Final Report – Project 2

Khalid AlAwadhi - Poorn Mehta

S.E.C.U.R.E.R.O.O.M

(**S**uper **E**ncrypted **C**ourse project which is a **U**seful **R**oom **E**mbedded system that communicates **R**emotely **O**ver the air and it's **O**bviously **M**arvelous)

GitHub Link: **https://github.com/K5Ma/APES_Prj2.git**

Documentation Folder Link:
https://drive.google.com/open?id=1lEjhavnkyza8AwUQry0yn8fN5KxVqA8M

Key Learnings

Khalid AlAwadhi

- This Project was one of the most challenging ones throughout my career, and I learned some very good things while working hard on the same – to make it fully functional.
- My part was to create a central structure, to provide a base for all of the inter-node, inter-thread, and inter-task communication, as well as to have a solid support for the Logging. Starting from developing base on BeagleBone – I learned the importance of proper use of **errno** as it can provide some excellent functionalities – related to system, and failures – which is very useful while debugging.
- I learned thoroughly, about implementing Interrupt based reception through UART on BeagleBone. It was a difficult task at an initial point, because I didn't have any idea how the Linux system will go about talking to Hardware with Interrupts.
- We have used Bluetooth Communication to setup a link between Control Node and the Remote Node, and I learned some key concepts like Bluetooth Pairing, Bonding, Connecting etc. while trying to interface the same with my nodes.
- Moreover, since my home has quite a few active Bluetooth Devices, I learned about the importance staying in close proximity – for my communication to work. Many times, the communication was failing because the noise from other devices on same Frequency was quite a lot.
- One other important learning was how to deal with two completely different nodes – running different multitasking algorithms/platforms/OS (Linux vs FreeRTOS) – and to make them talk to each other without any issues. I had to implement some novel approaches to get this done.
- Software Flow Control was something that I never really required in the projects I have worked on before, but this project demanded it. It was because the processing speed, time slicing, etc. were completely different for both boards.
- Developing a customized protocol – as a layer on a standard protocol, is complex – but very effective procedure as well. I did it in order to have reliable communication of not just strings, but standard C structures – between nodes. Formatting them into a Byte Stream, Appending proper testing Bytes with Them, Identifying them, and Decode the same on the other side – it was a challenging task which taught me a lot of things.
- Handling IPC on both Nodes, while also keeping track of sent and received messages – required a lot of time, as well as coding and debugging skills. Proper use of Mutex Locks, and Critical Sections was often found out to be an absolute requirement.

- When I started to Integrate the individual I/O Device Driver source files wrote by my teammate, I found that often – the drivers weren't working as intended, because of some strange issues caused by FreeRTOS's intense time slicing. Functions such as vTaskDelay() often rendered rather random behavior, finding a workaround of which took a lot of creativity.

Poorn Mehta

- For this project, I developed standalone driver source codes for all the I/O devices. It proved to be much more difficult than what I thought – since multitasking of FreeRTOS, the somewhat erratic behavior of some specific I/O devices etc., were quite tough to resolve and it consumed a great amount of time. Apart from technical learnings, I have gained substantial knowledge about project management – from experience, while working on this Project.
- Fault Tolerance was probably one of the key areas which taught me some very good lessons on clean coding, having lesser dependency on shared variables, making the portions of code atomic – while designing others to be multitasking friendly, and other such relevant fields.
- In case of NFC Module, detecting that the Module was offline proved to be difficult, especially when trying to scan an NFC Tag. This took place since the module itself doesn't have anything built-in to check the availability of the Tag, and thus, it just waits for tag to show up. Transmitting another command was overriding the previous reading tag, and thus, both commands had to take turns for execution.
- There were some very specific requirements for a few I/O devices, to drive which – I had to dig much deeper in documentations, to fully understand and implement some functionalities.
- Detecting the End of Response for Electronic Paper was really tricky – because it would take anywhere from a few milli seconds to a few seconds – to even start responding back. Moreover, it doesn't have a start or end frame or byte – which can be used to detect the valid response length. I had to develop a unique algorithm to deal with this highly dynamic response pattern.
- At some instances, I/O Devices wouldn't work properly – because they might not have been connected to a solid ground. This was a quite low-level fault – consuming a good deal of time. Eventually, this led me to connect every Power Connection of the TIVA Board, and of all the I/O Devices on the same node, using breadboard.
- Driving Servo proved to be a challenge, because of undocumented erratic behavior. It works well at various PWM frequencies, but the position to duty cycle correlation changes widely for each of those frequencies. After multiple observations, and trial & error cycles, I was able to figure out the correct values to drive servo in specific positions.
- While writing driver for communicating with BME280 Sensor through SPI bus, I learned that some sensors require 'pulse' on the Slave Select pin, before they begin to communicate. This was vaguely mentioned in the datasheet with loose terms, and was counterintuitive – which resultantly took me many hours to resolve the error, and finally get the interface done.
- At some point, one part of my code had partial recursive function calls, which cause the Task to run out of stack at some specific points. I gained knowledge about some key concepts on stack management while resolving this issue.
- Overall, I learned many things in this Project, however I must say that it was a bit – over the head, owing to the 'size' of the project itself.

