Parse of GPS location information

1. Learning Objectives

In this course, we mainly learn to use arduino and GPS module to realize the function of location information analysis and printing.

2. Preparation before class

The GPS module uses UART and USB communication. Here, the UART port of the Arduino UNO is used to read the information, and the TX of the module is connected to the D0 pin of the Arduino UNO board. VCC and GND are connected to 5V and GND respectively.

3. Procedure

Initialize the serial port.

```
void setup()  //Initialize content
{
  GpsSerial.begin(9600);      //Define baud rate 9600
  DebugSerial.begin(9600);

  Save_Data.isGetData = false;
  Save_Data.isParseData = false;
  Save_Data.isUsefull = false;
}
```

Read serial port data

```
void gpsRead() {
  while (GpsSerial.available())
  {
    gpsRxBuffer[ii++] = GpsSerial.read();
    if (ii == gpsRxBufferLength)clrGpsRxBuffer();
  }

  char* GPS_BufferHead;
  char* GPS_BufferTail;
  if ((GPS_BufferHead = strstr(gpsRxBuffer, "$GPRMC,")) != NULL || (GPS_BufferHead = strstr(gpsRxBuffer, "$GNRMC,")) != NULL )
  {
    if (((GPS_BufferTail = strstr(GPS_BufferHead, "\r\n")) != NULL) && (GPS_BufferTail > GPS_BufferHead))
    {
      memcpy(Save_Data.GPS_Buffer, GPS_BufferHead, GPS_BufferTail - GPS_BufferHead);
      Save_Data.isGetData = true;

      clrGpsRxBuffer();
    }
  }
}
```

Parse serial port data.

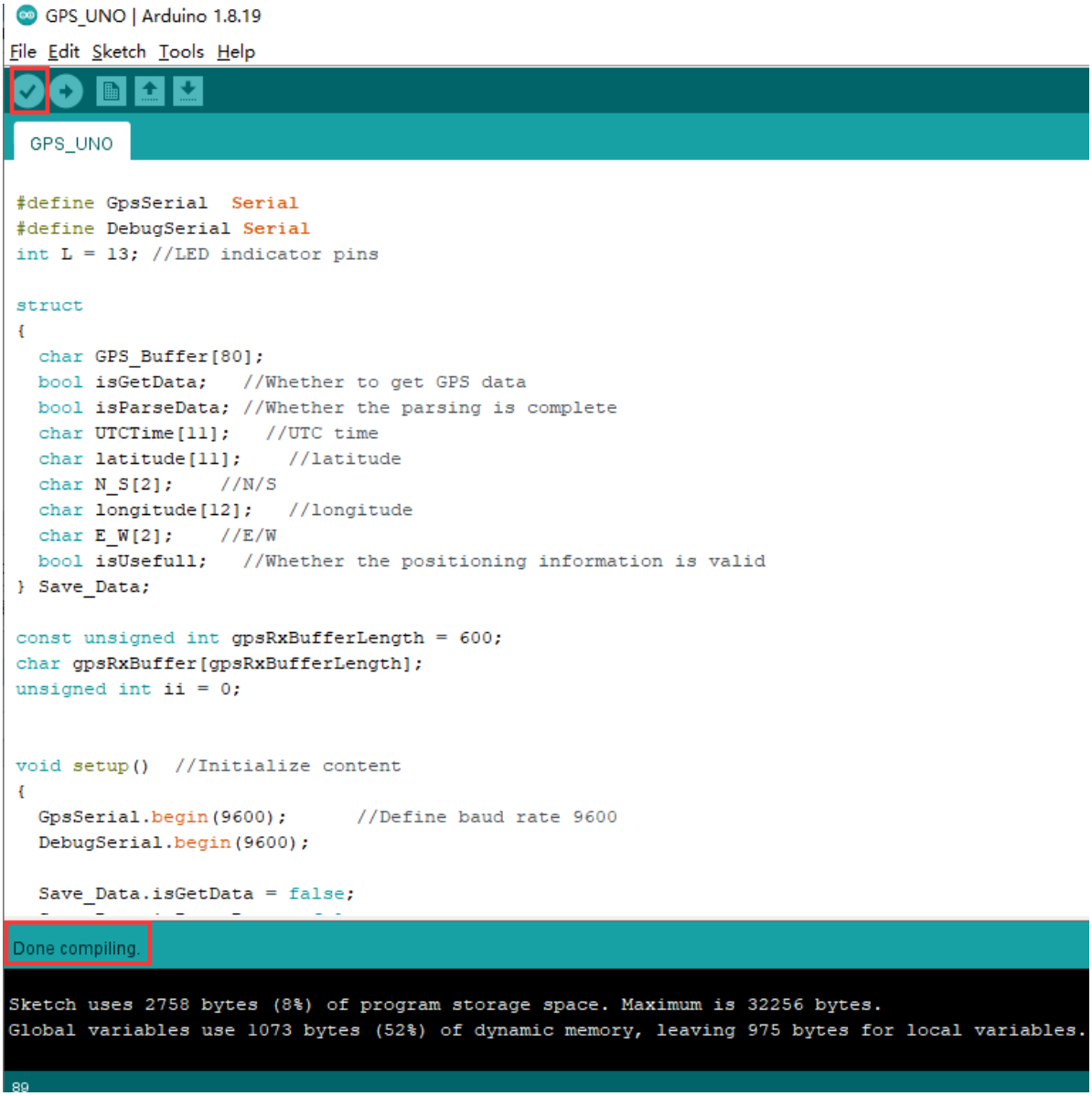
```
void parseGpsBuffer()
{
  char *subString;
  char *subStringNext;
  if (Save_Data.isGetData)
  {
    Save_Data.isGetData = false;
    DebugSerial.println("***************");
    DebugSerial.println(Save_Data.GPS_Buffer);


    for (int i = 0 ; i <= 6 ; i++)
    {
      if (i == 0)
      {
        if ((subString = strstr(Save_Data.GPS_Buffer, ",")) == NULL)
          errorLog(1);  //Parse error
      }
      else
      {
        subString++;
        if ((subStringNext = strstr(subString, ",")) != NULL)
        {
          char usefullBuffer[2];
          switch(i)
          {
            case 1:memcpy(Save_Data.UTCTime, subString, subStringNext - subString);break; //Get UTC time
            case 2:memcpy(usefullBuffer, subString, subStringNext - subString);break; //Get UTC time
            case 3:memcpy(Save_Data.latitude, subString, subStringNext - subString);break;  //Get latitude information
            case 4:memcpy(Save_Data.N_S, subString, subStringNext - subString);break; //Get N/S
            case 5:memcpy(Save_Data.longitude, subString, subStringNext - subString);break; //Get latitude information
            case 6:memcpy(Save_Data.E_W, subString, subStringNext - subString);break; //Get 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;
```

