Motors and the UNO

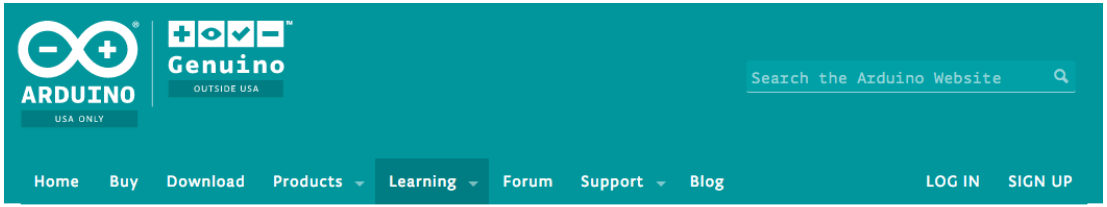
EE 189L (Space Vehicles LLC)

September 27, 2016

Resources for programming UNO

www.arduino.cc

examples in Arduino IDE (Sketch)



Reference Language | Libraries | Comparison | Changes

Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

Structure

- setup()
- loop()

Control Structures

- if
- if...else
- for
- switch case
- while

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT | INPUT_PULLUP
- LED_BUILTIN
- true | false
- integer constants
- floating point constants

Data Types

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

Variables

Memory allocated by name to store numbers. Favorite two types:

1. integer (int) uses 16 bits of memory to hold a number (no fractions) between $-2^{15} = -32{,}768$ and $2^{15}-1 = 32{,}767$

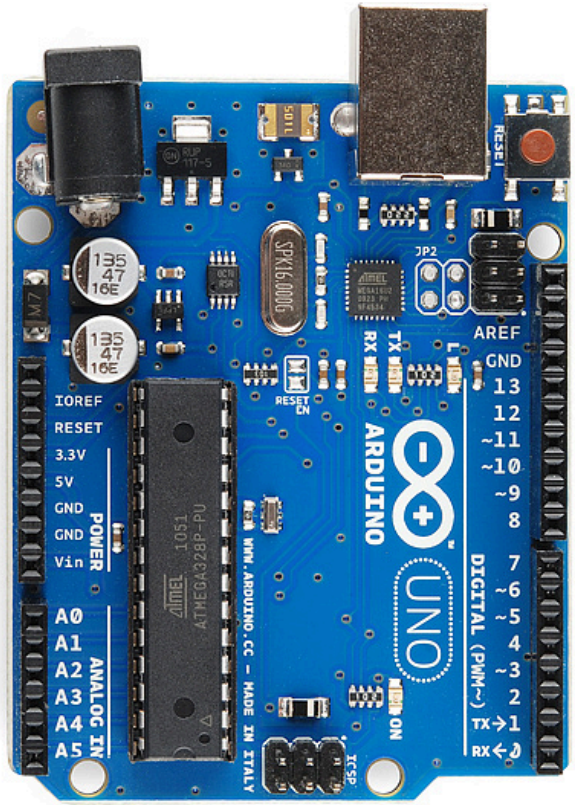
```
int aval, apin = 1;

aval = analogRead(apin);
```

2. floating point (float) uses 32 bits of memory to hold numbers with fractions (decimal places) between -3.4028×10^{38} and 3.4028×10^{38}; stored as sign, exponent and fraction after 1.__

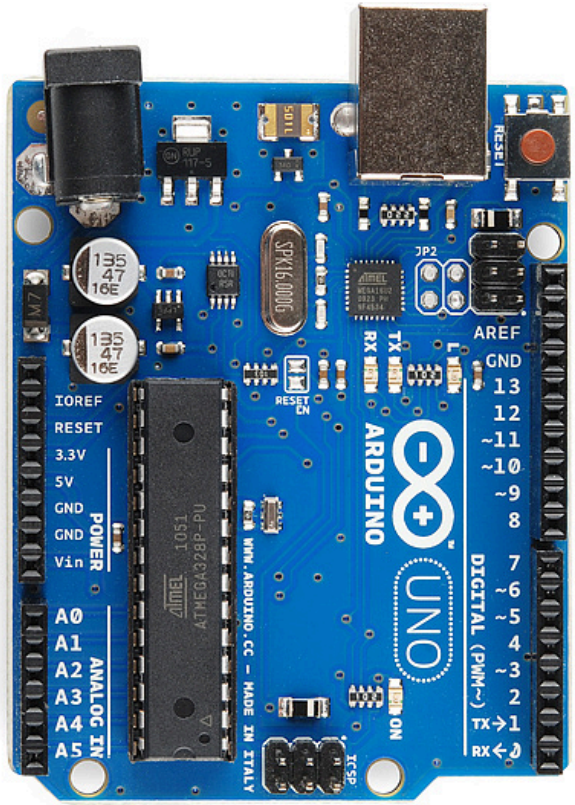
```
float pi = 3.14159, r = 1.0, area;

area = pi * r * r;
```

