Arduino Advanced Kit

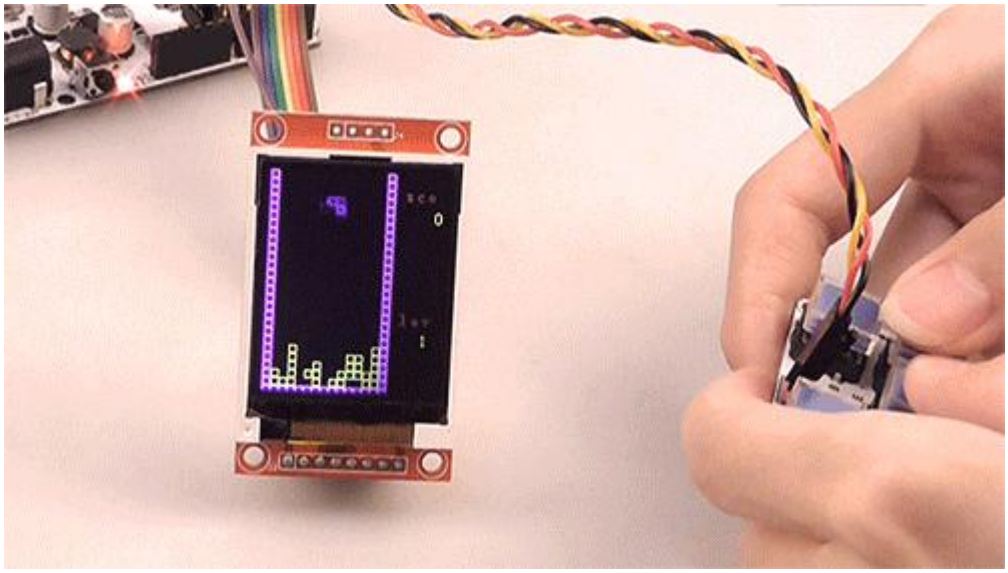
Introduction:

Arduino advanced kit provided by ELECFREAKS is based on our starter kit to expand more and more popular modules owing the characteristics of plug and use without any requirement of soldering. We have well prepared six courses for learners to learn the usage of modules in an intersting environment which is better for stimulating your thinking mind. For example,Tetris games help us to study the usage of the LCD screen and Arduino program. Color picker used can help us to learn the color sensor as well as BLE Bluetooth communication and Android interface development ... Practical Course included LCD screen, sensor, MP3, BLE, WIFI, Android development, and image interaction (processing), make you open-minded!

Part1: Arduino Play Tetris Game
Part2: Alcohol Tester New
Part3: Make an Mp3 Player New
Part4: Color Picking Plate New
Part5: Learning Processing with Arduino
Part6: Smarthome Wifi Remote Command

Part1 Arduino Play Tetris Game

Product introduction:



Use Arduino to develop Russia Brick Game and learn how to develop a analog game to present the classical taste.

Materials:

An Arduino maminboard,an analog keyboard, an LCD display

*1xFreaduino Uno;
*1xADkey;
*1xTFT01-1.8;

Parameter and IO definition:

1.5V power supply to mainboard

2.TFT 的 VCC-GND-CS-RESET-AO-SDA-SCK-LED successively connected with power supply(V)、ground(G)、D7、D6、D5、D4、D3、D2 of the mainboard.

3.ADK access to board analog A0, G (ground), V (power supply), S (IO port).

Working principles:

A.First of all, you need to set the size of the smallest grid. To set the pixels of the side can change the size of grids and calculate how many grids can be layed to the whole screen.

B.TO get a variety of graphics by changing different placement of grids. We have set 8 graphics in the game and we need to encapsulate it with array, because the biggest is strip which occupied four grids. There are four directions in each graphic and 4x4 two-dimensional array can be used to encapsulate the graphics data. That the array value is zero or not is used to make sure whether there is a grid existed.

C.When you put the whole screen as a large array and every grie is one of the elements, then you will be able to control the position of each graphics displayed on the screen and can calculate if it is in the end, beyond the border, or all of the elements has been filled in the line, then eliminate them.

D.If you are already clear of all this, just add a button then. Reading its value can be used to control the up and down about the game graphics. Well, introduction is over, let's start programming it.

Arduino code

part1.zip

<syntaxhighlight lang="php">

```
/*
 PART1 Arduino Play Tetris Game
 The Tetris game uno fitted to the inside, and then connect a LCD
and a ADK can be played
*/
```

1. include <UTFT.h>
2. include <Timer.h>
3. define blocksize 6
4. define xmax 16
5. define ymax 28
6. define LCD_BL 2

UTFT mlcd(ITDB18SP,4,3,7,6,5);//get lcd controler(model,data,clk,cs,rst,control) Timer timer; extern uint8_t SmallFont[]; // blockType // turnState byte level=1; byte levelid; byte blockType; byte score = 0; byte turnState; byte x=0; byte y=0; byte i = 0; byte j = 0; byte flag = 0;

```
//x=0-15,y=0-27;
byte map1[xmax][ymax];
```

// S、Z、L、J、I、O、T byte shapes[8][4][16]={ // i { {1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, {1, 0, 0,0,1, 0, 0, 0,1, 0, 0, 0, 1,0, 0, 0 }, { 1, 1, 1, 1,0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1,0, 0, 0, 1,0, 0, 0,1, 0, 0, 0, 1,0, 0, 0 } }, // s { { 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 }, { 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 } }, // z { { 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 }, { 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 } }, // j { { 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0 }, { 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 }, { 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 } }, // o { { 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 } }, // l { { 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0 }, { 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 }, { 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 } }, // _| { { 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 }, { 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0 } }, // t { { 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 },

{ 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 }, { 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }, { 1,0, 0, 0, 1, 1,0, 0,1, 0, 0, 0, 0, 0, 0, 0 } } };

// legitimacy detect byte blow(byte x, byte y, byte blockType, byte turnState) { for (byte a = 0; a < 4; a++) { for (byte b = 0; b < 4; b++) { if (((shapes[blockType][turnState][a * 4 + b] == 1) && (map1[x + b+1][y + a] == 1)) || ((shapes[blockType][turnState][a * 4 + b] == 1) && (map1[x + b+1][y + a] == 2))) {

```
   return 0;
```

