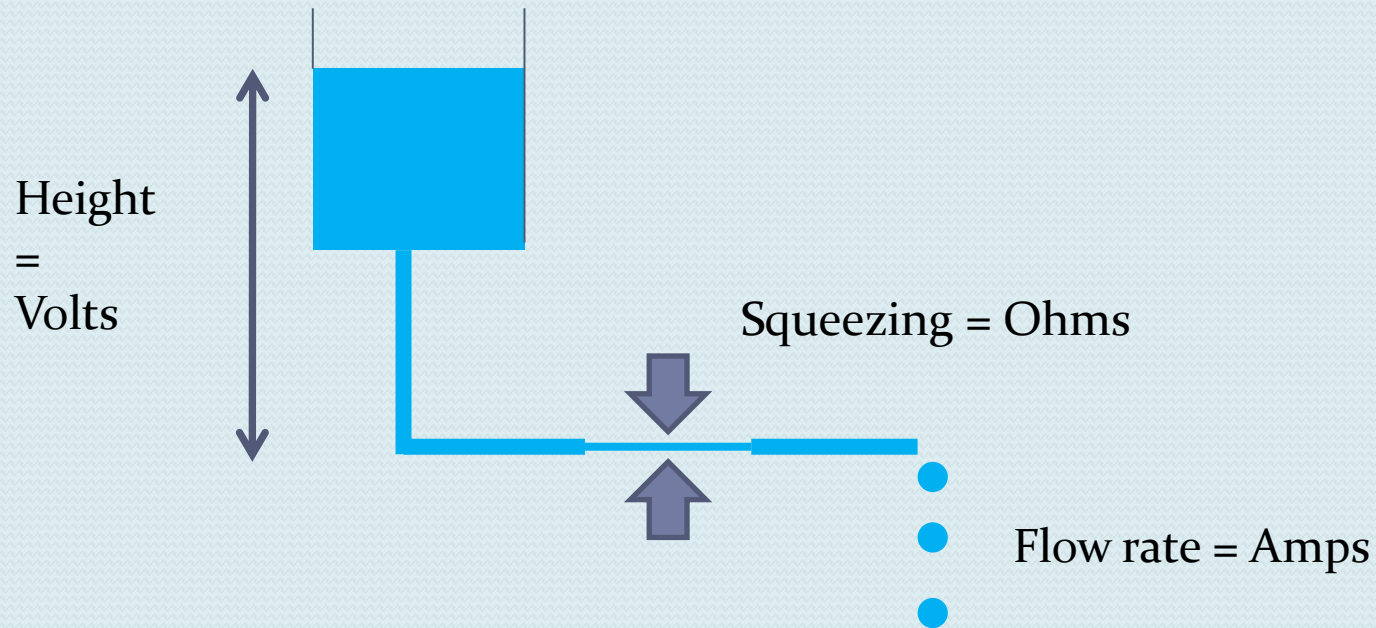


Arduino Sundae

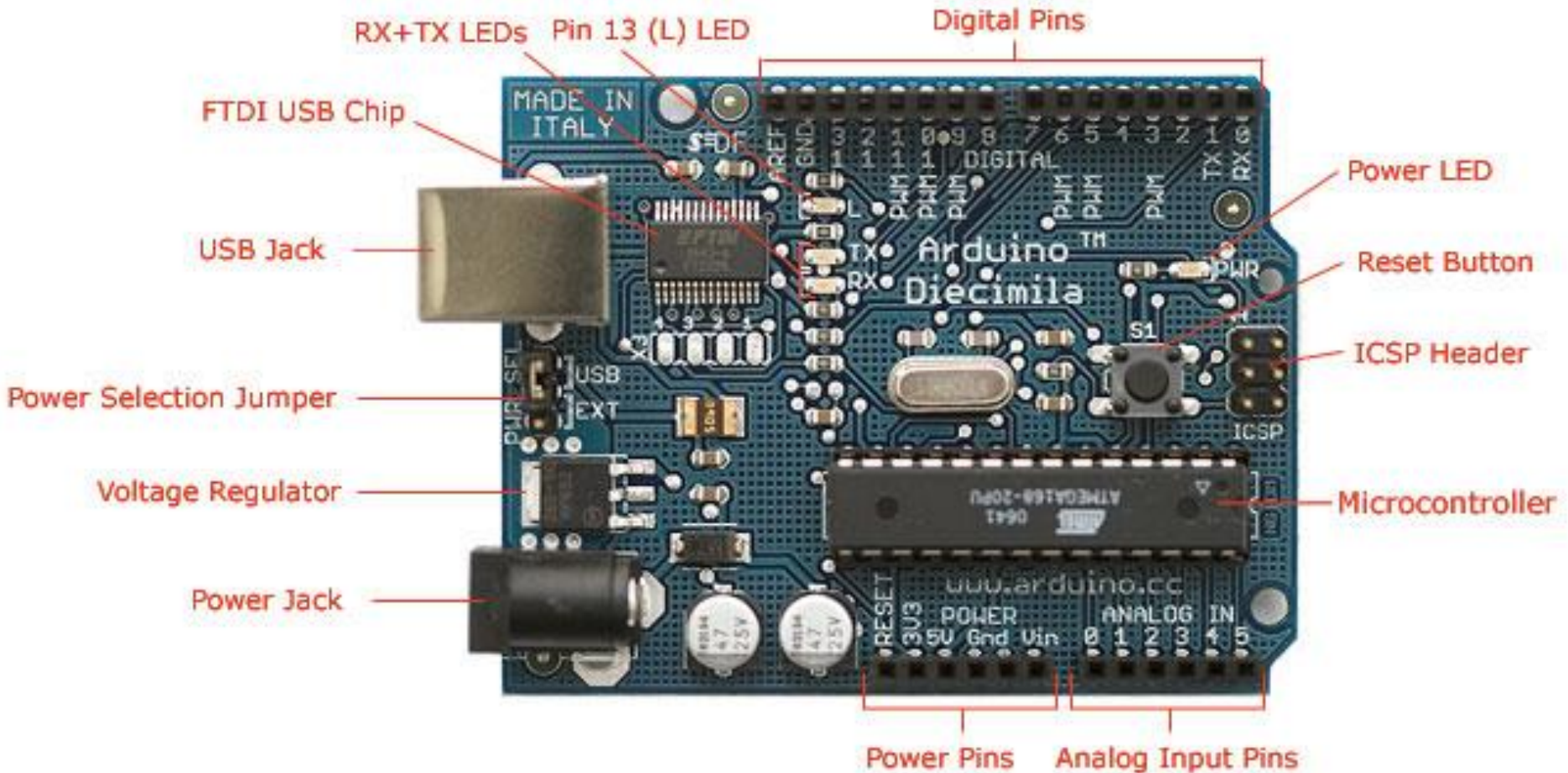
Nottinghack

Concepts

- Volts
 - Amps
 - Ohms
- Pressure
Flow
Resistance

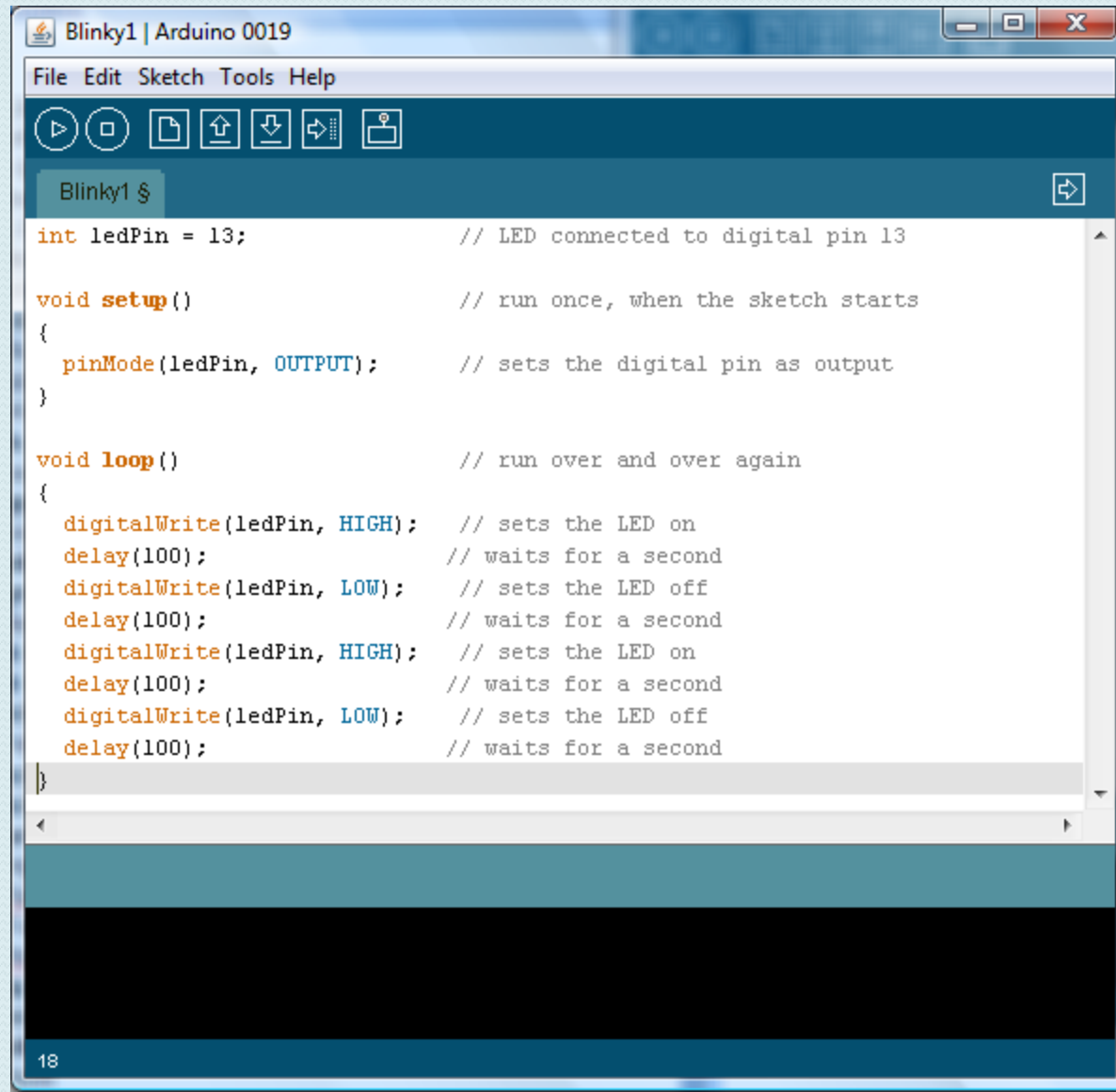


Arduino



Programming

- Arduino IDE
- Program is called a sketch



The screenshot shows the Arduino IDE interface. The title bar reads 'Blinky1 | Arduino 0019'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for running, stopping, saving, and other functions. The main text area contains the following code:

```
int ledPin = 13; // LED connected to digital pin 13

void setup() // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop() // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(100); // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(100); // waits for a second
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(100); // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(100); // waits for a second
}
```

The code is color-coded: keywords like 'int', 'void', 'pinMode', 'digitalWrite', and 'delay' are in orange, and literals like 'HIGH', 'LOW', and 'OUTPUT' are in blue. Comments are in grey. The IDE has a status bar at the bottom showing the number '18'.

