

XC800 System Architecture

Part 2 of 2



Agenda

- Block Diagram
- Architecture
- System Control Unit
 - EVR
 - RESET
 - Clock System
 - Interrupt Controller
 - Power Management
 - Embedded Memories
- Device Access

XC878 On-Chip Debug Support (OCDS)

- Two interfaces can be used to access the OCDS system
 - The **JTAG** interface is the primary channel
 - The **UART** is an alternative channel
- Two dedicated pins are used for external configuration
 - **TMS** – JTAG activate
 - ↪ active high
 - ↪ integrated pull-down
 - **MBC** – Bootstrap Loader activate
 - ↪ active low
 - ↪ external pull-up required

XC878 Bootstrap Loader

- Different boot modes on reset via different **pin configurations**
- BSL mode allows easy and quick flashing via serial interface

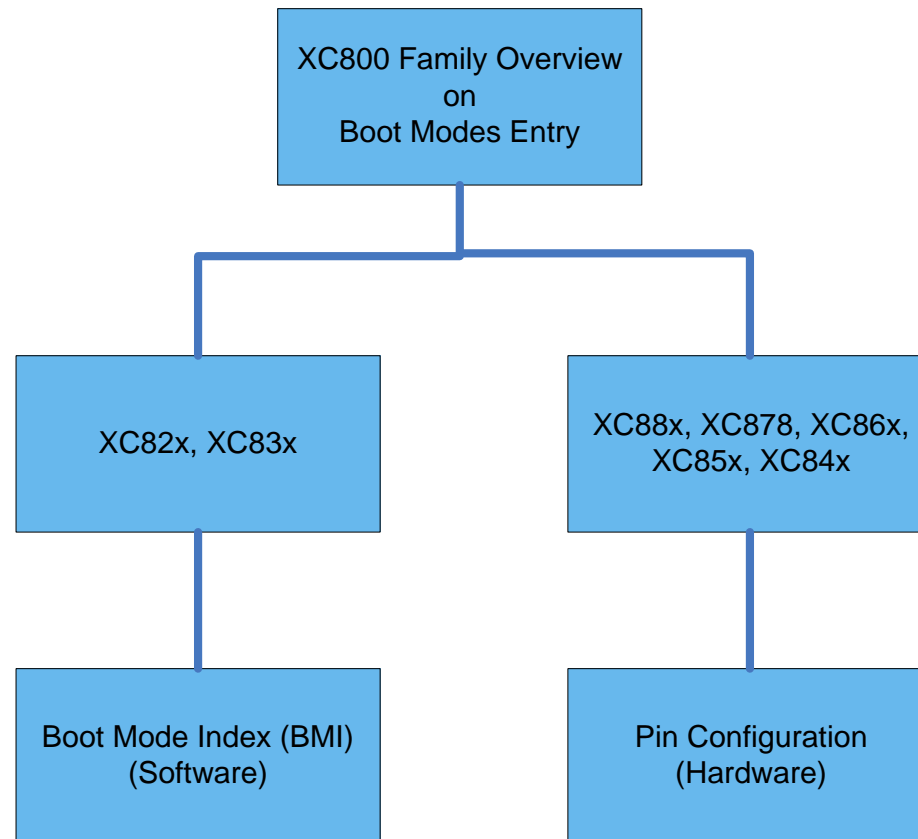
Bootstrap Loader Options	
UART	download to XRAM or Flash
LIN	
CAN	download to XRAM
ALTERNATE	jumps to user defined Flash address

Mark	„C“	„L“	„CM“	„ - “
Boot				
UART	x	-	x	x
LIN	-	x		-
CAN	x	-	x	-

- BSL is entered
 - with MBC = 0 (always)
 - with MBC = 1 (if flash @0x0000 is zero)
- Booting only on pins
 - P1.0/RXD/RXDC0_0
 - P1.1/TXD/TXDC0_0
- Booting always supports auto-baudrate detection
- MultiCan Booting
 - point-to-point connection
 - external clock required

