Neo Yu Yao Terence A0164651E, Neo Jia Wen Rachel A0176763R GA:

Table of Contents

[Introduction and objectives 1](#_Toc528925565)

[Flowcharts describing the system design and processes 1](#_Toc528925566)

[Detailed implementation 1](#_Toc528925567)

[Enhancement 1](#_Toc528925568)

[Significant problems encountered and solutions proposed 2](#_Toc528925569)

[Issues or suggestions 2](#_Toc528925570)

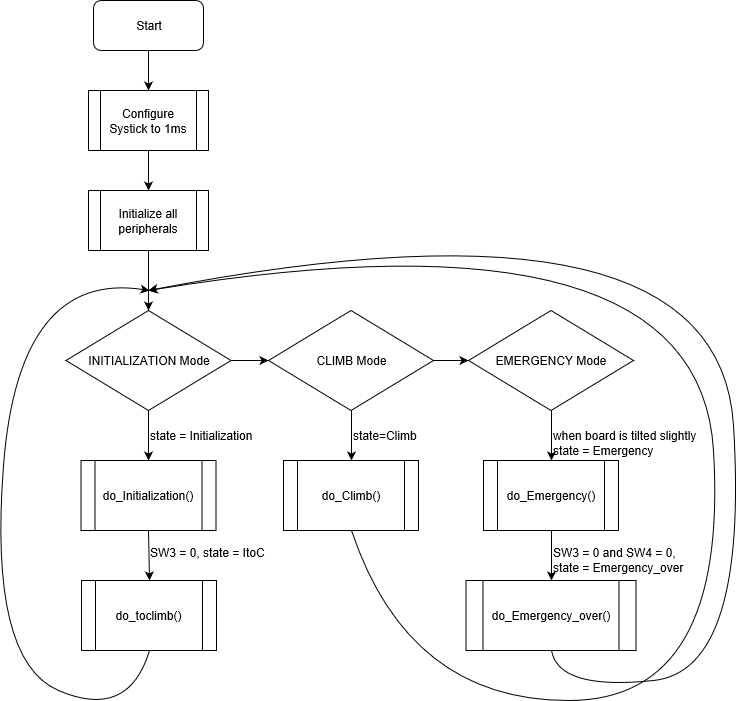
[Conclusion 2](#_Toc528925571)

# Introduction and objectives

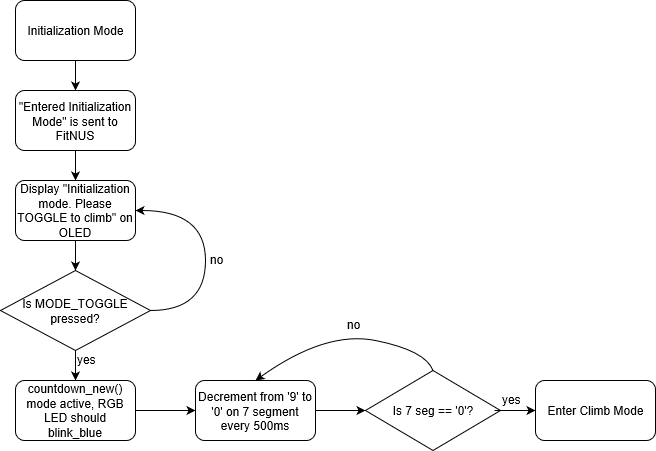
In this assignment, our group is tasked to implement a fitness tracking system, **FitNUS**. The main purpose of **FitNUS** is to boost daily workouts and make them easier to achieve. **FitNUS** detects acceleration/deceleration, light and temperature changes. **FitNUS** sends data periodically to a server known as **FiTrackX**.

# Flowcharts describing the system design and processes

* int main (void) flowchart
  + Configures systick timing to 1ms, initialize all peripherals and interrupts. Subsequently checks which mode **FitNUS** is when powered on.