Blink a led

You must have this – all initialisation goes here

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
```

You must have this – the main code goes here

```
void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
```

Serial monitor

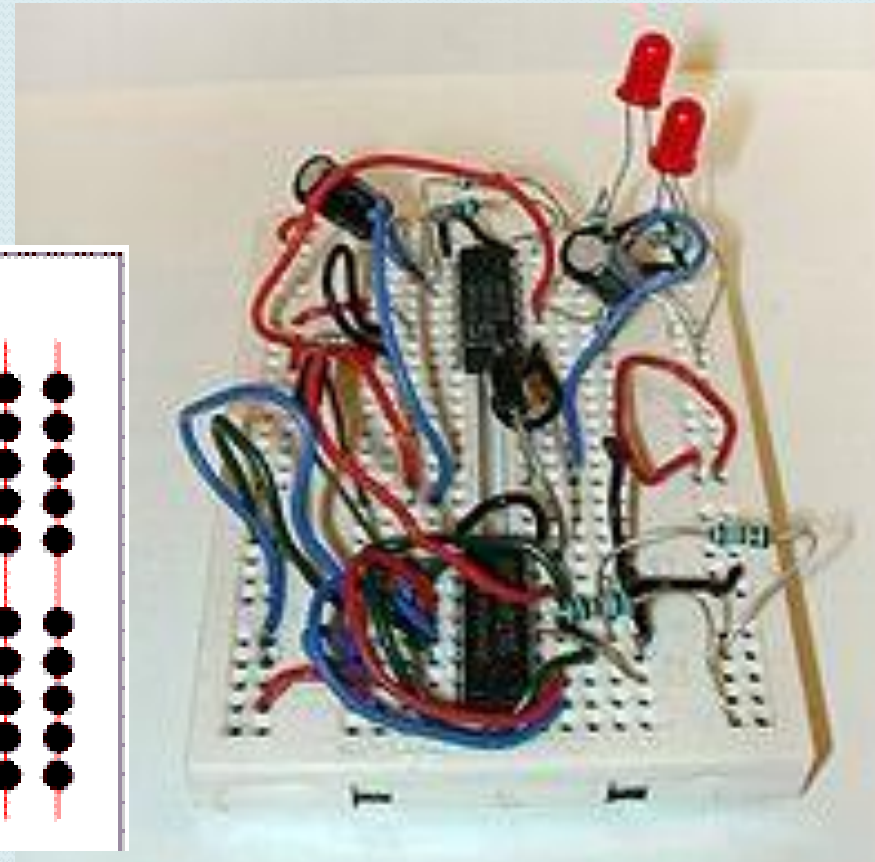
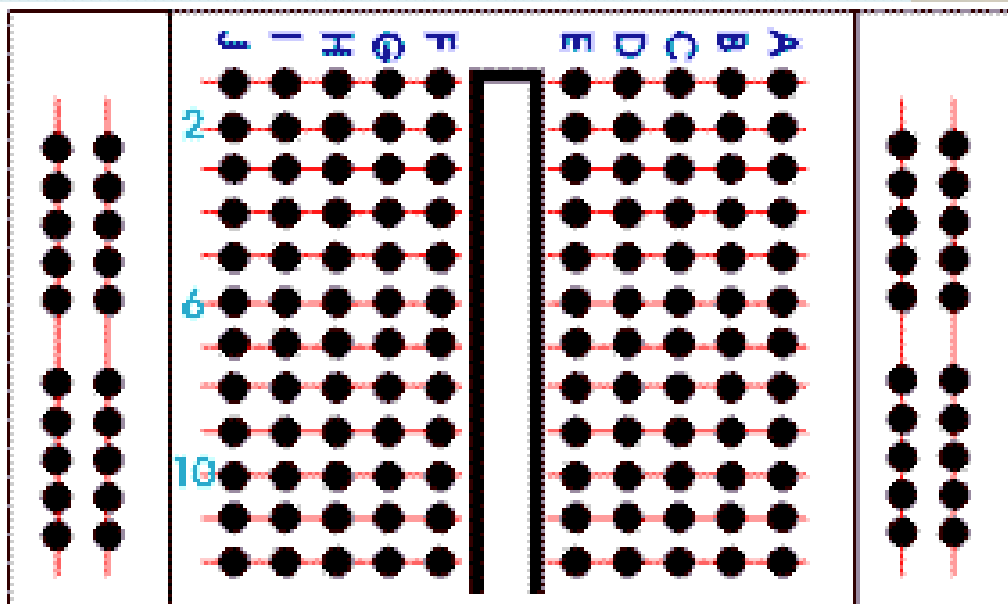
```
int ledPin = 13;           // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
