

# DXCPL DAP over CAN Physical Layer

## 32-bit TriCore™ AURIX™ TC3xx devices

### About this document

#### Scope and purpose

DAP over CAN Physical Layer (DXCPL) enables a tool connection through the regular CAN pins of the ECU connector. This tool connection is robust since it does not require a working flash or any software. DXCPL is in essence another option to transmit DAP telegrams.

This document gives an overview of DXCPL and provides some usage hints. Please note that this Application Note describes the DXCPL implementation for AURIX™ TC3xx devices, not for any other generation of AURIX™ devices.

**Note:** *This application note is valid only for AURIX™ TC3xx devices and not for AURIX™ TC2xx devices or older TriCore™ generations. For AURIX™ TC2xx, please refer to application note AP32264.*

#### Intended audience

Infineon® Tool partners and those seeking further information on DXCPL with AURIX™ TC3xx devices.

**Table of contents**

**Table of contents**

<b>About this document</b> .....	1
<b>Table of contents</b> .....	2
<b>1 Overview</b> .....	3
1.1 CAN pins to DAP module .....	3
1.2 DXCPL application .....	3
1.3 SPD (Single pin DAP) .....	4
1.3.1 Achievable bandwidth with a CAN transceiver .....	5
1.4 Enabling and pins .....	6
1.5 DXCPL activation .....	6
1.6 DXCPL de-activation .....	7
1.7 DXCPL activation prevention .....	7
1.7.1 DXCPL unintended activation .....	7
<b>2 Usage hints</b> .....	8
2.1 Activating DXCPL and unlocking the tool interface .....	8
2.1.1 Unlocking a device via DXCPL .....	8
2.1.2 Activate DXCPL and unlock the device .....	8
2.2 Halt After Reset (HARR) .....	8
2.3 Enabling of JTAG, DAP, DXCPL or DXCM .....	8
<b>3 Revision history</b> .....	10
<b>Disclaimer</b> .....	11

## 1 Overview

# 1 Overview

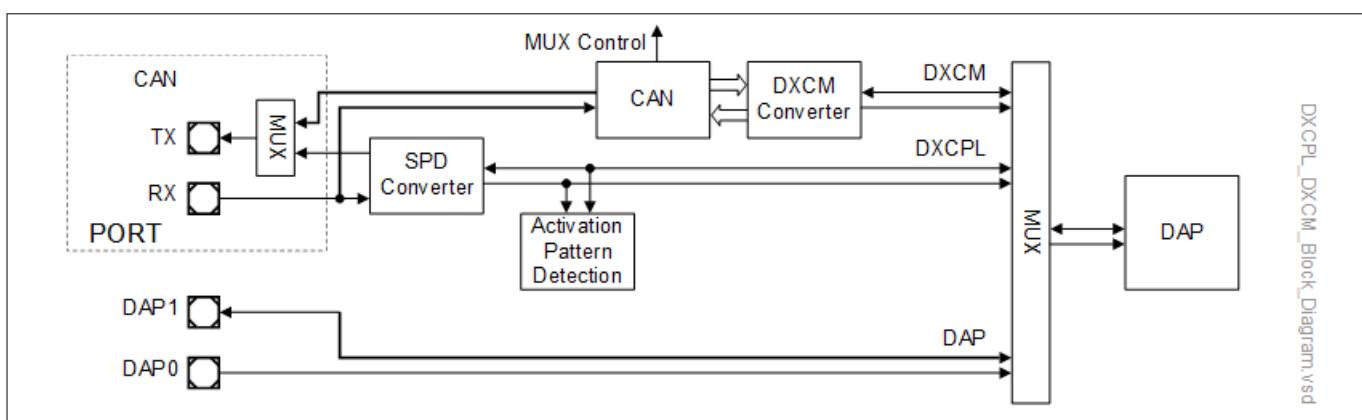
## 1.1 CAN pins to DAP module

The CAN bus physical layer is a bi-directional connection with differential encoding on a single pair of wires. When attaching a CAN node of the device to a CAN bus, an external transceiver is required. This transceiver has the role of translating the logical level of the TXD pin to opposing voltages on the CAN bus lines. Each voltage difference between the CAN bus lines is signaled to the RXD pin as a logical level (based on the sign of the difference). This direction separated signaling with TX/RX can be used to transmit SPD-encoded DAP messages.

