GPS parsing output

1. Learning objectives

In this course, we mainly learn to use STM32F103RCT6 and GPS module modules to realize the location information analysis output function.

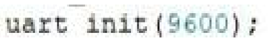
2. Prepare before class

The GPS module uses UART and USB communication, here use the UART port of STM32F103RCT6 to read the information, and connect the TXD of the module to the PA10 pin of the STM32F103RCT6 board.

VCC and GND are connected to the 5V and GND of the STM32F103RCT6, respectively.

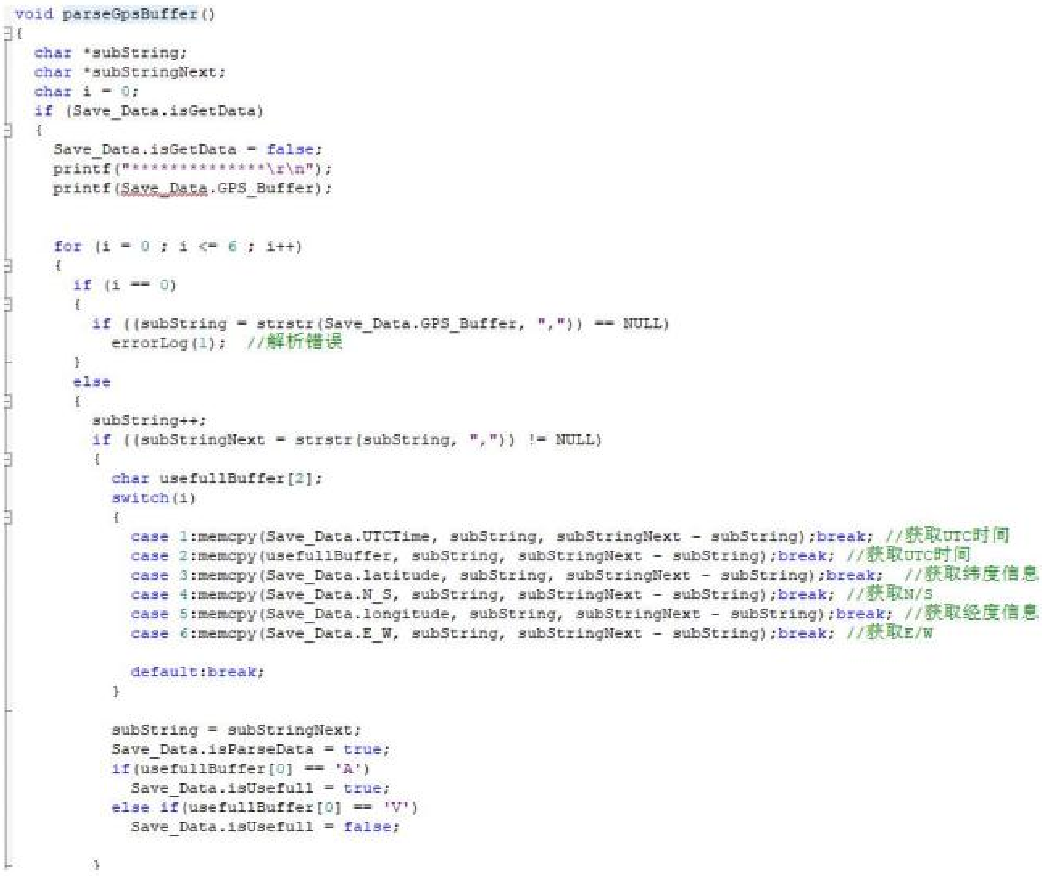
3. Programming

The baud rate of the module is 9600



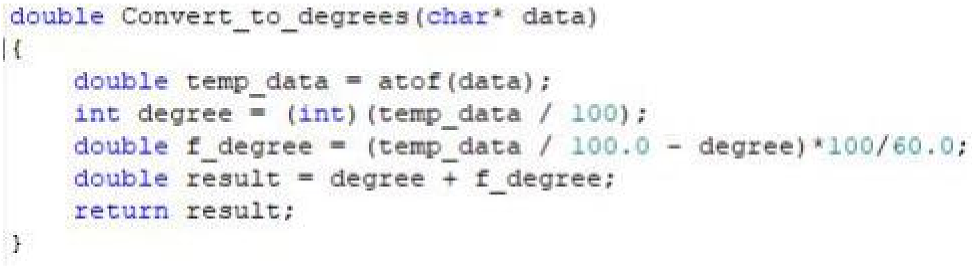
```
uart init(9600);
```

Read and parse the received data



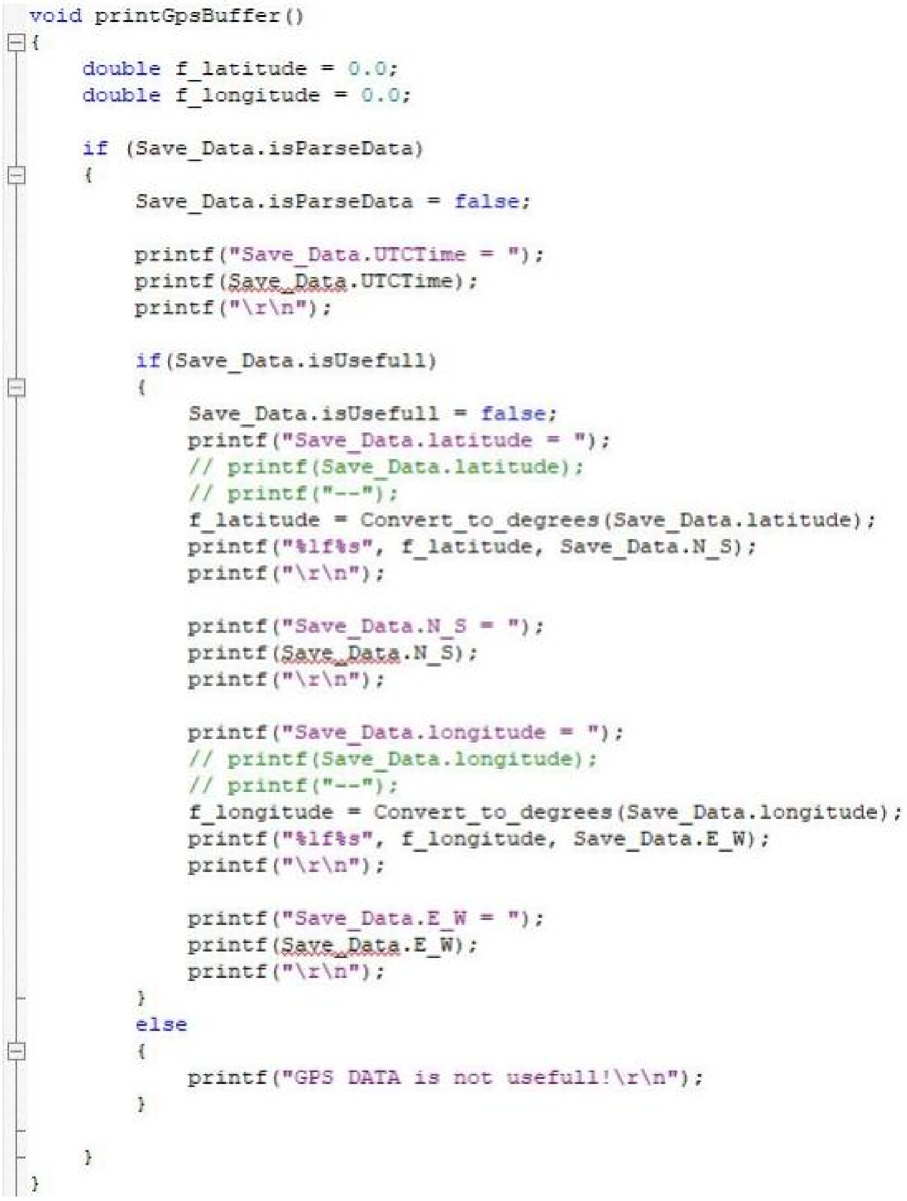
```
void parseGpsBuffer()
{
  char *subString;
  char *subStringNext;
  char i = 0;
  if (Save_Data.isGetData)
  {
    Save_Data.isGetData = false;
    printf("**************\r\n");
    printf(Save_Data.GPS_Buffer);


    for (i = 0 ; i <= 6 ; i++)
    {
      if (i == 0)
      {
        if ((subString = strstr(Save_Data.GPS_Buffer, ",")) == NULL)
          errorLog(1);  //解析错误
      }
      else
      {
        subString++;
        if ((subStringNext = strstr(subString, ",")) != NULL)
        {
          char usefullBuffer[2];
          switch(i)
          {
            case 1:memcpy(Save_Data.UTCTime, subString, subStringNext - subString);break; //获取UTC时间
            case 2:memcpy(usefullBuffer, subString, subStringNext - subString);break; //获取UTC时间
            case 3:memcpy(Save_Data.latitude, subString, subStringNext - subString);break;  //获取纬度信息
            case 4:memcpy(Save_Data.N_S, subString, subStringNext - subString);break; //获取N/S
            case 5:memcpy(Save_Data.longitude, subString, subStringNext - subString);break; //获取经度信息
            case 6:memcpy(Save_Data.E_W, subString, subStringNext - subString);break; //获取E/W

            default:break;
          }

          subString = subStringNext;
          Save_Data.isParseData = true;
          if(usefullBuffer[0] == 'A')
            Save_Data.isUsefull = true;
          else if(usefullBuffer[0] == 'V')
            Save_Data.isUsefull = false;

        }
```

