Code and Operation for Ripple_ID project

– Abhijit A. Bokil

Overview

- Code structure
- Build setup
- How to build
- Hardware test setup
- UID generation process
- Deliverables

Code structure

There are 2 software components in this project, one for the test host and the other (the firmware) meant for the target device. Note that the test host PC and associated hostside software play no role in the actual generation of the Unique ID. The hostside software merely listens on the serial (COM) port and displays the UID which is generated by the firmware on target.

The software deliverable for this project consists of the following files

Filename	Description
8051.h	Includes header for 8051 architecture
8052.h	Includes header for 8052 architecture
uid.c	Main firmware source file
uid.h	Firmware include file
uid_hostside.c	Hostside Windows application program source
Makefile	Build file for project (common for host & target)

The ultimate build consists of one hostside Windows application program (.exe) and one Intel format binary (.hex) file constituting the firmware.

Build setup

The software for this project was developed and tested using a Windows 11 host PC and a Nuvoton W78E052DDG 8051 (8052 variant) microcontroller development board. (The board was purchased from NSK Electronics, SP Road, Bangalore). The software was developed in C (C17 std default) using the VSCode IDE for Windows. Software was built using a single build (Makefile) file with the GNU Make utility. The gcc (version 6.3.0) compiler was used to build the hostside application, while the sdcc toolchain (version 4.4.1) was used to build the target firmware. Additionally, the srecord tool (version 1.65) was used to convert the .ihx file (which sdcc outputs) to Intel .hex format. Note that during the firmware compilation process, many intermediate files are generated.

The following softwares (for Windows 11) must therefore be installed and properly functioning in order to build & run this project.

- GCC – GNU C compiler (v6.3.0); available as part of the MinGW project from https://www.mingw-w64.org/downloads/
- SDCC – Small Device C Compiler(v 4.4.1); available for download from https://sdcc.sourceforge.net/snap.php#Windows
- Srecord tool – EPROM Load file manipulator (v1.65) available from https://srecord.sourceforge.net/download.html
- Make – GNU Make utility (v3.82.90) available at https://www.gnu.org/software/make/#download
- Nuvoton 8051 ISP-ICP Programmer tool – available at https://www.nuvoton.com/tool-and-software/software-tool/programmer-tool/ This tool is also available at https://drive.google.com/drive/folders/1lTpAuibYk6_VqTYofS_yUMFSmUmA8zdu

Please ensure that the following commands are accessible from the command line interface in Windows: ***gcc, sdcc, srec_cat, make***

If attempting any of these commands throws errors, ensure that the executables/binaries are in the Windows PATH by editing Windows environment variables.

How to build

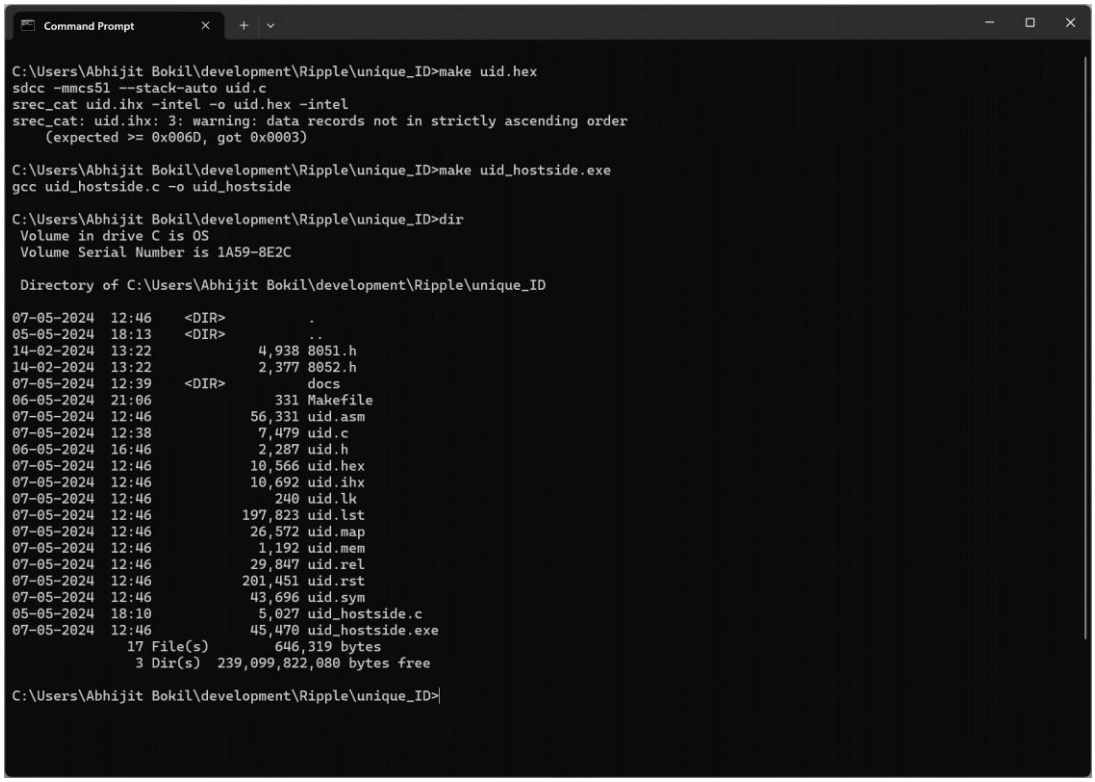
Open a windows command prompt and go to the folder where the software deliverable files are stored. Simply type

