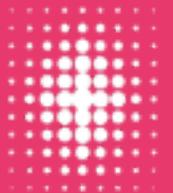


FPC2534 ALLKEY PRO DEVELOPMENT KIT

GETTING STARTED – USER GUIDE



FINGERPRINTS

Introduction

These slides provides guidelines on how to set up the FPC2534 AllKey Pro development kit and run provided example code on an STM32U545 development board. Prebuilt .elf files are available to get up and running quickly, and at the end of the slides we list the tools needed to make modifications and recompile the code.

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

⤵ Hardware components

- ⤵ FPC2534 AllKey Pro development kit, which includes:
 - ⤵ FPC5788 Arduino shield board
 - ⤵ FPC5789 Feather wing
- ⤵ STM32U545 development board NUCLEO-U545RE-Q (Not provided by FPC)
- ⤵ USB-C cable (Not provided by FPC)

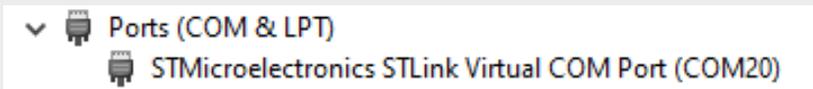
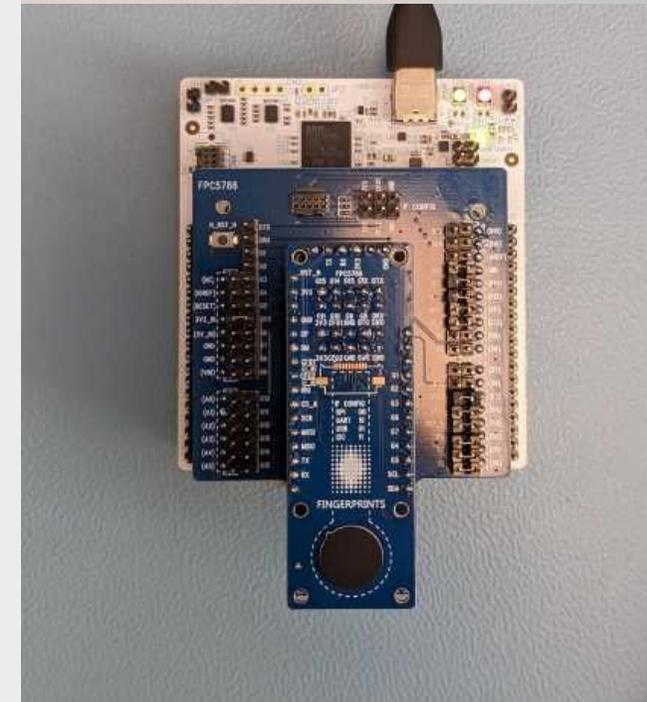
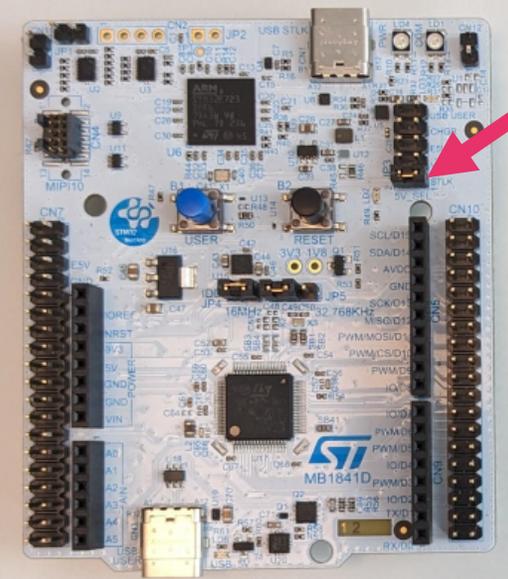
⤵ Software prerequisites

- ⤵ STM32CubeProgrammer (and STLink drivers) downloaded and installed
 - ⤵ Available at <https://www.st.com/en/development-tools/stm32cubeprog.html>
- ⤵ PuTTY downloaded and installed
 - ⤵ Available at: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
 - ⤵ **Note:** Other tools for reading serial ports work just fine

