



INVENTOR Electronic Kit



MAKER WORKS TECHNOLOGY INC

No. 426, F/4, Gonglehuating Business Building, Xinhu Road,
Xixiang Sub-district, Bao'an District, Shenzhen, 518102 China.

Please contact with us if you need some
technique support: support@makeblock.cc
www.makeblock.cc



www.makeblock.cc

Parts List

Pictures for reference only

1× Makeblock Orion



1× 9g Micro Servo



1× Me Ultrasonic Sensor



1× Me Light and Grayscale Sensor



1× Me PIR Motion Sensor



1× Me Sound Sensor



1× Me Temperature Sensor



1× Me 130 DC Motor



1× Fan Blade



4× Screw M4x8



4× Nylon Stud M4*30



1× Me RGB LED



1× Me Bluetooth Module



1× Me Potentiometer



1× Me RJ25 Adapter



1× Battery Holder for (6) AA



1× Me Joystick



1× Me 7-Segment Display



4× 6P6C RJ25 Cable-20cm



2× 6P6C RJ25 Cable-35cm



1× 6P6C RJ25 Cable-50cm



1× Micro USB cable



Contents

Pictures for reference only

Electronic Modules

1 ~ 14



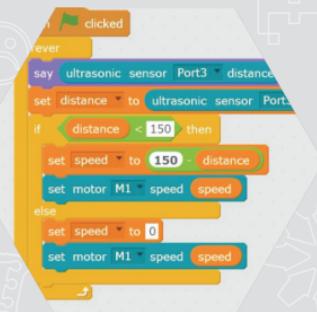
10 Fun Cases

15 ~ 25

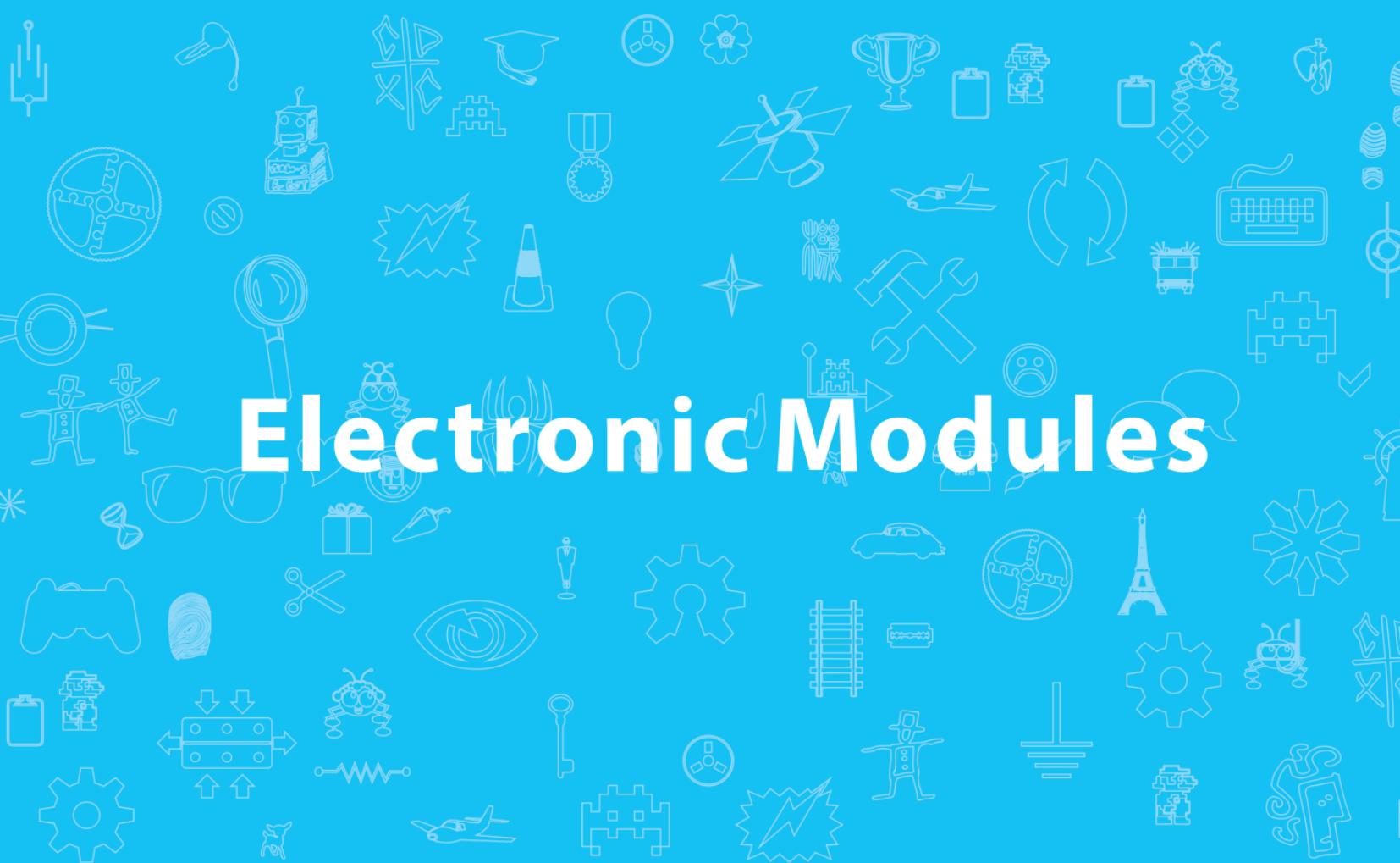


Programming Guide

26 ~ 28

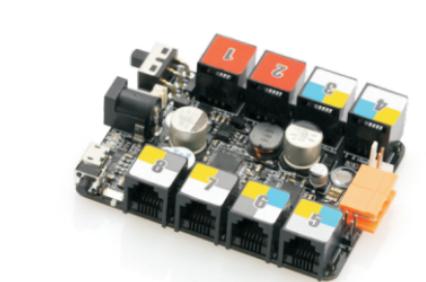


Electronic Modules



1
No.
Module

Makeblock Orion



Makeblock Orion is a mainboard based on Arduino Uno with improvements for ease of use. It provides 8 RJ25 ports to connect to all the other Me-series modules.

Features

Microcontroller: ATmega328;
100% Arduino compatible;
Two-channel motor driver integrated;
Switchable buzzer;
6-12V DC power supply;
Comes with Arduino library for easy programming;
Easy wiring with 6-Pin RJ25 interfaces;

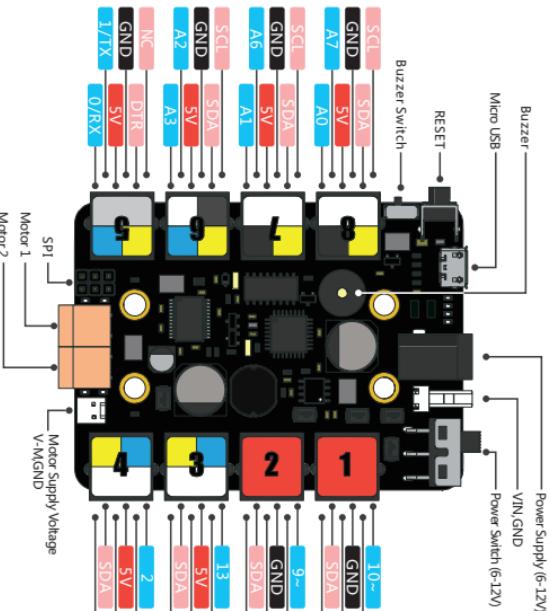
Note:

Intuitive connection by labeling tags with different color and number for each connector

-  Output voltage of power
-  One way digital interface
-  Dual digital interface
-  Dual & one way analog interface
-  Hardware serial port
-  I2C port

<http://learn.makeblock.cc/orion>

Main Control Board



Me Ultrasonic Sensor



Ultrasonic module works for measuring distance from 3cm to 400cm. The module's interface corresponds to the port with yellow tag in mainboard.



Features:

Rated Voltage: 5V DC power;
Detecting Range: 3cm ~ 400 cm;
Detecting Angle: about 30 degrees;
Dimension: 56 x 36 x 31mm (Length x Width x Height);

Function	Description
MeUltrasonicSensor (uint8_t port)	Constructs the object with port number.
long distanceInch()	Measures the distance in inches.
long distanceCm()	Measures the distance in centimeters.



TestUltrasonicSensor

Me 7-Segment Serial Display



Me 7-Segment Display combines a classic 4-digit 7-segment display and a TM1637 allowing you to control each segment individually and easily. This module's interface corresponds to the port with blue tag in mainboard.

```
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeUltrasonicSensor ultraSensor(PORT_6);
void setup()
{
//initialize serial communications at 9600 bps
Serial.begin(9600);
}

void loop()
{
Serial.print("Distance : ");
Serial.print(ultraSensor.distanceCm());
//print the measure centimeter of distance
Serial.println(" cm ");
Serial.print(ultraSensor.distanceInch());
//print the measure inch of distance
Serial.println(" inch ");
delay(100);
// the minimal measure interval is 100
milliseconds
}
```

Features:

Rated Voltage: 5V DC power;
Display Range: 0.000 ~ 9,999;
Dimension: 52 x 24 x 24 mm (Length x Width x Height);

Function	Description
Me7SegmentDisplay (uint8_t port)	Constructs the object with port number.
void init()	Initializes the module and clears the display.
void set(uint8_t brightness, uint8_t setData, uint8_t setAddr)	Adjusts the brightness and set data to specific address.
void display(float value)	
void display(int8_t value)	
void display(uint8_t BitAddr, int8_t DispData)	Displays numbers.

