Joy of innovation
nuvoTon

NuMicro® 8Bit

Quick Start

www.nuvoton.com

First use the NUVOTON microcontroller

- What do I need to know?
- What do to download?
- What are the steps?

Check Item

- **What compiler to use?**
  - KEIL, IAR
- **What can be used to emulator the execution of code?**
  - Nulink, Nulink me, Nulink pro, Nulink 2, Nulink 2 Me
  - NUVOTON Nutiny board
- **What tools are used to write BIN/HEX file to the chip?**
  - ICP (In circuit Programmer)
- **End customer need to updated FW? In UART, I2C, SPI…interface?**
  - ISP(In System Programmer)
- **Factory mass production tool?**
  - Nulink gang

KEIL Compiler

- **Downloading and installing each of the following steps is necessary.**
- **KEIL Evaluation form www.keil.com**
  - Install mdk.exe => Cortex M0/M23/M4
  - Install C51.exe => 8051
- **NUVOTON KEIL driver form www.nuvoton.com**
  - Nu-Link Driver Install Nu-Link_Keil_Driver.exe
- **Development Board Support Kit (BSP)**
  - Example MS51_Series_BSP_Keil_V1.00.003

IAR Compiler

- **Downloading and installing each of the following steps is necessary.**
- **IAR Evaluation form www.IAR.com**
  - Install IAR EMBEDDED WORKBENCH FOR 8051
- **NUVOTON IAR driver form www.nuvoton.com**
  - Nu-Link Driver Install IAR_Nu-Link_Driver
- **Development Board Support Kit (BSP)**
  - Example MS51_Series_BSP_IAR_V1.00.003

Emulator The Execution of Code

- **The compilation and driver software has been confirmed to be installed**
- **Find hardware NuMaker or NuTiny by development model**
  - https://direct.nuvoton.com/tw/8051-tool/

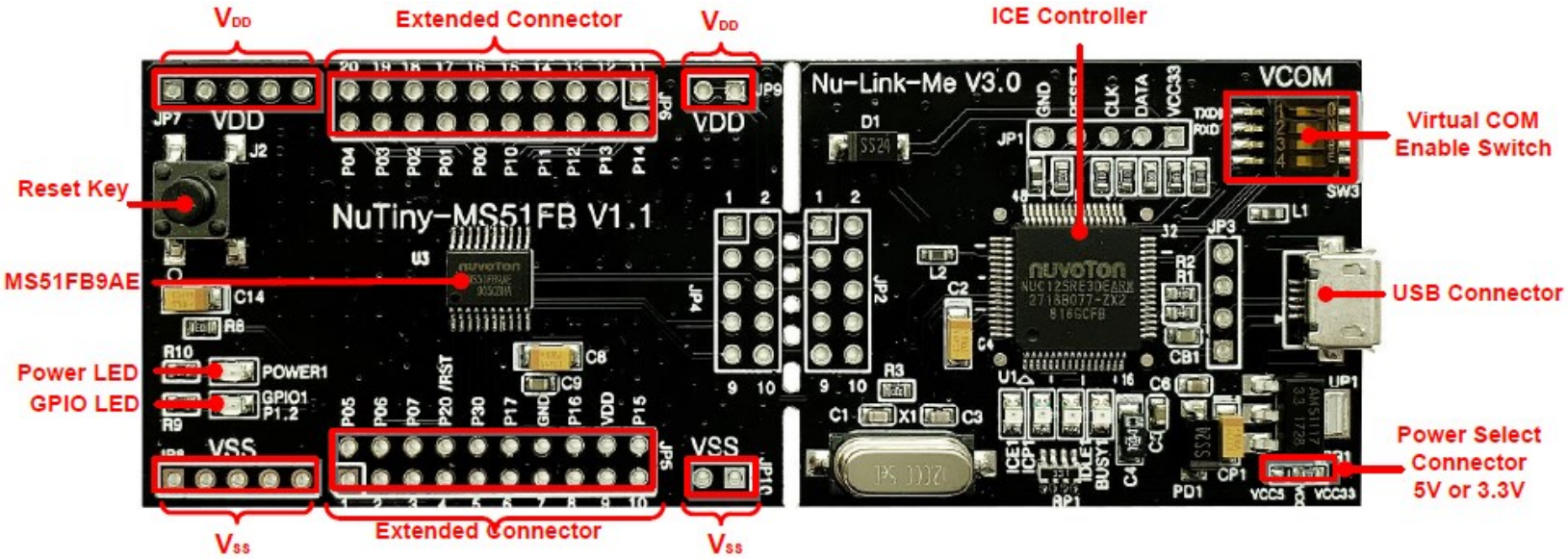
- **development board user manual, schematic, PCB file**
  - https://www.nuvoton.com/tool-and-software/evaluation-board/?index=1

NuTiny-ML51EB	ML51BB9AE
	ML51DB9AE
	ML51FB9AE
	ML51OB9AE
	ML51XB9AE
	ML51EB9AE
	ML51UB9AE
	ML51PB9AE
NuMaker-ML51PC	ML51TB9AE
	ML51EC0AE
	ML51UC0AE
	ML51PC0AE
	ML51TC0AE
NuTiny- MS51DA	MS51BA9AE
	MS51DA9AE
NuTiny-MS51FB	MS51XB9AE
	MS51XB9BE
	MS51FB9AE
NuMaker-MS51PC	MS51FC0AE
	MS51XC0BE
	MS51EC0AE
	MS51TC0AE
	MS51PC0AE
NuTiny-N76E003	N76E003AT20
	N76E003AQ20
	N76E003BQ20

NuTiny Board

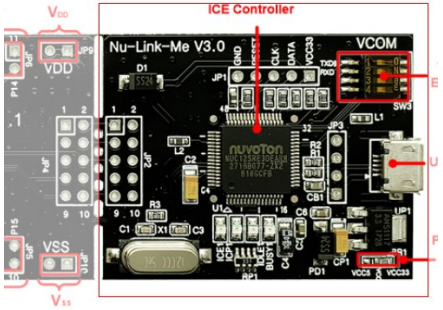
- An easy-to-develop platform for user to build the applications

VDD
DC 2.4 V ~ 5.5 V Input

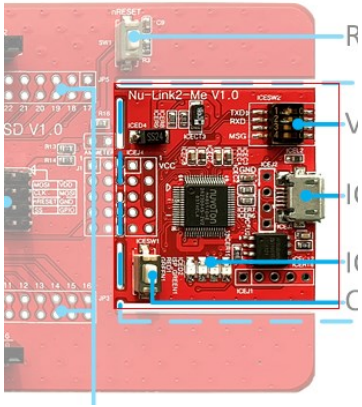


Write BIN/HEX File to The Chip

- **NULINK Serial**



Nu-Link

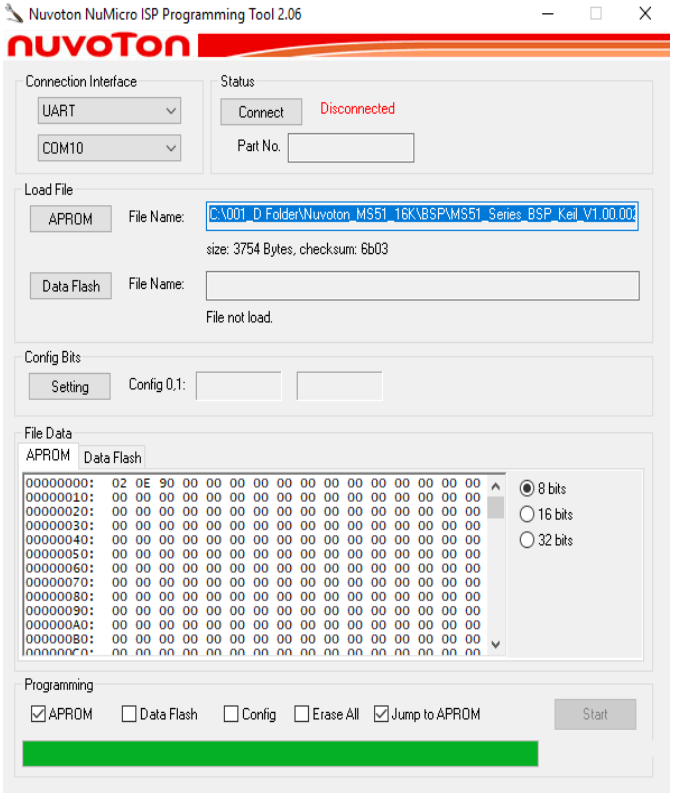


Nu-Link PRO

- **Software: NuMicro ICP Programmer Tool**

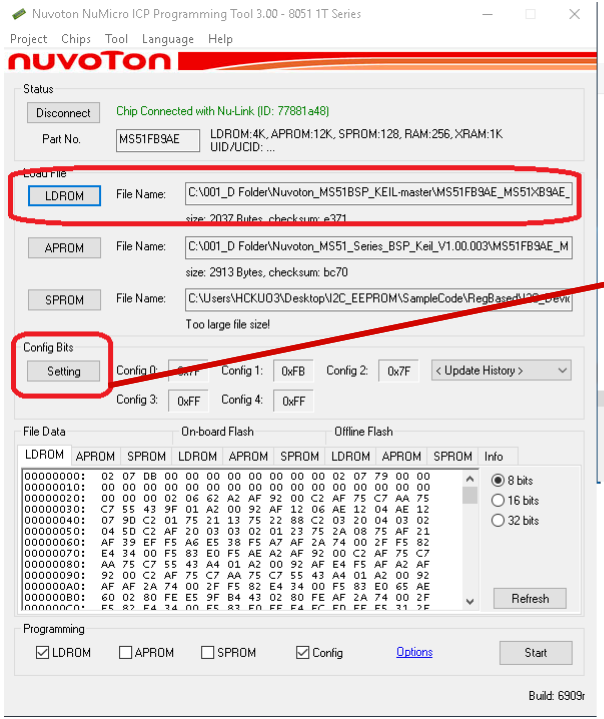
ISP(In System Programmer)

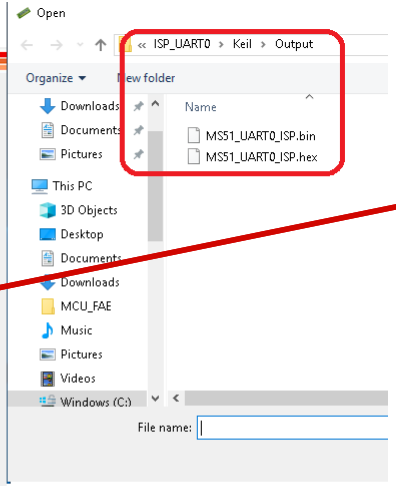
- **ISP Step**
  - The ISP bootloader code must first be written into the LDROM via ICP
  - Write APROM by communicating with PC's Nuvoton ISP Programming Tool through the UART of the IC
  - Bootloader code and ISP programmer tool open source code

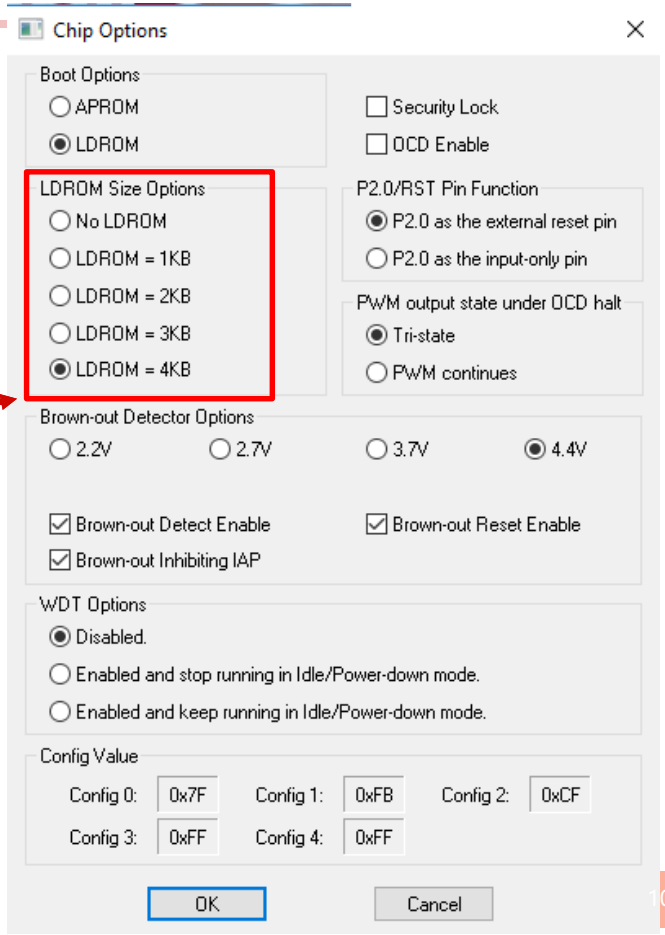


ICP Write boot loader Step

- Boot loader source: ISP_UART0
- LDROM Program boot loader







Mass Production Tools

- One time to 4 pcs
- Standalone word without PC
- NuLink Gang
  - https://www.nuvoton.com/tool-and-software/hardware-development-tool/programmer/

