Today's plan:

- Solution to Activity 4
- Announcements: status report
- Final presentations and reports
- Motor drivers
- Op-Amp continuation
- Powering your project
- Measuring capacitance

Activity 4

Complete the C program below so that it will:
1) configure pin P1.0 as an output and P1.3 as an input.
2) then enter a loop that continuously reads the
P1.3 value. Each time the program sees a
change from Low to High, it should toggle the P1.0 output.

```
#include <msp430.h>
int main(void){
	unsigned char oldval,newval;
	WDTCTL = WDTPW + WDTHOLD;
	P1DIR = 1; // P1.0 output, all others input
	oldval = P1IN;
	while(1){
		newval = P1IN;
		if (((newval & 8) == 8) && ((oldval & 8) == 0) )
			P1OUT ^= 1;
		oldval = newval;
	}
}
```

Activity 4

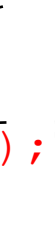
Complete the C program below so that it will:
1) configure pin P1.0 as an output and P1.3 as an input.
2) then enter a loop that continuously reads the P1.3 value. Each time the program sees a change from Low to High, it should toggle the P1.0 output.

```
#include <msp430.h>
int main(void){

        WDTCTL = WDTPW + WDTHOLD;
        P1DIR = 1; // P1.0 output, all others input

        while(1){
                while (P1IN & 8);
                while (! (P1IN & 8));
                P1OUT ^= 1;

        }
}
```



sits here as long as the pin is high

sits here as long as it is low.

so if we get here, we must have just gone from low to high.

Announcements: Final Reports

- At the end of the course you'll present your project in two ways:

1) Oral presentation in class. These will happen on Thursday April 8, Tuesday April 13 and Wednesday April 14. These are reasonably informal. We will all look at the presentation, with slides if any, and the working project.

2) Formal written report, due April 16, 10PM.

Announcements:

Materials return:

You will need to return all the materials borrowed from the lab: (breadboard, launchpad, motors, cars, etc).

This can be done after the presentations and not later than April 16th. You need your deposits back!

Announcements: Status Report

I would like a short written status report from everyone turned in at start of the third project session: Week of March 22-26.

The report should discuss your progress so far: what has been accomplished, what remains to be done. If you have encountered problems, discuss them, and your plans to move forward. If you need help to make progress, please mention it.

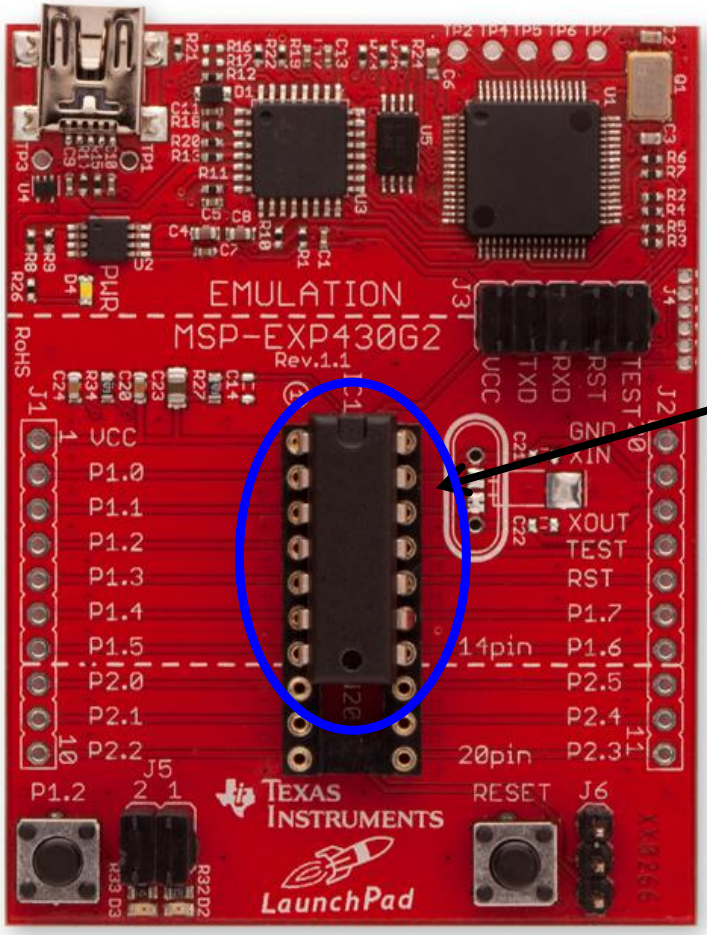
These reports need not be long, just a few sentences is fine.

Serial interfaces

Many sensors use “standard” interfaces such as I^2C (inter-integrated circuit) or SPI (serial peripheral interface) to talk to the microcontroller.

The MSP430 has a module that can ease using these interfaces (Universal Serial Communication Interface, USCI, which can “speak”: I^2C, SPI and UART).

Texas Instruments MSP430 LaunchPad

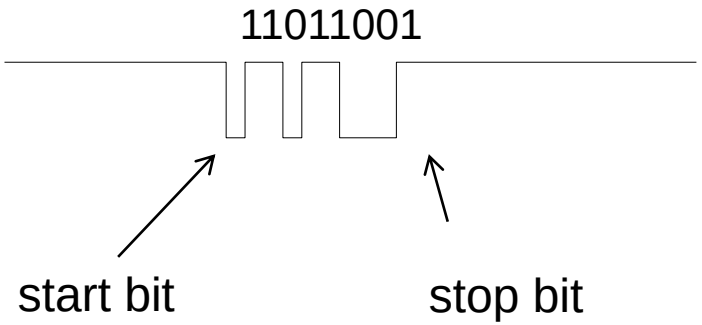


MSP430G2231

Seraial ports are build into n

http://www.ti.com/tool/msp-exp430g2

UART: Universal Asynchronous Receiver/Transmitter

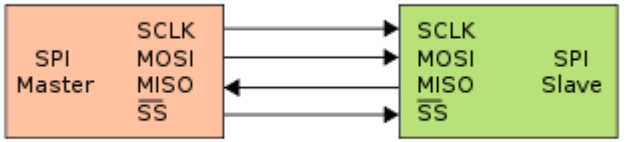


- 3 lines for bi-directional communication: ground, transmit, receive
- start bit is always low, stop bit is always high.
- usually have 8 data bits in between, (but sometimes 5 or 6 or 7)
- least significant bit first, most significant bit last
- sometimes there is parity bit after the data

The USCI can output bytes and decode incoming bytes.

