

ECU-Bus User Guide

by: GC AMP GPIS AE Team

1. Introduction

ECU-Bus is a general ECU Toolkits, which contains UDS Tester, S32K relative security tool(Mx matrix generator) and so on. Current version only support PEAK(CAN/LIN tool).

The tool based on electron(<https://www.electronjs.org/>),electron is easy to build cross-platform desktop apps with JavaScript, HTML, and CSS.

Tool link:

GitHub: <https://github.com/frankie-zeng/ECUBus>

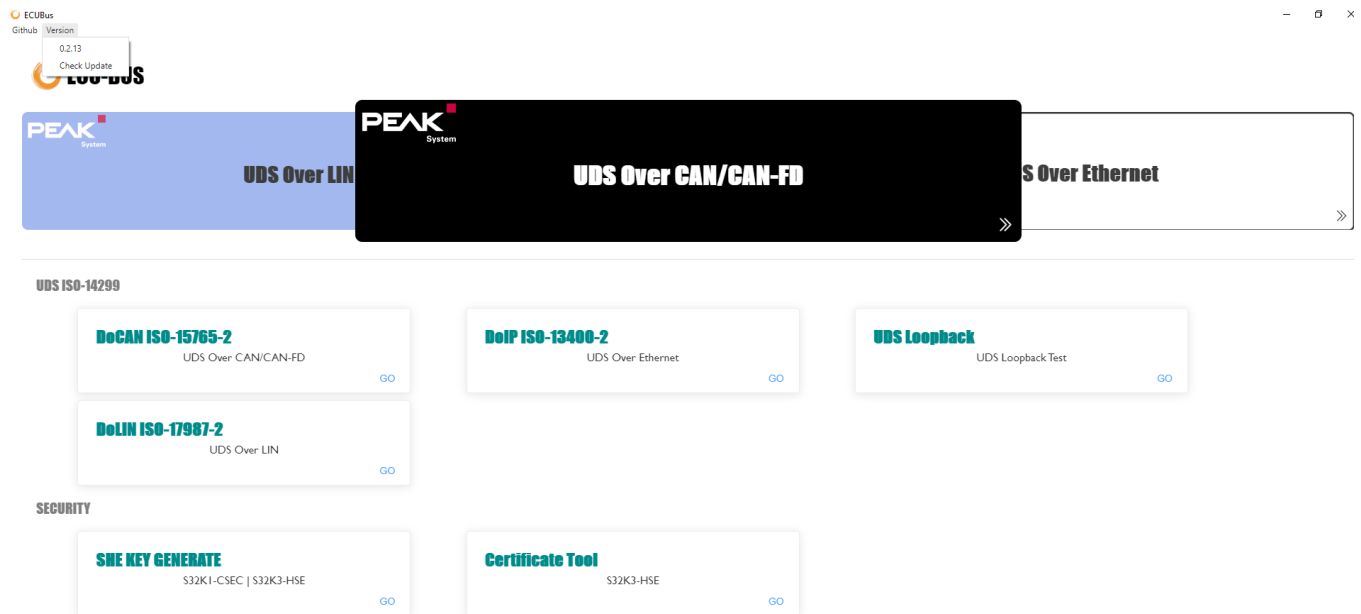
Contents

- 1. Introduction 1
- 2. Open ECU-BUS 2
- 3. UDSONCAN..... 2
- 4. UDSONLIN 5
- 5. Key features..... 6
 - 5.1. How to choose update file 6
 - 5.2. How to modify calculate CRC method 7
 - 5.3. How to modify access algorithm 9
 - 5.4. How to config different TX message ID 9
 - 5.5. How to config multi-schedule table..... 9
 - 5.6. How to export update process 10
- 6. UDSONDoIP..... 11
- 7. ISO-14299 UDS Feature 11
 - 7.1. Key Feature 11
 - 7.2. Loopback UDS Tester 18
 - 7.3. DoCAN UDS Tester..... 19
 - 7.4. DoIP UDS Tester..... 20
- 8. References 21



2. Open ECU-BUS

Open ECU-BUS as following:



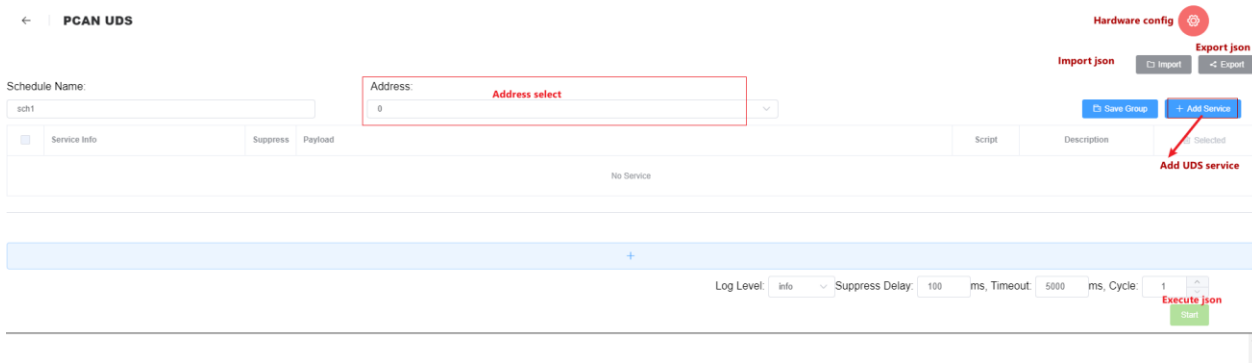
The version is display ECU-BUS version and check update. The ECUBUS page including UDS over LIN/CAN/Ethernet.