Print the parsed location information.

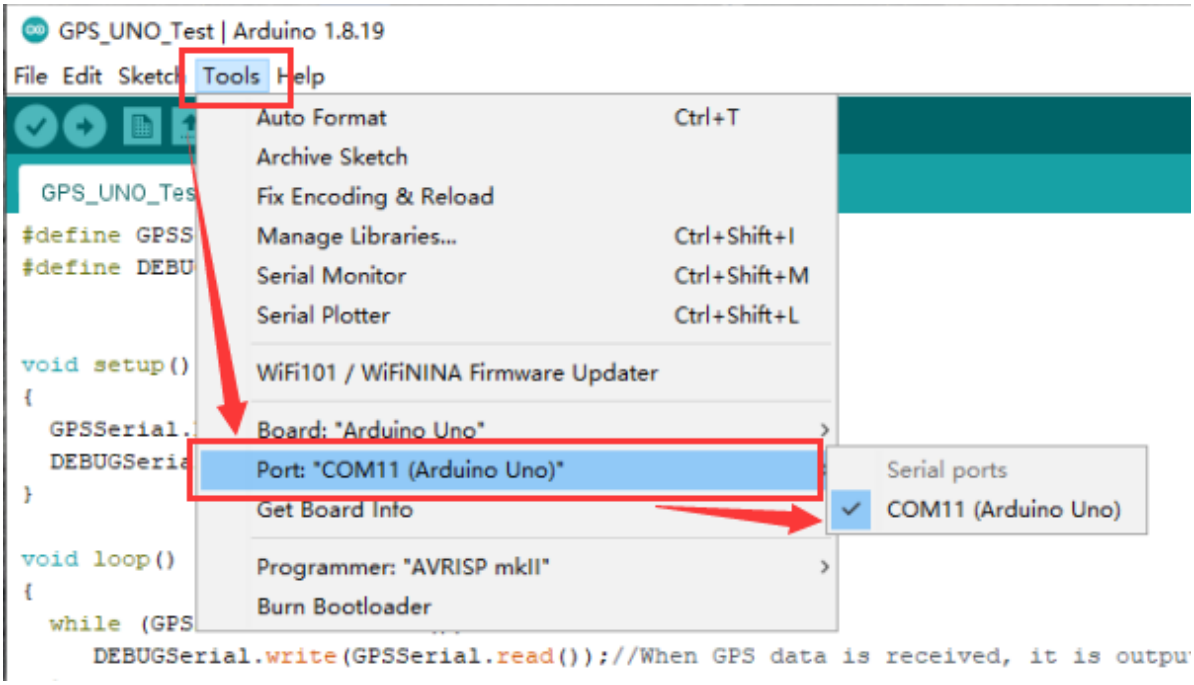
```
void printGpsBuffer()
{
  if (Save_Data.isParseData)
  {
    Save_Data.isParseData = false;

    DebugSerial.print("Save_Data.UTCTime = ");
    DebugSerial.println(Save_Data.UTCTime);

    if(Save_Data.isUsefull)
    {
      Save_Data.isUsefull = false;
      DebugSerial.print("Save_Data.latitude = ");
      DebugSerial.println(Save_Data.latitude);
      DebugSerial.print("Save_Data.N_S = ");
      DebugSerial.println(Save_Data.N_S);
      DebugSerial.print("Save_Data.longitude = ");
      DebugSerial.println(Save_Data.longitude);
      DebugSerial.print("Save_Data.E_W = ");
      DebugSerial.println(Save_Data.E_W);
    }
    else
    {
      DebugSerial.println("GPS DATA is not usefull!");
    }

  }
}
```

