WORKSHOP - USING SERIAL BOOTLOADER EXAMPLE

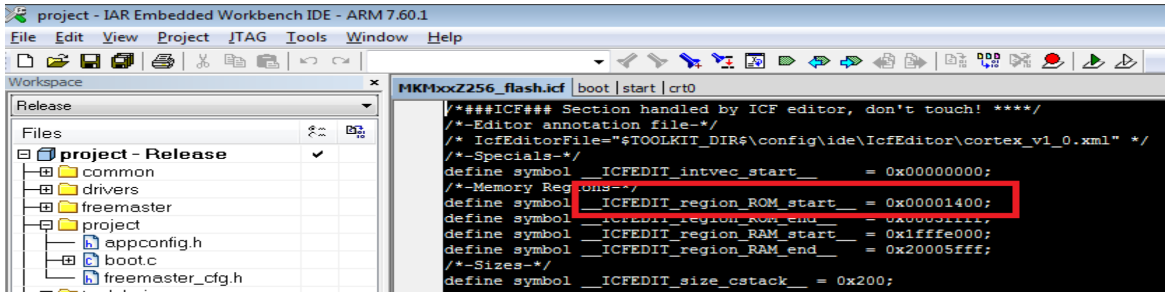
PAVEL KRENEK

26, APRIL, 2016

SECURE CONNECTIONS
FOR A SMARTER WORLD

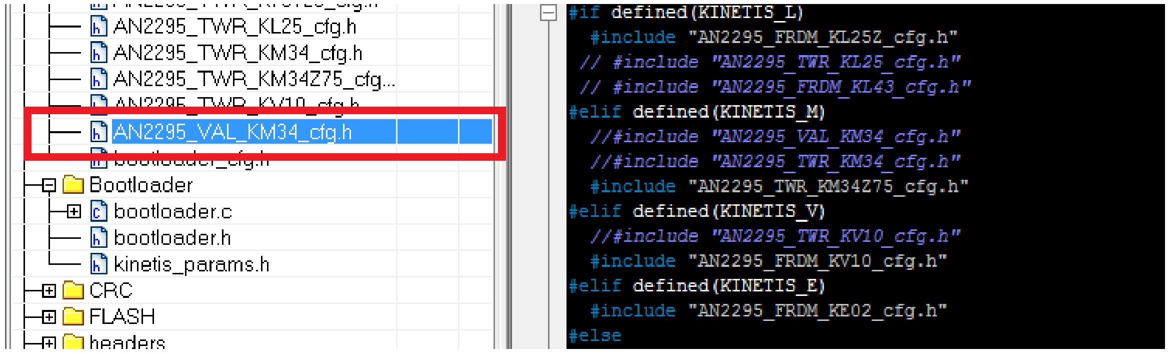
Bootloader workshop agenda

1. Short theory about the bootloader.
2. Create a short application on TWR-KM34Z75 with all needed modifications (using KM34Z75 bare metal drivers).



```
/*###ICF### Section handled by ICF editor, don't touch! ****/
/*-Editor annotation file-*/
/* IcfEditorFile="$TOOLKIT_DIR$\config\ide\IcfEditor\cortex_v1_0.xml" */
/*-Specials-*/
define symbol __ICFEDIT_intvec_start__     = 0x00000000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x00001400;
define symbol __ICFEDIT_region_ROM_end__   = 0x[illegible];
define symbol __ICFEDIT_region_RAM_start__ = 0x1fffe000;
define symbol __ICFEDIT_region_RAM_end__   = 0x20005fff;
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x200;
```

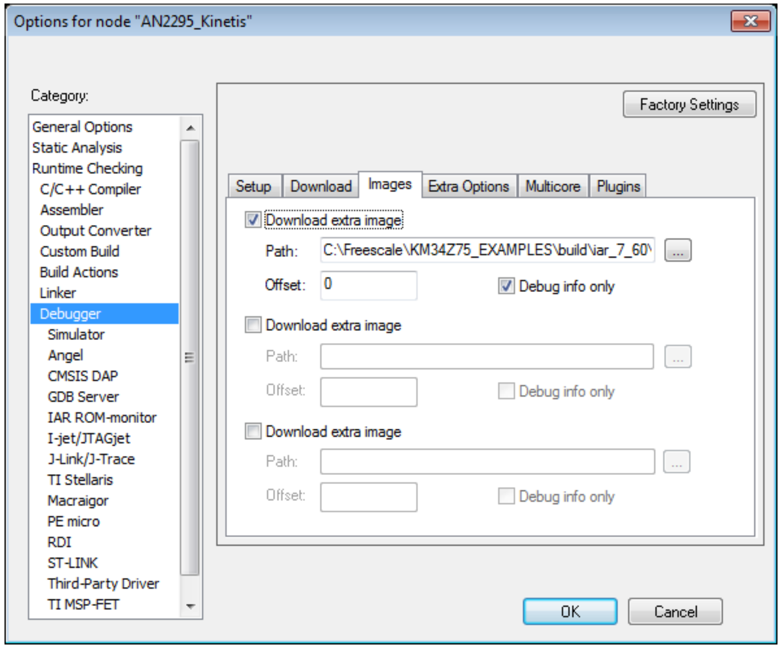
3. Correct configuration of the bootloader for (Kinetis M).



```
#if defined(KINETIS_L)
  #include "AN2295_FRDM_KL25Z_cfg.h"
 // #include "AN2295_TWR_KL25_cfg.h"
 // #include "AN2295_FRDM_KL43_cfg.h"
#elif defined(KINETIS_M)
  //#include "AN2295_VAL_KM34_cfg.h"
  //#include "AN2295_TWR_KM34_cfg.h"
  #include "AN2295_TWR_KM34Z75_cfg.h"
#elif defined(KINETIS_V)
  //#include "AN2295_TWR_KV10_cfg.h"
  #include "AN2295_FRDM_KV10_cfg.h"
#elif defined(KINETIS_E)
  #include "AN2295_FRDM_KE02_cfg.h"
#else
```

Bootloader workshop agenda

4. Using master PC application and final boot-loading procedure with extra image debugging in IAR (bootloader + user app).



Options for node "AN2295_Kinetis"

Category:

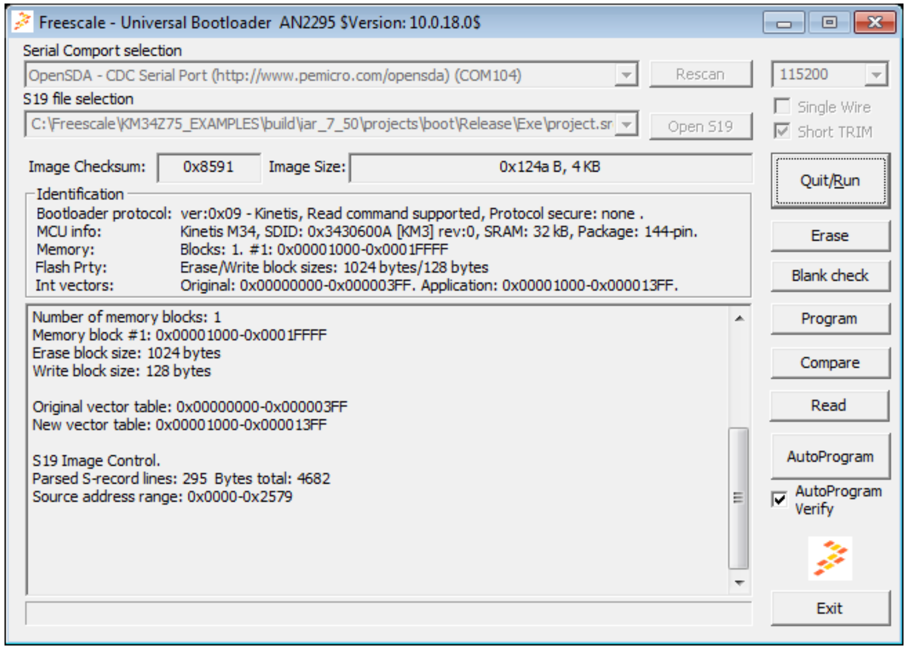
- General Options
- Static Analysis
- Runtime Checking
- C/C++ Compiler
- Assembler
- Output Converter
- Custom Build
- Build Actions
- Linker
- Debugger
- Simulator
- Angel
- CMSIS DAP
- GDB Server
- IAR ROM-monitor
- I-jet/JTAGjet
- J-Link/J-Trace
- TI Stellaris
- Macraigor
- PE micro
- RDI
- ST-LINK
- Third-Party Driver
- TI MSP-FET

Factory Settings
Setup | Download | Images | Extra Options | Multicore | Plugins

[✓] Download extra image
Path: C:\Freescale\KM34Z75_EXAMPLES\build\iar_7_60\
Offset: 0
[✓] Debug info only

[ ] Download extra image
Path:
Offset:
[ ] Debug info only

[ ] Download extra image
Path:
Offset:
[ ] Debug info only



Freescale - Universal Bootloader AN2295 $Version: 10.0.18.0$

Serial Comport selection
OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM104)
115200

S19 file selection
C:\Freescale\KM34Z75_EXAMPLES\build\iar_7_50\projects\boot\Release\Exe\project.sr

[ ] Single Wire
[✓] Short TRIM

Image Checksum: 0x8591   Image Size: 0x124a B, 4 KB

Identification

Bootloader protocol:	ver:0x09 - Kinetis, Read command supported, Protocol secure: none .
MCU info:	Kinetis M34, SDID: 0x3430600A [KM3] rev:0, SRAM: 32 kB, Package: 144-pin.
Memory:	Blocks: 1. #1: 0x00001000-0x0001FFFF
Flash Prty:	Erase/Write block sizes: 1024 bytes/128 bytes
Int vectors:	Original: 0x00000000-0x000003FF. Application: 0x00001000-0x000013FF.

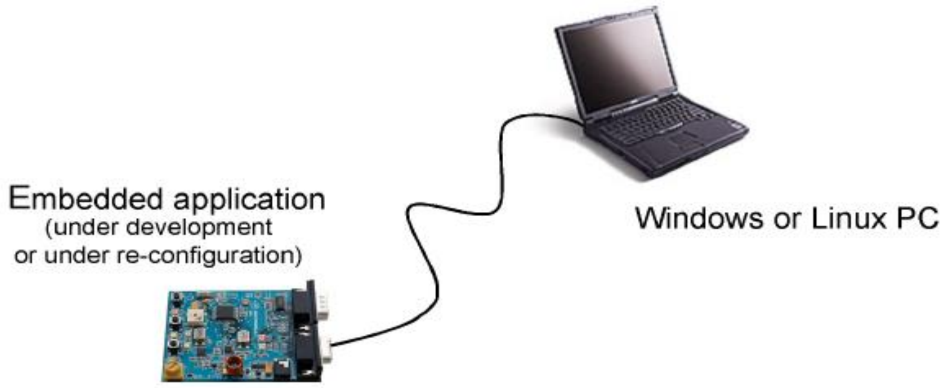
```
Number of memory blocks: 1
Memory block #1: 0x00001000-0x0001FFFF
Erase block size: 1024 bytes
Write block size: 128 bytes

Original vector table: 0x00000000-0x000003FF
New vector table: 0x00001000-0x000013FF

S19 Image Control.
Parsed S-record lines: 295  Bytes total: 4682
Source address range: 0x0000-0x2579
```

[✓] AutoProgram Verify

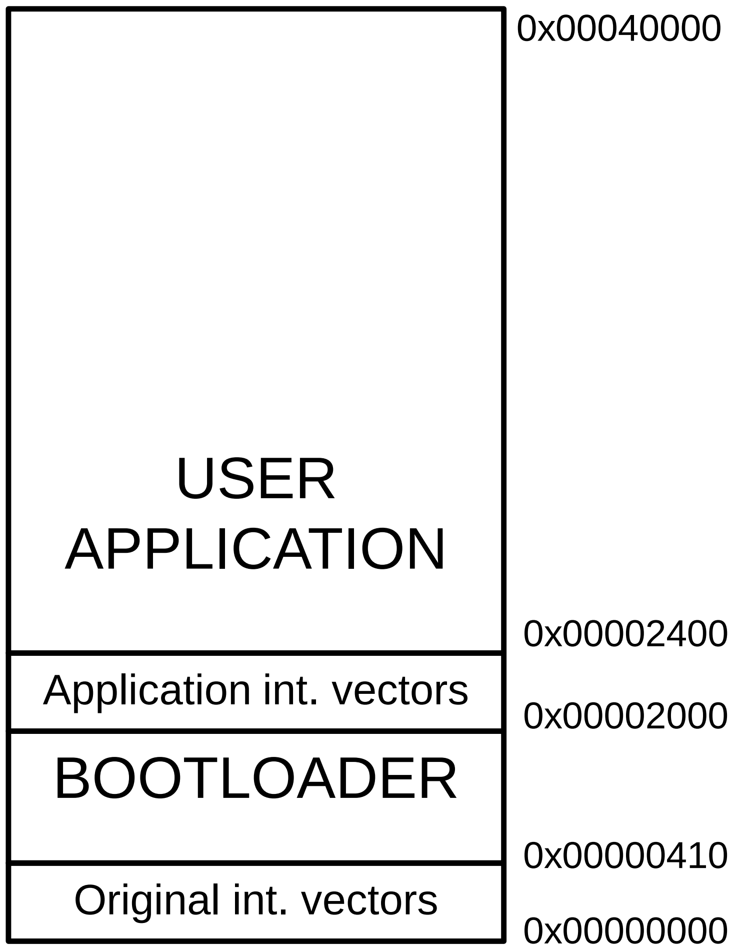
5. Q&A

“Short” about the theory of universal serial Bootloader

- A short program allows possibility to update existing firmware in MCU “in-circuit”
- Not intended to replace any of debugging tools
- Consist of Master PC app and Embedded slave application
- Uses simple SCI
- Offers a zero-cost solution to applications already equipped with a serial interface (USB)
- Supported platforms ?