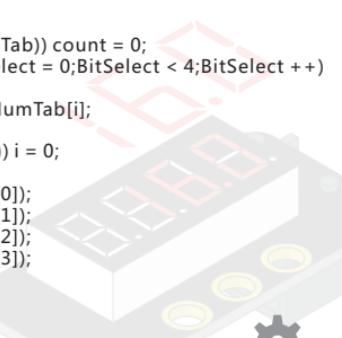


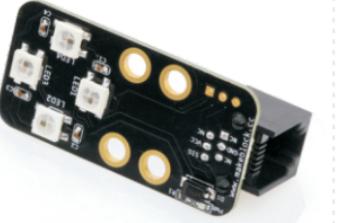
NumberFlow

```
#include <Makeblock.h>
#include <Wire.h>
#include <SoftwareSerial.h>

Me7SegmentDisplay disp(PORT_6);

void setup()
{
disp.init();
disp.set(2);
//Set the display brightness, the brightness range is from 0 to 7
}
void loop()
{
int8_t NumTab[] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
//0~9,A,b,C,d,E,F
int8_t ListDisp[4];
unsigned char i = 0;
unsigned char count = 0;
delay(150);
while(1)
{
i = count;
count++;
if(count == sizeof(NumTab)) count = 0;
for(unsigned char BitSelect = 0;BitSelect < 4;BitSelect++)
{
ListDisp[BitSelect] = NumTab[i];
i++;
if(i == sizeof(NumTab)) i = 0;
}
disp.display(0,ListDisp[0]);
disp.display(1,ListDisp[1]);
disp.display(2,ListDisp[2]);
disp.display(3,ListDisp[3]);
delay(300);
}
```



Me RGB LED

Me RGB LED module contains four programmable RGB LEDs. Color of each LED can be set by editing components of red, green, and blue. This module's interface corresponds to the port with yellow tag in mainboard.

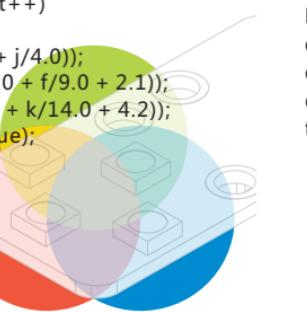
Features:

Rated Voltage: 5V DC Power;
LED Count: 4 RGB LEDs;
Peak Current: 60 mA each, 240 mA in total;
Dimension: 52 x 24 x 18 mm (Length x Width x Height);

Function	Description
MeRGBLed(uint8_t port)	Constructs the object with port number.
void show()	Applies all the settings of LEDs before.
void setNumber(uint8_t num_led)	Sets the total number of LEDs.
bool setColorAt(uint8_t index, uint8_t red, uint8_t green, uint8_t blue)	Sets the RGB components of one LED.
uint8_t getNumber	Gets the number of LEDs.
cRGB getColorAt(uint8_t index)	Gets the RGB components of one LED.

**Me Joystick**

Me Joystick is designed to control objects' movements in XY-axis direction. This module's interface corresponds to the port with black tag in mainboard;

**Features:**

Rated Voltage: 5V DC Power;
Control Direction: XY-axis;
Signal Type: Analog (range from 0 to 4.8 V);
Dimension: 52 x 24 x 32 mm (Length x Width x Height);

Function	Description
MeJoystick(uint8_t port)	Constructs the object with port number.
int readX()	Gets the X component of joystick.
int readY()	Gets the Y component of joystick.



```
#include <Makeblock.h>
#include <SoftwareSerial.h>
#include <Wire.h>
```

```
MeJoystick joystick(PORT_6);
```

```
int x = 0; // a variable for the Joystick's x value
int y = 0; // a variable for the Joystick's y value
```

```
void setup()
{
    // initialize serial communications at 9600 bps
    Serial.begin(9600);
}
```

```
void loop()
{
    // read the both joystick axis values:
    x = joystick.readX();
    y = joystick.readY();
```

```
// print the results to the serial monitor:
Serial.print("Joystick X = ");
Serial.print(x);
Serial.print("\t Joystick Y = ");
Serial.println(y);
// wait 100 milliseconds before the next loop
delay(100);
}
```



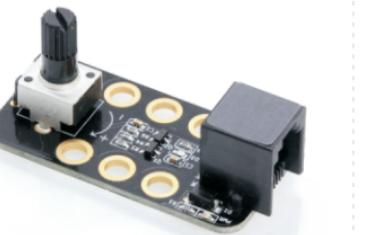
Me Light and Grayscale Sensor



Me Light and Grayscale Sensor can detect ambient light intensity and light intensity of reflection. There is a LED on circuit board for brightness compensation. This module's interface corresponds to the port with black tag in mainboard;



Me Potentiometer



Me Potentiometer module works as an adjustable potentiometer with 10K Rmax. This module's interface corresponds to the port with black tag in mainboard.

MeLightSensorTest

```
#include <Makeblock.h>
#include <SoftwareSerial.h>
#include <Wire.h>

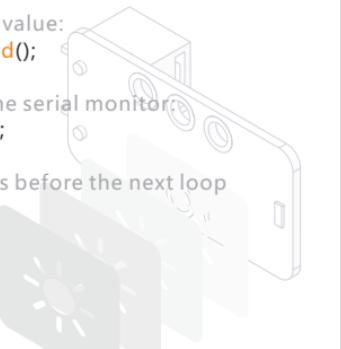
MeLightSensor lightSensor(PORT_6);

int value = 0;
// a variable for the lightSensor's value

void setup()
{
    // initialize serial communications at 9600 bps
    Serial.begin(9600);
}

void loop()
{
    // read the lightSensor value:
    value = lightSensor.read();

    // print the results to the serial monitor:
    Serial.print("value = ");
    Serial.println(value);
    // wait 100 milliseconds before the next loop
    delay(100);
}
```



Features:

Rated Voltage: 5V DC Power;
 Signal Type: Analog voltage(0V ~ 4.8V);
 Output Analog Values:
 Expose to daylight (> 500), night (0 ~ 100), indoor lighting situation (100 ~ 500);
 Dimension: 52 x 24 x 16 mm (Length x Width x Height);

Function	Description
MeLightSensor (uint8_t port)	Constructs the object with port number.
int read()	Gets the brightness
void lightOn()	Turns the LED on.
void lightOff()	Turns the LED off.



Me Potentiometer

```
#include <Makeblock.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MePotentiometer potentiometer(PORT_6);

int val = 0;

void setup()
{
    //initialize serial communications at 9600 bps
    Serial.begin(9600);
}

void loop()
{
    //read the potentiometer values:
    val = potentiometer.read();

    //print the results to the serial monitor:
    Serial.print("Potentiometer = ");
    Serial.println(val);

    //wait 100 milliseconds before the next loop
    delay(100);
}
```

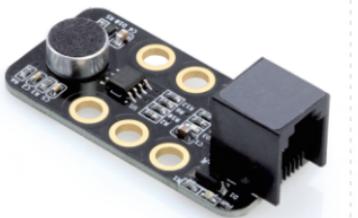


Features:

Rated Voltage: 5V DC Power;
 Signal Type: Analog voltage(0V ~ 4.8V);
 Dimension: 52 x 24 x 22 mm (Length x Width x Height);

Function	Description
MePotentiometer (uint8_t port)	Constructs the object with port number.
uint16_t read()	Reads the analog voltage of the potentiometer.

Me Sound Sensor



Me Sound Sensor is designed to measure the volume. The module's interface corresponds to the port with black tag in mainboard.

Features:

Rated Voltage: 5V DC Power;
Signal Type: Analog voltage(0.2V ~ 3V);
Dimension: 52 x 24 x 17 mm (Length x Width x Height);

Function	Description
MeSoundSensor (uint8_t port)	Constructs the object with port number.
int strength()	Measures the volume.



Me_Sound_Sensor

```
#include <Makeblock.h>
#include <SoftwareSerial.h>
#include <Wire.h>

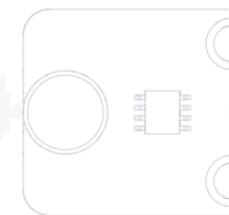
MeSoundSensor SoundSensor(PORT_6);
MeRGBLed led(PORT_3);

int analog_val = 0;
// a variable for the SoundSensor's analog result
int k=0;

void setup()
{
    led.setNumber(4);
}

void loop()
{
    //read the SoundSensor's result:
    analog_val = SoundSensor.strength();
    k=abs(analog_val-486);

    if(k>5 && k<480)
    {
        for(int t=0; t<4; t++)
            led.setColorAt(t,k,k);
        led.show();
        delay(50);
    }
    else
    {
        for(int t=0;t<4;t++)
            led.setColorAt(t,0,0,0);
        led.show();
    }
}
```



Me Temperature Sensor



Me Temperature Sensor is a stainless steel tube with DS18B20 sensor which is used for measuring temperature. RJ25 Adapter module is needed to connect it to port with blue tag in mainboard.