- to transmit a byte, just write it to UCA0TXBUF = byte;
- to receive a byte, set up interrupt to trigger when byte received, then read from receive buffer: byte = UCA0RXBUF;

SPI: Serial Peripheral Interface

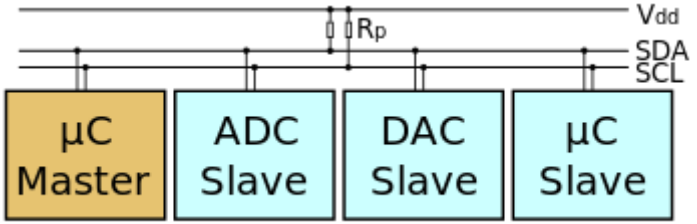


SCLK: clock from master
MOSI: Master out, slave in
MISO: Master in, slave out.
Slave Select

On every toggle of the clock, bits are transmitted in both directions, though not always useful. Communications controlled completely by the master.

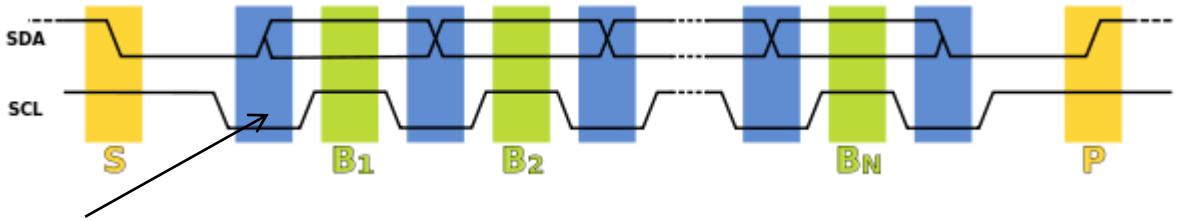
point-to-point, one master, one slave.

I2C: Inter-integrated Circuit



SDA – serial data
SCL – serial clock

Both lines are open-drain, pulled up with pull-up resistors



data line changes when clock is held low.

I2C is a bus: can be multiple masters, multiple slaves on the bus.

I2C: Inter-integrated Circuit info

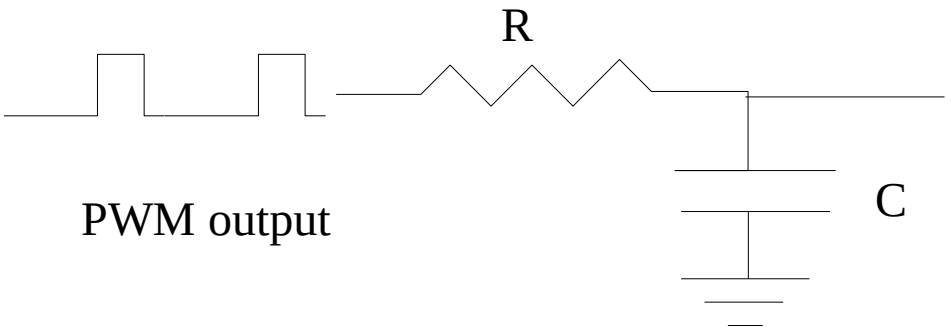
Some posted on our webpage.

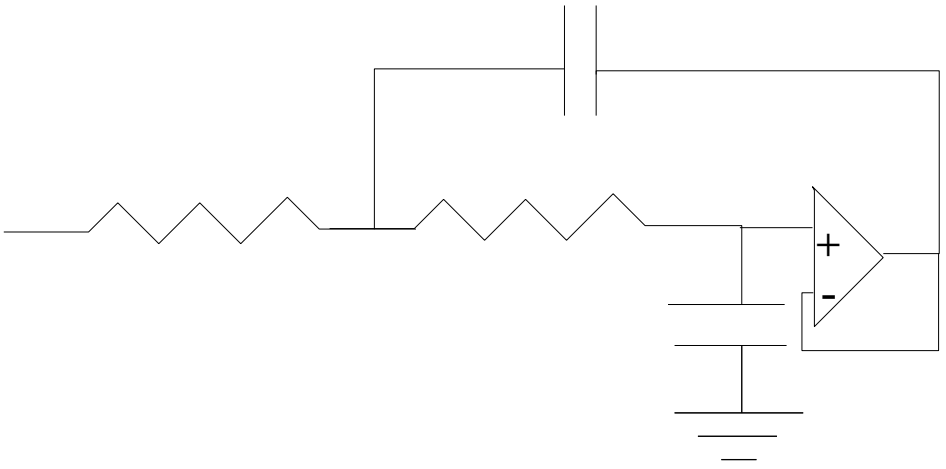
Search internet for descriptions and examples

Analog Output:

The MSP430G2553 doesn't have a DAC. If you want an adjustable analog output, there are a couple of options:

- Add an external DAC (serial or parallel)
- low-pass filter a PWM output.





2nd order Low-pass filter

Analog Output:

Parallel DAC: eg TLC7528, Dual 8bit multiplying digital to analog converter.

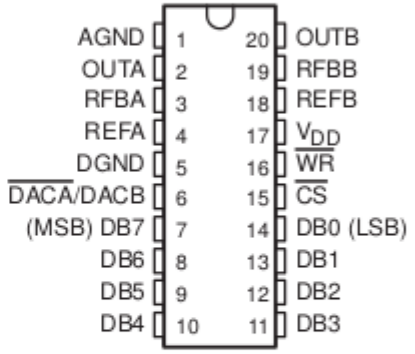
TLC7528C, TLC7528E, TLC7528I
DUAL 8-BIT MULTIPLYING
DIGITAL-TO-ANALOG CONVERTERS

SLAS062E – JANUARY 1987 – REVISED NOVEMBER 2008

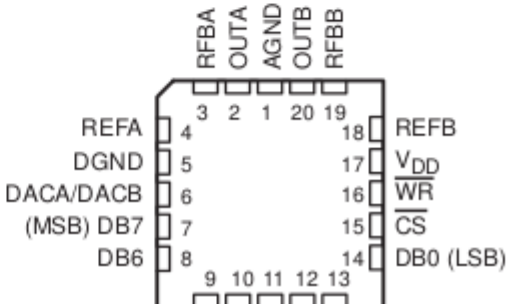
- **Easily Interfaced to Microprocessors**
- **On-Chip Data Latches**
- **Monotonic Over the Entire A/D Conversion Range**
- **Interchangeable With Analog Devices AD7528 and PMI PM-7528**
- **Fast Control Signaling for Digital Signal Processor (DSP) Applications Including Interface With TMS320**
- **Voltage-Mode Operation**
- **CMOS Technology**