The pins used for DXCPL are **P14.0 (CAN01\_TXD)** and **P14.1 (CAN01\_RXDB)**.

The alternative pins for DXCPL are **P33.12 (CAN00\_RXDD)** and **P33.13 (CAN00\_TXD)**.

The following figure shows the device internal DXCPL building blocks between CAN pins and the DAP module.



**Figure 1**      **DXCPL block diagram from CAN pins to DAP module**

Inside the device there is a pin monitor, which watches the input signal at the RXD pin of the CAN1 interface. In case the full activation pattern appears, the CAN module is disconnected and the pins are passed on to the DXCPL module. At the same time the dedicated DAP pins are disconnected.

**Note:** *The first part of the activation pattern switches only the multiplexer (MUX), but it does not disconnect the CAN module. Only when the second part of the activation pattern (dapisc telegram) is detected by the DAP module, the CAN module is disconnected from the pins. This was changed in AURIX™ TC3xx devices in order to improve the robustness.*

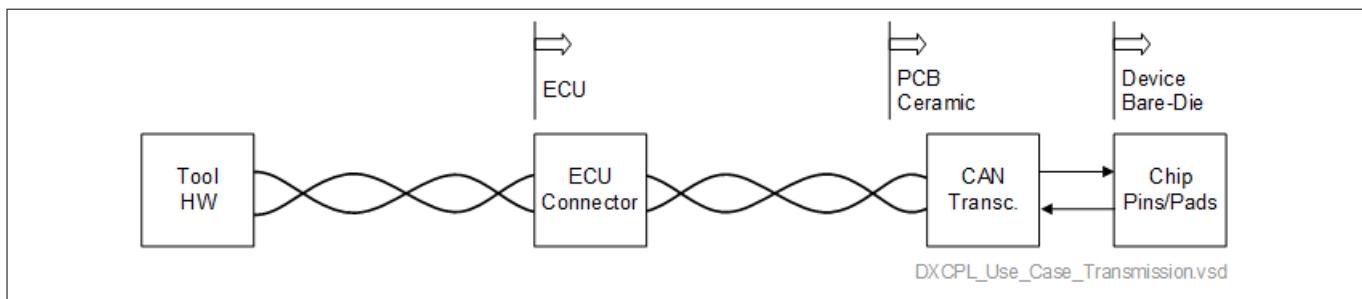
**Note:** Refer to [DXCPL activation](#) for the required steps to activate DXCPL.

The DXCPL module is able to use the physical CAN bus for bi-directional communication with a tool using the SPD protocol. The intended use case is to establish the connection between the tool and the bus side of the transceiver through either the ECU connector or the system's cable network.

## 1.2 DXCPL application

Since the CAN bus signals are available at the ECU connector, DXCPL allows tools to be connected without opening the ECU housing. This is very useful for a cost effective analysis of field returns for example.

## 1 Overview



**Figure 2 DXCPL Use Case Transmission ECU**

DXCPL can be used for three types of applications:

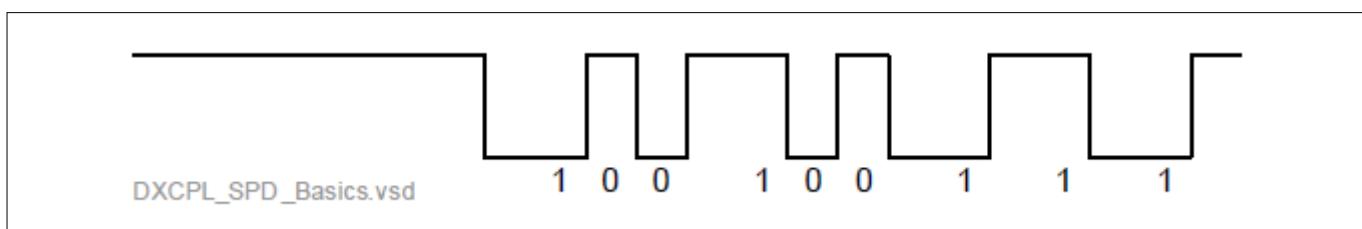
- **General debugging**
  - Usually it is used when no other debug interface is available or accessible. After activation the OCDS (On-Chip Debugging System) infrastructure is accessible provided that the tool access is enabled (OSTATE.IF\_LCK permits the IOClient to gain access). In this case the test modes and boundary scan (generally available in user-mode) are locked for safety reasons.
- **Measurement and calibration**
  - DXCPL is an electrically robust two wire measurement and calibration tool connection, which can be used for deeply embedded applications such as transmission ECUs within the gear box.
- **Application-Test-Mode (ATM)**
  - After SPD activation, entering this mode requires additional sequences on top of the dapisc DAP initialization telegram, which are complex enough to avoid unauthorized access. After activation the OCDS debug infrastructure is accessible with similar restrictions to those of a regular DAP connection, such as access of a locked device only after authentication. As a difference to DAP, boundary scan is not possible since the CAN pins are part of the scan chain.

