FPC2532 ALLKEY DEVELOPMENT KIT

GETTING STARTED – USER GUIDE





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:
 - O FPC5788 Arduino shield board
 - O 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
 - O STM32CubeProgrammer (and STLink drivers) downloaded and installed
 - O Available at https://www.st.com/en/development-tools/stm32cubeprog.html
 - O GNU Make (https://www.gnu.org/software/make/)
 - O Note: Preinstalled in most Unix-like environments and Windows alternatives listed below
 - o arm-none-eabi toolchain (Recommended version 10.3+)
 - O Available at Arm GNU Toolchain Downloads Arm Developer
 - O Note: Requires a Unix-like environment. If running on Windows, one can use e.g., MSYS2, Cygwin, or Windows Subsystem for Linux (WSL)
 - O PuTTy downloaded and installed
 - Available at: <u>https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html</u>



Connecting the device

- Make sure the bottom jumper (STLK) is connected, as seen in the top figure
- Attach the FPC2532 AllKey development kit onto the STM32U545 and connect to the PC via USB, as seen in the bottom figure
 - 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

Ports (COM & LPT)
 STMicroelectronics STLink Virtual COM Port (COM20)





Build and flash Example Application (1/2)

- Open a Linux based terminal, navigate into the main delivery folder and run *make* with *arm-none-eabi toolchain*
 - O 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:

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

Pro STI	M32CubeProgrammer			- 🗆 ×	
STAD2 C		Data Information Notice) f 🕒	🔰 🔀 🖅	
	Erasing & Programming		<u>.</u>	Not connected	
CPU SWV	Download File path File path Start address Skip flash erase before programming Verify programming Full Flash memory checksum Run after programming Start Programming	Erase flash memory Erase external memory Erase selected sectors Full chip erase Select Index Start Address Siz No sectors to display	ST-LINK Serial number Port Frequency (kHz) Mode Access port Reset mode Speed Shared	Connect UNK configuration U03A00423132511 SWD B000 Normal O Software reset Reliable Disabled	
	Full chip erase Log Live Upd	tate Verbosity level 1 2 3	Debug in Low Power r External loader Target voltage	mode 🗹 3.28 V V3J10M3 Firmware upgrade	
		*	Ta Board Device Type Device ID Revision ID	arget information - - - -	
?		100% 🛞	Flash size CPU Bootloader Version	- - -	



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

🕵 PuTTY Configuration	? ×
Putty Configuration Category: □- Session □- Terminal □- Weyboard □- Bell □- Features □- Window □- Appearance □- Behaviour □- Translation ⊡- Selection □- Colours □- Connection □- Data □- Proxy ⊡- SSH □- Serial □- Telnet □- Rlogin □- SUPDUP	? × Basic options for your PuTTY session Specify the destination you want to connect to Serial line Speed COM20 921600 Connection type: 921600 Connection type: 921600 Load, save or delete a stored session Saved Sessions Default Settings Load Default Settings Load Close window on exit: Delete Always Never
<u>A</u> bout <u>H</u> elp	<u>O</u> pen <u>C</u> ancel



Running the example code (2/3)

- On startup, the software version and number of templates enrolled is printed
 - This will be printed again after a HW reset (Black button)
- 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
- Once two fingers are enrolled, subsequent touches will try to match/identify against either of the enrolled fingers
 - To delete the stored templates and restart enrollment, press the USER button (blue) followed by touching the sensor

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

Putty	_		×
FPC2532 example app (SPI)			^
Got version: FPC2532 App (SiP)/2024.2.0.039/release/2024-08-15 :	11:14:44	/c23b23	39
Enroll samples remaining: 11 feedback: Progress (2)			
Enroll samples remaining: 10 feedback: Progress (2)			
Enroll samples remaining, 10, feedback, Flogress (2)			
Enroll samples remaining: 9, reedback: 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



Running the example code (3/3)

- O 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
 - The generated .elf file will be under *out/navigation/stm32u545/navigation.elf*
- Navigation means that the device continuously checks for gestures, i.e., swipes and presses on the sensor. No biometric operations are performed.
 - O To stop navigation, press the USER button (blue) on the STM device
 - 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/*

,	P COM35 - PuTTY	_	×
"	FPC2532 example app (SPI)		\sim
	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		

