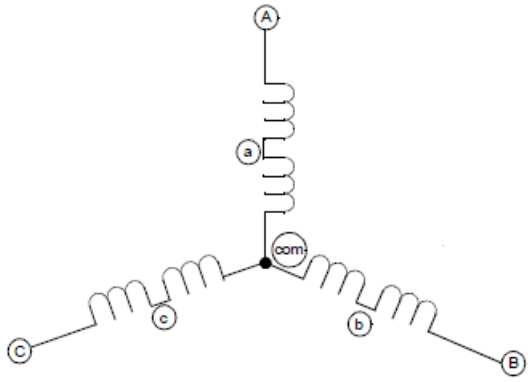
December 31, 2017

Simple Projects

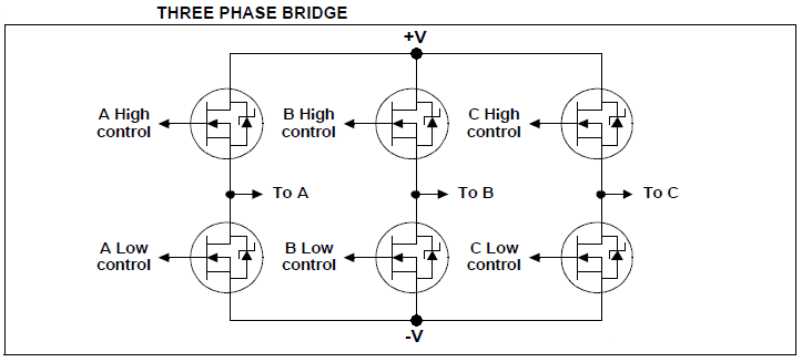
Sensored brushless DC motor control with Arduino

There are two types of BLDC motors: sensored and sensorless. Sensored BLDC motor uses hall effect sensors to detect rotor position where as the sensorless BLDC motor uses another technique which is BEMF (back electromotive force). This topic shows how to drive a sensored BLDC motor using an Arduino UNO board. The BLDC motor used in this project is just a PC CD-ROM motor (spindle motor).

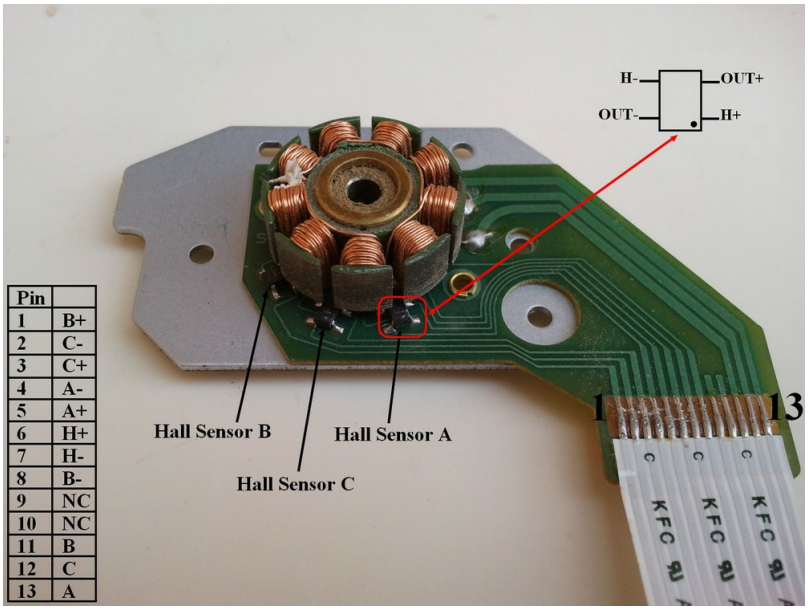
The BLDC motor (sensored or sensorless) is a 3 phase DC motor which means it has 3 winding on the stator core. Two coils are energized at a time to create a rotating electric field. This method is fairly easy to implement, but to prevent the permanent magnet rotor from getting locked with the stator, the excitation on the stator must be sequenced in a specific manner while knowing the exact position of the rotor magnets.



