

Arduino Quick Reference

BLINK EXAMPLE

```
/*This is a multi line comment
the setup runs once*/
void setup() {
  //This is a single line comment
  pinMode(LED_BUILTIN, OUTPUT); //Set the pin 'LED_BUILTIN' to be an output
}

// 'loop' function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second, 1000 milliseconds equals 1 second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(500); // wait half a second, now we jump to the start of 'loop'
}
```

CONTROL

To control how your program flows use these tools in your code

IF

```
if (x < 5){ //if x is less than run
  x = x + 1; //if x less than 5, do this
}else{
  x = 0; //if x is 5 or more, do this
}
```

SWITCH

```
switch ( myVar ){
  case 1:
    x = "hello"; //do this if myVar = 1
    break;
  case 2:
    x = "world"; //do this if myVar =2
    break;
  default:
    x = "oops"; //do this for no matches
}
```

WHILE

```
while ( x < 5 ){
  x = x + 1;
}
/* while x is less than 5, add one to x,
when x isn't less than 5, stop the loop */
```

FOR

```
for ( int i=0; i<10; i++){
  Serial.println("hello"+ String(i));
}
/* starting at 0, increment by 1, stop
when i isn't less than 10.
i++ means increment by 1*/
```

SERIAL EXAMPLE

//Great for debugging and displaying status to the screen
//After upload, go to Menu: Tools > Serial Monitor

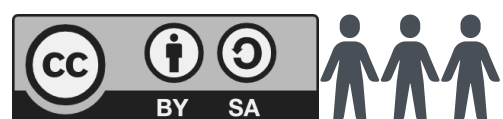
```
int x = 0; // create a global variable x and set it to zero

void setup(){
  Serial.begin(9600); // set the serial port speed to 9600 baud
}

void loop(){
  Serial.println("hello!"); // print the string "hello!" with a new line
  Serial.print("x = "); // print the string "x = " without a new line
  Serial.println(x); // print the variable x, with a new line
  x++; // increment x for the next loop,
} // same as x=x+1
```

