

---

# **Mind + Arduino**

## Tutorial for Beginners

---

# 0 Getting Started

**Windows 7 & XP   Windows 8**

4-9

10-16

**Intro**

17-20

# 1

## DIGITAL INPUT / OUTPUT

**Intro   Blink   Circuit   LED   Button   LED Matrix**

21-22

**Blink**

23-26

**Circuit**

27-29

**LED**

30-32

**Button**

33-35

**LED Matrix**

36-38

# 2

## ANALOG INPUT

**Intro   Rotation Sensor   Light sensor   Temp Sensor**

39-40

41-45

**Light sensor**

46-49

**Temp Sensor**

50-52

# 3

## ANALOG OUTPUT / PWM

**Intro   LED(PWM)   Servo   Capacity Sensor   IR Sensor**

53-54

**LED(PWM)**

55-58

**Servo**

59-60

**Capacity Sensor**

61-65

**IR Sensor**

66-70

# 0 Getting Started

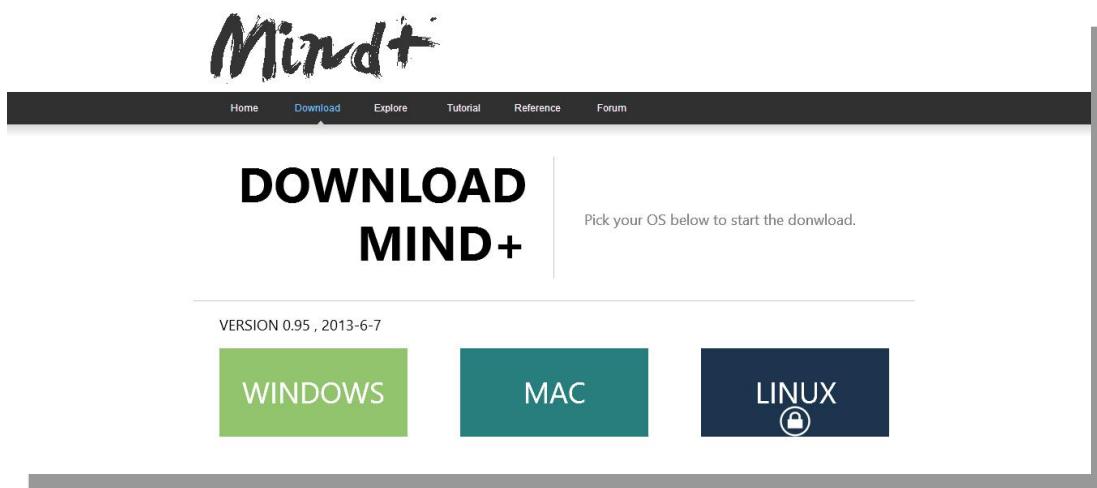
[Installation for Windows 7 & XP](#)

[Installation for Windows 8](#)

[Intro](#)

# Installation Guide for Windows 7 and XP Users

**STEP 1** . Enter <http://mindplus.cc/download.html> website, download the version corresponding to your operation system.



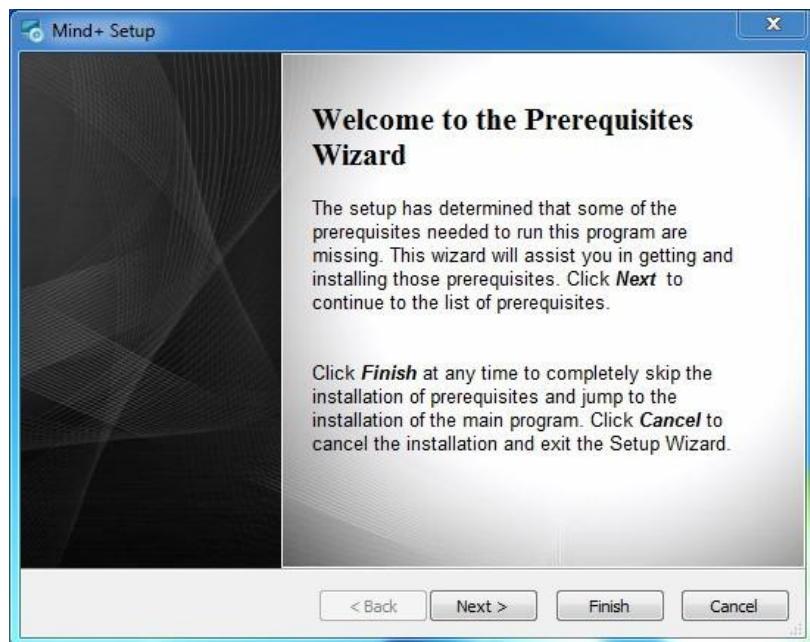
**STEP 2** After finishing the download, open the setup program.

\* Only if you install the Driver correctly, the Arduino board could run successfully. If you installed the old version of Mind+, please uninstall it before install the new one. \*

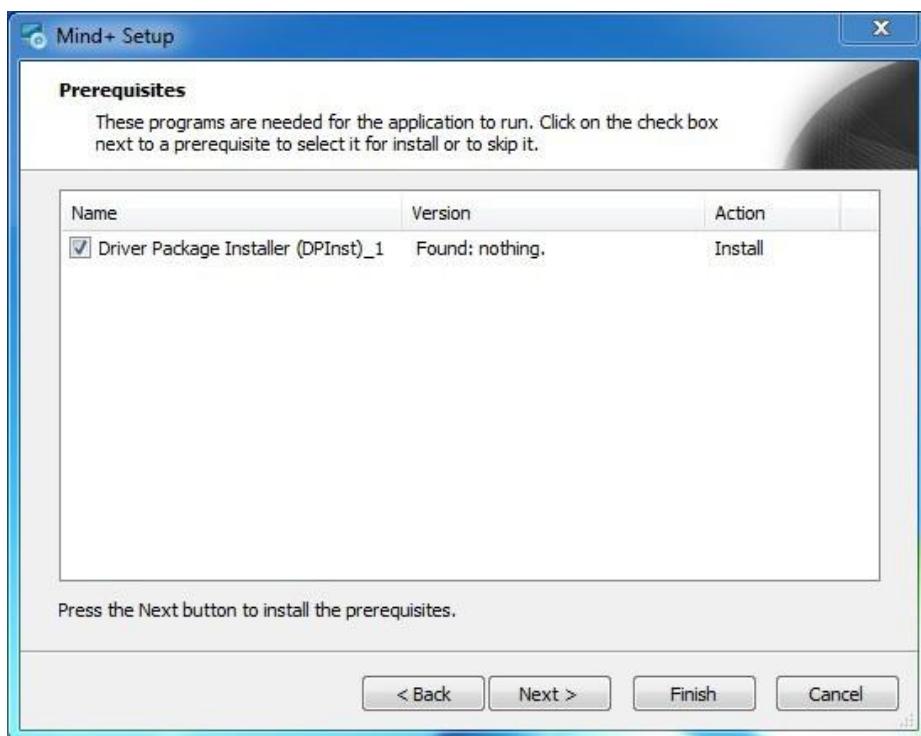
名称	修改日期	类型	大小
IR	2013/8/19 17:28	文件夹	
PORT	2013/8/19 17:29	文件夹	
Mind+V0.952_Win.exe	2013/8/13 14:50	应用程序	65,818 KB

Click Next Step on the installation interface.

\* If you ignored this step or clicked Finish, it may cause problems to the connection of Mind+ and the Arduino board. \*



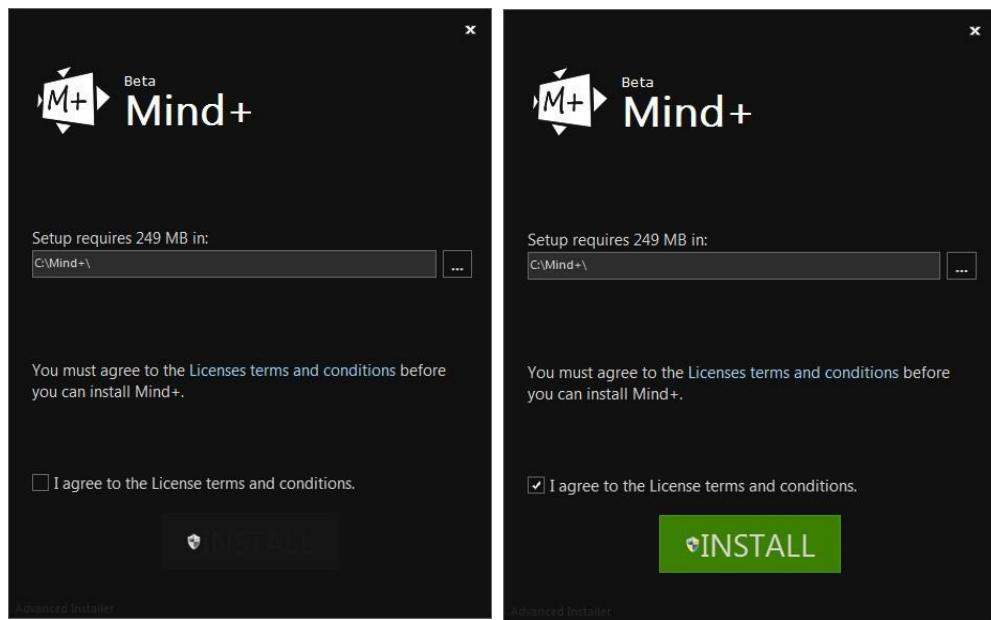
Tick off Driver Package Installer, and then click Next Step.

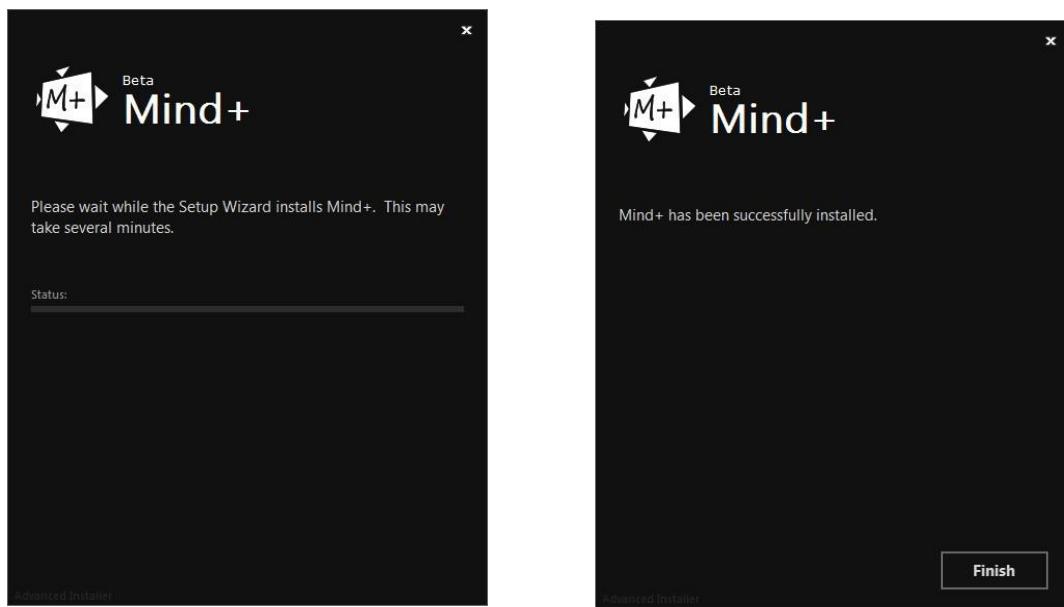


Find the Driver Installation Guide, and then click Next Step. If you could not find it, please minimize other windows. And click Finish.



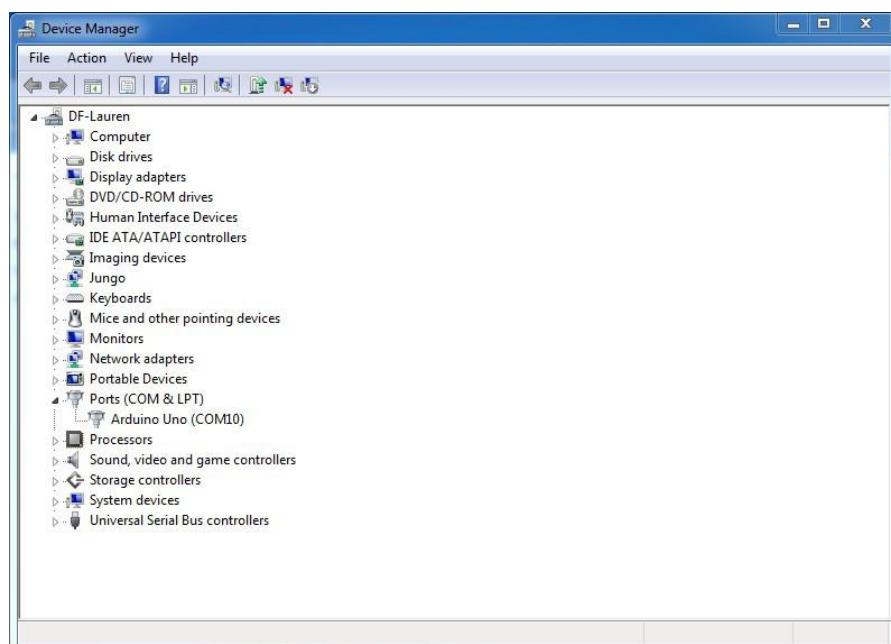
Tick off the Agreement, click Install, and then click Finish.





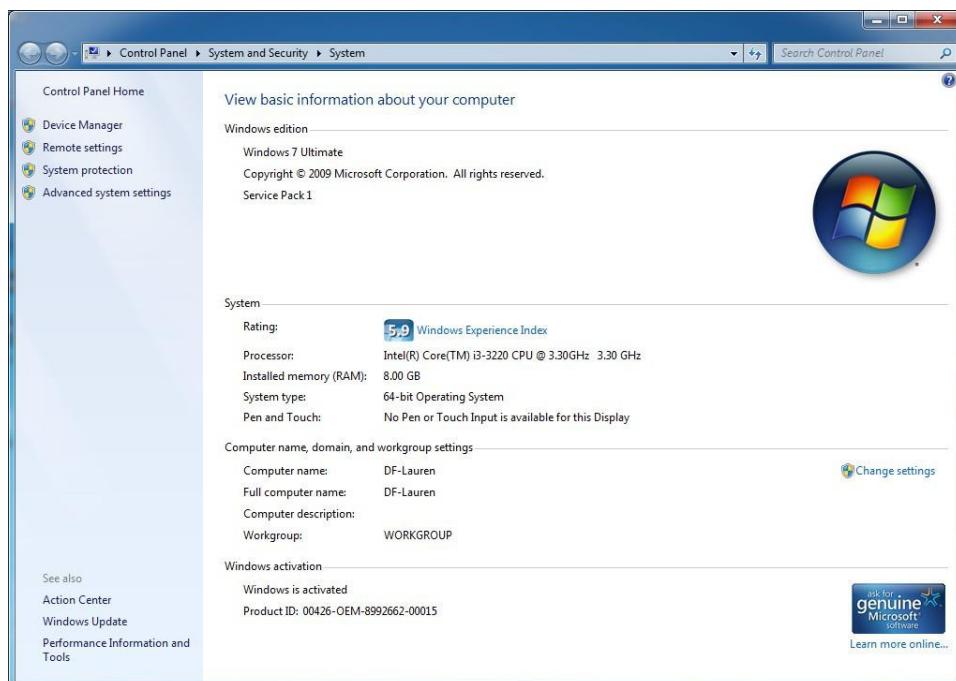
### STEP 3 . Check whether the Driver is installed correctly in the Device Manager.

If you installed it correctly, the Arduino Port would appear in the Port Section (COM and LPL), which means you have already installed Mind+ successfully. **If there were not any Arduino Port or there were Arduino Port but with exclamation mark in the Device Manager, you should turn to the Installation Guide below and install it manually.**



## Manual Installation

**STEP 1** Check the type of your operation system in the Control Panel or the Computer Properties, such as 32-bit operation system or 64-bit operation system.



**STEP 2** Install Driver, open X:\Mind+\drivers

If your computer had a 32-bit operation system, please run  
X:\Mind+\drivers\dpinst-x86.exe

If your computer had a 64-bit operation system, please run  
X:\Mind+\drivers\dpinst-amd64.exe

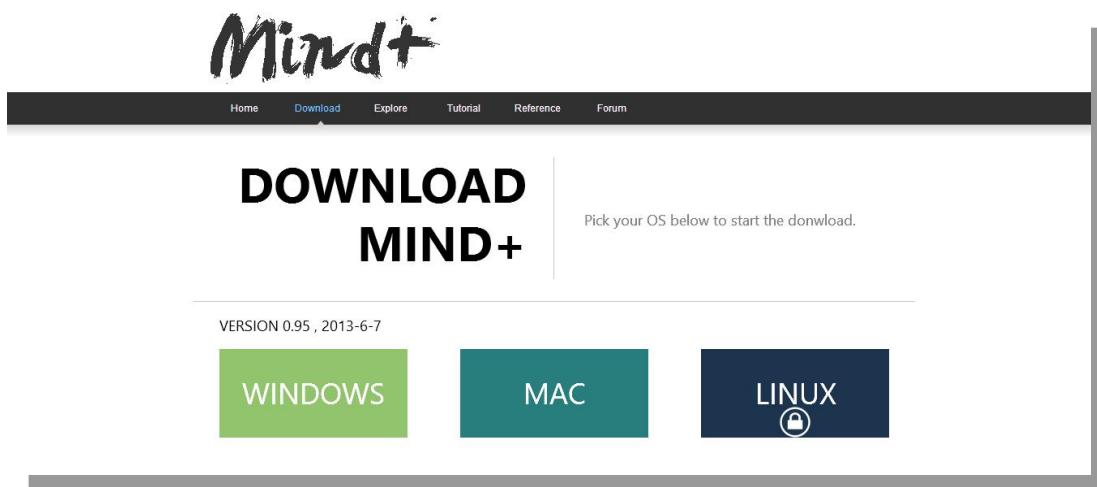
名称	修改日期	类型	大小
FTDI USB Drivers	2013/8/22 15:21	文件夹	
arduino.cat	2013/6/7 11:14	安全目录	7 KB
arduino.inf	2013/6/7 11:14	安装信息	6 KB
dpinst-amd64.exe	2013/6/7 11:14	应用程序	1,024 KB
dpinst-x86.exe	2013/6/7 11:14	应用程序	901 KB
Old_Arduino_Drivers.zip	2013/6/7 11:14	压缩(zipped)文件...	14 KB
README.txt	2013/6/7 11:14	文本文档	1 KB

### STEP 3 Click Next Step, wait a few seconds, and then click Finish.



# Installation Guide for Windows 8 Users

**STEP 1** . Enter <http://mindplus.cc/download.html> website, download the version corresponding to your operation system. Mind+ is available in Windows and MAC version.



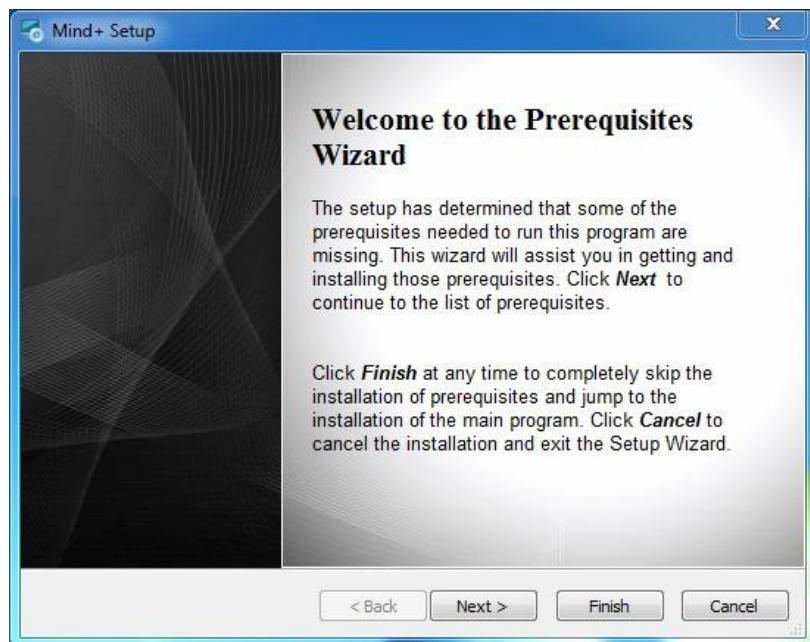
**STEP 2** After finishing the download, open the setup program.