} } } return 1; } void drawwall() { for (i = 0; i < (xmax-1); i++) { map1[i][(ymax-2)] = 2; } for (j = 0; j < (ymax-1); j++) { map1[(xmax-2)][j] = 2; map1[0][j] = 2; } }

void newmap() { for (i = 0; i < (xmax-1); i++) { for (j = 0; j < (ymax-1); j++) { map1[i][j] = 0; } } } // judgment over byte gameover(byte x, byte y) { if (blow(x, y, blockType, turnState) == 0) { return 1; } return 0; }

void newblock() { blockType =(byte) ((random()) % 8); turnState =(byte) ((random()) % 4); x=4; y=0; if(gameover(x, y) == 1) { mlcd.setColor(255,0,0); mlcd.print("GAME OVER",10,70); while(!pause()); newmap(); drawwall(); score = 0; mlcd.setColor(0,0,0); mlcd.print("GAME OVER",10,70); paint(); } } void paint() { //Videos have been fixed block for (j = 0; j < (ymax-1); j++) {

```
for (i = 0; i < (xmax-1); i++){
```

if (map1[i][j]==2) { mlcd.setColor(255,0,255); mlcd.drawRect(i * blocksize+1, j * blocksize+1,i * blocksize+blocksize, j * blocksize+blocksize); } else if(map1[i][j]==1) { mlcd.setColor(255,255,0); mlcd.drawRect(i * blocksize+1, j * blocksize+1,i * blocksize+blocksize, j * blocksize+blocksize); } else{ mlcd.setColor(0,0,0); mlcd.drawRect(i * blocksize+1, j * blocksize+1,i * blocksize+blocksize, j * blocksize+blocksize); } } } mlcd.setFont(SmallFont); mlcd.setColor(255,0,0); mlcd.print("sco",98,10); //Display Score mlcd.setColor(255,255,0); mlcd.printNumI(score,96,26,4); mlcd.setColor(255,0,0); mlcd.print("lev",98,100);//Display level mlcd.setColor(255,255,0); mlcd.printNumI(level,105,115,2); }

void turn() { byte tempturnState = turnState; turnState = (turnState + 1) % 4; if (blow(x, y, blockType, turnState) == 1) { turnState = tempturnState; blockundraw(0); turnState = (turnState + 1) % 4; blockdraw(); } if (blow(x, y, blockType, turnState) == 0) { turnState = tempturnState; }

} void left() { if(x>0) if (blow(x - 1, y, blockType, turnState) == 1) { blockundraw(0); x = x - 1; blockdraw(); }

} void right() {

if (blow(x + 1, y, blockType, turnState) == 1) { blockundraw(0); x = x + 1; blockdraw(); }} void delline() { byte c = 0; for (byte b = 0; b < (ymax-1); b++) { for (byte a = 0; a < (xmax-1); a++) { if (map1[a][b] == 1) { c = c + 1; if (c ==(xmax-3)) { score +=(xmax-3); if(score>=1000)nextlevel(); for (byte d = b; d > 0; d--) { for (byte e = 0; e < (xmax-2); e++) { map1[e][d] = map1[e][d - 1];

```
                                    }
                                }
```

paint();

```
                }
                  }
                                    }
```

c = 0;

```
                                    }
```

} // add to map void add(byte x, byte y, byte blockType, byte turnState) { int j = 0; for (byte a = 0; a < 4; a++) { for (byte b = 0; b < 4; b++) { if (map1[x + b + 1][y + a] == 0) { map1[x + b + 1][y + a] = shapes[blockType][turnState][j]; }

j++; } } }

```
void down() {
if (blow(x, y + 1, blockType, turnState) == 1) {
```

blockundraw(0);

```
y = y + 1;
    blockdraw();
}
;
if (blow(x, y + 1, blockType, turnState) == 0) {
add(x, y, blockType, turnState);
blockundraw(1);
delline();
newblock();
}
;
}
```

```

```

void blockdraw() { for (j = 0; j < 16; j++) { if (shapes[blockType][turnState][j] == 1) { mlcd.setColor(0,0,255); mlcd.drawRect((j % 4 + x + 1) * blocksize+1, (j / 4 + y) * blocksize+1,(j % 4 + x + 1) * blocksize+blocksize, (j / 4 + y) * blocksize+blocksize); } } } void blockundraw(byte a) { if(a)mlcd.setColor(255,255,0); else mlcd.setColor(0,0,0); for (j = 0; j < 16; j++) { if (shapes[blockType][turnState][j] == 1) {

```
  mlcd.drawRect((j % 4 + x + 1) * blocksize+1,
(j / 4 + y) * blocksize+1,(j % 4 + x + 1) * blocksize+blocksize,
(j / 4 + y) * blocksize+blocksize);
```

} } }

1. define buttonA 0
2. define buttonB 47
3. define buttonC 91
4. define buttonD 132
5. define buttonE 546

// ADK listener void keyListener() { pause(); int value=analogRead(A0); if(value<150&&value>100){ delay(140); right(); } else if(value>500&&value<600){ down(); } if(value<20){ delay(150);

```
  turn ();
```

} else if(value>70&&value<100){ delay(140); left(); }

} int pause() { int value=analogRead(A0); if(value>30&&value<60){ delay(550); value=0;

```
 while(!(value>30&&value<60)){
```

value=analogRead(A0); } delay(150); return 1; }else return 0;

} void nextlevel() { level++; timer.stop(levelid); newmap(); drawwall(); score=0; paint(); newblock(); levelid=timer.every((500/level),timelisten); }

```
// AUTO GAME
```

void timelisten() { blockundraw(0); if(blow(x, y + 1, blockType, turnState) == 1) {

```
y = y + 1;
delline();
```

```
}
if (blow(x, y + 1, blockType, turnState) == 0) {
if (flag == 1) {
add(x, y, blockType, turnState);
blockundraw(1);
delline();
newblock();
flag = 0;
}
flag = 1;
};
blockdraw();
}
```

void setup(){ pinMode(A0,0); //ADK pin A0

```
      pinMode(LCD_BL,OUTPUT);
      digitalWrite(LCD_BL,HIGH);
```

srand(1024); mlcd.InitLCD(0); mlcd.setFont(SmallFont); mlcd.clrScr(); //refresh the lcd newblock(); newmap(); drawwall(); timer.every(500,timelisten); //AUTO GAME timer.every(10,keyListener); //ADK listen

```
   paint();
```

}

void loop(){

timer.update(); }

</syntaxhighlight>

Alcohol Tester New

Introduction:
Arduino advanced kit provided by ELECFREAKS is based on our starter kit to expand more and more popular modules owing the characteristics of plug and use without any requirement of soldering. We have well prepared six courses for learners to learn the usage of modules in an intersting environment which is better for stimulating your thinking mind. For example,Tetris games help us to study the usage of the LCD screen and Arduino program. Color picker used can help us to learn the color sensor as well as BLE Bluetooth communication and Android interface development ... Practical Course included LCD

screen, sensor, MP3, BLE, WIFI, Android development, and image interaction (processing), make you open-minded!

Part2 Alcohol tester

Product introduction:

Using Arduino to make an alcohol tester to remind your safe travel.

Materials:
Easy to be made, because we provided all the hardware and software support
Only require:

- *1xFreaduino Uno;
- *1xMQ3 Sensor;
- *1xTFT01-1.8;
- *1xActive buzzer;

```
Parameter and IO definition:

1.5V power supply to mainboard

2.TFT 的 VCC-GND-CS-RESET-AO-SDA-SCK-LED successively connected
with power supply(V)、ground(G)、D7、D6、D5、D4、D3、D2 of the
mainboard.

3.MQ3 access to board analog A0 end, G (ground), V (power
supply), S (IO port).

4.Active buzzer access to analog A1 end, G (ground), V (power
supply), S (IO port).
```

```
What I need to reminde you is that if you did not see the
program, but prepared to use programming process directly, you
need to initialize MQ3 after writing the program at least ,
because the sensor would require heating for 5M's normal
execution.

After booting you can adjust potentiometer knob with a
screwdriver. Until number 20 is displayed on the screen, it can
be properly tested without correcting for many times.
```

```
Working principles:

A.Tin oxide (SNO2) in MQ3 is more sensitive to alcohol, when it meets a variety of alcohol, it will quickly change the resistance value. When through series voltage divider circuit, it will be able to get different voltage values

B.Since the unit used in alcohol tester is not the same as the official, do not rely on it too much, only as a reference.
```