Joy of innovation
nuvoTon

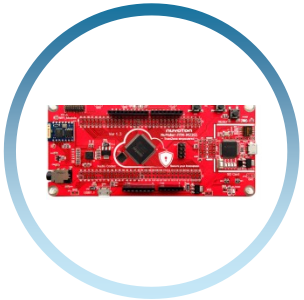
Nuvoton Development Environment

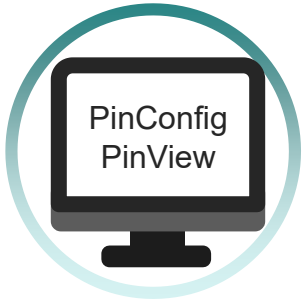
A Leading MCU Platform Provider

Sep, 2019

www.nuvoton.com

Development Environment and Tools





MCU Platform

- **Multiple Development IDE**
  - GNU GCC based NuEclipse on Windows or Linux
  - Keil MDK
  - IAR EWARM
- **Brand New Programming Platform - Nu-Link2**
  - Fast programming
  - ISP via multiple I/F
  - ETM
  - USB bridge
  - SPI / I^2C Monitoring
- **NuMicro IoT and HMI Development Platform**
  - Incl. sensor node, gateway, cloud database connection
  - Cloud service: Arm Pelion, Amazon AWS IoT , Alibaba Cloud
  - Comprehensive emWin GUI library
- **Easy-to-use Development Tool**
  - PinConfig: Graphical I/O setting
  - PinView: Real-time I/O monitoring
- **BSP (Board Supporting Package)**
  - Comprehensive drivers
  - Plentiful peripheral examples
  - API compatibility

Keil MDK Nuvoton Edition

– **Professional tool suite provided by arm, works with Nuvoton Cortex-M based MCU**

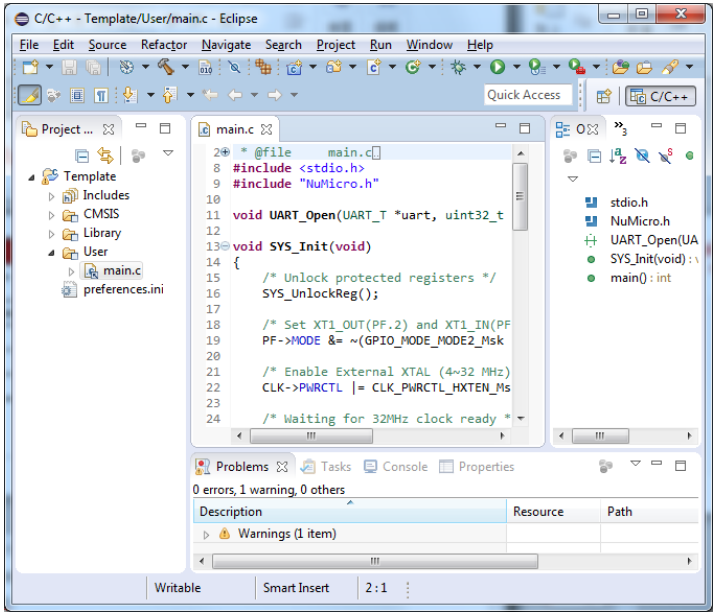
- Free-to-use professional tool suite for NuMicro® M0 and M23 product line
- Special offer to have a professional tool suite for NuMicro® M4 product line
- Includes the Arm C/C++ Compiler, the Keil RTX5 real-time operating system kernel, and the µVision IDE and debugger

Core	License Fee	License Term
Cortex-M0 /Cortex-M23	Free	1 year
Cortex-M4	US$395 for one user per year	1 year

NuEclipse (ARM Cortex series only)

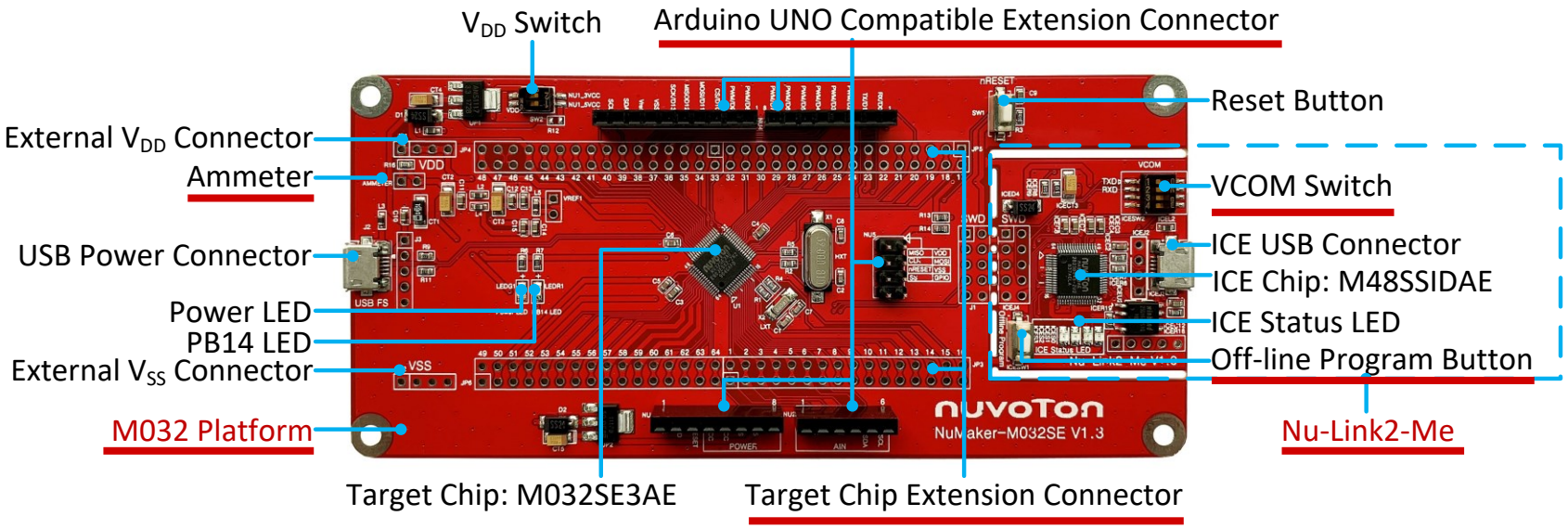
– **Free IDE to develop NuMicro® Family applications on Windows and Linux platform**

- Create, build, and debug projects of NuMicro® Family with in Eclipse framework
  - Create project by New Project Wizard.
  - Build project by GNU ARM Toolchain.
  - Debug project by GDB.
  - Supports Windows and GNU/Linux platform.
  - Designed for cross-platform embedded ARM development.



NuMaker EVB Board

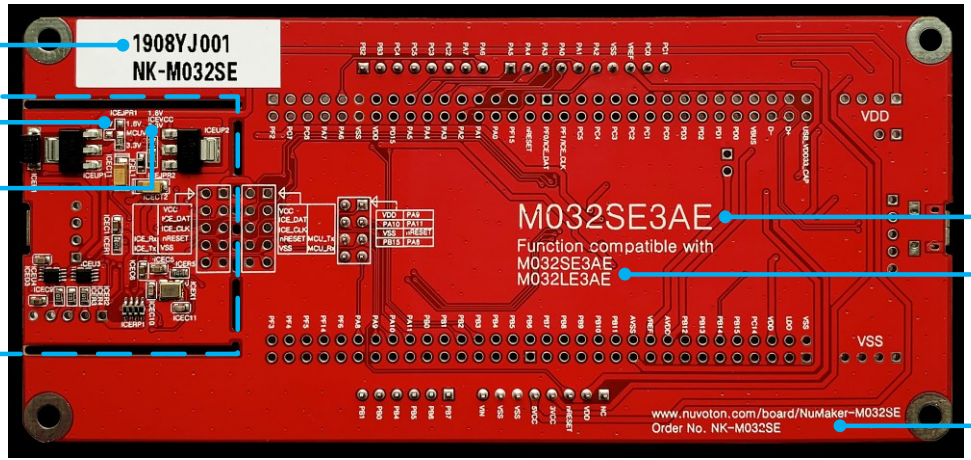
– An easy-to-develop platform for user to build the applications



- Extension connector: to expand the functionality and build the applications.
- Ammeter connector: easily monitor the target chip power consumption during development. (Remove R16)
- Nu-Link2-Me debugger and programmer
  - Supports on-line/off-line programming
  - Supports virtual COM port function

NuMaker EVB Board

– **An easy-to-develop platform for user to build the applications**



- Power Switch:
  - Configures the target chip operating voltage and ICE chip operating voltage

- Board Introduction page:

www.nuvoton.com/board/NuMaker-M032SE

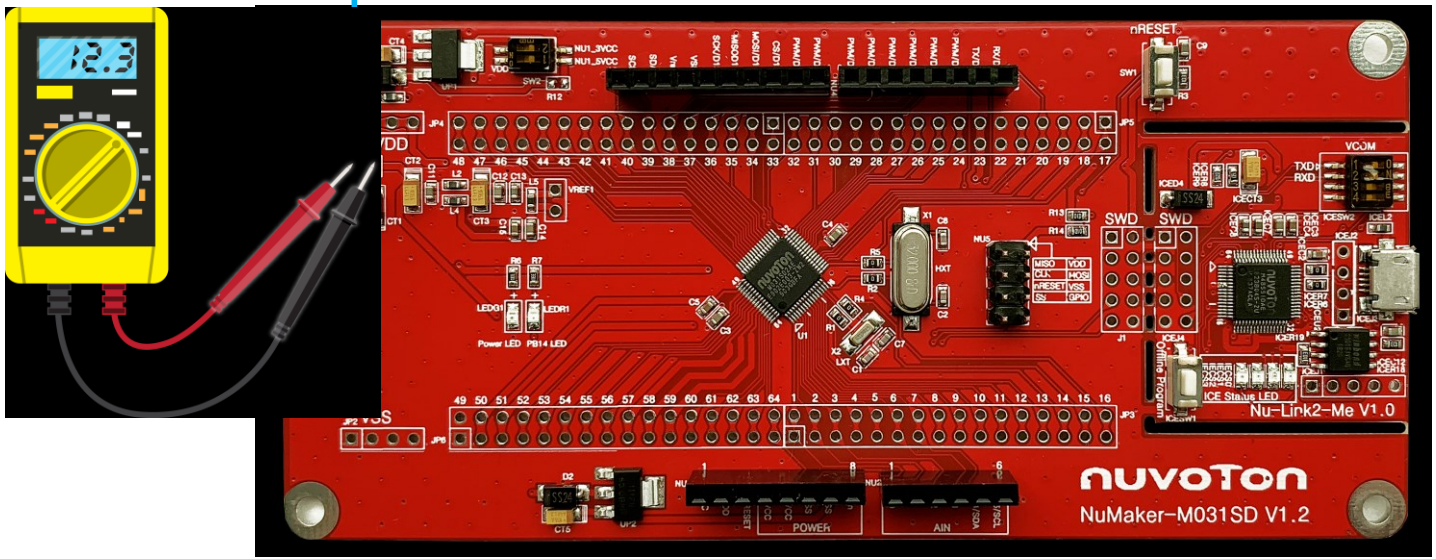
  - Board Introduction
  - User Manual
  - Quick Start Guides
    - Step-by-step build the development environment

NuMaker EVB Board - Ammeter Connector

– **An easy-to-develop platform for user to build the applications**

- **Connector for user to easily measure the target chip power consumption.**

Remove the R16 Resistor



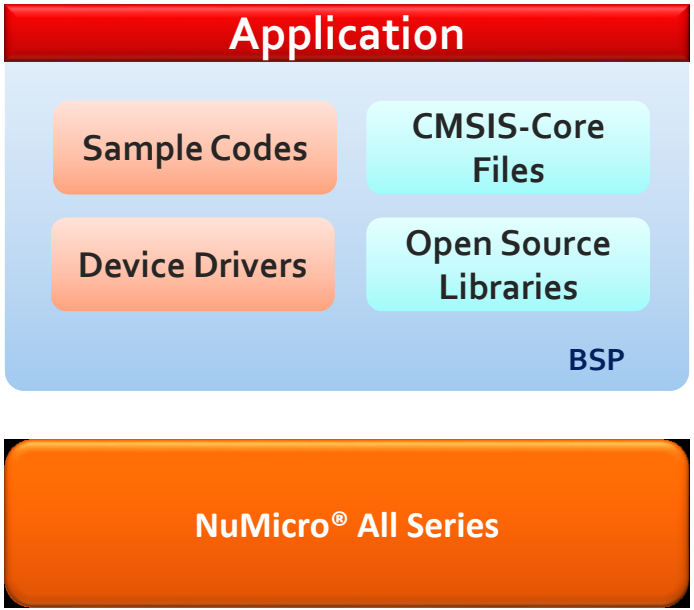
NuMaker EVB Board - Hardware Resource

Home > Support > Tool & Software > Development Tool Hardware > Development Board

- **NuMicro® Family MCUs and MPUs evaluation boards introduction**
- **Quick start**
  - Build the development environment by following the step-by-step guides
- **Download resources**
  - User manual
  - Schematic, PCB & Gerber File
  - NuMicro® Family chip library for Protel, OrCAD, and Pads Layout