Review of Analog I/O

analogRead(pin) where pin = 0, 1, … 5 and 10 bit integer value returned is between 0 and 2^{10}-1=1023



Review of Digital I/O



pinMode(pin, MODE)

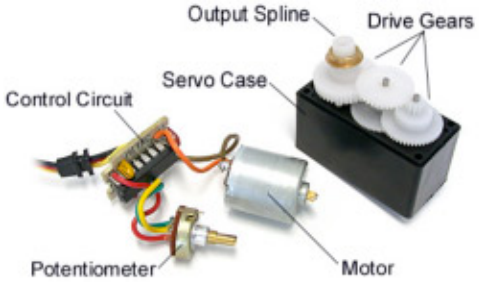
digitalRead(pin, value)
where value is 0 or 1
(LOW or HIGH)

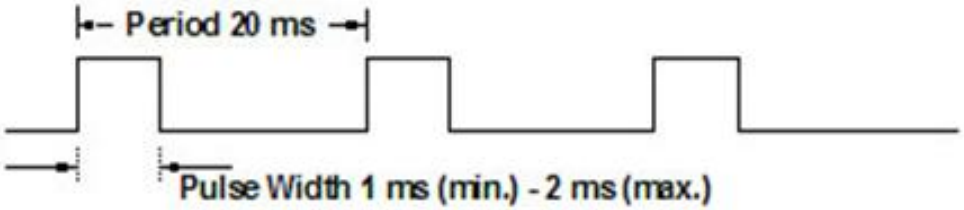
digitalWrite(pin)

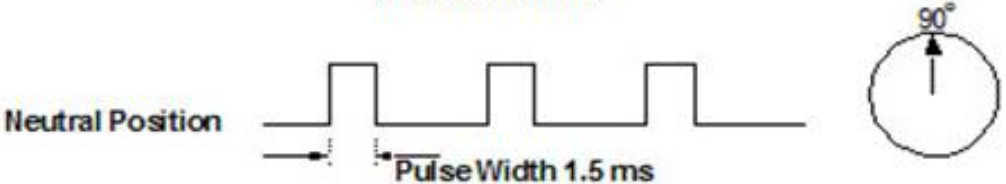
analogWrite(~pin, DC)
where duty cycle, DC,
between 0 and 255

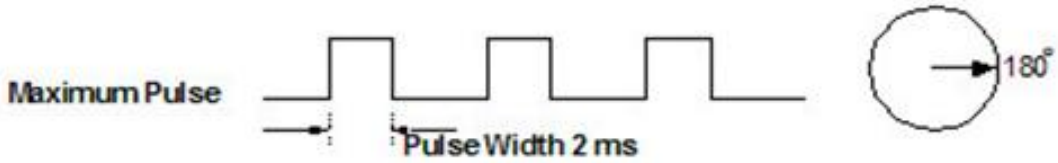
Servomotors

PWM = Orange (⎍⎍)
Vcc = Red (+)
Ground = Black (--)



