UM2495

User manual

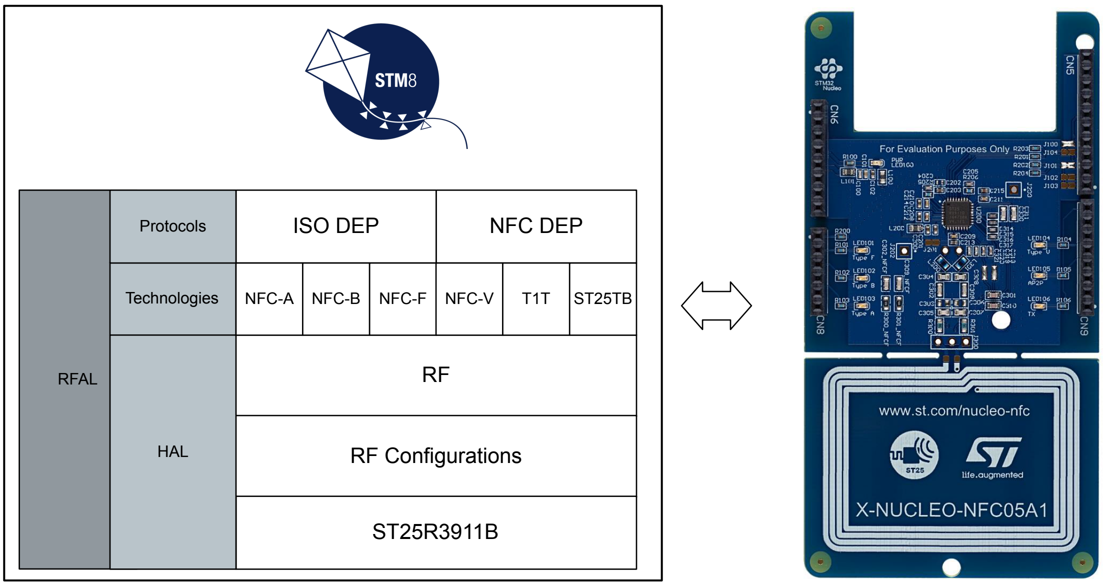
High performance HF reader / NFC initiator IC software expansion for NUCLEO-8S208RB

Introduction

The high performance HF reader / NFC initiator IC software expansion (STSW-STM8-NFC5) enables the NUCLEO-8S208RB board to operate with the X-NUCLEO-NFC05A1.

This package ports the RF abstraction layer (RFAL) onto the STM8 device to operate with the X-NUCLEO-NFC05A1, which contains the ST25R3911B high performance HF reader / NFC initiator. The package contains a sample application that detects different types of NFC tags and mobile phones supporting P2P.

The RFAL is the STMicroelectronics standard driver for ST25R NFC/RFID reader ICs. It is used, for instance, by the ST25R3911B-DISCO (STSW-ST25R002) and X-NUCLEO-NFC05A1 (X-CUBE-NFC5) firmware. STSW-STM8-NFC5 supports all the ST25R3911B lower-layer protocols and also some higher layer protocols to abstract RF communication. As the RFAL is written in a portable manner, it can run on a wide range of devices, from 8-bit MCUs up to 64-bit processors running Linux. All the code is highly portable and only minimal effort is needed in order for it to run on any STM8 platform.