\* Only if you install the Driver correctly, the Arduino board could run successfully. If you installed the old version of Mind+, please uninstall it before install the new one. \*

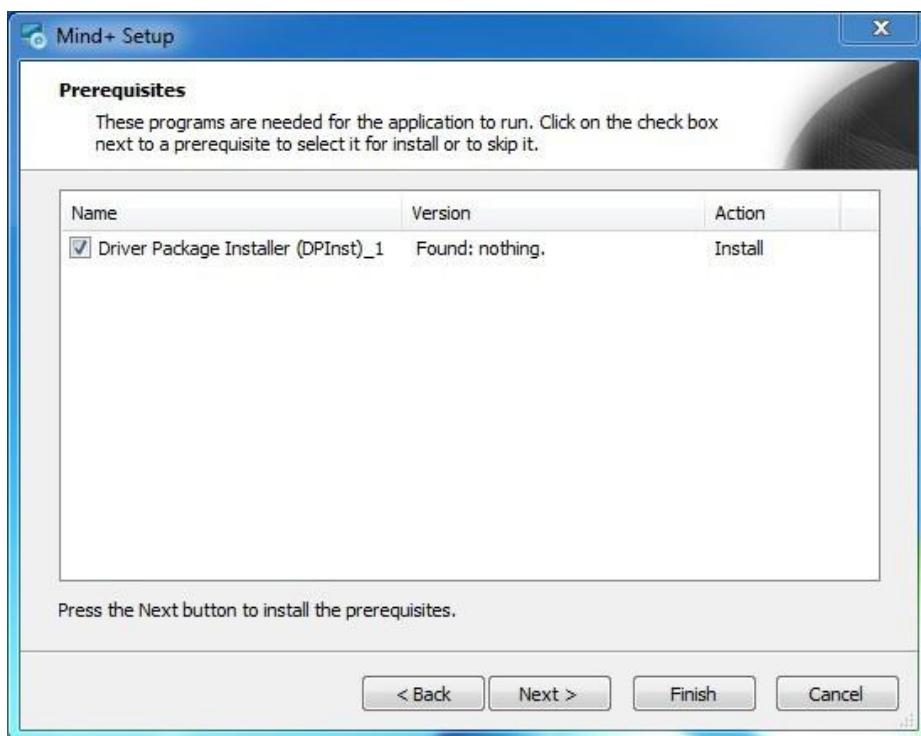
名称	修改日期	类型	大小
IR	2013/8/19 17:28	文件夹	
PORT	2013/8/19 17:29	文件夹	
Mind+V0.952_Win.exe	2013/8/13 14:50	应用程序	65,818 KB

Click Next Step on the installation interface.

\* If you ignored this step or clicked Finish, it may cause problems to the connection of Mind+ and the Arduino board. \*



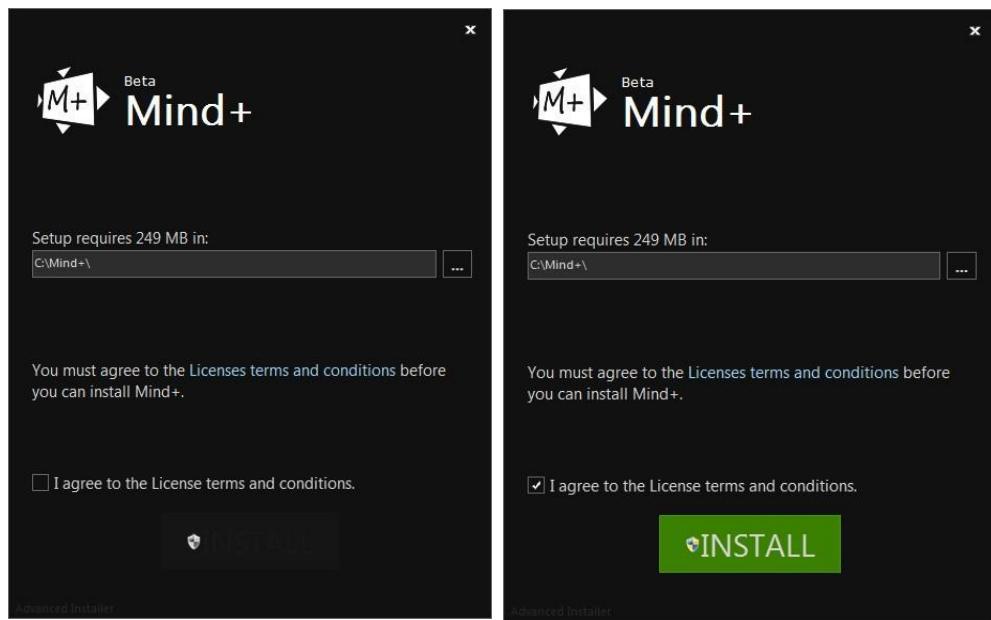
Tick off Driver Package Installer, and then click Next Step.

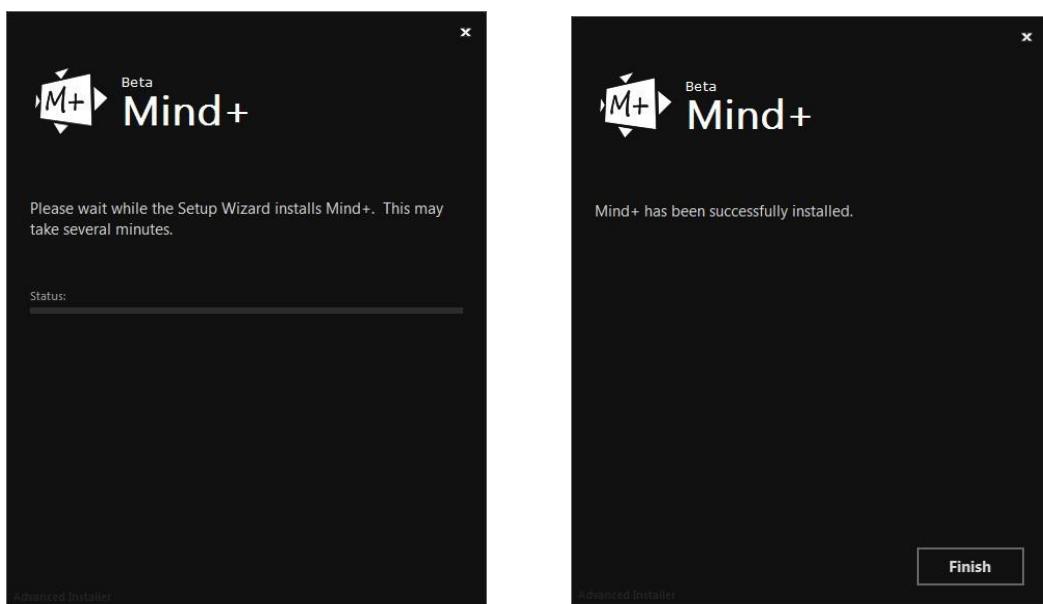


Find the Driver Installation Guide, and then click Next Step. If you could not find it, please minimize other windows. And click Finish.



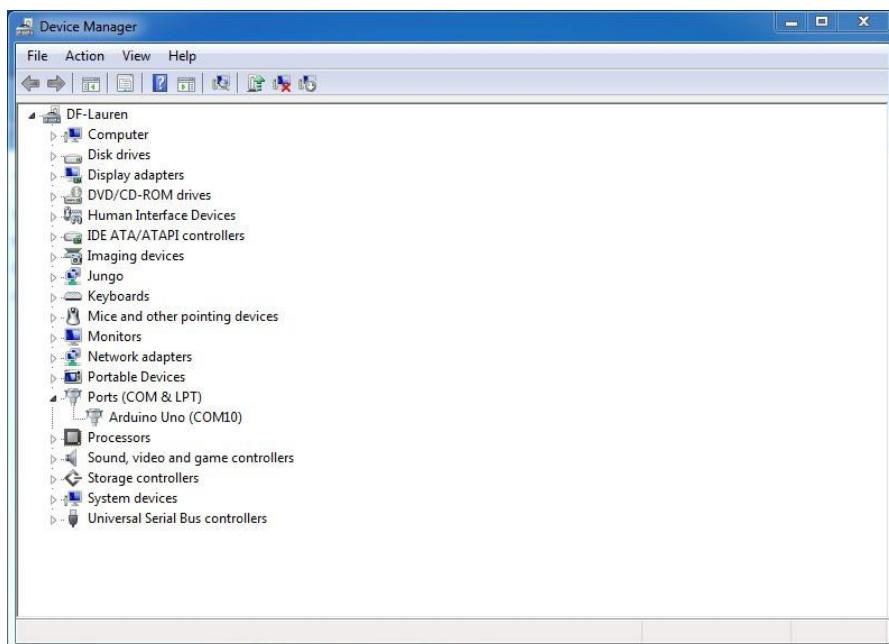
Tick off the Agreement, click Install, and then click Finish.





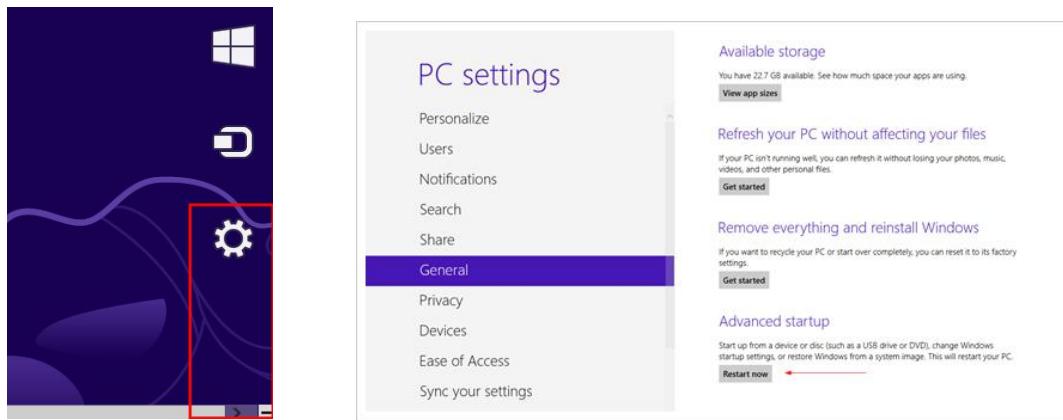
### STEP 3 . Check whether the Driver is installed correctly in the Device Manager.

If you installed it correctly, the Arduino Port would appear in the Port Section (COM and LPL), which means you have already installed Mind+ successfully. **If there were not any Arduino Port or there were Arduino Port but with exclamation mark in the Device Manager, you should turn to the Installation Guide below and install it manually.**

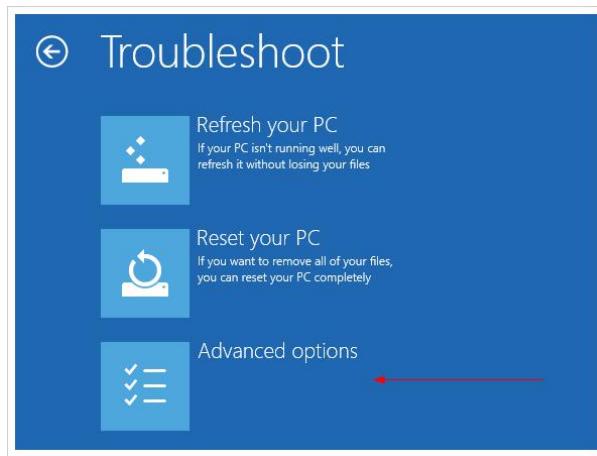


## Manual Installation

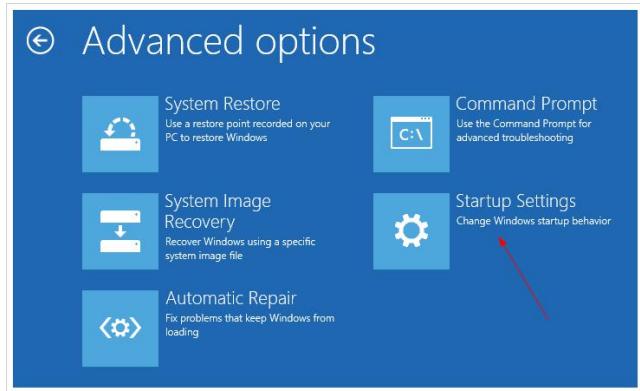
**STEP 1** Move the cursor over the top or bottom right corner or the screen and it will show extra options on the right side of the screen and click the settings button .  
Change PC settings > General > Advanced startup



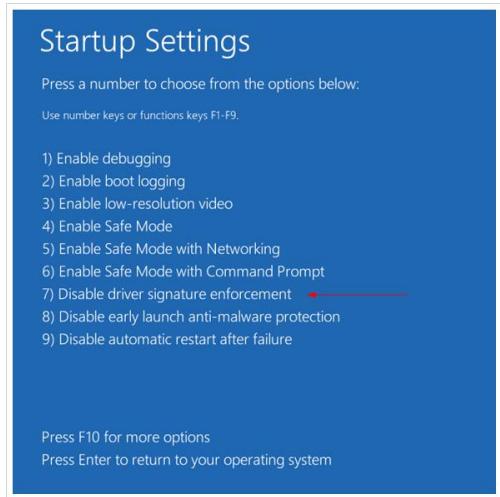
**STEP 2** Choose and option > Trouble shoot > Advanced option



### STEP 3 Advanced options > startup settings > restart



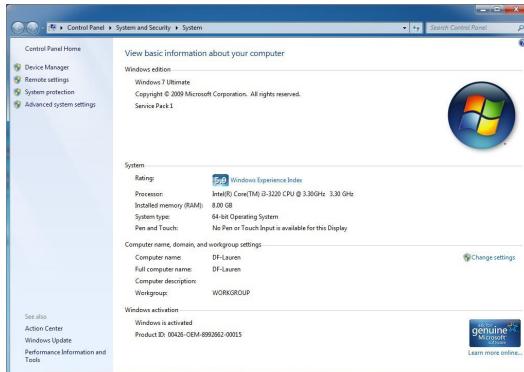
**STEP 4** The PC will now restart. Upon startup hit 7 on your keyboard to boot into Windows 8 with the driver signature check disabled.



**STEP 5** Now you can stall Arduino driver after restart. Click Install this driver software anyway when the Windows Security pops up.

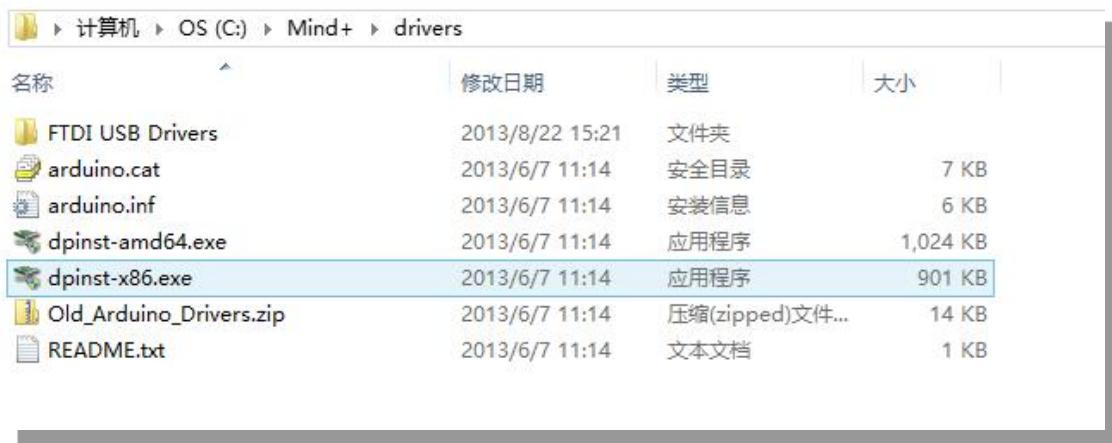


## STEP 6 Check the type of your operation system in the Control Panel or the Computer Properties, such as 32-bit operation system or 64-bit operation system.



## STEP 7 Install Driver, open X:\Mind+\drivers

If your computer had a 32-bit operation system, please run X:\Mind+\drivers\dpinst-x86.exe



## STEP 8 Click Next Step, wait a few seconds, and then click Finish.

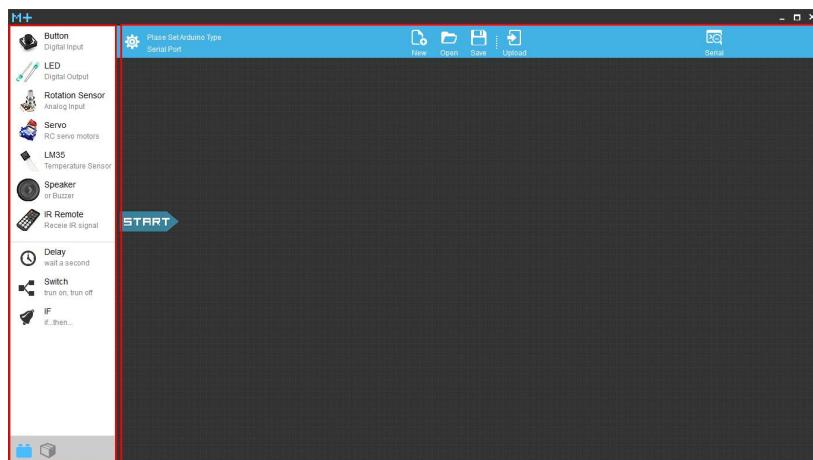


# Mind+ Intro

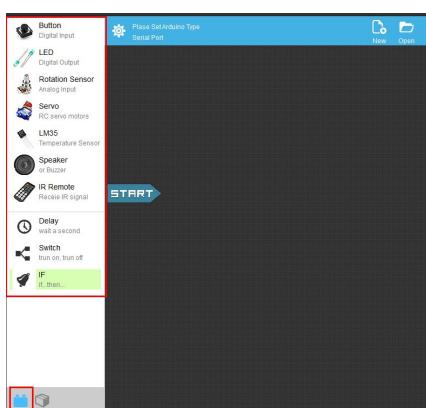
## Mind+ Interface:

block list

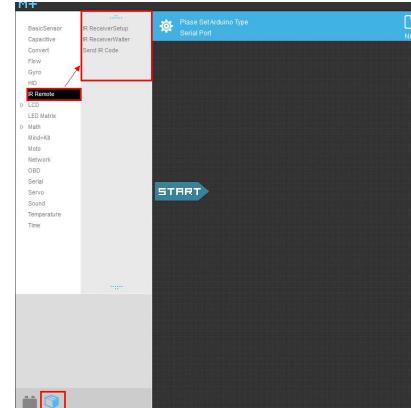
Work area



## block List :

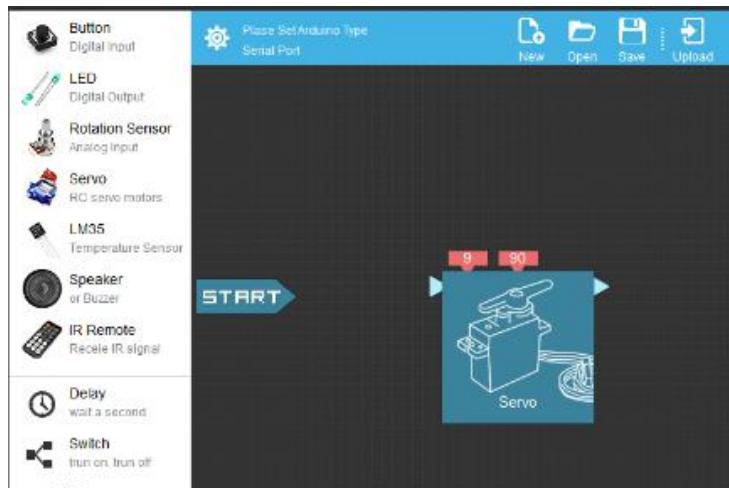


Basic block list

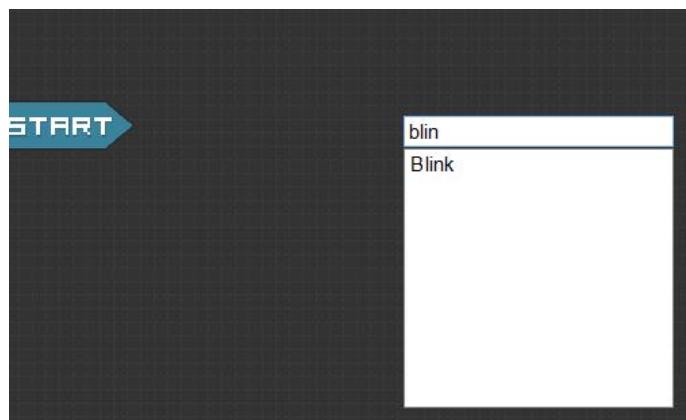


Advanced block list

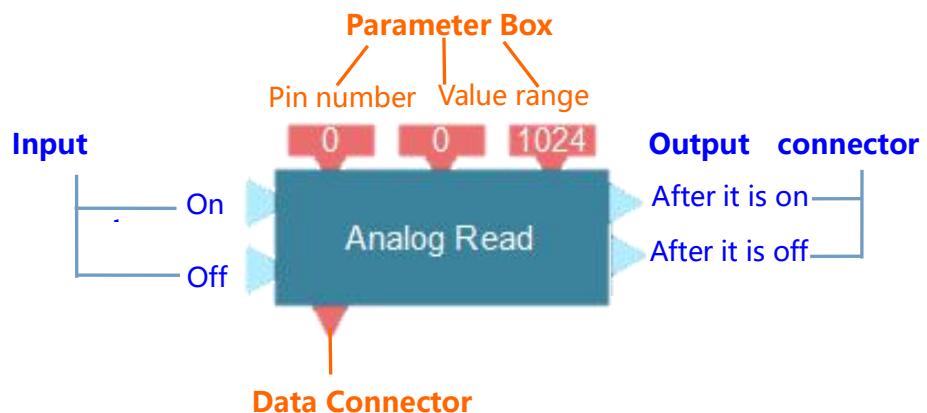
**Add block:** drag block from the list and drop it in the work area.



