



॥ सा विद्या या विमुक्तये ॥

भारतीय प्रौद्योगिकी संस्थान धारवाड़

Indian Institute of Technology Dharwad

Semester-7 Embedded Systems Laboratory  
Course Project

# Design of Stopwatch

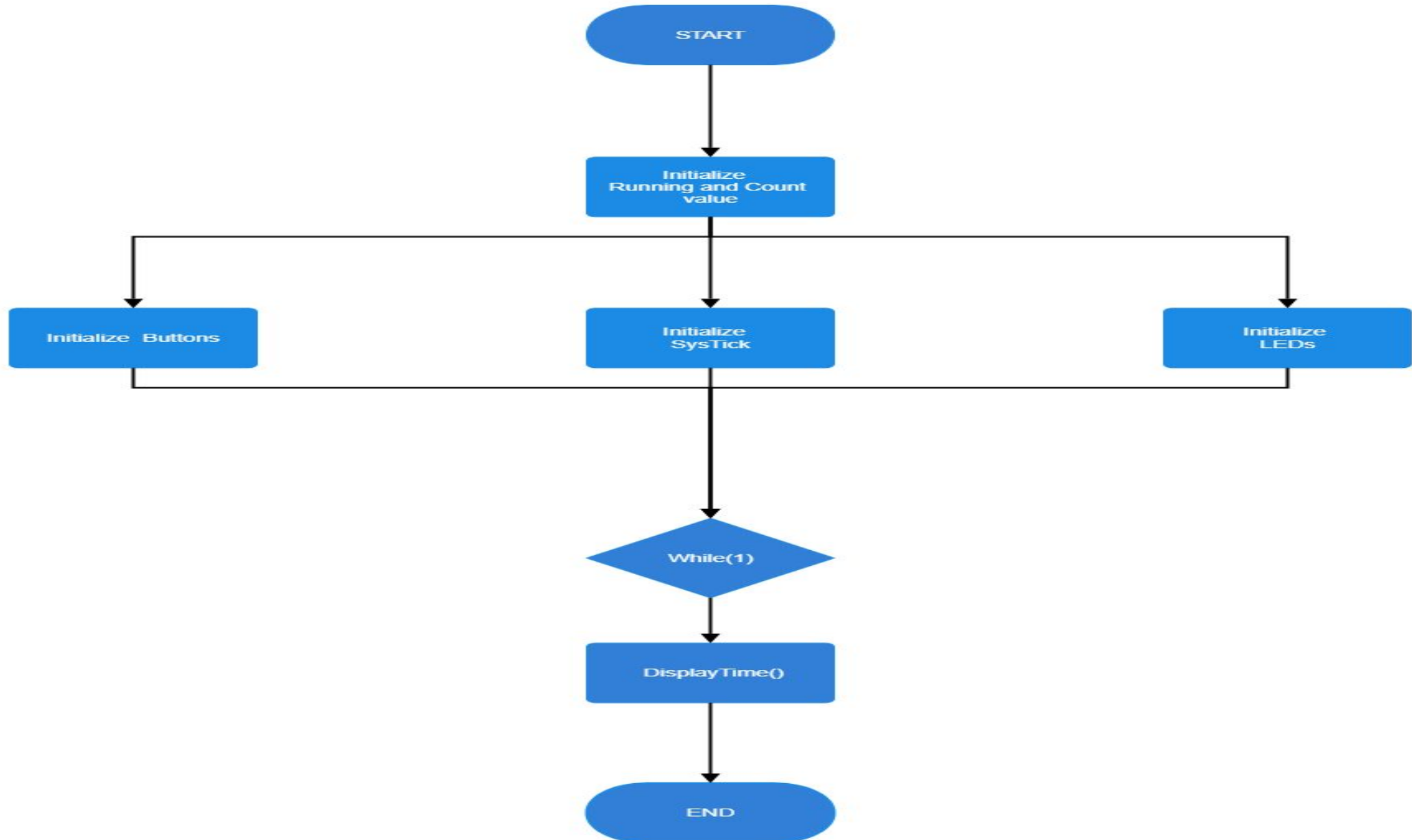
Supervisor:  
Dr Abhijit Kshirsagar  
Assistant Professor, Dept. of Electrical Engineering