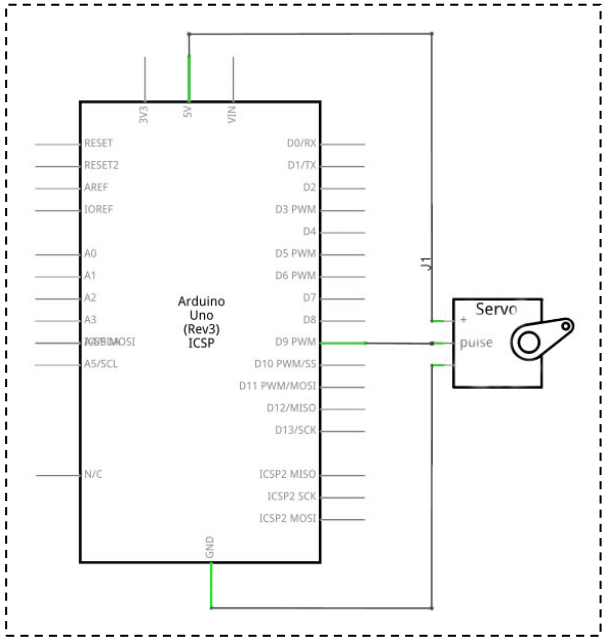


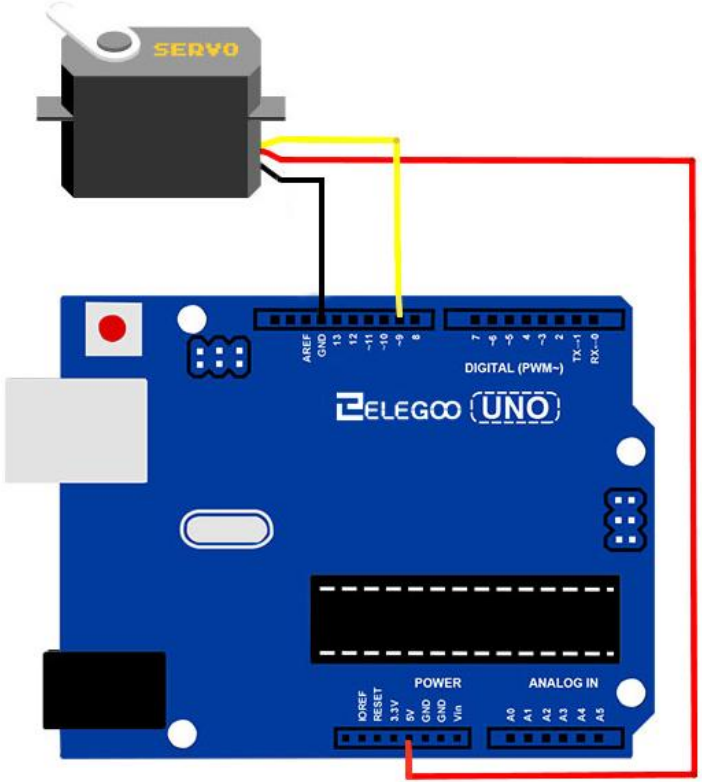




Servo library

Uses functionality of PWM on pins 9 and/or 10 and library of commands/functions to create repeating pulses as input to servo motor

