

Communication User Layer Module Design Reference

Version	V0.0.1
Status	Release
Authors	Mingfen XIAO

1.Documents Information

1.1 Change Log

Version	Date	Status	Authors	Comments
V0.0.1	2018.11.15	Draft	Mingfen XIAO	1.Inital version

1.2 Reference Documents

Index	Documents Name	Version
1		
2		
3		
4		
5		
6		
7		

2.Contents

1.	Documents Information	2
1.1	Change Log	2
1.2	Reference Documents	2
2.	Contents	3
3.	Introduction	4
3.1	Architecture Overview	4
3.2	Modules Files Structure	4
3.3	Modules Description	5
4.	Function Description	5
4.1	Initialization Function	5
4.2	Main Function	5
4.3	Read Signal	6
4.4	Write Signal	6
5.	API Description	6
5.1	ComUser_Init	6
5.2	ComUser_MainFunction	7
5.3	ComUser_RxMainFunction	7
5.4	ComUser_TxMainFunction	7
6.	Term	7

3.Introduction

3.1 Architecture Overview

图 3-1 为 Communication stack 的架构图。

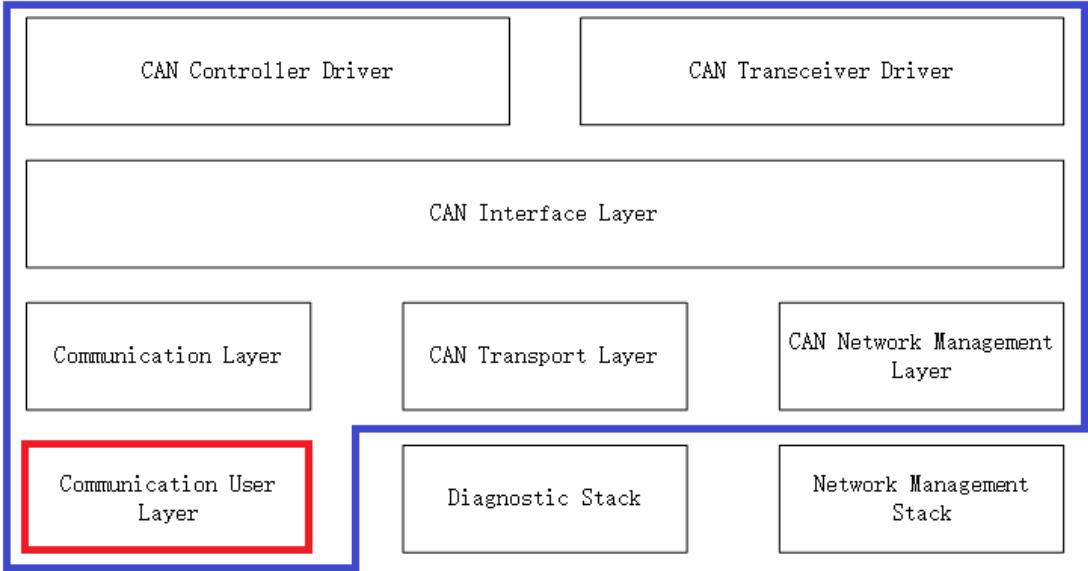


图 3-1 Communication Stack Architecture

如图 3-1 所示，蓝色区域为 Communication Stack 的各个模块。另外两个模块是和 Communication Stack 交互紧密的模块。

对于 Communication Stack 各个模块的功能和作用如下：

Can controller Driver Layer：对 CAN 控制器进行硬件的抽象以及驱动程序的实现。

Can Transceiver Driver Layer：对 CAN 收发器进行硬件的抽象以及驱动程序的实现。

Can Interface Layer：对硬件抽象层和服务层进行隔离。

Communication Layer：通信的服务层，主要对报文的编解码等操作。

Communication User Layer：通信的应用程序，主要是对信号的功能处理。

Can Transport Layer: 诊断报文的处理,主要是对 TP 层参数进行配置和实现 TP 层的逻辑。

Can Network Management Layer: 网络管理报文的处理,主要实现 CAN 网络网络的功能，但是对于不同总线的网络管理的功能不进行实现。

3.2 Modules Files Structure

File Name	File Type	Description
ComUser.c	Static	ComUser 模块的源文件。实现 ComUser 模块的函数
ComUser.h	Static	ComUser 模块的头文件。对 ComUser 模块的函数进行声明

