DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

ARCHITECTURAL DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
SUMMER 2019

TEAM ROCKET
ROCKET BOARD

DARION ADAMS
RAHASYA CHANDAN
ALEXANDER PAVLAK
NATHAN REAGAN
BRIANCA WASHINGTON

Revision History

Revision	Date	Author(s)	Description
0.1	08.08.2019	AP	document creation

Contents

List of Figures

List of Tables

1 Introduction

Rocket Board is a system that collects flight data and provides tracking capabilities to model rockets. Rocket Board will enable users to gather flight information about their rocket and provide GPS tracking capabilities for rocket recovery.

Rocket Board is a sensor suite intended collect data on board a model rocket during flight. Rocket Board will be accompanied by a separate base receiver to receive flight data live. The receiver will then re- transmit the data to a companion mobile app which will provide data visualization and GPS tracking capabilities. The data collected from the sensor suite can be used to analyze flight performance and inform future launch decisions.

Rocket Board is intended for use by hobby model rocketeers as well as in STEM classrooms. Model rocketeers will gain access to a wealth of information about the performance of their rocket previously unavailable to them. Educators in STEM classrooms will gain a fun and interactive tool to help teach concepts of aerospace engineering and data science.

2 System Overview

The Rocket Board will collect and send data at regular intervals from a model rocket while in flight. The data will be collected from various sensors on the product and will be transmitted over FM radio. A USB receiver connected to an Android smart phone will receive the data and will display it on the accompanying application along with visualization of the flight path. A GPS module in the app will allow the user to find the location of the rocket for easier retrieval. Rocket Board will not give constant real time data, nor will it directly guide a user to their rocket. The whole system will involve three primary parts: the physical board connected to the model rocket, the FM receiver, and the accompanying application.

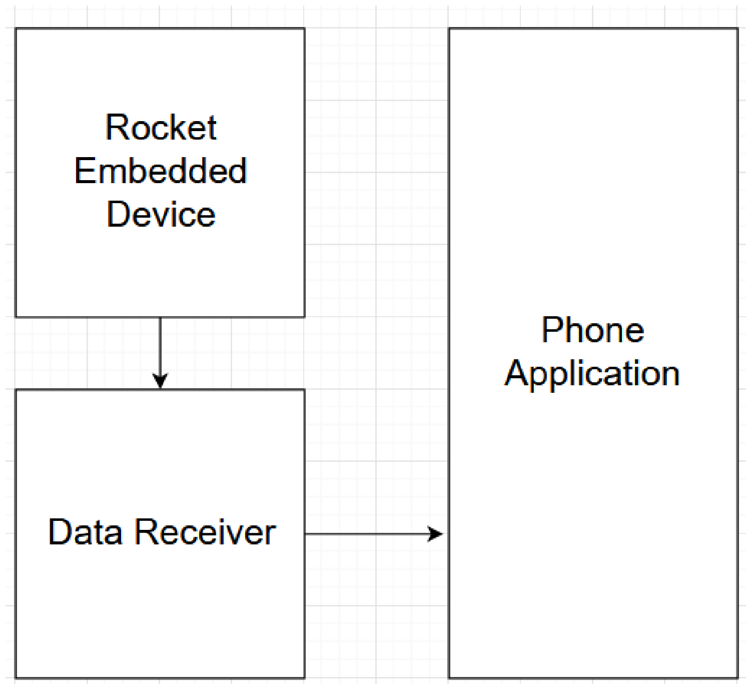

Figure 1: architectural layer diagram

2.1 Rocket Embedded Device

The embedded device serves to collect data and transmit the packaged data to the local receiver. The inertial measurement unit (IMU), GPS tracker, and altimeter collect data and send it to a central microcontroller. The microcontroller then validates and interprets the data from the raw voltage values to be sent to the FM transmitter. This does not include filtering of any kind, as the data will be interpreted later by the android application. The final packaged data will be sent to the FM transmitter to be received by the local receiver.

