M252 实现Google通讯协议缓冲区

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	本范例程序代码演示如何使用 Nanopb 在 M252 实现通讯协议缓冲区。Nanopb 是一个轻量、支持 C 语言的 Google 通讯协议缓冲区数据格式的实现。
BSP 版本	M251/M252/M254/M256/M258 Series CMSIS BSP V3.01.002
开发平台	NuMaker-M252SD Ver1.2

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 概述

通讯协议缓冲区 Protocol Buffers（简称：ProtoBuf）是一种串行化数据结构的协议，由 Google 所开发。对于通过管道(Pipeline)或储存数据进行通讯的程序开发上是很有用的。这个方法包含一个接口描述语言，描述一些数据结构，并提供程序工具根据这些描述产生程序代码，以产生数据结构或解析资料数据流。

微控制器 M252 提供范例程序代码，示范如何实现通讯协议缓冲区功能。请注意，此范例程序也适用于M251/M254/M256/M258 系列。

范例程序代码使用 Petteri Aimonen 的 Nanopb 开源函数库。Nanopb下载地址 https://jpa.kapsi.fi/nanopb/download/ 。

1.1 开源程序代码 Nanopb 使用

Nanopb 是一个 ANSI-C 函数库，用于以 Google 的通讯协议缓冲区格式对信息进行编码和译码，对内存和程序代码空间的要求最低。它主要适用于 32 位微控制器，但也适用于其他具有严格（<10 kB ROM，<1 kB RAM）内存限制的嵌入式系统。

我们这里使用nanopb-0.4.5-windows-x86。首先，使用 Sample Template 创建一个 Protobuf 项目，然后把nanopb-0.4.5-windows-x86 目录下七个文件 pb.h, pb_common.c, pb_common.h, pb_decode.c, pb_decode.h, pb_encode.c, pb_encode.h 复制到专案文件夹中，如图1-1所示：

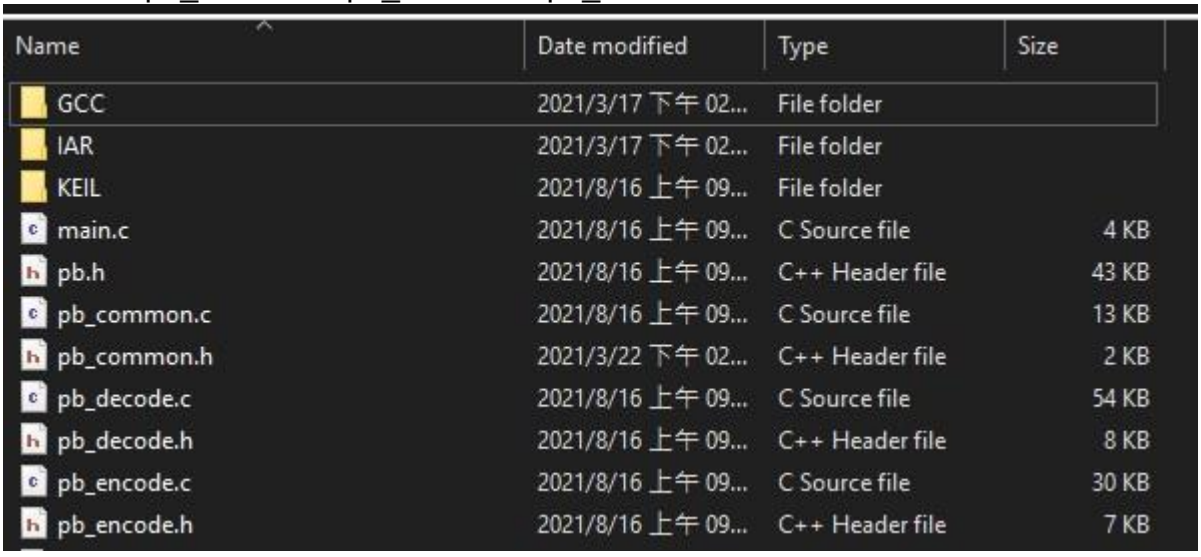

图 1-1 项目文件夹内容

Protobuf 的核心是一个 .proto 文件，我们自定义一个 .proto 来创建我们的协议数据，然后使用 protoc 工具编译产生 C 程序代码，有两个文件：一个头文件、一个原始档。 其中 protoc 工具在我们下载的 nanopb-0.4.5-windows-x86 的 generator-bin 文件夹中，如图1-2 所示：

