Lesson 8 Serial Port Controls Servo

1. Project Purpose

Learn the serial communication and use it to control serial bus servo to execute the corresponding commands

2. Project Principle

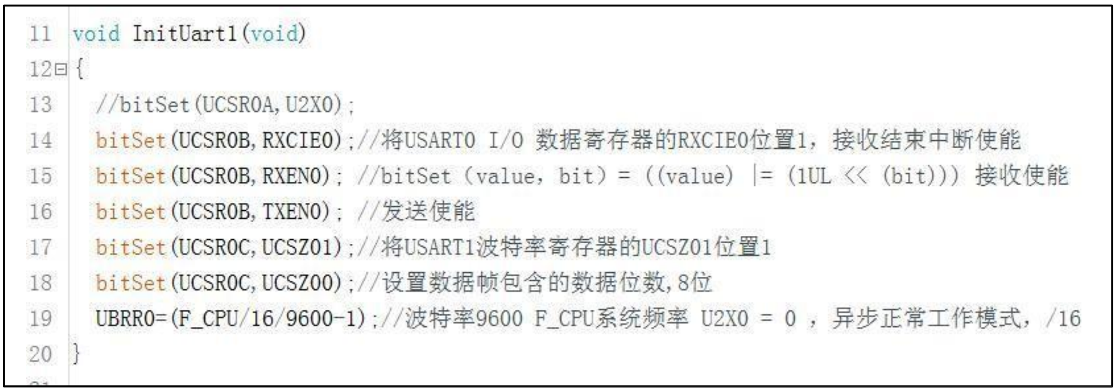
Serial communication interface that is shorts for the serial port refers to a serial communication port sending data one bit at a time. In the MCU and embedded environment, the serial port generally refers to the UART port.

According to the level standard of the interface, it can be divided into RS-232、RS-422、RS485、TTL, etc. TTL serial port is usually not converted by the specialized chip after we derive from the MCU chip.

Two types of physical ports are provided: DB9 connector and 4-pin header.

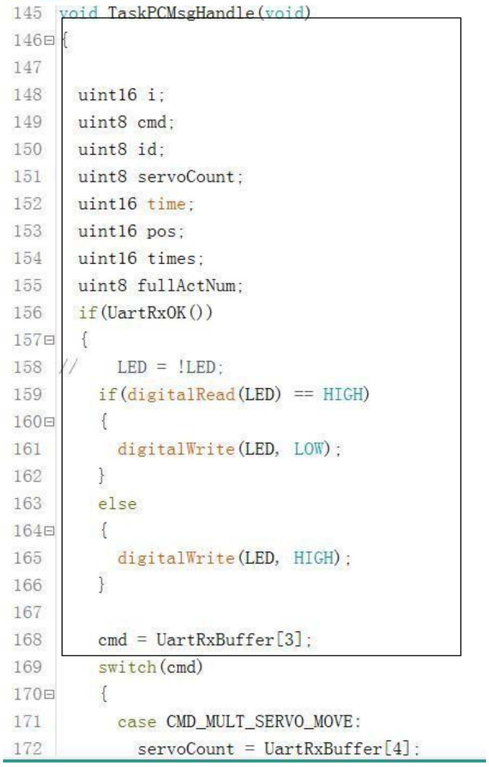
3. Program Analyst

1) Firstly, we need to initial the serial port. InitUart1 function is executed in Setup section.



```
void InitUart1(void)
{
  //bitSet(UCSR0A,U2X0);
  bitSet(UCSR0B,RXCIE0);//将USART0 I/O 数据寄存器的RXCIE0位置1，接收结束中断使能
  bitSet(UCSR0B,RXEN0); //bitSet (value, bit) = ((value) |= (1UL << (bit))) 接收使能
  bitSet(UCSR0B,TXEN0); //发送使能
  bitSet(UCSR0C,UCSZ01);//将USART1波特率寄存器的UCSZ01位置1
  bitSet(UCSR0C,UCSZ00);//设置数据帧包含的数据位数,8位
  UBRR0=(F_CPU/16/9600-1);//波特率9600 F_CPU系统频率 U2X0 = 0 ，异步正常工作模式，/16
}
```

2) Trough a series of operation on register, InitUart1 sets it to the transmittable data and its baud rate is 9600. Then, let’s look at the TaskPCMsgHandle function called in the loop.