2.2 Data Receiver

The local FM receiver exists purely to receive and transmit data from the rocket to the phone application. Data will be streamed straight from the FM receiver to the bluetooth transmitter, which contains its own protocol to be used with the Android application. No microcontroller exists between the two devices as data is streamed via I2C.

2.3 Phone Application

The phone application receives, interprets, and displays the data to the end user. Once the data is received through the bluetooth receiver, the raw data is stored. This data is then circularly filtered to provide a singular data measurement for each aspect at a rate of four times per second. The data is then used in two ways. In the first, the data is shown as a visual meter for properties such as pitch, roll, etc. This is to give a strict visual to the user for interpreting the rocket's current properties. This is then also used to control the rocket's 3D visualization within the application. The second aspect is GPS tracking. The live coordinates will be displayed on the application, but once the rocket's flight is completed, Google Maps' API will be used to track the rocket's end position through exporting the coordinates. The user will then be directed to the rocket for retrieval.

3 Subsystem Definitions & Data Flow

For the rocket embedded device layer, the various sensors gather input from the rocket during flight, and send them to the controller subsystem. The controller interfaces with the transmissions subsystem, it validates and interprets the data that will be sent. The transmissions subsystem transmits the data to the local receiver over an FM transmission.

For the data receiver layer, the FM Reception subsystem receives the data from the FM transmission system, and interfaces with the the Bluetooth Transmission subsystem. The Bluetooth Transmission sends the data straight to the phone application.

For the phone application layer, the data reception subsystem interfaces with the Bluetooth transmission to receive the complete packets and the current connection to the data receiver. The packets are send to data processing to be stored, filtered, and prepared to be displayed to the user. The data is then visualized in different ways on request in the data visualization subsystem which handles the user interface.

