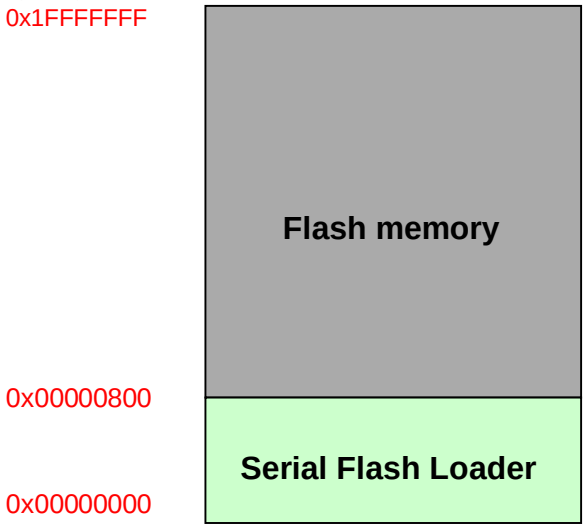
Stellaris Serial flash loader, Boot loader, LM Flash programmer

StellarisWare™

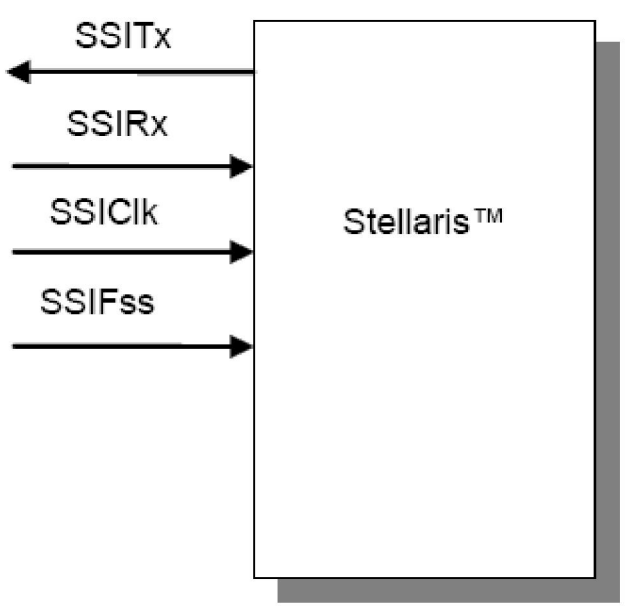
In-System Programming Options

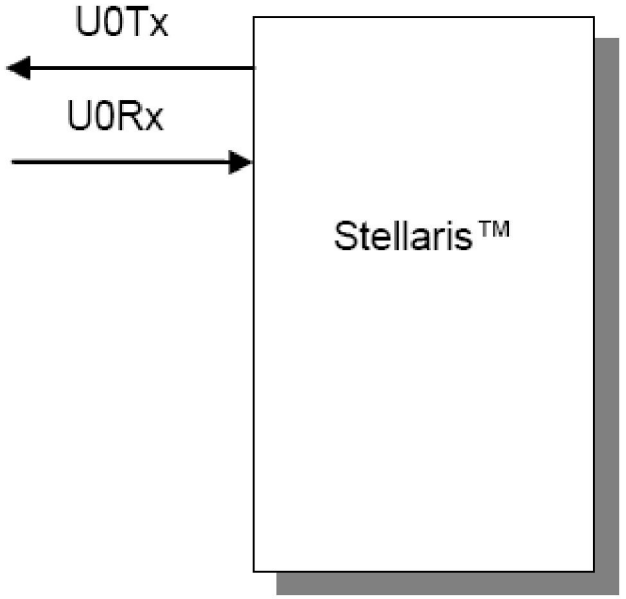
Stellaris Serial Flash Loader

- Small piece of code that allows programming of the flash without the need for a debugger interface.
- All Stellaris MCUs ship with this pre-loaded in flash memory
- Interface options include UART or SSI
- Luminary Micro supplies a Windows™ application (GUI or command line) that makes full use of all commands supported by the serial flash loader (LMflash.exe)
- See application note AN01242



SSI and UART connections





Only UART0 support serial flash loader

Command Definitions

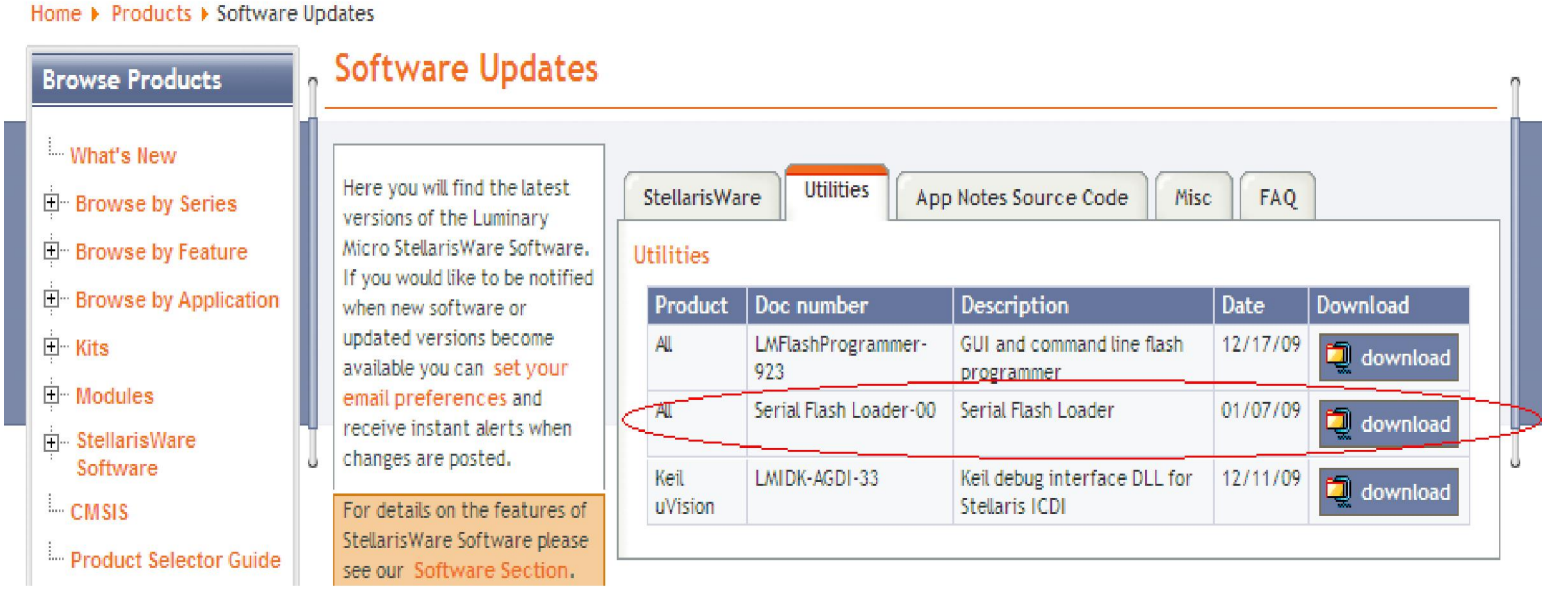
- COMMAND_PING (0x20)
  - Simply accepts the command and sets the global status to success.
- COMMAND_GET_STATUS (0x23)
  - Returns the status of the last command that was issued
- COMMAND_DOWNLOAD (0x21)
  - Sent to the flash loader to indicate where to store data and how many bytes to send with the COMMAND_SEND_DATA commands that follow
- COMMAND_SEND_DATA (0x24)
  - Should only follow a COMMAND_DOWNLOAD command
- COMMAND_RUN (0x22)
  - Tells the flash loader to execute from the address passed as the parameter in this command.
- COMMAND_RESET (0x25)
  - Tells the flash loader device to reset

User application

- The first is to completely overwrite the flash loader with a new application.
- The second method involves programming the application to an address not used by the flash loader(0x800 or later), and then executing this application by issuing a “run” command.