$ ***make uid.hex***

at the command prompt as displayed above. This will build the target firmware. Then type

$ ***make uid_hostside.exe***

at the command prompt. This will build the hostside application program.



The hostside file uid_hostside.exe is a Windows command line application meant to be executed on the test host during the UID generation procedure. The uid.hex file is a target binary firmware that is meant to be flashed to the 8051 microcontroller on the development board.

It is possible to delete all binaries and all intermediate files (except the original sources) by entering the command

$ ***make clean***

With the build completed, we are now ready to setup the test system hardware.

Hardware test setup

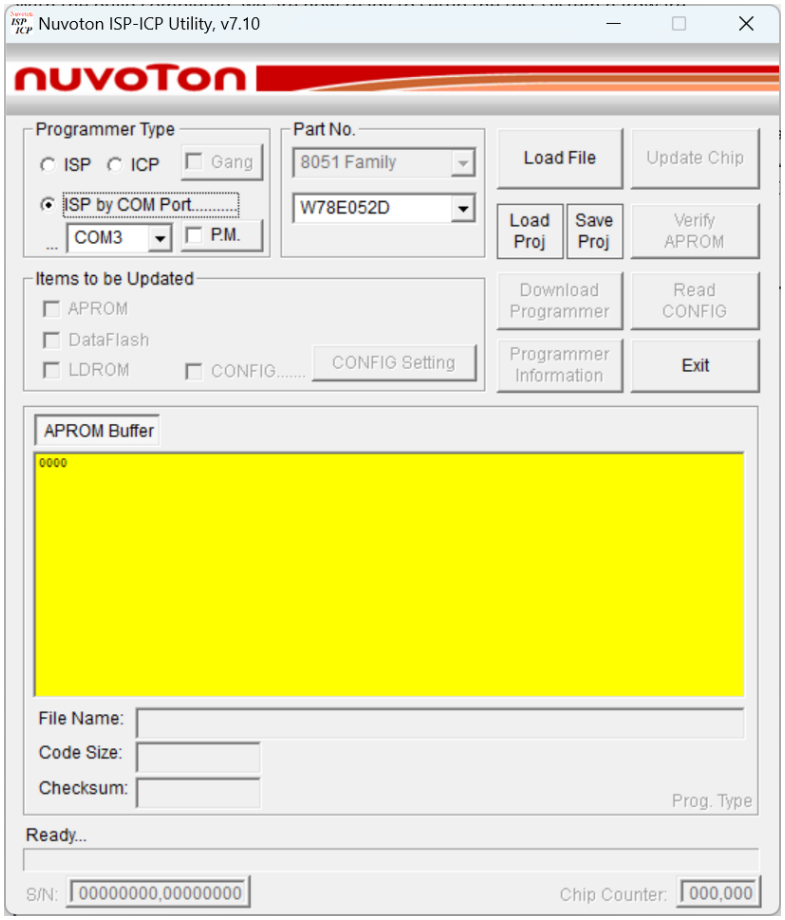
Board connections

Connect jumper wires between the following pins on the development board:

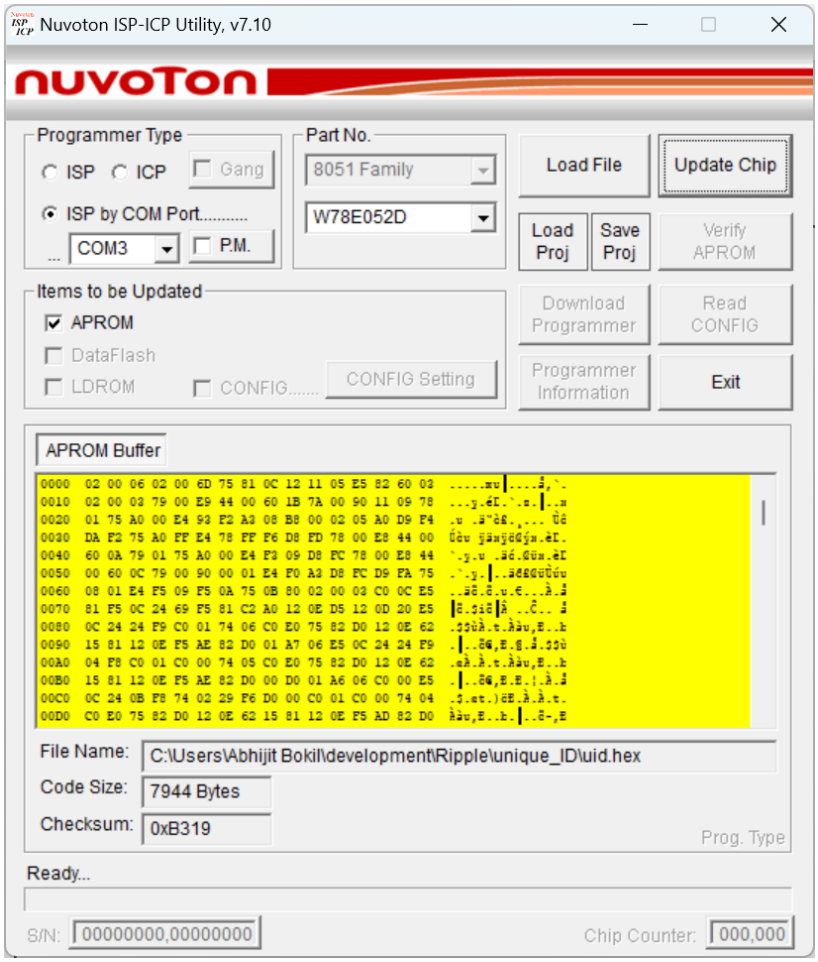
- Port 2, pin 0 ←→ Any LED input pin on the board
- Port 3, pin 6 ←→ SCL pin of the DS1307 chip
- Port3, pin 7 ←→ SDA pin of the DS1307 chip

The development board is powered on (12V) and connected to the test host via a (readily available) serial-to-USB converter cable. The Windows Device Manager utility should show that a new COMport device has been added.

We will now proceed to flash the board with our firmware binary. Execute the Nuvoton 8051 ISP-ICP Programmer Utility, as illustrated below:



Select the COMport at which the dev board sits. Choose “Load file” and select our “uid.hex” file from the folder in which code has been saved.



Click the “Update Chip” button, and ***immediately depress the RESET button*** on the development board. The ISP-ICP Programmer utility in conjunction with the pre-existing bootloader software on the microcontroller erases ROM and updates APROM (Application program memory) with our UID firmware binary.

This completes the firmware flashing process.

UID Generation Process

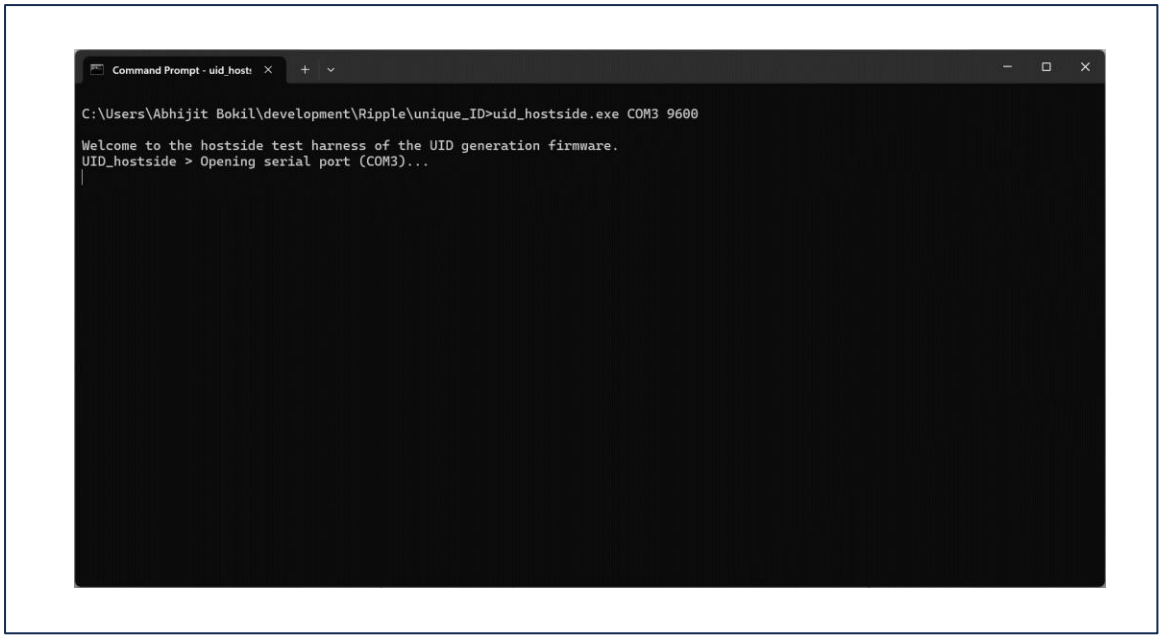
Next, open a command prompt on the Windows test host PC, and go to the deliverables folder.