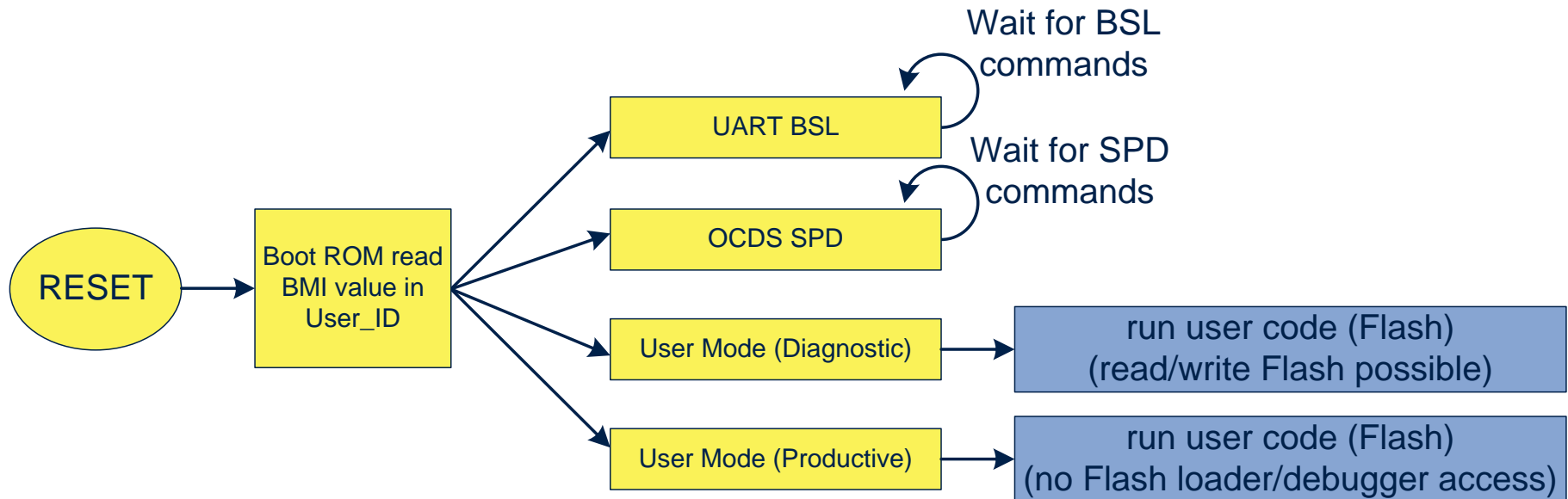
XC82x/XC83x

- After reset, XC800 devices will enter different Boot Modes via pin configurations
- For XC82x/XC83x however, the Boot Mode is determined by the BMI value
 - BMI and /BMI are part of User Identification (USER_ID) in Flash



XC82x/XC83x BMI Feature Overview

- Upon power up, XC82x/XC83x will enter one of the Boot Modes:
 - ❑ UART boot-loader mode (BSL)
 - ❑ On-chip debug (OCDS) mode
 - ❑ User mode (diagnostic)
 - ❑ User mode (productive)



BMI – UART Boot-Loader (BSL) Mode

- This is the default value
- Download and erase code via UART
 - After power-up, the chip will wait for baud rate synchronization command
 - Communicate with the device via full or half-duplex UART (P0.6/P0.5 for XC82x and P3.2/P0.7 for XC83x)
 - Download and execute code in XRAM/Flash, erase code in Flash, Program 4 bytes of USER_ID
 - FLOAD support
- BMI=0x00; /BMI=0x00 or 0xFF

BMI – OCDS Mode

- Recommended when the user has just started to develop and debug his code
 - After power-up, execution will stop at Flash address 0x0000
 - Access to full debug functionality using SPD pin
- e.g. BMI=0x60; /BMI=0x9F (SPD_0)

BMI – User Mode (Diagnostic)

- Recommended for testing the code in a real application, debug and programming functionality via SPD pin still available
 - After power-up, code execution will start at Flash address 0x0000
 - Specified SPD pin is automatically configured to allow debugger and Flash Loader access
 - The user can stop the code and read/write to Flash via SPD
- e.g. BMI=0x50; /BMI=0xAF (SPD_0)

BMI – User Mode (Productive)