3. UDSONCAN

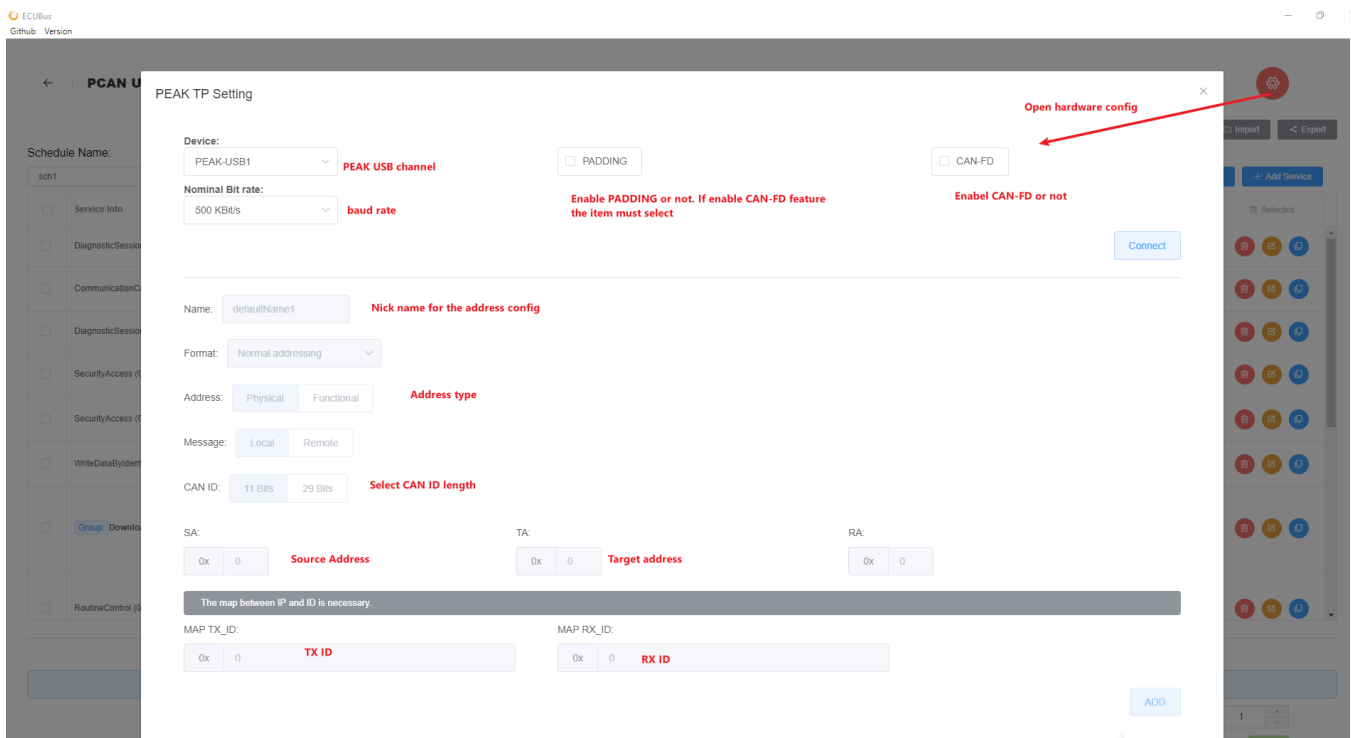
Use UDSONCAN for update APP as following step:

1. Open UDSONCAN
2. Open hardware button and config hardware information. E.g., baud rate, CAN or CAN FD, select CAN ID length, set TX & RX CAN ID, fill SA and TA address, add the address information in GUI.
3. Import update process or create process
4. Select schedule address(TX & RX message ID over nickname)
5. Click start button

Open UDSONCAN



Open hardware config as following:



As following table is an example for how to config hardware information.

✓ CAN

PEAK TP Setting

Device:	PEAK-USB1	<input type="checkbox"/> PADDING	<input type="checkbox"/> CAN-FD
Nominal Bit rate:	500 KBit/s	Connect HW	
			<input type="button" value="Connect"/>

✓ CAN-FD

PEAK TP Setting

Device: **Baud rate & PEAK-USB channel**

Nominal Bit rate:

PADDING **Data bit rate**

Data Bit rate:

CAN-FD **Payload length**

TLC:

After connect hardware

Device:

Nominal Bit rate:

PADDING **Data Bit rate:**

CAN-FD **TLC:**

Name: **Nick name**

Format:

Address: Physical Functional **UDS address type**

Message: Local Remote

CAN ID: 11 Bits 29 Bits **CAN ID length. If select 11bits the SA and TA have not used. Just fill different value.**

SA: **Source address**

TA: **Target address**

RA:

The map between IP and ID is necessary.

MAP TX_ID: **GUI TX ID is 0x784**

MAP RX_ID: **GUI RX ID is 0x7F0**

Add the address

Name	Send ID	Receive ID	Mapped	SA	TA	TA Type	RA	Format	Action
No Data									

After Add address

The map between IP and ID is necessary.

MAP TX_ID:

MAP RX_ID:

Name	Send ID	Receive ID	Mapped	SA	TA	TA Type	RA	Format	Action
1	defaultName1	1924	2032	<input checked="" type="checkbox"/>	0X11	0X22	Physical	0X0	Normal addressing <input type="button" value="Delete"/>

Import file

Github Version

← PCAN UDS

Schedule Name: sch1 Address: Address

Import file

Import Export Save Group Add Service

Service Info	Suppress	Payload	Script	Description	Selected
DiagnosticSessionControl (0X10)	<input type="radio"/>	subFunction : 3			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
CommunicationControl (0X28)	<input type="radio"/>	subFunction : 3 communicationType : 03			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
DiagnosticSessionControl (0X10)	<input type="radio"/>	subFunction : 2			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
SecurityAccess (0X27)	<input type="radio"/>	subFunction : 1 AccessData / securityKey :			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
SecurityAccess (0X27)	<input type="radio"/>	subFunction : 2 AccessData / securityKey :			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
WriteDataByIdentifier (0X2e)	<input type="radio"/>	dataIdentifier : f15a dataRecord : 5555			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Group: DownloadFile1		0-Download File : {"name": "C:\\Users\\NXF100351\\Desktop\\CAN_FD_bootloader\\S32K144_CANFD\\flash_api.bin", "size": 1476} 0-dataFormatIdentifier : 00 0-addressAndLengthFormatIdentifier : 44 0-memoryAddress : 1f8010			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
RoutineControl (0X31)	<input type="radio"/>	subFunction : 1 routineIdentifier : 0202 routineControlOptionRecord :			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Select address type(nickname)