Serial flash loader binary file

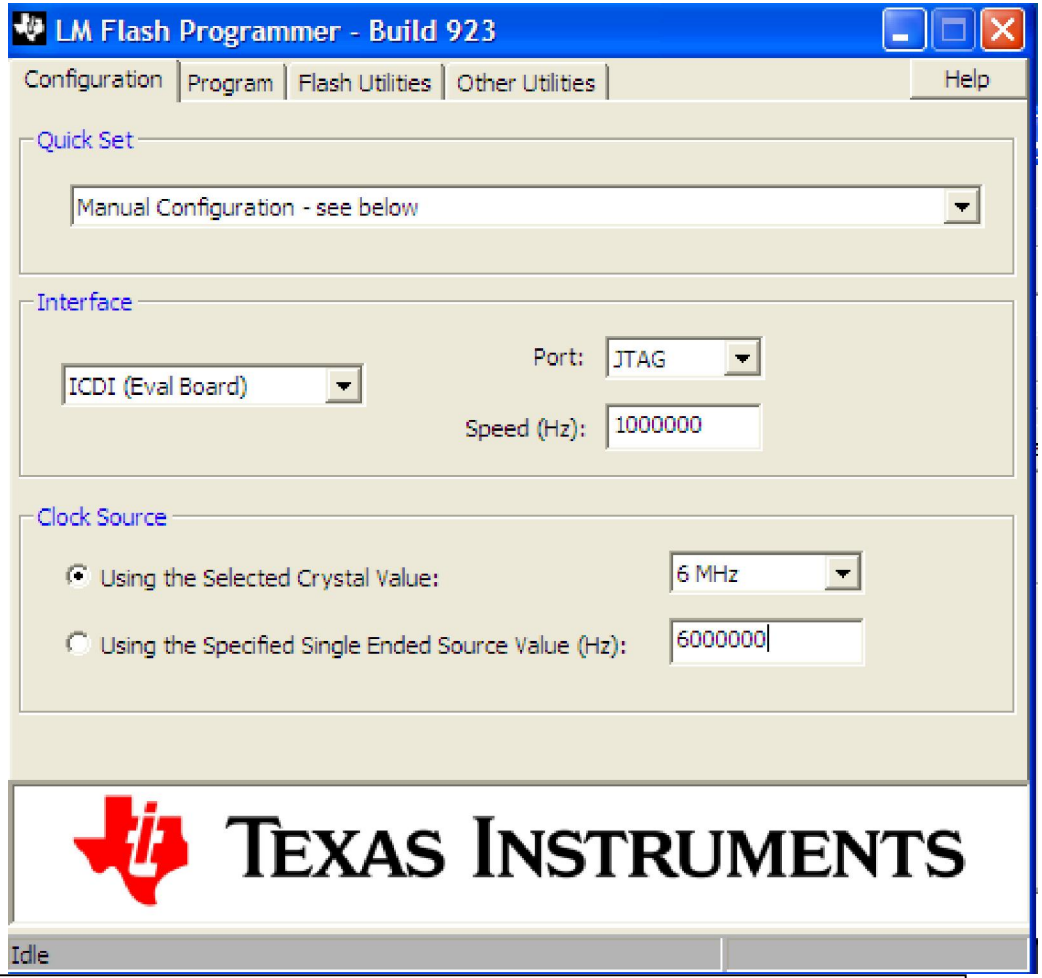
- Download the binary file serial_flash_loader.bin which pre-loaded in flash before factory from web side: https://www.luminarymicro.com/products/software_updates.html



- Re-program this binary file to the device by LM flash programmer to do the testing(use on-chip Serial Flash Loader to download user code)

Stellaris® Flash Programming GUI

- LM Flash Programming GUI
  - Simple graphical user interface
  - Support for all Evaluation Kits
  - Key features include:
    - Program
    - Verify
    - Erase
    - Read memory
  - Available now

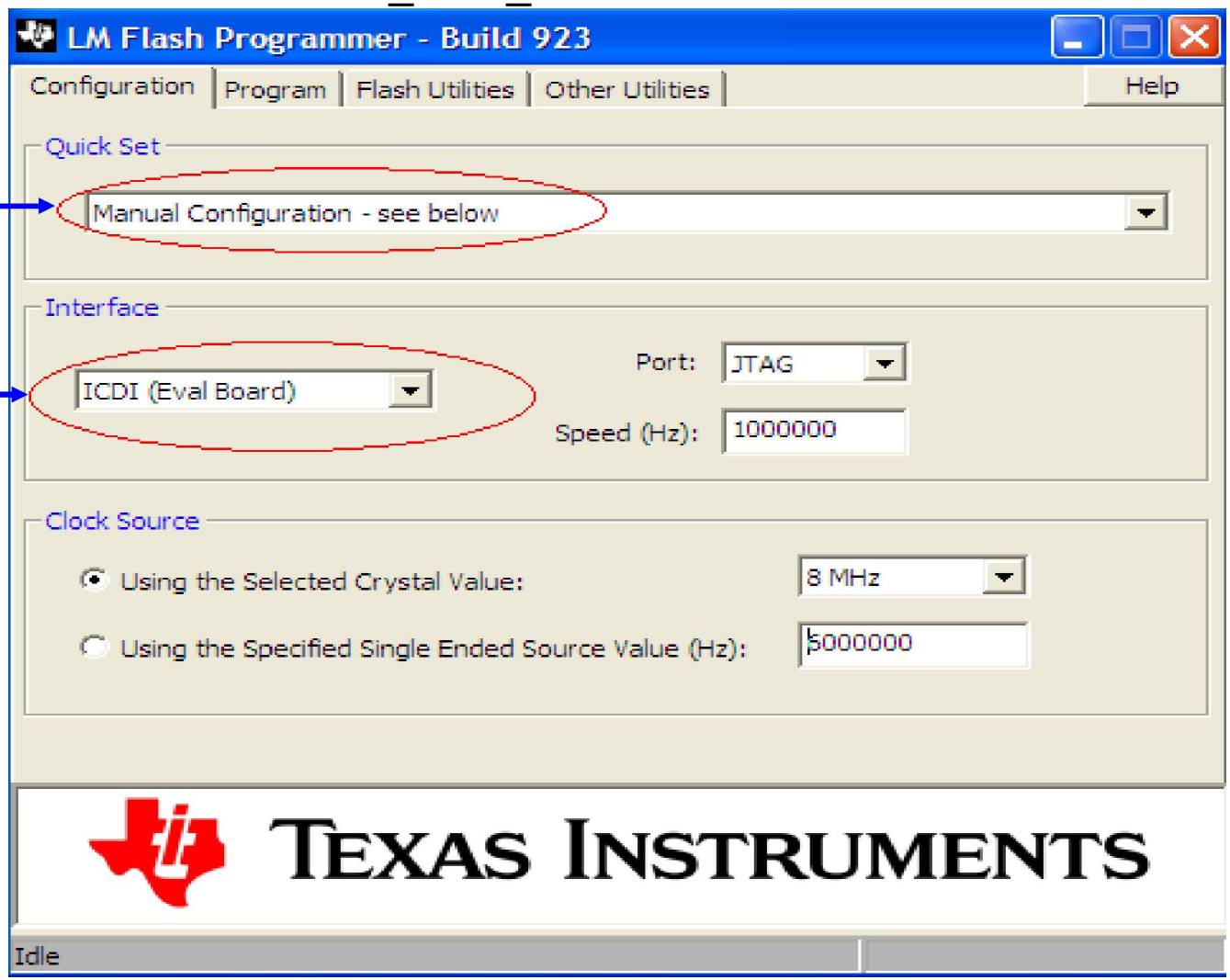


Reprogram device

Reprogram the device with serial_flash_loader.bin,which locate at C:\StellarisWare\Serial Flash Loader\serial_flash_loader.bin

Select Manual Configuration

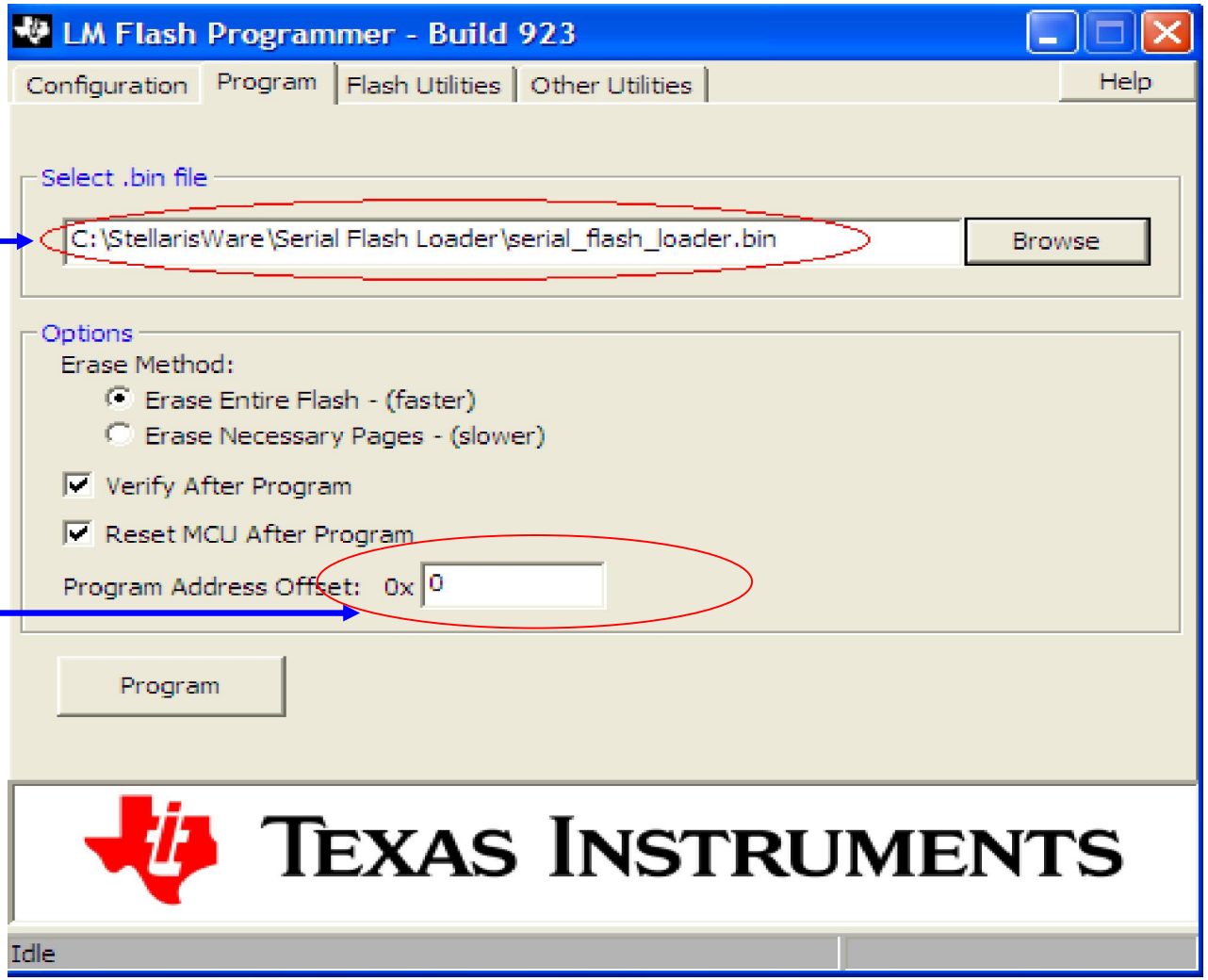
JTAG ICDI



Reprogram device(cnt)

Select the binary file

Start address



Use pre-load serial flash loader to download user code

Download user code to flash via pre-load serial flash loader, just like the device with factory default.