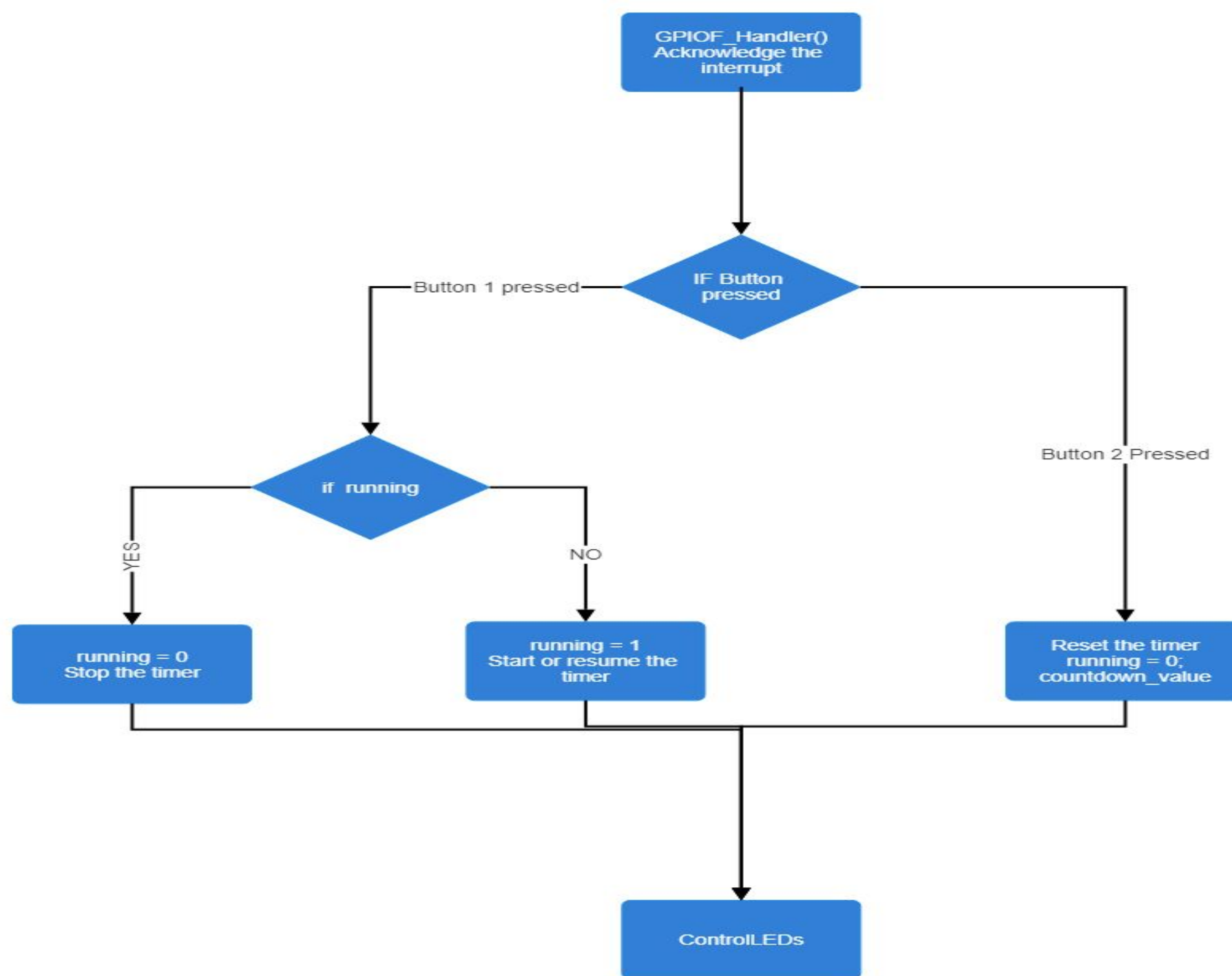
Group 12:  
Tirumula Reddy B H - EE23MT022  
Tanish H Talapaneni - 200020050

