Mini Project Report Cover Sheet

SRM Institute of Science and Technology College of Engineering and Technology Department of Electronics and Communication Engineering
18ECE204J ARM based Embedded System Design **Sixth Semester, 2020-21 (Even Semester)**

Name **:** Muralidhar B

Register No. **:** RA1811004010206

Title of the project **: Traffic light system using MBED simulator**

Lab Supervisor **:** Ms. Padmajothi.V Ma'am

Reg. No		RA1811004010206
Mark split up	Maximum Marks	Marks obtained
Novelty in the project work / Abstract	**5**	
Level of understanding of the design / Configuration	**10**	
Individual Contribution to the project	**5**	
Report writing	**5**	
Total	**25**	

REPORT VERIFICATION

Lab supervisor Signature with date :

Traffic Light System Using MBED Simulator

OBJECTIVE

To successfully design a fully functional traffic light system in MBED simulator using interrupts and timers.

SOFTWARE DETAIL

Mbed OS provides the Mbed C/C++ software platform and tools for creating microcontroller firmware that runs on IoT devices. It consists of the core libraries that provide the microcontroller peripheral drivers, networking, RTOS and runtime environment, build tools and test and debug scripts. These connections can be secured by compatible SSL/TLS libraries such as Mbed TLS or wolfSSL, which supports mbed-rtos.

Unlike Fast Models, the Mbed Simulator is not fully simulating a Cortex-M device. Instead, it cross-compiles Mbed OS 5 using Emscripten. Emscripten is an LLVM to JavaScript compiler, which takes in a complex C++ project and spits out something that can be run in a web environment

INTRODUCTION

Traffic lights, also known as traffic signals, are signaling devices positioned at pedestrian crossings, road intersections, and other locations to control Intersections for the flow of traffic. Currently, traffic lights have been installed in most small and major cities around the world to control the flow of traffic. It allocates the right of way to road users using lights in standard colors (Red - Yellow – Green). Traffic lights are used at busy intersections to aid the flow of traffic more smoothly and safely. The increasing amount of traffic in cities have a large impact on the overcrowding and the time it takes to reach destinations. The method of adding more roads is not enough, since they will always reach an endpoint, like intersections or traffic jams. Traffic jams cannot be prevented.

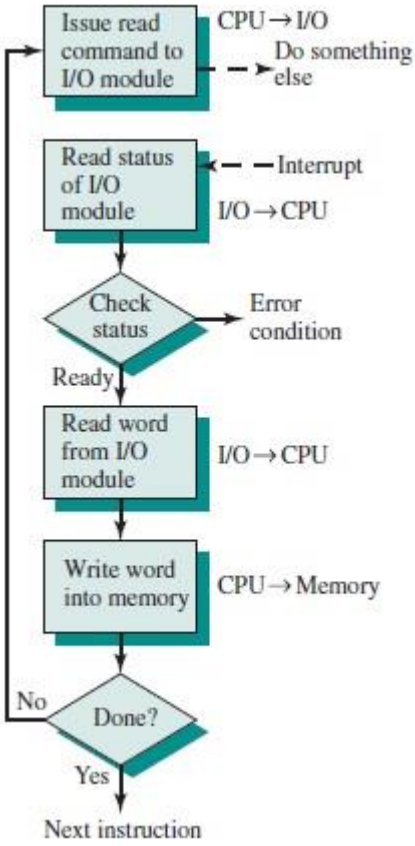
However, the way intersections are controlled has room for improvement. Intersections are controlled mainly by traffic lights and will reduce the amount of traffic jams, especially during rush hour. For this project, we created a prototype traffic light model using LEDs. Since LEDs consume less energy and generally last longer compared to any other light source. This project actively involved the use of a digital logic gate, integrated circuit, timer for our entire design and circuit simulation to help us analyze the entire system efficiently

An embedded system uses its input/output devices to interact with the external world. Input devices allow the computer to gather information, and output devices can display information. Output devices also allow the computer to manipulate its environment. The tight-coupling between the computer and external world distinguishes an embedded system from a regular computer system.