Connecting the device

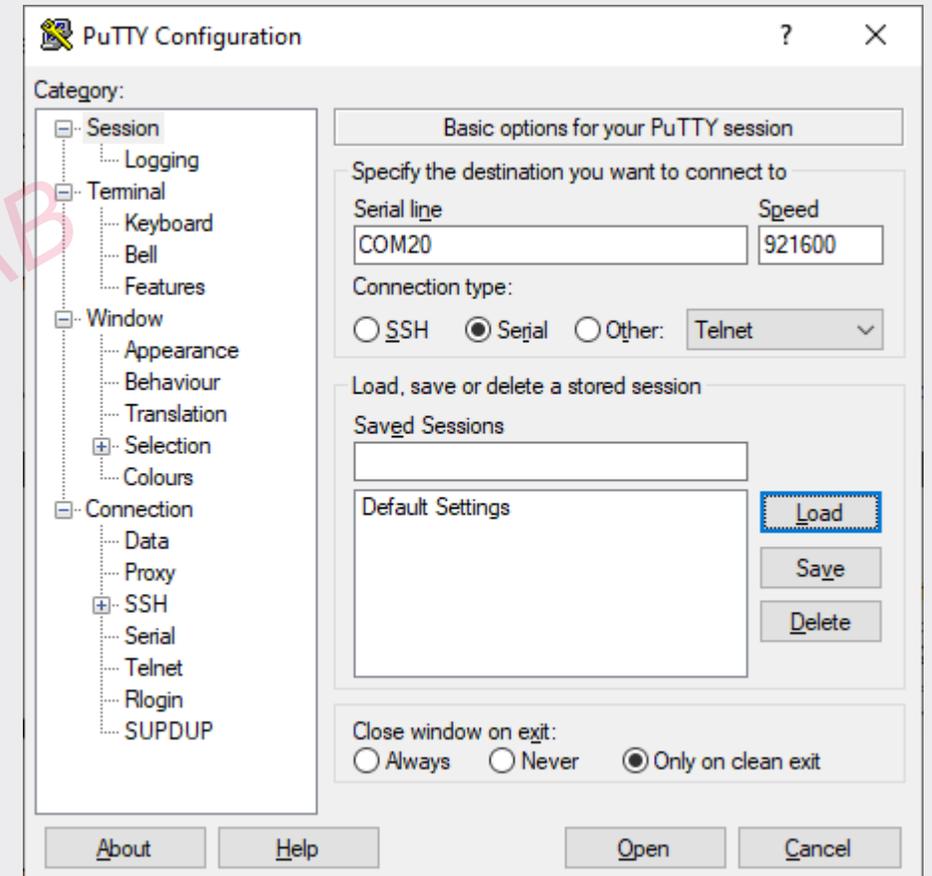
- ① Make sure the bottom jumper (STLK) is connected, as seen in the top figure
- ① Attach the FPC2534 AllKey Pro development kit onto the STM32U545 and connect to the PC via USB, as seen in the bottom figure
 - ① **Note:** The USER and RESET buttons on the board will be partially covered

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



Flash and run Example Application (1/2)