← PCAN UDS

Schedule Name: sch1 Address: Address

Select update progress address

SA:17.TA:34

Import Export Save Group Add Service

Service Info	Suppress	Payload	Script	Description	Selected
DiagnosticSessionControl (0X10)	<input type="radio"/>	subFunction : 3			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
CommunicationControl (0X28)	<input type="radio"/>	subFunction : 3 communicationType : 03			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
DiagnosticSessionControl (0X10)	<input type="radio"/>	subFunction : 2			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
SecurityAccess (0X27)	<input type="radio"/>	subFunction : 1 AccessData / securityKey :			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
SecurityAccess (0X27)	<input type="radio"/>	subFunction : 2 AccessData / securityKey :			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
WriteDataByIdentifier (0X2e)	<input type="radio"/>	dataIdentifier : f15a dataRecord : 5555			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Group: DownloadFile1		0-Download File : {"name": "C:\\Users\\NXF100351\\Desktop\\CAN_FD_bootloader\\S32K144_CANFD\\flash_api.bin", "size": 1476} 0-dataFormatIdentifier : 00 0-addressAndLengthFormatIdentifier : 44 0-memoryAddress : 1f8010			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
RoutineControl (0X31)	<input type="radio"/>	subFunction : 1 routineIdentifier : 0202 routineControlOptionRecord :			<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Start execute

+

Log Level: info Suppress Delay: 100 ms, Timeout: 5000 ms, Cycle: 1

Start

4. UDSonLIN

It's similar to UDSonCAN. Please reference the UDSonCAN config.

ECU-Bus User Guide, User guide, Rev. draft v0.1, 10/2020

5. Key features

5.1. How to choose update file

Click download file group and pop sub-window can select user' file. Currently the file format only support binary. The memory address should be fill correct address. The memory address is storage the file data. For auto fill memory size, please don't change the regular expression.

The screenshot shows the PCAN UDS software interface. At the top, there are fields for 'Schedule Name' (sch1) and 'Address' (defaultName1). Below these are buttons for 'Import', 'Export', 'Save Group', and 'Add Service'. A table lists various services with columns for 'Service Info', 'Suppress', 'Payload', 'Script', 'Description', and 'Selected'. The 'DownloadFile1' group is highlighted with a red box. The payload for this group is: `0-Download File : {"name": "C:\Users\NXF10035\Desktop\CAN_FD_bootloader\S32K144_CANFD\flash_api.bin", "size": 1476}`. Other services listed include DiagnosticSessionControl, CommunicationControl, SecurityAccess, and WriteDataByIdentifier.

The screenshot shows the 'Edit Service' dialog for 'DownloadFile1'. It has a 'Download File' section with a 'Choose File' button and a text field containing the file path: `C:\Users\NXF10035\Desktop\CAN_FD_bootloader\S32K144_CANFD\flash_api.bin 0x5c4`. Below this are three input fields: 'addressAndLengthFormatIdentifier' with value `0x 44`, 'memoryAddress' with value `0x 1ff8010` and a red label 'download file memory address', and 'memorySize' with value `len("${Download File}.size.parseInt("${addressAndLengthFormatIdentifier}",16)&0xf)` and a red label 'Auto fill file length'. There is also a 'Service Feature Description' field and a 'Change Group' button at the bottom right.

