OpenCPU Security Data Application Note

GSM/GPRS Module Series

Rev. OpenCPU_Security_Data_Application_Note_V1.0

Date: 2017-07-21

About the Document

History

Revision	Date	Author	Description
1.0	2017-07-21	Chunmao Li	Initial

Contents

Table Index

1 Introduction

This document illustrates OpenCPU Security Data Solution, and shows how to program the critical data in OpenCPU platform.

2 Security User Data

Every wireless device has some critical data that has a direct effect on the stability of module.

Take GPRS modem device for example, the configurations information of APN, server IP address and socket port number is the critical user data for the modem device. If the user data is lost or corrupted, the modem device will not be able to normally work anymore.

Usually, customer may store the user data in two places as below.

Table 1: Place to Store the User Data

Place	Advantage	Disadvantage
Constant variable, in code	Safe	Cannot be reconfigured according to actual case.
User config file, in user file system	Can be configured according to actual case.	A possible risk exists: the config file is lost or corrupted.

For wireless device, the inflexibility of configuration is the necessary consideration. So to use a file to store user data is more popular.

To protect the user data from losing, OpenCPU has designed the Security User Data Solution, and provided a couple of API functions to access security user data. The security data is stored in security data region, and the data in security data region is protected from losing.

OpenCPU has designed 13 blocks of system storage space to store some critical user data. Among the storage blocks, each of 1~8 blocks can store 50 bytes, each of 9~12 blocks can store 100 bytes, and the 13th block can store 500 bytes.

3 API Function

Customer App program may call *Ql_SecureData_Store()* to store the critical user data into security data region. And call *Ql_SecureData_Read()* to read back the critical user data.

3.1. Ql_SecureData_Store

This function stores the critical user data to security data space.

OpenCPU has designed 13 blocks of system storage space to store critical user data. Developer may specify the first parameter index [1-13] to specify different storage block. Among the storage blocks, 1~8 blocks can store 50 bytes for each block, 9~12 blocks can store 100 bytes for each block, and the 13th block can store 500 bytes.

- **Prototype**

```
s32 Ql_SecureData_Store(u8 index , u8* pData, u32 len);
```

- **Parameters**

index:
[in] The index of the user data, range from 1 to 13.

pData:
[in] Pointer to the data need to be stored.

len:
[in] The length of the user data. When the index is (1~8), len should be less than 50; when the index is (9~12), len should be less than 100; when the index is 13, len should be less than 500.

- **Return Value**

QL_RET_OK, this function succeeds.
QL_RET_ERR_PARAM, invalid parameter.

3.2. Ql_SecureData_Read

This function reads secure data which is previously stored by Ql_SecureData_Store.

- **Prototype**

```
s32 Ql_SecureData_Read(u8 index, u8* pBuffer, u32 len);
```

- **Parameters**

index:
[in] The index of the user data. Range from 1 to 13.

pBuffer:
Buffer to store the data to read.

len:
[in] The data length to read. When the index is (1~8), len should be less than 50; when the index is (9~12), len should be less than 100; when the index is 13, len should be less than 500.

- **Return Value**

If this function succeeds, the real read length is returned.
QL_RET_ERR_PARAM, invalid parameter.

3.3. Example

The following codes show how to store user data into security data region, and how to read back the data from security data region.

```
//Suppose that the goal user data is in the structure m_GprsConfig.
ST_GprsConfig m_GprsConfig={
    "CMNET",      //APN name
    "aaa",        //User name for APN
    "bbb",        //Password for APN
    0,
    NULL,
    NULL,
};
```

- Store to security data region

```
u8 idx;
u32 dataLen=sizeof(m_GprsConfig);
if (dataLen<=50)
{
    idx=1..8;
}
else if (dataLen<=100)
{
    idx=9..12;
}
else if (dataLen>100)
{
    idx=13;
}
else
{
    //Developer may choose multi-block to store the data.
    // …
}
ret=Ql_SecureData_Store(idx, (u8*)&m_GprsConfig, dataLen);
if (ret !=QL_RET_OK)
{
    Ql_Debug_Trace("<-- Fail to store critical data! Cause:%d -->\r\n", ret);
}
```

- Read back data from security data region

```
{
    //Retrieve config information from security data region
    u8 idx=9;
    ret=Ql_Userdata_Read(idx,   (u8*)&m_GprsConfig, sizeof(m_GprsConfig));
    Ql_Debug_Trace("<-- Read back data from security data region, ret=%d -->\r\n", ret);

    //Now, the config information is restored into m_GprsCofnig.
    //Program can directly use the data in m_GprsCofnig.
    //…
}
```

4 Appendix

Table 2: Reference Document

SN	Document Name
[1]	OpenCPU_User_Guide