

```

ret = tls_uart_read(TLS_UART_1, (u8 *)rec_buf, len); /* 读取串口的数据 存放到
rec_buf*/
if (ret <= 0)
{
    break;
}
// 解析数据 rec_buf      ret=30
for (int i = 0; i < ret; i++)
{
    if (rec_buf[i] == 0xAA && rec_buf[i + 1] == 0xFF)
    {
        if (i + 6 < ret)
        {
            // 解析x和y距离
            u16 x_distance = rec_buf[i + 4] | (rec_buf[i + 5] << 8);
            u16 y_distance = rec_buf[i + 6] | (rec_buf[i + 7] << 8);

            // 检查x_distance的最高位
            if (rec_buf[i + 5] & 0x80)
            {
                // x_distance最高位为1, 表示正数, 减去最高位的1。y永远为正直接减去0x8000
                x_distance -= 0x8000;
                y_distance -= 0x8000;
                // 输出距离
                printf("x:%d,y:%d\r\n", x_distance, y_distance);
            }
            else // 最高位为0 表示x负距离。y永远为正直接减去0x8000
            {
                y_distance -= 0x8000;
                // 输出距离
                printf("x:%d,y:%d\r\n", x_distance, y_distance);
            }

            // 偏移索引以跳过已处理的数据
            i += 7;
        }
    }
}
}

```