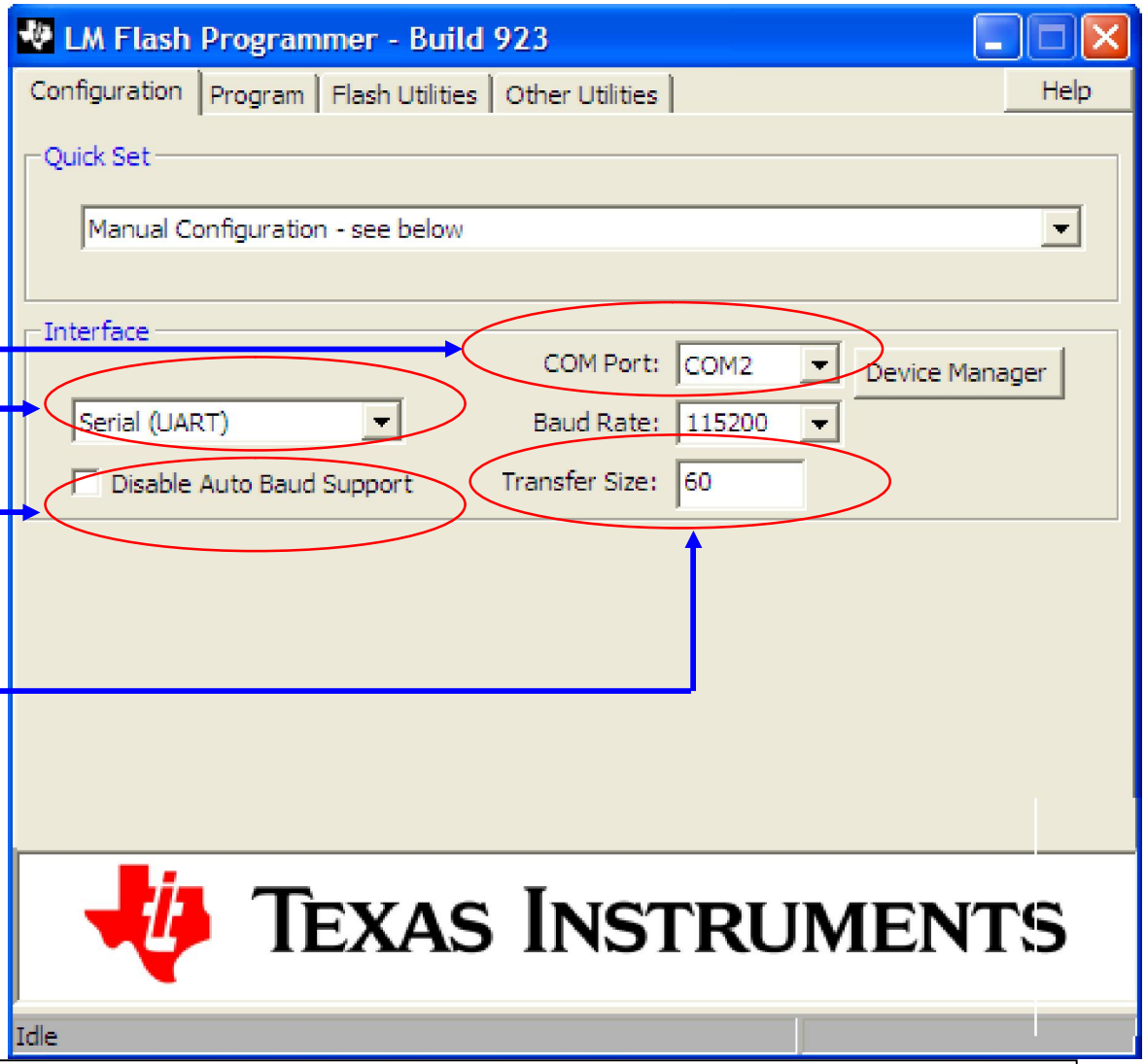
COM2?

Select Serial(UART)

Auto baud rate

Transfer size=60

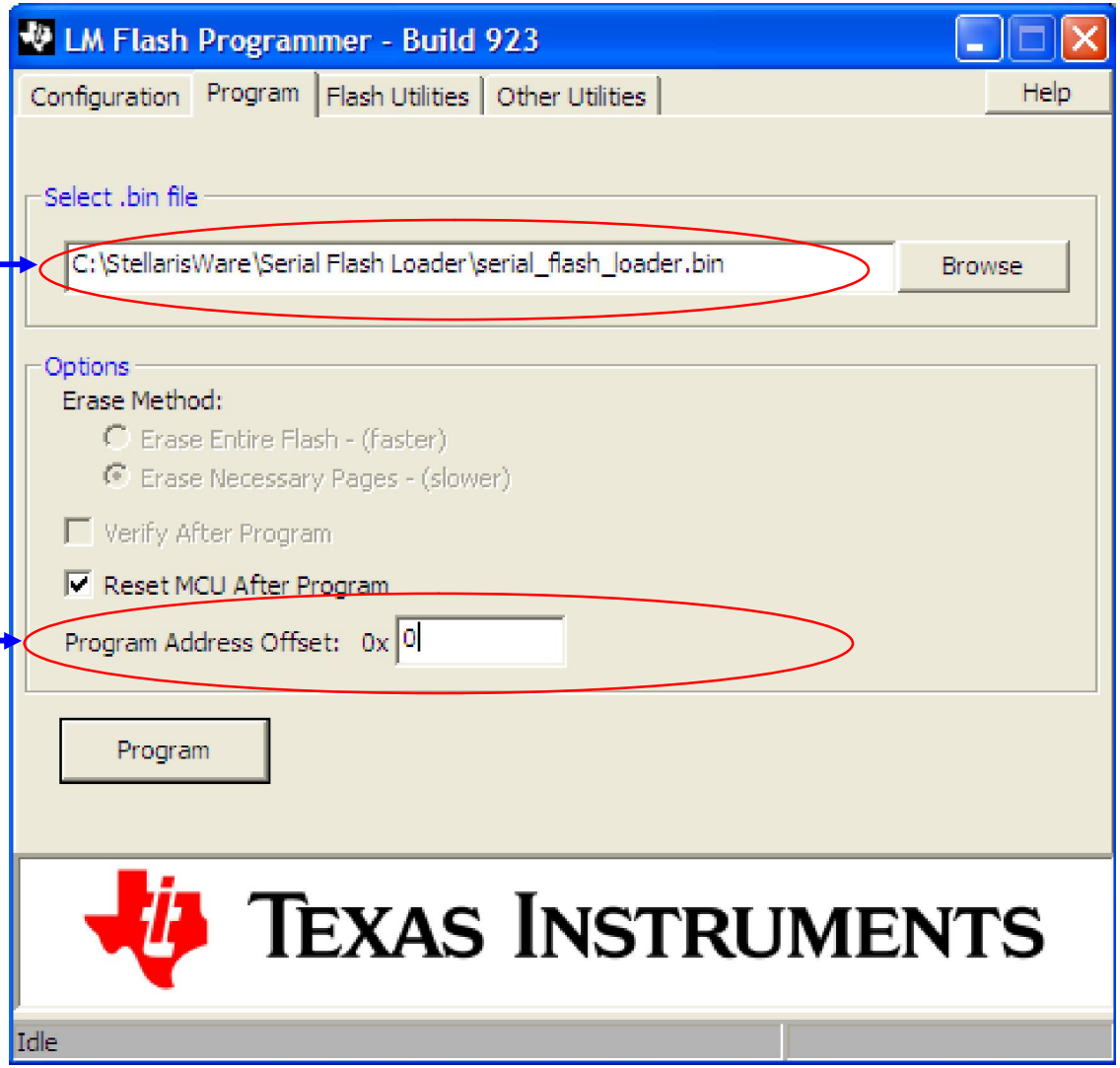
*Use device manager to check your visual COM port



Use pre-load serial flash loader to download(cnt)

Select your user code to download

Address will start from 0x00.



