I2C_Read_Ext_Device_1 for KIT_AURIX_TC397_TFT

Read external device MAC address via I2C

AURIX™ TC3xx Microcontroller Training
V1.0.1

Scope of work

An I2C module configured as I2C master is used to read a register of an external device.

An I2C module configured as I2C master is used to read the MAC address stored in MCP79411, a Real-Time-Clock device mounted on the board KIT_A2G_TC397_5V_TFT. The AURIX™ device reads the MAC address through the I2C module and stores it into a global variable.

Introduction

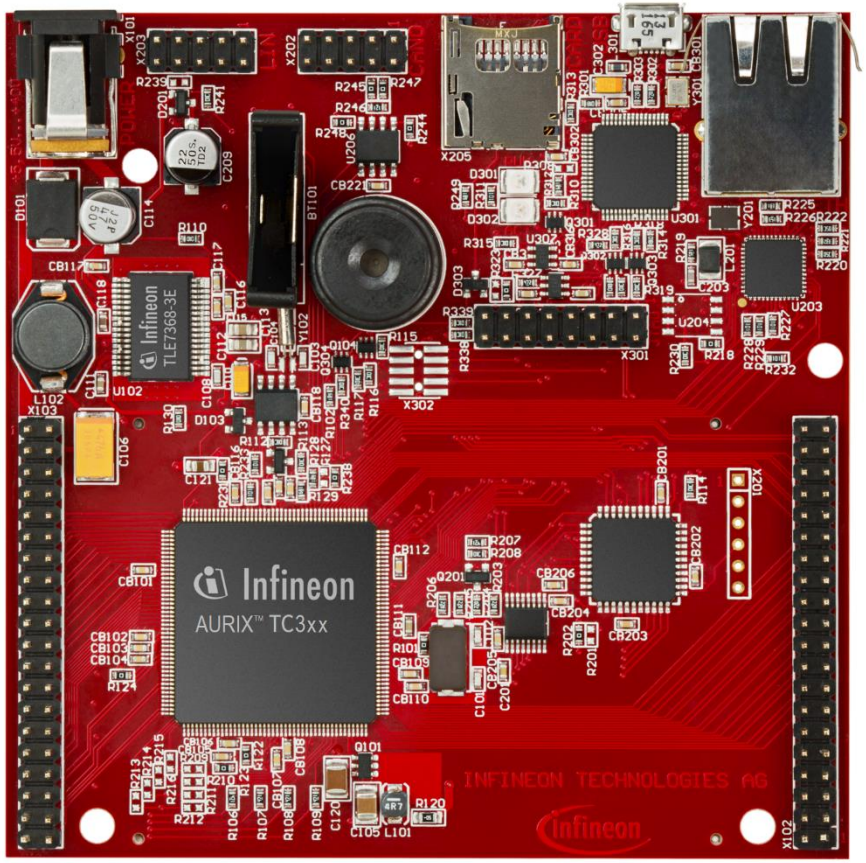
- The I2C protocol was developed to provide a simple and efficient data transfer between multiple devices over a short distance.
- It uses a bidirectional serial bus with two wires. A serial data line (SDA) and a serial clock line (SCL) are carrying the information between multiple devices.
- Both lines are connected to a positive supply voltage via pull-up resistors.
- An I2C device can work as a master or as a slave. The master, which is normally a microcontroller, initiates and terminates the transfer and generates the clock pulse.
- A specific slave can be addressed by the master via a 7- or 10-bit address. Afterwards the master starts the communication.
  - Data can flow in either direction and can be set via a data direction bit, which is transmitted by the master.

Hardware setup

This code example has been developed for the board KIT_A2G_TC397_5V_TFT.

The MCP79411 chip is mounted on the board and connected via the I^2C bus to the microcontroller.
The used data lines are connected to the positive power supply via two pull-up resistors.

The MCP79411 is a battery-backed I2C Real-Time Clock/Calendar (RTCC) device with SRAM, EEPROM and protected EEPROM.



Implementation

Configuring the I²C communication

The configuration of the I²C communication is done once in the setup phase in two different steps:

- The initialization of the I²C module by initializing an instance of the ***IfxI2c_I2c_Config*** structure
- The initialization of every device that is connected to the I²C module (in this case, the MCP79411 chip) by initializing an instance of the ***IfxI2c_I2c_deviceConfig*** structure for each device

Implementation

Configuring the I²C module

The function ***IfxI2c_I2c_initConfig()*** initializes an instance of the structure ***IfxI2c_I2c_Config*** with its default values.

The ***IfxI2c_I2c_Config*** structure allows setting the parameters to initialize the module:

- ***baudrate*** – to set the clock speed in bit/s. Typical values are 100 kbit/s in standard mode, 400 kbit/s in fast mode and 3.4 Mbit/s in high-speed mode.
- ***pins*** – a structure to set the port pins used for the communication.
  A serial data line (SDA) and a serial clock line (SCL) carry the information between the devices, therefore two port pins are required.

