

Arduino Starter Kit(Absolute Beginner)

Introduction

The Arduino Starter Kit provided by ElecFreaks is a great material to get users into learning step-by-step conveniently. For this kit, there is no need for soldering, plug then use, the construction of the working circuit can be done within one minute. It has 9 courses in total, content includes LED, infrared sensor, servo, and IR remote control.

The kit uses the Freaduino UNO, which is the improved version of the official UNO and 100% compatible with Arduino. It provides easy-to-use brick sensor interface, 3.3v or 5v I/O switch, power supply with DCDC circuit which support MAX 2A etc.

Getting Started with Arduino

Download IDE from : [Arduino Download](#)

Download Code and Libraries: [Arduino Starter Kit Demo Code](#)

Part1. Arduino Start blink

<syntaxhighlight lang="php">

```
/*
PART1 ARDUINO START Blink
Turns on LED for one second, then off for one second,
repeatedly.
Get the code from: Arduino
IDE->File->Example->Basics->Blink
Pin 13 has an LED connected on most Arduino boards.
*/
```

```
int led = 13;
```

```
// the setup routine runs once when you press reset:
```

```
void setup() {
```

```
// initialize the digital pin as an output.  
pinMode(led, OUTPUT);
```

```
}
```

```
// the loop routine runs over and over again forever: void loop() {
```

```
    digitalWrite(led, HIGH);    // turn the LED on (HIGH is  
the voltage level)  
    delay(1000);                // wait for a second  
    digitalWrite(led, LOW);    // turn the LED off by making  
the voltage LOW  
    delay(1000);                // wait for a second
```

```
} </syntaxhighlight>
```

Part2. Button control LED

```
<syntaxhighlight lang="php">
```

```
/*  
PART2 BUTTON CONTROL LED  
Press the button, led ON, press again led OFF  
*/
```

```
int led = 5; // The D5 pin,driving LED int button = A0; // The A0,read the button,Here used  
a analog pin as digital pin. void setup() {
```

```
    pinMode(led, OUTPUT);                // initialize the LED pin  
as an output.  
    pinMode(button, INPUT_PULLUP); // initialize the BUTTON  
pin as an input.
```

```
} void loop() {
```

```
    if (digitalRead(button) == LOW) {
```

```

    delay(200);          // wait for 200 microsecond,Avoid
    pressing the button and read many times in this very
    short time
    digitalWrite(led, HIGH); // turn the LED on (HIGH is the
    voltage level)
    while(1){
        if(digitalRead(button)==LOW){
            delay(200);
            digitalWrite(led, LOW); // turn the LED off (LOW is
            the voltage level)
            break;                //End of the while loop,Back to
            the main loop
        }
    }
}

```

</syntaxhighlight>

Part3. Vibration sensor control passive buzzer

<syntaxhighlight lang="php">

```

/*
PART3  Vibration sensors CONTROL Passive buzzer
Knock on the table, the buzzer will ring
*/

```

```

int vibration = A0;// The A0 pin,read Vibration sensors int buzzer = 6; // The D6 pin,driving
the Passive buzzer,the pin must PWM pin(3 5 6 9 10 11 on UNO)

```

```

void setup() {

```

```

    pinMode(vibration,INPUT_PULLUP);// initialize the
    vibration pin as an input.
    pinMode(buzzer,OUTPUT);          // initialize the buzzer
    pin as an output.

```

```

} void loop() {

```

```

    if(digitalRead(vibration)==HIGH){
        analogWrite(buzzer,200); //driver Passive buzzer must
        PWM,so analogWrite,200 is PWM value,max 1024
    }
}

```

```
    delay(1000);                //wait for 1000 microsecond
    analogWrite(buzzer,0);      //turn off the buzzer
  }
}
```

</syntaxhighlight>

Part4. PIR sensor control motor fan

<syntaxhighlight lang="php">

```
/*
PART4  PIR Sensor CONTROL Motor fan
If someone passing from the front, the fan will turn
*/
```

```
int pir = A0; // The A0 pin,read PIR int motor = 6; // The 6 pin,driving the motor

void setup() {
```

```
    pinMode(pir,INPUT); // initialize the PIR pin as an
    input.
    pinMode(motor,OUTPUT); // initialize the motor pin as an
    output.
```

```
} void loop() {
```

```
    if(digitalRead(pir)==HIGH) {
        digitalWrite(motor,HIGH);
        delay(5000); // wait for 5000 microsecond
        digitalWrite(motor,LOW); //turn off the motor
    }
}
```

</syntaxhighlight>

Part5. LDR sensor control motor fan

<syntaxhighlight lang="php">

```
/*
```

```
PART5 Photodiode sensor CONTROL Motor Fan
According to the intensity of light motor speed control
*/
```

```
int photodiode= A0; // The A0 pin,read Photodiode int motor = 6; // The 6 pin,driving the
motor
```

```
void setup() {
```

```
    pinMode(photodiode,INPUT); // initialize the photodiode
pin as an input.
    pinMode(motor,OUTPUT);    // initialize the motor pin as
an output.
```

```
}
```

```
void loop() {
```

```
    int speed=analogRead(photodiode)/2; //because the read
max value is 512
    analogWrite(motor,speed); //According to the intensity
of light motor speed control
}
```