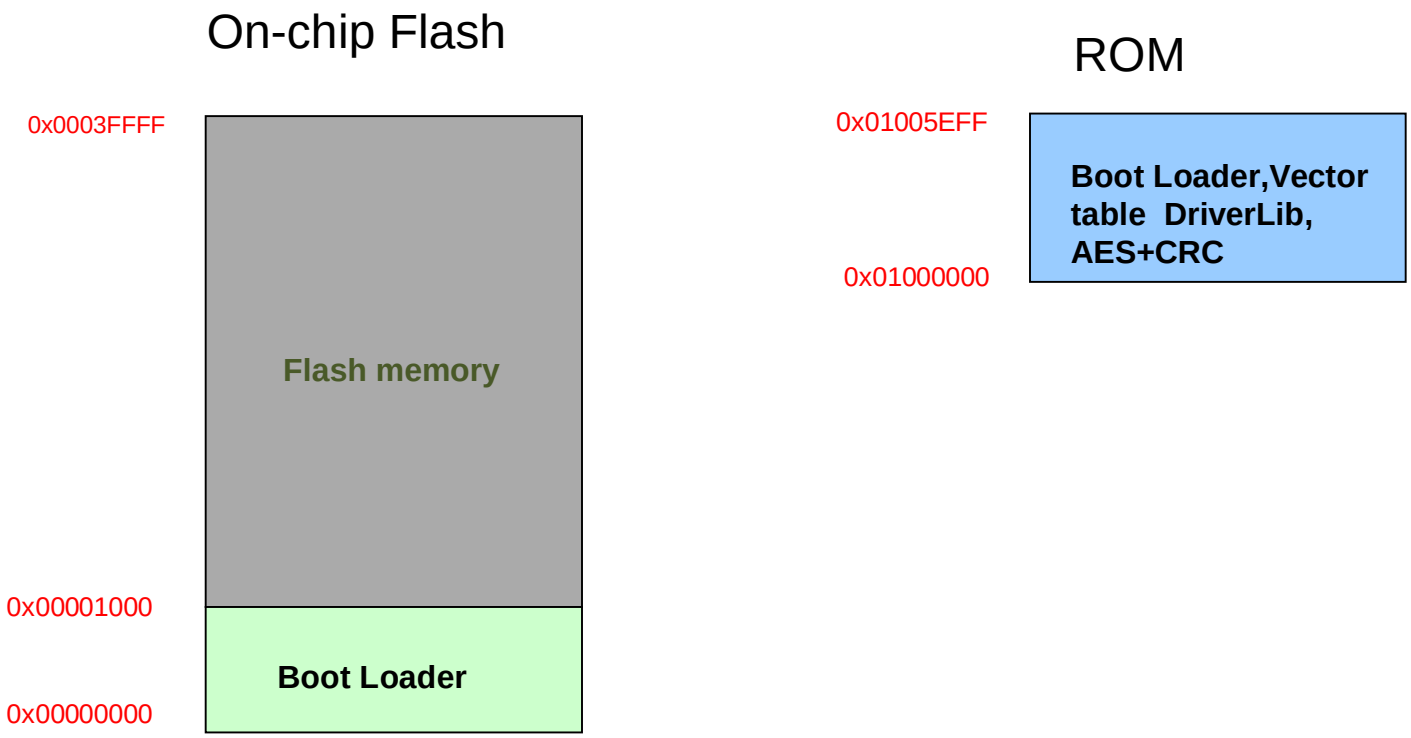
Sflash utility

- The sflash utility that we provide in
  C:\StellarisWare\Serial Flash Loader\sflash.exe
- Usage:
  >sflash test.bin -p 0x00 -r 0x00 -c 1
  -p [program address]
  -r [execution address]
  -c [COM port number]

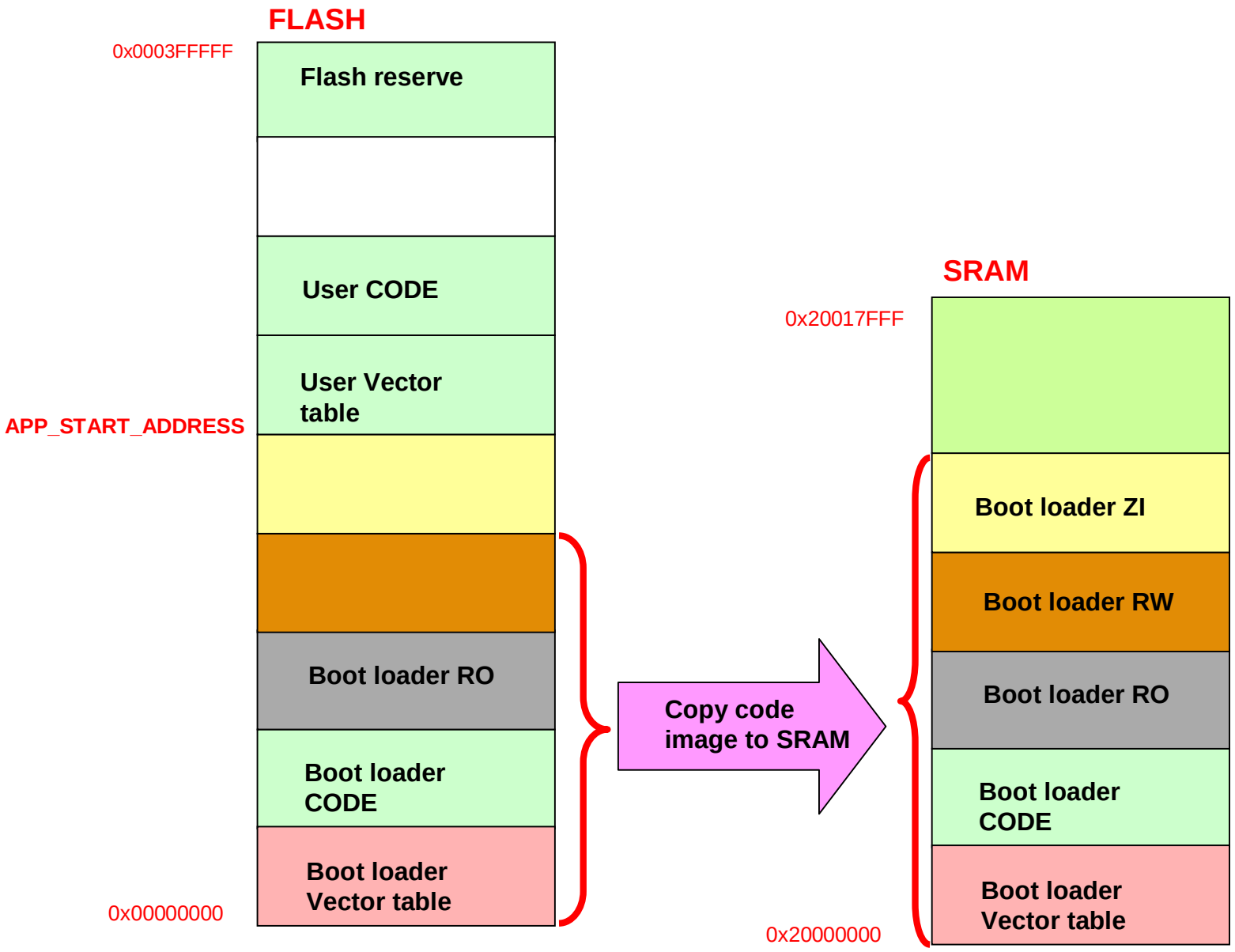
Stellaris Boot Loader

- Small piece of code that can be programmed at the beginning of flash to act as an application loader
- Also used as an update mechanism for an application running on a Stellaris microcontroller.
- Interface options include UART (default), I^2C, SSI, Ethernet, USB
- Included in the Stellaris Peripheral Driver Library with full applications examples
- See application note AN01248
- Preloaded in ROM on select Stellaris

Stellaris Memory map



Boot loader running procedure



User Applications

- On the Stellaris microcontrollers, minimum size of a flash block is 1024-byte,this is a1024-byte boundary,

- The boot loader at least need one flash block, so it requires that the application start at offset 2048 (0x0800) or more into the flash.