Project Plan

1	**Hands Free Startup**	Remote Node Start Up	Remote Node starts to create every task, and also performs BIST (whenever it applies), while printing step by step information to the the UART0 - and also, sending the same for logging purposes to the Control Node	BIST passes/fails - as expected (from physical state of the connections), Tasks are created, and starts to run concurrently.
2		Control Node Start Up	Control Node starts to create every thread, creates queues, and is all set to receive data from Remote Node	Control Node logs the information about starting, successfully sets up the resources required for functional operation, waits for Remote Node
3		Remote Node Arrival is detected by Control Node	Control Node is waiting for message from the Remote Node, when it's present, it logs the same	When Remote Node is functional, and within the range of BlueTooth Communication, Control Node Logs that it detected Remote Node
4		Remote Node Disappearance is detected by Control Node	Control Node has a timeout - within which if no message from the remote node is received, the situation is interpreted of Remote Node being Offline	When Remote Node is not functional and/or it out of range of BlueTooth Communication, Control Node Logs that the Remote Node has been disappereed

1	**Fault Tolerance (Built in Self Test Part)**	NFC BIST at Startup Success	NFC Task tries to talk to the NFC Module at the startup. If successful, it prints out message to the TIVA's UART0, and proceeds further	Prints out NFC Success Message
2		NFC BIST at Startup Failure	NFC Task tries to talk to the NFC Module at the startup. If failed, it prints out message to the TIVA's UART0, and cancels the Task	Prints out NFC Error Message, Task doesn't do anything after this point
3		EPaper BIST at Startup Success	EPaper Task tries to talk to the EPaper Module at the startup. If successful, it prints out message to the TIVA's UART0, and proceeds further	Prints out EPaper Success Message
4		EPaper BIST at Startup Failure	EPaper Task tries to talk to the EPaper Module at the startup. If failed, it prints out message to the TIVA's UART0, and starts retrying	Prints out EPaper Error Message, Task keeps on retrying
5		LoadCell BIST at Startup Success	LoadCell Task tries to talk to the LoadCell Module at the startup. If successful, it prints out message to the TIVA's UART0, and proceeds further	Prints out LoadCell Success Message
6		LoadCell BIST at Startup Failure	LoadCell Task tries to talk to the LoadCell Module at the startup. If failed, it prints out message to the TIVA's UART0, and starts retrying	Prints out LoadCell Error Message, Task keeps on retrying
7		BME280 BIST at Startup Success	BME280 Task tries to talk to the LoadCell Module at the startup. If successful, it prints out message to the TIVA's UART0, and proceeds further	Prints out BME280 Success Message
8		BME280 BIST at Startup Failure	BME280 Task tries to talk to the BME280 Module at the startup. If failed, it prints out message to the TIVA's UART0, and starts retrying	Prints out BME280 Error Message, Task keeps on retrying
9		Lux BIST at Startup Success	Lux Task tries to talk to the LoadCell Module at the startup. If successful, it prints out message to the TIVA's UART0, and proceeds further	Prints out Lux Success Message
10		Lux BIST at Startup Failure	Lux Task tries to talk to the Lux Module at the startup. If failed, it prints out message to the TIVA's UART0, and starts retrying	Prints out Lux Error Message, Task keeps on retrying
11		Gas BIST at Startup Success	Gas Task tries to talk to the LoadCell Module at the startup. If successful, it prints out message to the TIVA's UART0, and proceeds further	Prints out Gas Success Message
12		Gas BIST at Startup Failure	Gas Task tries to talk to the Gas Module at the startup. If failed, it prints out message to the TIVA's UART0, and starts retrying	Prints out Gas Error Message, Task keeps on retrying