  Serial.begin(9600);      // start the serial port at 9600
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  Serial.println("LED on");
  delay(1000);              // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  Serial.println("LED off");
  delay(1000);             // waits for a second
}
```

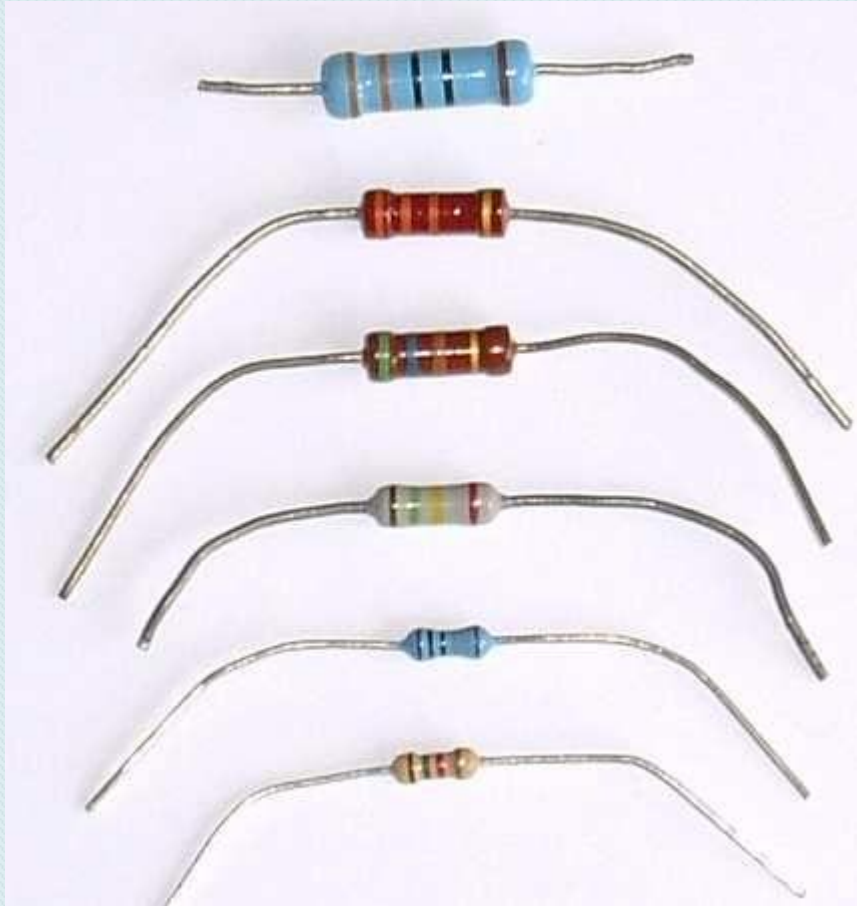
Tools

- Breadboard

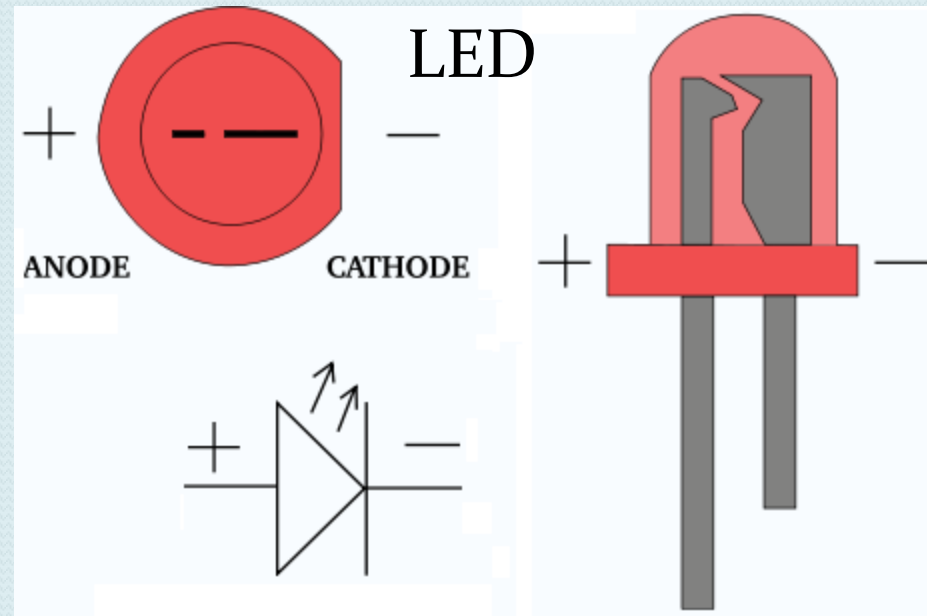
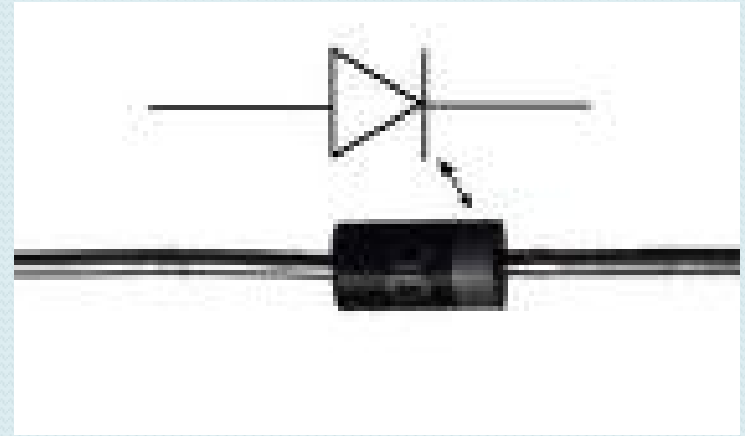