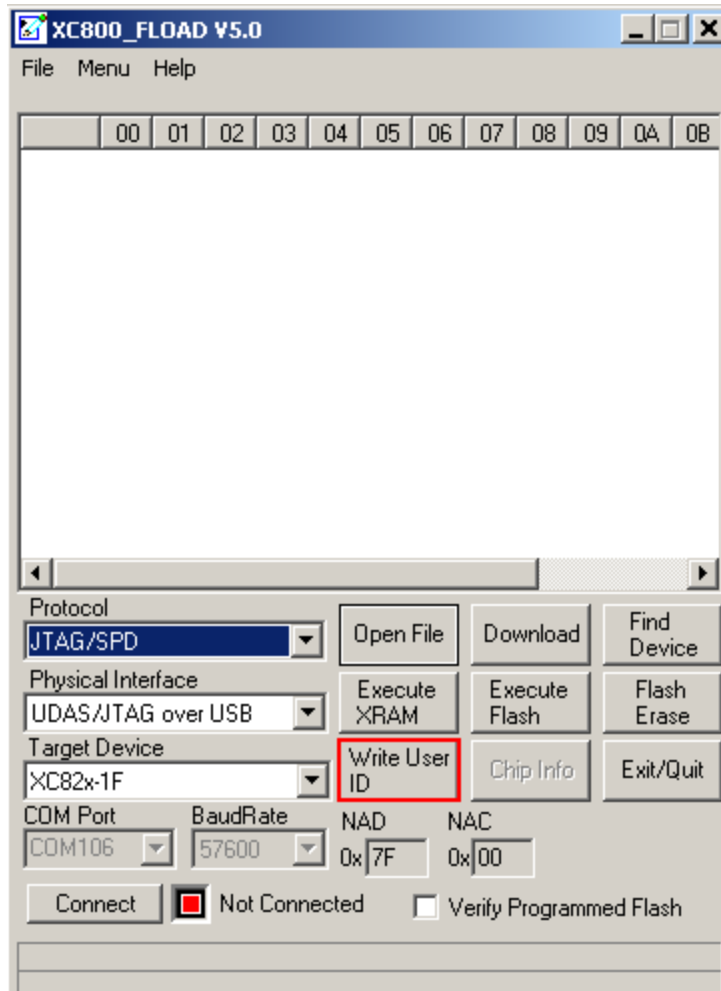
- Only if the code is **FULLY TESTED** and **NO MORE MODIFICATION** is necessary
 - Upon power up, code execution will start to from Flash address 0x0000
 - **MEMORY PROTECTION** by not allowing external tools to access (read/write) Flash memory
- UART BSL mode can be re-entered by embedding a call to BR_UART_BSL Boot ROM user routine (0xDFEA) in the user code. The USER_ID can also be reprogrammed by embedding a call in user code.
- If in doubt, use User Mode (Diagnostic)
- BMI=0x10; /BMI=0xEF

BMI – User Mode (Productive)

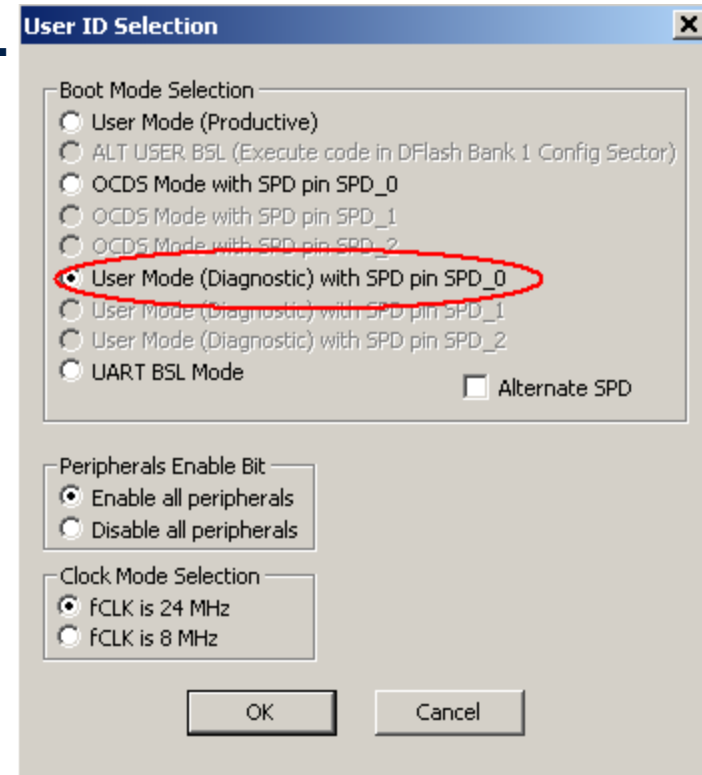
- No external access to the device once it is programmed to User Mode (Productive), unless there is a **specific routine** already ***embedded in user code***
 - To be called under ***user-defined conditions***
- This routine has one of these instructions:
 1. LJMP to BR_UART_BSL Boot ROM user routine (0xDFEA) to re-enter BSL Mode (recommended)
 2. LCALL of BR_PROG_USER_ID user routine (0xDFE4) to re-program USER_ID to enter other mode

Tools for programming the BMI FLOAD

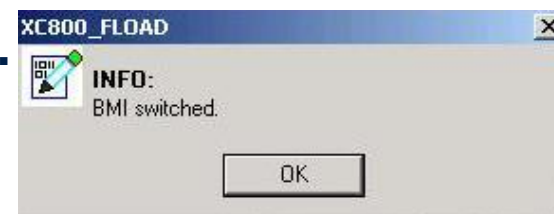
1.



2.



