

a07g-exploring-the-CLI

- Team Number: T06
- Team Name: Byte Crafter
- Team Members: Tony Yan & Yue Zhang
- GitHub Repository URL: <https://github.com/ese5160/final-project-a07g-a14g-t06-byte-crafter>
- Description of test hardware: ROG Zephyrus G14, HUAWEI 14

1. Software Architecture

1.1 Revisit Hardware and Software Requirements Specification (HRS & SRS) in A00G

1. Hardware Requirements Specification (HRS)

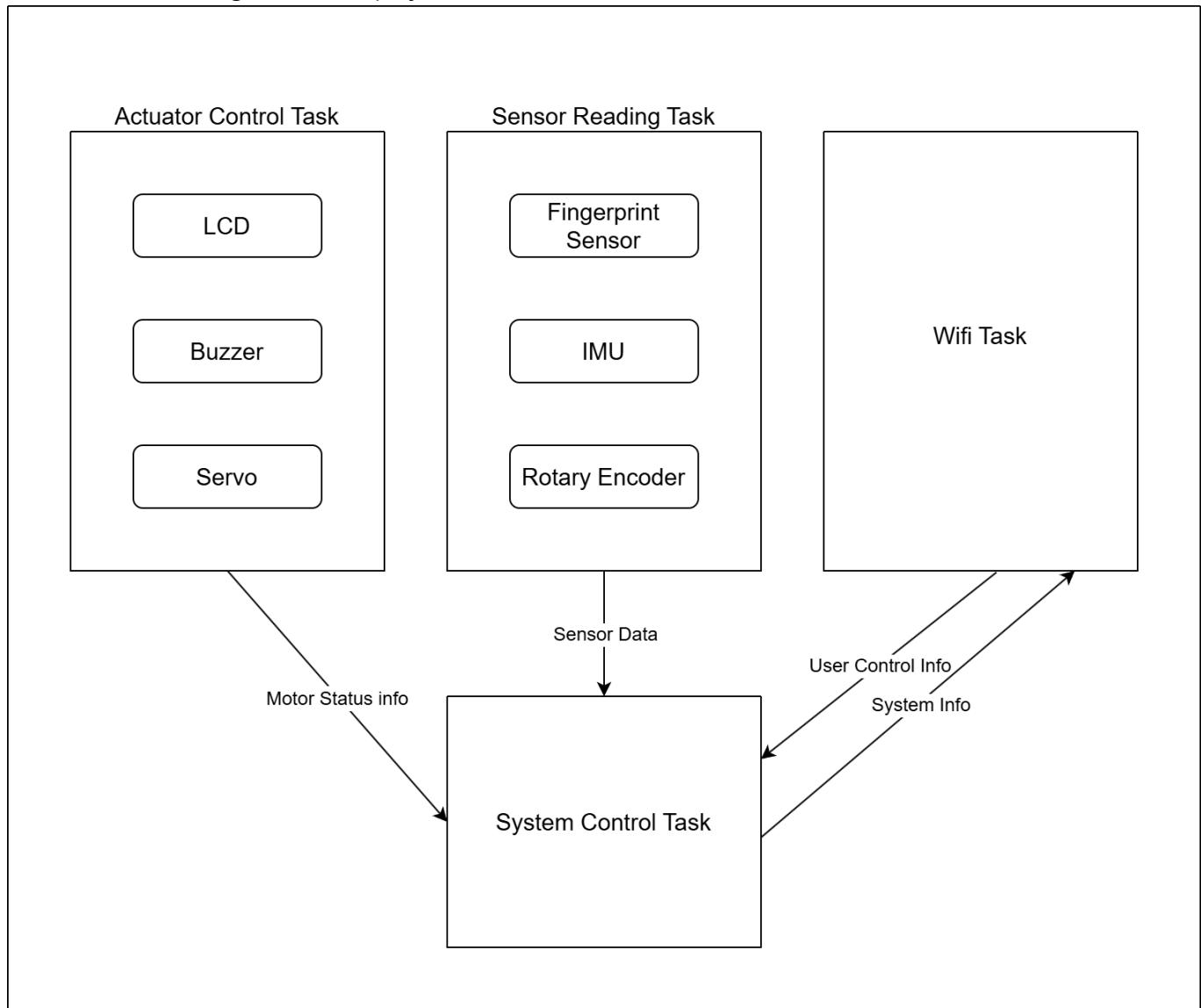
1. **HRS 01:** SAMW25 microcontroller shall be used for this project.
2. **HRS 02:** LCD display shall be used for user interface with correct information and the refresh rate of the screen should not be too slow (e.g. 3s), via SPI protocol.
3. **HRS 03:** The rotary encoder shall allow the user to navigate the menu on the LCD and select it, through GPIO pins configured for interrupt-driven input.
4. **HRS 04:** The fingerprint module shall achieve greater than 95% correctness within 1 second, via a UART interface.
5. **HRS 05:** The IMU shall detect vibrations caused by violent tampering or forced entry attempts, via an I2C interface.
6. **HRS 06:** The buzzer shall be sounded when specific requirements are met, driven via a PWM-capable GPIO pin.

2. Software Requirements Specification (SRS)

1. **SRS 01:** The LCD shall have a menu and allow users to add or delete fingerprints, after the master fingerprint has been successfully detected.
2. **SRS 02:** The fingerprint module shall open the lock after the authorized fingerprint be detected.
3. **SRS 03:** The fingerprint module shall trigger an alarm(buzzer) when it didn't detect authorized fingerprint three times or accelerometer detect someone wants to break the lock.
4. **SRS 04:** The system shall send a warning message to cloud when duress fingerprint be detected.
5. **SRS 05:** The system shall allow user to view real-time lock status and fingerprint library via cloud.
6. **SRS 06:** The system shall allow user to open or close the lock via the cloud.
7. **SRS 07:** The system shall allow user to add or delete fingerprint via the cloud.
8. **SRS 07:** The system shall lock itself after 5s after open.

