

Беспроводной джойстик PS2

Указания к использованию



supereye.ru
СУПЕР

eyes.ru
РАЙС

Оглавление

1 Описание джойстика PS2.....	3
2 Указания к использованию и подключению джойстика	5
3 Подключение к компьютеру.....	6
4 Составление скриптов.....	6
5 Загрузка и проверка	12



1 Описание джойстика PS2

Джойстик PS2 состоит из управляющего и приемного устройств, главная задача самого джойстика – отправление сигналов с клавиш. Приемник и микроконтроллер (далее – устройство, например, сама игровая консоль PS2) сопряжены друг с другом. Информация, полученная приемником, поступает на устройство, которое, иногда с помощью приемника, передает команды на джойстик и определяет режим отправки данных с джойстика.

Примечание: внешний вид джойстиков и приемников может отличаться в зависимости от партии или завода-изготовителя (на всех приемниках присутствуют индикаторы, но на некоторых дополнительно присутствуют индикаторы питания), но назначение контактов у всех приемников одинаковое, тип дешифрования одинаковый.

Выходные контакты приемника:

1	2	3	4	5	6	7	8	9
DI/DAT	DO/CMD	NC	GND	VDD	CS/SEL	CLK	NC	ACK

Приемник показан на фото ниже (обратите внимание на порядок портов).

DI/DAT: сигнал направлен от джойстика к устройству, последовательный 8bit, синхронная передача заднего фронта тактового сигнала. Сигнал читается сверху вниз относительно тактового сигнала.

DO/CMD: сигнал направлен от устройства к джойстику противоположно сигналу DI, последовательный 8bit, синхронная передача заднего фронта тактового сигнала.

NC: пустой интерфейс.

GND: заземление.

VDD: порт питания приемника, диапазон напряжения 3-5 В.

CS/SEL: предназначен для подачи триггерного сигнала джойстика, во время передачи данных работает на низком уровне.

NC: пустой интерфейс.

ACK: передает ответный сигнал с джойстика на устройство. После каждой серии данных посылается ответный бит, затем уровень сигнала понижается обратно. CS сигнал всегда находится на низком уровне. Если сигнал CS не понижается, то через 60 мс устройство PS попытается подключиться к другому периферийному устройству. В процессе составления скрипта интерфейс ACK не задействуется.



Рисунок 1 – Обозначение интерфейсов приемника

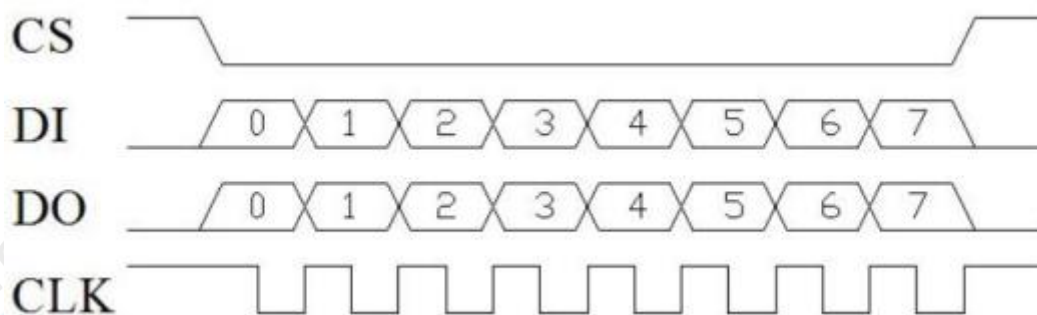


Рисунок 2 Временная диаграмма интерфейсов

Тактовая частота 250 кГц (4 мкс), если данные с приемника нестабильны, допускается увеличить тактовую частоту.

После передачи одной серии данных (а не после передачи первого бита) сигнал CS возрастает. Во время передачи его уровень всегда низкий.