ComUser_Type.h	Static	ComUser 模块数据类型的定义
VFB_ComUser.h	Dynamic	ComUser 需要使用外部模块的接口进行 API 的映射
ComUser_Cfg_Define.h	Dynamic	ComUser 模块用户功能开关选择以及参数的配置
ComUser_Cfg_Table.h	Dynamic	ComUser 模块用户列表的定义

3.3 Modules Description

Communication User Layer 是通信的应用层，是处于 communication stack 的最顶层。

Communication User Layer 主要由用户根据实际需求来实现的。一般情况下，为了增加模块的独立性，此模块可以作为应用层和服务层对报文收发的唯一接口。应用层其他模块对于总线信号的获取和发送也是通过此模块来实现的。

实际上此模块主要是根据项目来进行实现，基本上不存在和其他项目的共用性，特别是跨车厂之间的复用。

在此模块中，主要是定义相关的架构。对于具体的实现，将不做要求。一般情况下，ComUser 模块主要的功能为：

1. 读取信号或者信号组；
2. 发起信号或者信号组的发送请求；
3. 获取 message timeout 的标志；
4. 实现报文相关的诊断 monitor，主要作为诊断的 tester；
5. 用户自定义的需求。

4.Function Description

4.1 Initialization Function

定义一个 init function 的框架，具体作用根据需求实现。

一般情况下，是初始化 ComUser 的状态机控制信息，数据等。

4.2 Main Function

定义了一个 main function 的框架，其中包括 RX function 和 TX function。在现阶段是把 RX 和 TX function 直接放在 Main function 来执行。但是这样是不太科学的。因为每个报文的周期是不一样的，所以处理周期也不尽相同。

对于 main function 的调用方式大致为了分为两种：

- 1.将 main function 的运行周期作为为一个 comuser 模块的 tick，并且 TX 和 RX 也是以 tick 的周期来运行的，但是在 TX 和 RX 的 main function 增加 counter 以确定各个报文的处理周期。这样的方式是基于计数器的方式实现的，对于时间的精度不是很准确。
- 2.对于不同的报文处理创建对应周期的 handler 函数，并且将这些函数 mapping 到对于周期时间的任务中，这样的方式将会比较准确。

[注]: 1.对于 Main function 怎么和任何进行映射，这个不做要求。

2.还需要根据操作系统的情况，以及项目对时间精度的要求进行设计。

3.对于 Main function 只是当前设计创建的一个函数，实际项目中是否使用取决于项目的实际情况。

4.3 Read Signal

此功能是依赖于 Com 模块的 read signal 功能。在 ComUser 的 VFB 层中对于各个信号或者信号组进行封装，在 ComUser 中只需要调用对应的 API 即可实现信号的读取。

此操作是 ComUser 主动发起的操作，对于什么时候需要获取 signal 完全由用户决定。

4.4 Write Signal

此功能是依赖于 Com 模块的 write signal 功能。在 ComUser 的 VFB 层中对于各个信号或者信号组进行封装，在 ComUser 中只需要调用对应的 API 即可实现信号的发送。

此操作是 ComUser 主动发起的操作，对于什么时候需要发起发送信号的请求完全由用户决定。

5.API Description

5.1 ComUser_Init

Function Prototype	void ComUser_Init(void)
Parameters List	
Return Value	
Function Description	
Limitations	
Note	

5.2 ComUser_MainFunction

Function Prototype	void ComUser_Init(void)
Parameters List	
Return Value	
Function Description	
Limitations	
Note	

5.3 ComUser_RxMainFunction

Function Prototype	void ComUser_RxMainFunction (void)
Parameters List	
Return Value	
Function Description	
Limitations	
Note	

5.4 ComUser_TxMainFunction

Function Prototype	void ComUser_TxMainFunction (void)
Parameters List	
Return Value	
Function Description	
Limitations	
Note	

6.Term

Term	Description
API	Application Programming Interface
CAN	Controller Area Network