The challenge is under most situations the software executes much faster than the hardware. E.g., it might take the software only 1 us to ask the hardware to clear the LCD, but the hardware might take 1 ms to complete the command.

During this time, the software could execute tens of thousands of instructions. Therefore, the synchronization between the executing software and its external environment is critical for the success of an embedded system.

An interrupt is the automatic transfer of software execution in response to a hardware event that is asynchronous with the current software execution. This hardware event is called a trigger. The hardware event can either be a busy to ready transition in an external I/O device (like the UART input/output) or an internal event (like bus fault, memory fault, or a periodic timer).
When the hardware needs service, signified by a busy to ready state transition, it will request an interrupt by setting its trigger flag.
A thread is defined as the path of action of software as it executes. The execution of the interrupt service routine is called a background thread. This thread is created by the hardware interrupt request and is killed when the interrupt service routine returns from interrupt.
A new thread is created for each interrupt request. It is important to consider each individual request as a separate thread because local variables and registers used in the interrupt service routine are unique and separate from one interrupt event to the next interrupt.



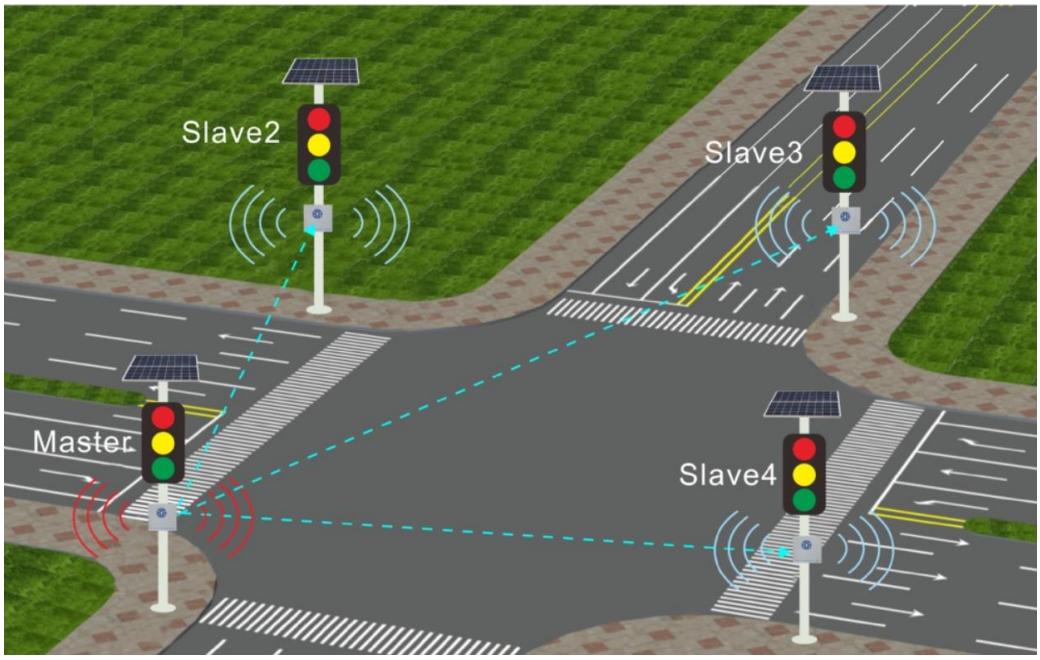
1. A device driver initiates an I/O request on behalf of a process.
2. The device driver signals the I/O controller for the proper device, which initiates the requested I/O.
3. The device signals the I/O controller that is ready to retrieve input, the output is complete or that an error has been generated.
4. The CPU receives the interrupt signal on the interrupt-request line and transfer control over the interrupt handler routine.
5. The interrupt handler determines the cause of the interrupt, performs the necessary processing and executes a "*return from*" interrupt instruction.
6. The CPU returns to the execution state prior to the interrupt being signalled.
7. The CPU continues processing until the cycle begins again.