1 Overview

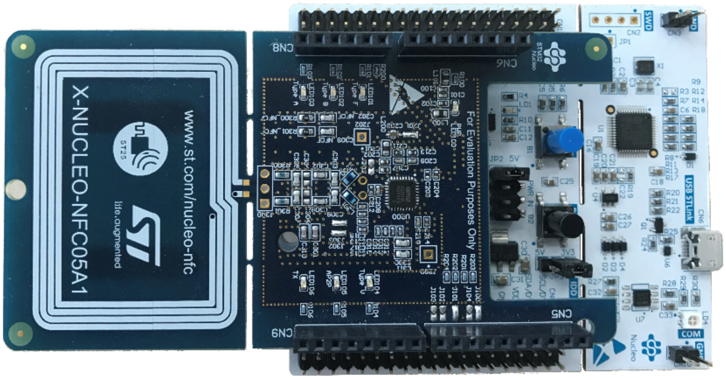
1.1 STSW-STM8-NFC5 features

- Complete driver (RF abstraction layer) to build NFC-enabled applications using the ST25R3911B or ST25R391x high performance HF reader / NFC initiator with up to 1.4 W output power
- Communication with the ST25R3911B or ST25R391x high performance NFC HF reader / NFC initiator using SPI interface
- Complete RF/NFC abstraction (RFAL) for all major technologies and higher layer protocols:
  - NFC-A (ISO14443-A)
  - NFC-B (ISO14443-B)
  - NFC-F (FeliCa)
  - NFC-V (ISO15693)
  - P2P (ISO18092)
  - ISO-DEP (ISO data exchange protocol, ISO14443-4)
  - NFC-DEP (NFC data exchange protocol, ISO18092)
  - Proprietary technologies (Kovio, B’, iClass, Calypso, among others)
- Sample application to detect several NFC tag types and mobile phones supporting P2P
- Free user-friendly license terms

1.2 Hardware setup

The setup is composed by the NUCLEO-8S208RB STM8 Nucleo-64 board together with the X-NUCLEO-NFC05A1 expansion board which supports the ST25R3911B device.

Figure 1. X-NUCLEO-NFC05A1 expansion board



1.3 Sample application

A sample application using the X-NUCLEO-NFC05A1 expansion board with the NUCLEO-8S208RB development board is provided in the *Projects* directory. A ready-to-build project is available for ST Visual Development IDE. On this application, NFC tags of different types or mobile phones supporting P2P are detected by the ST25R3911B high performance HF reader / NFC initiator IC.

After system initialization and clock configuration, LED101, LED102, LED103, LED104 and LED105 blink briefly if initialization is successful. Then the demo starts polling for nearby tags/devices.

When a tag/device is detected, a LED is lit on the X-NUCLEO-NFC05A1 shield according to the table below:

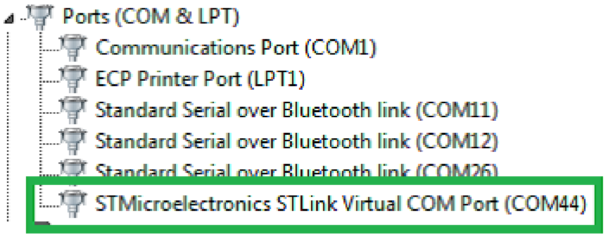
Table 1. **X-NUCLEO-NFC05A1 LED lit by detected device**

NFC technology	LED lit on tag detection
NFC-F	LED101 / Type F
NFC-B	LED102 / Type B
NFC-A	LED103 / Type A
NFC-V	LED104 / Type V
Active P2P	LED105 / Type AP2P

Additionally the user can enable the Wakeup mode by pressing the USER (blue) button on the NUCLEO-8S208RB. When pressed, the ST25R3911B is placed into the low-power wake-up mode on which it monitors the antenna's surroundings for any approaching tag/device and informs the MCU of that variation. The MCU reacts by triggering another poll cycle as described above, and then the ST25R3911B returns to the low-power wake-up mode.

When the USB cable is connected to a PC, a virtual COM port is listed as shown in the figure below.

Figure 2. **Example of virtual COM listed when USB cable is connected to a PC**



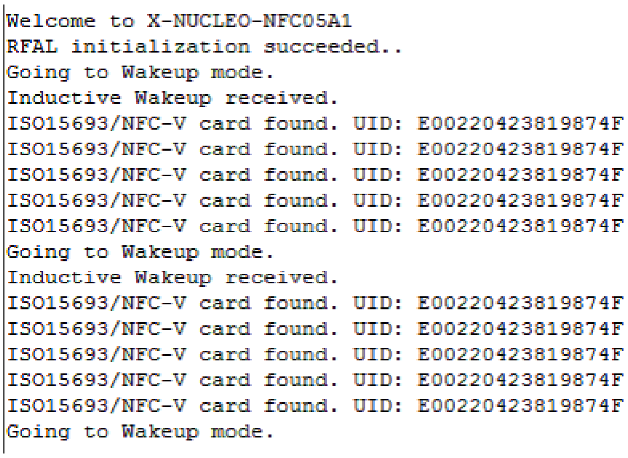
Ports (COM & LPT)
Communications Port (COM1)
ECP Printer Port (LPT1)
Standard Serial over Bluetooth link (COM11)
Standard Serial over Bluetooth link (COM12)
Standard Serial over Bluetooth link (COM26)
STMicroelectronics STLink Virtual COM Port (COM44)

After checking the virtual COM port number, the user has to open a connection on Hyperterminal (or similar) with the following configuration:

- Baud: 115200
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: no

Following a successful connection, the user can view the messages on the Hyperterminal, as shown below.