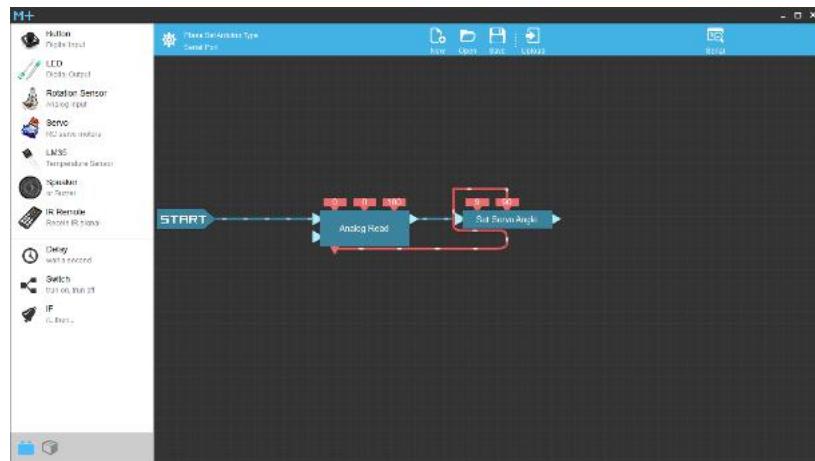
Or double click in the work area.



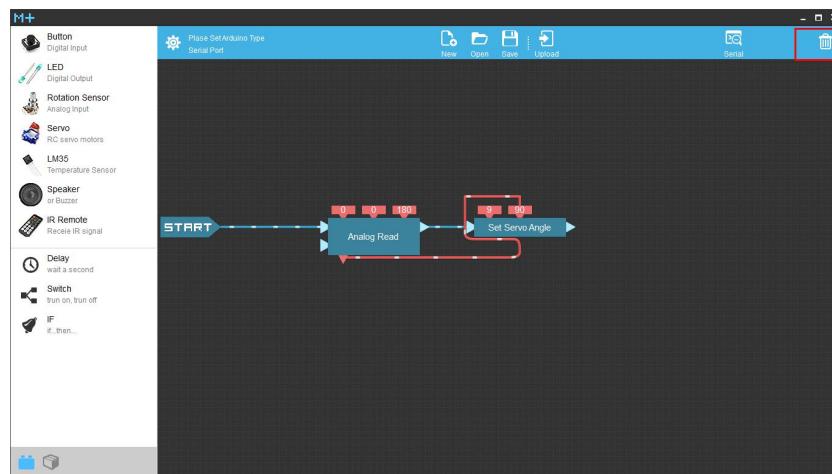
## Structure of blocks



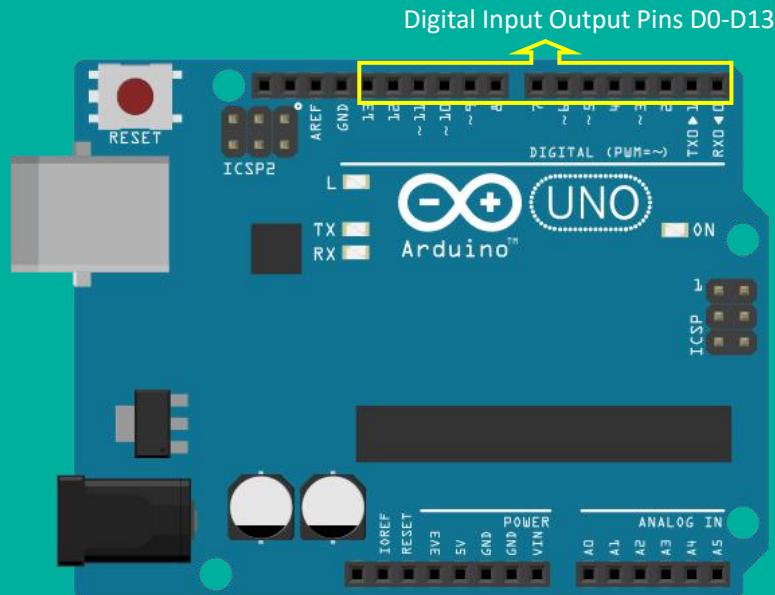
**Connect blocks:** Press a connector and drag it to another connector. Connectors only connect when in same color. Blue connectors simulate flow of energy , red connectors simulate flow of data. Color of line depends on color of pins. START simulates power source,



**Cancel block or line :** Choose a block or a line, press DELETE or click on the BIN icon on the upper right side.



# 1 DIGITAL INPUT / OUTPUT



If you closely look at the Arduino board, you will find many of the devices, such as LEDs and pushbuttons, have only two possible states: on and off, or as they're known to the Arduino board, "HIGH" (5 volts) and "LOW" (0 volts).

These digital pins on Arduino board are labeled DIGITAL, from D0-D13. But we recommend not to use D0 and D1 which are mainly used to transmit data.

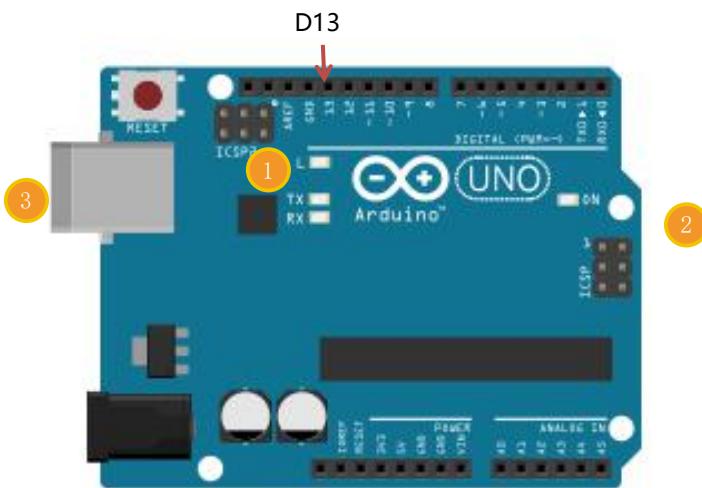
In the following projects, we will start with some very common sensors and actuators to practice, for instance, a button-controlled LED(Project#4) and 8x8 LED matrix for heartbeat animation (Project#5).

# 01 Blink

## Testing Arduino Board

Blinking a on board LED is a very simple programming for Arduino which is often used to test if the Arduino board has some errors.

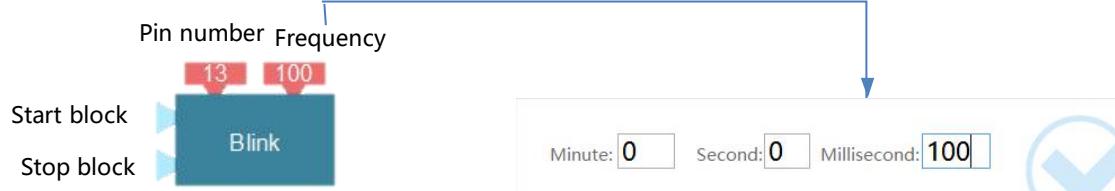
### Arduino



- 1 L, onboard LED, with default pin in D13 (D stands for Digital, which means digital pin)
- 2 ON indicates whether the board is powered.
- 3 Power source, where you can plug in with a USB cable to connect Arduino with a computer.

### Mind+

#### block description

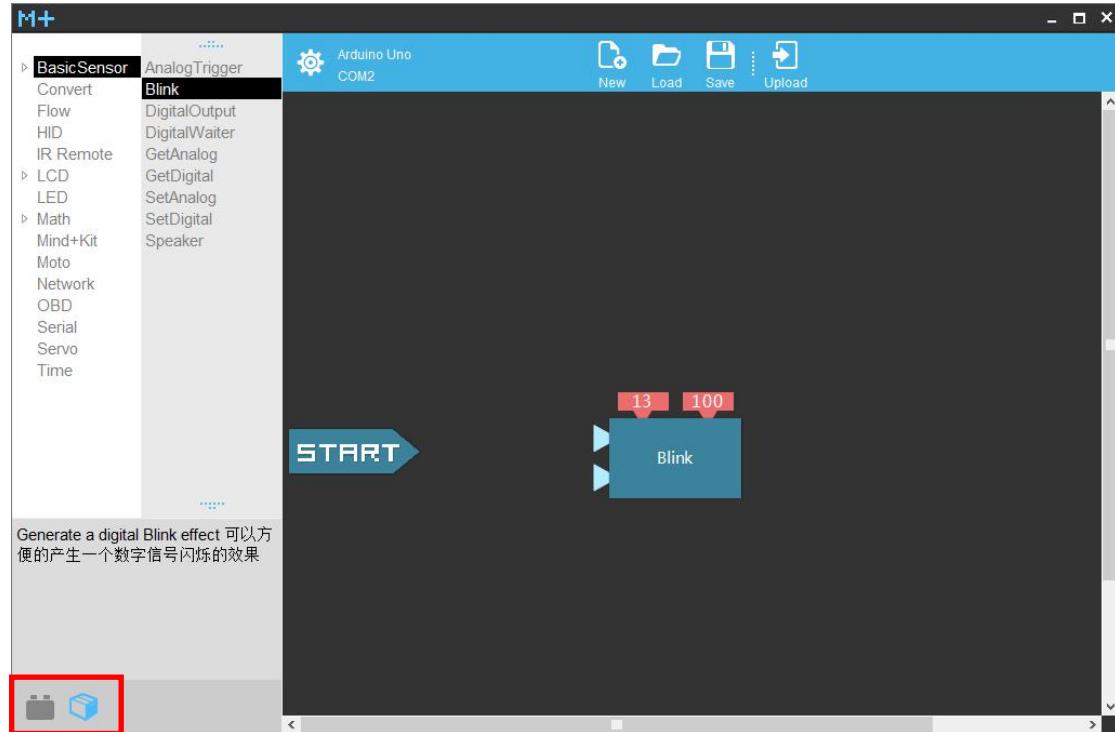


#### Blink

- + **Location:** advanced block list -- Basic Sensor—Blink
- + **Function:** Blink the on board LED

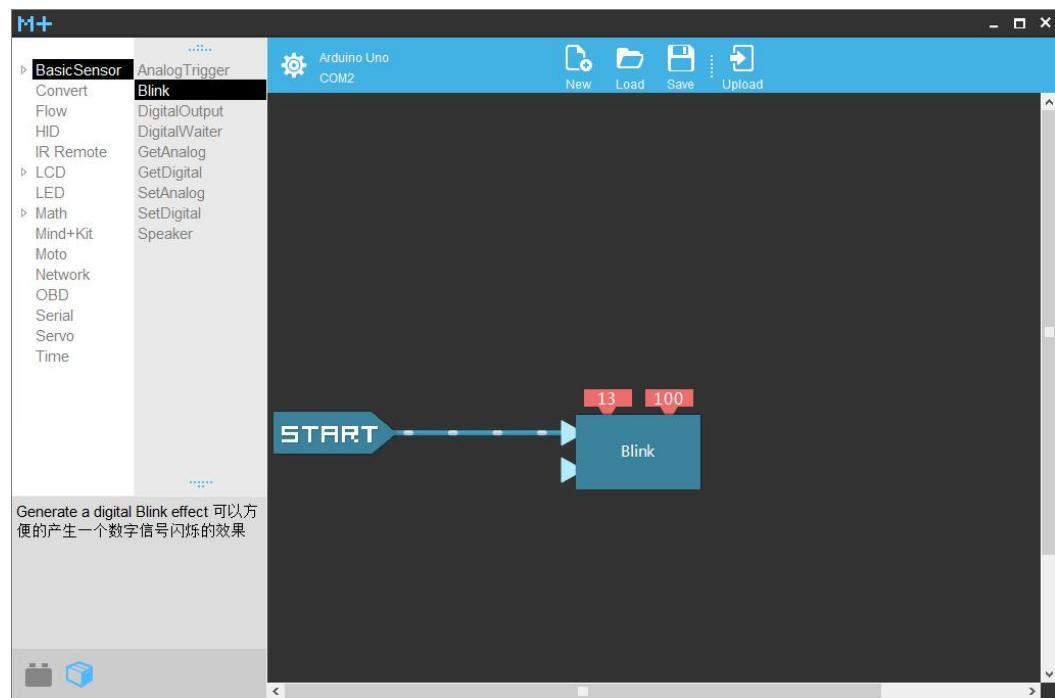
## Mind+ Interface

1. Open Mind+, Find Blink block in the advance block list   he left, and drag it to the operating area on the right. You can change the frequency of blink as you want.

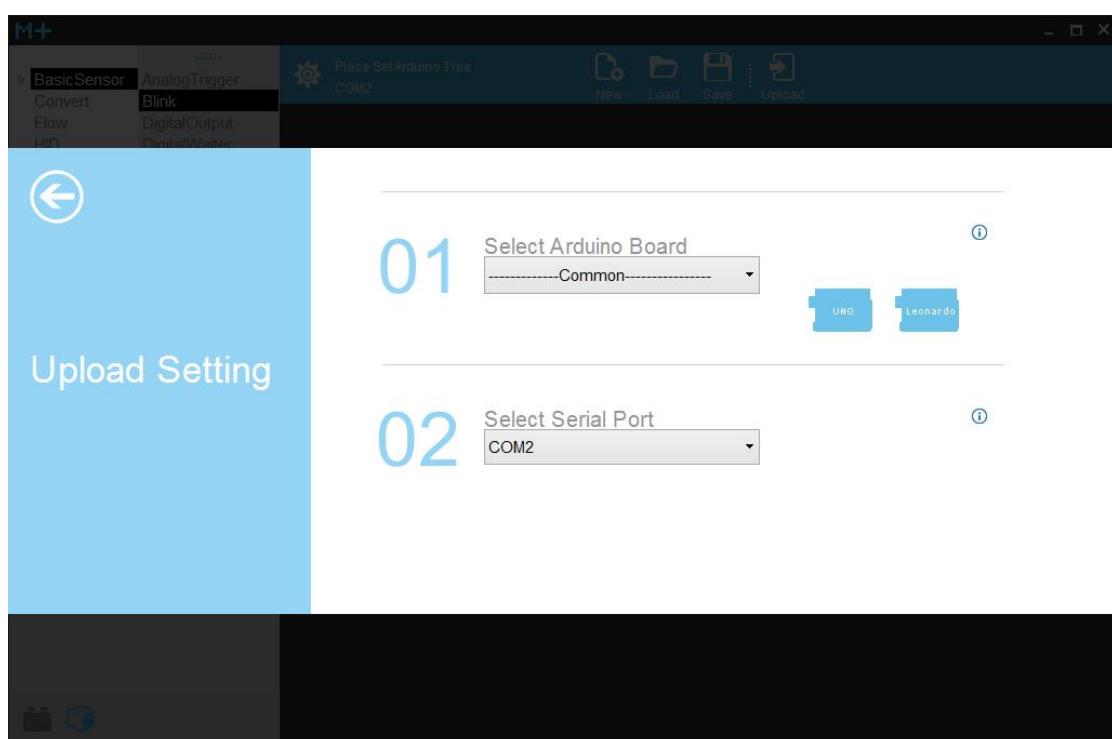
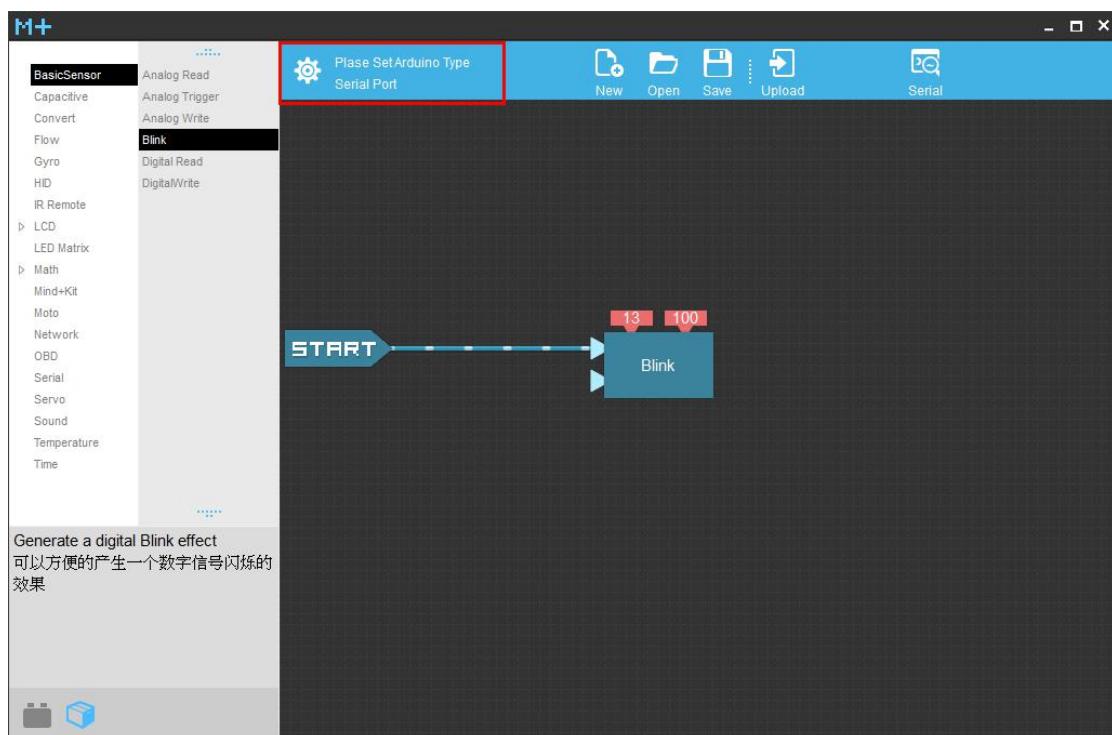


i

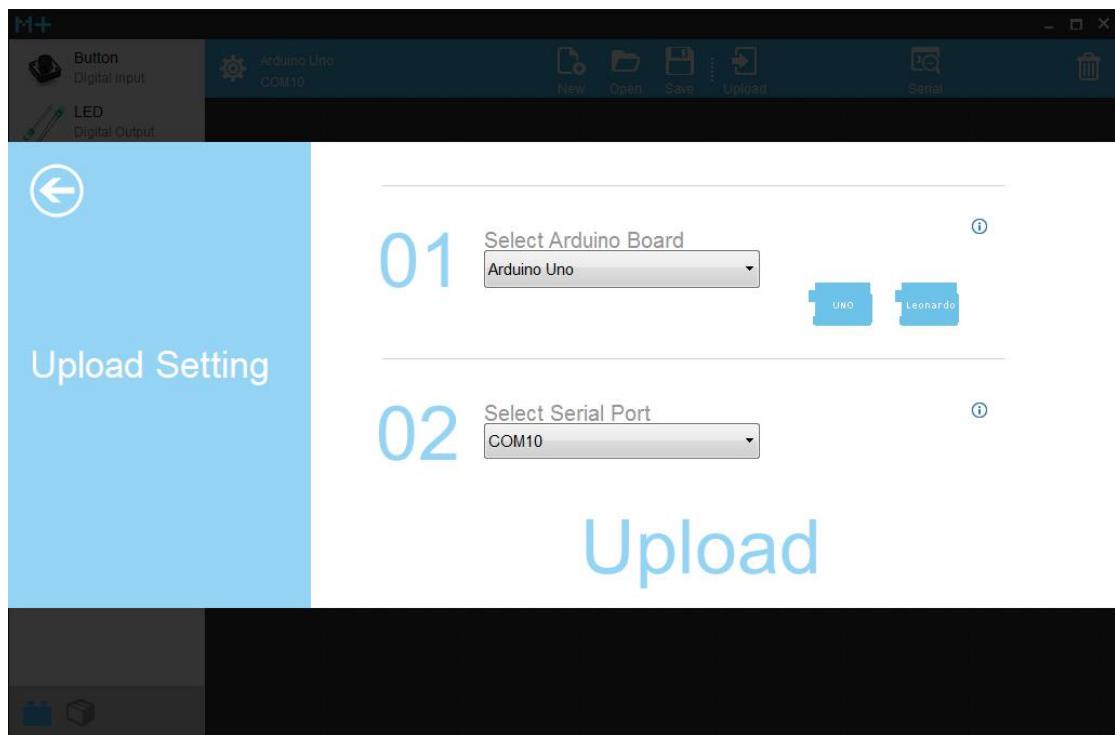
2 . START indicates power source. Left click and drag a line from START, connect it to the upper input connector of Blink.