1	**Fault Tolerance (Runtime Error Occurrence Part)**	EPaper Runtime Error	EPaper Task keeps on checking the Module state on regular basis, and if the Module is detached - it will print out message to the TIVA's UART0, and starts	Prints out EPaper Error Message, Task keeps on retrying
2		LoadCell Runtime Error	LoadCell Task keeps on checking the Module state on regular basis, and if the Module is detached - it will print out message to the TIVA's UART0, and starts	Prints out LoadCell Error Message, Task keeps on retrying
3		BME280 Runtime Error	BME280 Task keeps on checking the Module state on regular basis, and if the Module is detached - it will print out message to the TIVA's UART0, and starts	Prints out BME280 Error Message, Task keeps on retrying
4		Lux Runtime Error	Lux Task keeps on checking the Module state on regular basis, and if the Module is detached - it will print out message to the TIVA's UART0, and starts	Prints out Lux Error Message, Task keeps on retrying
5		Gas Runtime Error	Gas Task keeps on checking the Module state on regular basis, and if the Module is detached - it will print out message to the TIVA's UART0, and starts	Prints out Gas Error Message, Task keeps on retrying
6		Remote Node Suddenly Goes Offline	Control Node is expecting an 'alive check' message at every fixed interval. If that is not received, it will detect that the communication link is Broken. It will log the same message.	If Remote Node is Powered Off (or made unavaible to the Control Node in Some other way) - the Control Node detects it within reasonably short period of time, and then Logs the message to reflect the same
7		Control Node Suddenly Goes Offline	Remote Node is expecting an 'alive check' message at every fixed interval. If that is not received, it will detect that the communication link is Broken. It will print out the same message on TIVA's UART0.	If Control Node is Powered Off (or made unavaible to the Remote Node in Some other way) - the Remote Node detects it within reasonably short period of time, and then prints the message to reflect the same
8		When Remote Node is Offline, Control Node keeps Scanning for it	While the Remote Node is offline, the Control Node keeps on searching for any message sent from the Remote Node. It will keep on logging this state of operation as well	The Control Node keeps logging that it is still missing any valid message reception from the Remote Node, while Remote Node is offline
9		When Control Node is Offline, Remote Node keeps Scanning for it	While the Control Node is offline, the Remote Node keeps on searching for any message sent from the Control Node. It will keep on printing this state of operation on TIVA's UART0 as well	The Remote Node keeps printing message that it is still missing any valid message reception from the Control Node, while Control Node is offline

1	**Fault Tolerance (Runtime Error Resolution Part)**	EPaper Runtime State Test Succeeds	This Test only applies when EPaper Module was detected to be Offline (either at the startup, or at runtime) - and then it comes back Online.	Prints out EPaper is Online message, Task returns to normal state
2		LoadCell Runtime State Test Succeeds	This Test only applies when LoadCell Module was detected to be Offline (either at the startup, or at runtime) - and then it comes back Online.	Prints out LoadCell is Online message, Task returns to normal state
3		BME280 Runtime State Test Succeeds	This Test only applies when BME280 Module was detected to be Offline (either at the startup, or at runtime) - and then it comes back Online.	Prints out BME280 is Online message, Task returns to normal state
4		Lux Runtime State Test Succeeds	This Test only applies when Lux Module was detected to be Offline (either at the startup, or at runtime) - and then it comes back Online.	Prints out Lux is Online message, Task returns to normal state
5		Gas Runtime State Test Succeeds	This Test only applies when Gas Module was detected to be Offline (either at the startup, or at runtime) - and then it comes back Online.	Prints out Gas is Online message, Task returns to normal state
6		Remote Node Comes back Online	This Test only applies when Remote Node was detected to be Offline by Control Node, and it was driven in a state to trying to find it continuously	Control Node Logs the message that Remote Node is back Online, Returns to Normal Operation
7		Control Node Comes back Online	This Test only applies when Control Node was detected to be Offline by Remote Node, and it was driven in a state to trying to find it continuously	Remote Node prints the message that Control Node is back Online, Returns to Normal Operation

