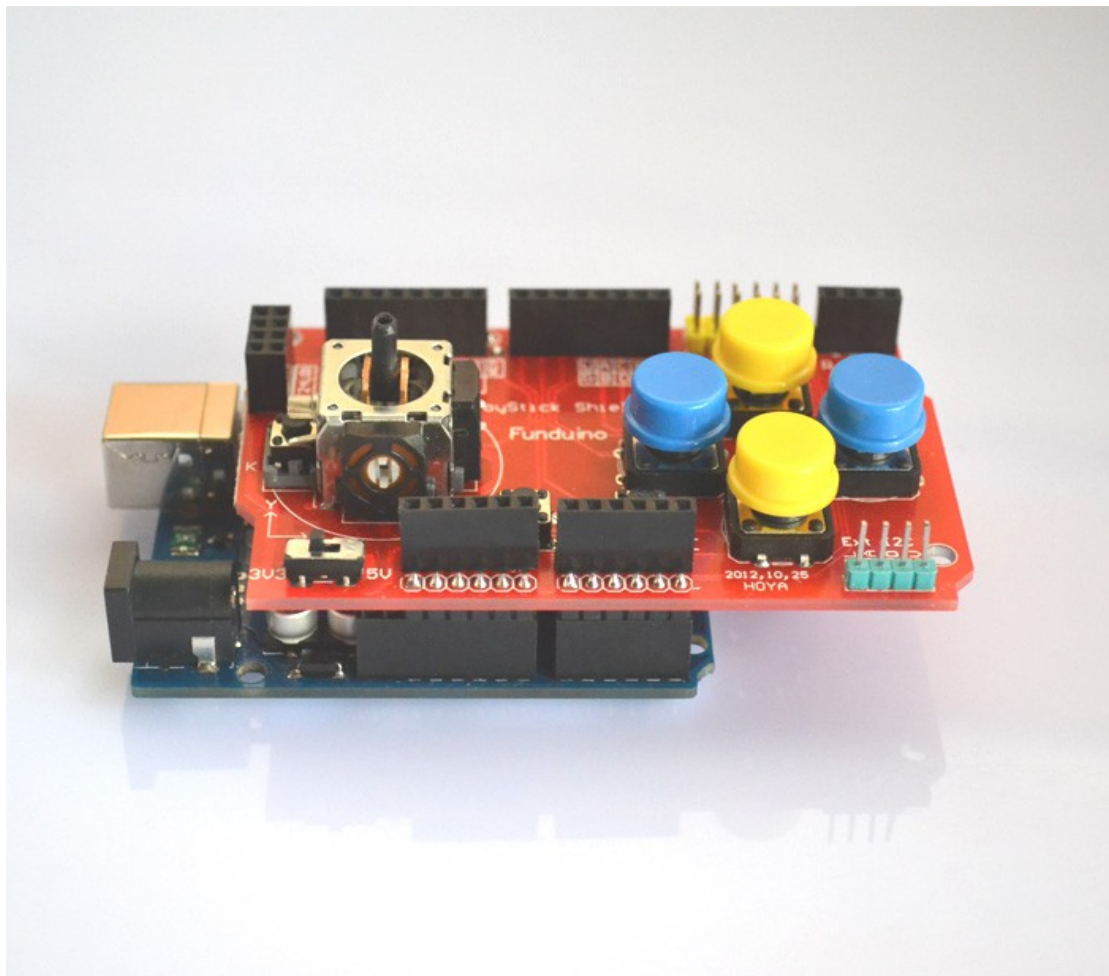


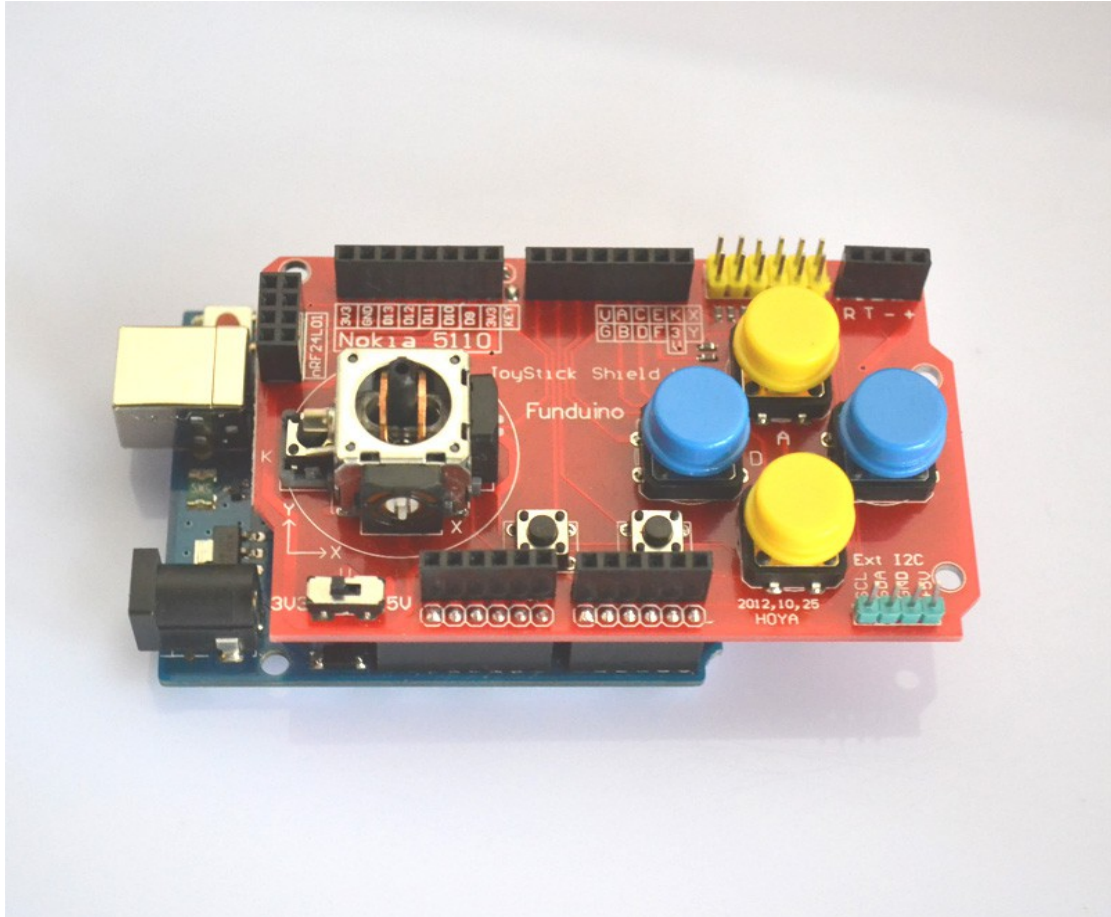
Joystick shield 游戏手柄扩展板

Joystick Shield 添加了 nRF24L01 的 RF 接口和 Nokia 5110 LCD 接口，这样非常方便二次的游戏开发。之所以添加 nRF24L01 的 RF 接口原因是此 RF 模块性能稳定，而且最大支持 1 拖 6 的玩法，可以让开发者充分的自由发挥。

另外，采用 Nokia 5110 LCD 显示屏是因为此屏在各种开发平台都大量使用，性价比高，并且有强大的库支持，这里我们提供的增强的 Nokia5110 的显示库，包括显示位图，画点线圆翻转等等

如下图，将游戏模块插在控制主板上就可以开始测试了





侧测试代码如下：

```
int FirstShotX , FirstShotY;
```

```
void setup()
```

```
{  
  for(int i=0; i<19; i++)  
  {  
    pinMode(i, INPUT);  
    digitalWrite(i, 1);  
  }  
}
```

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

```
  FirstShotX = 0;
```

```
  FirstShotY = 0;
```

```
}
```

```
void loop(){
```

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

```
  for(i=4; i<11; i++)
```

```
{
  someInt = digitalRead(i);
  if(someInt == 0)
  {
    flag =1;
    break;
  }
}
if(flag == 1)
{
  switch(i)
  {
    case 4: Serial.println("-----> Button A"); break;
    case 5: Serial.println("-----> Button B"); break;
    case 6: Serial.println("-----> Button C"); break;
    case 7: Serial.println("-----> Button D"); break;
    case 8: Serial.println("-----> Button E"); break;
    case 9: Serial.println("-----> Button F"); break;
    case 10: 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);  
}
```

下面是本次测试结果，仅供参考。根据不同的操作手法，会出现不同的数值。