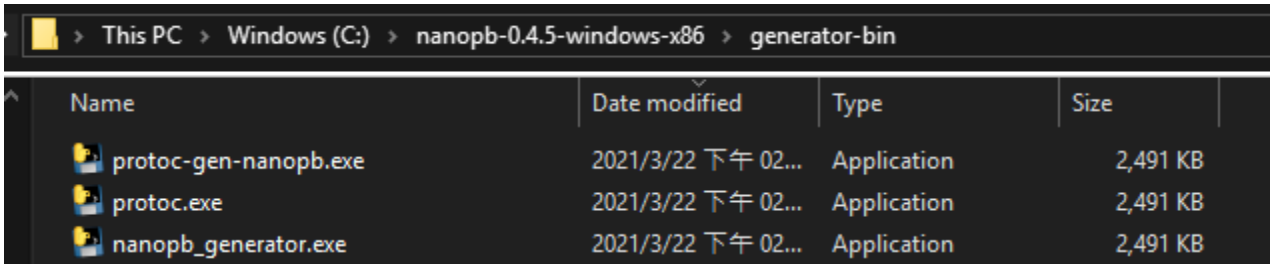

图 1-2 protoc 工具在 generator-bin 文件夹

为了能在命令行中任何路径下使用 protoc 工具，我们需要把 protoc.exe 所在路径添加到环境变量中。然后，我们在 Protobuf 文件夹中创建一个 test.proto 文件。

```
syntax = "proto2";

message Student
{
 required uint32 num      = 1;
 required uint32 py_score = 2;
 required uint32 c_score  = 3;
}
```

进入命令行窗口，在命令行 cd 进入 test.proto 所在目录；输入如下命令编译 test.proto： protoc --nanopb_out=. test.proto 产生 test.pb.c 与 test.pb.h 文件，如图1-3 与图1-4 项目文件夹内容所示：