Timer and counter is very important feature which allows us to provide time variable to our microcontroller based project. Most microcontrollers comes with built-in timer peripheral. The LPC2148 has two functionally identical general purpose timers: Timer0 and Timer1. These both timers are 32-bit along with 32-bit prescaler.

Timer allows us to generate precise time delay. For Example: In blink LED example, we've generated random delay of approximate 1 Sec. but using Timers we can generate accurate time delay. Apart from this we can use timers as pulse width modulator and also as free running timer. The heart of timers of the LPC2148 Microcontroller is a 32-bit free running counter, which is designed to count cycles of the Peripheral Clock (PCLK) or an external clock, this counter is programmable with 32-bit prescaler.

The tick rate of the Timer Counter (TC) is controlled by the 32-bit number written in the Prescaler Register (PR) in the following way. There is a Pre scale Counter (PC) which increments on each tick of the PCLK. When it reaches the value in the prescaler register, the timer count is incremented and the Prescaler Counter (PC) is reset, on the next PCLK. This cause the timer counters to increment on every PCLK when PR=0, every 2 PCLKs when PR=1, etc.

PRE-EXECUTION DIAGRAM



PROGRAM

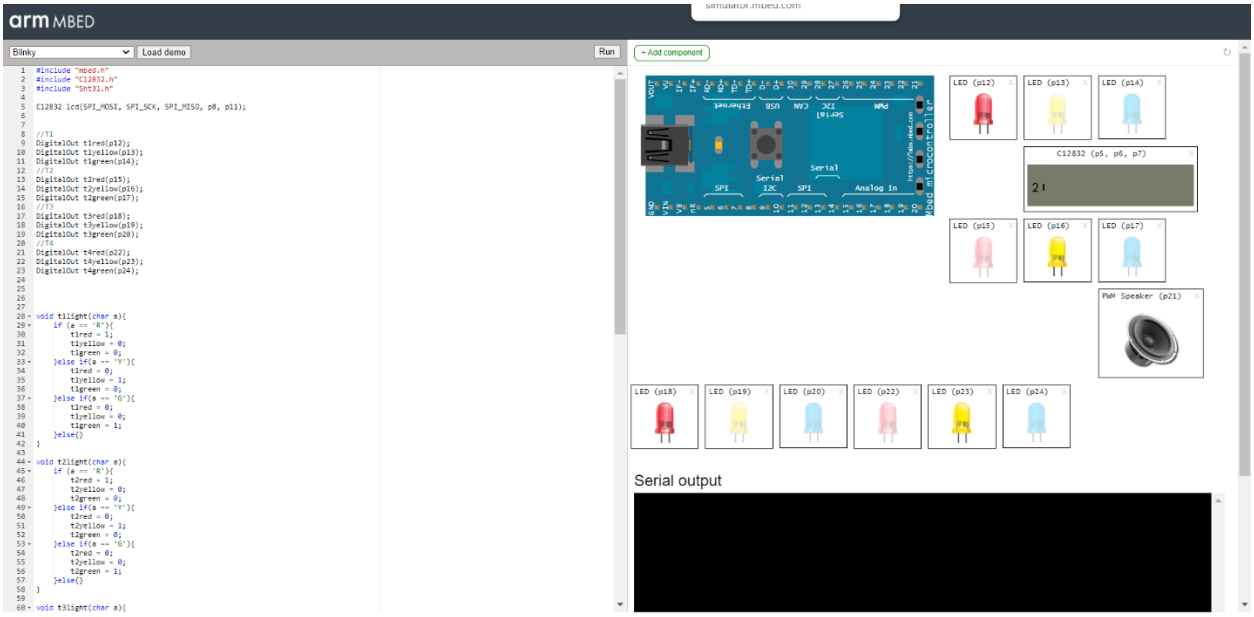
```
#include "mbed.h"
#include "C12832.h"

C12832 lcd(SPI_MOSI, SPI_SCK, SPI_MISO, p8, p11); //LCD Display as Timer
PwmOut speaker(p21); // Speaker to act as an alarm for pedestrians

void play_tone(float frequency, float volume, int interval, int rest) {
   speaker.period(1.0 / frequency);
   speaker = volume;
   wait(interval);
   speaker = 0.0;
   wait(rest);
}


//T1
DigitalOut t1red(p12);
DigitalOut t1yellow(p13);
DigitalOut t1green(p14);
//T2
DigitalOut t2red(p15);
DigitalOut t2yellow(p16);
DigitalOut t2green(p17);
//T3
DigitalOut t3red(p18);
DigitalOut t3yellow(p19);
DigitalOut t3green(p20);
//T4
DigitalOut t4red(p22);
DigitalOut t4yellow(p23);
DigitalOut t4green(p24);



void t1light(char a){
   if (a == 'R'){
```

