----------------------------------------------------------------------------------------------------------------------------------------------------------

**NAME:** Darshangouda Patil

**PROJECT NAME**: DC Motor Speed Control using PWM

**ABSTRACT:**

Pulse Width Modulation, or **PWM**, is a technique for getting analog results with digital **means**.This will help to increasing and reducing the speed of electronic components like DC motor.

**INTRODUCTION:**

This project help to control the speed of dc motor using PWM technique. User can set the duty cycle for each motor at any time using Interactive method that is UART communication channel.

**APPLICATION’S OF PWM:**

* PWM Techniques are used in Telecommunications for encoding purposes.
* Pulse Width Modulation helps in voltage regulation and thus finds its use in controlling Brightness in Smart Lighting Systems and also controls the speed of motors.
* Computer Motherboard requires PWM Signals that controls the heat generated in the board. 4 Pin PWM header is embedded in the fan that helps to dissipate the heat from the motherboard.
* It is also used in Audio/Video Amplifiers.

**ADVANTAGES OF PWM:**

* PWM technique helps in preventing overheating of LED’s while maintaining its brightness.
* Pulse Width Modulation provides accuracy and quick response time.
* It provides high input Power Factor.
* Initial cost is low.
* PWM technique helps the motors to generate maximum torque even when they are running at lower speeds.

**DISADVANTAGES OF PWM:**

* As the PWM frequency is high, switching losses is considerably high.
* It induces Radio Frequency Interference (RFI).

**LANGUAGES ,TOOLS AND TECHNOLOGY USED**

**Embedded C:**

Embedded C is most popular programming language in software field for developing electronic gadgets. Each processor used in electronic system is associated with embedded software.

Embedded C programming plays a key role in performing specific function by the processor. In day-to-day life we used many electronic devices such as mobile phone, washing machine, digital camera, etc. These all device working is based on microcontroller that are programmed by embedded C.

**Keil IDE:**

Keil MicroVision is a free software which solves many of the pain points for an embedded program developer. This software is an integrated development environment (IDE), which integrated a text editor to write programs, a compiler and it will convert your source code to hex files too.

**Proteus**:

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

**ARM DEVELOPMENT BOARD (LPC2148):**

Arm development boards are the ideal platform for accelerating the development and reducing the risk of new SoC designs. The combination of ASIC and FPGA technology in Arm boards delivers an optimal solution in terms of speed, accuracy, flexibility and cost.

**FALSH MAGIC:**

Flash Magic is a PC burner tool for programming flash memory based microcontroller using serial or Ethernet protocol built by NXP. This tool helps the developer to easily burn the hex file generated by the embedded software like Keil µvision for 8051 and ARM microcontrollers or MPLAB for PIC microcontrollers.

**DC MOTOR:**

A DC motor is any of a class of rotary electrical motors that converts direct current electrical energy into mechanical energy.

**Oscilloscopes**

Oscilloscopes (or scopes) test and display voltage signals as waveforms, visual representations of the variation of voltage over time. The signals are plotted on a graph, which shows how the signal changes. The vertical (Y) access represents the voltage measurement and the horizontal (X) axis represents time.

**WORKING MODULE:**

**VIRTUAL TERMINAL**

**ARM BOARD**

**DC motor**

**DC MOTOR DRIVER**

**BREAD BOARD**

**FIGURE: Block diagram of project module**

**WORKING:**

Connection as given as shown in below Fig 1 figure. Connect development board through RS232 cable dump hex code on board with help of FLASH magic. After that without removing RS232 cable Select virtual terminal in Flash magic software set baud rate to 9600 (as we set in programm) in flash magic. By default some duty cycle will be set through programme now u can change that on virtual terminal.

**SOURCE CODE :**

///LPC1768 PWM1 ,PWM2 PWM3 and UART0 used

#include<lpc17xx.h>

#include<stdio.h>

int UART\_RX(void);

void DISPLAY(char str[],char);

int flag=0,m1\_speed=25,m2\_speed=50,m3\_speed=75;

char check[]={"0123456789"};

#define SBIT\_CNTEN 0

#define SBIT\_PWMEN 2

#define SBIT\_PWMMR0R 1

#define SBIT\_LEN0 0

#define SBIT\_LEN1 1

#define SBIT\_LEN2 2

#define SBIT\_LEN3 3

#define SBIT\_LEN4 4

#define SBIT\_PWMENA1 9

#define SBIT\_PWMENA2 10

#define SBIT\_PWMENA3 11

#define SBIT\_PWMENA4 12

#define PWM\_1 0 //P2\_0 (0-1 Bits of PINSEL4)

#define PWM\_2 2 //P2\_1 (2-3 Bits of PINSEL4)

#define PWM\_3 4 //P2\_2 (4-5 Bits of PINSEL4)

#define PWM\_4 6 //P2\_3 (6-7 Bits of PINSEL4)

void PLL()