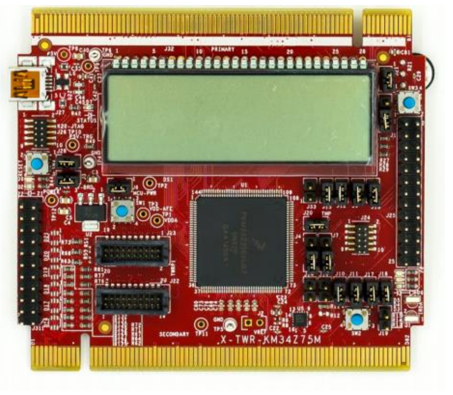
Memory structure of Kinetis M serial Bootloader



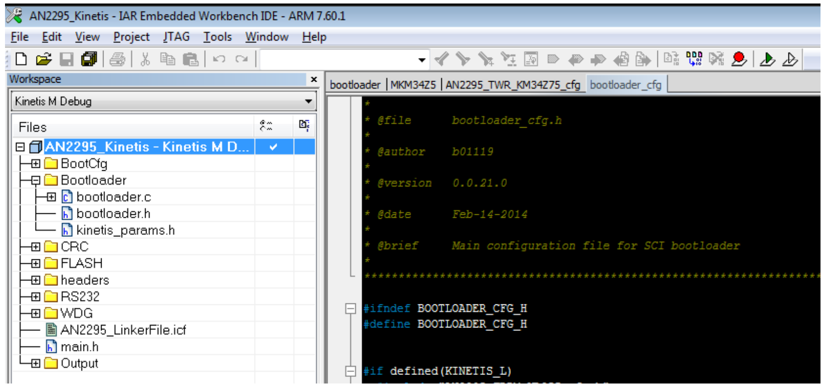
Pre-requisites

The following are needed to complete this work shop:

- Boards:
  - TWR-KM34Z75M (USB cable B-mini)

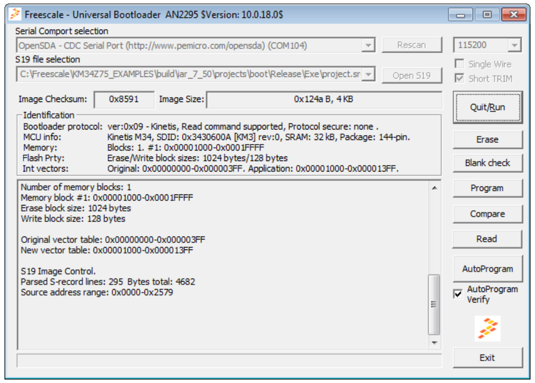


- Software:
  - IAR Embedded Workbench 7.60.1 or later: http://iar.com
  - AN2295SW Universal Bootloader sw package
  - Master Universal Bootloader AN2295 ver: 10.0.18.0



AN2295_Kinetis - IAR Embedded Workbench IDE - ARM 7.60.1

```
*
* @file      bootloader_cfg.h
*
* @author    b01119
*
* @version   0.0.21.0
*
* @date      Feb-14-2014
*
* @brief     Main configuration file for SCI bootloader
*
***********************************************************************

#ifndef BOOTLOADER_CFG_H
#define BOOTLOADER_CFG_H


#if defined(KINETIS_L)
```



Freescale - Universal Bootloader AN2295 $Version: 10.0.18.0$

Serial Comport selection: OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM104)

S19 file selection: C:\Freescale\KM34Z75_EXAMPLES\build\iar_7_50\projects\boot\Release\Exe\project.sr

115200

☐ Single Wire
☑ Short TRIM

Image Checksum: 0x8591 Image Size: 0x124a B, 4 KB

Identification

Bootloader protocol:	ver:0x09 - Kinetis, Read command supported, Protocol secure: none .
MCU info:	Kinetis M34, SDID: 0x3430600A [KM3] rev:0, SRAM: 32 kB, Package: 144-pin.
Memory:	Blocks: 1. #1: 0x00001000-0x0001FFFF
Flash Prty:	Erase/Write block sizes: 1024 bytes/128 bytes
Int vectors:	Original: 0x00000000-0x000003FF. Application: 0x00001000-0x000013FF.

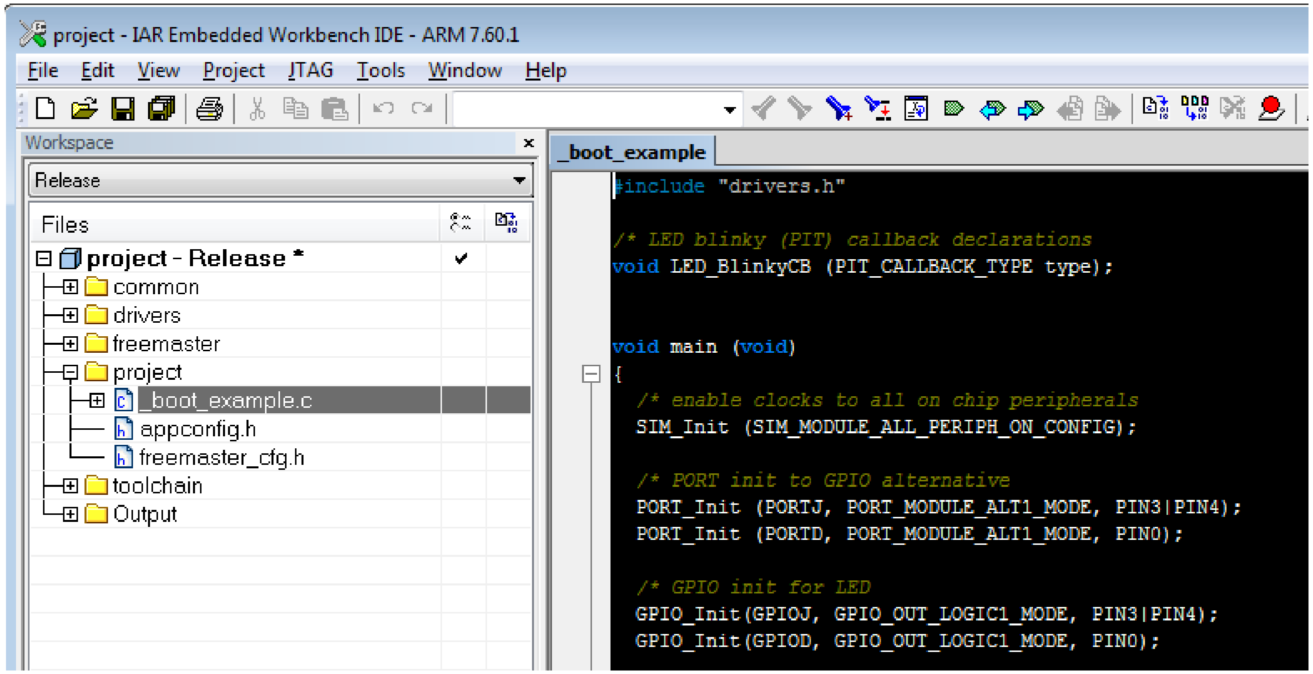
```
Number of memory blocks: 1
Memory block #1: 0x00001000-0x0001FFFF
Erase block size: 1024 bytes
Write block size: 128 bytes

Original vector table: 0x00000000-0x000003FF
New vector table: 0x00001000-0x000013FF

S19 Image Control.
Parsed S-record lines: 295  Bytes total: 4682
Source address range: 0x0000-0x2579
```

☑ AutoProgram Verify

Prepare the user application

Go the location where you installed the KM34 bare metal examples, and open the IAR Workspace found at:

<KM34Z75 bare_ installation path> \KM34Z75_EXAMPLES\build\iar_7_50\projects_boot_example\ project.eww

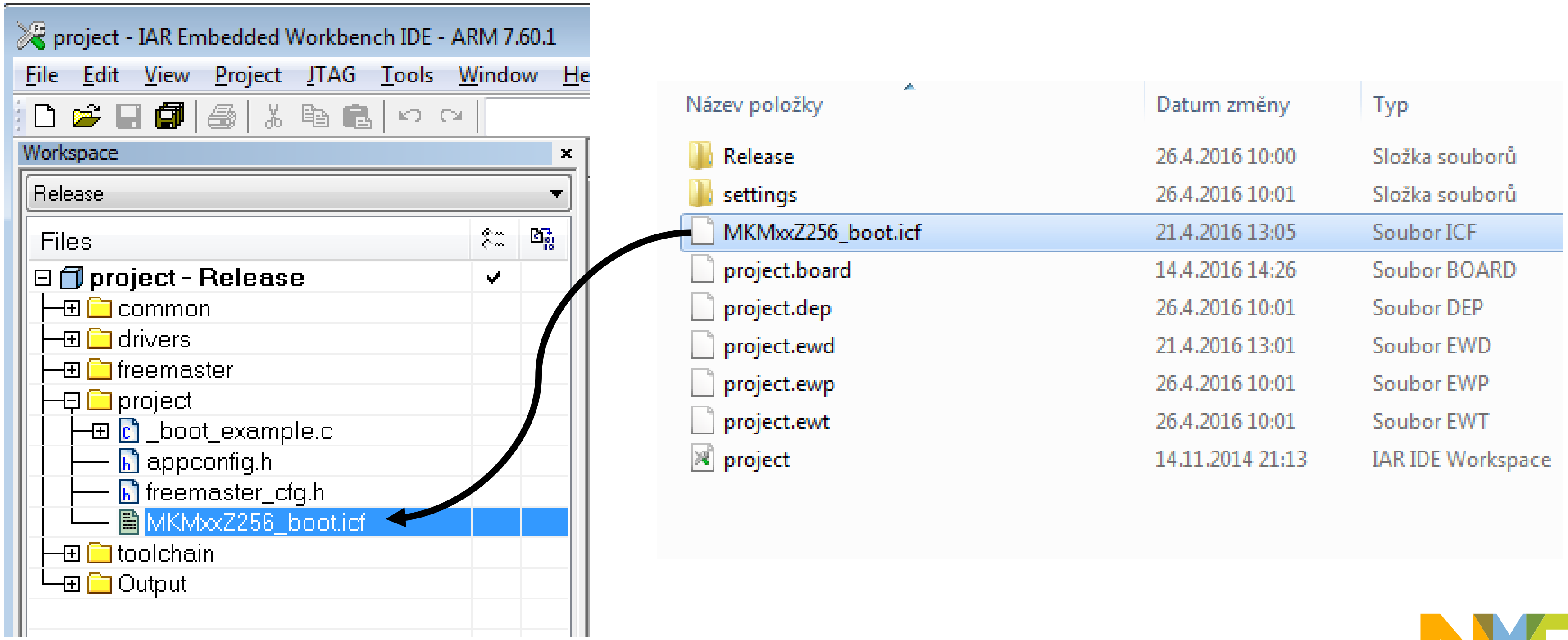


```
#include "drivers.h"

/* LED blinky (PIT) callback declarations
void LED_BlinkyCB (PIT_CALLBACK_TYPE type);


void main (void)
{
  /* enable clocks to all on chip peripherals
  SIM_Init (SIM_MODULE_ALL_PERIPH_ON_CONFIG);

  /* PORT init to GPIO alternative
  PORT_Init (PORTJ, PORT_MODULE_ALT1_MODE, PIN3|PIN4);
  PORT_Init (PORTD, PORT_MODULE_ALT1_MODE, PIN0);

  /* GPIO init for LED
  GPIO_Init(GPIOJ, GPIO_OUT_LOGIC1_MODE, PIN3|PIN4);
  GPIO_Init(GPIOD, GPIO_OUT_LOGIC1_MODE, PIN0);
```