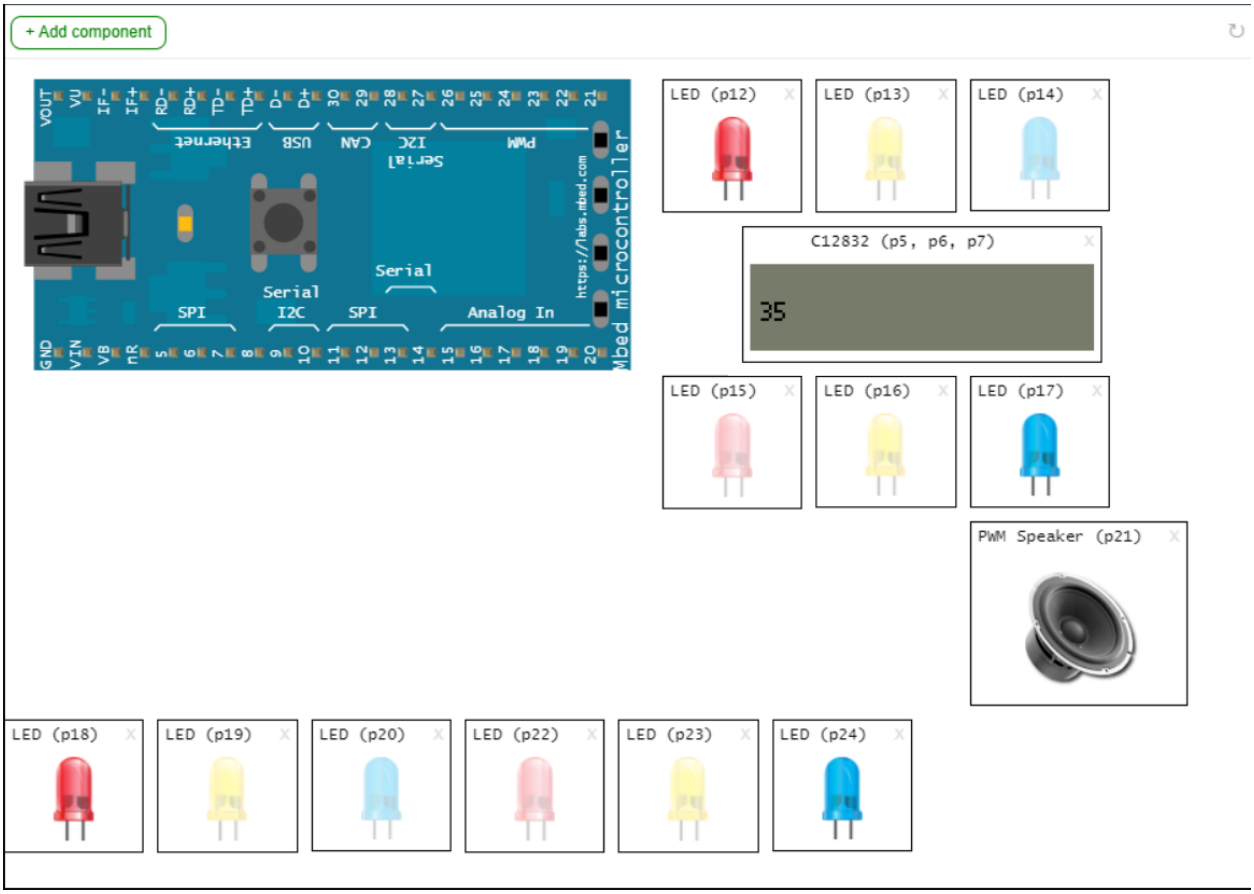
```
        t1red = 1;
        t1yellow = 0;
        t1green = 0;
    }else if(a == 'Y'){
        t1red = 0;
        t1yellow = 1;
        t1green = 0;
    }else if(a == 'G'){
        t1red = 0;
        t1yellow = 0;
        t1green = 1;
    }else{}
}

void t2light(char a){
    if (a == 'R'){
        t2red = 1;
        t2yellow = 0;
        t2green = 0;
    }else if(a == 'Y'){
        t2red = 0;
        t2yellow = 1;
        t2green = 0;
    }else if(a == 'G'){
        t2red = 0;
        t2yellow = 0;
        t2green = 1;
    }else{}
}

void t3light(char a){
    if (a == 'R'){
        t3red = 1;
        t3yellow = 0;
        t3green = 0;
    }else if(a == 'Y'){
        t3red = 0;
        t3yellow = 1;
        t3green = 0;
    }else if(a == 'G'){
```

```
        t3red = 0;
        t3yellow = 0;
        t3green = 1;
    }else{}
}

void t4light(char a){
    if (a == 'R'){
        t4red = 1;
        t4yellow = 0;
        t4green = 0;
    }else if(a == 'Y'){
        t4red = 0;
        t4yellow = 1;
        t4green = 0;
    }else if(a == 'G'){
        t4red = 0;
        t4yellow = 0;
        t4green = 1;
    }else{}
}


void traffic_default (char t1, char t2, char t3, char t4, int time){
    t1light(t1);
    t2light(t2);
    t3light(t3);
    t4light(t4);
    wait(time);
    }
void normal_traffic(){
    traffic_default('G','R','G','R',47);
    traffic_default('Y','R','Y','R',3);
    traffic_default('R','G','R','G',47);
    traffic_default('R','Y','R','Y',3);
    }

int main() {
    while(1){
    normal_traffic();
```

```
        lcd.cls();

        for(int i=47;i>=0;i--)
        {
        lcd.locate(3, 13);
        lcd.printf("%d",i);
            wait(1);
        if((i<=5)&&(i>=0))
        {play_tone(200.0, 0.5, 1, 0);
        play_tone(150.0, 0.5, 1, 0);
        play_tone(125.0, 0.5, 1, 2);}
        else{};
        }
wait(3);

    }
}
```

