LAB NO : 11 DATE : 26/03/2025

Title : PROGRAM ON PULSE WIDTH MODULATION (PWM)

# Solved Exercise 1: WAP to vary the intensity of an LED using PWM.

## Code :

#include <LPC17xx.h>

void initPWM(void) {

LPC\_PINCON->PINSEL3 |= 0x8000; //Select PWM1.4 output for Pin1.23, function 2

LPC\_PWM1->PCR = 0x1000; //enable PWM1.4, by default it is single Edged

LPC\_PWM1->PR = 0;

LPC\_PWM1->MR0 = 30000; //period=10ms if pclk=cclk/4

LPC\_PWM1->MCR = (1<<1); //Reset PWM TC on PWM1MR0 match

LPC\_PWM1->LER = 0xff; //update values in MR0 and MR1

LPC\_PWM1->TCR = 0x00000002; //RESET COUNTER TC and PC

LPC\_PWM1->TCR = 0x00000009; //enable TC and PC

}

void updatePulseWidth(unsigned int pulseWidth){

LPC\_PWM1->MR4 = pulseWidth; //Update MR4 with new value

LPC\_PWM1->LER = 0xff; //Load the MR4 new value at start of next cycle

}

void delayMS(unsigned int milliseconds) //Using Timer0

{

LPC\_TIM0->CTCR = 0x0; //Timer mode

LPC\_TIM0->PR = 2; //Increment TC at every 3 pclk

LPC\_TIM0->TCR = 0x02; //Reset Timer

LPC\_TIM0->TCR = 0x01; //Enable timer

while(LPC\_TIM0->TC < milliseconds); //wait until timer counter reaches the desired delay

LPC\_TIM0->TCR = 0x00; //Disable timer

}

int main(void){

int pulseWidths[] = {0, 3000, 6000, 9000, 12000, 15000, 18000, 21000, 24000, 27000}; //Pulse Widths for varying LED Brightness

const int numPulseWidths = 10;

int count=1;

int dir=0; //direction, 0 = Increasing, 1 = Decreasing

initPWM(); //Initialize PWM

while(1) {

updatePulseWidth(pulseWidths[count]); //Update LED Pulse Width

delayMS(100000);

if(count == (numPulseWidths-1) || count == 0)

dir = !dir; //Toggle direction if we have reached count limit

if(dir) count--;

else count++;

}

}

LAB NO : 12 DATE : 26/03/2025

Title : PROGRAM ON STEPPER MOTOR

# Solved Exercise 1: To rotate the stepper motor in clockwise and anticlockwise direction at a particular speed continuously.

## Code :

#include <LPC17xx.H>

void clock\_wise(void);

void anti\_clock\_wise(void);

unsigned long int var1,var2;

unsigned int i=0,j=0,k=0;

int main(void) {

SystemInit();

SystemCoreClockUpdate();

LPC\_PINCON->PINSEL0 = 0xFFFF00FF; //P0.4 to P0.7 GPIO

LPC\_GPIO0->FIODIR = 0x000000F0; //P0.4 to P0.7 output

while(1){

for(j=0;j<50;j++)

clock\_wise();

for(k=0;k<65000;k++);// Delay to show anti\_clock Rotation

for(j=0;j<50;j++)

anti\_clock\_wise();

for(k=0;k<65000;k++); // Delay to show clock Rotation

} // End of while(1)

} // End of main

void clock\_wise(void) {

var1 = 0x00000008; //For Clockwise

for(i=0;i<=3;i++) // for A B C D Stepping

{

var1 = var1<<1; //For Clockwise

LPC\_GPIO0->FIOPIN = var1;

for(k=0;k<3000;k++); //for step speed variation

}

}

void anti\_clock\_wise(void)

{

var1 = 0x00000100; //For Anticlockwise

for(i=0;i<=3;i++) // for A B C D Stepping

{

var1 = var1>>1; //For Anticlockwise

LPC\_GPIO0->FIOPIN = var1;

for(k=0;k<3000;k++); //for step speed variation

}

}

# Exercise 1: Write a C program to rotate the stepper motor in the clockwise direction when SW2 is high and anticlockwise direction when SW2 is low.

## Code :

#include <LPC17xx.H>

void clock\_wise(void);

void anti\_clock\_wise(void);

unsigned long int var1,var2;

unsigned int i=0,j=0,k=0;

unsigned int switchState=1;

int main(void) {

SystemInit();

SystemCoreClockUpdate();

LPC\_PINCON->PINSEL0 = 0xFFFF00FF; //P0.4 to P0.7 GPIO

LPC\_GPIO0->FIODIR = 0x000000F0; //P0.4 to P0.7 output

LPC\_PINCON->PINSEL4 &= 0xFCFFFFFF; //For Key

LPC\_GPIO2->FIODIR &= ~(1<<12);//0x00001000;

while(1){

switchState = (LPC\_GPIO2->FIOPIN >> 12) & 1;

if (switchState == 0)

clock\_wise();

else

anti\_clock\_wise();

for(k=0;k<1300;k++); // Delay to show clock Rotation

} // End of while(1)

} // End of main

void clock\_wise(void) {

var1 = 0x00000008; //For Clockwise

for(i=0;i<=3;i++) // for A B C D Stepping

{

var1 = var1<<1; //For Clockwise

LPC\_GPIO0->FIOPIN = var1;

for(k=0;k<3000;k++); //for step speed variation

}

}

void anti\_clock\_wise(void)

{

var1 = 0x00000100; //For Anticlockwise

for(i=0;i<=3;i++) // for A B C D Stepping

{

var1 = var1>>1; //For Anticlockwise

LPC\_GPIO0->FIOPIN = var1;

for(k=0;k<3000;k++); //for step speed variation

}

}