Connection interface

- UART0 (default)
- I^2C
- SSI
- Ethernet
- USB(DFU)

Boot loader file

File Name	Description
bl_startup_ewarm.S	Boot Loader start code
bl_config.h	Define Boot Loader option
bl_config.c	
bl_commands.h	Define Updater command
bl_packet.h	Serial data receive/transmit control
bl_packet.c	
bl_enet.c	Ethernet data transmit
bl_uart.h	UART data transmit
bl_uart.c	
bl_autobaud.h	
bl_ssi.h	SSI data transmit
bl_ssi.c	
bl_i2c.h	I2C data transmit
bl_i2c.c	
bl_usb.c	USB data transmit
bl_check.h	Check Updater condition
bl_check.c	
bl_main.c	Updater main

Definition in Bl_config

CRYSTAL_FREQ	Define crystal frequency for BootLoader
APP_START_ADDRESS	Define Application start Address
FLASH_RSVD_SPACE	Define Flash reserve size
STACK_SIZE	Define BootLoader stack
BUFFER_SIZE	Define BootLoader data receive buffer
ENABLE_BL_UPDATE	Enables updates to the boot loader
FLASH_CODE_PROTECTION	Flash code protection,erase the entire flash
ENABLE_DECRYPTION	Enables the call to decrypt
ENABLE_UPDATE_CHECK	Check Updater condition
FORCED_UPDATE_PERIPH	Forced Updater check for GPIO module

Definition in BI_config

- FORCED_UPDATE_PORT — Forced Updater check for GPIO base address
- FORCED_UPDATE_PIN — Forced Updater check for GPIO pin
- FORCED_UPDATE_POLARITY — Define the polarity of the GPIO pin that results in a forced update
- UART_ENABLE_UPDATE — Enable UART to update Flash
- UART_AUTOBAUD — Enable auto baud
- UART_FIXED_BAUDRATE — Define fixed baud
- SSI_ENABLE_UPDATE — Enable UART to update Flash
- I2C_ENABLE_UPDATE — Enable I2C to update Flash
- ENET_ENABLE_UPDATE — Enable Ethernet to update Flash

Bl_config.h

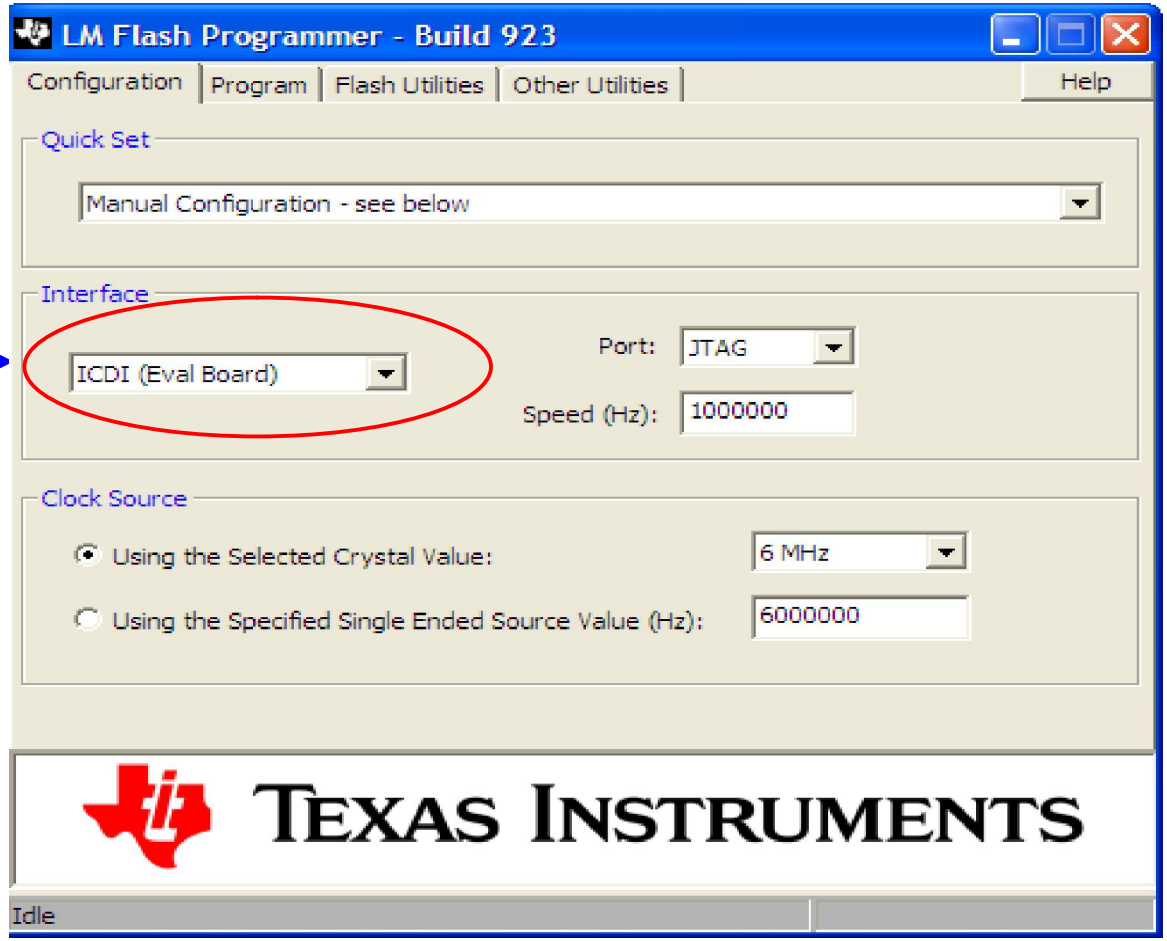
C:\Stellarisware\boards\ek-lm3s8962\boot_serial

```
#ifndef __BL_CONFIG_H__
#define __BL_CONFIG_H__
#define  CRYSTAL_FREQ              8000000
#define  APP_START_ADDRESS         0x1000
#define  STACK_SIZE                 48
#define  BUFFER_SIZE                20
#define  UART_ENABLE_UPDATE
#define  UART_FIXED_BAUDRATE       115200
```

Download user boot loader(serial) to flash

1. You could recompile the boot_serial project to get the binary file
2. Locate at C:\Stellarisware\boards\ek-lm3s8962\boot_serial

JTAG FTDI

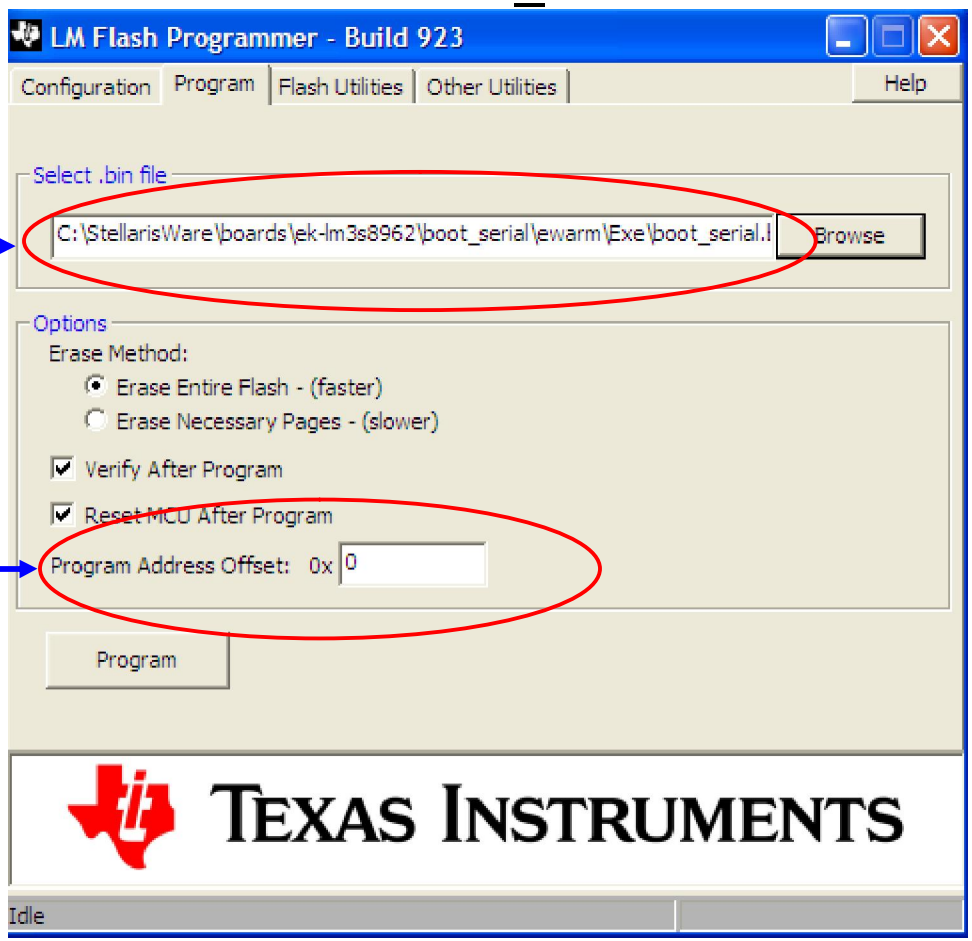


Download user boot loader to flash(cnt)

- Select the user boot loader file from C:\Stellarisware\boards\ek-lm3s8962\boot_serial

Select user binary file

Set start address



Program user code via user boot loader with UART0

- User code locate at C:\Stellarisware\boards\ek-lm3s8962\boot_demo1

COM2?