</syntaxhighlight>

Part6. Soil moisture sensor control relay

<syntaxhighlight lang="php">

/*

```
PART6 Soil moisture Sensor CONTROL Relay
According to the intensity of light motor speed control
*/
```

```
int soil= A0; // The A0 pin,read Soil moisture int relay = 6; // The 6 pin,driving the Relay  
void setup() {
```

```
    pinMode(soil,INPUT); // initialize the soil pin as an  
    input.  
    pinMode(relay,OUTPUT); // initialize the relay pin as an  
    output.
```

```
} void loop() {
```

```
    int value=analogRead(soil);  
    if(value>200){ //set the default value ,you can set it  
then more or less to do something  
        digitalWrite(relay,HIGH); //turn on the relay  
    }  
    else digitalWrite(relay,LOW); //turn off the relay  
  
}
```

</syntaxhighlight>

Part7. Encoder sensor control servo

<syntaxhighlight lang="php">

```
/*  
PART7  Encode Sensor CONTROL Servos  
Turn the rotary encoder control servos  
*/
```

1. include <Servo.h>

```
int encodeB= A0; // The A0 pin,read encodeB int servos = 6; // The 6 pin,driving the  
servos Servo servo; //Get a servo controller int angle=90; //set the servo angle void  
setup() {
```

```
    pinMode(encodeB,INPUT); // initialize the encodeB pin as  
an input.  
    servo.attach(servos);  
    attachInterrupt(0,start,FALLING); //set encodeA  
interrupt,this board interrupt0 is pin 2
```

```
} void loop() { } void start(){
```

```
    if (digitalRead (encodeB) == HIGH) {  
        angle -= 30;  
    } else angle += 30;  
    if (angle >= 180) angle = 180;  
    else if (angle <= 0) angle = 0;  
    servo.write (angle); }
```

</syntaxhighlight>

Part8. Display Temperature and Humidity

<syntaxhighlight lang="php"> /* Part 8 USE DHT11 Temperature and humidity sensor and Segment

```
* display Temperature and humidity*/
```

1. include "DHT11.h" //load Temperature and humidity sensor library
2. include "TM1637.h" //load Segment display library
3. define CLK 4 //pins definitions clk for TM1637
4. define DIO 5 //pins definitions dio for TM1637

```
TM1637 tm1637(CLK,DIO); //get Segment display controller DHT11 dht11(A0); //DHT11  
A0 void setup(){ tm1637.init(); tm1637.set(BRIGHT_TYPICAL);} void loop(){ dht11.start();  
tm1637.display(3,12); //Temperature Unit
```

```
tm1637.display(2, (dht11.DHT11data) [2] %10);
```

```
tm1637.display(1,(dht11.DHT11data)[2]%100/10); delay(1000); tm1637.clearDisplay();  
tm1637.display(3,(dht11.DHT11data)[0]%10); // humidity  
tm1637.display(2,(dht11.DHT11data)[0]%100/10); delay(1000); }
```

</syntaxhighlight>

Part9. Display Number Of IRremote

Note: If you used IRremote.h on 1.6.5 ,which need change RECV_PIN = A0 . That's why we do not recommend.

<syntaxhighlight lang="php"> /* Part9 USE IRreceive and IR remote Displayed on the segment code */

1. include <IRremote.h>//load IRremote library
2. include "TM1637.h"//load Segment display library
3. define CLK 4//pins definitions clk for TM1637
4. define DIO 5//pins definitions dio for TM1637

TM1637 tm1637(CLK,DIO);//get Segment display controller IRecv ir(A0);//an instance of the IR receiver object,A0 is IRreceive pin; decode_results result; // container for received IR codes long codes[10]= // this array is used to store infrared codes
{ 0xFD708F,0xFD08F7,0xFD8877,0xFD48B7,0xFD28D7,0xFDA857, //0 1 2 3 4 5

```
0xFD6897,0xFD18E7, 0xFD9867,0xFD58A7}; // 6 7 8 9
```

```
void setup(){ tm1637.init(); tm1637.set(BRIGHT_TYPICAL); ir.enableIRIn();} void  
loop(){ if(ir.decode(&result)){
```

```
int i=-1;
```

```
while(!(i>9||result.value==codes[++i]));
```

```
ir.resume(); // resume receiver
```

```
if(i<10){
```

```
tm1637.clearDisplay();  
tm1637.display(3,i);//IRremote value  
}}}
```

</syntaxhighlight>

Download

Download IDE from : [Arduino Download](#)

Download Code and Libraries: [Arduino Starter Kit Demo Code](#)

How to buy

<http://www.electfreaks.com/estore/arduino-starter-kit-absolute-beginner.html>

Licensing

This documentation is licensed under the Creative Commons [Attribution-ShareAlike License 3.0](#) Source code and libraries are

licensed under [GPL/LGPL](#), see source code files for details.