KEY PERFORMANCE SPECIFICATIONS	
Resolution	8 bits
Linearity Error	1/2LSB
Power Dissipation at V_{DD} = 5V	20mW
Settling Time at V_{DD} = 5V	100ns
Propagation Delay Time at V_{DD} = 5V	80ns

DW, N OR PW PACKAGE
(TOP VIEW)



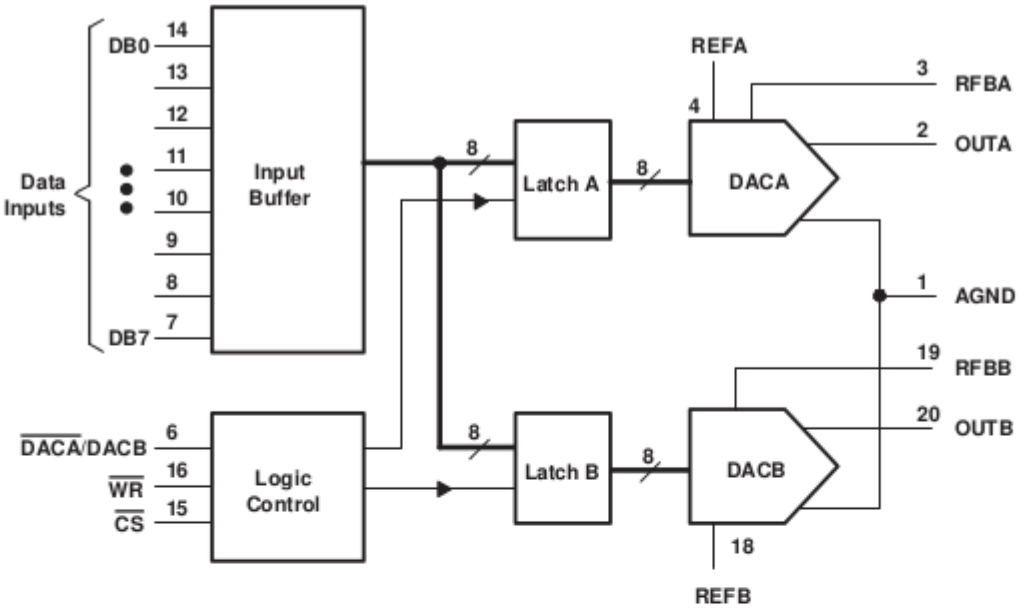
FN PACKAGE
(TOP VIEW)



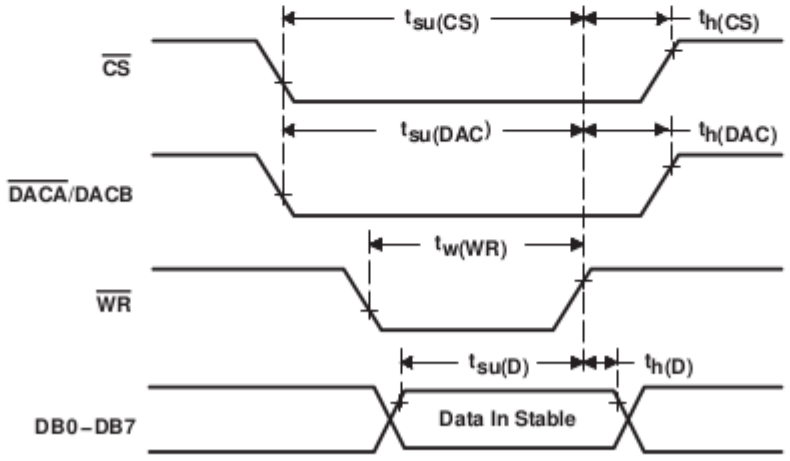
description

The TLC7528C, TLC7528E, and TLC7528I are dual, 8-bit, digital-to-analog converters (DACs) designed with separate on-chip data latches and feature exceptionally close DAC-to-DAC match-

DAC: TLC7528



operating sequence



Controlling things with the microcontroller

MSP430 P1.x maximum output current: +/- 6 mA (x 3.3V = 20mW)

To drive external loads that are more demanding than logic chips, the MSP430

Some possibilities:

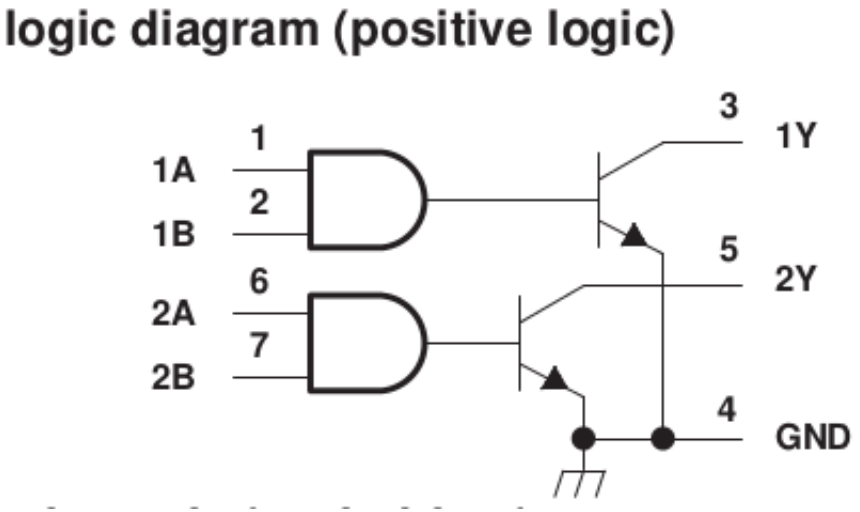
- Beefier logic
- op-amps
- Buffer/driver
- Transistor (bipolar or MOSFET)
- opto-isolators
- Relay
- Solid-state relay
- H Bridge chip (eg for bi-directional motors)

REFERENCE: The Art of Electronics (Horowitz and Hill)

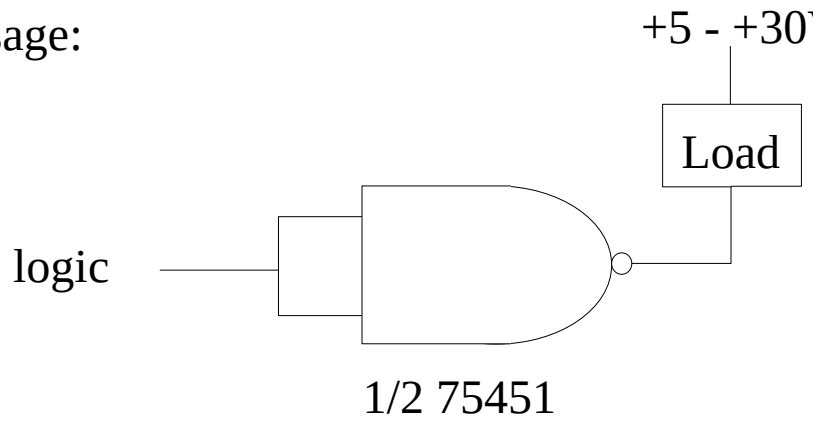
Controlling things with the microcontroller