Features:

Rated Voltage: 3.0V ~ 5.5V DC Power;
Working Range: -55 ~ 125 (degrees centigrade);
Pins: Red (VCC), Yellow (DATA), Black (GND);
Dimension: Stainless steel (6 x25mm);
Cable length: (50cm);

Function	Description
MeTemperature (uint8_t port, uint8_t slot)	Constructs the object with port number and slot number.
void reset(uint8_t port, uint8_t slot)	Re-initializes the port and slot.
float temperature()	Measures the temperature.



TestTemperature

```
#include <Makeblock.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeTemperature Temperature_val(PORT_6, SLOT1);

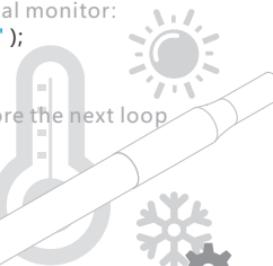
float Temp = 0; // a variable for the Temperature

void setup()
{
    // initialize serial communications at 9600 bps
    Serial.begin(9600);
    Temperature_val.reset(PORT_6, SLOT1);
}

void loop()
{
    // read the Temperature values:
    Temp = Temperature_val.temperature();

    // print the results to the serial monitor:
    Serial.print("Temperature = ");
    Serial.println(Temp);

    // wait 100 milliseconds before the next loop
    delay(100);
}
```



Me PIR Motion Sensor



9g Micro Servo



Me PIR Motion Sensor can detect infrared ray which derives from animals/humans in less than 6 meters. This module's interface corresponds to the port with blue tag in mainboard.

Function	Description
MePIRMotionSensor (uint8_t port)	Constructs the object with port number.
bool isPeopleDetected()	Detect whether there's people moving or not.

Features:

Rated Voltage: 5V DC Power;
Holding Time: 2s;
Detecting Angle: 120 Degrees;
Dimension: 51 x 24 x 16 mm (Length x Width x Height);

[Test_PIRMotionSensor](#)

```
#include <Makeblock.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MePIRMotionSensor PIRmotion_val(PORT_3);

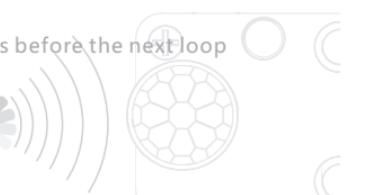
bool value = 0;
// a variable for the PIR Motion Sensor

void setup()
{
    // initialize serial communications at 9600 bps
    Serial.begin(9600);
}

void loop()
{
    // read the PIR Motion Sensor values:
    value = PIRmotion_val.isPeopleDetected();

    // print the results to the serial monitor:
    Serial.print("PIR Motion Values = ");
    Serial.println(value);

    // wait 100 milliseconds before the next loop
    delay(100);
}
```


<http://learn.makeblock.cc/pir-motion-sensor/>


Me Micro Servo weights 9 grams and can rotate from 0 to 120 degrees. RJ25 Adapter module is needed to connect it to port with blue tag in mainboard.

Function	Description
void attach (int pin)	Attaches the servo to the pin.
void write (byte angle)	Sets the angle of servo.

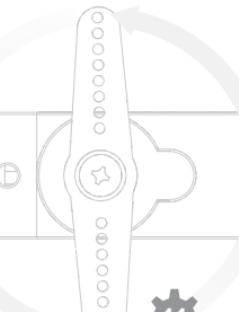
```
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>
```

```
#include <Servo.h> //include the Servo library;
```

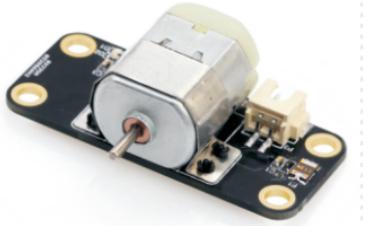
```
MePort port(PORT_6);
Servo myservo1;
int servo1pin = port.pin1();
```

```
void setup()
{
    myservo1.attach(servo1pin);
}
```

```
void loop()
{
    myservo1.write(0);
    delay(2000);
    myservo1.write(180);
    delay(2000);
}
```


<http://learn.makeblock.cc/micro-servo/>

Me 130 DC Motor



Me 130 DC Motor works with 3V to 12V power supply. It needs to be connected to motor interface (M1 or M2) in mainboard.

Features:

Rated Voltage: 3V ~ 12V;
Shaft Diameter: 2mm;
Shaft Length: 7.5mm;
Rotation Speed: 7 500 rpm (no-load);
Dimension: 56 x 24 x 17mm (Length x Width x Height);

Function	Description
MeDCMotor (uint8_t port)	Constructs the objects with port number.
void run (int speed)	Sets the speed of motor.
void stop()	Stops the motor.

<http://learn.makeblock.cc/130-dc-motor/>



Me_130_DC_Motor



Me RJ25 Adapter



13

Features:

Rated Voltage: 5V DC power;
Connector: two signal connectors;
Dimension: 52 x 24 x 16mm(Length x Width x Height);

Function	Description
MePort (uint8_t port)	Constructs the object with port number.
void dWrite1 (bool value)	Sets the digital voltage of pin 1.
bool dRead1()	Reads the digital voltage of pin 1.

<http://learn.makeblock.cc/rj25-adapter/>



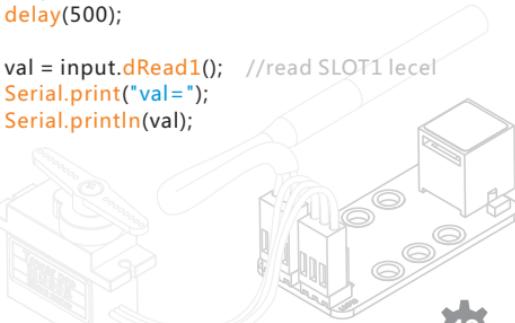
TestRJ25Adapter

```
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>
```

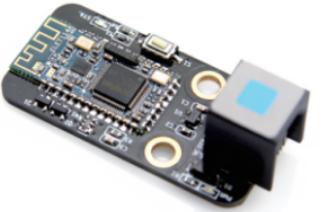
```
MePort output(PORT_4);
MePort input(PORT_3);
```

```
int val;
void setup()
{
    Serial.begin(9600);
}
void loop()
{
    output.dWrite1(HIGH); //SLOT1 HIGH
    delay(500);
    output.dWrite1(LOW); //SLOT1 LOW
    delay(500);

    val = input.dRead1(); //read SLOT1 lecel
    Serial.print("val=");
    Serial.println(val);
}
```



Me Bluetooth Module (Dual Mode)



Me Bluetooth Module is designed to communicate with devices with bluetooth function through specific software. We provide an Android App to use it. It is compatible with bluetooth 2.0 and 4.0. This module should be connected to PORT_5 in mainboard.

Features:

Rated Voltage: 5V DC power;
Bluetooth 2.0 and 4.0 compatible;
Level Output Voltage: 5V/high, 0V/low;
Dimension: 52 x 24 x 16mm (Length x Width x Height);
LED Indicator:
PWR (Red LED) - Power indicator.
STA (Blue LED) - Slow blinking indicates waiting to be paired. Always ON indicates the connection has been established;

Function	Description
MeBluetooth (uint8_t port)	Constructs the object with port number.
void begin (long baudrate)	Sets the baudrate of serial port and start it.
byte available()	Gets the number of data available for reading.
int read()	Reads one byte from received buffer.
size_t write (byte value)	Writes one byte through the serial port.

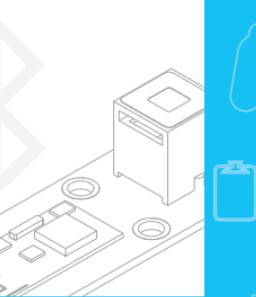
<http://learn.makeblock.cc/bluetooth-module/>

```
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

MeBluetooth bluetooth(PORT_5);
//Bluetooth is the use of a hardware serial port.
//The hardware serial port of Makeblock Orion is PORT_5
// The hardware serial port of Me Baseboard is PORT_4

void setup()
{
  Serial.begin(115200);
  bluetooth.begin(115200);
  Serial.println("Bluetooth Start!");
}

void loop()
{
  char inDat;
  char outDat;
  if(bluetooth.available())
  {
    inDat = bluetooth.read();
    Serial.print(inDat);
  }
  if(Serial.available())
  {
    outDat = Serial.read();
    bluetooth.write(outDat);
  }
}
```



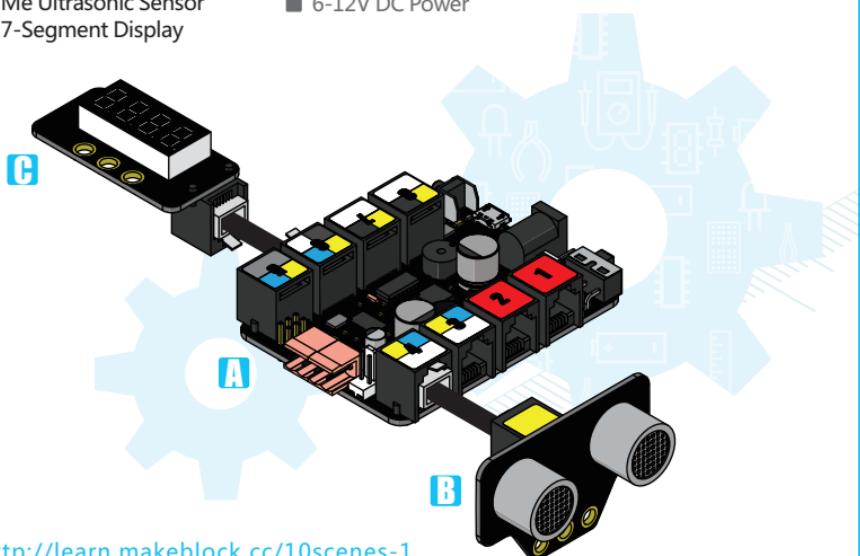
DistanceDisplay

Difficulty: *

Detected by Ultrasonic Sensor, the 7-Segment Display can show the distance data.