1	**Local Safety Control**	Temperature Too High Test	When Temperature of the Atmosphere surrouding the Valuable Contents - is found to be Higher than the preset Threshold, then appropriate actions should be taken by Remote Node in order to facilitate Rescual of these Valuables	Prints Temperature Warning message, Servo Rotates to Open the Door, Buzzer Starts up to provide Auditory Alert
2		Temperature Too Low Test	When Temperature of the Atmosphere surrouding the Valuable Contents - is found to be Lower than the preset Threshold, then appropriate actions should be taken by Remote Node in order to facilitate Rescual of these Valuables	Prints Temperature Warning message, Servo Rotates to Open the Door, Buzzer Starts up to provide Auditory Alert
3		Humidity Too High Test	When Humidity of the Atmosphere surrouding the Valuable Contents - is found to be Higher than the preset Threshold, then appropriate actions should be taken by Remote Node in order to facilitate Rescual of these Valuables	Prints Humidity Warning message, Servo Rotates to Open the Door, Buzzer Starts up to provide Auditory Alert
4		Lux Too High Test	When the Authorization is Not Completed, the Room is supposed to be in Pitch Black envrionment. In case of a Fire (even just a spark), or Breach of the Room by means of Breaking of Walls - or any such thing, the Internal Light Level will suddenly Rise, raising an Alert	Prints Lux Warning message, Buzzer Starts up to provide Auditory Alert
5		Lux is Disabled Test	The Lux Sensor should be Disable when there is a valid Authorization - because the opening of the Door, Switching on of the internal Lights, etc. is expected and should not result in a False Alarm	Even if the Lux Level is raised, it Does Not result in a Warning and/or Buzzer Sound
6		Gas Too High Test	The Gas Module is capable of detecting multiple types of Harmful Gases. This represents any set of Gases that could be potentially Harmful for the Valuables placed inside the Room/Lethal for the Authorized individual planning to Enter the Room. Thus if concentration of any of these Gas(es) is raised above predetermined safe value - appropriate actions should be taken to prevent a disaster	Prints Gas Warning message, Servo Rotates to Open the Door, Buzzer Starts up to provide Auditory Alert
7		PIR Breach Test	In the event where the Locked Door is broke Open, the PIR Module will detect motion, and Raise the Alert	Prints PIR Warning message, Buzzer Starts up to provide Auditoy Alert
8		PIR is Disabled Test	The PIR Module should go Offline, when the Authorized Individual is gaining Entry in the Room, because at this time - the opening of the Door, and resultant motion detection is Expected.	Even if the Object is Presented in front of the PIR Module, it Does Not result in a Warning and/or a Buzzer Sound
9		Local Safety is Not Breached Test	If all of the Local Safety Sensors are not Detecting anything Unusual, then the Buzzer should stay Off (Turn it Off if it was previously On), and the Door should stay Closed (Close it if it was previously Open)	The Buzzer is Off (Turned Off following Local Safety Control Breach), as well the Servo is on Close Position (Turned to move in this state following Local Safety Control Breach)

Note: Following Test Conditions are applicable only when there is no Ongoing RunTime Error with respect I/O Device. In Case of the Failure, the Test will Fail as well (behavior is similar to that of described in the Fault Tolerance Section)