Prepare the user application

Add bootloader linker file ***MKMxxZ256_boot.icf*** into the project structure:

<KM34Z75 bare_ installation path> \KM34Z75_EXAMPLES\build\iar_7_50\projects_boot_example\ MKMxxZ256_boot.icf



Název položky	Datum změny	Typ
Release	26.4.2016 10:00	Složka souborů
settings	26.4.2016 10:01	Složka souborů
MKMxxZ256_boot.icf	21.4.2016 13:05	Soubor ICF
project.board	14.4.2016 14:26	Soubor BOARD
project.dep	26.4.2016 10:01	Soubor DEP
project.ewd	21.4.2016 13:01	Soubor EWD
project.ewp	26.4.2016 10:01	Soubor EWP
project.ewt	26.4.2016 10:01	Soubor EWT
project	14.11.2014 21:13	IAR IDE Workspace

Prepare the user application

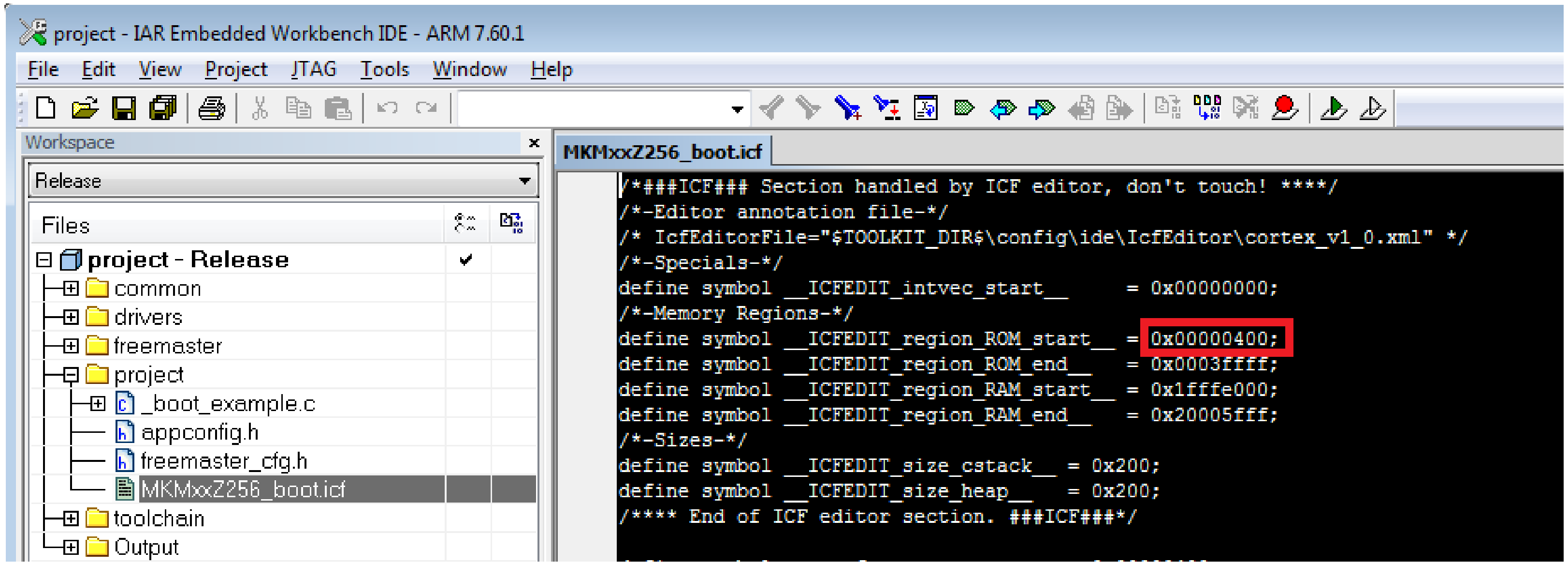
Now we will change flash starting address of the user application in linker file *MKMxxZ256_boot.icf*:

define symbol __ICFEDIT_region_ROM_start__ = 0x00000400;

User application must be moved above the bootloader region:

define symbol __ICFEDIT_region_ROM_start__ = ***0x00002400****;*

Protected vs non-protected version *(0x2400 vs 0x1400)*



```
/*###ICF### Section handled by ICF editor, don't touch! ****/
/*-Editor annotation file-*/
/* IcfEditorFile="$TOOLKIT_DIR$\config\ide\IcfEditor\cortex_v1_0.xml" */
/*-Specials-*/
define symbol __ICFEDIT_intvec_start__     = 0x00000000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x00000400;
define symbol __ICFEDIT_region_ROM_end__   = 0x0003ffff;
define symbol __ICFEDIT_region_RAM_start__ = 0x1fffe000;
define symbol __ICFEDIT_region_RAM_end__   = 0x20005fff;
/*-Sizes-*/
define symbol __ICFEDIT_size_cstack__ = 0x200;
define symbol __ICFEDIT_size_heap__   = 0x200;
/**** End of ICF editor section. ###ICF###*/
```

Prepare the user application

Protected vs non-protected version *(0x2400 vs 0x1400)*

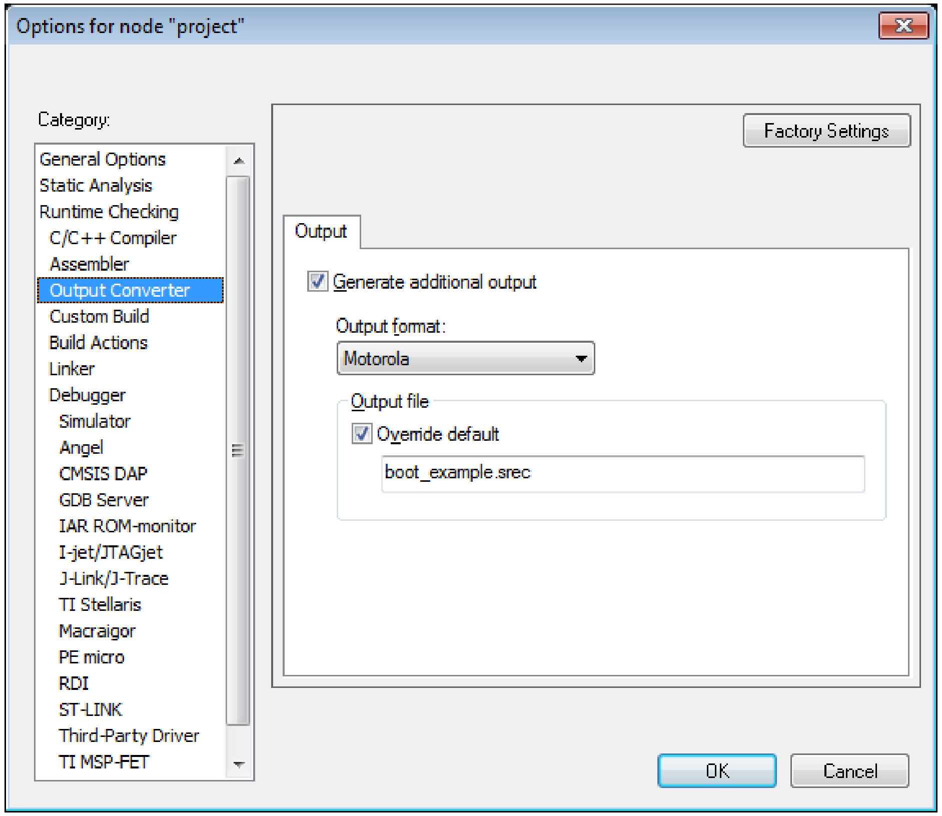
```
/** Bootloader flash protection */
#define BOOTLOADER_FLASH_PROTECTION          0

/** Bootloader flash protection */
#define BOOTLOADER_FLASH_PROTECTION          1
```

Actual size of the bootloader is ~2,3KB (0x900). This region cannot be in collision with user application.

Minimal protection are for the KM34Z75 is (256KB / 32) = 0x2000.

Prepare the user application

1. Go to the “Output Converter” category and set the settings according to the following picture:
   a) Change the Output format to Motorola
   b) Flag the “Override default” box



☑ Generate additional output
Output format: Motorola
Output file
☑ Override default
boot_example.srec

Prepare the user application

2. Click OK button and rebuild the boot_example project.
   a) Please remember to set the project as active.
   b) Generated “boot_example.srec” file will be in location:

<KM34Z75 bare_ installation path> \KM34Z75_EXAMPLES\build\iar_7_50\projects_boot_example\Release\Exe\boot_example.srec

Bootloader configuration

1. Plug in the TWR-KM34Z75M via the included USB cable:
   - It will begin to install some drivers, this is normal and should be allowed to complete before downloading the application.

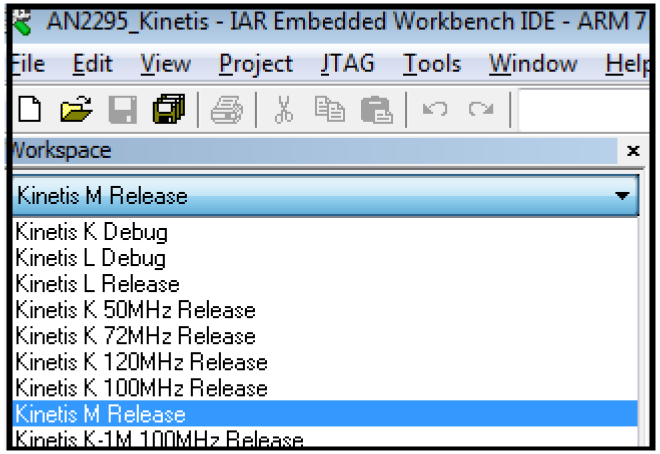
2. Go the location where you installed the AN2295 universal bootloader software, and open the IAR Workspace found at:

   <AN2295 software installation path>\an2295sw\src\Kinetis\IAR\AN2295_Kinetis.eww

3. Now we will change some configuration options of the bootloader application specific for the TWR-KM34Z75 board.

Bootloader configuration

- Set target to *Kinetis M Release*



AN2295_Kinetis - IAR Embedded Workbench IDE - ARM 7

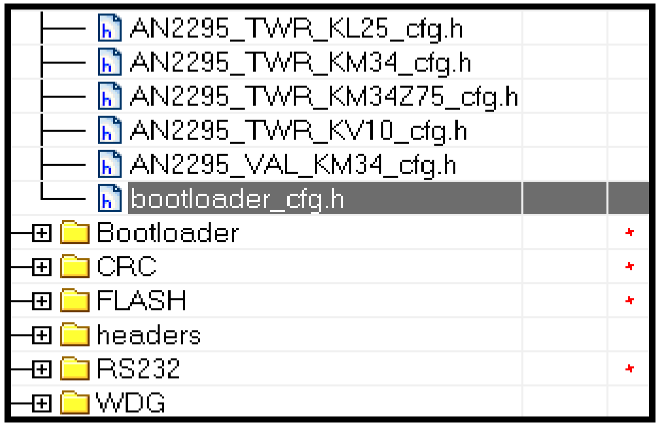
File Edit View Project JTAG Tools Window Help

Workspace

Kinetis M Release

Kinetis K Debug
Kinetis L Debug
Kinetis L Release
Kinetis K 50MHz Release
Kinetis K 72MHz Release
Kinetis K 120MHz Release
Kinetis K 100MHz Release
Kinetis M Release
Kinetis K-1M 100MHz Release

- Check file *bootloader_cfg.h*, if correct header file is included:

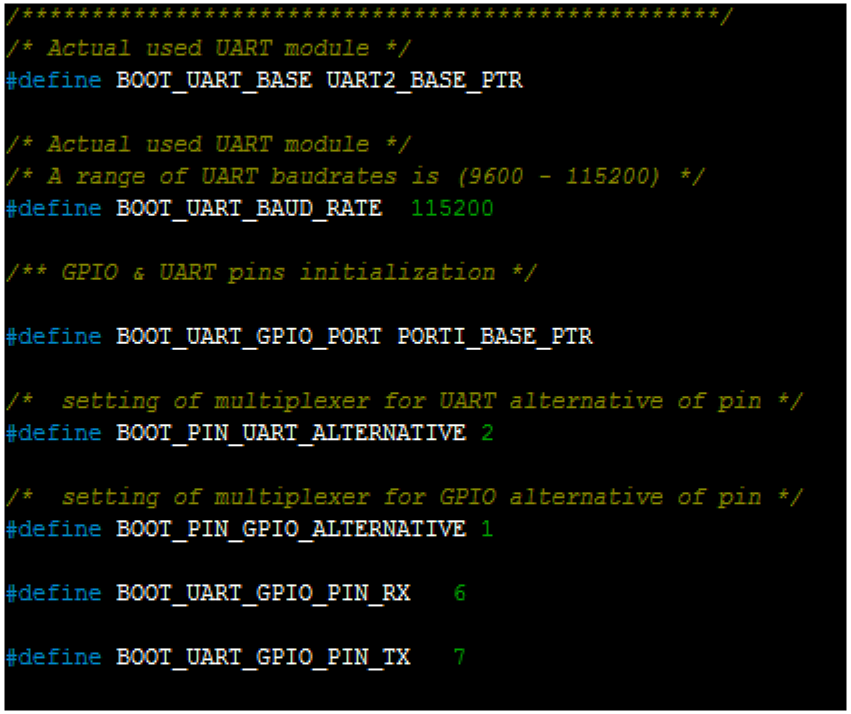


AN2295_TWR_KL25_cfg.h
AN2295_TWR_KM34_cfg.h
AN2295_TWR_KM34Z75_cfg.h
AN2295_TWR_KV10_cfg.h
AN2295_VAL_KM34_cfg.h
bootloader_cfg.h
Bootloader
CRC
FLASH
headers
RS232
WDG

```
#elif defined(KINETIS_M)
  // #include "AN2295_VAL_KM34_cfg.h"
  // #include "AN2295_TWR_KM34_cfg.h"
  #include "AN2295_TWR_KM34Z75_cfg.h"
```

Bootloader configuration

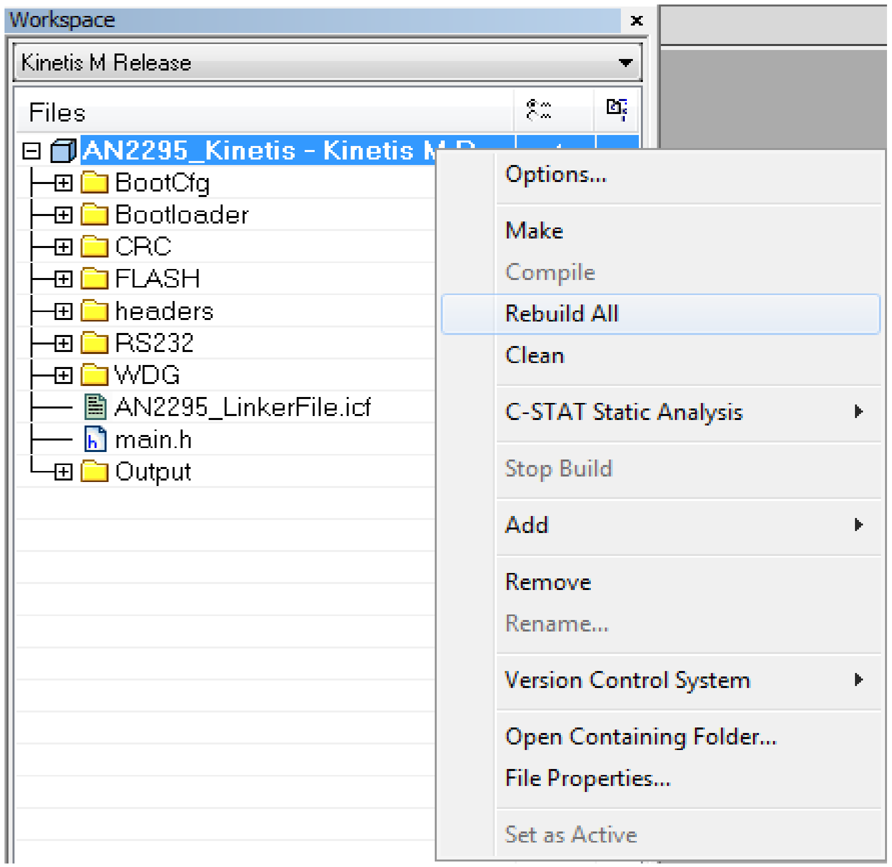
- Actual settings corresponds to UART via open SDA interface
- We can modify the config file *AN2295_TWR_KM34Z75_cfg.h:*
  - *Important parameters are:*



```
/**************************************************/
/* Actual used UART module */
#define BOOT_UART_BASE UART2_BASE_PTR

/* Actual used UART module */
/* A range of UART baudrates is (9600 - 115200) */
#define BOOT_UART_BAUD_RATE  115200

/** GPIO & UART pins initialization */

#define BOOT_UART_GPIO_PORT PORTI_BASE_PTR

/*  setting of multiplexer for UART alternative of pin */
#define BOOT_PIN_UART_ALTERNATIVE 2

/*  setting of multiplexer for GPIO alternative of pin */
#define BOOT_PIN_GPIO_ALTERNATIVE 1

#define BOOT_UART_GPIO_PIN_RX   6

#define BOOT_UART_GPIO_PIN_TX   7
```

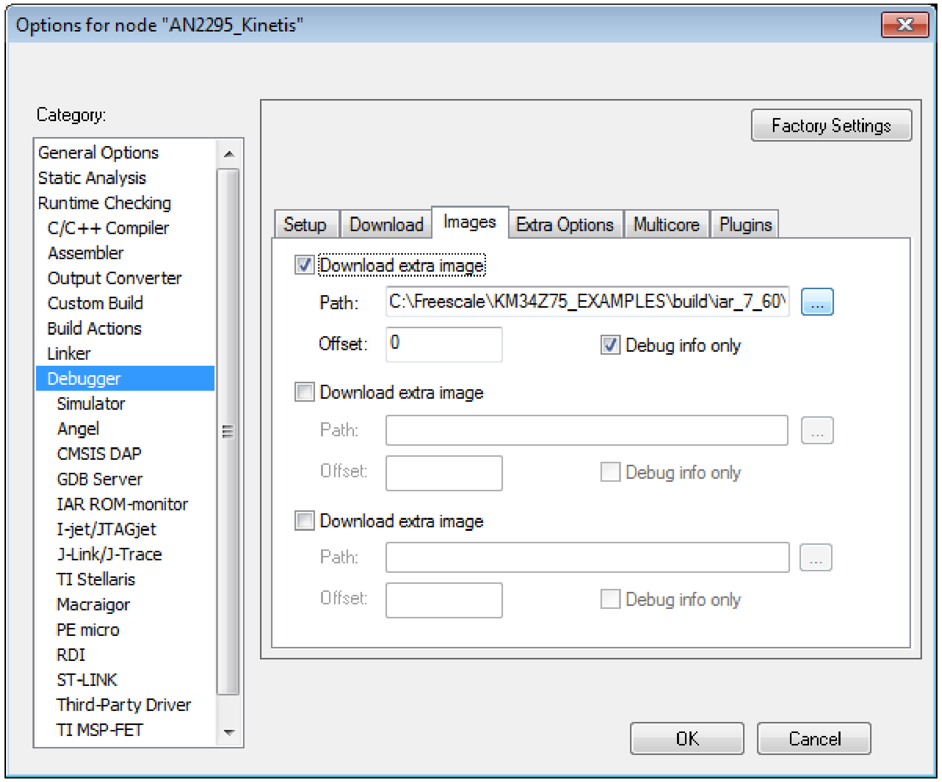
Bootloader configuration

- Rebuilt the complete bootloader application and program:



Bootloader configuration

- Now we can extra image of the user application (only debug info):



Options for node "AN2295_Kinetis"

Category:

[✓] Download extra image
Path: C:\Freescale\KM34Z75_EXAMPLES\build\iar_7_60\
Offset: 0
[✓] Debug info only

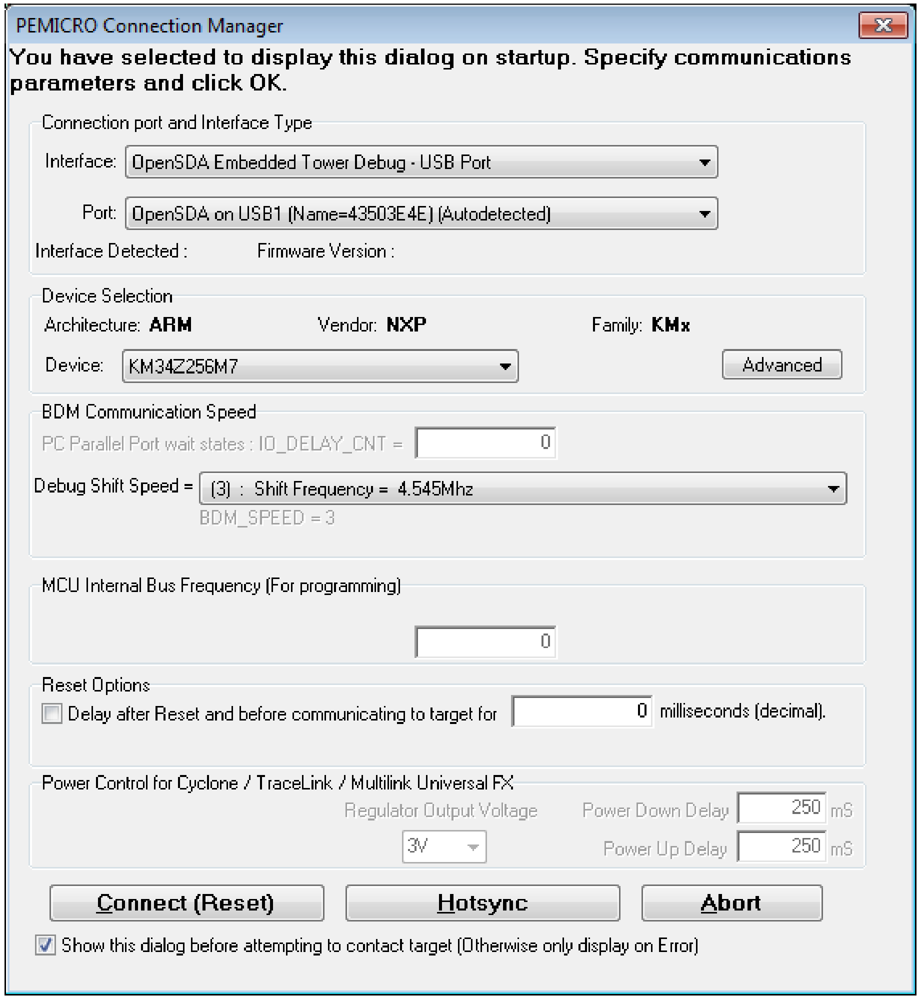
[ ] Download extra image
Path:
Offset:
[ ] Debug info only

[ ] Download extra image
Path:
Offset:
[ ] Debug info only

* This feature allows debugging of both application (bootloader & usr app)in the same time.

Bootloader configuration

- Download bootloader to TWR-KM34 using OpenSDA interface:



PEMICRO Connection Manager

You have selected to display this dialog on startup. Specify communications parameters and click OK.

Connection port and Interface Type

Interface: OpenSDA Embedded Tower Debug - USB Port

Port: OpenSDA on USB1 (Name=43503E4E) (Autodetected)

Interface Detected :

Firmware Version :

Device Selection

Architecture: **ARM**

Vendor: **NXP**

Family: **KMx**

Device: KM34Z256M7

BDM Communication Speed

PC Parallel Port wait states : IO_DELAY_CNT = 0

Debug Shift Speed = (3) : Shift Frequency = 4.545Mhz

BDM_SPEED = 3

MCU Internal Bus Frequency (For programming)

0

Reset Options

☐ Delay after Reset and before communicating to target for 0 milliseconds (decimal).