Prepare:

- A Makeblock Orion
- B Me Ultrasonic Sensor
- C 7-Segment Display
- D DIY Mount Support
- E 6-12V DC Power



<http://learn.makeblock.cc/10scenes-1>

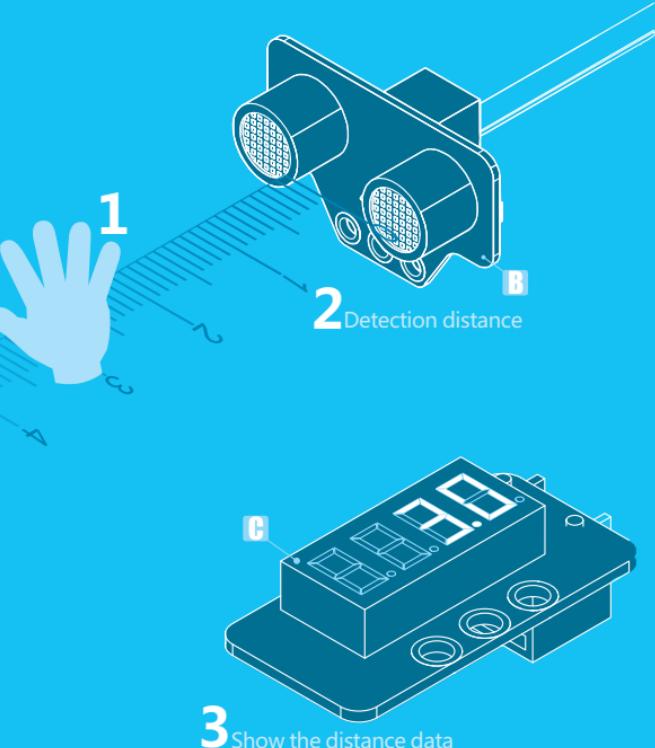
DistanceDisplay

Difficulty: *

The closer Ultrasonic Sensor approach the obstacle, the faster fan rotate.

Prepare:

- A Makeblock Orion
- B Me Ultrasonic Sensor
- C Me 130 DC Motor
- D DIY Mount Support
- E 6-12V DC Power



<http://learn.makeblock.cc/10scenes-2>

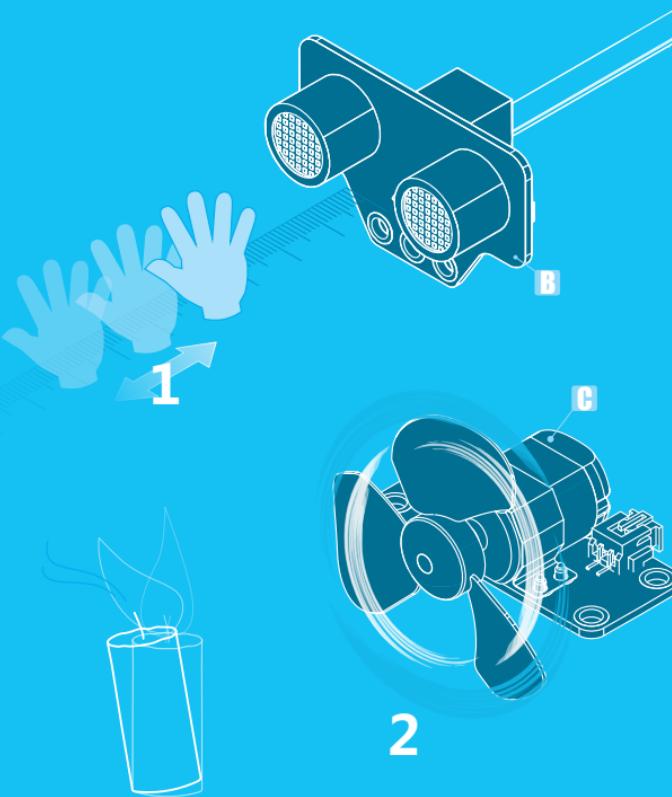
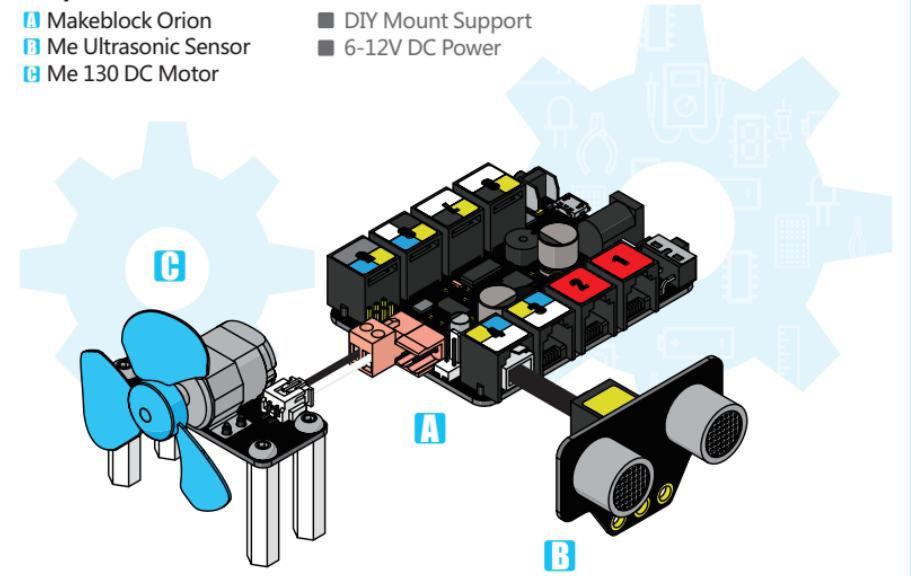
Smart Fan

Difficulty: **

The closer Ultrasonic Sensor approach the obstacle, the faster fan rotate.

Prepare:

- A Makeblock Orion
- B Me Ultrasonic Sensor
- C Me 130 DC Motor
- D DIY Mount Support
- E 6-12V DC Power



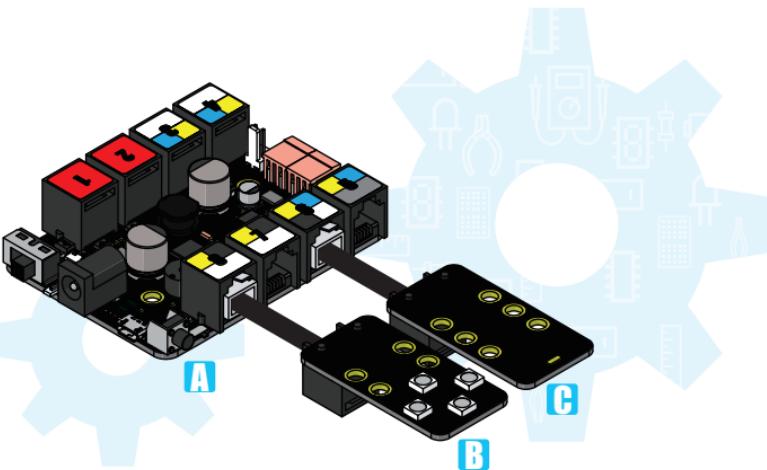
Romantic Sky

Difficulty: ★★

The LED rear of the card will be lighted in the dark.

Prepare:

- A Makeblock Orion
- B Me RGB LED
- C Me Light and Grayscale Sensor
- Carton, Scissors, Glue
- 6-12V DC Power



<http://learn.makeblock.cc/10scenes-3>

Romantic Sky

Auto Fish Feeder

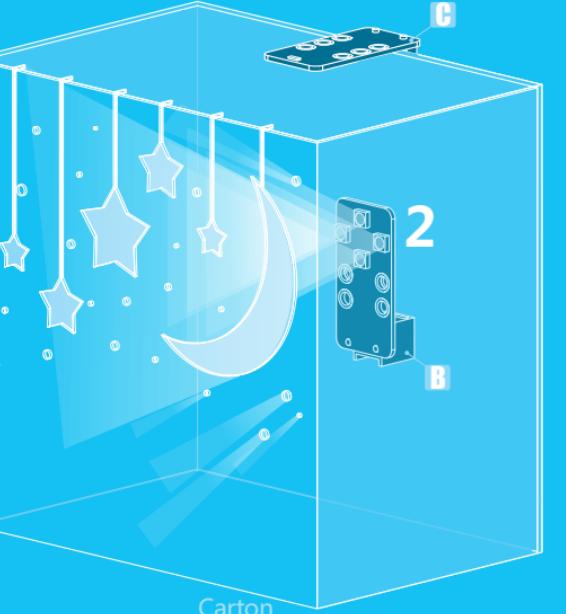
Difficulty: ★★

The servo is connected to the food container to dispense fish feed.

