



VKware AUTOSAR4.3 BSW

V1.2

通信栈用户手册

2021.05.07

VKware



文档控制

日期	作者	版本	状态	说明
2021. 04. 23	程成	V0. 1	【初稿】	
2021. 05. 07	程成	V1. 0	【初版】	



目录

1.0 关于本文档.....	1
1.1 概述.....	1
1.1.1 目的.....	1
1.1.2 受众.....	1
2.0 简介.....	2
3.0 配置与集成.....	3
3.1 配置.....	3
3.1.1 开启通信模块.....	3
3.1.2 配置通信模块.....	4
3.1.3 校验.....	4
3.1.4 下载.....	5
3.2 集成.....	7
3.2.1 基于 Demo 工程.....	7
3.2.2 手动集成.....	9
4.0 详细配置示例.....	10
4.1 通信未使用 CanTp 模块.....	10
4.1.1 配置 EcuC.....	11
4.1.2 抽取 Can 驱动适配的配置.....	12
4.1.3 配置 CanIf.....	14
4.1.4 配置 Com.....	20
4.1.5 配置 PDUR.....	24
4.1.6 配置 ComM.....	28
4.1.7 配置 CanSm.....	30
4.2 通信使用 CanTp 模块.....	31
4.2.1 配置 EcuC.....	32
4.2.2 抽取 Can 驱动适配的配置.....	33
4.2.3 配置 CanIf.....	34
4.2.4 配置 CanTp.....	40
4.2.5 配置 Com.....	43
4.2.6 配置 PDUR.....	48
4.2.7 配置 ComM.....	51
4.2.8 配置 CanSm.....	51
5.0 配置参数介绍.....	52
6.0 API 接口.....	53



1.0 关于本文档

1.1 概述

1.1.1 目的

本文档旨在引导用户从零开始通过 VKware 线上工具系统配置符合 Autosar 4.3 通信栈及配置代码、源代码下载和集成。

1.1.2 受众

本文档仅限于基于 VKware 线上工具系统做开发的工程人员。



2.0 简介

通信栈（除 MCAL）是与硬件平台无关的。因此，在完成通信栈配置并下载源码及配置代码后可结合相关驱动在任意开发平台上测试。本次发布版本为 1.2，有以下限制：

- 源码、配置、示例均基于 Autosar 4.3 版本；
- 仅提供基于 CAN 总线的通信配置和源码下载；
- COM 模块中基于 signal 的网关功能暂未开放；
- ComM 模块中 PNC 相关功能暂未开放；

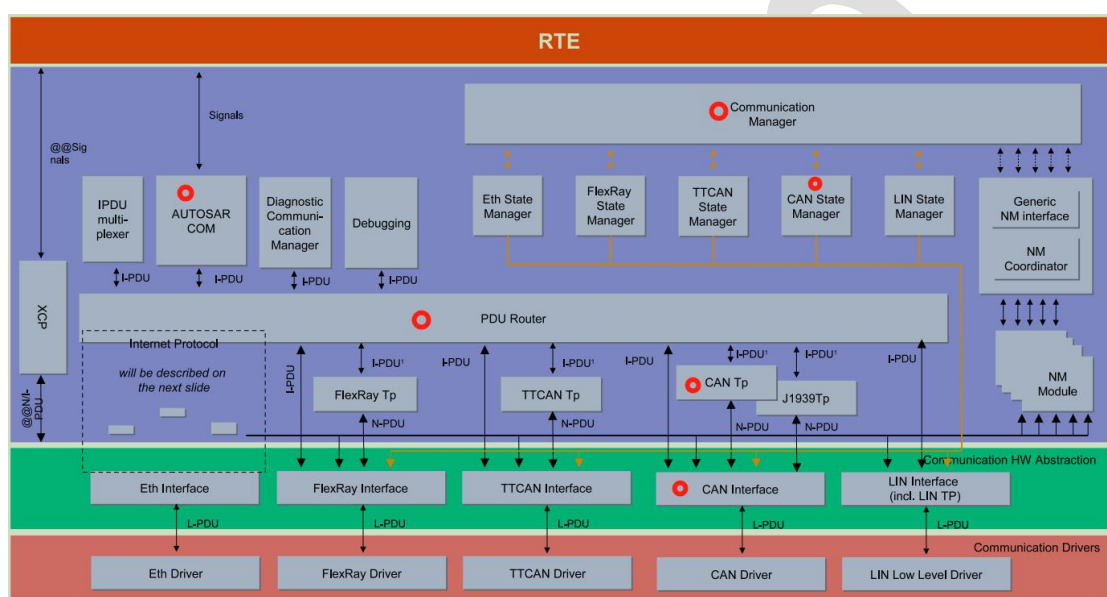


图 2-1 Autosar 通信层次结构

如图 2-1 所示，以红圈标记的模块可以配置和下载。各个部分的功能如下：

- 1、CAN Interface: 为上层模块（CanTp、PduR、CanSm）提供统一的接口与 CAN Driver 进行交互。
- 2、CAN Tp: 用于传输大于 8 字节（传统 CAN）或 64 字节（CANFD）的 PDU。
- 3、PDU Router: 为上层（Com）和下层（CanTp、CanIf）提供 PDU 路由服务。
- 4、AUTOSAR COM: 分析接收到的 PDU 和组织发送 PDU 的发送。
- 5、CAN State Manager: 通过 CanIf 提供的接口控制 Can 控制器和 Can 收发器的状态。
- 6、Communication Manager: 对通信栈通信状态进行管理。



3.0 配置与集成

3.1 配置

说明：本次发布未提供 CanDriver 的实现与配置，需用户自行按照 Autosar 4.3 中 CAN 控制器驱动和 CAN 收发器驱动规范进行开发、集成。

3.1.1 开启通信模块

通过侧边栏 Communication 项点击“Overview”，在“Overview”配置页面选择所需要的通信模块，如图 3-1。然后点击“Save”即可对选择的通信模块进行开启，如图 3-2。

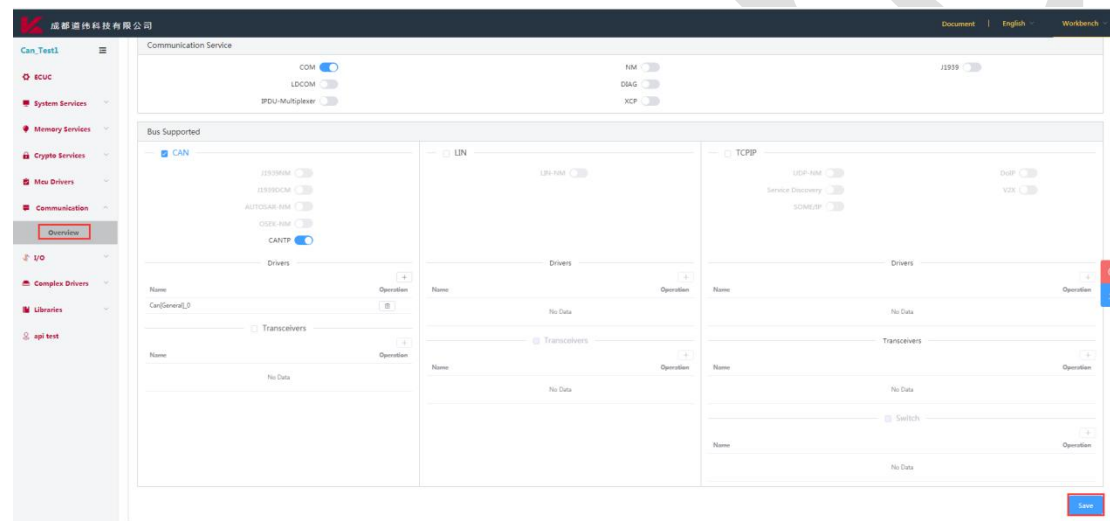


图 3-1 通信模块的选择

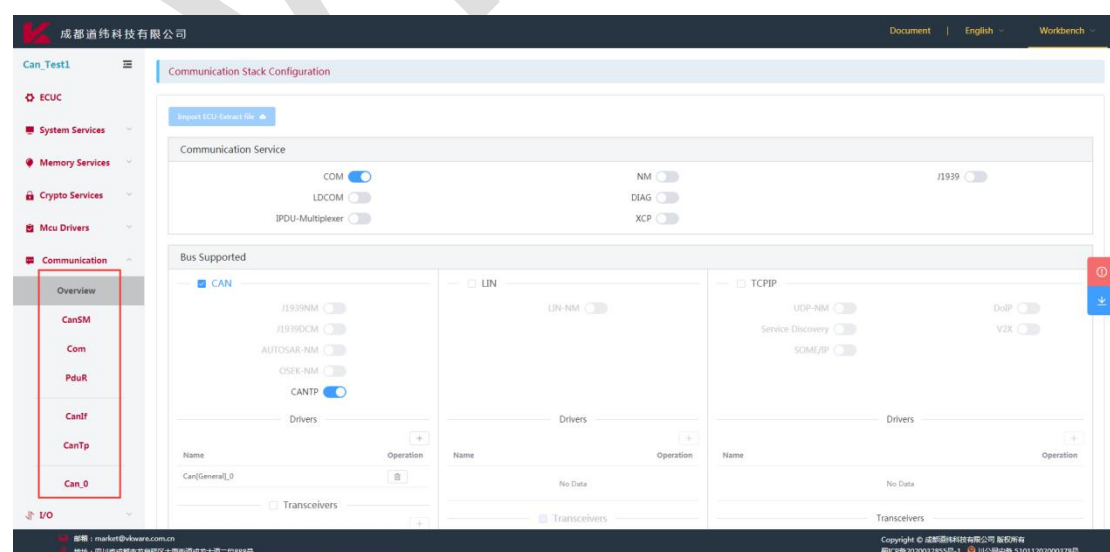




图 3-2 通信模块开启后

3.1.2 配置通信模块

如下图 3-3 所示，分别点击每个模块可对相应的模块进行详细的配置。

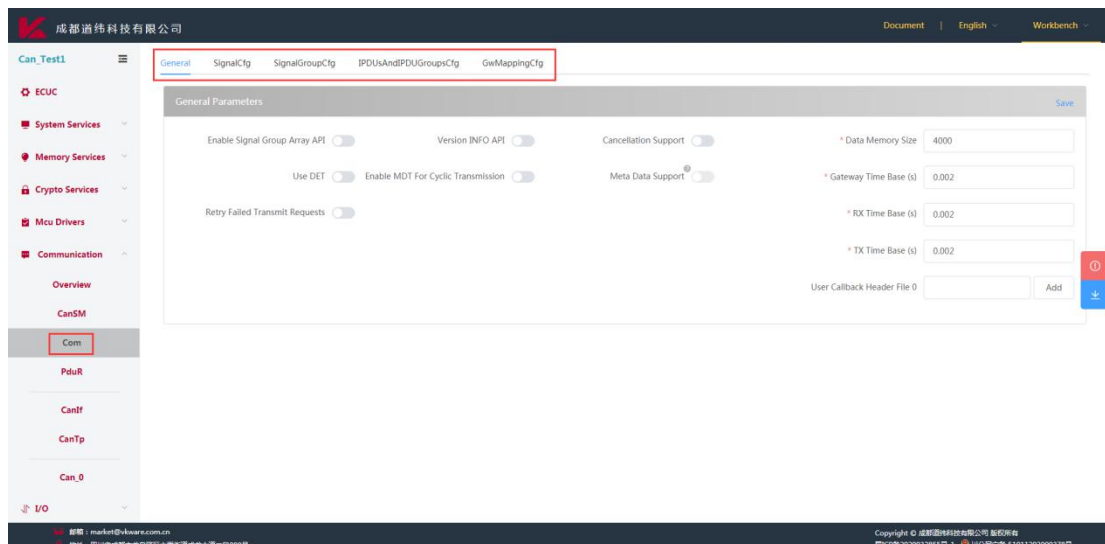


图 3-3 详细配置界面

3.1.3 校验

配置完成后，可以点击“”校验配置内容是否正确，但不会生成配置文件与提取源码文件。

配置内容错误信息将在错误日志页面显示，每个配置错误会有一条配置错误信息描述，以及该错误配置在配置中的位置，以便查找更正，如图 3-4。



图 3-4 校验错误日志



3.1.4 下载

配置完成后，配置内容没有错误，才能正常下载配置代码和源代码。在点击“generate”时，工具系统也会先校验配置内容是否正确，配置正确后，才会生成配置文件与提取源码文件。若有配置错误，错误信息也将在错误日志页面显示。

根据配置可能提取到文件列表如下表所示：

源文件	源文件
CanIf.c	CanSM.c
CanIf.h	CanSM.h
CanIf_Cbk.h	CanSM_BswM.h
CanIf_Ctrl.c	CanSM_Cbk.h
CanIf_DevErrCheck.c	CanSM_ComM.h
CanIf_Internal.h	CanSM_DevErrCheck.c
CanIf_MemMap.h	CanSM_EcuPassive.c
CanIf_Trcv.c	CanSM_ICom.c
CanIf_Types.h	CanSM_Internal.h
CanIf_Version.c	CanSM_TxTimeoutException.h
Com_Version.c	CanSM_Version.c
Com.c	ComM.c
Com.h	ComM.h
Com_64Bit.c	ComM_BusSM.h
Com_Cbk.h	ComM_Com.h



Com_DevErrorCheck.c	ComM_DevErrorCheck.c
Com_Float.c	ComM_EcuMBswM.h
Com_Internal.h	ComM_Internal.h
Com_Internal64.h	ComM_Version.c
Com_InternalFloat.h	PduR_Com.c
Com_IpduGroupCtrl.c	PduR_Com.h
Com_Rx.c	PduR_DevErrorCheck.c
Com_Tx.c	PduR_Internal.h
PduR.c	PduR_Types.h
PduR.h	PduR_Version.c
PduR_CanIf.c	PduR_XIf.c
PduR_CanIf.h	PduR_XTp.c
PduR_CanTp.c	PduR_CanTp.h
配置文件	配置文件
CanIf_Cfg.h	CanSM_Cfg.c
CanIf_Lcfg.c	CanSM_Cfg.h
CanIf_MemMap.h	CanSM_MemMap.h
CanIf_PBcfg.c	CanSM_PBcfg.c
CanIf_PBcfg.h	ComM_Cfg.c
Com_Cfg.c	ComM_Cfg.h
Com_Cfg.h	ComM_MemMap.h



Com_Lcfg.c	ComM_PBCfg.c
Com_MemMap.h	PduR_Cfg.c
Com_PBcfg.c	PduR_Cfg.h
Com_PBcfg.h	PduR_MemMap.h
SchM_PduR.h	PduR_PBcfg.c
SchM_PduR_Type.h	PduR_PBcfg.h
SchM_CanIf.h	SchM_Com.h
SchM_CanIf_Type.h	SchM_Com_Type.h
SchM_CanSM.h	SchM_ComM.h
SchM_CanSM_Type.h	SchM_ComM_Type.h
依赖文件	依赖文件
Platform_Types.h	Can_GeneralTypes.h
Std_Types.h	Compiler_Cfg.h
Compiler.h	ComStack_Cfg.h
ComStack_Types.h	

3.2 集成

3.2.1 基于 Demo 工程

Demo 工程可以在成都道伟科技有限公司官网获取：<https://www.vkware.com.cn/onlinevk>。如下图所示。



Demo下载



NXP S32K14x

支持的开发板：蓝宙FRDM-S32K144
支持的编译器：IAR Embedded Workbench for Arm 8.40.2
模块：通信

下载



NXP S32K14x

支持的开发板：蓝宙FRDM-S32K144
支持的编译器：IAR Embedded Workbench for Arm 8.22.2
模块：OS

下载



AURIX TC27x

支持的开发板：TriBoard-TC2X5-V2.0
支持的编译器：HighTec v4.9.2.0

下载



AURIX TC23x

支持的开发板：APPLICATION KIT TC2X4 V1.0
支持的编译器：HighTec v4.9.2.0

下载

图 3-5 Demo 下载

若需更新配置,将 Demo 工程的 src 及 config 文件夹中所有文件删除后添加下载的文件即可。

说明:

- 1、本次发布仅提供基于 **NXP S32K14x** 集成通信示例。
- 2、Gpt 驱动未实现,但集成中会用到,因此为了方便采用 2ms 任务中累加全局计时变量 **Gpt_preTimer** 间接实现 **Gpt**。
- 3、CAN 驱动中,接收与发送均采用中断方式。在配置通信模块前先在 **OS** 模块中添加如下配置 **Can** 中断。(参考: <https://www.vkware.com.cn/vkdoc/chapter2/OS.html> 相关文档)

中断 (name)	中断类别	优先级	中断嵌套	堆栈大小 (Bytes)	中断源
OS_ISR_MCA N_IRQ_CORE0	二类	3	FALSE	200	OS_CAN0_ORed_0_15_MB_IRQn