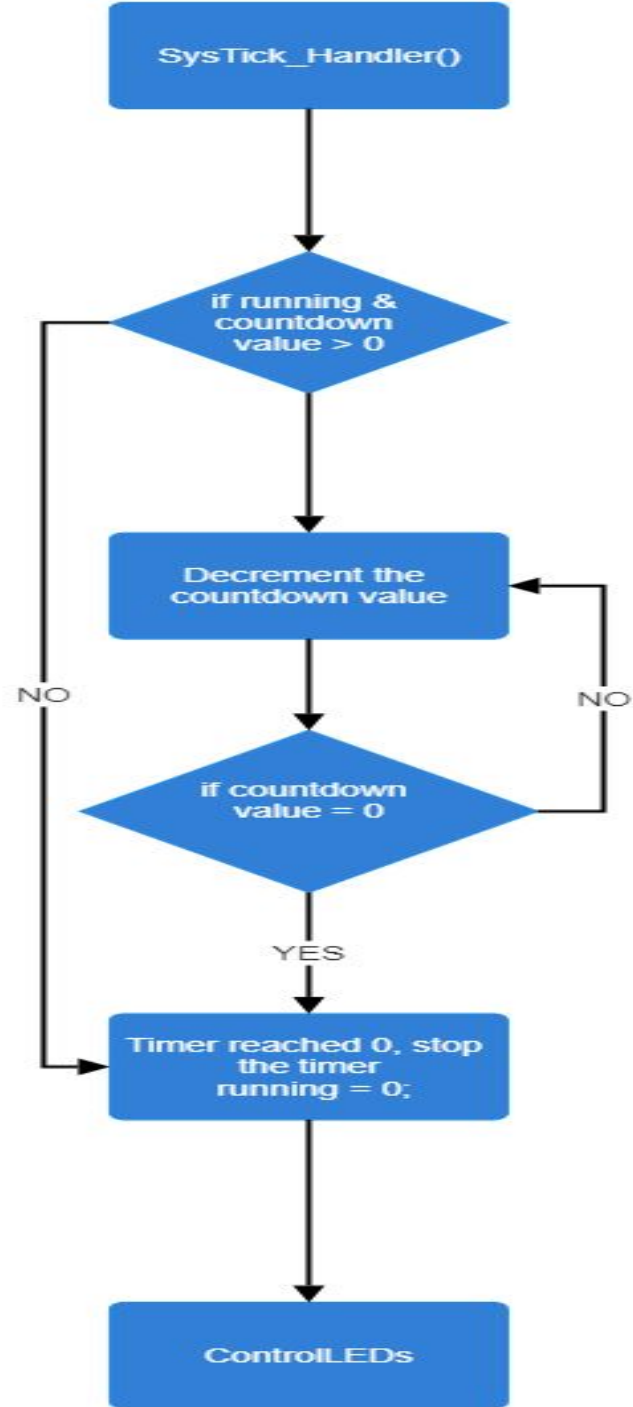
# Regular Timer

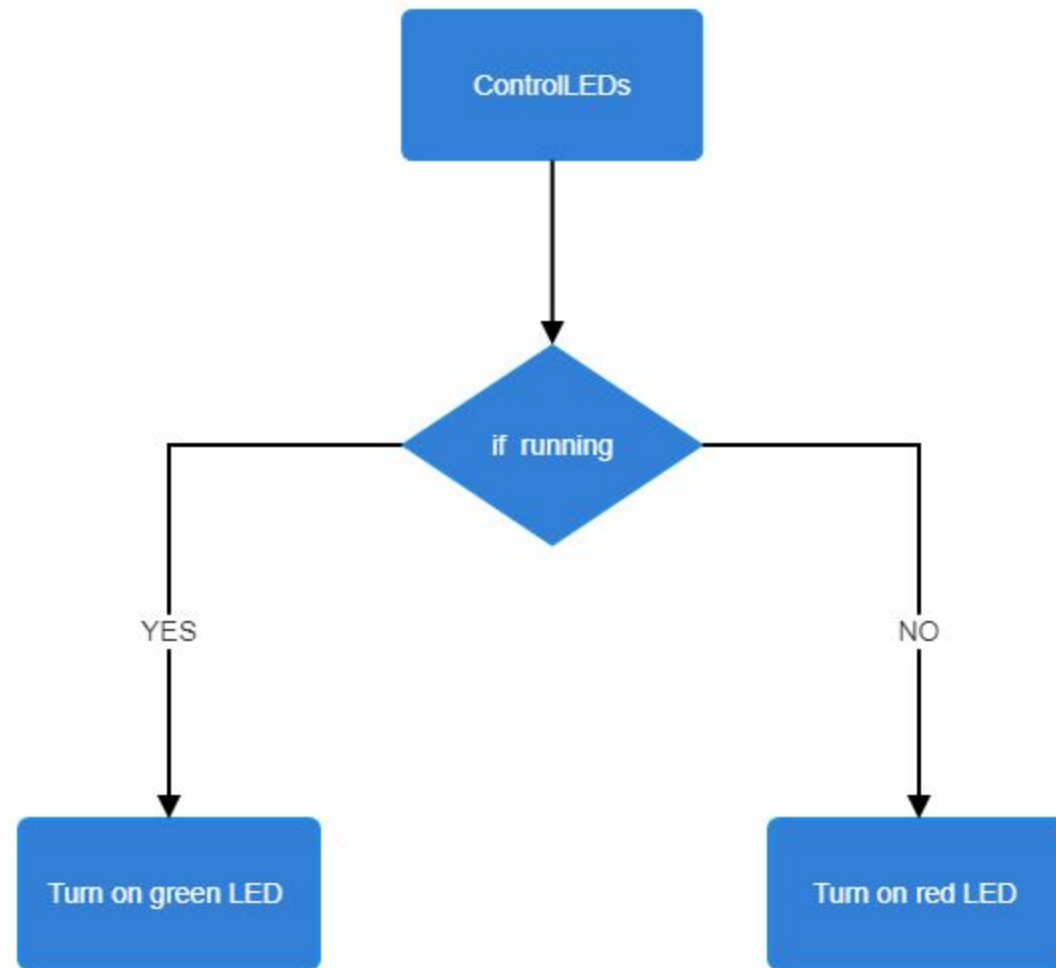
- User presses button 1 to start the timer.
- button 2 pressed it will reset.
- Button 1 for start, pause, resume
- LEDs show the status of the timer.

# Flowchart




















# sample output

Expression	Type	Value	Address
 hours	unsigned int	0	0x20000210
 seconds	unsigned int	10	0x20000208
 countdown_value	unsigned int	10000	0x20000200
 minutes	unsigned int	0	0x2000020C
 Add new expression			