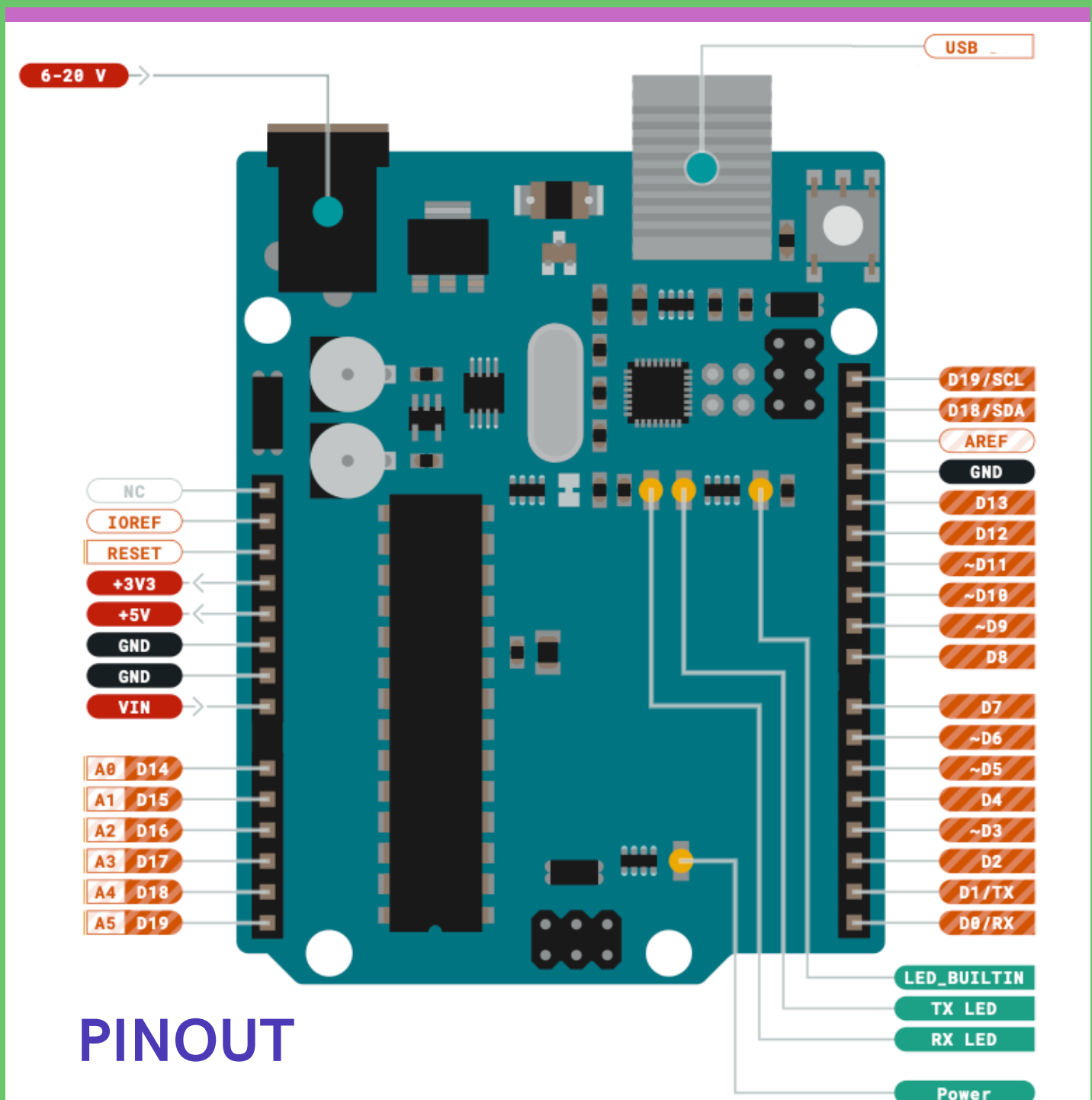
FREE TO SHARE

Sharable under the Creative Commons Licence, simply attribute TinkerElectric.com



DATA TYPES

boolean	(0 / FALSE or 1/TRUE)
char	(character a, A, b,B ... -128 to 127)
byte	(0 to 255)
int	(-32,768 to 32,767)
long	(-2,147,483,648 to 2,147,483,647)
float	(-3.4028235E+38 to -3.4028235E+38)



PINOUT

Ground	Digital Pin	MAXIMUM current per I/O pin is 20mA
Power	Analog Pin	MAXIMUM current per +3.3V pin is 50mA

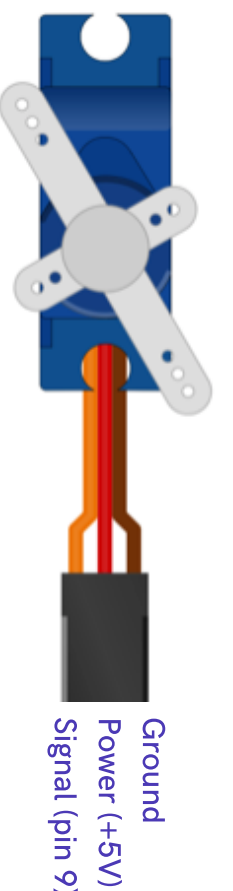
Image: Arduino.cc under CC licence

SERVO EXAMPLE

```
//Servos accept an angle
#include <Servo.h> // include the servo library
Servo myservo; // create a servo instance
int pos = 0; // global variable to set the servos angle (degrees)

void setup(){
  myservo.attach(9); // attach the servo signal to pin 9 (other pins ok)
}

void loop(){
  for (pos = 0; pos <= 180; pos += 1) {
    myservo.write(pos); // write the servo position 0 to 180 degrees
    delay(15); // pause for 15 milliseconds between changes
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    myservo.write(pos); // repeat the process in reverse, 180 to 0 degrees
    delay(15);
  }
} // end of program
```



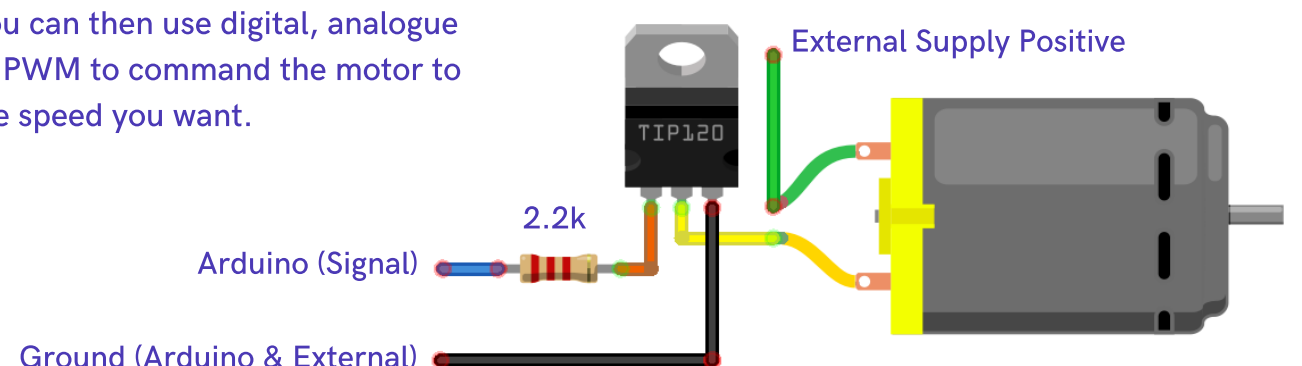
MORE POWER TO YOU

Devices like Pumps, Motors, bright LEDs and Solenoids need more power than the Arduino can provide.

To overcome this issue you can use a transistor, MOSFET or motor driver to do the job. These devices accept a signal from the Arduino and take power from an external source such as another battery or power pack.

Below is one way of doing this. The TIP120 transistor is fed a signal from the Arduino and then conducts the power from the external supply to the motor when needed.

You can then use digital, analogue or PWM to command the motor to the speed you want.



Want to shorten the time from idea to done?
Simply email us!

Q&A, classes & to share your project!...
hello@TinkerElectric.com