Driver eg SN75451

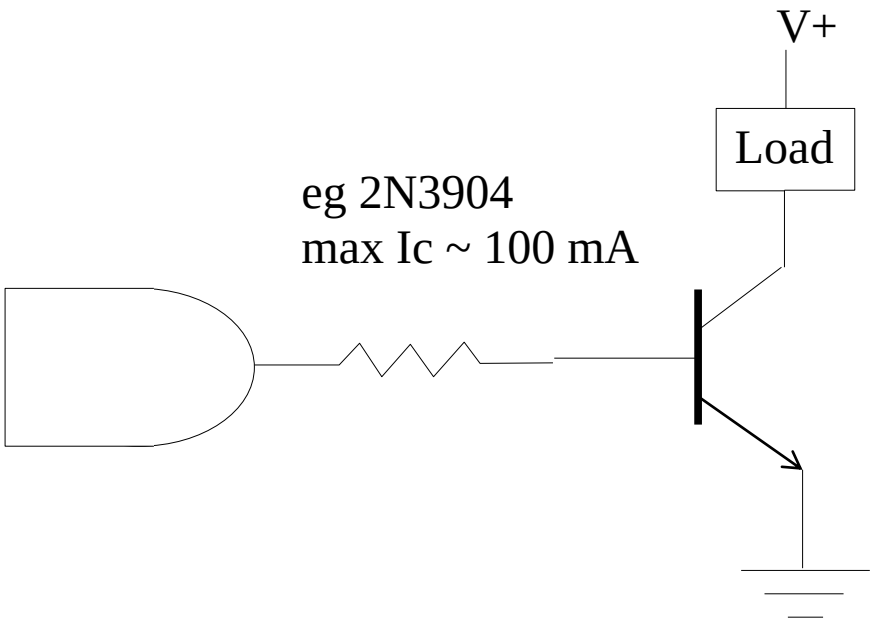
up to 300 mA



usage:

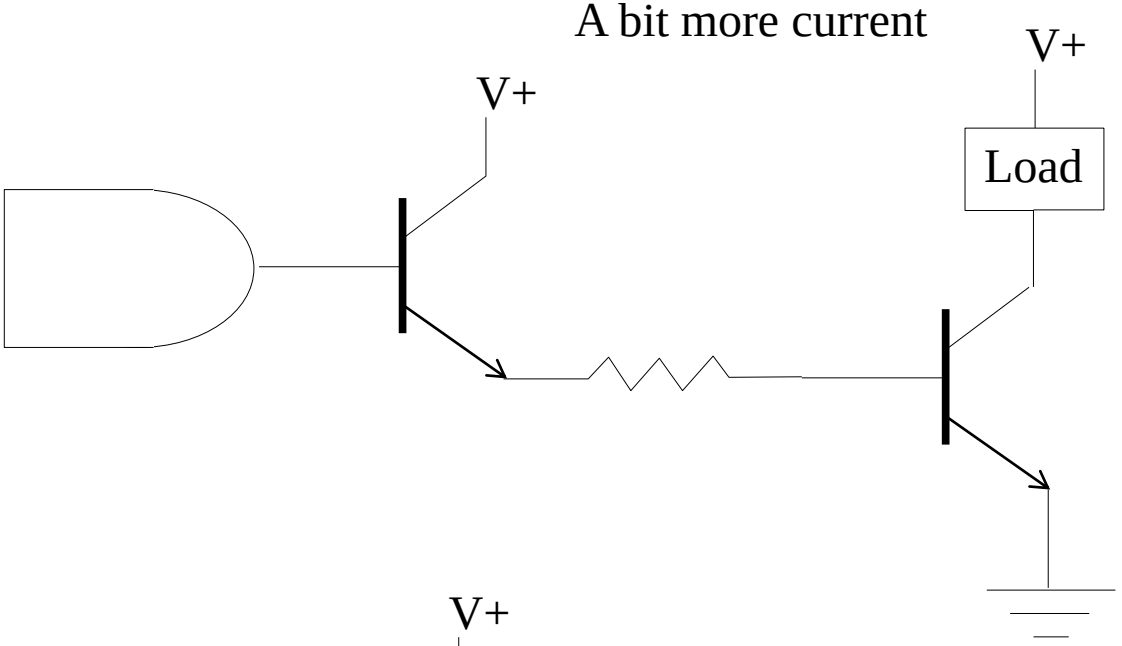


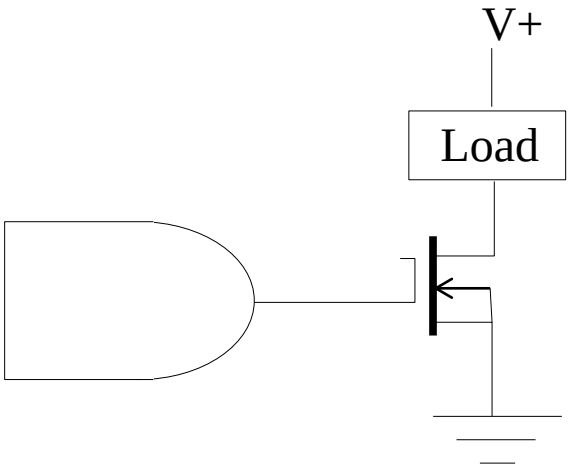
Driving loads: Transistors



eg 2N3904
max Ic ~ 100 mA

A bit more current

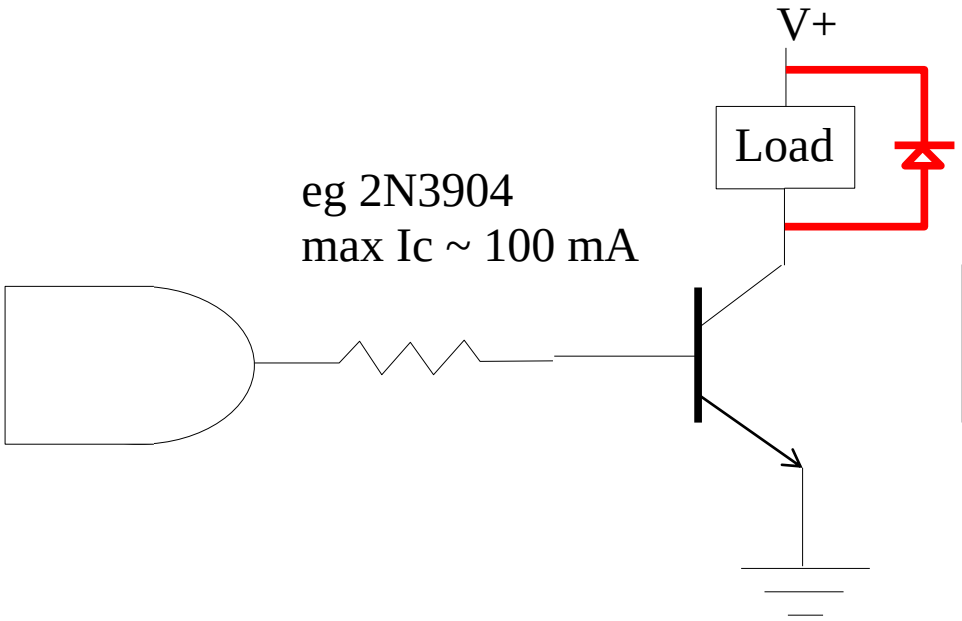




maximum current:
how much do you want?

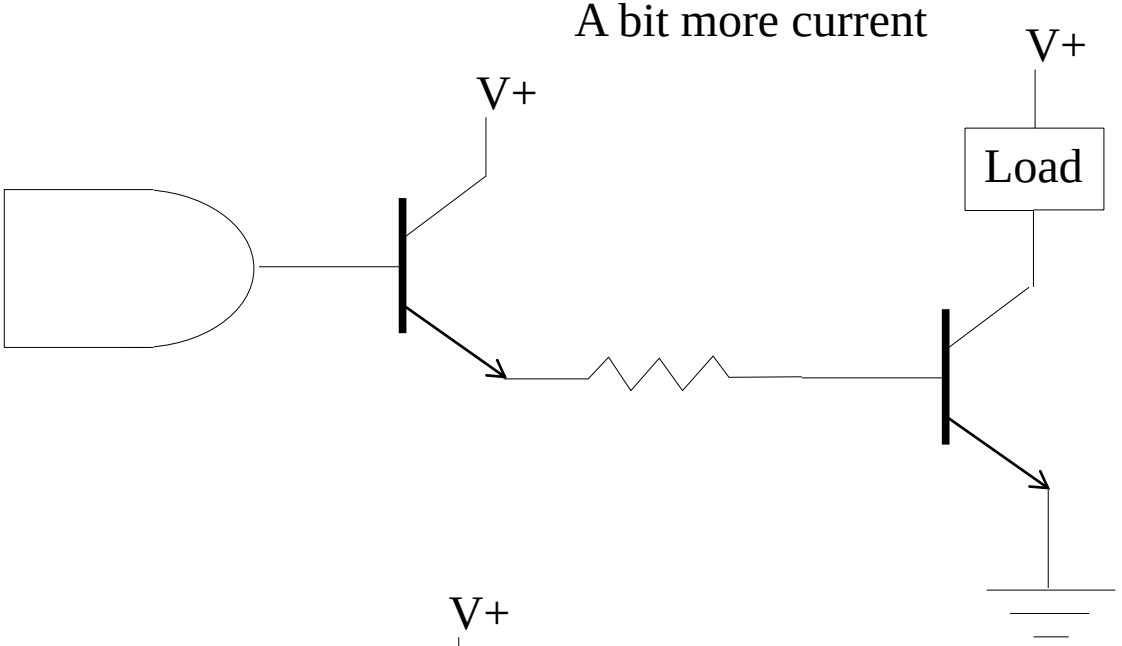
N-channel mosfet

Driving loads: Transistors

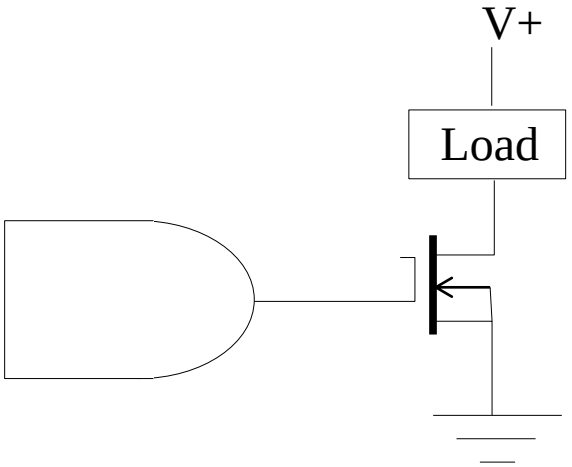


eg 2N3904
max Ic ~ 100 mA

A bit more current



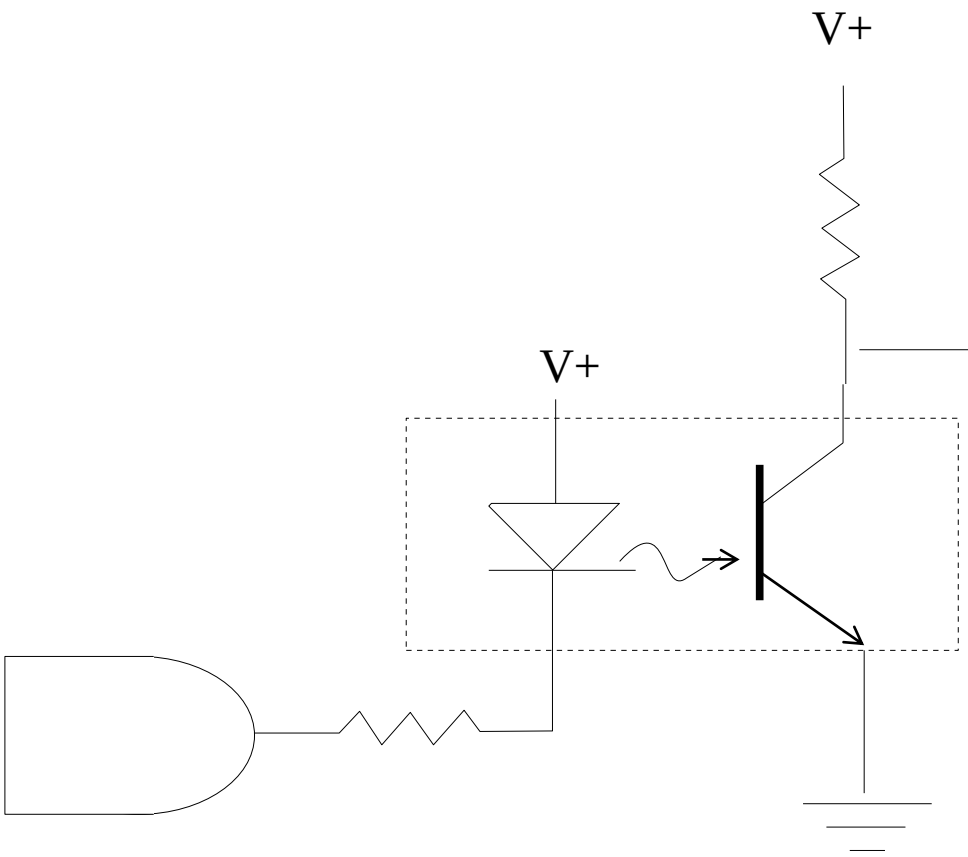
Inductive loads require that
you protect the transistor with a diode!



maximum current:
how much do you want?