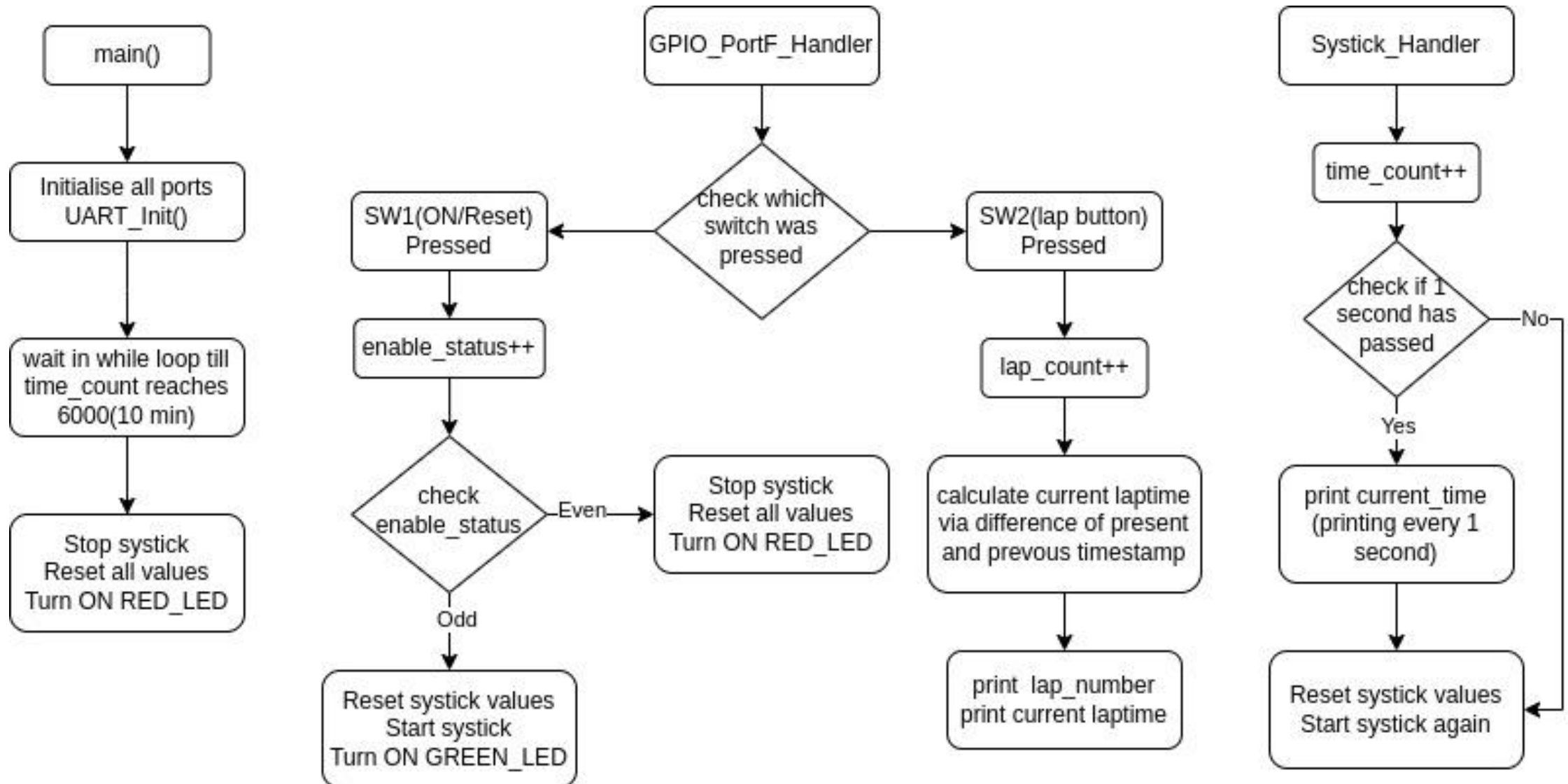
Expression	Type	Value	Address
 hours	unsigned int	0	0x20000210
 seconds	unsigned int	10	0x20000208
 countdown_value	unsigned int	10000	0x20000200
 minutes	unsigned int	0	0x2000020C
 seconds	unsigned int	4	0x20000208
 Add new expression			

# Laptime mode Operation

- SW1 is for start and reset.
- Time is displayed at regular intervals via UART.
- SW2 acts as lap button. When user presses it, the current laptime is shown and the current time is stored in an array of timestamps .
- The present laptime is calculated by subtracting the previous timestamp from the present timestamp.
- LEDs show the status of the timer:
  - Red: Timer not running
  - Green: Timer running



