Navigation with Augmented Reality on Raspberry Pi 3

A Project Report Presented to
The faculty of the Department of Electrical Engineering
San José State University

In Partial Fulfillment of the Requirements for the Degree
Master of Science

By

Patel, Dhatri Navinbhai SJSU ID: 010692526
EE297B Section 2, Spring 2017
E-mail: dhatrimaharshipatel@gmail.com Contact No.: +1 (669) 254-7961

Pandya, Samvid Bhargav SJSU ID: 010152610
EE297B Section 1, Spring 2017
E-mail: samvid.pandya150@gmail.com Contact No.: +1 (408) 455-2881

Department Approval

_____

_____ _____
Dr. Thuy T. Le *(signature)* Date
Project Advisor

_____ _____
Dr. Thuy T. Le *(signature)* Date
Graduate Advisor

Department of Electrical Engineering
Charles W. Davidson College of Engineering
San José State University
San Jose, CA 95192-0084

Abstract

The main aim of the project is using Augmented Reality, an application of Computer Vision. The Computer Vision is the use of computer graphics in real world. The Augmented Reality is also application of placing an object on the real world with help of Computer Graphics or also known as CGI. The use of Augmented Reality for navigation is new concept. It requires the knowledge of Computer Vision.

The Navigation with Augmented Reality is usage of AR in Navigation with synchronizing the camera and other required modules to reach the destination with help of real world images and in real time with help of Real Time Operating System. The Real Time Operating System is base for Augmented Reality as AR works in Real Time on Raspberry Pi 3 Model B.

Table of Contents

List of Figures

1. Introduction

1.1 Motivation

The Navigation with Augmented Reality is a new concept of using Augmented Reality for the purpose of Navigation. The motivation behind this project is that the user who uses conventional maps like Google Maps, Here WeGo Maps, Tomtom and Garmin Maps. They sometimes find it difficult to reach the destination by taking the wrong exit from the freeways or highways. The reason behind is that the maps are basically in Two Dimensions. Which gives only information of length and breadth.

The main usage in this project is of Augmented Reality. The Augmented Reality will help the user to take a proper exit from the freeways and will give a user an edge for driving the vehicle with help of camera. The user will take the exit through checking the camera (Display) in the car will take the exit according to the maps and triangle and shapes created for a route to a destination.

The augmentation will help the user to know the path to reach a destination and will tell the user which way is shorter to reach the destination with help of distance to take the exit. The navigation with augmented reality will be work with Raspberry Pi 3 Model B. Raspberry Pi 3 will work as a microcomputer that works on Real Time Operating System.

1.2 Objective of Project

The main objective of this project is to use Raspberry Pi 3 as a microcomputer for navigation purposes using Augmented Reality. The augmented reality works as the main keyword in this

project to work with raspberry pi 3 with Real Time Operating System as the base operating system.

1.3 Current Work

The company's like Here Maps, Google Maps are right now introduced street view that works like augmented reality but it is an application of image processing that works with ArcGIS tool to create a point of interest and street view uses images taken from the company's vehicle. The car or any vehicle which takes the picture of the area by driving through the street and then upload it to the maps with help of image processing. The current work is basically with only image processing but not with Augmented Reality.

The augmented reality is basically used for gaming and video sharing but not for navigation. The navigation is simply a new concept for augmented reality. The Current work is basically in street view of google earth. The project is basically works on for new concept. The companies are looking forward for the technology to be implemented in day to day life. It require a little bit of time but will be on its way.

2. Real Time Operating System

In this Project, Real Time Operating System is a base for implementation of Augmented Reality as it works in real time. Thus, detailed information regarding RTOS is mentioned in this section. Real Time Operating System is an operating system for the devices which work in real time. In other words, RTOS is used for applications which need a quick response. To discuss it in depth, RTOS is nothing but an operating system which works

differently from General Purpose Operating System like Windows, Linux, Ubuntu, MAC OS or Android. RTOS is not limited to computers only. Rather, it can be applied to any electronic devices which keep tendency to work in Real Time. Real Time Operating System, the name itself indicates its meaning that it is a type of operating system that works on the concept of Real time that means the response of particular task is quick and fast. Thus, it is specially designed for real time operations.

For RTOS, a critical factor is Time. Unlike General Purpose Operating System (GPOS), the response and efficiency of the operating system depend on whether the task will be completed within its deadline or not. Thus, all the time constraints should be met in real time systems. Thus, RTOS is a Time Critical Operating System. In other words, RTOS can be defined as the ability of an operating system to complete the task within decided time. Whereas, general purpose operating system is not a time critical system. It only looks at the throughput i.e. how many numbers of tasks are completed within given time. Thus, RTOS is used for applications in which the required response is quick and time is an important constraint.

Moreover, the other important feature of RTOS is that it is able to do multitasking. That means it supports the execution of multiple tasks. Multiple users can execute multiple tasks concurrently. That is the best advantage for many applications and at many fields where thousands of tasks need to be completed within a given deadline. So, multitasking within a given boundary time is a plus point of a Real time operating system compared to a General purpose operating system. Thus, there is a long list of all the features of a Real time operating system. To discuss them in depth, it is necessary to understand about Classifications of Operating Systems, fundamental components or structure of RTOS, how RTOS works, its

methodology, what makes it different from general purpose operating system etcetera. In this case, all the aspects of real time operating system are discussed below.

2.1 Classifications of Operating Systems

First of all, an Operating System is a critical program that runs on the computer which controls other programs and applications running on a computer. An operating system is one kind of a system software which is responsible for managing all the software and hardware that are associated with a computer. Now, there are numerous different programs running at the same time i.e. concurrently on a computer. In that case, input and output are not the only factors for those programs but other parameters and functions also need to be taken into consideration for processing on an input to get the required output. For that purpose system units like Central Processing Unit (CPU), Arithmetic and Logic Unit (ALU), System Memory, Memory required for processing and storing the result, Software and several peripheral devices used for the process, configuration of all the peripheral devices, tracking of all the processes and many other required processes. Thus, to make ensure about whether all programs gets what it needs or not, an operating system performs coordination of all these and provides synchronization between all of these to complete the task.

Apart from doing these basic functions, an operating system supports features like multiple users can work on system platform and they can do multiple tasks concurrently. That shows its ability to handle multiple users concurrently doing multiple tasks that are called as multitasking. An operating system is also able to run the same program on more than one CPU that is called as multiprocessing. Moreover, the other concept is operating system can also handle the execution of several parts of a single program at the same time. This concept is

known as multithreading. Thus, from this paragraph, it can be clearly understood that an operating system supports four different concepts to execute the program that is multiuser, multitasking, multiprocessing and multithreading. Now, it is necessary to understand how these concepts can be handled by this most important program i.e. operating system. Based upon this project, it is necessary to understand related to the real time operating system.

So, basically, operating systems can be classified into three categories based on their working factors. These are Non-Real Time Systems, Soft Real-Time Systems, and Hard Real-Time Systems. All of these are described below in detail.

1). Non-Real Time Operating System

It is a system in which time does not perform an important role. There is no deadline to complete the tasks. It only sees the throughput of the system. That means time or deadline does not matter to analyze the performance but the thing that matters in a performance of the system is a total number of tasks can be completed by the system. If a total number of completed tasks are high that means the throughput of the system is also high and the performance of the system or the efficiency of the system is greater. As the response of the input does not quick or it is not in real time, these type of operating systems are known as Non-Real Time systems.

2). Hard Real-Time Operating System

It is a type of real time system in which time is a most important factor which is taken into consideration to measure the performance. Every task must be completed within its specified deadline. If the system is unable to meet the time constraints or deadline then it can

result into catastrophic effect. Thus, hard real time systems require strict time limit. If the system is not able to do so, it is considered as a system failure. The task must be completed quickly, faster or within its specified duration. Thus, these requirements of meeting the timing constraints are so hard or strict, this type of system is known as Hard Real Time Operating system.

3). Soft Real-Time Operating System

It is a type of real time system in which time performs as an important factor but not as strict as a hard real time system. In other words, if the system is unable to meet the deadline then it does not result in catastrophic effect or it does not turn into system failure. It is undesirable to not to complete the task in deadline but also it will not cause any dangerous effect due to that. In any system, for a particular program, there can be several tasks need to be complete within one program and each of these tasks has some deadline. In this case, if a single task or computation is late and not able to meet the deadline then it cannot cause the system failure and usually, it is not significant to measure the performance of the whole system. Thus, in this case, the performance of the system depends on the average response time of each task. Thus, this type of operating system where not meeting a deadline is not desirable but also does not cause any dangerous effect, is known as soft real time operating system.

After understanding of the classification of an operating system, to know more about real time operating system like how it works, it is necessary to understand about its

fundamental components that make its working in real time. This information is discussed below.

2.2 Fundamental Components of Real Time Operating System

Basic fundamental components or functionality of real time operating system: 1). Scheduler, 2). RTOS Services, 3). Synchronization and messaging. The detailed information about all these functionality and components are explained below.

2.2.1 Scheduler

As mentioned above, an operating system is responsible for the execution of several tasks and programs within its deadline or sometimes the requirement of response may be at the same time. Specially, in real time operating system, as the input is given to the system, it is required to get the output quickly. To understand this, take an example of ATM machine. ATM machine works based on the concept of a real time operating system. As the user enters the amount of withdrawal, the response from the machine is quick or of fewer seconds. And the user gets the money in fewer seconds. So, it is a faster response from the machine. Thus, this example explains the concept of a real time operating system. That means it is the requirement of RTOS to give the quick response as soon as the user enters the input in a system.

As it is the responsibility of an operating system to handle or to coordinate all above-mentioned requirements in order to achieve higher efficiency and performance of the system, the operating system makes or organize the execution of these events or tasks in a proper manner. Generally, there are multiple tasks running on the system and multiple users try to attempt the task. To control all of these tasks, dividing these tasks into different states is an

easy and efficient method. So, generally, each task has three states. These are: i). Ready to Run, ii). Running, iii). Waiting and iv). Blocked. These states are explained in detail.

i). Ready to Run

This state indicates that the current task has all the things or resources are available and ready to start the execution. This state shows that task is ready to execute but not in running state.

ii). Running

By this state, an operating system comes to know that the task is currently executing. The particular task is currently in running phase.

iii). Waiting/Blocked

This state indicates that the particular task is currently not able to run because currently, it is busy in executing other or previous tasks. Thus, the next task has to wait for its turn for execution. So, as its name suggests it can be said that the task is in waiting or blocked state.

iv). Inactive

In case, when a task does not have anything to do or it does not have any resources to execute, it shows this inactive state. Sometimes it also indicates the completion of the task or the task has not started yet.

So, this is all about different states of a task. Now, as mentioned above in this report, real time operating system have the ability to control multitasking with multiple users. To handle this, it works on the mechanism of scheduling of each task. Also, for scheduling these more than one tasks, there are mainly three techniques like co-operative scheduling, round robin scheduling, and preemptive scheduling. Based on the type of scheduling, efficiency of the real time operating system can be determined. In other words, scheduling is the most

important functionality of a real time operating system. Three techniques of scheduling multiple tasks are described in detail. This is explained below.

1). Cooperative Scheduling

This is a type of technique to schedule multiple tasks in which each task run until it finishes its execution. In other words, once the system or program started, scheduler forces the task to run continuously until the task renounce itself or until it achieves its aim. So, the scheduler tracks the status of all the tasks and accordingly give a schedule to other tasks.

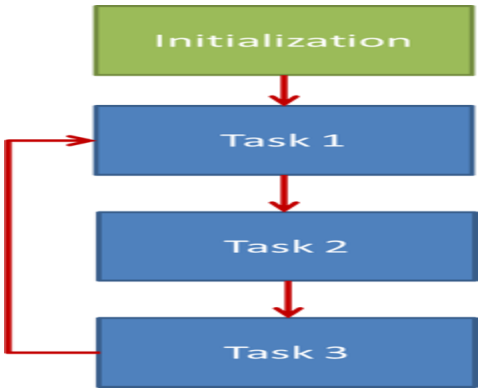


Figure 1. Co-operative Scheduling

As per figure 1, it shows that once the system is started or initialization is done, execution of tasks begins but in cooperative scheduling, task 1 runs continuously until it completes its execution. Only after task 1 completes its execution, task 2 can start its turn to run and completes its duty. So, this is all about cooperative scheduling.

2). Round Robin Scheduling

In this technique of scheduling a task, multiple tasks are handled in such a way that every task will have their turn to execute in a given time. Every task is allocated a fixed given time. When the task has its turn, tasks should complete its execution within that allocated time.

If it is unable to complete execution within given time then it may lose its turn and cannot complete its determined flow of execution. In this case, the task may also lose its generated data and it has to wait for getting its turn next time.

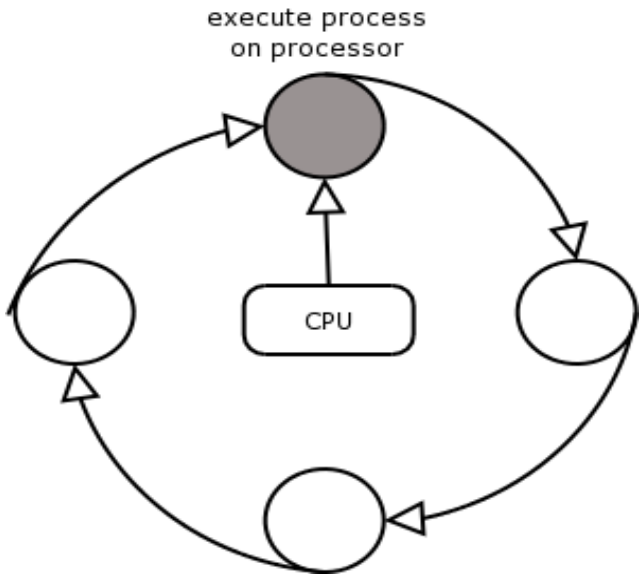


Figure 2. Round robin scheduling

Above figure (figure 2) shows the concept of round robin scheduling. As mentioned above that in round robin scheduling, every task gets their turns to complete their execution. Every task is given fixed time slots and within that time slots, they can complete their execution. If particular task unable to complete within that time, then it can complete on the next turn. So, this is the basic concept of round robin scheduling.

3). Preemptive Scheduling

Preemptive scheduling is a type of technique in which the scheduling is based on a priority of tasks. Thus, it is priority based time allocation. To decide about the priority of the task, the scheduler gives each tasks priority levels. According to priority levels, each task

executes in the system. The task with the highest priority will execute first. Thus, during execution of another task, if any tasks have the highest priority, it is possible that it may interrupt the system and that task can be forcefully suspended. In that case, the previous task may lose the data. So, basically preemptive scheduling schedules the tasks based on their priorities.

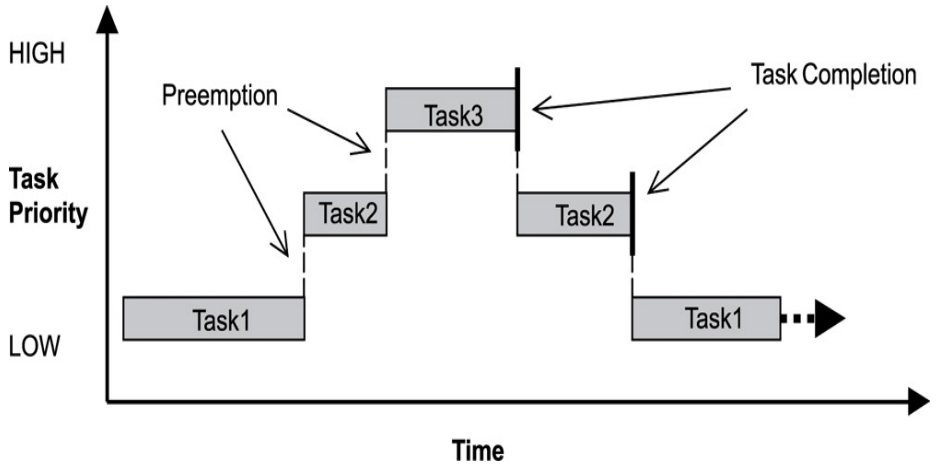


Figure 3. Preemptive Scheduling

Figure 3 shows the concept of preemptive scheduling. So, as mentioned above, preemptive scheduling works on priority based concept. The task with the highest priority will execute first and so on. The first scheduler checks priority level of a task if the priority level is lower, then it will look for next tasks and compare priority levels of all tasks. By comparing them, it is easy to recognize the task with the highest priority and this task will begin its execution first. So, it is easily be recognized in figure 3 about this type of schedule. As it can be seen that first priority levels are assigned to each of the tasks (i.e. task 1, task 2, and task 3) by a scheduler. After that, it comes to know that task 3 has the highest priority to run. So, task 3 will begin its execution and completes its execution first. Then the priority is like task3 > task2 > task1. So, the execution will also be like task3 then task2 and at last task1 completes execution. So, this is preemptive scheduling.

Now, after understanding the concepts of three techniques of scheduling, it is necessary to understand how this task should be handled and who is responsible for handling scheduling in multitasking. So, to control these scheduling, a kernel is responsible. The kernel is nothing but one program which controls and takes care of all these tasks and scheduling. It keeps tracking of all the process done in the operating system. The kernel takes care of creating, removing, keeping track of all states running under the task, assigning and changing the priorities of the task, other computations etcetera. Thus, a kernel is the central and most important component of operating systems. It is the heart of operating systems.

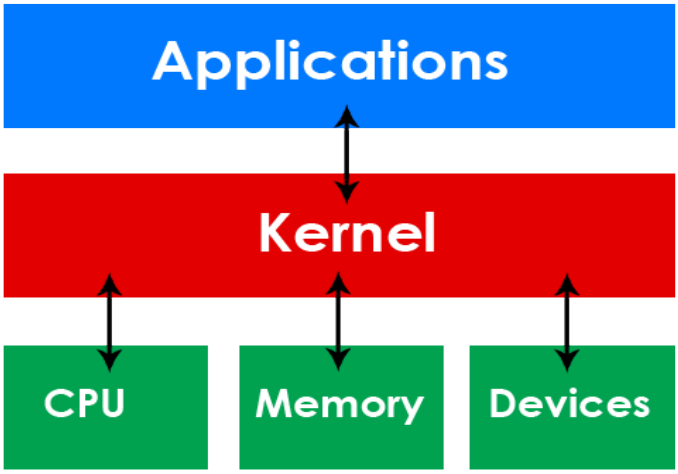


Figure 4. Kernel configuration

Figure 4 shows the coordination of whole architecture. As shown in the figure, a kernel is the central module of architecture. It monitors all the tasks running in the system and also provides necessary services to the respective task or program to complete its execution because tasks cannot obtain the attention of CPU all the time. It provides basic services like input/output management, interrupts handling services, time management, scheduling, peripheral device management and memory management services. So, by all these, it is clearly understandable

that kernel is the central unit or module of the operating system. It is a part of an operating system. Basically, its location is in main memory and especially in the protected area of memory. As it is very critical and most important program of the system, it is necessary to protect the program of a kernel to avoid overwritten by other programs and other parts of an operating system. Thus, after all this discussion about the kernel, it is understandable that kernel is an essential program of an operating system to provide various services to the software or hardware associated with an operating system. Now, by understanding all the services provided by the kernel in an operating system, it is necessary to understand services provided by real time operating system for this project. All these services provided by RTOS are mentioned below.

2.2.2 RTOS Services

As with scheduler, RTOS services are also considered as a fundamental component of real time operating a system. To provide different services to the system, the kernel performs an important function. As discussed before, a kernel is responsible for services. Because, CPU cannot give attention to each task all the time, the kernel performs a role of mediator and provides services to execution. The performance of an operating system also depends on this factor.

There are different types of services provided by the real time operating system like interrupt handling services, input/output handling services, peripheral device or other device management, memory organization services etcetera. With the help of interrupt handling services, it is possible to control program or execution of multiple tasks with multiple users by giving an interrupt to the CPU or system. For example, during preemptive scheduling in RTOS,

if kernel finds any task with the highest priority then it is necessary to execute it first. So, in that case, interrupt request can be given by kernel to stop the current process and to handover the control to highest priority task. Thus, in this type of processes, interrupt is necessary and it should be handled in such a way that it does not affect rest of the processes in a negative way. So, this type of interrupt handling services is provided by a real time operating system.

Next services provided by RTOS are input/output handling services. It indicates that kernel in the real time operating system is responsible for controlling all the incoming data or information. It is the responsibility of kernel to keep track of all the multiple inputs given to a system, how the processing is going on all of the inputs, from when and where the output will be generated etcetera. Input can be from any resources like CPU, peripheral hardware devices or from memory and any applications cannot connect directly to any of these resources all the time because of multitasking. Because of this reason, kernel comes into the picture. It provides services to both devices and applications by taking care of all the required tasks and processing to generate an output corresponding to the input. It also takes care about correct output reaches at right time to each task particularly in a real time operating system. As real time system requires quick output response to the input, the kernel takes care of all these tasks.

Moreover, as mentioned, an operating system is connected to any hardware devices i.e. peripheral devices and for storage it is connected to memory, for this purpose, to get better performance, it is necessary to manage or organize all these devices and memory in a system. To handle all of these, real time operating system i.e. kernel organizes all the devices connected to a system. Thus, apart from interrupt and input/output services, a real time operating system also provides services like device and memory management.

2.2.3 Synchronization and Messaging

The third important functionality of RTOS structure is synchronization and messaging. Synchronization as its name suggests, it is the critical factor for real time operating system as it requires correct output response quickly within fewer seconds. So, it is necessary to give the correct output by synchronizing all of the tasks. Because of quick response, real time operating system processes all required tasks in real time. To avoid loss of any important data, it is necessary that all the processes are done in synchronized manner. Synchronization is important especially when each task takes different time for processing. In this case, synchronization between all of these tasks is necessary. So, this is taken care by an operating system.

Furthermore, sometimes it is possible that there is a requirement of communication with other operating systems or requirement between tasks within a program. In this case, Messaging performs an important role. It helps to provide better communication by means of transferring packets or information in terms of messages. By using various kinds of messages, the destination can come to know about a type of information given by source.

RTOS provides different kinds of messages and their services. These various types of services are: event flags, semaphores, mailboxes, message queues and pipes. Each service has some purpose to use. Firstly, event flags are used to indicate synchronization between all activities done within tasks that is inter-task activities. The second type of services is semaphores. They are used to provide synchronization between an operating system and its shared resources. So, whenever it is necessary to give access to shared peripheral resources, semaphore service is given by the real time operating system to destination and source. Other services like mailboxes, message queues, and pipes are required to send information in terms

of messages between tasks. So, this is all about the concept of services provided by real time operating system for better efficiency and performance.

So, after discussing all of the aspects of real time operating system in detail, all of the basic concepts starting from classification of an operating system to basic components and functionality of the real time operating system is clearly understandable. Now, it's time to understand the methodology of the real time operating system especially used in this project. It is discussed in next section.

2.3 Creation and Simulation of RTOS

The Screenshot of RTOS Code is below with simulation and RTOS Kernel and Task Scheduler. It uses main four states as mentioned above. Invalid, Ready, Blocked and Delayed State. The RTOS code starts with a kernel which needs to be in the RTOS for booting purposes. The kernel is developed and it works with a scheduler which will decide the working of RTOS. The real time operating system in this project is of preemptive type. It works according to a priority of task and function.

The RTOS has basically a destroy process function. If the user modifies the destroyer function, then it will destroy every task on which RTOS is working on. The code basically is written in C language.

The RTOS States are basically for its working if the RTOS is in an INVALID state then there will be no operation will be done. The Ready state suggests that RTOS is ready for the working and will start on an execution of instruction provide to it.

