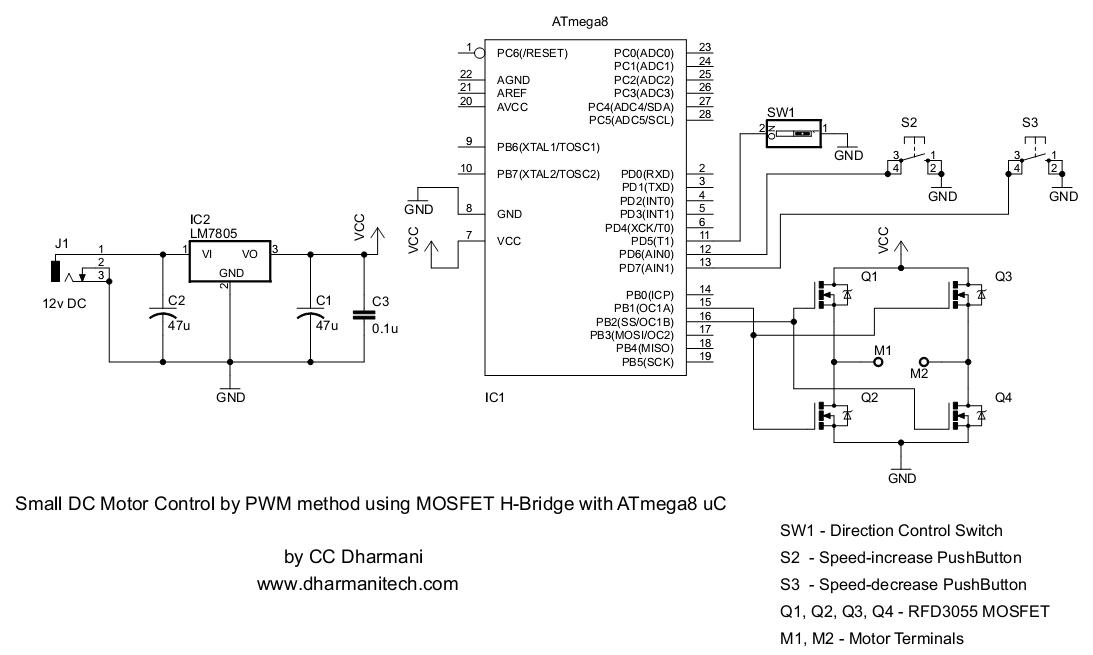
**Small Dc Motor Control by PWM Method Using MOSFET H-Bridge with Atmega8 Microcontroller**

BY CC Dharmani, [www.dharmanitech.com](http://www.dharmanitech.com/)

Hi Friends,

Here I’m discussing DC motor control using the PWM counters of AVR ATmega8 microcontroller. I had used a DC motor from an old personal stereo cassette player. The circuit provides speed and direction control of the motor. The PWM waveforms are used for driving the MOSFET H-bridge as shown in the schematic:



At a time only one of the two PWM channel is active, driving only two MOSFETS (either Q1-Q4 or Q3-Q2). Other two MOSFETs remain OFF. Whenever the Direction Control switch is toggled, the PWM channel is also changed, driving the alternative pair of MOSFET, which changes the direction of current flow through motor, resulting in the direction change in rotation of motor shaft.

**Code:**

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// \*\*\*\*\*\*\*\*\*\*\* PWM DC MOTOR CONTROL \*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Controller  : ATmega8 (1MHz internal Crystal)

//Compiler : ICCAVR

//Author   : CC Dharmani, Chennai(India)

//Date : Nov 2008

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <iom8v.h>

#include <macros.h>

#define increaseButton\_PRESSED !(PIND & 0x40)

#define increaseButton\_OPEN (PIND & 0x40)

#define decreaseButton\_PRESSED !(PIND & 0x80)

#define decreaseButton\_OPEN (PIND & 0x80)

#define DIRECTION\_FORWARD  !(PIND & 0x20)

#define DIRECTION\_REVERSE   (PIND & 0x20)