### 3. Select your Arduino type and Serial Port



4. Click **UPLOAD**.



You will see the L light blinking.

\*If you could not find the serial port, you might need to install Mind+ driver manually. Please check installation guide. [http://www.mindplus.cc/?page\\_id=1241](http://www.mindplus.cc/?page_id=1241)

\*If you found there is a Serial Port Mistake as below, please unplug the USB cable and plug in again.



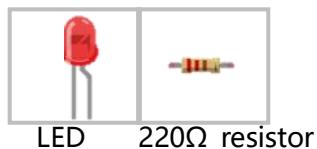
# 02 Circuit

## Building a Complete Circuit

LEDs (light-emitting diodes) are small, powerful lights that are used in many different applications. Almost all modern flat screen televisions and monitors have LED indicators to show they are on or off. In this project, we try to build a complete circuit to understand basic electronics.

### Arduino

#### Part list :



#### A Complete Circuit

- + **Connection** : Arduino has 3 negative pins --GND (Ground) pins and 2 positive pins --3.3V and 5V.
- + **Function** : Electricity flows from positive pole to negative pole.



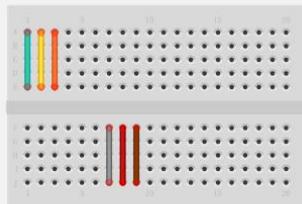
#### LED

- + **Connection:** shorter leg goes to GND and longer leg goes to 5V.

#### Resistor

- + **Connection** : There is no positive or negative polarity on resistor.
- + **Function:** What we get from the Arduino board is 5V, which is above the limited voltage of red light 2.2V. So we need a 220Ω resistor in order not to burn the LED.
- + **Usage:** Bend two legs of the resistor and plug onto the breadboard. Different resistance values has different banding colors. But it is quite complicated to tell so we use multimeter instead.



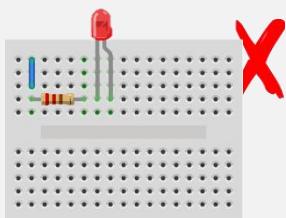


### Breadboard

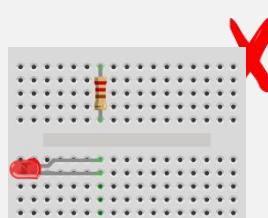
+ **Function:** The surface of the bread board is plastic with plenty of holes, just like bread. There are some metal strips on the bottom. The elements are plugged into the holes according to some rules and get in touch with metal strips so that they are conductive without the need of welding.

+ **Connection:** The five holes in the vertical column on the bread board are conducting.

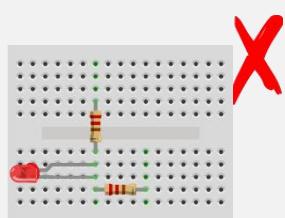
\*\* The green spots in the picture means the holes are conducting



The horizontal columns are non-conducting.



The notch on the bread board is non-conducting.



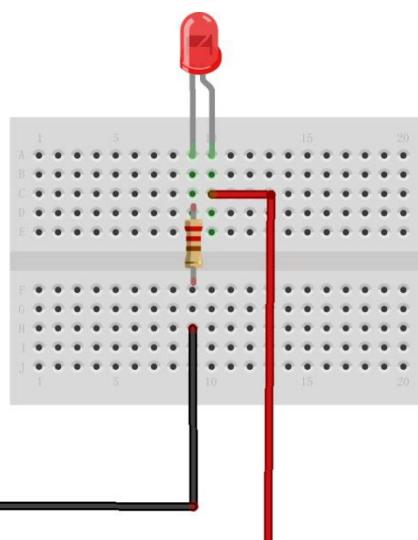
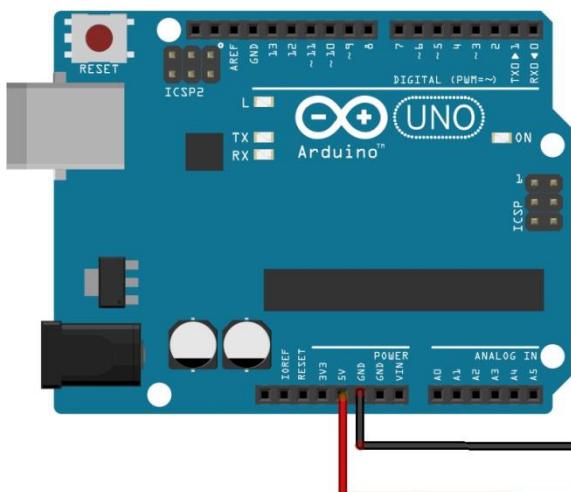
The current flows through the conducting holes instead of

### Jumper Wires

+ **Connection:** This guide will show the wires with different colored insulations for clarity, but using different combinations of colors is completely acceptable. Red wire is connected to 5V or 3.3V while black wire is connected to GND(ground).



## Electrical Circuit

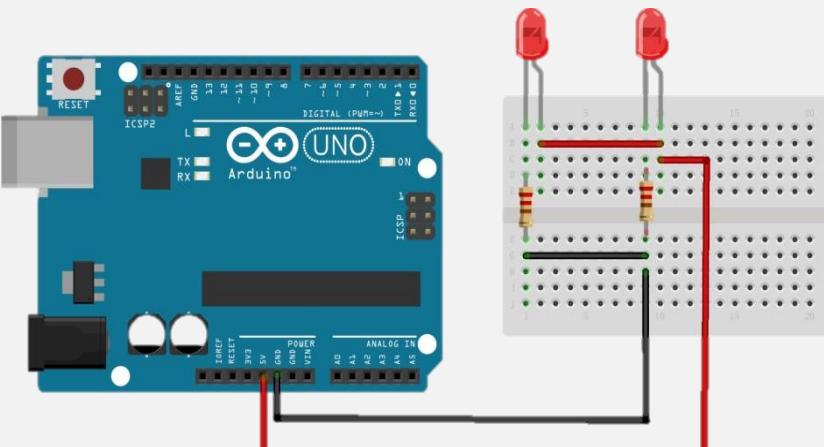


## Extended Exercise: Blinking 2 LEDs

We just used one LED to build a complete circuit. Now can you light up 1 more LED to based on the current circuit?

**\*\* tip: There are only one 5V on the Arduino board, so you need to divide the pin into 2 in order to share with another LED.**

### Reference



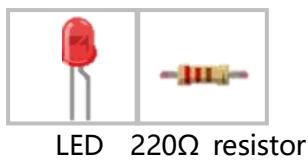
# 03 LED

## Making a LED Blink

In this project, we will use Delay block and Loop to make a LED Blink.

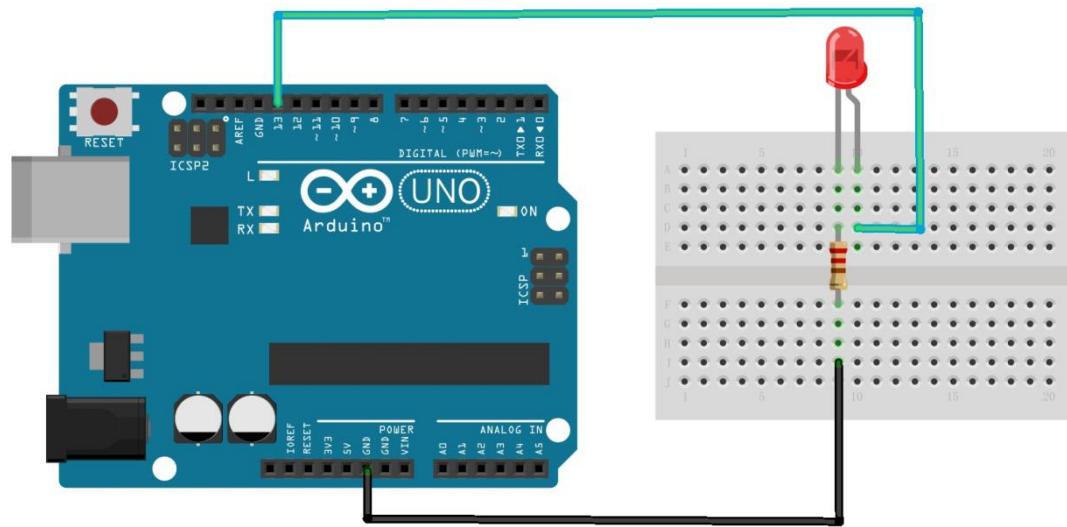
### Arduino

#### Part list



LED 220Ω resistor

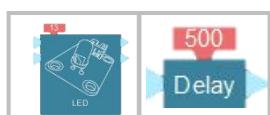
#### Electrical Circuit



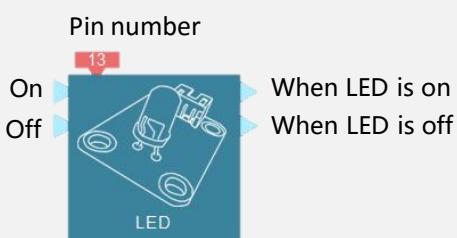
Made with Fritzing.org

## Mind+

### block List



LED      Delay



### LED

+ Location: basic block list -- LED



\*1 second=1000 mille second

Minute: 0      Second: 0      Millisecond: 500

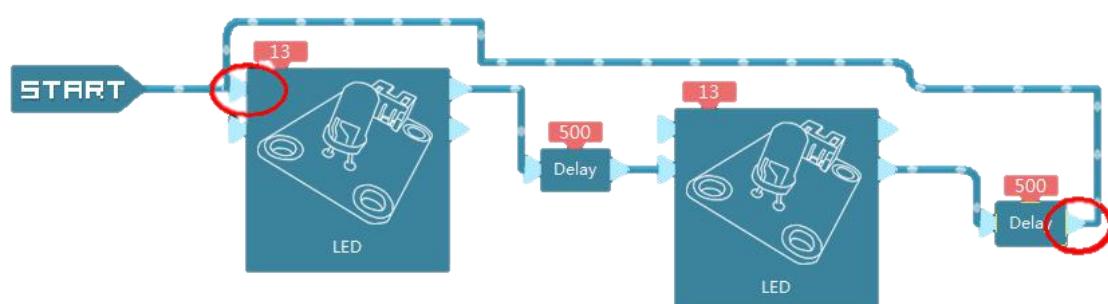


### Delay

+ Location: Basic block list – LED

## Mind+ Interface

The upper connector usually indicates on/true while the lower indicates off/false. Here we draw a line from START to the upper left connector, set a Delay of 500 milliseconds, connect it to the lower connector of same LED and another Delay to shut it down for another 500 milliseconds. In the end, we make a cycle from the first connector to the last connector to make the program running continuously. (equivalent to Loop in IDE)

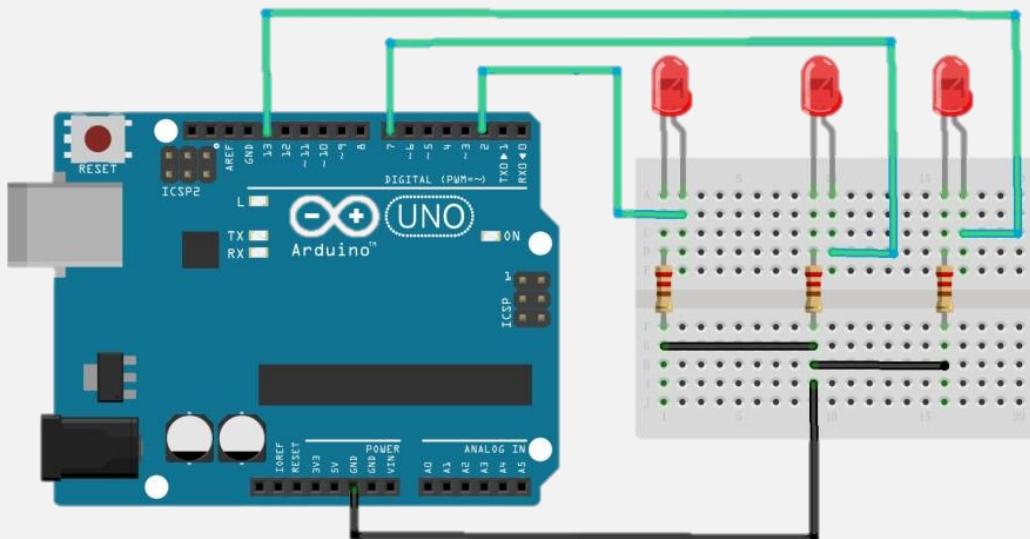


## Extended Exercise: Rolling LED

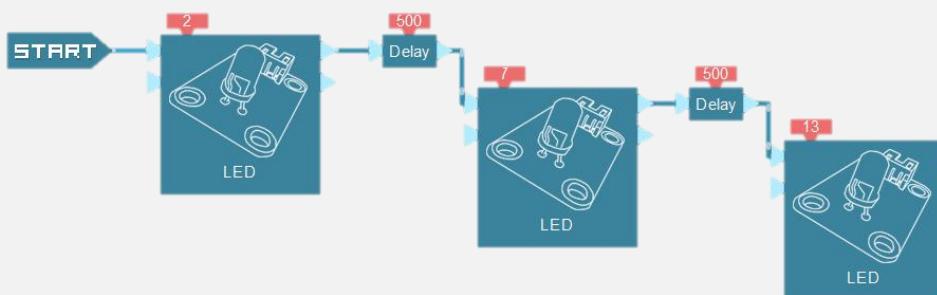
Turn three red LED lights on one by one every 500 milliseconds, creating an effect like floating water.

### Reference

#### Electrical Circuit



#### Mind+ Interface



# 04 Button

## Button-Controlled LED

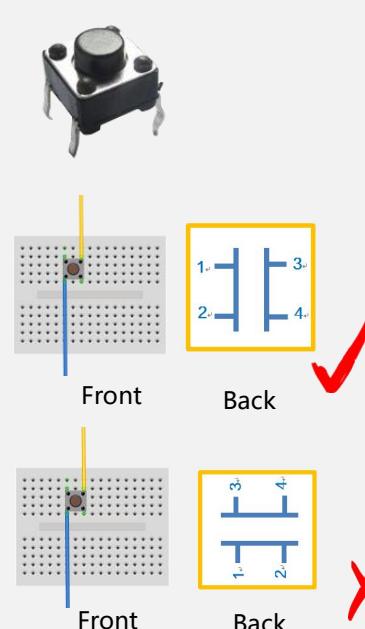
pushbutton is a very common electronic component to connect and disconnect circuit, in order to turn on and off devices. This project is to control LED with a pushbutton just like what we do at home . Meanwhile, you'll learn digital input and output signal, and how a sensor controls an actuator.

### Arduino

#### Part List :



pushbutton    LED    220 $\Omega$  resistor



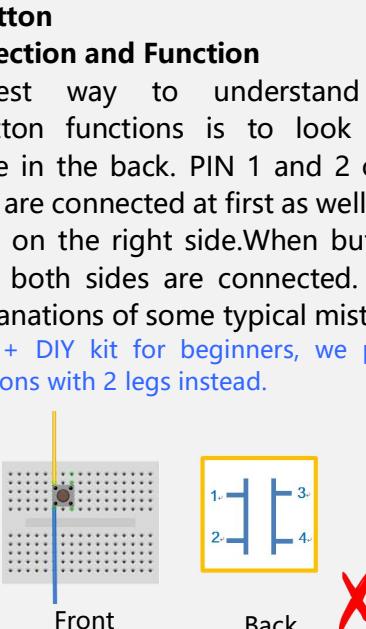
**pushbutton**  
+ Connection and Function

The best way to understand how pushbutton functions is to look at its structure in the back. PIN 1 and 2 on the left side are connected at first as well as pin 3 and 4 on the right side. When button is pushed, both sides are connected. Below are explanations of some typical mistakes.

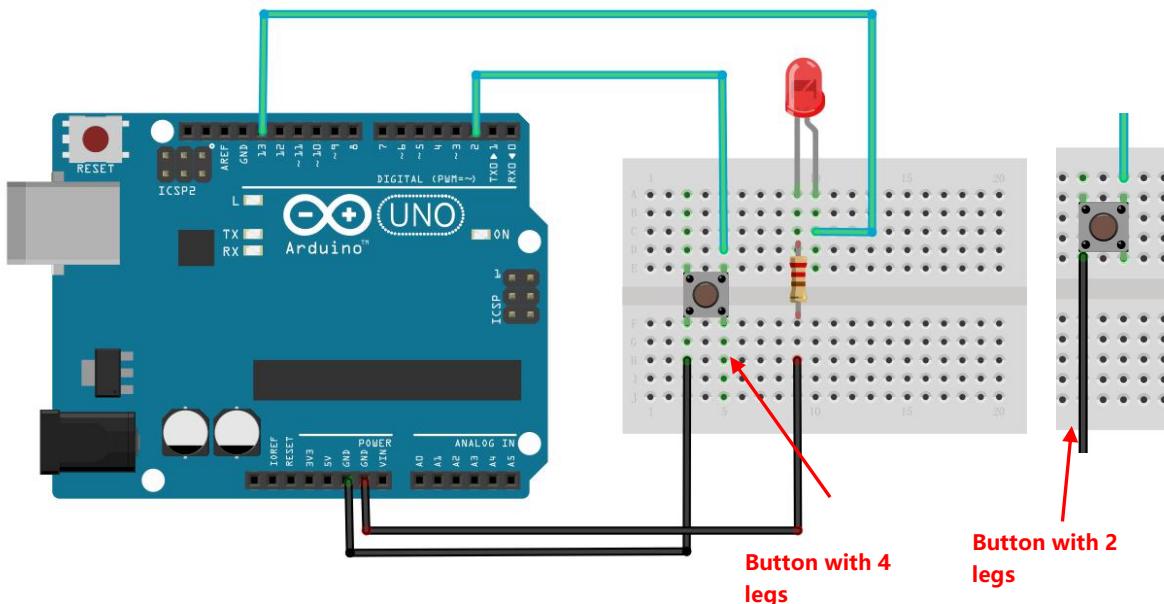
\*In Mind+ DIY kit for beginners, we provide pushbuttons with 2 legs instead.

**Wrong Example #1**  
When you switch the position of pushbutton, LED will be lit even the button is released because pin 1 and 3 are connected on vertical sockets of breadboard.

**Wrong Example #2**  
If you follow the instruction but connect 2 wires on the same side, the LED will be on even the button is released because pin 1 and 2 are connected on vertical sockets of breadboard anyways.



## Electrical Circuit



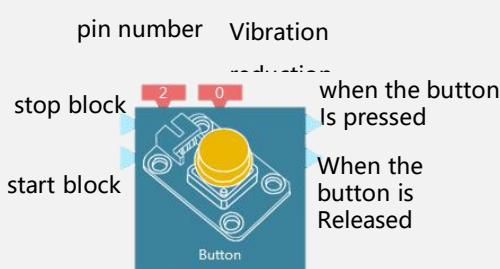
### How can a pushbutton control LED?

The Arduino board is like a micro-computer. In the real world, lightness levels, temperature, pressure can be transmitted into signal of environment through sensors. Arduino board receives this signal and sends out other specific signals as we program.

In this project, pushbutton is a sensor and LED is an actuator. We push and release button as digital input signal, then the board reads it and turns on the LED as digital output signal.

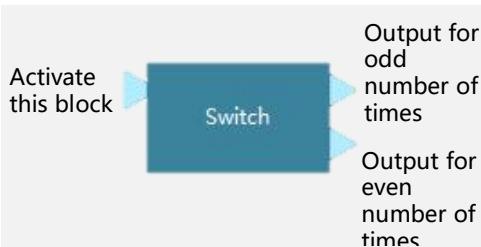
## Mind+

### block Description



#### Button

**+ Location:** basic block list -- Button  
**+ Usage:** When the button is pressed, 2 sheet irons inside keep shaking hence quickly connecting and disconnecting circuit. Vibration reduction is a duration of time to confirm an operation, similar to delay.



