Universal Telemetry and Tracking System (UTTS)

CSC490 - Senior Project I

Fall 2021

Project Requirements

10/17/2021

Instructor Comments/Evaluation 92/100

Overall a good requirement document that followed the recommended outline. This document helps readers to understand the project better. References & Citations were well done. User cases were good.

Things to improve:

1. Could be better to include some background of UTTS by including an example.
2. Functional requirements could be improved

Table of Contents

Abstract

Universal Telemetry and Tracking System (UTTS) is a project intended to make the world of remote telemetry and tracking more open and widely available. While there are existing solutions for a variety of industries, most are specifically designed for a single application. We seek to develop a system which is fully user customizable and is based on widely available hardware. This allows the user to adapt UTTS to suit whatever application they may have even if that application changes completely between sessions. The focus on openly available hardware also decreases the chances that a user's system will become entirely obsolete due to advances in a single component. With this document, we seek to better define UTTS and begin developing a plan for how we will proceed with the project.

Introduction

Background:

Telemetry and tracking are a critical part of our modern world. Many of the technological systems which we interact with each day would not be possible without the use of telemetry. Take, for example, your preferred social media platform. Regardless of what you may have chosen, that platform would be difficult if not impossible to run without telemetry in the form of usage data to decide how many servers are needed to service the clients connected at any given time. While the stakes may not be quite as high for potential users of UTTS, there still exists a clear need for telemetry and tracking systems. By providing environmental telemetry data and high refresh rate GPS tracking, UTTS provides a simple yet expandable system for data logging on mobile platforms.

Objective & Overview of Project:

The objective of UTTS is to create an open system for telemetry and tracking utilizing widely available hardware. Additionally, this software system is to be customizable by the user to allow them to adapt UTTS to best suit their specific needs. To accomplish this, the project will consist of two hardware portions: the ground station and the Mobile Telemetry & Tracking Unit (MTTU); and their associated software portions. Both the ground station and the MTTU will be based off Raspberry Pi single board computers and will feature low power radio frequency (RF) transceivers to allow wireless bidirectional communication. The ground station software will provide the user with a simple GUI which allows them to view live processed telemetry from the MTTU and choose which, if any, data will be recorded and saved for later use. The MTTU software will be responsible for broadcasting its GPS location along with any telemetry data from an onboard sensor package. The project will utilize only widely available and relatively inexpensive hardware, targeting a final cost per system of under $500.00 USD. Additionally, the software will be released fully open source to allow users complete control over their system.

Team Details & Dynamics:

While the entire team will be necessary to ensure the success of this project, members have elected to take on leadership roles based on the phases they feel most comfortable in, have prior experience, and that relate to their majors. These roles, along with member majors, are shown in the table below. By assigning leadership based on these criteria, we hope to ensure that our final product will be of the highest possible quality. Each leader will be responsible for ensuring that their respective tasks are completed correctly and on time.

Team Member:	Major:	Leadership Phase:
	CS	Requirements/Analysis, Design

	CET	Implementation
	CS	Presentation/User Manual

In order to facilitate the timely completion of the project, the team has established a Discord server which will act as our primary means of communication as well as an additional offsite backup for any code produced. The team meets virtually on a flexible schedule based on the task at hand but expects to meet at least weekly for organizational purposes. Members also maintain some in person availability to facilitate physical hardware development as necessary.

Application Domain

Project Context:

UTTS is intended to fill a gap which exists in the world of tracking and telemetry. There are few, if any, fully open-source solutions for obtaining custom tracking and telemetry data from any mobile platform. Since we are working towards a fully customizable solution, the domain for this project could become very large. UTTS might be of particular interest to those interested in unmanned aerial systems (UAS) or high-powered rocketry, as these are fields which are expanding and growing more complex each day. This project might also be useful for action photographers or videographers looking for a solution to automate camera tracking.

For information on the technical terms included in the document as they relate to UTTS, please refer to the technical glossary within the appendices.