### 1.3 SPD (Single pin DAP)

The SPD protocol encodes the DAP bits based on the distances between the SPD signal edges. A DAP1 value of '0' is encoded with a short distance and a value '1' with twice the distance (see [Figure 3](#)).

**Table 1 DXCPL SPD Parameters**

Parameter	Description
Bandwidth	Similar with regular DAP with 1.6 MHz clock with standard settings. It can be increased by a factor of 2, 3, or 4 (use of DAPISC.CONF).
Clock source	EVR oscillator (100 MHz nominal)
Edge distance '0'	0.4 µs (nominal)
Edge distance '1'	0.8 µs (nominal)
Decision point	0.6 µs (nominal)



**Figure 3 SPD basic protocol**

## 1 Overview

The robustness of the SPD protocol is dependent on the decision point. For getting the best results, the SPD receiver implementation needs to fix the decision point between '0' and '1' centered with '0' being 2/3 of the decision point and '1' being 4/3. When using the standard SPD encoding distance, the optimum decision point is 0.75 µs.

### Notes:

1. A cycle of an applied 'clock' signal contains two edges, which means two SPD bits.
2. SPD bandwidth figures assume an even distribution of the short '0' and twice as long '1' bits.

Any SPD/DAP communication is initiated by the tool. Therefore the initial state of the SPD signal will be low and the SPD module will wait for the first telegram from the tool, which always starts with a positive and long ('1' bit) SPD signal pulse, which is the SPD encoded start bit. The device detects this state due to the telegram protocol and starts creating the response, which is either an SPD telegram or only an SPD start bit as defined in the DAP protocol.

**Note:** If the SPD signal level is high after the last bit of the telegram, the tool has to add a '0' bit which means a final negative edge.

Given the fact that DAP is a clean request response protocol, the tool is always the requestor and the device executes and sends a response. After receiving the telegram from the tool and changing the direction, the response is sent. The response is basically a DAP reply telegram with one or two additional bits:

- A new bit is inserted after the start bit (when SPD is overlaid to a GPIO pin, it can be the value of the port output bit, a trigger indicator like DAP TGIP or a constant '0'). Please note that this bit is not included in the CRC of the DAP reply telegram.
- If the SPD output value is 1 after the last telegram bit, the SPD module will also automatically append a SPD '0' bit to bring the SPD value back to 0.

### 1.3.1 Achievable bandwidth with a CAN transceiver

The bandwidth is limited by the posedge/nededge asymmetry and the jitter of the transceiver. For CAN transceivers, a variation of up to 100 ns has to be taken into account. Because of this variation a short '0' edge distance is prolonged or a long '1' edge distance is shortened and the receiver detects the wrong value, then the SPD detection fails. It also fails when a short distance is shortened to less than one sample and a long distance is prolonged to more than the timeout value of the receiver. This effect needs to be taken into account for the SPD parameterization for both tool and device.

For a minimized quantization loss, an over-sample rate of 5 or more is required.

The following parameters are used by the on-chip SPD module:

- Oversample rate (time step) is 5
- Transmitter outputs a short edge distance ('0') with 4 time steps
- Transmitter outputs a long edge distance ('1') with 11 time steps
- Receiver recognizes an edge distance of 1 to 7 time steps as '0'
- Receiver recognizes an edge distance of 8 to 14 time steps as '1'

If the same SPD parameters are used on tool side, a bandwidth of 600KB/s can be achieved. This requires good CAN (FD) transceivers at both ends which ensure a 90 ns or less edge distance variation for the whole signal path.

## 1 Overview

### 1.4 Enabling and pins

DXCPL is by default enabled after Power-on-Reset for a certain time window. When enabled, it listens on the default RX pin for the activation sequence. Only after the activation sequence, including the dapisc DAP initialization telegram was sensed, the RX/TX pins will be taken over exclusively for DXCPL.