{

LPC\_SC->SCS=(1<<5);

while ((LPC\_SC->SCS& (1<<6))==0);

LPC\_SC->CLKSRCSEL=(1<<0);

LPC\_SC->PLL0CON=(1<<0);

LPC\_SC->PLL0CFG=(14<<0);

LPC\_SC->PLL0FEED=0XAA;

LPC\_SC->PLL0FEED=0X55 ;

LPC\_SC->CCLKCFG=(5<<0);

while ((LPC\_SC->PLL0STAT&(1<<26))==0);

LPC\_SC->PLL0CON|=(1<<1);

LPC\_SC->PLL0FEED=0XAA;

LPC\_SC->PLL0FEED=0X55 ;

}

void UART\_INIT()

{

LPC\_PINCON->PINSEL0=(1<<4)|(1<<6);

LPC\_UART0->LCR=(0X03<<0)|(1<<7);

LPC\_UART0->DLL=97;

LPC\_UART0->DLM=0;

LPC\_UART0->LCR &= ~(1<<7);

}

void PWM\_BASIC()

{

LPC\_PINCON->PINSEL4 = (1<<PWM\_1) | (1<<PWM\_2) | (1<<PWM\_3);

LPC\_PWM1->TCR = (1<<SBIT\_CNTEN) | (1<<SBIT\_PWMEN);

LPC\_PWM1->PR = 0x0;

LPC\_PWM1->MCR = (1<<SBIT\_PWMMR0R);

LPC\_PWM1->MR0 = 100; // set PWM cycle(Ton+Toff)=100)

LPC\_PWM1->MR1 = m1\_speed; // Set 50% Duty Cycle for all four channels

LPC\_PWM1->MR2 = m2\_speed;

LPC\_PWM1->MR3 = m3\_speed;

LPC\_PWM1->LER = (1<<SBIT\_LEN0) | (1<<SBIT\_LEN1) | (1<<SBIT\_LEN2) | (1<<SBIT\_LEN3);

LPC\_PWM1->PCR = (1<<SBIT\_PWMENA1) | (1<<SBIT\_PWMENA2) | (1<<SBIT\_PWMENA3);

}

void PWM\_CALL(int op)

{

switch(op)

{

case 1:LPC\_PWM1->MR1 = m1\_speed; LPC\_PWM1->LER = (1<<SBIT\_LEN1);break;

case 2:LPC\_PWM1->MR2 = m2\_speed; LPC\_PWM1->LER = (1<<SBIT\_LEN2);break;

case 3:LPC\_PWM1->MR3 = m3\_speed; LPC\_PWM1->LER = (1<<SBIT\_LEN3);break;

}

}

void MENU()

{ char str[4];

int flag,j,num,\*motor\_speed;

char motor\_text[]="SET Mn SPEED =";

DISPLAY("SET MOTOR SPEED(press enter to skip)",1);

DISPLAY("MOTOR SPEED M1=",0);

sprintf(str,"%d",m1\_speed);

DISPLAY(str,0);

DISPLAY(" M2=",0);

sprintf(str,"%d",m2\_speed);

DISPLAY(str,0);

DISPLAY(" M3=",0);

sprintf(str,"%d",m3\_speed);

DISPLAY(str,1);

while(1)

{

for(j=1;j<=3;j++)

{

if(j==1){ motor\_speed=&m1\_speed; motor\_text[5]='1'; };

if(j==2){ motor\_speed=&m2\_speed; motor\_text[5]='2'; };

if(j==3){ motor\_speed=&m3\_speed; motor\_text[5]='3'; };

flag=1;

do{

DISPLAY(motor\_text,0);

num=UART\_RX();

if(num==-2)

{

DISPLAY("PREVIOUS VALUE SET ->",0);

sprintf(str,"%d",\*motor\_speed);

DISPLAY(str,1);

flag=0;

}

else if(num<=99 && num>=0)

{

\*motor\_speed=num;

PWM\_CALL(j);

flag=0;

}

else

{

DISPLAY("ENTER VALID NUMBER (0 to 99)",1);

}

}while(flag);

}//for end

}

}

int GETNUM(char str[4],int len)

{

int i,num=0;

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

num=(num\*10)+(str[i]-'0');

return num;

}

int UART\_RX()

{

unsigned char temp;

int i=0,j,flag=0;

char str[4]={"000"};

do

{

while((LPC\_UART0->LSR&(1<<0))!=1);

temp=LPC\_UART0->RBR;

if(temp==13 && i==0)return -2;

if(temp==13)return GETNUM(str,i);

flag=0;

for(j=0;j<10;j++)

if(temp==check[j]) flag++;

if(flag!=1) return -1;

str[i++]=temp;

if(i==3)return -1;

}while(1);

}

void UART\_TX(unsigned char var)

{

while((LPC\_UART0->LSR&(1<<5)>>5)!=0);

