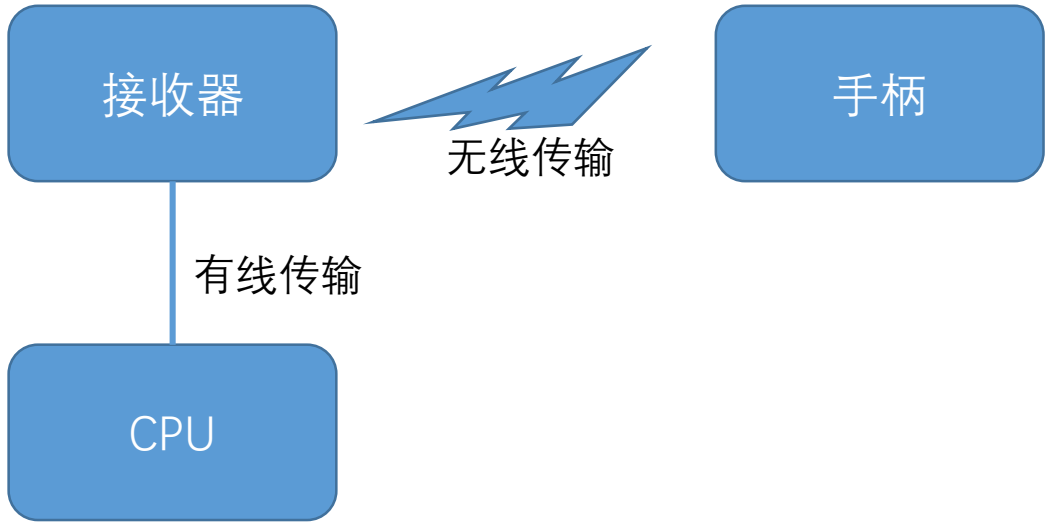
无线遥控手柄

无线遥控手柄原理

- 手柄与接收器之间内置无线传输协议，只需要简单的配对即可使用
- 接收器与CPU使用有线数据传输

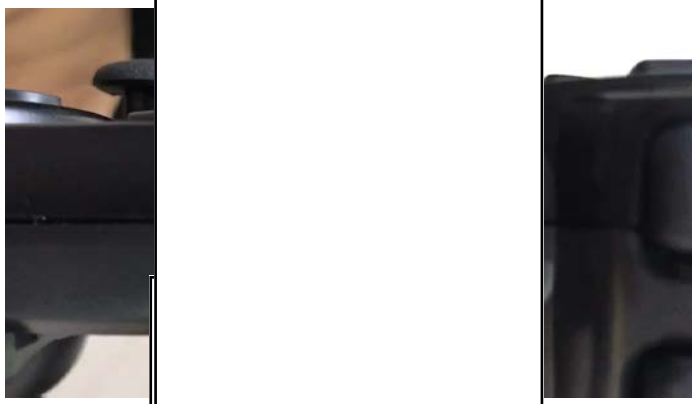


遥控器原理示意

手柄与接收器

无线遥控手柄配对

- 未配对的手柄和接收器在上电后会自动配对，在手柄和接收器数量比较多的时候（比如现在的教室），容易出现A手柄配到B小车的情况，如果出现这种情况，需要大家反复拔插小车电源重试，或者找一个人少的位置进行配对



