Application note

GCC development environment based on Windows Application Note

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

Contens

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

1. Overview

Taking N32G430 series MCU as an example, this paper introduces the methods of setting up development environment, compiling, firmware downloading and code debugging based on VScode editor, GCC compilation tool chain and GDB debugging tool under Windows environment.

2. Development Tools

2.1 Software

1) Editor Visual Studio Code 1.5x.x or above
2) Compile toolchain arm-none-eabi-gcc 6.3.1 or above
3) Make for Windows
4) Download and debugging tool JLink_v6.40(need to be no higher than the hardware support version) or above

2.2 Hardware

1) Development board N32G430C8L7-STB V1.0
2) JLink Downloader V9.2(need to be no lower than the software support version) or above

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

3. Development Environment Setup

3.1 Installing VScode

- **Download the software: https://code.visualstudio.com/**

VScode is used for code viewing and editing, and it also provides powershell and bash terminals for command-line operations, which will be used throughout our development process.

3.2 Installing the GCC Compilation tool chain

- **Download address:**
  https://launchpad.net/gcc-arm-embedded/+announcement/28093
  example version: 10-2020-q4-major

Check whether the installation is successful: Open the DOS command line window, type arm-none-eabi-gcc -v,
The installation is successful if:

```
C:\Users\tan.dengwang>arm-none-eabi-gcc --version
arm-none-eabi-gcc (GNU Arm Embedded Toolchain 10-2020-q4-major) 10.2.1 20201103
(release)
Copyright (C) 2020 Free Software Foundation, Inc.
```

If you don't succeed
1. Check whether environment variables are properly added
2. Go to "*C:\Program Files (x86)\GNU Arm Embedded Toolchain\10-2020-q4-major\bin* " and check whether the arm-none-eabi-gcc.exe file name is correct

3.3 Installing Make for Windows

This tool is used to parse Makefile scripts and can be installed with either of the following software.

- **Install the cmake.exe tool**
  Download address: http://www.equation.com/servlet/equation.cmd?fa=make
- **Install MinGW software and use its own make tool.**

Check whether the installation is successful: Open the DOS command line window and enter make -v as follows:

```
C:\Users\tan.dengwang>make -v
GNU Make 3.82.90
Built for i686-pc-mingw32
Copyright (C) 1988-2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

If you don't succeed

1，Check that the environment variables are properly added
2，Go to the bin folder of the corresponding make installation directory to check whether the

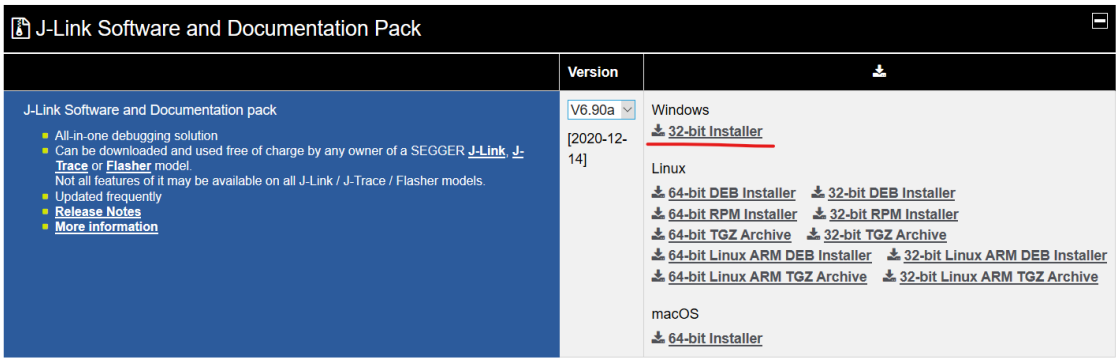
NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

make. exe file is correctly named

3.4 Installing the JLink Tool

- **Download the JLINK installation package, V6.90a or others version**

https://www.segger.com/downloads/jlink/#-LinkSoftwareAndDocumentationPack



3.5 Adding Chip Support

After installing JLink, we need to add our company's chip patch package to JLink, so that we can get the download algorithm correctly during downloading and debugging.
For details, see <jlink Tool Adding Nations Chip.7z>.

3.6 JLink download test

- **Test the JLink environment installation**
  1，Connect the PC and j-Link debugger, connect the development board, and power on;
  2，Open cmd.exe command line tool, go to JLink installation directory C:\Program Files (x86)\SEGGER\JLink_V640, type jlink.exe.