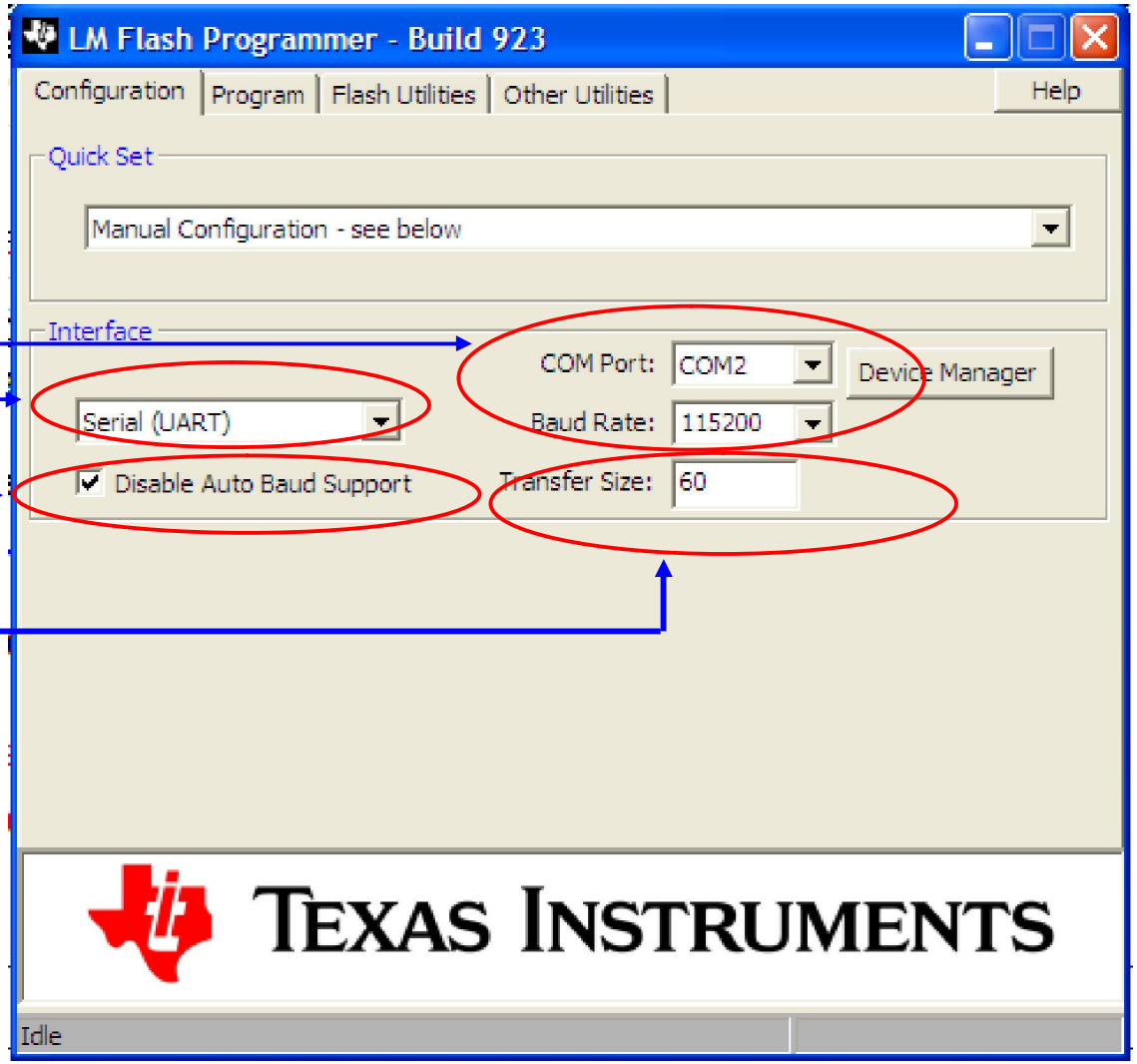
Serial(UART)

Dis Auto baud rate

Transfer size=60

***Use device manager to check your visual COM port**

***ISP function**

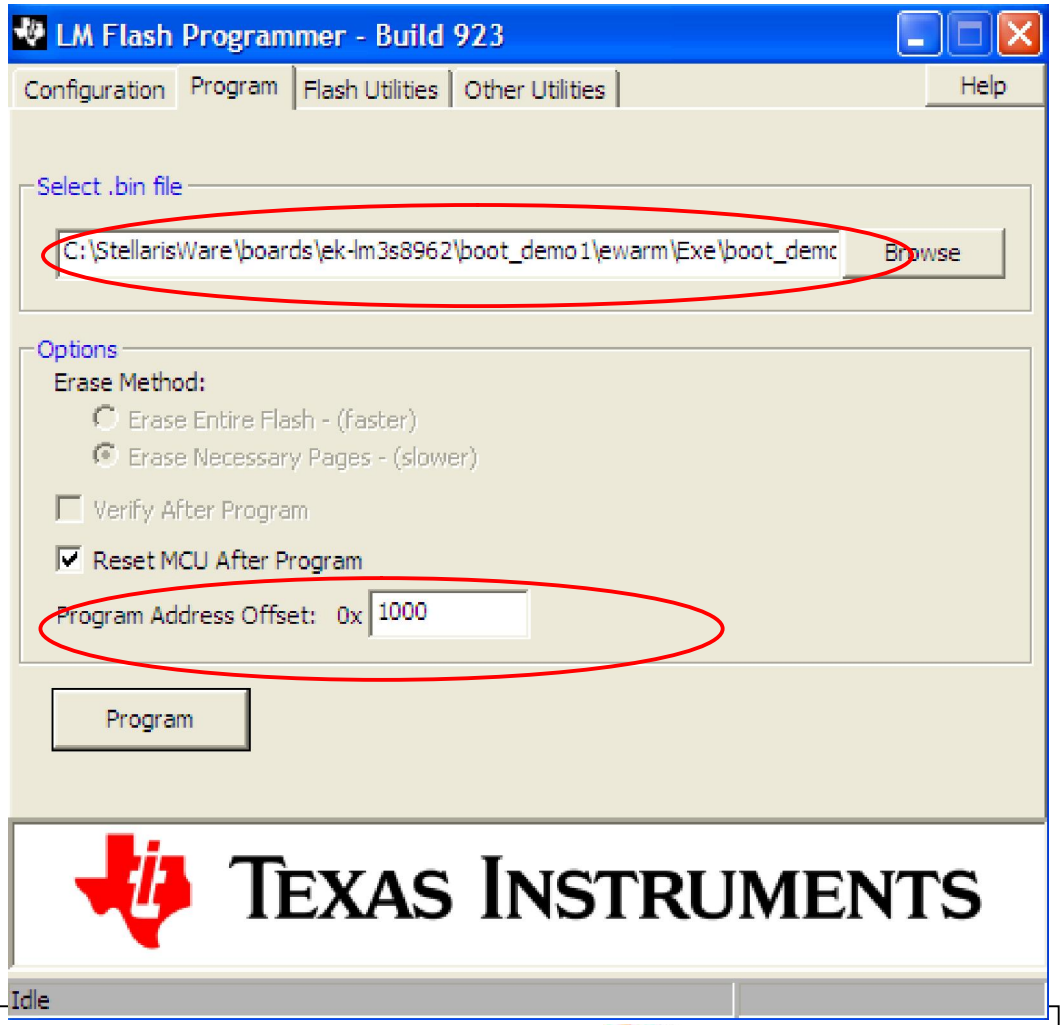


Program user code via user boot loader with UART0 (cnt)

- You could recompile the boot_demo1 project to get the binary file
- Locate at C:\Stellarisware\boards\ek-lm3s8962\boot_demo1

Select your user code

User code start address=0x1000

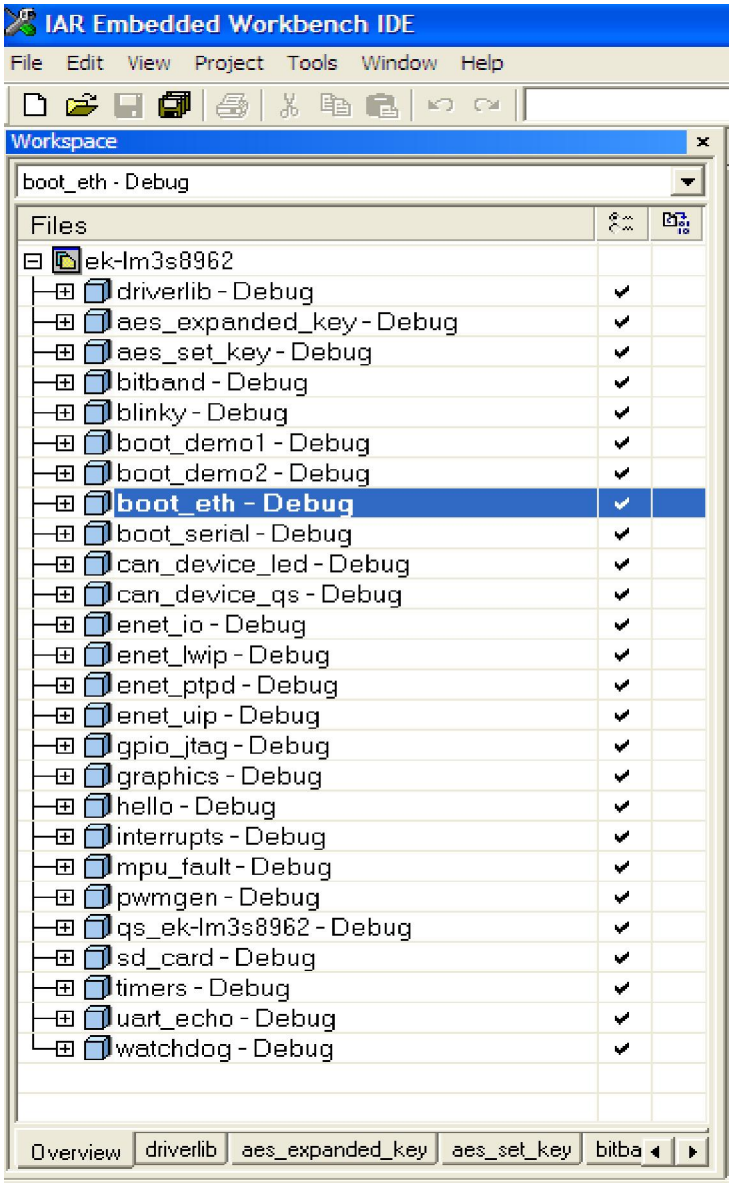


*IAP function

Download user boot loader(Ethernet) to flash

- Import the examples for the EK-LM3S8962
- C:\Stellarisware\boards\ek-lm3s8962\boot_eth

The boot_eth is a small piece of code that can be programmed at the beginning of flash to act as an application loader as well as an update mechanism for an application running on a Stellaris Microcontroller.