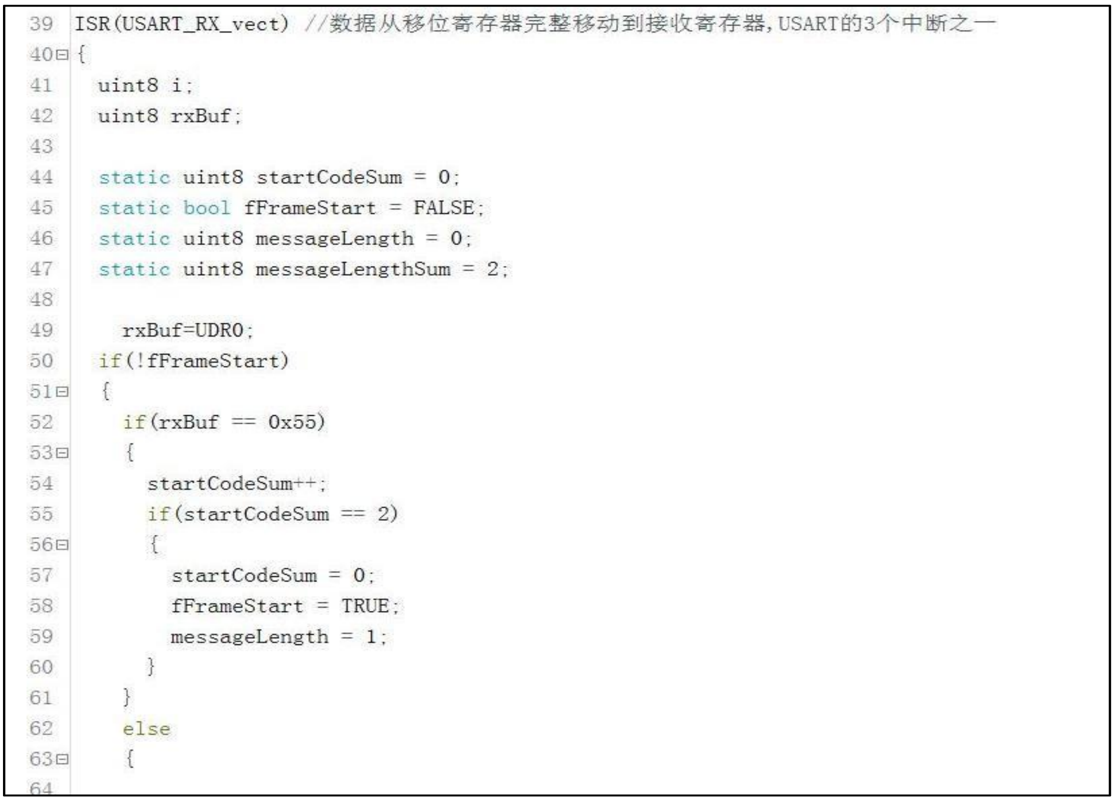


```
void TaskPCMsgHandle(void)
{

  uint16 i;
  uint8 cmd;
  uint8 id;
  uint8 servoCount;
  uint16 time;
  uint16 pos;
  uint16 times;
  uint8 fullActNum;
  if(UartRxOK())
  {
//    LED = !LED;
    if(digitalRead(LED) == HIGH)
    {
      digitalWrite(LED, LOW);
    }
    else
    {
      digitalWrite(LED, HIGH);
    }

    cmd = UartRxBuffer[3];
    switch(cmd)
    {
      case CMD_MULT_SERVO_MOVE:
        servoCount = UartRxBuffer[4];
```