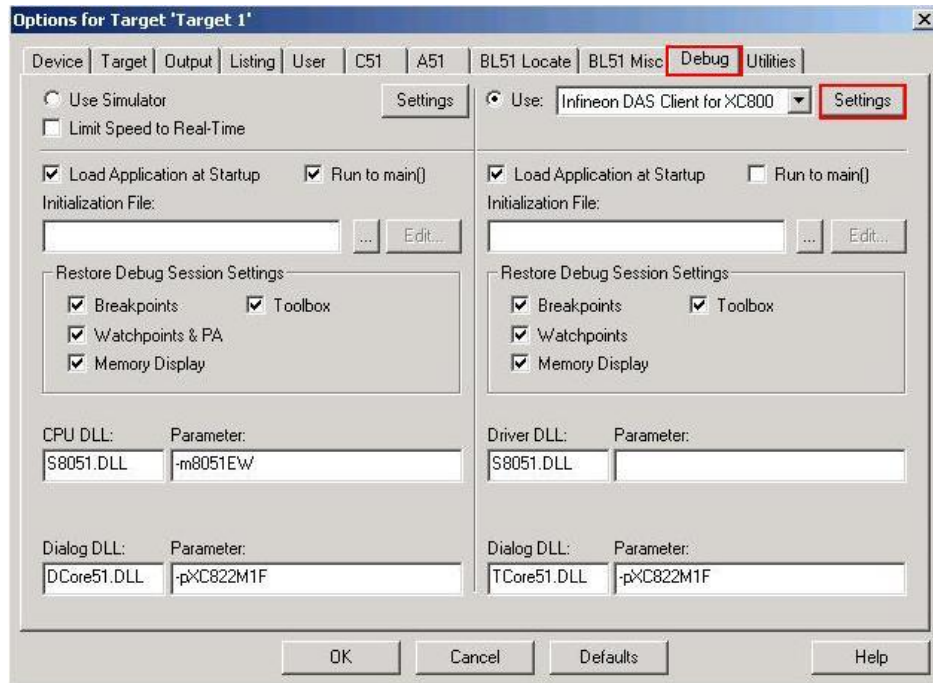
3.



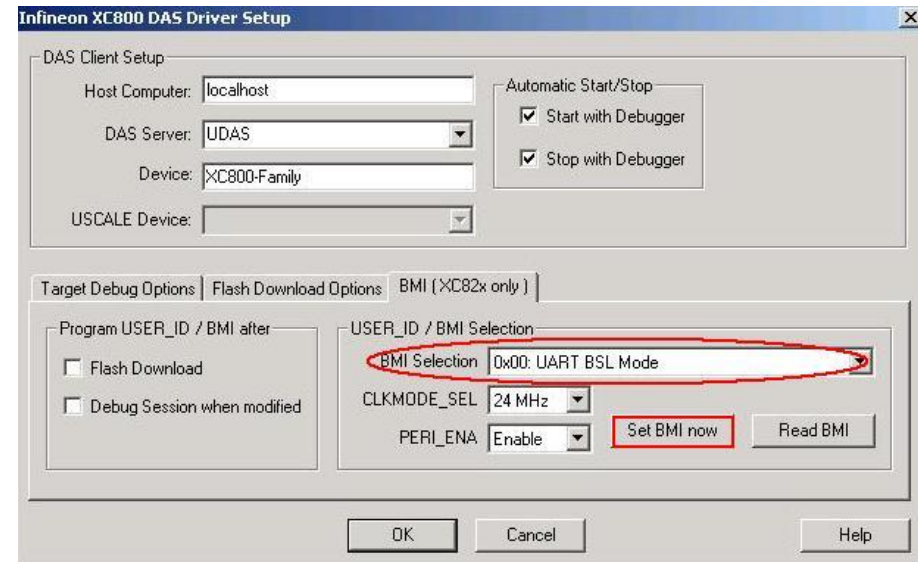
Tools for programming BMI value

KEIL uVision 4

1.



2.