Figure 3. Hyperterminal successful connection message



```
Welcome to X-NUCLEO-NFC05A1
RFAL initialization succeeded..
Going to Wakeup mode.
Inductive Wakeup received.
ISO15693/NFC-V card found. UID: E00220423819874F
ISO15693/NFC-V card found. UID: E00220423819874F
ISO15693/NFC-V card found. UID: E00220423819874F
ISO15693/NFC-V card found. UID: E00220423819874F
ISO15693/NFC-V card found. UID: E00220423819874F
Going to Wakeup mode.
Inductive Wakeup received.
ISO15693/NFC-V card found. UID: E00220423819874F
ISO15693/NFC-V card found. UID: E00220423819874F
ISO15693/NFC-V card found. UID: E00220423819874F
ISO15693/NFC-V card found. UID: E00220423819874F
ISO15693/NFC-V card found. UID: E00220423819874F
Going to Wakeup mode.
```

2 System setup guide

This section describes how to setup the different components before writing and executing an application on the STM8 Nucleo board with the X-NUCLEO-NFC05A1 expansion board.

2.1 Hardware description

This section describes the hardware components required for the setup described in this user manual.

2.1.1 X-NUCLEO-NFC05A1 expansion board

The X-NUCLEO-NFC05A1 is a high performance HF reader / NFC initiator expansion board usable with the STM8 Nucleo boards. It is also compatible with Arduino™ UNO R3 connector layout, and is designed around the STMicroelectronics IC ST25R3911B high performance HF reader / NFC initiator (with 1.4 W supporting VHBR and AAT). The X-NUCLEO-NFC05A1 interfaces with the STM8 MCU via SPI. Information regarding the X-NUCLEO-NFC05A1 expansion board is available on *www.st.com*.

2.2 Software description

The following software components are needed in order to setup the suitable development environment for creating applications for the STM8 Nucleo equipped with the NFC expansion board:

- ST Visual Develop. Refer to STVD-STM8 on *www.st.com*
- Cosmic compiler CXSTM8 for STM8 (COS-C-COMPILER). Refer to the *Cosmic STM8 Cross Development Tools* section on www.cosmic-software.com.

3 Hardware and software setup

This section describes the hardware and software setup procedures as well as the required system setup.

3.1 System setup guide

This section describes how to setup the different components before writing and executing an application on the STM8 Nucleo board with the X-NUCLEO-NFC05A1 expansion board.

3.1.1 STM8 Nucleo and X-NUCLEO-NFC05A1 expansion board setup

The STM8 Nucleo board integrates the ST-LINK/V2-1 debugger/programmer. The user can download the ST-LINK/V2-1 USB driver by searching STSW-LINK009 software on *www.st.com*.

The X-NUCLEO-NFC05A1 expansion board is easily plugged onto the STM8 Nucleo development board through the Arduino™ UNO R3 extension connector. It interfaces with the STM8 microcontroller on the STM8 Nucleo board through the SPI interface.

3.1.2 Hardware required for system setup

The following hardware components are required:

- One STM8S Nucleo development platform (order code: NUCLEO-8S208RB)
- One ST25R3911B high performance HF reader / NFC initiator IC expansion board (order code: X-NUCLEO-NFC05A1)
- One USB type A to Mini-B USB cable to connect the STM8 Nucleo to the PC

3.1.3 Software required for system setup

This section lists the minimum requirements for the developer to set up the SDK and how to use the sample project.

3.1.3.1 Development toolchains and compilers

- Download and install STVD-STM8
- Download and install Cosmic STM8 compiler CXSTM8
- Setup ST Visual Develop to use the Cosmic compiler