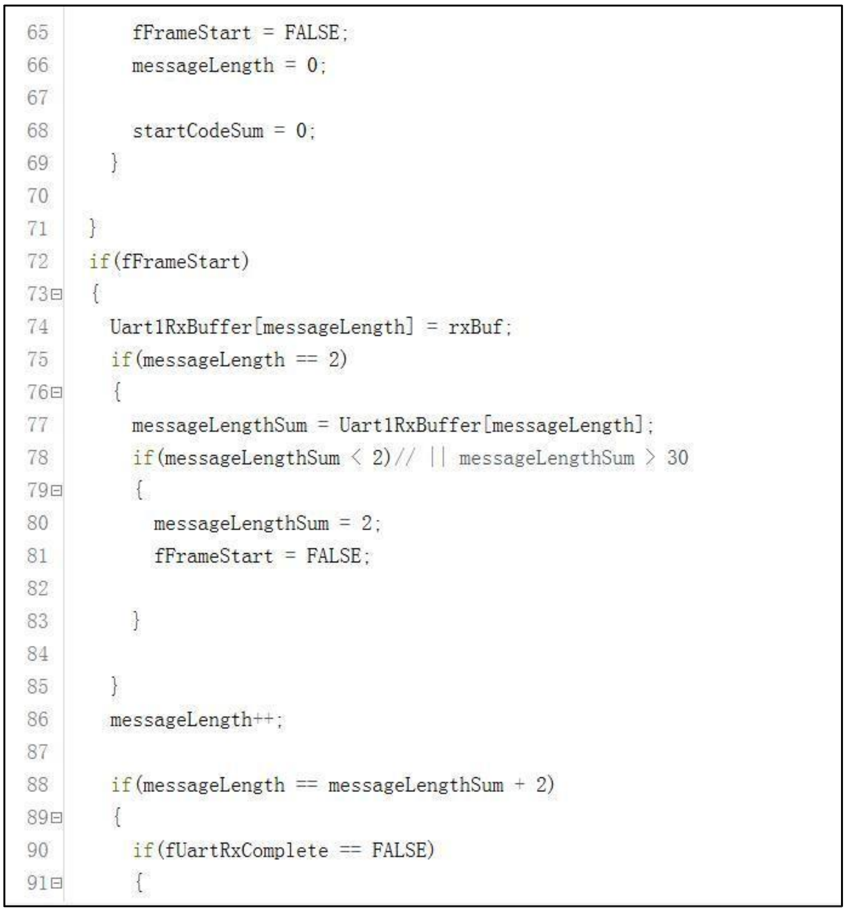
3) Then, judge the returned value of UartRxOK function. UartRxOK function is pulled upward first and then UartRxComplete value is judged.



```
ISR(USART_RX_vect) //数据从移位寄存器完整移动到接收寄存器,USART的3个中断之一
{
  uint8 i;
  uint8 rxBuf;

  static uint8 startCodeSum = 0;
  static bool fFrameStart = FALSE;
  static uint8 messageLength = 0;
  static uint8 messageLengthSum = 2;

   rxBuf=UDR0;
  if(!fFrameStart)
  {
    if(rxBuf == 0x55)
    {
      startCodeSum++;
      if(startCodeSum == 2)
      {
        startCodeSum = 0;
        fFrameStart = TRUE;
        messageLength = 1;
      }
    }
    else
    {
```



```
    fFrameStart = FALSE;
    messageLength = 0;

    startCodeSum = 0;
  }

}
if(fFrameStart)
{
  Uart1RxBuffer[messageLength] = rxBuf;
  if(messageLength == 2)
  {
    messageLengthSum = Uart1RxBuffer[messageLength];
    if(messageLengthSum < 2)// || messageLengthSum > 30
    {
      messageLengthSum = 2;
      fFrameStart = FALSE;

    }

  }
  messageLength++;

  if(messageLength == messageLengthSum + 2)
  {
    if(fUartRxComplete == FALSE)
    {
```

4) Suppose that the data frame sent to this program is "0x55 0x55 0x05 0x03 0x01 0xD0 0X07", this message refers to controlling the No. 1 servo to rotate to the 2000 position within 1000ms.

5) Suppose that the data frame sent to this program is "0x55 0x55 0x05 0x03 0x01 0xD0 0X07", this message refers to controlling the No. 1 servo to rotate to the 2000 position within 1000ms.

6) The received data is saved in rxBuf. If the data frame 0x55 is received , then startCodeSum+1.

7) When receiving two 0x55, startCodeSum will be cleared to 0. If the fFrameStart is set to true, the next frame of data will be saved. If no data is received, then judge again.

8) If fFrameStart is true, 0x55 will be saved to the second position in Uart1RxBuffer after receiving two 0x55 (with subscript 1).

9) After all the data has been received according to the data format, then we go back to the TaskPCMsgHandle function and UartRxOK will return “True”.

10) We classify the received data according to the command value, and then convert the other data values into the rotation time and position of the servo ID. Finally, realize the servo rotation through the ServoSetPluseAndTime function .