# **ARDUBLOCK Graphical Programming Starter Kit**



Ardublock zero-based programming kit is a new graphical programming study kit which is based on Arduino enthusiasts and developed by keyestudio. It greatly lowers the difficulty in program learning for enthusiasts. All computer language and professional statements of a program can be simplified into blocks. Compiling is as easy as block building. Interactive with Arduino development board, they can achieve exciting works.

The open source feature of Arduino makes it extremely popular. You can find a many learning communities on the Internet. Ardublock is a graphical programming environment, just like blocks building, which greatly lowers the learning difficulty for starters. As long as you love it, everyone can play with Arduino without professional knowledge or relevant background. As for parents who stress on the education of next generation, this will be one of your best way to interact with your children.

1. BOM of ArduBloc	k Starter Kit	3
2. Application of Ard	luino	7
Introduction		7
Arduino Installa	ation and Program Upload	7
3. Introduction of Are	duBlock	16
1. Installation		16
2.Projects		16
Project 1	LED Blinking	16
Project 2	PWM Regulating Light Brightness	18
Project 3	Flowing Light	21
Project 4	Button-controlled LED	23
Project 5	Passive Buzzer Sound Production	26
Project 6	Active Buzzer Sound Production	29
Project 7	Reading-out Analog Value	31
Project 8	Light-controlled Sound	34
Project 9	Sound-controlled Light	36
Project 10	Servo Motor Control	39
Project 11	Ultrasonic Distance Measurement	42
Project 12	PS2 Joystick Module	45
Project 13	LM35 Temperature Sensor	49
Project 14	5V Relay Module	52
Project 15	Tilting Switch Module	54
Project 16	Capacitive Touch Module Control LED	57
Project 17	Flame Alarm Module	59
Project 18	Reed Switch Module	62
Project 19	PIR Motion Sensor Module	64
Project 20	Line Tracking Sensor	67
Project 21	MQ-2 Combustible Gas and Smoke Sensor	69
Project 22	Soil Module	72

# **1. BOM of ArduBlock Starter Kit**

No.	Product Name	Quantity	Picture
1	KEYESTUDIO UNO Control Board	1	
2	V5 Sensor Shield	1	RET GRO 1312 1110 9 8 7 6 5 4 3 2 1 PESET OUT OF THE PEREFECTION OF
3	Joystick Module	1	
4	5V Single Channel Relay	1	
5	Adjustable Potentiometer Module	1	The second se
6	Piranha LED	3	

			Contraction of the second seco
7	Active Buzzer	1	A CONTRACTOR OF THE PARTY OF TH
8	Passive Buzzer	1	Keynandis
9	LM35 Temperature Sensor	1	town may and the second
10	Tilting Module	1	a la antime a
11	Photoresistor	1	terrano.
12	Button Module	1	and a second sec
13	Capacitance Touch Module	1	

14	Sound Module	1	The second second
15	Flame Module	1	
16	Reed Switch Module	1	Ra annua
17	MQ-2 Combustible Gas and Smoke Module	1	
18	Soil Module	1	
19	Line Tracking Module	1	
20	Human Body IR Pyroelectricity Module	1	Ren and a state of the state of
21	Micro Servo Module	1	
22	Ultrasonic Module	1	

23	USB Cable	1	
24	40PIN Male to Male Dupont Line	1	

# 2. Application of Arduino



# Introduction

What's Arduino?

Arduino is an open-source hardware project platform. This platform includes a circuit board with simple I/O function and program development environment software. It can be used to develop interactive products. For example, it can read signals of multiple switches and sensors, and control light, servo motor and other various physical devices. It's widely applied in robot field.

# **Arduino Installation and Program Upload**

First, download Arduino development software, click below hyperlink: https://www.arduino.cc/en/Main/OldSoftwareReleases#1.5.x Downloaded file is an arduino-1.5.6-r2-windows.zip compressed folder, and unzip it to your hard drive. Double click Arduino-1.5.6 .exe. and select "I agree".



#### Click "Next".



#### And then "Install".

∞ Arduino Setup: Installation Folder	
Setup will install Arduino in the following folder. To insta folder, click Browse and select another folder. Click Inst installation.	l in a different all to start the
Destination Folder	Browse
Space required: 375.7MB Space available: 42.1GB	7
Cancel Nullsoft Install System v2,46 < Back	Install

Wait for the installation to be completed, and click close.

∞ Arduino Setup: Completed	
Show <u>d</u> etails	
	<u>_</u>
	, y
Cancel Nullsoft Install System v2,	46 < <u>B</u> ack <b>Close</b>

Below is how Arduino 1.5.6 looks like.