Components

- Resistor



Diode



Blink another led

Alter the previous sketch

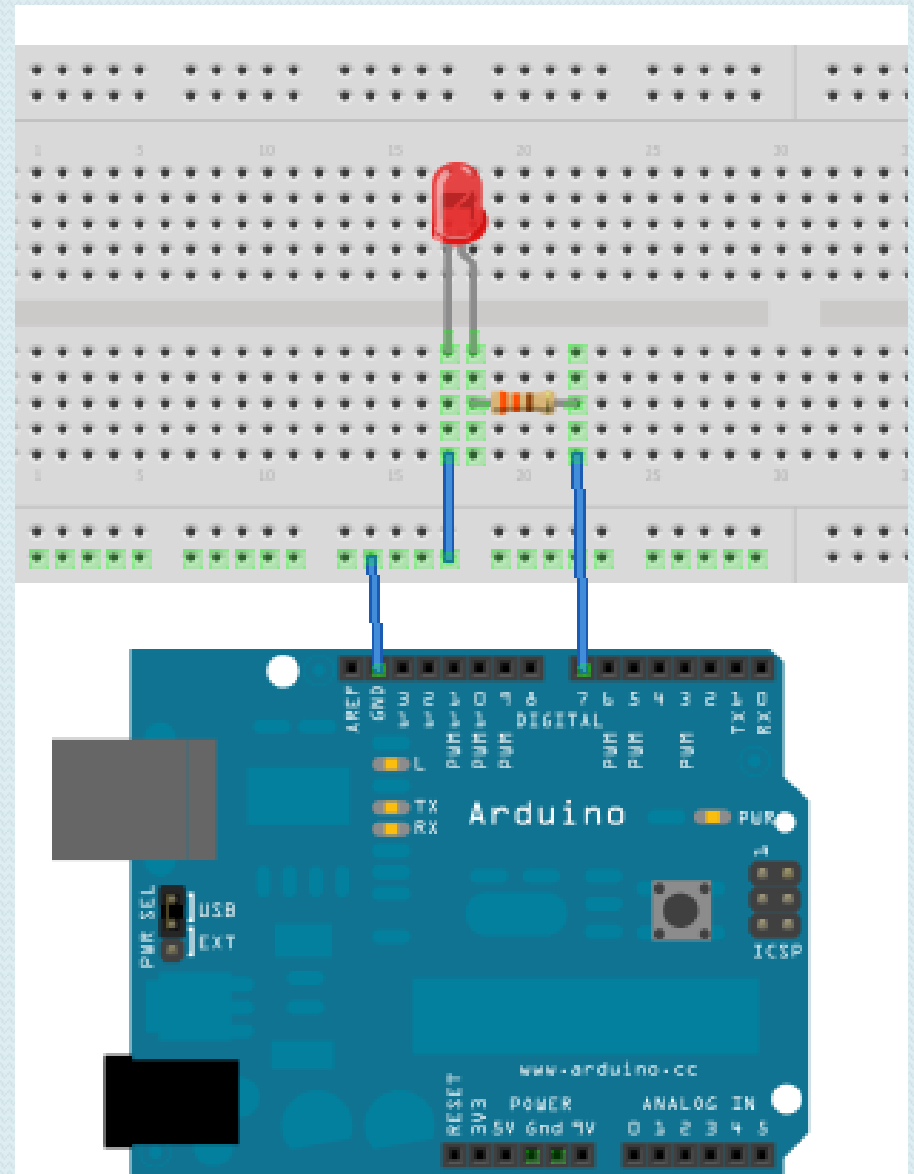
```
int ledPin = 13;
```

To

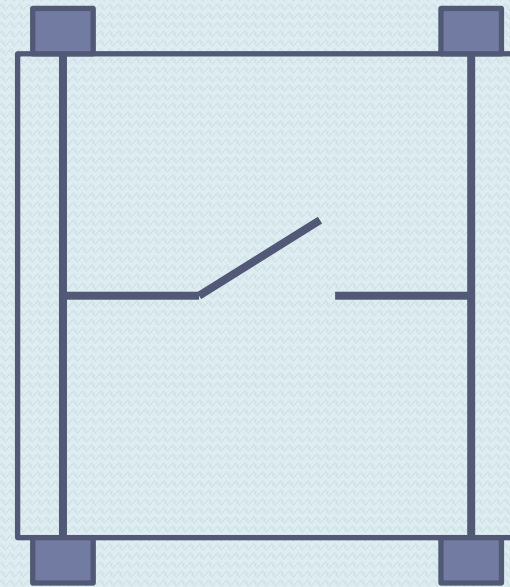
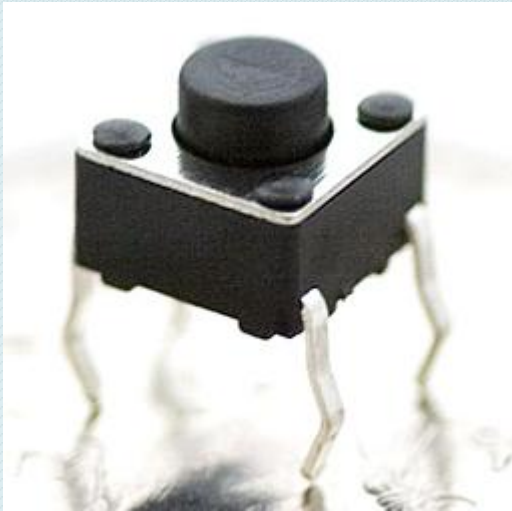
```
int ledPin = 7;
```

Exercises

- Blink both LEDs alternately
- Alter the blink frequency
- Two short blinks and one long blink



Switch



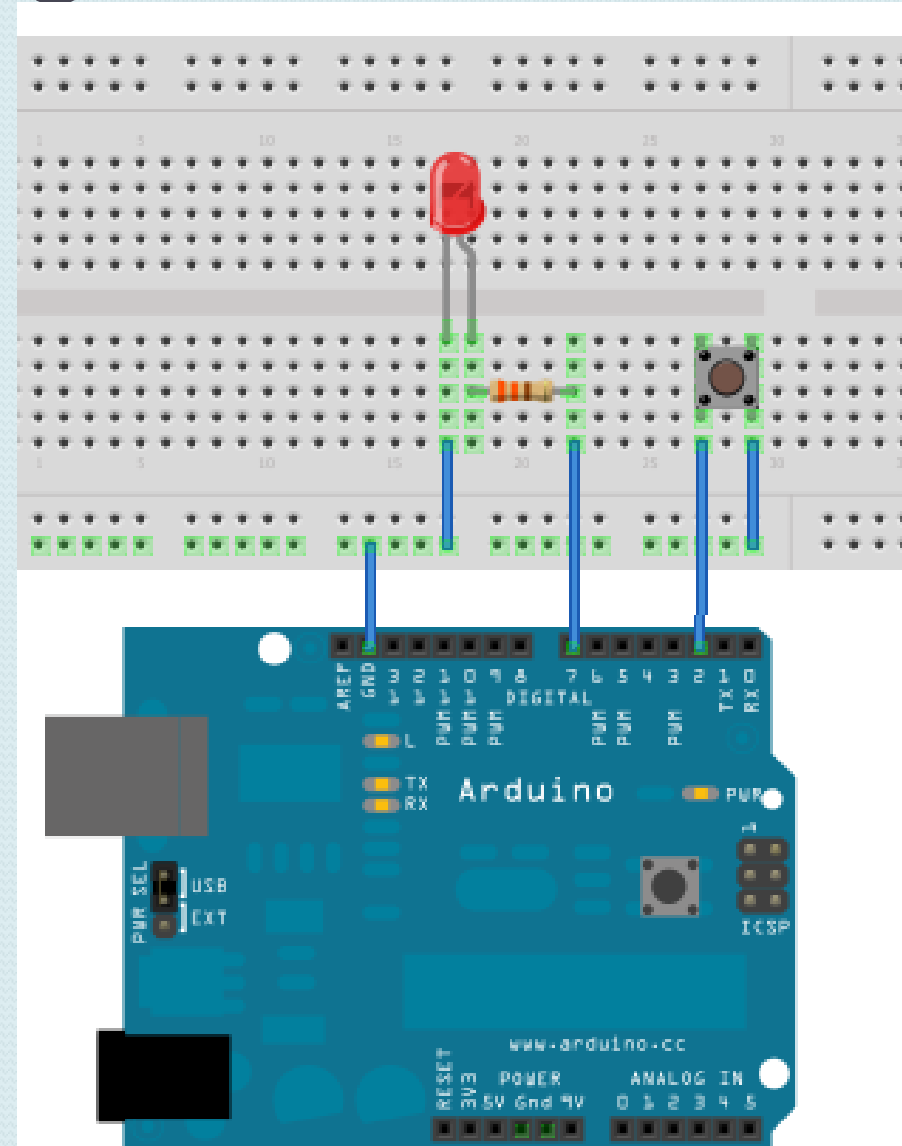
Switch controlling Led