3.1.3.2 Using the project

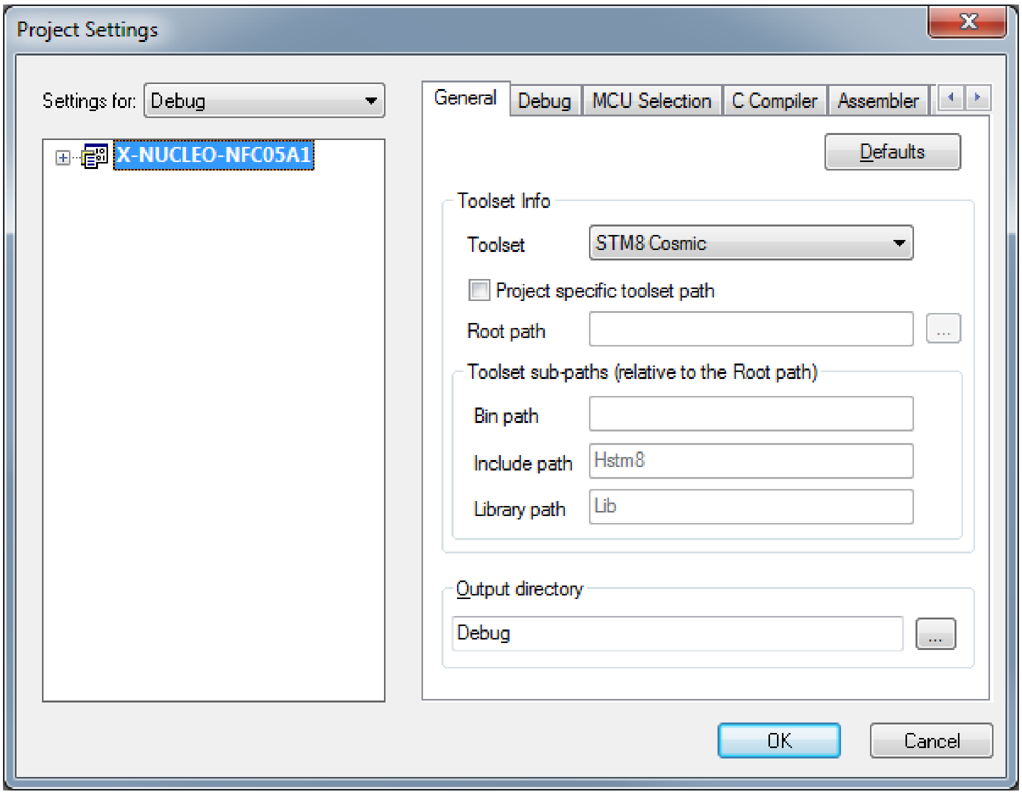
This section describes how to compile the sample application and then how to program it on the board.

Compiling

After installling all the tools, follow the steps below:

1. Open ST Visual Develop IDE
2. Go to *File* → *Open Workspace* and select the file *Project.stw*
3. Go to *Project* → *Settings* and ensure to select STM8 Cosmic on the Toolset Info

Figure 4. Project Settings window



Settings for: Debug

X-NUCLEO-NFC05A1

Toolset: STM8 Cosmic

□ Project specific toolset path

Root path:

Bin path:

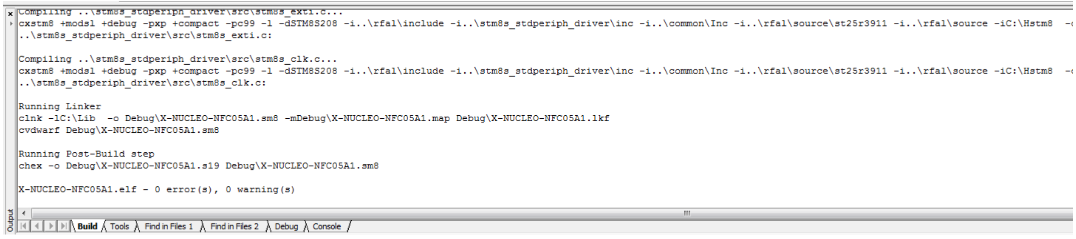
Include path: Hstm8

Library path: Lib

Output directory: Debug

4. To compile the project go to *Build → Rebuild All*

Figure 5. "Rebuild All" example at the compiling step

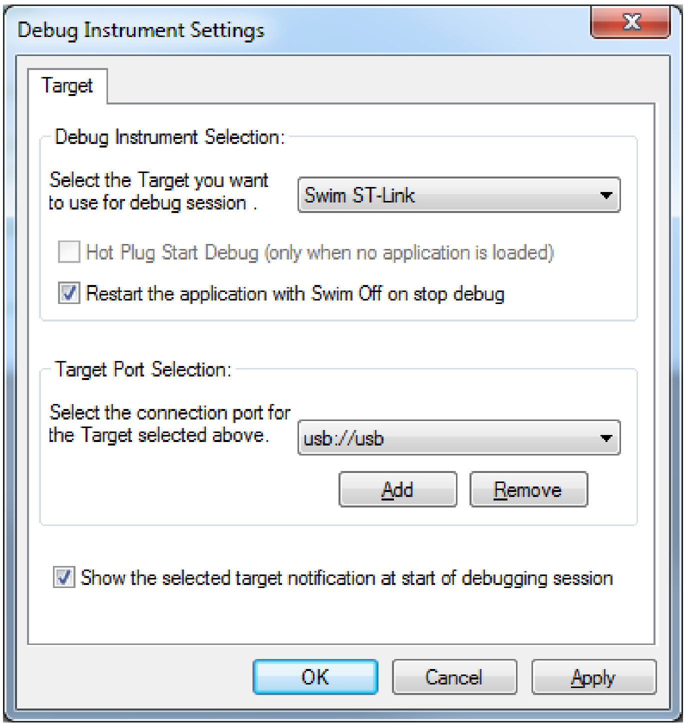


```
Compiling ..\stm8s_stdperiph_driver\src\stm8s_exti.c...
cxstm8 +modsl +debug -pxp +compact -pc99 -l -dSTM8S208 -i..\rfal\include -i..\stm8s_stdperiph_driver\inc -i..\common\Inc -i..\rfal\source\st25r3911 -i..\rfal\source -iC:\Hstm8  -
..\stm8s_stdperiph_driver\src\stm8s_exti.c:

Compiling ..\stm8s_stdperiph_driver\src\stm8s_clk.c...
cxstm8 +modsl +debug -pxp +compact -pc99 -l -dSTM8S208 -i..\rfal\include -i..\stm8s_stdperiph_driver\inc -i..\common\Inc -i..\rfal\source\st25r3911 -i..\rfal\source -iC:\Hstm8  -
..\stm8s_stdperiph_driver\src\stm8s_clk.c:

Running Linker
clnk -lC:\Lib  -o Debug\X-NUCLEO-NFC05A1.sm8 -mDebug\X-NUCLEO-NFC05A1.map Debug\X-NUCLEO-NFC05A1.lkf
cvdwarf Debug\X-NUCLEO-NFC05A1.sm8

Running Post-Build step
chex -o Debug\X-NUCLEO-NFC05A1.s19 Debug\X-NUCLEO-NFC05A1.sm8

X-NUCLEO-NFC05A1.elf - 0 error(s), 0 warning(s)
```

Debugging

1. When debugging for the first time, it may be required to set the Target to be used. Go to *Debug Instrument → Target Settings* and select *SWIM ST-Link*

Figure 6. **Debug Instrument Settings window**



Debug Instrument Settings

Target

Debug Instrument Selection:

Select the Target you want to use for debug session . Swim ST-Link

□ Hot Plug Start Debug (only when no application is loaded)

☑ Restart the application with Swim Off on stop debug

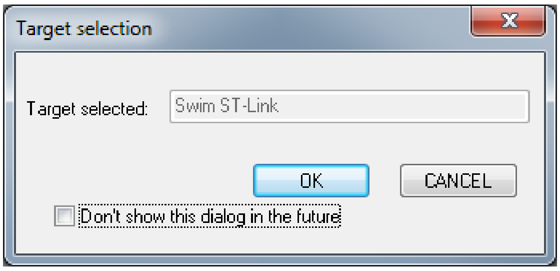
Target Port Selection:

Select the connection port for the Target selected above. usb://usb

☑ Show the selected target notification at start of debugging session

2. In order to program the compiled FW and start a debug session go to *Debug* → *Start Debugging*, and select *Swim ST-Link*