To drive this motor we need a 3-phase bridge, the basic elements of it are the 6 MOSFETs. General circuit schematic diagram of the 3-phase bridge is shown below:



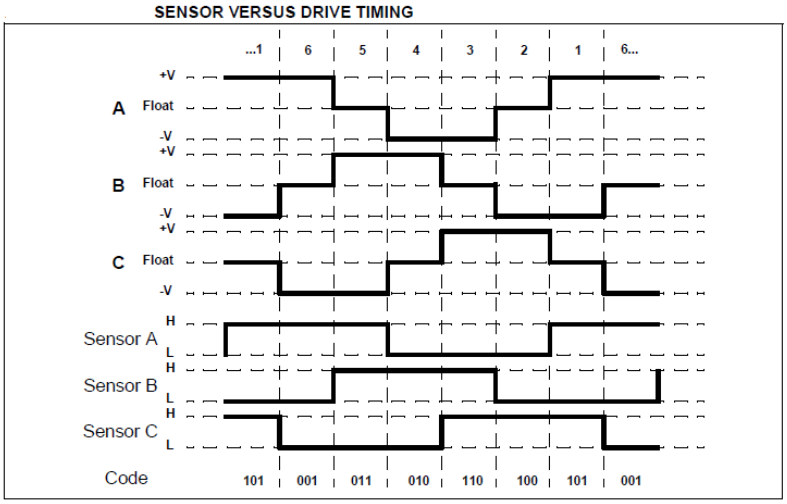
The sensored BLDC motor has 3 hall effect sensors (A, B and C) to sense rotor position, these sensors are placed as shown in the following picture. The motor which I used in this project has pinout as shown below (other motors may have a different pinout).
In this motor each hall effect sensor has 4 pins: VCC (H+), GND (H-) and two outputs (some sensors come with 3 pins: VCC, GND and output).



Pin	
1	B+
2	C-
3	C+
4	A-
5	A+
6	H+
7	H-
8	B-
9	NC
10	NC
11	B
12	C
13	A

Since I have 4-pin hall effect sensors, I added an analog comparator (I used LM339N quad comparator IC) to each one so each sensor outputs (2 outputs: + and -) are connected to the inputs (2 inputs: non-inverting and inverting) of the comparator as shown in the circuit schematic below, finally I got 3 outputs from the 3 hall effect sensors.

Each sensor outputs a digital high for 180 electrical degrees and outputs a digital low for the other 180 electrical degrees. The following figure shows the relationship between the sensors outputs and the required motor drive voltages for phases A, B and C.



According to the hall effect sensors, the 3-phase bridge is controlled as shown in the following table:

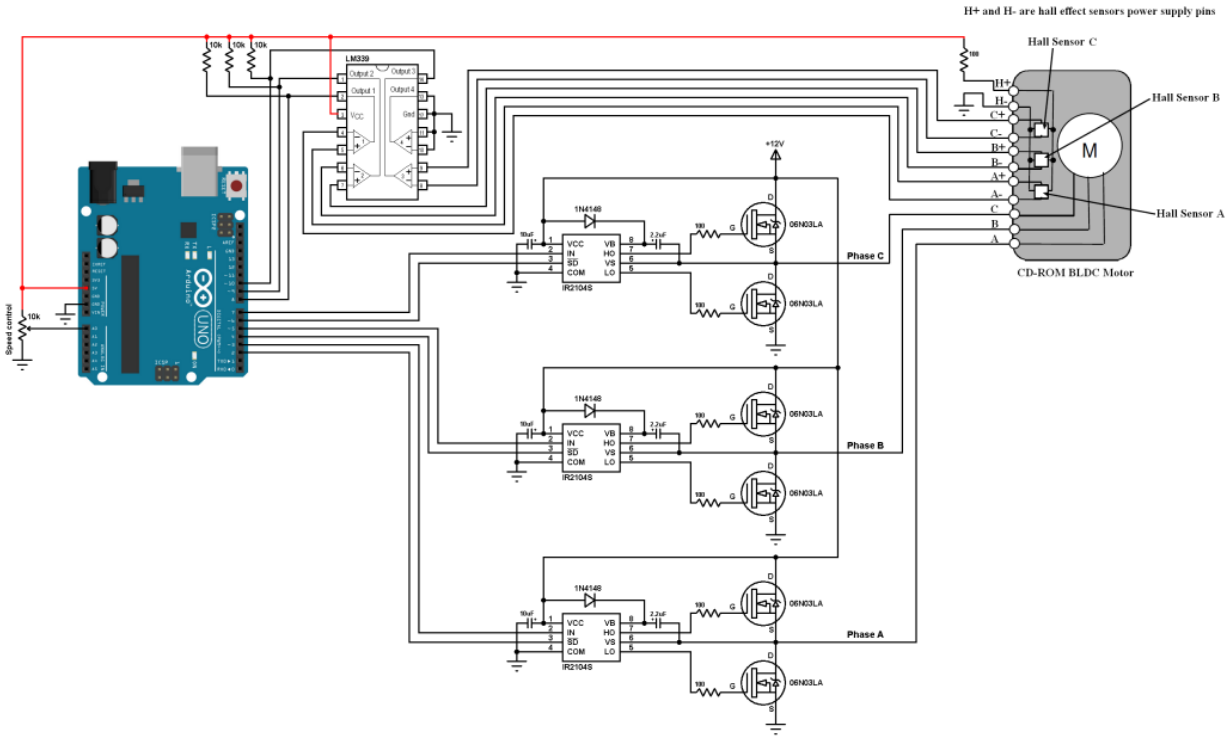
Phase	Sensor C	Sensor B	Sensor A	C High Drive	C Low Drive	B High Drive	B Low Drive	A High Drive	A Low Drive
6	0	0	1	0	1	0	0	1	0
4	0	1	0	0	0	1	0	0	1
5	0	1	1	0	1	1	0	0	0
2	1	0	0	1	0	0	1	0	0
1	1	0	1	0	0	0	1	1	0
3	1	1	0	1	0	0	0	0	1

Components Required:

- Arduino board
- Sensored brushless DC motor
- 6 x 06N03LA N-type mosfet (or equivalent) –**datasheet**
- 3 x IR2104S gate driver IC –**datasheet**
- LM339N (or LM339) quad comparator IC –**datasheet**
- 10k ohm (or less) potentiometer
- 3 x 10k ohm resistor
- 7 x 100 ohm resistor
- 3 x IN4148 diode
- 3 x 10uF capacitor
- 3 x 2.2uF capacitor
- 12V source
- Breadboard
- Jumper wires

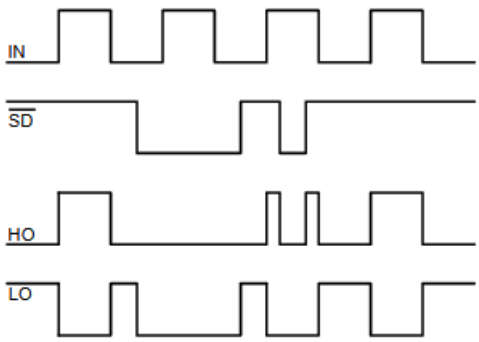
Sensored brushless DC motor control with Arduino circuit:

The overall circuit diagram is shown below.



(All grounded terminals are connected together).

In the circuit there are three IR2104S gate driver IC, each one is used to drive one high side mosfet and one low side mosfet, the switching between the high side and the low side is done according to the control lines which are: IN and SD. The figure below shows input and output timing diagram:



Input/Output Timing Diagram

The 10k potentiometer is used to control the brushless DC motor speed, it is controlled using PWM technique (pwming high sides only). Any time there is one active high side mosfet and one active low side mosfet, that means always there is one active PWM pin (Arduino pin 2, 4 or 6).