The 7-Segment Display shows the time before next feeding.

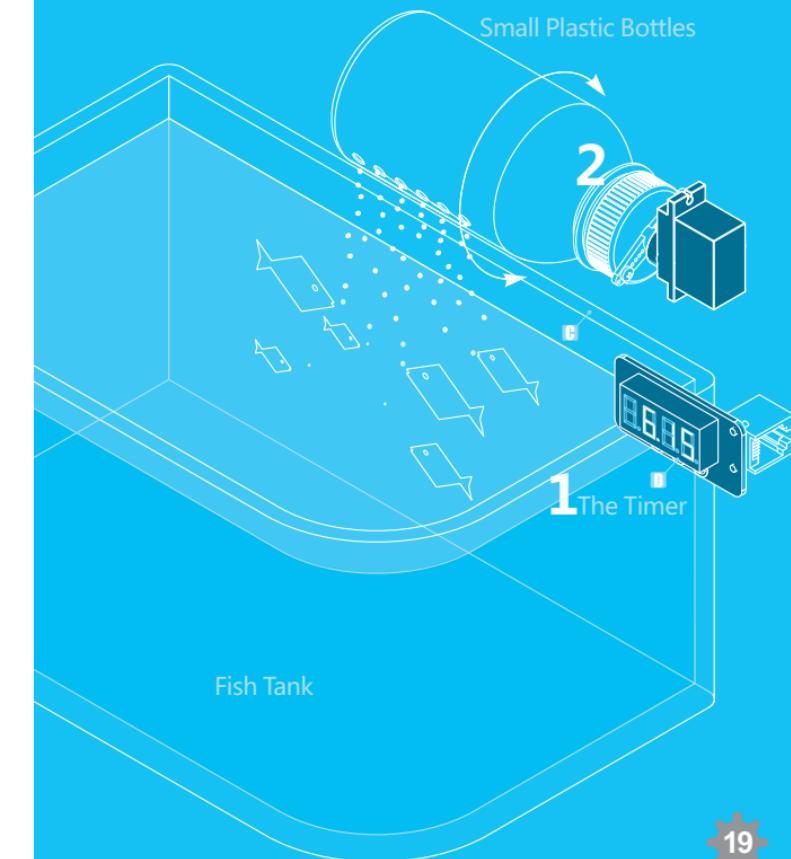
Prepare:

- A Makeblock Orion
- B Me RJ25 Adapter
- C 9g Micro Servo
- D 7-Segment Display
- Small Plastic Bottles
- DIY Mount Support
- 6-12V DC Power



<http://learn.makeblock.cc/10scenes-4>

Auto Fish Feeder



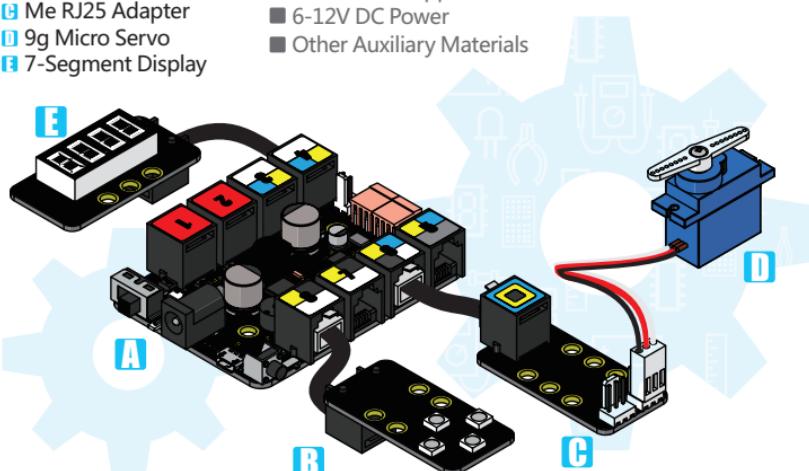
Simulation Traffic Light

Difficulty: ★★★

To simulate the RGB as the traffic lights , the 7-Segment Display show the interval time of the RGB change, the servo motor can pivot the fence as well.

Prepare:

- A Makeblock Orion
- B Me RGB LED
- C Me RJ25 Adapter
- D 9g Micro Servo
- E 7-Segment Display
- Small Fence
- DIY Mount Support
- 6-12V DC Power
- Other Auxiliary Materials


<http://learn.makeblock.cc/10scenes-5>

Simulation Traffic Light

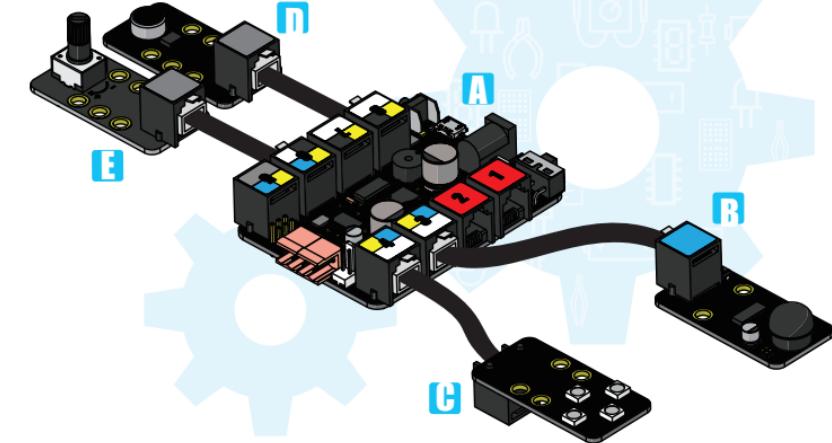
Difficulty: ★★★



The lamp can light up when someone makes sound or goes through. The Brightness of the lamp is adjustable.

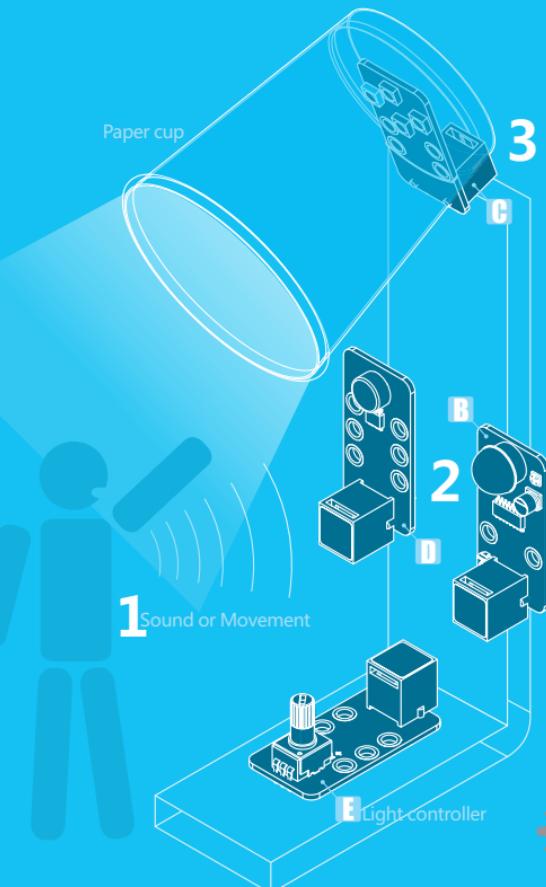
Prepare:

- A Makeblock Orion
- B Me PIR Motion Sensor
- C Me RGB LED
- D Me Sound Sensor
- E Me Potentiometer
- Paper Cup
- DIY Mount Support
- 6-12V DC Power


<http://learn.makeblock.cc/10scenes-6>

Smart Table Lamp

Difficulty: ★★★



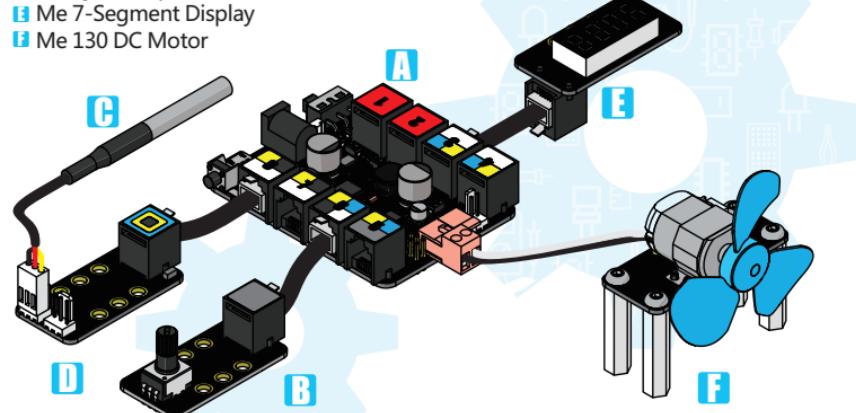
Cooling Machine

Difficulty: ★★★

Potentiometer sets the temperature as reservation, 7-Segment Display shows the setting data. If the temperature of the beverage is higher than the reserved one, 130 DC motor can work, conversely, it can stop.