Next, let's install Arduino driver.

For different operating system, there may be slight difference in installation method. Below is an example in WIN 7.

When you connect Arduino Uno to your computer the first time, right click "Computer" —> "Properties"—> "Device manager", you can see "Unknown devices".

A Device Manager	- • ×
File Action View Help	
4 🚔 1306-PC	
> 🦉 Batteries	
p - 🚯 Bluetooth Radios	
> 🛃 Computer	
👂 👝 Disk drives	
b - La Display adapters	
b and b vD/CD-ROM drives	
▷ - U詞 Human Interface Devices	
DE ATA/ATAPI controllers	
b - a Imaging devices	
- Keyboards	
Description of the second s	
b - A Mice and other pointing devices	
b Monitors	
b - See Network adapters	
a - 🚯 Other devices	
L 🐚 Unknown device	
Processors	
b - 4 Sound, video and game controllers	
b 4 System devices	
▷ - ♥ Universal Serial Bus controllers	

Click "Unknown devices", select "Update Driver software".

PF serve monager	
File Action View Help	
I 1306-PC     Batteries     Butetooth Radios     Computer     Disk drives     Disk drives     Disk drives     Disk drives     DVD/CD-ROM drive     DVD/CD-ROM drives     DVD/CD-ROM drives     DV	

In this page, click "Browse my computer for driver software".



Find the "drivers" file.



Click "Next"; select "Install this driver software anyway" to begin the installation.



Installation is completed; click "Close".



After driver is installed, go to "Device manager" again. Right click "Computer" ->>

"Properties"—> "Device manager", you can see UNO device as below figure shown, also the Com port info.



Following is a sketch uploading example called "Hello World!".

First, open Arduino IDE. In this example sketch, we program Arduino to display "Hello World!" in serial monitor when it receives a specific character string "R"; also the on-board D13 LED will blink once each time it receives "R".

First, set up board; In "Tools", select "Arduino Uno".



Next, set up COM port; In "Tools", select "COM3".

Setch_jan04a   Ar	rduino 1.5.6-r2				X	1
sketch_ian04a	Auto Format Archive Sketch Fix Encoding & Reload Serial Monitor Board	Ctrl+T Ctrl+Shift+M	,		2	
3	Port			COM3		
void loop() { // put your me	Programmer Burn Bootloader		•			
¢					Þ	•
4			1	Arduine Une e	n COM4	

After selection, you can see indicated area is the same with settings in "Device manager".



Copy the example sketch and paste it to the IDE; click "Verify "to check compiling mistakes;

click "Upload **D**" to upload the program to the board.



After uploading is done, open "serial monitor **P**"; enter "R"; click "Send", the serial monitor will display "Hello World!" and the D13 LED will blink once.



Congratulations! Your first sketch uploading is successful!

#### Operating video and related material download address:

http://7326097.s21d-7.faiusrd.com/0/ABUIABAAGAAggfjavAUopJzsjAI?f=operating+ve dio+and+related+material.zip&v=1469496323

# **3. Introduction of ArduBlock**

Ardublock, developed by Shanghai Xinchejian, is a graphical programming environment software. ArduBlock is a third-party software of Arduino's official programming environment. It can only work with Arduino software. Different from Arduino's text-based programming environment, it's in graphical form. Programming is like building blocks, which makes programming more visualized and interactive. It also greatly lowers the difficulties in programming. People with no programming experience can also compile a program for Arduino controller.

## 1. Installation

Put "ardublock-beta-20140828"this file into the installing catalog of arduino IDE software, if no installing catalog, create one, for example, I put the file in D:\Program Files\Arduino\tools\ArduBlockTool\tool **Download Address of ardublock-beta-20140828:** 

# http://7326097.s21d-7.faiusrd.com/0/ABUIABAAGAAg5 6fbvAUoglzwugc?f=ardublock-beta-20140828.jar&v=1 469502439

# 2.Projects

After having a basic understanding of Arduino and ArduBlock, we are going to learn graphical programming by actual operation.

### Project 1 LED Blinking

This project is one of the basic experiments, which can achieve LED blinking by graphical programming.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Piranha LED\*1 Female to Female Dupont Line\*3

### **Connection Diagram**





After wiring, we can start programming. We will turn the LED on for one second, and off for one second. This program is simple and similar to one that comes with Arduino except it's connected to digital pin 10 rather than 13.



