

KE06 CAN Bootloader Design

1. Introduction

There are many applications or products that upgrade firmware to fix bugs or sometimes to improve performance. Most of these applications and products do not use the dedicated debug interface, but instead use communication interfaces, such as UART, USB, and I²C. In these cases, a serial boot loader is required to perform a firmware upgrade using one of the communication interfaces without debugger or dedicated program tools. This application note provides guidelines to design a boot loader on the KE06 MCU with a CAN interface.

Contents

1. Introduction.....	1
2. Overview	2
3. Hardware architecture.....	2
4. Software architecture.....	4
4.1. Convert board.....	4
4.2. Target board.....	4
5. Running the demo	5
6. Conclusion	8
7. Revision history	8



2. Overview

The boot loader is normally a built-in firmware that is implemented to program the application code to flash through the communication interface. This application note describes the procedure to use the FRDM-KE06Z board to convert the UART data from the PC terminal to the CAN bus. In addition, it explains the procedure to communicate with the target board, FRDM-KE06Z, and to implement the updates of target application code.

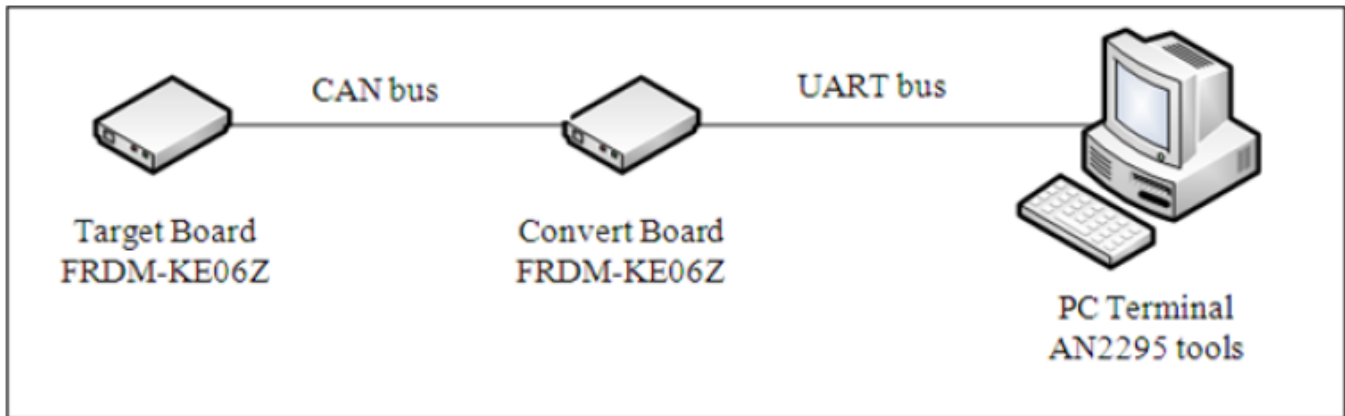


Figure 1. Top Level view

The boot loader uses the features of the XMODEM protocol.

The UART-CAN adapter board uses the Freedom board, FRDM-KE06Z, to convert the UART bus to CAN bus and to repackage the data transfer to the target board. The target board will program the application code to flash.

The CAN boot loader sample code can directly run on the FRDM-KE06Z board, and will be downloaded to the target board. A UART-CAN adapter code is also provided to run on FRDM-KE06 enabling the board to be used as a UART-CAN adapter. The project “PIT_demo” is used to generate a bin file, which can be downloaded using PC software.

3. Hardware architecture

The CAN boot loader requires two FRDM-KE06 boards as shown below:

Table 1. Hardware requirements

Board	Function
FRDM-KE06-A	UART-CAN adapter
FRDM-KE06-B	Boot loader target

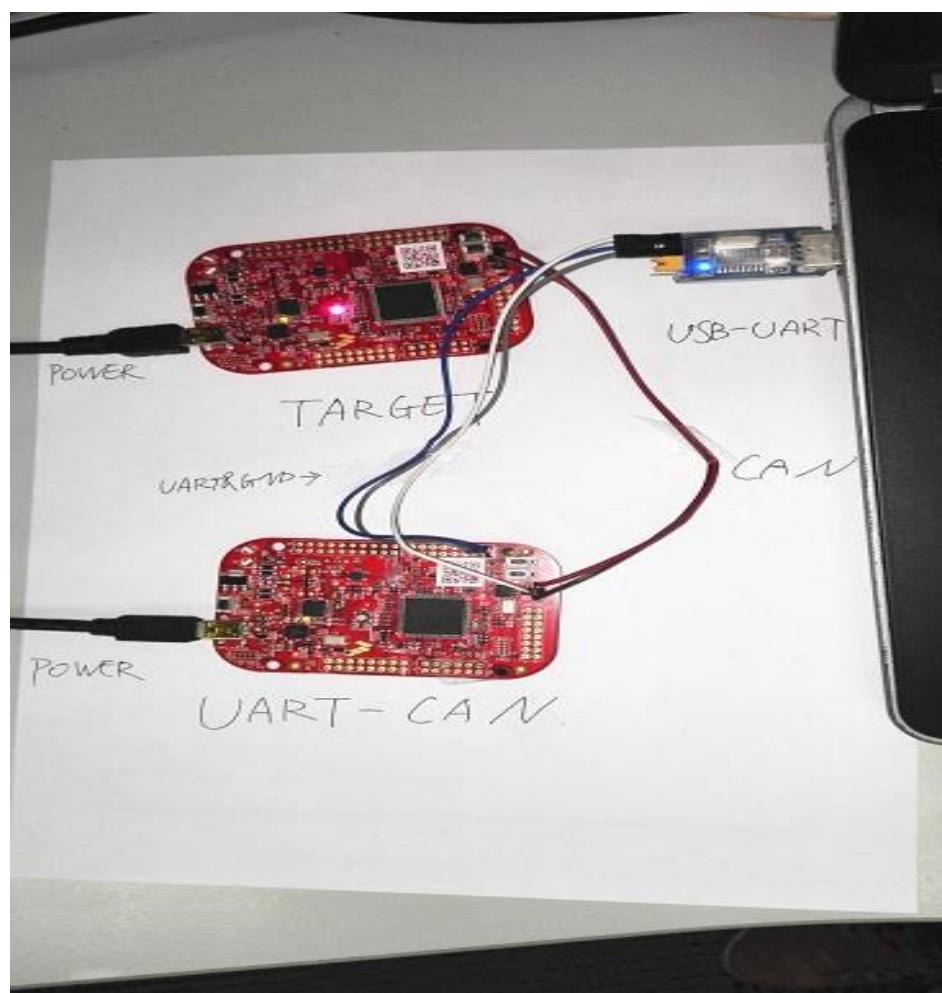


Figure 2. Hardware setup

NOTE

The UART-CAN adapter board requires some reworks because the on-board K20 OpenSDA debugger cannot meet the boot loader transmit speed requirements. One external USB-UART board is required in order to receive data from the PC.

To change the default UART port connected on an external UART-USB board, cut out the R53 and R52 and connect R78 and R79. This is shown in the following figure:

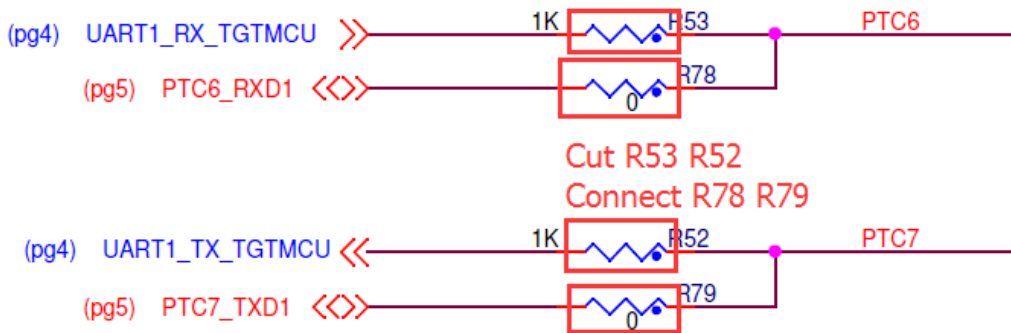


Figure 3. OpenSDA UART rework

The following steps show how to build demo hardware environments:

1. Ensure two FRDM-KE06 board are functional, the target board's OpenSDA has successfully connected to the PC, and the USB CDC COM port has been enumerated successfully.
2. Connect the CAN connector(J11) to the two boards

4. Software architecture

4.1. Convert board

The PC cannot communicate with the target board via CAN directly. To enable communication a convert board is required to transfer the UART signal (PC end) into the CAN signal (target board). Therefore, the convert board communicates with the PC terminal. The convert board repackages the data frame with data length, and also the receive or transmit data package with the target board using the CAN bus.

4.2. Target board

The target board contains a built-in boot loader code. After startup the target board firstly checks the work mode of the boot loader code. That is whether it is in boot mode or user code mode. There are various methods to perform this check. For example, by checking the level of an external GPIO, if the GPIO pin is low, then it will enter into boot mode to run boot loader and if the GPIO pin is high, then it will enter user code mode to run the application code.

If you enter boot loader mode, the boot loader will communicate with the USB-CAN adapter and download the target image via the Xmodem protocol. It will then jump to the user application. Or in user mode, the program will jump to boot vector directly and initialize SP and write the SCB->VTOR register with the user interrupt vector address.

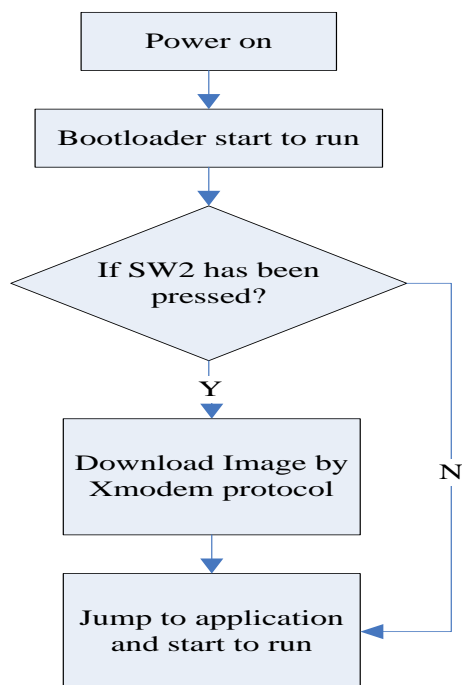


Figure 4. Software architecture

The software code package is designed based on the Freescale ke_drv_lib and Freescale Kobe boot loader framework.

The boot loader and CAN parameters are as listed below:

Table 2. Boot loader parameters

Parameters	Value
App start address	0x5000
Target board CAN Rx address	0x801
Target board CAN Tx address	0x802
UART-CAN adapter CAN Rx address	0x802
UART-CAN adapter CAN Tx address	0x801
UART-CAN adapter baudrate	115200

5. Running the demo

To run the CAN boot loader demo, follow the steps below:

1. Unzip the code package, open *ke06_can_boot loader\project\iar\ke06*. You should then see the following three project folders:

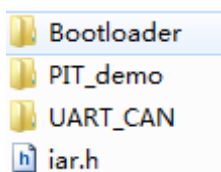


Figure 5. Demo root folder

The boot loader folder contains the boot loader program which must be programmed into the target board. The UART_CAN folder contains the firmware which turns the FRDM-KE06 board into an UART_CAN adapter. The PIT_demo folder provides a user application example.

Table 3. Boot loader parameters

Project	Description
Bootloader	Bootloader project, need to program to target board
PIT_demo	Example of application project. Need to be compiled to generate bin file.
UART-CAN	UART-CAN bridge board project. Should program into UART-CAN adapter board.

- Download the UART_CAN firmware onto one FRDM-KE06 board (should be reworked on OpenSDA UART).
- Download boot loader firmware onto target FRDM-KE06 board.
- Compile the PIT demo. After the demo is successfully compiled, the bin image can be found at:
ke06_can_boot loader\project\iar\ke06\PIT_demo\FLASH_128KB_PFLASH\Exe
- Open Tera Term, create a new session, and select the UART-CAN COM port open as: *115200-8-none-1-none*.