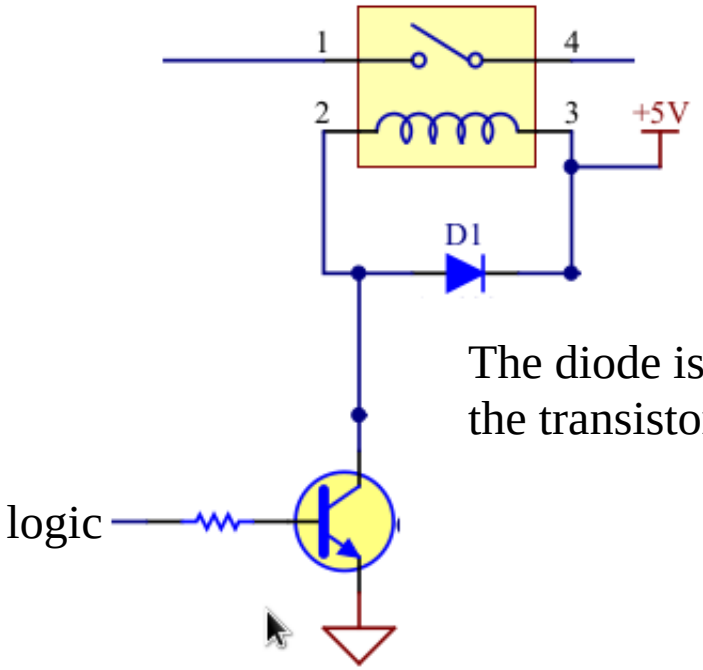
N-channel mosfet

Driving loads: Optoisolators



4N36, low current, but good isolation

Driving loads: Relays



The diode is essential to prevent destroying the transistor on turn-off!

There are some small low-current relays that can be driven directly by logic chips, again, a diode is essential to protect the logic circuit from the inductive spike on turn-off!

Driving loads: Solid-state Relays

good for AC, large loads, fast, repeated switching
(expensive, may need a heat sink), Often will synchronize to line voltage.

eg Crydom D2425: 280VAC, 25A !
$44

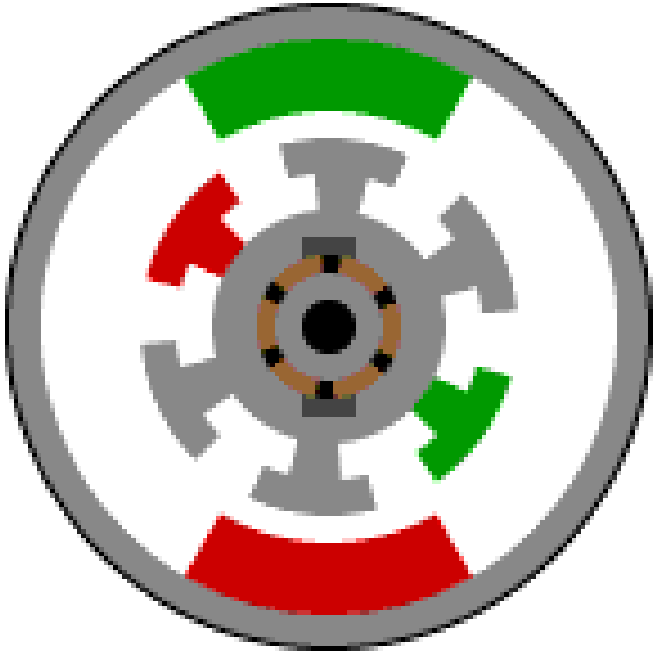
Driving loads: Solid-state Relays

good for AC, large loads, fast, repeated switching
(expensive, may need a heat sink), Often will synchronize to line voltage.

eg Crydom D2425: 280VAC, 25A !
$44

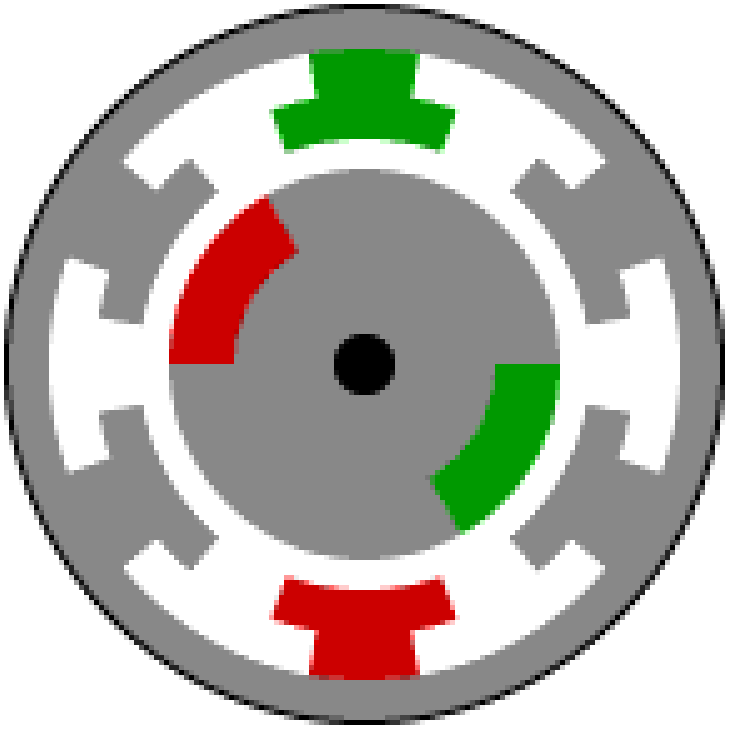
Be careful buying solid-state relays on the grey (Ebay etc) market, there are many 'fakes' out there.