**Note:** *If DXCPL is just enabled and not yet activated, it has no impact on any device behavior (it just listens). Only when the DXCPL activation pattern (out of CAN spec) and the dapisc DAP initialization telegram (50 significant bits) were received, the RX/TX pins will be taken over. With the default settings (DMU\_HF\_PROCONDBG.TIC), this needs to happen in the time window between Power-on-Reset release and Flash start-up (~1ms).*

**Note:** *A pre-requisite for DXCPL communication is an enabled CAN transceiver. This means that for the default DMU\_HF\_PROCONDBG.TIC setting, the DXCPL activation time window is already over when the user software enables the CAN transceiver. Only the other settings will allow DXCPL communication.*

It is recommended to ensure that the CAN transceiver is enabled already immediately after Power-on-Reset, otherwise it will not be possible to debug all kind of situations using DXCPL. The robustness of DXCPL is then just on a similar level as DXCM including that no Halt after Power-on-Reset is possible.

**Table 2 DXCPL enabling and pins controlled by DMU\_HF\_PROCONDBG.TIC**

TIC	Pins	Description
$0_H$	Default pins: P14.0 (CAN01_TXD) P14.1 (CAN01_RXDB)	DXCPL activation is prevented after flash startup. If already activated (including the step dapisc DAP initialization telegram), it will stay activated.
$1_H$		DXCPL will stay enabled/activated after Flash Start-up.
$2_H$	Alternative pins: P33.12 (CAN00_RXDD) P33.13 (CAN00_TXD)	DXCPL is enabled for alternative CAN pins after flash startup if it was not already activated for the default pins.

### 1.5 DXCPL activation

The DXCPL activation pattern is defined as 640 consecutive ‘0’-values in SPD format followed by a dapisc DAP initialization telegram. This means that the permanent signal changes occur at a slightly smaller time interval than the decision point of the SPD receiver of the device (approximately 0.55 µs).

**Note:** *In some cases (for example when the EVR oscillator is not trimmed), the device’s internal SPD decision point is biased. For this situation, the tool has to retry with a ± 20% edge distance until the device responds to a DAP sync telegram. At this moment, the device internal biasing of the decision point can be determined and the tool can adapt accordingly for a robust communication.*

The activation pattern will never occur inside any legal CAN traffic due to the structure of the CAN protocol:

- The CAN protocol consists of specific frames with a maximum length of up to 128 bits, where the last seven End-of-Frame bits are transmitted as ‘1’ and at least three more ‘1’ bits must follow before the next frame can start.
- Normal CAN data exchange is done with 512 kbit/s or fastest 1 Mbit/s, therefore a signal change will not occur faster than every 1-2 µs.
- For CAN FD, the data payload of a maximum of 64 bytes is too small to have an unintended activation by a data pattern.

## 1 Overview

- SPD will assume a ‘1’- value whenever the time between two signal changes is longer than 0.55 µs, which is the case at the end of every CAN FD frame.
- When SPD recognizes a ‘1’ before 640 ‘0’-values have been seen, the whole activation sequence must be restarted.

The DXCPL activations is based on 2 steps:

1. Sending the SPD activation pattern
2. Sending a valid dapisc DAP initialization telegram

Only after the second step will the CAN module be disconnected from the RX/TX pins.

**Note:** *For the special case, when the tool is plugged into a “live” bus, it has to be considered that the transmission of SPD-conforming ‘0’- values represents a violation of the CAN-protocol. Therefore any active CAN-node will react by actively sending an error-frame. This is repeated (by the same node or another) until the internal error counter(s) overflow(s). Sooner or later all nodes will reach this state and be disabled automatically. These error frames may disturb the SPD activation sequence when a ‘1’ results from a bus conflict. This may require that more than 640 ‘0’-values must be transmitted until SPD is active.*

The DXCPL/SPD active operating condition is also indicated to any user software by SCU register bit **STSTAT.SPDEN**. This is used to ignore errors reported by the disconnected CAN module.

### 1.6 DXCPL de-activation

The DXCPL de-activation and switching the interface pins back to regular CAN is possible only through Power-On Reset.

### 1.7 DXCPL activation prevention

DXCPL is available only when DAP is brought up into Locked Mode by Power-On Reset. This means that only one of the options (DAP, DXCM or DXCPL) can be used at any point in time.

