

СКАНИРОВАНИЕ ОТПЕЧАТКОВ ПАЛЬЦЕВ

1. Постановка задачи.

- Разработка устройства для считывания отпечатков пальцев на основе линейного сканера отпечатков.
- Осуществление сборки двумерного изображения отпечатка пальца из большого числа одномерных кадров (допускается решать эту задачу с помощью ПК).
- Относительное положение одномерных кадров должно корректироваться средствами корреляционного анализа.
- При тестировании устройства сторонним тестировщиком более чем в 50% случаев устройство должно выдавать корректные изображения отпечатков (допускаются незначительные артефакты, суммарная площадь которых не превышает 10% от площади изображения).

2. Компоненты.

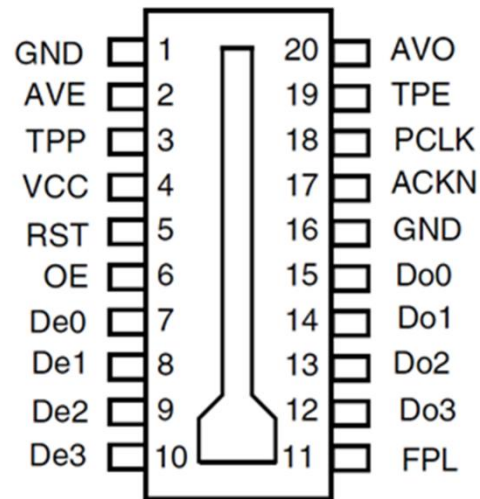
- Отладочная плата STM32F429 Discovery
- Линейный сканер отпечатков пальцев FCD4B14
- ПК (IDE для работы с мкк, IDE для работы с python 3)
- UART to USB bridge
- Рамка с припаянными проводами (17) и отладочная плата для сканера
- Провода (2 штуки) для подключения UART

3. Схема сборки.

- Ножки датчика с описанием подключения к ножкам МКК

	Обозначение	Название	Ножка на МКК
1	GND	GND	GND
2	AVE	Analog output (even)	-
3	TPP	Temp stabilization power	3V
4	VCC	Positive supply voltage	-
5	RST	Reset pin	PD10
6	OE	Output Enable	PD15
7	De0	Digital output	PB8
8	De1	Digital output	PB9
9	De2	Digital output	PB10
10	De3	Digital output	PB11
11	FPL	Front plane	GND
12	Do3	Digital output	PB7
13	Do2	Digital output	PB6
14	Do1	Digital output	PB5
15	Do0	Digital output	PB4
16	GND	GND	GND
17	ACKN	Acknowledge	PB2
18	PCLK	Clock pin	PB0
19	TPE	Temperature stabilization	PD14
20	AVO	Analog output (odd)	-

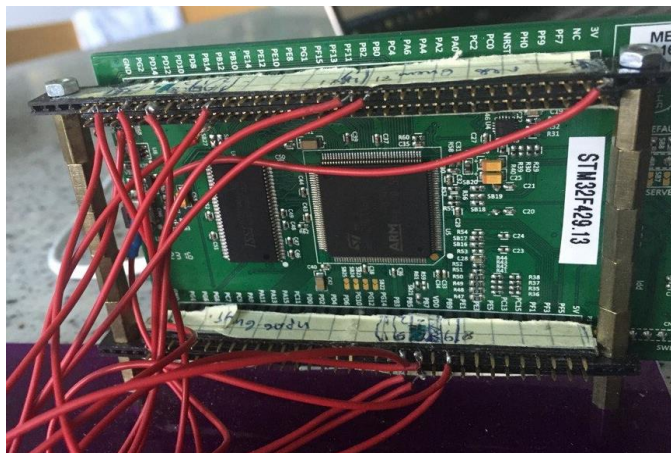
○ Схема датчика отпечатков пальцев



○ Подключение устройства к ПК:

1. Подключить провода UART: от UART to USB bridge GND к GND на МКК; от Rx на UART to USB bridge к PD5 на МКК.
2. Вставить UART to USB bridge в USB отверстие на ПК
3. Подключить отладочную плату МКК к ПК через USB кабель

○ Фотографии собранного устройства



4. Библиотеки.

- Для программы на python (использованная среда разработки Spyder; источник – Anaconda3):
 1. pyserial
 2. matplotlib.pyplot
 3. array
 4. numpy
- Для программы для МКК (IDE IAR Workbench):
 1. Библиотека STM32Cube_FW_F4_V1.18.0 (<http://www.st.com/en/embedded-software/stm32cubef4.html> - сейчас доступна более поздняя версия)
 2. Библиотека HAL (из п. 1.)

5. Описание работы устройства.

- На МКК:
 1. Задается частота тактирования датчика (ШИМ) (изначально 45МГц, с делителем и периодом – 25кГц).
 2. В прерывании таймера происходит считывание двух байтов (пикселей).
 3. После считывании одной полосы (281 колонка по 8 пикселей) происходит передача по UART DMA всей полосы.
- В программе на ПК:
 4. Считывание первой полосы.
 5. Нахождение синхронизационного байта (всегда есть последовательность 1111 0000 1111 0000).
 6. Считывание следующей полосы только до синхронизационной колонки.
 7. Считывание полноценной первой полосы.
 8. Последующее считывание полосок с их обработкой.
- Обработка:
 9. Проверка наличия данных больше 10 (1010). (если нет, то вообще не учитываем)
 10. Проверка количества вхождения в полосу элемента 15 (1111). Это сделано для того, чтобы отсечь вначале полосы с засветкой.
 11. Сравнение полосы с предыдущей (подсчет совпадающих пикселей).
 12. Вывод изображения (с помощью библиотеки matplotlib.pyplot)

6. Источники информации

- Документация к датчику FCD4B14
- Документация к библиотеке HAL: STM32CubeF4_Manual

7. Примеры отпечатков