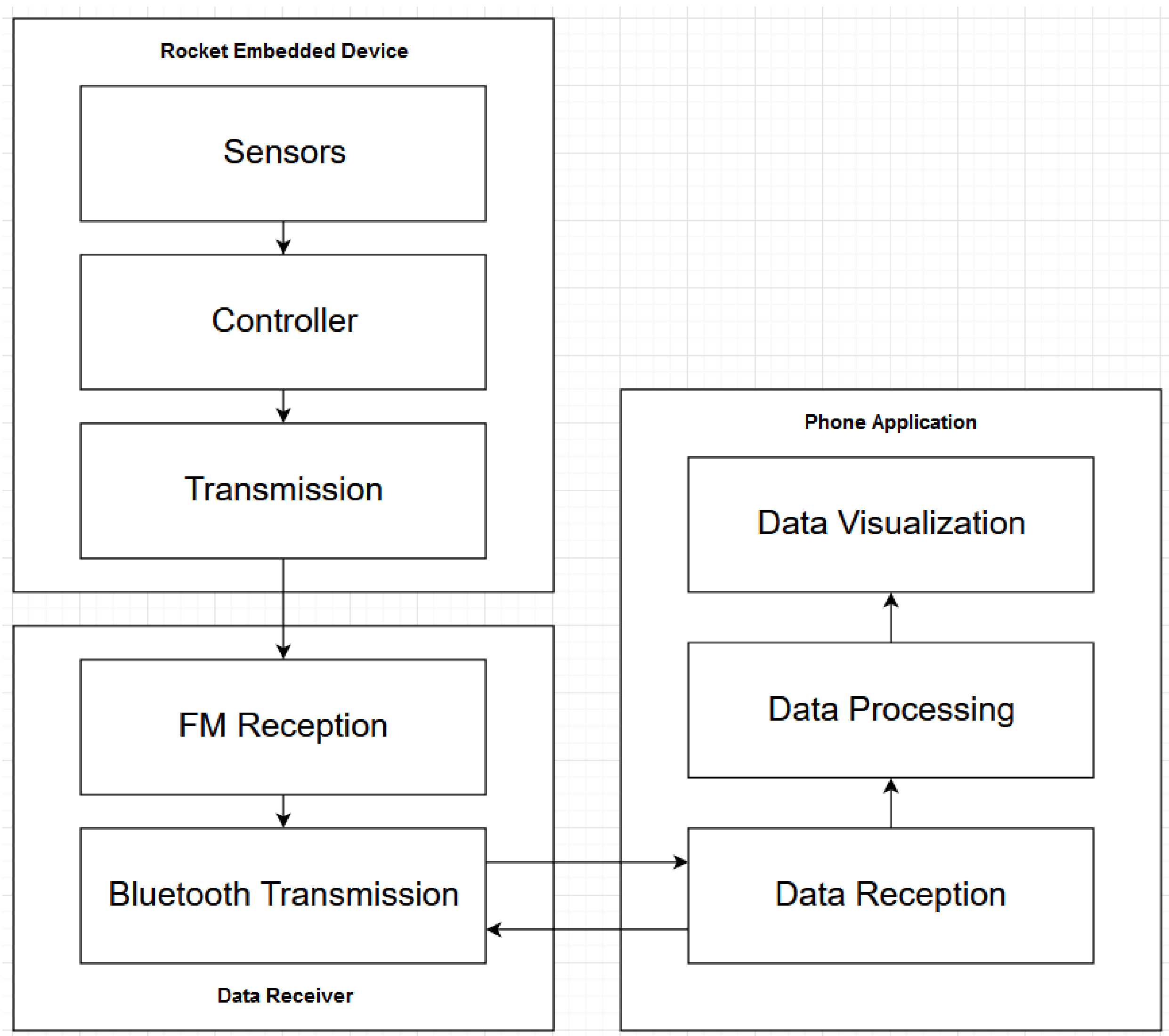

Figure 2: data flow diagram

4 Rocket Embedded Device

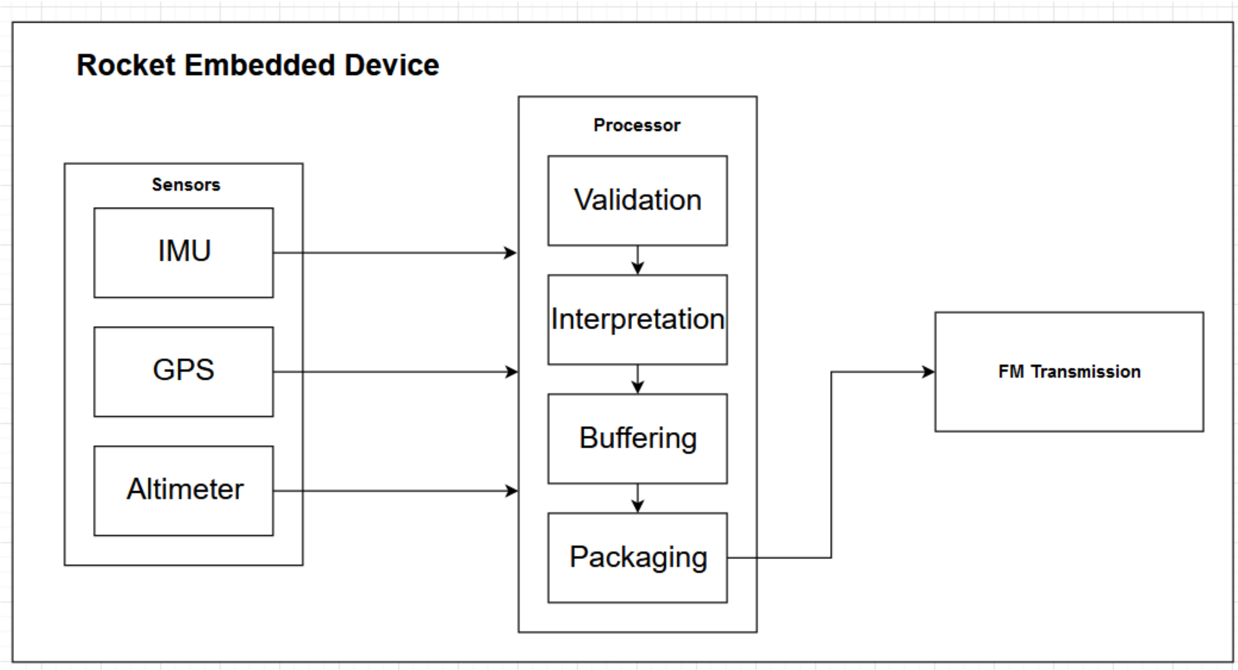

Figure 3: Example subsystem description diagram

4.1 Sensors

This system will be set up to take accurate readings from the flight of the rocket.

4.1.1 Assumptions

This system should gather the data continuously about the rockets flight. The data pins will be read continuously to put together packets for transmission. Each individual sensor will monitor will embedded inside of the rocket.

4.1.2 Responsibilities

The IMU will output readings such as force and angular rate, that are needed to plug into kinematic equations. This is responsible for readings of speed, acceleration and my other measurements.The altimeter will be responsible for sending the barometric readings This sensor needs to . The GPS system will report back and send coordinates relative to the last position. This system is to be accurately working so that it can be used for a map of tracked paths.

4.1.3 Subsystem Interfaces

Table 2: Sensors

ID	Description	Inputs	Outputs
GPS	A sensor collecting information	Relative position movements	Relative position measurements
IMU	A sensor detecting force and angular speed	Rocket	Force and Angular Speed
Altimeter	A sensor detecting barometrics	Rocket	Measurements

4.2 Processor

This system gathers the sensor data, monitors ready flags from the receiver, manipulates the data, and prepares the data to be sent.

4.2.1 Assumptions

