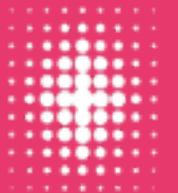


FPC2532 ALLKEY DEVELOPMENT KIT

GETTING STARTED – USER GUIDE



FINGERPRINTS

Introduction

These slides provides guidelines on how to set up the FPC2532 AllKey development kit, as well as build and flash example code provided by Fingerprint Cards AB (FPC) onto an STM32U545 development board. The first example demonstrates a basic user flow of enrolling and identifying a fingerprint, and the second shows the navigation functionality.

To get started, a few components and software tools are required:

Hardware components

- ⊗ FPC2532 AllKey development kit, which includes:
 - ⊗ FPC5788 Arduino shield board
 - ⊗ FPC5789 Feather wing
 - ⊗ FPC2532 (Mounted on the Feather wing)
- ⊗ STM32U545 development board NUCLEO-U545RE-Q (Not provided by FPC)
- ⊗ USB-C cable (Not provided by FPC)

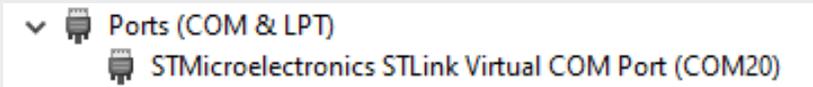
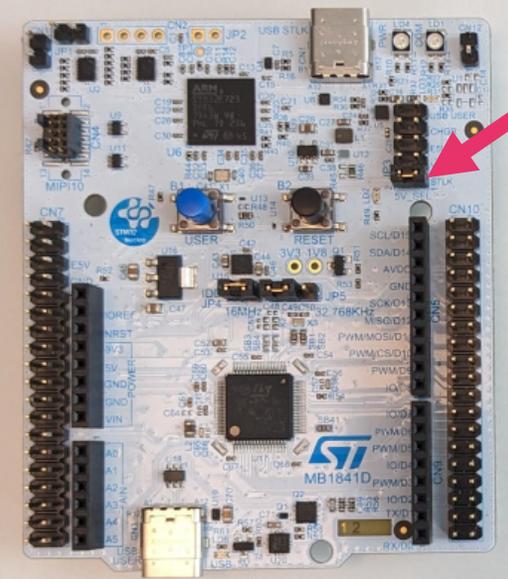
Software prerequisites

- ⊗ STM32CubeProgrammer (and STLink drivers) downloaded and installed
 - ⊗ Available at <https://www.st.com/en/development-tools/stm32cubeprog.html>
- ⊗ GNU Make (<https://www.gnu.org/software/make/>)
 - ⊗ **Note:** Preinstalled in most Unix-like environments and Windows alternatives listed below
- ⊗ arm-none-eabi toolchain (Recommended version 10.3+)
 - ⊗ Available at [Arm GNU Toolchain Downloads – Arm Developer](#)
 - ⊗ **Note:** Requires a Unix-like environment. If running on Windows, one can use e.g., MSYS2, Cygwin, or Windows Subsystem for Linux (WSL)
- ⊗ PuTTY downloaded and installed
 - ⊗ Available at: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Connecting the device

- ① Make sure the bottom jumper (STLK) is connected, as seen in the top figure
- ① Attach the FPC2532 AllKey development kit onto the STM32U545 and connect to the PC via USB, as seen in the bottom figure
 - ① **Note:** The USER and RESET buttons on the board will be partially covered

If drivers are installed correctly, it should appear in Device Manager under “Ports (COM & LPT)” as “STMicroelectronics STLink Virtual COM Port” once connected



Build and flash Example Application (1/2)

