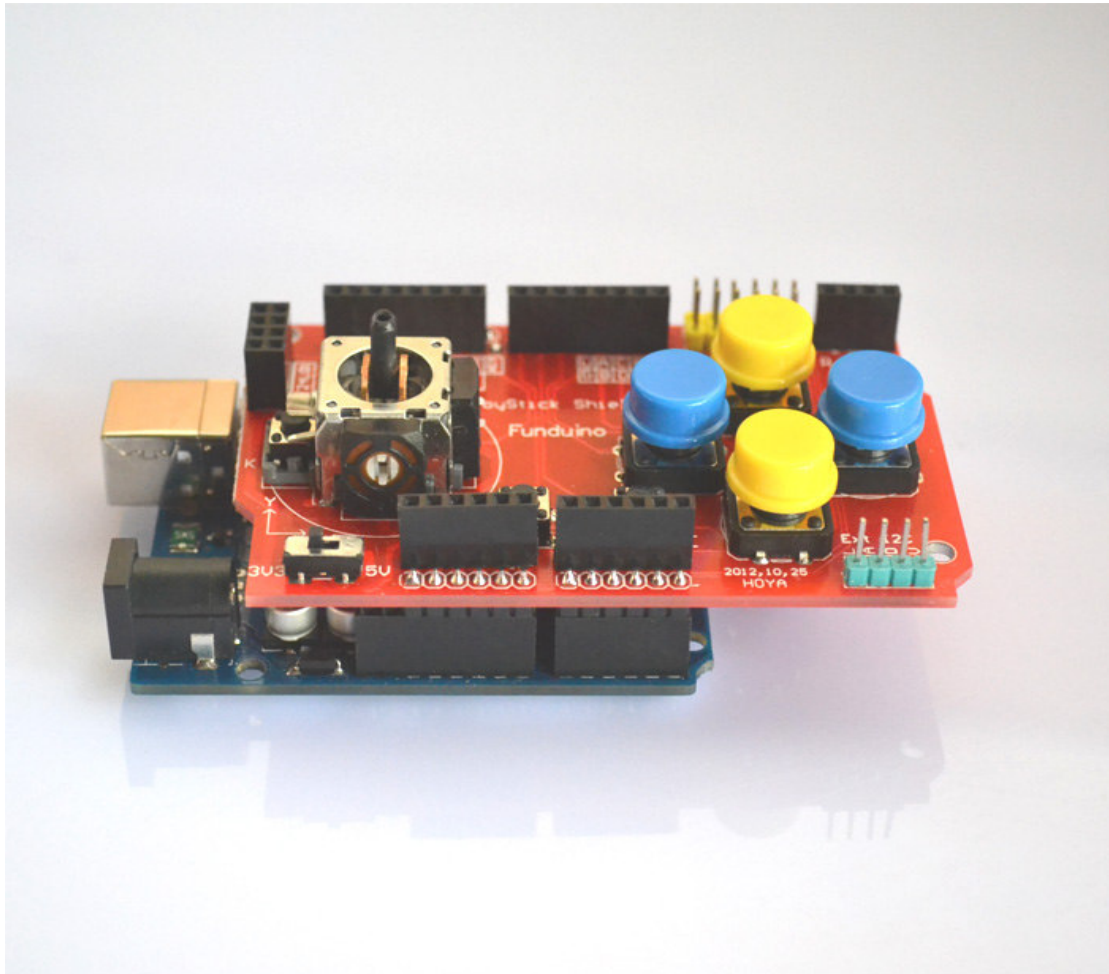


## Joystick shield



Joystick shield adds RF interface of nRF24L0 and interface of Nokia 5110 LCD , so it's easy for game development.

Adding a Bluetooth module interface, it's convenient for Bluetooth wireless serial communication.

Adding I2C communication interface, it's easy to connect to I2C devices.

Meanwhile, Joystick Shield is with a cross PS2 and buttons, there are four round buttons, 2 small buttons which Provide for the Arduino joystick and buttons input expansion. Onboard switches can switch between 3.3V and 5V, and can use this model for other 3.3V microcontroller platforms, such as STM32.