Прием и отправка, отправка и прием конечных данных (1 бит) заканчиваются одновременно с переключением на задний фронт тактового сигнала. Когда устройство захочет считать данные с джойстика или отправить команду на джойстик, уровень CS понизится, отправится команда «0x01»; джойстик отправит в ответ ID «0x41=зеленый индикатор, 0x73= красный индикатор». Одновременно при отправке ID джойстика устройство отправит 0x42, запрос данных. Затем джойстика отправит 0x5A, ответ устройству «данные поступили».

idle: канал данных пуст, передача отсутствует.

Некоторые серии данных содержат 9 байтов (8 знаков), такие данные отправляются побитно.

Таблица 1 – Сопоставительная таблица данных с интерфейсами

№	DO	DI	Bit0, Bit1, Bit2, Bit3, Bit4, Bit5, Bit6, Bit7
0	0X01	idle	
1	0x42	ID	
2	idle	0x5A	
3	WW	data	SELECT, L3, R3, START, UP, RIGHT, DOWN, LEFT
4	YY	data	L2, R2, L1, R1, Δ , \circ , \times , \square
5	idle	data	PSS_RX (0x00=left, 0xFF=right)
6	idle	data	PSS_RY (0x00=up, 0xFF=down)
7	idle	data	PSS_LX (0x00=left, 0xFF=right)
8	idle	data	PSS_LY (0x00=up, 0xFF=down)

При нажатии клавиш передаются комбинации из «0» и «1». Например, при нажатии кнопки SELECT передается массив Data[3]=1111110B.

После установки режима вибрации мы можем послать WW, YY, чтобы управлять вибрацией устройства. WW используется для управления мелкими вибрациями с правой стороны устройства, 0x00 закрыты, остальные значения открыты. YY используется для управления сильными вибрациями с левой стороны устройства, 0x40-0xFF включают устройство, чем выше аналоговые значения, тем сильнее вибрация. Конкретные настройки представлены ниже, в главе о составлении скриптов.

Режим красного индикатора: движения рычажка вправо-влево посылают аналоговый сигнал, диапазон 0x00-0xFF, также движения рычажка аналогичны значениям клавиш L3, R3.

Режим зеленого индикатора: движения рычажка вправо-влево неактивны, когда рычажок находится в крайних положениях, он посылает соответствующие кнопкам команды UP, RIGHT, DOWN, LEFT, Δ , \circ , \times , \square . Кнопки L3, R3 неактивны.

2 Указания к использованию и подключению джойстика

Необходимо вставить в джойстик две батарейки номиналом 1.5 В, приемник и главное устройство питаются от одного источника, диапазон напряжения 3-5 В. Неправильное подключение, работа при повышенных и пониженных напряжениях могут привести к поломке приемника.

На джойстике есть кнопка включения, ON – включено, OFF – выключено. Включите джойстик, если джойстик не обнаружил приемник, красный индикатор будет мерцать. Если в установленное время джойстик не свяжется с приемником, он перейдет в ждущий режим, красный индикатор погаснет. Чтобы выйти из ждущего режима, нажмите кнопку START.

Если на приемник поступает питание, но устройства не связываются, будет мигать зеленый индикатор.

Включите джойстик, подайте питание на приемник, джойстик и приемник должны связаться. Когда загорится красный индикатор, это означает, что связь установлена. Нажмите кнопку MODE (на некоторых джойстиках эта кнопка может называться ANALOG), выберите «режим красного индикатора» или «режим зеленого индикатора».

Некоторые пользователи пишут, что джойстик и приемник не сопрягаются правильно. В большинстве случаев это связано с неправильным подключением приемника или ошибками в скрипте.

Решение проблемы: подключите приемник непосредственно к источнику питания (питающий кабель должен быть без дефектов и подключен верно), не подсоединяйте никаких кабелей для передачи данных и тактового сигнала, при правильном подключении джойстик сразу же свяжется с приемником, загорится красный индикатор. Еще раз проверьте все соединения и скрипт на наличие ошибок.