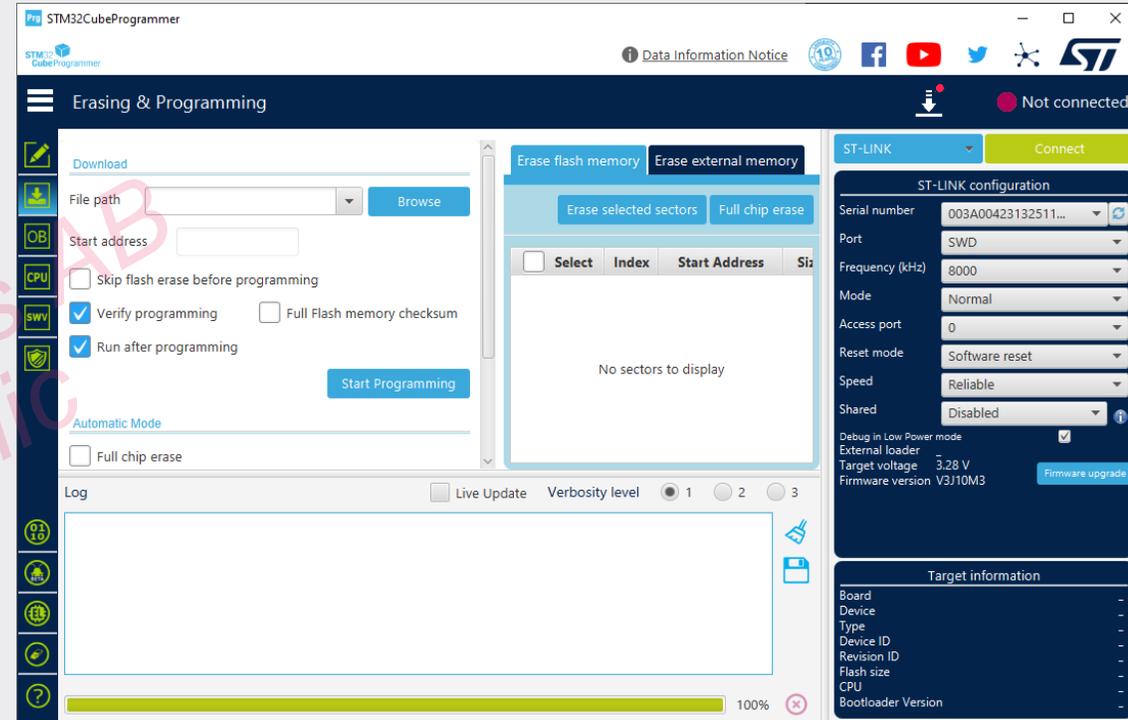
- ① Open a Linux based terminal, navigate into the main delivery folder and run *make* with *arm-none-eabi toolchain*
 - ① **Note:** Further build details and arguments can be found in README.md

If everything worked correctly, the .elf file should be available under *out/enroll_identify/stm32u545/enroll_identify.elf*

Build and flash Example Application (2/2)

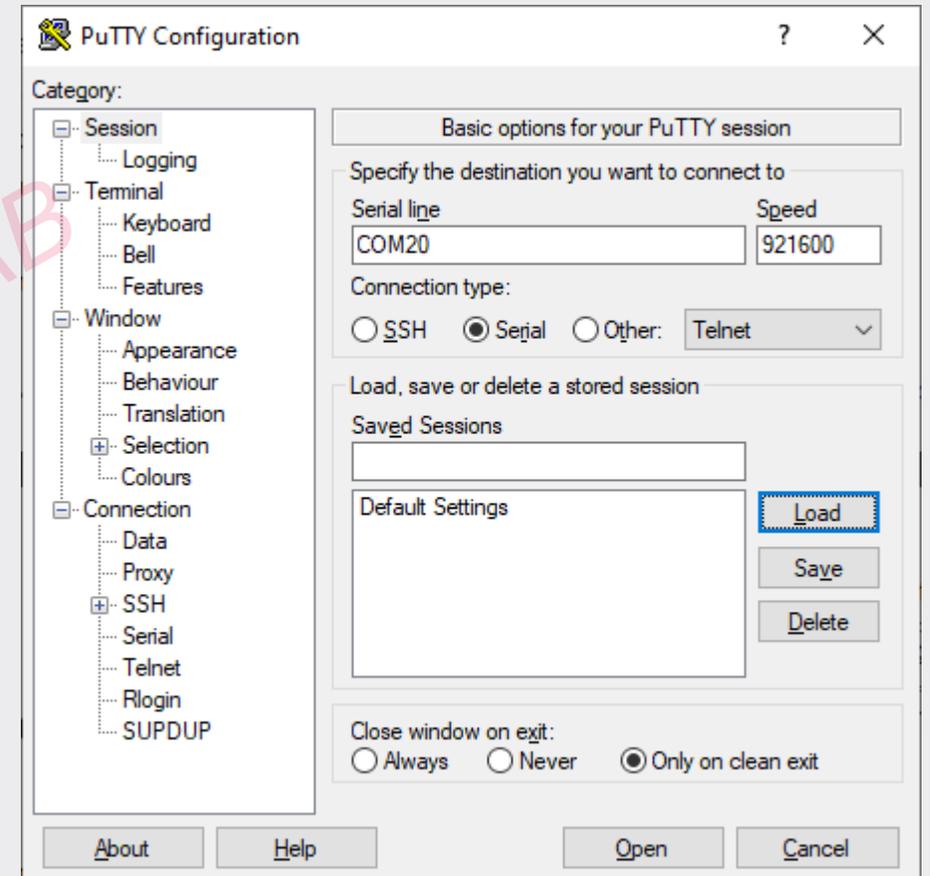
To flash the .elf file onto the development board, use the following steps:

- ① Open the STM32CubeProgrammer application
- ② Set Port to “SWD” and Frequency to 8000 kHz as seen in the figure, then press connect
- ③ Navigate to “Erasing & Programming” menu
- ④ Select the .elf file generated in the previous slide under “File path”
- ⑤ Check the “Verify programming” and “Run after programming” boxes
- ⑥ Click “Start Programming”, which should be followed by a Success message



Running the example code (1/3)