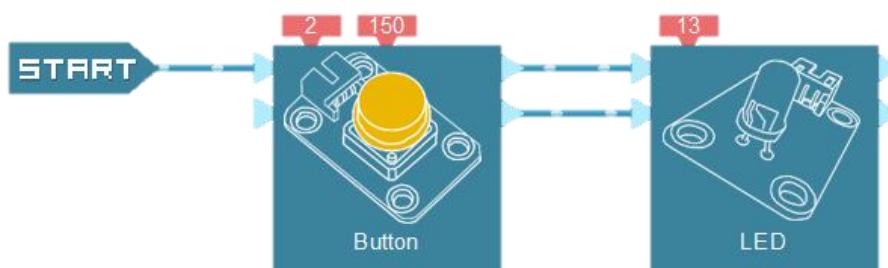
### Switch

**+ Location:** basic block list -- Switch  
**+ block Description:** Switch is a very common block to turn on and off devices which is similar to button. But it memorises number of times you push the button. When press at odd number of times, it operates the upper output connector, when it is at even number of times, it operates the lower output

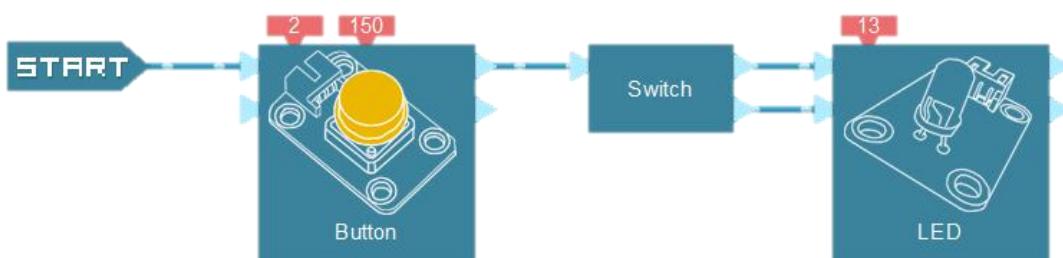
## Mind+ Interface

1. Drag Button and LED from tool bar and drag it to the work area, and connect them as below. Here

Upper connectors are connected together as well as lower connectors, meaning when the button is on, LED is on and when the button is off, LED is off.



2. What if we want to keep the light on after pushing the button just like in our home. Try Switch block from basic block list and connect it again to the LED! When the button is pushed for odd number of times, LED is on. When the button is pushed for even number of times, LED is off.



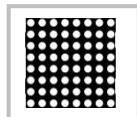
# 05 LED Matrix

## LED Animation

8x8 LED Matrix is consist of 64 LEDS in 8 rows and 8 columns. You can easily display numbers, letters or graphics on it.

### Arduino

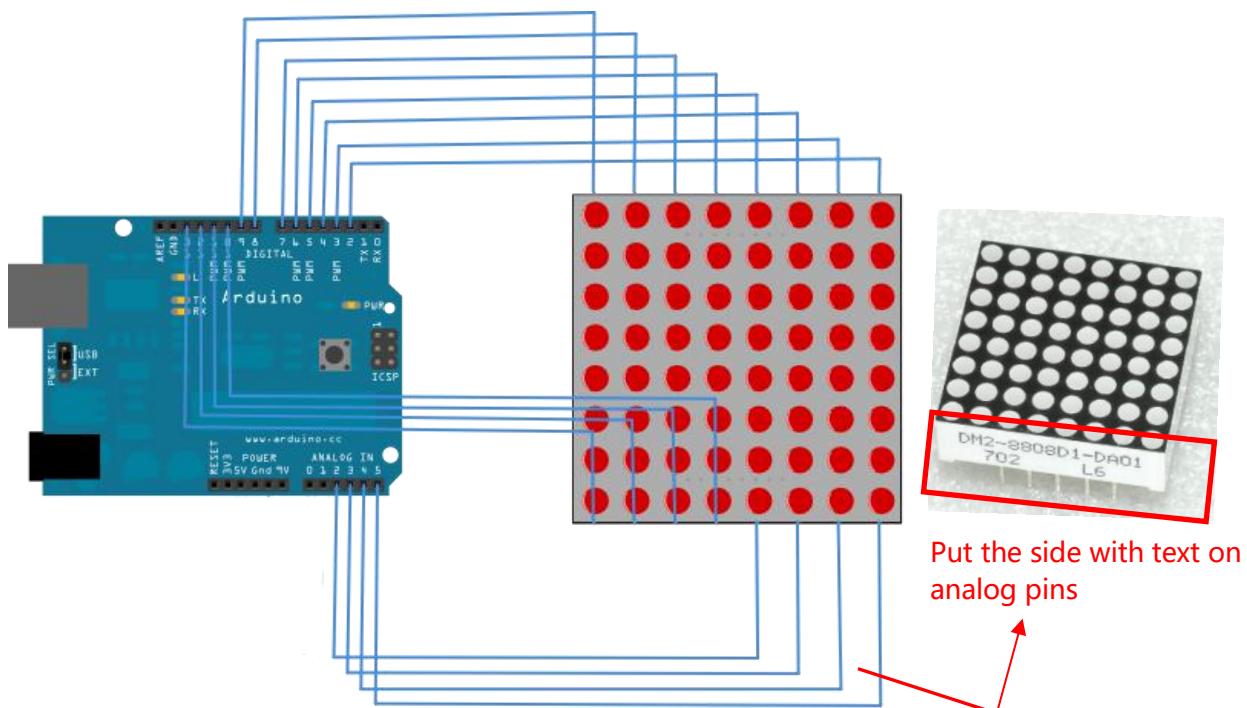
#### Part list :



8x8 LED Matrix

#### Electrical Circuit

Connect the wires as below and don't forget to put the side with text onto analog pins.



## Mind+

### block List :



8x8 LED Matrix   Delay

**8x8 LED Matrix**

- + Location :** advanced block list -- LED -- 8X8 LED Matrix
- + Usage :** double click the value parameter box on the upper left side, left click to fill color on dots, right click to cancel.

**Delay**

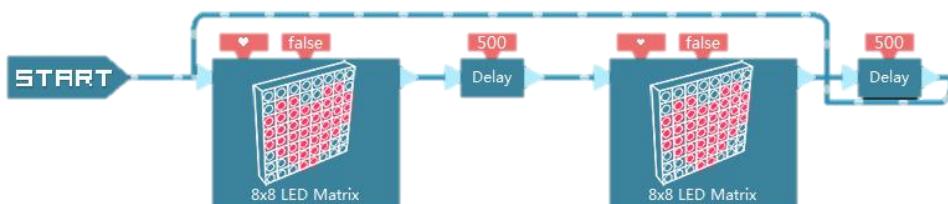
- + Location:** basic block list -- Delay

\*1 second =1000 mille second

### Mind+ Interface

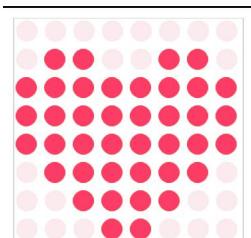
**Tips:** For a quick search to get block, you can double click the background of work area and enter key words after the bar pops up.

Drag and drop 2 8x8 LED Matrix and 2 Delay to the work area, and set up the graphic value and delayed time. Don't forget to make a loop from the first connector to the last connector so it can run continuously.

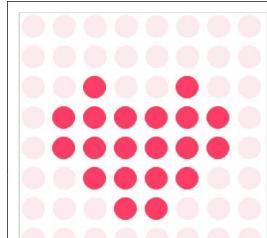


**Note:** If you follow the instruction but find it could not show the graphics you set up, try to change the default value false to true on the right parameter box.

Heartbeat graphics for your reference



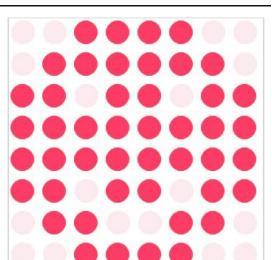
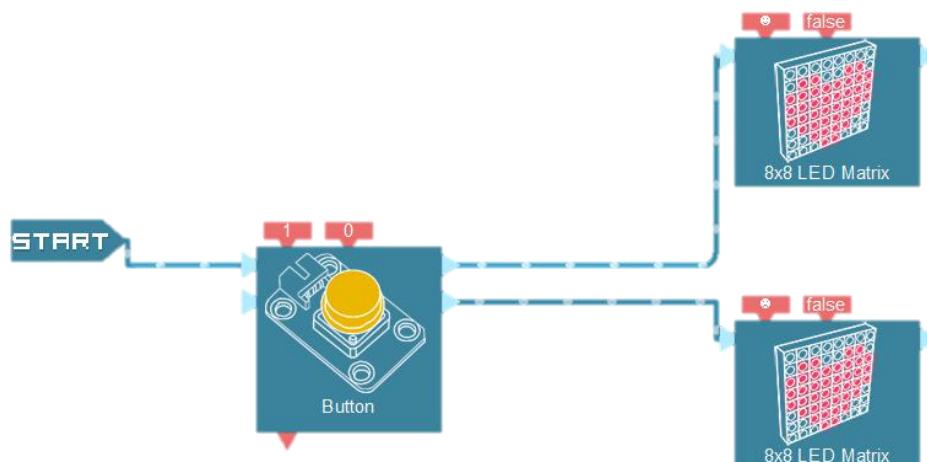
Done



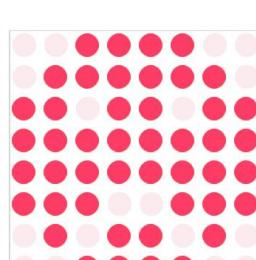
Done

## Extended Exercise

Now you can make a more interactive 8x8 animation. For example, use a button as a sensor to change graphics. When the button is pressed, it displays a happy face. When the button is released, it displays a sad face. Have fun!



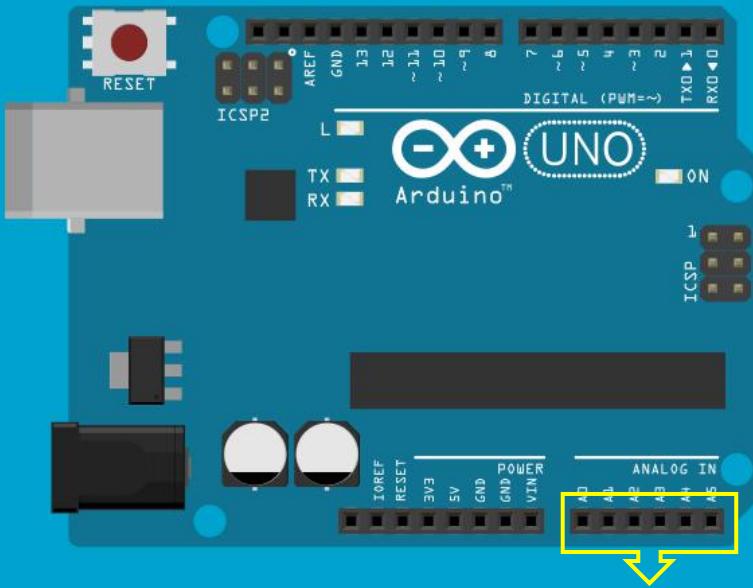
Done



Done

---

## **2 ANALOG INPUT**



Analog Input pins D3, D5, D6, D9, D10, D11

We've introduced digital input and output in the previous chapter, but there are also a lot of things out there that aren't just "on" or "o".

Temperature levels, control knobs, etc. all have a continuous range of values between HIGH and LOW.

For these situations, the Arduino board has six analog inputs ( A0-A5 )that translate an input voltage into a number that ranges from 0 (0 volts) to 1023 (5 volts). These pins are perfect for measuring all those "real world" values, and allow you to interface the Arduino board to all kinds of things. There are many components that measure analog signals. In the following projects, we will use rotation sensor to control make music (Project#6) , use photoresistor to control light automatically(Project#7), and use temp sensor to make a temp-sensible alarm(Project#8).

# 06 Rotation Sensor Tuning Machine

The first part of the project is to learn Analog value through rotation sensor and the second part is to make music with rotation sensor and piezo buzzer.

## Arduino

### Part list



Rotation sensor



### Rotation Sensor

#### + Connection:

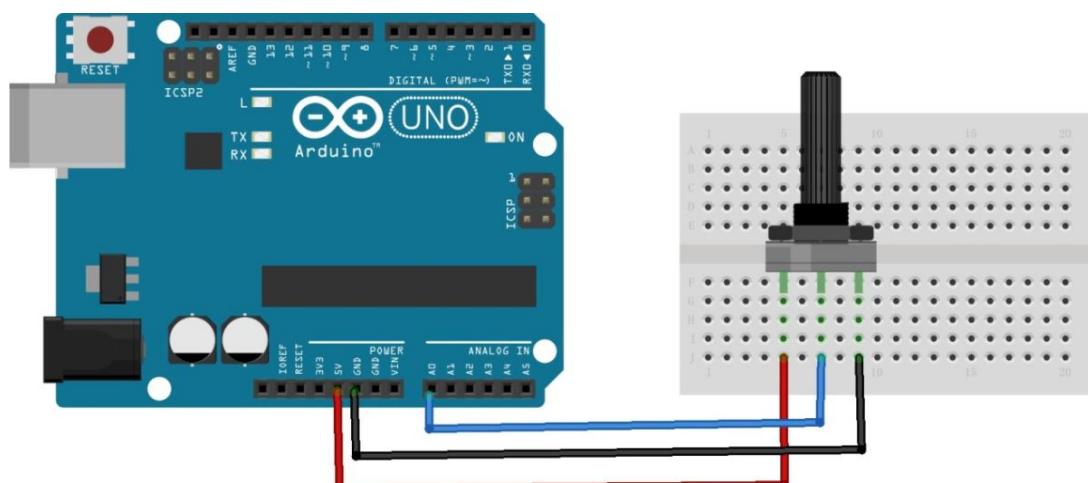
left pin and right pin to 5V and GND, middle pin to A0.

#### + Function:

Rotation sensor, also known as potentiometer, is consist of a round resistor and a knob. When we twist it, resistance value will change depending on where knob points to between 5V and GND. **The more distance between 5V pin and middle pin, the more resistance value on the round resistor.** That is why you can connect left and right pin to either 5V or GND. It is a perfect demonstration of a variable voltage divider circuit, especially for creating interactive effect and MIDI.

#### + Usage:

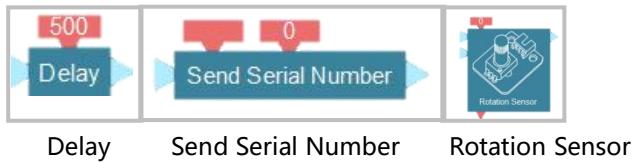
## Electrical Circuit



Made with Fritzing.org

## Mind+

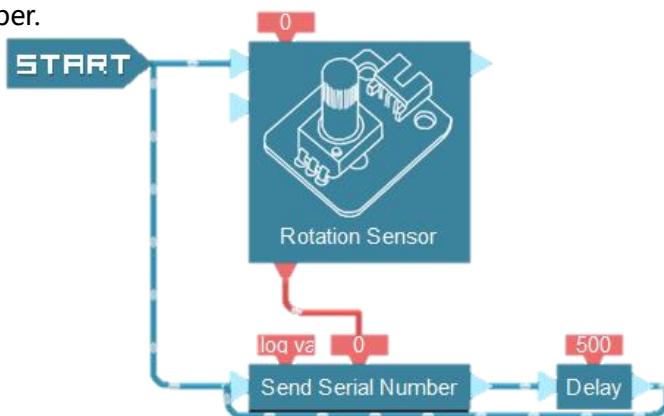
### Block list



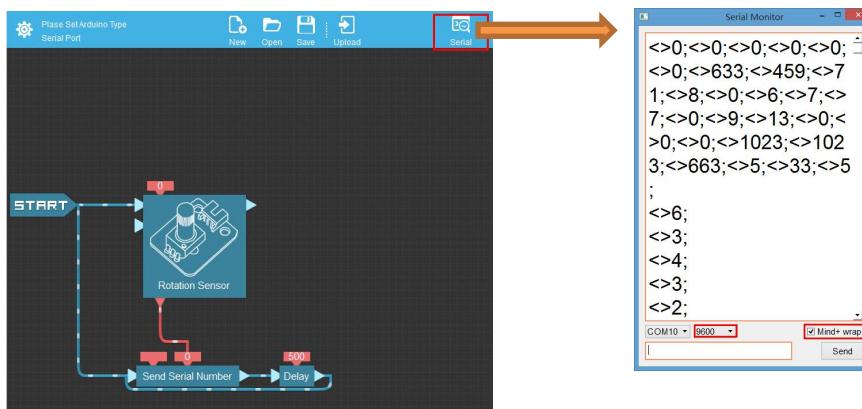
<p><b>Rotation Sensor</b></p> <p>+ Location basic block lists -- Rotation Sensor</p> <p>+ block Description Analog value of rotation sensor ranges from 0 to 1023. This block can be also used for slide position sensor.</p>	<p><b>Send Serial Number</b></p> <p>+ Location advanced block list -- Serial -- Send Serial Number</p> <p>+ block Description Send Serial Number sends value from serial port.</p>	<p><b>Delay</b></p> <p>+ Location: basic block list -- Delay</p>
---	--	--

### Mind+

Drag Rotation Sensor, Send Serial Number and Delay from the menu to the work area, and connect them as below. You can enter the value name on the left parameter box of Send Serial Number such as analogue value. Additionally, don't forget to draw a line between the red connector of both Rotation Sensor and Send Serial Number, so Send Serial Number can read analog value of rotation sensor. Delay is to set up the period of data collecting and operate constantly through making a loop with Send Serial Number.



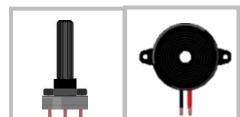
After uploading the program, you can open Serial Monitor on the right upper side of the interface and check real-time data from the serial port. 9600 is the default baud rate of serial port or the number of transmitting binary system every second. The slower the more stable the data processes. Additionally, tick the box of Mind+warp and each data can be shown in one separate row so you can see them clearly.



**In the project above, we've learned how to use rotation sensor, now we can add a piezo buzzer to play with music with various notes and rhythms.**

## Arduino

### Part list



Rotation sensor    buzzer

#### Speaker

##### + Connection

no polarity -- one leg on GND and the other on D7.

##### + Function

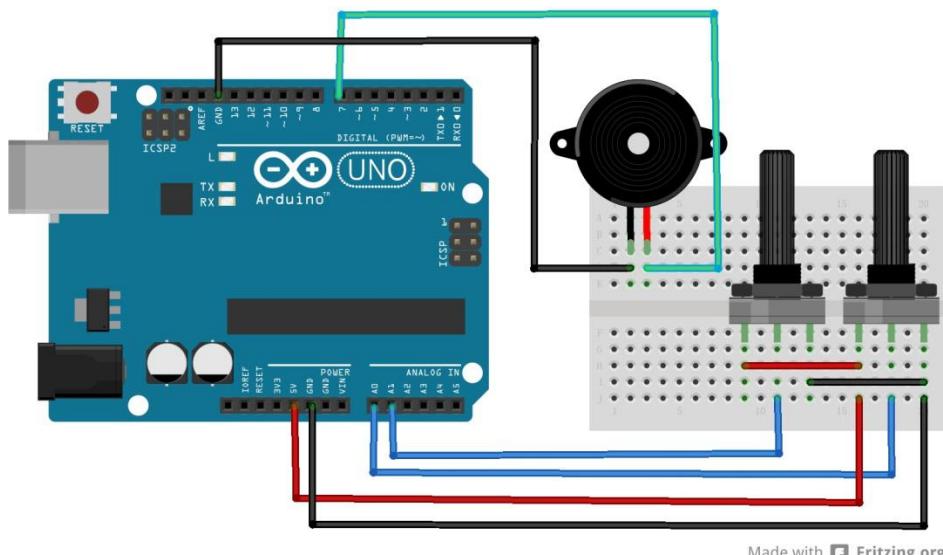
Speaker has no source of vibration. Similar to electromagnetic loudspeaker , it can make sound once connected to electricity.

##### + Usage

Speaker looks identical to buzzer. Speaker has no polarity so both legs have same length and there is an electronic circuit board in the back. Buzzer has polarity so it has a long leg and a short leg, and the back is covered with blacking coating.