未配对的手柄与接收器（视频）

手柄与接收器配对（视频）

读取接收器数据

- 手柄模块已经封装在PS2.ino中。并在范例中完整给出

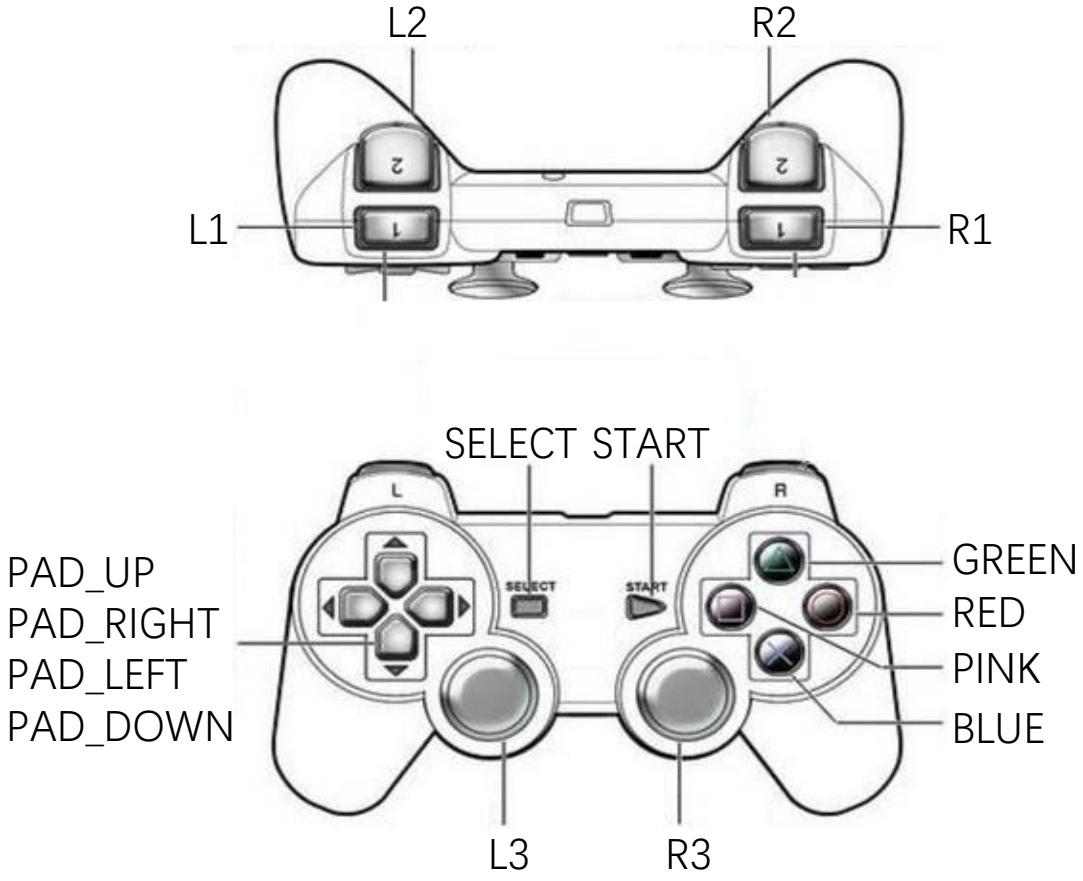
手柄初始化

读取手柄摇杆和按键状态
4个输入参数是分别用来存放
左右摇杆X和Y值的变量
返回值是按键按键的代码

```
void setup() {
  OLED_Init();//OLED初始化
  PS2Init();//PS2初始化
}
void loop() {
  unsigned long button = 0;//存放按键值
  int LX = 0, LY = 0, RX = 0, RY = 0;//存放左右摇杆值
  button = PS2Read(&LX, &LY, &RX, &RY);//读取PS2手柄按键和摇杆
  char a[20];//定义一个字符串变量（字符数组）
  sprintf(a,"LX:%3d LY:%3d",LX,LY);//填充字符串变量,左摇杆xy
  OLED12864_ShowStr(0,0,a) ; //用OLED输出字符串变量
  sprintf(a,"RX:%3d RY:%3d",RX,RY);//填充字符串变，右摇杆xy
  OLED12864_ShowStr(0,1,a) ; //用OLED输出字符串变量
  sprintf(a,"Button Code:0x%04x",button);//填充字符串变，按键值
  OLED12864_ShowStr(0,2,a) ; //用OLED输出字符串变量
}
```

手柄按键代码

- 摇杆每个维度的取值范围是0-255，中位值是127或128
- 按键的定义在PS2.ino中，如果要在主文件中，需要把这些定义拷贝到主文件




```
#define PSB_SELECT 0x0001
#define PSB_L3 0x0002
#define PSB_R3 0x0004
#define PSB_START 0x0008
#define PSB_PAD_UP 0x0010
#define PSB_PAD_RIGHT 0x0020
#define PSB_PAD_DOWN 0x0040
#define PSB_PAD_LEFT 0x0080
#define PSB_L2 0x0100
#define PSB_R2 0x0200
#define PSB_L1 0x0400
#define PSB_R1 0x0800
#define PSB_GREEN 0x1000
#define PSB_RED 0x2000
#define PSB_BLUE 0x4000
#define PSB_PINK 0x8000
```

判断按键与遥控小车

```
button = PS2Read(&LX, &LY, &RX, &RY);
if(button==PSB_L1)
{
  ° ° ° ° ° °
}
if(button&PSB_L1 && button&PSB_R1)
{
  ° ° ° ° ° °
}
```

判断L！是否按下

判断L！与R1是否同时按下，这种方式不是很好理解，但可以判断同时按下的几个按键，不强制要求

按键判断

简单的手柄运动控制

```
if(RY>200)
{
  //前进
}
else if(RY<50)
{
  //后退
}
else if(LX>200)
{
  //右转
}
else if(LX<50)
{
  //左转
}
else
{
  //停车
}
```

复杂点的的手柄运动控制

```
void carCtrl(int RX, int RY)
{
  int ySpeed = 0, xSpeed = 0, leftSpeed = 0,rightSpeed = 0;
  ySpeed = (0 - (RY - 0x80));
  xSpeed = (RX - 0x80);
  leftSpeed = ySpeed + xSpeed;
  rightSpeed = ySpeed - xSpeed;
  carMove(leftSpeed, rightSpeed);
}
```

0x80就是十六进制的128
大家读一读这段程序，看看是什么意思

任务

- 自定义手柄左右摇杆，控制机器人运动
- 通过按键控制机器人点动作