3 Подключение к компьютеру

Способы соединения приемника и stm32:

DI->PB12;
DO->PB13;
CS->PB14;
CLK->PB15.

4 Составление скриптов

Полные скрипты программ находятся в файле pstwo.c, ознакомьтесь с ним подробнее.

```
void PS2_Init(void)
{
    //вход DI->PB12
    RCC->APB2ENR|=1<<3; //тактовый сигнал PORTB
    GPIOB->CRH&=0xFFFF0FFF; // настройки PB12, ставим по умолчанию
    GPIOB->CRH|=0X00080000;
    // DO->PB13      CS->PB14      CLK->PB15
    RCC->APB2ENR|=1<<3; // тактовый сигнал PORTB
    GPIOB->CRH&=0X000FFFFF;
    GPIOB->CRH|=0X33300000; //выставляем выходы PB13, PB14, PB15
}
```

Инициализация интерфейсов: PB12 – вход; PB13, PB14, PB15 – выходы.

```
//отправляем команды на джойстик
void PS2_Cmd(u8 CMD)
{
    volatile u16 ref=0x01;
    Data[1] = 0;
    for(ref=0x01; ref<0x0100; ref<<=1)
    {
        if(ref&CMD)
        {
            DO_H; //вывод контрольного бита
        }
        else DO_L;
        CLK_H; //повышение тактового сигнала
        delay_us(10);
        CLK_L;
        delay_us(10);
        CLK_H;
    }
}
```

```

        if(DI)
            Data[1] = ref>Data[1];
    }
    delay_us(16);
}
//оцениваем, включен ли режим красного индикатора, 0x41=имитация зеленого
индикатора, 0x73=имитация красного индикатора
//ответное значение: 0, режим красного индикатора
//
//        другое, другие режимы

u8 PS2_RedLight(void)
{
    CS_L;
    PS2_Cmd(Comd[0]); //начало команды
    PS2_Cmd(Comd[1]); //запрос данных
    CS_H;
    if( Data[1] == 0X73)        return 0;
    else return 1;
}
//считывание данных с джойстика
void PS2_ReadData(void)
{
    volatile u8 byte=0;
    volatile u16 ref=0x01;
    CS_L;
    PS2_Cmd(Comd[0]);        //начало команды
    PS2_Cmd(Comd[1]);        //запрос данных
    for(byte=2;byte<9;byte++) //начало приема данных
    {
        for(ref=0x01; ref<0x100; ref<<=1)
        {
            CLK_H;
            delay_us(10);
            CLK_L;
            delay_us(10);
            CLK_H;
            if(DI)
                Data[byte] = ref>Data[byte];
        }
        delay_us(16);
    }
    CS_H;
}
}

```

Две функции, описанные выше, предназначены для отправки данных с устройства на джойстик и с джойстика на устройство.

Данные, отправляемые с джойстика на устройство, сохраняются в кэше в виде массива Data[], массив состоит из 9-ти элементов, значения элементов показаны в таблице 1. Еще одна функция определяет режим отправки данных с джойстика, а также устанавливает ID, то есть значение Data[1].

```

//управляем данными, считанными с PS2, только для управления кнопками
// кнопка нажата – 0, не нажата – 1
u8 PS2_DataKey()
{
    u8 index;
    PS2_ClearData();
    PS2_ReadData();
    Handkey=(Data[4]<<8)|Data[3];    //16 кнопок, нажата – 0, не нажата – 1
    for(index=0;index<16;index++)
    {
        if(Handkey&(1<<(MASK[index]-1)))==0)
            return index+1;
    }
    return 0;    //кнопки не нажаты
}
//получаем аналоговое значение для рычажка, диапазон 0-256
u8 PS2_AnalogData(u8 button)
{
    return Data[button];
}
//очищаем область данных
void PS2_ClearData()
{
    u8 a;
    for(a=0; a<9; a++)
        Data[a]=0x00;
}

```

Восьмибитные данные Data[3] и Data[4] соответствуют положениям 16-ти кнопок, нажата – 0, не нажата – 1. Для управления этими данными получаем информацию о положении кнопок и формируем ответное значение кнопки.

