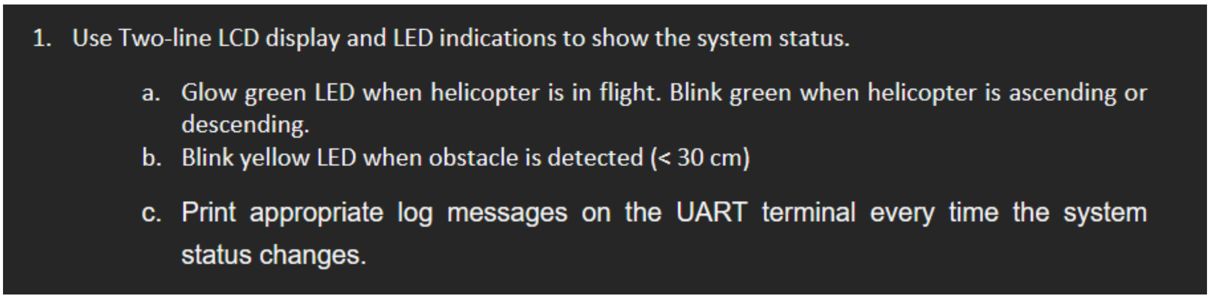
FINAL ASSIGNMENT E3 -257

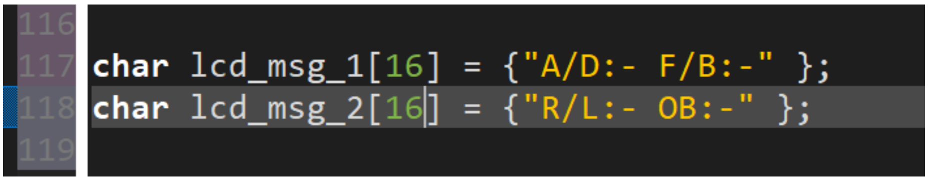
Primitive flight control system for helicopter

BASATI SIVAKRISHNA (22976)

M Tech, EPD.

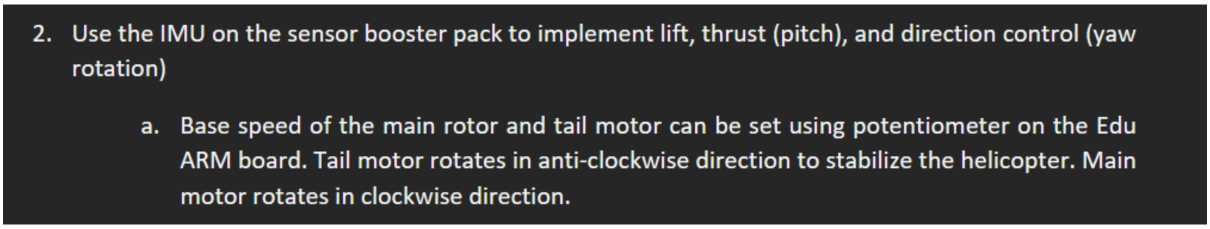


1. Use Two-line LCD display and LED indications to show the system status.
   a. Glow green LED when helicopter is in flight. Blink green when helicopter is ascending or descending.
   b. Blink yellow LED when obstacle is detected (< 30 cm)
   c. Print appropriate log messages on the UART terminal every time the system status changes.

- LCD is initialized and write_string function has written with string and line to be printed as parameter.
- Initially LCD screen will show the Motors checking status and after that it will show the system status.
- System status is nothing but showing whether there is an obstacle or not and there is forward/ backward movement and right or left movement and whether the flight is ascending or descending.
- MPU Sensor is initialized by using the compdcm libraries given by TI. And ultrasonic sensor is Initialised by using HC_SR04_setup () function.
- Using APIs provided by Tis compdcm Library, the acceleration towards Z direction is acquired and if there is a change in that Z value then accordingly LCD status and led status has updated.
- By using send_trigger_pulse() function, distance from the obstacle is calculated and if the distance is less than 30 cms then the yellow led will blink with 100mSec Rate.
- 