1.2 Block diagram outlining the different tasks

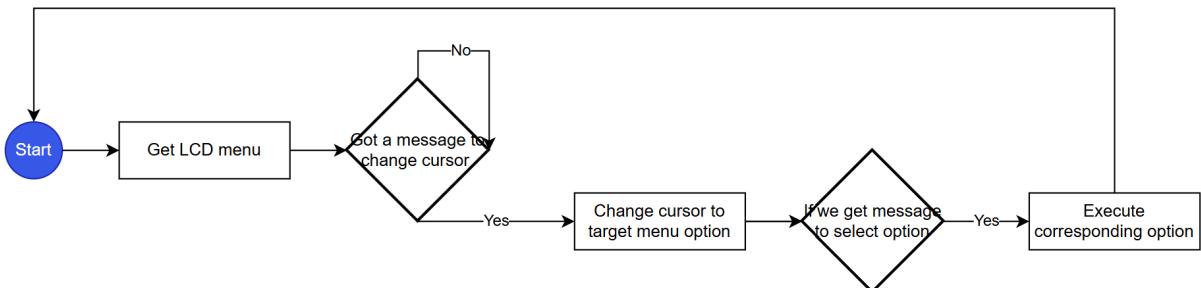
Please see below diagram for our project:



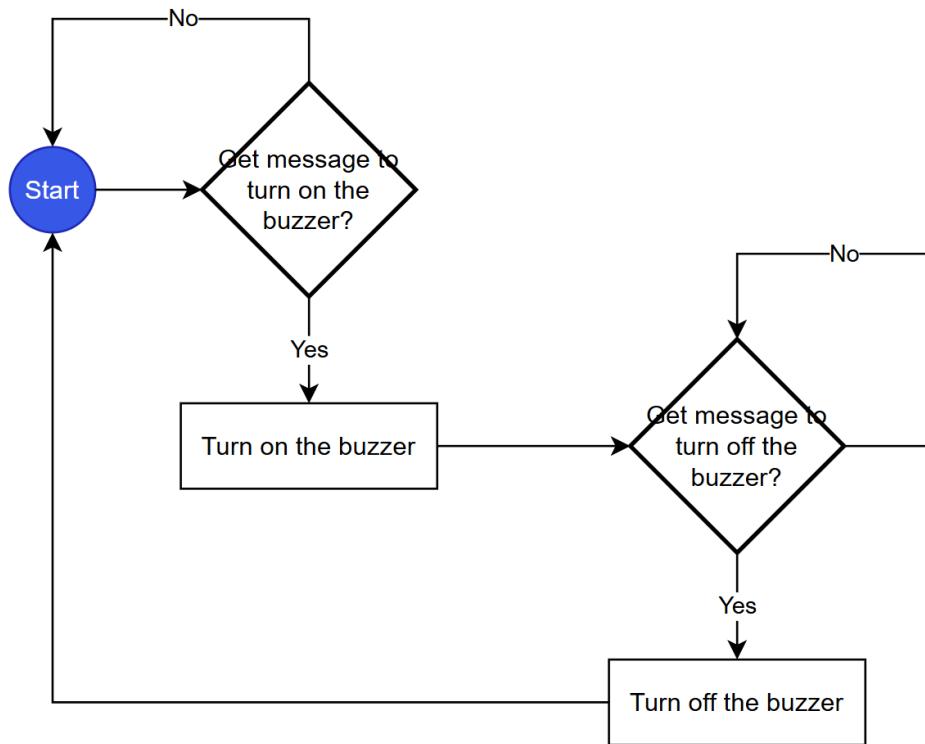
1.3 Flowcharts or state machine diagram

Please see below flowcharts for our tasks:

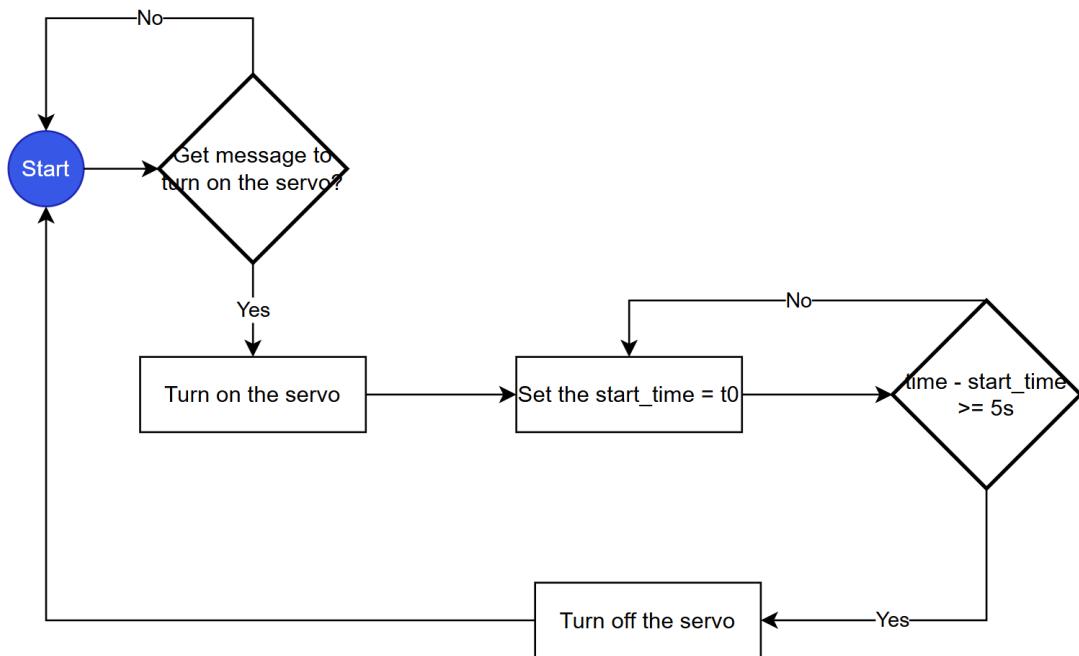
1. LCD flowchart:



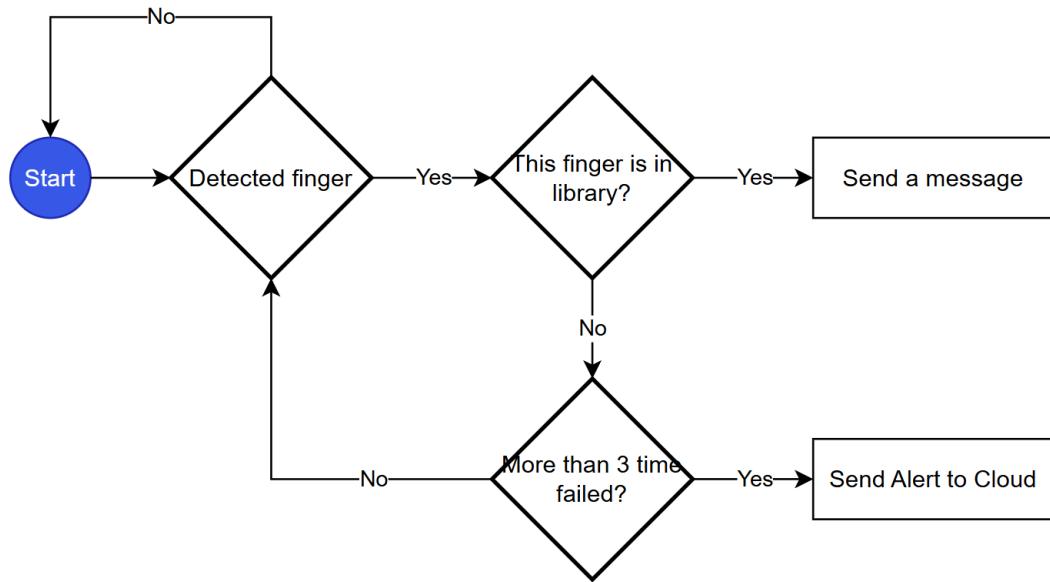
2. Buzzer flowchart:



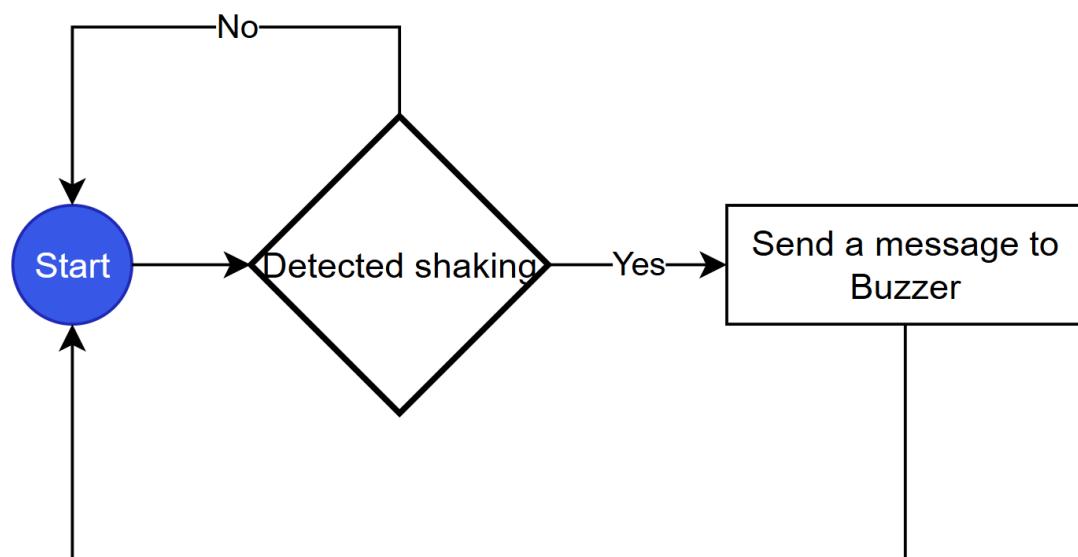
3. Servo flowchart:



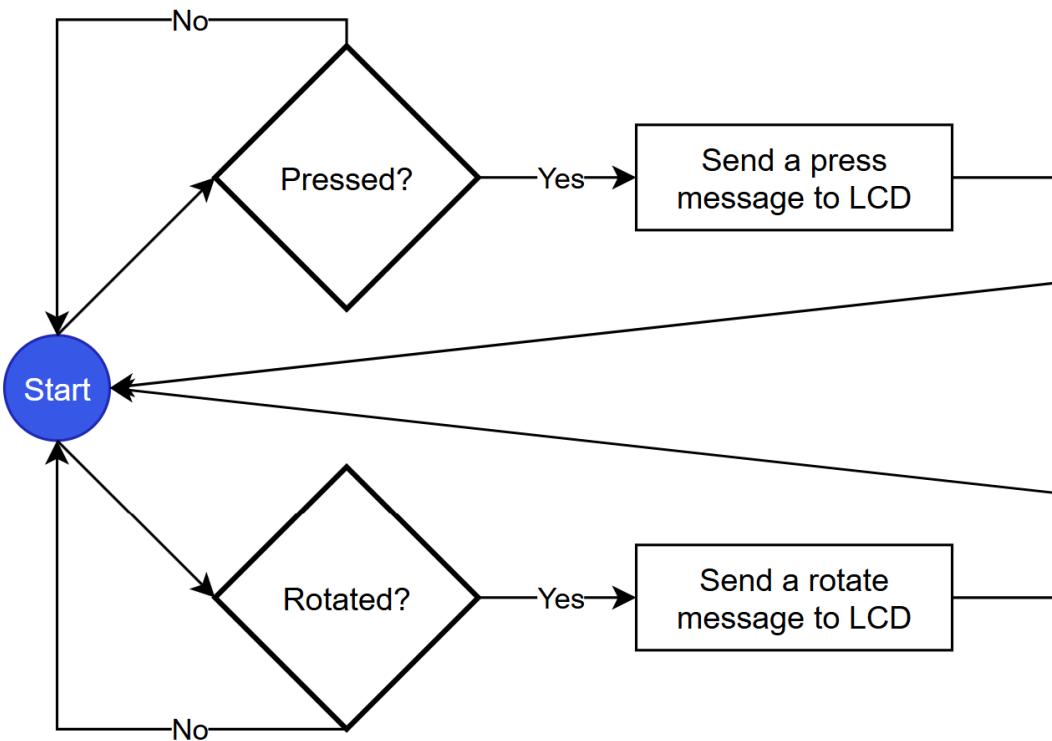
4. Fingerprint flowchart:



5. IMU flowchart:



6. Rotary Encoder flowchart:



2. Understanding the Starter Code

2.1

1. `InitializeSerialConsole()` function is used to initialize the UART and register the callback function, the main purpose is to prepare the serial port for communication: Initialize Ring Buffers (Circular Buffers);Configure USART (Universal Synchronous/Asynchronous Transceiver); Setting Interrupt Priority; Initiate Serial Data Read.
2. `cbufRx` is a Circular Buffer that points to the `rxCharacterBuffer`, which is used to store data received from the serial port. `cbufTx` is a Circular Buffer that points to the `txCharacterBuffer`, which holds the data to be sent.
3. Data structure: FIFO

2.2

`cbufRx` and `cbufTx` are initialized by `circular_buf_init()`, which allocates a `circular_buf_t` structure and they contain pointers to `rxCharacterBuffer` and `txCharacterBuffer`, respectively. These arrays serve as the data storage for the circular buffer, where incoming and outgoing serial port data is stored. The function also sets the maximum storage size of the circular buffer based on the size of these arrays. Additionally, the buffer must be cleared (using `circular_buf_reset()`) to ensure proper initialization.

Library: `SerialConsole.c`

2.3

RX: Array: `rxCharacterBuffer[]`; Size: `RX_BUFFER_SIZE` 512

TX: Array: `txCharacterBuffer[]`; Size: `TX_BUFFER_SIZE` 512

2.4

configure_usart_callbacks() _*usart_interrupt_handler* in *usart_interrupt.c*

2.5

a. TX: usart_write_callback

b. RX: usart_read_callback

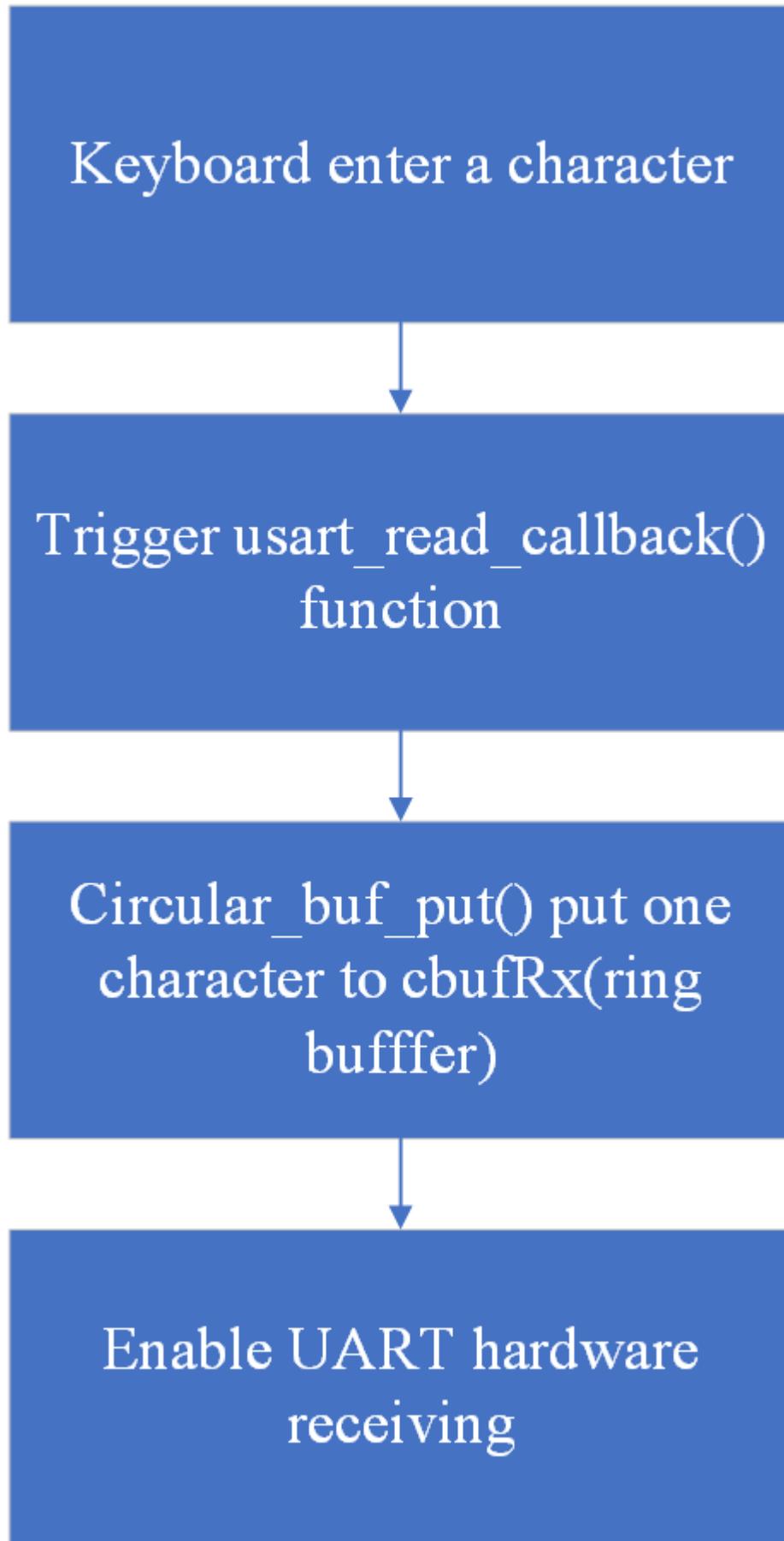
2.6

usart_write_callback: Get the next character to be sent from cbufTx and continue the send operation.

usart_read_callback: Put the received data into cbufRx for subsequent processing.

