<u>EDC Project Report</u>

<u>H Bridge DC Motor control with interface for coasting and braking</u>

Authors: Anupam Adhikari (14BEE0143), Karthik Bhardwaj (14BEE0070)

The objective of the project is to design a module using power electronics and microcontroller to control a DC Motor in four quadrant operation. The module should also contain interfaces using button to coast or brake the DC Motor.

Definitions

Coasting: Naturally stopping a Motor by disconnecting its power supply.

Braking: Forcefully stopping a Motor by short circuiting its power supply. Short-circuit creates a closed path for current to flow thus forcing the machine to act in generating mode and come to stop faster than Coasting.

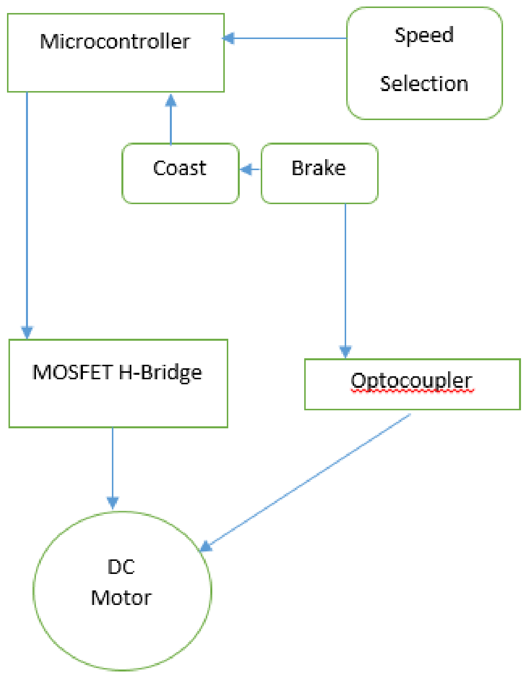
PWM: Pulse Width Modulation. One of the very popular methods to control power electronics. The power electronic device is rapidly switch on and off in order to create a variable average DC Voltage level. The switching frequency is kept high enough that the switching is not distinguishable to human eyes. The project uses variable time PWM.

Interrupt: A priority signal that forces the microcontroller to execute the routine for the said interrupt. Used for interrupting the drive for Coasting and Braking operation.

Opto-coupler: A power electronic device to seprate two different voltage level. Used to separate the MOSFET drive from the microcontroller interface to prevent any damage.

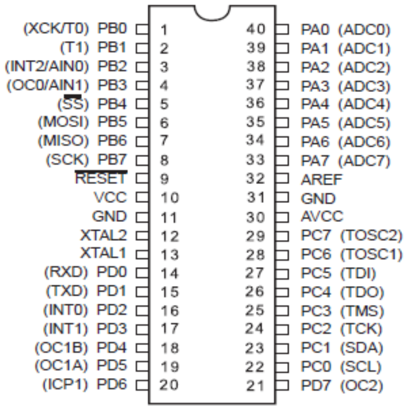
MOSFET: A power electronic voltage controlled switch.

Block Diagram

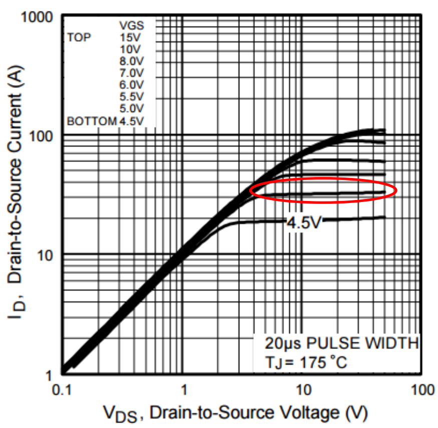


Description of components

The microcontroller acts as the central processing unit for the drive and needs to be powered separately by 5V DC. ATMega32A is one the popular and powerful microcontroller available in the market. Manufactured by Atmel Corporation the microcontroller has 4 GPIO Ports, 8 channel 10-bit ADC with noise cancellation, It has a 32KB ROM which is sufficient enough to store program of decent projects. It has 4 PWM channels that also function as two 8-bit and one 16-bit timer/counter. It has interface for the commonly used communication protocol like USART, I2C, SPI. PD2, PD3 and PB2 are dedicated to handle external interrupts and is used to interrupt the program for coasting and braking.



IRF540N is a power MOSFET that can handle Vds upto 100V and Ids upto 23A at 100C before breaking down. The MOSFET is used in an H bridge (fully controlled controller). The MOSFET is operated as a switch in the drive by alternately driving the MOSFET to saturation and cut off. Every high pulse from the PWM pin of microcontroller turns on the MOSFET while every low pulse turns it off. This turns on and off the motor simultaneously to attain speed control.