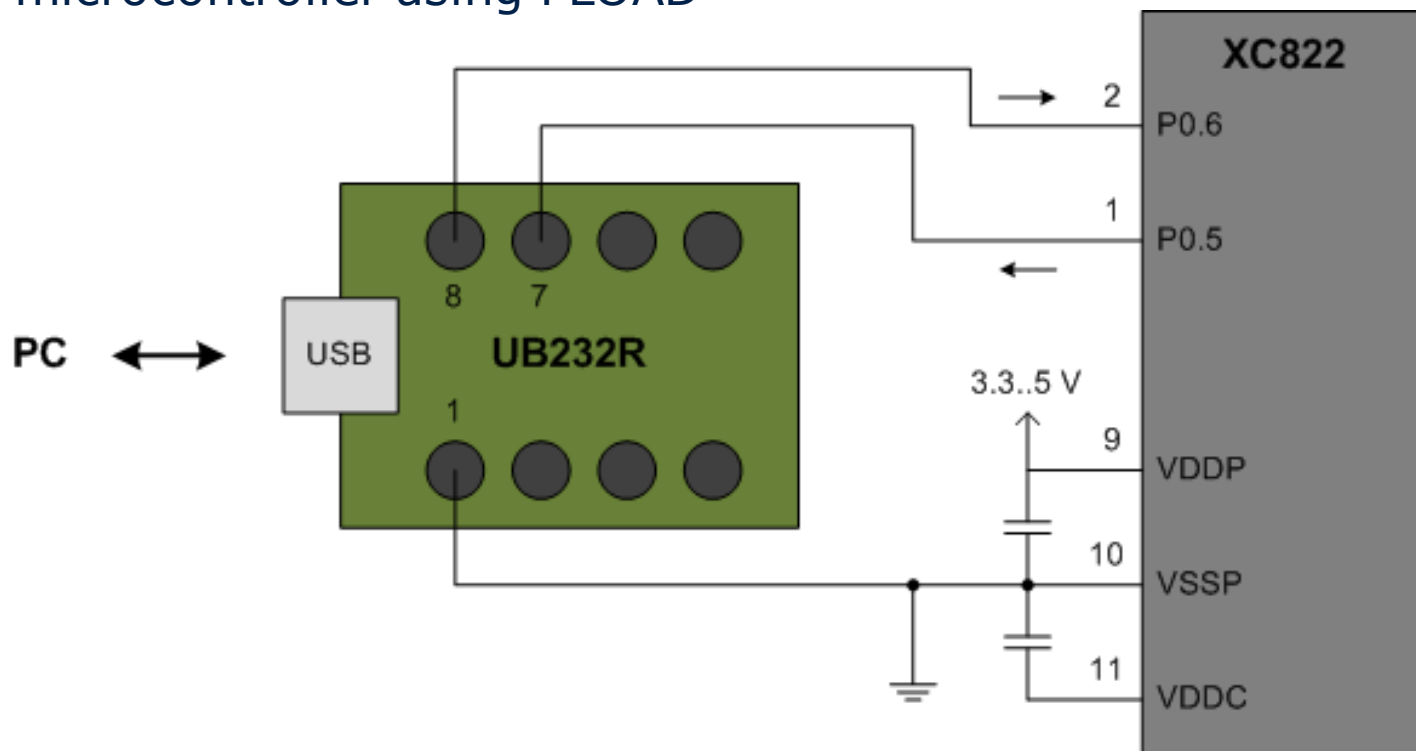
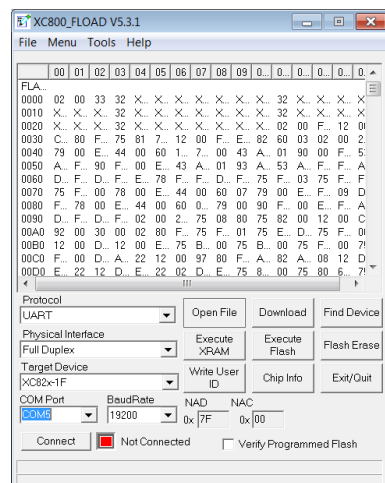
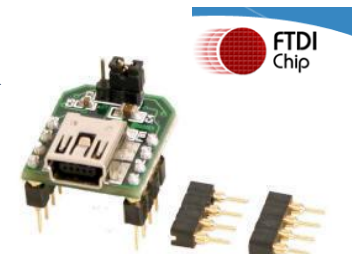


3.