User-friendly Board Support Package (BSP)

- Unified API names of all NuMicro® MCUs
- Supports KEIL MDK, IAR EWARM, and NuEclipse

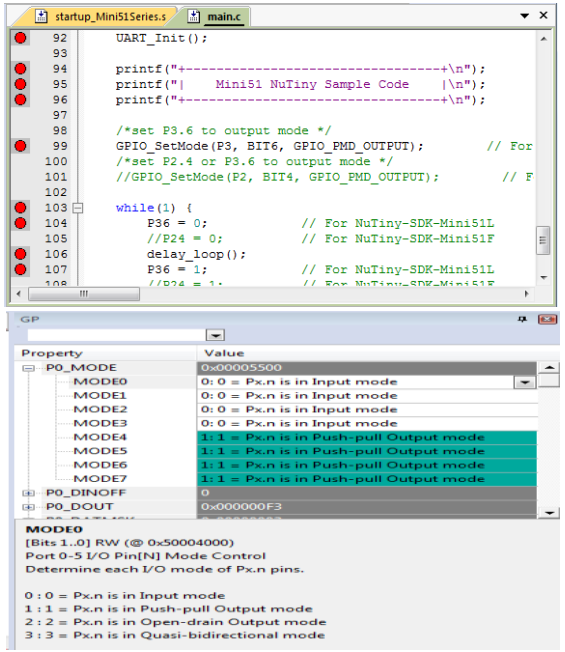


Nu-Link Driver for Keil & IAR

Joy of innovation
nuvoTon

– Supports Nu-Link to work under Keil MDK and IAR EWARM

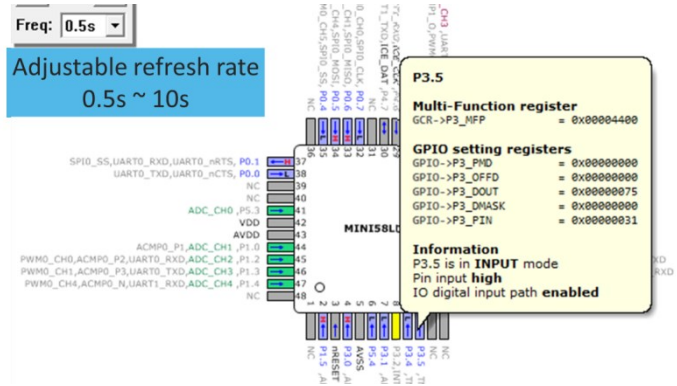
- Debugs, traces and analyzes programs running on the target board.
- Supports unlimited software breakpoints.
- Displays current register value on-line through SystemViewer and LiveWatch.
- Includes PinView to display real-time pin status and register settings.



PinView

– Monitors real-time pin status and register settings

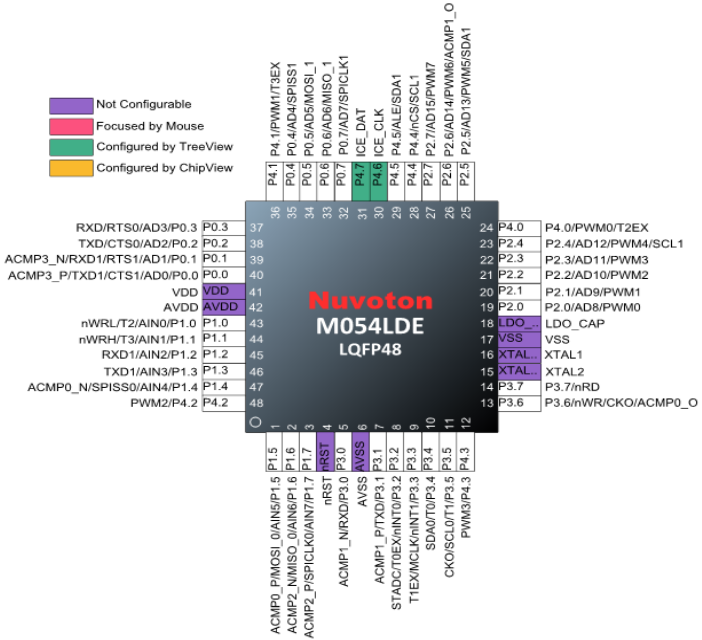
- Displays selected pin's configuration and highlights invalid settings.
- Supports two types of PinView:
  - Stand-alone PinView.
  - Add-on PinView for Keil µVision® IDE and IAR Embedded Workbench IDE.



PinConfig

– **Tool for pin assignment, initial code generation, and OrCAD/Protel part generation**

- Click and Select to configure pin setting.
  - Speed up pin assignment process.
- Generates GPIO function configuration code (.c file).
  - Speed up program development.
- Automatically generates corresponding OrCAD or Protel part.
  - Reduce the risk of erroneous hardware schematics.



Mass Production and Program Upgrade

- **Brand New Programming Platform - Nu-Link2**
- Fast programming
- ISP via multiple I/F
- ETM
- USB bridge
- SPI / I²C Monitoring

- **MP Tool - Nu-Link-Gang**
- One-to-four independent programming
- Auto Programming Machine

- **ICP Programming Tool**
- Online/offline programming
- Image bound to specific Nu-Link
- Encrypted project file for securing image
- Image serial number

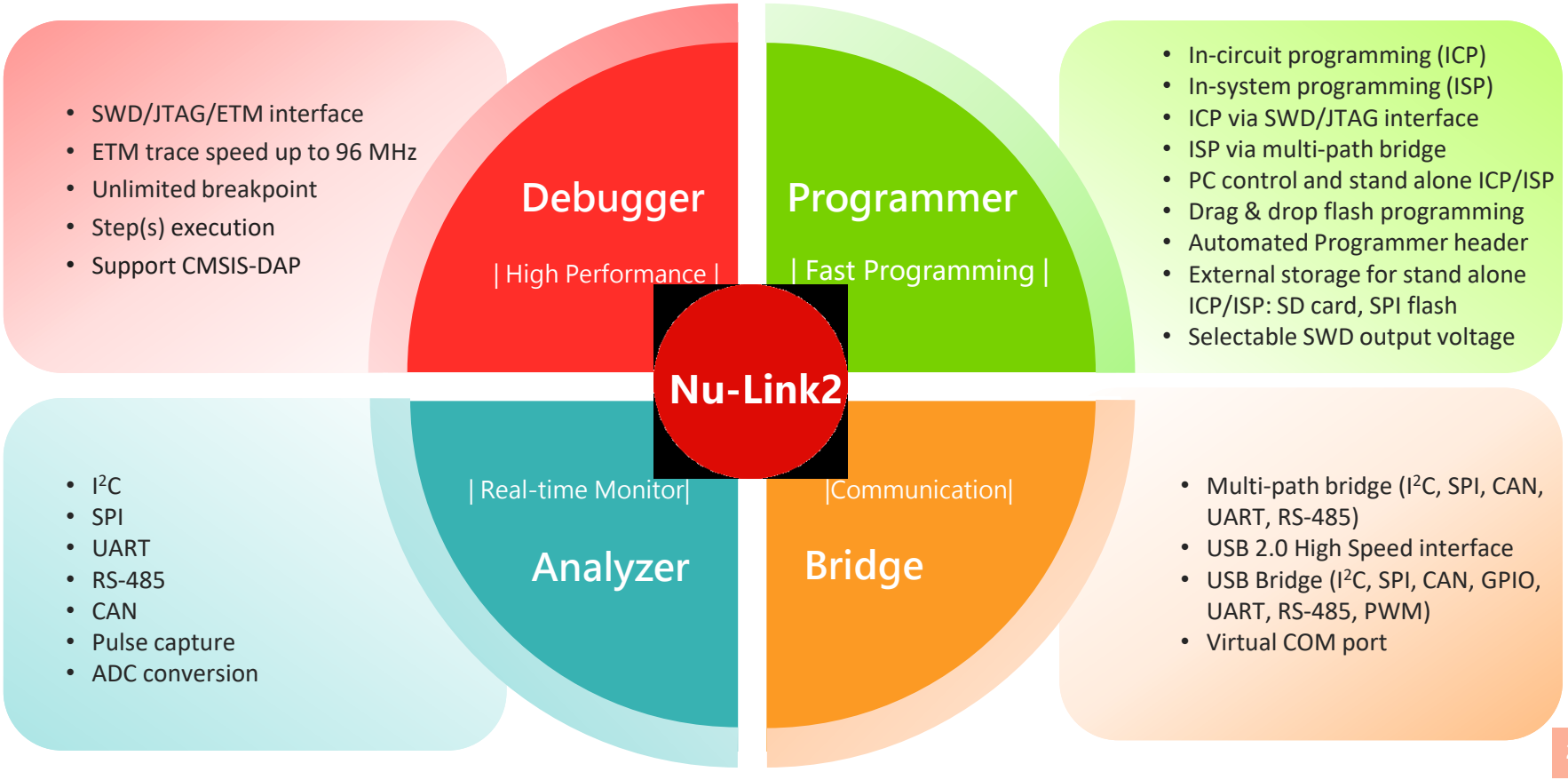
- **Customizable Programming Flow**
- Nu-Link Command Tool supports command execution from scripts or vendor GUI.
- Writer DIY supports flexible programming flow and self-defined test flow.

- **ISP Programming Tool**
- Standard ISP protocol via UART or USB.
- Source code of firmware and application tool.

Nu-Link Debugger

Function \ Type	Nu-Link-Pro	Nu-Link	Nu-Link-Me	Nu-Link2-Me
				
Debugging	v	v	v	v
Online Programming	v	v	v	v
Offline Programming	v	v	-	v
VCOM Port	-	-	v	v
SWD I/O Voltage Support	1.8 V, 2.5 V, 3.3 V, 5.0 V	5.0 V	3.3 V, 5.0 V	1.8 V, 3.3 V, 5.0 V
Control Bus (Automatic IC Programming System)	-	-	-	v

Nu-Link2



- SWD/JTAG/ETM interface
- ETM trace speed up to 96 MHz
- Unlimited breakpoint
- Step(s) execution
- Support CMSIS-DAP

- In-circuit programming (ICP)
- In-system programming (ISP)
- ICP via SWD/JTAG interface
- ISP via multi-path bridge
- PC control and stand alone ICP/ISP
- Drag & drop flash programming
- Automated Programmer header
- External storage for stand alone ICP/ISP: SD card, SPI flash
- Selectable SWD output voltage

- I^2C
- SPI
- UART
- RS-485
- CAN
- Pulse capture
- ADC conversion

- Multi-path bridge (I^2C, SPI, CAN, UART, RS-485)
- USB 2.0 High Speed interface
- USB Bridge (I^2C, SPI, CAN, GPIO, UART, RS-485, PWM)
- Virtual COM port

Nu-Link2

– All-in-one probe for NuMicro Family Microcontrollers

- **In-Circuit Programming**
  - via SWD/JTAG interface
    - Selectable SWD output voltage (1.8/2.5/3.3/5.0 V)
  - Supports PC control and stand alone ICP
    - PC control
      - ICP Programming Tool with image file protection
      - Drag & drop flash programming
    - Stand alone
      - SD card and SPI flash as image file storage
      - Press-to-start button
      - Automated Programmer header
      - Powered by USB or target-powered via SWD interface
- **In-System Programming**
  - via multi-path bridge
  - Supports PC control and stand alone ISP
    - PC control
      - ISP Programming Tool
    - Stand alone
      - SD card and SPI flash as image file storage
      - Press-to-start button
- **External Flash Programming**
  - On-board SPI flash Programming