- 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



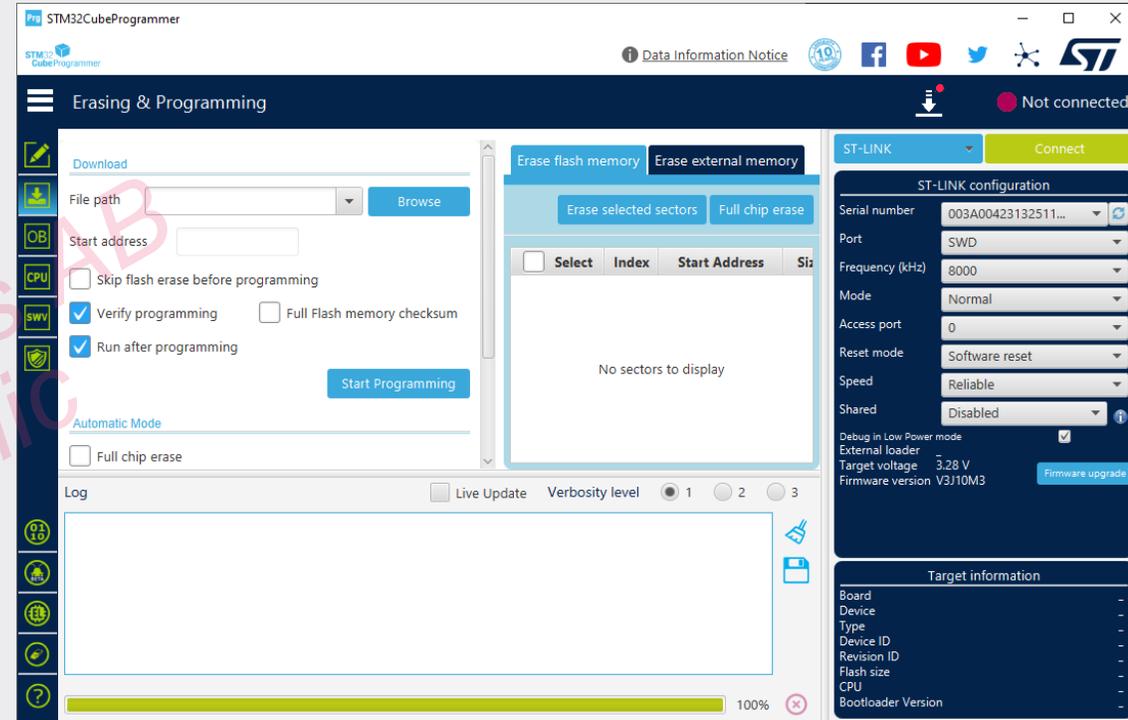
Flash and run Example Application (2/2)

There are three examples available: *enroll_identify*, *navigation*, and *secure_com*

- Pre-generated .elf files are available under `out/{example_app}/stm32u545/`

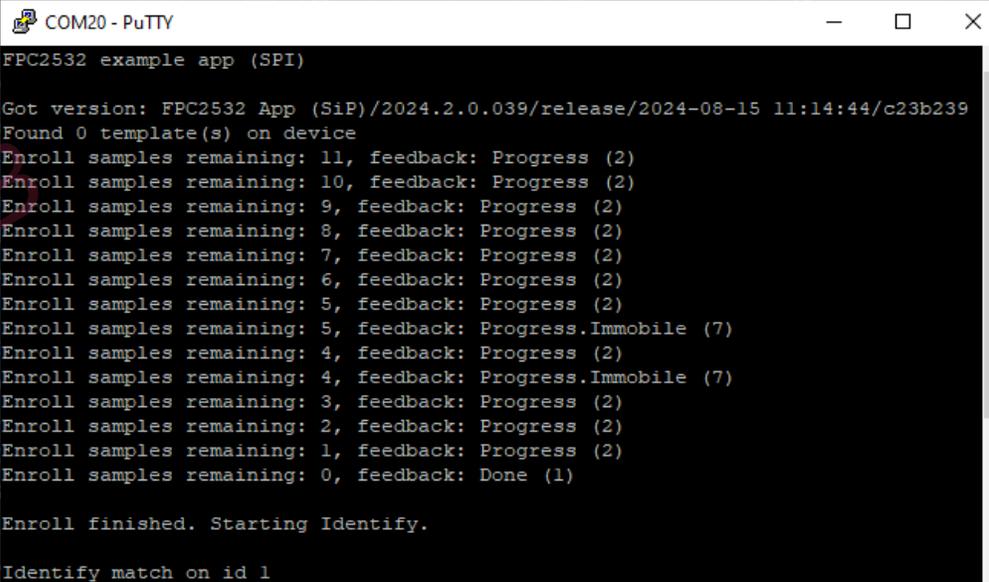
To flash the .elf file onto the development board, do the following:

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



Running Example Code – Enroll & Identify

- 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
```

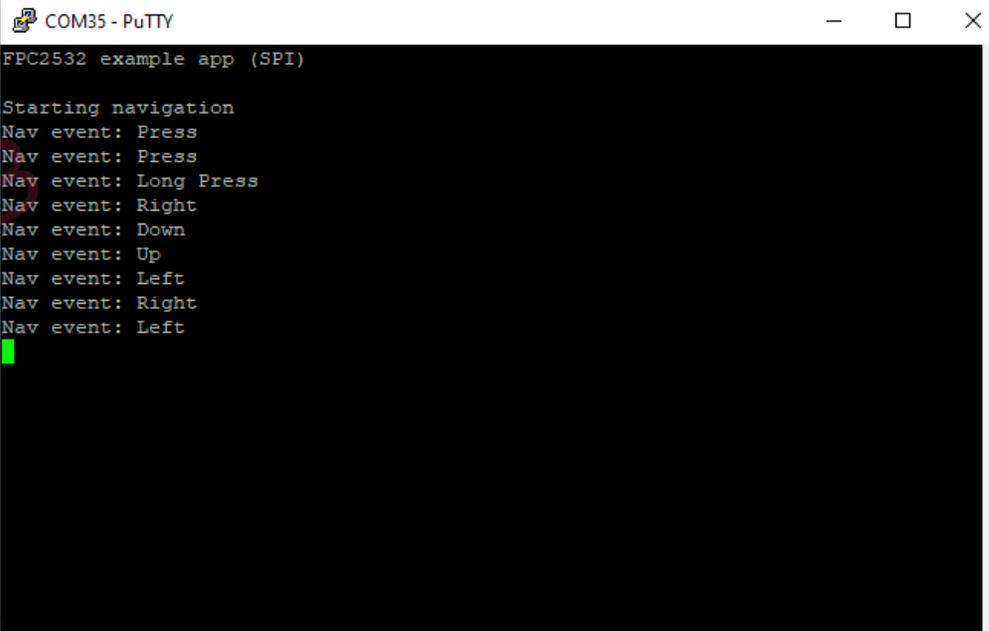
Note: Source code for the example application is found under `examples/enroll_identify/src/`

Running Example Code – Navigation

- ① The second example demonstrates navigation, which 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
```

Running Example Code – Secure Com

- ① The final example highlights features which are only available in the Pro version.
- ① Features demonstrated, in order of execution:
 - ① Get/set system configuration (Not Pro exclusive, but some required settings are)
 - ① Key provisioning for encrypted communication
 - ① Template export/import
 - ① Factory reset

Note: Details for the commands are found in the Product Specification and Feature Supplement, as well as in the `fpc_api.h` header file

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

Build and compile Example Applications

To modify and build upon the provided examples, we recommend the following:

- ③ GNU Make (<https://www.gnu.org/software/make/>)
 - ③ **Note:** Preinstalled in most Unix-like environments and Windows alternatives listed below
- ③ arm-none-eabi toolchain (Recommended version 10.3 or later)
 - ③ Available at [Arm GNU Toolchain Downloads – Arm Developer](#)
 - ③ **Note:** Requires a Unix-like environment. If running on Windows, one can use e.g., MSYS2, Cygwin, or Windows Subsystem for Linux (WSL)
- ③ Open a Linux based terminal, navigate into the main delivery folder and run *make* with *arm-none-eabi toolchain*
 - ③ **Note:** Further build details and arguments can be found in README.md

Output files are located at the same location pre-generated files are (*out/{app}/stm32u545/*)

(Optional) using I2C

- ① To run example apps over I2C interface, the SCL and SDA pins must be connected (see image)
 - ① Default interface SPI, unless otherwise specified when compiling
- ① Having them connected will **not** interfere when running any other interface, e.g., SPI or UART

Fingerprint Cards AB
Public