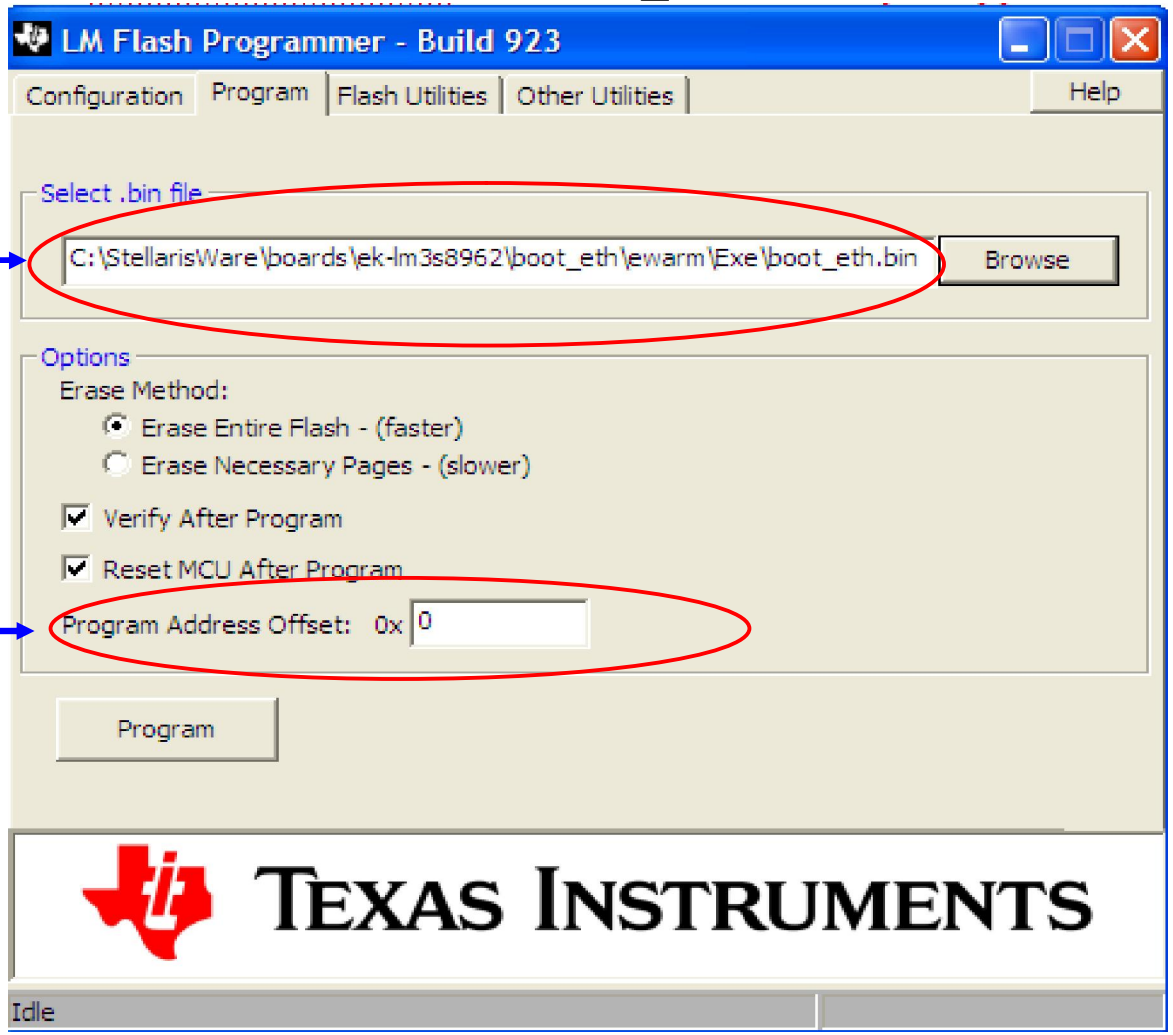


Download boot_eth.bin via LM Flash programmer

- You could recompile the boot_eth project to get the binary file
- Locate at C:\Stellarisware\boards\ek-lm3s8962\boot_eth