2.7

UART receive:



2.8

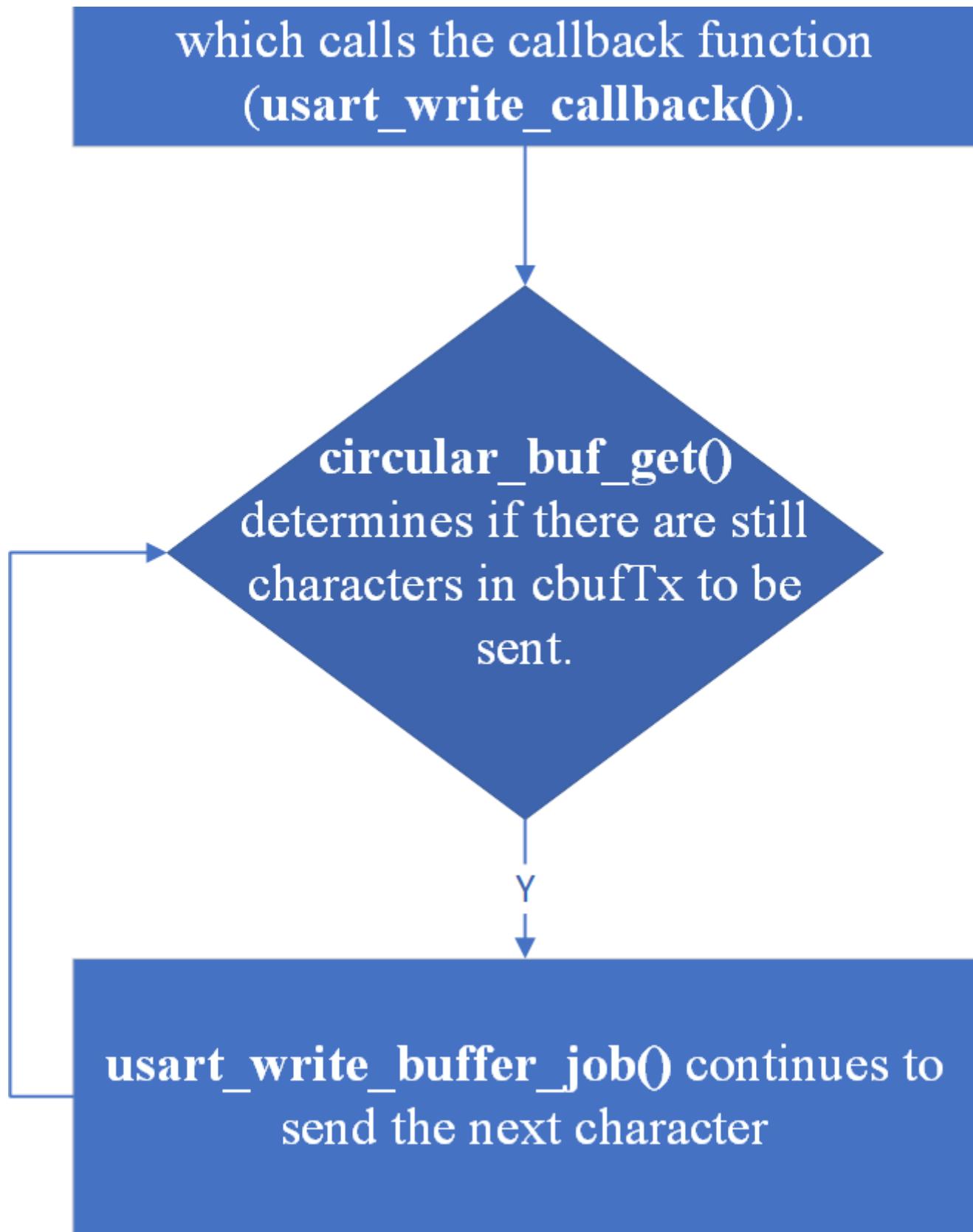
UART transmission:

Use **SerialConsoleWriteString()** to store the string into the circular buffer **cbufTx**.

Call **uart_get_job_status()** to check if the USART is idle.

