Embedded software document

In this topic, operation system is ubuntu 18.04 LTS, we build by makefile and debug by ozone, you need to install these software to implement

- Tools:
  - vs code:
    https://code.visualstudio.com/download
    - Download version for linux (.deb), extract file and open terminal and type this command
  - git: Open terminal and type this command
  - make file: Open terminal and type this command
  - NRF5x command line tool: reference here:
    http://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.tools%2Fdita%2Ftools%2Fnrf5x_command_line_tools
  - j-link: https://www.segger.com/downloads/jlink/
  - Download version for linux, extract file and open terminal and type this command
  - Ozone: https://www.segger.com/products/development-tools/ozone-j-link-debugger/
    - Download version for linux, extract file and open terminal and type this command
  - ceedling(for testing): Reference here :
    http://www.throwtheswitch.org/ceedling
    - You need to install python, ruby, gem firstly

```
$ sudo install ~/Downloads/"name of file"
```

```
$ sudo apt install git
```

```
$ sudo apt install make
```

```
$ sudo install ~/Downloads/"name of file"
```

```
$ sudo install ~/Downloads/"name of file"
```

- @NOTE: need to check before build project

Download GNU TOOL CHAIN version 7/2017 linux here

https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads

Must place TOOL_CHAIN in Makefile's path

GNU_INSTALL_ROOT := ../../../TOOL_CHAIN/7_2017-q4-major/bin/

How to build a project nrf52 with make file:

- cd into project path, you'll see 1 file name "Makefile", this file was coded follow template.
- Open terminal and type:

```
$ make
```

smartkey'll be built with release mode in default

esle type:

```
$ make config=Debug
```

smartkey'll be built with debug mode

It'll implement these commands:

```
SMART_KEY: LINKER_SCRIPT=Startup/SMART_KEY.ld
```

```
SMART_KEY: $(BUILD_DIRECTORIES) $(OBJECTS)
```

```
    @echo Linking target: $(OUTPUT_FILENAME).out
```

```
    $(NO_ECHO)$(CC) $(LDFLAGS) $(OBJECTS) $(LIBS) -lm -o $(
OUTPUT_BINARY_DIRECTORY)/$(OUTPUT_FILENAME).out
```

```
    $(NO_ECHO)$(MAKE) -f $(MAKEFILE_NAME) -C $(MAKEFILE_DIR
) -e finalize
```

BUILD_DIRECTORIES: define build path

OBJECTS: define prefix C GNU to build

OUTPUT_FILENAME: define name of file output

LDFLAGS: define linker flags

LIBS: define encrypt library path

OUTPUT_BINARY_DIRECTORY: folder to save the build binary

MAKEFILE_NAME: additional makefiles to be used

MAKEFILE_DIR: directory of the addtional makefiles

After makefile built already, there'll be 3 files in path $(BUILD_DIRECTORIES)

- In case make in default, it'll build 3 files release .bin, .hex, .out (only need file .hex for produce)
- In case make config=Debug, it'll build 3 files debug .bin, .hex, .out (we need file .out to debug with ozone)

@NOTE: Error can occur

- Lack of files makefile required:
  - Read error description in terminal firstly.
  - Make sure file s132_nrf52_6.0.0_softdevice.hex (in my case(smartfox) is version 6.0.0 with SDK15), with FOX, Pi, RHINO (file s132_nrf52_2.0.0_softdevice.hex with SDK11)) in path /Components/softdevice/s132/hex/.
  - Check library/source files in path defined at Makefile.
- Error in Makefile:
  - error syntax.