MQ3 reference data:

Arduino code

part2.zip

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

/*

```
PART2 Alcohol Tester
1. Use the UNO for programming;
2. Connect the MQ3 and the LCD, which can test and display the degree of alcohol;
3. Connect an active buzzer to sound a warning if the degree of alcohol surpasses;
4. Note that you must use a screwdriver to adjust the MQ3 potentiometer for initial value calibration;
5. If it displays 20, then it is ok and does not require repeated calibration.
*/
```

1. include <UTFT.h>
2. include <Timer.h>
3. include <avr/pgmspace.h>

// Declare which fonts we will be using extern uint8_t SmallFont[]; extern uint8_t BigFont[]; extern uint8_t SevenSegNumFont[];

Timer timer; //get a timer

1. define LCD_BL 2

UTFT mlcd(ITDB18SP,4,3,7,6,5);//get lcd controler(model,data,clk,cs,rst,control) char distances[5]={'0','0','0','0','\0'}; char* distancetostring(int distance){ distances[0]=distance%10000/1000+'0'; distances[1]=distance%1000/100+'0'; distances[2]=distance%100/10+'0'; distances[3]=distance%10/1+'0'; return distances; } char time[9]={'0','0',':','0','0',':','0','0','\0'}; void times(){ if(time[7]=='9'){ time[7]='0'; if(time[6]=='5'){ time[6]='0'; if(time[4]=='9'){ time[4]='0'; if(time[3]=='5'){ time[3]='0'; if(time[0]!='2'){ if(time[1]=='9'){ time[1]='0'; time[0]++; } else time[1]++; } else { if(time[1]=='3'){ time[1]='0';time[0]='0'; }else time[1]++; }

```
          }
```

else time[3]++; } else time[4]++; } else time[6]++; } else time[7]++; }

void start(){

```
myGLCD.setFont(SevenSegNumFont);
int len=analogRead(A0)-150;
if(len<0)len=0;
if(len<100){
myGLCD.setColor(0,255,0);}
else {
myGLCD.setColor(255,0,0);
}
myGLCD.print(distancetostring(len),15,55);
if(len>150){
```

analogWrite(A1,200); delay(500); analogWrite(A1,0); }

} void printtime(){ times(); myGLCD.setFont(SmallFont); myGLCD.setColor(255,255,0);

```
   myGLCD.print(time,5,113);
```

}

void setup() {

```
  pinMode(A0,0);  //MQ3 pin A0
  pinMode(A1,1);  //Active buzzer
  pinMode(LCD_BL,OUTPUT);
  digitalWrite(LCD_BL,HIGH);
 // Setup the LCD
 myGLCD.InitLCD();
 myGLCD.clrScr();
```

```
myGLCD.setFont(BigFont);
myGLCD.setColor(0,0,255);//R G B
myGLCD.print("init...",24,56);
while(millis()<20000); //Heating MQ3 20S
  analogWrite(A1,200); //Detection alarm
  myGLCD.clrScr();
  myGLCD.print("Alcohol%",16,15);
  myGLCD.setFont(SmallFont);
```

myGLCD.setColor(255,255,0);

```
  myGLCD.print(time,5,113);
  analogWrite(A1,0);
timer.every(10,start);
timer.every(1000,printtime);
Serial.begin(9600);
}
```

void loop() { timer.update(); }

```
</syntaxhighlight>
```

Make an Mp3 Player New

Introduction:
Arduino advanced kit provided by ELECFREAKS is based on our starter kit to expand more and more popular modules owing the characteristics of plug and use without any requirement of soldering. We have well prepared six courses for learners to learn the usage of modules in an intersting environment which is better for stimulating your thinking mind. For example,Tetris games help us to study the usage of the LCD screen and Arduino program. Color picker used can help us to learn the color sensor as well as BLE Bluetooth communication and Android interface development ... Practical Course included LCD screen, sensor, MP3, BLE, WIFI, Android development, and image interaction (processing), make you open-minded!

Part3 Make an Mp3 Player

Product introduction:

Use Arduino to make a Mp3 player by yourself would be interesting.

Materials:
an Arduino board, an ADkey, an Mp3 module, of course, you still need to prepare an additional headset(general 3.5MM) and an SD card(FAT32 format, 32G or less)

*1xFreaduino Uno;
*1xADkey;
*1xMP3 module;

```
Parameter and IO definition:

1.3.3V power supply to mainboard
2.ADK access to board analog A0, G (ground), V (power supply), S
(IO port).

3.Mp3Module 的 GND access to board black G port, VCC access to
board red V port, TX access to board D4 port, RX access to board
D5 port.
```

```
Working principles:

Via ADkey transferring to Uno board to control playing of Mp3
module, changing songs, and adjusting the volume. You need to
know serial communication and connect the RX, TX of MP3 module to
the TX, RX of Uno respectively. The rate is set to      9600bps.
The file of JQ6500 in part3.zip has some communication
introduction about MP3 module. What you must remember in mind is
that the command must be hexadecimal data for normal
communication. To insert SD card can immediately identify the
inside MP3 musics. Send play command: 7E 02 0D EF to play it, but
it must be sent in hexadecimal values, not strings.

Hardware Parameters

== Communications Directive ==
```

1. Send commands directly no need return parameters

For example, for next, send: 7E 02 01 EF

For previous, send: 7E 02 02 EF

For play, send: 7E 02 0D EF

2. Parameter Query

- Example: To read the volume, sent [7E 02 43 EF], it'll directly return to volume (16 bits)
- ADKEY resistance function: 0R PLAY 3.3K for Next, V + 6.2K for Pre, V-9.1K for mode switching.
- Note 1: The folders inside USB and TF card must be named 01 02...99; the files inside the folders must be named 001 002 003....

Arduino code

part3.zip

```
<syntaxhighlight lang="php">
/*
PART3  Make a Mp3 Player By Uno
1. You need a Uno
2. You need a Mp3 Module, using its Rx Tx pin to Uno's Tx Rx pin;
3. You also need to add an ADK on Uno, control the player of the
play & pause
*/
```

1. include <SoftwareSerial.h>
2. define mp3Rx 4
3. define mp3Tx 5

SoftwareSerial mp3(mp3Rx,mp3Tx);

1. define buttonA 0
2. define buttonB 47
3. define buttonC 91
4. define buttonD 132
5. define buttonE 546

int play=1; // ADK listener void keyListener() { int value=analogRead(A0); if(value<150&&value>100){ delay(300); //play/pause//7E 02 0D EF//7E 02 0E EF if(play){ mp3.write(0x7e); mp3.write(0x02); mp3.write(0x0d); mp3.write(0xef); play=0; } else{ mp3.write(0x7e); mp3.write(0x02); mp3.write(0x0e); mp3.write(0xef); play=1; } } else if(value>500&&value<600){ delay(300); //next//7E 02 01 EF mp3.write(0x7e); mp3.write(0x02); mp3.write(0x01); mp3.write(0xef); } if(value<20){ delay(300);

```
  // value-//7E 02 05 EF
```

mp3.write(0x7e); mp3.write(0x02); mp3.write(0x05); mp3.write(0xef); } else if(value>70&&value<100){ delay(300); //pre//7E 02 02 EF mp3.write(0x7e); mp3.write(0x02); mp3.write(0x02); mp3.write(0xef);