Send the first character using **uart_write_buffer_job()**

Triggers a "Transmission Complete (TXC)" interrupt,

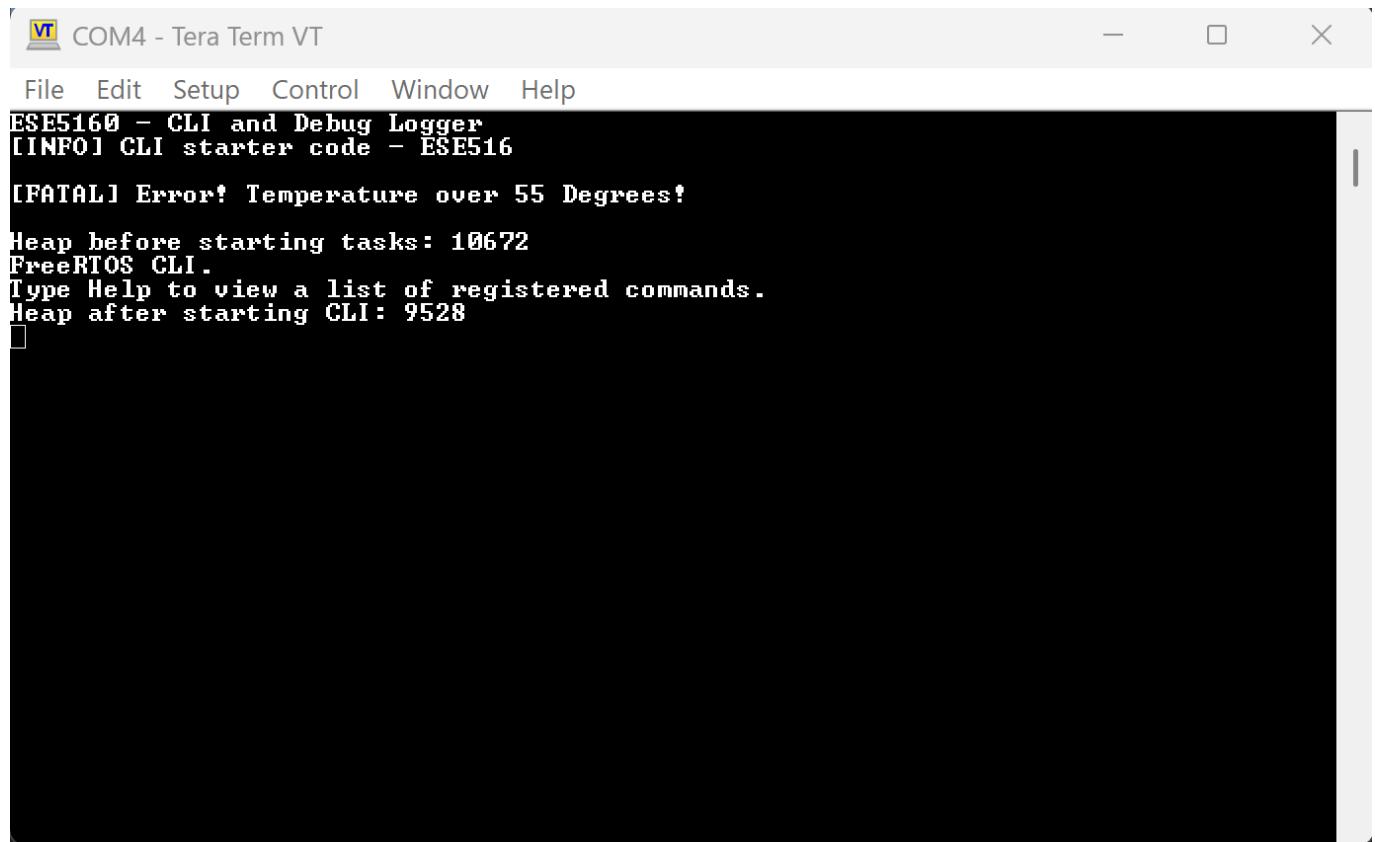


2.9

1. Functions:
 1. Initialize and create FreeRTOS tasks.
 2. Prints the heap memory size before and after task creation.
2. One thread (CLI_TASK) is started for the project.

3. Debug Logger Module

Done, code is under CLI Starter Code_A07G folder.



The screenshot shows a terminal window titled "COM4 - Tera Term VT". The window has a menu bar with "File", "Edit", "Setup", "Control", "Window", and "Help". Below the menu, the text "ESE5160 - CLI and Debug Logger" is displayed, followed by "[INFO] CLI starter code - ESE516". A red box highlights the error message "[FATAL] Error! Temperature over 55 Degrees!". Below the error message, the text "Heap before starting tasks: 10672" and "FreeRTOS CLI." is visible. A red box highlights the command "Type Help to view a list of registered commands.". At the bottom, the text "Heap after starting CLI: 9528" is shown. The terminal window has standard window controls (minimize, maximize, close) at the top right.