## Test source code is as follows:

```
/******  
** Device: Joystick **  
** File: EF_Joystick_Test.c **  
** **  
** Created by ElecFreaks Robi.W /10 June 2011 **  
** **  
** Description: **  
** This file is a sample code for your reference. **  
** **  
** Copyright (C) 2011 ElecFreaks Corp. **  
*****/
```

```
int FirstShotX , FirstShotY;
```

```
void setup()
```

```
{  
  for(int i=0; i<9; i++)  
  {  
    pinMode(i, INPUT);  
    digitalWrite(i, 1);  
  }  
}
```

```
Serial.begin(9600);
```

```
FirstShotX = 0;
```

```
FirstShotY = 0;
```

```
}
```

```
void loop(){
```

```
  int i, someInt, flag = 0;
```

```
  for(i=2; i<9; i++)
```

```
  {  
    someInt = digitalRead(i);  
    if(someInt == 0)  
    {  
      flag =1;  
      break;  
    }  
  }  
  if(flag == 1)  
  {
```

```

switch(i)
{
  case 2: Serial.println("-----> Button A"); break;
  case 3: Serial.println("-----> Button B"); break;
  case 4: Serial.println("-----> Button C"); break;
  case 5: Serial.println("-----> Button D"); break;
  case 6: Serial.println("-----> Button E"); break;
  case 7: Serial.println("-----> Button F"); break;
  case 8: Serial.println("-----> Button KEY"); break;
  default: break;
}
flag=0;
}

int sensorValue = analogRead(A0);
if(FirstShotX == 0)
{
  FirstShotX = sensorValue;
  Serial.print("FirstShotX = ");
  Serial.println(FirstShotX);
}

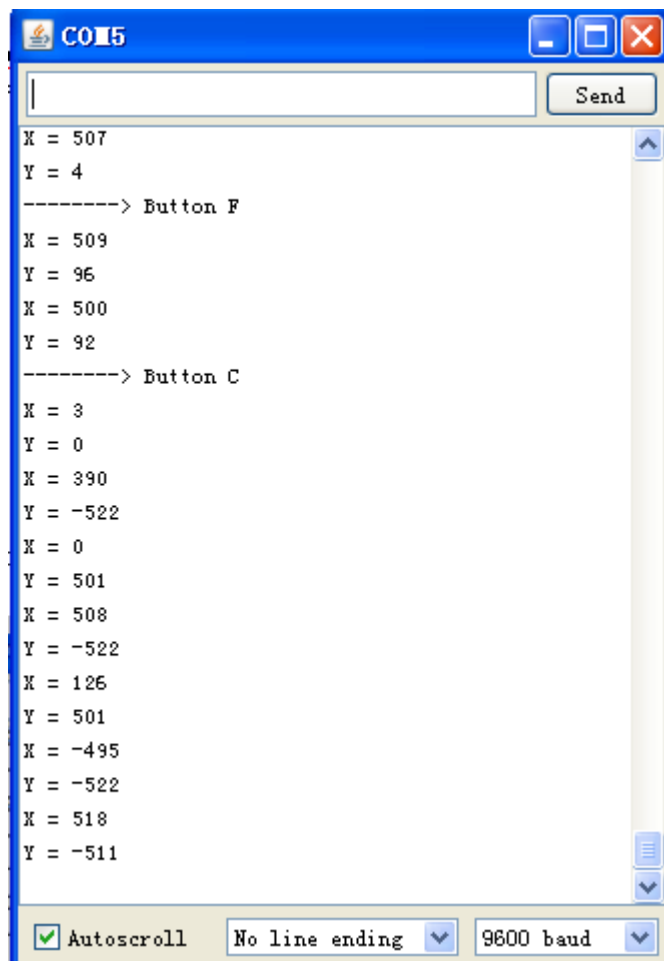
Serial.print("X = ");
Serial.println(sensorValue - FirstShotX);

sensorValue = analogRead(A1);
if(FirstShotY == 0)
{
  FirstShotY = sensorValue;
  Serial.print("FirstShotY = ");
  Serial.println(FirstShotY);
}
Serial.print("Y = ");
Serial.println(sensorValue - FirstShotY);

delay(200);
}

```

Following is the result of test, just for reference:



The image shows a screenshot of a Windows terminal window titled "COM15". The window has a blue title bar with standard minimize, maximize, and close buttons. Below the title bar is a text input field and a "Send" button. The main area of the window displays the following text:

```
X = 507
Y = 4
-----> Button F
X = 509
Y = 96
X = 500
Y = 92
-----> Button C
X = 3
Y = 0
X = 390
Y = -522
X = 0
Y = 501
X = 508
Y = -522
X = 126
Y = 501
X = -495
Y = -522
X = 518
Y = -511
```

At the bottom of the window, there are three control elements: a checked checkbox labeled "Autoscroll", a dropdown menu showing "No line ending", and another dropdown menu showing "9600 baud".