Click "download to Arduino" to see the codes in Arduino interface.

```
************
void setup()
{
 pinMode(10, OUTPUT);
}
void loop()
{
 digitalWrite( 10, HIGH );
 delay(1000);
 digitalWrite(10, LOW);
 delay(1000);
}
Then, you can see the high brightness LED blinking. We have now finished this project!
Note: Arduino software selection
On the TOOLS menu, select UNO.
COM port selects the same serial port with the first installation.
You cannot download program unless both requirements are met.
```

### Project 2 PWM Regulating Light Brightness

In this project, we should connect signal port of LED module to PWM port, and use graphical programming to control it to produce various PWM signals. So the brightness of LED changes, with effect of breathing LED.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Piranha LED\*1 Female to Female Dupont Line\*3

#### **Connection Diagram**



In this project, we will use PWM to control an LED, slowly lighting it up and then dimming it, and circulate it like this over and over again. Still, connect high bright LED module to pin 10 of digital IO port.



Click "download to Arduino" to see the codes in IDE interface.

```
int ABVAR_1 brightness = 0;
```

```
void setup()
{
  pinMode(10, OUTPUT);
}
void loop()
{
  while ( ( ( _ABVAR_1_brightness ) < ( 255 ) ) )
  {
    _ABVAR_1_brightness = ( _ABVAR_1_brightness + 1 );
    analogWrite(10, _ABVAR_1_brightness);
    delay(5);
  }
  while (((ABVAR_1_brightness) > (0)))
  {
    _ABVAR_1_brightness = ( _ABVAR_1_brightness - 1 );
    analogWrite(10, _ABVAR_1_brightness);
    delay(5);
```

LED is taking a breath. That's why we vividly call it breathing LED.

# **Project 3** Flowing Light

In this project, we use graphical programming to control 3 LED module . Light up and dim them according to given sequence and time, gaining a visual effect of flowing light.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Piranha LED\*3 Female to Female Dupont Line\*9





After wiring, we can start programming.



Click "download to Arduino" to see the codes in Aduino interface.

void setup()

{

pinMode( 3 , INPUT); pinMode( 2 , INPUT); pinMode( 4 , INPUT); pinMode( 2 , OUTPUT);

```
pinMode( 3 , OUTPUT);
pinMode( 4 , OUTPUT);
}
void loop()
{
    digitalWrite( 2 , !( digitalRead(2) ) );
    delay( 500 );
    digitalWrite( 3 , !( digitalRead(3) ) );
    delay( 500 );
    digitalWrite( 4 , !( digitalRead(4) ) );
    delay( 500 );
}
To realize flowing light function, we should connect three high bright LED module of IO port.
```

### Project 4 Button-controlled LED

Button is a common control electrical apparatus element, used to turn-on or turn-off circuit, as a switch of controlling motor or operating other device. In this project, we use graphical programming to control LED turning on or off.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Piranha LED\*1 Button Module\*1 Female to Female Dupont Line\*6





After wiring, we can start programming.



Click "download to Arduino" to see the following codes in IDE interface:

```
void setup()
{
  pinMode( 4 , INPUT);
 pinMode( 10 , OUTPUT);
}
void loop()
{
  if (( digitalRead(4) && HIGH ))
  {
    digitalWrite( 10 , HIGH );
    delay( 50 );
  }
  else
  {
    digitalWrite( 10 , LOW );
    delay( 50 );
  }
}
******
                                                    *****
                           ******
```

After finishing downloading, we can control LED by button module.

\*\*\*\*\*

## Project 5 Passive Buzzer Sound Production

Buzzers are divided into active and passive buzzers. Passive buzzers don't carry with vibrator inside, so it need external sine or square wave to drive. It can produces slight sound when connecting directly to power supply. It features controlling sound frequency and producing the sound of "do re mi fa so la si".

In this project, with passive buzzer, we use graphical programming to make this buzzer produce sound at 10~5000 frequency.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Passive buzzer\*1 Female to Female Dupont Line\*3





After wiring, we can start programming.



Click "download to Arduino" to see the following codes in IDE interface:

```
int _ABVAR_1_t1;
```

\*\*\*\*\*\*

```
void setup()
{
pinMode( 13 , OUTPUT);
_ABVAR_1_t1 = 0;
}
void loop()
{
analogWrite(_ABVAR_1_t1, 10);
digitalWrite( 13 , !(LOW ));
```

\*\*\*\*\*\*\*\*\*

```
if (( ( _ABVAR_1_t1 ) < ( 5000 ) ))
{
    digitalWrite( 13 , !(HIGH ));
    tone(8, _ABVAR_1_t1, 3000);
    _ABVAR_1_t1 = ( _ABVAR_1_t1 + 100 );
    delay( 2000 );
    digitalWrite( 13 , LOW );
    delay( 2000 );
    }
    digitalWrite( 13 , HIGH );
    delay( 1000 );
}</pre>
```

Ardublock graphical programming environment offers a buzzer module. There are several parameters, including one buzzer connecting port, and the other one sound frequency. In this project, t1 is a variable to change the frequency of sound producing, which begins from 10, and increases 100 each time, up to 5000. Figure pin 13 is used as prompt, so when making a sound , LED blinks. "Dang" module is a cycle structure on condition that the value of t1 is less than 5000. When more than 5000, the cycle structure ends. Please have a try, to find in which frequency the sound is sweeter from 10 to 5000 in this mystical sound program. After downloading the program, this passive buzzer project is finished.

