

Lesson 6 PS2 Handle Control

1. Project Purpose

Learn the principle of PS2 handle control and realize the data communication of PS2 handle.

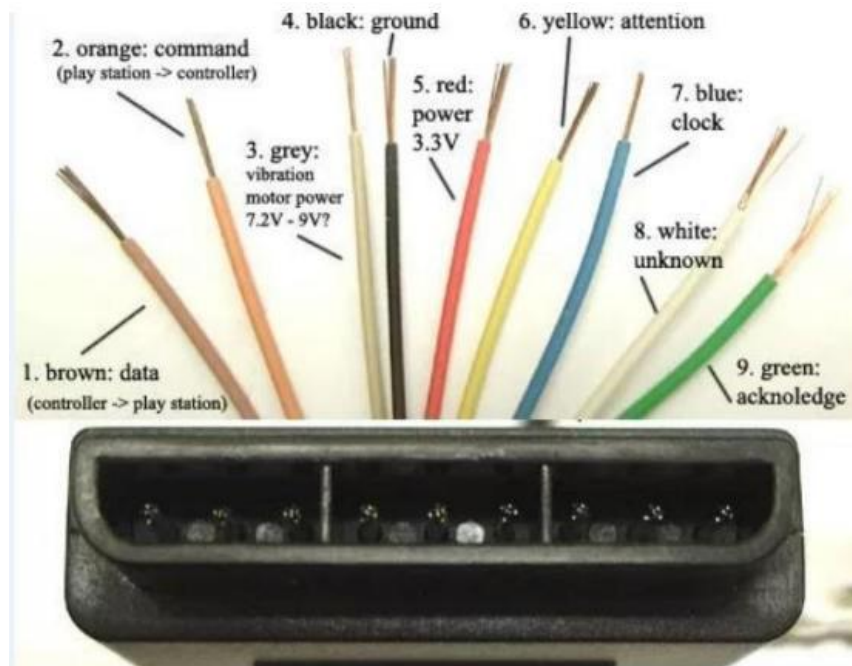
2. Project Principle

The human-computer interface is very important in control system. Handle is easy and convenient to operation and suitable for robot control. In this section, we choose a common PS handle as the control device.

The PS handle requires only four signal wires to communicate with the microcontroller. The communication method between the handle and microcontroller is serial mode, which occupies fewer I/O ports and the communication protocol is simpler. Therefore, it is very suitable for development. The following is the pin definition diagram of the PS handle receiver.

Pin	Definition	Application
1	DATA	The serial data line from the handle to the host, this signal is an 8-bit serial data, synchronously transmitted on the falling edge of the clock (input and output signals change from high to low in the clock signal, and all signals are read from the front edge of the clock to the level Done before the change.)
2	COMMAND	The serial data line from the host to the handle works in the same way as the DATA signal.
3	NC	No use
4	GND	Power ground and signal ground
5	3.3v	Power votage. The effective working voltage is 3V-5V.
6	Attention	Used to provide a handle trigger signal and the signal is at a low level during communication.

		Equivalent to chip select signal
7	CLOCK	Signal direction: from the host to the handle. Used to keep data in sync
8	NC	No use
9	acknolege	The response signal from the handle to the host. This signal becomes low in the last clock cycle after each 8Bit data is sent, and remains low. If the ACK signal does not go low for about 60us, the PS host will try another handle.



3. Program Analyst

1) Firstly, initialize PS2 handle in InitPS2 function of setup function.

```

14 PS2X ps2X; //Instantiate the handle class
15 void InitPS2()
16 {
17     ps2X.config_gamepad(A2, A4, A3, A5); //Set PS2 port, No.A2 IO port to clock, No.A4 IO to command, No.A3 IO to attention, No.A5 to data.
18 }
19

```

2) Then analyze the ps2Handle function.

```

181 void ps2Handle() { //PS2 handle handling
182     static uint32_t Timer; //Define static variable Timer for timing
183     if (Timer > millis()) //Return when Timer is larger than millis() (The total running milliseconds). Continue to run the following statement when Timer is less than the total running milliseconds.
184         return;
185     ps2X.read_gamepad(); //Read the data of the button of PS2 handle
186
187     if ( Mode == 0 )
188     {
189         if ( ps2X.Button( PSB_SELECT ) )
190         {
191             if ( ps2X.ButtonPressed( PSB_START ) )
192             {
193                 Mode = 1;
194                 manual = TRUE;
195                 FullActRun(0, 1);
196                 LedFlip();
197                 for (int i = 0 ; i < 8 ; i++)
198                 {
199                     ServoPwmDutyset[i] = 1500;
200                 }
201                 ServoSetPluseAndTime( 1, 1500, 1000 );
202                 ServoSetPluseAndTime( 2, 1500, 1000 );
203                 ServoSetPluseAndTime( 3, 1500, 1000 );
204                 ServoSetPluseAndTime( 4, 1500, 1000 );
205                 ServoSetPluseAndTime( 5, 1500, 1000 );
206                 ServoSetPluseAndTime( 6, 1500, 1000 );
207                 return;
208             }
209         }
210     }
211     else {
212         if (ps2X.ButtonPressed(PSB_START)) { //If the "up" button on the left side is pressed
213             LedFlip();
214             FullActRun(0, 1);
215             Timer = millis() + 50; //Timer adds 50ms to the total running milliseconds. It will run again after 50ms.
216             return; //Return, exit this function
217         }
218     }
219 }

```

- 3) After entering ps2Handle function, delay a period of time first before reading the data. After confirming that the data is read completely, the corresponding operation is executed according to the value of the key. LedFlip is a blinking LED light to indicate the completion of the key.
- 4) FullActRun is the action group running function and will be mentioned in the following chapter. It actually calls the downloaded action group in the controller, and then exit delay.