A validation byte will be created once all measurements are completed and the transmission validation byte is detected. Then conversions for all sensors will take place to produce meaningful data in the interpretation phase. The data will then enter the buffer. The buffer is the holding place for the data until last packet data has completely arrived or been trashed. Only once the buffers check has been completed, the packet can be completed. Then the data is ready to start transmission.

4.2.2 Responsibilities

The validation subsystem verifies that the last packet was sent successfully. It will let the processor collect official samples to include in flight information. The interpretation's job is to change raw input data meaningful data. The buffers job is to take in the data until the actual data in the last packet is verified as received. Once that is verified or deal with, data will be allowed to entered into the packet and ready for transmission.

4.2.3 Subsystem Interfaces

Table 3: Processor

ID	Description	Inputs	Outputs
Validation	Transmission Ready Signal	Receiver Validation Byte	Read Byte
Interpretation	Making the sensor values into useful info	Raw Data	Meaningful Data
Buffer	A holding place that will hold data	Data Receive byte	Data
Packaging	Gathering all the data in on packet	individual measurements	Data Packet

4.3 FM Transmission

4.3.1 Assumptions

The receiver system will be set to the same frequency as this subsystem. Once the last data bit is sent the receiver will send confirmation.

4.3.2 Responsibilities

This system will send out a start bit once ready to send data. Then, the packet will be send its data values on the frequency wave. Finally, an end bit is sent to confirm completion of transmission.

4.3.3 Subsystem Interfaces

.