Initial Business Model

Operational Environment:

UTTS will be designed primarily for hobbyists in UAS but will not be locked down to any one application. By targeting this very demanding use case, we hope to make the system well suited to any other application that a user might come up with. This environment means that the system will have to be user friendly while also allowing more advanced individuals to modify it to best suit their needs.

Since we are targeting UAS, the physical operating environment for UTTS is an extremely demanding one. Any system designed to fly onboard an unmanned aircraft must be light, rugged, and ideally power efficient. As such, UTTS will be designed with these aspects in mind.

Description of Data Sources:

UTTS will utilize data produced by sensors onboard the MTTU. These sensors will include a GPS, a compass, and an environmental sensor package which outputs pressure, temperature, and humidity. This data will be used to generate all telemetry outputs and can optionally be recorded to a file for later investigation by the user. This data will be wirelessly broadcasted by the MTTU and thus is susceptible to interference, interception, or tampering. This data is not sensitive and thus does not need any sort of encryption, but packets will be validated on arrival at the base station and some noise filtering will be performed to ensure only realistic location deltas will be accepted.

Use Case UML Diagrams and Descriptions:

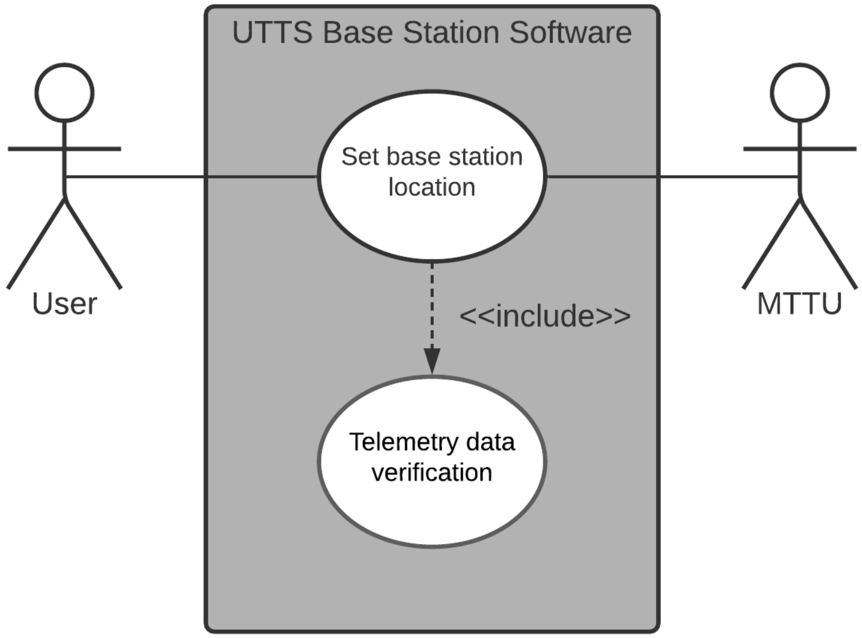

Figure 1: Use case UML diagram for user setting base station location.

Description: When starting the ground station software, the user will be given the option to use their previous location, enter a location manually, or use the MTTU's current location as the base station location. If the MTTU is selected, the GPS location portion of a validated telemetry packet will be used.

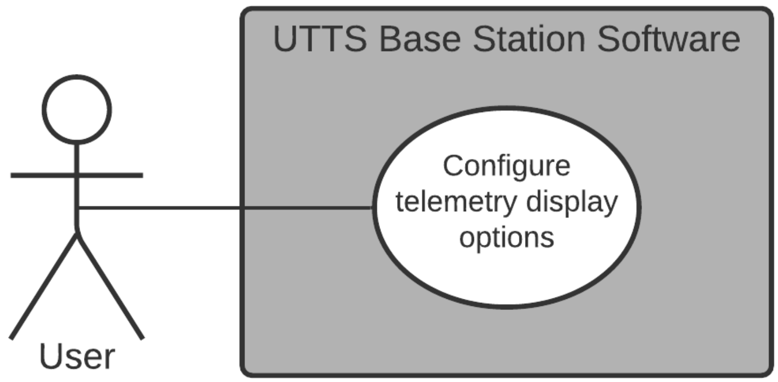

Figure 2: Use case UML diagram for user configuring telemetry display.