Nu-Link2

– **All-in-one probe for NuMicro Family Microcontrollers**

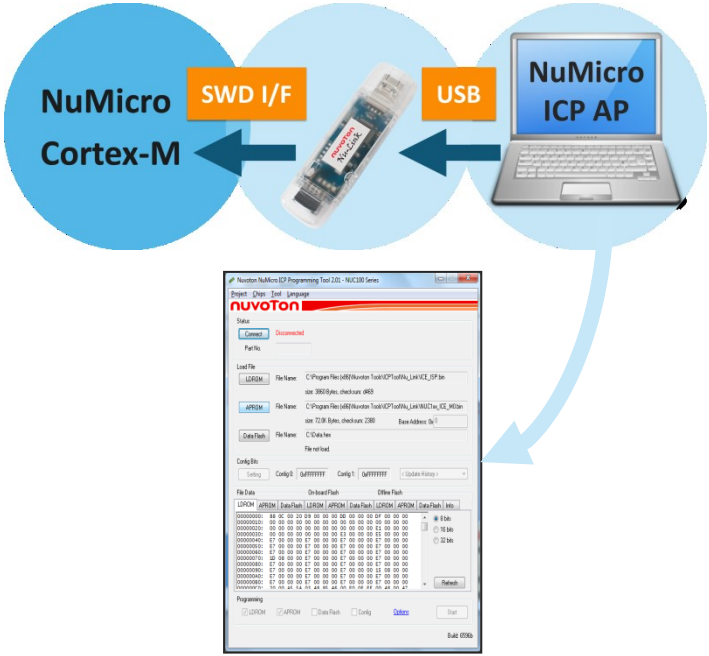
- **Debugger**
  - Interface: SWD / JTAG / ETM
  - ETM trace speed up to 96 MHz
  - Unlimited breakpoint and step execution
  - Support CMSIS-DAP
  - Monitor data in specific area (e.g. RAM, flash, or register)
- **Analyzer**
  - SPI, I^2C , CAN, UART and RS485 signal monitor
  - ADC conversion results
  - Pulse capture
- **Bridge**
  - Multi-path bridge (I^2C, SPI, CAN, UART, RS-485)
  - USB 2.0 High Speed interface
  - USB Bridge (I^2C, SPI, CAN, GPIO, UART, RS-485, PWM)
  - Virtual COM port

	Nuvoton Nu-Link2	ST ST-Link V3	TI MSP 430	ARM ULINK plus	NXP LPC-Link2	Microchip MPLAB PICkit 4
Core	M4	?	M4	?	M4/M0	M4
USB	USB HS	USB HS	USB HS	USB HS	USB HS	USB HS
Output Voltage	1.8~5.5 V	No support	1.8~3.6 V	?	?	1.2~5.5 V
SWD	16 MHz	24 MHz	8 MHz	50 MHz	V	V
JTAG	2 MHz	21 MHz	8 MHz	10 MHz	V	-
ETM	96 MHz	-	-	-	-	-
CMSIS-DAP	V	-	V	V	V	-
Virtual COM port	V	V	V		V	-
Drag-and-drop Flash programming	V	V	-	-	-	-
Program external SPI Flash	V	-	-	V	-	-
Offline Programming	V	-	-	-	-	V
Control Bus for IC programmer	V	-	-	-	-	-
SPI Flash	8MB	-	-	-	8MB	-
SDCARD	V	-	-	-	-	V
PWM Outout	1	-	-	-	-	-
Analog I/O	1	-	-	4	-	-
Data encryption	V	-	-	-	V	-
Monitor	I^2C/SPI/RS485/CAN/Capture	-	-	-	-	-
Bridge	I^2C/SPI/UART/ RS485/CAN	SPI/UART/I^2C/ CAN/GPIO	I^2C/SPI/UART	GPIO	I^2C/SPI/UART/ GPIO	I^2C/SPI/UART

ICP Programming Tool

– **Mass production software tool to program on-chip flash memory with Nu-Link**

- Supports on-line/off-line programming.
- Supports erasing/programming flash memory region(s).
- Supports setting and programming software serial number.
- Supports Nu-Link Certification
  - Supports firmware image files bound with certain Nu-Links.
  - Supports encrypted project file for electronic copy protection and content protection.



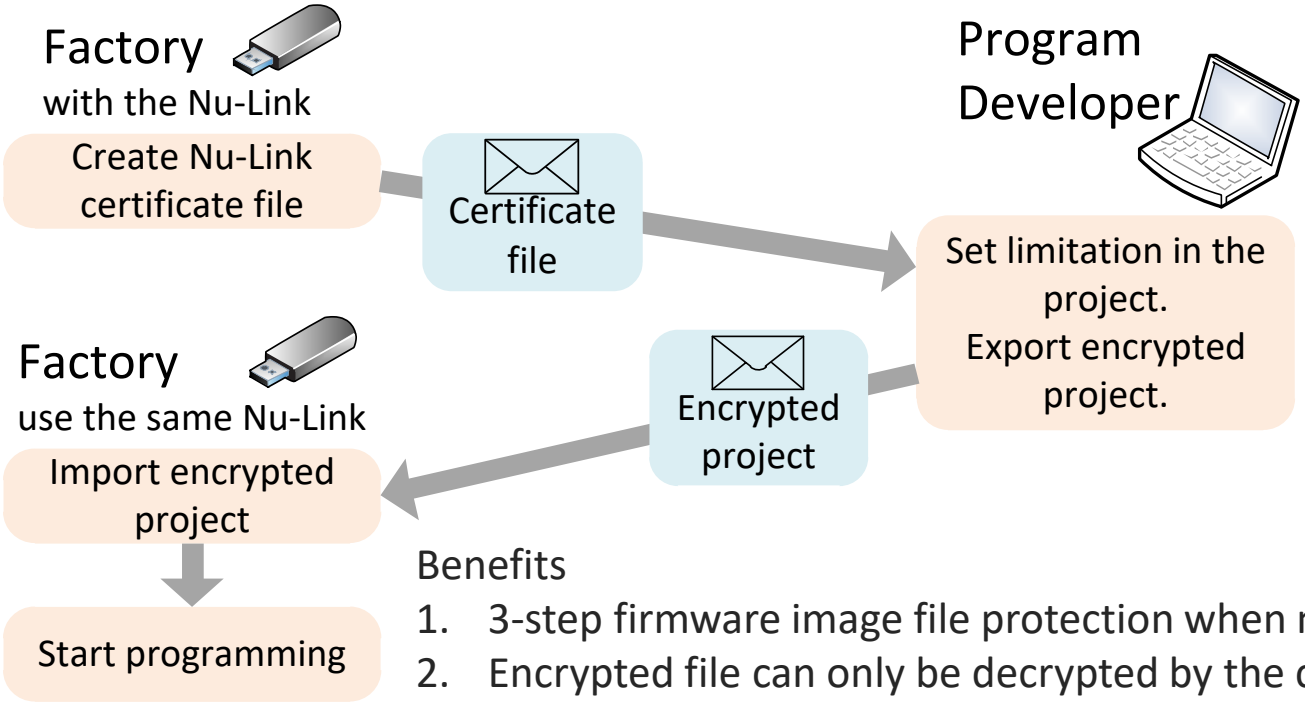
Nu-Link Certification

– Firmware image file protection supported by ICP Programming Tool

- **Scenario**
  - The product is expected to begin mass production.
  - How to pass the firmware image file to manufacturer in a secure way?
  - What if the manufacturer secretly use the firmware image file to make products by themselves?
  - What if the factory sell the firmware image file?
- **Tip**
  - Use Nu-Link and ICP Programming Tool to protect your hard work.
  - FAQ No.44 (https://www.nuvoton.com/hq/support/faq/801a0216-de9c-11e4-b643-4511a3410ebb/?__locale=en)

Nu-Link Certification (2/3)

– **Firmware image file protection supported by ICP Programming Tool**

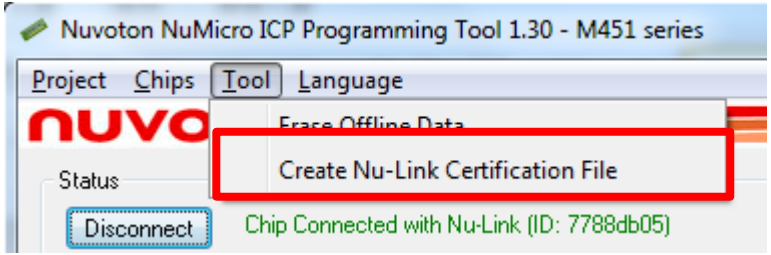


Benefits

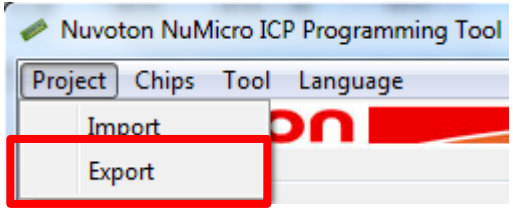
1. 3-step firmware image file protection when mass production.
2. Encrypted file can only be decrypted by the cert-creating Nu-Link.
3. Factory is not allowed to do unlimited mass production without the permission of developer.

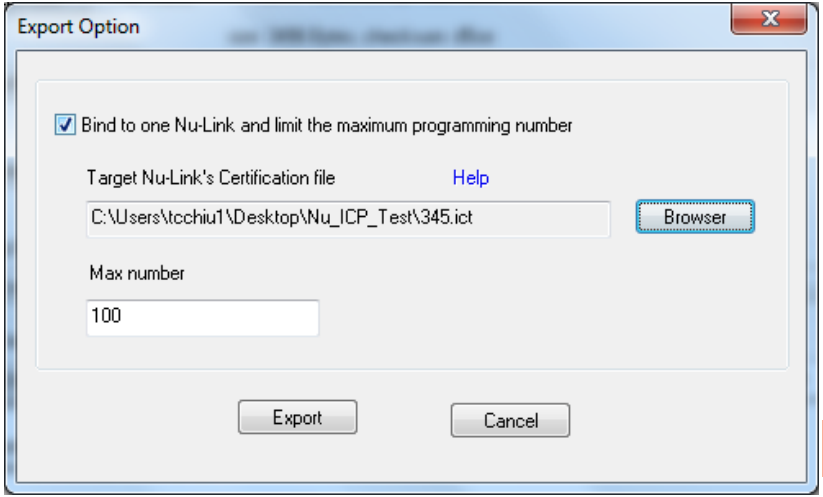
Nu-Link Certification (3/3)

- Step 1 – Factory creates Nu-Link certificate file and send to developer.

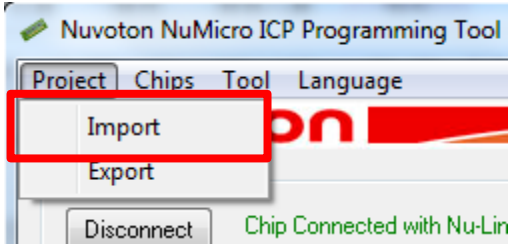


- Step 2 – Developer exports encrypted project with Nu-Link certificate file inside and send to factory.



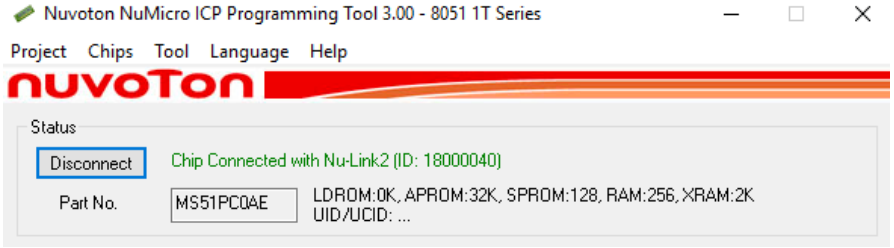


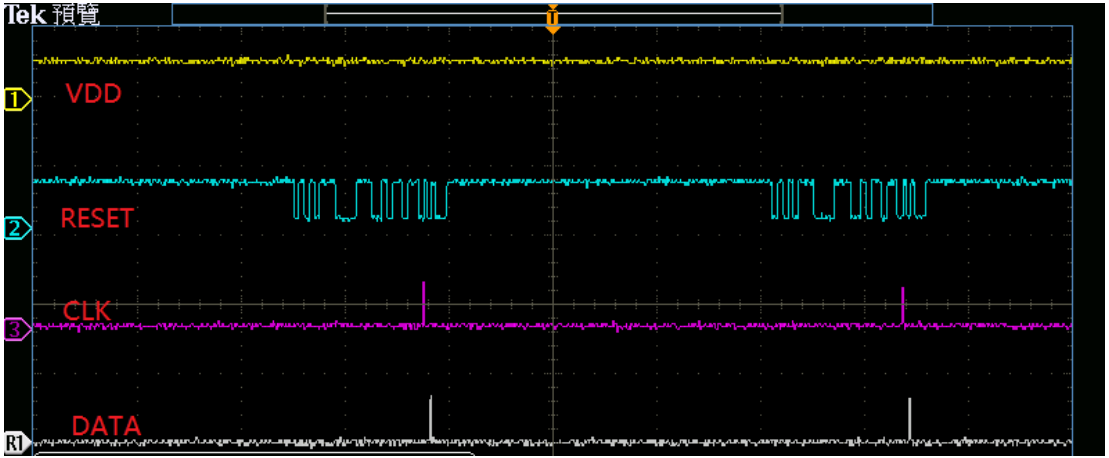
- Step 3 – Factory imports project and starts programming.



