# System Requirements Specification

emStart: A Satellite Transceiver and Small Radio Telescope Emulator

**Version 1.1 approved**

**Prepared by Ivan Borra, Matthew Gasper, Matthew Grabasch, TJ, Scherer, Matthew Selph**

**EECS Senior Design**

**10/26/2021**

# Table of Contents

# Revision History

| Name | Date | Changes | Version |
| --- | --- | --- | --- |
| All | 9/28/21 | Document creation | 1.0 |
| All | 10/26/21 | Document revision | 1.1 |

# 1. Introduction

## 1.1 Purpose

The goal of this project is to first create a model of a real-world antenna for data collection and then to move that implementation to an already existing full-scale antenna. The existing implementation of the antenna is a fixed arm that does not move with the Earth's rotation. The purpose of the Antenna is to capture satellite data from a satellite that will be moving with respect to the antenna. We aim to implement a robotic arm that can best match the rotation of the Earth, and therefore extend the capture time frame of data from the satellite, helping with debugging of the received signal.

This emulator will include the construction of a mockup satellite that will transmit data, and a model ground station that will receive that data, and attempt to move with the expected movement of the satellite.

## 1.2 Document Conventions

*N/A*

## 1.3 Intended Audience and Reading Suggestions

This document is meant for 3 entities all listed in the SCRUM Framework, as well as the senior design faculty. These include the product owner, which in this case is Dr. Liu, the scrum master, which was selected to be Matthew Gasper, and the remaining senior design team. Additionally, the senior design faculty may use this document to track the progress of the team.

## 1.4 Product Scope

There are 4 pieces of software being developed for this project. The first is the transmitter software. This aims to emulate a "satellite" and send out signals to be captured by the receiver. The second is the receiver software, which aims to interpret the data being sent to it to verify the accuracy of the tilt function that is described next. The third is the tilt function for the ground station, which has a goal of making the receiver always point in the direction of the transmitting "satellite". The final piece of software is the earth rotation emulation which will be installed on a robotic arm to simulate the movement of the ground station on Earth.

## 1.5 References

N/A

# 2. Overall Description

## 2.1 Product Perspective

emStart will be the first in an eventual series of products that will be open-sourced for others to create themselves. The goal of the project is to create a functioning prototype that can be refined and simplified to make it accessible as a science fair project. Overall, it will be a self-contained product with multiple variants depending upon the users' resources, ranging from very simple and cost-effective to more complex, more precise, but also more expensive. Providing these different tiers for the same product will make it much more accessible and adaptable to a variety of experimentation, research, development, and debugging efforts.

## 2.2 Product Functions

- The product must use astronomy data to determine the position of the Earth
- The product must use astronomy data to determine the position of the ground station antenna
- The product must be able to receive radio signals at the ground station
- The product must be able to send radio signals from the system in space

## 2.3 User Classes and Characteristics

This product will have a variety of user classes dependent upon the application. Keeping in mind the goal of open-sourcing the project and simplifying it to make it more accessible, there will be many different levels of expertise required for each tier. At the top, there will be the researchers and developers using the product as a test bench for their own designs. There will be a tier in between for those who do not require the same level of precision but still have a useful application for the design. At the lowest level, students and tinkerers will be able to use the design on a smaller scale which will not require the same levels of precision and accuracy.

Beginning at the highest level, with the most complex design, researchers and developers will use this design to its fullest. It will be used frequently to run tests on hardware and software in the user design, likely utilizing the majority of the functions the product has to offer. This application will require high precision, so technical expertise and experience will be critical to understanding the intricacies of the product to achieve the most desirable results.

In between, some applications may require some of the features but not as much precision, or vice versa.

On the lowest level, this product should be accessible to students creating a science fair project and tinkerers looking for something to do. With this will come significantly less frequent use, less access to complex functions, and less expertise. In this scenario, the product will serve as a proof of concept rather than a precise tool for conducting research.

### 2.4 Operating Environment

The hardware platform will be custom-built for this project but will consist of some standard hardware such as the myCobot robotic arm, Raspberry Pi 4, and HackRF One. The software for each of these will use manufacturer programming methods. However, in the case of the antenna movement, an Arduino will be used to directly control the servos. The software must coexist with the astronomy data which will be streamed into the software to determine the Earth and antenna positions. It will also need to interface to the myCobot API to allow control of the robotic arm for the Earth.