Description: The user is given the ability to select what live telemetry data is displayed by the UTTS base station software. They will also be given options for units and the placement of this data on the display screen.

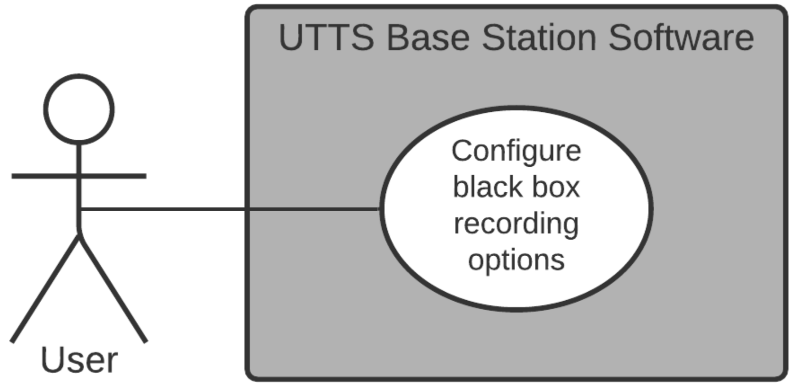

Figure 3: Use case UML diagram for user configuring black box recording.

Description: The user is given the ability to select what, if any, telemetry is stored in the black box log file for that session. Users can also select the file name, start time, end time, and how often the telemetry string is recorded.

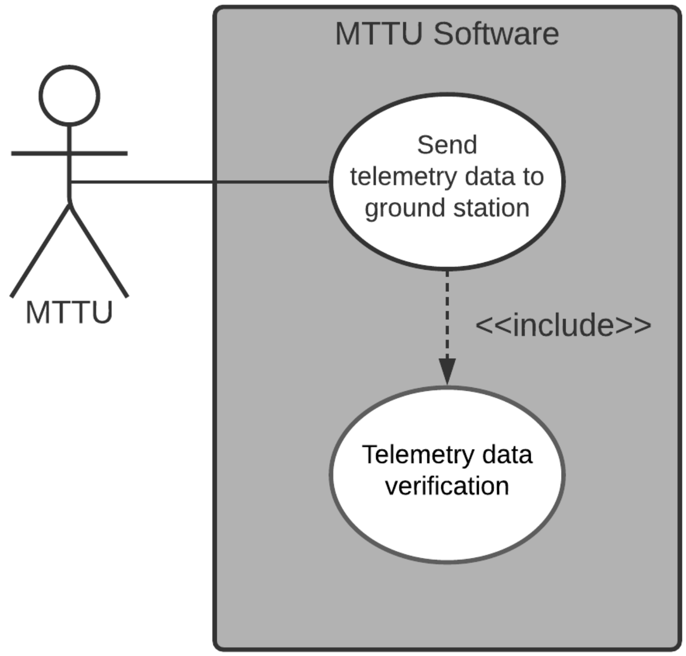

Figure 4: Use case UML diagram for MTTU telemetry transmission.

Description: The MTTU will transmit updated telemetry data packets to the ground station at least ten times each second. These packets will include information for verification which will be confirmed by the ground station on receipt.

Initial requirements

Functional:

This project will have several functional requirements to operate properly. These requirements include the connection of the mobile unit to the ground station as well as the selection of displayed data and a “black box” option for recording. With these options, the project will provide a high degree of customization to the user.

