The SPC5 series of micro-controllers is intended for automotive applications featuring functional safety (ASIL-d) and cyber security functions. The Debug-over-CAN is a new feature of the multi-core MCU. It is intended to enable debugging of ECUs in the field, where standard debug interfaces like JTAG or Nexus are usually not readily available, but access to the CAN interface is possible. Use of the CAN interface for debug purposes requires the use of the following resources to be exclusively used for that application:

- MCAN resources: Three CAN filters need to be configured for the three CAN debug messages. Furthermore two dedicated TX elements are needed for the transmission of the debug messages that are to be sent back to the external tool.
- DMA resources: A total of six DMA transfers is required per Debug-over-CAN cycle. When using the scatter/gather mechanism of the DMA module only one DMA channel is needed, plus 192 byte of flash memory (32 byte for each of the six required Transfer Control Descriptors).

Initialization of both MCAN and DMA via software is required, but once this initialization has completed, debugging over CAN is possible without any further software overhead. The initialization can be performed in the boot code.

As the Debug-over-CAN scheme generates internal JTAG messages based on the received CAN data, all JTAG clients, and included debug resources are accessible. Basic trace capability is also possible by configuring the trace hardware to stream to an overlay/trace RAM, which can be read later using debug over CAN.

No other (external) JTAG tool/debugger can be connected when using Debug-over-CAN. If a JTAG tool is connected, the JTAGM is not able to access the DCI resources.

In a typical application in the field, an external debug tool sends debug CAN messages to one of the MCAN
modules of the micro-controller. The debug messages have specific CAN-IDs and have to be sent/received in a specific order but may be interleaved with non-debug CAN messages. A mechanism internal to the MCAN module consisting of debug message filtering and a debug message state machine handles the incoming CAN debug messages, stores them correctly in a specified RX Buffer and triggers DMA data transfers between MCAN and JTAGM.

The JTAGM acts as a JTAG master within the device and is used to generate internal JTAG messages. Three CAN messages are necessary to generate the JTAG messages. This means three MCAN message filters need to be configured for the debug messages and the DMA channel needs to be enabled for MCAN triggering. The MCAN modules are part of the Controller Area Network (CAN) controller alongside the Time-Triggered CAN (TTCAN) modules and the CAN RAM controller. The CAN implementation complies with ISO 11898-1:2015 and supports Classical CAN as well as CAN FD.

The RX Handler and the TX Handler provide all functions concerning the handling of messages. The RX Handler manages message acceptance filtering, the transfer of received messages from the CAN core to the Message RAM as well as providing receive message status information. The TX Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN core as well as providing transmit status information. Acceptance filtering is implemented by a combination of up to 128 base filter elements or 64 extended filter elements. Each one can be configured as a range, as a bit mask, or as a dedicated ID filter.

The MCAN modules 1 and 2 can be used for the Debug-over-CAN applications. Their role in this application is to receive and handle the incoming debug messages from an external tool, trigger the DMA transfer sequence, and to send internal debug messages back to the external tool. Incoming messages are identified as debug messages through a unique CAN-ID and then stored in a specified location in the receive buffer. After the three CAN debug messages have been received correctly and stored in the debug receive buffer, the MCAN triggers a DMA transfer to send the data to the JTAGM. The MCAN also receives data back from the JTAGM via the DMA. This data is to be sent back to external tool. The CAN reception and transmission of these debug messages has to be configured by the user.

Description of the debug sequence

Typically debugging over CAN takes several individual Debug-over-CAN cycles to accomplish a given task, e.g. a Nexus Read/Write Access. In one Debug-over-CAN cycle...

CAN Products for your requirements

- Economical solutions for series applications
- Optimized for industrial applications
- Solutions for stationary and mobile use
- Software support for bus-analysis, measurement and control

EMS
Thomas Wünsche
Sonnenhang 3
D-85304 Ilmünster
Tel.: +49-8441-49 02 60
Fax: +49-8441-8 18 60
www.EMS-wuensche.com
an external tool sends three CAN debug messages A, B, and C sequentially to the MCAN of the micro-controller. The debug messages may be interleaved with none debug messages but must be in the correct order. The CAN debug messages are identified by a unique CAN-ID. When a debug message has been identified it is stored in the user defined RX Buffer in the CAN message RAM.