### 2.5 Design and Implementation Constraints

The options available to the developers will be limited by the hardware and by the external software required to run the entire system. Hardware limitations will mostly limit the level of precision that the system is able to achieve when translating the astronomy data into a physical emulation. The software will be able to achieve much higher precision than the hardware itself due to the inherent limitations of the servos selected for the project. As far as software limitations are concerned, our software will only be as capable as the interfaces to astronomy data and the robotic arm.

### 2.6 User Documentation

| | | |
|---|---|---|
| System Design Document | Revision 1.1 | 10/26/21 |

### 2.7 Assumptions and Dependencies

It is assumed that we will be able to achieve within one degree of precision in the hardware, and if that assumption is incorrect that may affect our software design, as it may require more mitigating factors to prevent uncertainties. This will help ensure that the emulator will be able to handle tests of longer duration which will make it a much more useful tool for research, development, and debugging.

## 3. External Interface Requirements

### 3.1 User Interfaces

The user shall have access to an interface that allows for basic control and the current status of different aspects of the antenna and signals being received.

3.1.1 Diagram showing current longitude angle of the antenna.

3.1.2 Diagram showing current latitude angle of the antenna.

3.1.3 Button allows the user to stop the movement of the antenna.

3.1.4 Button that allows manual control of the antenna for testing.

3.1.5 Diagram showing the current status of alignment for the antenna, depicting whether it is tracking the signal or not.

## 3.2 Hardware Interfaces

The hardware for the emulator will consist of three different base systems working in tandem, these systems being the Ground Base, Earth, and the Satellite.

3.2.1 The Earth System shall be comprised of a computer connected to a 3-DOF robotic arm via a USB cable.

3.2.2 The Ground Station shall consist of a computer connected to the antenna microcontroller via a USB, which is used to control the 2-DOF antenna.

3.2.3 The Ground Station shall work independently from the Earth System, meaning that the antenna must compensate for the movement of the “Earth”.

3.2.4 The System in Space shall work using a computer connected via USB to an SDR and GPSDO with a fixed antenna, sending the frequency towards the Ground Base.

## 3.3 Software Interfaces

The Ground station will consist of systems that move the antenna and work with the given astronomy data. This data collected will be used to convert into which position the antenna will be facing to properly receive the signal from the system in space. The system in space will only be sending signals, which will then be received by the antenna.The Earth system shall use the 6 DOF arm to move the “Earth” in different positions, thus causing the Ground Base to be utilized to keep the antenna aligned with the space object.

3.3.1 The Ground Station shall control a 2-DOF antenna.

3.3.2 The antenna shall be utilized using a microcontroller connected to a computer via USB.

3.3.3 The microcontroller will be fed the astrological data, which then can be converted into movements for the antenna.

3.3.4 The software-defined radio will be used to communicate analog signals to the Ground Station.

3.3.5 The “Earth” will be fully controlled by a computer, working without reference to the other systems via a 3-DOF robotic arm connected to a computer via USB.

## 3.4 Communications Interfaces

3.4.1 The Ground Base shall track the analog signal that is being transmitted by the System In Space.

3.4.2 The Ground Base shall use the astronomical program to aid in tracking the signal by converting the data into movements for the 2-DOF Antenna.

3.4.3 The Earth System shall have no communication with the Ground Station or System in Space.

# 4. System Features

Section 4 details a more specific list as put forward in section 2.2. This goes along with the functions associated with Earth Emulation in 4.1, Rotator Emulation in 4.2, Radio Communications in 4.3, and Mission Viewer in 4.4.

## 4.1 Earth Emulation

### 4.1.1 Description and Priority

The user shall be able to define a specified astronomical situation/path for the earth emulation to copy. The user will import a set of astronomical data that they wish to emulate, this data when imported will be used for emulation once the rotator emulation and communications of radio receiver and transceiver are also ready to execute.

### 4.1.2 Stimulus/Response Sequences

Once the user imports the correct astronomical data for emulation and selects “ready” the system will be on a waiting period until the other systems confirm their ready status. After emulation, the system will then return to a starting position for further emulations.

### 4.1.3 Functional Requirements

4.1.3.1 The system shall use a six degree of freedom robotic arm to emulate “Earth.”

4.1.3.2 The system shall have an attachment point to mount a flat plane to mount a rotator emulator system

4.1.3.3 The system shall allow imputed astronomy data to then emulate earth accordingly.

4.1.3.4 The system shall use a personal computer to control the system in its entirety.

## 4.2 Rotator Emulation

### 4.2.1 Description and Priority

The user shall be able to define a specified astronomical source for which the rotator will track, this information with a known earth positioning will be used to emulate a rotator’s movement when tracking.