```
int ledPin = 13;
int switchPin = 2;

int switchvalue; // store the switch value

void setup () {
  // sets the LED pin as output
  pinMode(ledPin, OUTPUT);
  // Set the switch pin as input
  pinMode(switchPin,INPUT);
  // activate the pullup
  digitalWrite(switchPin,HIGH);
}

void loop () {
  // read the switch value
  switchvalue = digitalRead(switchPin);
  // and write it to the LED
  digitalWrite(ledPin, !switchvalue);
}
```



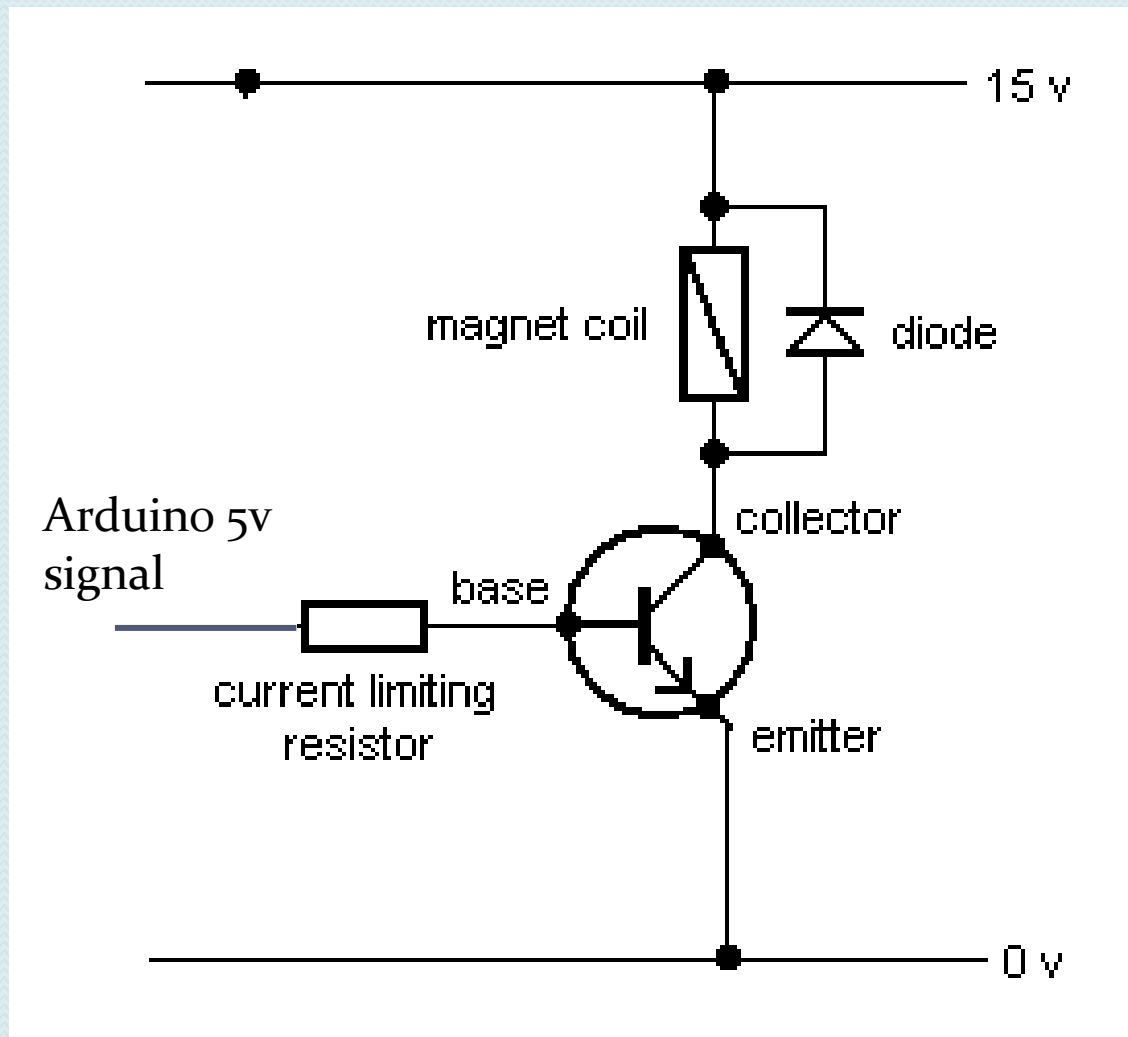
Exercise

- blink the LED twice each time the switch is pressed.
- Hint use an if statement
- `if (switchvalue == LOW)`
- `{`
- `... code to blink LED twice ...`
- `}`

Motors and relays

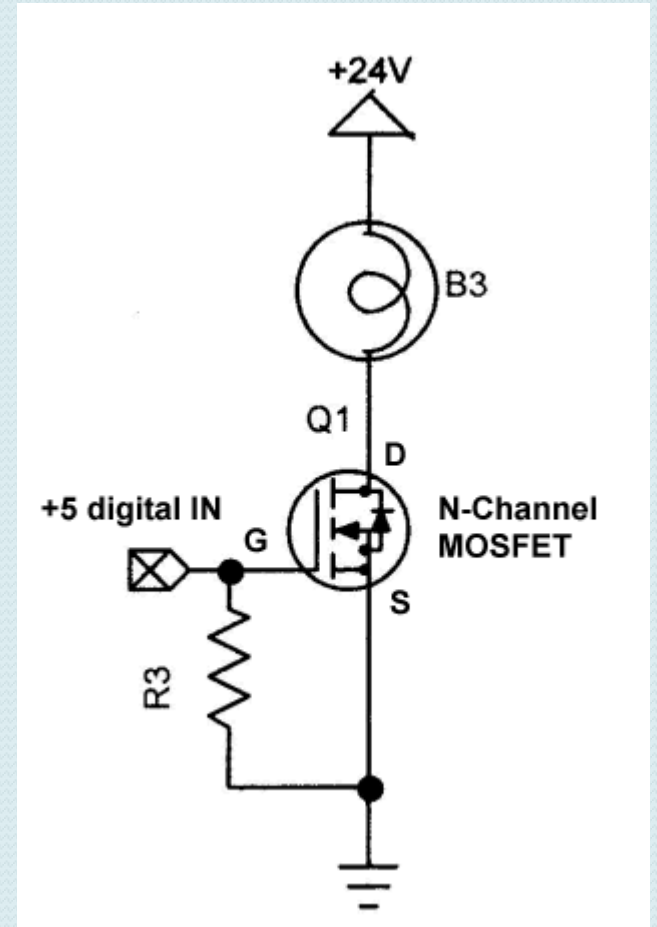
- These devices need more current than the Arduino can supply.