#### **Project 6** Active Buzzer Sound Production

Buzzers are divided into active and passive buzzers. Active buzzer can produce sound normally when connecting directly to power supply, usually with stable frequency. In this project, with active buzzer, we use graphical programming to make buzzer ring for 1s and stop for 1s, circulating like this.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Active Buzzer\*1 Female to Female Dupont Line\*3





After wiring, we can start programming.



Click "download to Arduino" to see the following codes in IDE interface:

```
void setup()
{
    pinMode( 8 , OUTPUT);
}
void loop()
{
    digitalWrite( 8 , HIGH );
    delay( 1000 );
    digitalWrite( 8 , LOW );
    delay( 1000 );
}
Then buzzer rings for 1 second and stop for 1 second, circulating like this.
```

# Project 7 Reading-out Analog Value

On a KEYESTUDIO UNO control board, there are 6 analog interfaces numbered from 0 to 5. Potentiometer is a typical output component of analog value. We will use it in this project. We simply use graphical programming read out analog value of analog 0 on serial monitor.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Adjustable Potentiometer\*1 Female to Female Dupont Line\*3





After wiring, we can start programming.

do Loop	serial printla	values	glue	Analog Rotation Sensor VI	pin# 0
	delay MILLIS mill	liseconds 🧹	1000		

Click "download to Arduino" to see the following codes in IDE interface:

There are reading-out analog values as follow.

© COM6		
		Send
value:658		*
value:668		
value:677		
value:687		
value:694		
value:703		
value:711		
value:718		_
value:725		
value:732		
value:739		
value:745		
value:751		
value:757		
		Ψ
Autoscroll	No line ending 💌	9600 baud

# Project 8 Light-controlled Sound

Photovaristor is a resistor using the photoelectric effect of a semiconductor. Its resistance changes along with the intensity of the incident light, the greater the intensity, the lower the resistance, the less the intensity, the higher the resistance. It has wide application in various light-controlled circuits, such as lighting control, regulation, light-controlled switch etc. In this project, we use graphical programming to display analog value of illumination intensity measured by photovaristor. And set this value as frequency of sound production.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Photovaristor Module\*1 Passive Buzzer\*1 Female to Female Dupont Line\*6





After wiring, we can start programming.

In this project, we apply similar method in "reading analog value". Directly connect the photovaristor to analog port. The analog value varies with the intensity of light and controls the sound frequency produced by passive buzzer.

do	variable val
	value pin# 0 set integer variable Analog Ambient Light Sensor
loop	serial printlo Llights Lglue val
	Buzzer val
	delay MILLIS milliseconds 500

Click "download to Arduino" to see the following codes in IDE interface:

int \_ABVAR\_1\_val = 0 ;

void setup()

```
{
    Serial.begin(9600);
}
void loop()
{
    _ABVAR_1_val = analogRead(0);
    Serial.print("light:");
    Serial.print(_ABVAR_1_val);
    Serial.println();
    tone(8, _ABVAR_1_val);
    delay( 500 );
}
```

After downloading the program, have a kook at the value of serial monitor.

COM6	
	Send
light: 428	
light:427	
light:427	
light:427	
light:426	
light:426	
light:424	
light:418	
light:410	
light:409	=
light:411	1
light:424	
light:436	
light:441	
	-
V Autoscroll	No line ending 💌 9600 baud

The value indicates the light intensity, and the larger the value, the stronger the light intensity. Meanwhile, the value belongs to the sound frequency produced by passive buzzer.

## Project 9 Sound-controlled Light

Sound sensor is used to detect the intensity of ambient sound; the intensity is proportional to output voltage.