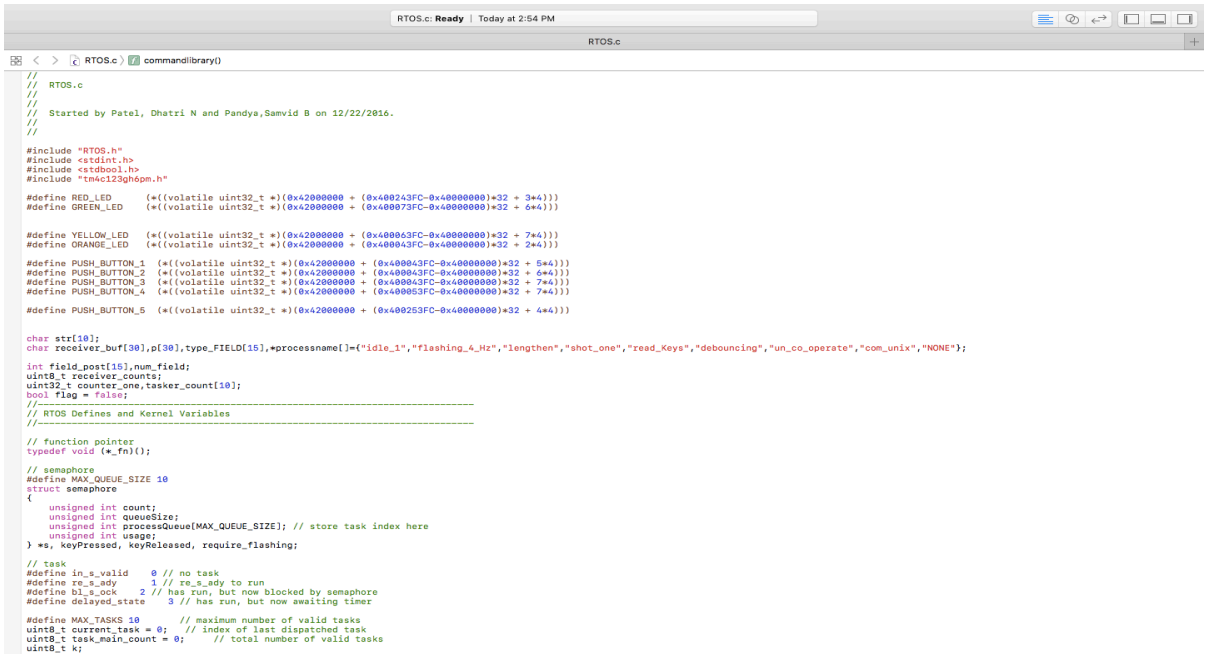


Figure 5. RTOS Code

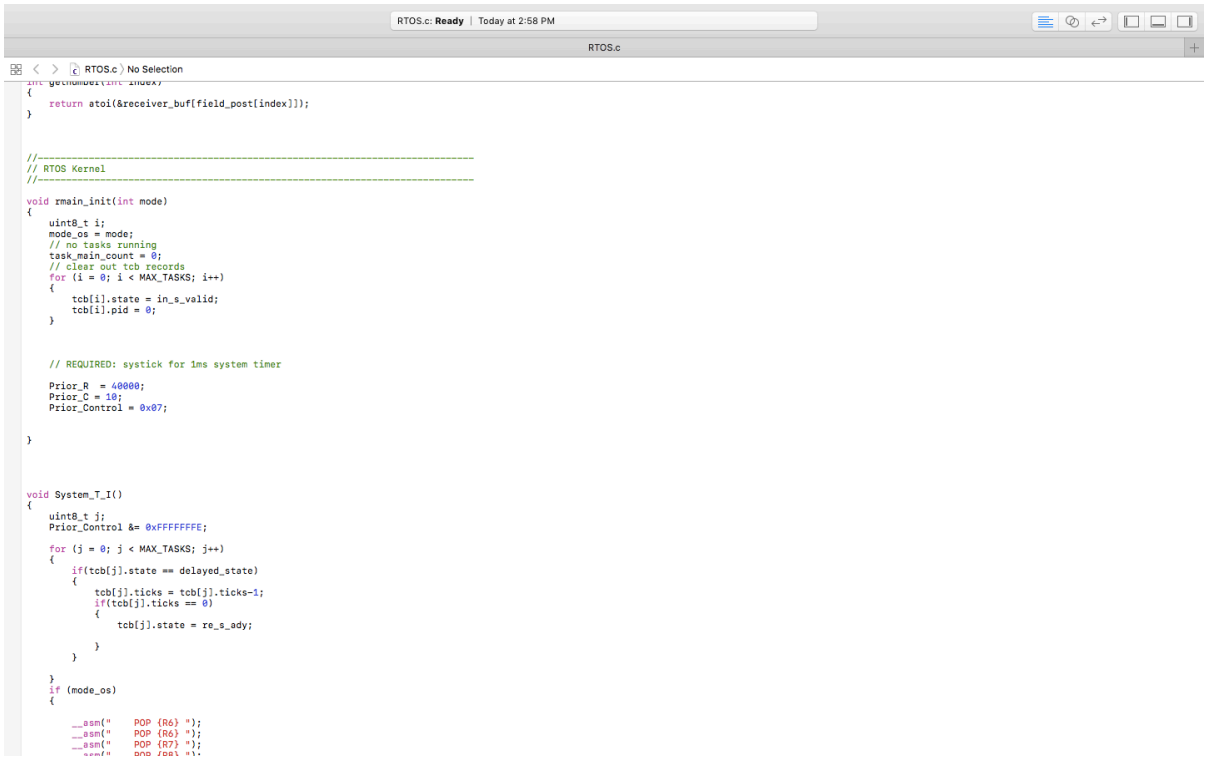


Figure 6. RTOS Kernel and Scheduler

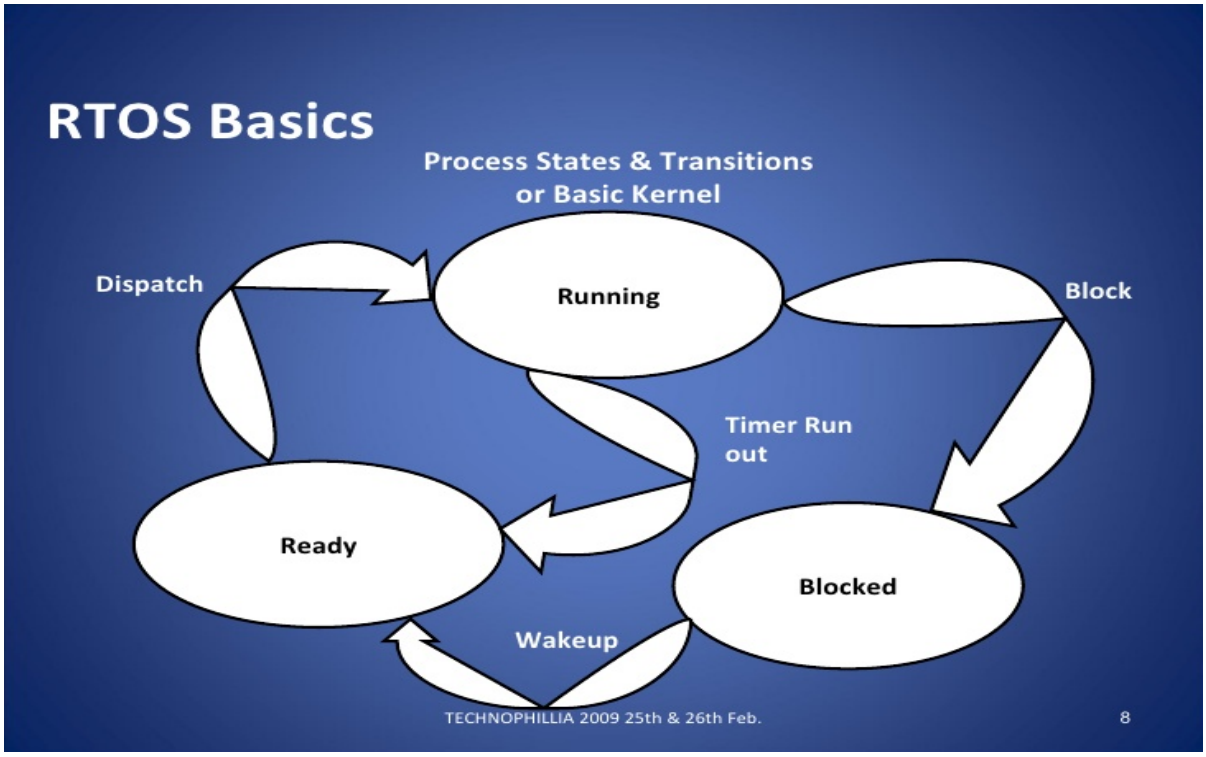


Figure 7. RTOS State for working

3. Navigation

The main aim this project is to use Augmented Reality for Navigation in Real World. The term to navigate means to reach the destination with help of some device or map. The basic idea of navigation is to locate the person or a device. The navigation in conventional two-dimensional might sometimes cause problem for the driver to take an exit from the freeway. Because of this problem, the driver has to drive more to reach the destination. The implementation of Augmented Reality with navigation will help a lot of to the driver. The Camera is being used to implement the real world in navigation. The basic requirement for navigation is the information of the locations.

The location of a particular destination can be known using Global Positioning System. The Global Positioning System determines the latitude and longitude of a place. The navigation route of user's current location and the destination is written in the device that supports the Global Positioning System. The Global Positioning System uses the satellite to obtain the information of the place and surrounding things that are near to that place.

3.1 The Global Positioning System

The government of United States of America has developed Global Positioning System for usage of their Air Force wing of Military in 1978. It is now used worldwide. To know the importance of GPS one should know how it works? The Global Positioning System has Twenty-Four or more satellite working together. To know the just basic location of a vehicle or device only Three to Four Satellites are required to obtain latitude and longitude. To get the exact location of a device or vehicle at least twelve satellites are required that works together and gives the exact location of the vehicle.

3.1.1 Global Positioning System – Working

The basic requirement of Global Positioning System working is that the device that has Global Positioning System Module it receives the radio signals from the satellites. If a device has it then following steps are the working of Global Positioning System.

The satellite (GPS Satellite) will send a RF signal with its position in the space and how much time it would take to reach the earth from space. These RF signals travel at speed of light that is about 3x10^8 m/s in space. The moving device will receive the RF signal sent from the satellite and notes its time of reception and will determine the movement in accordance to each satellite. The figure below shows how the Global Positioning System works with help of satellites and RF signals. The distance of vehicle can be obtained through speed and signal traveling time. The traveling time is the difference between time t1 and time t2, time t1 is time at which signal is sent from the satellite and time t2 is time at which the device has received the signal. The speed is of light. So the distance is the production of time and speed.

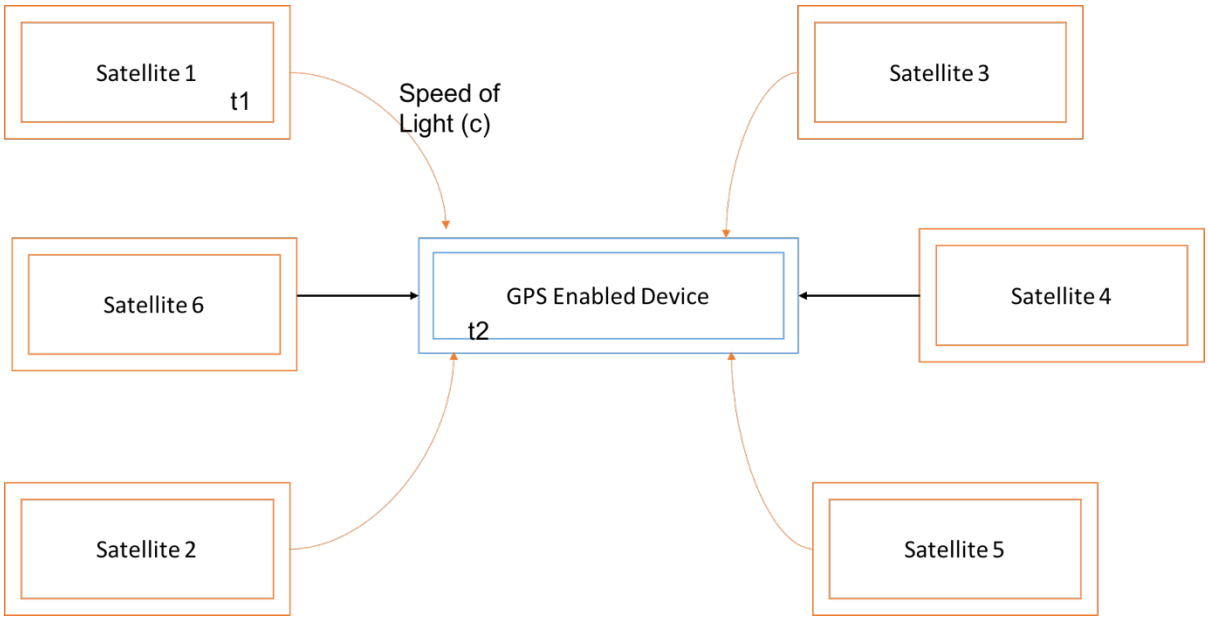


Figure 8. Working Block Diagram of Global Positioning System

There are many devices like an airplane, ships, cars, mobile phone, GPS receiver can be used as a receiver because these devices have Global Positioning System receiver module in them which receives the signal and come to know about the distance.

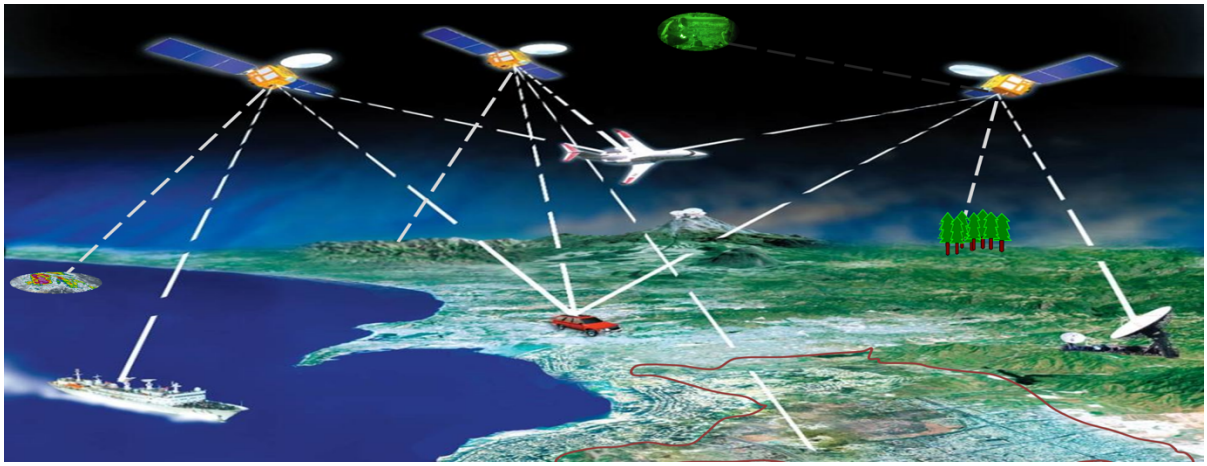


Figure 9. Basic Working of Global Positioning System

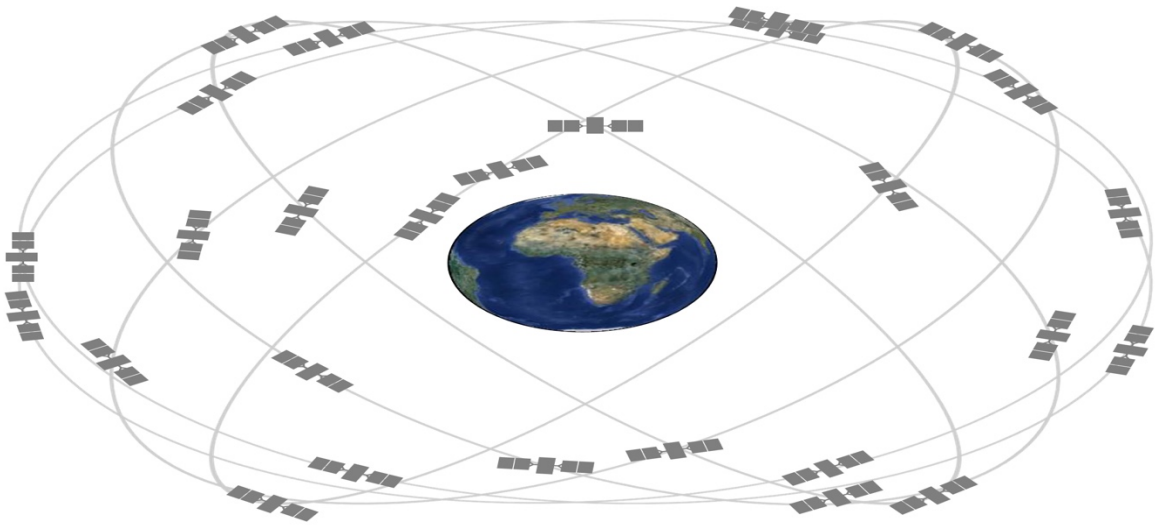


Figure 10. Global Positioning System: 24 Satellite Structure

3.2 GPS Module

The Global Positioning System receiver or module works for reception of signals. The Module basically works on the signal transmission and reception. The working block diagram of GPS Module is explained below.

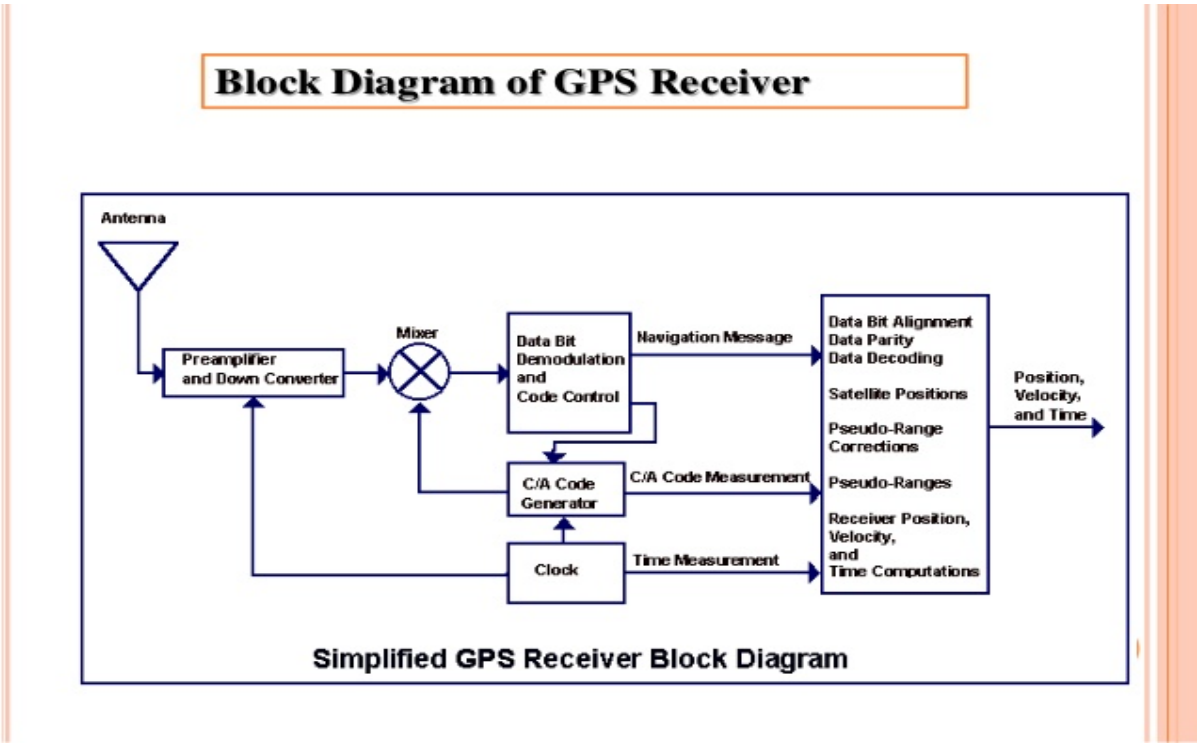


Figure 11. Receiver/Module Block Diagram

The RF Signal will be received through the antenna from the space. The Preamplifier and down converter block will down convert the RF signal received from the space and the amplifies the signal and sends it to Mixer where it mixes with the Coarse/Acquisition code generated through C/A code generator block.

The Navigation message is an outcome of Data bit demodulation and code control block. In this block, the data is demodulated and controlled by bit demodulation. The C/A code generator receives the signal from Data Bit Demodulation and Code Control Block and with help of clock it generates Coarse/Acquisition code. The navigation message, C/A code measurement and time (clock) is then sent to the main block of GPS Module where it processes various operation and the user will get information of Position of the device, time at which signal is received and speed of the device. This data will send to controller or processor for the display on the device board. This basically requires Analog to Digital converter that is inside the main block of the Global Positioning System's receiver.

The GPS Module for Raspberry Pi 3 Model B is shown in the figure below.

Figure 12. GPS Module for Raspberry Pi 3 Model B

Navigation in two dimension is basically given only x-axis and y-axis and if someone wants to know how it looks when the real world is implemented for navigation. It is basically developed using Augmented Reality. One can say the implementation of the human eye as background for an object for navigation. The Google Maps, Here Maps and other maps use two-dimensional and three-dimensional view but in terrain and satellite images.

The route between San Jose State University and 101 E San Fernando Street is shown in google maps APIs. The Project is basically using ArcGIS and Here Technologies' Application Programming Interface (APIs) for making maps. It is required for mapping of the image on the maps. ArcGIS tool is basically a tool for editing maps according to changes in the real world. It basically requires knowledge of the surrounding terrains.

Figure 13. Google MAPS APIs

The Google Maps' 2D view will give us the above figure. This image sometimes will cause a problem to the driver where population density or the road density is very high. The route in Los Angeles has many freeways interconnecting to each other so many times the driver takes the wrong exit and end up on the wrong path. This image is taken when the user is going to 101 E San Fernando Street from Washburn Hall, it is located at the intersection of S 8th Street and E San Salvador Street in San José, California.

The working of GPS module is explained through Flow Chart Algorithm. The GPS module starts the process when Vcc voltage is applied to the Vcc pin of GPS module. After the setup of an initial and important library, the receiver pin (Rx) will starts to receive data. It will also check whether the receiving data is corrupted? The corruption of data may occur because of bad weather or loss of GPS signal at an interior place like a basement, multistory buildings. The signal will be processed in processor and then will be stored into SD card or will be displayed on the display through Transmission pin of GPS Module(Tx). The Flow Chart of basic GPS Module working is shown below.

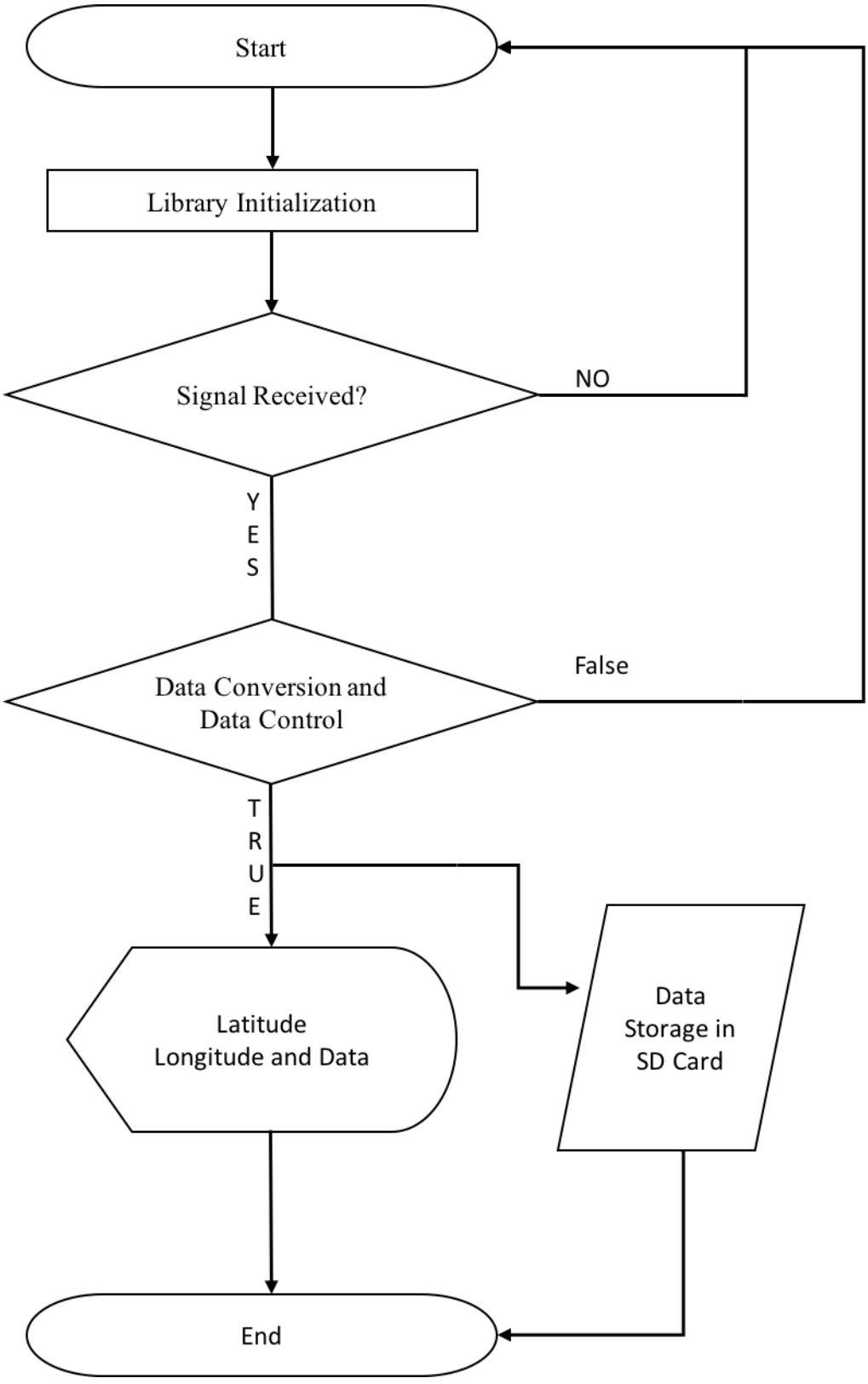


Figure 14. Flow Chart of GPS Module

3.3 ArcGIS Tool

The maps are edited and recreated using the ArcGIS Tool. It is basically used as GIS software. GIS means Geographic Information System. This tool has already maps that are used for changing the things or information of particular location with help of edit map. This project is a part of San José State University's Master project. It is basically ArcGIS API and Google Maps API. The main use of APIs is that they will provide exact interface as it appears on any website.