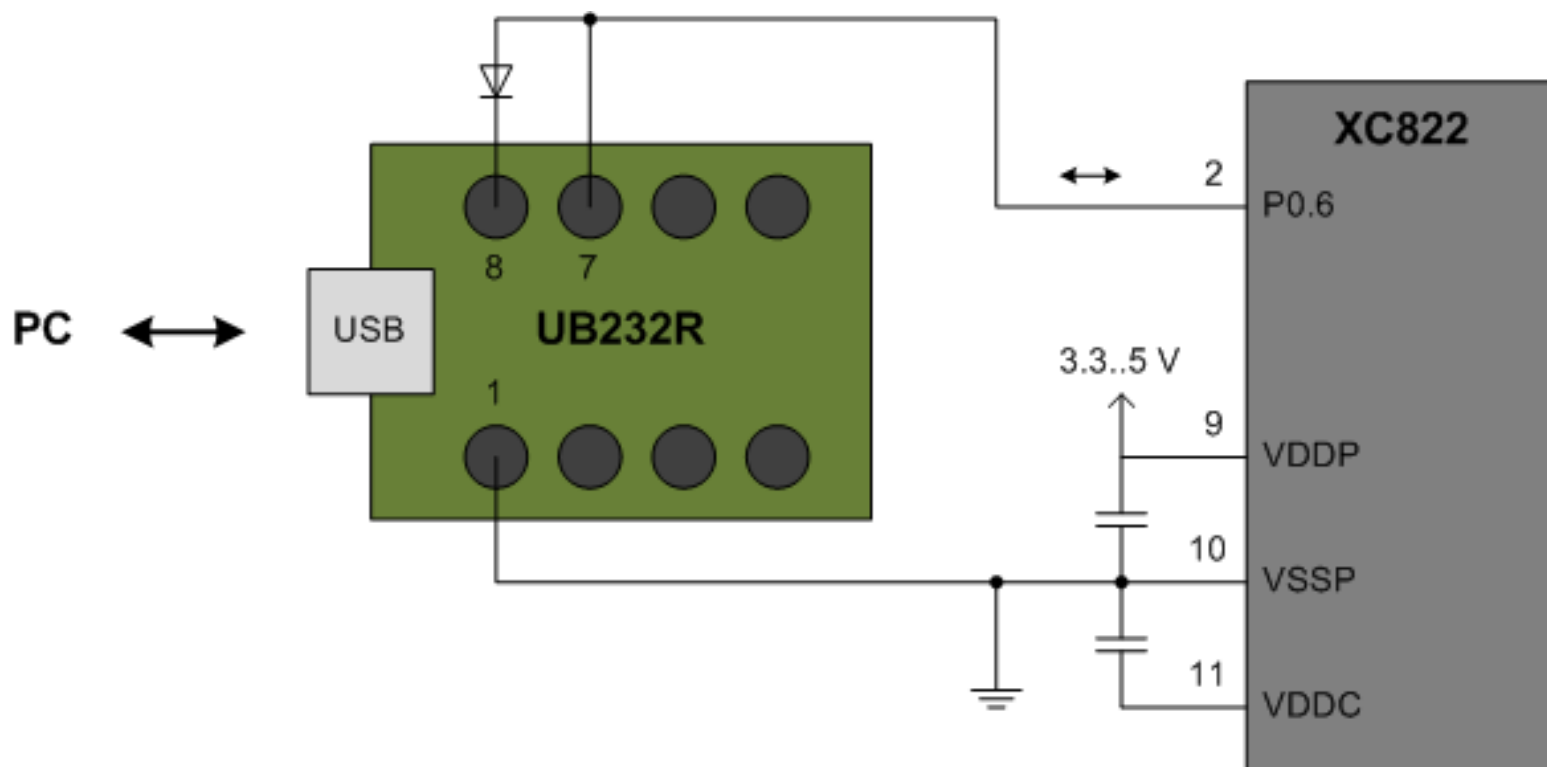
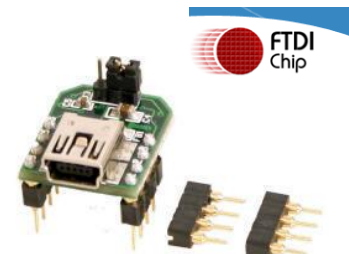
Programming Access – Example 1

- **UB232R:** <http://www.ftdichip.com/Products/Modules/DevelopmentModules.htm#UB232R>
- Full-Duplex UART protocol