After the correct reception of debug messages A, B, and C, the MCAN triggers a DMA transfer. First, the content of the JTAGM status register (JTAGM_SR) and the JTAGM DATA IN 0 (JTAGM_DIR0) and JTAGM DATA IN 1 (JTAGM_DIR1) registers are transferred to the user defined MCAN TX buffer in the CAN message RAM. JTAGM_DIR0 and _DIR1 hold the debug data from the previous Debug-over-CAN cycle and hence needs to be read back before starting a JTAG traffic generation sequence.

After reading back the data from the JTAGM Status and Data-In registers, the relevant data from the debug messages A, B, and C in the RX Buffer of the CAN message RAM is transferred to the JTAGM’s module control register (MCR), status register (SR) and data out registers (DORO-3).

Writing the LSB in DOR3 sets the JTAGM_DOR3_SEND bit and internal JTAG traffic is generated depending on the data written to the JTAGM registers. Parallel to that the MCAN sends the data that has just been stored in TX Buffers back to the external as debug messages D and E. The MCAN transmission is initiated using a DMA transfer to write transmission start bit of the given TX element.

Before starting the next Debug-over-CAN cycle the external tool waits for the two CAN messages D and E coming back from the MCU’s MCAN. This provides enough time for the internal JTAG traffic to be finished and the data in JTAGM_DIR0 and _DIR1 has been updated.

MCAN debug message handling

Generally, the CAN modules on the SPC57xx/SPC58xx MCUs share 16 KiB of common CAN message RAM space. Within this CAN message RAM the user defines an individual area for each of the CAN modules. This area contains the message filters, Rx FIFO blocks, Rx Buffers, a Tx Event FIFO block and Tx Buffers.

The CAN modules accept CAN messages with matching CAN-ID. Any CAN message that is accepted by the CAN module will be stored as an element in either Rx FIFO_0, Rx FIFO_1 or Rx Buffer depending on the message filter that is configured for a given CAN-ID.

The eight data bytes (DB0 to DB7) contain the data that will be transferred to the JTAGM by means of the DMA module. If an extended filter is used the XTD bit is set to ‘1’ and all 29 bits of the CAN-ID are used. For a base ID, the bits [28 to 18] are used for the 11-bit CAN-ID. In order to receive/accept messages and to handle debug messages as desired the correct filter settings have to be programmed into the filter blocks of the CAN Message RAM Configuration area.

As mentioned above, the three debug messages need to be received in the correct order. The DMS bit field in the JTAGM status register indicates the status of the FSA. Debug messages that arrive in the wrong order are rejected, and the state machine is reset to its initial state. The correct reception of debug message C triggers a DMA transfer (if configured correctly). However, before the debug data is transferred to the JTAGM, the status and the debug data from the previous Debug-over-CAN cycle must be read back from the JTAGM and transferred to the MCAN to be sent to the external tool.

DMA transfers between MCAN and JTAGM

The DMA transfer sequence is triggered by the MCAN when debug message C has been received successfully. A total of six DMA transfers are required to complete one sequence of debug message transfer between MCAN and JTAGM.
First, two transfers are required to send data from the JTAGM to the transmit buffers of the MCAN. Thereafter, three transfers are required to send the new debug messages from the MCAN debug message RX buffer to the JTAGM. One transfer is required to trigger the MCAN transmission to send data back to the external CAN debug tool.

One additional transfer is required when using the Debug-over-CAN feature with SPC5744K. Due to an errata the new data flags in the MCAN_NDAT registers have to be reset by a DMA transfer at the end of each Debug-over-CAN cycle. It is recommended to do that before triggering the MCAN transmission with the final DMA transfer.

For more details please contact CiA office at service@can-cia.org

CiA® plugfests

- **CANopen lift**
  Interoperability testing of the CANopen application profile for lift devices (CiA 417).

- **CANopen CiA 4xx**
  Interoperability testing of devices implemented according to generic CANopen profiles like CiA 401, CiA 402, CiA 406, CiA 410.

- **CAN FD**
  Proving the interoperability of nodes implementing the CAN FD CAN protocol.

CiA plugfests are for members only, participation is free of charge.

For more details please contact

*www.can-cia.org*