* do\_Initialization(), do\_toclimb() flowchart



* do\_Climb() flowchart
* do\_Emergency() flowchart, do\_Emergency\_over() flowchart
* UART flowchart
* EINT3\_IRQHandler flowchart
* Light sensor flowchart, Temperature flowchart, Acc flowchart
* Rotary switch flowchart
* Joystick flowchart

# Detailed implementation

Summary of **FitNUS**: There are 3 modes in our **FitNUS** system: INITIALIZATION Mode, CLIMB Mode and EMERGENCY Mode.

* INITIALIZATION Mode would be active when the **FitNUS** system is first switched ON.
  + The OLED should display “Initialization mode. Press TOGGLE to climb” and a message “Start” should be sent once to **FiTrackX**.
  + Sensors will not be reading any data (Temperature, Light and Accelerometer).
  + No UART transmission would be sent to **FitNUS**.
* CLIMB Mode would be active when MODE\_TOGGLE (SW3) is activated
  + The OLED should display “CLIMB mode” when **FitNUS** first enters this mode.
  + The 7-segment will display the countdown, where the value decrement from 9-0 every 500ms. When criteria are met, the RGB led will blink periodically.
  + The sensors will receive their respective values and store them in variables to be utilized by other functions.
  + The OLED should display the values obtained. Additionally, it should continue to update the values periodically.
  + **FitNUS** will transmit date wirelessly to **FiTrackX** at intervals (UART).
* EMERGENCY Mode would be active when the user decides to trigger fall detection in CLIMB Mode
  + The OLED should display “EMERGENCY!” when **FitNUS** first enters this mode.
  + **FitNUS** will transmit date wirelessly to **FiTrackX** at intervals (UART).
  + When MODE\_TOGGLE (SW3) and EMERGENCY\_OVER (SW4) are pressed simultaneously, the OLED should display “Emergency is cleared! Time consumed for recovery: xx sec”.

# Enhancement

If you have implemented any enhancement, give a detailed description. You might consider including several photos of your working board at some special steps. This will help to distinguish your system and report from others.

After meeting the basic requirements of **FitNUS**, our group implemented the following enhancements to make the system more user-friendly, responsive and useful.

* In order to make it more user-friendly, we modified the led7seg\_setChar() to display the inverted mode of characters on the 7 Segment to match the orientation of the OLED.
  + Since the 7 segment display is an active low device, a 0 in the bit pattern will make their respective strokes on the 7 segment light up, while a 1 will turn it off. To make a digit appear on the 7 segment, we followed the order of FBDpC AGED and put a 0 in the right position. We inverted numbers ‘0’ to ‘9’ and alphabets ‘S’, ‘A’, ‘U’, ‘E’, ‘D’.
  + **static** uint8\_t numbers\_inverted[] = {0x24, 0x7D, 0xE0, 0x70, 0x39, 0x32, 0x22, 0x7C, 0x20, 0x38, 0xFF};
  + **char** saued[] = {0x32, 0x28, 0x25, 0xA2, 0x24};
* Utilize additional peripherals (joystick, rotary) as interrupts
* Music playlist
* UART interrupt
* Watchdog timer(?)

# Significant problems encountered and solutions proposed

What did you learn? What are the significant problems you encountered and how did you solve them in this assignment? If your code did not work in the lab, explain why.

Green RGB conflicts with OLED, PIO1\_10, Port 2 Pin one used by RGB\_GREEN and OLED

Changing of jumper positions from default value

1. Remove j28 for sw4 to work
2. Remove j23 to turn off green\_led

# Issues or suggestions

These feedbacks, whether positive or negative, will not affect your marks in any way, but will make the report more complete.

As this is the first hardware programming project we did on LPC, there were many times when we got stuck, baffled by lines of codes that seems to work but did not. Fortunately, we were able to readily consult the various teachers, lab staff, and graduate assistances who are not only very knowledgeable, spotting our errors instantly, but also extremely patient when explaining the concepts to us. And we are very grateful for you all!

# Conclusion