- XC82x in UART BSL Mode
- On XC83x, connect P3.2 and P0.7 instead
- Connect to the microcontroller using FLOAD



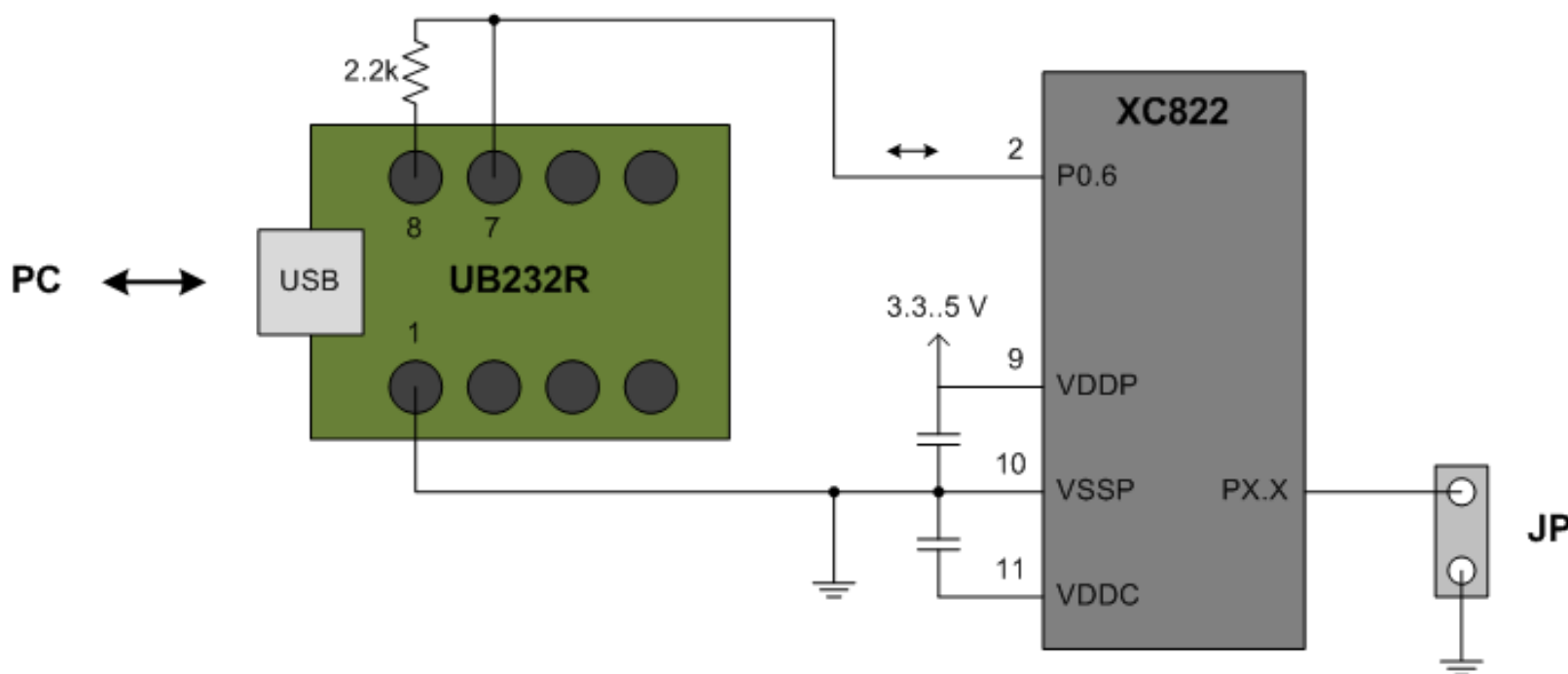
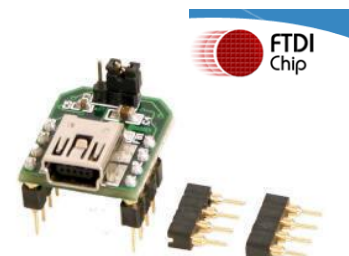
Programming Access – Example 2