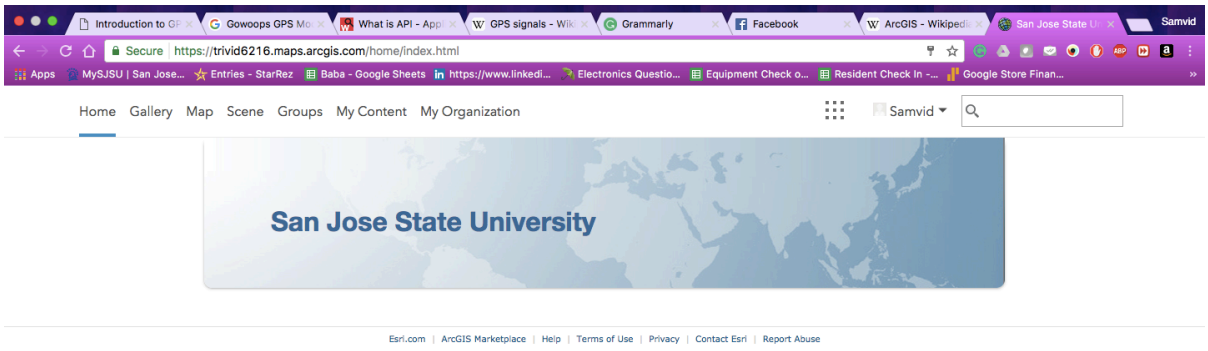


Figure 15. ArcGIS Main Page

The basic step to work with ArcGIS is to know the surroundings and If the user wants to add new things or needs to update about things of old places he or she must have that information with him or her. The ArcGIS Add Map feature is about this thing only. The user can add or delete the places according to its need and according to update available in the maps.

The Steps to Follow to create a Point of Interest in the Maps. It requires to learn some basic tools and editing features of ArcGIS powered by ESRI. The following steps and keywords are required. The following steps need to be followed. Steps to Create a map and map notes are as follow.

Step 1: **To select the place** to create or edit maps: The first for editing maps is to select the place or town in which the editing is going to be done. The basic requirement for it is placed. The place is selected for edit the maps is "**San José, California, United States of America.**" The selection of place is shown in figure 16.

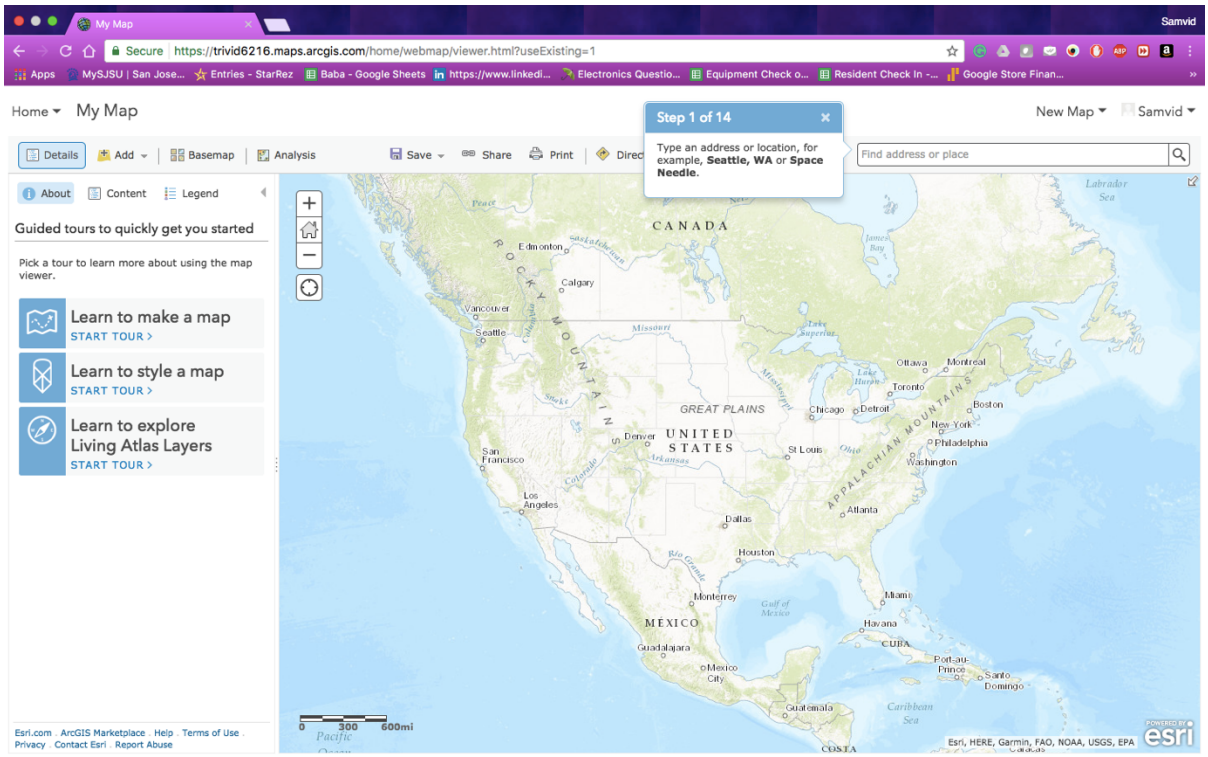


Figure 16. ArcGIS Edit Map - Step 1: Place Selection

Step 2: After selection of the place the second step is to **add the place** in the maps. The editing of maps is the main thing in this project. The user is giving the data of the destination using stick pin.

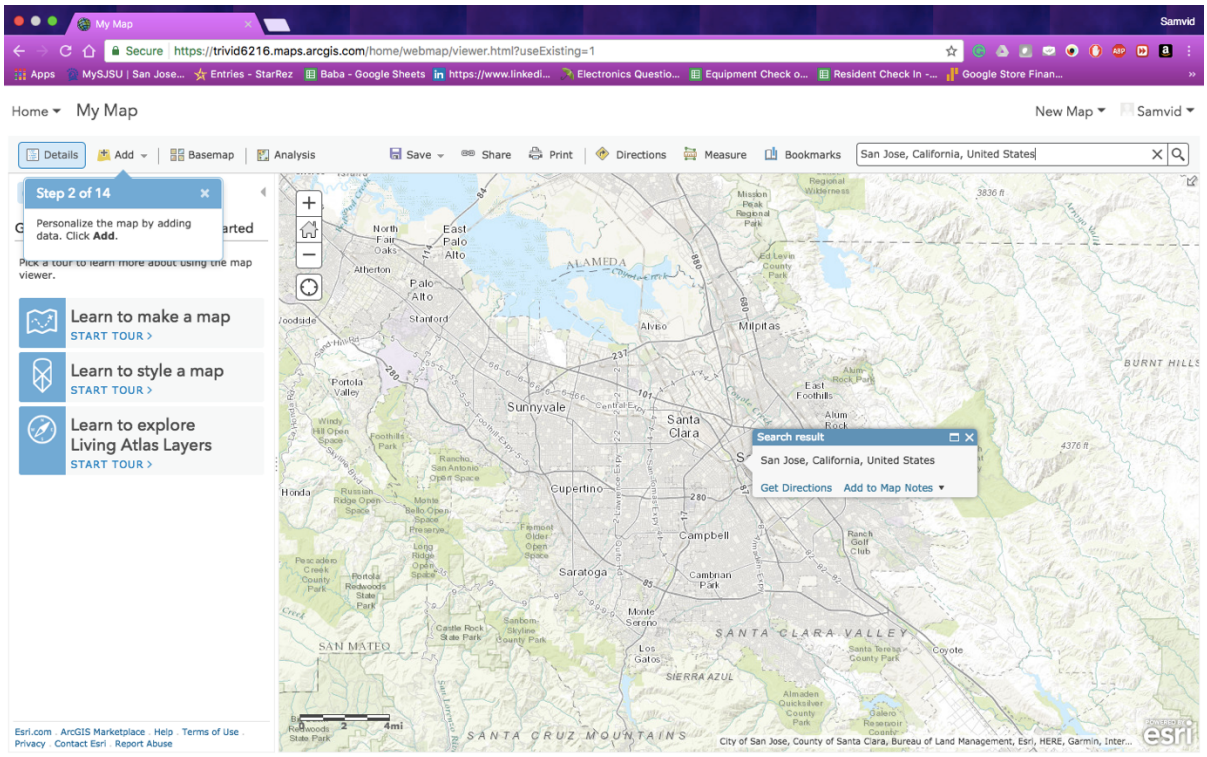


Figure 17. ArcGIS Edit Map Step-2: Adding the place

Step 3: Adding Map Notes: This step 3 is basically adding the map notes for the place in which the user is going to edit the details. This step is required to add the description of Area. The next step will be to an explanation of place.

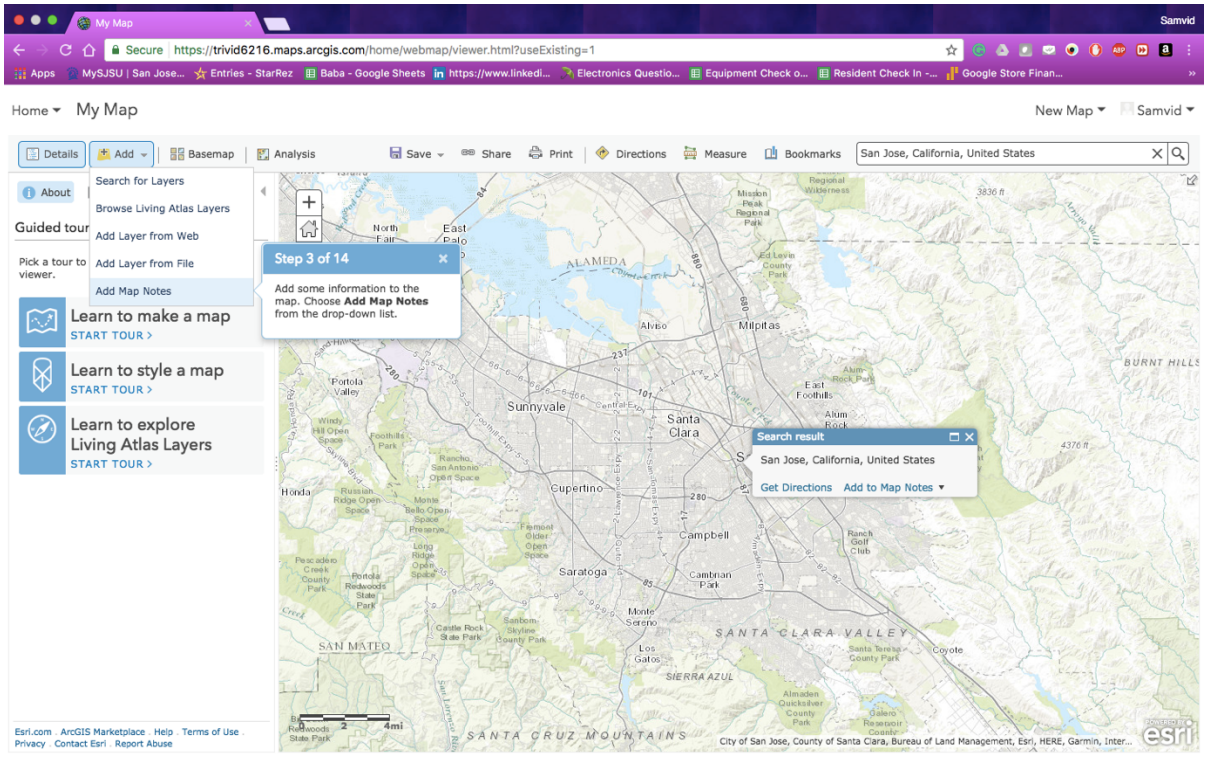


Figure 18. ArcGIS Edit Map Step-3: Adding Map Notes

Step 4: Explaining about Place: This step 4 is about the explanation about the place. The places to visits, places to hang out and gas station will be added using explanation portion of the maps. The figures 19.1 and 19.2 explains about it.

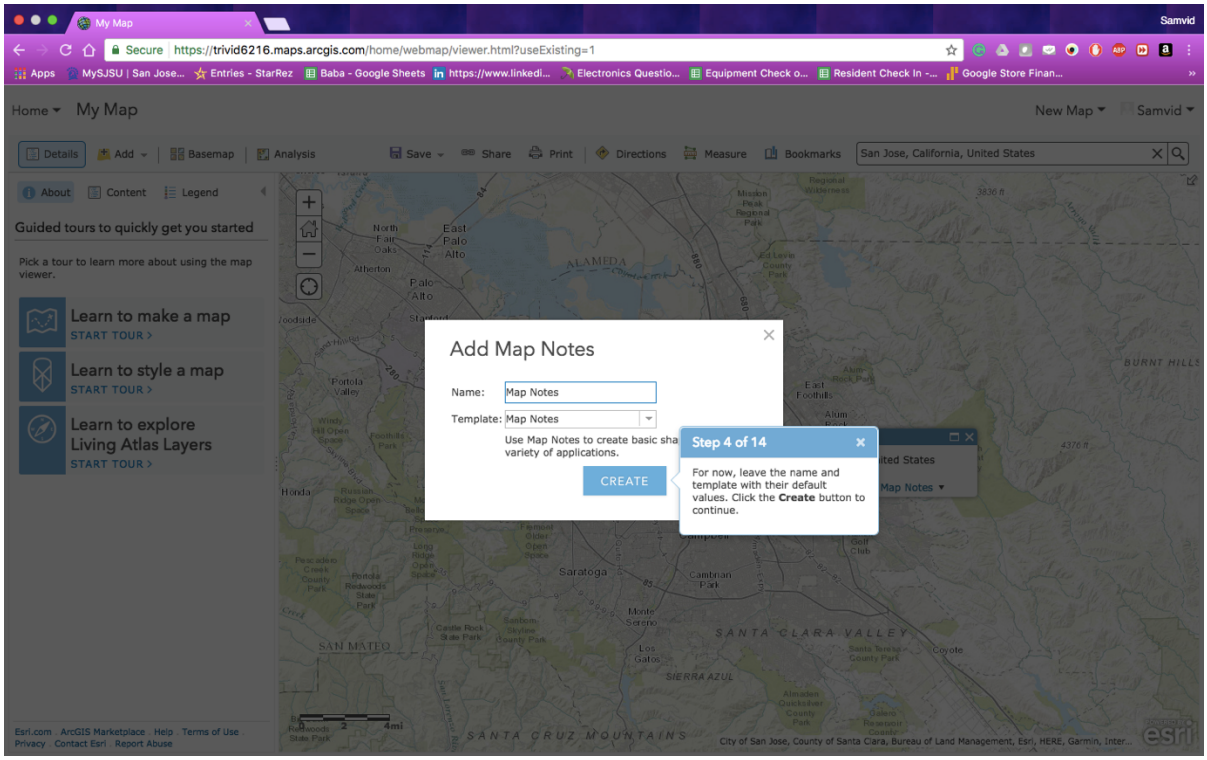


Figure 19.1. ArcGIS Edit Map Step-4

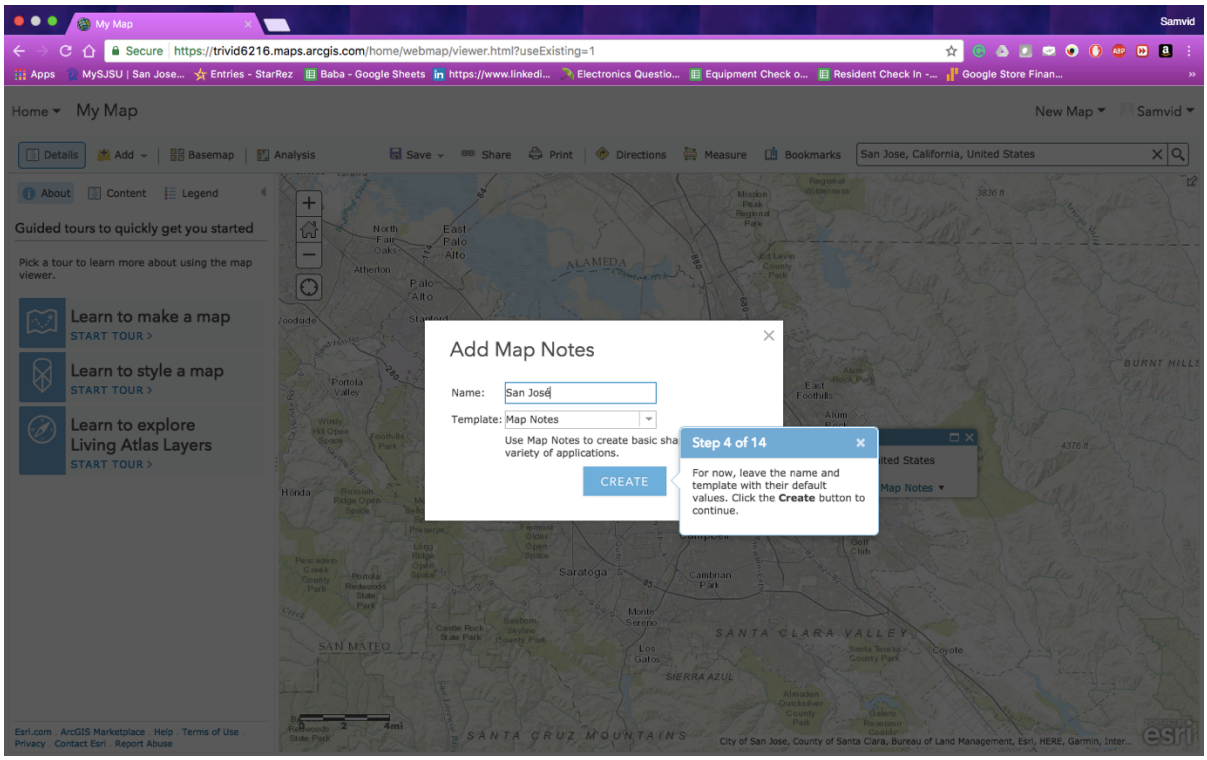


Figure 19.2. ArcGIS Edit Map Step-4

Step 5 & 6: Choosing a Point of Interest: These steps 5 & 6 is about choosing a point of interest. To choose a particular place for reaching. The Point of Interest will be pinned on images of a map using stick pin. Shown in Figure 20.

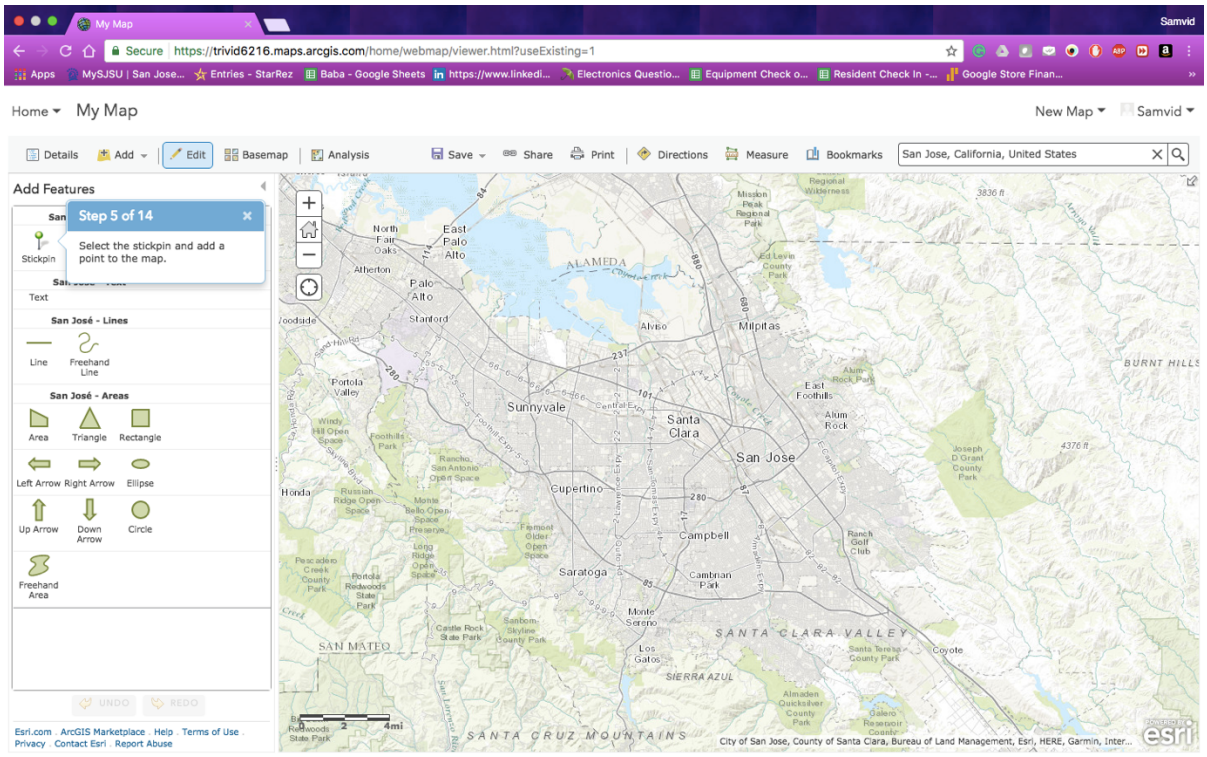


Figure 20. ArcGIS Edit Map Step-5 & 6: Point of Interest. Use of APIs

Step 7: Place of Interest: Step 7 is about the explaining about the place of interest. What is good? Why is it important? Benefits and usage to the society. The figures 21.1 and 21.2 are images of explanation of the place of interest and name of the place.