Prepare:

- A Makeblock Orion
- B Me Potentiometer
- C Me Temperature Sensor
- D Me Rj25 Adapter
- E Me 7-Segment Display
- F Me 130 DC Motor
- DIY Mount Support
- 6-12V DC Power


<http://learn.makeblock.cc/10scenes-7>

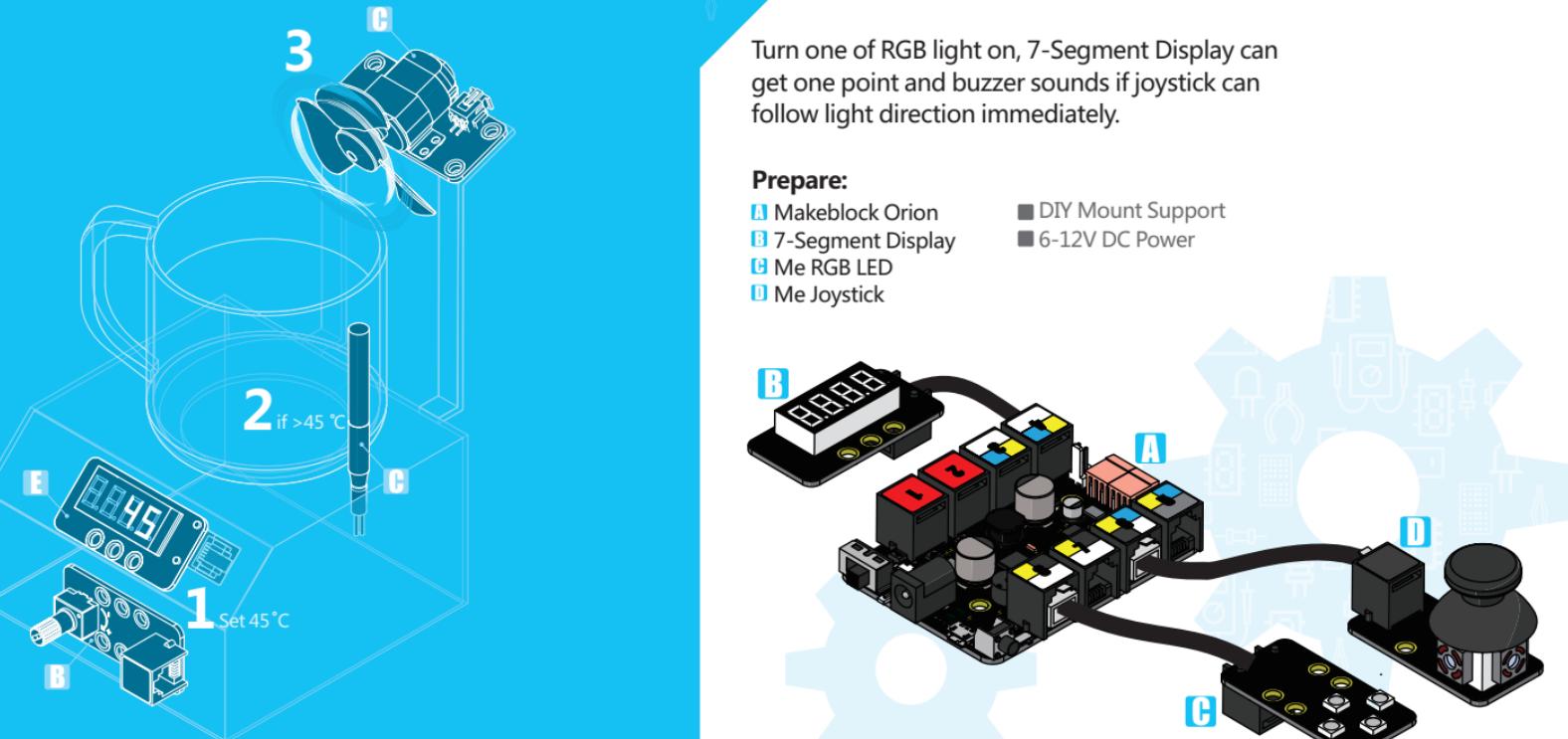
Cooling Machine

Difficulty: ★★★

Turn one of RGB light on, 7-Segment Display can get one point and buzzer sounds if joystick can follow light direction immediately.

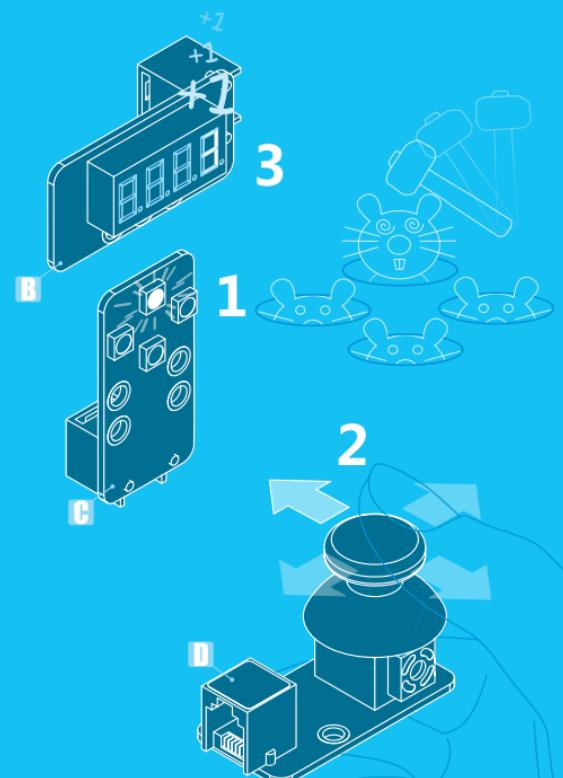
Prepare:

- A Makeblock Orion
- B 7-Segment Display
- C Me RGB LED
- D Me Joystick
- DIY Mount Support
- 6-12V DC Power


<http://learn.makeblock.cc/10scenes-8>

Whac-A-Mole

Difficulty: ★★★



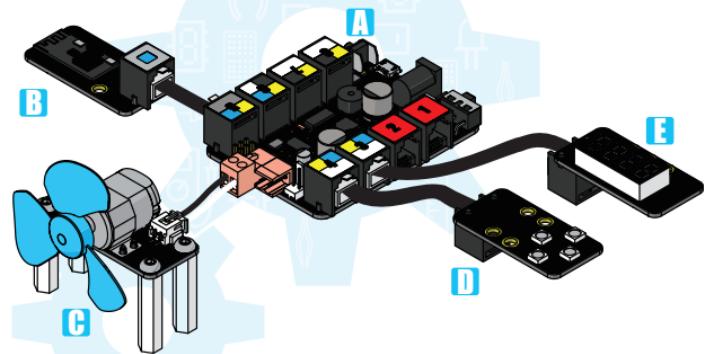
Bluetooth Control

Difficulty: ★★★

Bluetooth device(mobile device) control the color and brightness of RGB light, the speed of 130 DC motor as well, 7-Segment Display shows the time.

Prepare:

- A Makeblock Orion
- B Me Bluetooth
- C Me 130 DC Motor
- D Me RGB LED
- E Me 7-Segment Display
- F Paper Cup
- G Thin Paper
- H DIY Mount Support
- I 6-12V DC Power


<http://learn.makeblock.cc/10scenes-9>

Bluetooth control

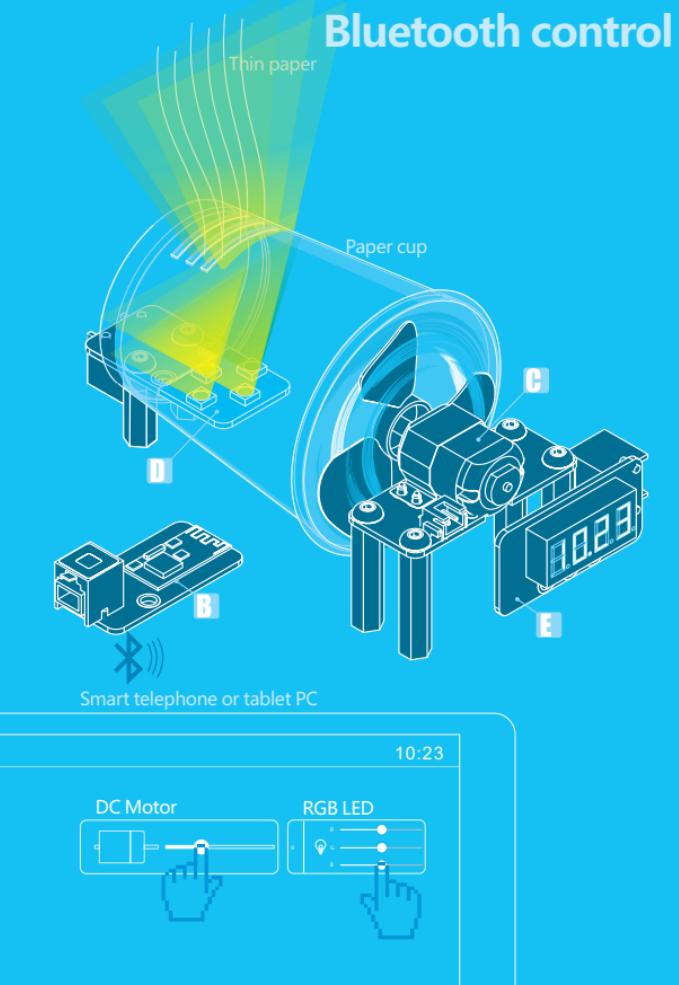
Bubble Machine

Difficulty: ★★★★

Being close to the body by PIR Motion Sensor, the bubble can be happen when 130 DC motor blow to the steel ring with soap.