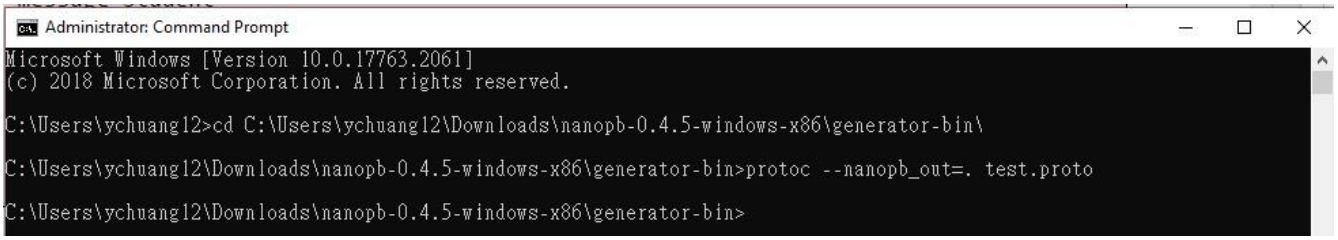

图 1-3 命令行窗口

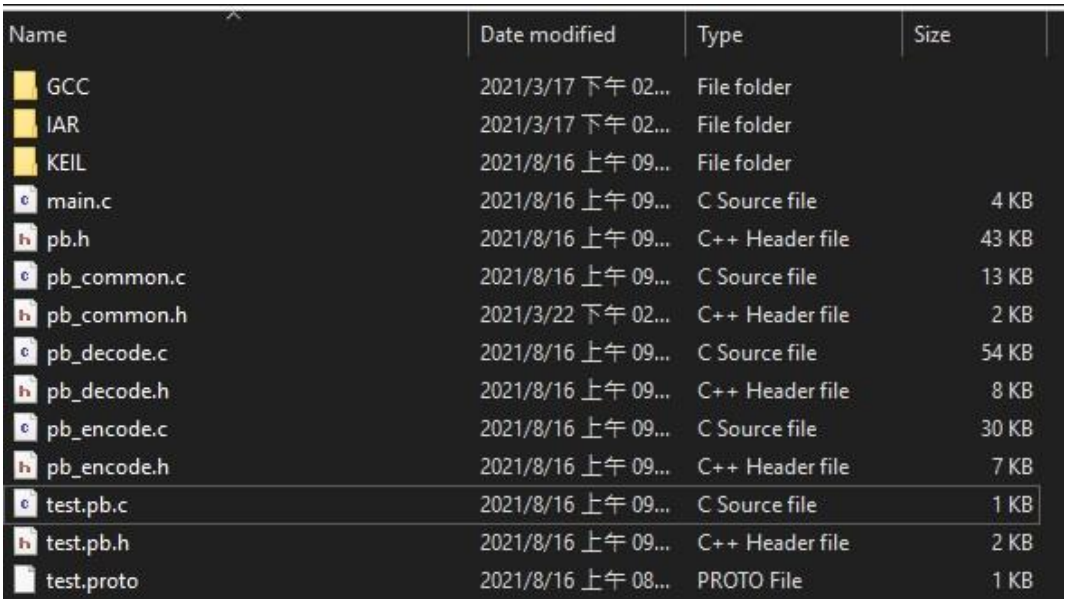

图 1-4 项目文件夹内容

把 Protobuf 文件夹里的文件添加到 mdk 环境，如图1-5 编译环境所示并增加头文件搜索路径：

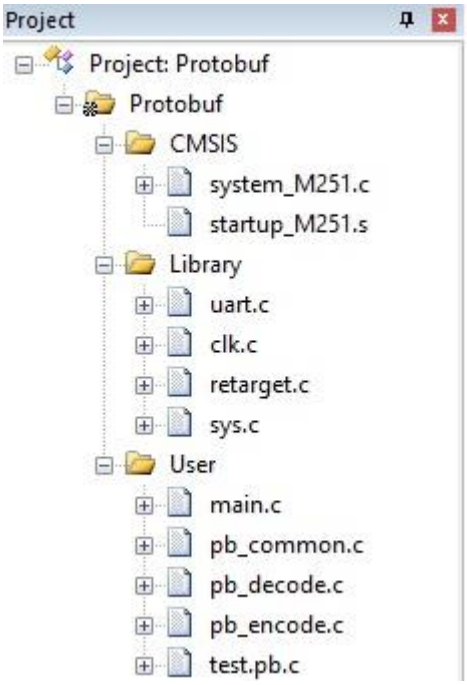

图 1-5 编译环境

Demo 很简单，编码就是建构一个协议数据结构，调用编码往 buffer 中写入数据；解码正好反过来，从 buffer 中读出数据解码到结构里。其中，编码相关的函数在 pb_encode.c 中，解码相关的函数在 pb_decode.c 中。

1.2 执行结果

执行结果，以及打印数据如图1-6 执行结果，以及打印信息所示。

图1-6 执行结果，以及打印信息

2 代码介绍

一开始初始化 pack_stu，编码后通过 o_stream 进行传送，再经由 i_stream 接收进行译码，得出数据写入 unpack_stu，比对结果如图 1-6 执行结果，以及打印信息：

```
        /* pack_stu and unpack_stu is Student type*/
        Student pack_stu = {0};
        Student unpack_stu = {0};
        /* pack_stu initialization */
        pack_stu.num  = 1024;
        pack_stu.py_score = 4096;
        pack_stu.c_score = 512;
        /* o_stream initialization */
        o_stream = pb_ostream_from_buffer(buffer, sizeof(buffer));
        /* encode all pack_stu fields from Student into o_stream */
        pb_encode(&o_stream, Student_fields, &pack_stu);
        /* i_stream initialization */
        i_stream = pb_istream_from_buffer(buffer, sizeof(buffer));
        /* decode all fields from i_stream into unpack_stu */
        pb_decode(&i_stream, Student_fields, &unpack_stu);
        /* print unpack_stu result */
        printf("unpack_stu.num = %d\n", unpack_stu.num);
        printf("unpack_stu.py_score = %d\n", unpack_stu.py_score);
        printf("unpack_stu.c_score = %d\n", unpack_stu.c_score);
```

编码所有字段：

```
bool checkreturn pb_encode(pb_ostream_t *stream, const pb_msgdesc_t *fields, const void
*src_struct)
{
    pb_field_iter_t iter;

    if (!pb_field_iter_begin_const(&iter, fields, src_struct))
        return true; /* Empty message type */

    do
    {
        if (PB_LTYPE(iter.type) == PB_LTYPE_EXTENSION)
        {
            /* Special case for the extension field placeholder */
            if (!encode_extension_field(stream, &iter))
                return false;
        }
```

```
        else
        {
            /* Regular field */
            if (!encode_field(stream, &iter))
                return false;
        }
    } while (pb_field_iter_next(&iter));

    return true;
}
```

解码所有字段：

```
bool checkreturn pb_decode(pb_istream_t *stream, const pb_msgdesc_t *fields, void
*dest_struct)
{
    bool status;

    status = pb_decode_inner(stream, fields, dest_struct, 0);

#ifdef PB_ENABLE_MALLOC
    if (!status)
        pb_release(fields, dest_struct);
#endif

    return status;
}
```

3 软件与硬件需求

3.1 软件需求

- BSP 版本
  - M251/M252/M254/M256/M258 Series CMSIS BSP V3.01.002
- IDE 版本
  - Keil uVersion 5.26

3.2 硬件需求

- 电路组件
  - NuMaker-M252SD Ver1.2
- 示意图

UART_TX (PB.13)	↔	RX
UART_RX (PB.12)	↔	TX
M252		PC

- 图 3-1 线路连接

4 目录信息

EC_M252_Protocol_Buffers_V1.00	
Library	Sample code header and source files
CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
Device	CMSIS compliant device header file
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	Source file of example code

图 4-1 目录信息

5 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 M252_Protocol_Buffers.uvprojx。
2. 进入编译模式界面
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开仿真模式
3. 进入仿真模式界面
   a. 执行代码

6 修订纪录

Date	Revision	Description
2021.09.24	1.00	1. 初始发布.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.