Другая возможность этой функции – ответное аналоговое значение, только в «режиме красного индикатора» действительны нижние значения. Значения изменяются только при движении рычажка, все эти значения по отдельности хранятся в массивах Data[5], Data[6], Data[7], Data[8].

Инициализация конфигурации джойстика:

```

//short poll
void PS2_ShortPoll(void)
{
    CS_L;
    delay_us(16);
    PS2_Cmd(0x01);
    PS2_Cmd(0x42);
    PS2_Cmd(0X00);
    PS2_Cmd(0x00);
    PS2_Cmd(0x00);
    CS_H;
    delay_us(16);
}
//входим в конфигурацию
void PS2_EnterConfing(void)

```



```

{
    CS_L;
    delay_us(16);
    PS2_Cmd(0x01);
    PS2_Cmd(0x43);
    PS2_Cmd(0X00);
    PS2_Cmd(0x01);
    PS2_Cmd(0x00);
    PS2_Cmd(0X00);
    PS2_Cmd(0X00);
    PS2_Cmd(0X00);
    PS2_Cmd(0X00);
    CS_H;
    delay_us(16);
}
//отправляем настройки режима
void PS2_TurnOnAnalogMode(void)
{
    CS_L;
    PS2_Cmd(0x01);
    PS2_Cmd(0x44);
    PS2_Cmd(0X00);
    PS2_Cmd(0x01); //analog=0x01; digital=0x00 настройки ПО для режима
    PS2_Cmd(0xEE); //0x03 блокирует настройки, нельзя задать настройки при
                    //помощи кнопки MODE
                    //0xEE не блокирует сохраненные настройки ПО, можно задать
                    //настройки при помощи кнопки MODE

    PS2_Cmd(0X00);
    PS2_Cmd(0X00);
    PS2_Cmd(0X00);
    PS2_Cmd(0X00);
    CS_H;
    delay_us(16);
}
//Настройки вибрации
void PS2_VibrationMode(void)
{
    CS_L;
    delay_us(16);
    PS2_Cmd(0x01);
    PS2_Cmd(0x4D);
    PS2_Cmd(0X00);
    PS2_Cmd(0x00);
    PS2_Cmd(0X01);
    CS_H;
    delay_us(16);
}
//завершаем и сохраняем настройки конфигурации
void PS2_ExitConfig(void)

```

```

{
    CS_L;
    delay_us(16);
    PS2_Cmd(0x01);
    PS2_Cmd(0x43);
    PS2_Cmd(0X00);
    PS2_Cmd(0x00);
    PS2_Cmd(0X5A);
    PS2_Cmd(0X5A);
    PS2_Cmd(0X5A);
    PS2_Cmd(0X5A);
    PS2_Cmd(0X5A);
    CS_H;
    delay_us(16);
}
//Инициализация конфигурации джойстика
void PS2_SetInit(void)
{
    PS2_ShortPoll();
    PS2_ShortPoll();
    PS2_ShortPoll();
    PS2_EnterConfig(); // входит в режим настройки конфигурации
    PS2_TurnAnalogModeOn; // конфигурация режимов «красного» или «зеленого
                           // индикатора», выбор и сохранение
    PS2_VibrationMode(); //включение режима вибрации
    PS2_ExitConfig(); // завершение и сохранение конфигурации
}

```

Вы можете увидеть, какие действия выполняет джойстик после отправки на него команд.