Motors: DC motors



http://www.freescale.com/webapp/sps/site/overview.jsp?code=WBT_MOTORDCTUT_WP

Motors: Brushless DC motors



http://www.freescale.com/webapp/sps/site/overview.jsp?code=WBT_MOTORBLDCTUT_WP

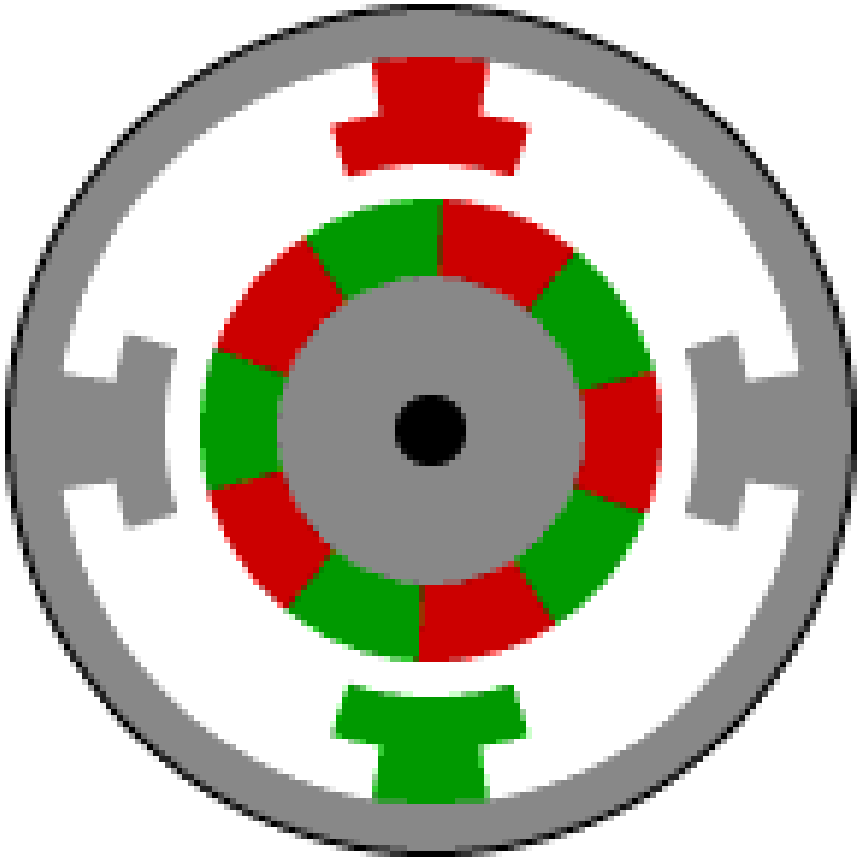
Motors: Servos

photo from Hobby Servo Fundamentals by Darren Sawicz

- precise position control
- provide a PWM signal to specify position
- typical range of 0-180 degrees

- contains a DC motor, gearing, a potentiometer, control electronics.
- The average PWM voltage is compared to the position, as measured by the potentiometer. The control electronics then drive the motor forward or backward to set the angle as requested.
- Typical PWM period of 20 ms with on time of ~1 – ~2 ms (1.5 ms is 'center'). Wire colours often: red = +5 V, black = ground, white = PWM control.
- Position depends on length of on-pulse.

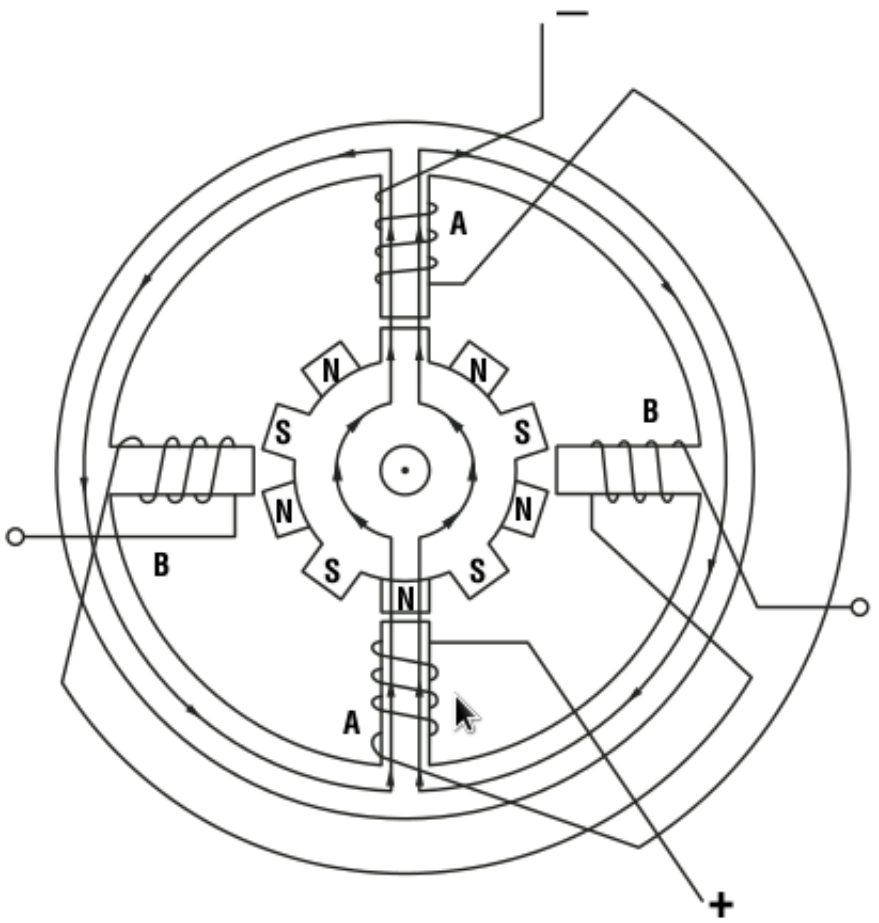
Motors: Stepper motors



Fixed step size, often 200 steps per revolution.