```
C:\Program Files (x86)\SEGGER\JLink_V640>jlink.exe
SEGGER J-Link Commander V6.40 (Compiled Oct 26 2018 15:06:29)
DLL version V6.40, compiled Oct 26 2018 15:06:02

Connecting to J-Link via USB...O.K.
Firmware: J-Link V9 compiled Dec 13 2019 11:14:50
Hardware version: V9.60
S/N: 69660532
License(s): RDI, GDB, FlashDL, FlashBP, JFlash
VTref=3.316V


Type "connect" to establish a target connection, '?' for help
J-Link>
```

The image above shows that the PC successfully connected to the JLink debugger.
3，Then according to the prompt input: “connect”, “N32G430C8”, “SWD”, “4000”, if the

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

previous operation is successful, you will see the following output information, JLink download debugging environment can be used normally.

```
Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: N32G030C8
Type '?' for selection dialog
Device>N32G430C8
Please specify target interface:
  J) JTAG (Default)
  S) SWD
  T) cJTAG
TIF>S
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>
Device "N32G430C8" selected.


Connecting to target via SWD
Found SW-DP with ID 0x2BA01477
Scanning AP map to find all available APs
AP[1]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x24770011)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, Little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
ROMTbl[0] @ E00FF000
ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB00C SCS-M7
ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
ROMTbl[0][3]: E0000000, CID: B105E00D, PID: 003BB001 ITM
ROMTbl[0][4]: E0040000, CID: B105900D, PID: 000BB9A1 TPIU
Cortex-M4 identified.
J-Link>
```

4. SDK Contents

SDK follows the issued SDK version, currently using V1.0.0, on this basis to make the following modifications to adapt to GCC development environment.

4.1 Makefile

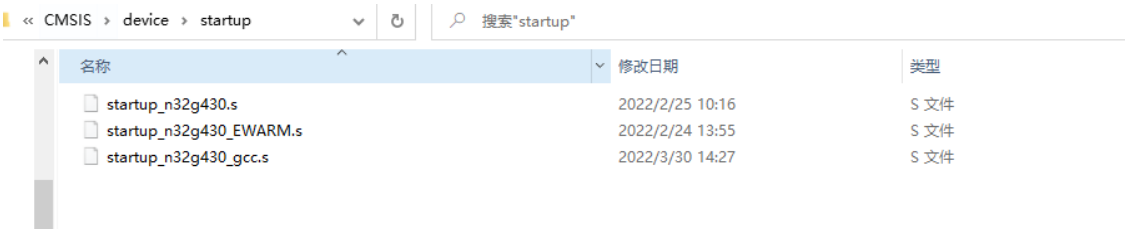
Added "GCC" folder under module routines directory in SDK package :(please copy "GCC" folder to each routine)

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

The "Makefile" file is the GCC compilation script file.

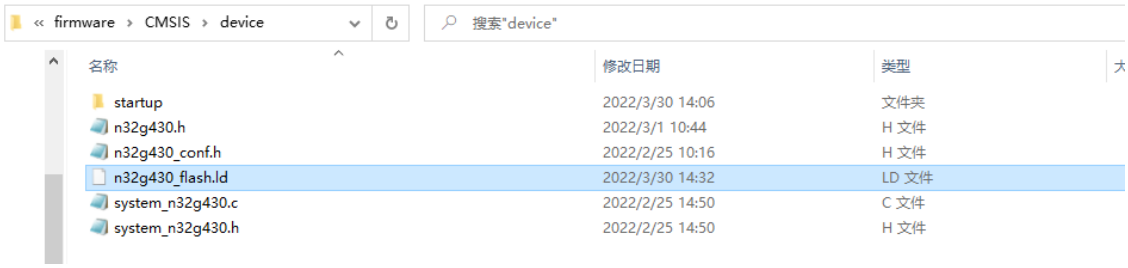
4.2 .s File

In the SDK package "Nations.n32g430_Library.1.0.0 \firmware\CMSIS\device\ startup" there is a GCC compiler .S file “startup_n32g430_gcc.s” in the corresponding path.



4.3 .ld File

In the SDK package, "Nations.N32G430_Library.1.0.0\firmware\CMSIS\ device" there is a .ld file "n32g430_flash.ld" in the corresponding path.



4.4 Printing Remapping

The “print_remap.c” file is added in the “bsp/src” directory of the SDK package for serial port printing remapping.

4.5 J-Link Script

Added the jlink folder in the SDK home directory, which contains a Jlink download script for

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

downloading firmware using the J-Link tool.

4.6 Clearing Scripts

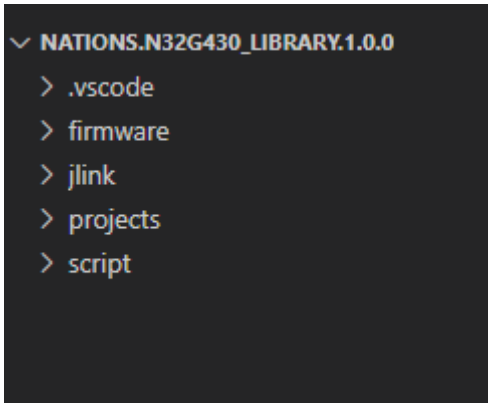
The “script” folder is added in the SDK package home directory, and there is a .bat script in the folder, which is used to clear intermediate files generated during compilation.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

5. Compile and download

5.1 Workspace

Open the SDK folder in VScode and save it as a workspace.At this point, the ".vscode "folder will be generated under the SDK folder to place the workspace configuration file.



5.2 Working Directory

Take the GPIO routine LedBlink as an example to enter the project directory:
"Nations. N32G430_Library. 1.0.0 \ projects \ n32g430_EVAL \ examples \ GPIO \ LedBlink"
KEIL project "MDK - ARM"
GCC project "GCC"
Project source file "src /xxx.c"
Project header file "inc/XXX.h"
Makefile file "GCC/Makefile"

5.3 Code Compilation

In the terminal of the VScode editor, switch to the "GCC" folder directory and type "make" to start compiling