}else if(value>30&&value<60){ delay(300); //value+//7E 02 04 EF mp3.write(0x7e); mp3.write(0x02); mp3.write(0x04); mp3.write(0xef); }

}

//The setup function is called once at startup of the sketch void setup() { // Add your initialization code here pinMode(A0,0); //A0 pin adk //mp3 module use RX(D0 pin)Tx(D1 pin) mp3.begin(9600);//open MP3 serial Serial.begin(9600);//open serial }

// The loop function is called in an endless loop void loop() { //Add your repeated code here keyListener();// listen adk if(Serial.available()){ mp3.print(Serial.readString()); } if(mp3.available()){ Serial.print(mp3.readString()); } }

```
</syntaxhighlight>
```

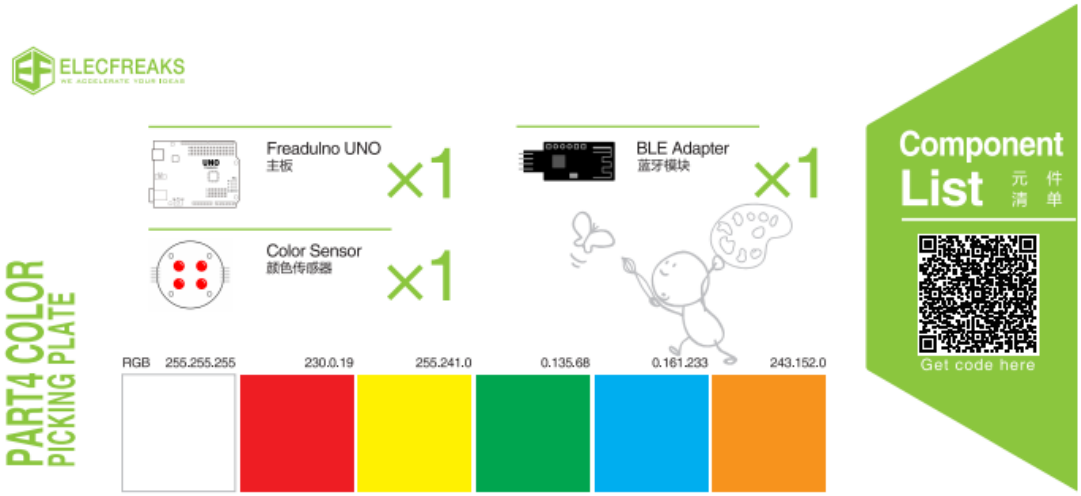
Color Picking Plate New

Introduction:
Arduino advanced kit provided by ELECFREAKS is based on our starter kit to expand more and more popular modules owing the characteristics of plug and use without any requirement of soldering. We have well prepared six courses for learners to learn the usage of modules in an intersting environment which is better for stimulating your thinking mind. For example,Tetris games help us to study the usage of the LCD screen and Arduino program. Color picker used can help us to learn the color sensor as well as BLE Bluetooth communication and Android interface development ... Practical Course included LCD screen, sensor, MP3, BLE, WIFI, Android development, and image interaction (processing), make you open-minded!

Part4 Color Picking Plate

Product introduction:

This is an interesting project, you can paint the color you see to the phone via Bluetooth.



Materials:

an Uno board, a color sensor, a mobile phone with Android 4.3 or above prepared by yourself, we now only developed APK for android phone

*1xFreaduino Uno;

*1xColor Sensor;

*1xBle;

```
Parameter and IO definition:

1.3.3V power supply to mainboard

2.Color Sensor access to board.
 #define S0     D6   // Please notice the Pin's define,D(digital
pin)
 #define S1     D5
 #define S2     D4
 #define S3     D3
 #define OUT    D2
 #define VCC    V(power supply)
 #define GND    G(ground)
 #define OE     G(ground)
```

```
Notes

A.Color Sensor power supply can not be higher than 4V, so please
adjust arduino board power supply to 3.3V
B.Color sensor needs white balance correction for normal use.
```

C.Uno board can not program procedures when connect to a Bluetooth module, because serial port is occupied. When programming process, you need to disconnect the Bluetooth module

Working principles:

1.After connection and at the beginning of powered Uno board, it will do white balance correction to color sensor. That is it will identify the first color it detected as white, ie RGB values are 255,255,255. So when you perform white balance correction, just try to capture an value very access to white, and distance is better to be within 10 cm. If you are not satisfied with the corrected color, you can also control it to do white balance correction again by phone

2.Apk introduction(need apk and Source code,please downloadpart4.zip):

A.Only support Android 4.3 or above, being restricted by Bluetooth 4.0 communications protocol, otherwise it can not open the program

B.Its interface is very simple, just with simple buttons, so you can operate easily.

C.If you are not satisfied with the color you got, try to test in a good lighting environmentor click "Balance" to do white balance correction again. If you want to improve it, we offer you Apk source code, so that you can develop it once again.

Color table:

Arduino code

part4.zip

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

```
 PART4 Communicate with Android phone via Bluetooth 4.0
and color sensors
 1. You need a Uno, a color sensor, a BLE and one
support, Bluetooth 4.0 android phone
 2. When you put these are connected, you also need to
install a specific apk to communicate with it
*/
```

1. include <TimerOne.h>

void TSC_WB(int Level0, int Level1);

1. define S0 6 // Please notice the Pin's define
2. define S1 5
3. define S2 4
4. define S3 3
5. define OUT 2

int g_count = 0; // count the frequecy int g_array[3]; // store the RGB value int g_flag = 0; // filter of RGB queue float g_SF[3]; // save the RGB Scale factor

// Init TSC230 and setting Frequency. void TSC_Init() {

```
 pinMode(S0, OUTPUT);
 pinMode(S1, OUTPUT);
 pinMode(S2, OUTPUT);
 pinMode(S3, OUTPUT);
 pinMode(OUT, INPUT);

 digitalWrite(S0, LOW);  // OUTPUT FREQUENCY SCALING 2%
 digitalWrite(S1, HIGH);
```

}

// Select the filter color void TSC_FilterColor(int Level01, int Level02) {

```
 if(Level01 != 0)
   Level01 = HIGH;

 if(Level02 != 0)
   Level02 = HIGH;

 digitalWrite(S2, Level01);
```

```
 digitalWrite(S3, Level02);
```

}

void TSC_Count() {

```
 g_count ++ ;
```

}

void TSC_Callback() {

```
 switch(g_flag)
 {
   case 0:
        Serial.println("->WB Start");
        TSC_WB(LOW, LOW);            //Filter without Red
        break;
   case 1:
        Serial.print("->Frequency R=");
        Serial.println(g_count);
        g_array[0] = g_count;
        TSC_WB(HIGH, HIGH);          //Filter without Green
        break;
   case 2:
        Serial.print("->Frequency G=");
        Serial.println(g_count);
        g_array[1] = g_count;
        TSC_WB(LOW, HIGH);           //Filter without Blue
        break;

   case 3:
        Serial.print("->Frequency B=");
        Serial.println(g_count);
        Serial.println("->WB End");
        g_array[2] = g_count;
        TSC_WB(HIGH, LOW);           //Clear(no filter)
        break;
  default:
        g_count = 0;
        break;
 }
```