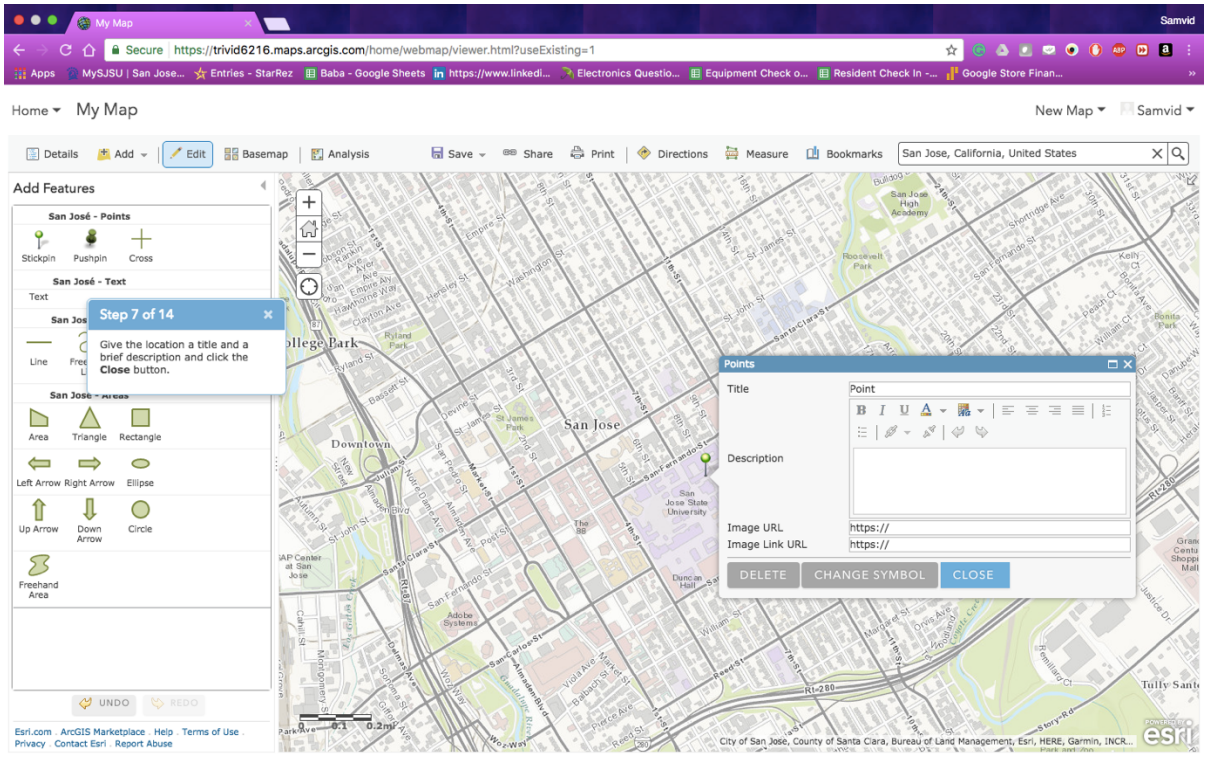


Figure 21.1: ArcGIS Edit Map Step-7: Name of the Place of Interest

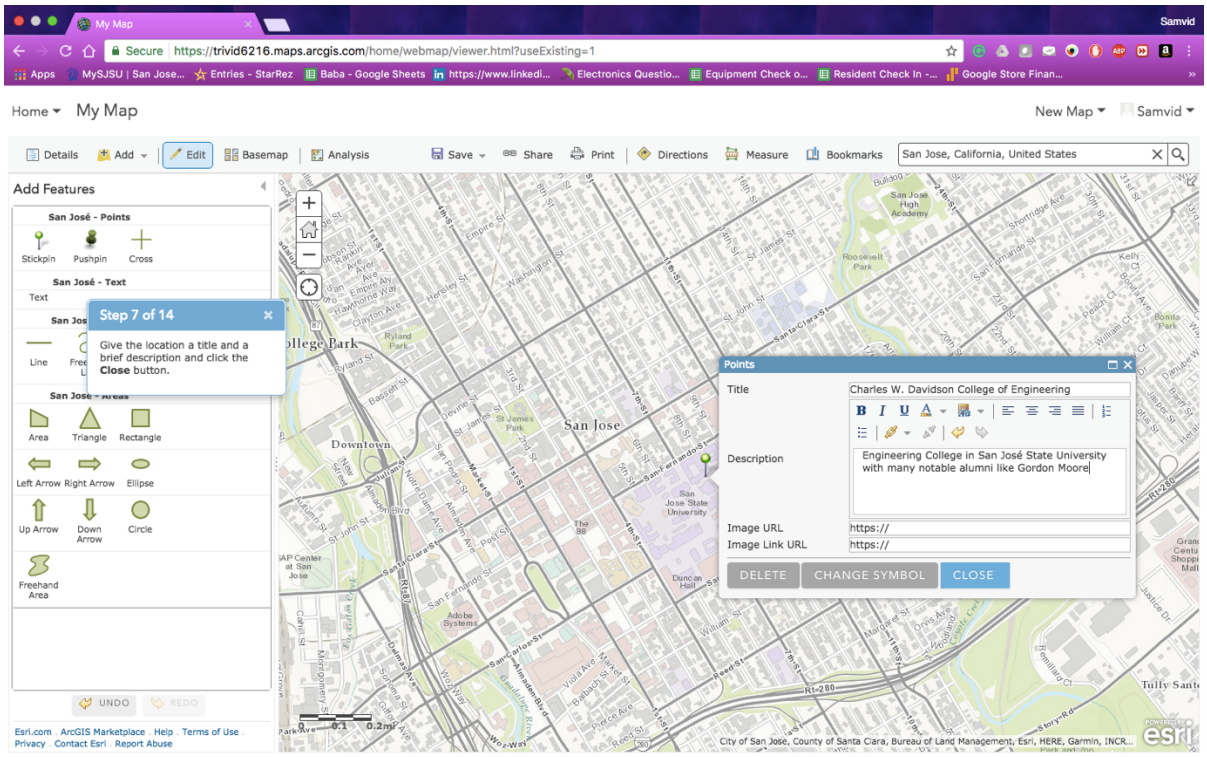


Figure 21.2: ArcGIS Edit Map Step-7: Explanation of the place

Step 8: Saving the Information: After giving details of the place of interest, the user will have to double check the details provided and then have to save the data using save as button in step 8. The figure below Figure 22 is shown below.

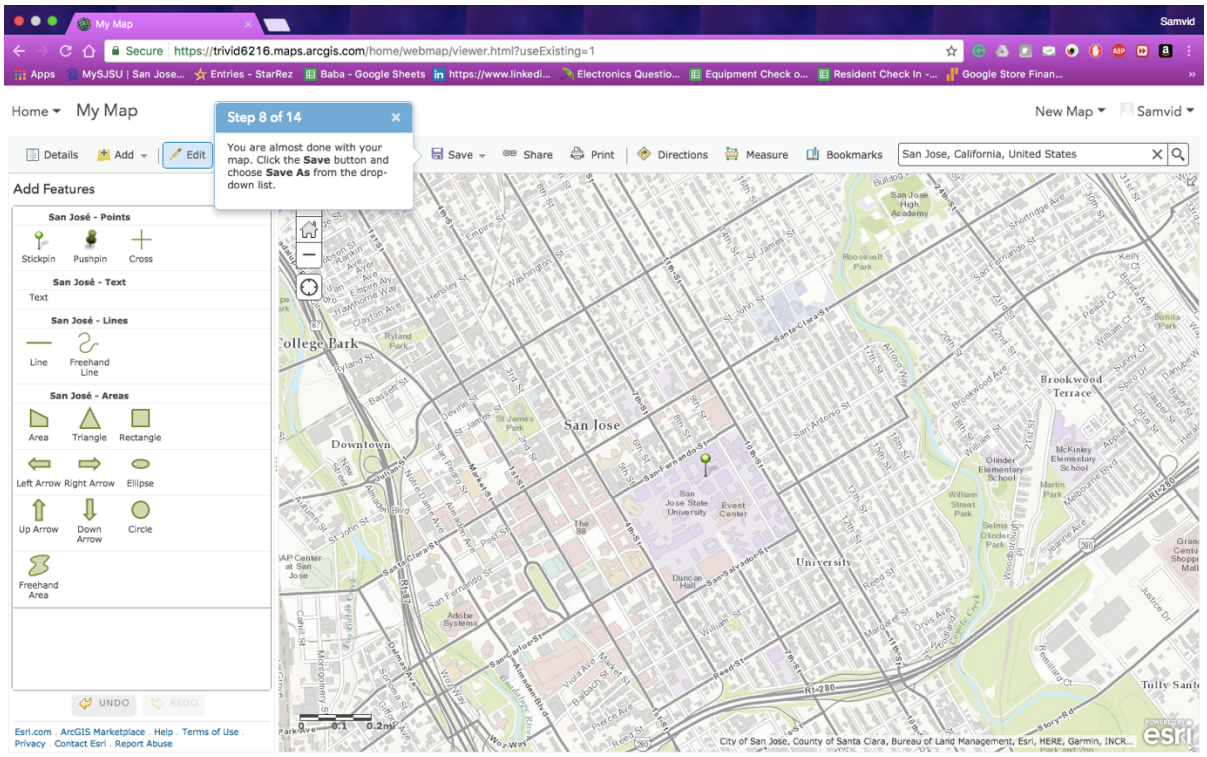


Figure 22. ArcGIS Edit Map Step-8: Saving the data.

Step 9: Saving the edited map: This step 9 is about to save the map after making additional changes like giving a particular short name to the place of interest. This step is shown in figure 23 below.

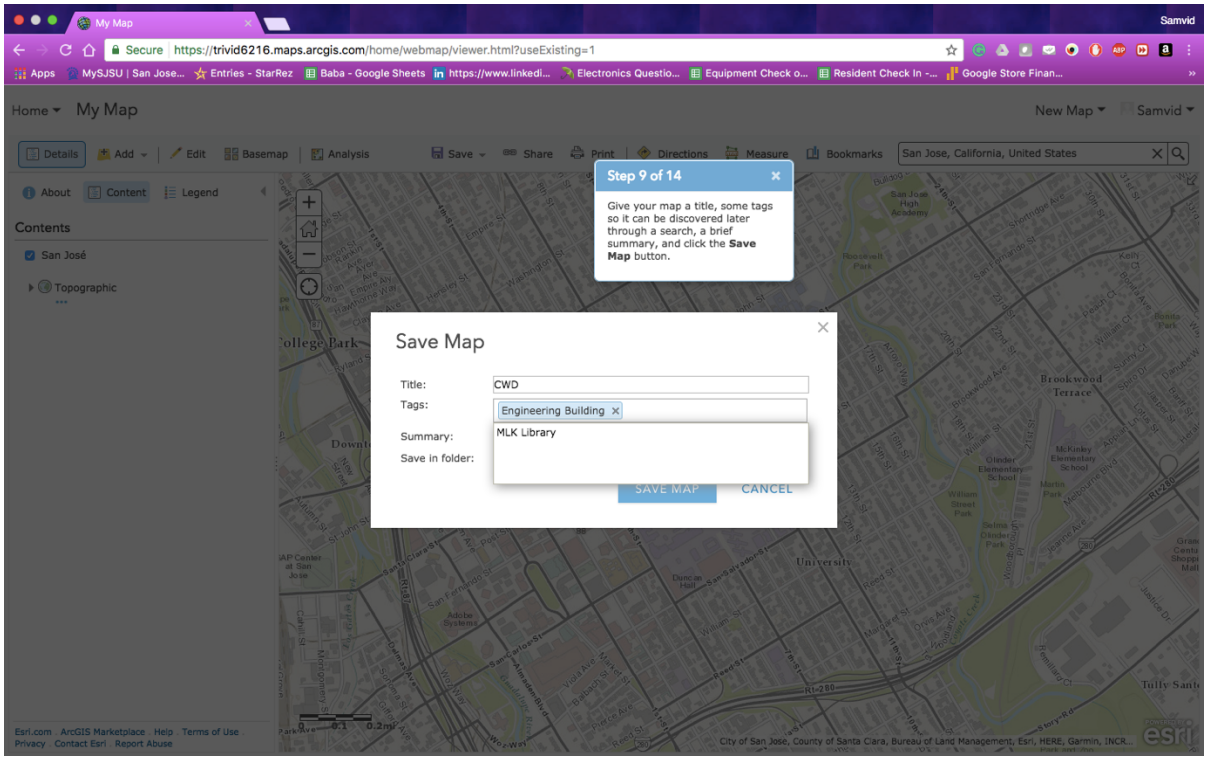


Figure 23. ArcGIS Edit Map Step-9: Saving the map

Step 10: Sharing the edited map: This step 10 is about to share the map after saving the maps and explaining every detail of it. The main aim of sharing the data is to help the others to know about the places. Figure 24 shows the data sharing in the maps.

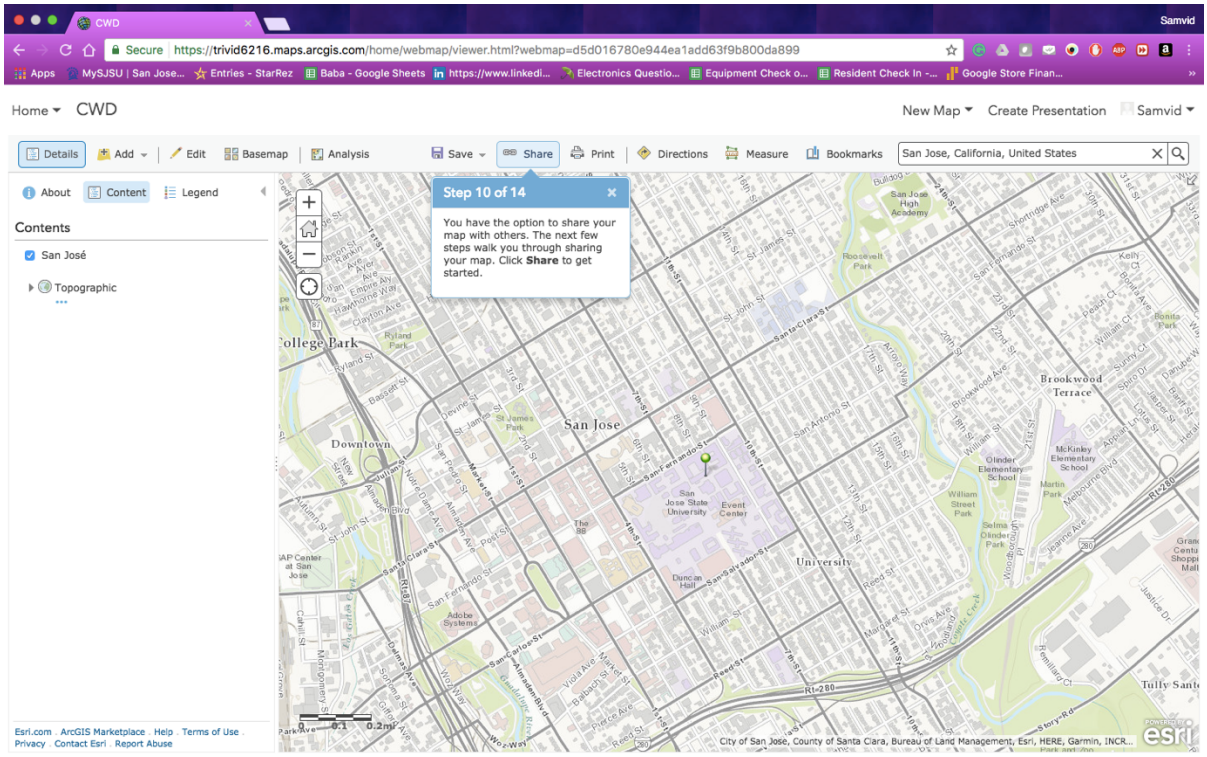


Figure 24. ArcGIS Edit Map Step-10: Sharing the map

Step 11: Sharing the edited map: This step 11 is about to share the map after saving the maps and explaining every detail of it. The main aim of sharing the data is to help the others to know about the places to the organization “San Jose State University.” Figure 25 shows the data sharing in the maps.

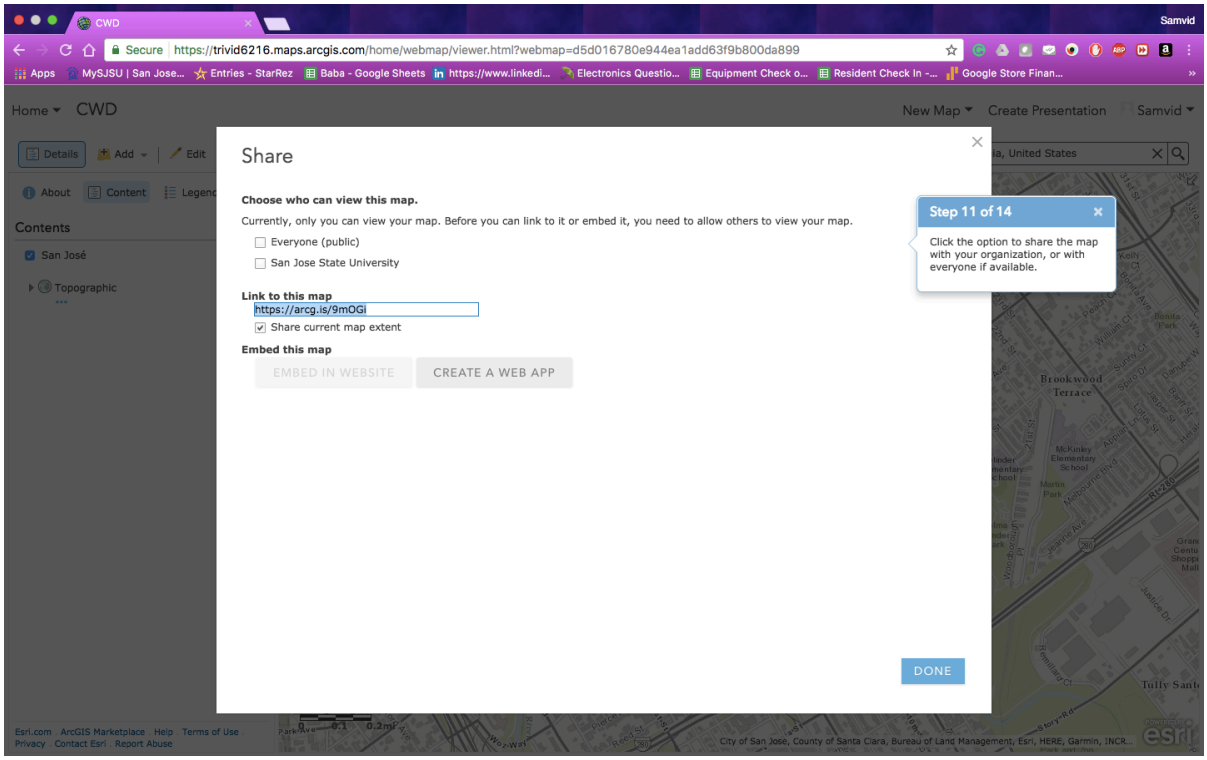


Figure 25. ArcGIS Edit Map Step-11

Step 12: Sharing the edited map: The link provided in URL link will have to copy and paste it another place. The user can share on its other media. Figure 26 shows in details of the above-mentioned step.

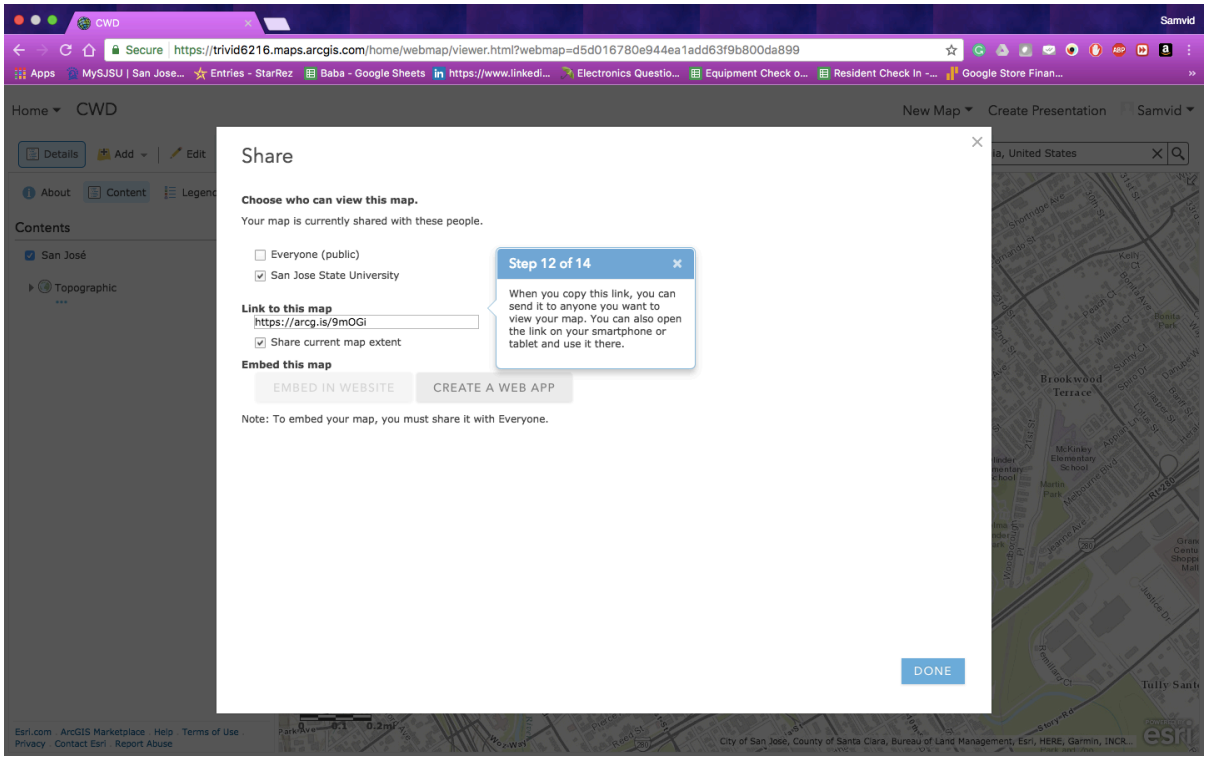


Figure 26. ArcGIS Edit Map Step-12

The next step after sharing the data the configure window will ask the user to update the information in the link and will give some general information like latitude, longitude, elevation and more. The configure part is shown in figure 27.

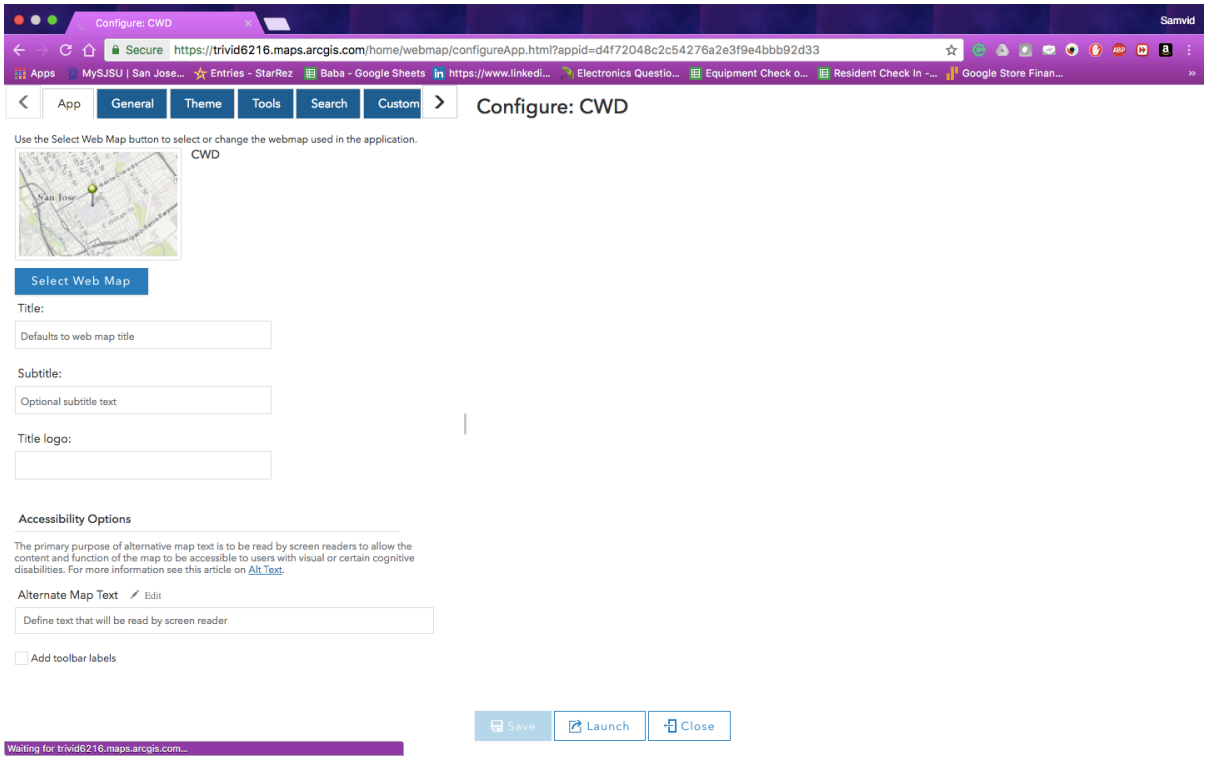


Figure 27. ArcGIS Outcome: Configuring Part of Map

Figure 28 and 29 shows the final outcome with stick pin at “Charles W. Davidson College of Engineering,” San Jose State University, California, United State of America. CWD is the short name given to the “Charles W. Davidson College of Engineering.”.