Region of operation

Fig 2. Typical Output Characteristics

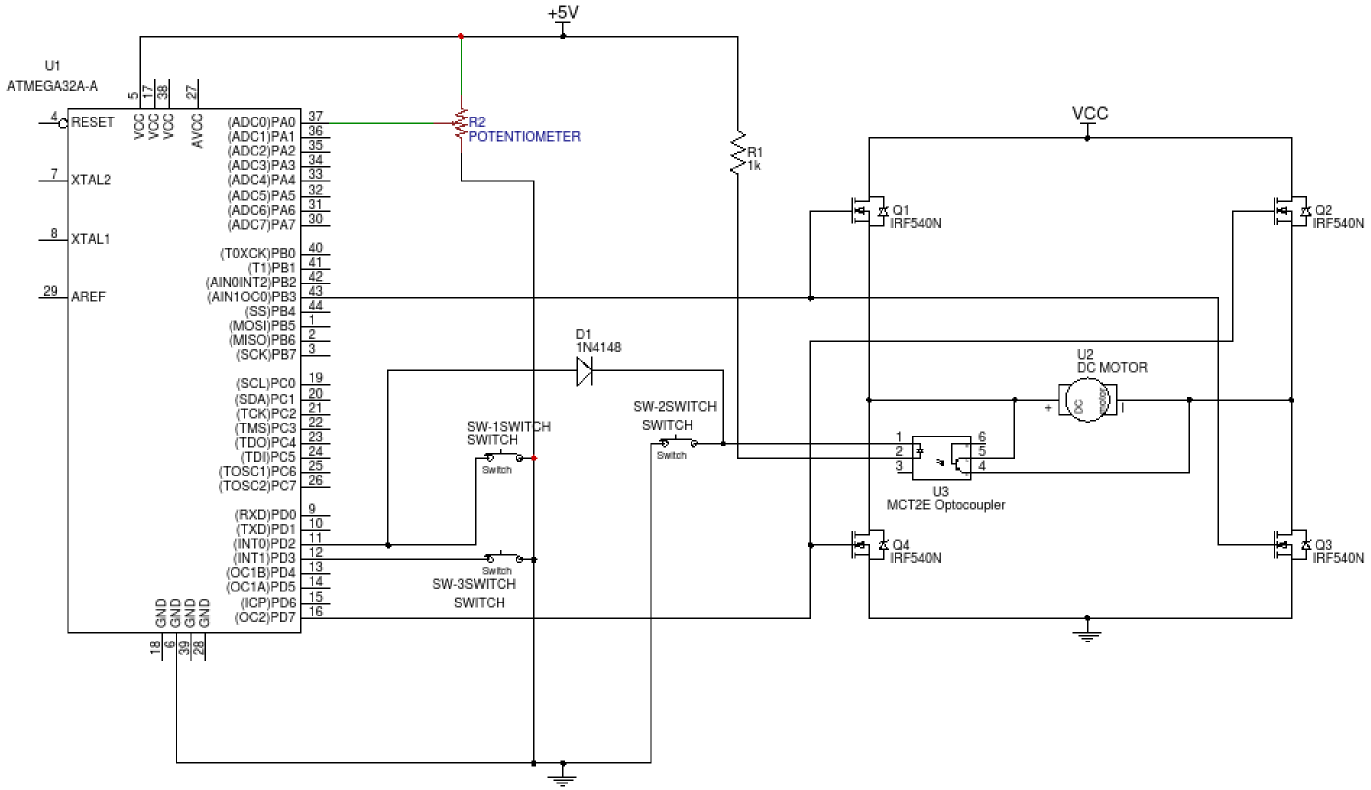
MCT2E is an optocoupler used to seprate two different voltage levels. It can handle upto 70V in the output side. The optocoupler's photo illumination and sensing to transfer pulses from input to output side. A high pulse on the input side light up a miniature IR/normal LED. This is sensed by the photo-transistor and conducts current from collector to emitter side. MCT2E is used to brake the motor by shorting its terminal when the input circuit of MCT2E is completed. MOSFET can also be used for this purpose however unlike MOSFET Optocoupler have a high temperature resistance and doesn't heat up easily.

To visualise the operation the load has been replaced to a DC PC cooling fan. It has a 1.68W BLDC motor that works on 12V DC and runs on 3000RPM while drawing 0.14A. It has a 3 pin connector; 2 power pins for power input and one pin for speed sensing which is floating in the current project.