}

void TSC_WB(int Level0, int Level1) //White Balance {

```
 g_count = 0;
 g_flag ++;
 TSC_FilterColor(Level0, Level1);
 Timer1.setPeriod(250000);           // set 1s period
```

}

void setup() {

```
 TSC_Init();
 Serial.begin(115200);
 Timer1.initialize(250000);           // defaulte is 1s
 Timer1.attachInterrupt(TSC_Callback);
 attachInterrupt(0, TSC_Count, RISING);

 delay(4000);

 for(int i=0; i<3; i++)
   Serial.println(g_array[i]);

 g_SF[0] = 255.0/ g_array[0];     //R Scale factor
 g_SF[1] = 255.0/ g_array[1] ;    //G Scale factor
 g_SF[2] = 255.0/ g_array[2] ;    //B Scale factor

 Serial.println(g_SF[0]);
 Serial.println(g_SF[1]);
 Serial.println(g_SF[2]);
```

}

void loop()
{ if(Serial.available()){ if(Serial.readString()=="Balance"){ Serial.println("Initializing... the ColorSensor , please wait..."); TSC_Init(); delay(1000);

for(int i=0; i<3; i++) Serial.println(g_array[i]);

g_SF[0] = 255.0/ g_array[0]; //R Scale factor g_SF[1] = 255.0/ g_array[1] ; //G Scale factor g_SF[2] = 255.0/ g_array[2] ; //B Scale factor

Serial.println(g_SF[0]); Serial.println(g_SF[1]); Serial.println(g_SF[2]); Serial.println("White Balance is OK"); } } g_flag = 0; for(int i=0; i<3; i++){ Serial.print(','); Serial.print(int(g_array[i] * g_SF[i])); } Serial.println(','); delay(1000);

}

```
</syntaxhighlight>
```

Learning Processing with Arduino

Introduction:

Arduino advanced kit provided by ELECFREAKS is based on our starter kit to expand more and more popular modules owing the characteristics of plug and use without any requirement of soldering. We have well prepared six courses for learners to learn the usage of modules in an intersting environment which is better for stimulating your thinking mind. For example,Tetris games help us to study the usage of the LCD screen and Arduino program. Color picker used can help us to learn the color sensor as well as BLE Bluetooth communication and Android interface development ... Practical Course included LCD screen, sensor, MP3, BLE, WIFI, Android development, and image interaction (processing), make you open-minded!

Part5 Learning Processing with Arduino

Product introduction:

It is connected to a 9-axis sensor on the Arduino board to control Processing programs' running on the computer.

Materials:

Uno board connected to a 9DOF (9-axis sensor), and then connect to the PC via usb cable

*1xFreaduino Uno;

*1x9DOF MODULE V1.2;

```
Parameter and IO definition:

1.5V power supply to mainboard

2.9DOF access to board
 #define SC   A5   // Please notice the Pin's define,A(Analog pin)
 #define SD   A4
 #define V    V(power supply)
 #define G    G(ground)
```

```
Notes

A.Development Tools of Processing Processing2.2.1(above 2.2) need to be downloaded and installed to run processing program on pc end(in part5.zip)

  And this program needs to be connected to one of Arduino ends via USB to get opened, or it can't identify Arduino board. Processing learnig please refer to Chiese website http://www.hiprocessing.net,English website http://www.openprocessing.org/browse/

B.Sensors must be placed horizontally and with terminals toward the right-hand side.

C.To run processing.pde must unzip the entire processing folder, and then load the resource file of the folder to run. The following is the example figure of running.
```

```
Working principles:

Arduino board reads the stance sensor data, and pass through the serial port to the computer, then processing will automatically connect serial ports occupied by Arduino board

to receive the data, so that control program running.
```

Arduino code

part5.zip

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

1. include <Wire.h>

/********** 传感器参数 **********/

1. define Acc (0x53) // ADXL345 地址
2. define Gyr (0x68) // L3205D 地址

3. define Mag 0x1E // HMC5883L 地址

1. define G_SMPLRT_DIV 0x15
2. define G_DLPF_FS 0x16
3. define G_INT_CFG 0x17
4. define G_PWR_MGM 0x3E

// 加速度传感器误差修正的偏移量

int a_offx = -30;

int a_offy = -8; int a_offz = 0; // 陀螺仪误差修正的偏移量

int g_offx = 67; int g_offy = 5; int g_offz = 41;

1. define Gry_offset -13 // 陀螺仪偏移量
2. define Gyr_Gain 0.07 // 满量程 2000dps 时灵敏度(dps/digital)
3. define pi 3.14159

void writeRegister(int device, byte address, byte val); void sensor_init(); int gDat(int device, int axis); void readFrom(int DEVICE, byte address, int num, byte buff[]); void getAccelerometerData(int * result); /********** 互补滤波器参数 *********/ unsigned long preTime = 0; // 采样时间 float f_angleX = 0.0; // 滤波处理后的角度值 float f_angleY = 0.0; // 滤波处理后的角度值 float f_angleZ = 0.0; // 滤波处理后的角度值

/********** PID 控制器参数 *********/ unsigned long lastTime; // 前次时间 float ITerm, lastInput; // 积分项、前次输入 float Output = 0.0; // PID 输出值

/************ 程序初始化 ***********/ void setup() { Wire.begin();

```
  sensor_init();        // 配置传感器

  Serial.begin(19200);  // 开启串口以便监视数据

  delay(1000);
}
```

/************** 主程序 *************/ int acc[3]; int gyro[4]; void loop() {

```
  unsigned long now = millis();                         //
当前时间(ms)

  float dt = (now - preTime) / 1000.0;
// 微分时间(s)

  int time=now-preTime;
```

```
  preTime = now;                                          // 记
录本次时间(ms)
```

/********** 读取姿态传感器 *********/

```
  // float Y_Acc = gDat(Acc, 1);
// 获取向前的加速度(digite)
  getAccelerometerData(acc);
  getAccelerometerData(gyro);
  float X_Acc = acc[0];                                   // 获取向前
的加速度(digite)
  float Y_Acc = acc[1];                                   // 获取向前
的加速度(digite)
  float Z_Acc = acc[2];                                   // 获取向下
的加速度(digite)
  float angleX = atan(Y_Acc / Z_Acc) * 180 / pi;   // 根据
加速度分量得到的角度(degree)
  float angleY = atan(X_Acc/Z_Acc  ) * 180 / pi;   // 根据
加速度分量得到的角度(degree)
  float angleZ = atan(X_Acc / Y_Acc) * 180 / pi;   // 根据
加速度分量得到的角度(degree)
 // float omega =  Gyr_Gain * (gDat(Gyr, 0) +
Gry_offset); // 当前角速度(degree/s)
  float omegaX =  Gyr_Gain * (gyro[0] +  Gry_offset); //
当前角速度(degree/s)
```

```
  float omegaY =  Gyr_Gain * (gyro[1] +  Gry_offset); //
当前角速度(degree/s)
  float omegaZ =  Gyr_Gain * (gyro[2] +  Gry_offset); //
当前角速度(degree/s)
```

