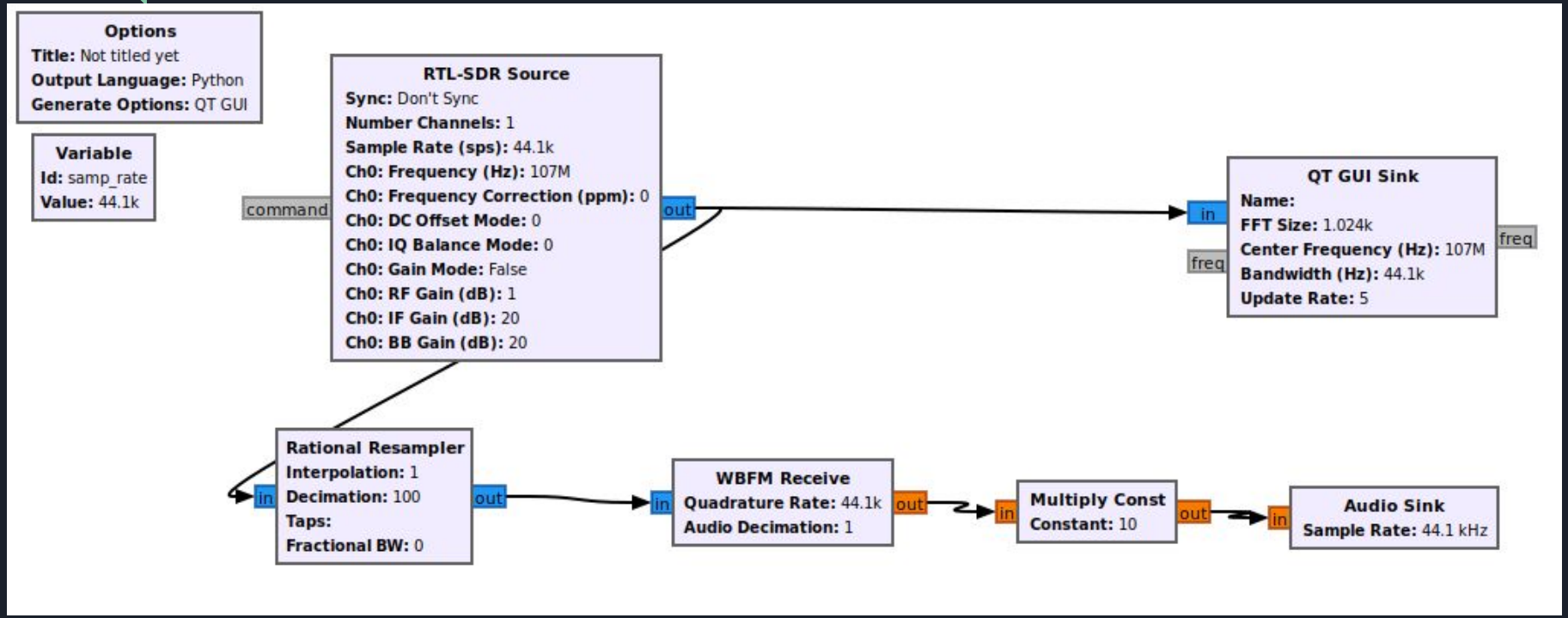
Demo



Radio Transmit Module



Radio Receive Module



Introduction Recap

Sprint 2 Activity

Remaining Tasks

Project Timeline

Remaining Tasks

Bug fixing, optimizing, and improving the Earth software

Integrating the robot arm skeleton code with the existing Earth software platform

Verifying communication between the Earth and the ground station

Verifying operation of the ground station antenna rotator

Mounting the ground station on the platform which will be mounted to the robot arm

Introduction Recap

Sprint 2 Activity

Remaining Tasks

Project Timeline

Project Timeline

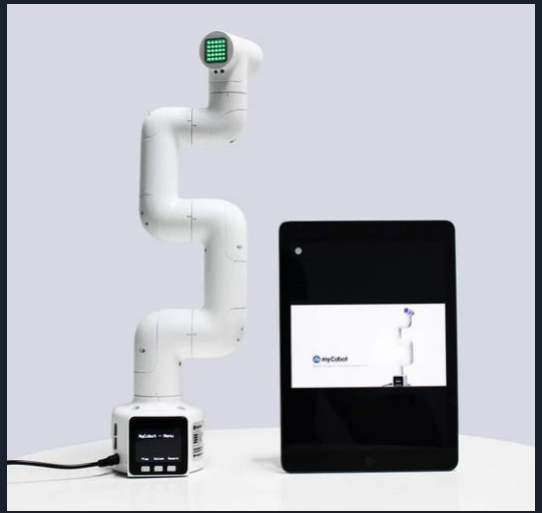
Going forward into sprint 3 is based off on the status of the hardware arriving

With Hardware:

- Integrating our different parts into the hardware and get basic functions working with more advanced parts being added should time permit.

Without Hardware:

- Integrating our software on the basic levels to get communication starting.



Questions?

Thank You!