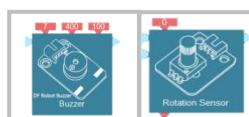


## Electrical Circuit



## Mind+

### block List



Buzzer      Rotation Sensor

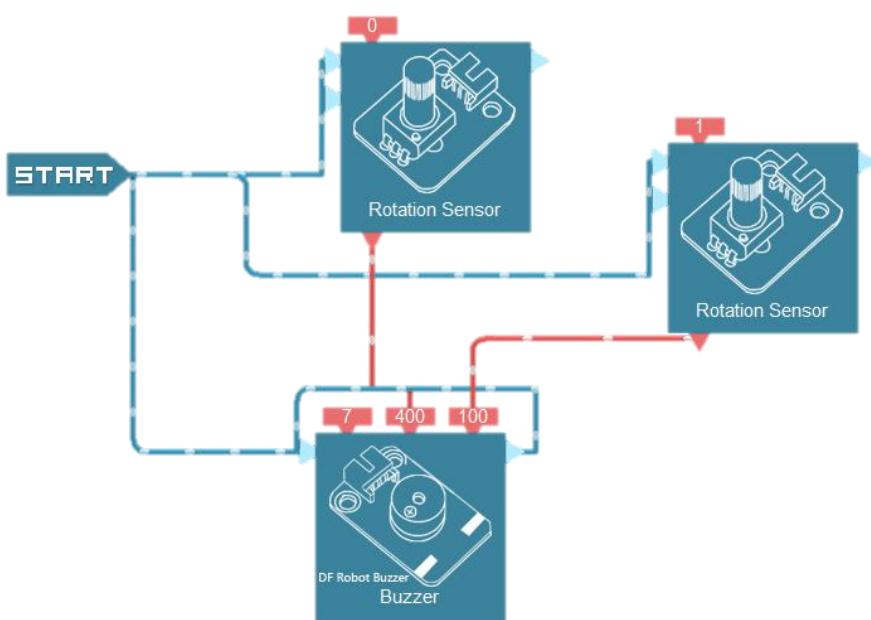
Pin number	frequency	time
7	400	100

Start block      Output pin

**Buzzer**  
+ **Location:** Basic block list-- Speaker or Buzzer  
+ **block Description :** 2 parameter boxes control note and rhythm.

## Mind+ Interface

Drag 2 Rotation Sensors and 1 Buzzer from the library, and connect them as below. Don't forget to connect red connector of 2 Rotation Sensor with Red connectors of Buzzer so the Buzzer can read the data from Rotation Sensor. Additionally, make a loop of Buzzer so it can keep making sound.



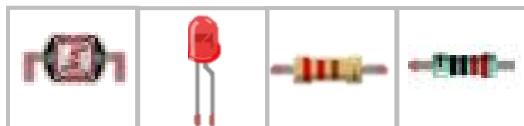
# 07 Light Sensor

## Auto Light

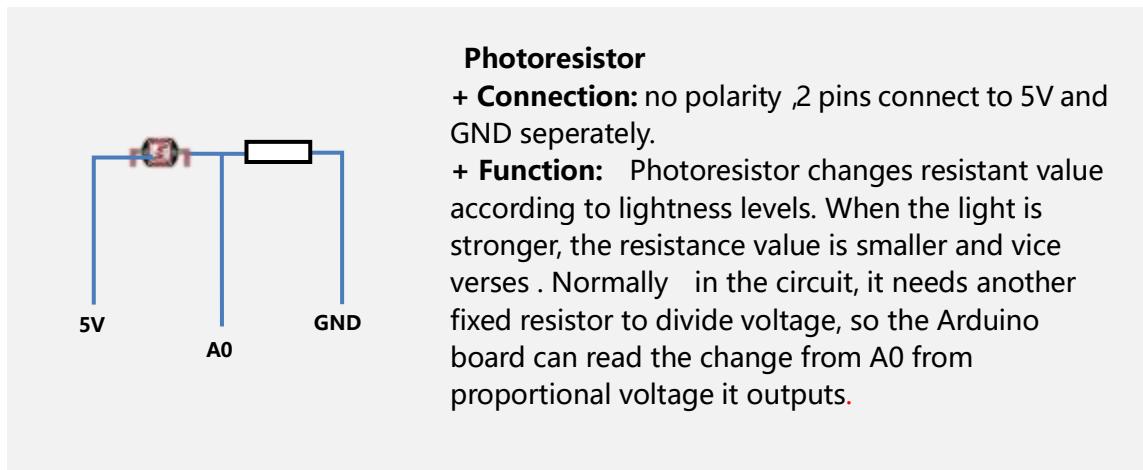
After getting to know the basics of analog input /output, we can use IF to evaluate conditions in the environment. In this project, we light up LED when the environment gets dark and turn off LED when it gets bright, just like street light.

### Arduino

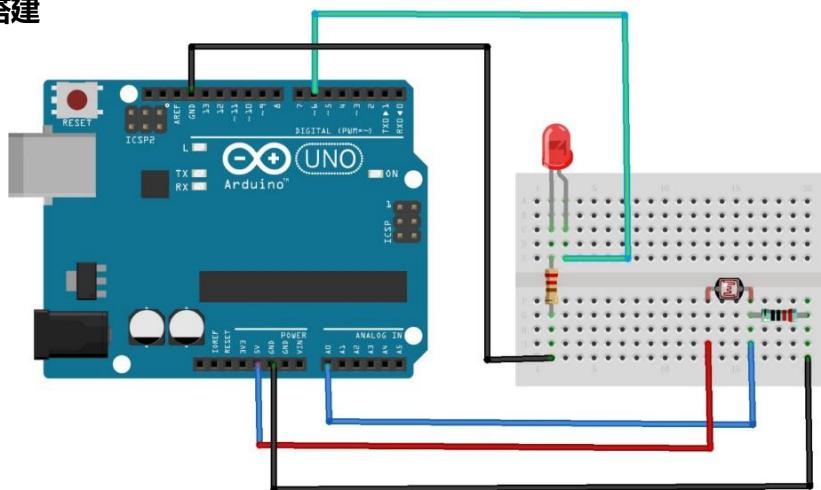
#### Part list



photoresistor LED 10k $\Omega$ resistor 220 $\Omega$ resistor



### 电路搭建



Made with Fritzing.org

## Mind+

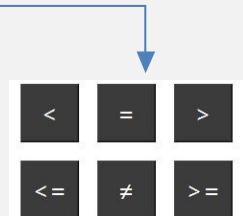
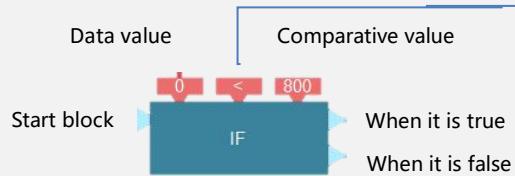
### block List



IF

Analog Read

LED



#### IF

- + **Location:** basic menu -- IF
- + **block Description**

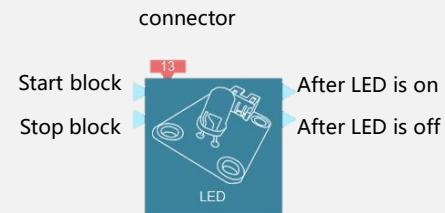
IF is used to compare value. The default data value is 0 and your can set up any comparative value and relation on the parameter box.



#### Analog Read

- + **Location:** Advanced block list -- Basic Sensor -- Analog Read

+ **block Description:** Analog Read is used to output voltages in various electronic components such as rotation sensors, photoresistor and temp sensor.

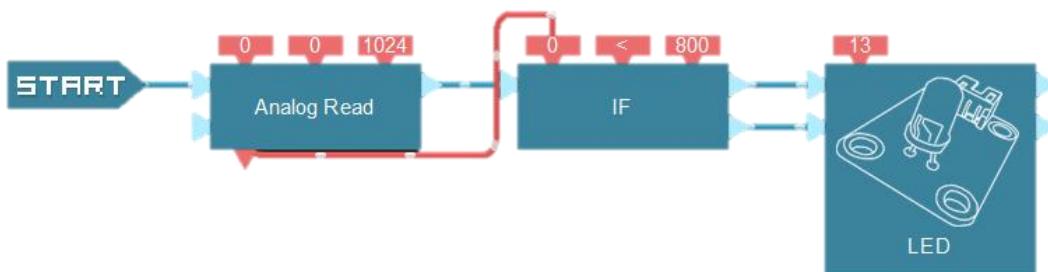


#### LED

- + **Location:** basic block list -- LED

## Mind+ Interface

Photoresistor , equivalent to Analog Read, reads the environmental light and inputs as analog value, then use IF to evaluate. If analog value is lower than 800, the light turns on automatically. If higher than or at 800, the light turns off. Additionally, you need to connect red connectors of Analog Read and IF to transmit data.



## Extended Exercise : Show various statuses through light

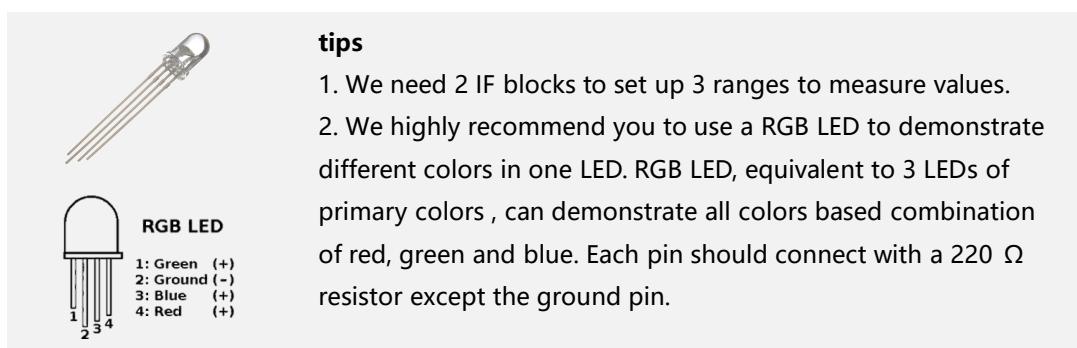
You might find light of your mobile phone or computer displaying various colors based on electricity status. We can imitate this effect to make a lightness-sensitive LED.

When analog value of photoresistor is between 0 -340, the color of RGB LED is red.

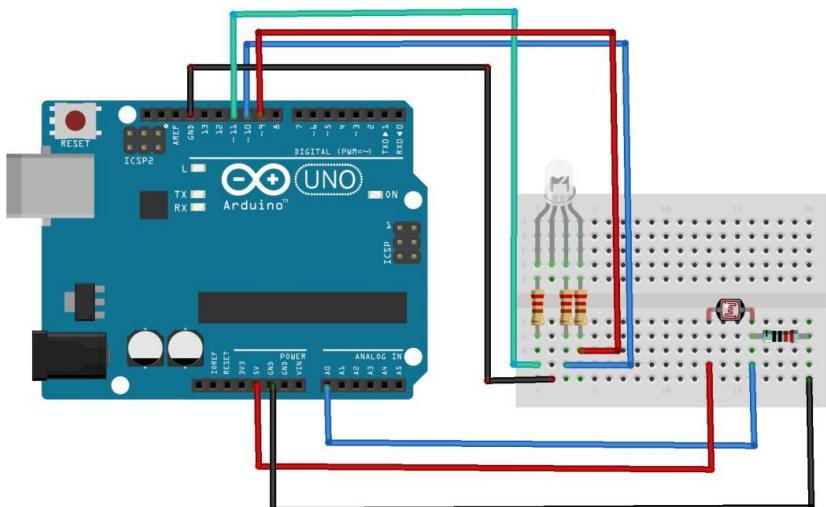
When analog value of photoresistor is between 341 - 680, the color of RGB LED is green.

When analog value of photoresistor is between 681 -1023, the color of RGB LED is blue.

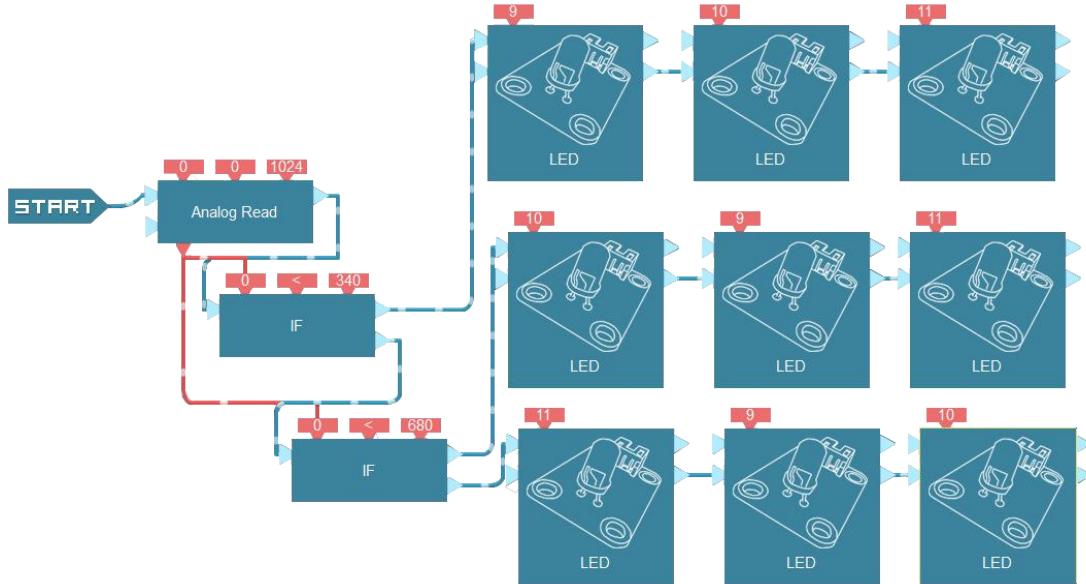
This project is a bit difficult, take you time to think about the programming part.



## 电路搭建



## Mind+ Interface

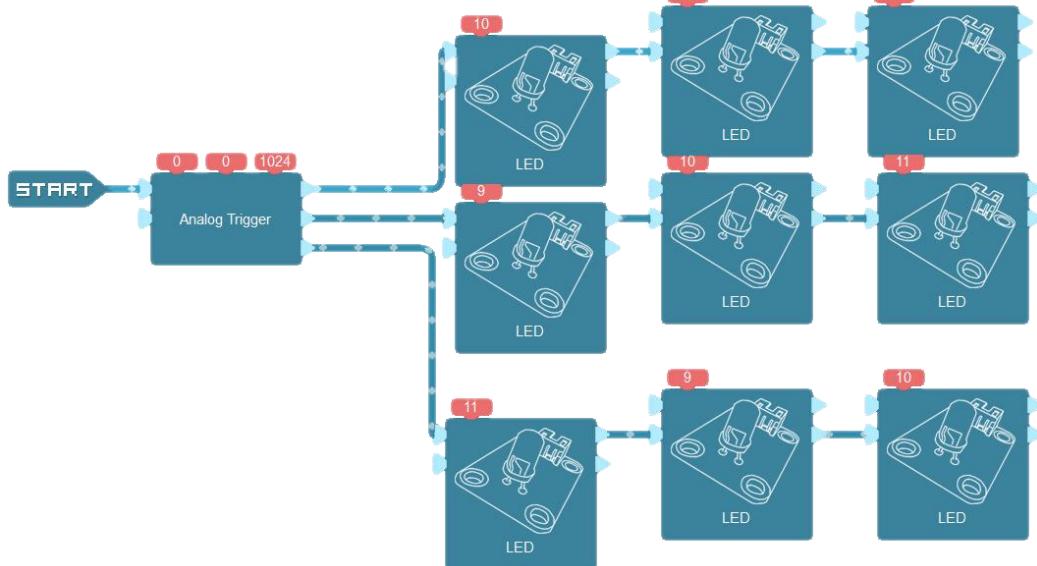


If you'd like to explore more, you could use Analog Trigger which is a combination of Analog Read and IF. Analog Trigger block has 2 comparative values, dividing values into 3 ranges and outputs 3 statuses. The electrical circuit is similar as above, but be careful that first output connector is the range between 2 comparative values which is different compared to IF.

Pin number	Smaller comparative value	Bigger comparative value
------------	---------------------------	--------------------------

Start block	0, 86, 170	When analog value is smaller than the smaller comparative value
Stop block	Analog Trigger	When analog value is between 2 comparative values
		When analog value is bigger than the bigger comparative value

## Mind+ Interface



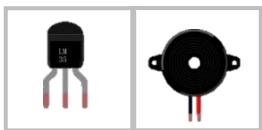
# 08 Temp Sensor

## Temp Alarm

Followed with last project, we use IF again to evaluate condition and switch to various statuses automatically according to different conditions in the environment. When temperature is higher than a certain degree, the buzzer makes noise as an alarm.

### Arduino

#### part list :



LM35

buzzer

#### LM35

##### + Connection:

The legs go to 5V, A0 and GND. (Put the component with the side marked LM 35 faced to you.)

##### + Function:

LM 35 can measure temperature from 0 -150 degree and output voltages ranging 0-1023 and convert into degrees.



#### Buzzer



Buzzer



Speaker

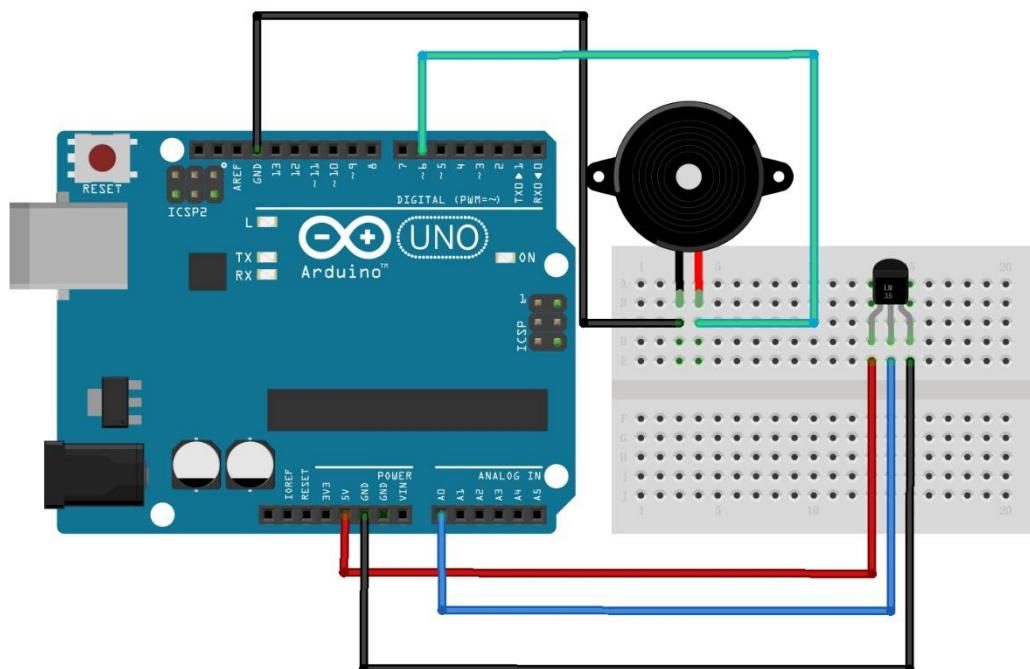
##### + Connection:

Buzzer contains a source of vibration inside. Once connected with electricity, it makes noise.

##### + Usage:

Speaker looks identical to buzzer. Speaker has no polarity so both legs have same length and there might be a small circuit board in the back. Buzzer has polarity so it has a long leg and a short leg, and the back is covered with black coating.

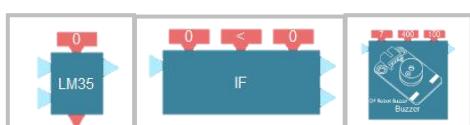
## Electrical Circuit



Made with Fritzing.org

## Mind+

### Part list

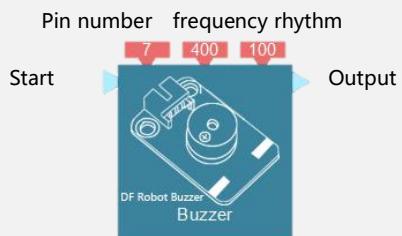


LM35

IF

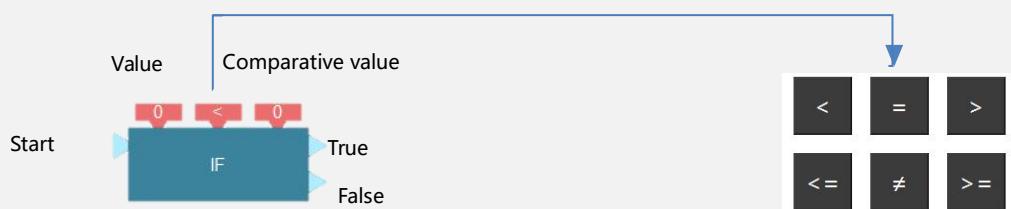
Speaker

<p>Pin number Start Stop Data Output</p>	<p><b>LM35</b></p> <ul style="list-style-type: none"> <li>+ <b>Location:</b> basic block list - LM 35</li> <li>+ <b>block description:</b> LM 35 can measure temperature from 0 -150 degree and can be converted to degree.</li> </ul>
--	--



### **Speaker**

- + **Location:** basic block list -- Speaker or Buzzer
- + **block Description:** Speaker and buzzer can use Buzzer block. If it is buzzer, you don't need to set rhythm and note.

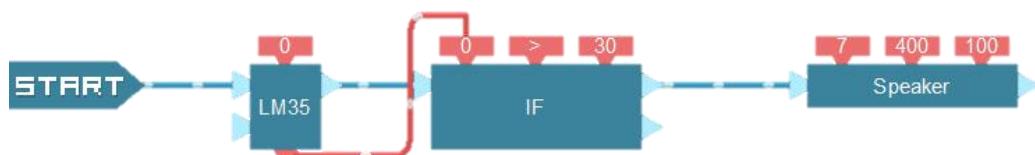


### **IF**

- + **Location:** basic block list - IF
- + **block Description:** IF can be used to change status under different environmental conditions. You can set up a comparative value and relation in parameter boxes. The upper output connector means true while the lower output connector means false.