Figure 7. **Target selection window**



Target selection

Target selected: Swim ST-Link

□ Don't show this dialog in the future

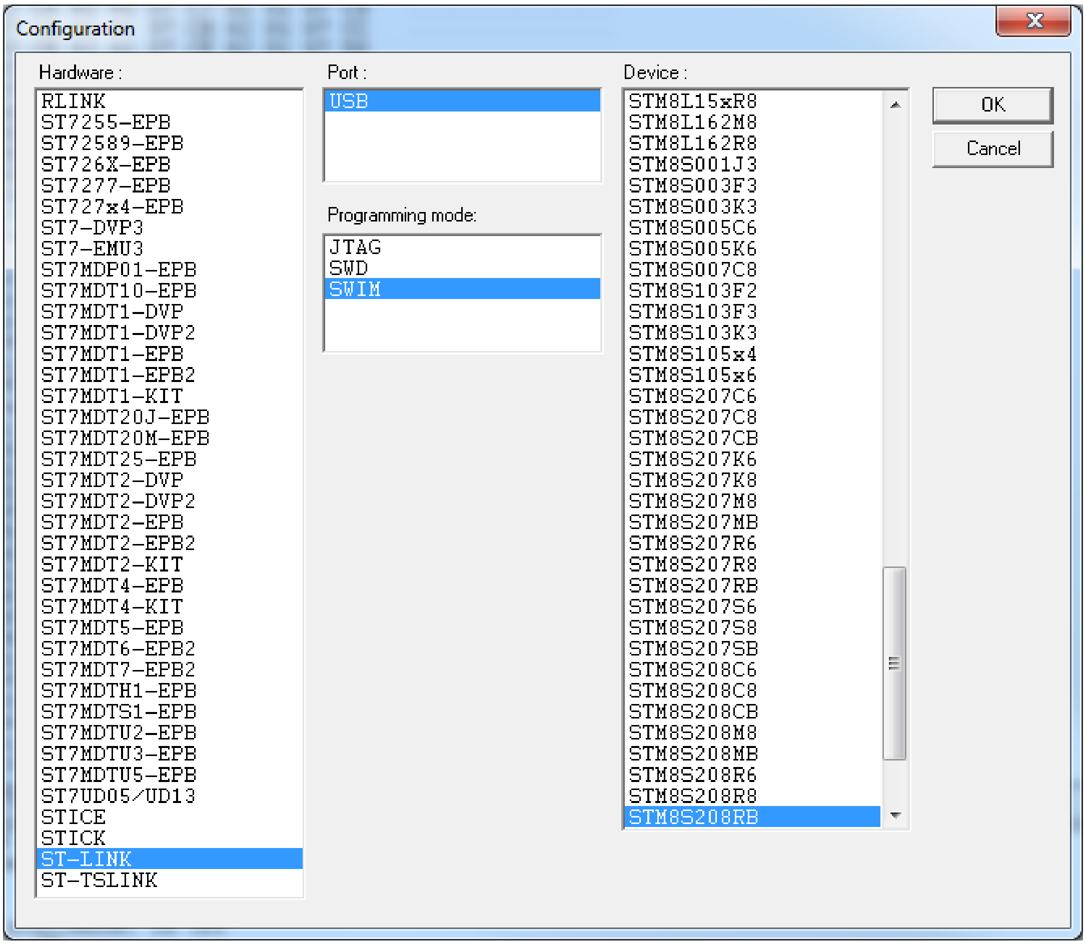
3. Press *Debug* → *Continue* to execute the application.

Flashing

If a precompiled FW is available, it is possible to simply flash it on the IDE by going to *Tools* → *Programmer*, or using another tool called *ST Visual Programmer (STVP)* following the steps below:

1. Open *ST Visual Programmer*
2. Configure the tool in *Configure* → *Configure ST Visual Programmer*

Figure 8. Configuration window



Hardware : ST-LINK

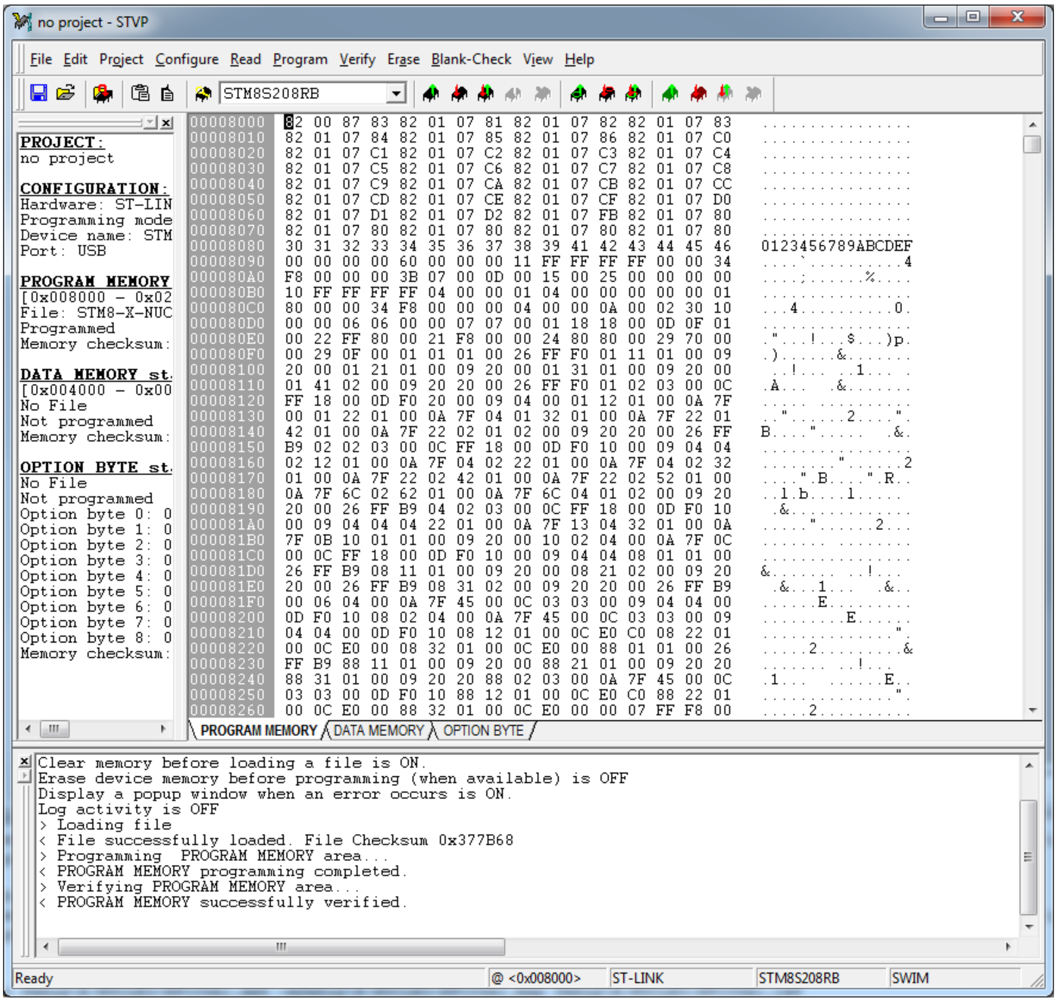
Port : USB

Programming mode: SWIM

Device : STM8S208RB

3. Open the program file by going into *File* → *Open* and selecting the generated file (*.s19)

Figure 9. **Example of file generated at the Flashing step**


no project - STVP
File Edit Project Configure Read Program Verify Erase Blank-Check View Help
STM8S208RB
PROGRAM MEMORY
DATA MEMORY
OPTION BYTE

```
PROJECT:
no project

CONFIGURATION:
Hardware: ST-LIN
Programming mode
Device name: STM
Port: USB

PROGRAM MEMORY
[0x008000 - 0x02
File: STM8-X-NUC
Programmed
Memory checksum:

DATA MEMORY st.
[0x004000 - 0x00
No File
Not programmed
Memory checksum:

OPTION BYTE st.
No File
Not programmed
Option byte 0: 0
Option byte 1: 0
Option byte 2: 0
Option byte 3: 0
Option byte 4: 0
Option byte 5: 0
Option byte 6: 0
Option byte 7: 0
Option byte 8: 0
Memory checksum:
```

```
00008000  82 00 87 83 82 01 07 81 82 01 07 82 82 01 07 83   ................
00008010  82 01 07 84 82 01 07 85 82 01 07 86 82 01 07 C0   ................
00008020  82 01 07 C1 82 01 07 C2 82 01 07 C3 82 01 07 C4   ................
00008030  82 01 07 C5 82 01 07 C6 82 01 07 C7 82 01 07 C8   ................
00008040  82 01 07 C9 82 01 07 CA 82 01 07 CB 82 01 07 CC   ................
00008050  82 01 07 CD 82 01 07 CE 82 01 07 CF 82 01 07 D0   ................
00008060  82 01 07 D1 82 01 07 D2 82 01 07 FB 82 01 07 80   ................
00008070  82 01 07 80 82 01 07 80 82 01 07 80 82 01 07 80   ................
00008080  30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46   0123456789ABCDEF
00008090  00 00 00 00 60 00 00 00 11 FF FF FF FF 00 00 34   ....`..........4
000080A0  F8 00 00 00 3B 07 00 0D 00 15 00 25 00 00 00 00   ....;......%....
000080B0  10 FF FF FF FF 04 00 00 01 04 00 00 00 00 00 01   ................
000080C0  80 00 00 34 F8 00 00 00 04 00 00 0A 00 02 30 10   ...4..........0.
000080D0  00 00 06 06 00 00 07 07 00 01 18 18 00 0D 0F 01   ................
000080E0  00 22 FF 80 00 21 F8 00 00 24 80 80 00 29 70 00   ."...!...$...)p.
000080F0  00 29 0F 00 01 01 01 00 26 FF F0 01 11 01 00 09   .)......&.......
00008100  20 00 01 21 01 00 09 20 00 01 31 01 00 09 20 00    ..!... ..1... .
00008110  01 41 02 00 09 20 20 00 26 FF F0 01 02 03 00 0C   .A...  .&.......
00008120  FF 18 00 0D F0 20 00 09 04 00 01 12 01 00 0A 7F   ..... ..........
00008130  00 01 22 01 00 0A 7F 04 01 32 01 00 0A 7F 22 01   .."......2....".
00008140  42 01 00 0A 7F 22 02 01 02 00 09 20 20 00 26 FF   B....".....  .&.
00008150  B9 02 02 03 00 0C FF 18 00 0D F0 10 00 09 04 04   ................
00008160  02 12 01 00 0A 7F 04 02 22 01 00 0A 7F 04 02 32   ........"......2
00008170  01 00 0A 7F 22 02 42 01 00 0A 7F 22 02 52 01 00   ....".B....".R..
00008180  0A 7F 6C 02 62 01 00 0A 7F 6C 04 01 02 00 09 20   ..l.b....l..... 