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Sends out data using FM Transmission	Data Packet	Data Packet

5 Data Receiver

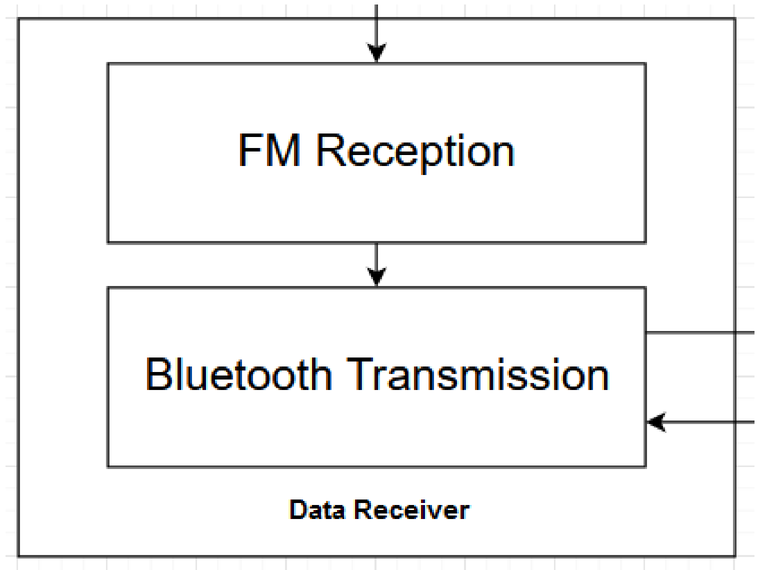

Figure 4: Example subsystem description diagram

5.1 FM Reception

FM Reception gathers frequency waves and converts the information carried by them to a usable form.The receiver uses electronic filters to separate the desired radio frequency signal from all the other signals picked up by the antenna

5.1.1 Assumptions

Assumes the transmitter is sending at a particular frequency, where the start and end bit packet will be a fixed size

5.1.2 Responsibilities

Reading the frequency waves and distributing the values in data structure

5.1.3 Subsystem Interfaces

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Receive Fm data	FM waves	Data structure

5.2 Bluetooth Transmission

The Bluetooth transmission uses wireless communication technology and allows devices such as mobile phones and computers to transmit data over a short distance.

5.2.1 Assumptions

Assume that if the Bluetooth receiver receives an incomplete packet then it would request re-transmission.

5.2.2 Responsibilities

It is responsible for transmitting data to the phone application via the the phone's Bluetooth radio.

5.2.3 Subsystem Interfaces

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Bluetooth Transmitter	Data Packets input 2	Data Packet