1T 8051 with ICP Programming Tool

- **Note: While creating connection with NuMicro® Family 1T 8051 MCUs, ICP Programming Tool will reset chip serval times.**

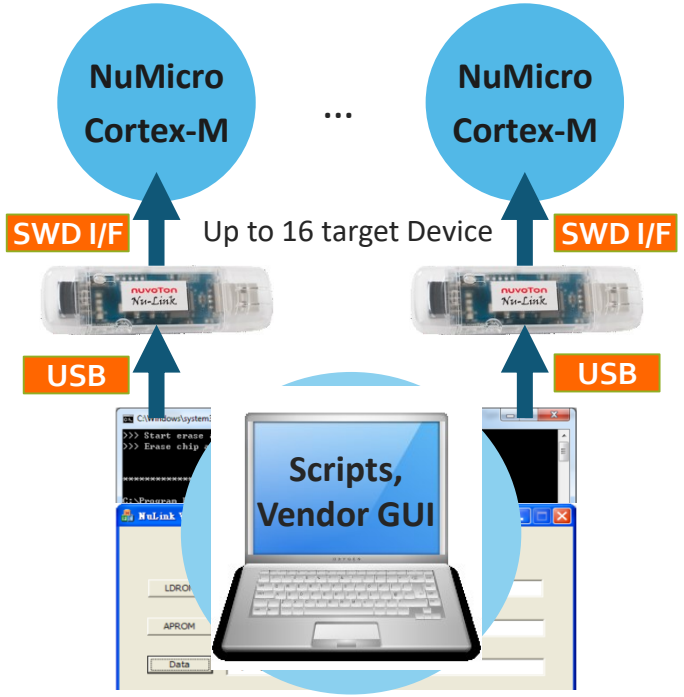




Nu-Link Command Tool

– **Customize Programming Tool for Nuvoton NuMicro Family microcontrollers**

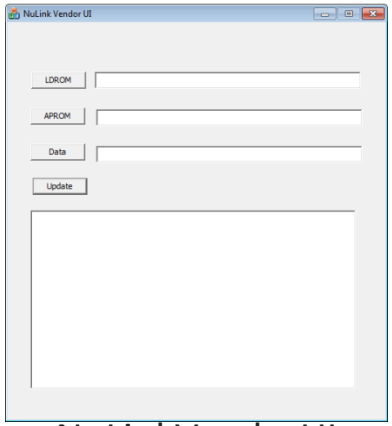
- Capable of managing up to 16 Nu-Link with 4 USB hubs
- Supports programming up to 16 target chips simultaneously
- Supports 11 commands and execution options
- Executes commands and log message from Vendor GUI or scripts to control the programming process



Nu-Link Command Tool - Resource

– **Customize Programming Tool for Nuvoton NuMicro Family microcontrollers**

- **Download Link**
  - http://www.nuvoton.com/opencms/resource-download.jsp?tp_GUID=SW0520160317094731



NuLinkVendorUI

- **Command List**
  - \Program Files\Nuvoton Tools\NuLink Command Tool 2.0\UM_EN_Nu-Link_Command_Tool.pdf

- **Error Code List**
  - \Program Files\Nuvoton Tools\NuLink Command Tool 2.0\ReturnCode.xlsx

- **UI for Customization**
  - \Program Files\Nuvoton Tools\NuLink Command Tool 2.0\NuLinkVendorUI.zip

Nu-Link Command Tool – Batch Example

– **Customize Programming Tool for Nuvoton NuMicro Family microcontrollers**

- **Batch Example for Erasing and Programming APROM**
  - Show message and log error code

```
@echo test
NuLink.exe -e APROM
@if %ERRORLEVEL% EQU 0 (
  @echo Erase APROM OK
  NULink.exe -w APROM C:\APROM.bin
  @if %ERRORLEVEL% EQU 0 (
    @echo Program APROM OK
    @goto Finish
  ) else (
    @echo Program APROM Error
    @goto ErrorLog
  )
) else (
  @echo Erase APROM Error
  @goto ErrorLog
)
```

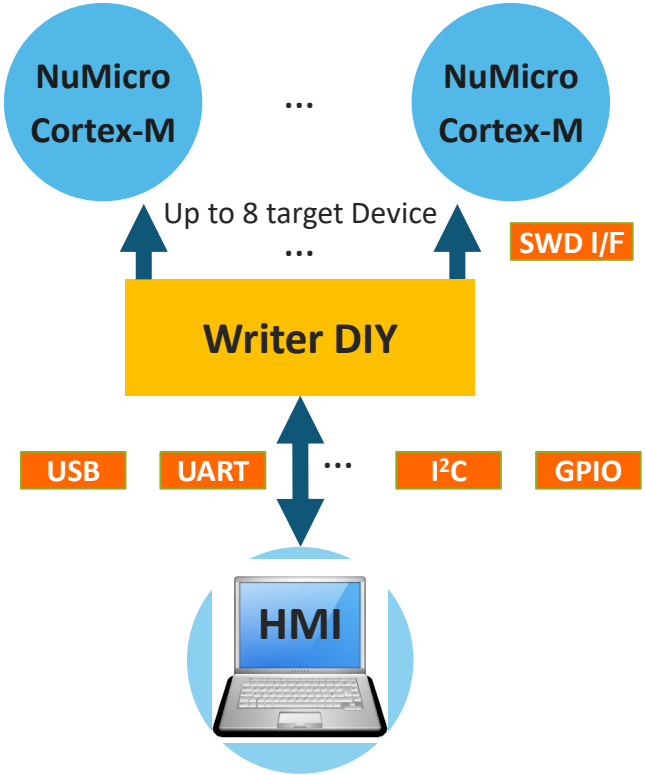
```
:ErrorLog
@set x=%date:~0,4%%date:~5,2%%date:~8,2%
@set y=%time:~0,2%%time:~3,2%%time:~6,2%
@echo Error Code = %ERRORLEVEL% >> Error-%x%-%y%.txt

:Finish
```

Writer DIY

– **Programmable writer with flexible programming flow and HMI connection**

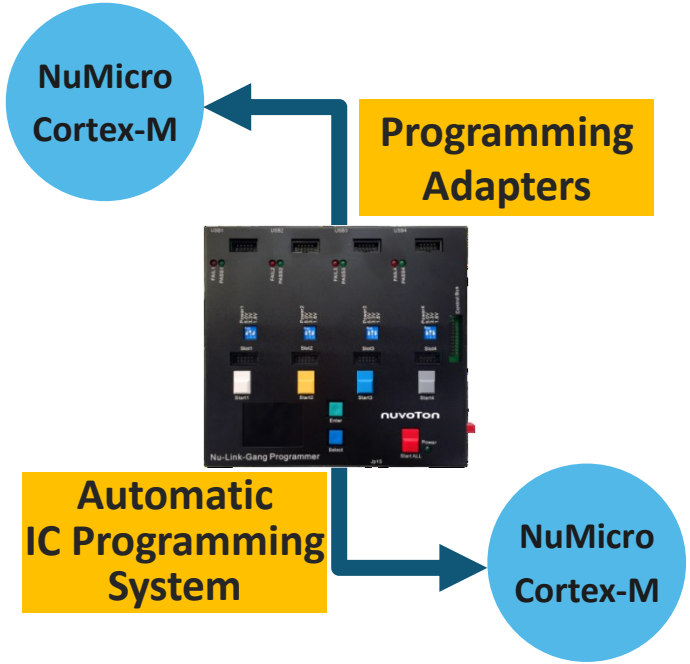
- Provides programming flow library and reference circuit
- Provides multiple programmable connection between Writer DIY and HMI, e.g. USB, UART, I²C, GPIO, etc.
- Supports programming up to 8 target HW simultaneously
- User-defined test flow can be programed into Writer DIY



Nu-Link-Gang Programmer

– **Mass production tool suitable for automatic IC Programming System**

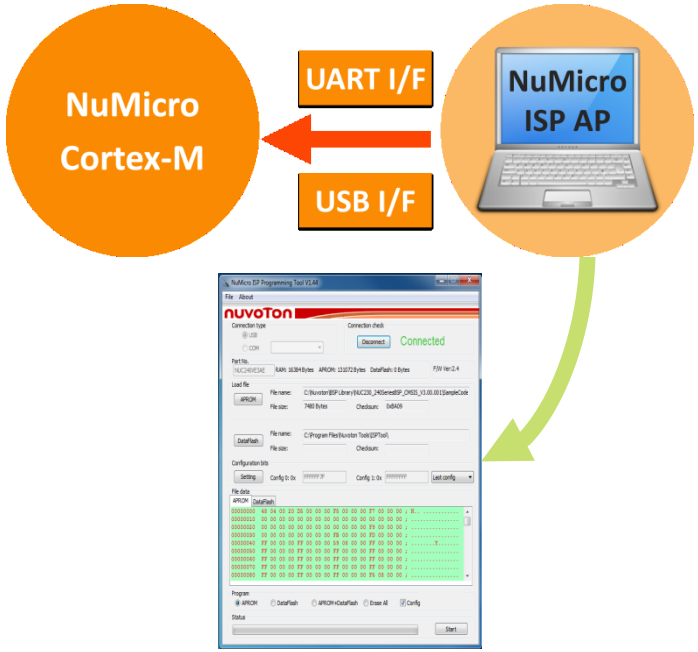
- Supports all Nuvoton NuMicro® Family and 8051 1T series and packages.
- Flexible Programming setting:
  - 3 options of programming voltage: 1.8 V, 3.3 V, and 5.0 V.
  - Off-line programming 4 different chips simultaneously or individually.
  - Supports automatic IC programming system.



ISP Programming Tool

– **Upgrade Tool for firmware upgrade through serial interface**

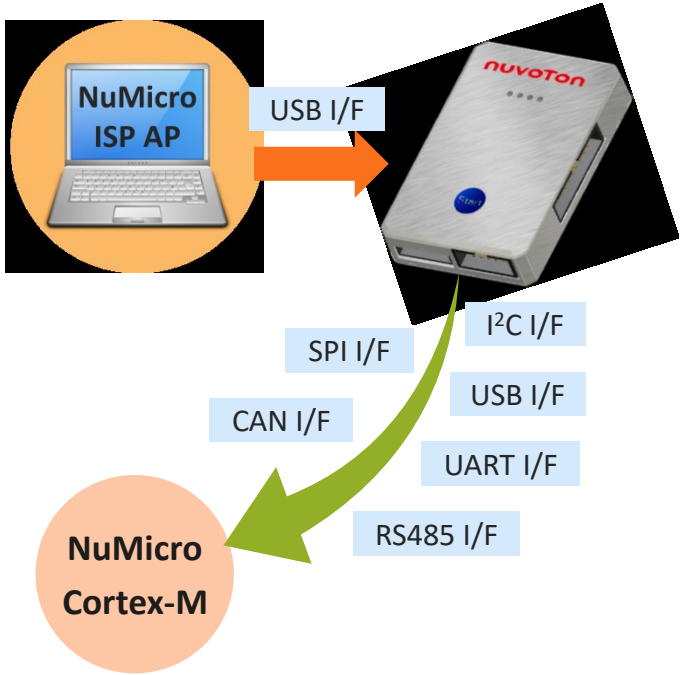
- Supports standard ISP protocols via UART and USB.
- Provides firmware and application tool source code.