- We use a transistor to increase the current available
- Transistor acts as a Arduino controlled switch.
- Need protection from inductive loads (usually wound coils such as motors and relays).
- NPN Bipolar and N-channel MOSFET are commonly used with the Arduino.
- Voltage of the load (motor etc.) can be higher than Arduino voltage.

NPN Bipolar transistor



N-channel MOSFET

- Voltage controlled device
- R_3 ensures that the MOSFET turns off if the input is disconnected



Speed control

Pulse width modulation (PWM)

Drives motor at full voltage.

This gives high torque even at low speeds.

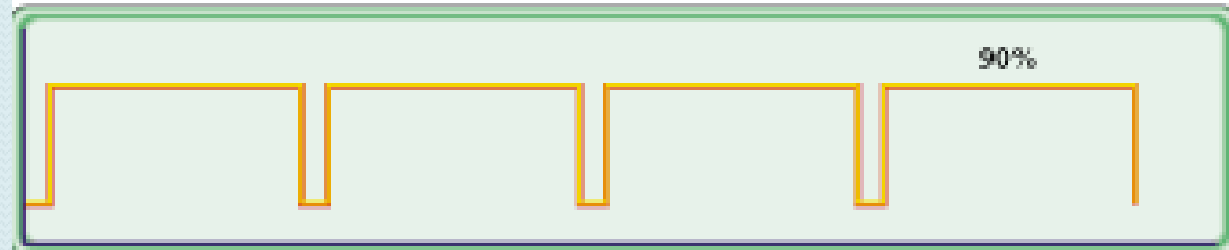
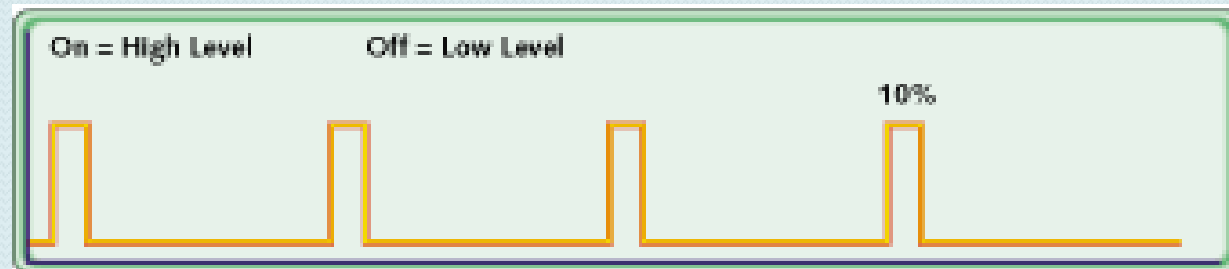
Used on Arduino pins

3, 5, 6, 9, 10, 11

Code:

```
analogWrite(pin,value);
```

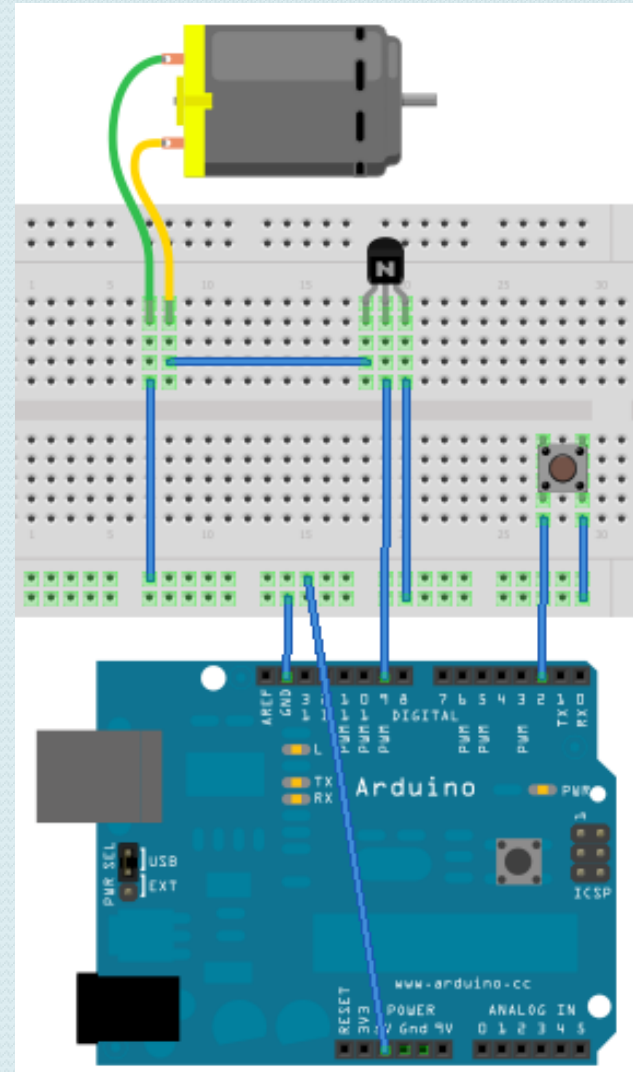
Value 0 - 255



Arduino motor control

Motor control according to switch

```
void setup() {  
  pinMode(9,OUTPUT);  
  pinMode(2,INPUT);  
  digitalWrite(2,HIGH);  
}  
  
void loop() {  
  if (digitalRead(2) == 0) {  
    analogWrite(9,100);  
  }  
  else {  
    analogWrite(9,0);  
  }  
}
```



Reading analog signals

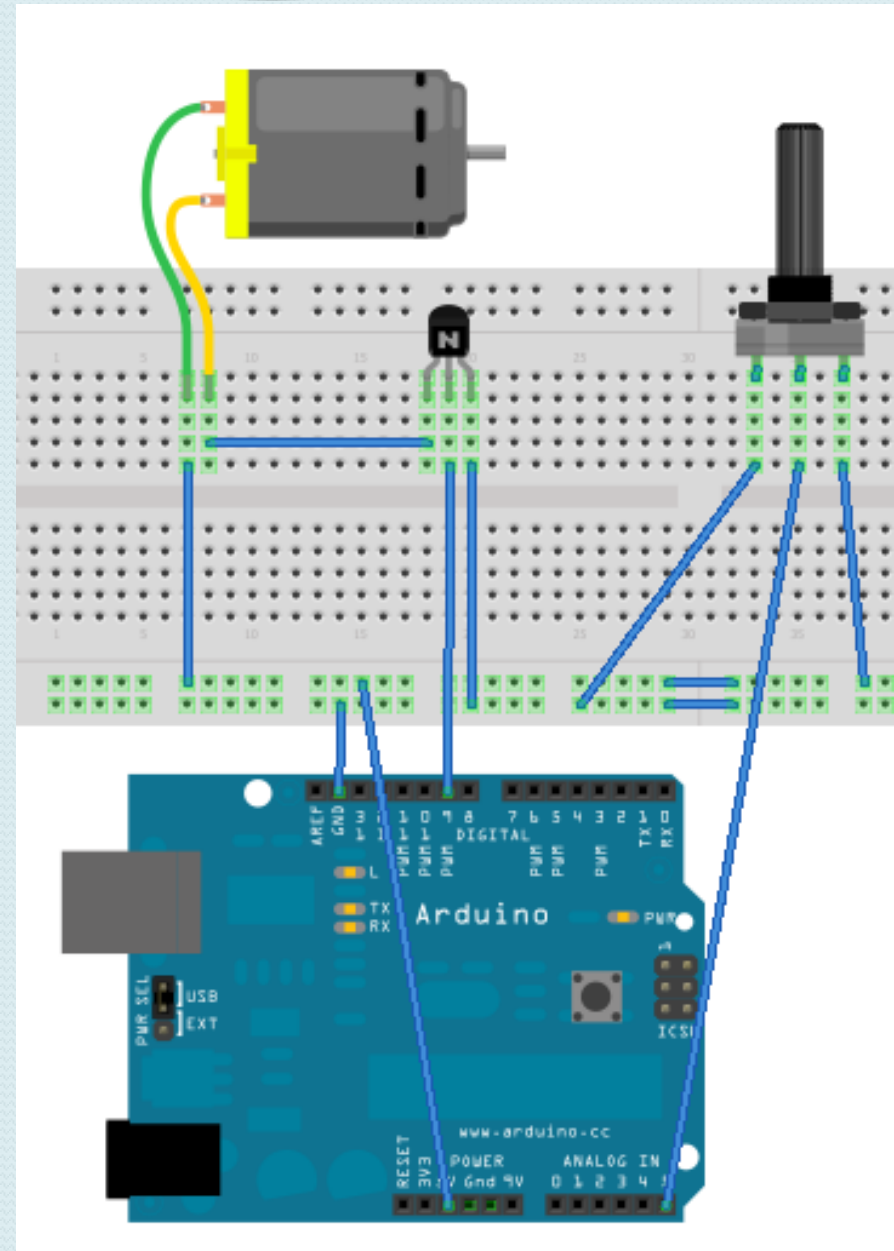
- Arduino has 6 analog inputs
- Read with
 - `int value = analogRead(pin);`
 - Pin is 0-5 or A0-A5
 - Value is from 0 to 1023 representing voltage of 0-5v

Potentiometer

Speed control using potentiometer

```
void setup() {  
  pinMode(9,OUTPUT);  
}
```

```
void loop() {  
  int light = analogRead(5);  
  analogWrite(9,light/4);  
}
```



Light dependant resistor

Speed control according to light level

```
void setup() {  
  pinMode(9,OUTPUT);  
}  
  
void loop() {  
  int light = analogRead(5);  
  analogWrite(9,light/4);  
}
```

