iSmartWays Technology Inc.

Mocar V2X SDK user manual

2019-3-26

Contents

1. Brief

Mocar V2X SDK is designed by iSmartWays Technology Inc. for V2X application development. The applications developed by the SDK are adapted to run on the Mocar OBE devices and the Mocar RSE devices.

It is straightforward and quick to build V2X applications, including V2V, V2I, V2P, V2N and others, with Mocar SDK.

Features:

- different communication: DSRC, C-V2X
- different data-sets: SAE J2735 – 2016, T/CSAE 53 - 2017
- simple and clear API: send and receive data

2. Contents

category	items	comments
data sheet files	head files includes definition of message according to SAE J2735 – 2016: BSM, RSA, SPAT, MAP, SSM, SRM, TIM	
	head files includes definition of message according to T/CSAE 53 – 2017: BSM, SPAT, MAP, RSI, RSM	
API files	head files to send API, receive API	
library files	C library	mocarv2x.so

3. Usage

Steps to use Mocar V2X SDK:

- Include all head files in your program, including data sheet and API files;
- Add library files into makefile in your program;
- Call initialization API to initialize Mocar SDK;
- Fill data structure following definition in data sheet files;

- Send out packet via send API;
- Receive and handle packet data via receive API.

4. License

4.1 License on development environment

- Execute application reg32 (on 32-bit Linux OS) or reg64 (on 64-bit Linux OS) to generate mk_reg.dat

  Example:

  mocar@ubuntu:~/mocar $ ./reg

  generate reg file success! reg file name: mk_reg.dat

- Email the mk_reg.dat file to support@ismartways.com to get the license file
- Put license file in the same path as Mocar C library

  mocar@ubuntu:~/mocar $ scp root@192.168.0.100:/mocar_license .

  mocar@ubuntu:~/mocar $ ls -l

  total 12

  -rwxrwxr-x 1 mocar mocar 3523 Dec 18 19:28 mocarv2x.so

  -rwxrwxr-x 1 mocar mocar 1182 Dec 18 19:28 mocar_license

4.2 License on device

- Copy the license file into the same path as Mocar C library on the device, such as RSE or OBE

5. Samples

- Under the samples directory is the send and receive code for each message set, Users can refer to it to develop programs.
- Users can compile these samples under the build directory, the bin file is generated in the bin directory, Users can run these bin files to test send and receive of message sets, include BSM , MAP, SPAT, RSA, TIM, SRM, SSM, RTCM.
- Users can follow these steps to run bin files.

  1. Copy bin file to mocar device directory, for example /var.
  2. Copy all SDK so lib(in SDK lib directory) to /usr/lib directory.
  3. Copy mocar_log.conf file to log path init by mde_v2x_init function.

4. Enter the /var directory, run ./bsm…., if don't have permission , chmod 777 bsm.

6. Debug

5.1 Trace log

- Initialize log file with Mocar SDK API in your program, for example bsm.log

  API to initialize log:

*int mde_v2x_init(const char * log_cfg_path, const char* module_name)* *Description:* *This API initializes v2x SDK and log system, it should be called before sending or receiving v2x message.* *Parameters:* *log_cfg_path : name and path of log file* *module_name: module name to genareate log file.* *Return Value:* *0 - success* *others - failure*

- Monitor log file on your device with command tail, like this:

root@ibox:~/mocar $ tail -f /usr/local/bsm.log

- If user want to add new module or change module name, for example custom module, if you want to check sdk log, follow these step:

  1. Init, mde_v2x_init("/usr/local/mocar_log.conf", "custom*").*
  2. Add new module in mocar_log.conf.

     custom.INFO $mocar_output, "/usr/local/custom.log"; normal

     custom.ERROR "/usr/local/custom.core.log", 5MB * 1; normal
  3. Then you can check log in /usr/local/custom.log.