```
PS E:\workspace_linqi\3605\GCC\Nations.N32G430_Library.1.0.0\projects\n32g430_EVAL\examples\GPIO\LedBlink\GCC> make
```

And the .elf, .bin and .hex files are generated when compiled error-free

```
/output.map -T../../../../../../firmware/CMSIS/device/n32g430_flash.ld -o build/output.elf
arm-none-eabi-size build/output.elf
   text    data     bss     dec     hex filename
   2264    1088    1572    4924    133c build/output.elf
arm-none-eabi-objcopy -O ihex -S build/output.elf build/output.hex
arm-none-eabi-objcopy -O binary -S build/output.elf build/output.bin
```

In this case, the "build" folder is created under the "GCC" folder. The compiled firmware and intermediate files are stored in this folder.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

5.4 Downloading Firmware

1. Connect PC->JLink->development board
2. On the terminal, type “make download”

```
PS E:\workspace_linqi\3605\GCC\Nations.N32G430_Library.1.0.0\projects\n32g430_EVAL\examples\GPIO\LedBlink\GCC> make download
```

Some information will be printed in the process...Finally, the download is complete

```
Writing target memory failed.
J-Link>r
Reset delay: 0 ms
Reset type NORMAL: Resets core & peripherals via SYSRESETREQ & VECTRESET bit.
Reset: Halt core after reset via DEMCR.VC_CORERESET.
Reset: Reset device via AIRCR.SYSRESETREQ.
J-Link>g
J-Link>qc

Script processing completed.