Power Control for Cyclone / TraceLink / Multilink Universal FX

Regulator Output Voltage 3V

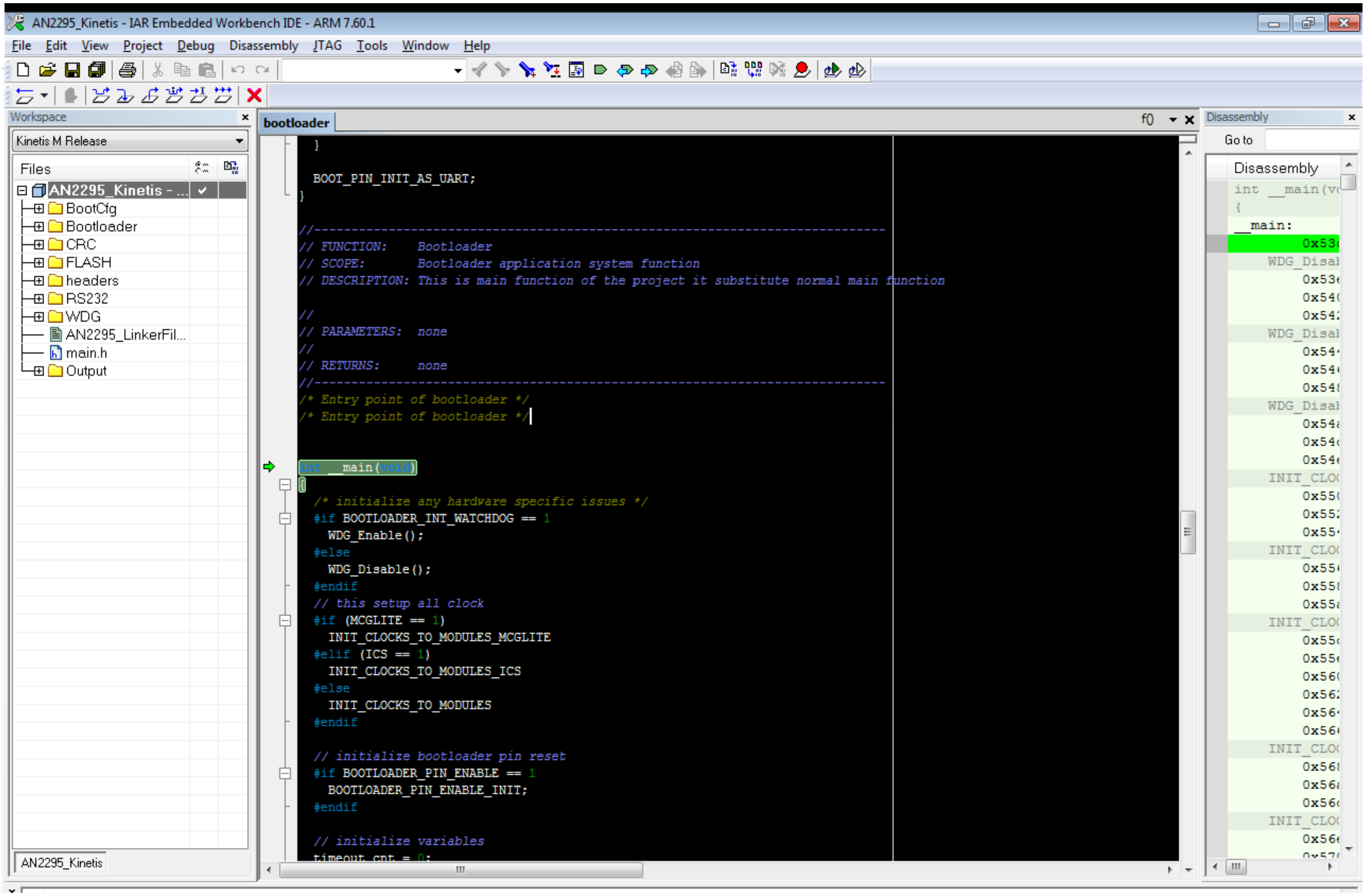
Power Down Delay 250 mS

Power Up Delay 250 mS

☑ Show this dialog before attempting to contact target (Otherwise only display on Error)

Bootloader configuration

Now everything is prepared for boot-loading procedure. ☺

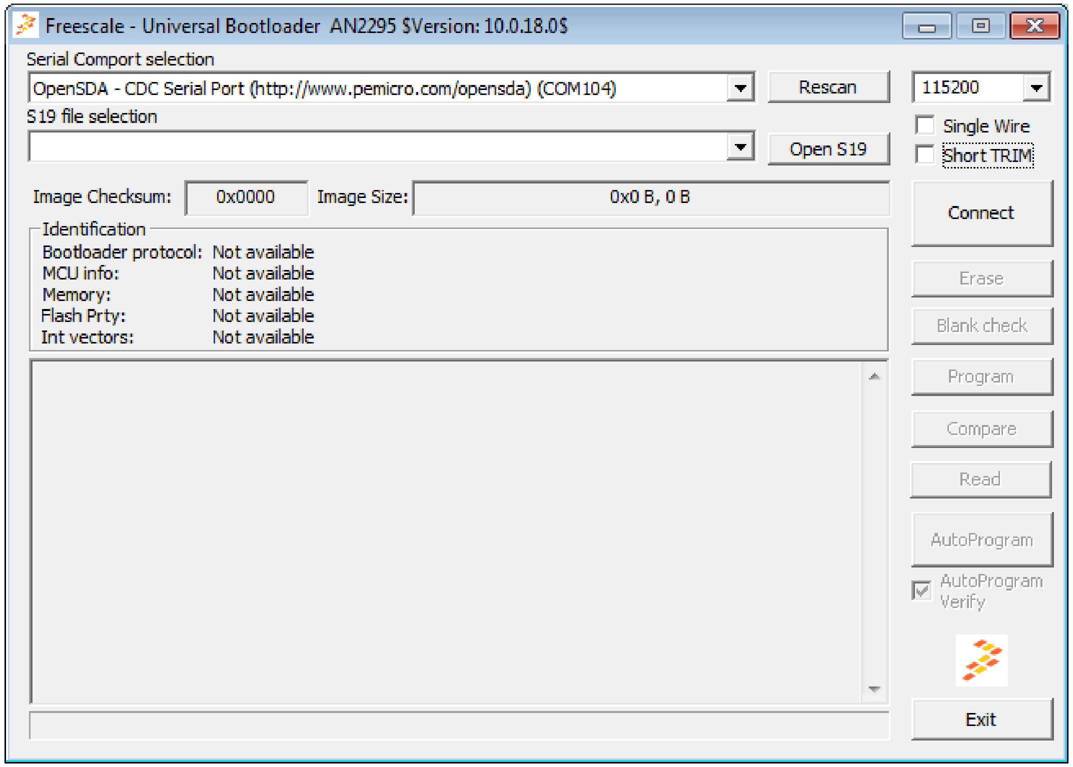


AN2295_Kinetis - IAR Embedded Workbench IDE - ARM 7.60.1

```
  }

  BOOT_PIN_INIT_AS_UART;
}

//-----------------------------------------------------------------------------
// FUNCTION:    Bootloader
// SCOPE:       Bootloader application system function
// DESCRIPTION: This is main function of the project it substitute normal main function

//
// PARAMETERS:  none
//
// RETURNS:     none
//-----------------------------------------------------------------------------
/* Entry point of bootloader */
/* Entry point of bootloader */


int __main(void)
{
  /* initialize any hardware specific issues */
  #if BOOTLOADER_INT_WATCHDOG == 1
    WDG_Enable();
  #else
    WDG_Disable();
  #endif
  // this setup all clock
  #if (MCGLITE == 1)
    INIT_CLOCKS_TO_MODULES_MCGLITE
  #elif (ICS == 1)
    INIT_CLOCKS_TO_MODULES_ICS
  #else
    INIT_CLOCKS_TO_MODULES
  #endif

  // initialize bootloader pin reset
  #if BOOTLOADER_PIN_ENABLE == 1
    BOOTLOADER_PIN_ENABLE_INIT;
  #endif

  // initialize variables
  timeout_cnt = 0;
```

Boot-loading procedure with master PC application

1. Go the location where you installed the AN2295 universal bootloader software and open the master PC application located here: ***<AN2295 software installation path>\an2295sw\masters\release\win_hc08sprg.exe***


Freescale - Universal Bootloader AN2295 $Version: 10.0.18.0$

Serial Comport selection: OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM104)

S19 file selection:

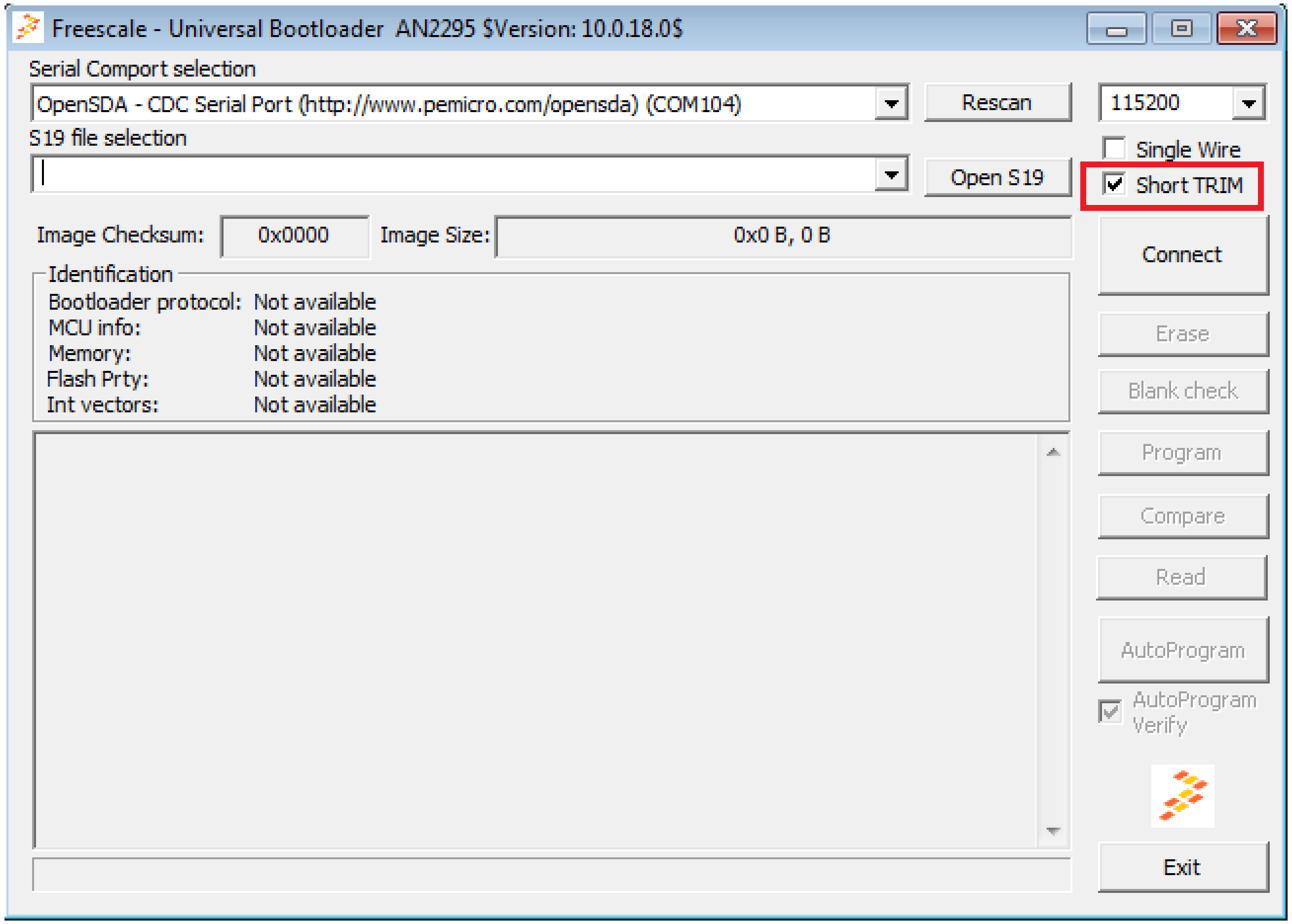
Image Checksum: 0x0000 Image Size: 0x0 B, 0 B

