UCLA BMES
Build Team Cheat Sheet

Circuit Components

 Diode

 Capacitor

 Resistor

 DC Voltage Source

 AC Voltage Source

 Ground

Arduino Specific Functions

`void setup()`
- Runs once to set up Arduino

`void loop()`
- Runs after setup continuously

`pinMode(pinNum, INPUT/OUTPUT)`
- Initializes pin on Arduino as input or output

`Serial.println(x);`
- Prints "x" on a new line in Serial Monitor

`digitalWrite(pinNum, LOW/HIGH)`
- Outputs to pin

`digitalRead(pinNum)`
- Reads input from pin and assigns to value

`delay(numOfMilliseconds)`
- Stops code for specific number of milliseconds

`millis()`
- Returns number of milliseconds the code has run for

`analogRead(pin)`
- Returns a reading ranging from 0-1023

`analogWrite(pin, value)`
- Can write a value from 0-255

`map(value, fromLow, fromHigh, toLow, toHigh)`
- Useful for rescaling range of analog input values to range of analog output values (Ex. map (value, 0, 1023, 0, 255))

`random(start of range, end of range)`
- Output random number between the defined number range

C++ Syntax

Boolean operators:

`A<B / A<=B`
- Returns "true" if A less than B / A less than or equal to B

`A>B / A>=B`
- Returns "true" if A greater than B / A greater than or equal to B

`A==B`
- Returns "true" if A equal to B

`A!=B`
- Returns "true" if A not equal to B

`A||B`
- Returns "true" if A **or** B is true

`A&&B`
- Returns "true" if A **and** B is true

If statements:

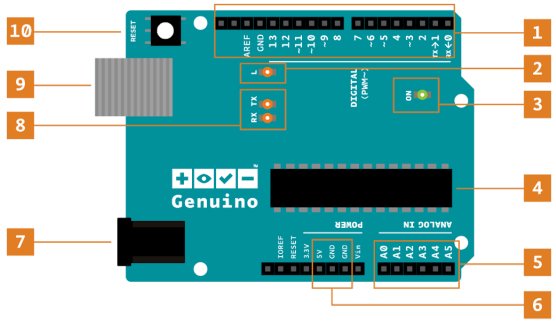
```
if (condition)
{
   // code
}
else if (condition)
{
    // code
}
else
{
    // code
}
```

Loops:

```
for (initialize; condition; increment)
{
    // code
}
```
- "Initialize" is run once at the start
- Loop continues until "condition" is not satisfied
- "Increment" is run at the end of each iteration of the loop

```
while (condition)
{
    // code
}
```
- Continuously loops while condition is satisfied

Arduino Components



For Analog:

- Input must use Analog In pins (i.e. A0, A1, etc.)
- Output must use PWM (pins with ~ in front of number)

Processing Specific Functions

Coordinate system:

- Y-axis is reversed on the vertical
- Origin is the top-left point

`size(400, 400)`

- Sets window size in pixels

`background(192, 64, 0)`

- Sets background color RGB values

`stroke(255)`

- Sets line color, with 0 → black and 255 → white

`line(150, 25, 270, 350)`

- Draws a line between coordinates (150, 25) and (270, 350)

`point(25, 45)`

- Draws a point at (25, 45)

For Arduino:

`class name;`

- To declare an object of a certain class

```
Servo pauley;
Arduino boelter;
```

`name = new class;`

- To initialize a new object in Java

```
pauley = new Servo;
boelter = new Arduino
(this,_port_,_baud rate ex.57600_);
```

`name.function();`

- To access specific class functions

```
pauley.write(angle);
boelter.AnalogRead(pin);
```

`class.variable;`

- To access specific class constants

```
Arduino.OUTPUT;
Arduino.HIGH;
```

Declare an Arduino object called myArduino
In void setup() insert:

```
printArray(Arduino.list()); // Lists USB ports, find the one connected to Arduino
myArduino = new Arduino(this, Arduino.list()[0], 57600);    // May need to change index value (where it says [0]) based on printed array
```

This initializes your myArduino object and links your Arduino to Processing

Processing Troubleshooting for Mac

Arduino port needs to be manually selected on Mac

- If you are having issues connecting , manually input the port number
- Check the Arduino IDE under “Tools” > “Port” to see which port number the Arduino is connected to

Setting Up Standard Firmata in Arduino IDE

THESE STEPS NEED TO BE DONE EVERY TIME BEFORE SWITCHING TO PROCESSING FROM ARDUINO IDE

To use Firmata in Arduino:

1. In the Arduino IDE, got to File and then Examples
2. Under Firmata, select StandardFirmata
3. Upload code to your Arduino

THESE LINES NEED TO BE INCLUDED AT THE BEGINNING OF EVERY PROCESSING CODE USING ARDUINOS

```
import processing.serial.*;
import cc.arduino.*;
import org.firmata.*;
```

OR

Sketch -> import library -> Serial or Firmata

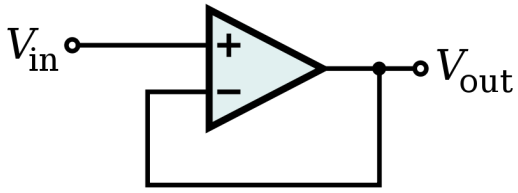
Tips:

- When declaring analog pins, just use the pin number without “A” in the front
  - Processing will differentiate pin type by function called for pin
- Arduino specific constants => Arduino.___
  - Use general Arduino class to call
  - Ex. HIGH, LOW, OUTPUT, INPUT
- Arduino specific functions => myArduino.___
  - Use Arduino object you created to call function