5.2. How to modify calculate CRC method



Click the script button, we can see the CRC table and how to calculate CRC value. If needed, customer can modify the CRC table.

```

1  const crc16_table = new Uint16Array([ CRC table
2  0x0000, 0x365e, 0x6cbc, 0x5ae2, 0xd978, 0xef26, 0xb5c4, 0x839a,
3  0xff89, 0xc9d7, 0x9335, 0xa56b, 0x26f1, 0x10af, 0x4a4d, 0x7c13,
4  0xb26b, 0x8435, 0xded7, 0xe889, 0x6b13, 0x5d4d, 0x07af, 0x31f1,
5  0x4de2, 0x7bbc, 0x215e, 0x1700, 0x949a, 0xa2c4, 0xf826, 0xce78,
6  0x29af, 0x1ff1, 0x4513, 0x734d, 0xf0d7, 0xc689, 0x9c6b, 0xaa35,
7  0xd626, 0xe078, 0xba9a, 0x8cc4, 0xf5e, 0x3900, 0x63e2, 0x55bc,
8  0x9bc4, 0xad9a, 0xf778, 0xc126, 0x42bc, 0x74e2, 0x2e00, 0x185e,
9  0x644d, 0x5213, 0x08f1, 0x3eaf, 0xbd35, 0x8b6b, 0xd189, 0xe7d7,
10 0x535e, 0x6500, 0x3fe2, 0x09bc, 0x8a26, 0xbc78, 0xe69a, 0xd0c4,
11 0xacd7, 0x9a89, 0xc06b, 0xf635, 0x75af, 0x43f1, 0x1913, 0x2f4d,
12 0xe135, 0xd76b, 0x8d89, 0xbbd7, 0x384d, 0x0e13, 0x54f1, 0x62af,
13 0x1ebc, 0x28e2, 0x7200, 0x445e, 0xc7c4, 0xf19a, 0xab78, 0x9d26,
14 0x7af1, 0x4caf, 0x164d, 0x2013, 0xa389, 0x95d7, 0xcf35, 0xf96b,
15 0x8578, 0xb326, 0xe9c4, 0xdf9a, 0x5c00, 0x6a5e, 0x30bc, 0x06e2,
16 0xc89a, 0xfec4, 0xa426, 0x9278, 0x11e2, 0x27bc, 0x7d5e, 0x4b00,
17 0x3713, 0x014d, 0x5baf, 0x6df1, 0xee6b, 0xd835, 0x82d7, 0xb489,
18 0xa6bc, 0x90e2, 0xca00, 0xfc5e, 0x7fc4, 0x499a, 0x1378, 0x2526,
19 0x5935, 0x6f6b, 0x3589, 0x03d7, 0x804d, 0xb613, 0xecf1, 0xdaaf,
20 0x14d7, 0x2289, 0x786b, 0x4e35, 0xcdaf, 0xfb1, 0xa113, 0x974d,
21 0xeb5e, 0xdd00, 0x87e2, 0xb1bc, 0x3226, 0x0478, 0x5e9a, 0x68c4,
22 0x8f13, 0xb94d, 0xe3af, 0xd5f1, 0x566b, 0x6035, 0x3ad7, 0x0c89,
23 0x709a, 0x46c4, 0x1c26, 0x2a78, 0xa9e2, 0x9fbc, 0xc55e, 0xf300,
24 0x3d78, 0x0b26, 0x51c4, 0x679a, 0xe400, 0xd25e, 0x88bc, 0xbee2,
25 0xc2f1, 0xf4af, 0xae4d, 0x9813, 0x1b89, 0x2dd7, 0x7735, 0x416b,
26 0xf5e2, 0xc3bc, 0x995e, 0xaf00, 0x2c9a, 0x1ac4, 0x4026, 0x7678,
27 0x0a6b, 0x3c35, 0x66d7, 0x5089, 0xd313, 0xe54d, 0xbfaf, 0x89f1,
28 0x4789, 0x71d7, 0x2b35, 0x1d6b, 0x9ef1, 0xa8af, 0xf24d, 0xc413,
29 0xb800, 0x8e5e, 0xd4bc, 0xe2e2, 0x6178, 0x5726, 0x0dc4, 0x3b9a,
30 0xdc4d, 0xea13, 0xb0f1, 0x86af, 0x0535, 0x336b, 0x6989, 0x5fd7,
31 0x23c4, 0x159a, 0x4f78, 0x7926, 0xfabc, 0xcce2, 0x9600, 0xa05e,
32 0x6e26, 0x5878, 0x029a, 0x34c4, 0xb75e, 0x8100, 0xdbe2, 0xedbc,
33 0x91af, 0xa7f1, 0xfd13, 0xcb4d, 0x48d7, 0x7e89, 0x246b, 0x1235
34  ]);
35  var crcResult = 0x00;
36  function crc16_byte(crc, data) Calculate CRC
37  |   return (crc >> 8) ^ crc16_table[(crc ^ data) & 0xff];
38  | }
39  function Crc16DNP(crc, data) Calculate CRC over DNP
40  {
41  |   for(var i in data){
42  |     crc = crc16_byte(crc,data[i]);
43  |   }
44  |   return crc;
45  | }

```


5.3. How to modify access algorithm

Schedule Name: Address:

<input type="checkbox"/>	Service Info	Suppress	Payload	Script	Description	<input type="checkbox"/> Selected
<input type="checkbox"/>	DiagnosticSessionControl (0X10)	<input type="radio"/>	subFunction : 3	<input type="button" value="Script"/>		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	CommunicationControl (0X28)	<input type="radio"/>	subFunction : 3 communicationType : 03	<input type="button" value="Script"/>		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	DiagnosticSessionControl (0X10)	<input type="radio"/>	subFunction : 2	<input type="button" value="Script"/>		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	SecurityAccess (0X27)	<input type="radio"/>	subFunction : 1 AccessData / securityKey :	<input type="button" value="Script"/>		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	SecurityAccess (0X27)	<input type="radio"/>	subFunction : 2 AccessData / securityKey :	<input type="button" value="Script"/>		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	WriteDataByIdentifier (0X2e)	<input type="radio"/>	dataIdentifier : f15a dataRecord : 5555	<input type="button" value="Script"/>		<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

User function

```
function preLoad(writeData) {
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
}

function afterLoad(writeData.readData) {
1 if ((0x67 == readData[0]) &&
2 (0x01 == readData[1]))
3 {
4   const crypto = require('crypto');
5   const algorithm = 'aes-128-cbc';
6   const iv = Buffer.alloc(16, 0); // Initialization vector.
7   const key=Buffer.from([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]);
8   const cipher = crypto.createCipheriv(algorithm, key, iv);
9   var res=cipher.update(Buffer.from(readData.slice(2)));
10  //var res=cipher.final();
11  this.log(res.toString('hex'));
12  this.changeNextFrame('AccessData / securityKey', [...res]);
13  return true;
14 }
15 else
16 {
17   return false;
18 }
```

Click the script button, user can modify the algorithm. Default is the AES. Key is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

5.4. How to config different TX message ID

In update process, send some services based on function ID, send some services based on physical ID.

5.5. How to config multi-schedule table

Usually, user need to send some UDS services as function ID. E.g., disable record DTC, disable non-UDS message transmit. Send some UDS services as physical ID. E.g., enter bootloader mode, update APP.