In this project, we use graphical programming to read out analog value of sound intensity measuring by sound module, ranging from 0 to 1023. And then map the value to 0-255, displaying on serial monitor. Finally set the mapped value as PwM value to control brightness of LED.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Piranha LED\*1
Sound Module\*1 Female to Female Dupont Line\*6







Click "download to Arduino" to see the codes in IDE interface:

int  $ABVAR_1$ \_sound\_value = 0;

void setup()

{

pinMode(11,OUTPUT);

```
Serial.begin(9600);
}
void loop()
{
    _ABVAR_1_sound_value = 0;
    _ABVAR_1_sound_value = (_ABVAR_1_sound_value + analogRead(0));
    analogWrite(11, map (_ABVAR_1_sound_value, 0, 1023, 0, 255));
    Serial.print("sound val:");
    Serial.print(_ABVAR_1_sound_value);
    Serial.print();
}
```

Alter dowinoading program, have a look at values on serial monitor	After downloading program,	have a	look at value	s on serial monitor
--	----------------------------	--------	---------------	---------------------

COM6			x
		Ser	nd
sound val:217			*
sound val:217			
sound val:217			
sound val:218			
sound val:217			
sound val:217			
sound val:218			
sound val:216			
sound val:217			
sound val:218			
sound val:216			
sound val:217			
sound val:218			
sound val:217			
sound val:218			-
Autoscroll	No line ending	▼ 9600 bau	ıd ·
*****	****	****	****

## Project 10 Servo Motor Control

Servomotor is a commonly used positional-servo actuator.

In this project, we use graphical programming . First , read out analog value controlled by adjustable potentiometer module, ranging from 0 to 1023. Then, map the value to 0-180. Finally, set the mapped value as angle of motor rotation.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Adjustable Potentiometer\*1 Micro Servo\*1 Female to Female Dupont Line\*3

## **Connection Diagram**

Note: do not use the USB port for power supply because the current it needs is more than 500ma, it may lead to USB port burnt out. We recommend external power supply.





```
After wiring, we can start programming.
```



Click "download to Arduino" to see the codes in Arduino interface.

#include <Servo.h>

int \_ABVAR\_1\_val = 0; int \_ABVAR\_2\_servo = 0; Servo servo\_pin\_9;

```
void setup()
{
    servo_pin_9.attach(9);
}
void loop()
{
    _ABVAR_1_val = analogRead(0);
    _ABVAR_2_servo = map (_ABVAR_1_val, 0, 1023, 0, 180) ;
    servo_pin_9.write(_ABVAR_2_servo);
}
Put the program into the development, and we can use potentiometer to control servomotor.
```

Note: place Servo folder into the compiler of \Arduino\libraries. Otherwise, you can't compile. Just like this D:\Program Files\Arduino\libraries

#### Servo folder downloading address:

http://7326097.s21d-7.faiusrd.com/0/ABUIABAAGAAggrHXvAUopNrTggY?f=Servo.zip&v=1469438 082

## **Project 11 Ultrasonic Distance Measurement**

In this project, we use graphical programming to measure the distance between ultrasonic module and front obstacle and display it on serial monitor.

### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Ultrasonic Module\*1 Female to Female Dupont Line\*4

### **Connection Diagram**

Let's start experiment, and connect ultrasonic waves to ARDUINO referring to the following figure.

Ultrasonic ranger	Arduino Uno
ECHO	D4
TRIG	D5
VCC	5V
GND	GND







Click "download to Arduino" to see the following codes in ARDUINO programming interface:

```
int ABVAR_1_t1 = 0;
```

int ardublock Ultrasonic Sensor Code AutoGenerated Return CM (int trigPin, int echoPin)

{

```
long duration;
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(20);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
duration = duration / 59;
if ((duration < 2) || (duration > 300)) return false;
return duration;
```

```
void setup()
{
    digitalWrite( 4 , LOW );
    Serial.begin(9600);
```

```
}
```

}

```
void loop()
```

```
{
    _ABVAR_1_t1 = ardublockUltrasonicSensorCodeAutoGeneratedReturnCM(4,5);
    Serial.print("diatance:");
    Serial.print(_ABVAR_1_t1);
    Serial.println();
    delay( 200 );
}
```

We can see the distance value measured by ultrasonic in the serial monitor of ARDUINO after downloading the program.

COM6	
	Send
diatance:176	
diatance:176	
diatance:175	
diatance:175	
diatance:176	
diatance:175	
diatance:175	
diatance:6	
diatance:5	
diatance:4	
diatance:3	
diatance:3	
diatance:4	
diatance:4	
diatance:3	
diatance:2	·
Autoscroll	No line ending - 9600 baud

## Project 12 PS2 Joystick Module

The joystick sensor module uses original high-quality metal PS2 joystick potentiometer, with (X, Y) 2-axis analog output and (Z) 1-channel button digital output. Together with Arduino sensor shield, you can make a remote control or other interactive projects. In this project, we use graphical programming to read out analog output of X, Y axis, and digital output of Z axis, displaying them on serial monitor.

### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Joystick Module\*1 Female to Female Dupont Line\*5





Click "download to Arduino" to see the following codes in ARDUINO programming interface:

```
int _ABVAR_1_X = 0 ;
int _ABVAR_2_Y = 0 ;
bool _ABVAR_3_Z= false ;
```

```
void setup()
{
    pinMode( 2 , INPUT);
    Serial.begin(9600);
}
void loop()
{
    _ABVAR_1_X = analogRead(0);
    _ABVAR_2_Y = analogRead(1);
    _ABVAR_3_Z = digitalRead(2);
    Serial.print("X:");
    Serial.print(_ABVAR_1_X);
```

```
Serial.println();
Serial.print("Y:");
Serial.print(_ABVAR_2_Y);
Serial.println();
Serial.print("Z:");
Serial.print(_ABVAR_3_Z);
Serial.println();
delay( 500 );
}
```

After downloading the program, we can see X and Y axis value and Z axis high and low level of joystick module in ARDUINO serial monitor.

COM6	
	Send
X:503	
¥:500	
Z:0	
X:503	
¥:500	
Z:0	
X:503	
¥:500	
Z:0	
X:503	
¥:500	_
Z:0	=
X:503	
¥:500	
Z:0	
	-
🔽 Autoscroll	No line ending 👻 9600 baud

Apply X and Y axis of joystick module, press Z axis and you can see obviously the change of numbers.



## Project 13 LM35 Temperature Sensor

LM35 is a common and easy-to-use temperature sensor.

In this project, we use graphical programming to read out analog value of temperature measured by LM35 temperature sensor module. Then after a serials of calculation we get specific temperature value. Finally display the value on serial monitor.

### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 LM35 Temperature Module\*1 Female to Female Dupont Line\*3







```
******
int \_ABVAR_1\_val = 0;
int \_ABVAR\_2\_dat = 0;
void setup()
{
 Serial.begin(9600);
}
void loop()
{
 _ABVAR_1_val = analogRead(0);
 _ABVAR_2_dat = ((5 * _ABVAR_1_val) / 10);
 Serial.print("TEMP=");
 Serial.print(_ABVAR_2_dat);
 Serial.println();
 delay( 100 );
}
**
```

After the program is downloaded, you can open the serial monitor to see current temperature.

O COM6			
		Send	
TEMP=29			
IEMP=29			
TEMP=29			
TEMP=29		r	
TEMP=29			=
TEMP=29			-
IEMP=27			
TEMP=29			
TEMP=29			
TEMP=29			
TEMP=28			
TEMP=29			
TEMP=28			Ŧ
Autoscroll	No line ending	▼ 9600 baud	
****	· · · · · · · · · · · · · · · · · · ·	****	**>

# Project 14 5V Relay Module

Relay is a automatic control device, the output of which can produce saltatory changes when the input(electricity, magnetism, sound, light and heat) arrives at a certain value. We usually need to use weak current to control strong current in life, namely, the low current controls large current. It looks like when Arduino controller control high power electric appliances like fan, we have to use relay.

In this project, we use graphical programming to control the on and off of 1-channel relay easily. The relay is on for 0.5S and off for 0.5S with an interval of 0.5S, circulating like this.

### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 5V 1-channel Relay Module\*1 Female to Female Dupont Line\*3





The relay works in high level, and we use digital port 3 in the Arduino controller. Output high level for 0.5 seconds, and then output low level for 0.5 seconds. That is to say, turn on switch for 0.5 seconds and turn off for 0.5 seconds.



Next, we can see the indicator light of the relay blinking when switch-on and off. Turn on switch for 0.5 seconds, and the led is on; turn off for 0.5 seconds and the led is off; meanwhile, the relay is ticking.

## Project 15 Tilting Switch Module

When one side of the switch tilts below horizontal position, the switch is on; When the other side of the switch tilts below horizontal position, the switch is off.

In the project, we use graphical programming to control the on and off of LED, tilting module used as a button.

### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Piranha LED\*1 Tilting Switch Module\*1 Female to Female Dupont Line\*6







Click "download to Arduino" to see the following codes in IDE interface:

```
void setup()
{
 pinMode(3, INPUT);
 pinMode( 6 , OUTPUT);
}
void loop()
{
 if (( digitalRead(3) && HIGH ))
  {
   digitalWrite( 6 , HIGH );
   delay( 100 );
  }
 else
  {
   digitalWrite( 6 , LOW );
   delay( 100 );
  }
}
*****************
After the program is downloaded, we can use tilting switch module to control LED.
```

\*\*\*\*\*

\*\*\*\*\*\*\*

# Project 16 Capacitive Touch Module Control LED

Capacitance touch module supplies one touch key to detect IC. In order to replace traditional key with changeable area key, touch detecting IC is designed.