Для надежного соединения инициализация конфигурации джойстика PS2_ShortPoll() выполняется 3 раза. Также предоставлены комментарии к конкретным конфигурациям.

```

/*****
Function:    void PS2_Vibration(u8 motor1, u8 motor2)
Description: функция вибрации джойстика
Calls:      void PS2_Cmd(u8 CMD);
Input:      motor1: слабые вибрации в левой части устройства, 0x00 закрыт, другие открыты
            motor 1: сильные вибрации в правой части устройства, 0x40-0xFF открыты, чем
            выше значение, чем сильнее вибрация
            *****/
void PS2_Vibration(u8 motor1, u8 motor2)

```

```

{
    CS_L;
    delay_us(16);
    PS2_Cmd(0x01); // начало команды
    PS2_Cmd(0x42); //запрос данных
    PS2_Cmd(0X00);
    PS2_Cmd(motor1);
    PS2_Cmd(motor2);
    PS2_Cmd(0X00);
}

```

```

PS2_Cmd(0X00);
PS2_Cmd(0X00);
PS2_Cmd(0X00);
CS_H;
delay_us(16);
}

```

Только при инициализации функции void PS2_SetInit(void) производится инициализация вибрации устройства (PS2_VibrationMode();// запуск режима вибрации), эта строка функции может повторяться.

Главная функция скрипта:

```

int main(void)
{
    u8 key;
    Stm32_Clock_Init(9);//настройки тактового сигнала системы
    delay_init(72); // инициализация задержки
    uart_init(72, 9600); //инициализация последовательного интерфейса 1
    LED_Init();
    PS2_Init(); //запуск инициализации интерфейса
    PS2_SetInit(); // инициализация конфигурации, «красный индикатор», можно
                    // выбрать другой режим или изменить настройки
                    // запуск режима вибрации
    while(1)
    {
        LED=!LED;
        key=PS2_DataKey();
        if(key!=0) //кнопка действительна
        {
            print(" \r\n  %d  id  pressed  \r\n", key);
            if(key == 11)
            {
                PS2_Vibration(0xFF,0x00); // задержка после вибрации
                delay_ms(1000);
                delay_ms(500);
            }
            else if(key == 12)
            {
                PS2_Vibration(0x00,0x41); // задержка после вибрации
                delay_ms(1000);
                delay_ms(500);
            }
            else
                PS2_Vibration(0x00, 0x00);
        }
        printf("%5d%5d%5d%5d\r\n",PS2_AnalogData(PSS_LX),PS2_AnalogData(PSS_LY),
        PS2_AnalogData(PSS_RX), PS2_AnalogData(PSS_RY));
        delay_ms(500);
    }
}

```

Когда кнопка нажата, с нее поступает значение.

5 Загрузка и проверка

Скомпилируйте скрипт и загрузите его. Если рычажок установлен в режим красного индикатора, индикатор будет гореть красным, выходное аналоговое значение с последовательного интерфейса 127 или 128. При движении рычажка соответствующее аналоговое значение изменится, в этот момент положение рычажка соответствует нажатой кнопке, формируется сигнал с кнопок, как показано на рисунке 4.



Рисунок 3



Рисунок 4

Нажмите клавишу Δ , выходное значение будет «13».



Рисунок 5

Нажмите по отдельности L1 и R1, на последовательном интерфейсе появится значение кнопки и соответствующая вибрация джойстика.



Рисунок 6

Нажмите **MODE**, переключитесь в режим зеленого индикатора, цвет индикатора на джойстике станет зеленым, аналоговое значение последовательного интерфейса будет 255, легонько наклоните рычажок – значение не изменится.



Рисунок 7

Наклоните рычажок в крайнее правое положение, на последовательном интерфейсе высветится «Mispressed», что соответствует значению кнопки \triangle , но аналоговое значение не меняется.



Рисунок 8

Главное отличие «режима красного индикатора» и «режима зеленого индикатора» в изменении аналогового значения при движении рычажка.

Попробуйте изменить настройки при инициализации: будут ли заблокированы настройки при выборе режима? Проверьте, есть ли какие-то изменения в работе джойстика. Установите значения PS2_Vibration(0x00, 0x41) и почувствуйте, как джойстик вибрирует с разной частотой.