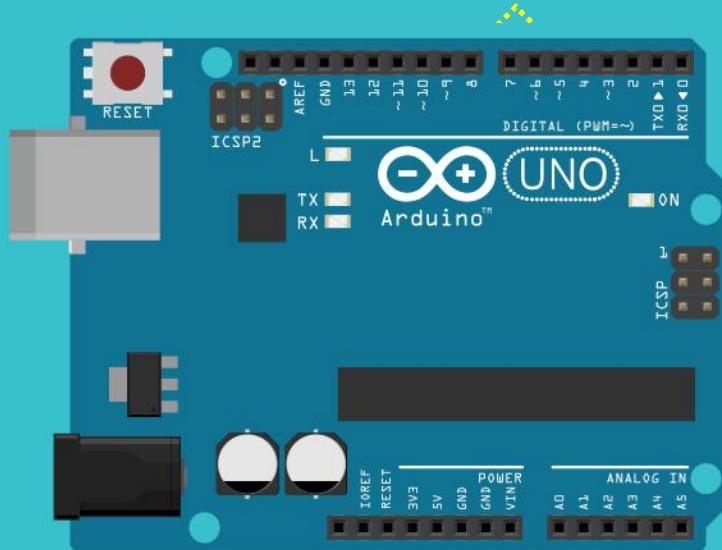
## **Mind+ Interface**

Drag LM 35, IF and Speaker blocks, and connect them as below. Don't forget to draw a line from data connector of LM 35 to parameter box of IF. If you set the comparative value 30 degrees, when the temperature is higher than 30 degree, the buzzer buzzes. If lower than 30 degree, nothing happens. You can hold LM 35 to manually increase temperature if needed.



# 3 ANALOG OUTPUT / PWM

## Analog Output / PWM pins D3、D5、D9、D10 and



We've seen that the Arduino board can read analog voltages (voltages between 0 and 5 volts) in pins labeled ANALOG IN pins. But there is not ANALOG OUT on the Arduino board. Actually the Arduino Board does not have a true analog voltage output. But because it is so fast that it can blink a pin on and off almost 1000 times per second, it can fake it using something called PWM ("Pulse-Width Modulation").

Pins with “~” next to them are PWM/Analog out compatible. They are D3,D5,D6,D9,D10 and D11. These 6 pins translate an output voltage into a number that ranges from 0 (0 volts) to 255 (5 volts).

Electronic components using PWM signals includes LED, servo and buzzer. In the following projects, we will make LED with simulated dimming (project#9) and a servo controlled by rotation sensor (project#10).

# 09 LED (PWM)

## Light Controller

We can turn on and off LED but also change its lightness levels. In this project, we use a curve generator block to control speed and lightness levels of a LED.

### Arduino

#### Part list



Rotation Sensor LED 220 $\Omega$  resistor

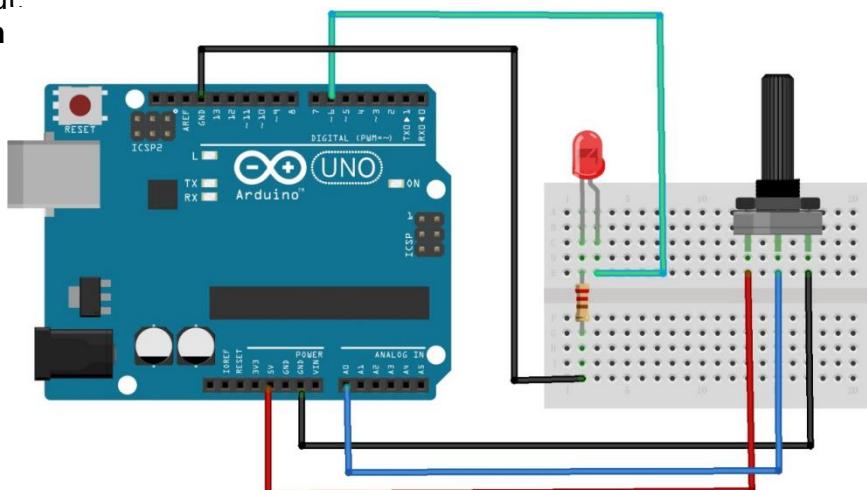
#### LED

- + **Connection** : short leg goes to GND and long leg goes to D6.
- \*\* long leg goes to digital output pin when you try to turn on/off it but goes to PWM pin when you try to change lightness levels.



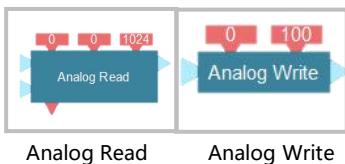
+ **Function** : The Arduino board can blink a pin on and off almost 1000 times per second using PWM ("Pulse-Width Modulation"). PWM varies the amount of time that the blinking pin spends HIGH vs. the time it spends LOW. If it spends most of its time HIGH, a LED connected to that pin will appear bright. If it spends most of its time LOW, the LED will look dim. Because the pin is blinking much faster than your eye can detect, the Arduino board creates the illusion of a "true" analog output.

#### Electrica



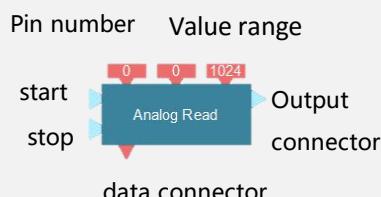
## Mind+

### block List



Analog Read

Analog Write



#### Analog Read

+ **Location:** Advanced block list -- Basic Sensor -- Analog Read  
+ **block Description:** Analog Read is used to output voltages in various electronic components such as rotation sensors, photoresistor and temp sensor.

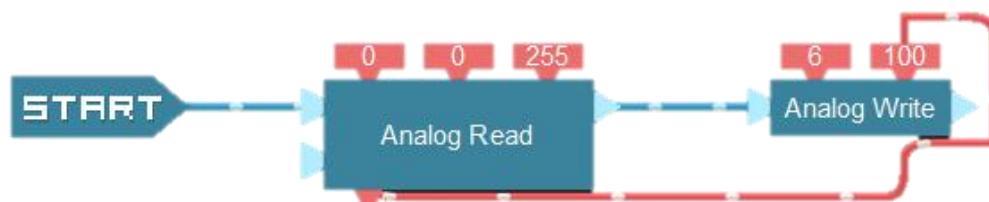
#### Analog Write

+ **Location:** Advanced block list -- Basic Sensor -- Analog Write  
+ **block Description:** Analog Write is used to input voltages in various electronic components such as LED and buzzer.

### Mind+ Interface

Drag Analog Read and Analog Write blocks and drop them in the work area, and connect them as below. Don't forget to set the value range of Analog Read 0-255 as it inputs PWM signal, and draw a line to connect data connectors of both blocks to simulate the flow of data.

**Note:** If you use rotation sensor block instead of Analog Read block, you will find LED dimming 4 times when twisting rotation sensor from 0-180. It is because the default analog value of Rotation Resistor block range from 0-1023, not 0-255 in consistent with PWM output analog voltage.



For sure you can use a rotation sensor to control lightness levels manually, but you can set parameter to control it with a fix pattern automatically.

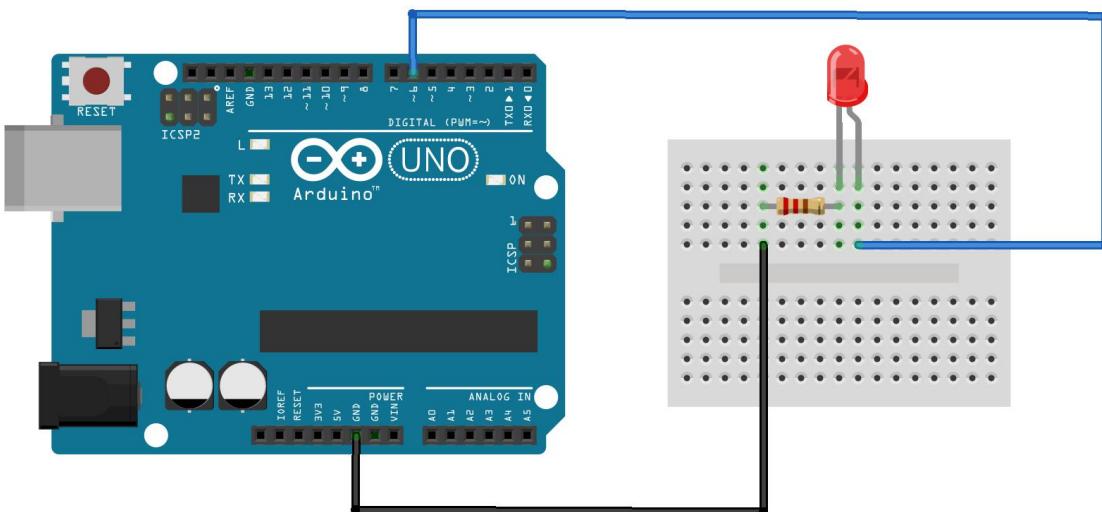
## Arduino

### Part list



LED      220  $\Omega$  resistor

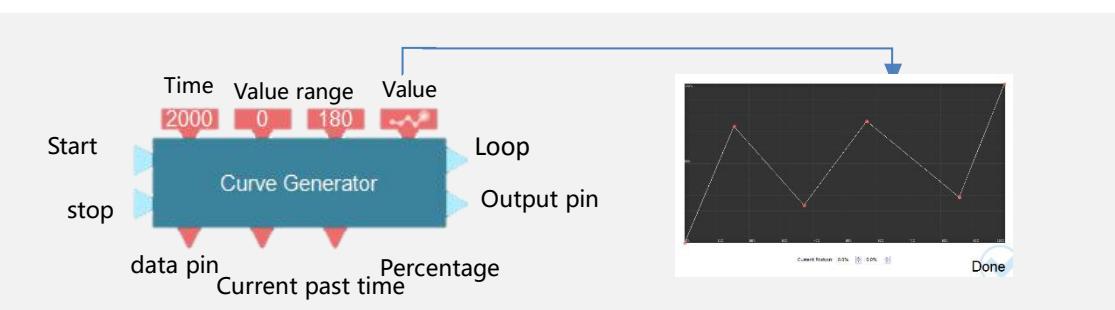
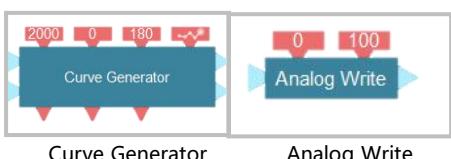
### Electrical Circuit



Made with Fritzing.org

## Mind+

### block list



### Curve Generator

+ **Location:** advance block list - Time -- Curve Generator

+ **block Description:** Curve Generator control speed, angle and frequency of a movement, perfectly designed for LED animation and servo movement. Double click the W shape parameter box to open the table. Double click the line to create a movable dot and drag it to the direction you want. Right click to cancel the dot. Press DONE when you think it is perfect.

### Analog Write

Pin number Analog value

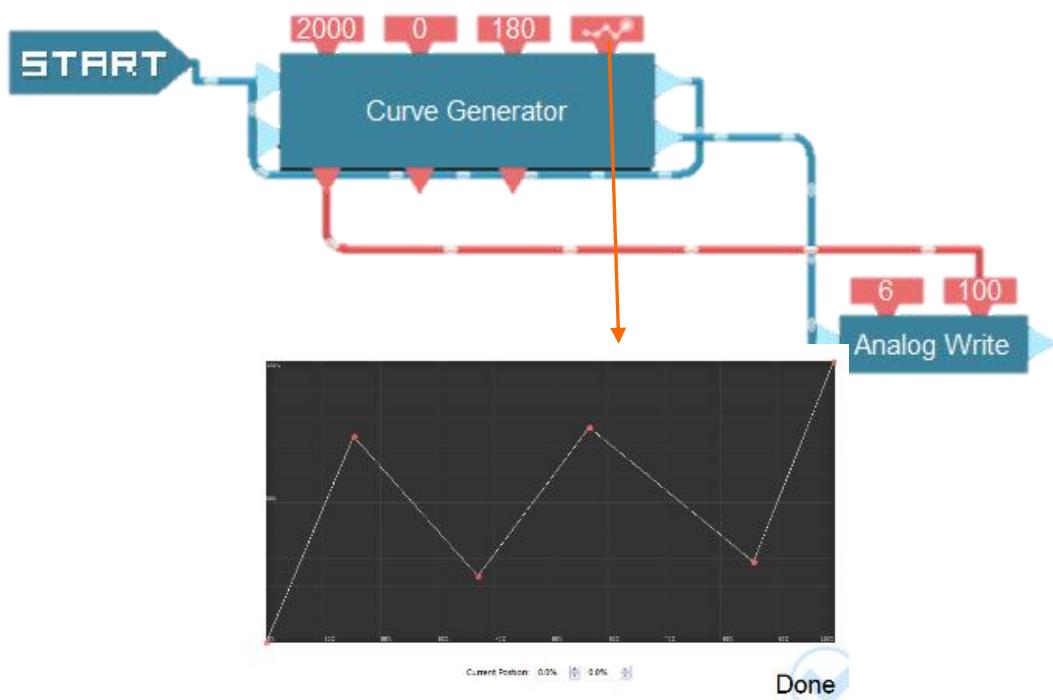


+ **Location:** Advanced block list -- Basic Sensor -- Analog Write

+ **block Description:** Analog Write is used to input voltages in various electronic components such as LED and buzzer.

## Mind+ Interface

Drag Curve Generator and Analog Write from the block list and drop them in the work area, and connect them as below. Don't forget to connect data connectors (red) of both blocks to simulate data transmitting. Also, we need to set the analog value range from 0-255, to go in consistent with PWM signal of LED.

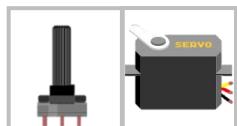


# 10 Servo

# Robotic Arm

## Arduino

## Part list



Rotation sensor servo

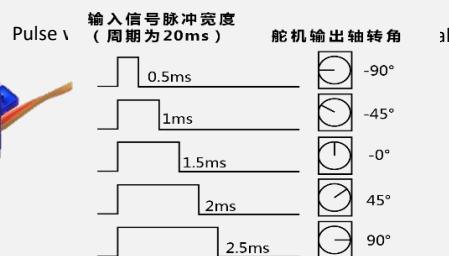
Servo

+ **Connection** : Black, red and white wire go to GND, 5V and D6

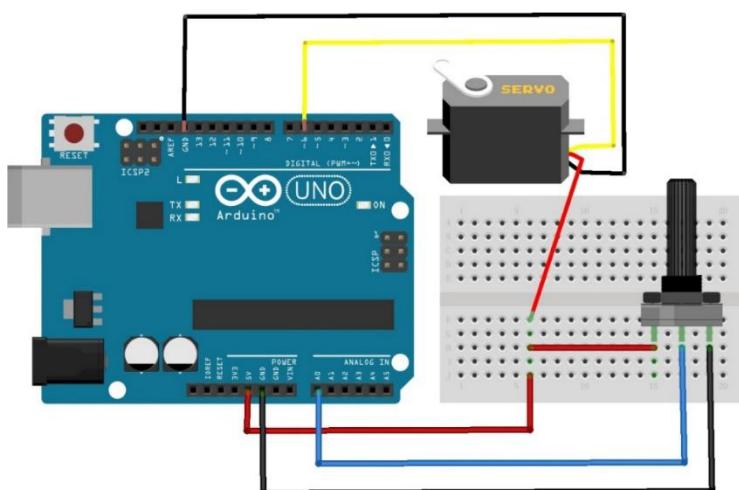
**+ Function:** Servos are ideal for embedded electronics applications because they do one thing very well that motors cannot: they can move to a position accurately. By varying the pulse width of the output voltage to a servo, you can move a servo to a specific position. For example, a pulse of 1.5 milliseconds will move the servo 90 degrees. In this circuit, you'll learn how to use PWM (pulse width modulation)

to control and rotate a servo. But the down side is that it can only turn 0-180 degree.

When it surpasses 180, say 190, it turns back to 10 degree due to its mechanism limitation.



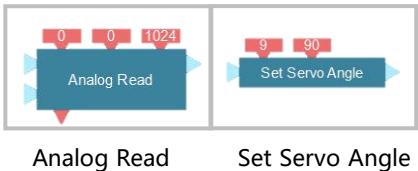
## Electrical Circuit



Made with  Fritzing.org

## Mind+

### block list



Pin number angle

Start Set Servo Angle Output connector

#### Set Servo Angle

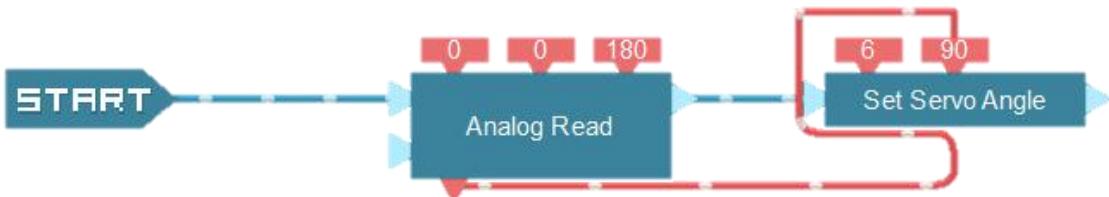
+ **Location:** advance block list -- Servo -- Set Servo Angle

+ **block Description:** set up the absolute angle of servo.

### Mind+ Interface

Drag Analog Read and Set Servo Angle in the block list and connect them as below. Don't forget to connect data connectors of both blocks and also change the value range of Analog Read from 0 to 180 in consistent to the value range of servo.

If you use Rotation Sensor instead of Analog Read which default value range is 0-1023, it cannot go with the value range of servo.



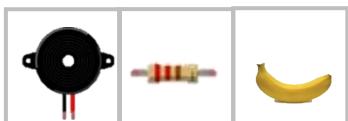
# 11 Capacity Sensor

## Banana Piano

Human touch can change the capacity of the circuit so we use it a lot to trigger some other effects. Inspired by Makey Makey, we use buzzers and resistors to create a banana piano which you can play music by touching bananas and also give an extended example of making an avatar plant in which hidden LEDs are lit when people touch it.

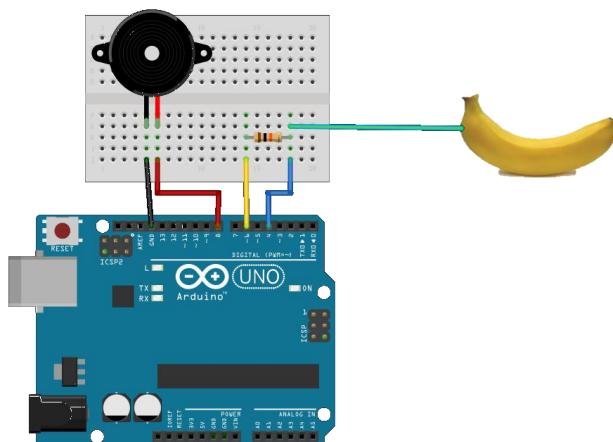
### Arduino

#### Part list



Buzzer 10k resistor bananas

#### Electrical Circuit

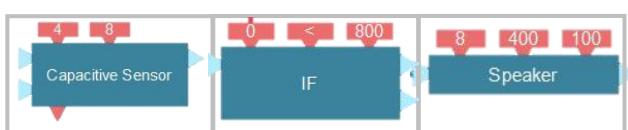


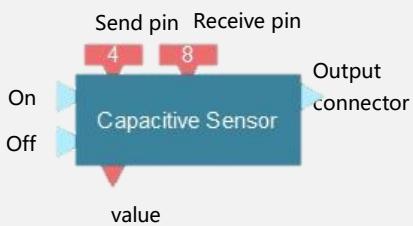
Capacitance is the ability of a body to store an electrical charge. Any object that can be electrically charged exhibits capacitance.

D6 pin sends signal and D4 pin receives signal. When we touch the banana, it changes the capacity of

### Mind+

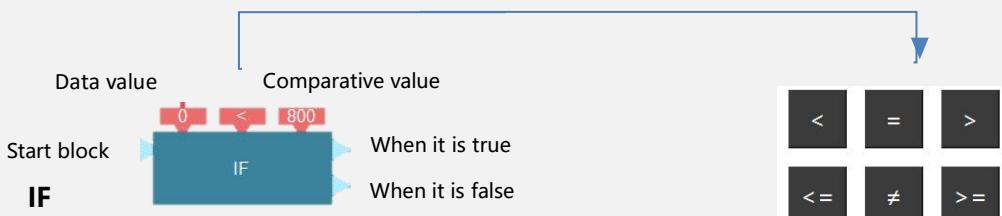
#### block list





### Capacity Sensor

+ **Location:** Advanced block list -- Basic Sensor -- Analog Read  
+ **block Description:** Analog Read is used to output voltages in various electronic components such as rotation sensors, photoresistor and temp sensor.