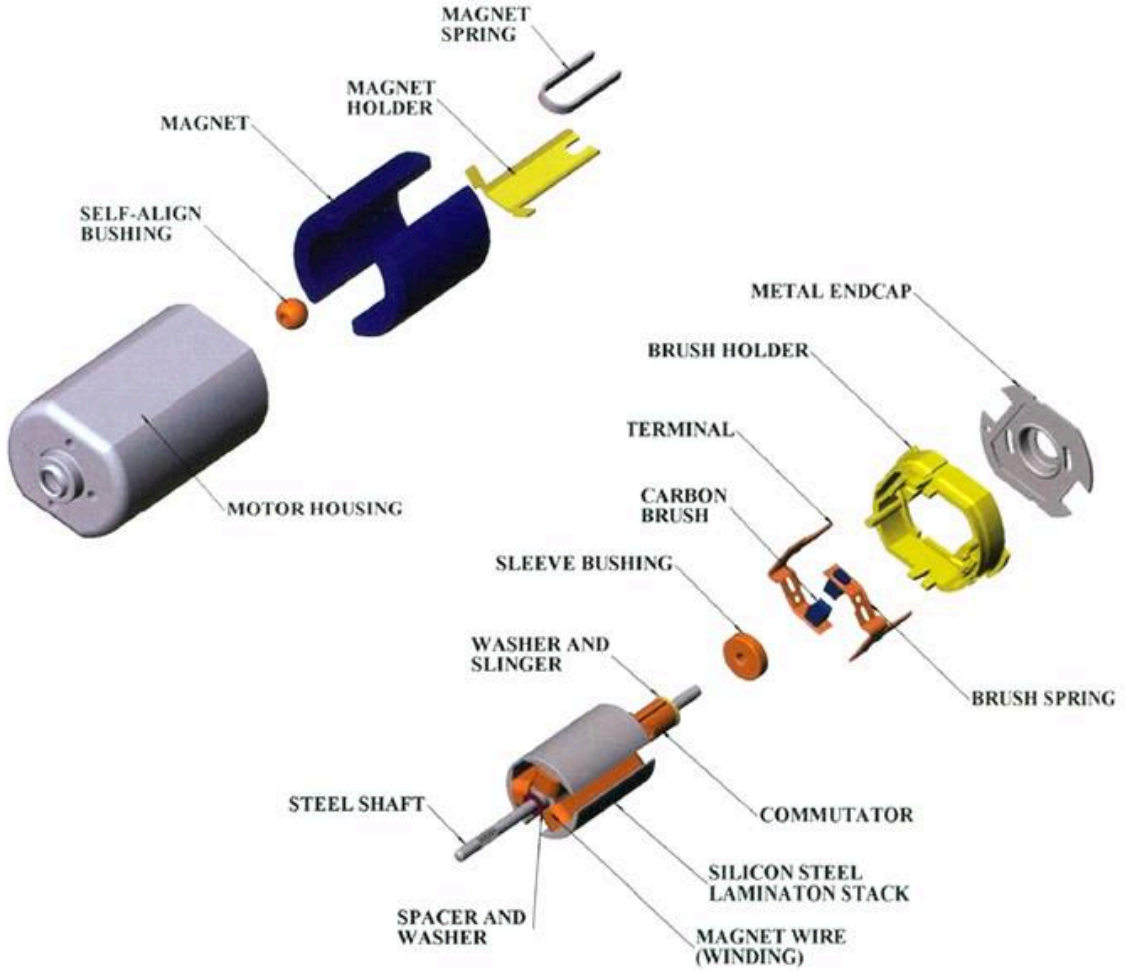




Servo library for “sweep”

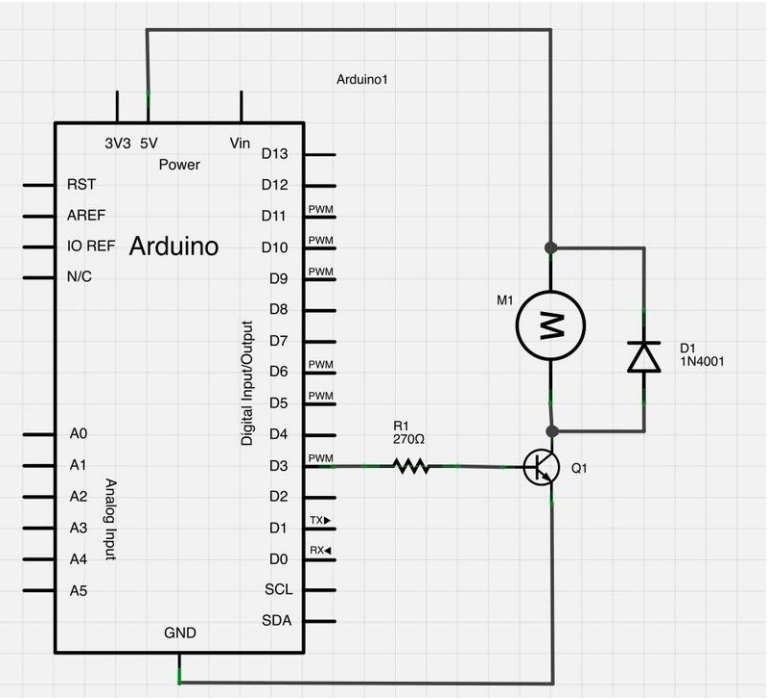
```
#include <Servo.h>
Servo myservo;                  // create servo object to control a servo
int pos = 0;                    // variable to store the servo position

void setup() {
                myservo.attach(9);              // attaches the servo on pin 9 to the servo object
}

void loop() {
                for (pos = 0; pos <= 180; pos = pos + 1) {    // goes from 0 degrees to 180 degrees in steps of 1 degree
                                myservo.write(pos);                             // tell servo to go to position in variable 'pos'
                                delay(15);                                      // waits 15ms for the servo to reach the position
                 }
                for (pos = 180; pos >= 0; pos = pos - 1) {    // goes from 180 degrees to 0 degrees in steps of 1 degree
                                myservo.write(pos);                             // tell servo to go to position in variable 'pos'
                                delay(15);                                      // waits 15ms for the servo to reach the position
                 }
}
```

