**REPORT**

**TRAFFIC LIGHT CONTROL USING LPC1768**

**1. Introduction**

Traffic signals are an essential part of road infrastructure that help regulate flow of vehicle, reduce congestion, and ensure road safety. For controlling traffic lights efficiently, microcontrollers are used widely, providing precise timing and control mechanisms for switching between signals (red, orange, and green). In this project, we implemented a traffic signal controller for 4 directions using the LPC1768 microcontroller and its timers. Ensuring that each direction gets the appropriate amount of time for green, orange, and red lights, allowing for an organized flow of traffic at intersections.

**2. Objective**

The main objective of this project is to implement a traffic signal system using the LPC1768 microcontroller. This project focuses on using timers to control the duration of red, orange, and green lights for each direction efficiently.

* **Precise Timing Control**

Accurate light timing is achieved using the internal timers of the LPC1768 microcontroller. The use of match registers allows for fine-grained control over the time intervals, ensuring reliable and consistent light switching.

* **Low-Cost Solution**

The project is implemented using the LPC1768, a low-cost yet powerful microcontroller, along with simple components such as LEDs. Using timers and GPIOs directly on the microcontroller avoids the need for additional external hardware like delay circuits or expensive controllers.

* **Flexibility and Programmability**

The system is highly programmable, allowing for easy customization of the timing intervals for different traffic scenarios. The flexibility of using timers means you can easily adjust the delay timings, switch sequence, or add new functionality without changing the hardware.

* **Scalability**

The project can easily be scaled to accommodate more lanes, traffic directions, or other types of signal control by simply adding more timers or match registers. By adjusting the timer configurations, this system can be adapted to manage complex traffic systems, including those at major intersections.

**3. Methodology**

**3.1 Hardware Components**

* **LPC1768 microcontroller:** The core component that controls the traffic signal system, utilizing its internal timers.
* **LEDs:** Represent the traffic signals (red, orange, and green lights).
* **Power Supply:** To power the microcontroller and LEDs.

**3.2 Software Tools**

* **Keil uVision:** For writing and compiling the C code.
* **Flash Magic:** For programming the LPC1768 microcontroller.
* **C language:** Used for programming the microcontroller.

**3.3 Steps Involved**

**GPIO Pin Configuration:**

The GPIO pins of the LPC1768 are configured to control LEDs that represent traffic lights.

12 LEDs are used (3 for each direction: red, orange and green).

**Timer Configuration:**

Two timers (Timer 0 and Timer 1) are used to control the light duration for the four directions. Each timer has 4 match registers (MR0, MR1, MR2, and MR3), which define when the timer will trigger events for switching lights.

**Match Registers:**

When the timer counter reaches the value in a match register, the corresponding light changes state, and the next light in the sequence is turned on.

**Interrupts:**

When a match condition is met (i.e., the timer counter matches the value in a match register), an interrupt is triggered.

The interrupt service routine (ISR) handles switching between the lights and resetting the timer for the next cycle.

**Delays:**

Delays are introduced between transitions to simulate real-time traffic light operation (e.g., after switching to red, the next light is delayed by a specific interval).

**Delay=(1/PCLK​)×(PR+1)×Timer Count**

**4. Working Principle**

The system uses the LPC1768's internal timers to create a cycle for switching the traffic lights in four directions. Each direction has three LEDs representing the red, orange, and green lights, which are controlled using GPIO pins.

**Timer Control:**

The timers are configured to generate match events at predefined intervals (based on the values loaded into the match registers). The prescaler in the timer divides the clock to provide practical timing intervals.

**Traffic Light Timing:**

For each direction, the red light is on for a certain duration (e.g., 20 seconds), followed by the orange light for a brief period (e.g., 5 seconds), and then the green light while the other directions are active.

The timers ensure that each light stays on for the appropriate time before switching to the next.

**Synchronization:**

The system ensures synchronization between all four directions by using two timers and carefully coordinating the match register values.

**Cycle Reset:**

Once all the match events for a cycle are complete (all directions have cycled through red, orange, and green), the timers reset, and the cycle starts again.

**5. Component Details**

**LPC1768 Microcontroller:**

The core of the system, running the code and controlling the timers, interrupts, and GPIO pins for light switching.

**LEDs:**

Twelve LEDs are used to represent the red, orange, and green lights in four directions.

**Power Supply:**

A 5V power supply is used to power the microcontroller and the LEDs.

**6. Code**

#include<lpc17xx.h> // Header file

#define r1 (1<<19) // defining pins with names

#define o1 (1<<14)

#define g1 (1<<20)

#define r2 (1<<21)

#define o2 (1<<15)

#define g2 (1<<22)

#define r3 (1<<23)

#define o3 (1<<16)

#define g3 (1<<24)

#define r4 (1<<25)

#define o4 (1<<17)

#define g4 (1<<26)

void delay(unsigned int); //declaration of delay function

int main()

{

LPC\_GPIO1->FIODIR |= (0xff<<19) | (0x0f<<14); // configuring 12 pins as output pins

LPC\_GPIO1->FIOCLR = (0xff<<19) | (0X0f<<14); // clearing 12 pins

//Timers Configuration

LPC\_SC->PCONP |= (0x03<<1); // Enabling Power configuration

LPC\_SC->PCLKSEL0 |= (1<<2) | (1<<4); // Sets bits 2 and 4 for clock of Timers 0 and 1.