#define STOP\_MOTOR TCCR1B = 0x00; TCCR1A = 0x00

#define START\_MOTOR TCCR1B = 0x09

#define set\_FORWARD TCCR1A = 0x81

#define set\_REVERSE TCCR1A = 0x21

//defining macros for setting minimum and maximum PWM counter values

//and step-size for controlling the voltage applied to MOSFETs base

#define COUNTER\_LOWER\_LIMIT 0x0090

#define COUNTER\_UPPER\_LIMIT 0x00f8

#define STEP\_SIZE 0x0008

**void** port\_init(**void**)

{

PORTB = 0x00;

DDRB = 0x06; //PWM pins OC1A & OC1B defined as outputs

PORTC = 0x00;

DDRC = 0x00;

PORTD = 0xE0; //internal pull-up enabled for three pins connected to switches

DDRD = 0x00;

}

//TIMER1 initialize - prescale:1

//PWM Frequency: 1KHz

**void** timer1\_init(**void**)

{

TCCR1B = 0x00; //stop

TCNT1H = 0xFC; //setup

TCNT1L = 0x18;

OCR1A = COUNTER\_LOWER\_LIMIT;

OCR1B = COUNTER\_LOWER\_LIMIT;

ICR1H = 0x03;

ICR1L = 0xE8;

TCCR1A = 0x81; //set forward; OC1A connected, OC1B disconnected

TCCR1B = 0x09; //start Timer

}

//call this routine to initialize all peripherals

**void** init\_devices(**void**)

{

//stop errant interrupts until set up

CLI(); //disable all interrupts

port\_init();

timer1\_init();

MCUCR = 0x00;

GICR = 0x00;

TIMSK = 0x00; //timer interrupt sources

SEI(); //re-enable interrupts

//all peripherals are now initialized

}

//\*\*\*\*\* FUNCTION FOR SOFTWARE DELAY OF 1 mSEC (appx.) \*\*\*\*\*\*\*

**void** delay\_ms(**int** miliSec) //for 1 Mhz crystal

{

**int** i,j;

**for**(i=0;i<miliSec;i++)

**for**(j=0;j<100;j++)

{

asm(*"nop"*);

asm(*"nop"*);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**void** main(**void**)

{

**unsigned** **int** counter = COUNTER\_LOWER\_LIMIT;

**unsigned** **char** dir = 0, dir1 = 0;

init\_devices();

**while**(1)

{

CHECK\_PB:

**while**(increaseButton\_OPEN && decreaseButton\_OPEN)

{ //loop here until any push-button is pressed

**if**(DIRECTION\_FORWARD) //check for Direction control switch status

dir = 0;

**else**

dir = 1;

**if**(dir != dir1) //chenge direction if switch position has changed

{

STOP\_MOTOR;

delay\_ms(500);

**if**(dir == 0)

set\_FORWARD;

**else**

set\_REVERSE;

START\_MOTOR;

dir1 = dir;

}

}

**if**(increaseButton\_PRESSED) //Speed-increase push-button is pressed

{

delay\_ms(20); //key debouncing delay after key-pressed

**if**(increaseButton\_OPEN) **goto** CHECK\_PB;

**while**(increaseButton\_PRESSED); //wait here till the push-button is kept pressed

delay\_ms(20); //key debouncing delay after key released

**if**(counter >= COUNTER\_UPPER\_LIMIT) //if speed is already maximum, no action

counter = COUNTER\_UPPER\_LIMIT;

**else**

counter += STEP\_SIZE; //increase speed by a fixed step

OCR1A = counter;

OCR1B = counter;

}

**else** //speed-decrease push-button is pressed

{

delay\_ms(20); //key debouncing delay after key-pressed

**if**(decreaseButton\_OPEN) **goto** CHECK\_PB;

**while**(decreaseButton\_PRESSED); //wait here till the push-button is kept pressed

delay\_ms(20); //key debouncing delay after key released

**if**(counter <= COUNTER\_LOWER\_LIMIT) //if speed is already minimum, no action

counter = COUNTER\_LOWER\_LIMIT;

**else**

counter -= STEP\_SIZE; //reduce speed by a fixed step

OCR1A = counter;

OCR1B = counter;

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\* CC Dharmani, [www.dharmanitech.com](http://www.dharmanitech.com/) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*