Prepare:

- A Makeblock Orion
- B Me PIR Motion Sensor
- C Me 130 DC Motor
- D 9g Servo Motor
- E Me RJ25 Adapter
- F Ring
- G Soap Water
- H Cup
- I DIY Mount Support
- J 6-12V DC Power


<http://learn.makeblock.cc/10scenes-10>

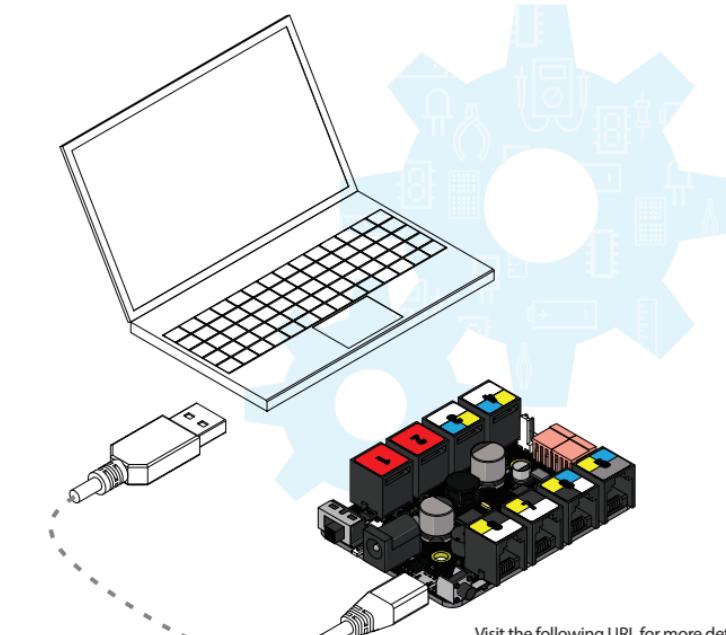
Bubble Machine

Programming Guide

If you want to creat advanced and customized programs, please refer to this part.

Makeblock Orion Driver Installation

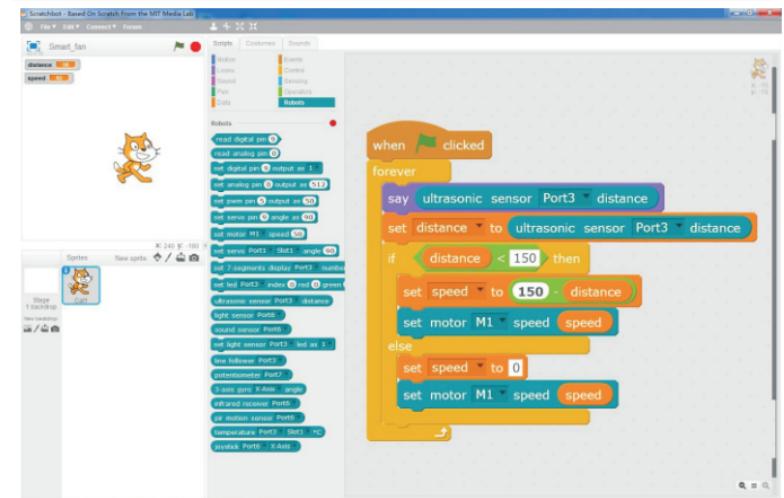
The USB interface of Makeblock Orion uses the chip CH340,you may need to install the chip CH340 drivers manually.



Visit the following URL for more details:
http://learn.makeblock.cc/driver_installation/

Working with Scratch —Hack the Physical World

The Inventor Electronic Kits helps you seamlessly combine the Scratch programming language and electronic module. Simply drag and joint the blocks to run your creations without difficult coding.



Visit the following URL for more details:
<http://learn.makeblock.cc/learning-scratch>

Scratch is a programming language and online community where you can create your own interactive stories, games, and animations – and share your creations with others around the world. In the process of designing and programming Scratch projects, young people learn to think creatively, reason systematically, and work collaboratively. Scratch is a project of the Lifelong Kindergarten group at the MIT Media Lab. It is available for free at <http://scratch.mit.edu>

Working with Arduino IDE

—Learn Programming the Fun Way

```
TestSlaveBluetoothBySoftSerial | Arduino 1.0.5-r2
File Edit Sketch Tools Help
TestSlaveBluetoothBySoftSerial $ http://learn.makeblock.cc/
#include <Makeblock.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <Wire.h>

void setup()
{
    Serial.begin(9600);
    bluetooth.begin(9600);
    Serial.println("Bluetooth Start!");
}

void loop()
{
    char inDat;
    char outDat;
    if(bluetooth.available())
    {
        inDat = bluetooth.read();
        Serial.print(inDat);
    }
    if(Serial.available())
    {
        outDat = Serial.read();
        bluetooth.write(outDat);
    }
}
```

Learn and play at the same time! Enjoy it your own way.

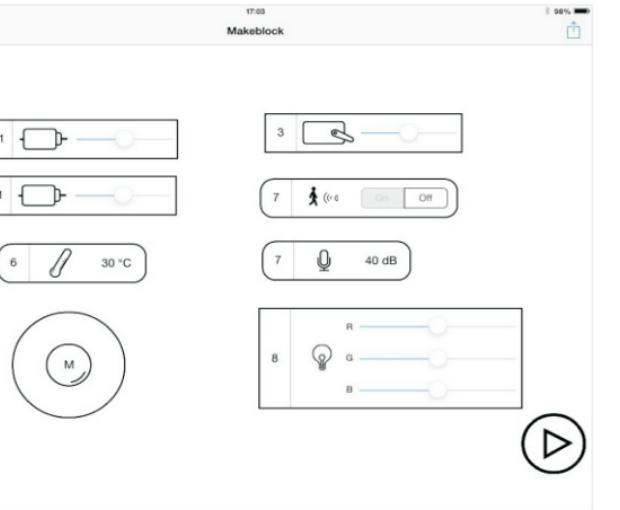
Visit the following URL for more details:
<http://learn.makeblock.cc/learning-arduino>

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. The Arduino software consists of a development environment (IDE) and the core libraries. The IDE is written in Java and based on the Processing development environment. The core libraries are written in C and C++ and compiled using avr-gcc and AVR Libc.