Import Export

Schedule Name: sch1 Address: NXP_S32K3XX_OTAD

Sch1: all the services will send have the same address. The address selected by user

Save Group Add Service

Service Info	Suppress	Payload	Script	Description	Selected
DiagnosticSessionControl (0X10)	⊙	subFunction : 3			⊙
CommunicationControl (0X28)	⊙	subFunction : 3 communicationType : 03			⊙

Schedule Name: sch2 Address:

Delete Schedule Save Group Add Service

Service Info	Suppress	Payload	Script	Description	Selected
DiagnosticSessionControl (0X10)	⊙	subFunction : 2			⊙
SecurityAccess (0X27)	⊙	subFunction : 1 AccessData / securityKey :			⊙
SecurityAccess (0X27)	⊙	subFunction : 2 AccessData / securityKey :			⊙
WriteDataByIdentifier (0X2e)	⊙	dataIdentifier : 1f5a dataRecord : 5555			⊙
Group: DownloadFile1		0 Download File : [{"name": "C:\Users\NXP10035\Desktop\CAN_FD_bootloader\S32K146_CANFD\flash_api.bin", "size": 1476} 0 dataFormatIdentifier : 00 0 addressAndLengthFormatIdentifier : 44 0 memoryAddress : 1FFF800			⊙
RoutineControl (0X31)	⊙	subFunction : 1 routineIdentifier : 0202 routineControlOptionRecord :			⊙

Here have an **known-issue** is that config ECU BUS receive the same message ID when TX physical/function ID. Due to 3rd party DLL prevent config 2 TX message ID and have the same RX message ID for normal addressing mode.

5.6. How to export update process

User can export the update process.

Github Version

← DOIP UDS

Import Export

Schedule Name: sch1 Address: NXP_S32K3XX_OTAD

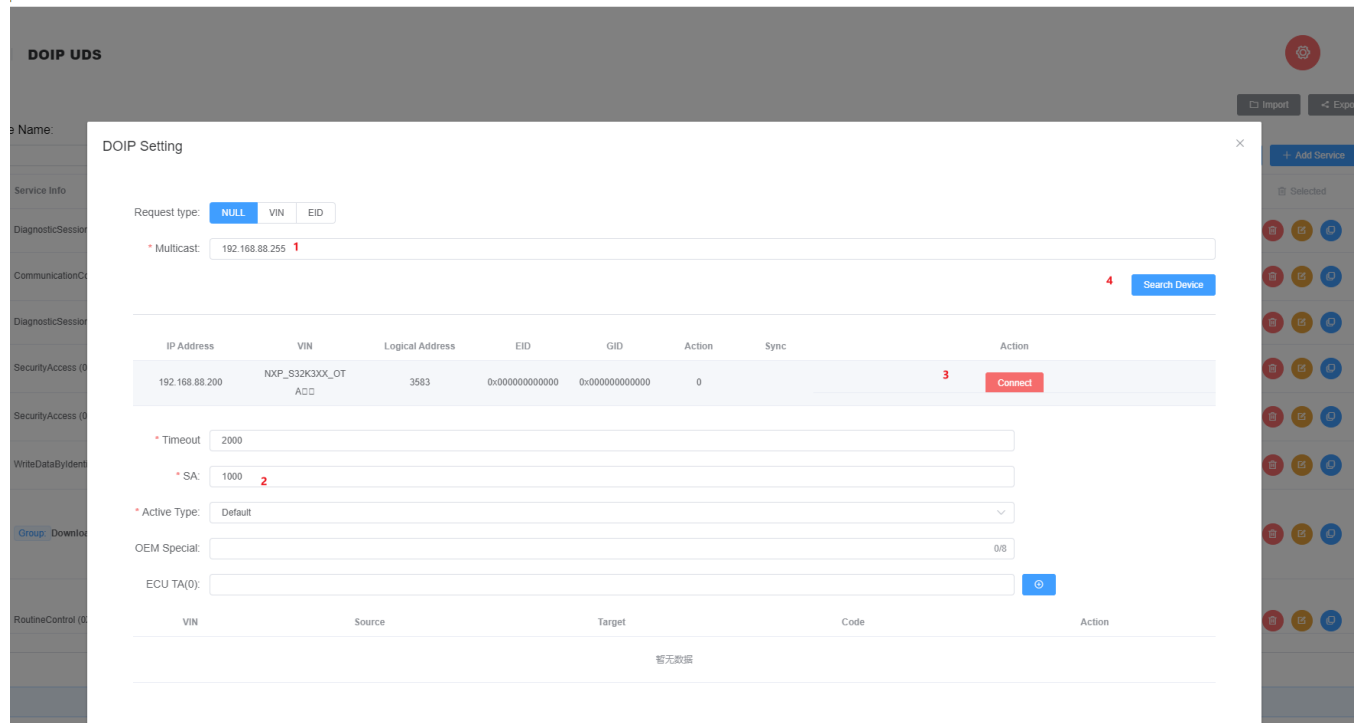
Save Group Add Service

Service Info	Suppress	Payload	Description	Selected
DiagnosticSessionControl (0X10)	⊙	subFunction : 3		⊙
CommunicationControl (0X28)	⊙	subFunction : 3 communicationType :		⊙
DiagnosticSessionControl (0X10)	⊙	subFunction : 2		⊙
SecurityAccess (0X27)	⊙	subFunction : 1 AccessData / secu		⊙
SecurityAccess (0X27)	⊙	subFunction : 2 AccessData / secu		⊙
WriteDataByIdentifier (0X2e)	⊙	dataIdentifier : 1f5a dataRecord : 5555		⊙
Group: DownloadFile1		0 Download File : [{"name": "C:\Users\NXP10035\Desktop\CAN_FD_bootloader\S32K146_CANFD\flash_api.bin", "size": 1476} 0 dataFormatIdentifier : 00 0 addressAndLengthFormatIdentifier : 44 0 memoryAddress : 1FFF800		⊙
RoutineControl (0X31)	⊙	subFunction : 1 routineIdentifier : 0202 routineControlOptionRecord :		⊙

Save As dialog box showing file name and save as type: JSON (*.json)

6. UDSONDoIP

Config DoIP as following:



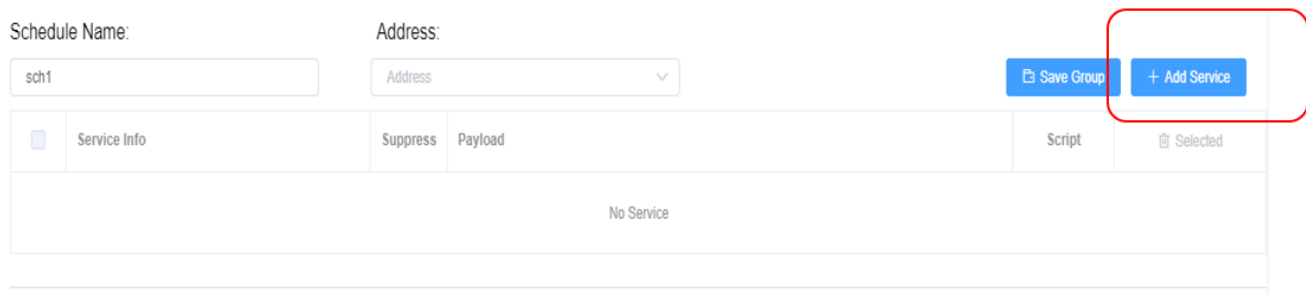
Step 1 is config IP address, step 2 fill SA(source address), step 3 search the target, step 4 is connect the target.

7. ISO-14299 UDS Feature

7.1. Key Feature

7.1.1. Service

Service is same as UDS service, all UDS services are supported in ECU-Bus.



Click Add Service Button, you will see

Add UDS Service

UDS GROUP

Service DiagnosticSessionControl Service List

DiagnosticSessionControl * subFunction Each Service Special Parameters

Sub-Function Suppress

User function `function(writeData,readData){
1 return true;
}` User Define Script

Add Service

- **Service List:** Here to select a service
- **Service Special Parameters:** Input parameters of a service.
- **User Define Script:** See [2.1.4](#)

Service

DiagnosticSessionControl

DiagnosticSessionControl	0x10
ECUReset	0x11
SecurityAccess	0x27
CommunicationControl	0x28
TesterPresent	0x3e
AccessTimingParameter	0x83
SecuredDataTransmission	0x84

Every parameter will check the format correct real time, most of all parameters are hex format.

UDS GROUP

Service

RoutineControl

RoutineControl

* subFunction

startRoutine Suppress

* routineIdentifier

0x 11

Please Input HEX Format

routineControlOptionRecord

0x

routineControlOptionRecord-ascii

I want to add a RoutineControl service, I write '11' in routineIdentifier field, '11' is a correct hex format, but routineIdentifier need 2bytes value, so the tool will say 'Please Input HEX Format'.

UDS GROUP

Service

RoutineControl

RoutineControl

* subFunction

startRoutine Suppress

* routineIdentifier

0x 1122

routineControlOptionRecord

0x

routineControlOptionRecord-ascii

After Click Add Service Button, You will see the service was show in a schedule(See [2.1.3](#)).

Schedule Name: Address:

<input type="checkbox"/>	Service Info	Suppress	Payload	Script	<input type="checkbox"/> Selected
<input type="checkbox"/>	RoutineControl (0X31)		<pre>subFunction : 1 routineIdentifier : 1122 routineControlOptionRecord : routineControlOptionRecord-ascii :</pre>		

You also can edit or delete the service from the schedule.

7.1.2. Group

Multi UDS services can be combined into a single group, which will help user to add some complex service, such as 0x34->0x36->0x37.

The screenshot shows a configuration interface with a 'Schedule Name' field containing 'sch1' and an 'Address' dropdown menu. A red box highlights a 'Save Group' button and an 'Add Service' button. Below is a table with columns: Service Info, Suppress, Payload, Script, and Selected. Two rows are visible, both highlighted with a red border:

Service Info	Suppress	Payload	Script	Selected
<input type="checkbox"/> RoutineControl (0X31)	<input type="radio"/>	subFunction : 1 routineIdentifier : 1122 routineControlOptionRecord : routineControlOptionRecord-ascll :		
<input type="checkbox"/> DiagnosticSessionControl (0X10)	<input type="radio"/>	subFunction : 1		

I want combine a 0x31 and a 0x10 into a group, click Save Group button. You will see

The screenshot shows two panels. The left panel, titled 'Service 0/5', contains a list of parameters with checkboxes:

- 0-subFunction
- 0-routineIdentifier
- 0-routineControlO...
- 0-routineControlO...
- 1-subFunction

Between the panels are left and right arrow buttons. The right panel, titled 'Group 0/0', contains the text '无数据' (No data).

All parameters were list in left table, you can chose which parameters as the new input parameters of the group.

The screenshot shows the same two panels as the previous image. The 'Service 0/3' panel now has three items:

- 0-subFunction
- 0-routineControlO...
- 0-routineControlOption...

The 'Group 0/2' panel now contains two items:

- 0-routineIdentifier
- 1-subFunction

I choose 0x31-routineIdentifier and 0x10-subFunction as the new input parameters and name the group test.

Add UDS Service

✕

UDS **GROUP**

Group

test

test

* routineIdentifier

0x 1122

User function

* subFunction

defaultSession Suppress

User function

Click Add service group type, you will see the group you defined in schedule.

Schedule Name: Address:

<input type="checkbox"/>	Service Info	Suppress	Payload	Script	<input type="checkbox"/> Selected
<input type="checkbox"/>	RoutineControl (0X31)	<input type="checkbox"/>	subFunction : 1 routineIdentifier : 1122 routineControlOptionRecord : routineControlOptionRecord-ascii :	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	DiagnosticSessionControl (0X10)	<input type="checkbox"/>	subFunction : 1	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Group: test	<input type="checkbox"/>	0-routineIdentifier : 1133 0-routineControlOptionRecord : 0-routineControlOptionRecord-ascii : 1-subFunction : 1 <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>

7.1.3. Schedule

Schedule Define services execute process, every schedule can be assigned a address. A schedule contains zero or multi services.