// TIMER 0

LPC\_TIM0->PR = 3; // Value in Prescalar Register

LPC\_TIM0->MR0 = 3000000; // MR0 Interrupt value

LPC\_TIM0->MR1 = 6000000; // MR1 Interrupt value

LPC\_TIM0->MR2 = 7000000; // MR2 Interrupt value

LPC\_TIM0->MR3 = 10000000; // MR3 Interrupt value

LPC\_TIM0->MCR = (1<<0)|(1<<3)|(1<<6)|(1<<9); // Match Control Register to generate interrupts at all four MRs

LPC\_TIM0->TCR = (1<<1); // Timer Control Register to reset the counter

LPC\_TIM0->TCR = (1<<0); // TCR for start the counter

//TIMER 1

LPC\_TIM1->PR = 3;

LPC\_TIM1->MR0 = 11000000;

LPC\_TIM1->MR1 = 14000000;

LPC\_TIM1->MR2 = 15000000;

LPC\_TIM1->MR3 = 18000000;

LPC\_TIM1->MCR = (1<<0)|(1<<3)|(1<<6)|(1<<9)|(1<<10);

LPC\_TIM1->TCR = (1<<1);

LPC\_TIM1->TCR = (1<<0);

while(1)

{

if(LPC\_TIM0->IR & (1<<0)) // Checking for Interrupt at MR0

{

LPC\_GPIO1->FIOSET = g1 | r2 | r3 | r4;

LPC\_GPIO1->FIOCLR = o1 | g4;

delay(200);

LPC\_TIM0->IR = (1<<0); // Writing 1 to the 0th bit clears the pin. So it clears MR0 Interrupt

}

if(LPC\_TIM0->IR & (1<<1)) // Checking for Interrupt at MR1

{

LPC\_GPIO1->FIOSET = o2 | r3 | r4 | g1;

LPC\_GPIO1->FIOCLR = r2;

delay(200);

LPC\_TIM0->IR = (1<<1); // Writing 1 to the 1st bit clears the pin. So it clears MR1 Interrupt

}

if(LPC\_TIM0->IR & (1<<2)) // Checking for Interrupt at MR2

{

LPC\_GPIO1->FIOSET = g2 | r1 | r3 | r4;

LPC\_GPIO1->FIOCLR = o2 | g1;

delay(200);

LPC\_TIM0->IR = (1<<2); // Writing 1 to the 2nd bit clears the pin. So it clears MR2 Interrupt

}

if(LPC\_TIM0->IR & (1<<3)) // Checking for Interrupt at MR3

{

LPC\_GPIO1->FIOSET = o3 | r1 | r4 | g2;

LPC\_GPIO1->FIOCLR = r3;

delay(200);

LPC\_TIM0->IR = (1<<3); // Writing 1 to the 3rd bit clears the pin. So it clears MR3 Interrupt

}

if(LPC\_TIM1->IR & (1<<0))

{

LPC\_GPIO1->FIOSET = g3 | r1 | r2 | r4;

LPC\_GPIO1->FIOCLR = o3 | g2;

delay(200);

LPC\_TIM1->IR = (1<<0);

}

if(LPC\_TIM1->IR & (1<<1))

{

LPC\_GPIO1->FIOSET = o4 | r1 | r2 | g3;

LPC\_GPIO1->FIOCLR = r4;

delay(200);

LPC\_TIM1->IR = (1<<1);

}

if(LPC\_TIM1->IR & (1<<2))

{

LPC\_GPIO1->FIOSET = g4 | r1 | r2 | r3;

LPC\_GPIO1->FIOCLR = o4 | g3;

delay(200);

LPC\_TIM1->IR = (1<<2);

}

if(LPC\_TIM1->IR & (1<<3))

{

LPC\_GPIO1->FIOSET = o1 | r2 | r3 | g4;

LPC\_GPIO1->FIOCLR = r1;

delay(200);

LPC\_TIM1->IR = (1<<3);

LPC\_TIM0->TC = 0;

}

}

}

void delay(unsigned int k) // Delay function.

{

unsigned int i,j;

for(i=0;i<k;i++)

{

for(j=0;j<3000;j++);

}

}

**6. Result**

The system successfully controls the traffic lights for all four directions using timers through match registers in the LPC1768 microcontroller. Each direction is provided with adequate green, orange, and red-light intervals, ensuring that the traffic flows smoothly. The timers operate in a continuous loop, switching the lights at regular intervals.

The system runs efficiently, and the use of timers ensures precise timing for each light, leading to a smooth traffic control system.