Identification

Bootloader protocol:	Not available
MCU info:	Not available
Memory:	Not available
Flash Prty:	Not available
Int vectors:	Not available

Boot-loading procedure with master PC application

1. Check the Short TRIM checkbox due to using UART via USB.



Freescale - Universal Bootloader AN2295 $Version: 10.0.18.0$

Serial Comport selection: OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM104)

S19 file selection:

115200

[ ] Single Wire

[✓] Short TRIM

Image Checksum: 0x0000 Image Size: 0x0 B, 0 B

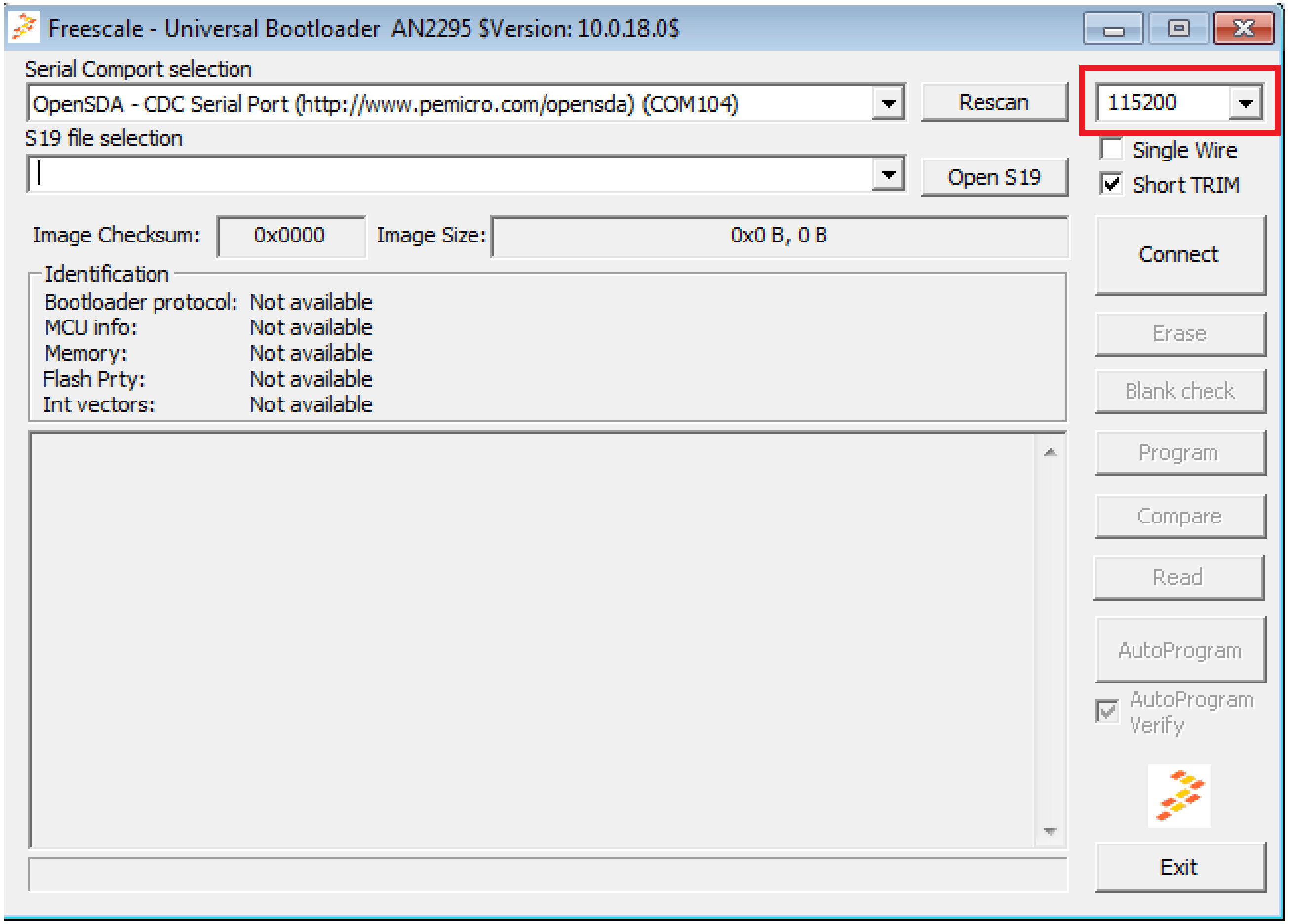
Identification

Bootloader protocol:	Not available
MCU info:	Not available
Memory:	Not available
Flash Prty:	Not available
Int vectors:	Not available

[✓] AutoProgram Verify

Boot-loading procedure with master PC application

1. Set the correct baud rate of UART (in example 115200 Baud).



Freescale - Universal Bootloader AN2295 $Version: 10.0.18.0$

Serial Comport selection: OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM104)

115200

S19 file selection:

- [ ] Single Wire
- [✓] Short TRIM

Image Checksum: 0x0000

Image Size: 0x0 B, 0 B

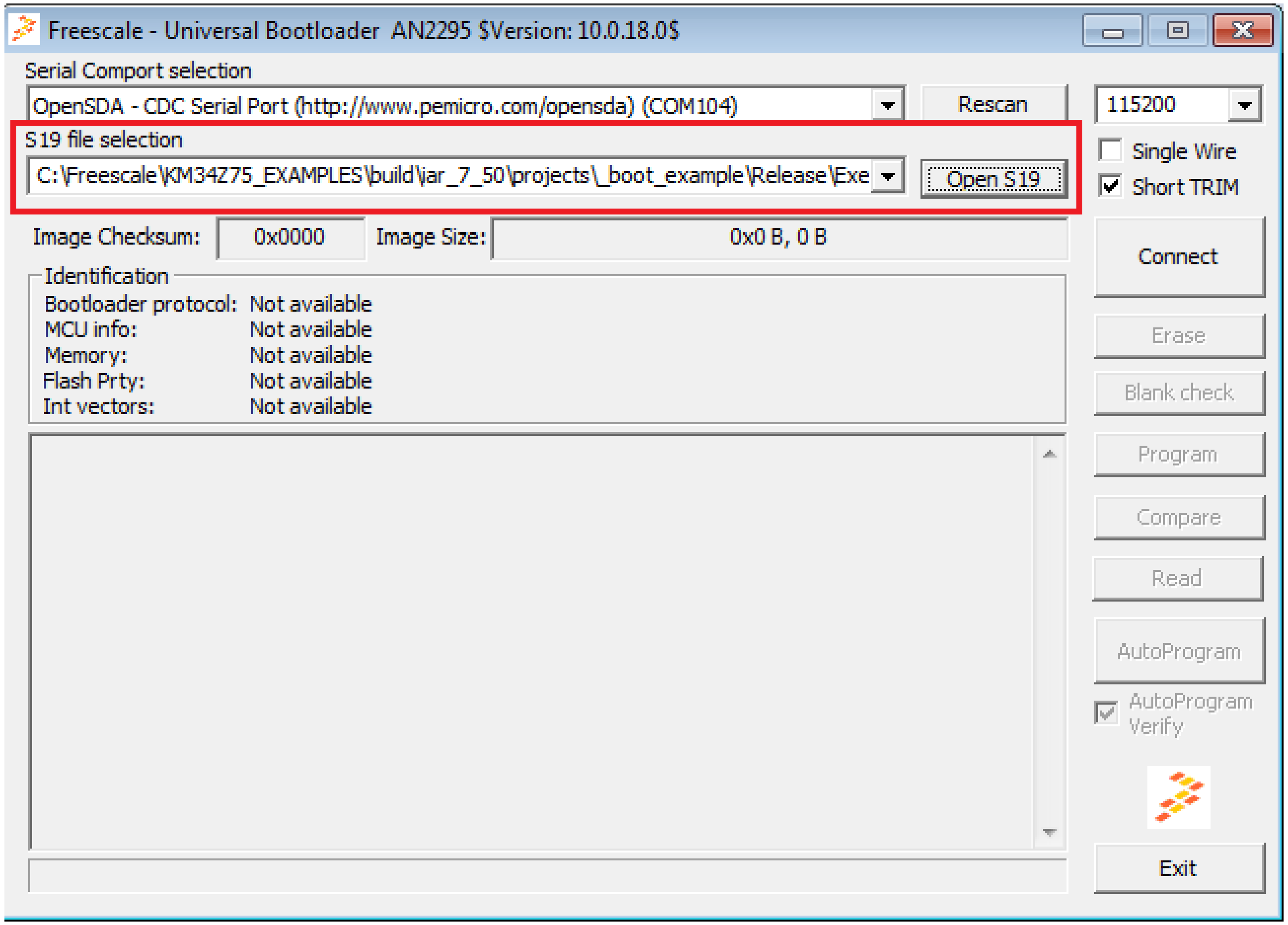
Identification

Bootloader protocol:	Not available
MCU info:	Not available
Memory:	Not available
Flash Prty:	Not available
Int vectors:	Not available

- [✓] AutoProgram Verify

Boot-loading procedure with master PC application

1. Open generated S19 file of user example application.



Freescale - Universal Bootloader AN2295 $Version: 10.0.18.0$

Serial Comport selection

OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM104)

115200

S19 file selection

C:\Freescale\KM34Z75_EXAMPLES\build\iar_7_50\projects_boot_example\Release\Exe

☐ Single Wire

☑ Short TRIM

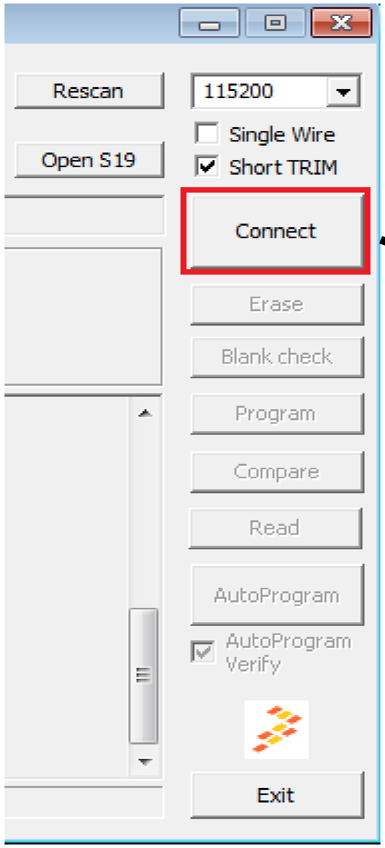
Image Checksum: 0x0000 Image Size: 0x0 B, 0 B

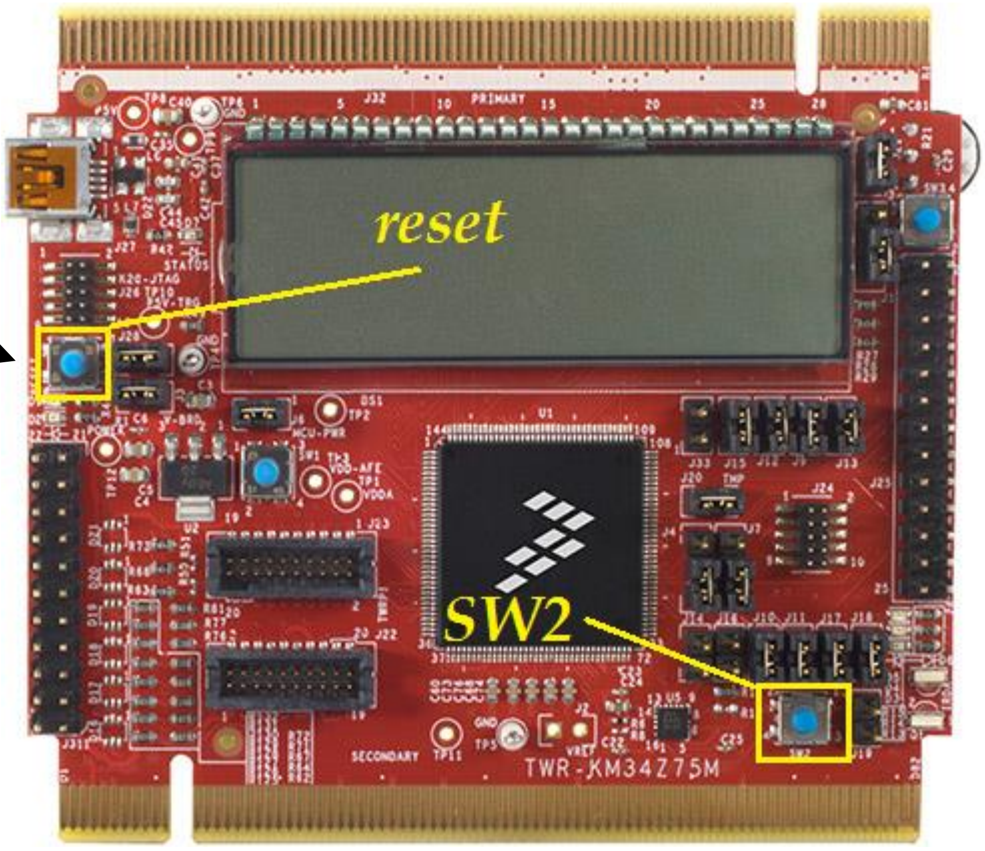
Identification

Bootloader protocol:	Not available
MCU info:	Not available
Memory:	Not available
Flash Prty:	Not available
Int vectors:	Not available