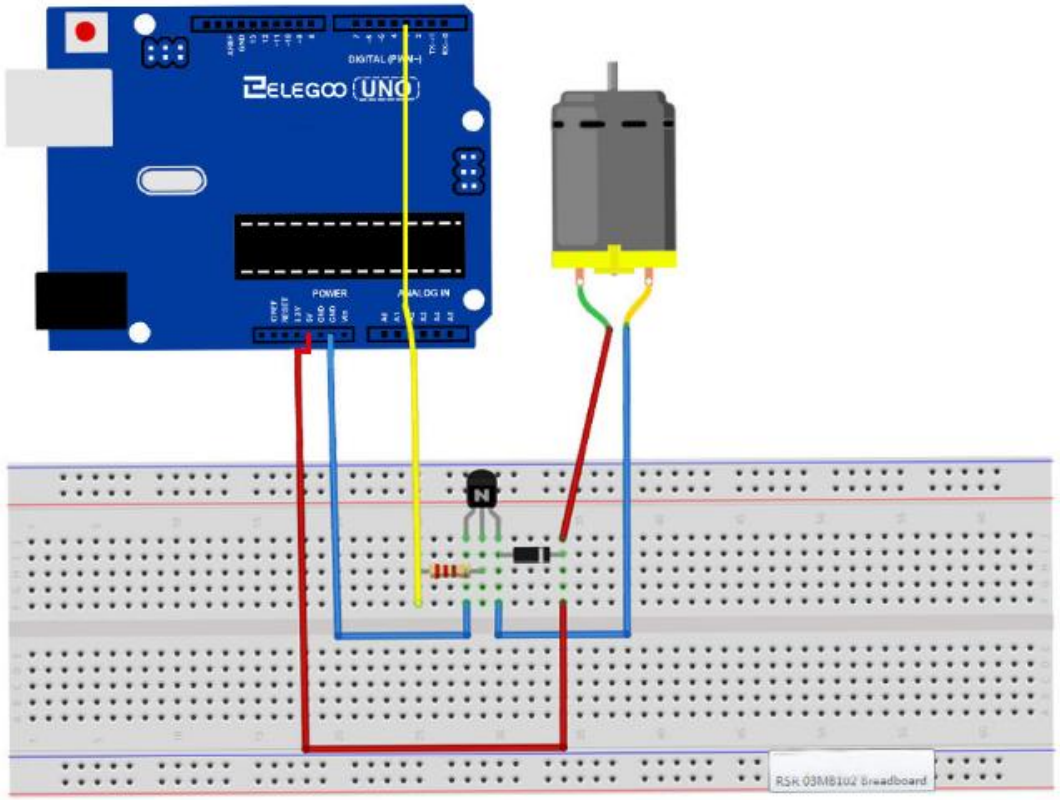
DC Motors


MAGNET SPRING
MAGNET HOLDER
MAGNET
SELF-ALIGN BUSHING
MOTOR HOUSING
METAL ENDCAP
BRUSH HOLDER
TERMINAL
CARBON BRUSH
SLEEVE BUSHING
WASHER AND SLINGER
BRUSH SPRING
STEEL SHAFT
COMMUTATOR
SILICON STEEL LAMINATON STACK
SPACER AND WASHER
MAGNET WIRE (WINDING)

DC Motors

We'll control with PWM (analogWrite()), but will need "amplification" via a transistor to supply higher currents required by motor and back-emf protection (diode)





DC Motors – “sweep” speed

```
int pwm = 0;                                    // variable to store the value for pwm

void setup() {
        pinMode(3, OUTPUT);       // use digitial/PWM pin 3 as output
}

void loop() {
        for (pwm = 0; pwm <= 255; pwm = pwm + 1){          // goes from 0 PWM (off) to 255 PWM (on) in steps of 1
                analogWrite(3, pwm);                       // output pwm
                delay(15);                                 // wait 15ms
         }
        for (pwm = 255; pwm >= 0; pwm = pwm - 1) {         // goes from 255 PWM (on) to 0 PWM (off) in steps of 1
                analogWrite(3, pwm);                                  // output pwm
                delay(15);                                 // wait 15ms
         }
}
```

Today’s goal

Write programs and build electronics to “drive” servomotor and DC motor

1. Sweep servomotor back and forth between 0 and 180 degrees
2. Control servomotor with joystick with “left” mapped to 0 degrees and “right” mapped to 180 degrees.
3. Sweep DC motor from stopped (0 PWM to full speed 255 PWM) and back
4. Control DC motor’s speed with joystick where “centered” is stopped and “right” is full speed