/********** 一阶互补滤波 **********/

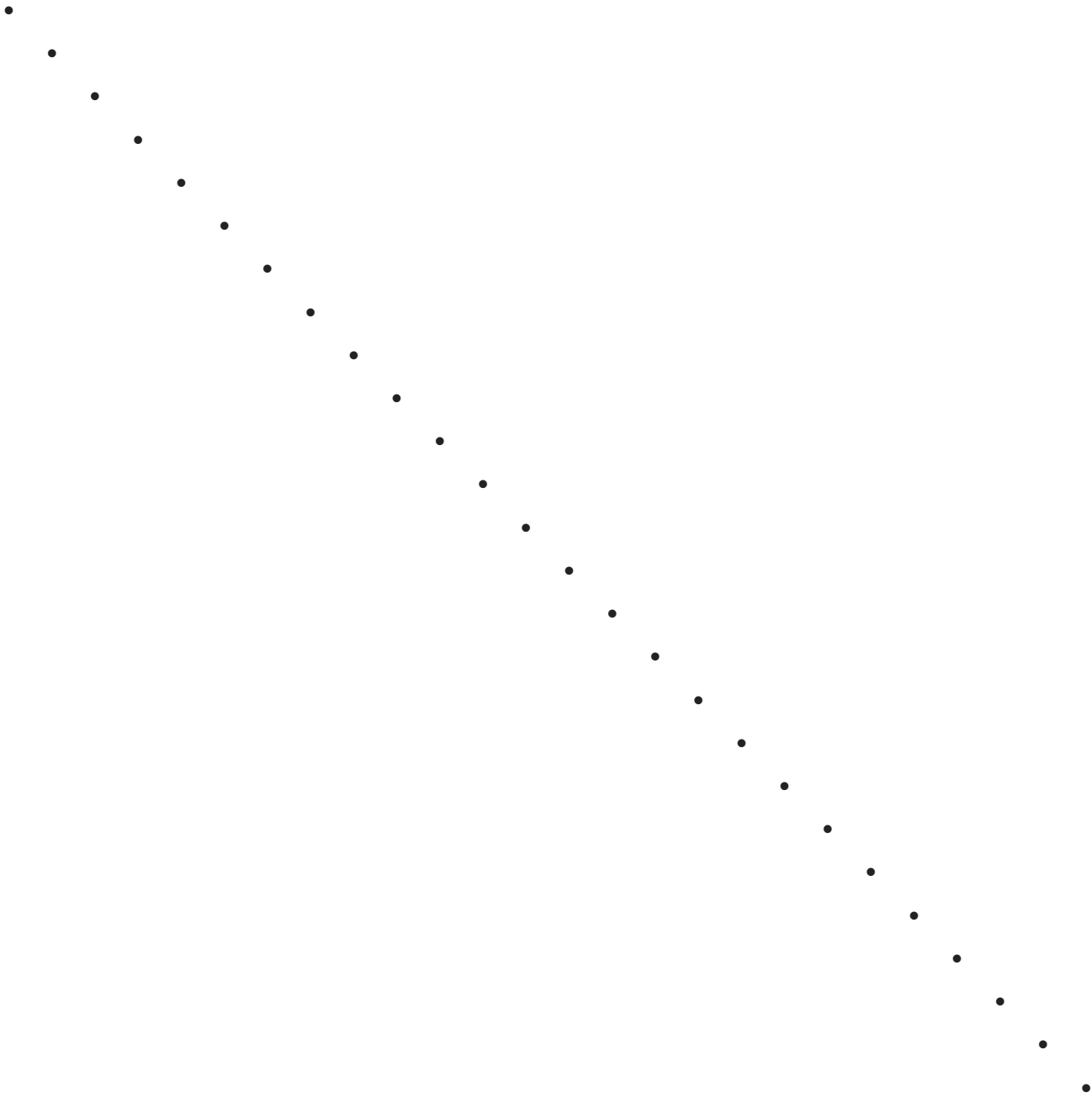
```
  float K = 0.8;                                    // 取值
权重
  float A = K / (K + dt);                            // 加
权系数
  f_angleX =(A *(f_angleX + omegaX * dt) + (1-A) *
angleX);  // 互补滤波算法
  f_angleY =(A *(f_angleY + omegaY * dt) + (1-A) *
angleY);  // 互补滤波算法
  f_angleZ =(A *(f_angleZ + omegaZ * dt) + (1-A) *
angleZ);  // 互补滤波算法
```

```
  Serial.print(",");
  Serial.print("X=");   // 偏离角度
  Serial.print(",");
  Serial.print(f_angleX, 2);   // 偏离角度
  Serial.print(",");
  Serial.print("Y=");   // 偏离角度
  Serial.print(",");
  Serial.print(f_angleY, 2);   // 偏离角度
  Serial.println(",");
```