1	**Normal Verification Operation**	NFC Tag ID Reading Test	When a NFC Tag is Presented to the Module, it should read the unique ID, and print the same on the TIVA's UART0.	When NFC Tag is Presented, it Prints out message containing the ID of the same
2		NFC Tag is Received by T_BBComm Test	When a NFC Tag is Read by the NFC Task, it should then be sent to the T_BBComm Task through IPC (Queue) so that it can be sent to the Control Node.	T_BBComm Confirms the reception of the Tag ID by printing message
3		NFC Tag is Received on Control Node Test	Control Node receives structure related to the NFC Tag ID from Remote Node - in BB_TIVAComm Thread. It logs the information received in the file.	BB_TIVAComm Logs message, indicating successful Reception of the Tag ID
4		NFC Tag Received by NFC Thread Test	After the NFC Tag ID is received on the BB_TIVAComm Thread, it should then be sent to the NFC Thread for further processing through IPC (Queue)	NFC Thread Logs message, indicating successful Reception of the Tag ID
5		NFC Tag ID Verified Test	If the presented NFC Tag at the Module on Remote Node is Valid (ID matching to one of the preprogrammed ones), then it should proceed accordingly - with the rest of the verification stages	NFC Thread Logs message, along with the Name of the Person to whom - the presented Tag Belongs
6		NFC Tag ID Not Verified Test	If the presented NFC Tag at the Module on Remote Node is Not Valid (ID not matching to any of the preprogrammed ones), then it should just Log this and not proceed further with the rest of the verification stages	NFC Thread Logs Error message, indicating that the Received Tag ID was found out to be Invalid/Missing from Internal Database
7		Verification Stage 2 is Initialized Test	On the Control Node, following the Successful verification of the Tag ID, the NFC Function (residing inside NFC Thread) should set the appropriate pair of Image (to be Displayed on EPaper) and 6 letter code (to be entered via Keypad), and send the structure to the BB_TIVAComm Task using IPC (Queue) for relaying it to the Remote Node	BB_TIVAComm Logs message, indicating successful Reception of the Structure Required to Start Verification Stage 2
8		Verification Stage 2 Structure is Received on Remote Node Test	The Structure (required for starting Verification Stage 2) should be Received on the Remote Node - after the Control Node is done Sending the Same, the confirmation of which should be Sent to the TIVA's UART0	T_BBComm Task prints out a message, indicating successful Reception of the Structure that belongs to the Epaper_Keypad Task
9		Verification Stage 2 Initialization Structure Received by EPaper_Keypad Task Test	On the Remote Node, if the Structure (required for kicking off Verification Stage 2) is received on T_BBComm, then it Should Pass it to the EPaper_Keypad Task for further processing using IPC (Queue)	EPaper_Keypad Task prints out message indicating that it Received a valid Strucutre from T_BBComm
10		EPaper Update Image Test	Once the EPaper Update is Enabled, the EPaper Module reads the Image Name string, and Displays the Same from SD Card. This image is to be used as a base to choose a CodeWord from the predetermined ones, for the User	Epaper_Keypad Task prints out the message Containing the Requested Image Name & the Response of EPaper Module containing File Name, both of which would be the same
11		Keypad Code Entered Test	Once the KeyPad Polling is Enabled, the Keypad Starts to Poll to Enable the User to input a 6 letter long Keyword - to be matched against predetermined values, stored Internally on the Control Node	EPaper_Keypad Task prints out the message Containing Entire KeyWord Entered
12		Keypad Code Error Test	When Keypad Polling is On, and if Keypad Times out before all 6 characters are entered, it will be printed to TIVA's UART0 - and the strcuture will not be sent to the Control Node	Prints out Keypad Error Message, Task returns back to Normal Functionality
13		Keypad Code Sent to T_BBComm Task Test	Once the CodeWord is Entered, it should then be sent to the T_BBComm Task in order to get it Processed on Control Node	T_BBComm Confirms the reception of the CodeWord by printing message
14		Keypad Code Received on Control Node Test	Control Node receives structure related to the Keypad Code from Remote Node - in BB_TIVAComm Thread. It logs the information received in the file.	BB_TIVAComm Logs message, indicating successful Reception of the CodeWord
15		CodeWord Received by EPaper_Keypad Thread Test	After the CodeWord is received on the BB_TIVAComm Thread, it should then be sent to the KE Thread for further processing through IPC (Queue)	KE Thread Logs message, indicating successful Reception of the CodeWord
16		CodeWord Verified Test	Once the KE thread has the CodeWord, it checks whether it is what it's supposed to be (related to the image that the NFC Thread set), and if it's correct, then it proceeds further with Verification Stage 3	KE Thread Logs message, indicating successful verification of the Received CodeWord
17		CodeWord Not Verified Test	Once the KE thread has the CodeWord, it checks whether it is what it's supposed to be (related to the image that the NFC Thread set), and if it's incorrect correct, then it should just log this and not proceed with Verification Stage 3	KE Thread Logs Error message, indicating that the Received CodeWord did not match the preset (related to the Image being Displayed on Epaper) one