Converts units of latitude and longitude information into degrees.



```
double Convert_to_degrees(char* data)
{
    double temp_data = atof(data);
    int degree = (int)(temp_data / 100);
    double f_degree = (temp_data / 100.0 - degree)*100/60.0;
    double result = degree + f_degree;
    return result;
}
```

Print the received data through the serial port



```
void printGpsBuffer()
{
    double f_latitude = 0.0;
    double f_longitude = 0.0;

    if (Save_Data.isParseData)
    {
        Save_Data.isParseData = false;

        printf("Save_Data.UTCTime = ");
        printf(Save_Data.UTCTime);
        printf("\r\n");

        if(Save_Data.isUsefull)
        {
            Save_Data.isUsefull = false;
            printf("Save_Data.latitude = ");
            // printf(Save_Data.latitude);
            // printf("--");
            f_latitude = Convert_to_degrees(Save_Data.latitude);
            printf("%lf%s", f_latitude, Save_Data.N_S);
            printf("\r\n");

            printf("Save_Data.N_S = ");
            printf(Save_Data.N_S);
            printf("\r\n");

            printf("Save_Data.longitude = ");
            // printf(Save_Data.longitude);
            // printf("--");
            f_longitude = Convert_to_degrees(Save_Data.longitude);
            printf("%lf%s", f_longitude, Save_Data.E_W);
            printf("\r\n");

            printf("Save_Data.E_W = ");
            printf(Save_Data.E_W);
            printf("\r\n");
        }
        else
        {
            printf("GPS DATA is not usefull!\r\n");
        }

    }
}
```

Note: In fact, the coordinate system value of GPS/Beidou positioning is not a simple 100 times relationship, but needs to be converted once in degrees, minutes and seconds. Then the GPS/Beidou coordinate values we obtain, such as 2429.53531 north latitude and 11810.78036 east longitude, need to be calculated as follows:

24+（29.53531/60）≈ 24.49225517

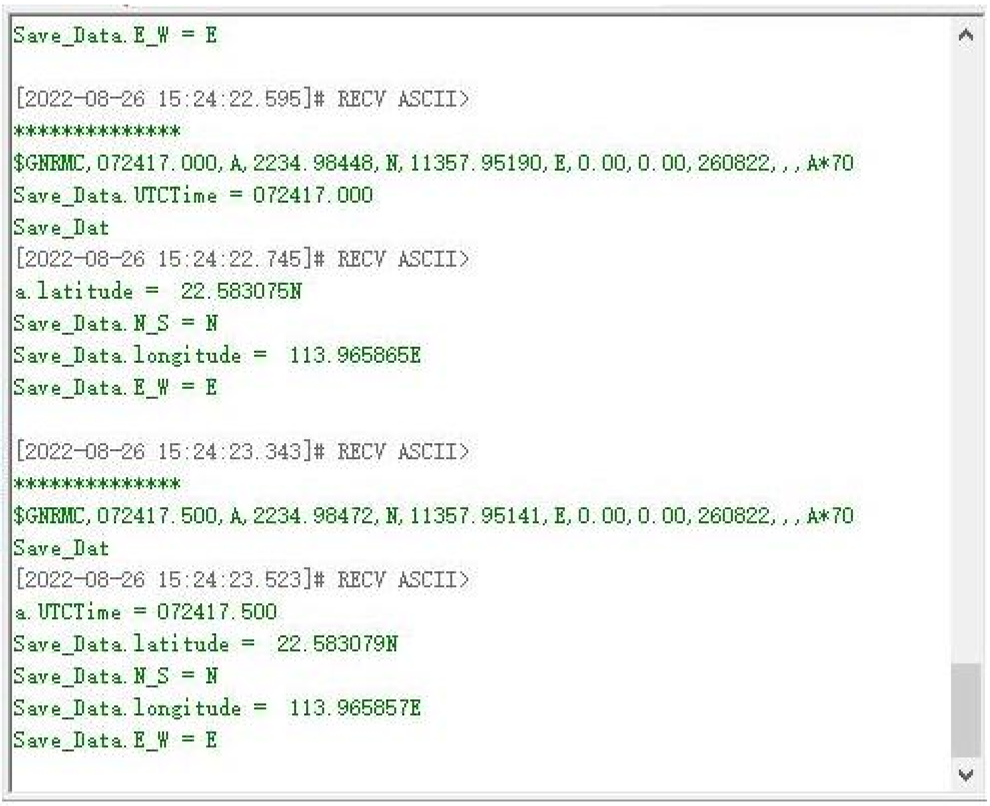
118+（10.78036/60）≈118.17967267

And different single-chip microcomputers may have problems with data conversion accuracy and have certain errors.

4. Experimental phenomenon

After the module is powered on, it takes about 32s to start, and then the serial port printing status light on the module will continue to flash, and the data can be received normally.

After the program is downloaded, run it, open the serial port software, the baud rate is set to 9600, and the serial port will print the current location information in a loop.



```
Save_Data.E_W = E

[2022-08-26 15:24:22.595]# RECV ASCII>
**************
$GNRMC,072417.000,A,2234.98448,N,11357.95190,E,0.00,0.00,260822,,,A*70
Save_Data.UTCTime = 072417.000
Save_Dat
[2022-08-26 15:24:22.745]# RECV ASCII>
a.latitude =  22.583075N
Save_Data.N_S = N
Save_Data.longitude =  113.965865E
Save_Data.E_W = E

[2022-08-26 15:24:23.343]# RECV ASCII>
**************
$GNRMC,072417.500,A,2234.98472,N,11357.95141,E,0.00,0.00,260822,,,A*70
Save_Dat
[2022-08-26 15:24:23.523]# RECV ASCII>
a.UTCTime = 072417.500
Save_Data.latitude =  22.583079N
Save_Data.N_S = N
Save_Data.longitude =  113.965857E
Save_Data.E_W = E
```

Note: The module antenna needs to be outdoors, otherwise the GPS signal may not be searched.