- UB232R: <http://www.ftdichip.com/Products/Modules/DevelopmentModules.htm#UB232>
- Half-Duplex UART protocol
- XC82x in UART BSL Mode
- On XC83x, connect P3.2 instead



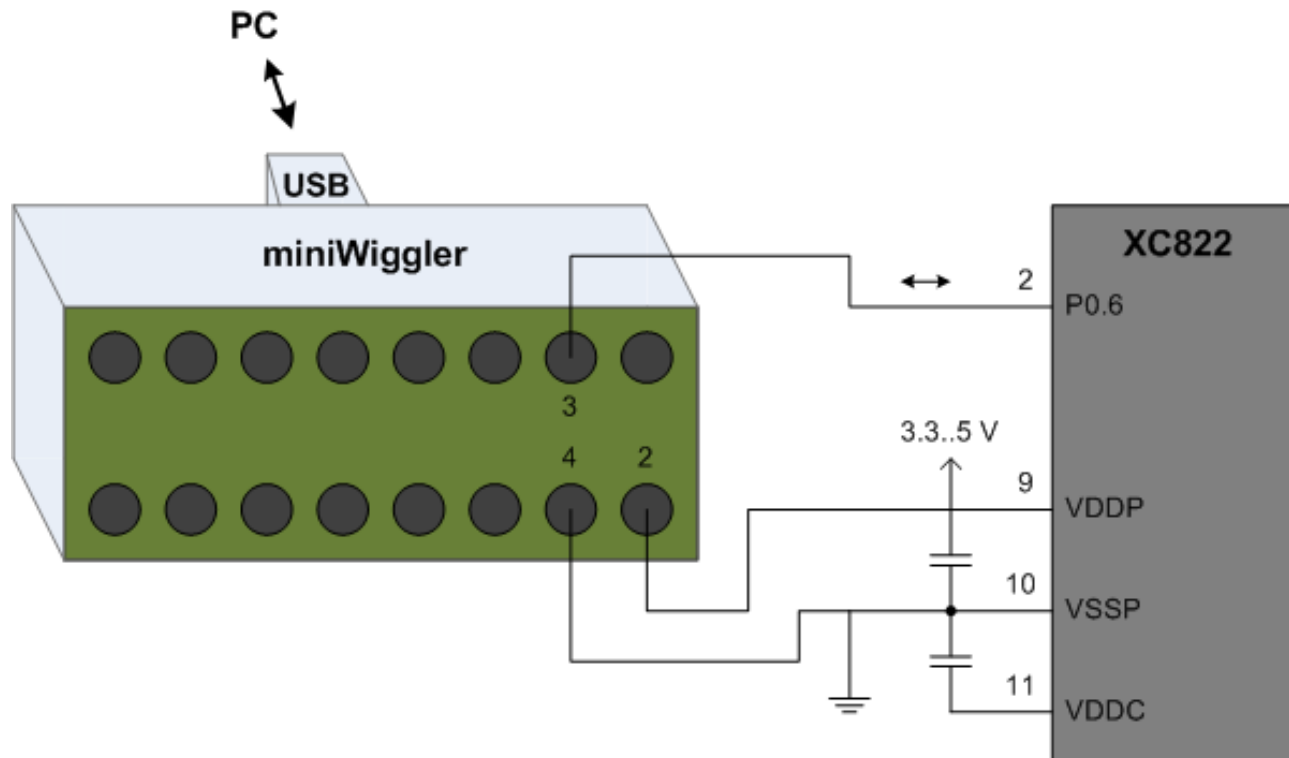
Programming Access – Example 3

- **UB232R:** <http://www.ftdichip.com/Products/Modules/DevelopmentModules.htm#UB232>
- Half-Duplex UART protocol
- XC82x in User Mode (Productive)
 - User software ensures that UART Boot-loader mode is entered if PX.X is connected to ground on startup
- On XC83x, connect P3.2 instead



Programming Access – Example 4

- DAP miniWiggler: www.infineon.com/miniwiggler
- SPD protocol
- XC82x in OCDS Mode or User Mode (Diagnostic)
- On XC83x, connect P3.2 instead
- External 3.3-5V supply must be provided



Programming Access to a New Device

■ New XC82x and XC83x devices are in UART BSL mode

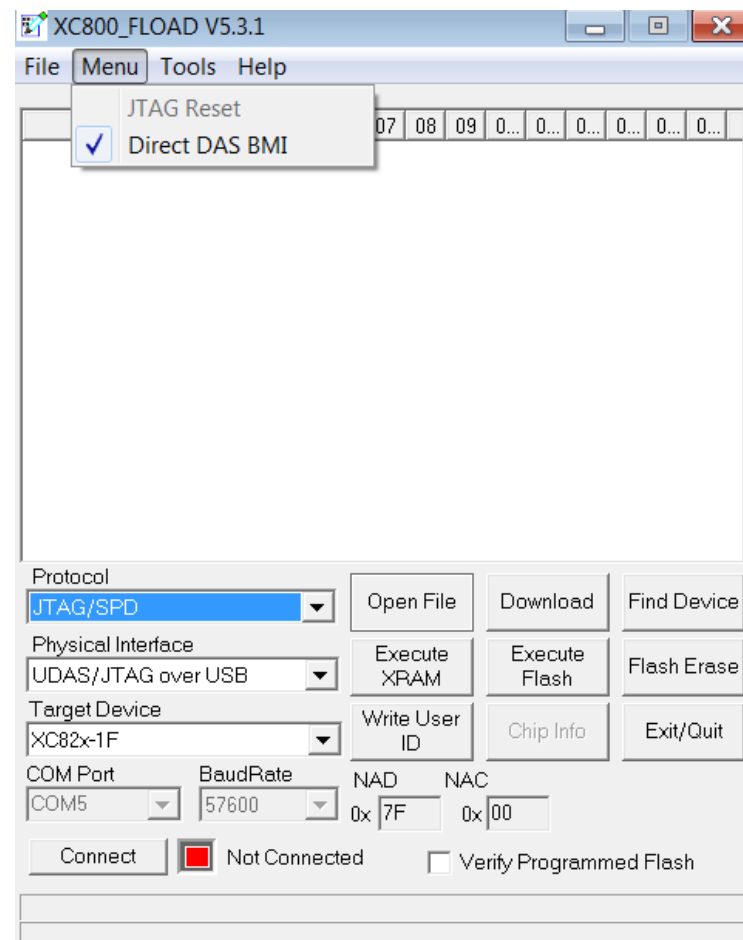
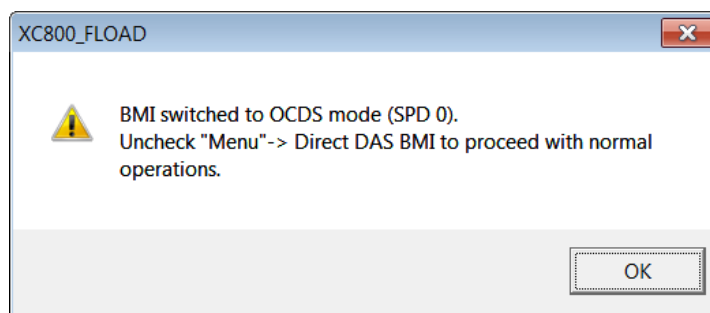
□ Device access via UART

→ Use UB232R

□ Alternatively use the miniWiggler

→ First-time connection steps with FLOAD:

1. In "Menu" → check "Direct DAS BMI"
2. Click on "Connect"
3. The BMI is switched to OCDS mode
4. In "Menu" → uncheck "Direct DAS BMI"
5. Click on "Connect"



www.infineon.com/xc800

support@infineon.com

ENERGY EFFICIENCY COMMUNICATIONS SECURITY

Innovative semiconductor solutions for energy efficiency, communications and security.