"Download Completed!"
```

3. After downloading, the system will automatically reset and start running
4. If the download fails, check the JLink configuration

5.5 Clearing Intermediate Files

Type "make clean" on the terminal to clear the intermediate files generated by the compilation.

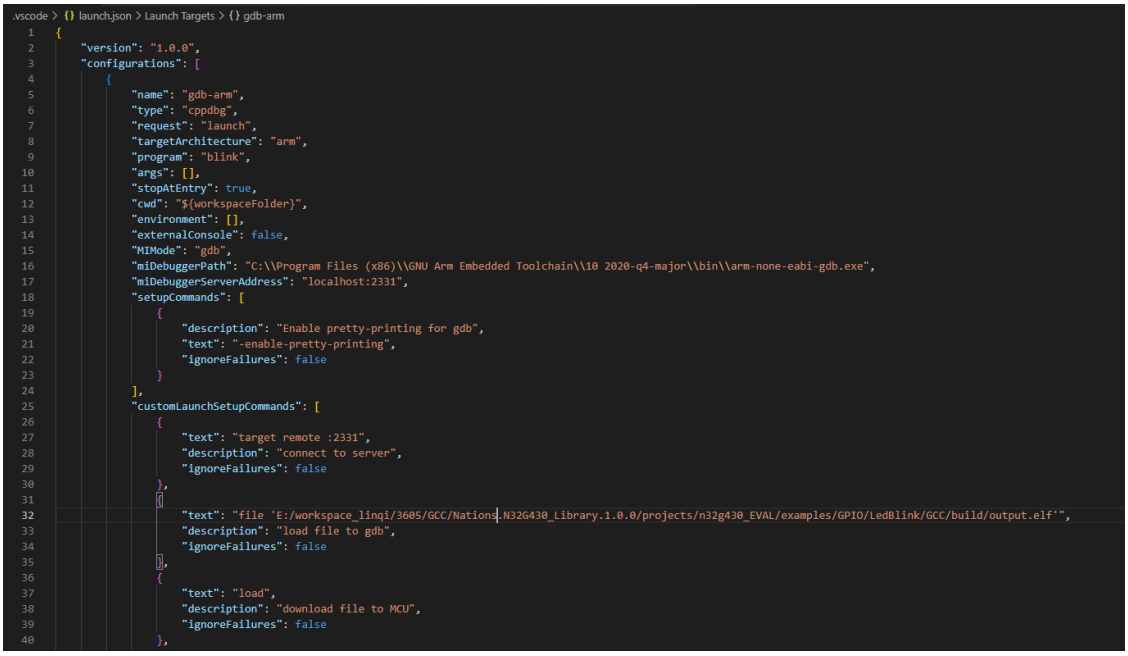
NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

6. Code debugging

6.1 VSCode set

There is a ".vscode "folder in the SDK working path, which contains “launch.json” workspace configuration files that need to be configured for code debugging:

launch. json:



```
.vscode > {} launch.json > Launch Targets > {} gdb-arm
{
    "version": "1.0.0",
    "configurations": [
        {
            "name": "gdb-arm",
            "type": "cppdbg",
            "request": "launch",
            "targetArchitecture": "arm",
            "program": "blink",
            "args": [],
            "stopAtEntry": true,
            "cwd": "${workspaceFolder}",
            "environment": [],
            "externalConsole": false,
            "MIMode": "gdb",
            "miDebuggerPath": "C:\\Program Files (x86)\\GNU Arm Embedded Toolchain\\10 2020-q4-major\\bin\\arm-none-eabi-gdb.exe",
            "miDebuggerServerAddress": "localhost:2331",
            "setupCommands": [
                {
                    "description": "Enable pretty-printing for gdb",
                    "text": "-enable-pretty-printing",
                    "ignoreFailures": false
                }
            ],
            "customLaunchSetupCommands": [
                {
                    "text": "target remote :2331",
                    "description": "connect to server",
                    "ignoreFailures": false
                },
                {
                    "text": "file 'E:/workspace_linqi/3605/GCC/Nations.N32G430_Library.1.0.0/projects/n32g430_EVAL/examples/GPIO/LedBlink/GCC/build/output.elf'",
                    "description": "load file to gdb",
                    "ignoreFailures": false
                },
                {
                    "text": "load",
                    "description": "download file to MCU",
                    "ignoreFailures": false
                },
                {
                    "text": "monitor reset",
                    "description": "reset MCU",
                    "ignoreFailures": false
                },
                {
                    "text": "b main",
                    "description": "set breakpoints at main",
                    "ignoreFailures": false
                },
            ],
            "launchCompleteCommand": "None",
            //"preLaunchTask": "build"
        }
    ]
}
```

This is the vscode debugger configuration file, and the following changes should be made according to your project path:

1, specify the path to the GDB debugger :(absolute path)

```
"miDebuggerPath": "C:\\Program Files (x86)\\GNU Arm Embedded Toolchain\\10-2020-q4-major\\bin\\arm-none-eabi-gdb.exe",
```

The version of the GDB tool must match the version of the compiler tool. Otherwise, errors will be reported or some functions will be unavailable. The arm-none-eabi-gdb.exe tool is usually in the

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

same directory as the arm-none-eabi-gcc.exe tool.
2, specify debug code xxx.elf file path: (Note: path cannot be too long)

```
32              "text": "file 'E:/workspace_linqi/3605/GCC/Nations.N32G430_Library.1.0.0/projects/n32g430_EVAL/examples/GPIO/LedBlink/GCC/build/output.elf'",
```

6.2 Makefile Settings

Open the routine "GCC/Makefile" file:

```
download:
    @$(JK_DPATH)JLink.exe -device $(CHIP_TYPE) -if SWD -speed 4000 -autoconnect 1 -CommanderScript $(JKS_DIR)/flash.jlink
    @echo "Download Completed!"