In this project, we use graphical programming and control the on and off of LED by capacitive touch module.

### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Piranha LED\*1 Copacitive Touch Module\*1 Female to Female Dupont Line\*6







Click "download to Arduino" to see the following codes in IDE interface:

```
****
            *****
void setup()
{
 pinMode( 4 , INPUT);
 pinMode( 10 , OUTPUT);
}
void loop()
{
 if (( digitalRead(4) && HIGH ))
 {
   digitalWrite( 10 , LOW );
 }
 else
 {
   digitalWrite( 10 , HIGH );
 }
}
******
After the program is downloaded, we can use capacitive touching module to control LED.
                                          *****
                 *****
```

## Project 17 Flame Alarm Module

Infrared ray is very sensitive to flame, which is the feature of flame sensor. It utilizes specific infrared receive tube to detect flame, and then converts the brightness into changed level signal. In this project, we use graphical programming and flame sensor to control the ringing and stop of a buzzer.

### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Flame Sensor\*1 Active Buzzer\*1 Female to Female Dupont Line\*3







Click "download to Arduino" to see the following codes in IDE interface:

```
int _ABVAR_1_val = 0 ;
```

```
void setup()
{
  Serial.begin(9600);
  pinMode( 10 , OUTPUT);
}
void loop()
{
  _ABVAR_1_val = analogRead(0);
  Serial.print("val:");
  Serial.print(_ABVAR_1_val);
  Serial.println();
  if (( ( _ABVAR_1_val ) < ( 600 ) ))
  {
     digitalWrite( 10 , HIGH );
  }
  else
  {
     digitalWrite( 10 , LOW );
  }
}
```

After the program is downloaded, we can use flame sensor to check whether there is flame around.

## Project 18 Reed Switch Module

The reed switch grass tube is full of highly purified inactive gas. Before operating, the two reeds are separated. Extra magnetic field produces different polarity around the endpoint of the two reeds, so that two different polar reeds attract each other and close. In this project, we use graphical programming and regard reed switch module and magnet as a switch to control the on and off of LED.

### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Piranha LED\*1 Reed Switch Module\*1 Female to Female Dupont Line\*6





After wiring, we can start programming.



Click "download to Arduino" to see the following codes in IDE interface:

```
*****
void setup()
{
 pinMode(3, INPUT);
 pinMode( 6 , OUTPUT);
}
void loop()
{
 if (( digitalRead(3) && HIGH ))
 {
   digitalWrite( 6 , HIGH );
 }
 else
 {
   digitalWrite( 6, LOW );
 }
}
**************
After downloading is done, you can check whether there is magnetic field using reed switch
module.
```

## Project 19 PIR Motion Sensor Module

It is a human body detecting sensor based on infrared pyroelectricity, which can only detect moving body. That is to say, when you don't move, it can do nothing. Still ,it is a digital sensor. In this project, we use graphical programming and regard PIR Motion Sensor Module and human body as a button to control the on and off of LED.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Piranha LED\*1 PIR Motion Sensor Module\*1 Female to Female Dupont Line\*6

# **Connection Diagram**







Click "download to Arduino" to see the following codes in IDE interface:

```
void setup()
{
  pinMode( 3 , INPUT);
  Serial.begin(9600);
  pinMode( 6 , OUTPUT);
}
void loop()
{
  if (( digitalRead(3) && HIGH ))
  {
     Serial.print("Somebody is in this area!");
     Serial.println();
     digitalWrite( 6 , LOW );
     delay( 1000 );
  }
  else
  {
     Serial.print("No one!");
     Serial.println();
```

```
digitalWrite( 6 , HIGH );
  delay( 1000 );
}
```

After the downloading is done, use reed switch module to check whether there is someone moving. If someone moving, piranha LED on and the serial monitor indicates "Somebody is in this area!"; if nobody moving, the LED off and the serial monitor indicates "No one!".

COM6		
	Send	
No onel		-
No onel		
Somebody is in this area!		
Somebody is in this area!		
Somebody is in this area!		
No one!	•	
Autoscroll	No line ending - 9600 baud	-

## Project 20 Line Tracking Sensor

Line tracking sensor has a TCRT5000 photoelectric sensor. Infrared reflectivity of color is different ,which is applied to convert strong and weak echoed signal into current signal by photoelectric sensor. The potentiometer is used to adjust sensitivity of this sensor. In this project, we use graphical programming to detect high and low level of line tracking sensor signal port and the color of objects, displaying it on serial monitor.