- 4、若配置中开启了 **CanTp** 模块,则将 **main.h** 中的宏定义 **CANTP_USED** 为 1, 否则定义为 0。



3.2.2 手动集成

3.2.2.1 添加文件至工程

将下载的源文件添加至工程 `src`、将下载的配置文件添加至工程 `config`。下载的依赖文件自行根据工程进行添加。

3.2.2.2 添加头文件

将下载的源文件和配置文件中 “.h” 结尾的文件添加至工程 `include` 目录。

3.2.2.3 添加配置数据声明

a、将 `CanIf.h`、`CanSm.h`、`ComM.h`、`PduR.h`、`CanTp.h`、`Com.h` 头文件添加至 `xxx.c`。

b、声明 `CanIf_PBCfgData`、`CanSM_PBCfgData`、`Com_PbCfgDataRoot`、`PduR_PBCfgData`、`CanTp_PBCfgData` 全局变量。

注：未使用 `CanTp` 则不用添加 `CanTp.h` 头文件和 `CanTp_PBCfgData` 全局变量。

3.2.2.4 初始化

完成 CAN 驱动的初始化后，依次调用 `CanIf_Init(&CanIf_PBCfgData)`、`CanSM_Init(&CanSM_PBCfgData)`、`ComM_Init(NULL_PTR)`、`PduR_Init(&PduR_PBCfgData)`、`CanTp_Init(&CanTp_PBCfgData)`、`Com_Init(&Com_PbCfgDataRoot)` 进行通信模块的初始化。

注：未使用 `CanTp` 则不用调用 `CanTp_Init(&CanTp_PBCfgData)`。

3.2.2.5 任务函数

在某一周期任务中调用 `ComM_MainFunction(OU)`、`CanSM_MainFunction()`、`CanTp_MainFunction()`、`Com_MainFunctionTx()`、`Com_MainFunctionRx()`。

注：未使用 `CanTp` 则不用调用 `CanTp_MainFunction()`。

3.2.2.6 启动通信

完成以上操作后，依次调用 `ComM_CommunicationAllowed(OU, TRUE)`、`ComM_RequestComMode(OU, COMM_FULL_COMMUNICATION)` 即可启动通信。

说明：以上步骤假设工程除通信栈外其它模块已正确构建。



4.0 详细配置示例

4.1 通信未使用 CanTp 模块

本实例演示如何配置短 PDU 的收、发。通信栈功能开启如图 4.1-1，PDU 流向如图 4.1-2。

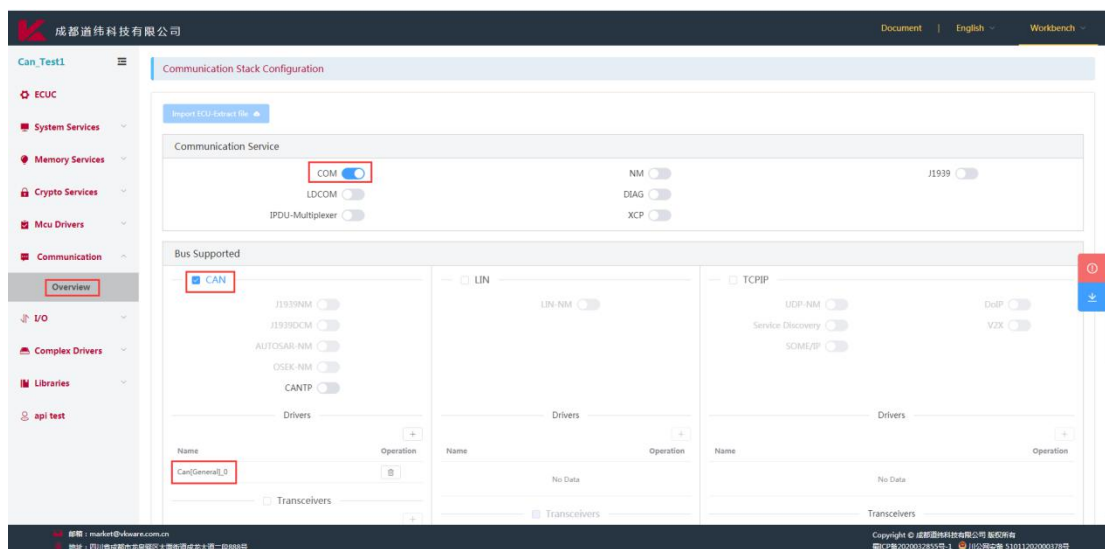


图 4.1-1 通信栈功能配置

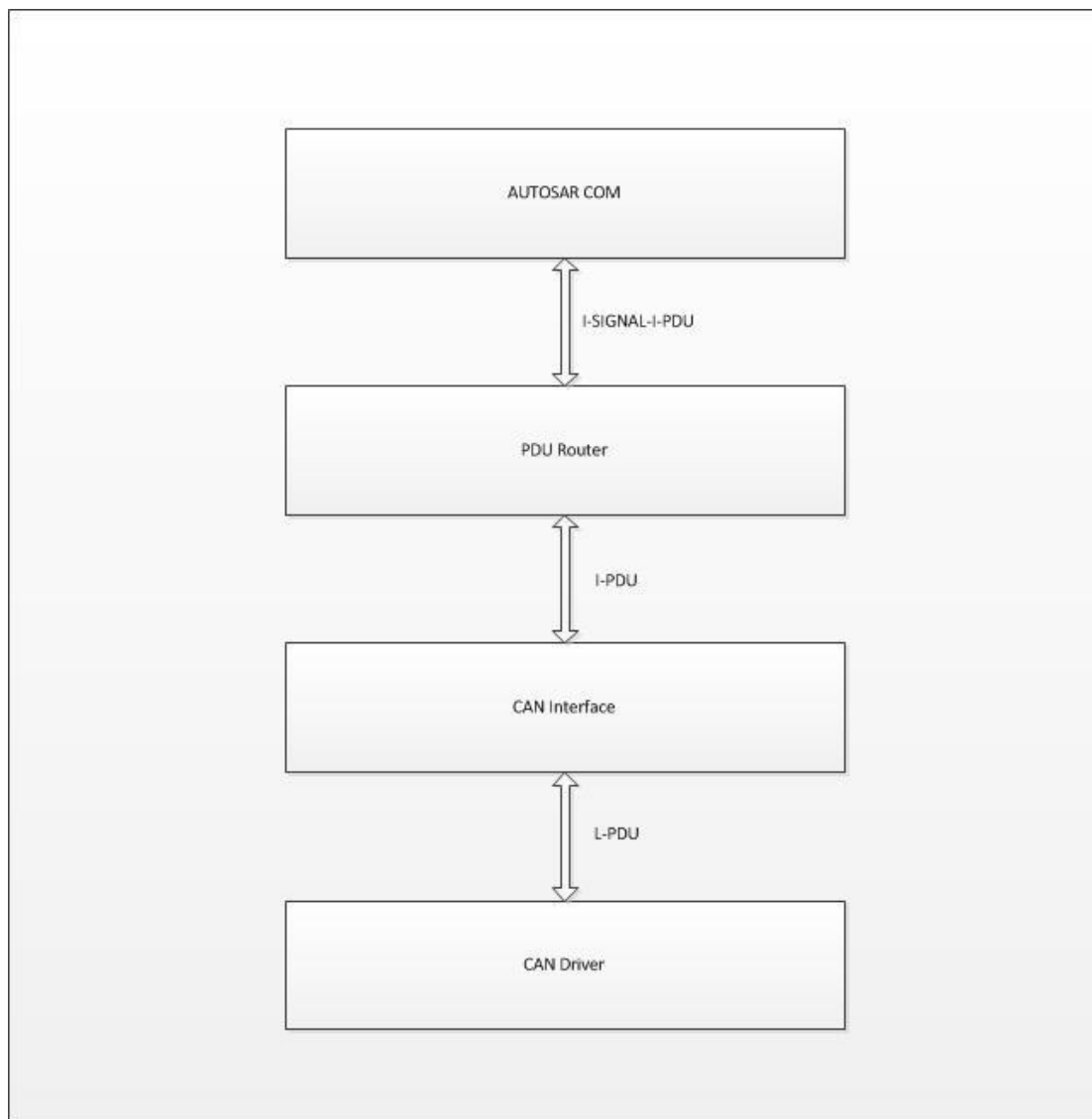


图 4.1-2 PDU 流向示例

4.1.1 配置 EcuC

前提：

1、根据 Autosar 方法学，通信栈中所有使用到的 PDU 在 EcuC 中进行配置。

假设：

1、所有 PDU 不含 metaData。



添加 PDU:

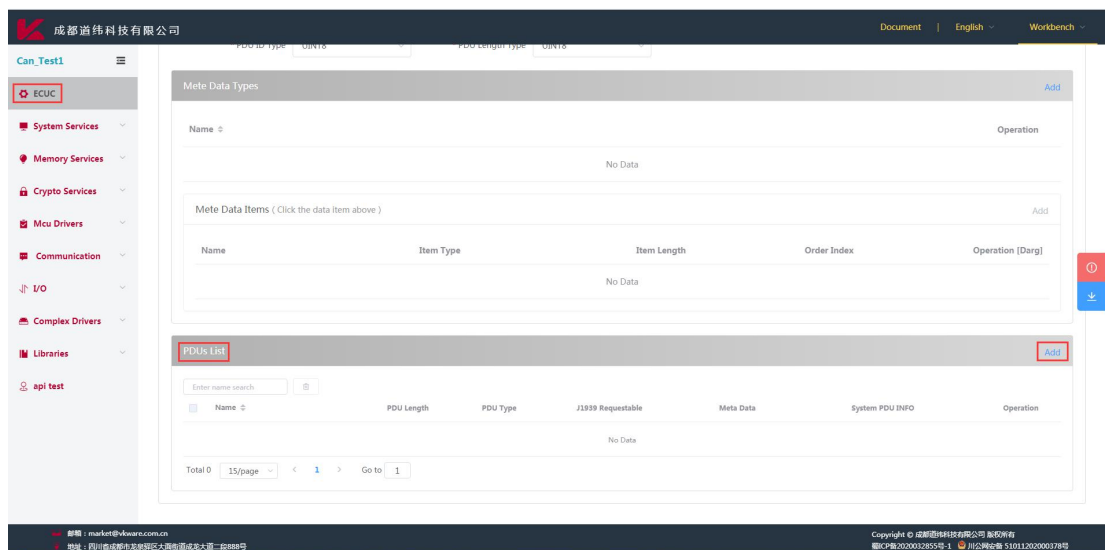


图 4.1-3 EcuC 中添加 PDU

配置参数:

Name	PDU Length	PDU Type	其它
IPDU_rx_360	8	空	默认
IPDU_tx_361	8	空	默认
LPDU_rx_360	8	CAN	默认
LPDU_tx_361	8	CAN	默认

4.1.2 抽取 Can 驱动适配的配置

前提:

1、Can 驱动按照 Autosar 4.3 标准实现。

假设:

1、Can 驱动提供两个邮箱，接收一个、发送一个。

2、Can 驱动不支持 CanFD。

3、Can 驱动提供一个 Can 控制器。

4、Can 控制器没有唤醒功能。



5、未包含 Can 收发器驱动。

关闭供应商 Api 标识：

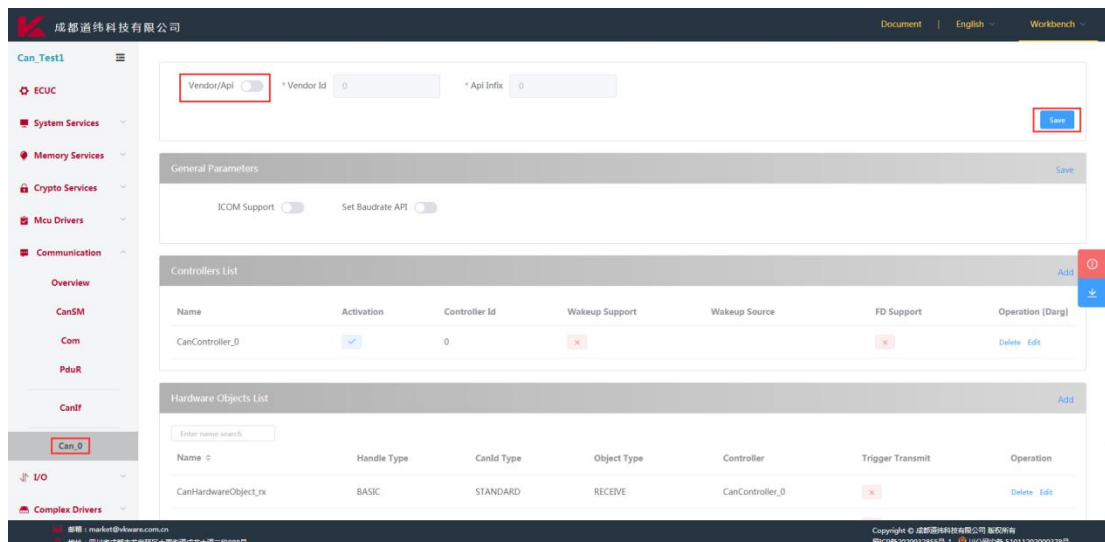


图 4.1-4 关闭供应商 Api 标识

添加一个 Controller：

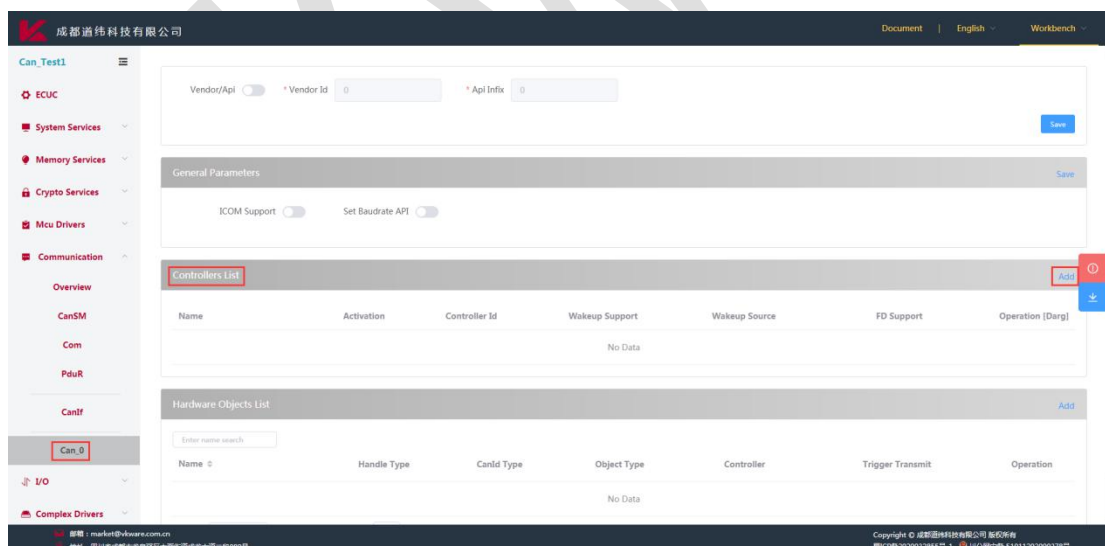


图 4.1-5 添加 Controller

配置参数：

Name	Activation	Wakeup Support	FD Support
默认	开启	关闭	关闭



添加一个接收邮箱：

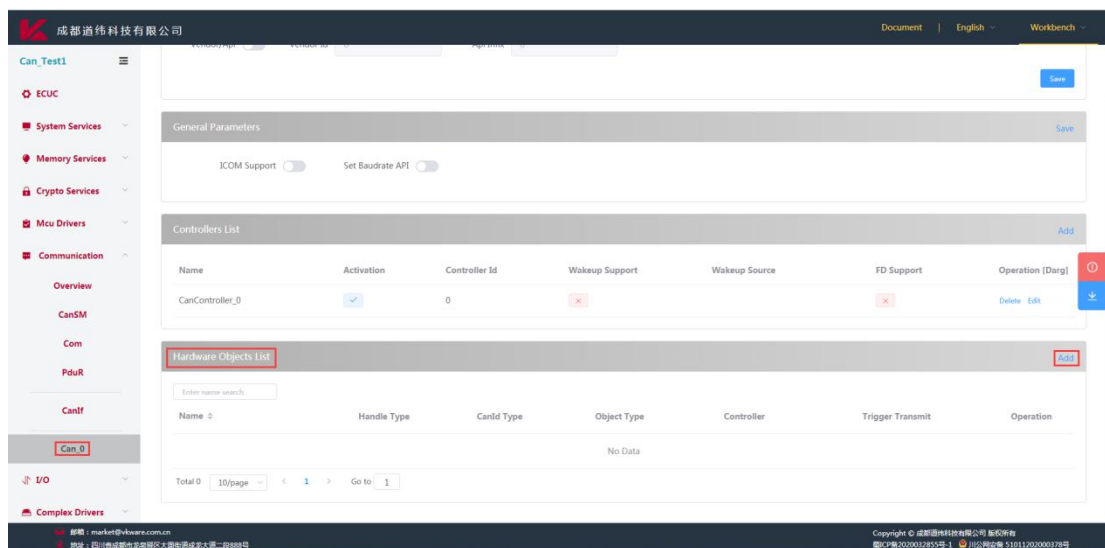


图 4.1-6 添加接收邮箱

配置参数：

Name	Handle Type	CanId Type	Object Type	Controller
CanHardwareObject_rx	BASIC	STANDARD	RECEIVE	CanController_0