LPC\_UART0->THR=var;

}

void DISPLAY(char str[],char newline)

{

int i;

for(i=0;str[i]!='\0';i++)

{

UART\_TX(str[i]);

}

if(newline==1)

UART\_TX(13);

}

main()

{

PLL();

UART\_INIT();

PWM\_BASIC();

PWM\_CALL(1);

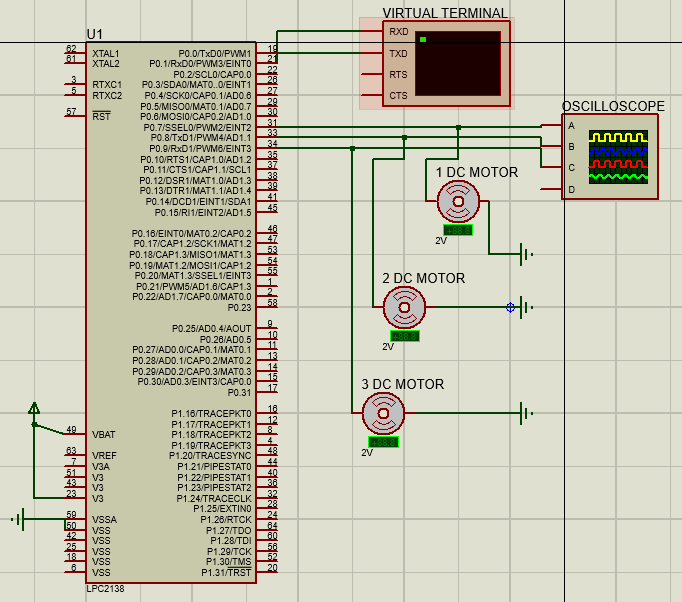
PWM\_CALL(2);

PWM\_CALL(3);

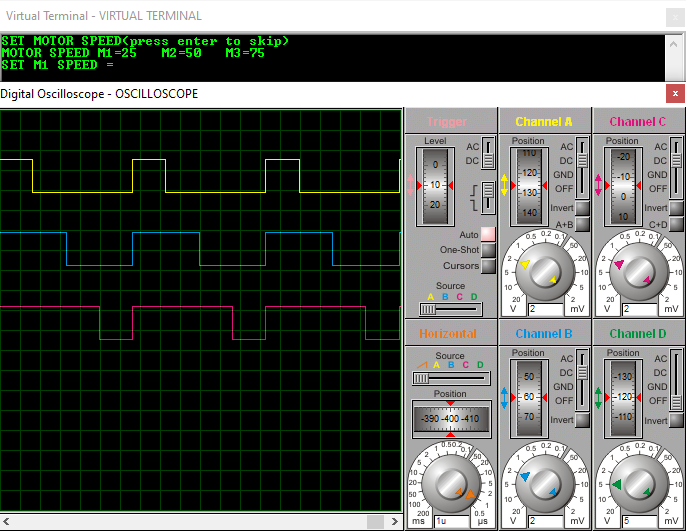
MENU();

}

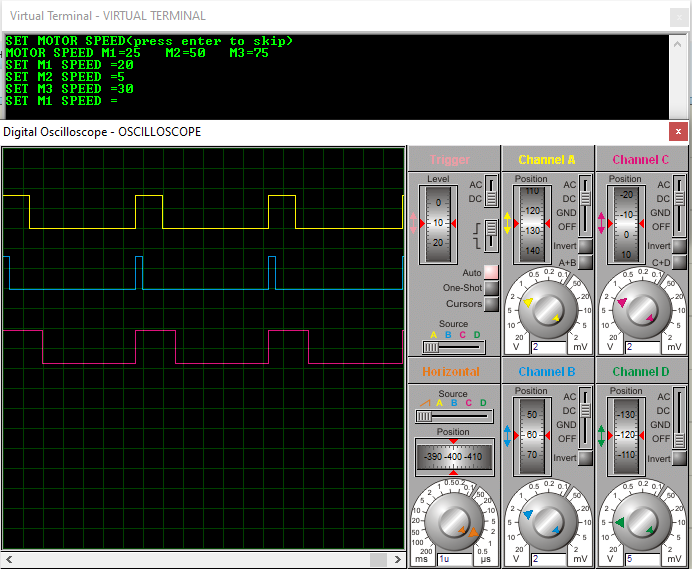
**OUTPUT:**

****

**Fig1:Scematic of Project**

****

**Fig 2: Duty cycle for 1 motor=25, 2 motor=50, 3 motor=75**

****

**Fig 3: Duty cycle for 1 motor=20, 2 motor=5, 3 motor=30**

**CONCLUSION:**

DC motor’s speed can be increase and decreased at any time with the help of PWM technique. User can set duty cycle from 0 to 99 at any time through virtual terminal that is UART (RS232) communication.

We can use PWM technique in various application as we discussed earlier.