The function ***IfxI2c_I2c_initModule()*** initializes and activates the I2C module with the user configuration in master mode.
The functions above are provided by the iLLD header ***IfxI2c_I2c.h***.

Implementation

Configuring the I²C device

The function ***IfxI2c_I2c_initDeviceConfig()*** initializes an instance of the structure ***IfxI2c_I2c_deviceConfig*** with its default values.

Afterwards, the 7-bit slave address can be set through the parameter ***deviceAddress.***

The function ***IfxI2c_I2c_initDevice()*** finalizes the I²C initialization by connecting the device configuration with the preconfigured I²C module.

The functions above are provided by the iLLD header ***IfxI2c_I2c.h***.

Implementation

Establish I²C communication

Data transfer between the external device and the microcontroller is divided into two steps:

- Firstly, the microcontroller is transmitting the address of the register, in which the requested data is stored on the external device (the register containing the MAC address of MCP79411 is 0xF2). This is done using the ***IfxI2c_I2c_write()*** function.
- Then, the reading of the MAC address is started with the function ***IfxI2c_I2c_read()***.

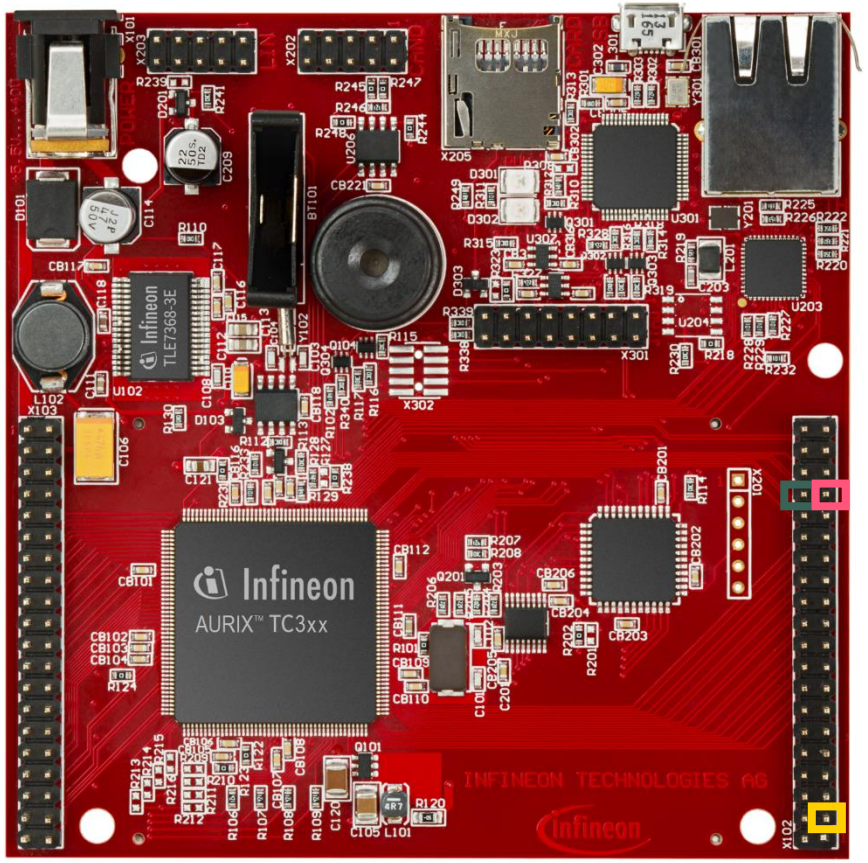
Both the write and read functions are defined in the iLLD header ***IfxI2c_I2c.h***.

Run and Test

After code compilation and flashing the device, perform the following steps:

- Use the debugger to watch the hexadecimal value of the global array ***g_macAddr***. The MAC address is unique for each board.

Run and Test



- A second test can be performed with an oscilloscope. Two oscilloscope probes can be connected to SDA and SCL pins to observe the generated and received signals.