18		Verification Stage 3 is Initialized Test	On the Control Node, following the Successful verification of the CodeWord, the KE Function (residing inside KE Thread) should set the appropriate structure to initialize the LoadCell polling, and send it to the BB_TIVAComm Task using IPC (Queue) for relaying it to the Remote Node	BB_TIVAComm Logs message, indicating successful Reception of the Structure Required to Start Verification Stage 3
19		Verification Stage 3 Structure is Received on Remote Node Test	The Structure (required for starting Verification Stage 3) should be Received on the Remote Node - after the Control Node is done Sending the Same, the confirmation of which should be Sent to the TIVA's UART0	T_BBComm Task prints out a message, indicating successful Reception of the Structure that belongs to the LoadCell Task
20		Verification Stage 3 Initialization Structure Received by LoadCell Task Test	On the Remote Node, if the Structure (required for kicking off Verification Stage 3) is received on T_BBComm, then it Should Pass it to the LoadCell Task for further processing using IPC (Queue)	LoadCell Task prints out message indicating that it Received a valid Strucutre from T_BBComm
21		LoadCell Sampling Test	Once the LoadCell Polling is Enabled, the LoadCell Module Starts to Collect Samples with preset Frequency and Filters, to Enable the User to input a unique pattern of grip - to be matched against predetermined values, stored Internally on the Control	LoadCell Task prints out the message Containing Entire Array which gather Samples
22		All LoadCell Samples Sent to T_BBComm Task Test	Once the LoadCell Task has finished Gathering Samples, the whole Array should then be sent to the T_BBComm Task in order to get it Processed on Control Node	T_BBComm Confirms the reception of the LoadCell Samples Array by printing message
23		Al LoadCell Samples Received on Control Node Test	Control Node receives structure related to the LoadCell Samples from Remote Node - in BB_TIVAComm Thread. It logs the information received in the file.	BB_TIVAComm Logs message, indicating successful Reception of the LoadCell Samples Array
24		LoadCell Samples Received by LoadCell Thread Test	After the LoadCell Samples Array is received on the BB_TIVAComm Thread, it should then be sent to the LC Thread for further processing through IPC (Queue)	LC Thread Logs message, indicating successful Reception of the LoadCell Samples Array
25		LoadCell Samples Verified Test	Once the LC thread has all the LoadCell Samples, it checks all of them in order to make a decision about the 'right' grip - which depends on the predetermined pattern (and resultant Algorithm), and if the Algortihm Verifies it Successfully, then it proceeds further with Controlling Output Device properly - to reflect the Success	LC Thread Logs message, indicating successful verification of the Received LoadCell Samples, and that the Person has been Granted Access
26		LoadCell Samples Not Verified Test	Once the LC thread has all the LoadCell Samples, it checks all of them in order to make a decision about the 'right' grip - which depends on the predetermined pattern (and resultant Algorithm), and if the Algortihm fails to Verify the same, then it should just Log the relevant message, and not proceed with driving Output Devices	LC Thread Logs Error message, indicating that the Received LoadCell Samples did not match with Predetermined Pattern
27		Granting Access Stage is Initialized Test	On the Control Node, following the Successful verification of the LoadCell Samples - and resultantly successful completion of All the Verification Stages, the LC Function (residing inside LC Thread) should set the appropriate structure to Contorl Output Devices Properly to reflect the Granted Access, and send it to the BB_TIVAComm Task using IPC (Queue) for relaying it to the Remote Node	BB_TIVAComm Logs message, indicating successful Reception of the Structure Required to Grant Access on Remote Node
28		Granting Access Stage Structure is Received on Remote Node Test	The Structure (required for Driving Output Devices to reflect Granted Access) should be Received on the Remote Node - after the Control Node is done Sending the Same, the confirmation of which should be Sent to the TIVA's UART0	T_BBComm Task prints out a message, indicating successful Reception of the Structure that belongs to the OutputIndicators Task
29		Grant Access Stage Initialization Structure Received by OutputIndicators Task Test	On the Remote Node, if the Structure (required for kicking off Grant Access) is received on T_BBComm, then it Should Pass it to the Output Indicators Task for further processing using IPC (Queue)	OutputIndicators Task prints out message indicating that it Received a valid Strucutre from T_BBComm
30		Grant Access Stage Completed Test	Once the Data required to Drive the Output Devices has been Received, The OutputIndicators Task Decodes the data, and Drives the Devices Connected to it Accordingly	Servo Turns to Open the Door - and automatically Closes after a Timeout, SpeakJet Plays a Welcome Message Containing Authorized Person's Name