# Flowchart



# Laptime mode Operation Stats

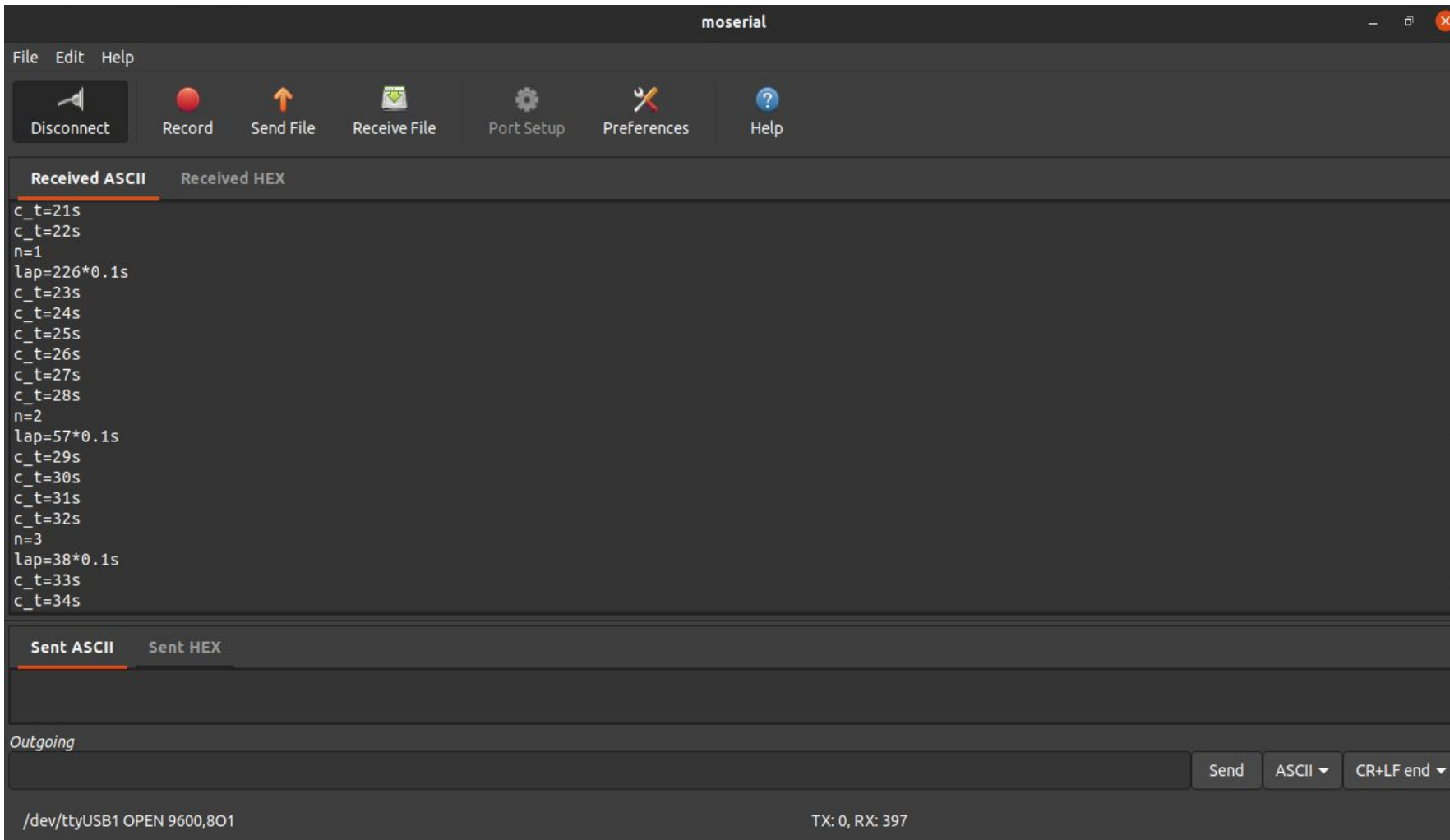
- Range= 0-10min
- Resolution= 0.1s (each systick cycle is for a duration of 100ms)

Typical use case: In a multi-lap race, a person's different laptimes can be recorded.