Schedule Name: Address:

<input type="checkbox"/>	Service Info	Suppress	Payload	Script	<input type="checkbox"/> Selected
<input type="checkbox"/>	RoutineControl (0X31)	<input type="radio"/>	<pre>subFunction : 1 routineIdentifier : 1122 routineControlOptionRecord : routineControlOptionRecord-ascii :</pre>	<input type="button" value="Script"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
<input type="checkbox"/>	DiagnosticSessionControl (0X10)	<input type="radio"/>	<pre>subFunction : 1</pre>	<input type="button" value="Script"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
<input type="checkbox"/>	Group: test		<pre>0-routineIdentifier : 1133 0-routineControlOptionRecord : 0-routineControlOptionRecord-ascii : 1-subFunction : 1</pre>	<input type="button" value="Script"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>

+

Log Level: Suppress Delay: ms, Timeout: ms, Cycle:

Click + to add a new schedule,

Schedule Name: Address:

<input type="checkbox"/>	Service Info	Suppress	Payload	Script	<input type="checkbox"/> Selected
<input type="checkbox"/>	RoutineControl (0X31)	<input type="radio"/>	<pre>subFunction : 1 routineIdentifier : 1122 routineControlOptionRecord : routineControlOptionRecord-ascii :</pre>	<input type="button" value="Script"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
<input type="checkbox"/>	DiagnosticSessionControl (0X10)	<input type="radio"/>	<pre>subFunction : 1</pre>	<input type="button" value="Script"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
<input type="checkbox"/>	Group: test		<pre>0-routineIdentifier : 1133 0-routineControlOptionRecord : 0-routineControlOptionRecord-ascii : 1-subFunction : 1</pre>	<input type="button" value="Script"/>	<input type="button" value="Delete"/> <input type="button" value="Edit"/>

Schedule Name: Address:

<input type="checkbox"/>	Service Info	Suppress	Payload	Script	<input type="checkbox"/> Selected
No Service					

+

Every schedule has a name to describe the purpose of these services. User can drag a service from a schedule to another schedule.

Log Level control the log information display level. **Suppress Delay** setup the delay time for suppress request service(without response). **Timeout** define the time waiting response message. **Cycle** define all services execute times for press test.

7.1.4. User Define Script

ECU-Bus supports flexible JS script to handle received data.

Service

DiagnosticSessionControl

DiagnosticSessionControl

subFunction

defaultSession

 Suppress

User function ⓘ

function(writeData,readData){

1 return true;

}

Every function must return a bool variable, true means correct response. Every function has two input parameter(writeData, readData), writeData is an object, which contains all information input, readData is an array.

function(writeData,readData){

```
1 this.info(writeData);
2 this.info(readData);
3 return true;
```

}

We use this.info to display the data. log windows:

```
[3933ms] [sch1] [{"name":"subFunction","subFunction":1,"suppress":false,"type":"subfunction"}]
[3933ms] [sch1] [80,1]
[3933ms] [sch1]"Excute successful"
```

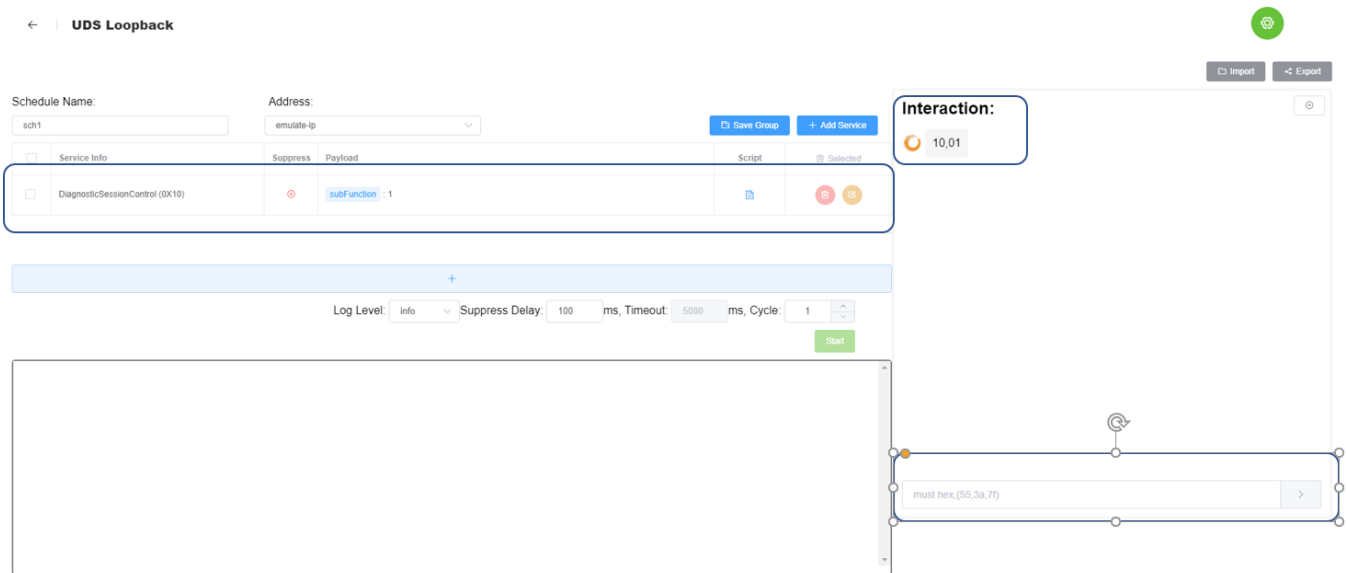
There are some function support by ECU-Bus:

Name	Parameters	Description
------	------------	-------------

this.log(msg,type='debug')	type:'error, warn, info, verbose, debug, silly'	Ouput log information
this.error(msg)	msg:any type	Equal this.log(msg,'error')
this.debug(msg)	msg:any type	Equal this.log(msg,'debug')
this.info(msg)	msg:any type	Equal this.log(msg,'info')
this.delay(ms)	ms,type int	Insert a new delay and wait another new reponse
this.openFile(filename,flag='r')	filename is a absolute file name path,flag:default is 'r'	Open a file,this function must be call before using readfile,writefile and closefile
this.readFile(size)	size,type int	Read data from a file,return type is a array, the length of array maybe less than size
this.writeFile(data)	data,type array of buffer	Write data to a file
this.closeFile()	null	Close a file
this.changeNextFrame(name,value)	name:should be payload name,value:the change value	Change the next service data in the schedule table
this.set(key,value)	key:string,value:any type	Store value in all life cycle
this.get(key)	key:string	Return last store value in this.set

7.2. Loopback UDS Tester

Loopback doesn't send data output, which help user to debug the script your write.

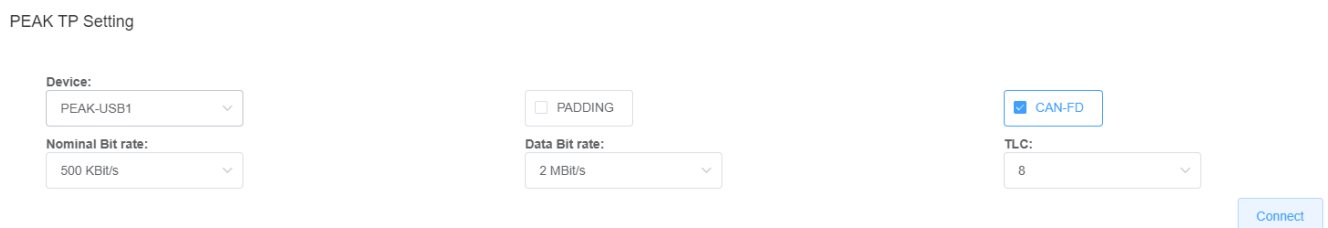
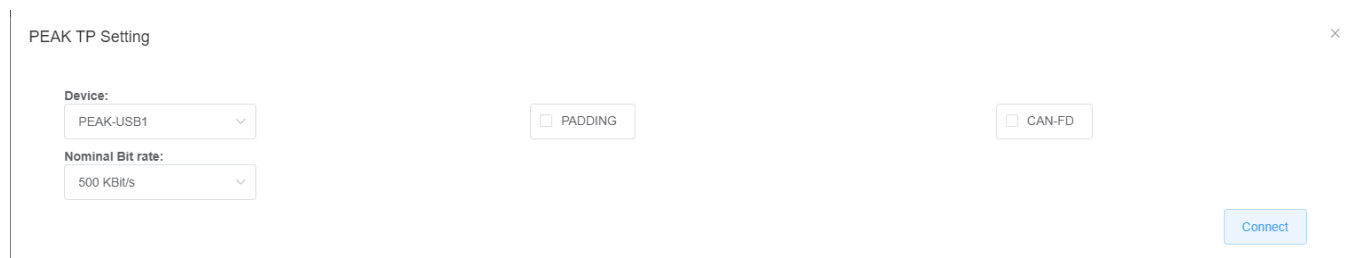


In interaction windows, user can edit the response data manual.

7.3. DoCAN UDS Tester

DoCAN only supports PEAK USB device right now.

First thing is to connect a USB PEAK device, support CAN and CAN-FD right now.



After connected PEAK, you must add at least one address. ECU-Bus supports these address format:

- Normal address
- Normal fixed address

- Extended address
- Mixed address

Name:

Format:

Address: Physical Functional

Message: Local Remote

CAN ID: 11 Bits 29 Bits

SA:

TA:

RA:

The map between IP and ID is necessary.

MAP TX_ID:

MAP RX_ID:

Name	Send ID	Receive ID	Maped	SA	TA	TA Type	RA	Format	ID	Action
defaultName1	85	52	<input checked="" type="checkbox"/>	0X11	0X22	Physical	0X0	Normal addressing		Delete

Then in schedule you can choose this address as target address.

7.4. DoIP UDS Tester

Firstly, user need use UDP to find device in target network.

Request type: NULL VIN EID

* Multicast:

If the tool find a device, which will list the device information in below table.

DOIP Setting ×

Request type: NULL VIN EID

* Multicast:

IP Address	VIN	Logical Address	EID	GID	Action	Sync	Action
192.168.1.3	NXP_S32K2TV_DC MC□	3583	0xa55aa55a0500	0x0000a55aa55a	17		<input type="button" value="Connect"/>

Then user can click connect button, the tool will send **active router** DOIP message firstly, if receive a success response, the TCP socket about this address is stable.

DOIP Setting

✕

Request type: **NULL** VIN EID

* Multicast: 192.168.1.255

Search Device

IP Address	VIN	Logical Address	EID	GID	Action	Sync	Action
192.168.1.3	NXP_S32K2TV_DC M□□	3583	0xa55aa55a0500	0x000a55aa55a	17		Connect

* Timeout: 2000

* SA: 1000

* Active Type: Default

OEM Special: 0/8

ECU TA(0):

VIN	Source	Target	Code	Action
NXP_S32K2TV_DC M□□	1000	3583	0x10	Close

Timeout defines the socket max wait time, **SA** define the source address(0-65535).

Then in schedule you can choose this address as target address.

8. References

NULL

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C 5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, AMBA, Arm Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and μ Vision are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. Arm7, Arm9, Arm11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, Mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2020 NXP B.V.

Document Number: ANXXXXX
Rev. draft v0.1
10/2020