1	**Self Requirements**	Verification Stage 1 Passes, Starts Verification Stage 2	The Verification Stage 2 is Initiated Following the Successful Completion of Stage 1	Valid NFC Tag Updates EPaper, Starts polling Keypad
2		Verification Stage 2 Passes, Starts Verification Stage 3	The Verification Stage 3 is Initiated Following the Successful Completion of Stage 2	Correct CodeWork Starts LoadCell Polling
3		Verification Stage 3 Passes, Opens the Door	The Door is Opened Following the Successful Completion of Stage 3	Correct pattern Moves the Servo to Open Condition

1	**Logging**	Remote Node Arrival/Disappearance Log Test	The Arrival/Disappearance of Remote Node should be Logged	Control Node Logs the message, indicating the Arrival/Disappearance of Remote Node
2		Startup Test (BIST) Success/Failure Log Test	All of I/O Devices' Startup Test Results should be Logged	Control Node Logs the message, along with Name of the I/O Device - and result of it's own Startup Test (Performed on Remote Node)
3		Runtime Error of I/O Device Log Test	If there is any change in the Runtime State of any I/O Device on Remote Node, (Either Going Offline from Online, or Coming Back Online from Offline State) - it should be logged on the Control Node	Control Node Logs the message, along with Name of the I/O Device - and related Change in Operation State
4		Local Safety Control Log Test	The Control Node should Log all the Significant Events that are Taking Place on the Remote Node with respect to Local Safety Control	Control Node Logs the message, reflecting Current Important State of the relevant Local Safety Control I/O Devices
5		Normal Operation Log Test	When Normal Operation is Going, Each Significant Event Should be Logged	Control Node Logs the message, reflecting Current Important State of the relevant Normal Operation I/O Devices
6		Readable Logger File Test	The Logger File Format should be simple Text Friendly	Able to Open the Logger File using User Interface, and Able to Read the Contents of the Same
7		Logger Surviving Power Cycles Test	The Logger File Should Not be Erased/Overwritten after going through a Power Cycle	The Logger File is Appending to the Previsouly Created Logger
8		All Logs have Time Stamp Test	Every Log message should be stamped with specific Time	The Events match with the Time being Displayed on the Control Node