00008190  20 00 26 FF B9 04 02 03 00 0C FF 18 00 0D F0 10    .&.............
000081A0  00 09 04 04 04 22 01 00 0A 7F 13 04 32 01 00 0A   ....."......2...
000081B0  7F 0B 10 01 01 00 09 20 00 10 02 04 00 0A 7F 0C   ....... ........
000081C0  00 0C FF 18 00 0D F0 10 00 09 04 04 08 01 01 00   ................
000081D0  26 FF B9 08 11 01 00 09 20 00 08 21 02 00 09 20   &....... ..!... 
000081E0  20 00 26 FF B9 08 31 02 00 09 20 20 00 26 FF B9    .&...1...  .&..
000081F0  00 06 04 00 0A 7F 45 00 0C 03 03 00 09 04 04 00   ......E.........
00008200  0D F0 10 08 02 04 00 0A 7F 45 00 0C 03 03 00 09   .........E......
00008210  04 04 00 0D F0 10 08 12 01 00 0C E0 C0 08 22 01   ..............".
00008220  00 0C E0 00 08 32 01 00 0C E0 00 88 01 01 00 26   .....2.........&
00008230  FF B9 88 11 01 00 09 20 00 88 21 01 00 09 20 20   ....... ..!...  
00008240  88 31 01 00 09 20 20 88 02 03 00 0A 7F 45 00 0C   .1...  ......E..
00008250  03 03 00 0D F0 10 88 12 01 00 0C E0 C0 88 22 01   ..............".
00008260  00 0C E0 00 88 32 01 00 0C E0 00 00 07 FF F8 00   .....2..........
```

```
Clear memory before loading a file is ON.
Erase device memory before programming (when available) is OFF
Display a popup window when an error occurs is ON.
Log activity is OFF
> Loading file
< File successfully loaded. File Checksum 0x377B68
> Programming  PROGRAM MEMORY area...
< PROGRAM MEMORY programming completed.
> Verifying PROGRAM MEMORY area...
< PROGRAM MEMORY successfully verified.
```

```
Ready    @ <0x008000>    ST-LINK    STM8S208RB    SWIM
```

4. To program the STM8 go to *Program → Current Tab.*

4 Automotive compliance

ST's 8-bit microcontroller platform (STM8) offers MCUs for automotive applications with its STM8AF Series and STM8AL Series.

To better enable automotive applications, the RF/NFC library (RFAL) included in this demo is fully compliant with the MISRA C:2012 standard.

MISRA C is a set of guidelines for programming in the C language which have been developed and published by the *Motor Industry Software Reliability Association* (MISRA).

These guidelines aim to improve code safety, security, portability and reliability by identifying aspects of the C language that should be avoided due to their ambiguity and susceptibility to common programming mistakes. The MISRA standard is widely used by the automotive industry for embedded software.

For further detail on RFAL’s MISRA compliance please refer to the Compliance Report provided on RFAL’s documentation folder.

Note: *This demo application is using a STM8S Series device which is not automotive grade. The demo relies on the Nucleo-64 platform and no STM8A Series device is yet available in this platform.*

The STM8S and STM8L Series are very similar to the STM8AF and STM8AL Series, and even pin compatible in some packages, which alleviates porting efforts.

Revision history

Table 2. Document revision history

Date	Version	Changes
05-Nov-2018	1	Initial release.

Contents

List of tables

List of figures