debug:
    @$(JK_DPATH)JLinkGDBServer.exe -select USB -device $(CHIP_TYPE) -if SWD -speed auto -noir -LocalhostOnly

# *** EOF ***
```

1, you can see that there is a debug startup configuration pointing to the JLinkGDBserver server in the JLink installation directory.
2. The make command is in debug mode by default, with some debugging information.If you want to switch to the release version, compile the code with the following command: make Release =y

6.3 Debugging Examples

Using the GPIO LedBlink project as an example, see how to start code debugging:
1. Open SDK project in vscode, switch to LedLink/GCC directory in terminal, and type make to compile code

```
PS E:\workspace_linqi\3605\GCC\Nations.N32G430_Library.1.0.0\projects\n32g430_EVAL\examples\GPIO\LedBlink\GCC> make
ild/n32g030_lpuart.o build/n32g030_opamp.o build/n32g030_pwr.o build/n32g030_rcc
030_wwdg.o build/startup_n32g030_gcc.o -mcpu=cortex-m0 -mthumb   -Wl,--gc-sectio
 build/output.elf
arm-none-eabi-size build/output.elf
   text    data     bss     dec     hex filename
   1508    1080    1572    4160    1040 build/output.elf
arm-none-eabi-objcopy -O ihex -S build/output.elf build/output.hex
arm-none-eabi-objcopy -O binary -S build/output.elf build/output.bin
```

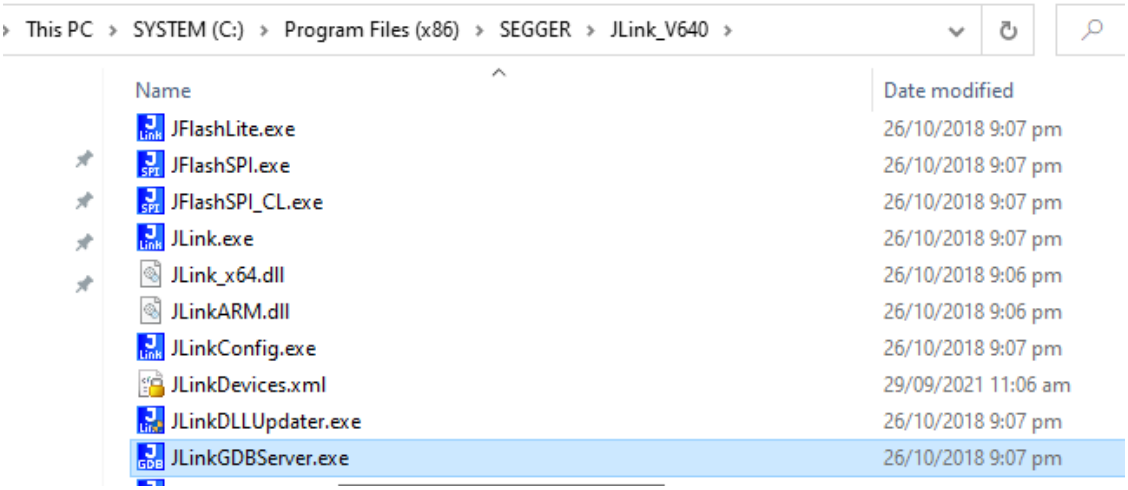
output.elf, output.bin, output.hex files are generated in GCC/build folder.

2. Refer to 6.1 and 6.2 section to configure the path in the launch.json files.

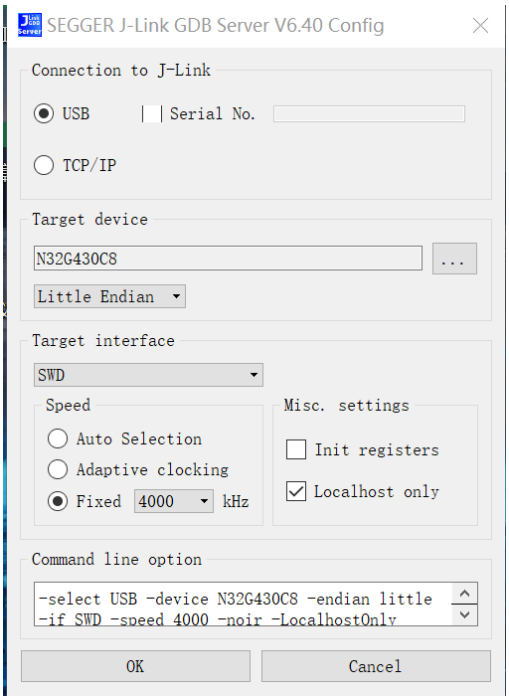
3, connect the JLink debugger to the development board, power on and prepare.

4, Go to your JLink installation directory and double-click JlinkGDBServer.exe

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

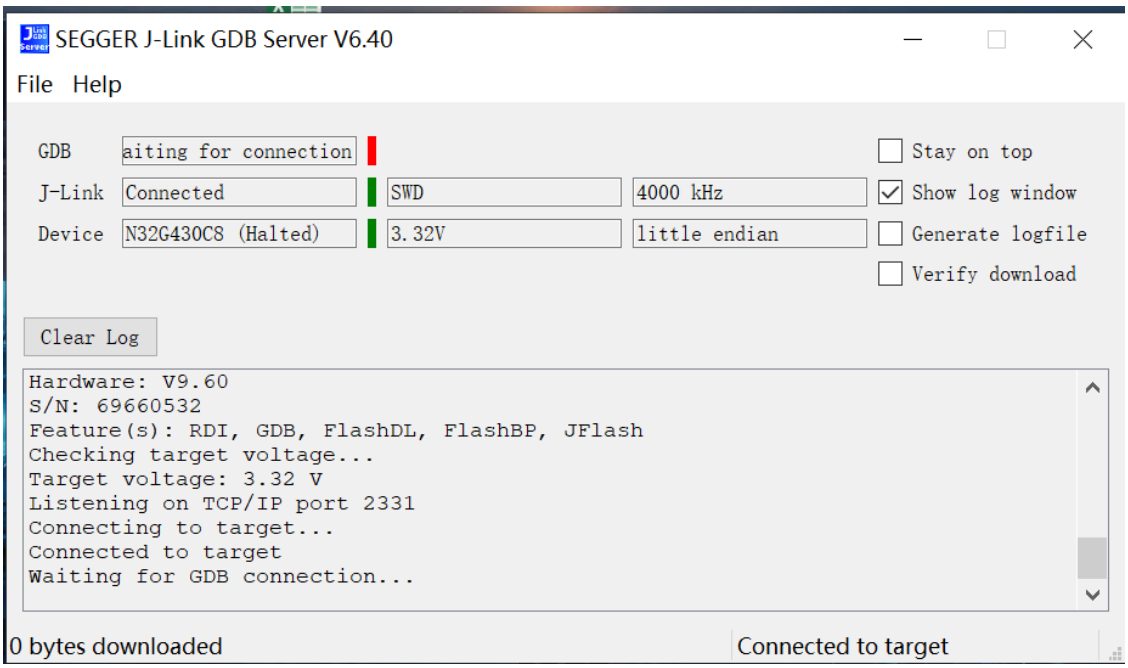


To configure ports, protocols, and chip models, click OK

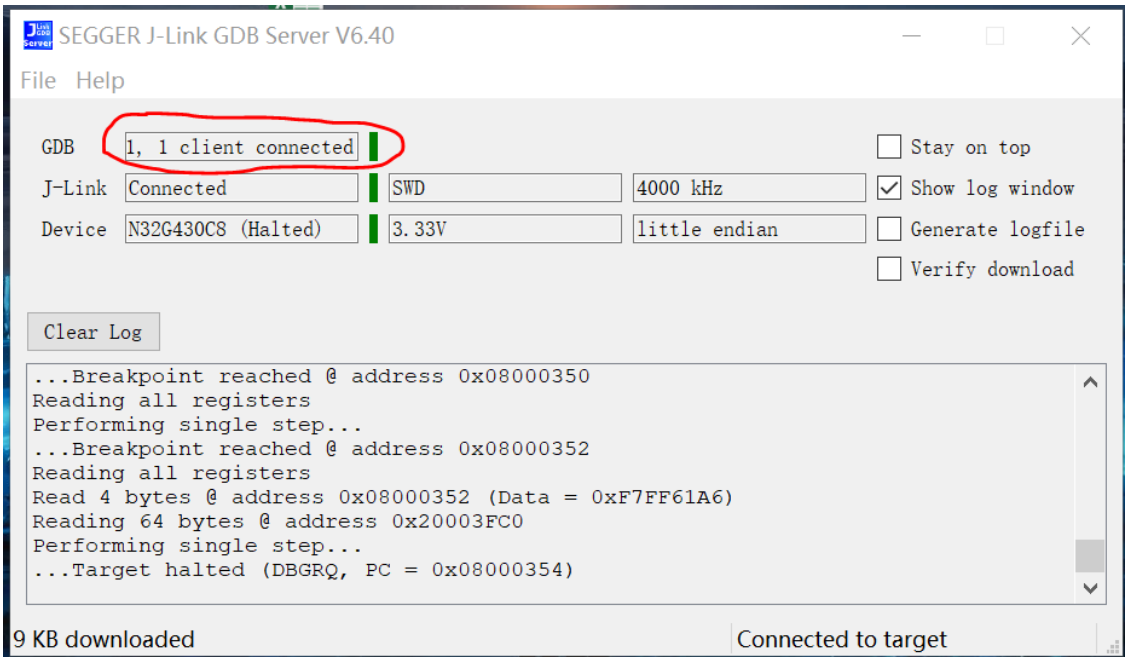