Test Results

1	**Hands Free Startup**	Remote Node Start Up	Pass	
2		Control Node Start Up	Pass	
3		Remote Node Arrival is detected by Control Node	Skipped	Not Enough Time to Test
4		Remote Node Disappearance is detected by Control Node	Skipped	Not Enough Time to Test
1	**Fault Tolerance (Built in Self Test Part)**	NFC BIST at Startup Success	Pass	
2		NFC BIST at Startup Failure	Pass	
3		EPaper BIST at Startup Success	Pass	
4		EPaper BIST at Startup Failure	Pass	
5		LoadCell BIST at Startup Success	Pass	
6		LoadCell BIST at Startup Failure	Pass	
7		BME280 BIST at Startup Success	Skipped	Not Enough Time to Test
8		BME280 BIST at Startup Failure	Skipped	Not Enough Time to Test
9		Lux BIST at Startup Success	Skipped	Not Enough Time to Test
10		Lux BIST at Startup Failure	Skipped	Not Enough Time to Test
11		Gas BIST at Startup Success	Skipped	Not Enough Time to Test
12		Gas BIST at Startup Failure	Skipped	Not Enough Time to Test
1	**Fault Tolerance (Runtime Error Occurrence Part)**	EPaper Runtime Error	Pass	
2		LoadCell Runtime Error	Pass	
3		BME280 Runtime Error	Skipped	Not Enough Time to Test
4		Lux Runtime Error	Skipped	Not Enough Time to Test
5		Gas Runtime Error	Skipped	Not Enough Time to Test
6		Remote Node Suddenly Goes Offline	Skipped	Not Enough Time to Test
7		Control Node Suddenly Goes Offline	Skipped	Not Enough Time to Test
8		When Remote Node is Offline, Control Node keeps Scanning for it	Skipped	Not Enough Time to Test
9		When Control Node is Offline, Remote Node keeps Scanning for it	Skipped	Not Enough Time to Test
1	**Fault Tolerance (Runtime Error Resolution Part)**	EPaper Runtime State Test Succeeds	Pass	
2		LoadCell Runtime State Test Succeeds	Pass	
3		BME280 Runtime State Test Succeeds	Skipped	Not Enough Time to Test
4		Lux Runtime State Test Succeeds	Skipped	Not Enough Time to Test
5		Gas Runtime State Test Succeeds	Skipped	Not Enough Time to Test
6		Remote Node Comes back Online	Skipped	Not Enough Time to Test
7		Control Node Comes back Online	Skipped	Not Enough Time to Test
1	**Local Safety Control**	Temperature Too High Test	Skipped	Not Enough Time to Test
2		Temperature Too Low Test	Skipped	Not Enough Time to Test
3		Humidity Too High Test	Skipped	Not Enough Time to Test
4		Lux Too High Test	Skipped	Not Enough Time to Test
5		Lux is Disabled Test	Skipped	Not Enough Time to Test
6		Gas Too High Test	Skipped	Not Enough Time to Test
7		PIR Breach Test	Skipped	Not Enough Time to Test
8		PIR is Disabled Test	Skipped	Not Enough Time to Test
9		Local Safety is Not Breached Test	Skipped	Not Enough Time to Test

1	**Normal Verification Operation**	NFC Tag ID Reading Test	Pass
2		NFC Tag is Received by T_BBComm Test	Pass
3		NFC Tag is Received on Control Node Test	Pass
4		NFC Tag Received by NFC Thread Test	Pass
5		NFC Tag ID Verified Test	Pass
6		NFC Tag ID Not Verified Test	Pass
7		Verification Stage 2 is Initialized Test	Pass
8		Verification Stage 2 Structure is Received on Remote Node Test	Pass
9		Verification Stage 2 Initialization Structure Received by EPaper_Keypad Task Test	Pass
10		EPaper Update Image Test	Pass
11		Keypad Code Entered Test	Pass
12		Keypad Code Error Test	Pass
13		Keypad Code Sent to T_BBComm Task Test	Pass
14		Keypad Code Received on Control Node Test	Pass
15		CodeWord Received by EPaper_Keypad Thread Test	Pass
16		CodeWord Verified Test	Pass
17		CodeWord Not Verified Test	Pass
18		Verification Stage 3 is Initialized Test	Pass
19		Verification Stage 3 Structure is Received on Remote Node Test	Pass
20		Verification Stage 3 Initialization Structure Received by LoadCell Task Test	Pass
21		LoadCell Sampling Test	Pass
22		All LoadCell Samples Sent to T_BBComm Task Test	Pass
23		Al LoadCell Samples Received on Control Node Test	Pass
24		LoadCell Samples Received by LoadCell Thread Test	Pass
25		LoadCell Samples Verified Test	Pass
26		LoadCell Samples Not Verified Test	Pass
27		Granting Access Stage is Initialized Test	Pass
28		Granting Access Stage Structure is Received on Remote Node Test	Pass
29		Grant Access Stage Initialization Structure Received by OutputIndicators Task Test	Pass
30		Grant Access Stage Completed Test	Pass

1	**Self Requirements**	Verification Stage 1 Passes, Starts Verification Stage 2	Pass	
2		Verification Stage 2 Passes, Starts Verification Stage 3	Pass	
3		Verification Stage 3 Passes, Opens the Door	Pass	
1	**Logging**	Remote Node Arrival/Disappearance Log Test	Skipped	Not Enough Time to Test
2		Startup Test (BIST) Success/Failure Log Test	Pass	
3		Runtime Error of I/O Device Log Test	Pass	
4		Local Safety Control Log Test	Skipped	Not Enough Time to Test
5		Normal Operation Log Test	Pass	
6		Readable Logger File Test	Pass	
7		Logger Surviving Power Cycles Test	Pass	
8		All Logs have Time Stamp Test	Pass	