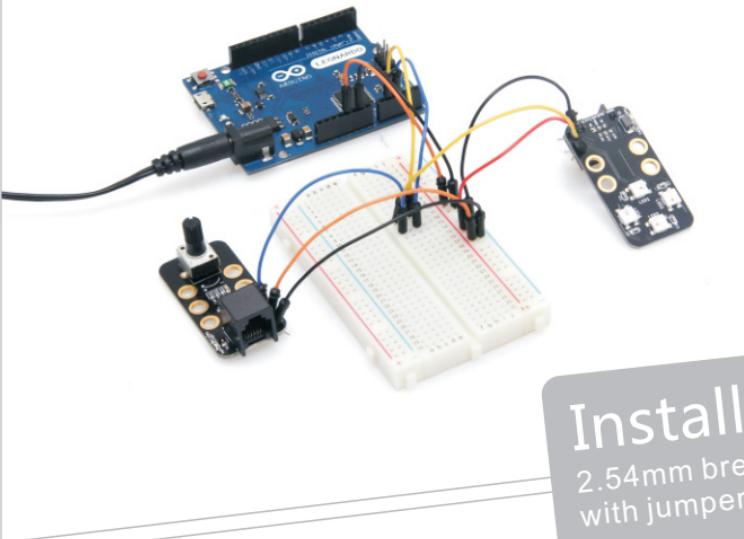
Working with the Makeblock App

—Construct It Once, Control It Everywhere

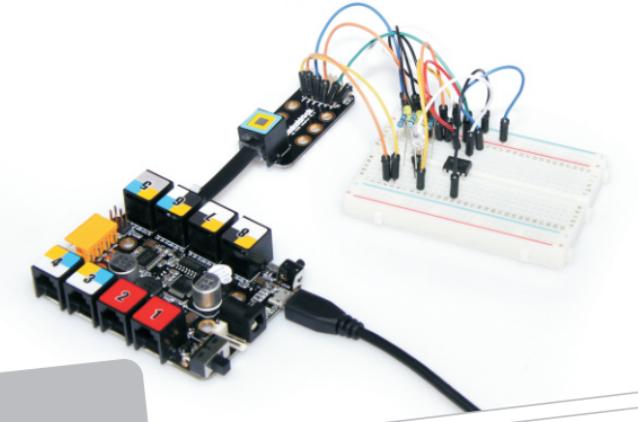
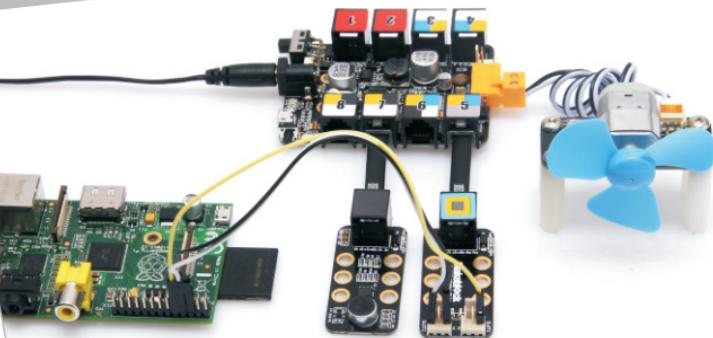
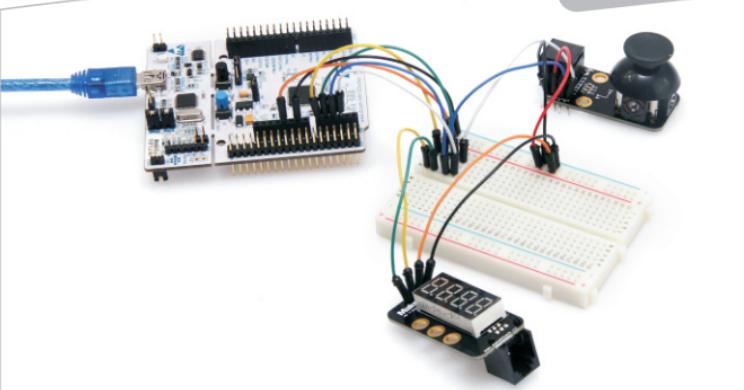
Install the Makeblock app to control your creations with your Android or iOS device.

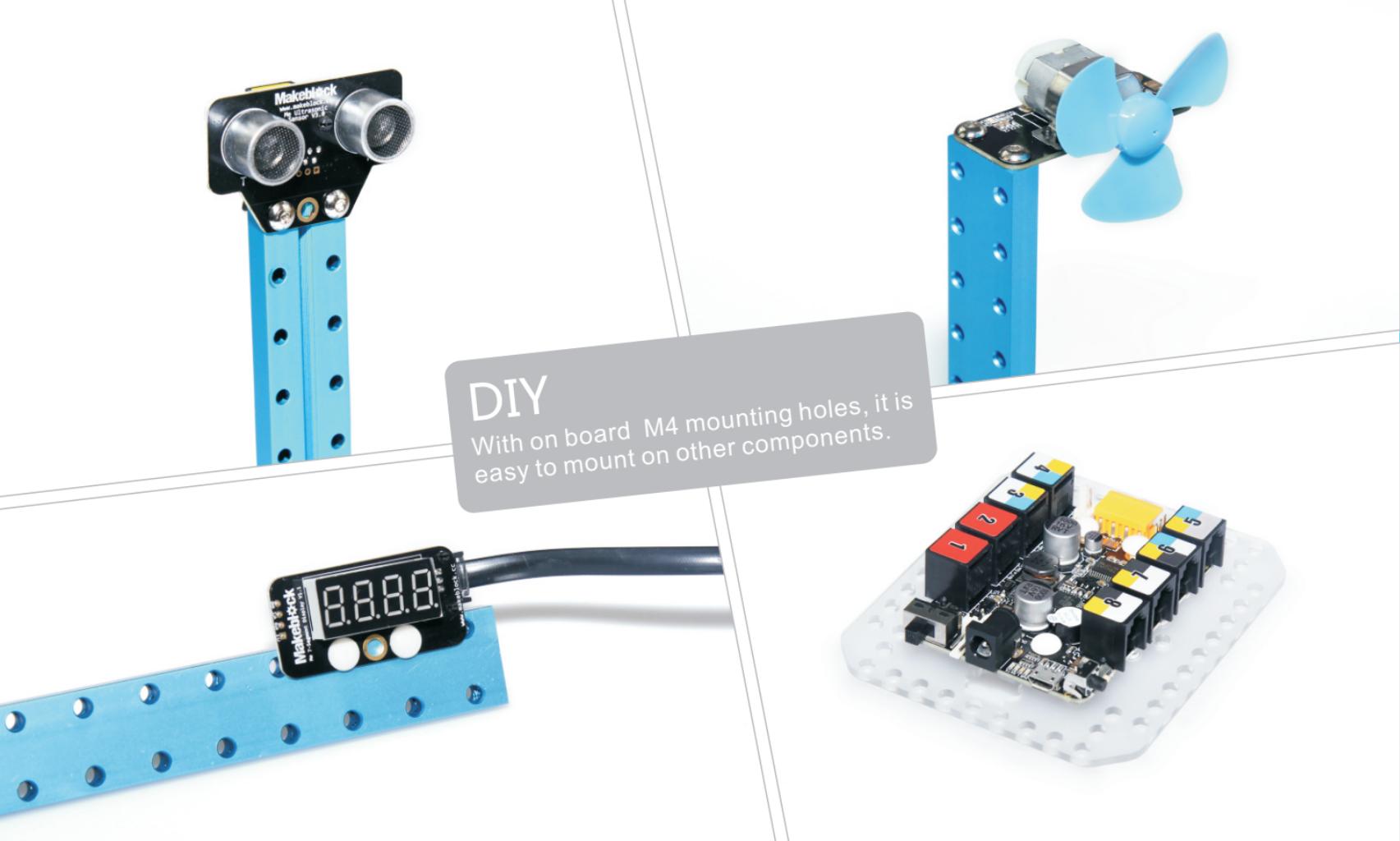


Visit the following URL for more details:
<http://learn.makeblock.cc/learning-APP>



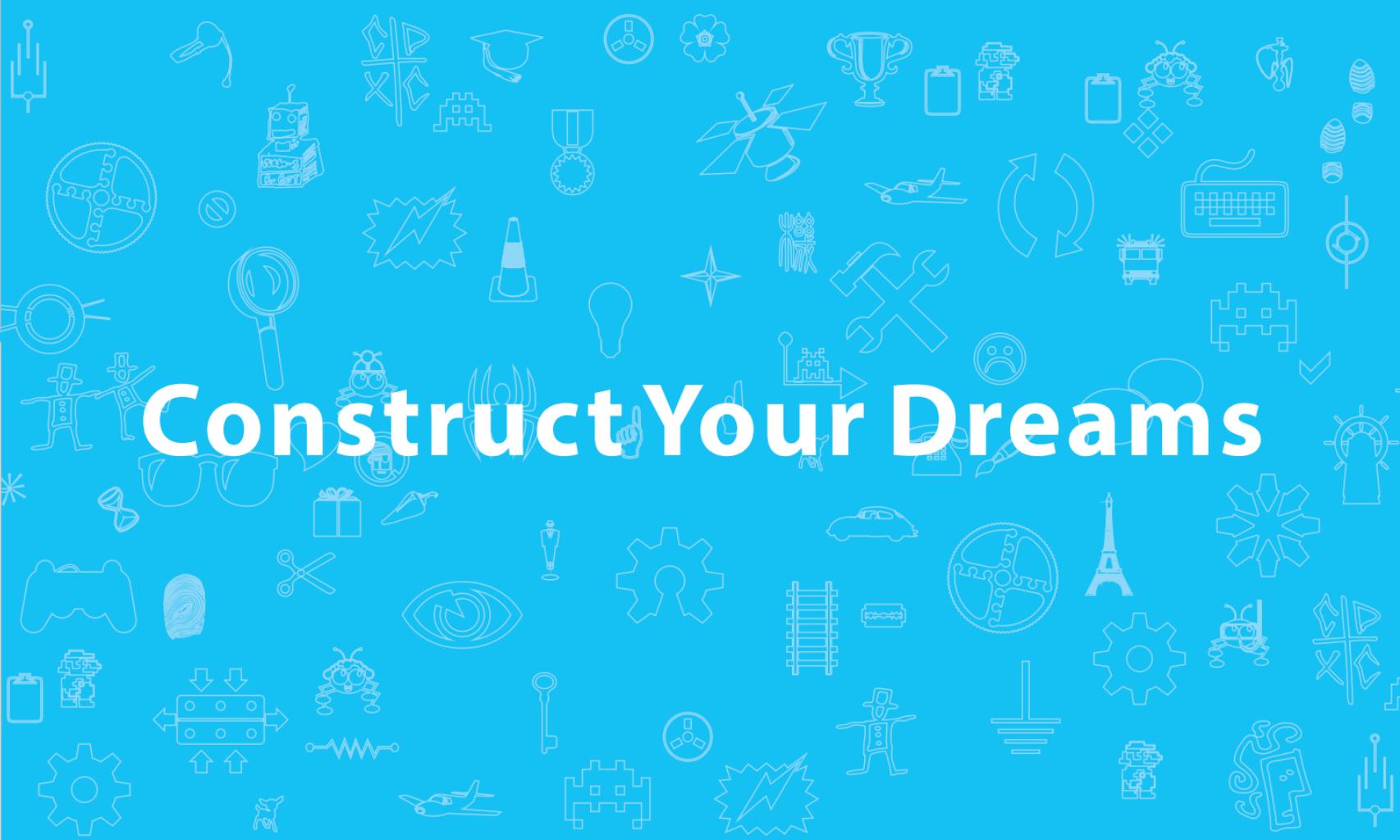
Installation
2.54mm breakout pins for connecting
with jumper wires





DIY

With on board M4 mounting holes, it is
easy to mount on other components.



Construct Your Dreams