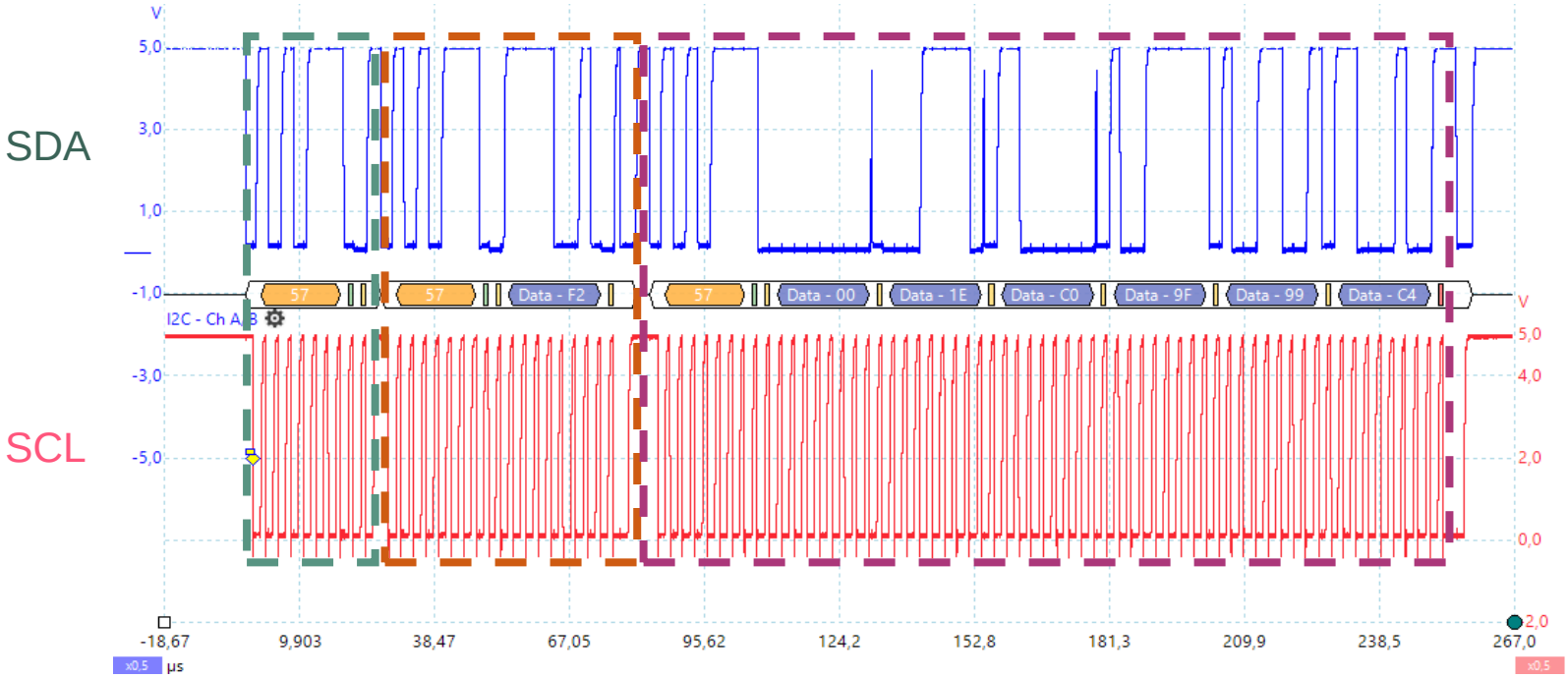
	X102		
P14.5	40	39	P14.4
P33.10	38	37	P20.9
P15.7	36	35	P15.6
P15.5	34	33	P15.4
P15.3	32	31	P15.2
P22.3	30	29	P22.2
P22.1	28	27	P22.0
P33.11	26	25	P23.4
P23.3	24	23	P23.2
P23.1	22	21	P23.0
P33.6	20	19	P33.8
P33.12	18	17	P33.1
P33.2	16	15	P33.3
P33.4	14	13	P33.5
AN0	12	11	AN8
AN2	10	9	AN3
AN11	8	7	AN13
AN20	6	5	AN21
GND	4	3	GND
V_UC	2	1	VCC_IN

Run and Test

The following waveforms should be seen on the oscilloscope after pressing the PORST push button:

- First data section: device address byte
- Second data section: device address byte and register address
- Third data section: device address byte and six bytes of data

(Please refer to the next slide for more details about the data sections)



Run and Test

- First data section: First attempt to establish communication with the device by transmitting the device address (0x57).
- Second data section: After the external device has successfully acknowledged the transmission, the write process of the register location (0xF2) is started (transmitting the device address (0x57) with the Read/Write bit set to „write“ and transmitting 0xF2 afterwards).
- Third data section: Reading process is started by transmitting the device address (0x57) and setting Read/Write bit to „read“: six bytes from the RTCC device, containing the MAC address, are then received.

References

- AURIX™ Development Studio is available online:
- https://www.infineon.com/aurixdevelopmentstudio
- Use the *„Import...“* function to get access to more code examples.

- More code examples can be found on the GIT repository:
- https://github.com/Infineon/AURIX_code_examples

- For additional trainings, visit our webpage:
- https://www.infineon.com/aurix-expert-training

- For questions and support, use the AURIX™ Forum:
- https://www.infineonforums.com/forums/13-Aurix-Forum

Revision history

Revision	Description of change
V1.0.1	Update of version to be in line with the code example’s version
V1.0.0	Initial version

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

Edition 2020-12
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
I2C_Read_Ext_Device_1_KIT_TC397_TFT

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie") .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

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.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

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.