同上添加一个发送邮箱：

配置参数：

Name	Handle Type	CanId Type	Object Type	Controller
CanHardwareObject_tx	BASIC	STANDARD	TRANSMIT	CanController_0

4.1.3 配置 CanIf

前提：

1、Can 驱动适已正确配置。

假设：

1、接收 canId 为 0x360。

2、发送 canId 为 0x361。



配置 General:

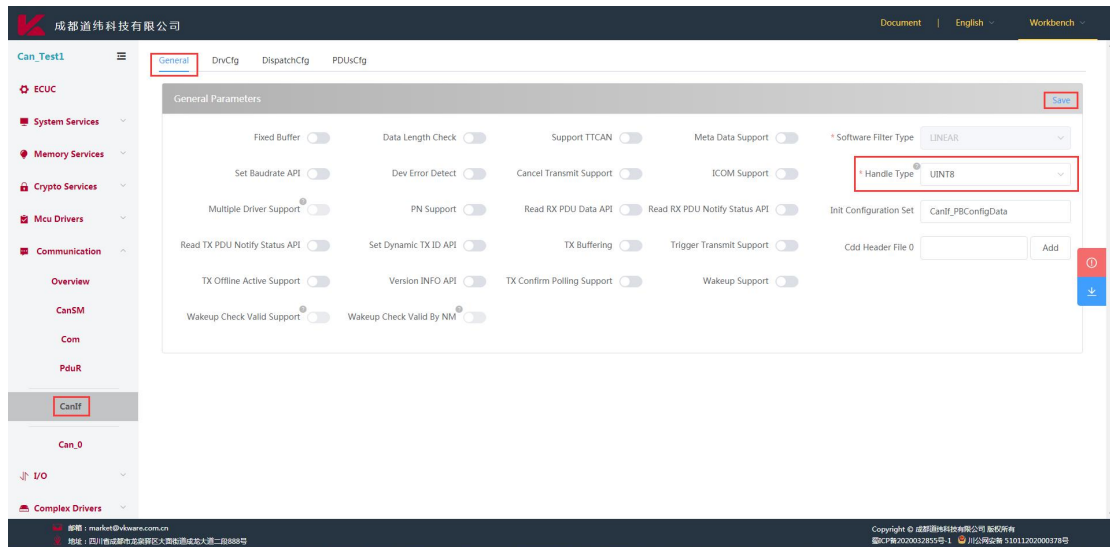


图 4.1-7 配置 General

配置参数:

Handle Type	其它
UINT8	默认

配置 DrvCfg:

关联 Can 驱动的 Controller:

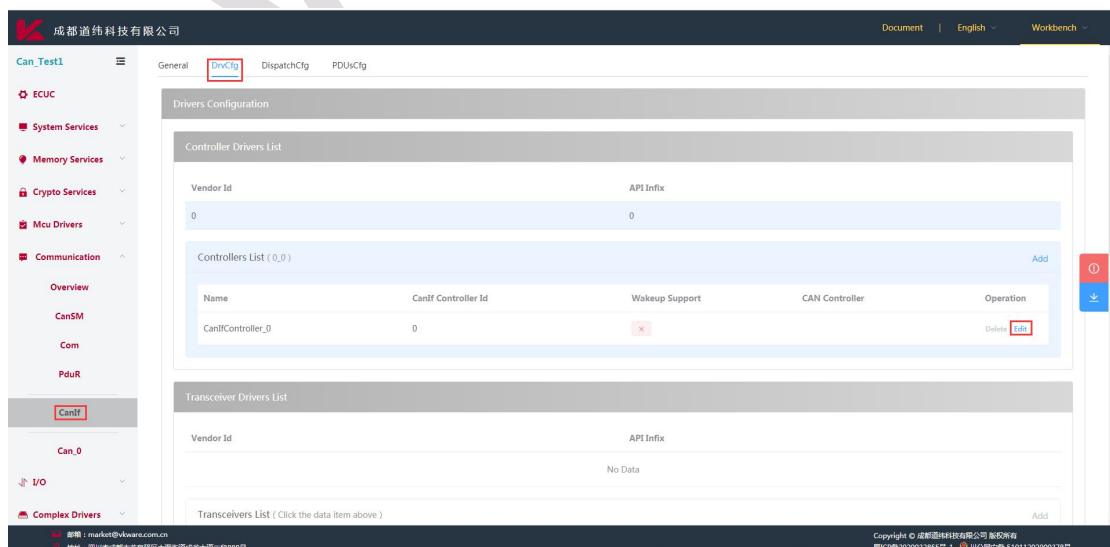




图 4.1-8 关联 Controller

配置参数：

Name	CAN Controller
默认	CanController_0

说明：此项配置由系统默认为每一个 CanDriver 生成一个 Controller，但需要关联，CanDriver 中的其它 Controller 需要通过点击“Add”进行添加和关联。

添加一个 HRH：

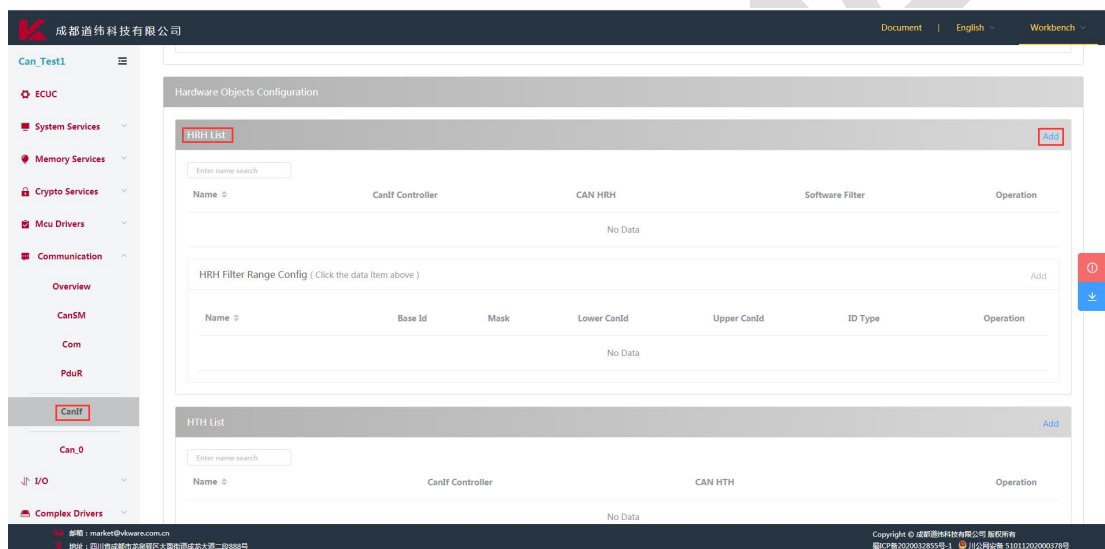


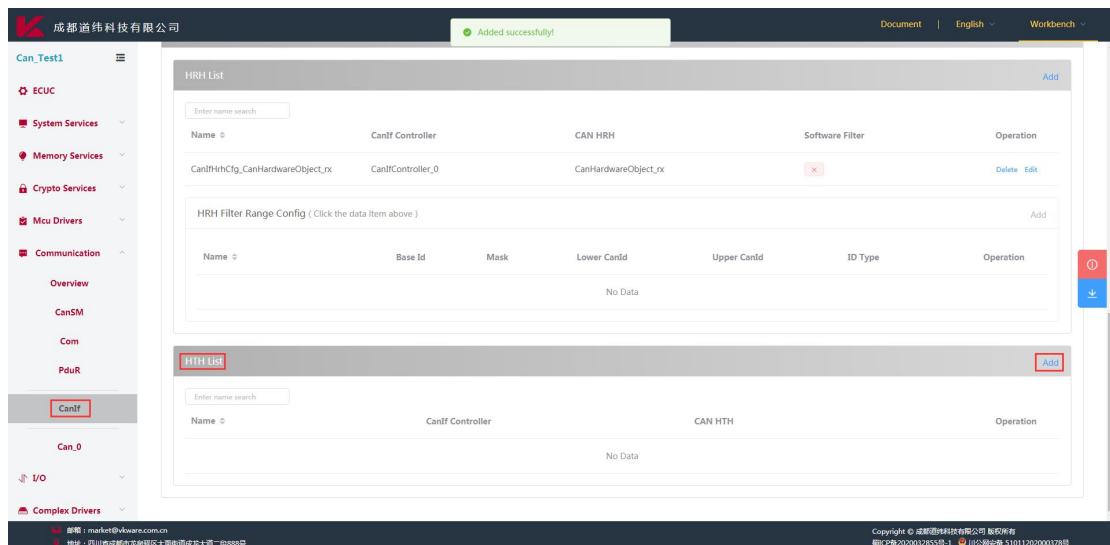
图 4.1-9 添加 HRH

配置参数：

Name	CanIf Controller	CAN HRH	Software Filter
默认	CanIfController_0	CanHardwareObject_rx	关闭



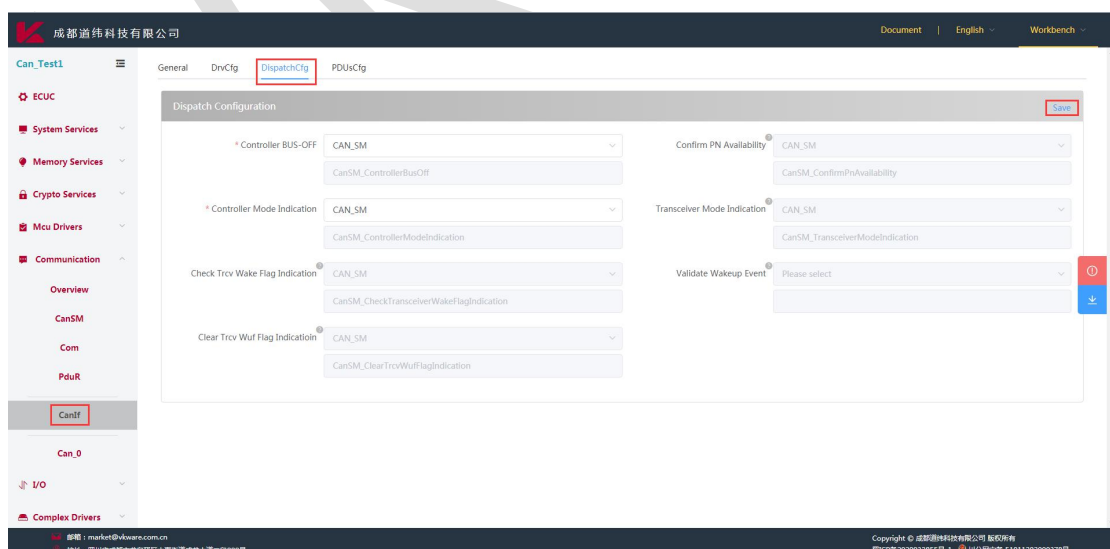
添加一个 HTH:



配置参数:

Name	CanIf Controller	CAN HTH
默认	CanIfController_0	CanHardwareObject_tx

配置 DispatchCfg:



配置参数:



Controller BUS-OFF	Controller Mode Indication
CAN_SM	CAN_SM

配置 PDUscfg:

配置接收和发送 PDU 的数量限制

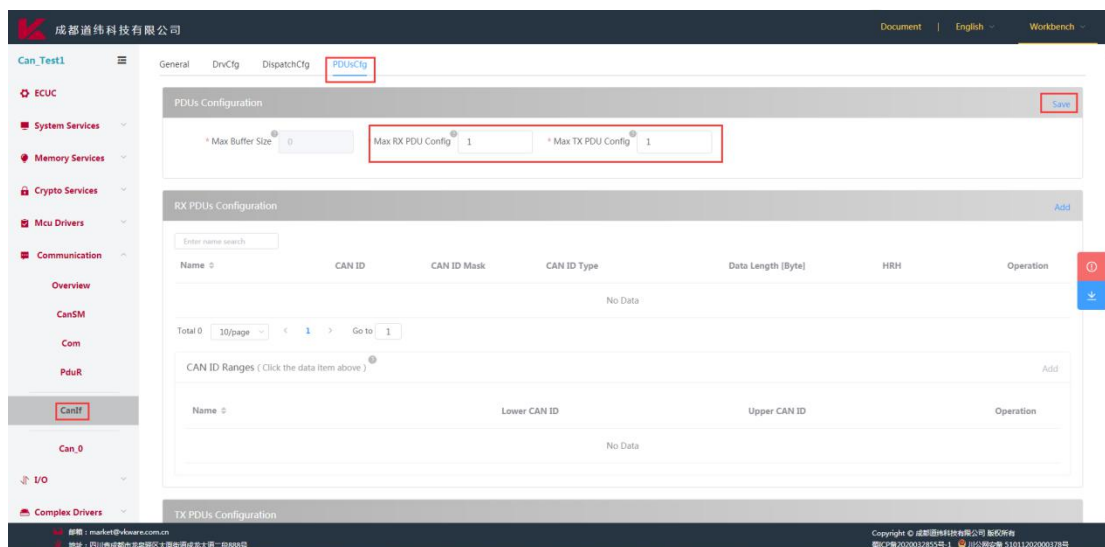


图 4.1-12 配置 PDUscfg

配置参数:

Max RX PDU Config	Max TX PDU Config
1	1



添加一个接收 PDU:

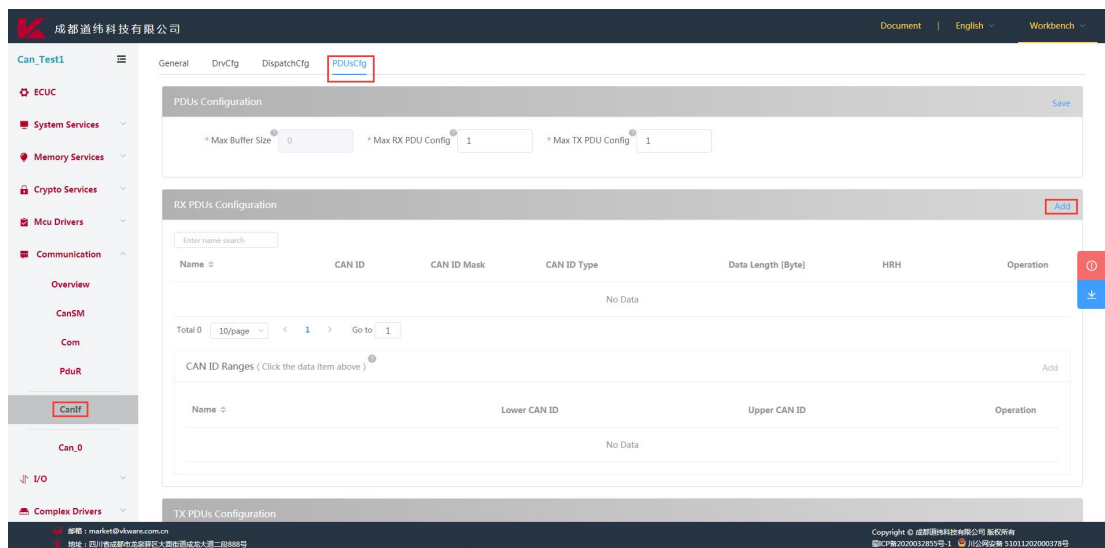


图 4.1-13 添加接收 PDU

配置参数:

Name	HRH	PDU Ref	CAN ID Type	CAN ID	CAN ID Mask	其它
默认	CanIfHrhCfgr_CanHa rdwareObject_rx	LPDU_rx_360	STANDARD_NO _FD_CAN	360	7ff	默认

添加一个发送缓存:

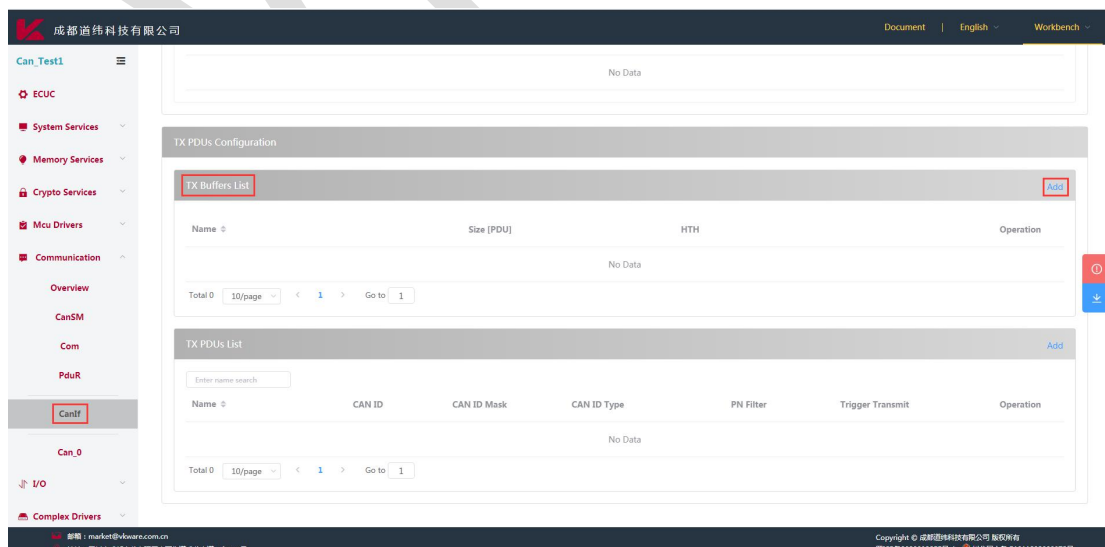


图 4.1-14 添加发送缓存



配置参数：

Name	HTH
默认	CanIfHrhCfg_CanHardwareObject_tx

添加一个发送 PDU：

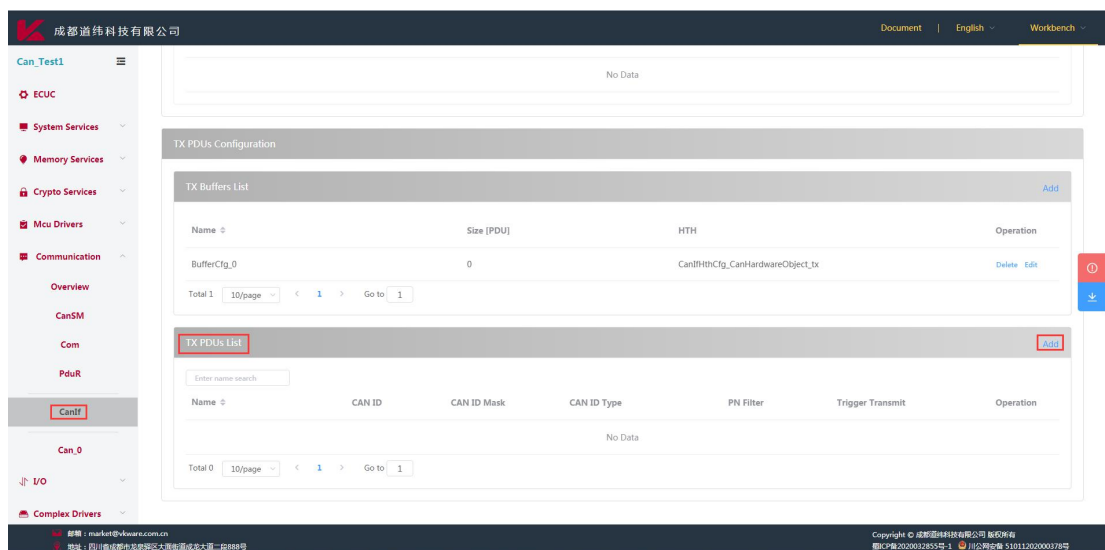


图 4.1-15 添加发送 PDU

配置参数：

Name	TX PDU Buffer Ref	PDU Ref	CAN ID Type	CAN ID	CAN ID Mask	其它
默认	BufferCfg_0	LPDU_tx_361	STANDARD_CA N	361	7ff	默认

4.1.4 配置 Com

前提：

无。

假设：

- 1、接收与发送方向均有一个 PDU。
- 2、接收 PDU 关联一个信号。
- 3、发送 PDU 关联一个信号。



配置 General:

默认

配置 SignalCfg:

添加一个接收方向信号:

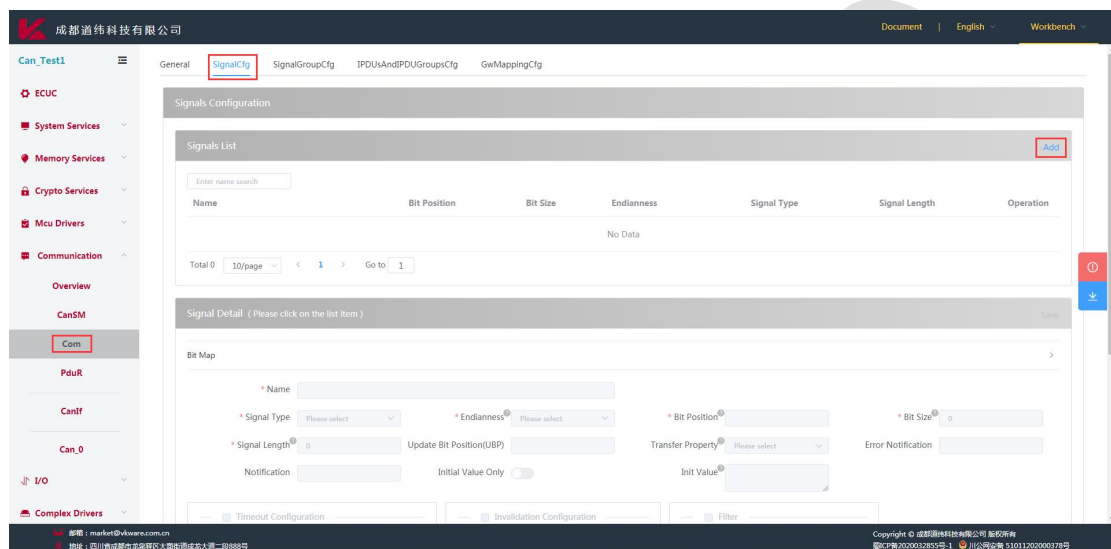


图 4.1-16 添加接收方向信号

配置参数:

Name	Signal_rx_360_vehicleSpeed
Signal Type	UINT16
Endianness	BIG_ENDIAN
Bit Position	8
Bit Size	12
Notification	Rte_ComCbk_Signal_rx_360_vehicleSpeed_RxNotification
Init Value	0
其它	默认



同上添加一个发送方向信号：

配置参数：

Name	Signal_tx_361_ambientTemperature
Signal Type	UINT8
Endianness	BIG_ENDIAN
Bit Position	0
Bit Size	8
Transfer Property	PENDING
Notification	Rte_ComCbk_Signal_tx_361_ambientTemperature_TxNotification
Init Value	0
其它	默认

配置 IPDUsAndIPDUGroupsCfg:

配置 IPDU 组和 IPDU 的数量限制

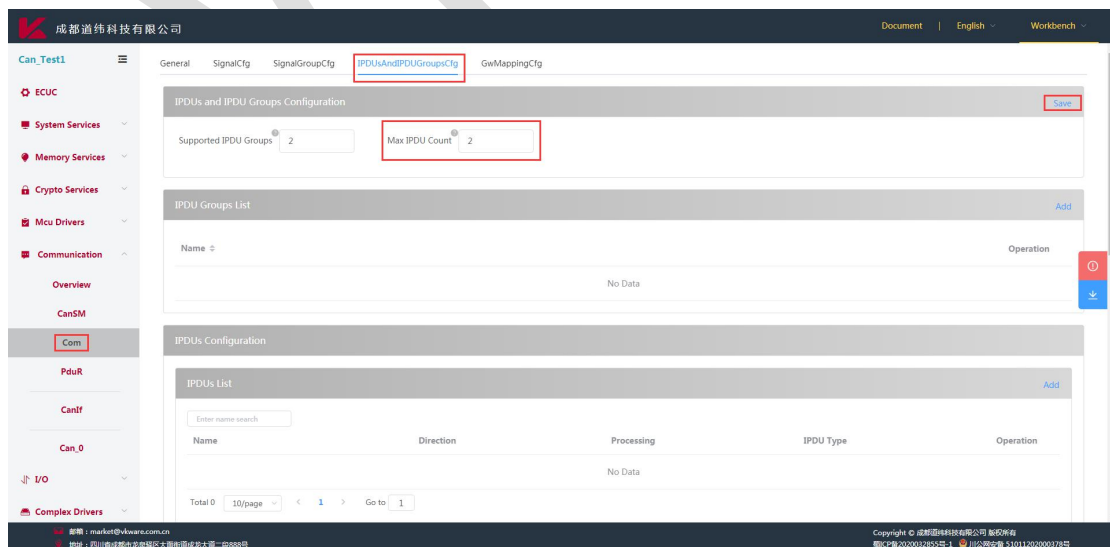




图 4.1-17 配置 IPDUsAndIPDUGroupsCfg

配置参数:

Supported IPDU Groups	Max IPDU Count
默认	2

添加一个接收 IPDU:

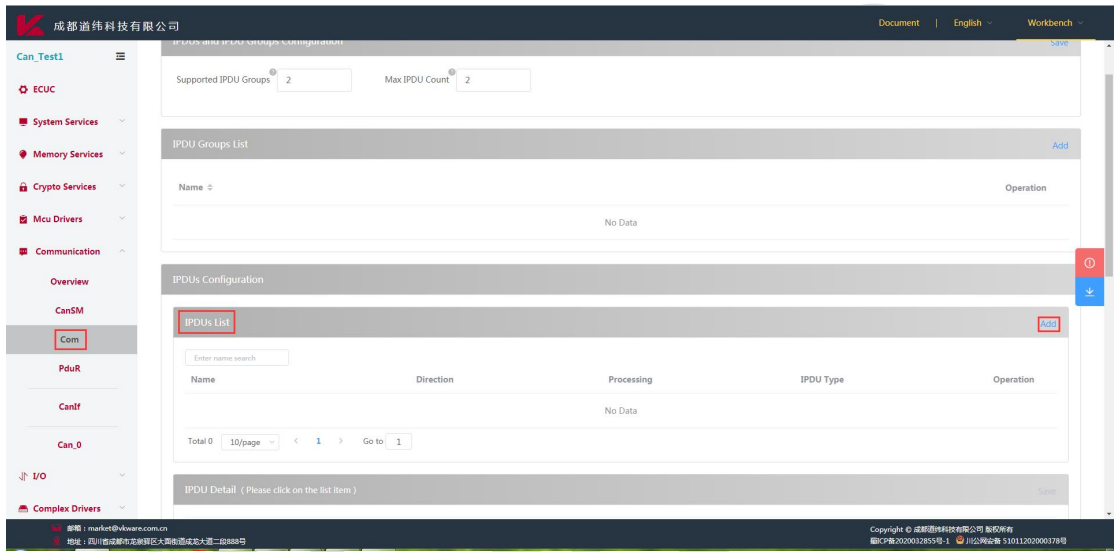


图 4.1-18 添加接收 IPDU

配置参数:

Name	IPdu_rx_360
PDU Ref	IPDU_rx_360
Direction	RECEIVE
Processing	IMMEDIATE
IPDU Type	NORMAL
Signal Refs	Signal_rx_360_vehicleSpeed
其它	默认



同上添加一个发送 IPDU:

配置参数:

Name	IPdu_tx_361
PDU Ref	IPDU_tx_361
Direction	SEND
Processing	DEFERRED
IPDU Type	NORMAL
Signal Refs	Signal_tx_361_ambientTemperature
TRUE Mode	勾选
Mode	PERIODIC
Time Period	0.5
其它	默认

4.1.5 配置 PDUR

前提:

1、无。

假设:

1、不使用 Zero Cost 功能。

配置 General:

默认



配置 BswModule:

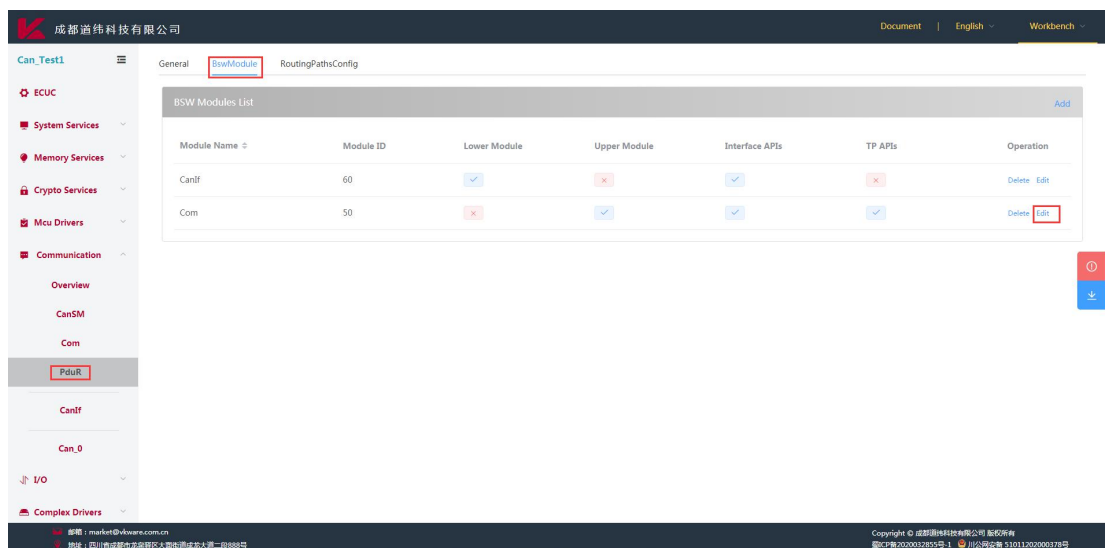


图 4.1-19 配置 BswModule

配置参数:

BSW Module	COM
Communication Interface	勾选
Transport Protocol	不勾选
其它	默认



配置 RoutingPathsConfig:

配置路由路径的数量限制:

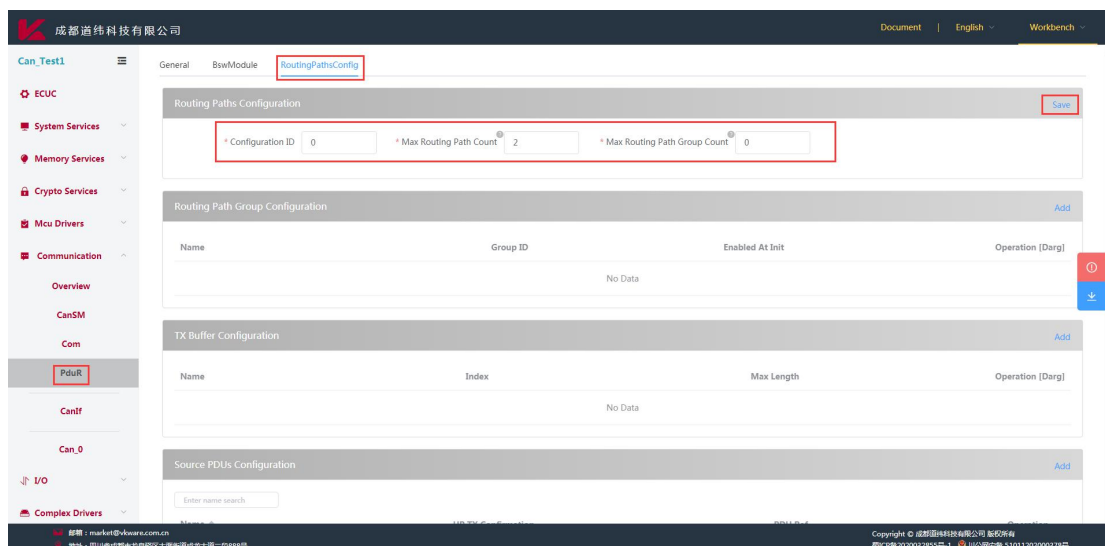


图 4.1-20 配置 RoutingPathsConfig

配置参数:

Configuration ID	Max Routing Path Count	Max Routing Path Group Count
0	2	0

添加两个源 PDU:

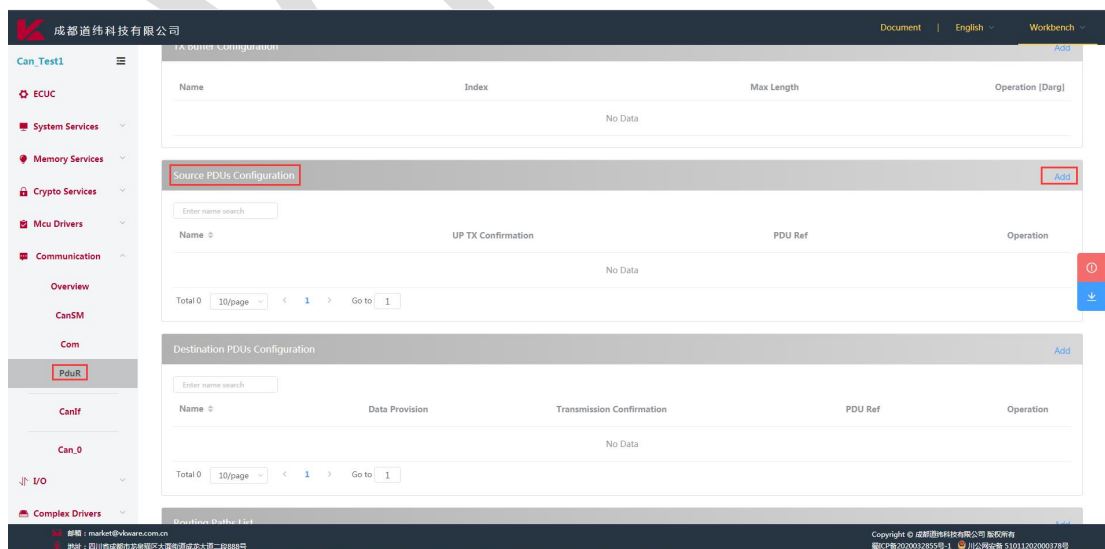




图 4.1-21 添加源 PDU

配置参数：

Name	Source Pdu Block Size	PDU Ref	UP TX Confirmation
默认	0	LPDU_rx_360	关闭
默认	0	IPDU_tx_361	打开