The table below summarizes the active Arduino pins according to the hall effect senors states (pins: 8, 9, and 10):

Pin / Phase	10	9	8	7	6	5	4	3	2
6	0	0	1	0	1	0	0	1	PWM
4	0	1	0	0	0	1	PWM	0	1
5	0	1	1	0	1	1	PWM	0	0
2	1	0	0	1	PWM	0	1	0	0
1	1	0	1	0	0	0	1	1	PWM
3	1	1	0	1	PWM	0	0	0	1

Sensored brushless DC motor control with Arduino code:

In this project I implemented a simple software PWM code because I had needed an active PWM signal on pin 2, 4 or 6 (only one is active at a time), for that I used Timer2 module and I configured it with a prescaler of 1/8 which means the PWM signal frequency is about 7.8KHz (equal to: 16MHz/(8 * 256)) and 8-bit resolution.

The ADC module is configured to read from channel 0 only.

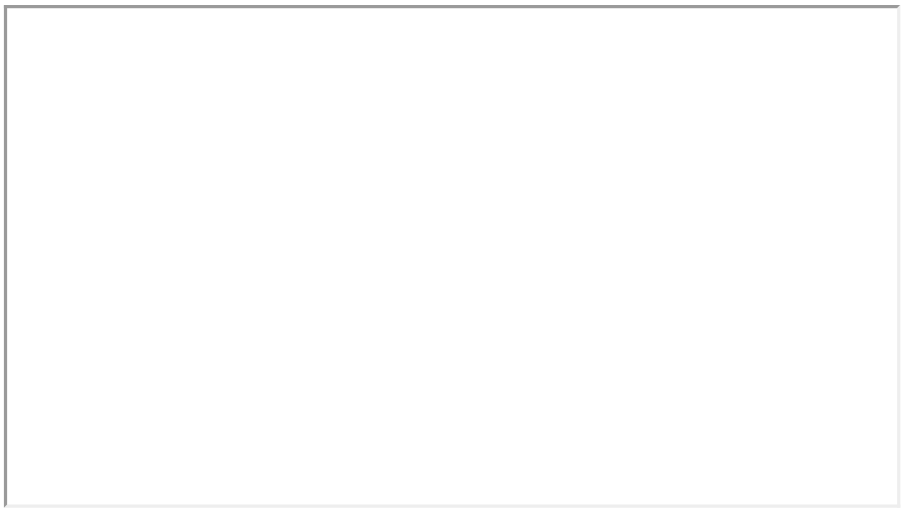
Arduino interrupt on change is activated for pins 8, 9 and 10 (hall effect sensors inputs) for better commutation.

PORTB are Arduino uno pins: 8 ... 13.
PORTD are Arduino uno pins: 0 ... 7.

```
// Arduino sensored BLDC (brushless DC) motor control code

byte motor_speed, bldc_step, pwm_pin = 2;
void setup() {
  DDRD  |= 0xFC;                    // Configure pins 2, 3, 4, 5, 6 and 7 as outputs
  PCICR  = 1;                       // Enable pin change interrupt for pins 8 to 13
  PCMSK0 = 7;                       // Enable pin change interrupt for pins 8, 9 and 1
  // Timer2 configuration
  TCCR2A = 0;
  TCCR2B = 2;                       // Timer2 prescaler = 1/8 (Timer2 clock = 2MHz)
  TCNT2  = 0;                       // Timer2 preload value = 0
  // ADC module configuration
  ADMUX  = 0x60;                    // Configure ADC module and select channel 0
  ADCSRA = 0x84;                    // Enable ADC module with 16 division factor (ADC
  // BLDC motor first move
  bldc_step = PINB & 7;             // Read hall effect sensors status (PINB: read fro
  bldc_move();                      // Move the BLDC motor (first move)
}
ISR (PCINT0_vect){
  bldc_step = PINB & 7;             // Read and save hall effect sensors status (PINB:
  bldc_move();                      // Move the BLDC motor
}
void bldc_move(){                   // BLDC motor move function according to hall effe
  switch(bldc_step){
    case 1:
      PORTD = 0x48;
      pwm_pin = 2;
```

The video below shows a hardware circuit of the project:



Related project:

[CD-ROM Sensored BLDC motor control with Arduino]

Share this:

312

BLDC MOTOR

« Arduino ST7735 1.8″ TFT display example

Sensorless BLDC motor control with Arduino – DIY ESC »

16 comments

ajith says:

August 17, 2018 at 3:25 am

This article is great. I like it very much

But I have a liite question. You say Arduino pins 2, 4 and 6 as PWM capable but I find only pins 3,5,6,9,10,11 are PWM cable.