Nu-Link2 ISP Programming Mode

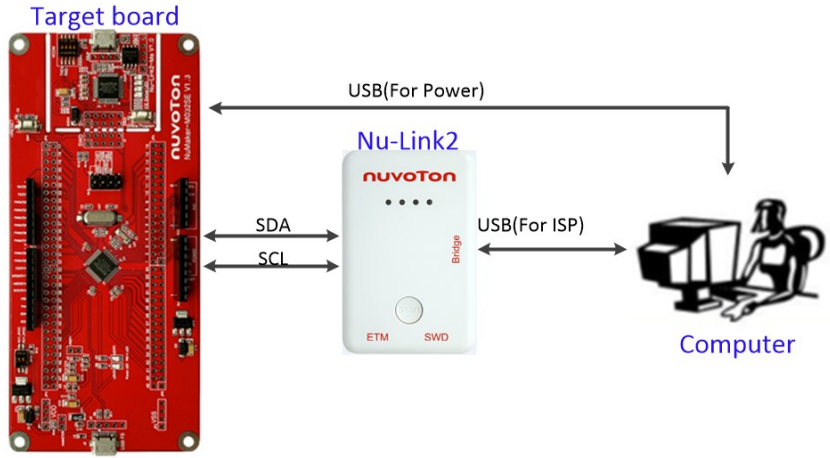
– Firmware upgrade through Nu-Link2

- On-line and off-line firmware upgrade
- Supports standard ISP protocols via SPI, I^2C, UART, RS485, CAN or USB
- Provides firmware and application tool source code



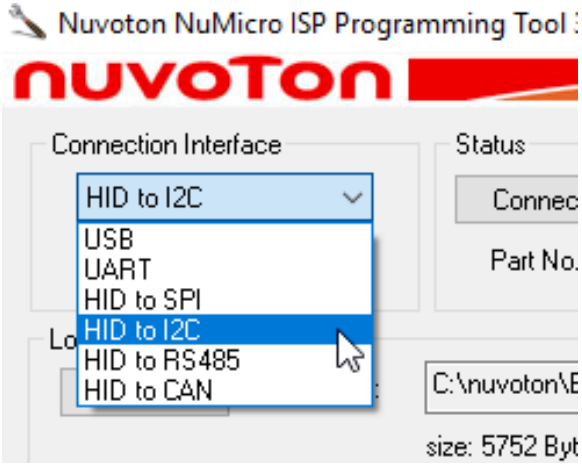
Nu-Link2 On-line ISP Programming Demo (1/3)

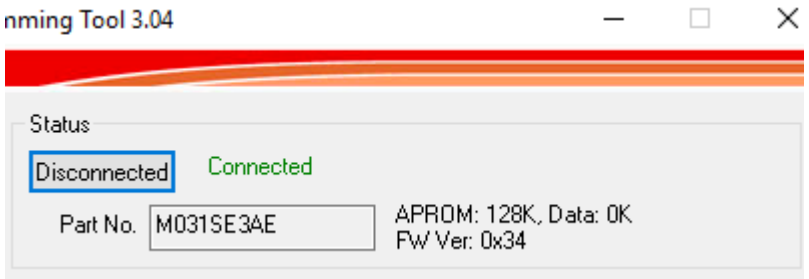
- **Preparation**
  - ISP Programming Tool (latest version)
  - Nu-Link2 (latest version)
  - Target board with standard ISP code in chip's LDROM
  - USB wire between Nu-Link2 and PC
  - Communication wire between Nu-Link2 and the target board
- **Connection (use I²C I/F for example)**



Nu-Link2 On-line ISP Programming Demo (2/3)

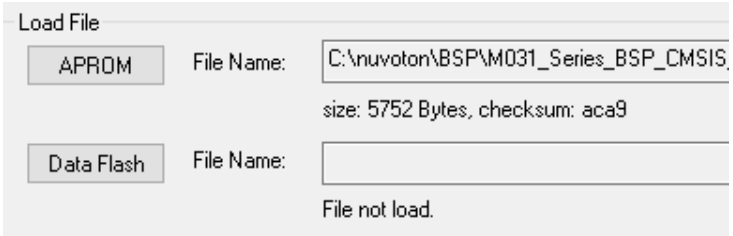
- Step 1 – Open ISP Programming Tool and select connection interface to “HID to I2C” to create connection.



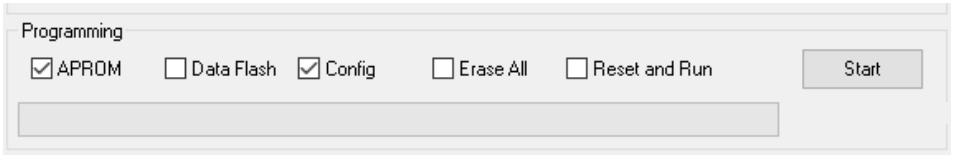


Nu-Link2 On-line ISP Programming Demo (3/3)

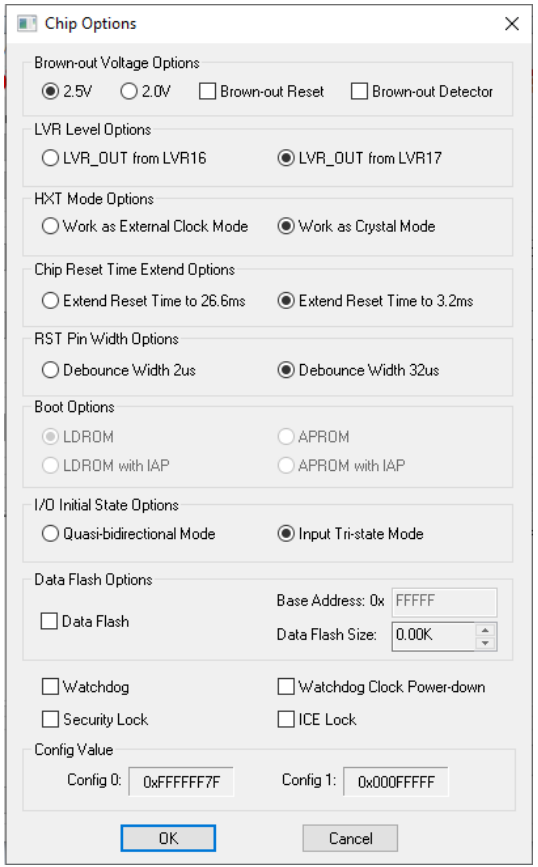
- Step 2 – Load APROM firmware image file.



- Step 3 – Set config value.
- Step 4 – Set programming area.



- Step 5 – Start programming.



Nuvoton 8051 Supported Tool List

Joy of innovation
nuvoTon

IDE and Compiler

Tool Name	Support Nuvoton MCUs
KEIL C51	8-bit
IAR EW8051	8-bit (1T)

Debugger & Programmer

Tool Name	Support Nuvoton MCUs
Nu-Link Adapters	8-bit (1T)
Nu-Link-Gang	8-bit (1T)

Programming Software Tool

Tool Name	Support Nuvoton MCUs
ICP Programming Tool	8-bit (1T)
Nu-Link Command Tool	8-bit (1T)
ISP Programming Tool	8-bit (1T)

Development Software Tool

Tool Name	Support Nuvoton MCUs
Nu-Link Keil Driver	8-bit(1T)
Nu-Link IAR Driver	8-bit(1T)
PinConfigure	ML51 Series

Online Resource

Nuvoton Website www.nuvoton.com

Forum

- http://forum.nuvoton.com
- http://www.nuvoton-MCU.com
- http://bbs.21ic.com/iclist-187-1.html

Media

- https://www.facebook.com/NuvotonNuMicro/
- ID: nuvoton_mcu
- @NuvotonMCU

BSP

- https:// github.com/OpenNuvoton
- https://gitlab.com/OpenNuvoton
- https://gitee.com/OpenNuvoton

eShop

- https://direct.nuvoton.com
- http://nuvoton.tmall.com/
- www.techdesign.com/market/nuvoton

Joy of innovation
nuvoTon

NuMicro® 8Bit

FAQ

Sep, 2019

www.nuvoton.com

The System Debug

- **Check MCU power voltage (VDD and AVDD) in work range.**

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
T_A	Temperature	-40	-	105	℃	
V_{DD}	Operation voltage	2.4	-	5.5	V	
AV_{DD}[*1]	Analog operation voltage	V_{DD}				
V_{BG}	Band-gap voltage[2]	1.17	1.22	1.30		T_A = 25 °C
		1.14		1.33		T_A = -40°C ~105 °C,

Note:

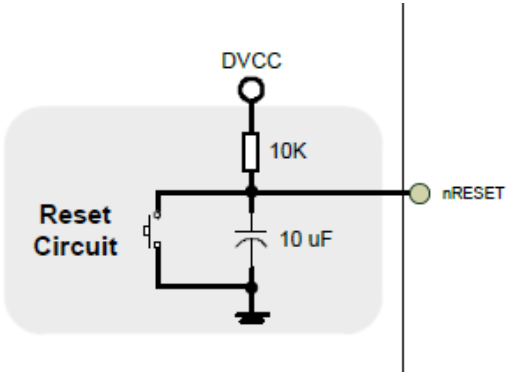
1. It is recommended to power V_{DD} and AV_{DD} from the same source. A maximum difference of 0.3V between V_{DD} and AV_{DD} can be tolerated during power-on and power-off operation .
2. Based on characterization, tested in production.

The System Debug

- Check reset pin voltage and schematic.
- If reset pin in low state, MCU is keep reset.

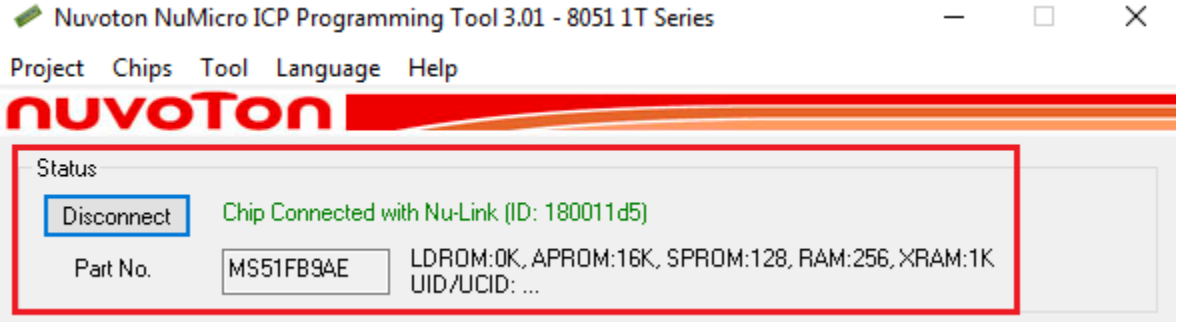
Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{ILR}	Negative going threshold, nRESET	-	-	$0.3*V_{DD}$	V	
V_{IHR}	Positive going threshold, nRESET	$0.7*V_{DD}$	-	-	V	
R_{RST}[1]	Internal nRESET pull up resistor	45	-	60	KΩ	V_{DD} = 5.5 V
		45	-	65		V_{DD} = 2.4 V
t_{FR}[1]	nRESET input response time	-	1.5	-	µs	Normal run and Idle mode
		10	-	25		Power down mode

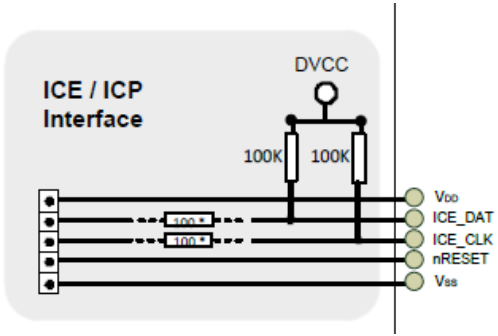
Notes:
1. Guaranteed by characterization result, not tested in production.
2. It is recommended to add a 10 kΩ and 10uF capacitor at nRESET pin to keep reset signal stable.



The System Debug

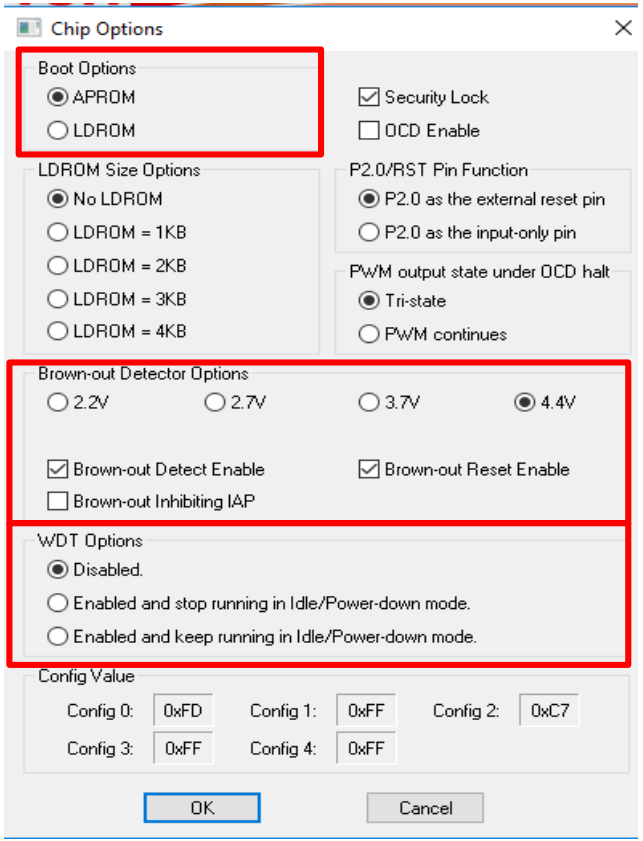
- Check ICE/ICP pin for schematic.
- Use NuLink and ICP Tool to connect target chip.