NOTE

The Tera term default baud rate is 9600, not 115200.

- Select File->Transfer->XMODEM->send, select PIT_demo_frdm.bin

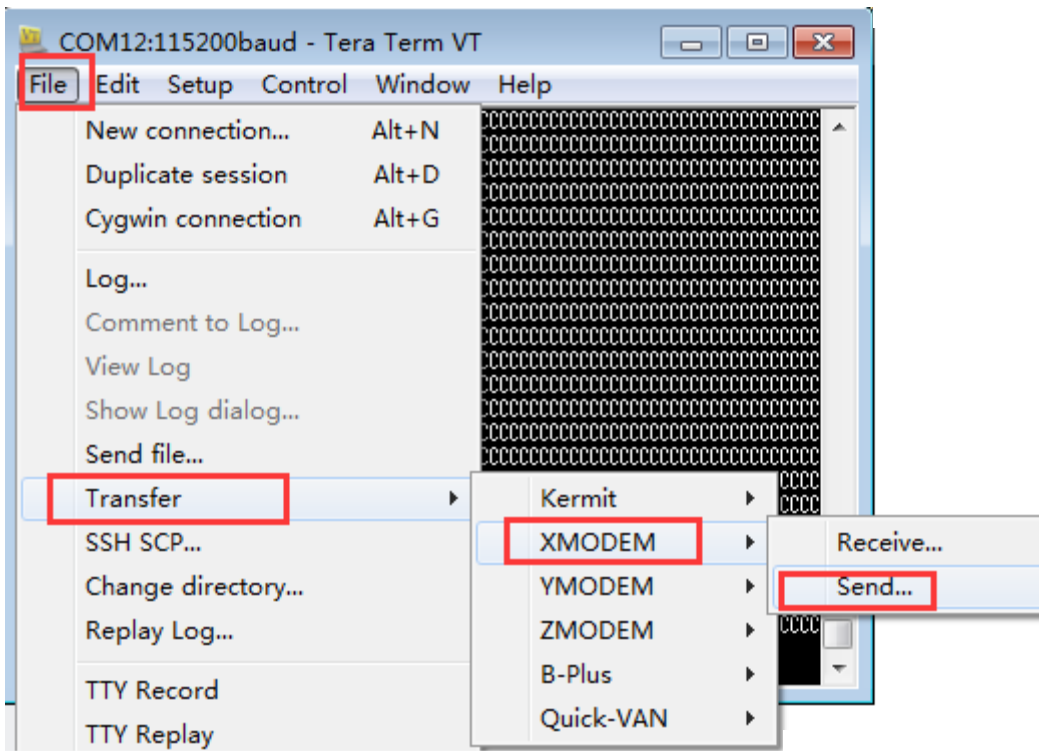


Figure 6. Send XMODEM option to send file

7. Press and hold the SW2 button on the target board while pushing the RESET button. Then release the SW2 button. This will put boot loader code into boot mode and Tera Term should print 'C' which indicates that the boot loader is ready to receive data:

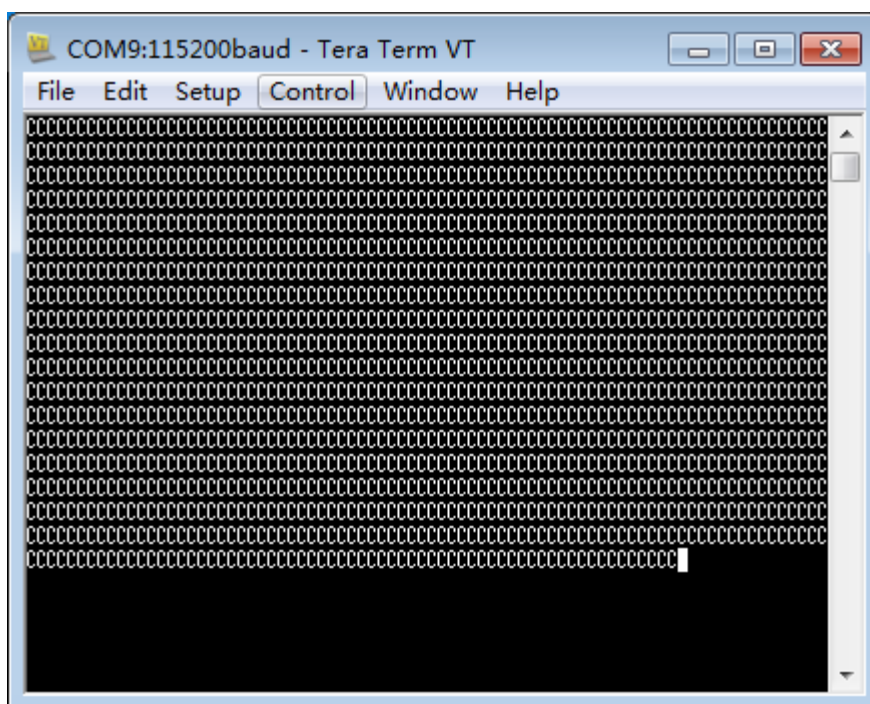


Figure 7. Tera term connected with target board

8. The downloading process will begin automatically. After image download is complete, the boot loader will jump to the application code. The red LED should be blinking.

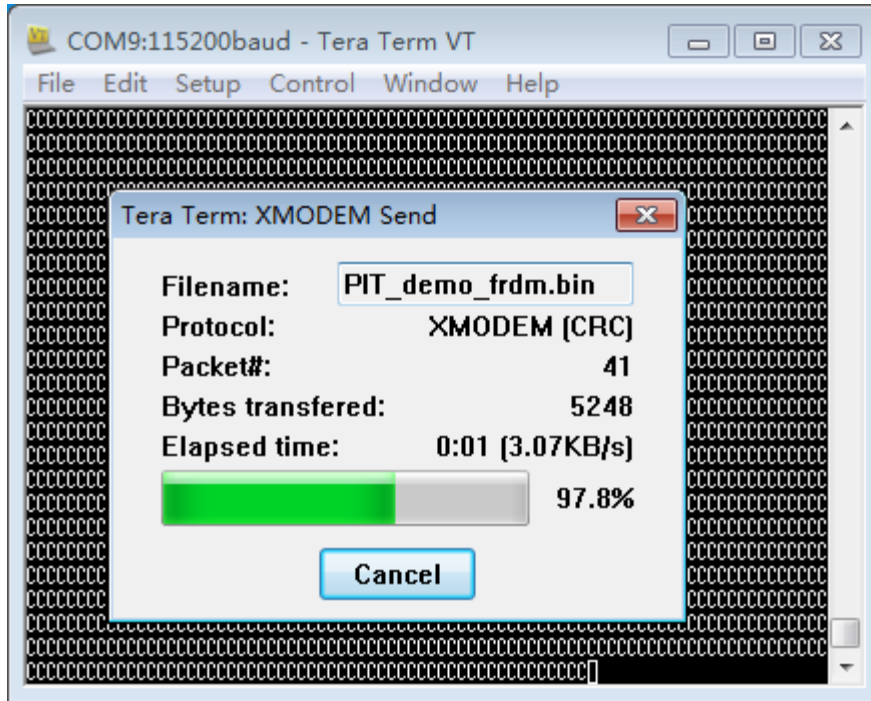


Figure 8. Downloading image

6. Conclusion

This application note describes the procedure to implement the CAN boat loader by using a bridge board as a convert board and another board as target board. All C source file described in this application note can be found in the attached software package, that also includes ready-to-build tool chain projects (IAR and Keil).

7. Revision history

Table 4. Revision history

Revision number	Date	Substantive changes
0	11/2015	Initial release
1	02/2016	Sections 5, “Running the demo” and 6, “Conclusion” updated

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off.

ARM, the ARM Powered logo, and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All other product or service names are the property of their respective owners. All rights reserved.

© 2016 Freescale Semiconductor, Inc.

Document Number: AN5219
Rev. 1
02/2016