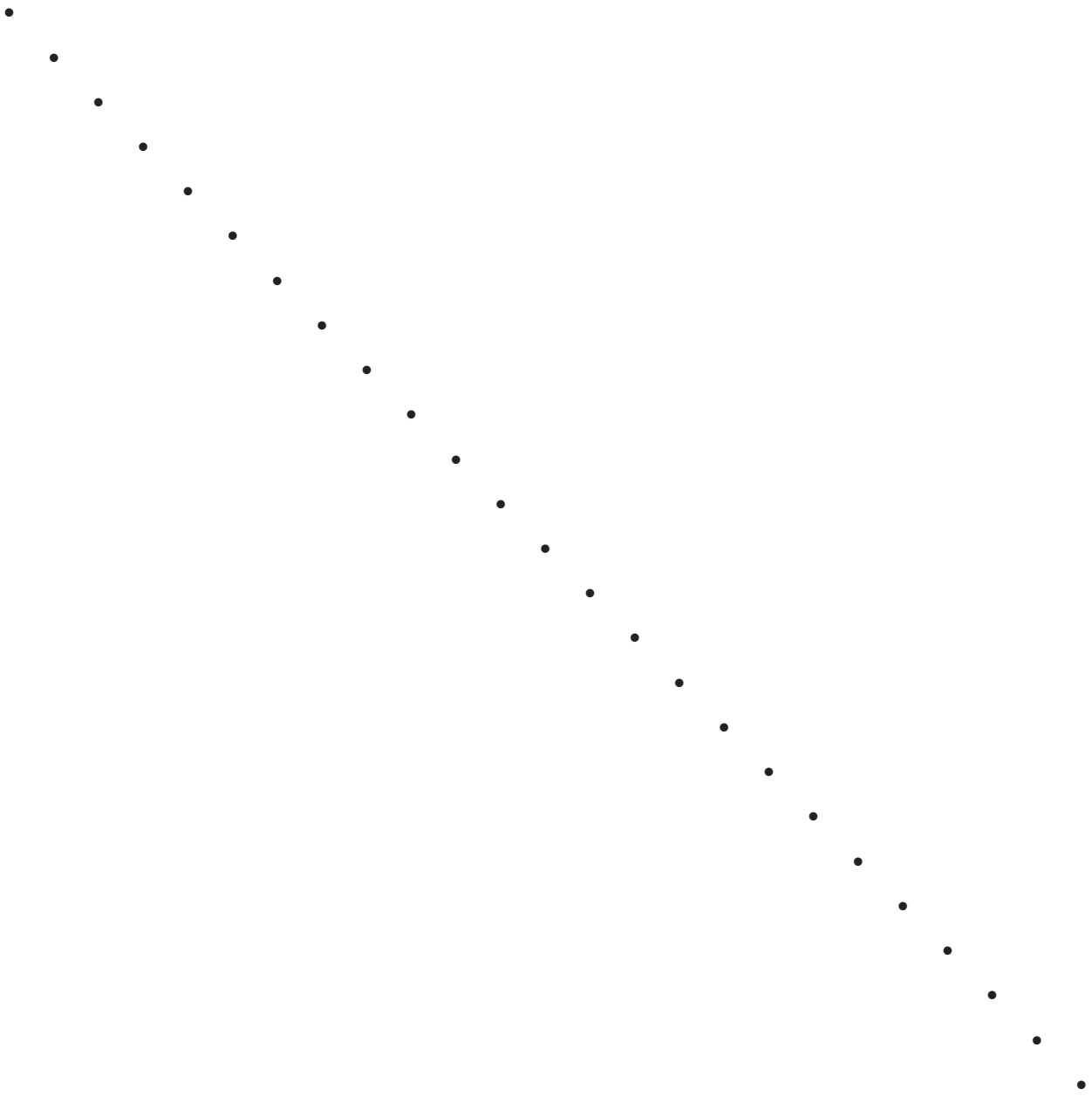
```
  delay(10);
```

}

/************************************ 九轴姿态传感器寄存器读取函数 For Arduino,



调用参数表

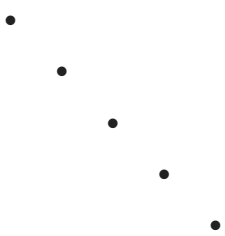


```
type    device      axis
                0   1   2
```

ADXL345 Acc x y z L3G4200D Gyr x y z HMC5883L Mag x z y

Example

```
00 #include <Wire.h> 01 #define Acc 0x1D; 02 #define Gyr 0x69; 03 #define Mag 0x1E;
04 05 void setup() { 06 sensor_init(); 07 delay(1000); 08 } 09 10 void loop() { 11 int Z-
Gyroscope; 12 Z-Gyroscope = gDat(Gyr, 2); 13 delay(50); 14 }
```



```
int gDat(int device, int axis) {
```

```
  int v;
  byte vL, vH, address;              // 存放 byte 数值
  if (device == Acc) address = 0x32;  // ADXL345 的读数地址
  if (device == Gyr) address = 0x1B;  // L3G3200D 的读数地址
  if (device == Mag) address = 0x03;  // HMC5883L 的读数地址
  address = address + axis * 2;       // 数据偏移-坐标轴
  Wire.beginTransmission(device);     // 开始传输数据
  Wire.write(address);                // 发送指针
  Wire.endTransmission();
  Wire.beginTransmission(device);     // 开始传输数据
  Wire.requestFrom(device, 2);        // 请求 2 byte 数据
  if(Wire.available())vL = Wire.read();       // 成功获取前等待
  if(Wire.available())vH = Wire.read();      // 读取数据
  Wire.endTransmission();             // 结束传输
  if (device == Mag) v = (vL << 8) | vH;
  else v = (vH << 8) | vL;            // 将 byte 数据合并为 Int
  return v;                           // 返回读书值
```

} //得到 加速度的数据 void getAccelerometerData(int * result) {

```
 int regAddress = 0x32;    //加速度传感器 ADXL345 第一轴的数据的设定
 byte buff[6];
```

```
 readFrom(Acc, regAddress,6, buff); //读取加速度传感器

ADXL345 的数据

 //每个轴的读数有 10 位分辨率，即 2 个字节.

 //我们要转换两个 bytes 为一个 int 变量

result[0] = (((int)buff[1]) << 8) | buff[0] + a_offx;
result[1] = (((int)buff[3]) << 8) | buff[2] + a_offy;
result[2] = (((int)buff[5]) << 8) | buff[4] + a_offz;
```

} //得到陀螺仪的值 void getGyroscopeData(int * result)

{

```
 int regAddress = 0x1B;
 int temp, x, y, z;
 byte buff[8];

 readFrom(Gyr, regAddress,8, buff); //读取陀螺仪 ITG3200 的数
据
 result[0] = ((buff[2] << 8) | buff[3]) + g_offx;
 result[1] = ((buff[4] << 8) | buff[5]) + g_offy;
 result[2] = ((buff[6] << 8) | buff[7]) + g_offz;

 result[3] = (buff[0] << 8) | buff[1]; // 温度
```

}

//读 IIC 数据 void readFrom(int DEVICE, byte address, int num, byte buff[]) {

```
 Wire.beginTransmission(DEVICE); //开始传送至加速度传感器

 Wire.write(address);        //发送读取的地址

 Wire.endTransmission(); //结束传输

 Wire.beginTransmission(DEVICE); //开始传送到 ACC
```

```
 Wire.requestFrom(DEVICE, num);    // 要求从加速度传感器中发送
6 个字节的数据
 int i = 0;
 while(Wire.available())    //当加速度传感器返回的数据小于要求值
时（异常情况）
 {
   buff[i] = Wire.read(); // 接收数据
   i++;
 }
 Wire.endTransmission(); //结束传输
```

}/**********配置九轴姿态传感器

void sensor_init() { // 配置九轴姿态传感器

```
//writeRegister(Acc, 0x2D, 0b00001000);    // 测量模式
                    // 配置 ADXL345
//调用 ADXL345
```

writeRegister(Acc, 0x2D, 0);

```
writeRegister(Acc, 0x2D, 16);
writeRegister(Acc, 0x2D, 8);
```

//设定在 +-2g 时的默认读数

// writeRegister(Gyr, 0x20, 0b00001111); // 设置睡眠模式、x, y, z 轴使能 // writeRegister(Gyr, 0x21, 0b00000000); // 选择高通滤波模式和高通截止频率 // writeRegister(Gyr, 0x22, 0b00000000); // 设置中断模式 // writeRegister(Gyr, 0x23, 0b00110000); // 设置量程(2000dps)、自检状态、SPI 模式 // writeRegister(Gyr, 0x24, 0b00000000); // FIFO & 高通滤波 // // 配置 L3G4200D(2000 deg/sec)

```
  writeRegister(Gyr, G_PWR_MGM, 0x00);
  writeRegister(Gyr, G_SMPLRT_DIV, 0x07); // EB, 50, 80,
7F, DE, 23, 20, FF
  writeRegister(Gyr, G_DLPF_FS, 0x1E);    // +/- 2000
dgrs/sec, 1KHz, 1E, 19
  writeRegister(Gyr, G_INT_CFG, 0x00);
 // writeRegister(Mag, 0x02, 0x00);         // 连续测量

                        // 配置 HMC5883L
```

} /********** 寄存器写入函数

void writeRegister(int device, byte address, byte val) {

```
  Wire.beginTransmission(device);          // 写入的传感器
  Wire.write(address);                     // 写入地址
  Wire.write(val);                         // 写入值
  Wire.endTransmission();                  // 结束传输
```

}

```
</syntaxhighlight>
```

Smarthome Wifi Remote Command

Introduction:

Arduino advanced kit provided by ELECFREAKS is based on our starter kit to expand more and more popular modules owing the characteristics of plug and use without any requirement of soldering. We have well prepared six courses for learners to learn the usage of modules in an intersting environment which is better for stimulating your thinking mind. For example,Tetris games help us to study the usage of the LCD screen and Arduino program. Color picker used can help us to learn the color sensor as well as BLE Bluetooth communication and Android interface development ... Practical Course included LCD screen, sensor, MP3, BLE, WIFI, Android development, and image interaction (processing), make you open-minded!