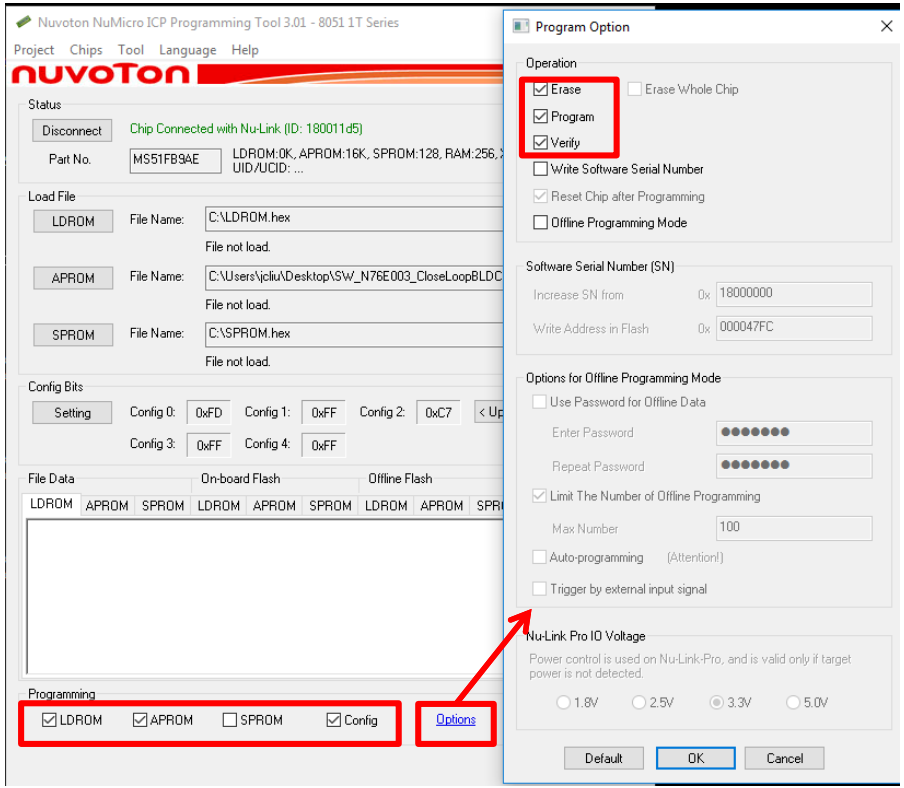
The System Debug

- **Check MCU Config Setting.**
  - MCU boot in APROM or LDROM.
  - WDT is enable, code no clear WDT counter, MCU always reset in 26 sec later.
  - Brown-out voltage (< system voltage) and reset enable.



The System Debug

- **Check Enable Erase, program and verify in ICP**



Is There a Way to Protect MCU

The APROM, LDROM is protect by config setting, code lock for security by CONFIG

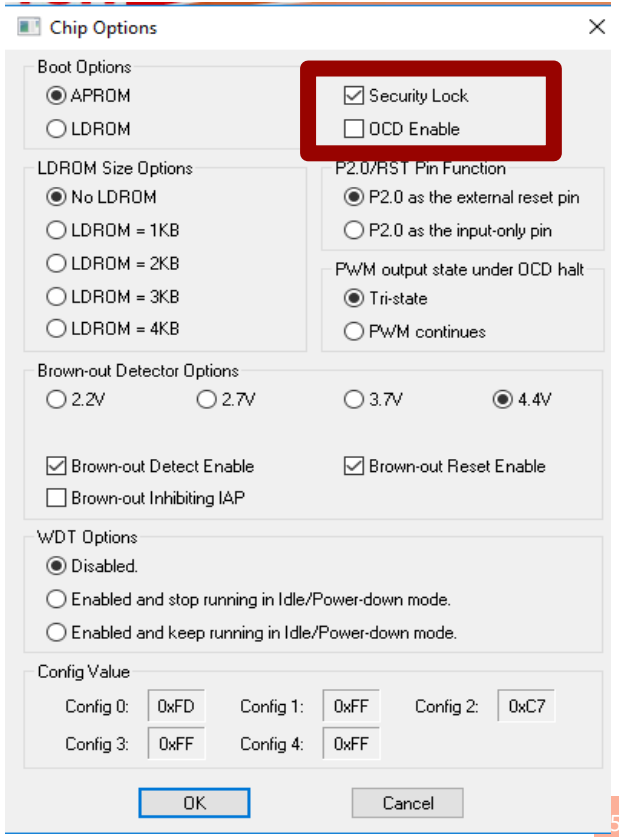
Yes, the prohibit access (Read / Write prohibition) of the whole ROM area(APROM, LDROM).

If MCU is lock (config lock is set), the APROM, LDROM is protect state, the general EPROM programmer/GANG programmer and NUVTON ICP)cannot read flash context.

The general EPROM programmer/GANG and NUVTON ICP programmer operation flow,

Erase APROM, LDROM, data flash => write APROM, LDROM, data flash => verify APROM, LDROM, data flash => lock chip in config

After chip is lock state, all flash area can't read by general EPROM programmer/GANG and NUVTON ICP programmer, only erase whole chip to unlock MCU.



Why SFRS is "push" / "pop" with Interrupt?

- **It want to avoid unexpected results while the value of SFRS Register changed before ISR occurs**

6.1.7.1 SFR Page Selection

To accommodate more than 128 SFR in the 0x80 to 0xFF address space, SFR paging has been implemented. By default, all SFR accesses target SFR Page 0. During device initialization, some SFR located on SFR Page 1/2 may need to be accessed. The register SFRS is used to switch SFR addressing page.

Note: In particular, since the interrupt is unpredictable, it is recommended to add following action in each interrupt subroutine. Push SFRS in the stack at the beginning of the interrupt vector program and pop SFRS when exiting the interrupt.

```
void PinInterrupt_ISR (void) interrupt 7
{
  _push_(SFRS);

  PIF = 0;

  _pop_(SFRS);
}
```

What`s GPIO Application

- **A, Output Mode:**
  - A-1, Push-Pull mode high source and sink current, it without external resistor to output “High or Low state” in user code.
  - A-2, Open-Drain mode need external or internal resistor to output high,
  - A-3, Quasi mode have “very weak” pull-high register(110KΩ-300KΩ@5V-2.5V) can driver high.
  - High driver: Push-Pull mode > Open-Drain mode > Quasi mode
- **B, Input mode:**
  - B-1, Input mode for pure input function.
  - B-2, Open-Drain Mode need external or internal resistor to output high, The MCU need output high to read high or low in pin register
  - B-3, Quasi mode have “very weak” pull-high register, it can driver high for read high or low in pin register

What`s GPIO Application

- **C, input/out mode:**
  - C-1, Open-Drain Mode need external or internal resistor to output high, the NMOS can driver to low, so need output high for external device drive to low. Link i2c application.
  - The Open-Drain Mode can pull external 3V-5V voltage with external devices.
  - C-2, Quasi mode have “very weak” pull-high register. it can driver high for external device to drive to low.
  - Output high: Open-Drain Mode > Quasi Mode
  - Speed: Open-Drain Mode > Quasi Mode
- **D, NC(float) mode:**
  - D-1, For the floating pin please use quasi mode connect to GND, avoid GPIO leakage and noise input.

What`s the internal pull high resister value

R_{PU} [*1] [*3]	Pull up resistor	40	-	60	kΩ	V_{DD} = 5.5 V, Quasi mode and Input mode with pull up enable
		40	-	60		V_{DD} = 3.3 V, Quasi mode and Input mode with pull up enable
		40	-	70		V_{DD} = 1.8 V, Quasi mode and Input mode pull up enable

What`s PWM output highest frequency @50% duty

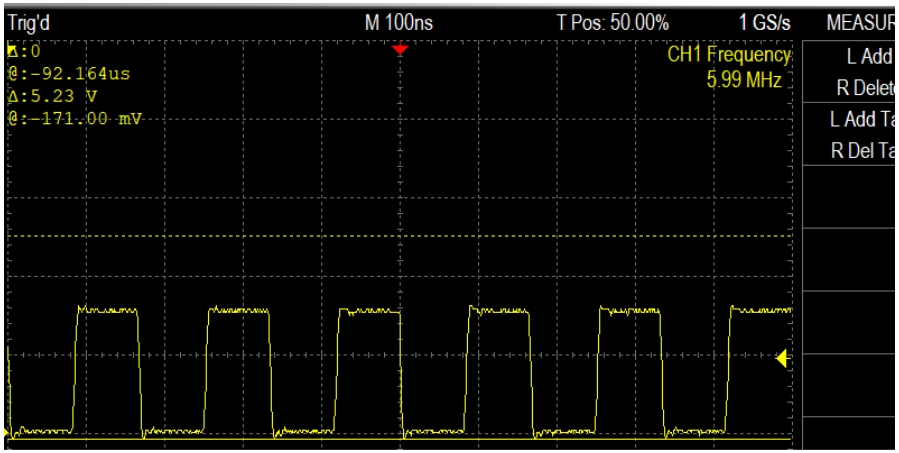
The PWM hardware design is same for MS51 and ML51.

PWM clock source 24Mhz, PWM clock divider 1.

PWMnPH, PWMnPL register value =>3

PWMnCHxH, PWMnCHxL register value=>2

The PWM output max speed 6Mhz for 50% duty @3.3V



Why to Updated PWM Frequency and Duty in Code

If your want load new value to PWM Frequency and duty register, you must set LOAD bit in PWM control register.

```
PWM1PH = 0x03; /*Setting PWM period */

PWM1PL = 0xFF;

PWM1C0H = 0x0;  /*PWM1 high duty = 1/2 PWM period */

PWM1C0L = 0xFF;

PWM0CON0 =PWM0CON0|0x40;  //set PWM0CON0 load new value to PWM duty
```

6	LOAD	PWM new period and duty load This bit is used to load period and duty control registers in their buffer if new period or duty value needs to be updated. The loading will act while a PWM period is completed. The new period and duty affected on the next PWM cycle. After the loading is complete, LOAD will be automatically cleared via hardware. The meaning of writing and reading LOAD bit is different. Writing: 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. Reading: 0 = A loading of new period and duty is finished. 1 = A loading of new period and duty is not yet finished.

What is the pull-up resistor value for I²C bus?

Joy of innovation NUVOTON

For the standard-mode I²C-bus system, the resistor value Rp in the figure above is determined by the following parameters:

I²C speed;

Supply voltage;

Bus capacitance;

The number of connected devices (input current + leakage current).

The supply voltage (V_{DD}) and the maximum output LOW level determine the minimum value of pull-up resistor Rp.

For example, a supply voltage of 4.5V, V_{OL}= 0.4 V and Rp is 4.7 kΩ.

Sink Current PA, PB, PC, PD, PE (Quasi-bidirectional and Push-pull Mode)	I_{SK11}	10	16	20	mA	V_{DD} = 4.5 V, V_S = 0.45 V
	I_{SK12}	7	10	13	mA	V_{DD} = 2.7 V, V_S = 0.45 V
	I_{SK13}	6	9	12	mA	V_{DD} = 2.5 V, V_S = 0.45 V

What is the pull-up resistor value for I²C bus

(4.5 - 0.45) / 4700 = 8.617 mA < Sink current.

The maximum bus capacitance of the value Rp limits is approximately 400 pf (capacitive load), calculated by cut-off frequency (1/2 * pi * Rp * Cb), to meet the requirement that the maximum tr (SCL/SDA rising time) is 300 ns.

I²C Dynamic Characteristics

Symbol	Parameter	Standard mode[1][2]		Fast mode[1][2]		Unit
		Min	Max	Min	Max	
t_{LOW}	SCL low period	4.7		1.2		uS
t_{HIGH}	SCL high period	4		0.6		uS
$t_{SU;STA}$	Repeated START condition setup tim	4.7		1.2		uS
$t_{HD;STA}$	START condition hold time	4		0.6		uS
$t_{SU;STO}$	STOP condition setup time	4		0.6		uS
t_{BUF}	Bus free time	4.7[3]		1.2[3]		uS
$t_{SU;DAT}$	Data setup time	250		100		nS
$t_{HD;DAT}$	Data hold time	0[4]	3.45[5]	0[4]	0.8[5]	uS
t_r	SCL/SDA rise time		1000	20+0.1Cb	300	nS
t_f	SCL/SDA fall time		300		300	nS
C_b	Capacitive load for each bus line		400		400	pF

Do Nu-Link Debugging Tools Support 8051 Series Chips

- Nuvoton Nu-Link Debugging Tools include Nu-Link-Me (ICE), Nu-Link-Pro (ICE), and Nu-Link (ICE), and support 1T 8051 chips, such as ML51, MS51, N76E003, N76E616, and N76E885.
- Other older 8051 series are not currently supported and require additional tools for debugging.