SIMULATION OUTPUT:





SIMULATION EXPLANATION:

The project is designed to imitate a 4-way traffic signal, which is an idea of how to simulate a traffic signal system at a 4-way junction.

Consider a 4-way junction with traffic flow showing a loop of traffic light operations. The project is also implemented in the same manner. In this junction, firstly, Lane 1 gets its Green light turned. Hence, in all the other lanes apart from the opposite lane to lane 1 i.e. lane 3, their corresponding Red lights are turned on. After a delay of predefined time, in this case we have considered 47 seconds, after which the orange light in these two lanes must be turned on for three seconds, to provide a warning to the pedestrians to stop crossing the road and also a buzzer sounds when the timer reaches 5 seconds to alert visually-challenged pedestrians via sound.

After the three seconds are up, the yellow lights in lane 1 and 3 turn to red and the lights in lanes 2 and 4 turn green, again having the same timing system that the lights in lane 1 and 3 had.

The system then loops back to Lane 1 where the process mentioned above will be repeated all over again.

ADVANTAGES OF INTERRUPTS IN THE TRAFFIC LIGHT SYSTEM:

- It increases the efficiency of System.
- It decreases the waiting time of CPU.
- Stops the wastage of instruction cycle.

DISADVANTAGES:

- CPU has to do a lot of work to handle interrupts, resume its previous execution of programs.
- Overhead required to handle the interrupt request.

ADVANTAGES OF TIMERS IN THE TRAFFIC LIGHT SYSTEM:

- Maintains the timing of an operation in sync with the system clock or an external clock.
- Time ticks for synchronizing operations.
- For sampling of the Data.
- A normal delay function might be used for loop iterating for a few 1000 cycles. But these types of delays might not be accurate. So, Timer is a software designed to count the time interval between events and has proven to be more efficient and accurate.

DISADVANTAGES:

- Can be modulated for frequency, but only in a narrow range.
- It is not very accurate with temperature changes. It has been designed to reduce changes from the power supply voltage, but it still has an effect on timing accuracy.

Limitations

- The project is not suitable for actual implementation but just a demonstration of the process behind the system.

- Real time traffic light controller systems are generally run time programmable i.e. the operator (usually a policeman) can change the timings of each lane as per the intensity of the traffic in each lane.
- There will also be a provision for either manual operation or pre-programmed operation.

APPLICATIONS

- Optimum control of fluctuating traffic volumes such as over saturated or unusual load conditions.
- Improve the vehicular throughput and maximizes the traffic flow rate.
- Decrease delays of vehicles and environmental effects.
- Increase intersection capacity at same time.
- Potential reduction of accidents, conflicts and ensuring safety.
- The timer that has been implemented in the system can also be used for pulse generation and as an Oscillator.
- We can control Servo Motors, Sensors, Multiplexers and De-multiplexers and control the signal transmission at any voltage level using a few add-ons.
- Another typical use is to generate periodic interrupts by dividing the output of a crystal oscillator and having an interrupt handler count the interrupts in order for a processor to keep time.
- The interrupts are often used by the OS's task scheduler to reschedule the priorities of running processes.
- Other interrupts exist to transfer data bytes using UARTs or Ethernet; sense key-presses.

CONCLUSION

For this project, it is intended for the design and implementation of four-way traffic light control system. The design is accomplished by taking the tasks of searching, brainstorming, creating, and reviewing our design to producing a functioning

product. The design was preceded by getting all necessary components to initialize and maintain the proper functions of the desired design code. The control circuit was then designed and faults were corrected before the final project was presented. It was quite challenging to accomplish the perfection required to achieve an accurate output or result due to some software bug issues. Proper functioning of the circuit and an error free connection was safeguarded. The circuit is finally put to test and automatic control of the traffic light is achieved by the ARM mbed code we created.

Some of the improvements that we suggest for the system are detection of traffic volume by adding more logic into the coding and sensors; Emergency vehicle detection such as ambulance, police etc. by using wireless sensor network at the signal intersection; system can give more time to an intersection approach that is experiencing heavy traffic, or shorten or even skip a phase that has little or no traffic waiting for a green light; or traffic signals are activated to detect the approach of a train near a rail crossing.

REFERENCES

1] https://binaryupdates.com/timers-in-lpc2148-arm7-microcontroller/

[2] https://openlabpro.com/guide/timers-8051/

[3] https://www.sciencedirect.com/topics/computer-science/software-interrupt#:~:text=A%20trap%2C%20also%20known%20as,explicitly%20generates%20an%20exception%20condition.&text=The%20ARM%20provides%20the%20SWI,be%20read%20by%20the%20handler.

[4] Venkata Naga Rohit Guntur Electronics and communication engineering department, Anna University, Chennai. Microcontroller Based Automatic Led System International Journal of Advancements in Research & Technology, Volume 2, Issue4, April-2013 194 ISSN 2278-7763 Copyright © 2013 SciRes Pub.

[5] https://www.tutorialspoint.com/embedded_systems/es_timer_counter.htm