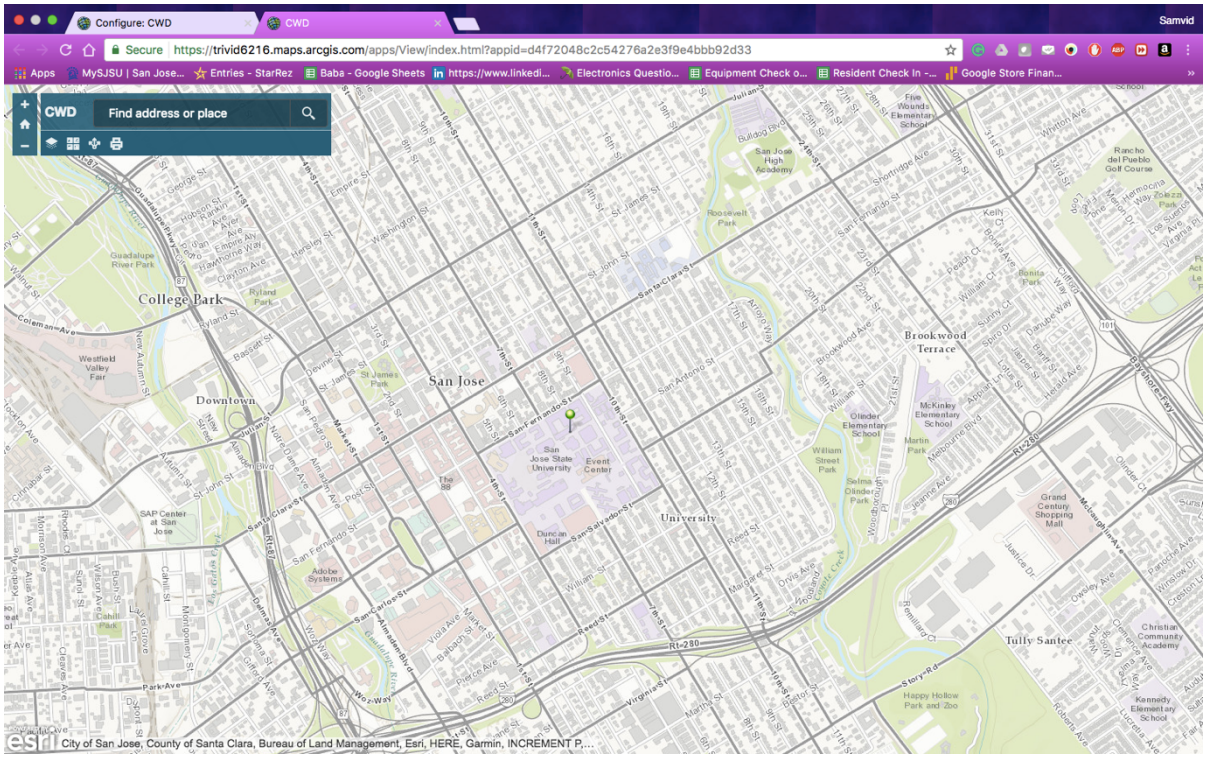


Figure 28. ArcGIS Outcome

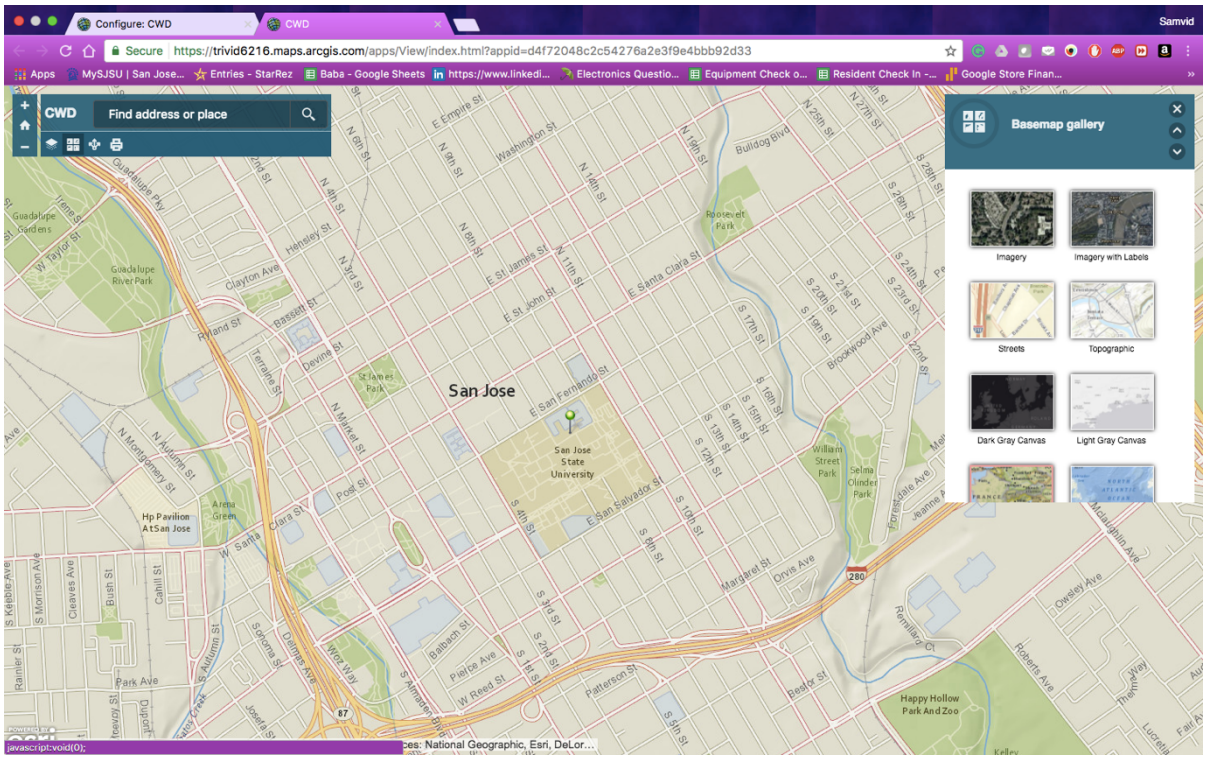


Figure 29. ArcGIS Outcome

The Application Programming Interface is used for generating maps and outlines of the maps. The basic needs for API are required in Mapping for Raspberry Pi 3. The Real Time Operating System basically requires APIs to be used by synchronizing the camera with augmented reality and maps. The APIs is provided from Here Technologies as mentioned above for the project purposes.

The ArcGIS tool will provide APIs for particular things to the user who requires it. The main aim behind it is that to use the maps and geographic location of a user with GPS module. The works as the same mobile device save the location with an image for the memory purposes. It uses GPS signal provided through a mobile network.

4. Augmented Reality

The augmented reality is implementing an object (animation or shape) in the real world. The augmented reality is basically an application of Computer Vision. Computer Vision. The Augmented Reality can be created using many Augmented Reality Tools. The Recognition of any shape or object with help of Augmented Reality tools and camera. The main algorithm to follow to get an object or shape synchronized with the real world. The main thing is to select the object and acquiring and figure the object.

The Unity 3D is a tool that is used to create Augmented Reality. This tool is basically being used for making an augmented reality for games. It is basically one type of game

mechanism that recognizes the object and implements it on the real world with help of Camera. This camera is known as Augmented Reality Camera.

4.1 Augmented Reality Camera (AR Camera)

The Augmented Reality Camera is basically the main element of the Augmented Reality. The camera will take an object as a reference then try to implement it with the real word. To select AR Camera in Unity 3D. Go to Assets > Vuforia > Prefabs > AR Camera. The AR Camera works with a camera that is integrated with a device like a laptop, personal computer, PDAs, Navigational Device with GPS Antenna and Module. The object will be at the center of a Three-Dimensional image. The object scene in the Figure is at the central of the axis in a three-dimensional view. The object is a teapot and the same implies with arrows used in Augmented Reality for Navigation.

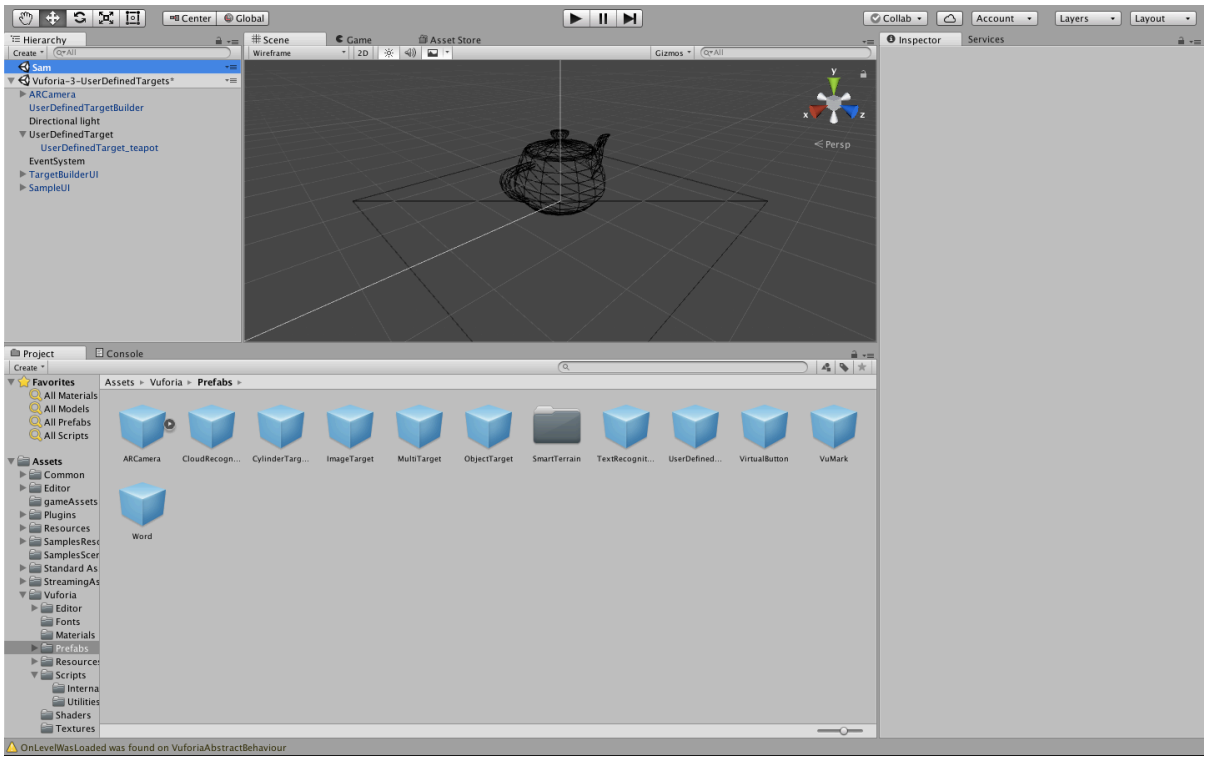


Figure 30. Unity 3D ARCamera

The Unity 3D software and tool is basically a graphical representation of the real world and object. The Vuforia Assets is basically the placement of Augmented Reality Camera and objects through Vuforia Development Portal for Mac OS or Windows. The Figure shows the login page of Vuforia Development portal.

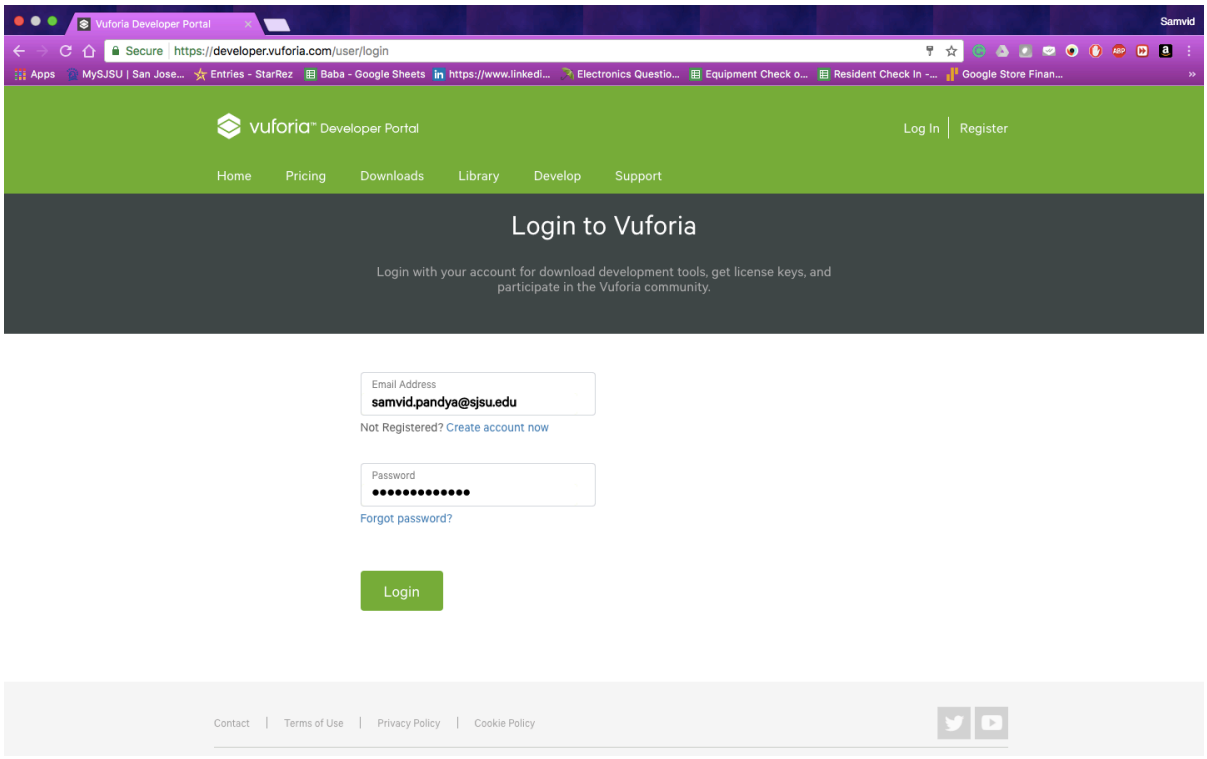


Figure 31. Login page of vuforia development portal

The license manager will add the license key for every object the user has to License key to AR Camera for AR Camera's license key. The target manager has the object name and object image. Figure 32 and 33 explains the above-mentioned detail.

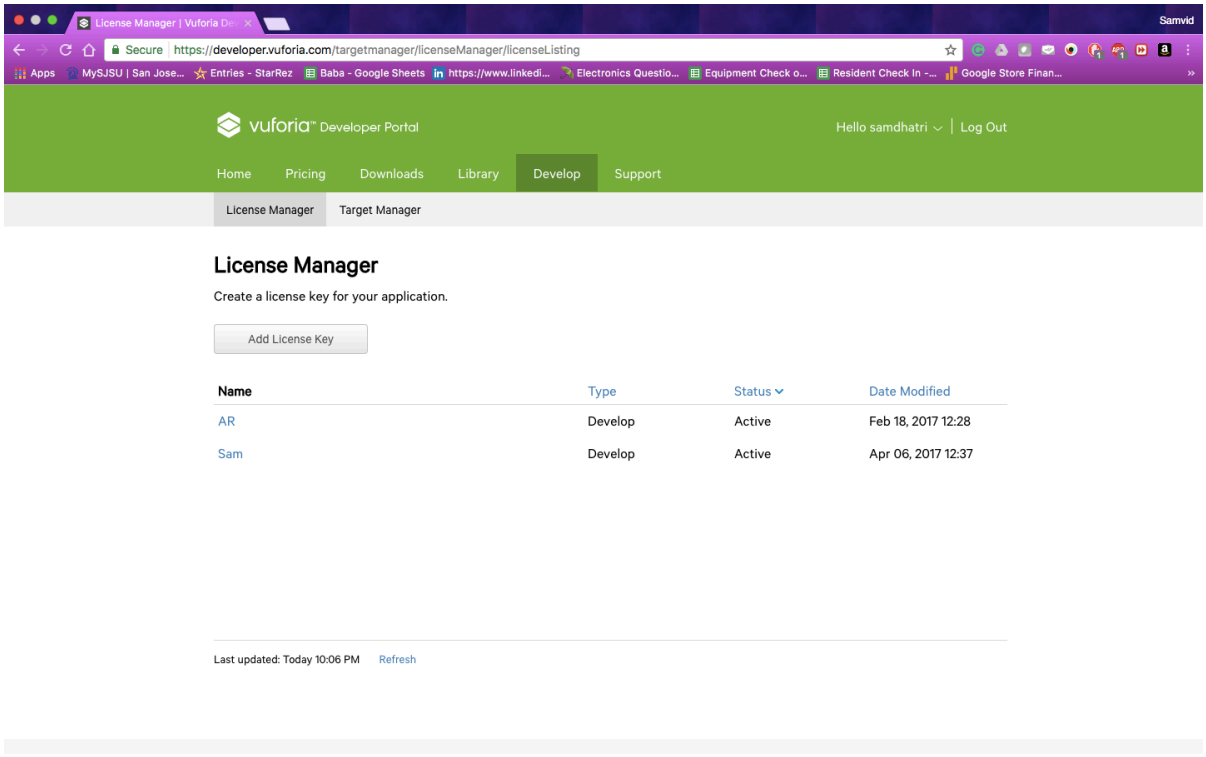


Figure 32. License manager

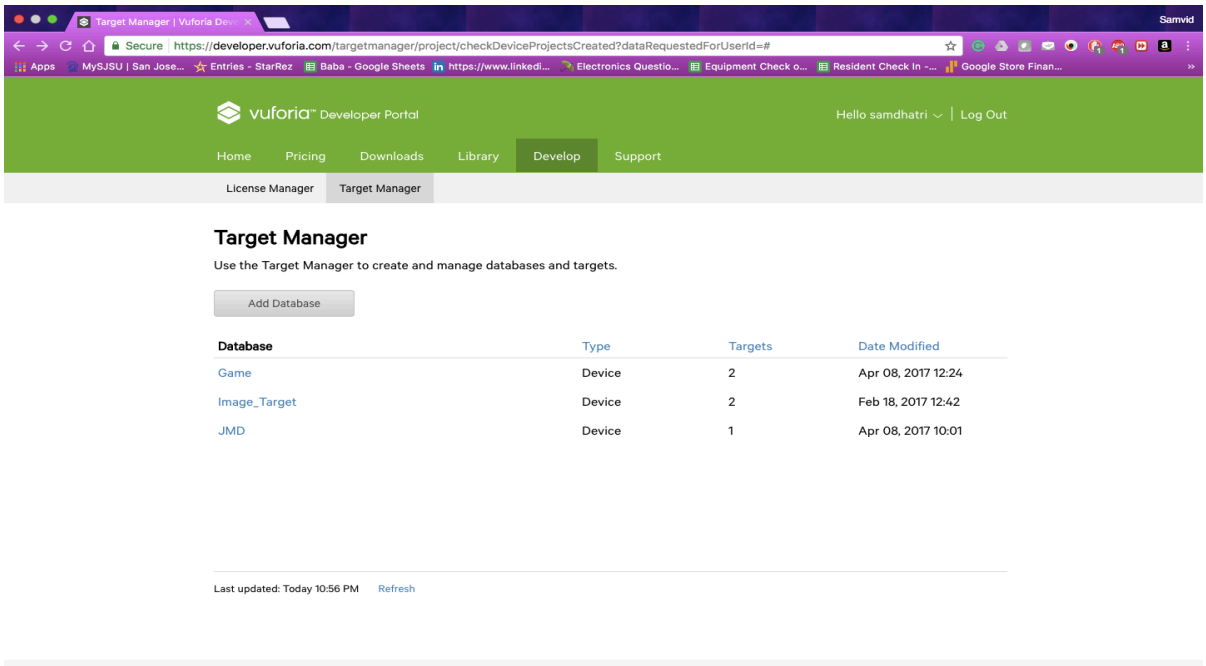


Figure 33. Target manager

The object is seen in Figure 34. A teapot. The teapot's picture is taken using a mobile camera and then will be used for the object. The three-dimensional object is shown in three axes, the object will be seen according to axes. The scale of an object will be decided through scale option in vuforia development and Unity 3D.

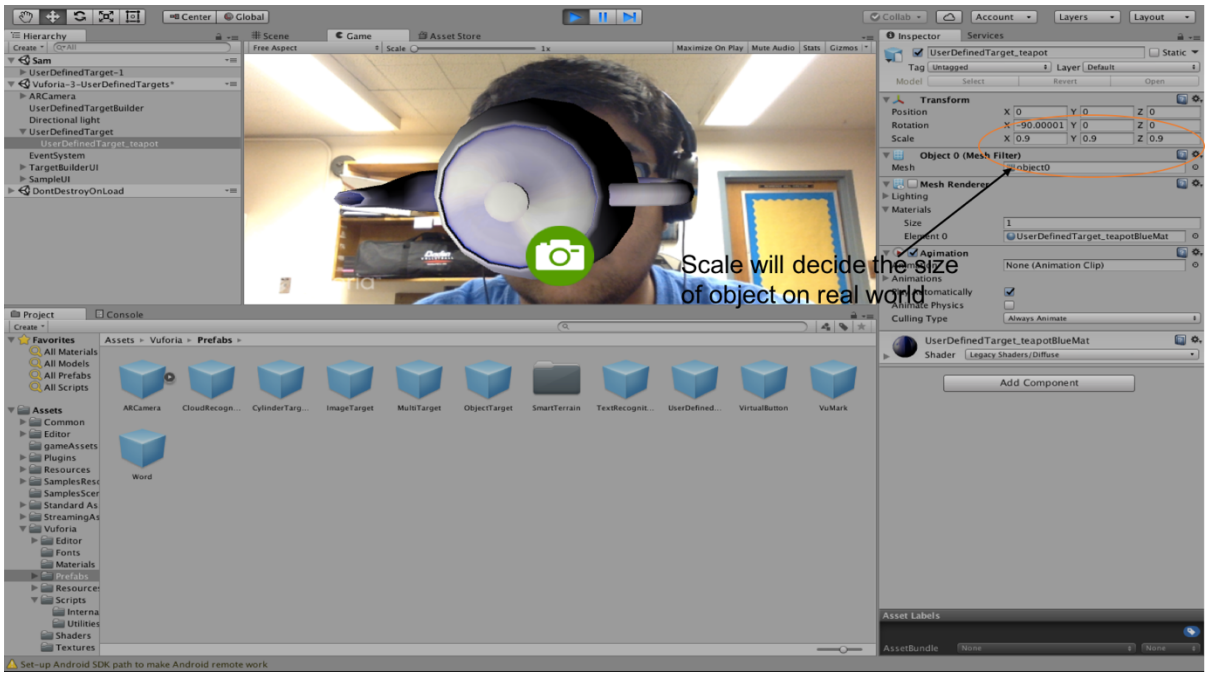


Figure 34. Augmented reality with Scale

5. Raspberry Pi 3 Model B

5.1 Introduction

When hearing the word like "Raspberry Pi" everyone wonders what this is about. By hearing the name raspberry pi, everyone may think that this must be a name of any sweet dish. But obviously, it is not about food for this project. So, first, some introductory information is explained here to get familiar with this new technology. Raspberry pi is nothing but an integrated circuit board which works as a computer. Sometimes it is also considered as a mini computer. Like a computer, all the required configurations are integrated on a single chip. Sometimes it also uses as an FPGA board depends on various applications. Now, several

versions of Raspberry pi has released in the market. These versions are raspberry pi Zero, raspberry pi 1, raspberry pi 2 and raspberry pi 3. The last one is the latest version with the latest technology with the addition of all of the configurations needed for the computer, other peripheral hardware, and software. So, raspberry pi 3 is used in this project. Specifically, model B of raspberry pi 3 is taken into consideration.

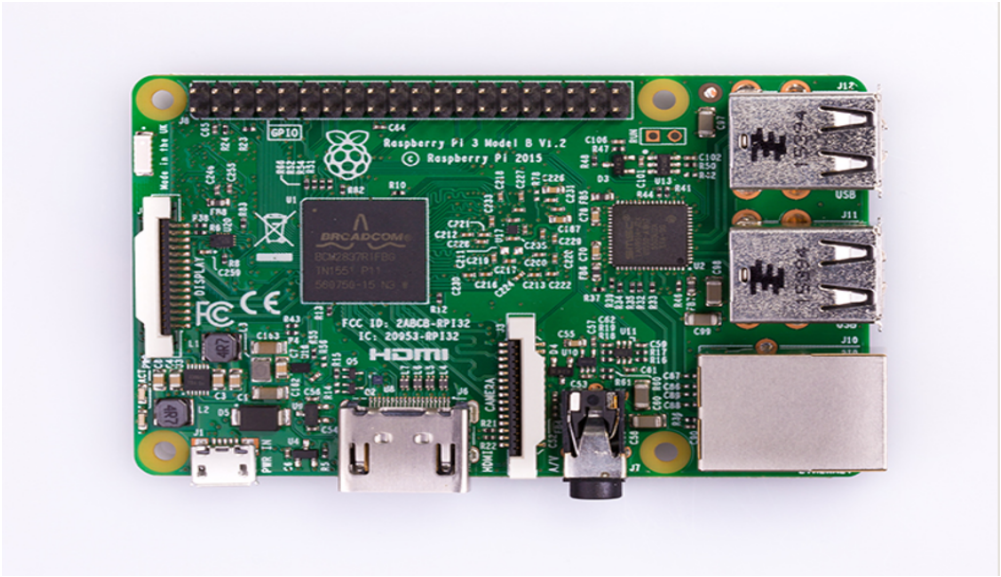


Figure 35: Raspberry Pi 3 Model B

Above figure (figure 35) shows the hardware of raspberry pi 3 model B. It is the third version of raspberry pi. Now, the model shown in figure 35, is very important for this project because of its specifications. So, to understand how this raspberry pi 3 is important, its specifications are discussed here.

5.2 Specifications of Raspberry Pi 3 Model B

As mentioned above, this is the third generation of raspberry pi. As it is an updated version, it has many new features added in it. It has System on Chip (SoC) of Broadcom BCM 2837. Compared to version 2, it is faster. Furthermore, 1.2 GHz quad-core ARM Cortex A53 with ARMv8 (instruction set) works as a processor in raspberry pi 3 model B. Broadcom Video

core IV with 400 MHz performs the role of GPU. As a memory, it has 1 GB of SDRAM. Compared to previous versions, various specs like USB ports (4 ports), Bluetooth 4.0, Wi-Fi, HDMI ports for video / audio, Slot for Camera, SD card slot for loading an operating system or for storage purpose, Ethernet slot, slot for display and slots for various general purpose input/ output (GPIO) are added in this version of raspberry pi. By this, it can be said that raspberry pi 3 model B can work as a mini computer. It is a mirror of a computer. So, most of the tasks or applications that can be done with a desktop computer, the same can be done with raspberry pi 3. The plus point is that it is portable as CPU and display both are on the single chip, unlike desktop computers. Thus, there is a long list of advantages of raspberry pi 3 through which numerous applications can be implemented on raspberry pi 3 model B.