Select the bin file

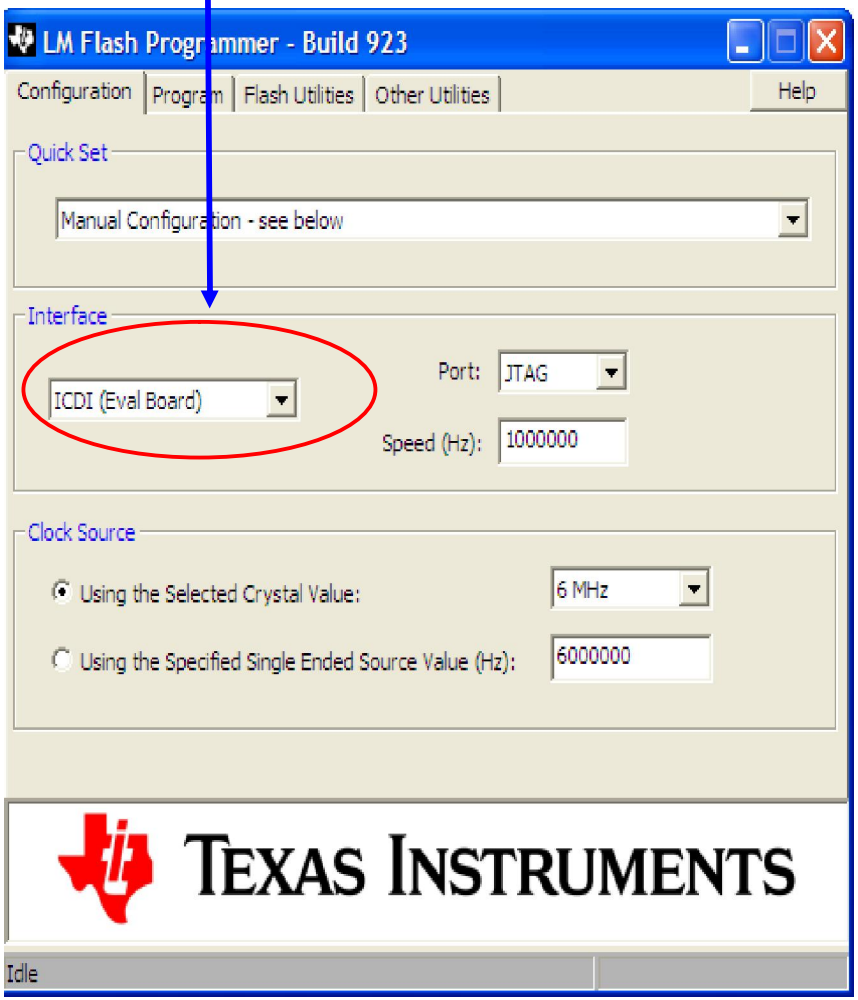
Set start address

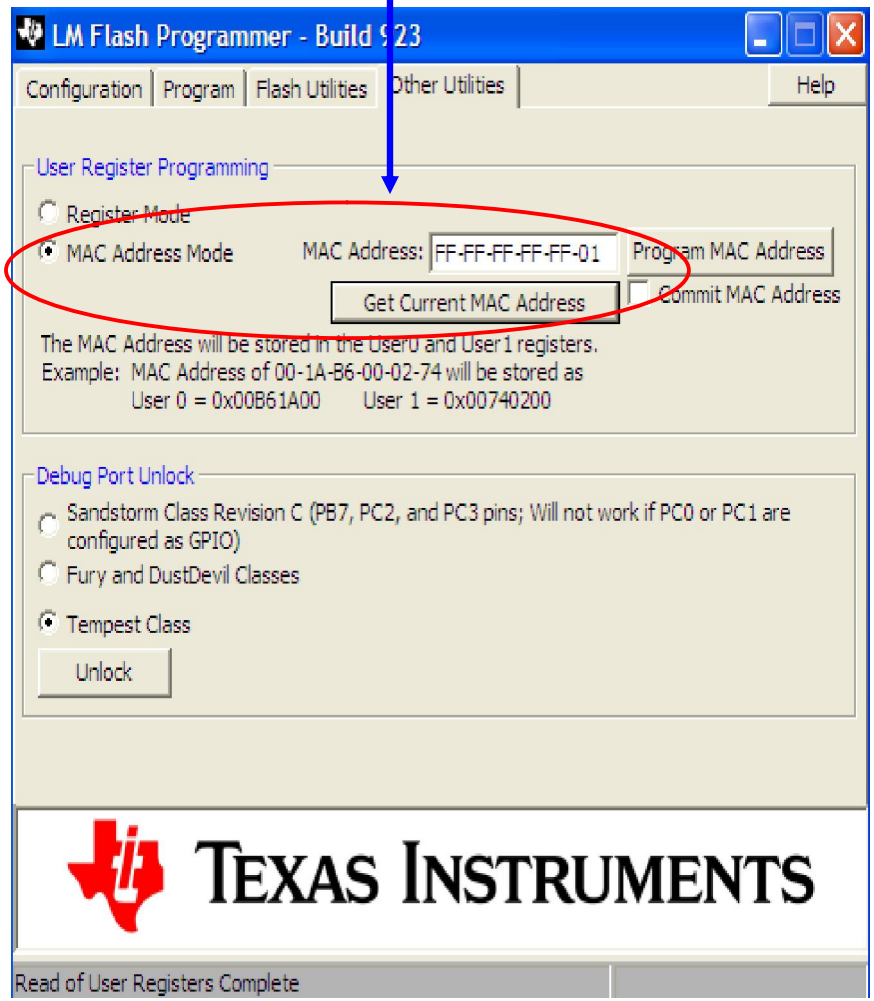


Get current board MAC address

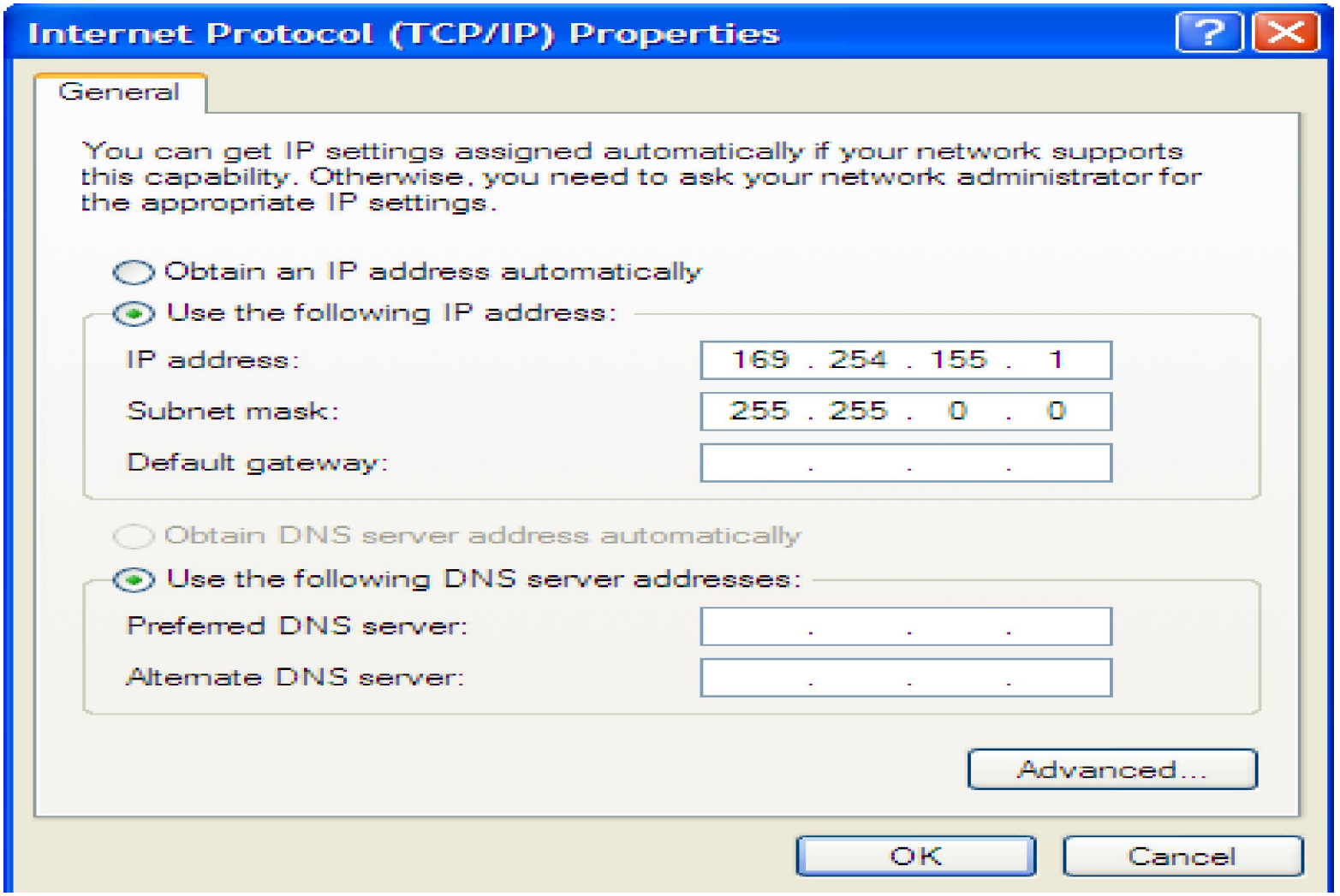
Select ICDI

Get current MAC address



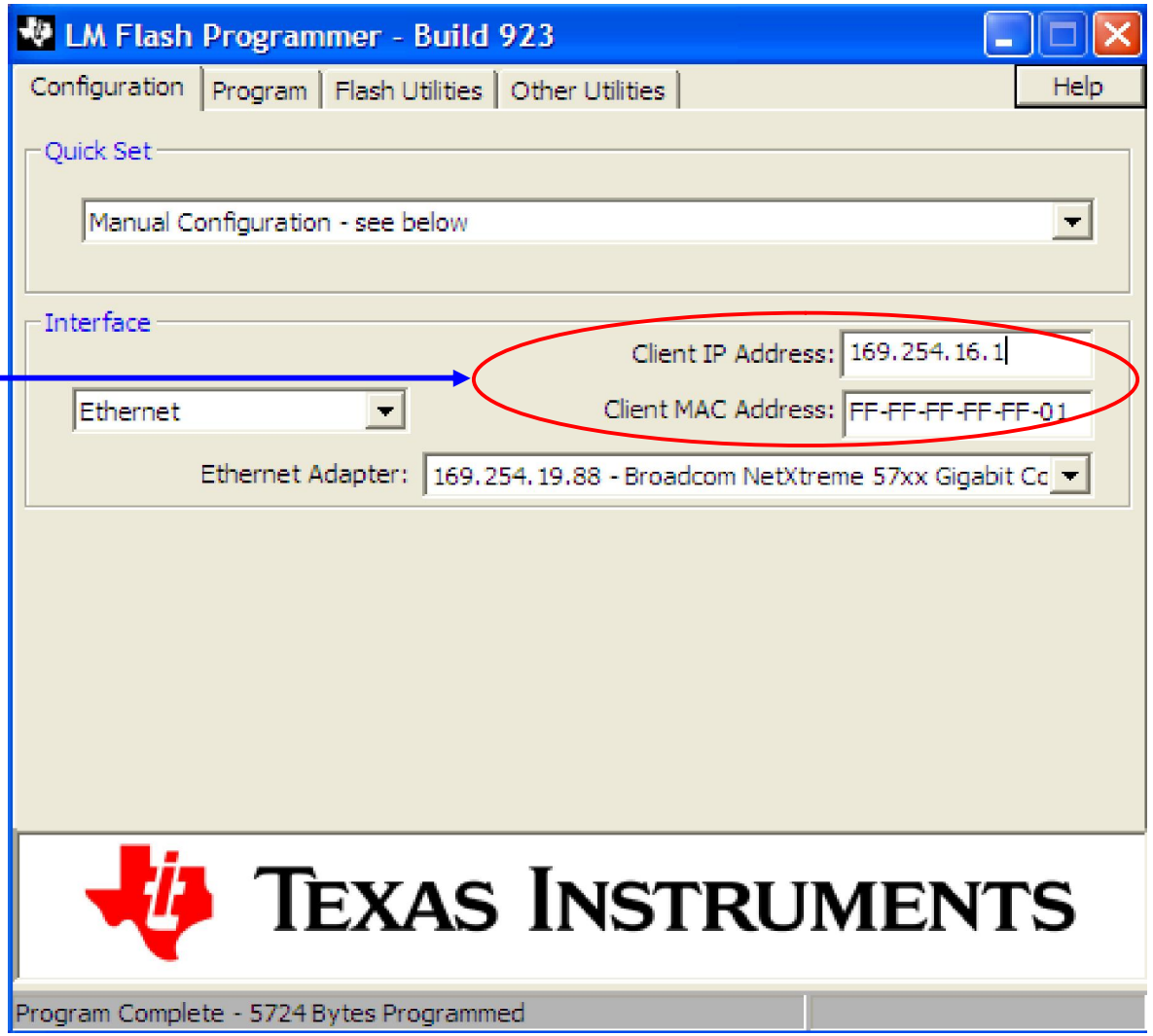


Configure your PC to static IP



Configure the Client IP,MAC address

Configure Client IP,MAC address



Download user code via Ethernet

- You could recompile the boot_demo1 project to get the binary file
- Locate at C:\Stellarisware\boards\ek-lm3s8962\boot_demo1

Select user code