http://www.freescale.com/webapp/sps/site/overview.jsp?code=WBT_MOTORSTEPTUT_WP

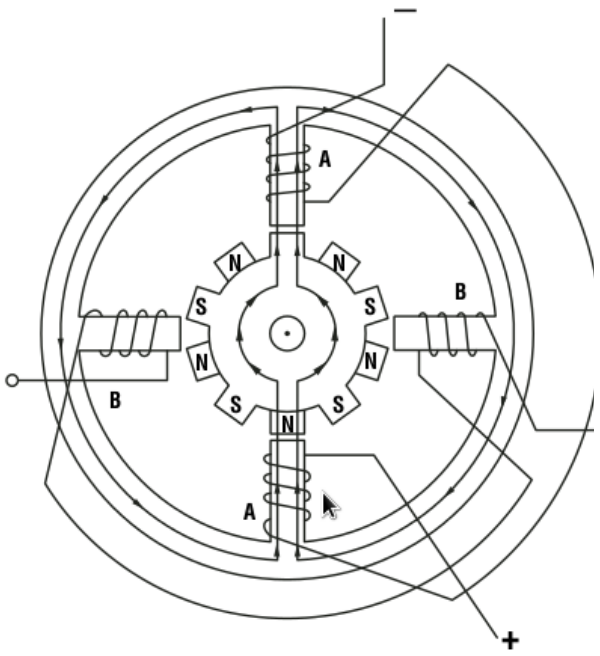
Motors: Stepper motors

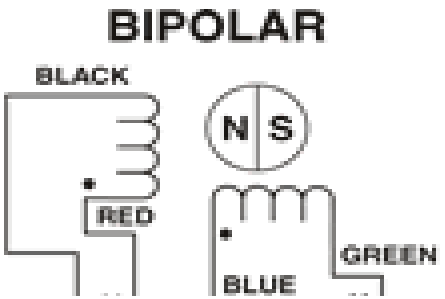


From: Introduction to Step Motors, Applied Motion Products. http://www.omega.ca/auto/pdf/REF_IntroStepMotors.pdf

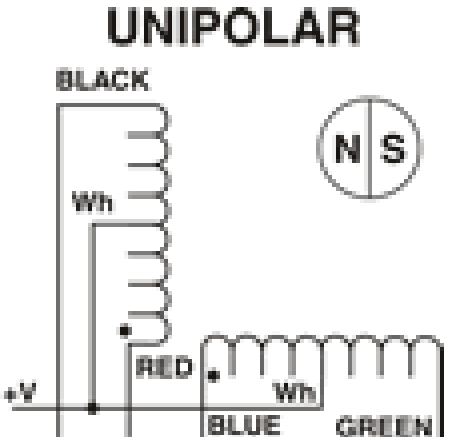
Motors: Stepper motors

Unipolar vs bipolar windings:





4 wires (usually)

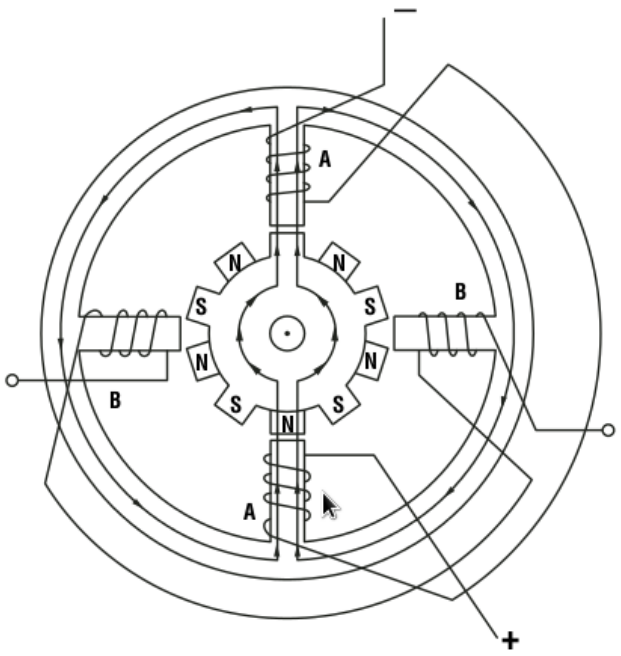


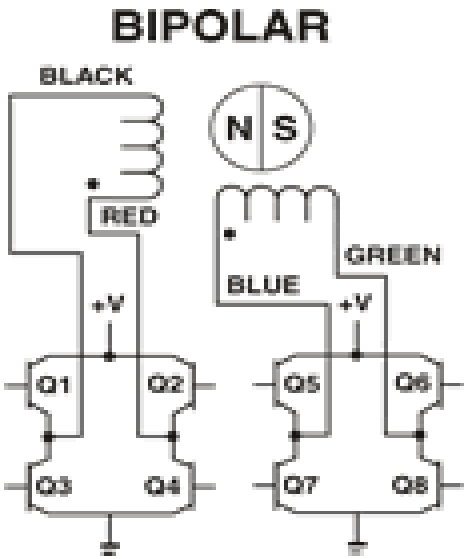
Unipolar stepper (5, 6 or 8 wires)

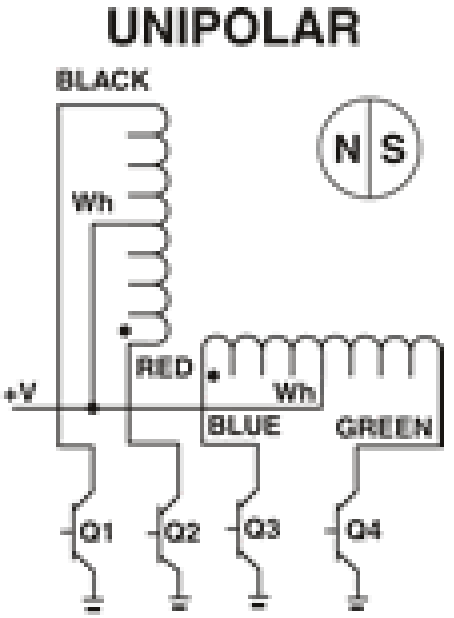
http://www.haydonkerk.com/Resources/StepperMotorTheory/tabid/192/Default.aspx

Motors: Stepper motors

Unipolar vs bipolar windings:



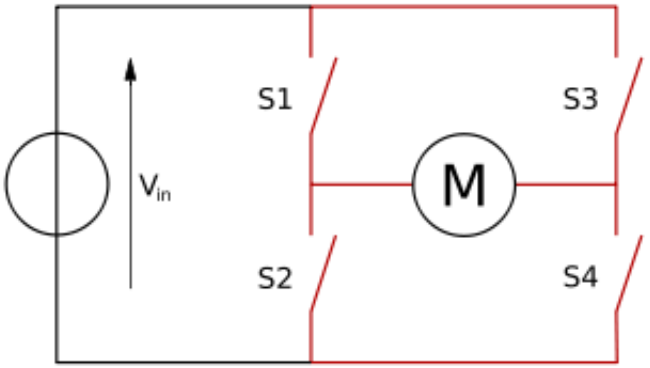




http://www.haydonkerk.com/Resources/StepperMotorTheory/tabid/192/Default.aspx

Driving Motors: H-bridge

To drive a dc motor in either direction with a single power supply, close S1 and S4 OR S2 and S3.



http://en.wikipedia.org/wiki/File:H_bridge.svg

The switches are often transistors: bipolar or MOSFETs