This is clear from the picture of the UNO in your schematic.

Could you kindly explain me this discrepancy.

Thanks

Reply

Simple Projects says:

August 17, 2018 at 7:30 am

In this project a software PWM is used which can be used with any pin, but it's recommended to use hardware PWM as what was done in this project:

CD-ROM Sensored BLDC motor control with Arduino

Reply

akhalesh yadav says:

February 23, 2019 at 8:32 pm

Actually I am using bldc controlling in order to obtain regenerative braking ; would this program be enough to get regenerative braking voltage

Reply

ajith says:

August 17, 2018 at 7:29 am

Hi, I suppose I understood that you are using software based PWM.

Thanks

Reply

ajith says:

August 17, 2018 at 9:30 am

Thanks very much

Reply

Moose says:

November 26, 2018 at 10:45 am

Hi,

I am trying to make a 5 phase BLDC motor controller and your 3 phase example is by far the best I've seen! Is it possible for me to implement code like this to make a 5 phase controller?

I also want to add that I am quite the noob with arduino and although I understand the read/write parts of the code, I do not understand how to enable and assign the interrupt pins using this port manipulation method and the software PWM section.

I looked around and could not find a good enough tutorial using data port manipulation.

Will you please explain this to me?

Thanks

Reply

Dominique Gyselinck says:

February 13, 2019 at 9:34 pm

How you change direction ?

Reply

Karthik says:

April 2, 2019 at 11:15 am

Can we control the speed using push buttons? please help me with the changes in code to do so.

Reply

Alex says:

April 11, 2019 at 7:09 am

Thank you for your contributions, this very illustrative and schematic project, you learn from your projects, they serve as guides and inspirations.
a question I would like to know how to adapt this project to a hall sensor of three terminals, what change should be made to use the sensor hall three terminals, thank you for your ingenuity and creativity

Reply

william says:

August 7, 2019 at 8:49 am

How to change the program from 180degree bldc timing to 120 degree. because mine is 120 degree

Reply

Sharjeel Ahmed says:

September 20, 2019 at 9:29 pm

https://community.parker.com/technologies/electromechanical-group/w/electromechanical-knowledge-base/174/hall-sensors-60-vs-120degrees

Reply

Sharjeel Ahmed says:

September 20, 2019 at 9:34 pm

@ william Its 120 phase difference between hall sensors. In this post Note that Hall states retain for

180 deg, But the difference between each hall is 120

Reply

Davide says:

March 12, 2020 at 10:21 am

Hi. Thanks for sharing this project. It was a good inspiration.
I noticed that you have the SOIC IR2102 mounted into some sort of socket … What are those sockets and where can I buy them ?

Reply

Simple Projects says:

March 12, 2020 at 12:46 pm

They are IR2104S, soldered in SOP to DIP converter PCBs. The sockets are just pin headers, you can get them from online shopping websites (Aliexpress, ebay …).

Reply

Sebastian Reich says:

May 6, 2020 at 7:32 am

Thank you for this beautiful and efficient jewel of a code.
it was very inspiring for me.

Sebastian

Reply

praveenkumar says:

June 29, 2020 at 11:39 am

pin no 3,5 and 7 whe
re these pins are what was actually dose in the program??

Reply

Leave a Reply

Email (required) (Address never made public)

Name (required)

Website

Save my name, email, and website in this browser for the next time I comment.

Notify me of new comments via email.

Notify me of new posts via email

Post Comment

This site uses Akismet to reduce spam. **Learn how your comment data is processed**.

Facebook Page:

Sponsored Links:

PCBWay

Simple Projects 2020