If the JLink debugger is successfully connected to the chip:

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

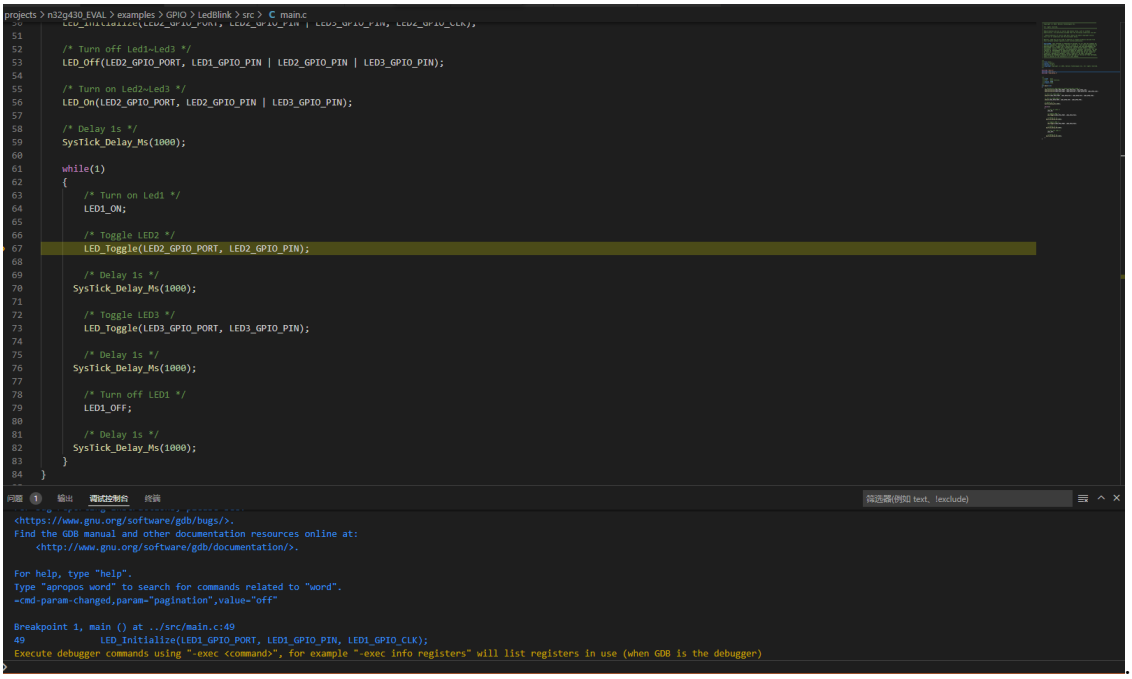


5. Under vscode working environment, press "F5" or click "Run" -> "Start debugging". At this time, it can be seen that the label below turns green, indicating that gdb tool successfully connects to JLinkGDBserver.

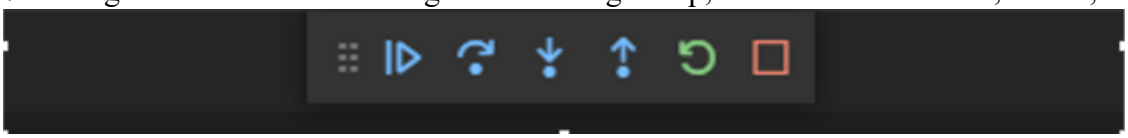


6, vscode automatically switches to the debug window

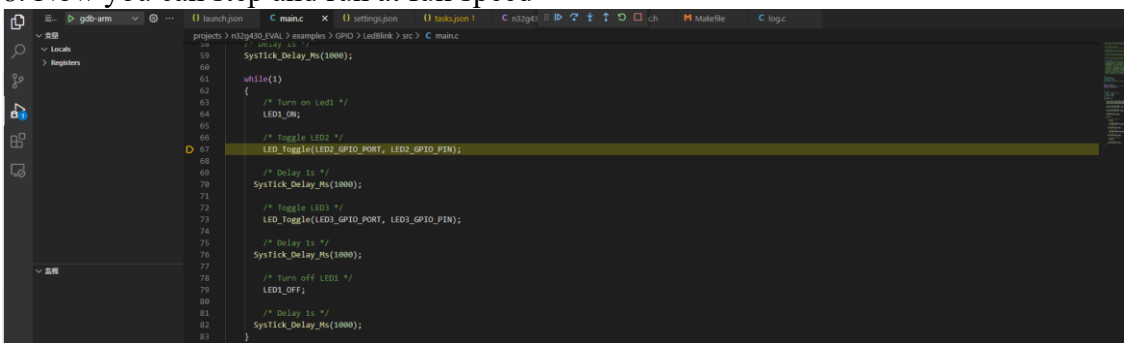
NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg



7. Debug buttons above the debug window: single step, continuous execution, restart, stop, etc



8. Now you can step and run at full speed



NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

7. Configuration Changes

7.1 Chip Models

If you are using chips other than the N32G430 family, you need to modify the variables "TARGET_PLATFORM" and "DEFS" in the makefile.