### 4.2.2 Stimulus/Response Sequences

Once the user imports their specified astronomical source via initial azimuth and elevation followed by selecting “ready” the system will be on a waiting period until the other systems confirm their ready status. After emulation, the system will then return to a starting position for further emulations.

### 4.2.3 Functional Requirements

4.2.3.1 The system shall have an attachment point to mount a patch antenna.
4.2.3.2 The system shall have a mounting point for a raspberry pi 4 in order for it to be then routed to a hack RF.
4.2.3.3 The system shall have a mounting point for a hack RF in order to support the system.
4.2.3.4 The system shall have a movement ability of two degrees of freedom in order to emulate the actual system.
4.2.3.5 The system shall have a mounting point in which to attach itself onto a wooden plane.
4.2.3.6 The system shall allow imputed astronomy coordinates in order for it to emulate said tracking.

## 4.3 Radio Communications

### 4.3.1 Description and Priority

The system shall communicate between a patch antenna mounted on the rotator emulator's endpoint, from a transmitting free form radio transmitter consisting of a simple wire antenna connected to an SDR and mini computer.

### 4.3.2 Stimulus/Response Sequences

Radio communications are all done automatically during the emulation process with no user input required.

### 4.3.3 Functional Requirements

4.3.3.1 The system shall use a patch antenna in order to receive electromagnetic signals.
4.3.3.2 The system shall use a software-defined radio in order to allow for easier control over incoming frequencies.
4.3.3.3 The system shall use a personal computer to take in signals coming from a software-defined radio in order to analyze them.
4.3.3.4 The system shall be mounted atop a rotator emulator in order to perform its operations.
4.3.3.5 The system shall transmit at a frequency of 1.4ghz in order to emulate that of an actual received frequency of a ground station.
4.3..3.6 The system shall receive a frequency of 1.4ghz in order to emulate that of the actual frequency transmitted from a satellite.

### 4.4 Mission Viewer (GUI)

#### 4.4.1 Description and Priority

The Mission Viewer will be the front end of the entire system allowing for a simplified view of the entire system. It will provide the means to import astronomy data and other such information for emulation. This includes the emulation of the earth in 4.1 and of the rotator in 4.2 furthermore, a readout of radio communications will also be displayed within the mission viewer.

#### 4.4.2 Stimulus/Response Sequences

During the operation of the emulation station, a readout of radio communications and up-to-date positional data will be provided. This on top of the current operation being completed will also be displayed.

#### 4.4.3 Functional Requirements

4.4.3.1 The system shall visually represent the physical disposition relating to the rotator emulator.
4.4.3.2 The system shall visually represent the physical disposition relating to the earth emulator.
4.4.3.3 The system shall provide a visual for the current state of the overarching platforms attached to it being the earth and rotator emulator along with the communications system.
4.4.3.4 The system shall provide an ability to import astronomy data for it to be then sent to the earth and rotator emulator for emulation.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

5.1.1 The system shall be able to carry the antenna system on the arm.
5.1.2 The system shall receive data into the antenna system.
5.1.3 The system shall rotate the joints so the antenna is pointing to the satellite.

### 5.2 Safety Requirements

5.2.1 The system base should be secured to table
5.2.2 While the system is operational the users shall maintain a safe operating range.

### 5.3 Security Requirements

5.3.1 The system shall not be left connected unsupervised to the internet.
5.3.2 The system shall be monitored while connected to the internet.

### 5.4 Software Quality Attributes

5.4.1 The antenna shall not exceed 250 grams.
5.4.2 The ground station software shall display the correct overlay selected by the user.
5.4.3 The system shall be able to emulate the earth's rotation.

### 5.5 Business Rules

5.5.1 The system shall use astronomical data from astroPy
 5.5.1.1 The system shall use the astronomical data to adjust the arm

# 6. Other Requirements

# Appendix A: Glossary

SRT - small radio telescope
SDR - software-defined radio
GPSDO - GPS disciplined oscillator
DOF - degrees of freedom
SRS - System Requirements Specification
SMA - SubMiniature version A

# Appendix B: Analysis Models

N/A

# Appendix C: To Be Determined List

The position of wiring for the SRT.