+ **Location:** basic menu -- IF

+ **block Description**

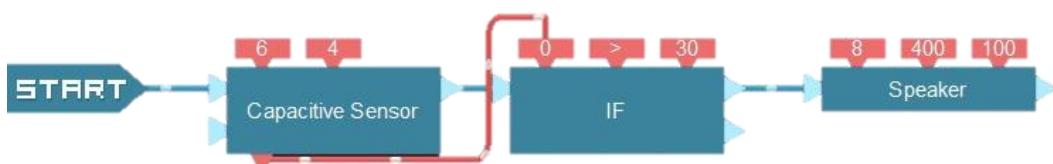
IF is used to compare value. The default data value is 0 and you can set up any comparative value and relation on the parameter box.



### Speaker

+ **Location:** basic block list -- Speaker or Buzzer  
+ **block Description:** Speaker and buzzer can use Buzzer block. If it is buzzer, you don't need to set rhythm and note.

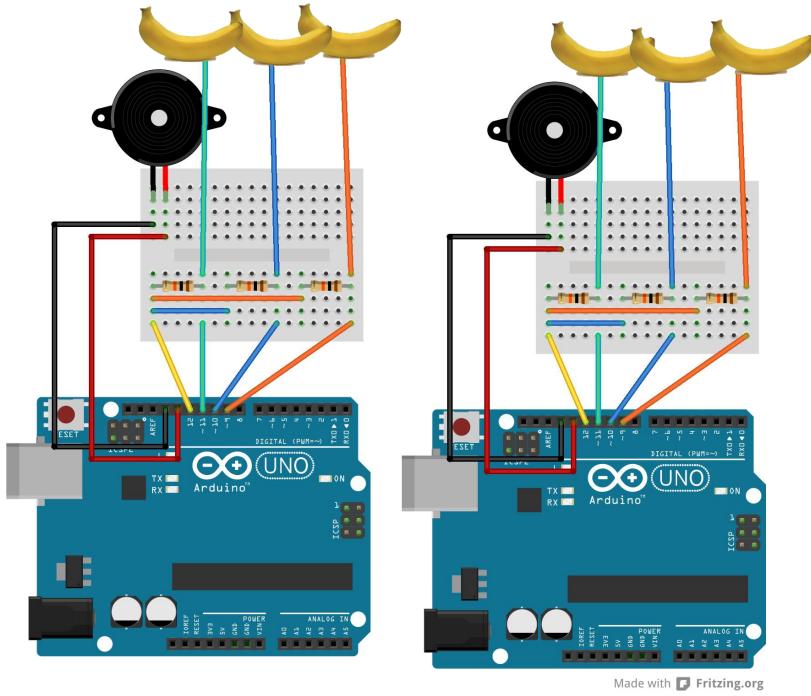
## Mind+ Interface



Here we use IF to evaluate whether someone touches the object. If touched, it changes the capacitance and trigger the buzzer. Otherwise, nothing happens. **The second and the third parameter box indicate the frequency and rhythm of the buzzer.**

**Now you are able to create one key for the banana piano. What about playing a song on multiple keyboards, like Twinkle Twinkle Little Star lol**

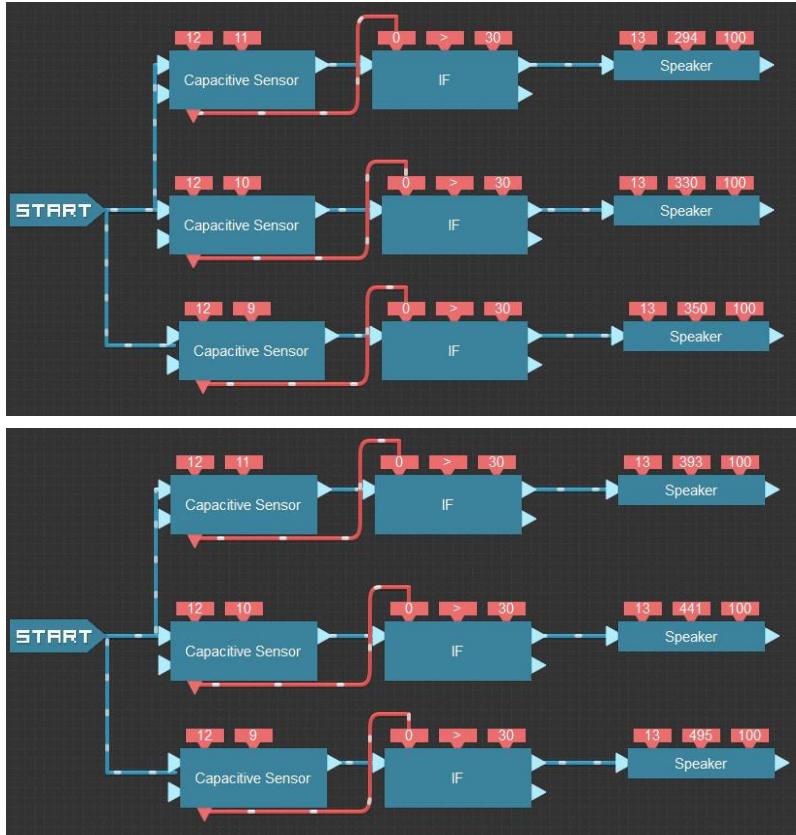
## Electrical Circuit



Made with Fritzing.org

## Mind+ Interface

**Tips:** We don't have a copy-paste function yet, but you can try the quick search function instead when you need to use same block for multiple times. Just double click on the background and enter key words after the bar pops up.



## Twinkle Twinkle Little Star

1 1 | 5 5 | 6 6 | 5 - | 4 4 | 3 33 | 2 2 | 1 - |

Twinkle, twinkle, little star. How I wonder what you are.

5 5 | 4 4 | 3 3 | 2 - | 5 5 | 4 4 | 3 3 | 2 - |

Up a - bove the world so high, Like a diamond in the sky!

5 | 6 6 | 5 - | 4 4 | 3 33 | 2 2 | 1 - ||

Twinkle, twinkle, little star. How I wonder what you are.

### Reference of frequency of notes

	1	2	3	4	5	6	7
A	221	248	278	294	330	371	416
B	248	278	294	330	371	416	467
C	131	147	165	175	196	221	248
D	147	165	175	196	221	248	278
E	165	175	196	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	234	262	294	330	371
	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	294	330	350	393	441	495	556
E	330	350	393	441	495	556	624
F	350	393	441	495	556	624	661
G	393	441	495	556	624	661	742
	1	2	3	4	5	6	7
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1178	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	589	661	700	786	882	990	1112
E	661	700	786	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

## Extended Exercise : Avatar Plant

Apart from using capacitive sensor to trigger a buzzer, we can also use it to trigger a LED. For example, make an avatar plant. When it is touched, the lights hidden in the plant are lit.

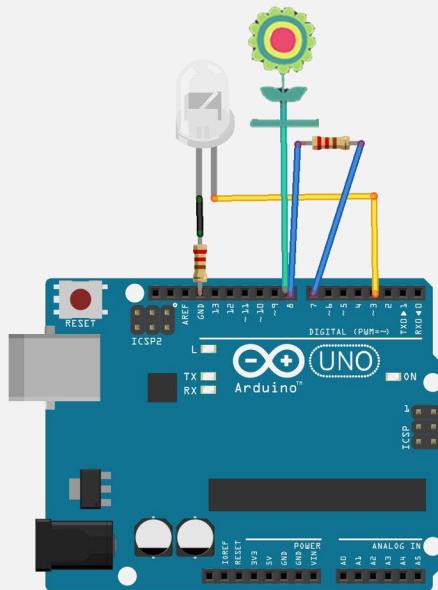


Before

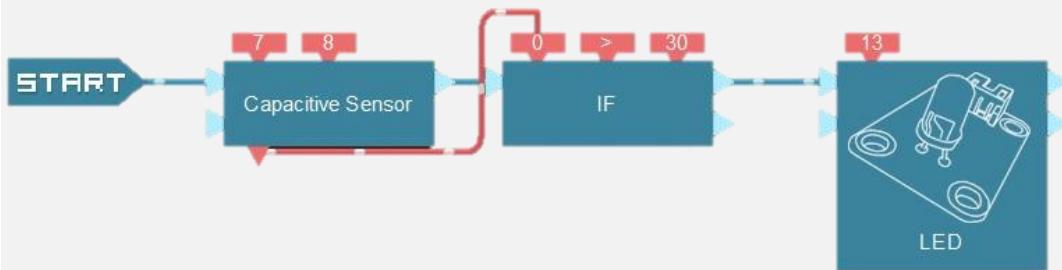


After

### Electrical Circuit



### Mind+ Interface



For more detail instruction, you can follow our step by step tutorial on Instructables.com.  
<http://www.instructables.com/id/Avatar-PlaNT-Touch-Sensitive/>

# 12 IR Sensor

## Infrared TV Killer

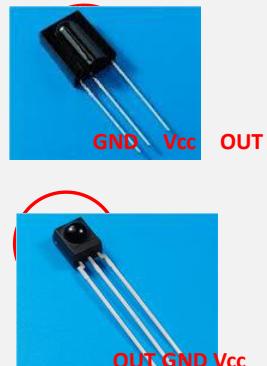
Infrared sensor is widely used for remote control detection in a short distant of 3-4 meters. In this project, we hack a TV remote controller by decoding and resending IR signal inspired by Mitch Altman' s creation TV-B-GONE, a universal remote control resembles other TV remote controls. To better demonstrate the process for all DIYers, we replace an IR remote controller with TV controller and on-board LED with TV.

### Arduino

#### Part list



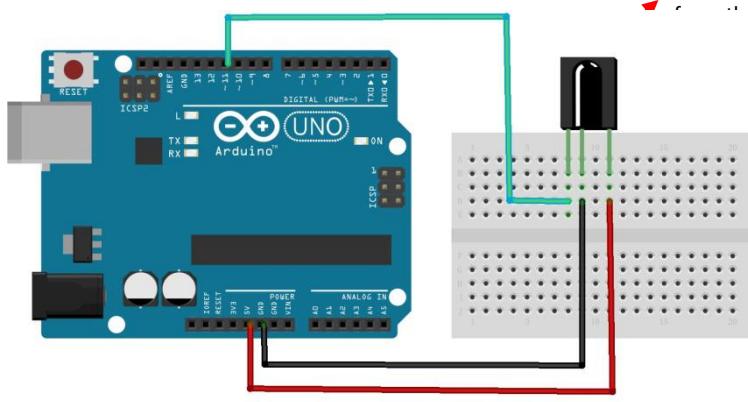
IR Detector    IR Remote Controller



#### IR Detector

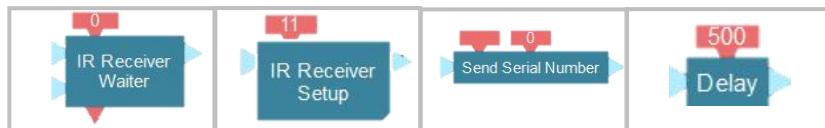
+ **Connection:** Check pictures on the left for different detectors.  
+ **Function:** IR detectors are little microchips with a photocell that are tuned to listen to infrared light. They are almost always used for remote control detection - every TV and DVD player has one of these in the front to listen for the IR signal from the clicker.

### Electrical Circuit



## Mind+

### block list



IR Receiver Waiter    IR Receiver Setup    Send Serial Number    Delay

Code you want to wait

The code examples show how each block is used in a sequence:

- IR Receiver Waiter**: On stop, it waits for a match IR command (Pin number 0) and outputs "Get match IR command".
- IR Receiver Setup**: on (Pin number 11), it finishes setup.
- Send Serial Number**: Start block, Value name R value, Data 0.
- Delay**: Start block, time 500, Output connector.

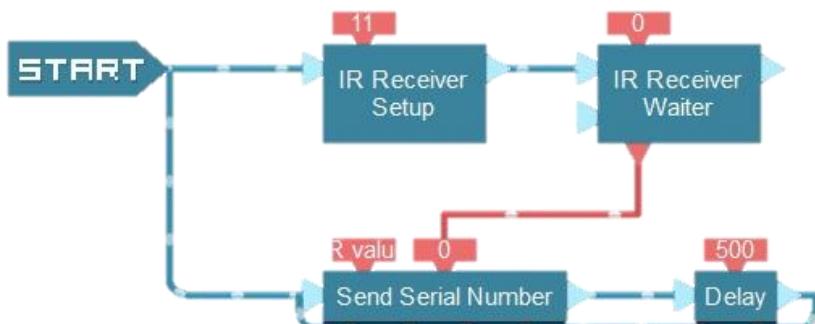
**IR Receiver Waiter**  
+ Location  
advanced block list -- IR Remote

**IR Receiver Setup**  
+ Location  
advanced block list -- IR Remote

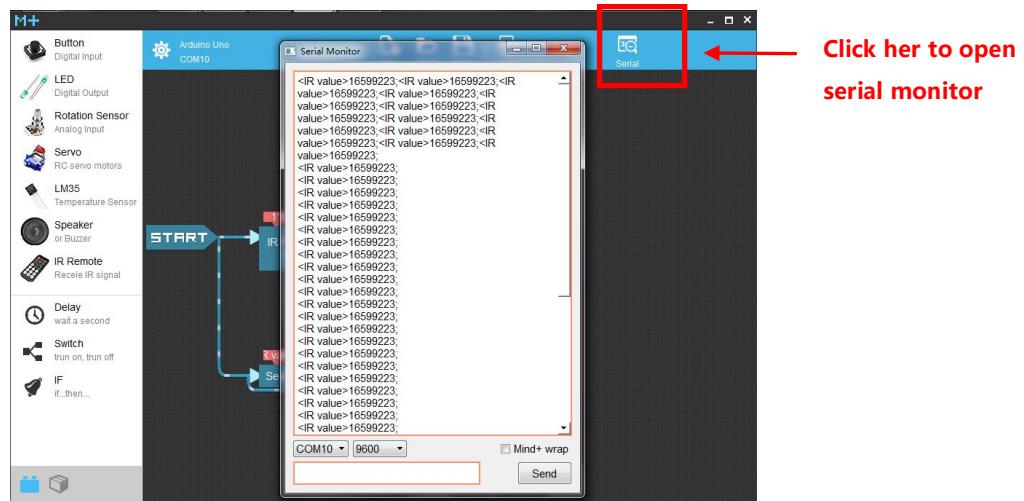
**Send Serial Number**  
+ Location  
advanced block list -- Serial -- Send Serial Number  
+ block Description  
Send Serial Number sends value from serial port.

**Delay**  
+ Location: basic block list -- Delay

### Mind+ Interface



Open the **serial monitor** from the upper right side of the interface(more details of serial monitor please check Project# 6)

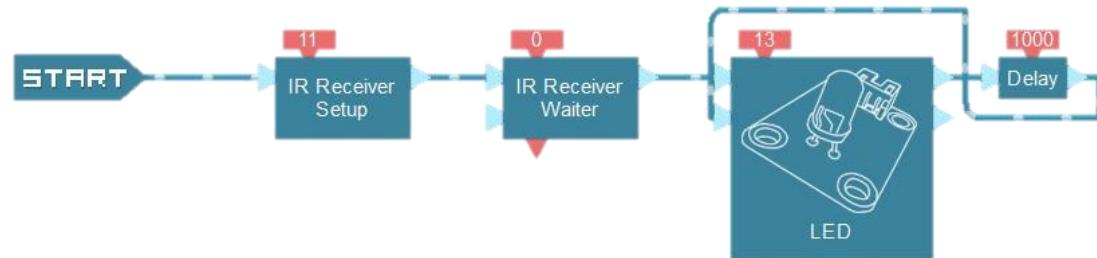


In this way, you can decode every button on the IR remote controller just like below.

<b>16580863</b>	16613503	16597183
16589023	16621663	16605343
16584943	16617583	16601263
16593103	16625743	16609423
16582903	16615543	16599223
16591063	16605343	16607383
16586983	16619623	16603303

**Now let us copy and paste IR code of the on/off button (16580863) and make a program to turn on the on board LED after receiving this code. After this, we will send the same code from another circuit to turn on and off the LED in the first circuit.**

**Note:** We need 2 Arduino boards for the following project. One to receive code as we did above, and one to send code.



# Arduino

## Part list



Infrared LED    220 resistor    pushbutton



### Infrared LED

**Connection:** long leg goes to 5V and short leg goes to GND.

**Function:** It works the same way as LED. It functions well in a distance of 3-4 meters but light cannot pass through obstacles.

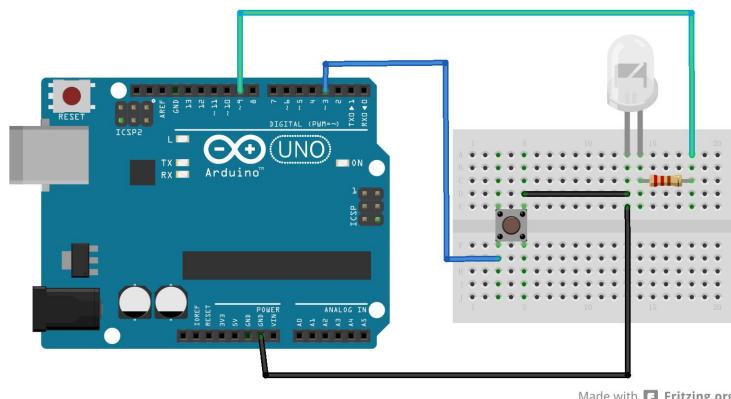
**Usage:** Infrared LED and White LED look identical but



### How to tell an infrared LED from a white LED?

Take out the button from the IR controller and attach 2 legs of the LED to the button. Long leg on + side and short leg on the - side. Now turn on the camera on your cellphone, put the top of the LED towards your camera. If you attach 2 legs to the battery , you can see light from the LED. If release, the light goes off.

## Electrical Circuit

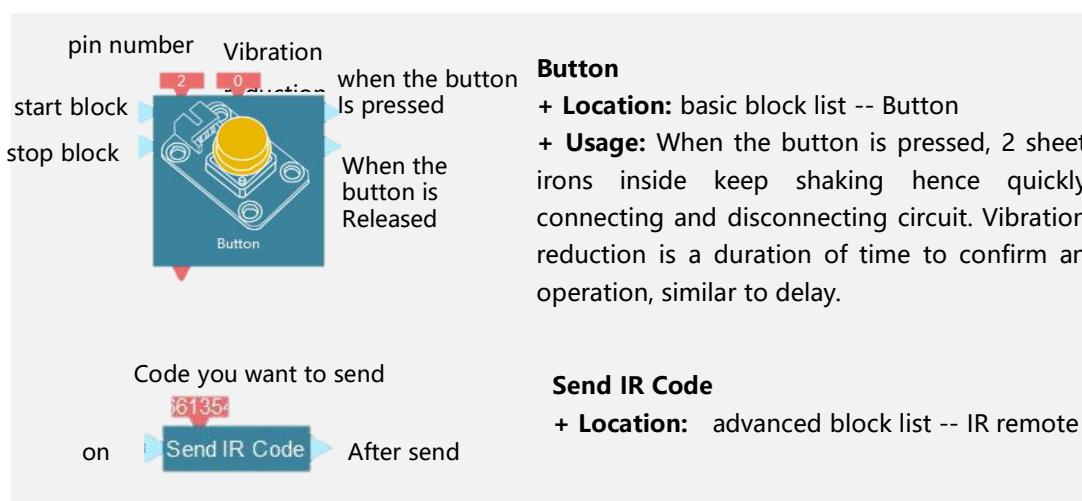


Made with Fritzing.org

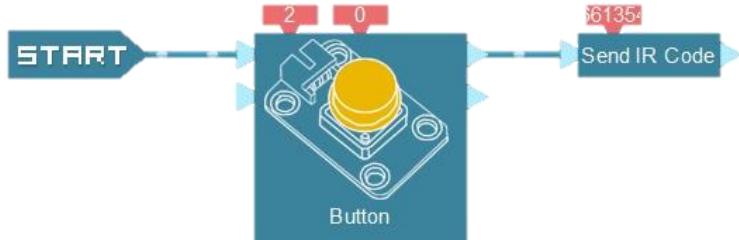
We use another Arduino to make a program to send code from the button.

## Mind+

### block List



### Mind+ Interface



When the button is pressed, it sends out IR Code to turn the on-board LED on on the first Arduino.

### CONGRATS!

So now you have finished all projects of the Mind+ DIY Starter kit. For more information, please visit our website [www.mindplus.cc](http://www.mindplus.cc) or like us on Facebook (Mindplus official). Thank you:)