On startup, the ground station will connect to the MTTU, allowing information to be sent back to the ground station. Once the MTTU is connected, the user will set the base station location. Telemetry output will then begin, and the user will be able to select what information to display, where to display it, and what units to use. This information will consist of the MTTU's GPS location, distance, elevation, and azimuth. The user will also be given the option to enable black box recording for the MTTU. This option will record selected telemetry data from the MTTU. This data will be time stamped for later analysis. This option provides an easier method for recovery if there is a failure or anomaly. It can also be used as a training aid in certain scenarios. With these functional requirements, the user has everything that would be needed to monitor and track the MTTU once it has been connected to the ground station.

Nonfunctional:

This project also has several non-functional requirements. These include the hardware's size and weight, the GUI's responsiveness, and the MTTU's packet rate. As these are largely quality-of-life issues, they will be considered nonfunctional for the purposes of this document.

A more compact and lightweight device could potentially increase appeal to enthusiasts, especially those aviation related fields. The size and weight of sensor packages are always a significant limitation in UAS, and a smaller, lighter package would always be appreciated.

Additionally, people such as photographers and videographers, who often carry a lot of equipment heavy, would benefit from a lighter device which is easier to transport and handle.

A more responsive user interface would benefit all, from enthusiast to industrial applications. No one enjoys using slow software, especially when that software is intended to provide a live data output. By optimizing the GUI for speed and responsiveness, the user experience could be greatly enhanced.

Finally, the MTTU's sensor packet rate could greatly impact the usefulness of the entire system. Applications where the MTTU will be attached to fast moving or rapidly accelerating platforms would be impossible with a low packet rate. We expect our GPS unit to be the bottleneck for the MTTU's packet rate, but we are targeting at least 10 packets per second which should be sufficient for any platform moving less than 100m/s.

Documentation:

I. Proposal Document
   - The initial document used to propose and define the project.

II. Requirements Document
   - The document used to communicate the goals of the project and establish a basic development plan.

III. Specifications Document
   - The document that outlines the actual requirements, standards, testing, and parameters of the final product.

IV. Design Document

- The document used to describe the projects capabilities, appearance, and functions that are required for the project. This is used to accelerate the development process.

V. Project Log

- Document that will provide information that will aid in the ability to track and assess any issues that could affect the time that the project is completed or even halt the progress of the project.

VI. User Manual

- The document used to communicate as well as educate the users on technical information associated with the project. This will include a quick start guide to get users up to speed with the system as easily as possible.

Testing / Revisions

This document was created through the combined efforts of the team members. This collaboration was achieved with the document being shared online allowing each member to work on it independently as well having the ability to be completed conjointly. The document was completed in sections with multiple sections being drafted each day. Multiple revisions were saved locally as a backup, each with the own timestamp to indicate age.

A similar method will be utilized when working on the software. To achieve this the writer/technician will test each module as it is developed. The module will then be passed to the remaining members for quality assurance testing. Any issues discovered can be discussed during set meeting times where a member or team will be appointed to fix the issue. Software will follow a similar method of revision with copies being saved in multiple locations after each work session. This provides an emergency backup in the event of any computer system failure as well as providing a rollback point if major bugs are accidentally introduced.

Hardware testing will be conducted by the individual team member who has the components. Upon the next meeting time, progress will be demonstrated to the team where additional troubleshooting may be conducted based on any outstanding issues or future issues that arise.

List of References

Python Software Foundation. (2021, October 7). *Welcome to Python*. python.org. Retrieved October 12, 2021, from https://www.python.org/

Raspberry Pi Documentation. (2021, October 12). raspberrypi.com. Retrieved October 12, 2021, from https://www.raspberrypi.com/documentation/

TIOBE Software BV. (2021). *TIOBE Index for October 2021*. tiobe.com. Retrieved October 12, 2021, from https://www.tiobe.com/tiobe-index/

U.S. Space Force. (2021, February 22). *GPS Overview*. gps.gov. Retrieved October 12, 2021, from https://www.gps.gov/systems/gps/

Appendix: Technical Glossary

Black Box - A device used to store telemetry data for analysis in the event of an anomaly or failure. In the case of UTTS, this is simply a text file which stores time stamped telemetry data for later review.