4. Wiretap the convo

4.1

1. SERCOM configurations:

1. TX: SERCOM4_PAD2;
2. RX: SERCOM4_PAD3.

2. Physical Pin:

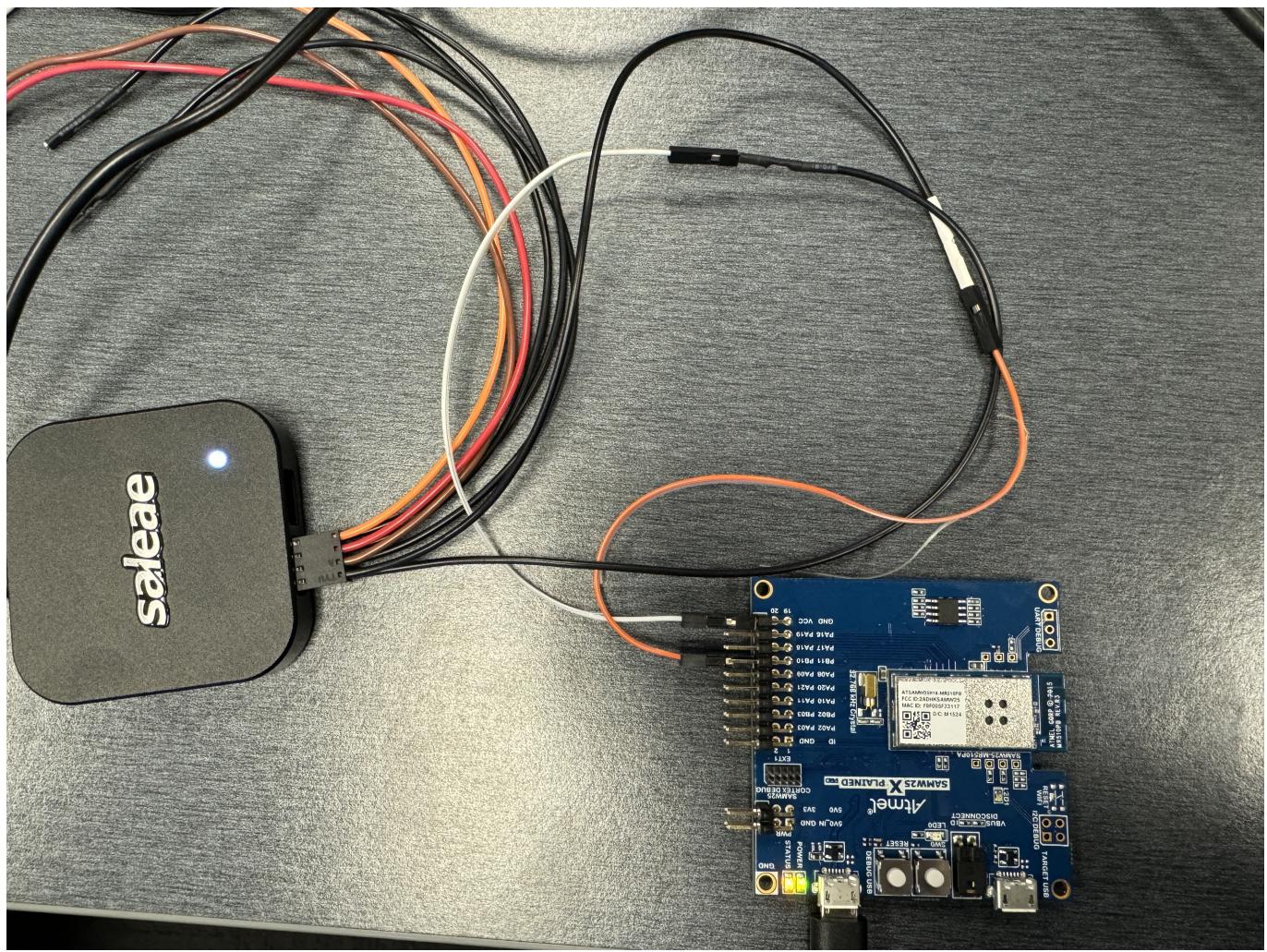
1. TX: PB10;
2. RX: PB11;
3. GND

3. Logic Analyzer Setting:

1. Baud rate: 115200;
2. Data Bit: 8 bit;
3. Stop bit: 1 bit;
4. Parity: None;

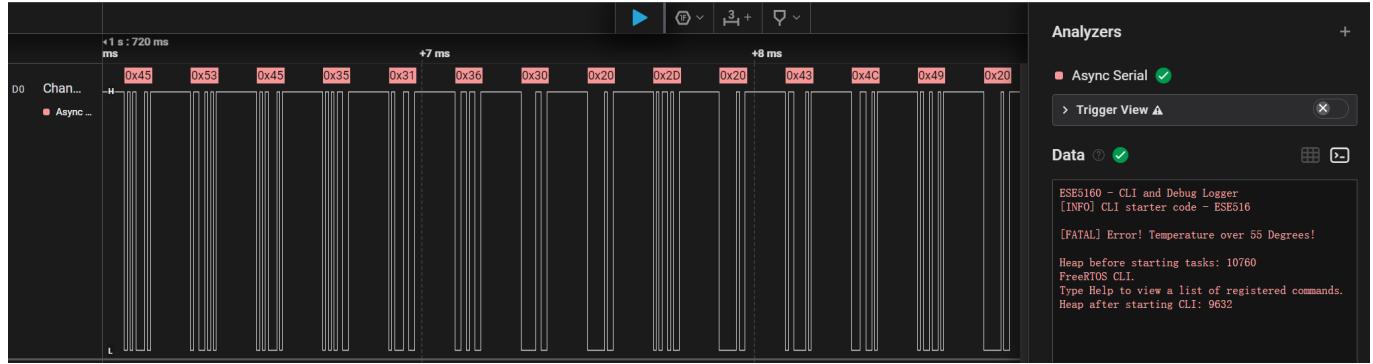
4.2

Hardware Connection:



4.3

Decoded message:

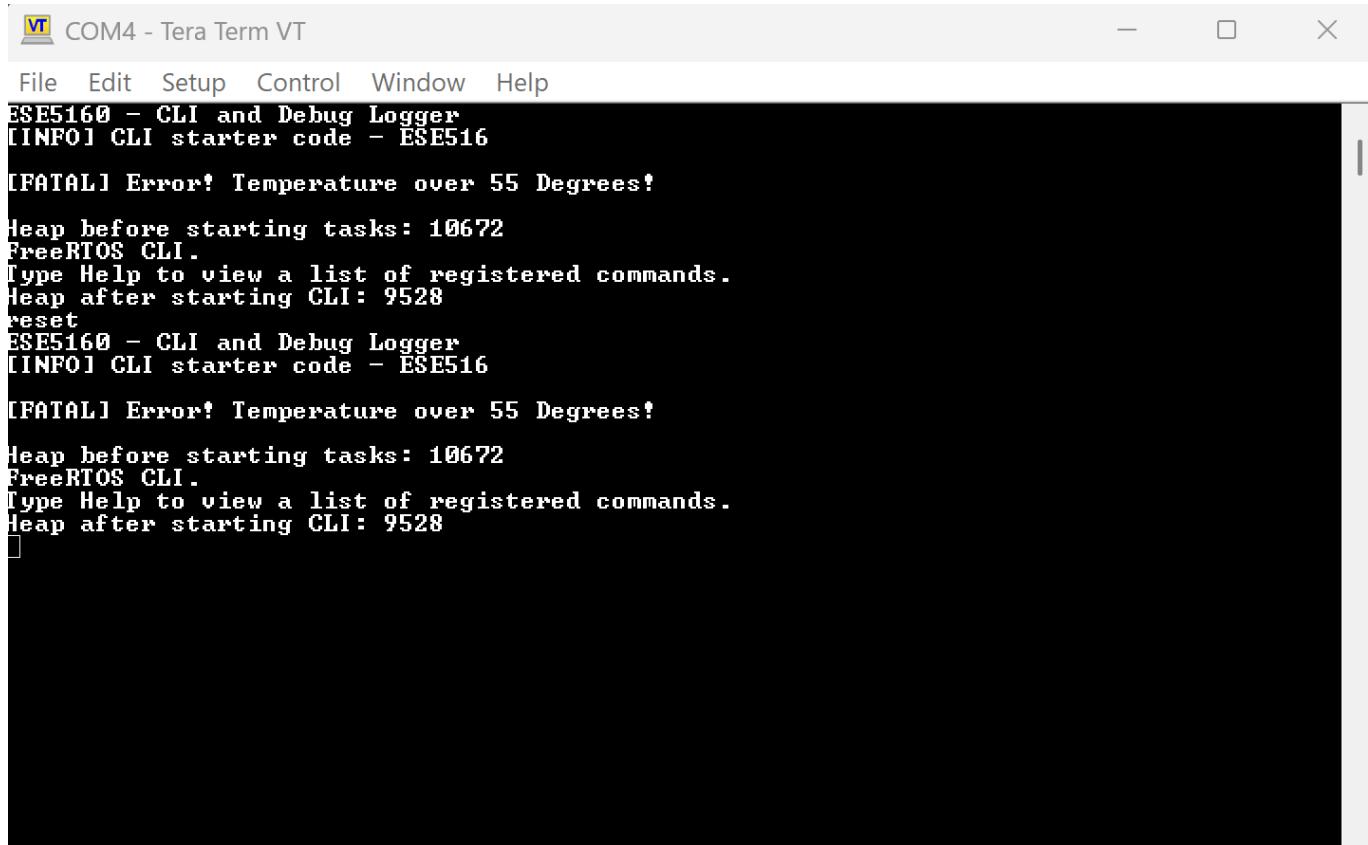


4.4

File path: [uart_capture.sal](#)

5. Complete the CLI

Done, code is under CLI Starter Code_A07G folder.