5.3 Use of Raspberry Pi 3 Model B in this project

To meet the objective of this project - navigation with augmented reality on raspberry pi 3, first it is necessary to boot up the raspberry pi 3 with a real time operating system. As real time operating system is the base for augmented reality, first and foremost it is necessary to load the real-time operating system on raspberry pi 3. To accomplish this purpose, first real time operating system is designed on a desktop computer as mentioned in RTOS section. Then with the help of SD card of 32 GB, RTOS is loaded into SD card. Now, as mentioned above, raspberry pi 3 has a slot for memory card, this SD card is configured on the board and RTOS is boot up with the help of memory card. Now, raspberry pi 3 board has its real-time operating system.

The next step is an implementation of Global Positioning System (GPS) for navigation purpose. To achieve this, GPS code is implemented and to improve accuracy ArcGIS tool is used. To know more about this in detail, it is explained in Navigation section. The next level is a use of display and camera slots to implement and to observe augmented reality which is designed with the help of unity 3D and vuforia tool. How this augmented reality is created is mentioned in Augmented Reality Section? How these display and camera slots used on raspberry pi 3 board is explained in detail in the next section of implementation on raspberry pi 3 (Simulation and results). Thus, the goal is achieved on raspberry pi 3 Model B.

6. Implementation on Raspberry Pi Three (Simulation and Results)

The implementation of project on raspberry pi 3 model will as the first of all development of Real Time Operating System on the Raspberry Pi 3. As Raspberry Pi 3 working on an Operating System that drives the Raspberry Pi 3 model B.

The Raspberry Pi 3 Model B is booted with Real Time Operating System with help an SD Card. This SD Card is used to store the C code of Real Time Operating System. This code has some kernel or root values which will work as a base for Raspberry Pi 3. After booting the Real Time Operating System on Raspberry Pi 3 it will take some time to execute the instruction given it to like to add a camera, add a display.

6.1 SD Card

The SD Card is basically used as the storage device for Raspberry Pi 3. It requires for the booting process of Raspberry Pi 3 for any given operating system. The main requirement to boot the system is to have roots or kernel. This kernel will be then saved into SD Card. The

SD Card is also used to save data coming from the GPS Transmitter to store the information of latitude and longitude.

6.2 Camera and Display

The camera is used in Raspberry Pi 3 for creating Augmented Reality. The camera will take picture and implement the augmented object on the display used for the Raspberry Pi 3. The camera is 5 Mega Pixel High Definition Camera specially designed for Raspberry Pi 3 Model B. The Camera is connected and updated through terminal windows with “sudo apt-get update” command. The connection is shown in below figure. The Display is used to see the results of the final result.

Figure 36. Camera installation on Raspberry Pi 3

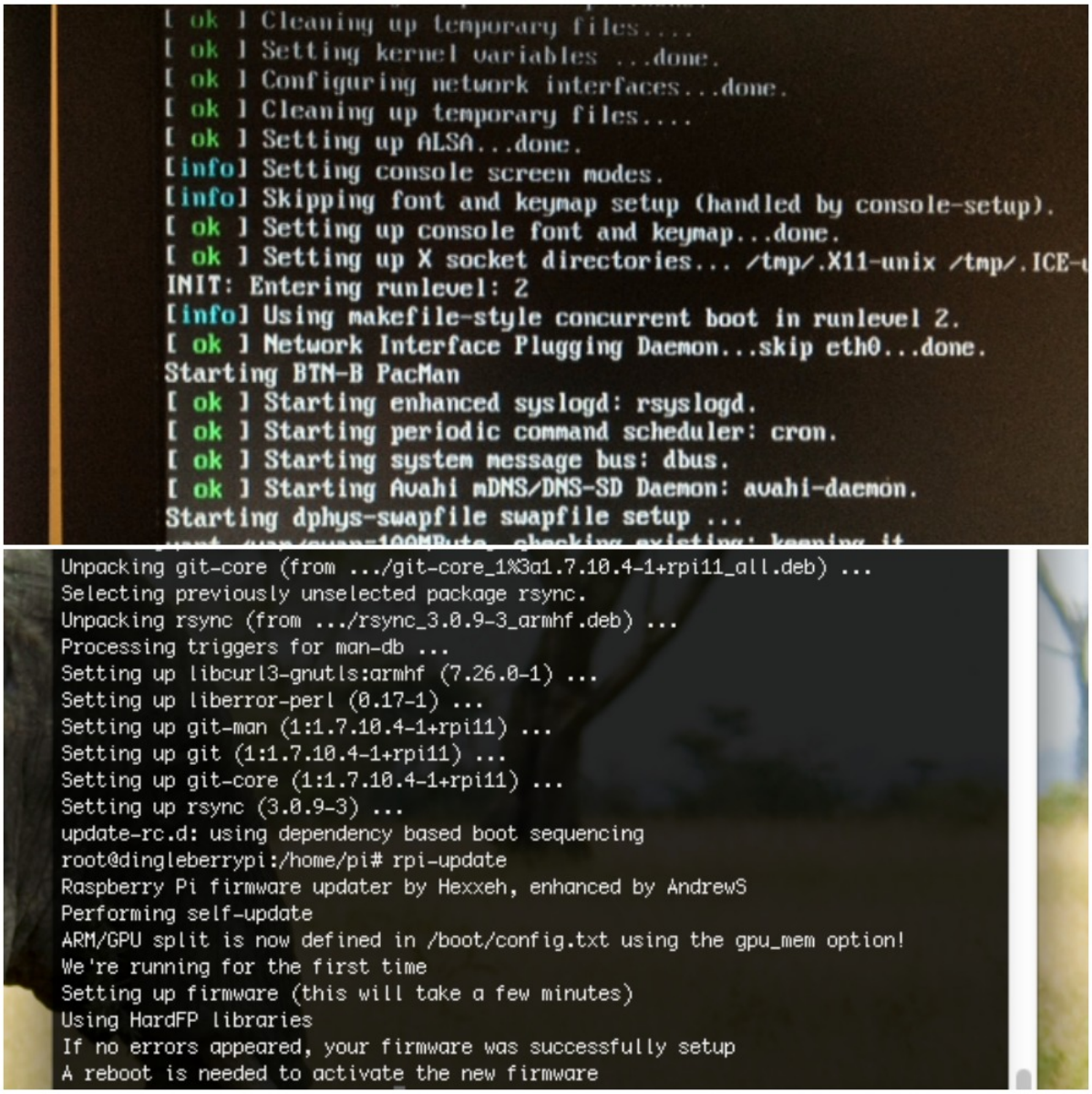


Figure 37. RTOS booting on Raspberry Pi 3 on Windows and MAC OS X

6.3 Result

The outcome of this project is shown below in the figure. The Description of figure is that the navigation is done with help of Augmented Reality. Various triangles and shape and stick pins are used to notify the user how to follow the path according to the maps. The stick pin with written 1 is suggest that the user has to take right/left from that point of interest to reach the destination. The path is basically between 101 E San Fernando Street and 377 S 7th

Street going across Robert D. Clark Hall, Student Union Inc. of SJSU. To the Event Center of SJSU.

Figure 38. Navigation with Augmented Reality on Raspberry Pi 3: Robert D Clark Hall

Figure 39. Navigation with Augmented Reality on Raspberry Pi 3: Student Union Inc., SJSU

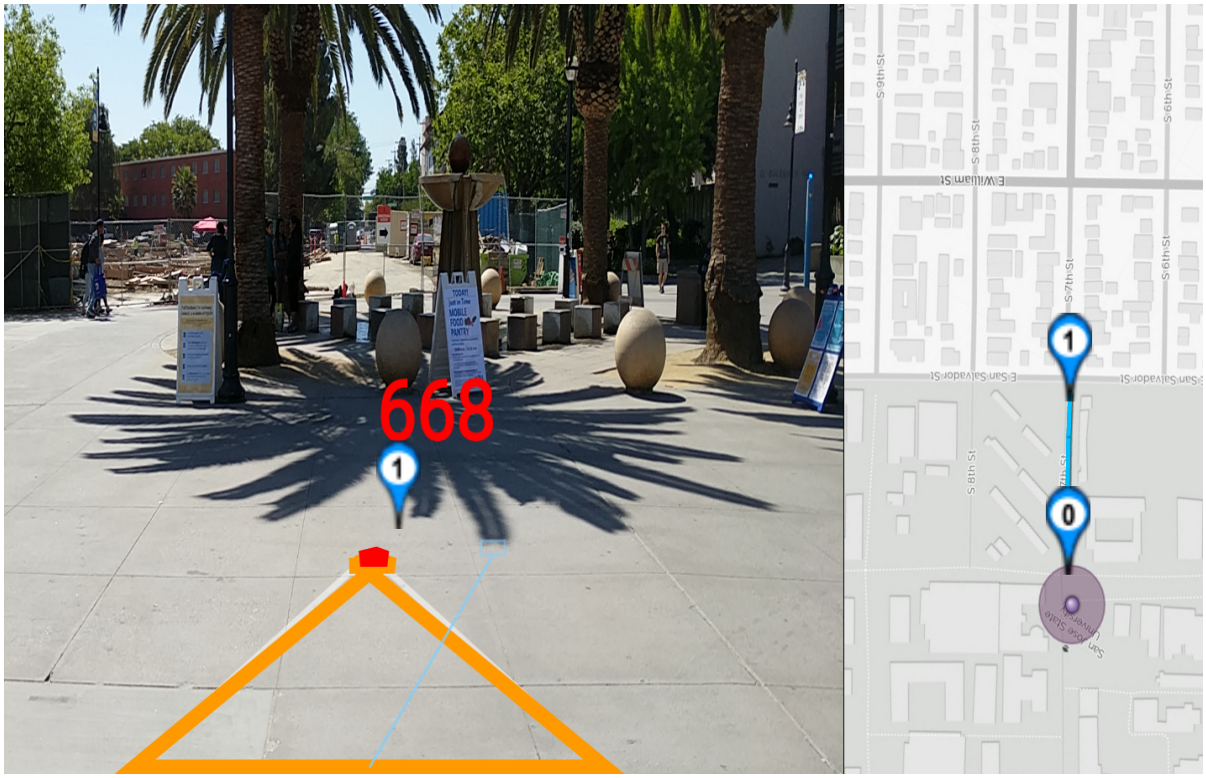


Figure 40. Destination is 668 Feet ahead. Event Center SJSU

7. Conclusion

In a nutshell, this project is about an implementation of a navigation system with the addition of augmented reality feature. As a platform for implementation, Raspberry Pi 3 Model B is configured with required peripheral devices. As to make the navigation system works in augmented reality, first augmented reality is implemented. AR works in real time as the response is quick and faster. For this, real time operating system is designed in C language. To make the use of raspberry pi 3 board designed RTOS is loaded into a board with the help of SD card. The second step is to design navigation system. For that GPS code is also designed in C language using IDE. With the help of GPS, a user can get latitude and longitude of the entered destination address. This GPS code is also loaded into a board using SD card. In

navigation, to improve an accuracy of path and direction of the destination place, ArcGIS tool is also used.

At last, as the goal is to extend this with augmented reality, Unity 3D and Vuforia tools are used to design Augmented Reality and RTOS is the base for it. AR is the presentation of 3D images or real world view of computer generated 2D inputs of any type like images or video. So, whenever a user enters the destination address as an input to this whole structure, GPS code calculates the path and direction to destination address from the current location or desired location. This designated path with its direction is given as the input to AR tools. With the help of these AR tools, a path in 2-dimentional superimposes onto real world 3-dimentional view. So, in the result, once the destination address is entered as an input, a user will get the path with the direction to a destination in 3-dimentional real world image quickly or in fewer seconds.

The significance of this project is that it enhances the concept of a conventional 2D navigation system to the 3-dimentional navigation system with the view of the physical world by including new technology of AR. Moreover, the use of raspberry pi 3 gives the advantage of portability, less area and power, faster response and better performance due to all in one chip. So, if a user is at any unknown place or even at a known place, the user can easily reach to his destination with the help of this structure.

If this project will be made on a big scale, there can be a great future scope which can give big rise to Augmented Reality. With the extension of this project, the user can get a three-dimensional view of destination place with its detailed information like other nearer places associated with entered destination address, information like opening-closing time of that

place, specialties, and an importance of that place. So, there are numerous opportunities by extending this project on a big scale.

8. References

[1] A.H. Behzadan and V.R. Kamat, "Visualization of construction graphics in outdoor augmented reality", in Simulation Conferences, 2005, 27th International Conference of IEEE, December 4, 2005

[2] Azuma R., J. W. Lee, B. Jiang, and J. Park. 1999. Tracking in unprepared environments for Augmented Reality systems. Computers and Graphics 23(6): 787-793

[3] J.H. Lin, C.M. Lin and C.R. Dow, "Design and Implement Augmented Reality For Supporting Driving Visual Guidance", in 2011 Second International Conference on Innovations in Bio-inspired Computing and Applications, IEEE, 2011.

[4] W. Narzt, G. Pomberger, A. Ferscha, C. Lindlinger, "Augmented Reality Navigation Systems",in Johannes Kepler University, Linz, Austria, in 2004.

[5] Ono, Ooe, Sakamoto, "Navigation and Communication Systems", 1989 International Conference on Consumer Electronics, IEEE, 1989

[6] J.J.Hall, B.Erol, J. Graham, Q. Ke , H. Kishi, J. Moraleda, D. G. Van Olst, “Paper-Based Augmented Reality”, 17th International Conference Artificial Reality and Telexistence(ICAT), 28-30 November, 2007. Jylland, Denmark, 2007

[7] W.E. McKay, “Augmented Reality: Linking real and virtual worlds A new paradigm for interacting with computers”, in University of Paris, 1999

[8] Su-Lim Tan, T.N. Bao Anh, “Real Time Operating System for small microcontroller”, 13th International Symposium for Consumer Electronics, IEEE, 2009, Kyoto, Japan

APPENDIX A

Appendix A consists of navigational functional code. The Functional Code of Global Positioning System is basically working with raspberry pi 3 and will receive the GPS signals coming from GPS Satellite. The code will give the information about latitude and longitude of the particular location. It sometimes provides the information about the number of satellites available within a radius of GPS Module.

I). Module Code

```
#include <NewSoftSerial.h>
NewSoftSerial gps_data(2,3); // (Rx, Tx)

void setup()
{
      Serial.begin(9600);
      gps_data.begin(9600);
}

void loop()
{
      if(gps_data.available())
      {
            char i = gps_data.read();
            Serial.print(i);
      }
}
```

Figure A1. Code for GPS

II). Code for saving the data

```
#include <LiquidCrystal.h>
#include <NewSoftSerial.h>
#include <TinyGPS.h>
#include <SD.h>
const int chipSelect = 10;
string dataString = "latitude,longitude";
LiquidCrystal lcd (A0, A1, A2, A3, A4, A5);

TinyGPS gps; //GPS Library
NewSoftSerial GPS(2,3); //pin 2 as Rx and pin 3 as Tx.

void setup()
{
      Serial.begin(9600); //baud rate//
      GPS.begin (9600);
      lcd.begin(16,2);
      lcd.print("gps testing…");
      delay(2000);
```

Figure A2. GPS code

```
 Serial.print("Initializing SD card…");

 pinMode(10,OUTPUT);
 pinMode(4,OUTPUT);
 lcd.setCursor(0,0);

 delay(2000);
 if(!SD.begin(chipSelect))
 {
       Serial.println("Card failed, or not present");
```

Figure A3. GPS code

```
      lcd.setCursor(0,0);
      lcd.print("Card failed");
      delay(2000);
      lcd.clear();
      return;
}
Serial.println("Card initialized");
lcd.setCursor(0,0);
lcd.print("Card initialized");
delay(2000);
lcd.clear();
delay(2000);
File dataFile = SD.open("datalog.csv", FILE_WRITE);
if(dataFile)
{
      dataFile.println(dataString);
      dataFile.close();
```

Figure A4. Data Log

```
      if(dataFile)
      {
            dataFile.println(dataString);
            dataFile.close();
            Serial.println(dataString);
      }
      else
      {
            Serial.println("error opening datalog.txt");
            lcd.setCursor(0,0);
            lcd.print("err open in file");
            delay(2000);
      }
}

void loop()
{
```

Figure A5. Datalog (continue)

```
      bool newData = false; // data from Satellite
      unsigned long chars;
      unsigned short sentences, failed;

      for(unsigned long start = millis(); millis()-start<1000;)
      {
            while(GPS.available())
            {
                  char c = GPS.read();
                  if(gps.encode(c))
                        newData = true;
            }
      }
      if(newData)
      {
            float flat, flon;
            unsigned long age;
            gps.f_get_position(&flat, &flon, &age);
```

Figure A6. Datalog

```
Serial.println(flat, 6); //LAT
Serial.println(flon, 6); // LONG
Serial.println(gps.satellite()); //SET

file dataFile = SD.open("datalog.csv", FILE_WRITE);

if(dataFile)
{
      digitalWrite(4,HIGH);
      dataFile.print(flat,6);
      dataFile.print(",");
      dataFile.println(flon,6);
```

Figure A7. GPS Data logger

```
            lcd.setCursor(0,0);
            lcd.print(flat,6);
            lcd.setCursor(0,1);
            lcd.print(flon,6);
            dataFile.close();
            digitalWrite(4,LOW);
        }
    }
}
```

Figure A8. GPS Data logger

APPENDIX B

Appendix B consists of Real Time Operating System Code. The Functional Code of Real Time Operating System will be run on Raspberry Pi 3 and work with Augmented Reality. The RTOS is required because AR works in Real Time. The following images are the code of Real Time Operating System.