```
########################################
# chip platform info
########################################
TARGET_PLATFORM := n32g430
DEFS += -DN32G430
DEFS += -DUSE_STDPERIPH_DRIVER
```

7.2 Firmware Download Algorithm

You need to type the full chip model so that JLink can properly match the download algorithm.

```
#Chip type
CHIP_TYPE = N32G430C8
```

Configure the path to download the tool: configure it according to your installation directory

```
########################################
# download .hex/.bin by jlink
########################################
#Your JLink installation directory
PATH_WINPC = 'C:/Program Files (x86)/SEGGER/JLink_V640/'
#PATH_LINUX = /opt/SEGGER/JLink_V640b/JLinkExe
JK_DPATH = $(PATH_WINPC)
```

7.3 Using the SDK algorithm library

By default, the library is not used. Please modify the variable USELIB = 1 to use the library.

```
36  ########################################
37  # Algo libs
38  ########################################
39  USELIB = 0
```

7.4 DEBUG configuration

The default "make" compilation is with "-g" debugging information. If you want to build a release

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

version, use "make release =y".

7.5 Optimization Grade

The default optimization level is "-Os", which takes into account both code size and execution speed.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

8. Version History

Version	Date	Changes
V1.0	2022/03/30	The initial release

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg

9. Disclaimer

This document is the exclusive property of NSING TECHNOLOGIES PTE. LTD.(Hereinafter referred to as NSING). This document, and the product of NSING described herein (Hereinafter referred to as the Product) are owned by NSING under the laws and treaties of Republic of Singapore and other applicable jurisdictions worldwide. The intellectual properties of the product belong to Nations Technologies Inc. and Nations Technologies Inc. does not grant any third party any license under its patents, copyrights, trademarks, or other intellectual property rights. Names and brands of third party may be mentioned or referred thereto (if any) for identification purposes only. NSING reserves the right to make changes, corrections. enhancements, modifications, and improvements to this document at any time without notice. Please contact NSING and obtain the latest version of this document before placing orders. Although NATIONS has attempted to provide accurate and reliable information, NATIONS assumes no responsibility for the accuracy and reliability of this document. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. In no event shall NATIONS be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this document or the Product.

NATIONS Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, Insecure Usage'. Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, all types of safety devices, and other applications intended to supporter sustain life. All Insecure Usage shall be made at user's risk. User shall indemnify NATIONS and hold NATIONS harmless from and against all claims, costs, damages, and other liabilities, arising from or related to any customer's Insecure Usage Any express or implied warranty with regard to this document or the Product, including, but not limited to. The warranties of merchantability, fitness for a particular purpose and non-infringement are disclaimed to the fullest extent permitted by law. Unless otherwise explicitly permitted by NATIONS, anyone may not use, duplicate, modify, transcribe or otherwise distribute this document for any purposes, in whole or in part.

NSING Technologies Pte. Ltd.
Add: NSING, Teletech Park #02-28, 20 Science Park Road,
Singapore 117674
Tel: +65 69268090
Email: sales@nsing.com.sg