# Limitations

- Time lost while transmitting via UART.
- Stack size had to be set high to account for large strings and to use sprintf function.

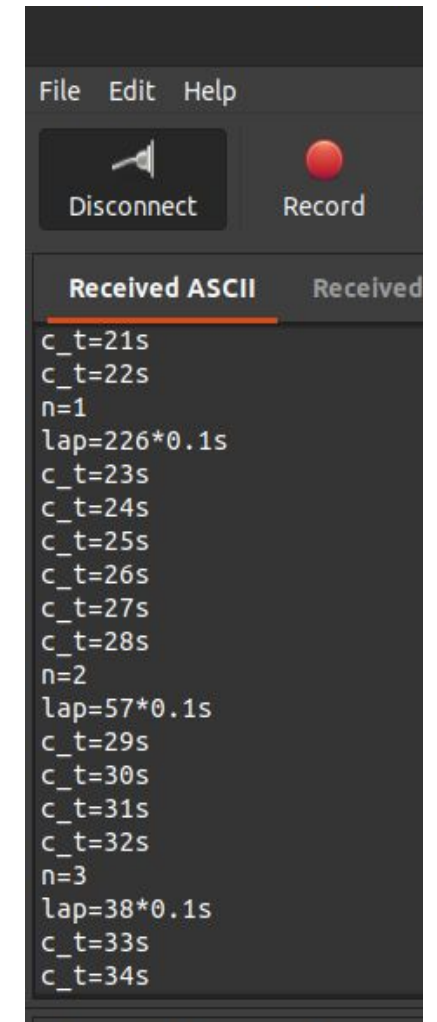
# Sample Output



The screenshot shows the 'moserial' application window. The menu bar includes 'File', 'Edit', and 'Help'. The toolbar contains icons for 'Disconnect', 'Record', 'Send File', 'Receive File', 'Port Setup', 'Preferences', and 'Help'. Below the toolbar, there are two tabs: 'Received ASCII' (selected) and 'Received HEX'. The main text area displays the following received data:

```
c_t=21s  
c_t=22s  
n=1  
lap=226*0.1s  
c_t=23s  
c_t=24s  
c_t=25s  
c_t=26s  
c_t=27s  
c_t=28s  
n=2  
lap=57*0.1s  
c_t=29s  
c_t=30s  
c_t=31s  
c_t=32s  
n=3  
lap=38*0.1s  
c_t=33s  
c_t=34s
```

Below the main text area, there are two tabs: 'Sent ASCII' (selected) and 'Sent HEX'. At the bottom, there is an 'Outgoing' section with a text input field, a 'Send' button, and dropdown menus for 'ASCII' and 'CR+LF end'. The status bar at the bottom shows the port path '/dev/ttyUSB1 OPEN 9600,801' and the transmission/reception counts 'TX: 0, RX: 397'.



This screenshot shows a zoomed-in view of the 'moserial' application window, focusing on the 'Received ASCII' tab. The received data is the same as in the first screenshot:

```
c_t=21s  
c_t=22s  
n=1  
lap=226*0.1s  
c_t=23s  
c_t=24s  
c_t=25s  
c_t=26s  
c_t=27s  
c_t=28s  
n=2  
lap=57*0.1s  
c_t=29s  
c_t=30s  
c_t=31s  
c_t=32s  
n=3  
lap=38*0.1s  
c_t=33s  
c_t=34s
```

# Further Scope

- Configuration of Wide Timer in capture mode(better resolution)
- Integration with SD Storage to store the recorded time data

**Thank you!!**

Questions?