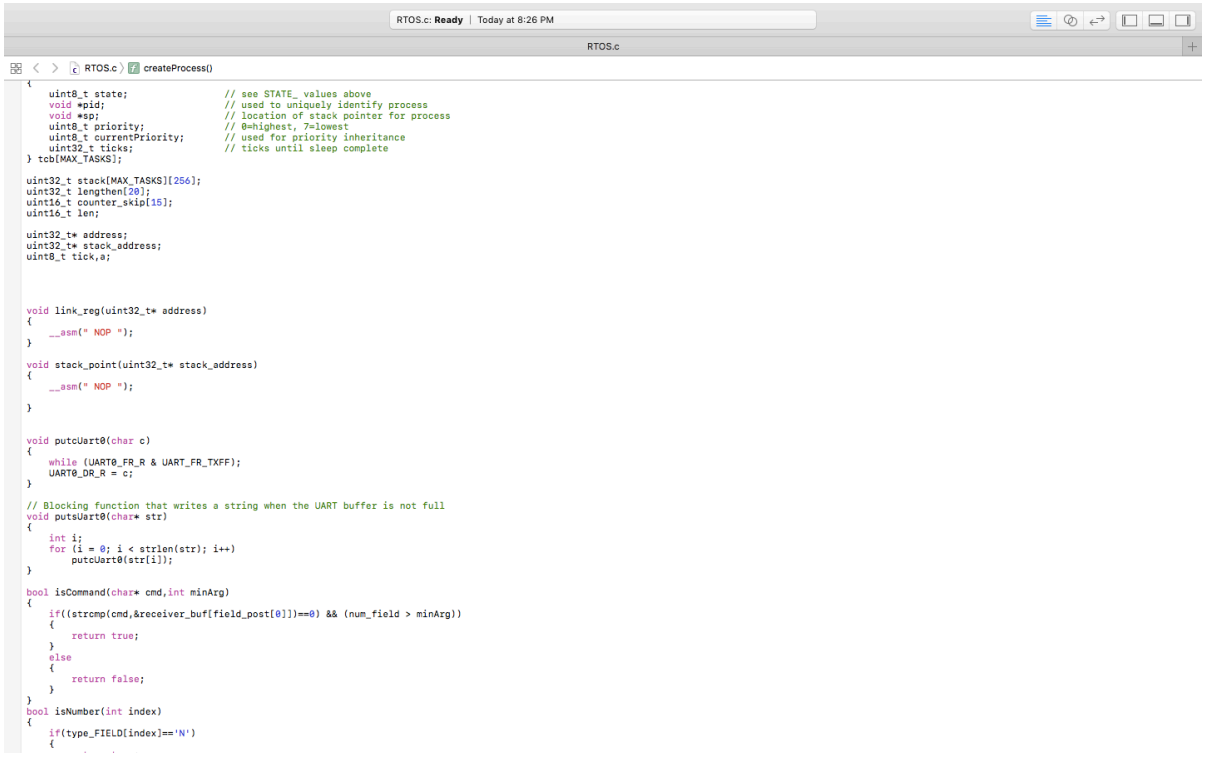


Figure B1. RTOS Code

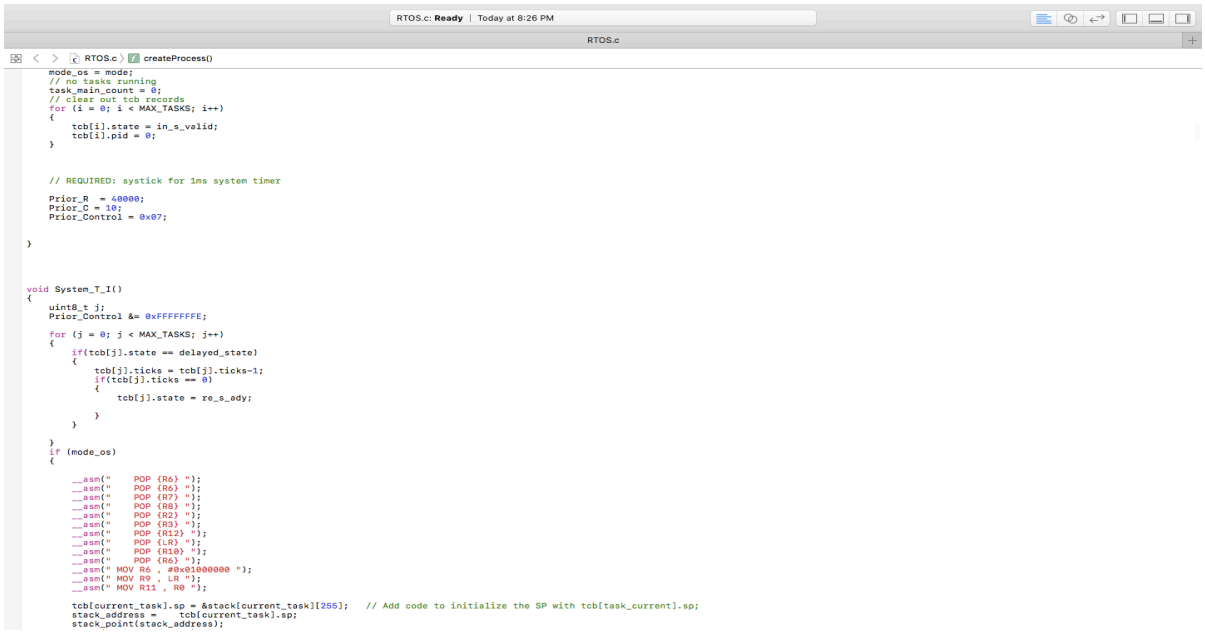


Figure B2. RTOS Code

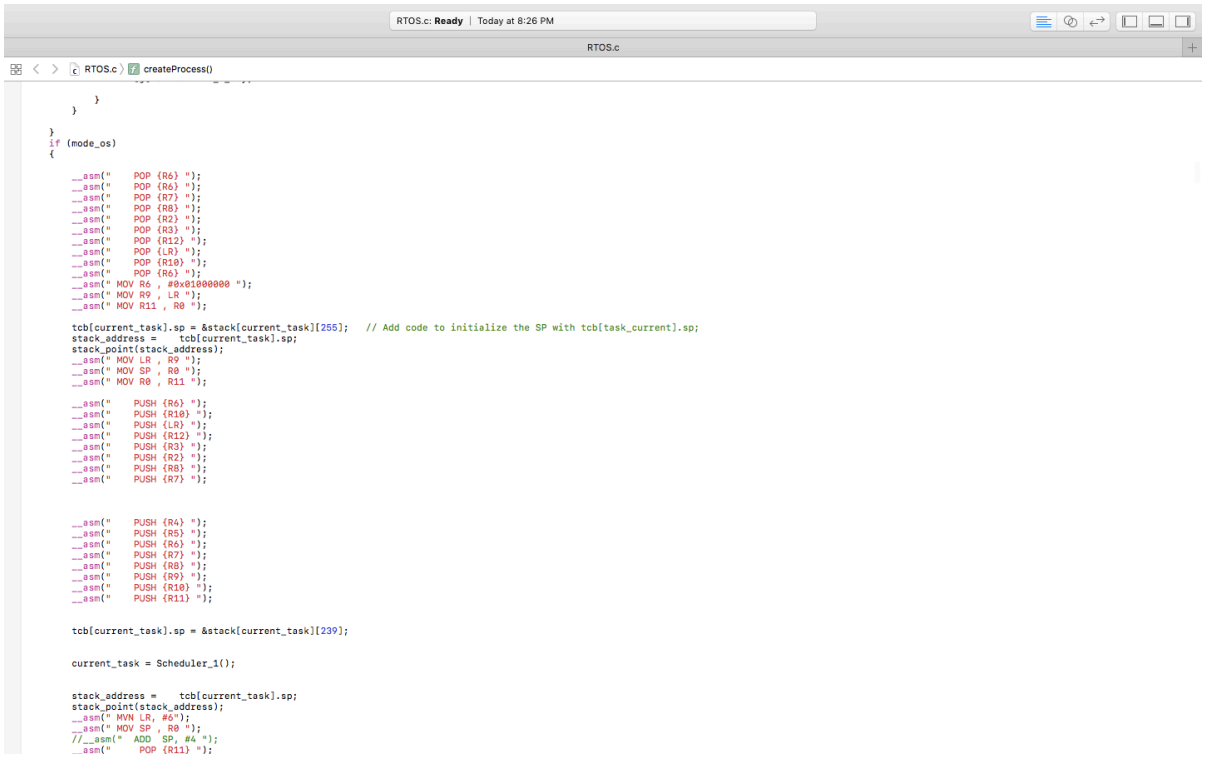


Figure B3. RTOS Code

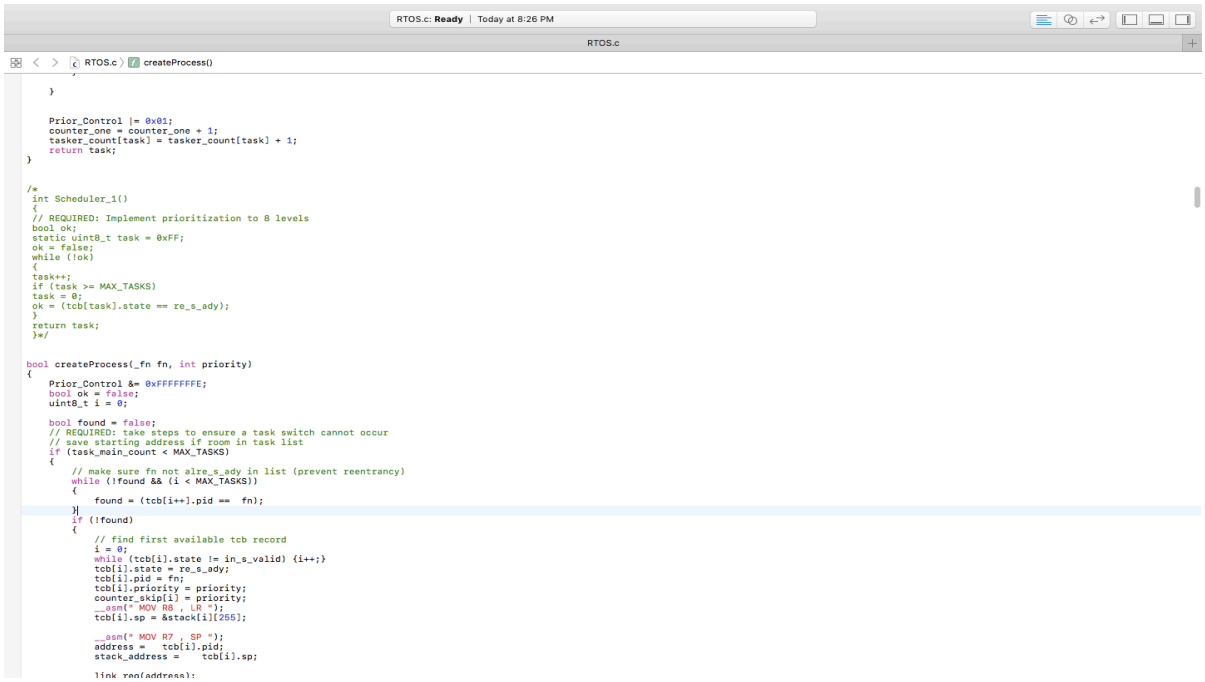


Figure B4. RTOS Code

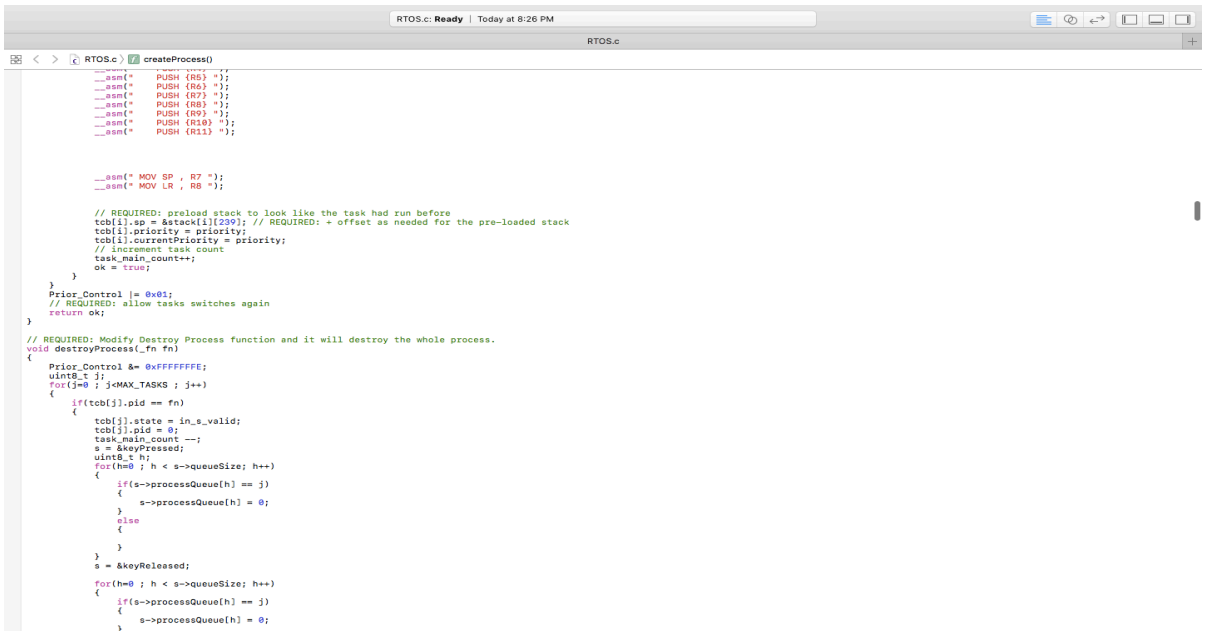


Figure B5. RTOS Code

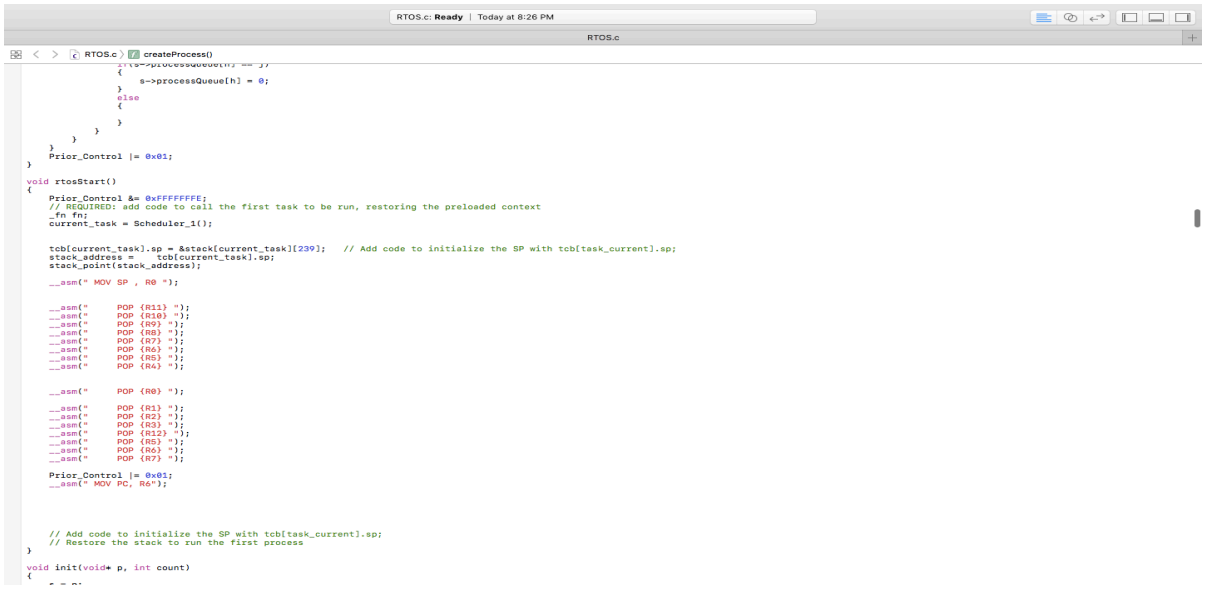


Figure B6. RTOS Code

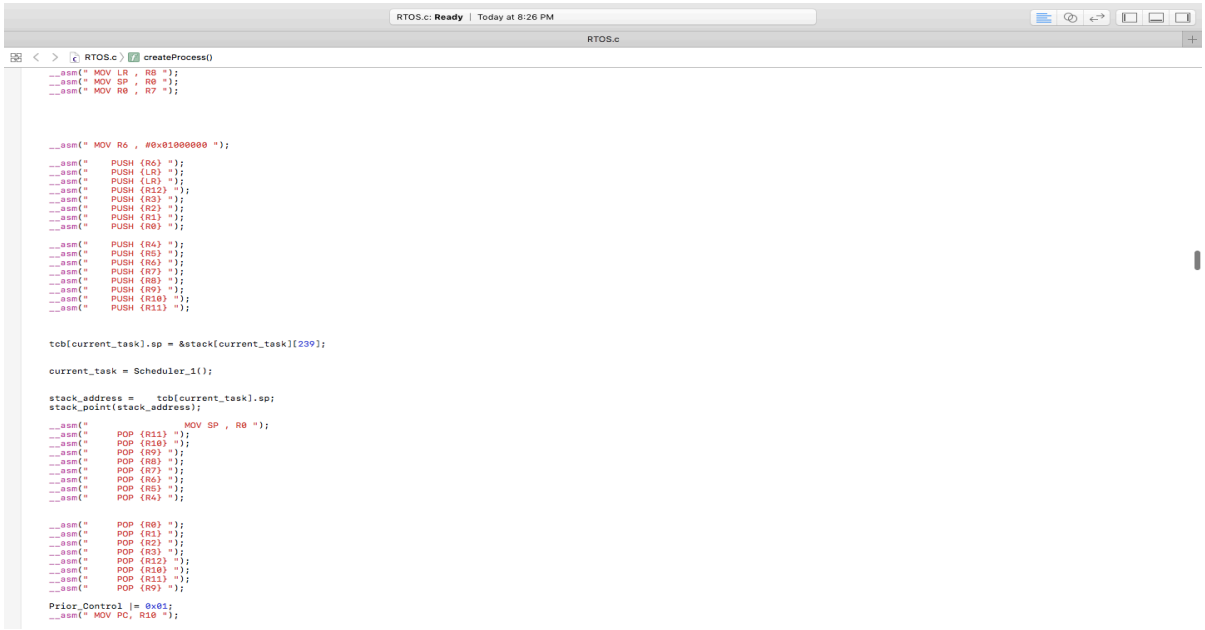


Figure B7. RTOS Code

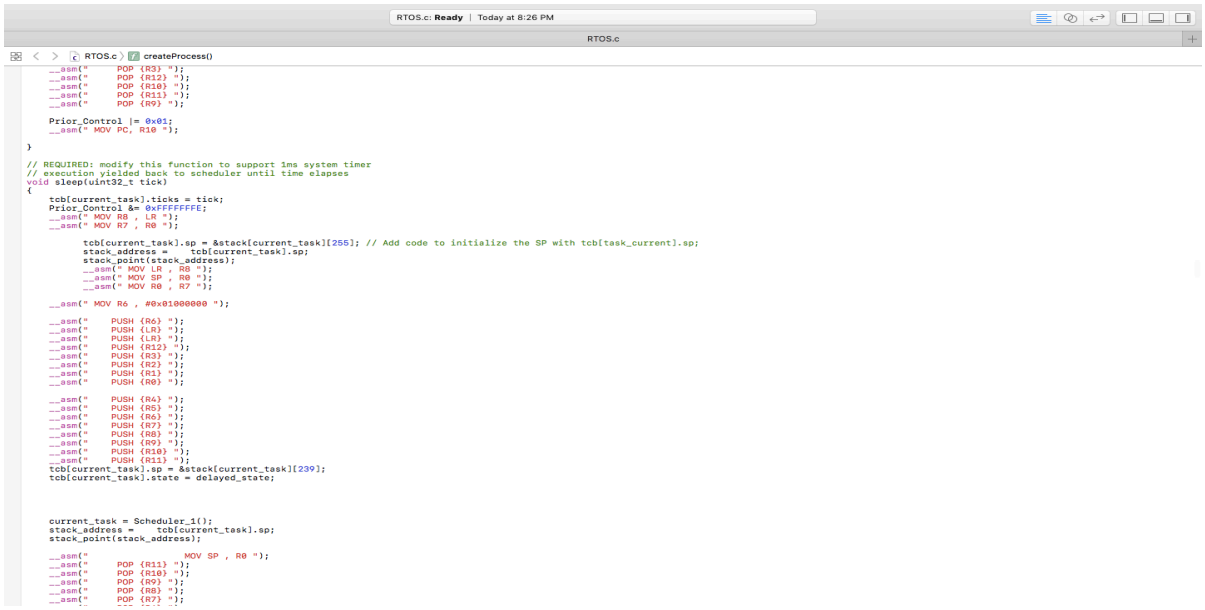


Figure B8. RTOS Code

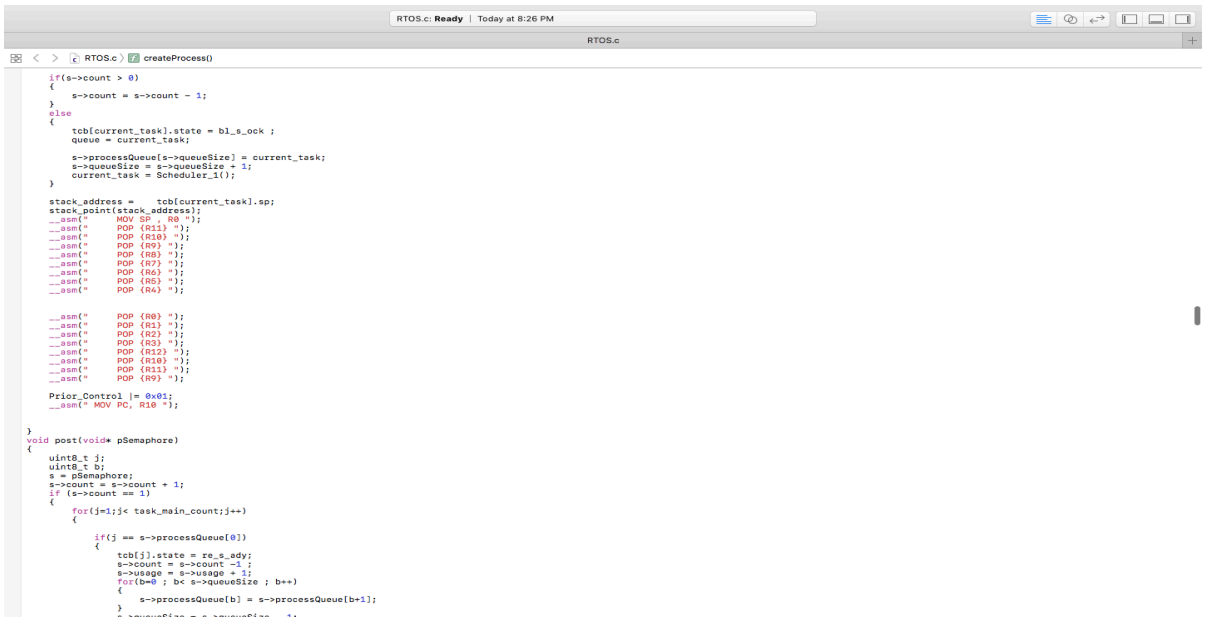
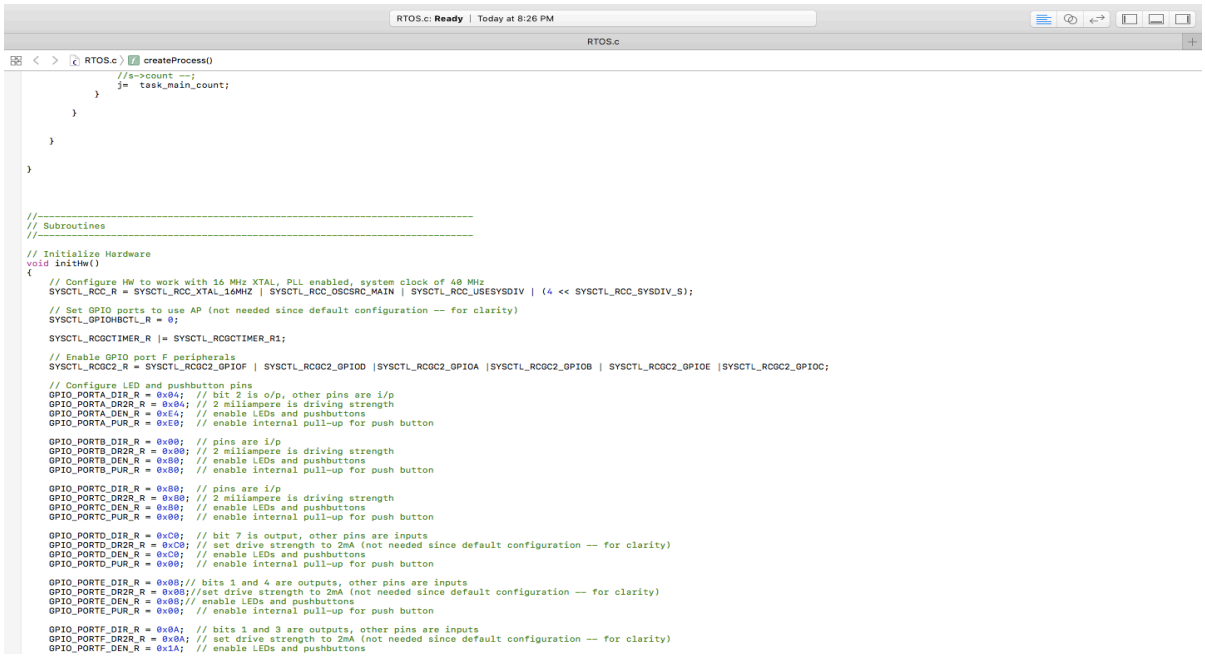
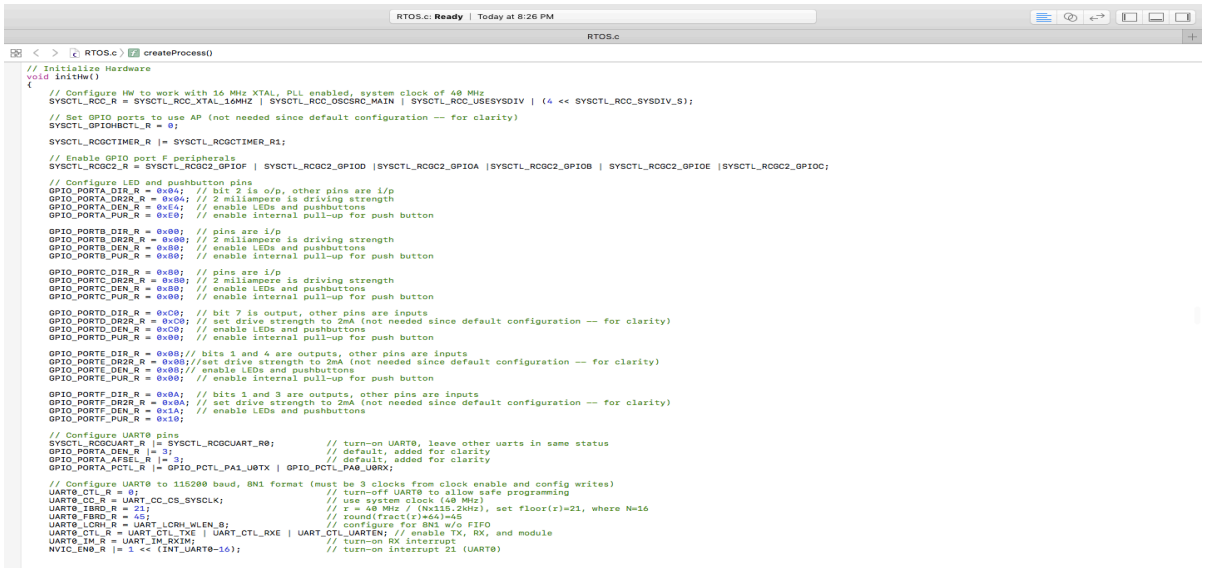


Figure B9. RTOS Code




```
                //s->count --;
                j=  task_main_count;
            }

        }


    }

}



//-----------------------------------------------------------------------------
// Subroutines
//-----------------------------------------------------------------------------

// Initialize Hardware
void initHw()
{
    // Configure HW to work with 16 MHz XTAL, PLL enabled, system clock of 40 MHz
    SYSCTL_RCC_R = SYSCTL_RCC_XTAL_16MHZ | SYSCTL_RCC_OSCSRC_MAIN | SYSCTL_RCC_USESYSDIV | (4 << SYSCTL_RCC_SYSDIV_S);

    // Set GPIO ports to use AP (not needed since default configuration -- for clarity)
    SYSCTL_GPIOHBCTL_R = 0;

    SYSCTL_RCGCTIMER_R |= SYSCTL_RCGCTIMER_R1;

    // Enable GPIO port F peripherals
    SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOF | SYSCTL_RCGC2_GPIOD |SYSCTL_RCGC2_GPIOA |SYSCTL_RCGC2_GPIOB | SYSCTL_RCGC2_GPIOE |SYSCTL_RCGC2_GPIOC;

    // Configure LED and pushbutton pins
    GPIO_PORTA_DIR_R = 0x04;  // bit 2 is o/p, other pins are i/p
    GPIO_PORTA_DR2R_R = 0x04; // 2 miliampere is driving strength
    GPIO_PORTA_DEN_R = 0xE4;  // enable LEDs and pushbuttons
    GPIO_PORTA_PUR_R = 0xE0;  // enable internal pull-up for push button

    GPIO_PORTB_DIR_R = 0x00;  // pins are i/p
    GPIO_PORTB_DR2R_R = 0x00; // 2 miliampere is driving strength
    GPIO_PORTB_DEN_R = 0x80;  // enable LEDs and pushbuttons
    GPIO_PORTB_PUR_R = 0x80;  // enable internal pull-up for push button

    GPIO_PORTC_DIR_R = 0x80;  // pins are i/p
    GPIO_PORTC_DR2R_R = 0x80; // 2 miliampere is driving strength
    GPIO_PORTC_DEN_R = 0x80;  // enable LEDs and pushbuttons
    GPIO_PORTC_PUR_R = 0x00;  // enable internal pull-up for push button

    GPIO_PORTD_DIR_R = 0xC0;  // bit 7 is output, other pins are inputs
    GPIO_PORTD_DR2R_R = 0xC0; // set drive strength to 2mA (not needed since default configuration -- for clarity)
    GPIO_PORTD_DEN_R = 0xC0;  // enable LEDs and pushbuttons
    GPIO_PORTD_PUR_R = 0x00;  // enable internal pull-up for push button

    GPIO_PORTE_DIR_R = 0x08;// bits 1 and 4 are outputs, other pins are inputs
    GPIO_PORTE_DR2R_R = 0x08;//set drive strength to 2mA (not needed since default configuration -- for clarity)
    GPIO_PORTE_DEN_R = 0x08;// enable LEDs and pushbuttons
    GPIO_PORTE_PUR_R = 0x00;  // enable internal pull-up for push button

    GPIO_PORTF_DIR_R = 0x0A;  // bits 1 and 3 are outputs, other pins are inputs
    GPIO_PORTF_DR2R_R = 0x0A; // set drive strength to 2mA (not needed since default configuration -- for clarity)
    GPIO_PORTF_DEN_R = 0x1A;  // enable LEDs and pushbuttons
```