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Line Tracking Sensor\*1 Female to Female Dupont Line\*3





do loop	serial printla	Line Tracking	glue	Line Tracking Sensor	3
	delay MILLIS mil	liseconds 1000	•		

Click "download to Arduino" to see the following codes in IDE interface:

```
void setup()
{
```

\*\*\*\*\*\*

```
pinMode( 3 , INPUT);
Serial.begin(9600);
}
void loop()
{
Serial.print("Line Tracking");
Serial.print(digitalRead(3));
Serial.println();
delay( 1000 );
}
```

After downloading program, we can use line tracking sensor to detect the color of line tracking object. If it is white, the light on the sensor turns on, and display "Line Tracking0" on serial monitor; if black, the light off, and display "Line Tracking1"

💿 COM6			
		Send	
Line IrackingU			
Line Tracking0			
Line Tracking1		=	
Line Tracking1			
		-	
Autoscroll	No line ending 🔻	9600 baud	
****	****	*******	<i>*</i> ***********************************

# Project 21 MQ-2 Combustible Gas and Smoke Sensor

Arduino MQ-2 combustible gas sensor is mainly used to detect whether the gas leaks in industry and family. It can detect liquefied petroleum gas, methane, ethane, iso-butane, ethyl alcohol, hydrogen, smoke and ect. Potentiometer can be used to adjust the sensitivity of the sensor. In this project, we use graphical programming to detect analog value of signal port of this sensor. When gases leak, the value increases and the LED is on .

#### **Materials Required**

KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 MQ-2 Combustible Gas and Smoke Sensor\*1 Female to Female Dupont Line\*3





do Loop	serial printla	sensorValue;	glue	Analog Cas Sensor	pin# ()
	delay MILLIS mill	iseconds 1000			

Click "download to Arduino" to see the following codes in IDE interface:

}

After downloading program, you can use this sensor to detect whether the gas leaks. Open serial monitor, you can see the analog value of ambient gas. In this experiment, we detect the gas of a lighter. Open a lighter and close to the sensor after closing it, the analog increasing, and the red light on the sensor is on; the lighter away from this sensor, the analog value decreasing, the red light off.

© COM6	
	Send
sensorValue;130	
sensorValue;130	
sensorValue;129	
sensorValue;128	
sensorValue;126	
sensorValue;127	
sensorValue;129	
sensorValue;312	
sensorValue;385	
sensorValue; 406	
sensorValue; 413	
sensorValue; 426	
sensorValue; 435	
sensorValue; 426	
sensorValue; 424	=
sensorValue;393	=
sensorValue;353	
sensorValue;322	
sensorValue;293	
sensorValue;271	
sensorValue;254	-
Autoscroll	No line ending 👻 9600 baud

# Project 22 Soil Module

This is an easy moisture sensor detecting the moisture of soil. If soil is lack of water, the output analog value decreases; if not, the value increases.

In this project, we use graphical programming to detect analog value of signal port of this sensor, displaying it on serial monitor.

### **Materials Required**
KEYESTUDIO UNO Control Board\*1 V5 Sensor Shield\*1 Soil Module\*1 Female to Female Dupont Line\*3

## **Connection Diagram**





After wiring, we can start programming.



Click "download to Arduino" to see the following codes in IDE interface:

```
void setup()
{
    Serial.begin(9600);
}
void loop()
{
    Serial.print("sensorValue:");
    Serial.print(analogRead(0));
    Serial.println();
    delay( 1000 );
}
```

After downloading program, we can use soil module to detect moisture in the soil. Open serial monitor, we can see corresponding value.

This is a printscreen under normal situation.

COM6	
	Send
sensorValue:U	
sensorValue:0	-
sensorValue:0	=
	-
Autoscroll	No line ending 🔻 9600 baud

This is a printscreen when putting soil module into water.

© COM6	
	Send
sensorValue:151	A
sensorValue:153	
sensorValue:149	
sensorValue:153	
sensorValue:155	
sensorValue:155	
sensorValue:155	
sensorValue:157	
sensorValue:152	
sensorValue:158	
sensorValue:156	
sensorValue:153	
sensorValue:159	
sensorValue:160	
sensorValue:159	
sensorValue:160	
sensorValue:164	
sensorValue:160	
sensorValue:166	=
sensorValue:166	
sensorValue:166	
sensorValue:165	· ·
Autoscroll	No line ending 🔹 9600 baud

\*\*\*\*\*\*\*\*\*\*\*\*\*

## Download address of all courses:

http://7326097.s21d-7.faiusrd.com/0/ABUIABAAGAAg27TXvAUogIqf-wE?f=program+download +address+of+all+courses.zip&v=1469438555

Referred Website: http://blog.ardublock.com/