6 Phone Application

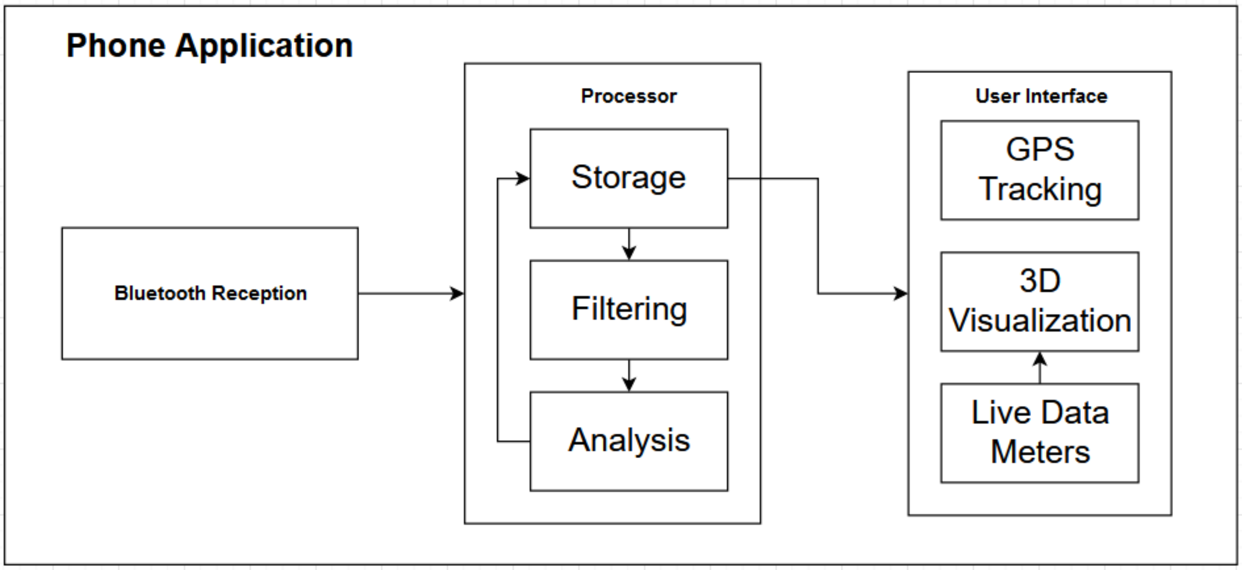

Figure 5: Example subsystem description diagram

6.1 Bluetooth Reception

The phone application starts with receiving data from the receiver via the phones bluetooth radio. Once received the data proceeds up the application eventually through to the UI when requested.

6.1.1 Assumptions

The phones bluetooth receiver will assume that the Data Receiver module sent a complete packet.

6.1.2 Responsibilities

The bluetooth receiver will be responsible for ensuring a complete packet has been received before passing the data onto the processing system. The bluetooth module shall also be responsible for informing the processor about the status of its connection to the data receiver.

6.1.3 Subsystem Interfaces

Table 7: Phone Bluetooth Interfaces

ID	Description	Inputs	Outputs
#1	Bluetooth Receiver	Data Packet from Data Receiver	N/A
#2	Device data bus	N/A	Validated Packet

6.2 Processor

The processor subsystem receives and prepares the data for presentation on the UI.

6.2.1 Assumptions

The processor assumes that all packets received from the bluetooth system are complete and valid.

6.2.2 Responsibilities

Each of the responsibilities/features/functions/services of the subsystem as identified in the architectural summary must be expanded to more detailed responsibilities. These responsibilities form the basis for the identification of the finer-grained responsibilities of the layer's internal subsystems. Clearly describe what each subsystem does.

We use the Kalman filter to essentially makes an educated guess on what the next data point will be based on the current data point. Then depending on how much the next data point differs from the current, it will weight the actual value verses the predicted value accordingly. What we are left with is a relatively simple and lightweight method for extracting meaningful trends from noisy data in real time. We apply the filter to remove the noise.

Once filtered the data is moved to the storage system. This system is responsible for storing the filtered data in a manner that is able to be worked with in the analysis section. This means storing it as data types or in structures native or supported by the Android operating system. The storage system provides an interface to the analysis and UI systems to be able to access the data when needed.

The Analysis system takes stored data and prepares it display to the user. This system will be responsible for notifying the UI when key events occur. This includes things such as the rocket moving out of range and sudden changes of direction. The system will track key metrics related to the flight and provide them to the interface when requested.

6.2.3 Subsystem Interfaces

Table 8: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Bluetooth to filter	Bluetooth Packet	Filtered Data
#2	Filter to Storage	Filtered Data	N/A
#3	Storage to Analytics	N/A	Filtered Data
#4	Analytics to Storage	N/A	Metrics
#5	Storage to UI	Data Request	Requested Data

6.3 User Interface

The user interface subsystem displays the in-flight data, the 3D visualization, and the Google Maps location to the user.

6.3.1 Assumptions

The user interface assumes that all of the data received from the processor are in the proper format required for display.

6.3.2 Responsibilities

The UI is responsible for displaying the necessary in-flight data to the user at a rate of four times per second, allowing for the user to have a visual understanding of the rocket's current flight. After a flight, the UI will use all of the data to create a 3D visualization of the rocket's flight path. Live coordinates will be displayed during flight, so that the user will have a basic understanding of the location of the rocket. Also, Google Maps' API will be utilized to displayed the current location of the rocket and the user. This allows for the user to be easily directed to the location of the rocket for retrieval.

6.3.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 9: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Storage to 3d Visualization	Metrics	Data Visualization
#2	Storage to Data Meters	Metrics	Metrics

References