Boot-loading procedure with master PC application

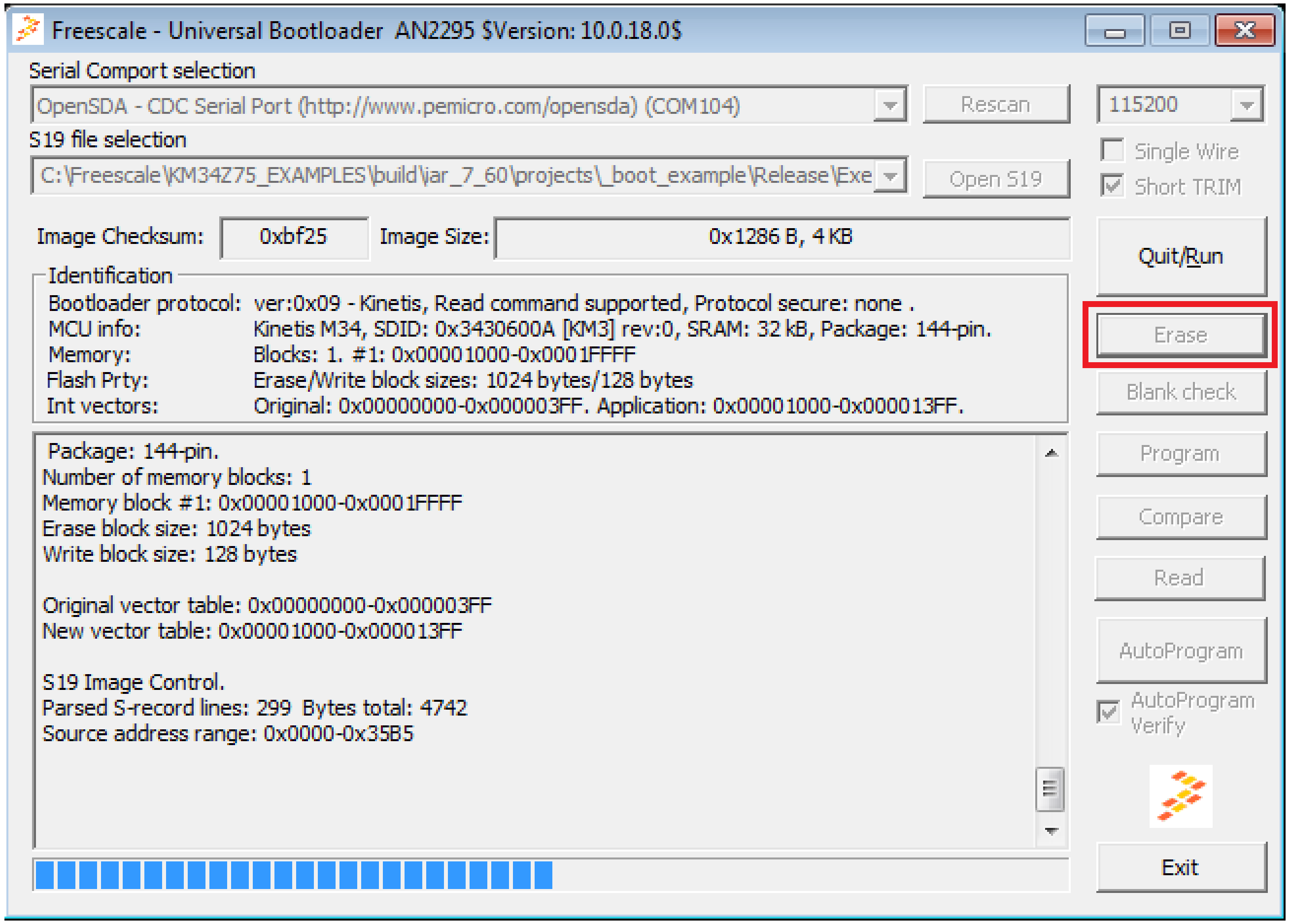
1. Plug TWR-KM34Z75M board to PC.
2. Try to connect Master application: Master app->Button Connect-> Board->reset button->switch SW2->go to bootloader





Boot-loading procedure with master PC application

1. Now you are connected with the bootloader on TWR-KM34 board. First must be erased the application part of flash memory:



Serial Comport selection: OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM104)

115200

S19 file selection: C:\Freescale\KM34Z75_EXAMPLES\build\iar_7_60\projects_boot_example\Release\Exe

Image Checksum: 0xbf25 Image Size: 0x1286 B, 4 KB

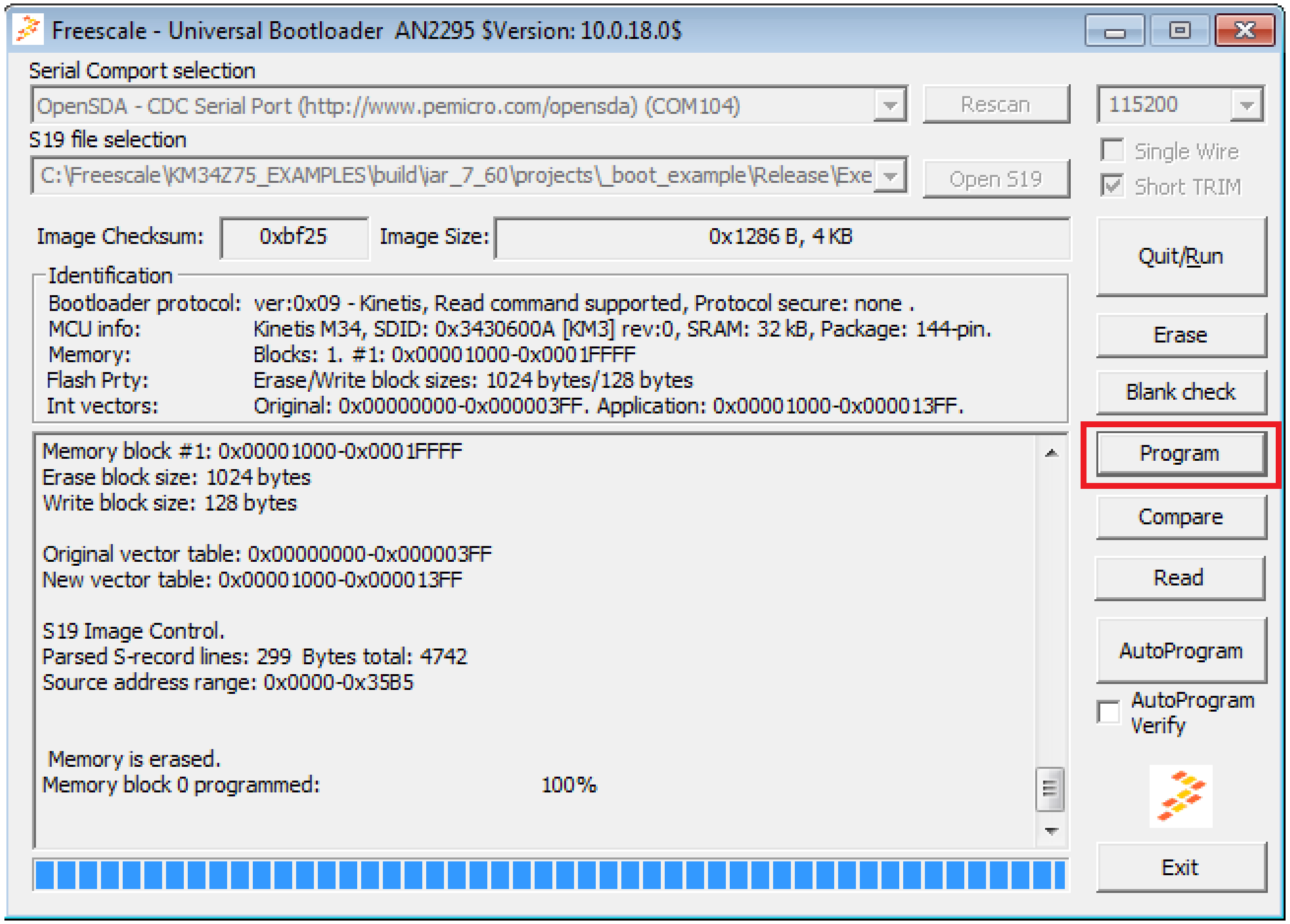
Identification

Bootloader protocol:	ver:0x09 - Kinetis, Read command supported, Protocol secure: none .
MCU info:	Kinetis M34, SDID: 0x3430600A [KM3] rev:0, SRAM: 32 kB, Package: 144-pin.
Memory:	Blocks: 1. #1: 0x00001000-0x0001FFFF
Flash Prty:	Erase/Write block sizes: 1024 bytes/128 bytes
Int vectors:	Original: 0x00000000-0x000003FF. Application: 0x00001000-0x000013FF.

```
Package: 144-pin.
Number of memory blocks: 1
Memory block #1: 0x00001000-0x0001FFFF
Erase block size: 1024 bytes
Write block size: 128 bytes

Original vector table: 0x00000000-0x000003FF
New vector table: 0x00001000-0x000013FF

S19 Image Control.
Parsed S-record lines: 299  Bytes total: 4742
Source address range: 0x0000-0x35B5
```

Boot-loading procedure with master PC application

1. After the correct flash memory erasing we can continue with self-programming by using button **Program**:



Freescale - Universal Bootloader AN2295 $Version: 10.0.18.0$

Serial Comport selection: OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM104)

115200

S19 file selection: C:\Freescale\KM34Z75_EXAMPLES\build\iar_7_60\projects_boot_example\Release\Exe