At the command line type the following:

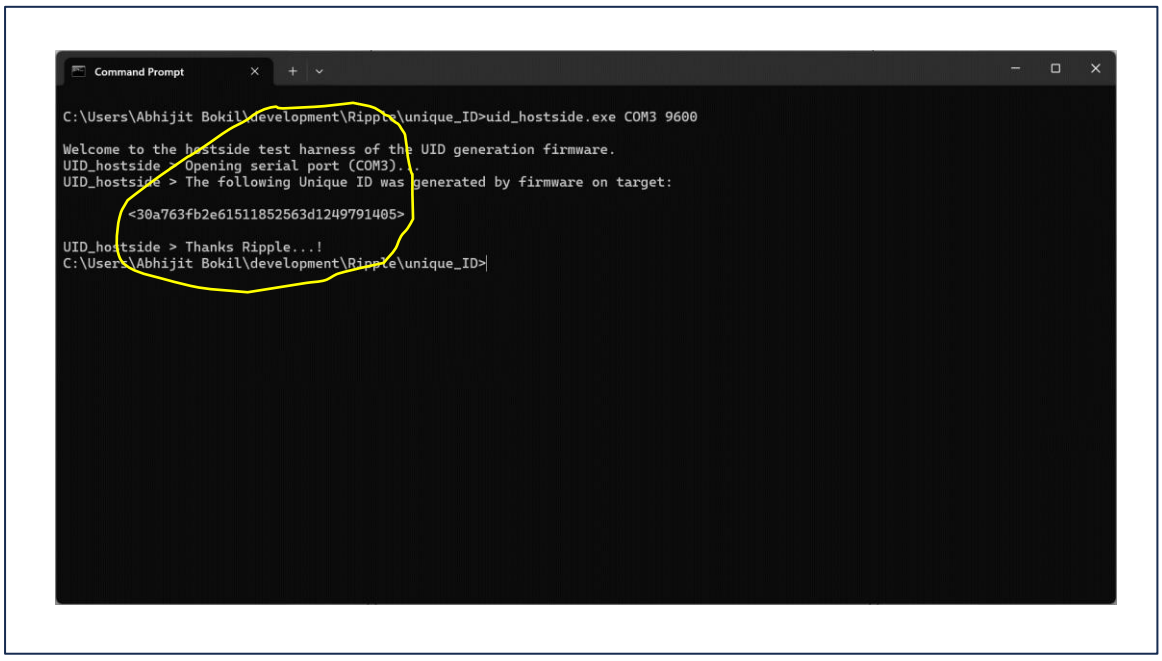
$ ***uid_hostside.exe COM3 9600***

Replace COM3 in the above command with the particular COMport on your PC where the devboard sits (e.g. COM2, COM5, etc)

The hostside application executes, as shown below:



As soon as the hostside program executes, press the RESET button on the dev board. The firmware on the board will then execute and a UID will be generated and written to the serial port. The UID is then displayed by the hostside program as illustrated below:



Repeating the UID generation process by executing hostside application followed by pressing RESET on the devboard will generate a new different UID.

Deliverables

All project artefacts including a Research report, a Design document and a how-to Code_Operation document (this doc), as well as all software deliverables are available online on Github at

www.github.com/aabcode/ripple_uid

To fetch all project files into a local folder, execute the following command

$ ***git clone https://github.com/aabcode/ripple_uid*** from the command line.