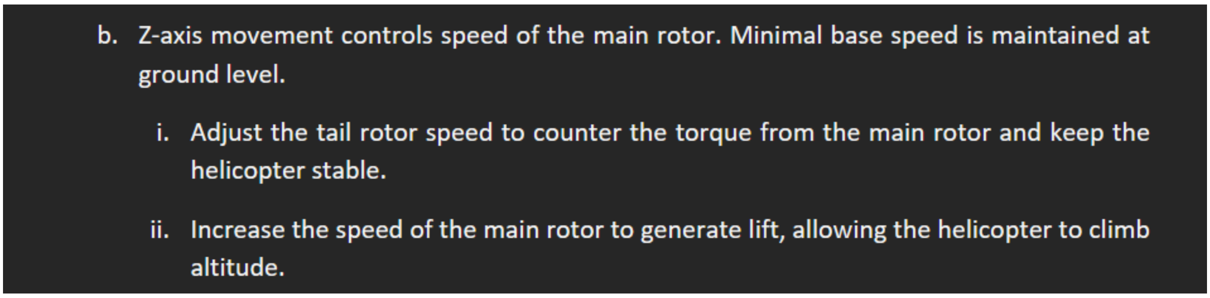
```
char lcd_msg_1[16] = {"A/D:- F/B:-" };
char lcd_msg_2[16] = {"R/L:- OB:-" };
```

- Above string will be printed on the LCD screen.
- And UART terminal will have all sensor values i.e.
  - Accelerometer (X, Y, Z)
  - GYROSCOPE (Pitch, Yaw, Roll)
  - Ultrasonic sensor distance.



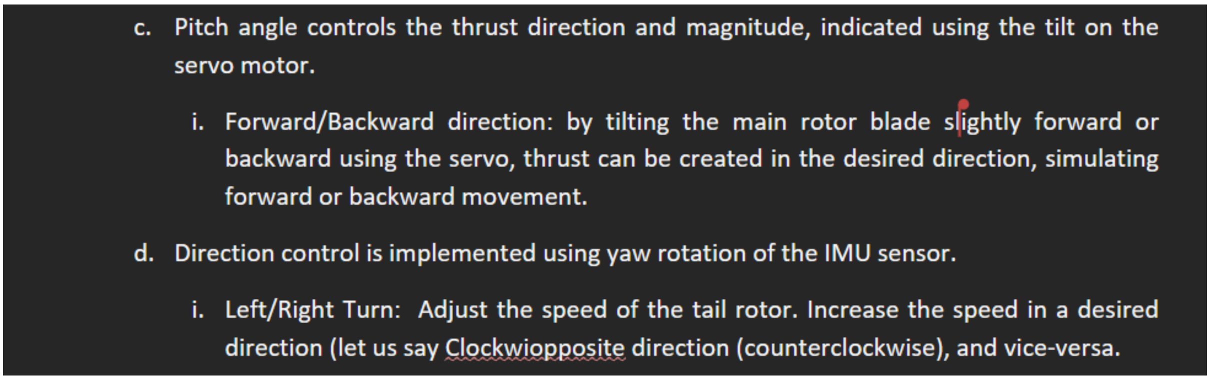
2. Use the IMU on the sensor booster pack to implement lift, thrust (pitch), and direction control (yaw rotation)
   a. Base speed of the main rotor and tail motor can be set using potentiometer on the Edu ARM board. Tail motor rotates in anti-clockwise direction to stabilize the helicopter. Main motor rotates in clockwise direction.

- Potentiometer connected to Edu-Arm pin is initialized as ADC pin and the ADC value (ranging from 0 to 4095) is mapped to equivalent duty cycle of PWM for both.
- Each PWM is generated by using Systick handler and respective timer.
- **<u>GENERATING PWM:</u>**
  - **P**WM is generated by using timer. Systick is configured to generate an interrupt for every 10mSec, it will pull the 3 GPIO pins to Low.
  - Accordingly, systick handler will also turn off the timers and load pwm equivalent value into load registers of the timers and turn them again.
  - Equivalent PWM value will be generate an interrupt for the next loaded value which is less than 10mSec. And the Timer handler will pull the respective pin as high.
  - So, for example if we want 75% duty cycled PWM signal we should load timer with 2.5mSec equivalent value (should calculate from the clock frequency).



b. Z-axis movement controls speed of the main rotor. Minimal base speed is maintained at ground level.
   i. Adjust the tail rotor speed to counter the torque from the main rotor and keep the helicopter stable.
   ii. Increase the speed of the main rotor to generate lift, allowing the helicopter to climb altitude.

- **O**nce the base speed has set, according to the change in acceleration towards Z value is mapped into the PWM equivalent value of main motor.
- To increase the speed, we need to increase the duty cycle. This can be done by loading timer load register with lower value.



c. Pitch angle controls the thrust direction and magnitude, indicated using the tilt on the servo motor.