C/C++ - C and C++ are high-level computer programming languages. They are simple but powerful languages and are used in many of today's computer applications. Despite their age, they remain some of the most popular programming languages in use today (TIOBE Software BV, 2021).

GPS - Acronym for "Global Positioning System." This is a method of location finding and navigation using radio signals transmitted by satellites in orbit (U.S. Space Force, 2021). GPS generally refers to the system owned by the United States government however this document will use it to refer to all radionavigation satellites that transmit publicly.

GUI - Acronym for "Graphical User Interface." This refers to the graphical menus which a user interacts with in a software application.

Location Delta - The change in the MTTU's GPS location between two received telemetry packets. This difference over time equates to the mobile platform's speed.

MTTU - Acronym for "Mobile Telemetry and Tracking Unit." This refers to the mobile portion of the project including both mobile hardware and mobile software.

OS - Acronym for "Operating System." This is the software responsible for managing user interactions with a computer. Examples include Linux, Windows, and MacOS.

Packet Rate - The rate at which the MTTU sends complete telemetry packets. This is measured in packets per second.

Python - Python is a high-level computer programming language (Python Software Foundation, 2021). Python is included in the default operating system of Raspberry Pi computers and is a popular language for beginners in programming, and is, overall, one of the most popular programming languages in use today (TIOBE Software BV, 2021).

Raspberry Pi - Raspberry Pi refers to a popular series of single board computers developed by the Raspberry Pi Foundation. These are full-fledged computers in a very small form factor capable of running a desktop operating system. By default, these systems usually run a Linux based operating system known as Raspbian (*Raspberry Pi Documentation*, 2021).

RF Transceiver - Also abbreviated as XCVR. This is a device which uses radio waves to enable wireless communication.

Telemetry - Within the context of the project, telemetry refers to any data wireless transmitted by the mobile platform to the base station. This includes GPS latitude and longitude, barometric pressure, ambient temperature, and all other data contained within a telemetry packet.

Telemetry Packet - A single string of telemetry data from the MTTU including all sensor outputs and GPS location data. Packets are transmitted by the MTTU multiple times per second.

UAS - Acronym for "Unmanned Aircraft Systems." This refers to any unmanned aircraft. Examples include radio-controlled quadcopters, fixed-wing aircraft, rotary-wing aircraft, and lighter-than-air aircraft, as well as fully autonomous drones of any type.

UTTS - Acronym for "Universal Telemetry and Tracking System." This refers to the entirety of the project herein.

Appendix: Team Details

This document was developed in cooperation with all team members over the course of approximately two weeks under the leadership of Dalton Obitko. While members' work was largely spread out across every section of the document, the majority contributors for each section are described as follows:

- was a major contributor to the "Initial Requirements," "Testing/Revisions," and "Workflow Authentication" sections of the document.
- was a major contributor to the "Initial Business Model" and "Initial Requirements" sections of the document.
- was a major contributor to the "'Introduction," "Application Domain," and "Technical Glossary" sections of the document.

All other major sections of the document were developed in tight coordination during meetings and thus members contributed equally in these sections. Additionally, all members joined in the effort to proofread and finalize the formatting of the document as well as work on any graphs or figures included herein.

Further development will occur as described previously. Members chosen as development phase leaders will be responsible for the successful completion of the tasks assigned during their phase, however, leadership is largely symbolic, and all members are free to make suggestions and vote on changes at any time.

Appendix: Workflow Authentication

I, , attest that the work that has been completed on this document as well as future work on this project was done so as stated per the description of this document.

Signature:  **Date**: 10/16/2021

I, , attest that the work that has been completed on this document as well as future work on this project was done so as stated per the description of this document.

Signature:  **Date**: 10/16/2021

I, , attest that the work that has been completed on this document as well as future work on this project was done so as stated per the description of this document.

Signature:  **Date**: 10/16/2021

Appendix: Report from Writing Center

See 10/14/2021 email from Kylie Kurtz.