COM4 - Tera Term VT

File Edit Setup Control Window Help

ESE5160 - CLI and Debug Logger
[INFO] CLI starter code - ESE516

[FATAL] Error! Temperature over 55 Degrees!

Heap before starting tasks: 10672
FreeRTOS CLI.
Type Help to view a list of registered commands.
Heap after starting CLI: 9528
reset

ESE5160 - CLI and Debug Logger
[INFO] CLI starter code - ESE516

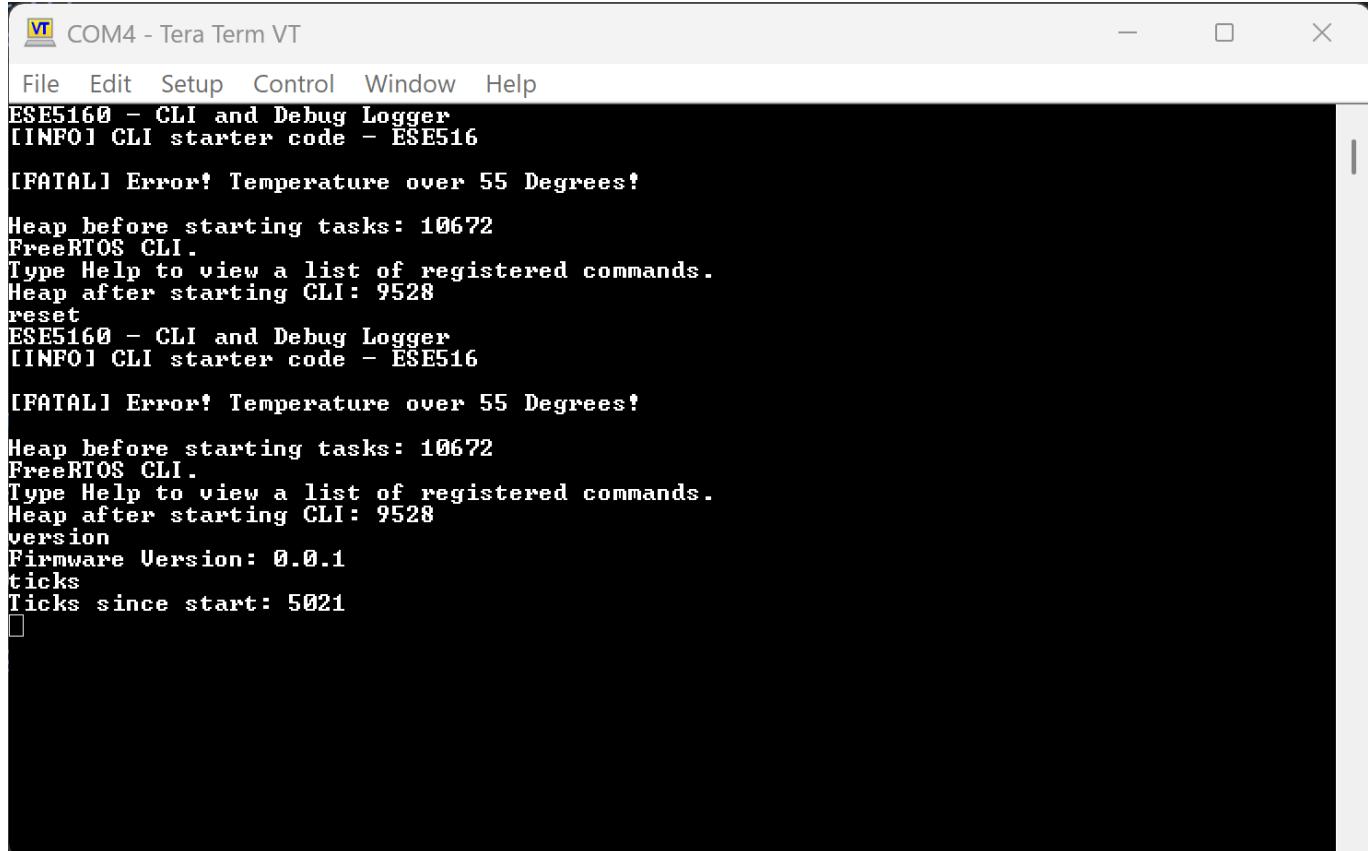
[FATAL] Error! Temperature over 55 Degrees!

Heap before starting tasks: 10672
FreeRTOS CLI.
Type Help to view a list of registered commands.
Heap after starting CLI: 9528

6. Add CLI commands

6.1

Done, code is under CLI Starter Code_A07G folder.



COM4 - Tera Term VT

File Edit Setup Control Window Help

ESE5160 - CLI and Debug Logger
[INFO] CLI starter code - ESE516

[FATAL] Error! Temperature over 55 Degrees!

Heap before starting tasks: 10672
FreeRTOS CLI.
Type Help to view a list of registered commands.
Heap after starting CLI: 9528
reset

ESE5160 - CLI and Debug Logger
[INFO] CLI starter code - ESE516

[FATAL] Error! Temperature over 55 Degrees!

Heap before starting tasks: 10672
FreeRTOS CLI.
Type Help to view a list of registered commands.
Heap after starting CLI: 9528
version
Firmware Version: 0.0.1
ticks
Ticks since start: 5021

6.2

[version_ticks_command_functionality](#)