4. Compile and download the program

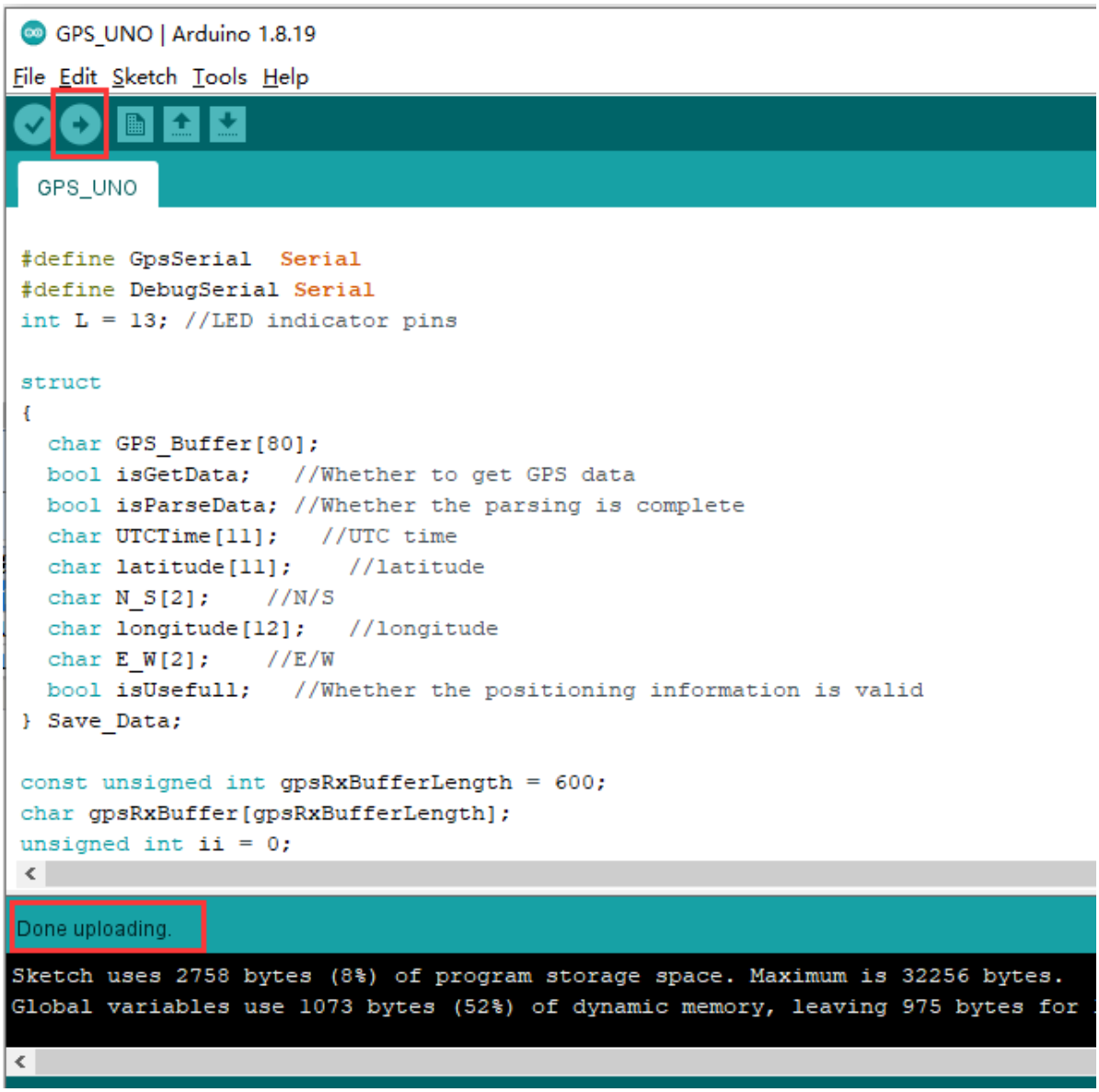
4.1 We need the general Arduino IDE software to open the file, then click "√" in the menu bar to compile the program, and wait for the words "Done compiling" to appear in the lower left corner.



4.2 In the menu bar of Arduino IDE, we need to select [Tools]---[Port]---select the port number just displayed in the device manager, as shown in the figure below.



4.3 After the selection is complete, click "→" under the menu bar to upload the code to the UNO board. When the word "Done uploading" appears in the lower left corner, it means that the program has been successfully uploaded to the UNO board, as shown in the figure below.



5. Experimental phenomenon

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

After the program is downloaded and run, open the serial port monitoring window, open the serial port software, set the baud rate to 9600, and the serial port will cyclically print the parsed real-time location information.

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