Part6 Smarthome Wifi Remote Command

Product introduction:

When your Arduino is connected to the network via wifi module, you can control your Arduino board via the Internet. Because of the limited conditions, here with LAN case.

Materials:

an Arduino board, a wifi module, a relay, of course you still need a wireless router and an internet device, we also prepared an Android APK for demo.

*1xFreaduino Uno;
*1xESP Wifi Module;
*1x1 Channel Relay;

```
Parameter and IO definition:

1.Board requires a 3.3V power supply. Higher than this voltage
will lead to unstable working of wifi module.

2.ESP WIFI access to board
  #define V    V(power supply)
  #define TX   D4(RX)   // Please notice the Pin's define,D(Digital
pin)
  #define RX   D5(TX)
  #define G    G(Ground)
```

```
  #ESP WIFI GIO0 used when updating firmware and you need to
connect this pin to ground
```

```
Notes:

  1、First install the part6 APK in the Android cell phone
  2、upload code to UNO, and according to the card after
connection and power
  3、Search and connect WiFi with elecfreaks_ap
  4、Open the app and click the X button on the app, if it turns
into a V icon to indicate the connection is successful.
  5、AT commands please view ESP WIFI reference materials below,
or you can also downloaded part6.zip for full profile.
```

```
ESP WIFI reference materials:
```

Arduino code

part6.zip

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

1. include <SoftwareSerial.h>
2. include <Timer.h>

1. define TCPServer_IP "192.168.1.100"
2. define TCPServer_Port "200"
3. define TCPServer_OverTime "1800"
4. define macUser_ESP8266_BulitApSsid "ELECFREAKS_AP"
5. define macUser_ESP8266_BulitApEcn 0
6. define macUser_ESP8266_BulitApPwd "elecfreaks"
7. define wifiBaud 9600
8. define wifiRx 4
9. define wifiTx 5

Timer timer; SoftwareSerial wifi(wifiRx,wifiTx);

int sendATcommand(char* ATcommand, char* expected_answer1, unsigned int timeout){

```
   return sendATcommand2(ATcommand, expected_answer1, " ",
timeout);
```

}

int sendATcommand2(char* ATcommand, char* expected_answer1, char* expected_answer2,

```
   unsigned int timeout){
   uint8_t x=0,  answer=0;
   char response[100];
   unsigned long previous;
   memset(response, '\0', 100);    // Initialize the
string

   delay(100);

   while( wifi.available() > 0) wifi.read();    // Clean
the input buffer

   wifi.println(ATcommand);    // Send the AT command
```

```
   x = 0;
   previous = millis();
   // this loop waits for the answer
   do{
      // if there are data in the UART input buffer,
reads it and checks for the asnwer
```

// if(mySerial.available() != 0){ // response[x] = mySerial.read(); // x++;

```
          if(wifi.available())
          {
            do{
              response[x++] = wifi.read();
              }while(wifi.available());
```

```
          // check if the desired answer 1 is in the
response of the module
          if (strstr(response, expected_answer1) != NULL)
          {
             answer = 1;
          }
          // check if the desired answer 2 is in the
response of the module
          if (strstr(response, expected_answer2) != NULL)
          {
             answer = 2;
          }
       }
       Serial.println(response);
   // Waits for the asnwer with time out
   }while((answer == 0) && ((millis() - previous) <
timeout));
   return answer;
```

}

void ESP8266_AT_Test (void) {

```
 while ( ! sendATcommand2 ( "AT", "OK", NULL, 200 ) ) ;
```

}

void ESP8266_Net_Mode_Choose(void) {

```
 sendATcommand2 ( "AT+CWMODE=2", "OK", "no change",
2500 );
```

}

void ESP8266_ExitUnvarnishSend (void) { delay (1000);

wifi.print("+++");

delay (500);

}

uint8_t ESP8266_CIPAP (char * pApIp) { char cCmd [30];

sprintf (cCmd, "AT+CIPAP=\"%s\"", pApIp);

```
if ( sendATcommand2 ( cCmd, "OK", 0, 5000 ) )
```

return 1;

else return 0;

}

int ESP8266_BuildAP (char * pSSID, char * pPassWord, int mode) { char cCmd [120];

sprintf (cCmd, "AT+CWSAP=\"%s\",\"%s\",1,%d", pSSID, pPassWord, mode);

return sendATcommand2 (cCmd, "OK", 0, 1000);

}

int ESP8266_Enable_MultipleId (int mode) { char cStr [20];

sprintf (cStr, "AT+CIPMUX=%d", mode);

return sendATcommand2 (cStr, "OK", 0, 500);

}

int ESP8266_StartOrShutServer (int mode, char * pPortNum, char * pTimeOver) { char cCmd1 [120], cCmd2 [120];

if (mode) { sprintf (cCmd1, "AT+CIPSERVER=%d,%s", 1, pPortNum);

sprintf (cCmd2, "AT+CIPSTO=%s", pTimeOver);

return (sendATcommand2 (cCmd1, "OK", 0, 500) && sendATcommand2 (cCmd2, "OK", 0, 500)); }

else { sprintf (cCmd1, "AT+CIPSERVER=%d,%s", 0, pPortNum);

return sendATcommand2 (cCmd1, "OK", 0, 500); }

}

String info; void receiveInfo(){ if(wifi.available()){ info=wifi.readString(); Serial.print(info); if(info.lastIndexOf("ON")>info.lastIndexOf("OFF")){ digitalWrite(A0,1); }else if(info.lastIndexOf("ON")<info.lastIndexOf("OFF")) { digitalWrite(A0,0); } else return; //http response /* String id=info.substring(info.indexOf(',')+1,info.lastIndexOf(',')); wifi.write("AT+CIPSEND="); wifi.write(id); wifi.write(",126\r\n"); wifi.println("HTTP/1.1 200 OK"); wifi.println("Cache-Control: no-cache"); wifi.println("Pragma: no-cache"); wifi.println("Expires: -1"); wifi.println("Content-Type: text/html"); wifi.println("Content-Length: 12"); wifi.println(" "); wifi.println("it received"); */ }}

void setup(void) {

```
Serial.begin(9600);
wifi.begin(9600);
pinMode(A0,OUTPUT);
//ESP8266_ExitUnvarnishSend();
```

```
 ESP8266_AT_Test();//AT command test
 ESP8266_Net_Mode_Choose();//set ap mode
 //while ( ! ESP8266_CIPAP ( TCPServer_IP ) );
 //ESP8266_CIPAP ( TCPServer_IP );
 ESP8266_BuildAP ( macUser_ESP8266_BulitApSsid,
macUser_ESP8266_BulitApPwd,
macUser_ESP8266_BulitApEcn );//set ap ssid passwd
 ESP8266_Enable_MultipleId(1);//set multi connect
 ESP8266_StartOrShutServer ( 1, TCPServer_Port,
TCPServer_OverTime );//set server port,over time
```

}

void loop() {

```
 receiveInfo();
 while(Serial.available()){
   wifi.print(Serial.readString());
 }
```

}

```
</syntaxhighlight>
```