Op-Amp Circuits

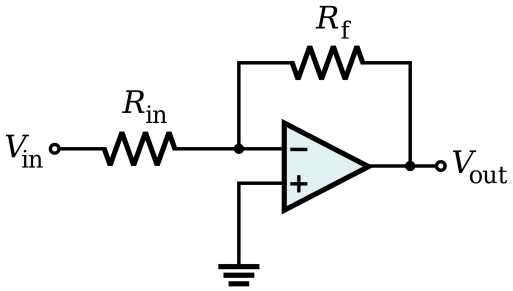
Voltage Follower

- Keeps the voltage the same
  - Vout = Vin
- Gain = 1 (unity)



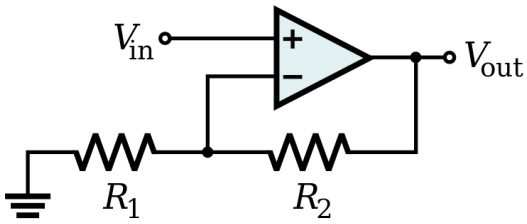
Inverting Amplifier

- Inverts and amplifies the signal
- Gain = - R_f/R_{in}



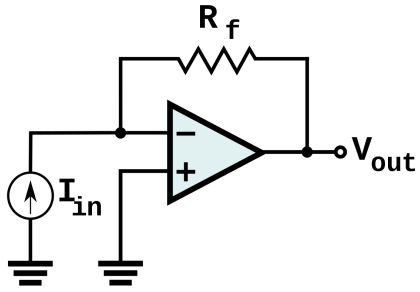
Non-Inverting Amplifier

- Amplifies the signal
- Gain = 1 + (R_2/R_1)



Transimpedance Amplifier

- Converts current into voltage
- V_{out} = - I_{in} * R_f



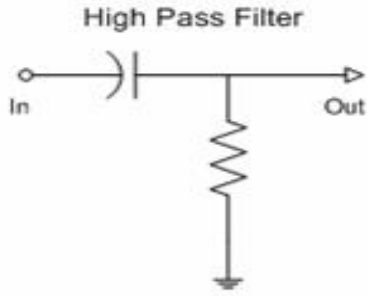
Op-Amp Negative Feedback

Filters

$$f = \frac{1}{2\pi RC}$$

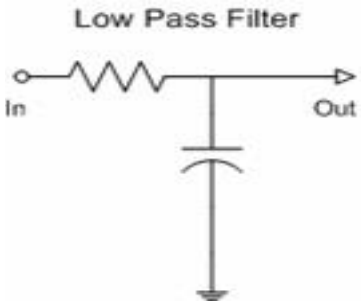
High Pass Filters

- Capacitor placed before resistor
- DC and low frequencies can't pass



Low Pass Filters

- Capacitor placed after resistor
- High frequencies can't pass
- Noise is generally high frequency

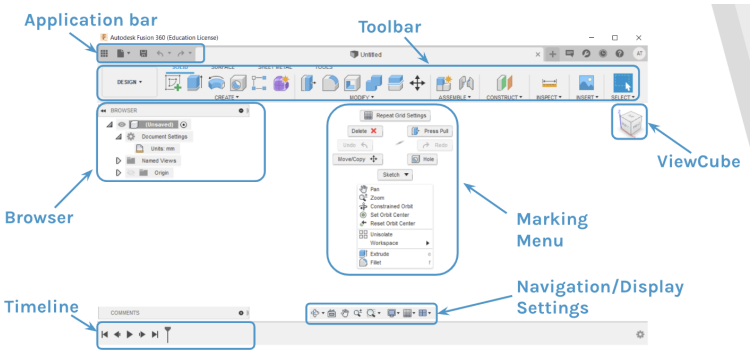


Helpful resources to understand Op-Amp:
Op-Amp Negative Feedback
Voltage Follower
Inverting Amplifier
Non-inverting Amplifier
Transimpedance Amplifier

Tips:

- Arduino's signal reading ideal resolution= 5V/2^10 bits
  - Theoretic value not particularly high already, often in reality even worse
  - Might be unable to differentiate close to 0 values, easily affected by noise
- Good resistor range for op-amp: 1-100K ohm

Fusion 360



Setting Units

Under Browser ➔ Document Settings ➔ Units: mm

Shortcuts

S	Search	C	Circle
L	Line	T	Trim
R	Rectangle	E	Extrude
F	Fillet		

Function	**Mouse**	**Trackpad**
Zoom	Scroll Wheel	Pinch on trackpad
Pan View	Hold Middle Mouse Button	
Orbit View	Hold Shift + Hold Middle Mouse Button	

Autodesk Eagle

Video Tutorials: Autodesk Webinars

- Getting Started Pt. 1
- Getting Started Pt. 2

Parts Libraries for Arduino Components

- You'll need these if you want to use an Arduino chip or Arduino Nano
- Arduino Eagle Cad Library
- Adafruit Library

Icons

Select		Net (draw wire)	
Copy		Invoke	
Add Part			

Notes

Moving parts

- left click the + in its center and drag

Rotating parts

- left click and hold, then right click
- Or use the rotate tool
- To mirror, click your scroll wheel

Search Window (search bar input)

- Exact match search (not like Google!)
- Must use wildcards
    - ? = single character
    - * = multiple characters

Helpful Commands

esc	go back to add part menu
Right click	rotate
Scroll wheel click	mirror
Copy function	left toolbar
Red stop button	top toolbar