Single Wire ☐
Short TRIM ☑

Image Checksum: 0xbf25 Image Size: 0x1286 B, 4 KB

Identification

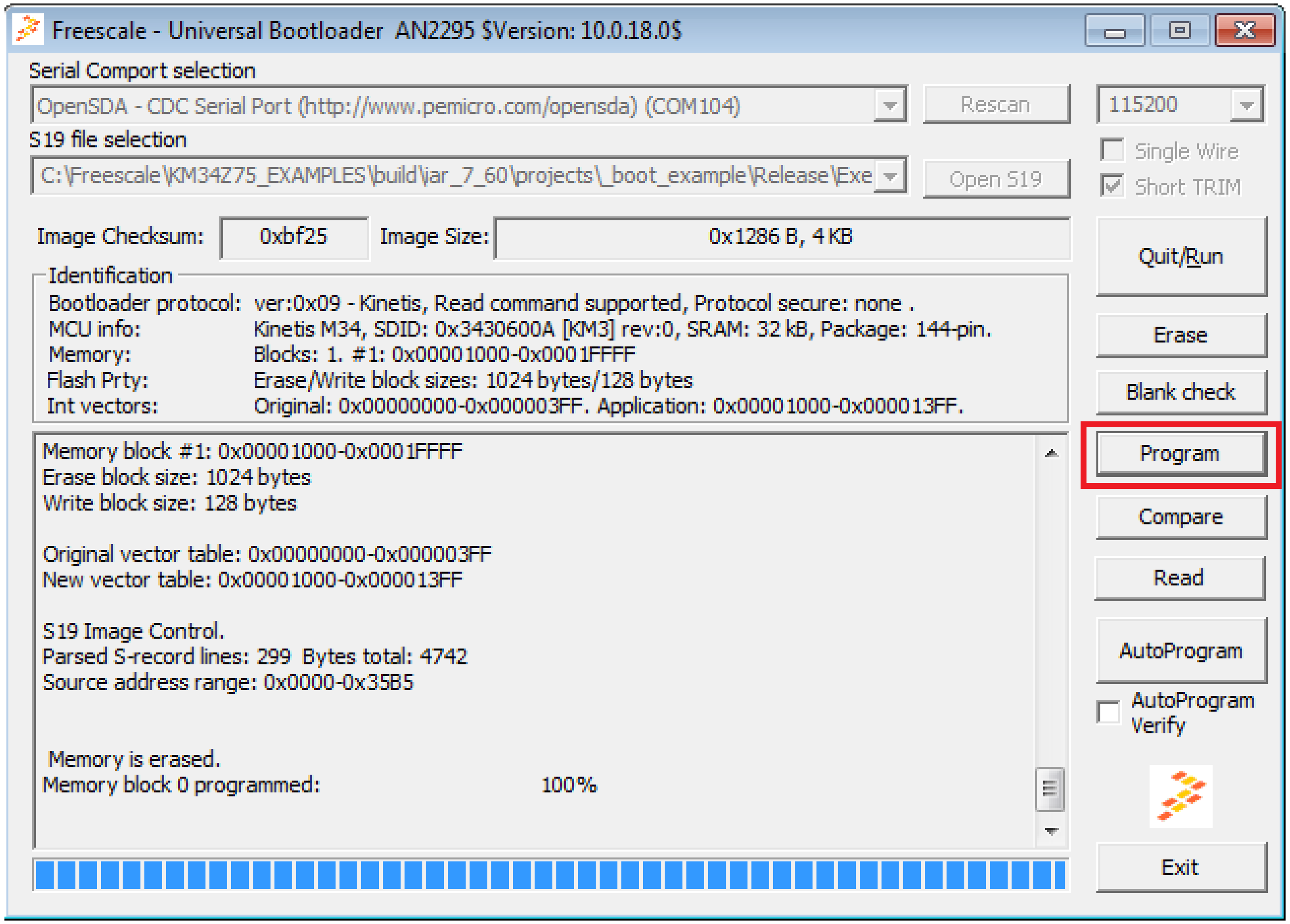
```
Bootloader protocol:  ver:0x09 - Kinetis, Read command supported, Protocol secure: none .
MCU info:             Kinetis M34, SDID: 0x3430600A [KM3] rev:0, SRAM: 32 kB, Package: 144-pin.
Memory:               Blocks: 1. #1: 0x00001000-0x0001FFFF
Flash Prty:           Erase/Write block sizes: 1024 bytes/128 bytes
Int vectors:          Original: 0x00000000-0x000003FF. Application: 0x00001000-0x000013FF.
```

```
Memory block #1: 0x00001000-0x0001FFFF
Erase block size: 1024 bytes
Write block size: 128 bytes

Original vector table: 0x00000000-0x000003FF
New vector table: 0x00001000-0x000013FF

S19 Image Control.
Parsed S-record lines: 299  Bytes total: 4742
Source address range: 0x0000-0x35B5


 Memory is erased.
Memory block 0 programmed:                    100%
```

AutoProgram Verify ☐

Boot-loading procedure with master PC application

1. Now the application was successfully loaded into the MCU, we can leave Master Bootloader application by using button **Quit/Run**:



OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM104)

115200

C:\Freescale\KM34Z75_EXAMPLES\build\iar_7_60\projects_boot_example\Release\Exe

Image Checksum: 0xbf25 Image Size: 0x1286 B, 4 KB

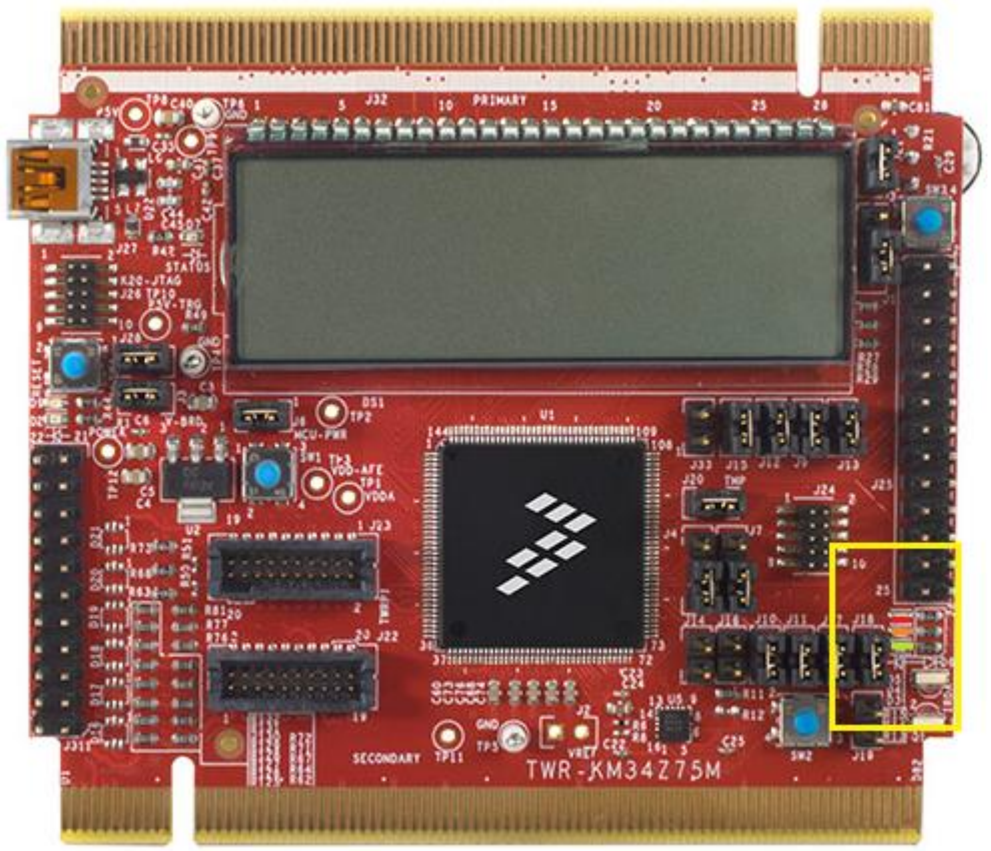
Identification

Bootloader protocol:	ver:0x09 - Kinetis, Read command supported, Protocol secure: none .
MCU info:	Kinetis M34, SDID: 0x3430600A [KM3] rev:0, SRAM: 32 kB, Package: 144-pin.
Memory:	Blocks: 1. #1: 0x00001000-0x0001FFFF
Flash Prty:	Erase/Write block sizes: 1024 bytes/128 bytes
Int vectors:	Original: 0x00000000-0x000003FF. Application: 0x00001000-0x000013FF.

```
Memory block #1: 0x00001000-0x0001FFFF
Erase block size: 1024 bytes
Write block size: 128 bytes

Original vector table: 0x00000000-0x000003FF
New vector table: 0x00001000-0x000013FF

S19 Image Control.
Parsed S-record lines: 299  Bytes total: 4742
Source address range: 0x0000-0x35B5


 Memory is erased.
Memory block 0 programmed:                    100%
```

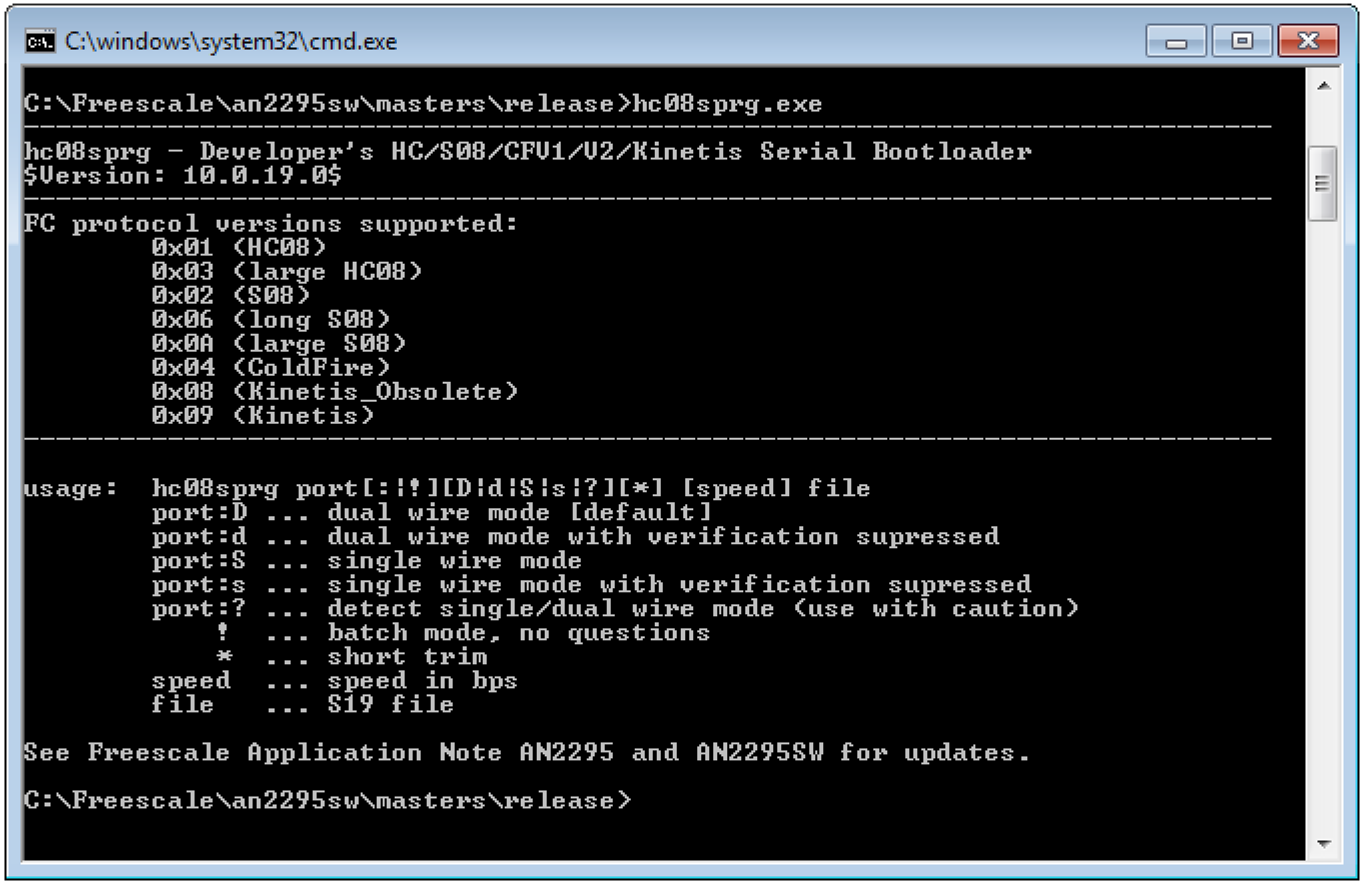
Boot-loading procedure with master PC application

1. If the procedure was done correctly, application is running right now:



Boot-loading procedure with master PC application

1. The same procedure can be achieved by using our command line “hc08sprg.exe” application which is in the sw package.



```
C:\Freescale\an2295sw\masters\release>hc08sprg.exe
--------------------------------------------------------------------------------
hc08sprg - Developer's HC/S08/CFV1/V2/Kinetis Serial Bootloader
$Version: 10.0.19.0$
--------------------------------------------------------------------------------
FC protocol versions supported:
        0x01 (HC08)
        0x03 (large HC08)
        0x02 (S08)
        0x06 (long S08)
        0x0A (large S08)
        0x04 (ColdFire)
        0x08 (Kinetis_Obsolete)
        0x09 (Kinetis)
--------------------------------------------------------------------------------

usage:  hc08sprg port[:|!][D|d|S|s|?][*] [speed] file
        port:D ... dual wire mode [default]
        port:d ... dual wire mode with verification supressed
        port:S ... single wire mode
        port:s ... single wire mode with verification supressed
        port:? ... detect single/dual wire mode (use with caution)
            !  ... batch mode, no questions
            *  ... short trim
        speed  ... speed in bps
        file   ... S19 file

See Freescale Application Note AN2295 and AN2295SW for updates.

C:\Freescale\an2295sw\masters\release>
```

Q&A

1. Document AN2295 application note

THANK YOU FOR YOUR ATTENTION

SECURE CONNECTIONS
FOR A SMARTER WORLD