Figure B10. RTOS Code




```
// Initialize Hardware
void initHw()
{
    // Configure HW to work with 16 MHz XTAL, PLL enabled, system clock of 40 MHz
    SYSCTL_RCC_R = SYSCTL_RCC_XTAL_16MHZ | SYSCTL_RCC_OSCSRC_MAIN | SYSCTL_RCC_USESYSDIV | (4 << SYSCTL_RCC_SYSDIV_S);

    // Set GPIO ports to use AP (not needed since default configuration -- for clarity)
    SYSCTL_GPIOHBCTL_R = 0;

    SYSCTL_RCGCTIMER_R |= SYSCTL_RCGCTIMER_R1;

    // Enable GPIO port F peripherals
    SYSCTL_RCGC2_R = SYSCTL_RCGC2_GPIOF | SYSCTL_RCGC2_GPIOD |SYSCTL_RCGC2_GPIOA |SYSCTL_RCGC2_GPIOB | SYSCTL_RCGC2_GPIOE |SYSCTL_RCGC2_GPIOC;

    // Configure LED and pushbutton pins
    GPIO_PORTA_DIR_R = 0x04;  // bit 2 is o/p, other pins are i/p
    GPIO_PORTA_DR2R_R = 0x04; // 2 miliampere is driving strength
    GPIO_PORTA_DEN_R = 0xE4;  // enable LEDs and pushbuttons
    GPIO_PORTA_PUR_R = 0xE0;  // enable internal pull-up for push button

    GPIO_PORTB_DIR_R = 0x00;  // pins are i/p
    GPIO_PORTB_DR2R_R = 0x00; // 2 miliampere is driving strength
    GPIO_PORTB_DEN_R = 0x80;  // enable LEDs and pushbuttons
    GPIO_PORTB_PUR_R = 0x80;  // enable internal pull-up for push button

    GPIO_PORTC_DIR_R = 0x80;  // pins are i/p
    GPIO_PORTC_DR2R_R = 0x80; // 2 miliampere is driving strength
    GPIO_PORTC_DEN_R = 0x80;  // enable LEDs and pushbuttons
    GPIO_PORTC_PUR_R = 0x00;  // enable internal pull-up for push button

    GPIO_PORTD_DIR_R = 0xC0;  // bit 7 is output, other pins are inputs
    GPIO_PORTD_DR2R_R = 0xC0; // set drive strength to 2mA (not needed since default configuration -- for clarity)
    GPIO_PORTD_DEN_R = 0xC0;  // enable LEDs and pushbuttons
    GPIO_PORTD_PUR_R = 0x00;  // enable internal pull-up for push button

    GPIO_PORTE_DIR_R = 0x08;// bits 1 and 4 are outputs, other pins are inputs
    GPIO_PORTE_DR2R_R = 0x08;//set drive strength to 2mA (not needed since default configuration -- for clarity)
    GPIO_PORTE_DEN_R = 0x08;// enable LEDs and pushbuttons
    GPIO_PORTE_PUR_R = 0x00;  // enable internal pull-up for push button

    GPIO_PORTF_DIR_R = 0x0A;  // bits 1 and 3 are outputs, other pins are inputs
    GPIO_PORTF_DR2R_R = 0x0A; // set drive strength to 2mA (not needed since default configuration -- for clarity)
    GPIO_PORTF_DEN_R = 0x1A;  // enable LEDs and pushbuttons
    GPIO_PORTF_PUR_R = 0x10;

    // Configure UART0 pins
    SYSCTL_RCGCUART_R |= SYSCTL_RCGCUART_R0;         // turn-on UART0, leave other uarts in same status
    GPIO_PORTA_DEN_R |= 3;                           // default, added for clarity
    GPIO_PORTA_AFSEL_R |= 3;                         // default, added for clarity
    GPIO_PORTA_PCTL_R |= GPIO_PCTL_PA1_U0TX | GPIO_PCTL_PA0_U0RX;

    // Configure UART0 to 115200 baud, 8N1 format (must be 3 clocks from clock enable and config writes)
    UART0_CTL_R = 0;                                 // turn-off UART0 to allow safe programming
    UART0_CC_R = UART_CC_CS_SYSCLK;                  // use system clock (40 MHz)
    UART0_IBRD_R = 21;                               // r = 40 MHz / (Nx115.2kHz), set floor(r)=21, where N=16
    UART0_FBRD_R = 45;                               // round(fract(r)=64)=45
    UART0_LCRH_R = UART_LCRH_WLEN_8;                 // configure for 8N1 w/o FIFO
    UART0_CTL_R = UART_CTL_TXE | UART_CTL_RXE | UART_CTL_UARTEN; // enable TX, RX, and module
    UART0_IM_R = UART_IM_RXIM;                       // turn-on RX interrupt
    NVIC_EN0_R |= 1 << (INT_UART0-16);               // turn-on interrupt 21 (UART0)
```

Figure B11. RTOS Code

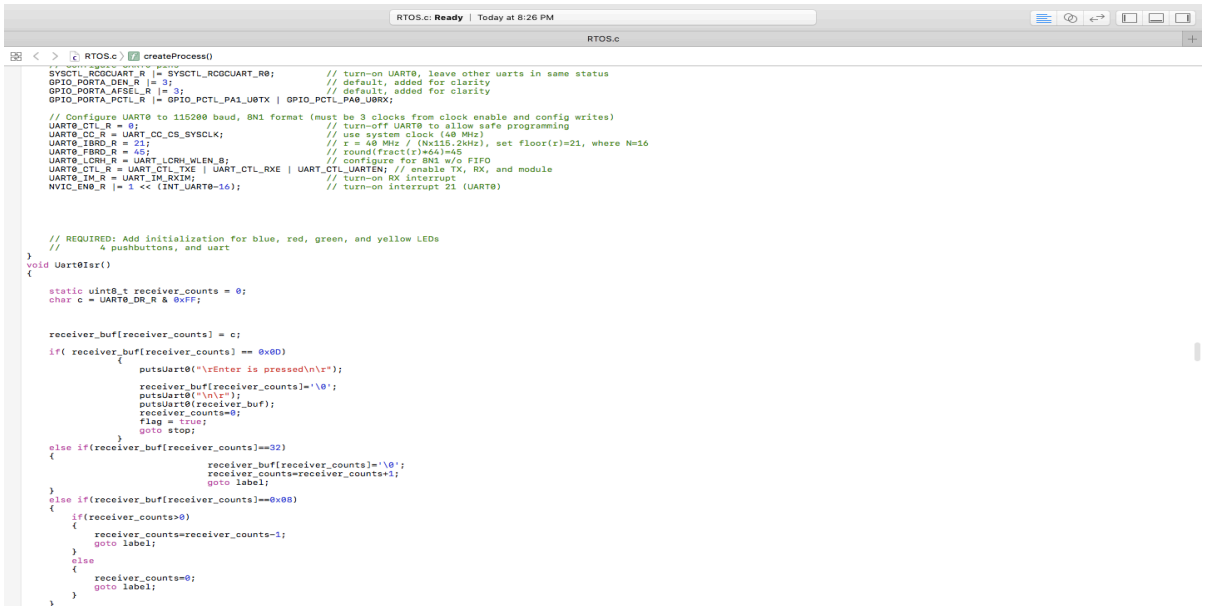


Figure B12. RTOS Code

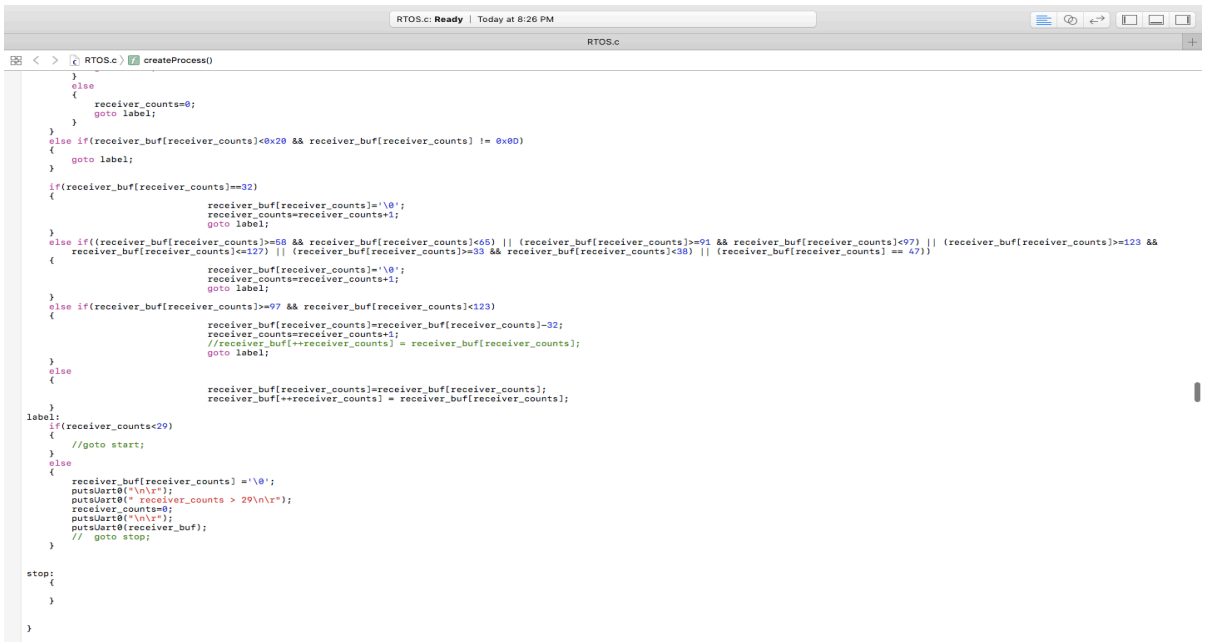


Figure B13. RTOS Code

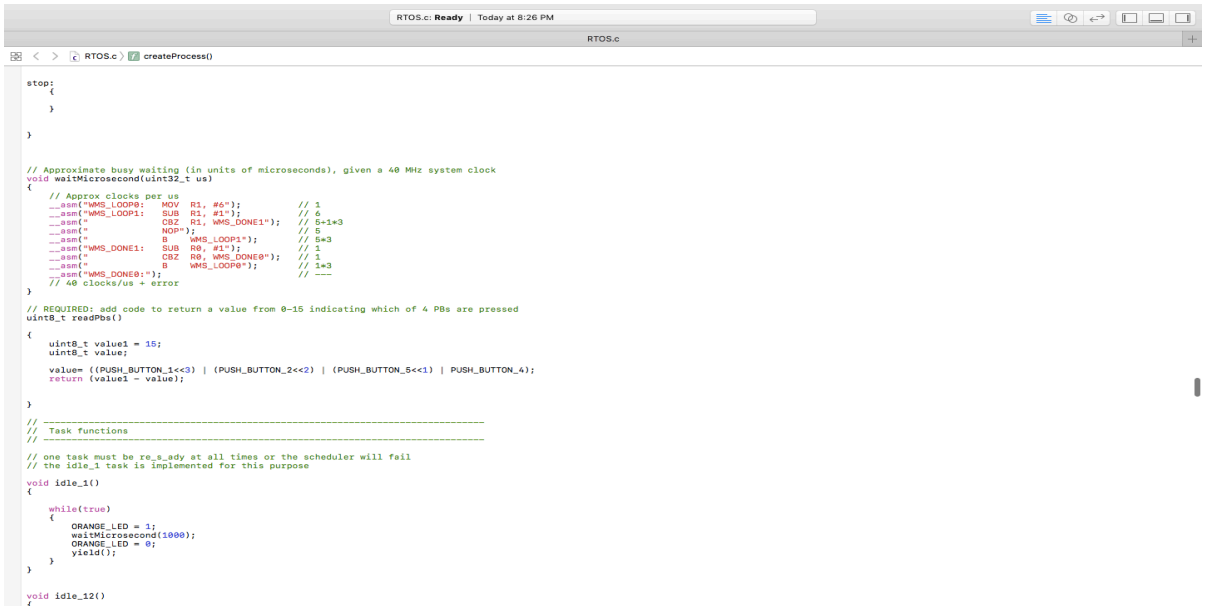


Figure B14. RTOS Code

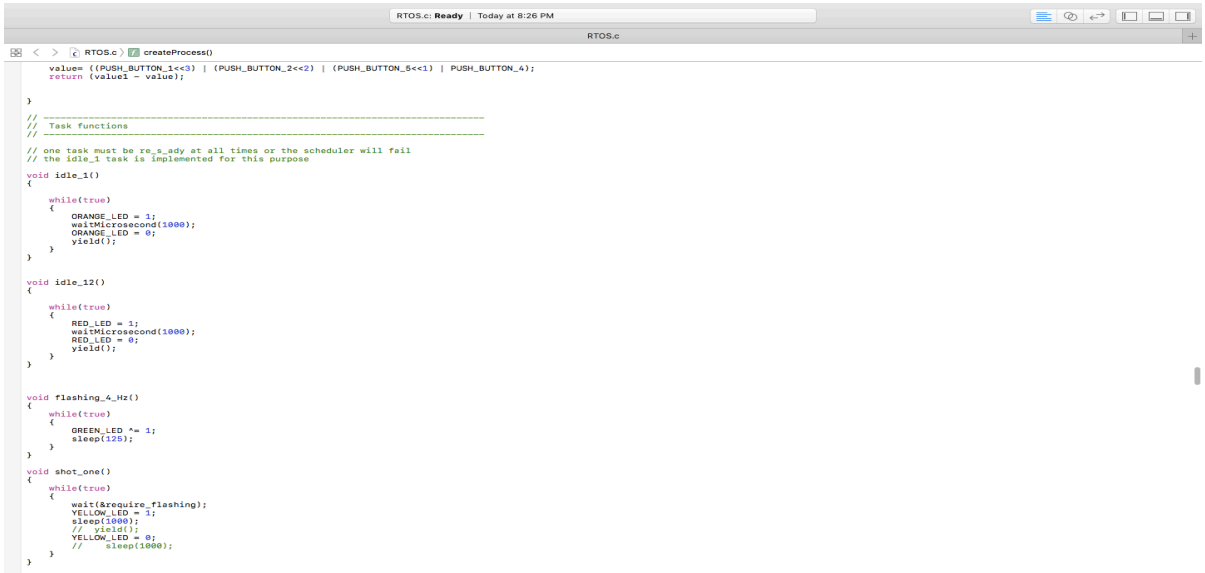


Figure B15. RTOS Code

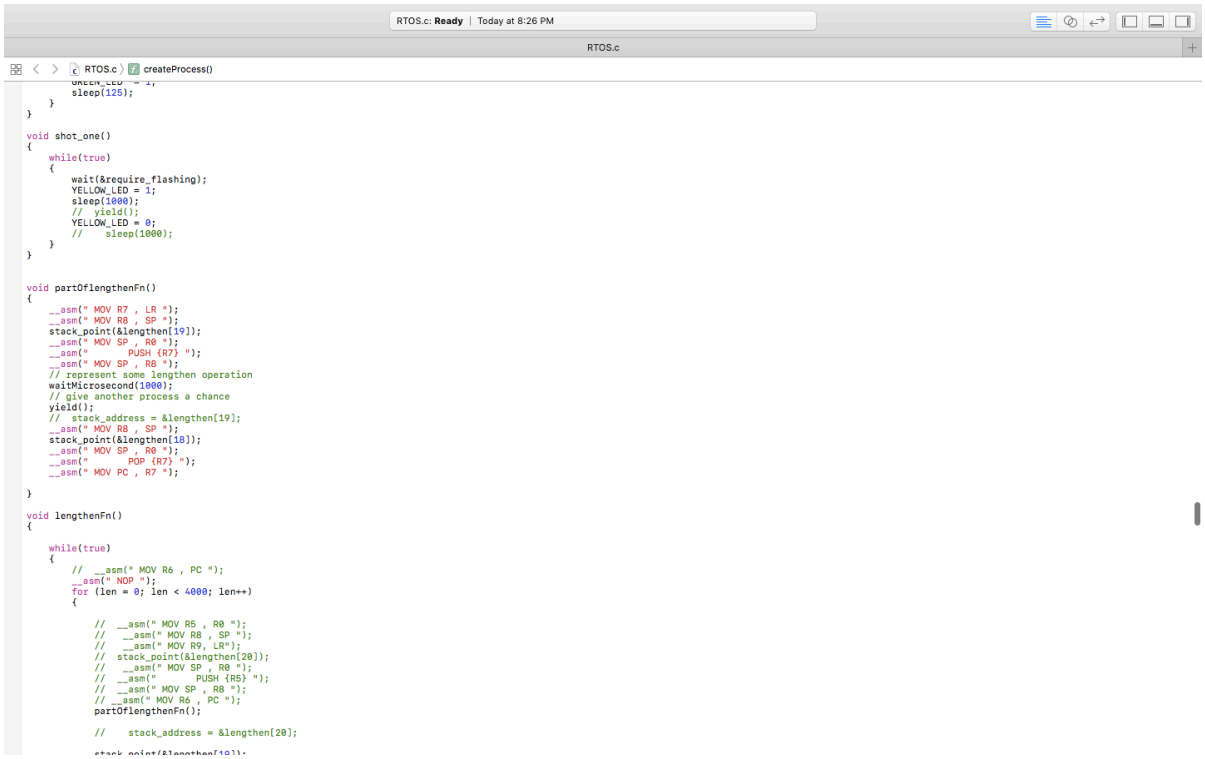


Figure B16. RTOS Code

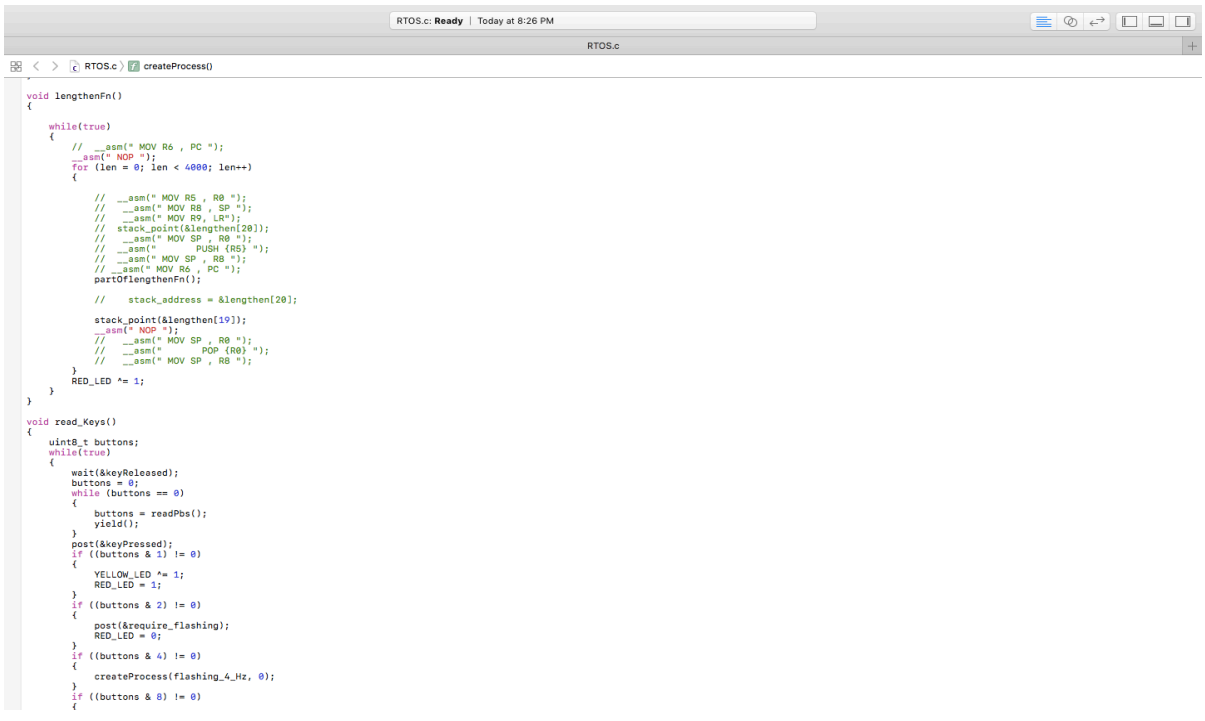


Figure B17. RTOS Code

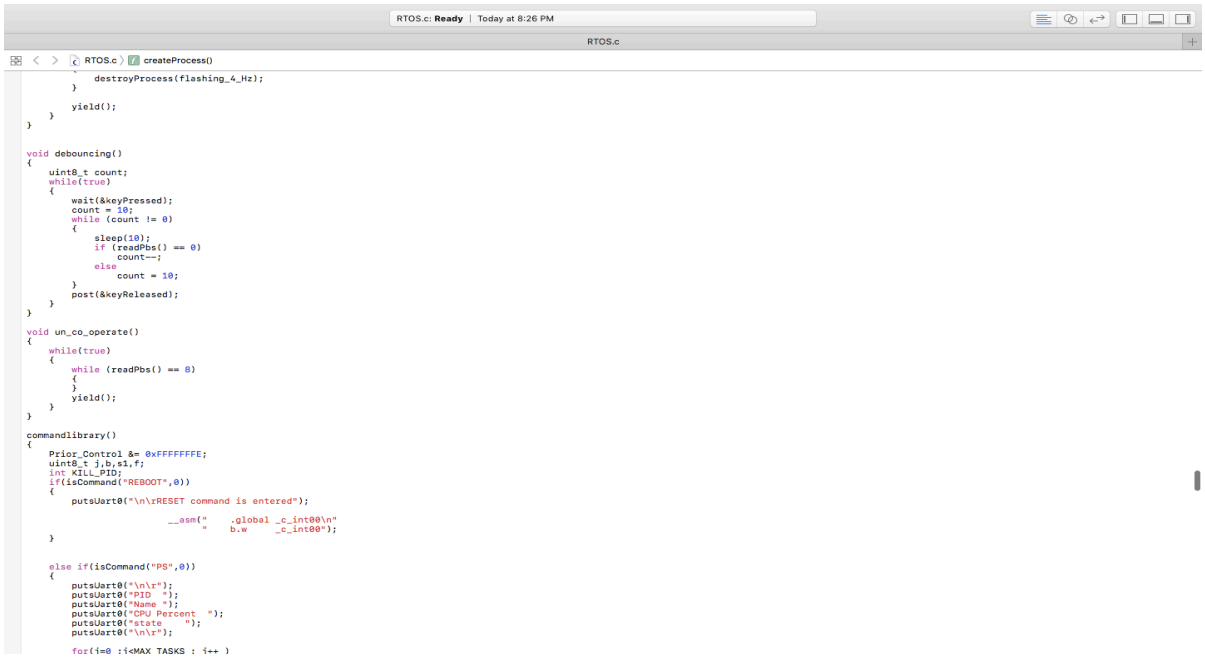


Figure B18. RTOS Code

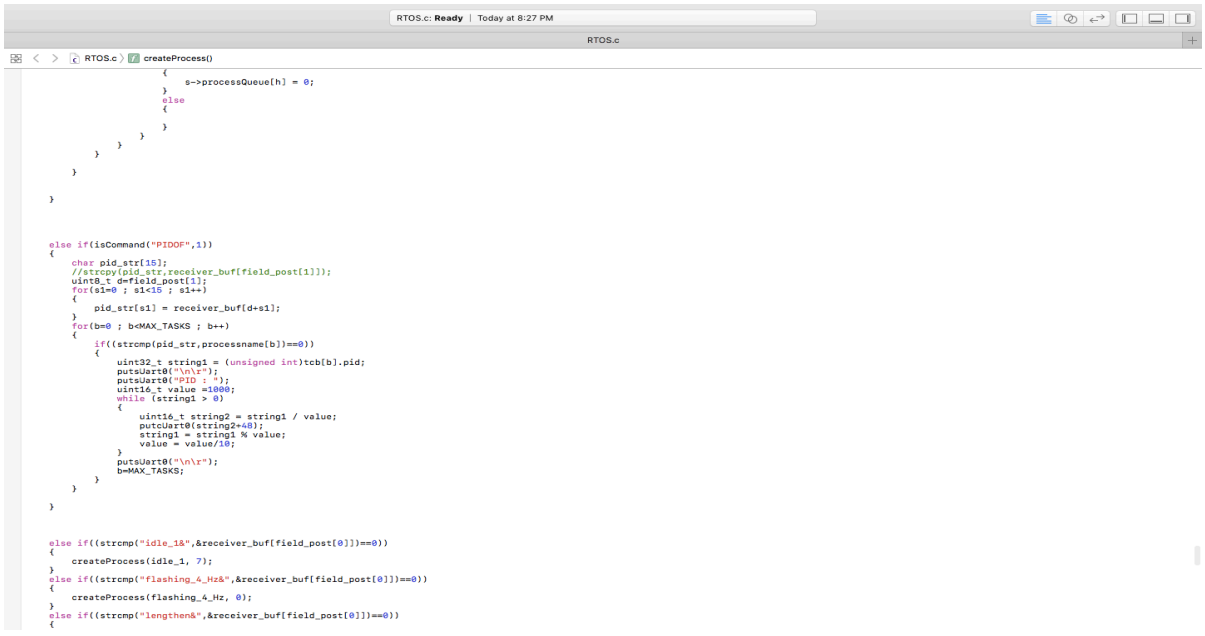


Figure B19. RTOS Code