- Start PuTTY and configure for “Serial” connection type and speed 921600, as seen in the image
- **Note:** Serial line (COM port) might differ. Check in Device Manager which port is assigned to the device
- **Note:** For readability, consider checking “Implicit CR in every LF” under the “Terminal” category
- Press “Open” and a new PuTTY terminal should open



Running the example code (2/3)

- ① On startup, the software version and number of templates enrolled is printed
 - ① This will be printed again after a HW reset (Black button)
- ① Enrollment starts as soon as you place your finger on the sensor. Repeatedly lift and place the same finger on the sensor to progress enrollment
- ① Once two fingers are enrolled, subsequent touches will try to match/identify against either of the enrolled fingers
 - ① To delete the stored templates and restart enrollment, press the USER button (blue) followed by touching the sensor

```
COM20 - PuTTY
FPC2532 example app (SPI)

Got version: FPC2532 App (SiP)/2024.2.0.039/release/2024-08-15 11:14:44/c23b239
Found 0 template(s) on device
Enroll samples remaining: 11, feedback: Progress (2)
Enroll samples remaining: 10, feedback: Progress (2)
Enroll samples remaining: 9, feedback: Progress (2)
Enroll samples remaining: 8, feedback: Progress (2)
Enroll samples remaining: 7, feedback: Progress (2)
Enroll samples remaining: 6, feedback: Progress (2)
Enroll samples remaining: 5, feedback: Progress (2)
Enroll samples remaining: 5, feedback: Progress.Immobile (7)
Enroll samples remaining: 4, feedback: Progress (2)
Enroll samples remaining: 4, feedback: Progress.Immobile (7)
Enroll samples remaining: 3, feedback: Progress (2)
Enroll samples remaining: 2, feedback: Progress (2)
Enroll samples remaining: 1, feedback: Progress (2)
Enroll samples remaining: 0, feedback: Done (1)

Enroll finished. Starting Identify.
Identify match on id 1
```

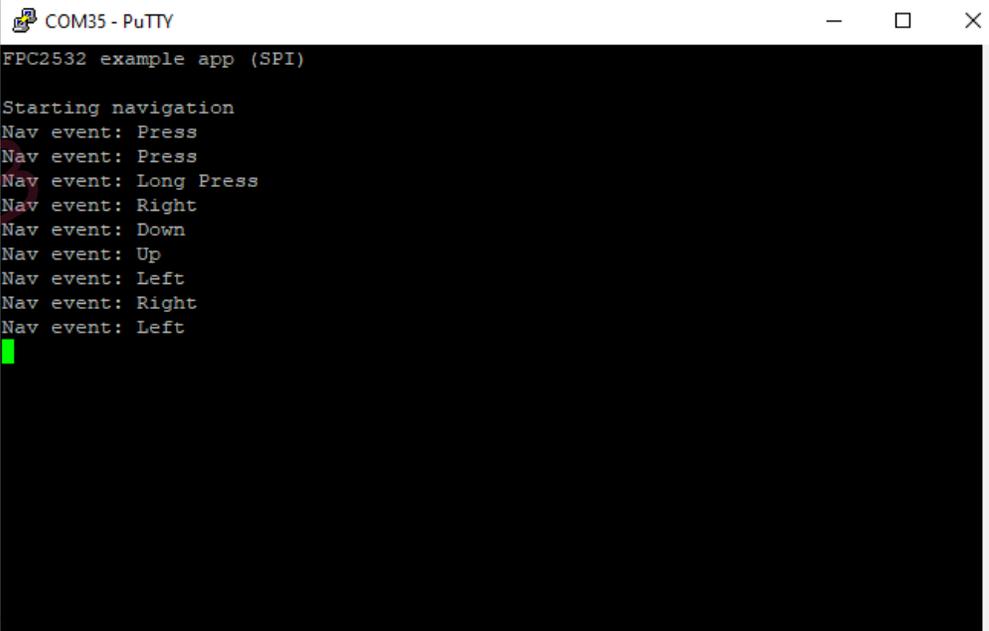
Note: Source code for the example application is found under `examples/enroll_identify/src/`

Running the example code (3/3)

- ① There is also support for a navigation example. To run this, repeat the steps from “Build and flash Example application” but run *make APP=navigation*
 - ① The generated .elf file will be under *out/navigation/stm32u545/navigation.elf*
- ① Navigation means that the device continuously checks for gestures, i.e., swipes and presses on the sensor. No biometric operations are performed.
 - ① To stop navigation, press the USER button (blue) on the STM device
 - ① To start again, press the black RESET button on the STM device

Note: The “up” direction in navigation is by default what might be seen as right from the user's perspective. The directions can be easily configured in the software.

Note: Source code for the example application is found under *examples/navigation/src/*



```
COM35 - PuTTY
FPC2532 example app (SPI)
Starting navigation
Nav event: Press
Nav event: Press
Nav event: Long Press
Nav event: Right
Nav event: Down
Nav event: Up
Nav event: Left
Nav event: Right
Nav event: Left
```