In the case DXCPL has not yet been activated, the user can prevent any later activation by writing 1 to the SYSCON.DDC bitfield. This write has no effect once DXCPL is activated.

Another way of preventing activation is to set the HF\_PROCONDBG.TIC bitfield to 0b000.

#### 1.7.1 DXCPL unintended activation

For AURIX™ TC3xx, the option of permanently disabling DXCPL is not needed anymore, due to the introduction of the dapisc DAP initialization telegram, which is well defined, in the activation sequence. Therefore the risk of unintentional activation of DXCPL does not exist anymore.

**Note:** *For protection against hacking, the locking mechanism available for all tool interfaces can be used, therefore protecting the device by password against any unauthorized access.*

## 2 Usage hints

### 2 Usage hints

#### 2.1 Activating DXCPL and unlocking the tool interface

##### 2.1.1 Unlocking a device via DXCPL

- Setting IO client to 1
- Read IOINFO
- Issue a system reset via OJCONF with HARR
- Read IOINFO
- Wait pprox. 500 us (time might be different depending on the previous steps)
- Proceed with the COMDATA key exchange
- Read IOINFO
- If IOINFO.IF\_LCK = 1, proceed with the 256 bit password exchange

**Note:** *The wait time of 500 µs can slightly vary depending on the device (consider an interval between 400 us and 650 µs).*

##### 2.1.2 Activate DXCPL and unlock the device

When it is necessary to activate DXCPL and also the unlock the device, the following sequences can be used:

- Power-On Reset -> Activate DXCPL -> Device Locked Status -> System Reset -> Device unlock sequence
- Power-On Reset -> Activate DXCPL -> Device unlock sequence

#### 2.2 Halt After Reset (HARR)

It is possible to have a reliable Halt After Reset Request (HARR) with DXCPL. For locked devices the HARR sequence needs to be extended by the password exchange. The tool needs to execute the following steps:

##### HARR by controlling the PORST Pin

- Release the PORST pin.
- Wait approximately 100 µs for on-board and device internal Power-on Reset release.
- Apply the [DXCPL activation](#) including dapisc DAP initialization telegram (takes about 300 µs)
- Proceed with the DAP HARR telegram sequence.

##### HARR without controlling the PORST Pin

It is possible to have a reliable HARR even when the tool does not control the PORST pin.

- The tool periodically sends the [DXCPL activation](#) sequence including the dapisc DAP initialization telegram
- When the device responds to the telegram, the tool proceeds with the usual DAP HARR sequence

**Note:** *For “small” devices with a short HARR time window, the DXCPL Activation Sequence can be optional shortened to 170 µs by using half the regular SPD ‘0’ bit width (0.2 µs) for the 640 ‘0’-values. The dapisc DAP initialization telegram needs to be sent with regular bit width.*

#### 2.3 Enabling of JTAG, DAP, DXCPL or DXCM

This section is used to give an overview of the interface selection. The below table lists the main decision points for the tool interface enabling.

## 2 Usage hints

**Table 3 Decision points for tool interface enabling**

<b>Decision</b>	<b>Interface</b>	<b>Condition</b>	<b>Disabled tool interfaces</b>
Select one of: DAP DXCPL DXCM	DAP	Enabled per default	DXCPL, DXCM (DAP turn-off to JTAG possible)
	DXCPL	DXCPL activation pattern detected on CAN RX	JTAG, DAP, DXCM
	DXCM	DXCM is enabled with bit CAN0_MCR.DXCM	JTAG, DAP, DXCPL

For this Application Note only the decision point (2) is of interest, which represents a simplified rule: first claims the interface.

### 3 Revision history

## 3 Revision history

<b>Document version</b>	<b>Date of release</b>	<b>Description of changes</b>
V1.0	June 2019	First release
V1.1	September 2022	Corrected header Added additional details about activation prevention of DXCPL
V1.2	2024-03-11	Template update; no content update.

## **Trademarks**

All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2024-08-14**

**Published by**

**Infineon Technologies AG  
81726 Munich, Germany**

**© 2024 Infineon Technologies AG  
All Rights Reserved.**

**Do you have a question about any aspect of this document?**

**Email: [erratum@infineon.com](mailto:erratum@infineon.com)**

**Document reference  
IFX-qbu1709805363396**

## **Important notice**

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

## **Warnings**

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.