添加两个目的 PDU：

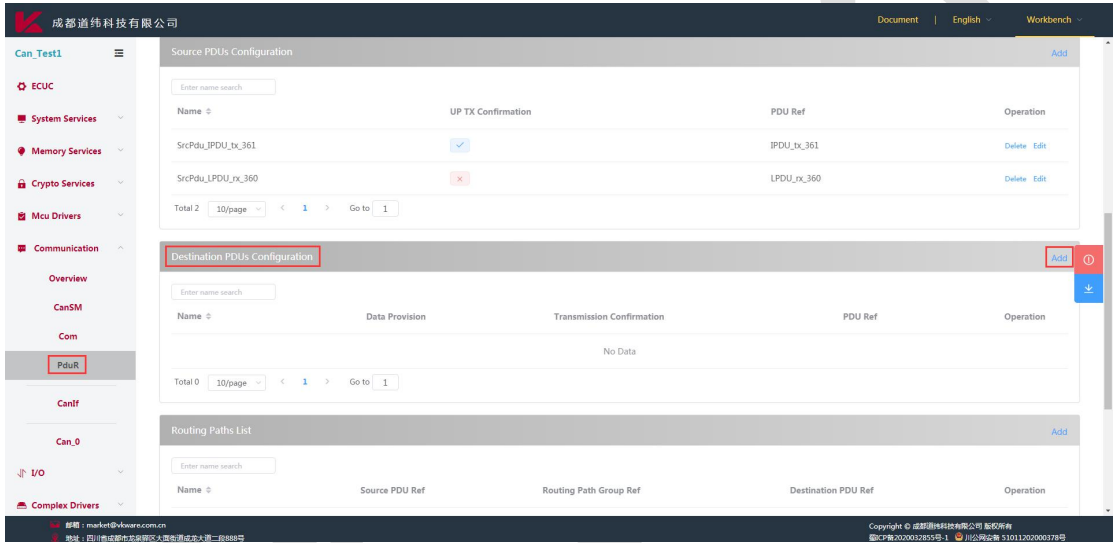


图 4.1-22 添加目的 PDU

配置参数：

Name	PDU Ref	Transmission Confirmation	其它
默认	IPDU_rx_360	关闭	默认
默认	LPDU_tx_361	打开	默认



配置 PDU 路径:

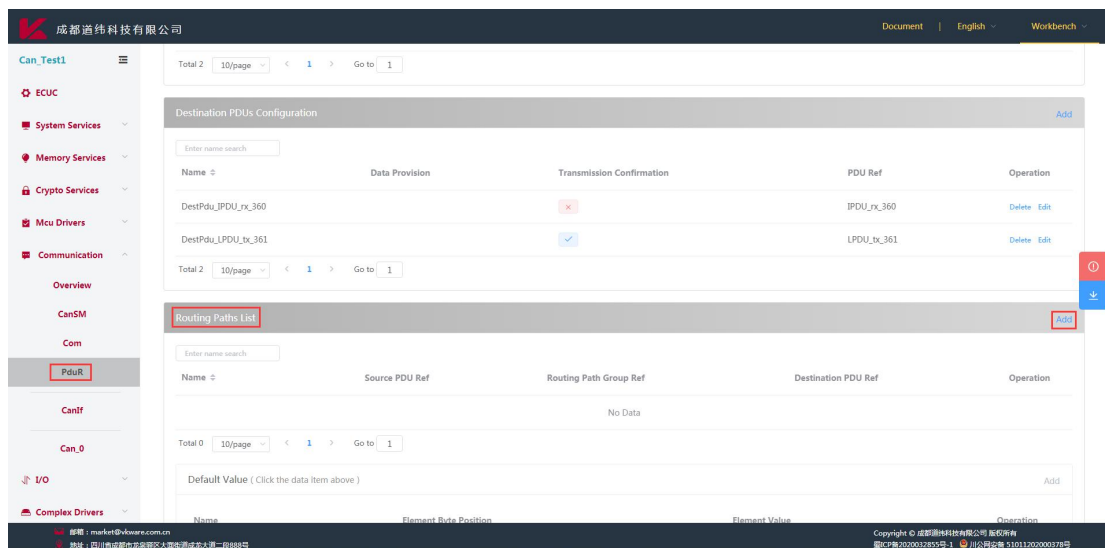


图 4.1-23 配置 PDU 路径

配置参数:

Name	Source PDU Ref	Destination PDU Ref	其它
默认	SrcPdu_LPDU_rx_360	DestPdu_IPDU_rx_360	默认
默认	SrcPdu_IPDU_tx_361	DestPdu_LPDU_tx_361	默认

4.1.6 配置 ComM

前提:

无。

假设:

- 1、仅有一个 user。
- 2、仅有一个通信通道。

配置 General:

默认



添加一个 user:

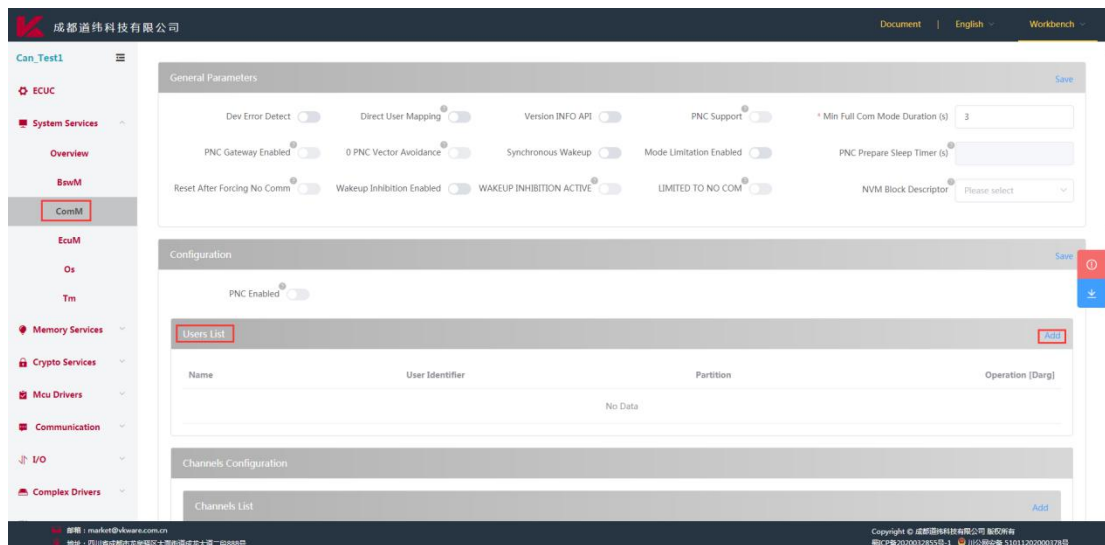


图 4.1-24 添加 user

配置参数:

Name	其它
ComMUser_can0	默认

添加一个通信通道:

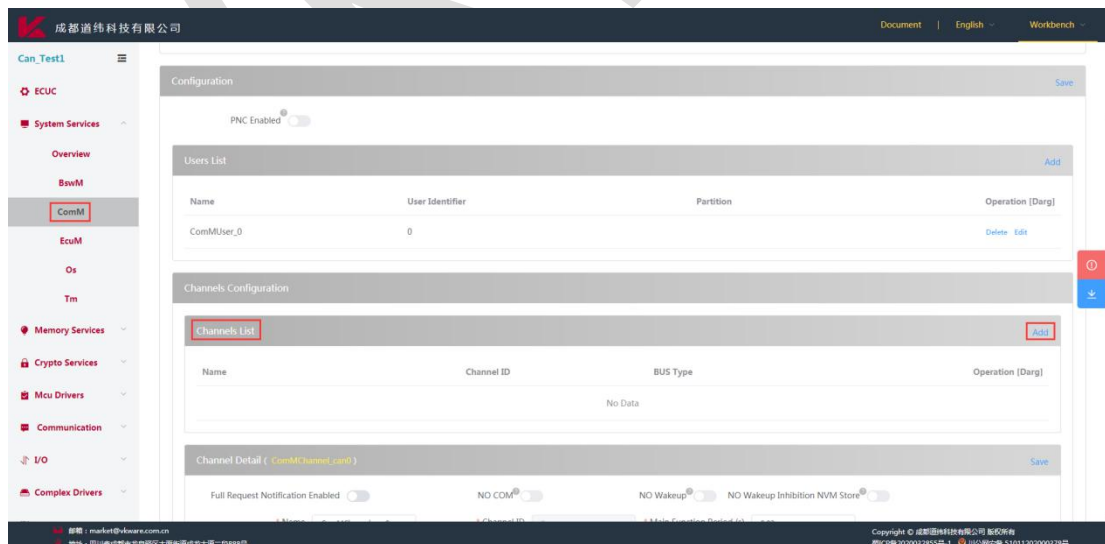


图 4.1-25 添加通信通道

配置参数:



Name	BUS Type	Channel Users	NM Variant	其它
ComMChannel_can0	BUS_TYPE_CAN	ComMUser_can0	NONE	默认

4.1.7 配置 CanSm

前提：

1、ComM 已正确配置。

假设：

无。

配置 General:

默认

添加一个网络配置：

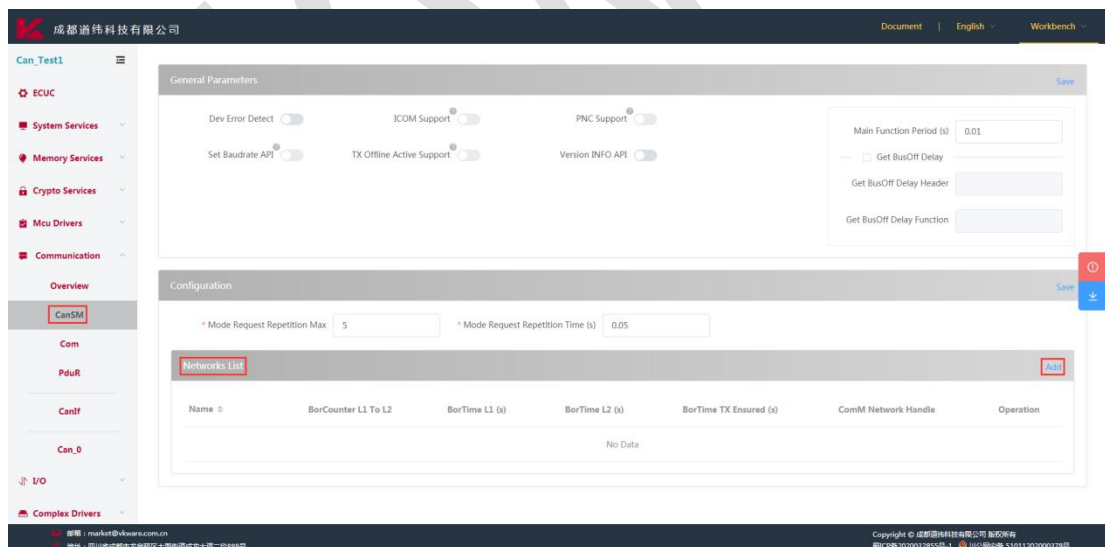


图 4.1-26 添加网络配置

配置参数：

Name	ComM Network Handle	CANIF Controllers	其它
默认	ComMChannel_can0	CanIfController_0	默认



4.2 通信使用 CanTp 模块

本实例演示如何配置长 PDU 的收、发。通信栈功能开启如图 4.2-1，PDU 流向如图 4.2-2。

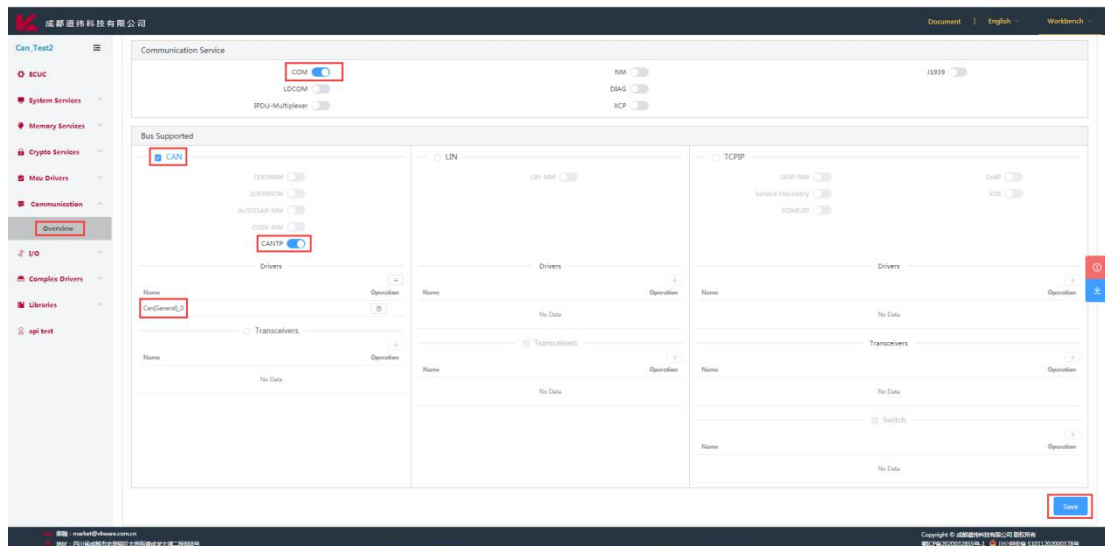


图 4.2-1 通信栈功能配置

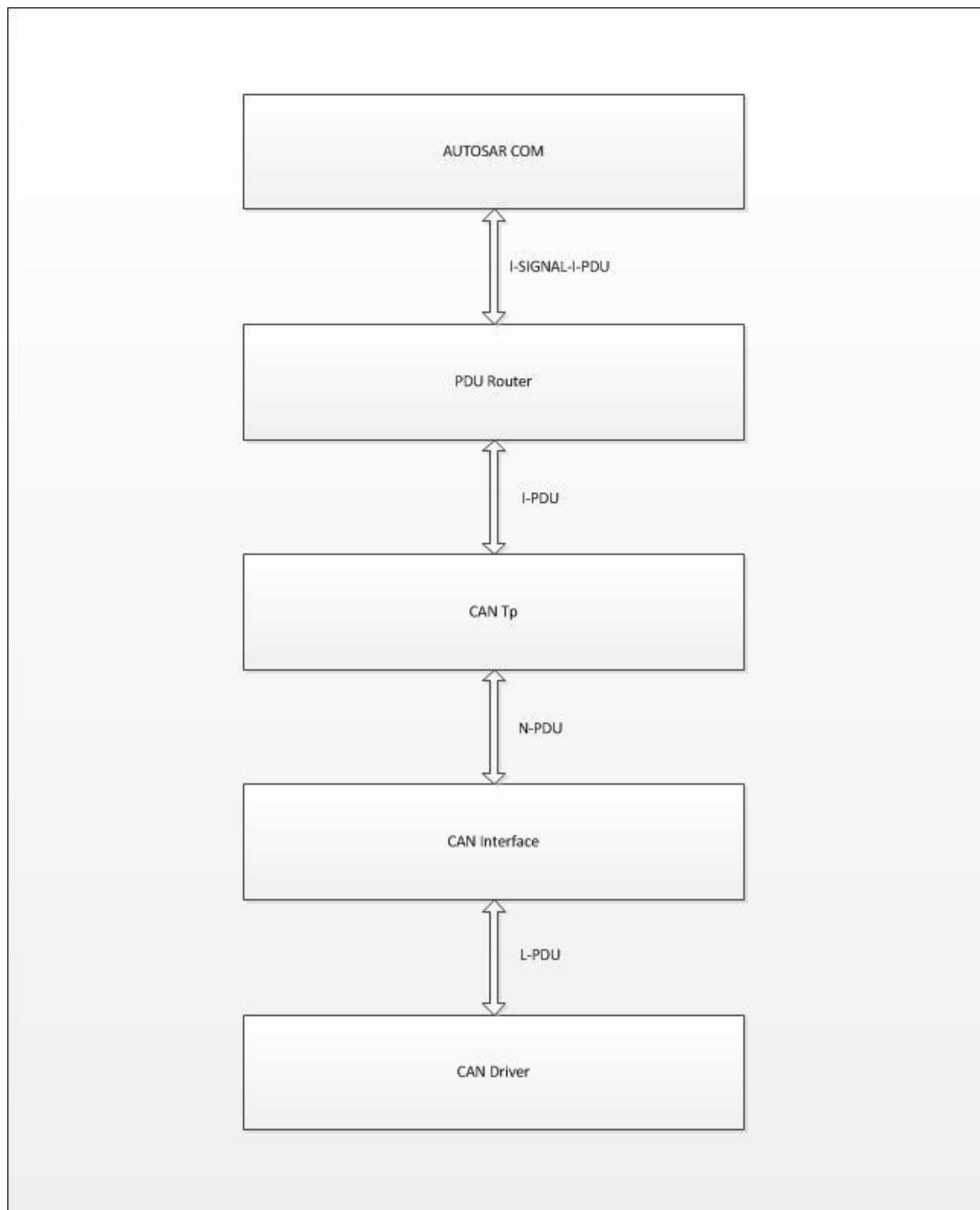


图 4.2-2 PDU 流向示例

4.2.1 配置 EcuC

前提：

1、根据 Autosar 方法学，通信栈中所有使用到的 PDU 在 EcuC 中进行配置。

假设：

1、所有 PDU 不含 metaData。



添加 PDU:

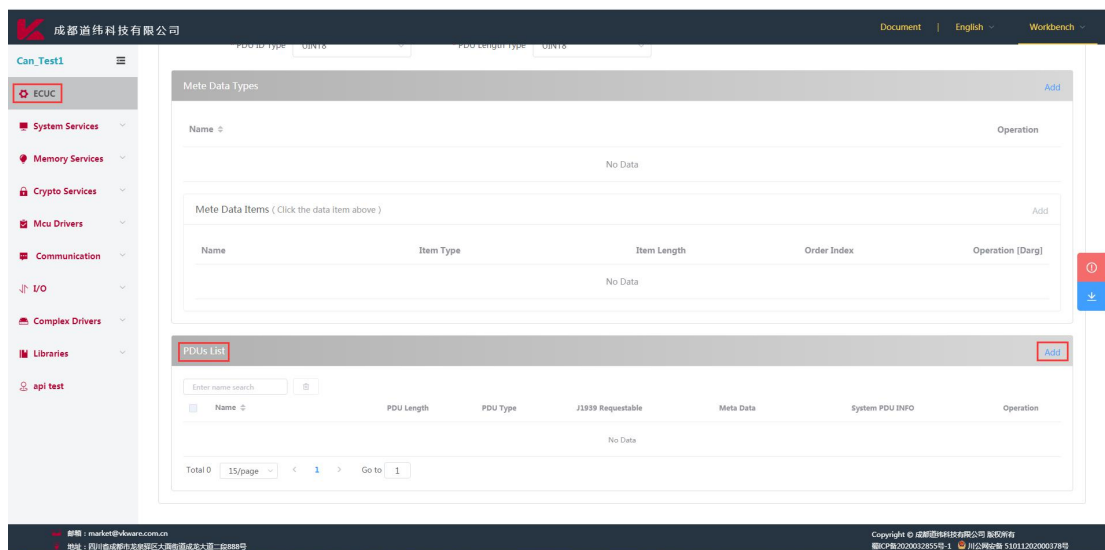


图 4.2-3 EcuC 中添加 PDU

配置参数:

Name	PDU Length	PDU Type	其它
LPDU_rx_360	8	CAN	默认
NSDU_rx_360	10	空	默认
IPDU_rx_360	10	空	默认
LPDU_tx_368	8	CAN	默认
IPDU_tx_361	10	空	默认
NSDU_tx_361	10	空	默认
LPDU_tx_361	8	CAN	默认
LPDU_rx_369	8	CAN	默认

4.2.2 抽取 Can 驱动适配的配置

同 4.1.2。



4.2.3 配置 CanIf

前提：

前提：

1、Can 驱动适已正确配置。

假设：

1、接收 canId 为 0x360、0x369。

2、发送 canId 为 0x361、0x368。

配置 General：

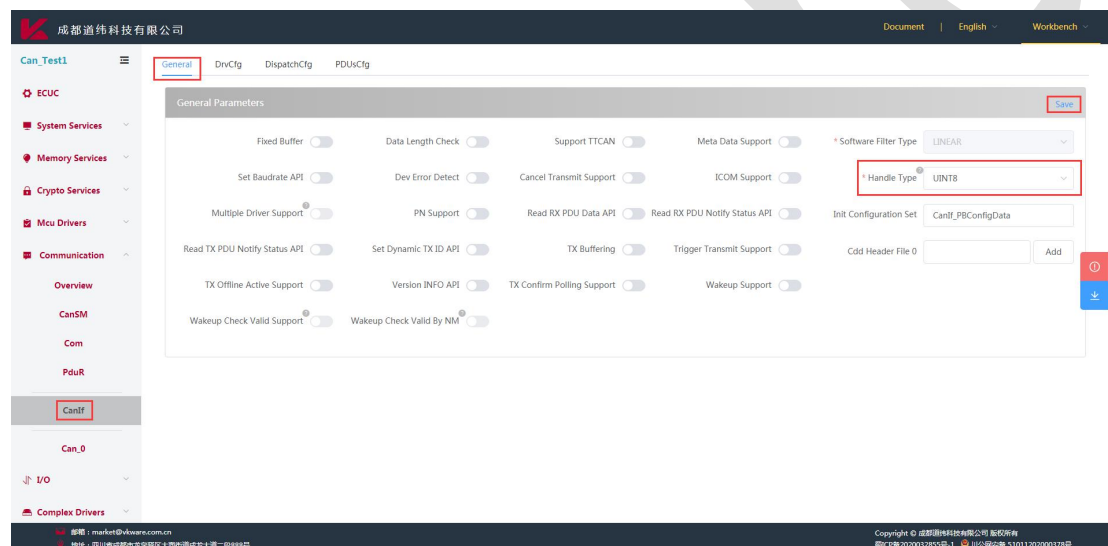


图 4.2-4 配置 General

配置参数：

Handle Type	其它
UINT8	默认



配置 DrvCfg:

关联 Can 驱动的 Controller:

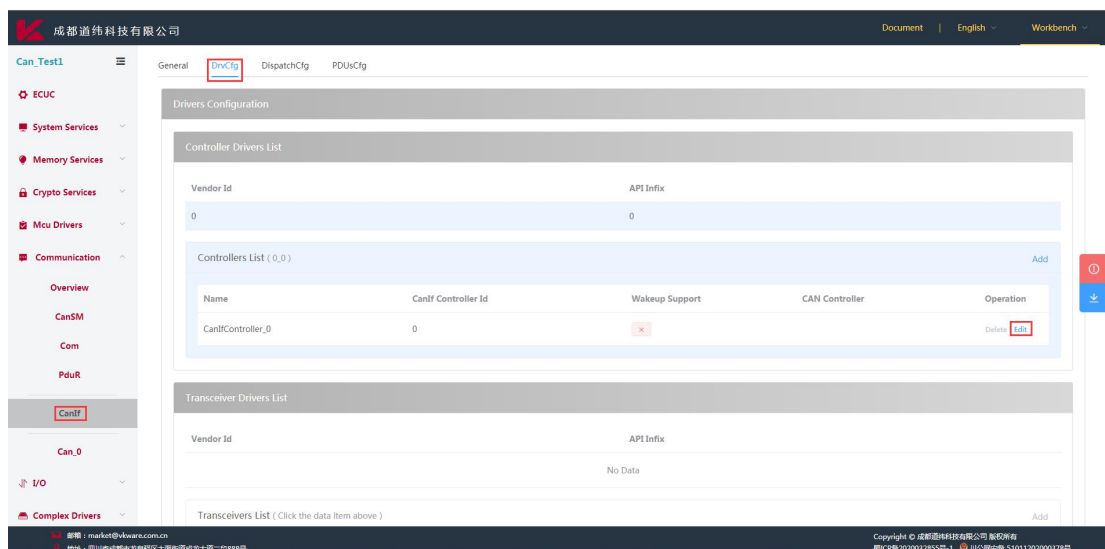


图 4.2-5 关联 Controller

配置参数:

Name	CAN Controller
默认	CanController_0

说明: 此项配置由系统默认为每一个 CanDriver 生成一个 Controller, 但需要关联, CanDriver 中的其它 Controller 需要通过点击 “Add” 进行添加和关联。



添加一个 HRH:

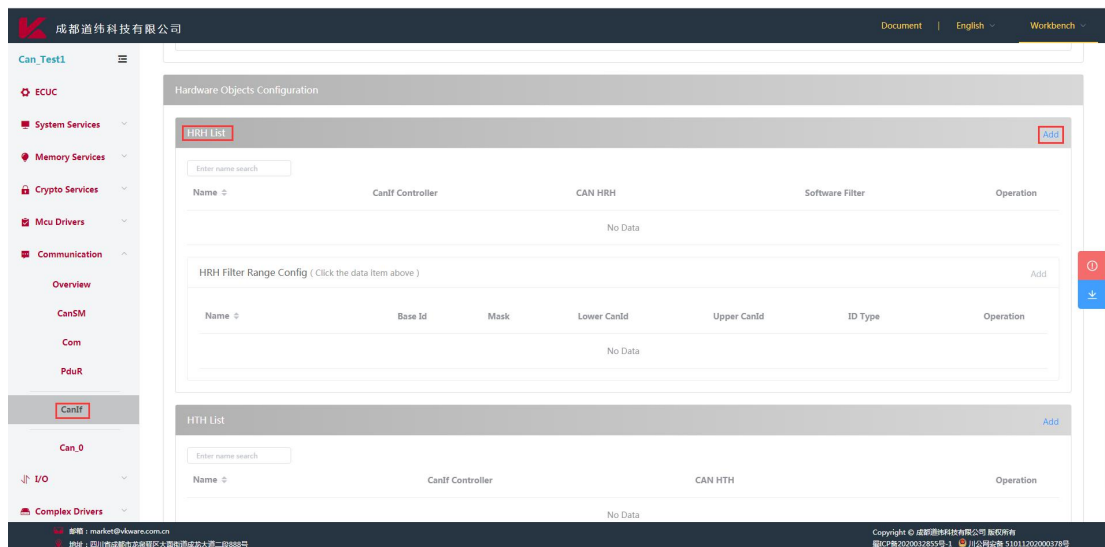


图 4.2-6 添加 HRH

配置参数:

Name	CanIf Controller	CAN HRH	Software Filter
默认	CanIfController_0	CanHardwareObject_rx	关闭

添加一个 HTH:

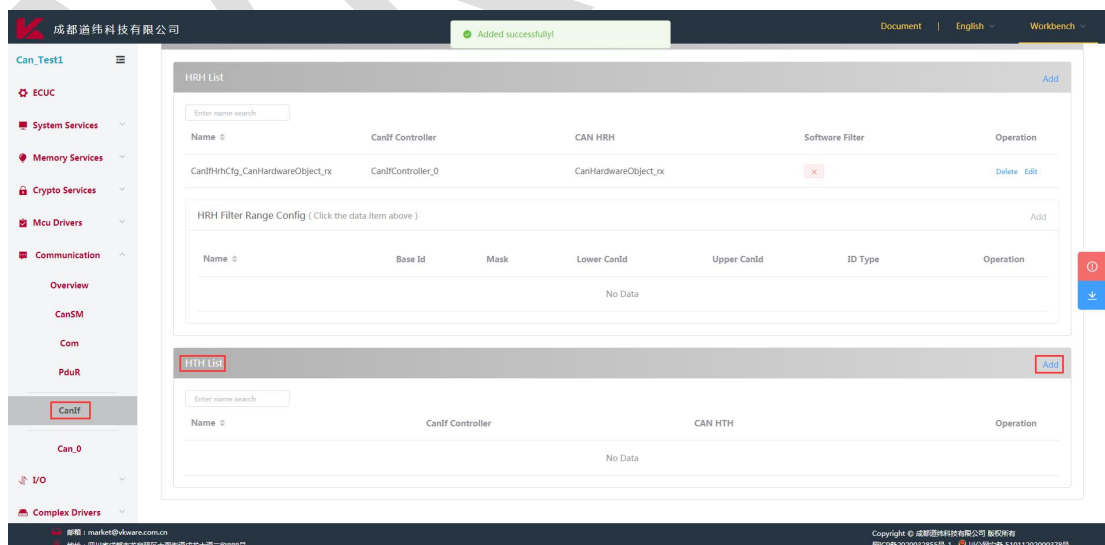


图 4.2-7 添加 HTH

配置参数:



Name	CanIf Controller	CAN HTH
默认	CanIfController_0	CanHardwareObject_tx

配置 DispatchCfg:

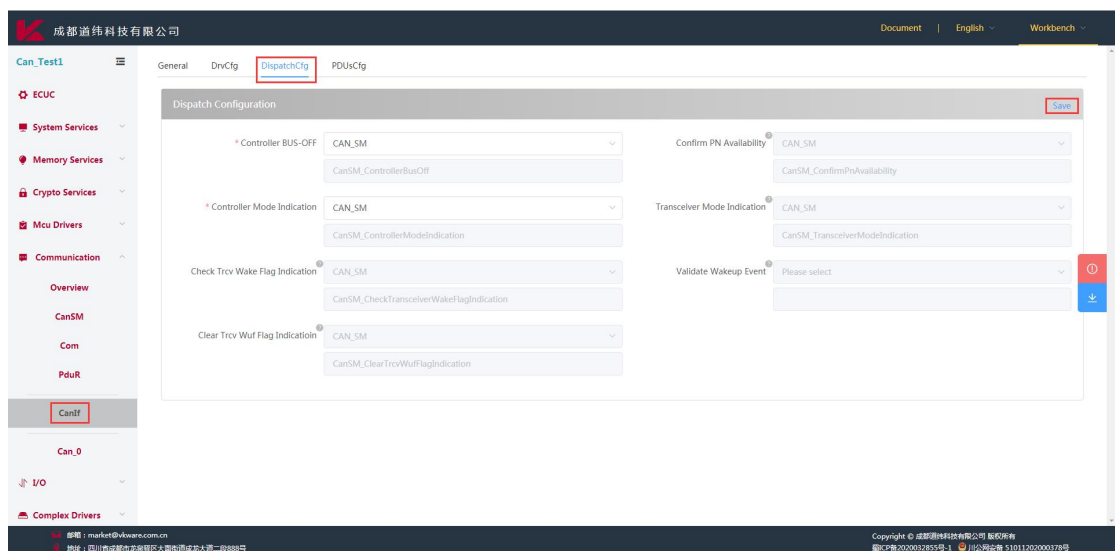


图 4.2-8 配置 DispatchCfg

配置参数:

Controller BUS-OFF	Controller Mode Indication
CAN_SM	CAN_SM



配置 PDUscfg:

配置接收和发送 PDU 的数量限制

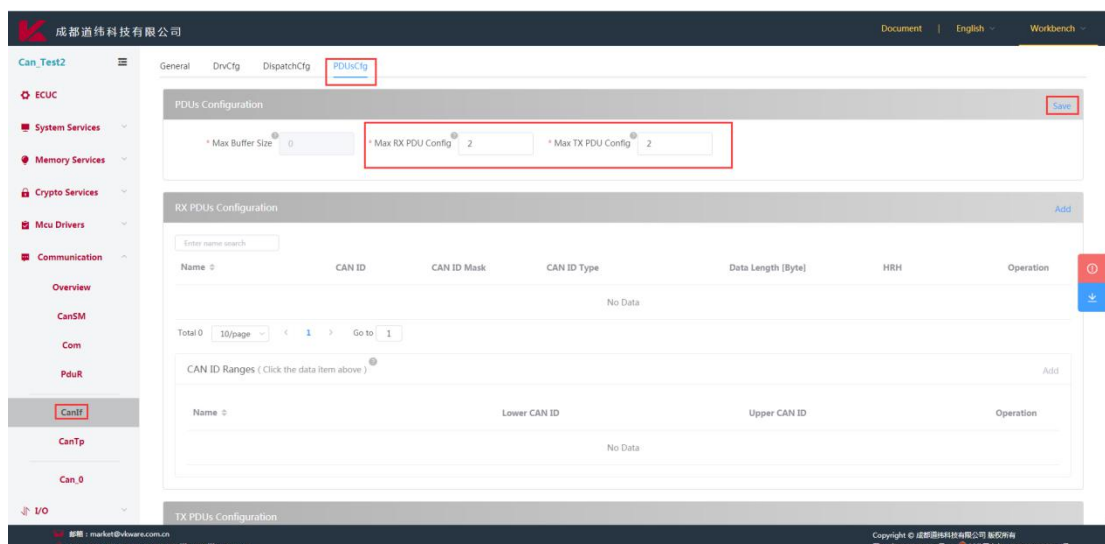


图 4.2-9 配置 PDUscfg

配置参数:

Max RX PDU Config	Max TX PDU Config
2	2

添加两个接收 PDU:

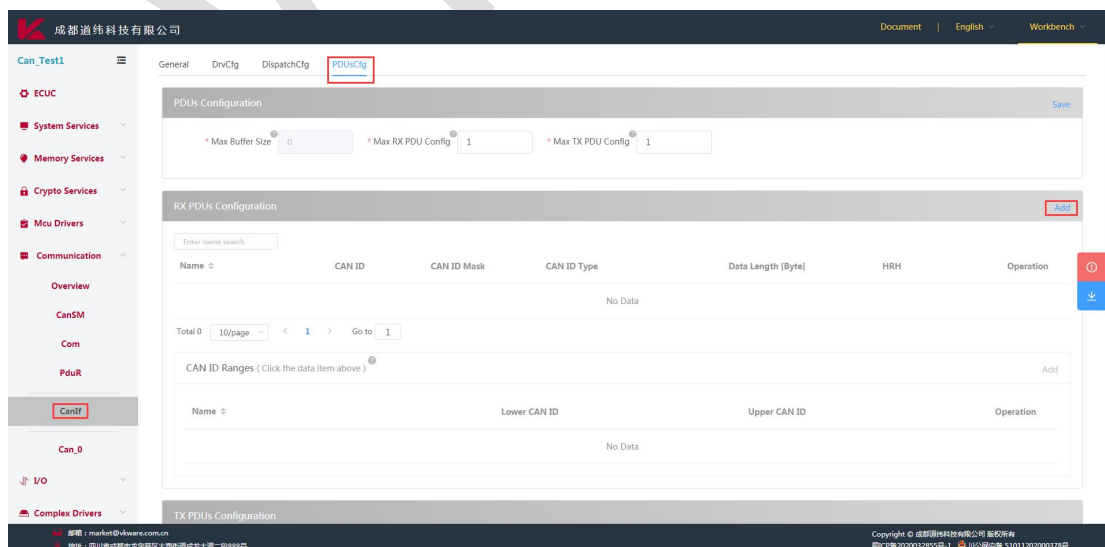




图 4.2-10 添加接收 PDU

配置参数：

Name	HRH	PDU Ref	CAN ID Type	CAN ID	CAN ID Mask	RX Indication UL	其它
默认	CanIfHrhCfg_CanHardwareObject_rx	LPDU_rx_360	STANDARD_NO_FD_CAN	360	7ff	CAN_TP	默认
默认	CanIfHrhCfg_CanHardwareObject_rx	LPDU_rx_369	STANDARD_NO_FD_CAN	369	7ff	CAN_TP	默认

添加一个发送缓存：

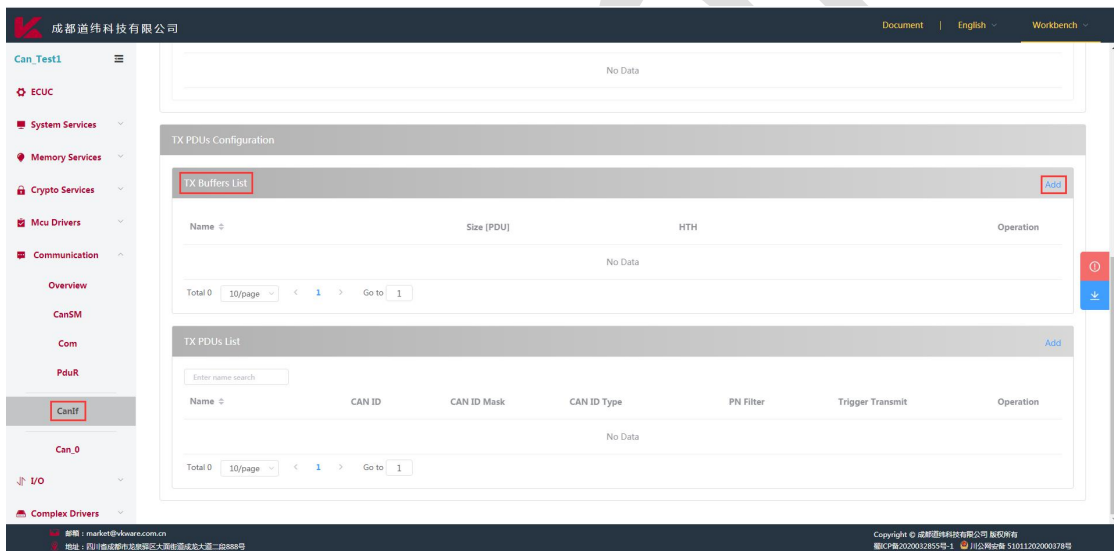


图 4.2-11 添加发送缓存

配置参数：

Name	HTH
默认	CanIfHrhCfg_CanHardwareObject_tx



添加两个发送 PDU:

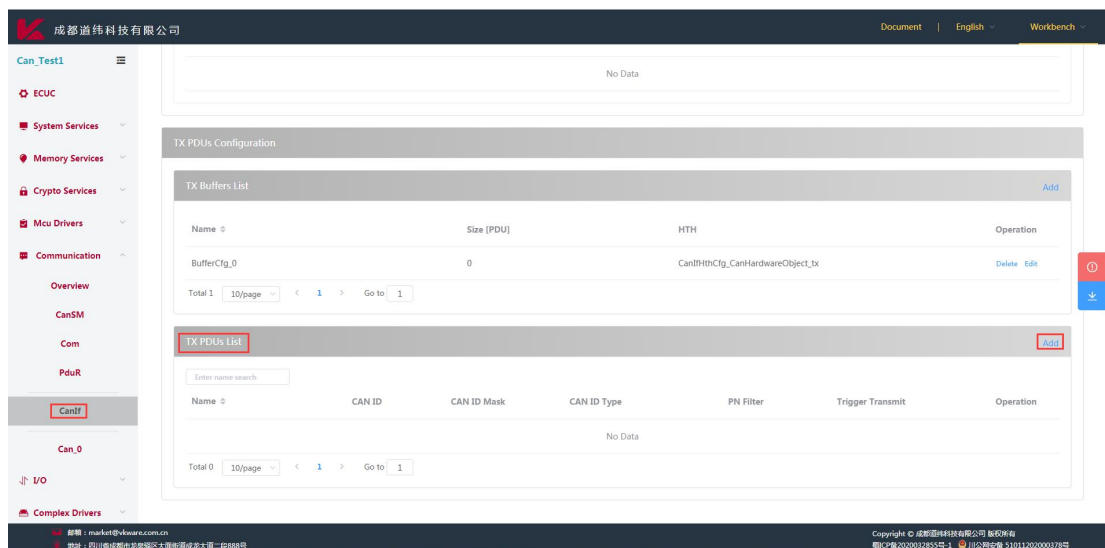


图 4.2-12 添加发送 PDU

配置参数:

Name	TX PDU Buffer Ref	PDU Ref	CAN ID Type	CAN ID	CAN ID Mask	User TX Confirmation UL	其它
默认	BufferCfg_0	LPDU_tx_361	STANDARD_CAN	361	7ff	CAN_TP	默认
默认	BufferCfg_0	LPDU_tx_368	STANDARD_CAN	368	7ff	CAN_TP	默认

4.2.4 配置 CanTp

前提:

1、无。

假设:

- 1、寻址方式为标准寻址。
- 2、发送和接收在不同通道。



配置 General:

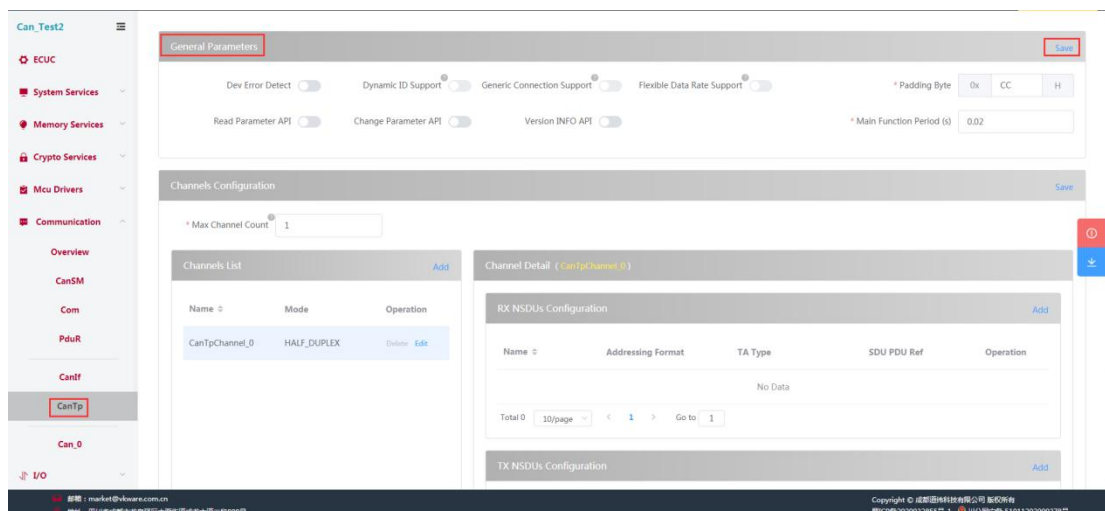


图 4.2-13 配置 General

配置参数:

默认。

配置 Channels List:

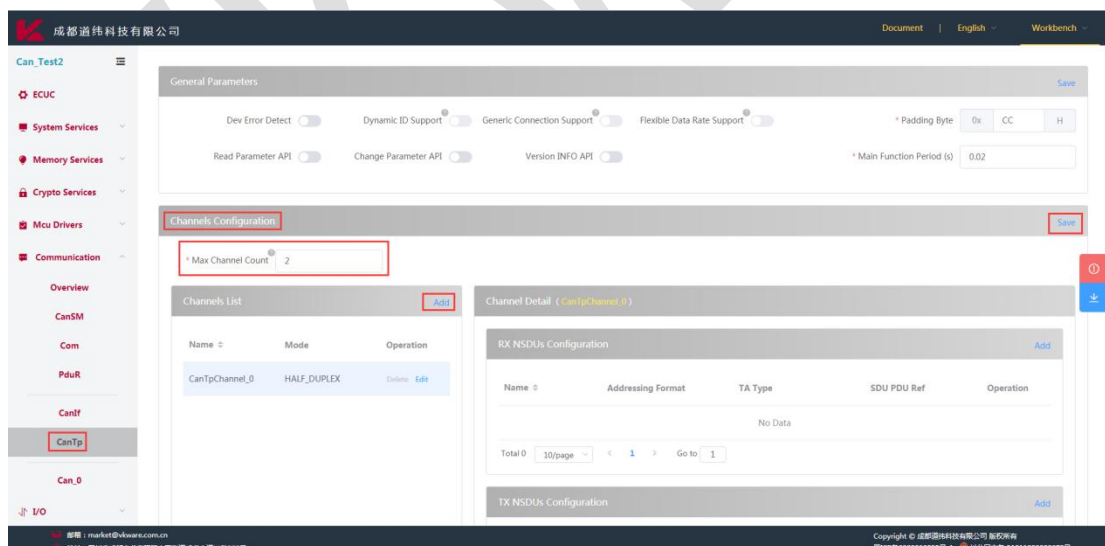


图 4.2-14 配置 Channels List

配置参数:

Max Channel Count: 2



Name	Mode
CanTpChannel_rx	HALF_DUPLEX
CanTpChannel_tx	HALF_DUPLEX

配置 CanTpChannel_rx 下的 RX NSDUs Configuration:

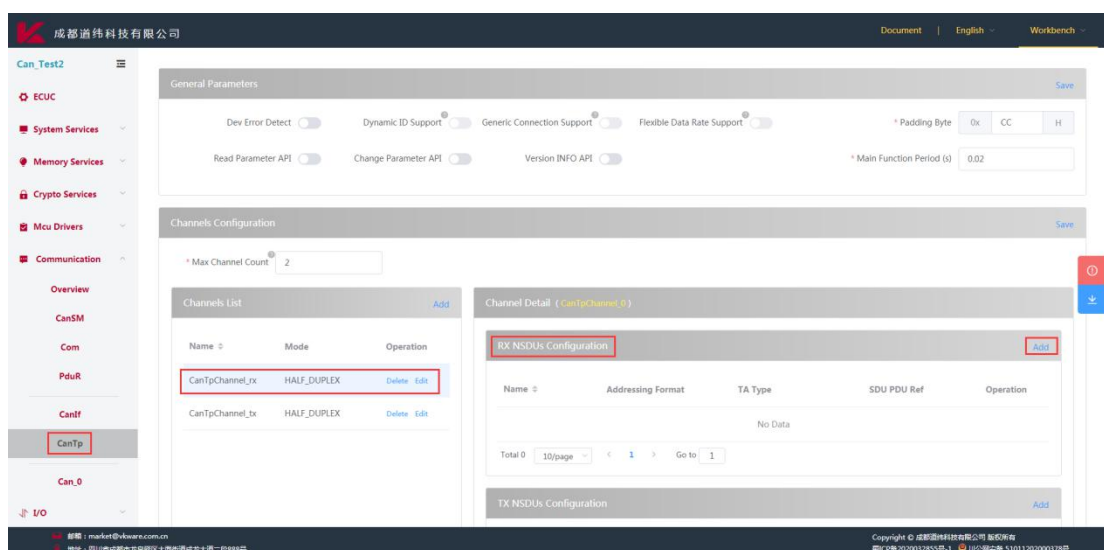


图 4.2-15 配置 RX NSDU

配置参数:

Name	默认
SDU PDU Ref	NSDU_rx_360
RX NPDU PDU Ref	LPDU_rx_360
Addressing Format	STANDARD
TA Type	PHYSICAL
TX FC NPDU PDU Ref	NPDU_tx_368
其它	默认



配置 CanTpChannel_tx 下的 TX NSDUs Configuration:

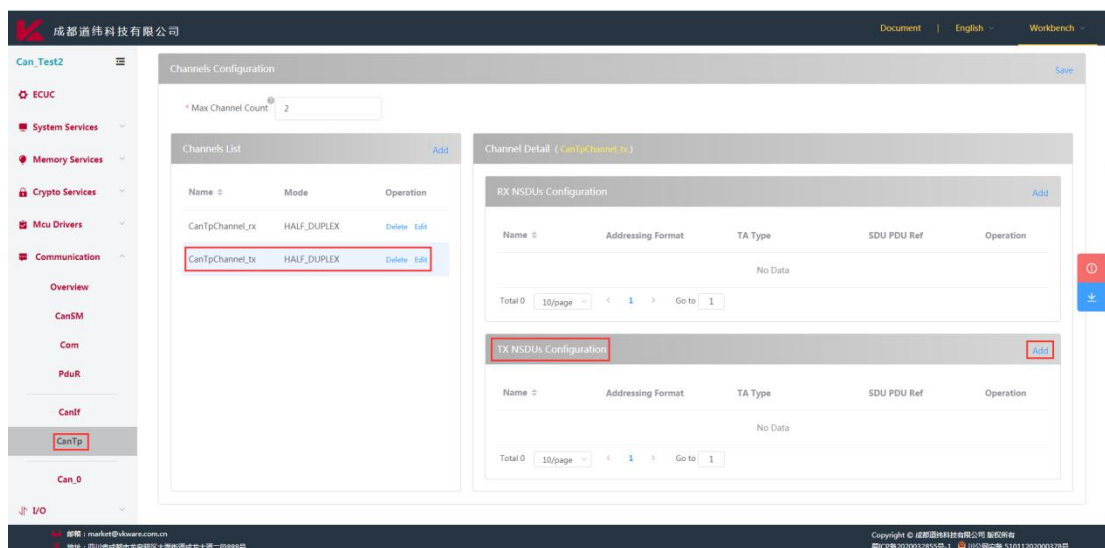


图 4.2-16 配置 TX NSDU

配置参数:

Name	默认
SDU PDU Ref	NSDU_tx_361
TX NPDU PDU Ref	LPDU_tx_361
Addressing Format	STANDARD
TA Type	PHYSICAL
TX FC NPDU PDU Ref	NPDU_rx_369
其它	默认

4.2.5 配置 Com

前提:

无。

假设:

1、接收与发送方向均有一个 PDU。



- 2、接收 PDU 关联一个信号。
- 3、发送 PDU 关联一个信号。

配置 General:

默认

配置 SignalCfg:

添加一个接收方向信号:

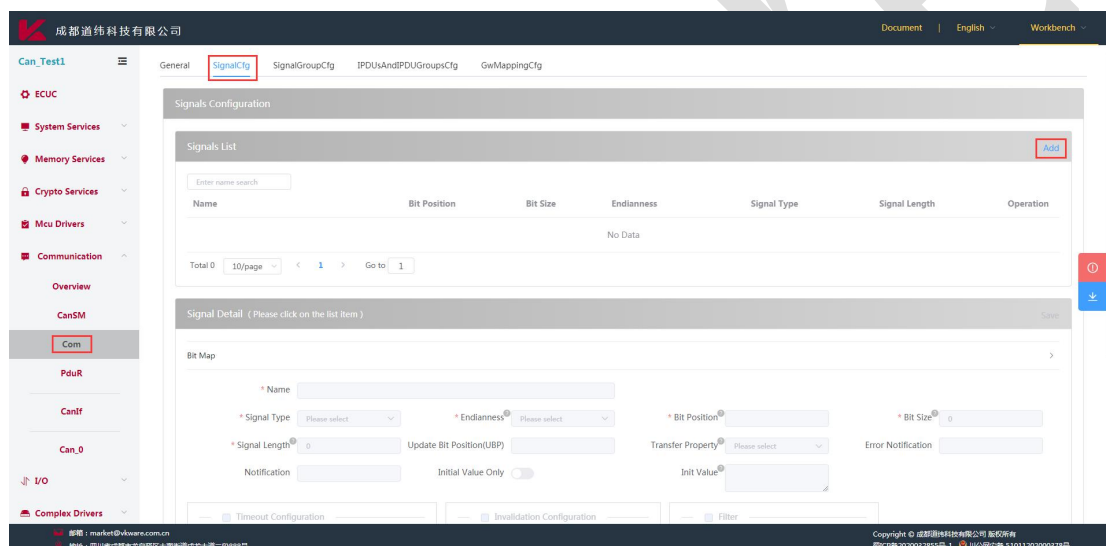


图 4.2-17 添加接收方向信号

配置参数:

Name	Signal_rx_360_vehicleSpeed
Signal Type	UINT16
Endianness	BIG_ENDIAN
Bit Position	8
Bit Size	12
Notification	Rte_ComCbk_Signal_rx_360_vehicleSpeed_RxNotification



Init Value	0
其它	默认

同上添加一个发送方向信号：

配置参数：

Name	Signal_tx_361_ambientTemperature
Signal Type	UINT8
Endianness	BIG_ENDIAN
Bit Position	0
Bit Size	8
Transfer Property	PENDING
Notification	Rte_ComCbk_Signal_tx_361_ambientTemperature_TxNotification
Init Value	0
其它	默认



配置 IPDUsAndIPDUGroupsCfg:

配置 IPDU 组和 IPDU 的数量限制

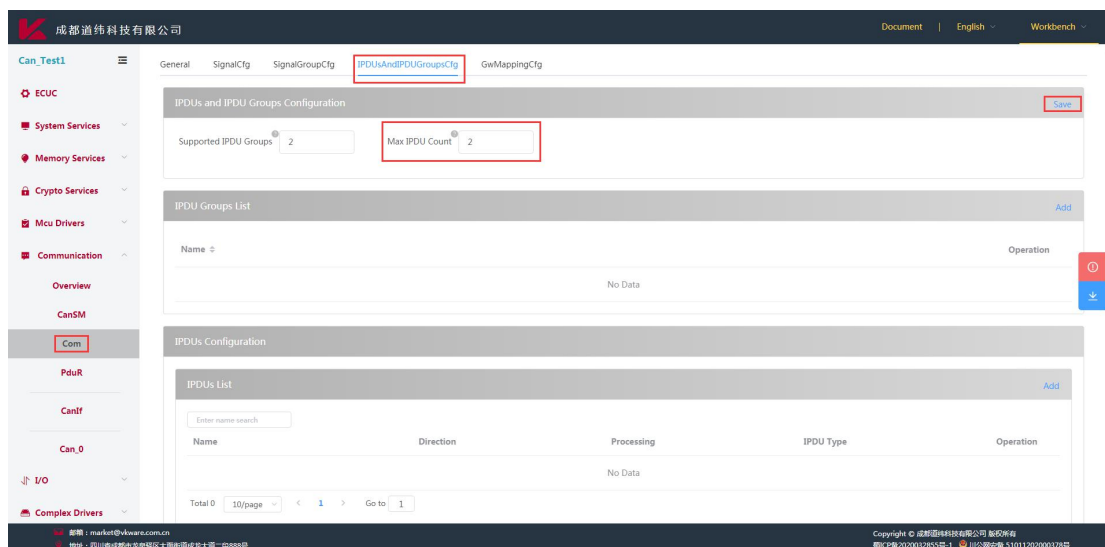


图 4.2-18 配置 IPDUsAndIPDUGroupsCfg

配置参数:

Supported IPDU Groups	Max IPDU Count
默认	2

添加一个接收 IPDU:

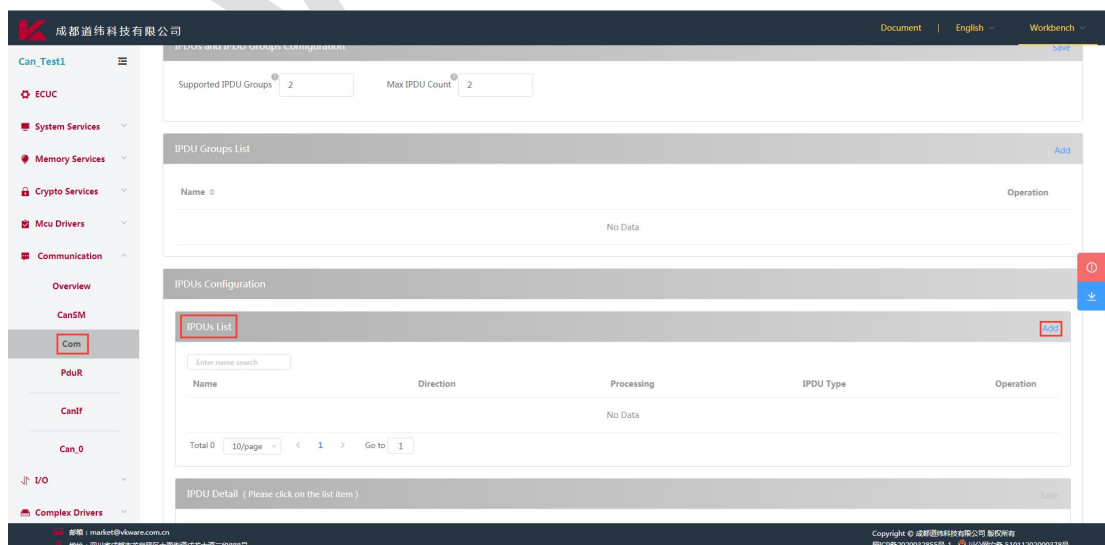




图 4.2-19 添加接收 IPDU

配置参数:

Name	IPdu_rx_360
PDU Ref	IPDU_rx_360
Direction	RECEIVE
Processing	IMMEDIATE
IPDU Type	TP
Signal Refs	Signal_rx_360_vehicleSpeed
其它	默认

同上添加一个发送 IPDU:

配置参数:

Name	IPdu_tx_361
PDU Ref	IPDU_tx_361
Direction	SEND
Processing	DEFERRED
IPDU Type	TP
Signal Refs	Signal_tx_361_ambientTemperature
TRUE Mode	勾选
Mode	PERIODIC
Time Period	0.5
其它	默认



4.2.6 配置 PDUR

前提：

1、无。

假设：

1、不使用 Zero Cost 功能。

配置 General:

默认

配置 BswModule:

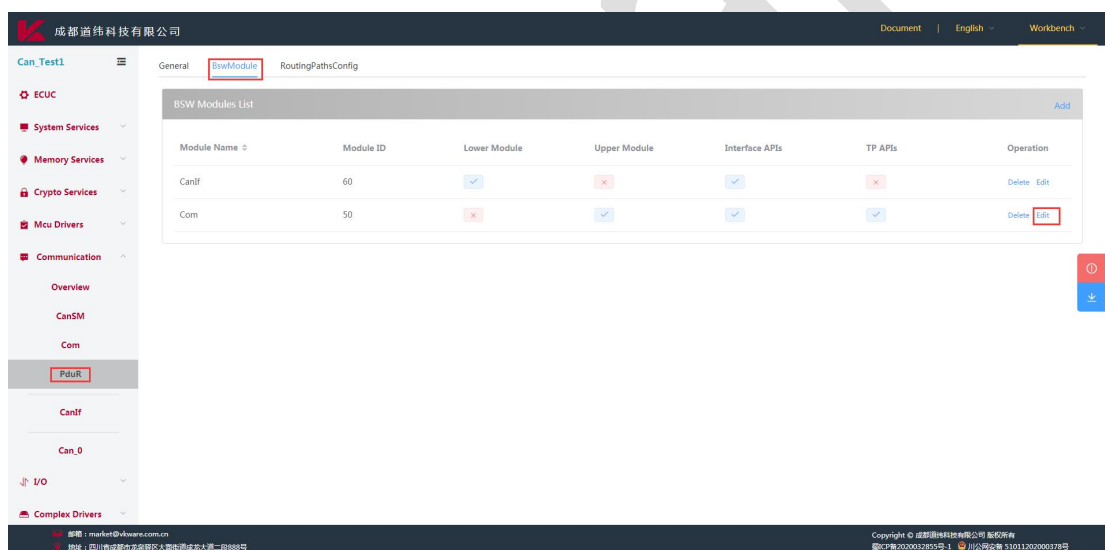


图 4.1-20 配置 BswModule

配置参数：

BSW Module	COM
Communication Interface	不勾选
Transport Protocol	勾选
其它	默认



配置 RoutingPathsConfig:

配置路由路径的数量限制:

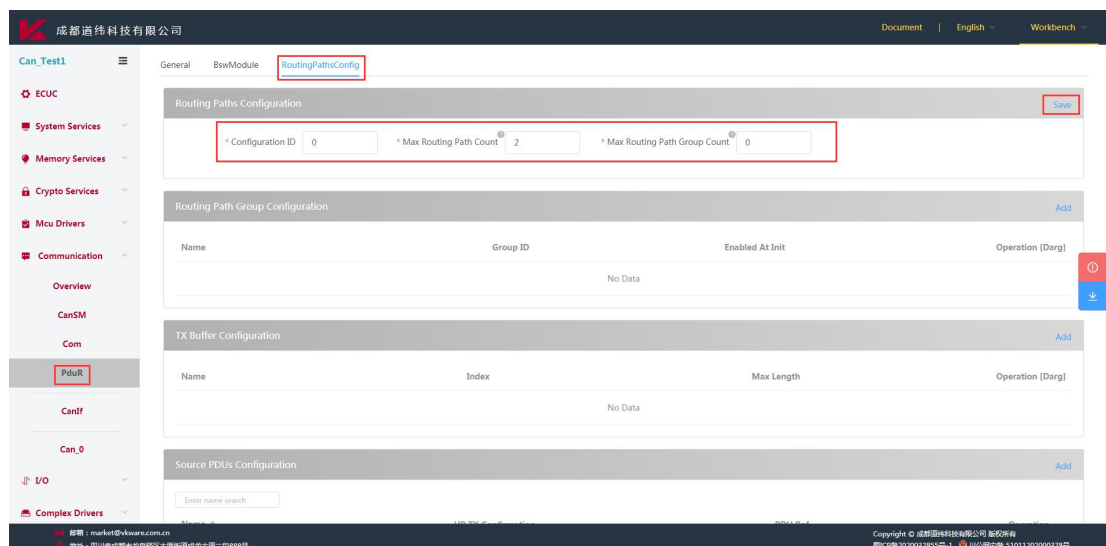


图 4.2-21 配置 RoutingPathsConfig

配置参数:

Configuration ID	Max Routing Path Count	Max Routing Path Group Count
0	2	0

添加两个源 PDU:

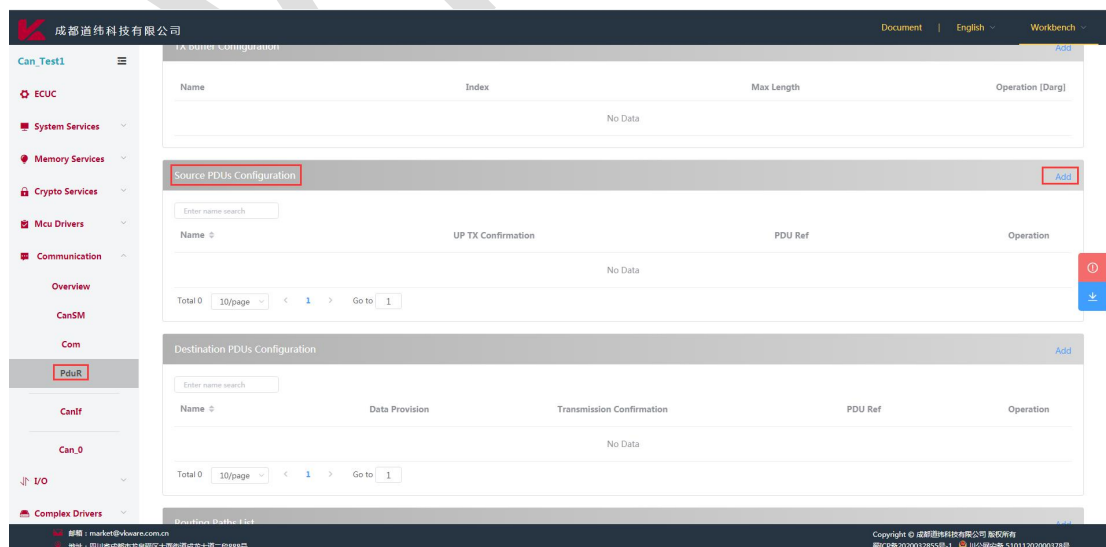




图 4.2-22 添加源 PDU

配置参数：

Name	Source Pdu Block Size	PDU Ref	UP TX Confirmation
默认	0	NSDU_rx_360	关闭
默认	0	IPDU_tx_361	打开

添加两个目的 PDU：

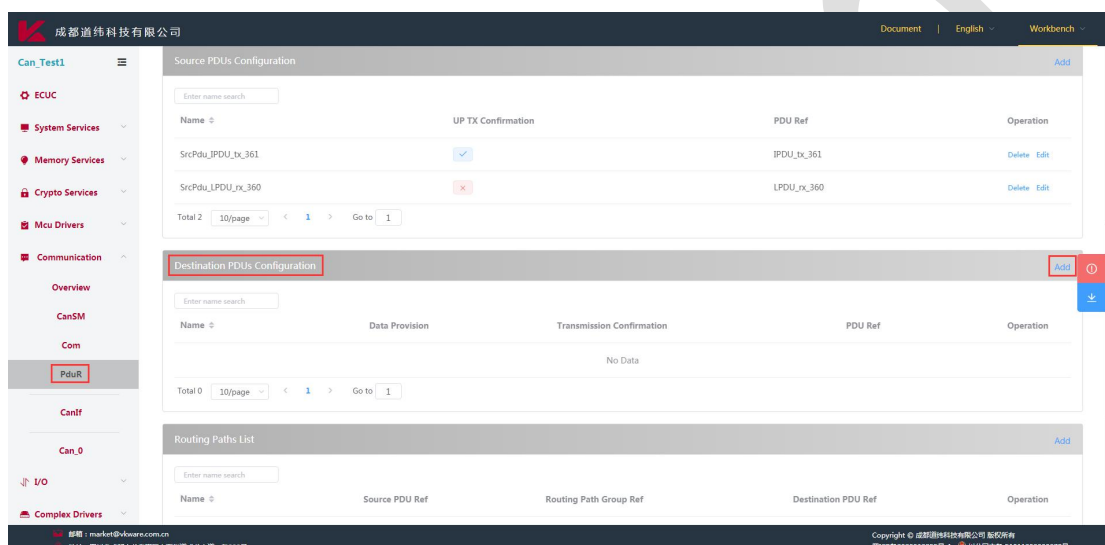


图 4.2-23 添加目的 PDU

配置参数：

Name	PDU Ref	Transmission Confirmation	其它
默认	IPDU_rx_360	关闭	默认
默认	NSDU_tx_361	打开	默认



配置 PDU 路径:

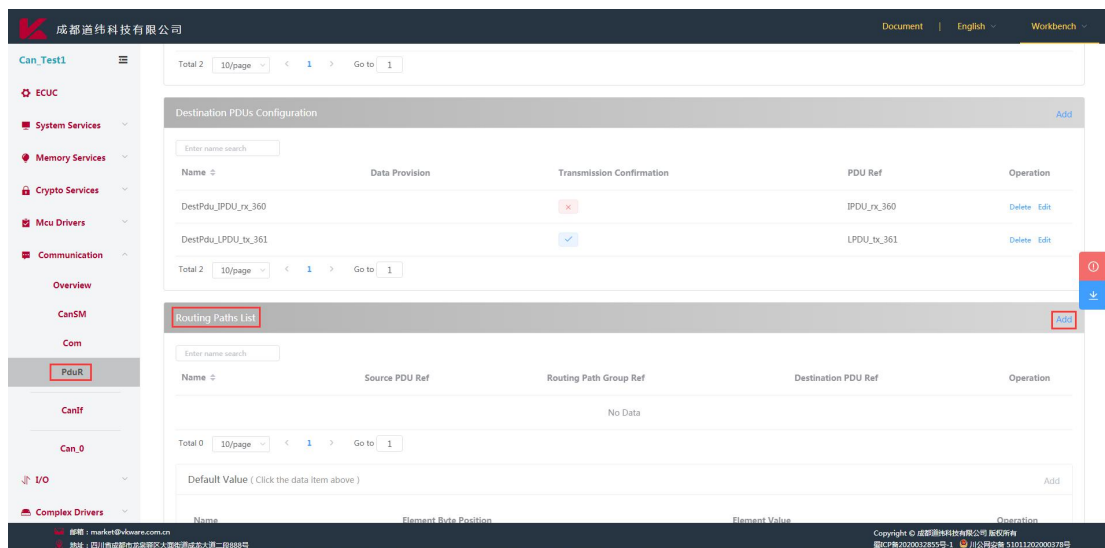


图 4.2-24 配置 PDU 路径

配置参数:

Name	Source PDU Ref	Destination PDU Ref	其它
默认	SrcPdu_NSDU_rx_360	DestPdu_IPDU_rx_360	默认
默认	SrcPdu_IPDU_tx_361	DestPdu_NSDU_tx_361	默认

4.2.7 配置 ComM

同 4.1.6。

4.2.8 配置 CanSm

同 4.1.7。



5.0 配置参数介绍

详见各模块 Autosar SWS 文档说明。

VKware



6.0 API 接口

详见各模块 Autosar SWS 文档说明。

VKware