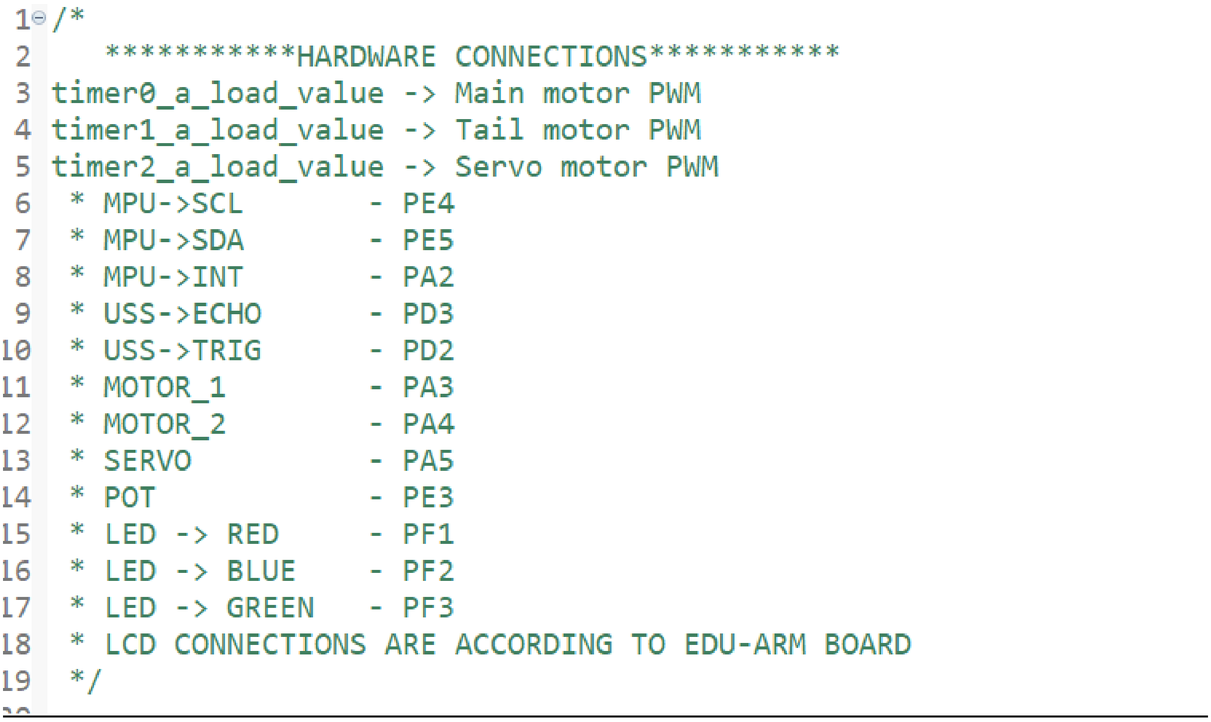
   i. Forward/Backward direction: by tilting the main rotor blade slightly forward or backward using the servo, thrust can be created in the desired direction, simulating forward or backward movement.

d. Direction control is implemented using yaw rotation of the IMU sensor.

   i. Left/Right Turn: Adjust the speed of the tail rotor. Increase the speed in a desired direction (let us say Clockwiopposite direction (counterclockwise), and vice-versa.

- **B**y using the EULERS formula and APIs provided by comp_dcm.h library provided by the we have obtained Pitch, Roll and Yaw interms of degrees.
- If copter wants to move toward forward direction, then Pitch will be increase and resultant degrees has mapped to servo motor rotation.
- Note: Servo motor needs only 2.5% to 12.5% duty cycled PWM to complete entire 180 degrees rotation.
- So the value came from changing Yaw is mapped to the PWM value of the tail motor.

Hardware Connections:



```
/*
   ***********HARDWARE CONNECTIONS***********
timer0_a_load_value -> Main motor PWM
timer1_a_load_value -> Tail motor PWM
timer2_a_load_value -> Servo motor PWM
 * MPU->SCL       - PE4
 * MPU->SDA       - PE5
 * MPU->INT       - PA2
 * USS->ECHO      - PD3
 * USS->TRIG      - PD2
 * MOTOR_1        - PA3
 * MOTOR_2        - PA4
 * SERVO          - PA5
 * POT            - PE3
 * LED -> RED     - PF1
 * LED -> BLUE    - PF2
 * LED -> GREEN   - PF3
 * LCD CONNECTIONS ARE ACCORDING TO EDU-ARM BOARD
 */
```

Note: The code provided will work fine only with ARM compiler version 5.2.6