Algorithm

For normal operation

1. Read ADC input of the speed selection potentiometer.
2. Convert the ten bit result into eight bit result by multiplying with 0.25.
3. Send the eight bit result to set_duty routine
4. If mode is forward motoring write OC0=the eight bit result and OC2=0 else write OC2= the eight bit result and OC0=0.
5. Go to step 1



TITLE: H Bridge Control of DC Motor	REV: 1.0
Date: 2017-05-05	Sheet: 1/1
EasyEDA V4.4.4	Drawn By: Anupam Adhikari

For ISR

For Coasting

1. Interrupt Service is called
2. Set OC0=0 and OC2=0
3. RTI

For Braking

1. Coasting Interrupt is triggered
2. The optocoupler shorts the motor terminals.

For Reversal

1. Negative edge is detected at INT1
2. If mode==1 mode=0 else mode=1; mode1= Forward Motoring, mode2= Reverse Motoring

Operating Instructions

Avoid using at higher voltage for longer period of time unless heat sinks are attached to MOSFETs

Before breaking ensure that the MOSFETs are switched off to prevent short-circuit and heating of MOSFET eventually leading to damage (This is taken care by the circuit).

Do not block the rotor of the fan or touch during operation.

The drain to source voltage of the MOSFET should be above 3V to prevent the MOSFET from going into active/amplification region. Resulting in a nonlinear speed control.

Working of H Bridge

Under normal operation the microcontroller controls Q1, Q2, Q3 and Q4 using PWM. When under forward motoring Q1 and Q3 are switched on while Q2 and Q4 are switched off (duty=0%). Rapid on and off of the MOSFET via PWM gives us a variable average voltage. Changing the voltage across the terminals of DC Motor controls its speed as

$$V = Eb - Ia \times Ra$$

$$N = \frac{60\ A\ Eb}{\phi\ P\ Z}$$

When the motor is to be operated in reverse motoring mode Q2 and Q4 are turned on via PWM and Q1 and Q3 are turned off (duty=0%).

For coasting operation all the four MOSFETs are turned off i.e. (OC2 and OC0 =0). Therefore there is no power supply to the motor and it comes to halt naturally because of drag and rotational friction.

For braking the motor is first coasted i.e. power supply is disconnected. This is done to prevent short circuit from VCC to GND which may lead to electrical fires or unnecessary heating of power electronic devices. The opto-isolator then shorts the terminal of the machine and allows short circuit current to flow. The result is a shorter halt time.

Vt\Mode	Coasting	Braking
5V DC	3.2s	2.5s
12V DC	4.3s	3.3s

The time recorded is the average time of 5 observation using smartphone's stopwatch.

Embedded C Code

```c
/*
 * EDC_project.c
 * Program to perform four quadrant operation of DC Motor
 * Created: 18-03-2017 07:49:08 PM
 * Author : Anupam
 */

#include <avr/io.h>
#include <avr/interrupt.h>

/*
Speed control at ADC0
Coasting operation interrupt at INT0
Reversing interrupt/pin at INT1
Forward Motoring PWM at OC0
Reverse Motoring PWM at OC2
*/
int mode=1;

void set_duty(uint8_t duty)
{
		if(mode=1) {OCR0=duty;OCR2=0;}
		else { OCR2=duty;OCR0=0;PORTB|=1<<PINB0;}

}

void init_ADC()
{
	ADMUX|=1<<REFS0|1<<ADLAR;
	ADCSRA|=1<<ADEN|1<<ADPS2|1<<ADATE;
}

void init_PWM()
{
	TCCR0|=1<<WGM00|1<<WGM01|1<<COM01|1<<CS02;
	OCR0=0;
	TCCR2|=1<<WGM21|1<<WGM20|1<<COM21|1<<CS22;
	OCR2=0;
}

void init_SYS()
{
	DDRD|=1<<PIND7;
	DDRB|=1<<PINB3;
	MCUCR|=1<<ISC11;
	GICR|=1<<INT0|1<<INT1;
	ADCSRA|=1<<ADSC;
	sei();


}

int main()
{
	DDRB|=1<<PINB0;
	DDRD|=1<<PIND3;
	PORTD|=1<<PIND3;
	init_ADC();
	init_PWM();
	init_SYS();
```

```
        while(1);
        return 0;
}
ISR(INT0_vect)
{
        OCR0=0;
        OCR2=0;
}

ISR(INT1_vect)
{

        if(mode==0) mode=1;
        else {mode=0; }
}

ISR(ADC_vect)
{
        set_duty(ADCH<<2|ADCL>>6);
        ADCSRA|=1<<ADSC;
}
```

Result