Timer Stopped During Single-step In Keil Debug Mode

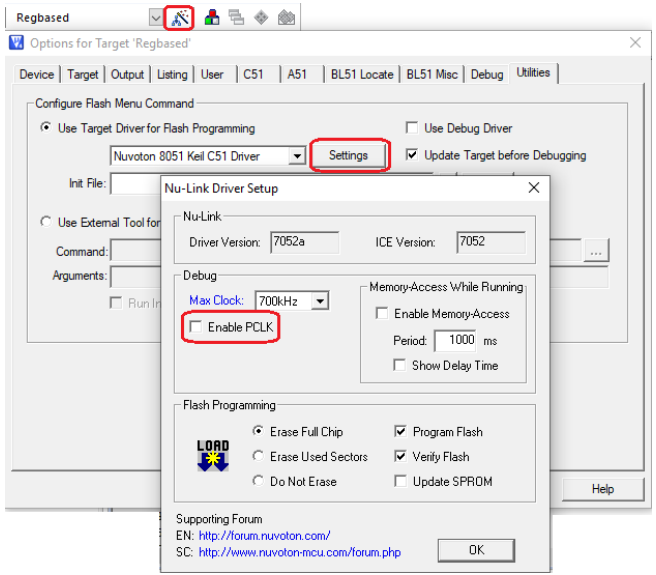
Keil can turn off the peripheral clock in Debug mode to achieve the effect of stopping the Timer in a single-step. However, after this setting is made, the peripherals will have no action during single-step execution such as PWM output.

To suspend the Timer in Debug mode, please follow the steps below:

(1) Click the " Options for target" icon

(2) Click the Settings button

(3) Uncheck "Enable PCLK"



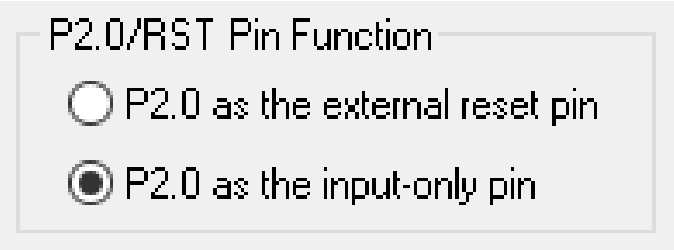
How to configure UART Rx Function if pin has UART Rx and nRESET functions

Take MS51DA9AE for example. After power-on, P2.0 can configure as the external reset pin or an input-only pin. By default, P2.0 is configured as external reset pin.

Only when P2.0 is configured as an input-only pin can user select alternate function. Take UART Rx function for example. Please follow the steps below.

Program RPD (CONFIG0.2) as 0 by ICP tool. The purpose is to configure P2.0 as input pin available.

Remove pull-up resistor from reset circuit. The purpose is to avoid that IO can't drive the signal low by external sources with pull-up resistor.



What are The Different Points Among INT0/1 and PIN Interrupt

- The pin interrupt only support port select, pin interrupt only support port select is a mount of GPIO (Enable interrupts only for ports required by PINENx or PIPENx) share same interrupt handler. Once interrupt event happened, software should check which I/O is the trigger source. Hence, it will spend some time for check (if… else if …) the event source.
- EINT0 and EINT1 is only service for one GPIO and it should not spend time to check the trigger source. Hence it can respond the interrupt event immediately.

How to Estimate The Total ADC Conversion Time

The total ADC conversion time follows the equation below:

$T_{ADC} = T_{SMP} + T_{conv}$

Where

T_{ADC} = total conversion time

T_{SMP} = sampling time

T_{conv} = conversion time = 1.625us

The sampling time follows the equation below:

$T_{SMP} = \frac{4*ADCAQT+6}{F_{ADCAQT}}$ where $F_{ADCAQT} = F_{sys}$

With the condition F_{sys} = 16MHz, $T_{SMP(min)} = \frac{4*0+6}{16MHz}$ = 0.375us

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
T_A	Temperature	-40	-	105	℃	
AV_{DD}	Analog operating voltage	2.7	-	5.5	V	$AV_{DD} = V_{DD}$
V_{REF}	Reference voltage	2.7	-	AV_{DD}	V	$V_{REF} = AV_{DD}$
V_{IN}	ADC channel input voltage	0	-	V_{REF}	V	
I_{ADC}[*1]	Operating current (AV_{DD} + V_{REF} current)	-	-	418	μA	$AV_{DD} = V_{DD} = V_{REF}$ = 5.5 V F_{ADC} = 500 kHz T_{CONV} = 17 * T_{ADC}
N_R	Resolution		12		Bit	
F_{ADC}[1]	ADC conversion rate	-	-	500	kHz	$F_{ADC} = 1/T_{ADC}$ $T_{ADC} = T_{SMP} + T_{CONV}$
T_{SMP}	Sampling Time [2]	0.375	-	2.12	μs	Fsys = 16MHz;
		0.417	-	1.54	μs	Fsys = 24MHz; For Min. ADCAQT = 1 [3]
T_{CONV}	Conversion time	-	-	1.625	μs	
T_{EN}	Enable to ready time	20	-	-	μs	
INL[*1]	Integral Non-Linearity Error	-3	-	+3	LSB	$V_{REF} = AV_{DD} = V_{DD}$
DNL[*1]	Differential Non-Linearity Error	-2	-	+4	LSB	$V_{REF} = AV_{DD} = V_{DD}$
E_G[*1]	Gain error	-3.5	-	+0.4	LSB	$V_{REF} = AV_{DD} = V_{DD}$
E_O[*1]	Offset error	-2	-	+2.8	LSB	$V_{REF} = AV_{DD} = V_{DD}$
E_A[*1]	Absolute Error	-7		+7	LSB	$V_{REF} = AV_{DD} = V_{DD}$

Notes:

1. Guaranteed by characterization result, not tested in production.
2. ADC sampling time = $\frac{4 * ADCAQT + 6}{F_{ADCAQT}}$, F_{ADCAQT} is defined in ADCDIV (ADCCON2[3:1]). As default $F_{ADCAQT} = F_{SYS}$ (ADCDIV=0),
3. Since the minima sampling time must over 370ns that means when F_{ADCAQT} = 24MHz, ADCAQT should be defined as 1 at least. This value is defined by software.

The ADC controller supports continuous conversion mode and interrupt conversion mode, and the specific calculation formula is detailed below.

How to Estimate The Total ADC Conversion Time

1. **Continuous Conversion Mode**

   The total ADC conversion time is $T_{ADC(min)} = T_{SAMP(min)} + T_{CONV} = 0.375us + 1.625us = 2us$

   then the ADC conversion rate is $F_{ADC} = \frac{1}{T_{ADC(min)}} = 500\ kHz$

2. **Interrupt Conversion Mode**

   Setting ADC converting software start trigger bit and waiting for ADC flag to be set requires about 9 cycles, as shown in the figure below.

   The total ADC conversion time is $T_{ADC(min)} = T_{SMP(min)} + T_{CONV} + T_{instruction\ cycle} = 0.375us + 1.625us + 9*\frac{1}{16M}us = 2.5625us$

   then the ADC conversion rate is $F_{ADC} = \frac{1}{T_{ADC(min)}} = 390\ kHz$ Thus, the ADC conversion rate is 390 kHz.

```
Disassembly
C:0x0EFF    7E36      MOV       R6,#0x36
C:0x0F01    7D6E      MOV       R5,#0x6E
C:0x0F03    7C01      MOV       R4,#0x01
C:0x0F05    120703    LCALL     Timer2_Delay(C:0703)
    42:       clr_ADCCON0_ADCF;
C:0x0F08    53E87F    ANL       ADCCON0(0xE8),#0x7F
    43:       set_ADCCON0_ADCS;                  // ADC start trig signal
C:0x0F0B    43E840    ORL       ADCCON0(0xE8),#BYTE_TMP(0x40)
    44:       while(ADCF == 0);
C:0x0F0E    30EFFD    JNB       ADCF(0xE8.7),C:0F0E
    45:       temp=(ADCRH<<4)+(ADCRL&0x0F);
C:0x0F11    AFC3      MOV       R7,ADCRH(0xC3)
C:0x0F13    EF        MOV       A,R7
C:0x0F14    75F010    MOV       B(0xF0),#0x10
C:0x0F17    A4        MUL       AB
```

What Type of RAM in 8 Bit MCU

In 8 Bit MCU only support Code, data, idata, bdata, xdata.

Memory Type	Description
code	Program memory (64 KBytes); accessed by opcode MOVC @A+DPTR.
data	Directly addressable internal data memory; fastest access to variables (128 bytes).
idata	Indirectly addressable internal data memory; accessed across the full internal address space (256 bytes).
bdata	Bit-addressable internal data memory; supports mixed bit and byte access (16 bytes).
xdata	External data memory (64 KBytes); accessed by opcode MOVX @DPTR.

Refer:

http://www.keil.com/support/man/docs/c51/c51_le_memtypes.htm

What is 8bit Timer

8051 timer is up counter, if th0, th1 initial value set 0xfff0, the Timer will count up from 0xfff0->0xfff1->…->0xffff->timer overflow->timer interrupt

Timer Mode 0-3:

- **Mode 0: 8192-(THx(8bit), TLx(5bit)) * timer clock source**
- **Mode 1: 65535-(THx(8bit), TLx(8bit)) * timer clock source**
- **Mode 2: 256-(THx(8bit)) * timer clock source**
- **Mode 3: 256-(THx(8bit)) * timer clock source**

M1	M0	Mode	Operation
0	0	0 (13-bit timer mode)	13-bit timer/counter, 8-bit of THx & 5-bit of TLx
0	1	1 (16-bit timer mode)	16-bit timer/counter, THx cascaded with TLx
1	0	2 (8-bit auto reload mode)	8-bit timer/counter (auto reload mode), TLx reload with value held by THx each time TLx overflow
1	1	3 (split timer mode)	Split 16-bit timerx into two 8-bit timer i.e. THx and TLx like two 8-bit timer

How to Initial RAM to 0

In “STATUP.A51”

The code in the following red frame of "STARTUP.A51" recognized that the memory was cleared to 0.

STARTUP.A51

```
107
108  IF IDATALEN <> 0
109  ·····MOV·····R0,#IDATALEN - 1
110  ·····CLR·····A
111  IDATALOOP:···MOV·····@R0,A
112  ·····DJNZ····R0,IDATALOOP
113  ENDIF
114
115  IF XDATALEN <> 0
116  ·····MOV·····DPTR,#XDATASTART
117  ·····MOV·····R7,#LOW (XDATALEN)
118  ··IF (LOW (XDATALEN)) <> 0
119  ·····MOV·····R6,#(HIGH (XDATALEN)) +1
120  ··ELSE
121  ·····MOV·····R6,#HIGH (XDATALEN)
122  ··ENDIF
123  ·····CLR·····A
124  XDATALOOP:···MOVX····@DPTR,A
125  ·····INC·····DPTR
126  ·····DJNZ····R7,XDATALOOP
127  ·····DJNZ····R6,XDATALOOP
128  ENDIF
```

How to Initial RAM to 0

In the above assembler, the clear area of XDATA is specified from 0[H] to XDATALEN (2FF [H] = 767).

STARTUP.A51

```
13  ;  BL51 invocation:
14  ;
15  ;     BL51 <your object file list>, STARTUP.OBJ <controls>
16  ;
17  ;------------------------------------------------------------------------------
18  ;
19  ;  User-defined Power-On Initialization of Memory
20  ;
21  ;  With the following EQU statements the initialization of memory
22  ;  at processor reset can be defined:
23  ;
24  ;               ; the absolute start-address of IDATA memory is always 0
25  IDATALEN        EQU     80H     ; the length of IDATA memory in bytes.
26  ;
27  XDATASTART      EQU     0H      ; the absolute start-address of XDATA memory
28  XDATALEN        EQU     0FFFH   ; the length of XDATA memory in bytes.
29  ;
30  PDATASTART      EQU     0H      ; the absolute start-address of PDATA memory
31  PDATALEN        EQU     0H      ; the length of PDATA memory in bytes.
32  ;
33  ;  Notes:  The IDATA space overlaps physically the DATA and BIT areas of the
34  ;          8051 CPU. At minimum the memory space occupied from the C51
35  ;          run-time routines must be set to zero.
36  ;------------------------------------------------------------------------------
37  ;
```

How to Keep RAM Value in Software Reset

A, "STARTUP.A51" set xdatalen to "0x3ff"=>Change it to match the MCU XRAM size, In MCU power reset or software reset, all XRAM is initial to 0.

B, "STARTUP.A51" set xdatalen to "0x2ff"=>the revise XRAM(0x300-0x3ff) no initial to 0 and keep the XRAM context value (0x300-0x3ff) in software reset.

User can use _at_ (KEIL define) key word to access the memory address in user define.

Refer:

http://www.keil.com/support/man/docs/c51/c51_ap_at.htm

How to Set Default Variables to External RAM

Set Memory mode in “large: Variables in XRAM”