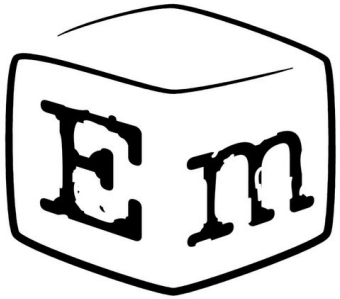
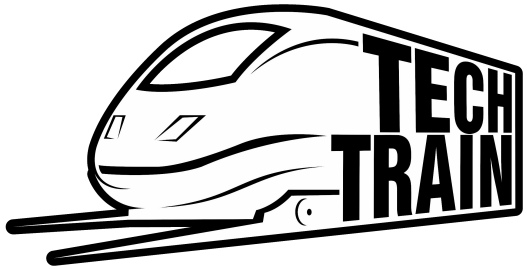




Распознаем образы на микроконтроллере, используя библиотеку OpenCV

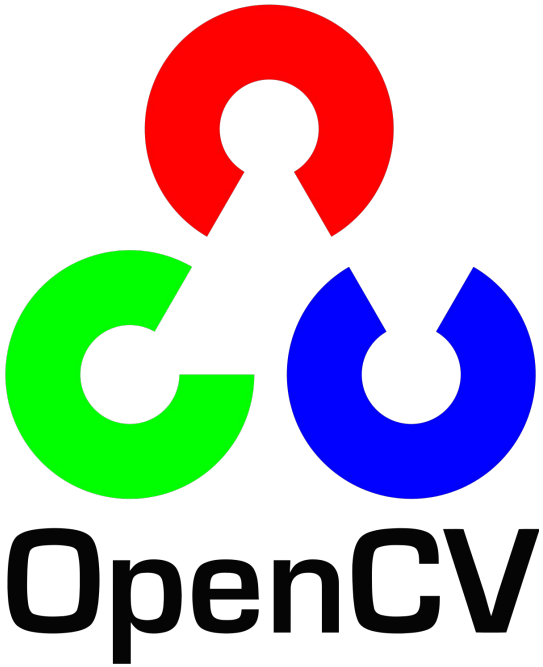
Александр Калмук

TechTrain, 2019

OpenCV

OpenCV — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом.

Хорошая документация, много примеров, кроссплатформенность.



А зачем на микроконтроллере то?

- *В интернете много запросов про OpenCV на STM32, а значит это интересно людям.*
- У нас был ранее положительный опыт с библиотекой PJSIP.
- Некоторая нацеленность других крупных проектов на рынок микроконтроллеров:
    - Qt - кроссплатформенный фреймворк для разработки программного обеспечения на языке программирования C++
    - TensorFlow - открытая программная библиотека для машинного обучения

STM32F746G-DISCOVERY

- ARM® Cortex®-M7 core
- 1 Mbytes of Flash memory and 340 Kbytes of RAM
- 128-Mbit SDRAM (64 Mbits accessible)
- 128-Mbit Quad-SPI Flash memory

STM32F769I-DISCOVERY

- ARM® Cortex®-M7 core
- 2 Mbytes of Flash memory and 512+16+4 Kbytes of RAM
- 128-Mbit SDRAM
- 512-Mbit Quad-SPI Flash memory

В чем сложность?

- Если скомпилировать OpenCV даже с минимальным набором модулей, во внутреннюю память микроконтроллера библиотека не влезет (даже без учёта ОС) из-за очень большого размера кода.
- Нужна поддержка POSIX со стороны ОС.
- Нужна поддержка C++:
    - Конструкторы, исключения, rtti
    - Помимо стандартной библиотеки требуется свежий STL

Embox

Embox — свободная операционная система реального времени (RTOS), разрабатываемая для встроенных систем.

Особенность:

Обеспечивает возможность запуска больших (“десктопных”) библиотек на микроконтроллере.

А теперь все по порядку :)

Конфигурируем и собираем OpenCV.

Посмотрим на файл конфигурации в Embox:
third-party/lib/opencv/Makefile

Поддержка C++

- Constructors / Destructors
- Exceptions

Другие внешние зависимости

- POSIX (например, работа с файлами)
- Standard Template Library

Сборка проекта и запуск

- Посмотрим конфигурацию, проанализируем требуемые ресурсы
- Скомпилируем, загрузим образ на плату.
- Запустим распознавание границ на картинке (детектор границ Кэнни).

Запуск из QSPI на STM32F746G

- Нужно учесть, что программа будет физически располагаться по другим адресам (не в быстрой памяти SRAM).
- Нужен загрузчик, который разместит код программы и ОС в память QSPI
- “Прыгнуть” на стартовый адрес программы.

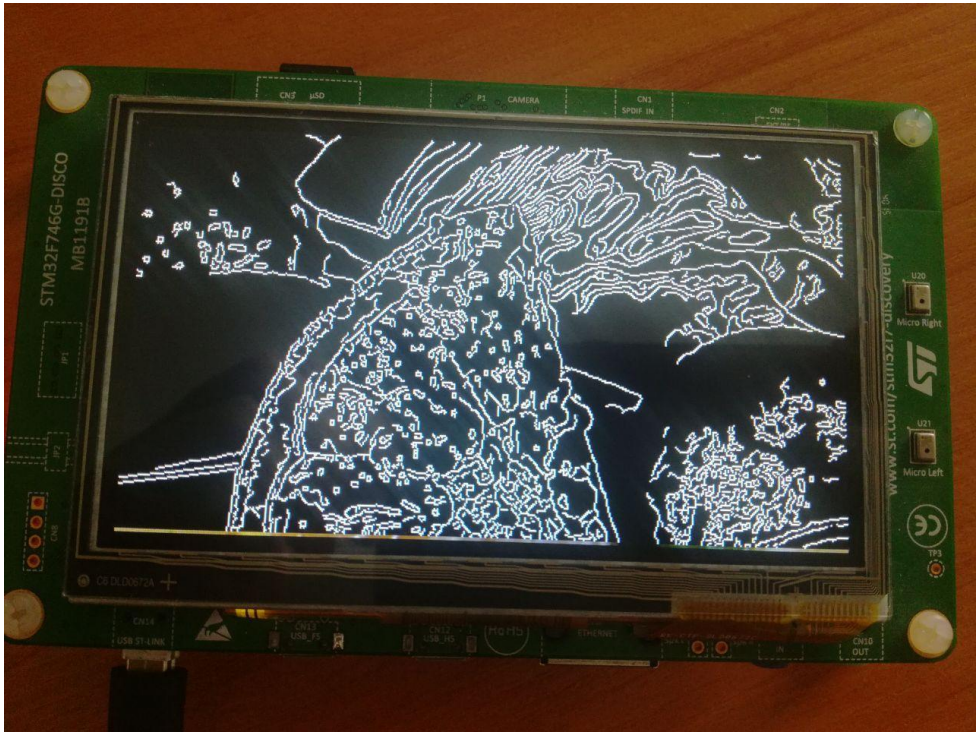
Пока идет сборка…
Типы памяти STM32F746G

- SRAM (Static RAM) - самая быстрая.
  - 1 Mbyte
- SDRAM (Synchronous Dynamic RAM)
  - 8 MBytes
- QSPI (Quad SPI) - самая медленная.
  - 16 Mbytes

Пока идет сборка... Qt.

- На самом деле всё что мы сейчас проделываем можно повторить и для многих других проектов. Пример - QT.

Запуск из QSPI на STM32F746G



Недостатки и преимущества запуска из внешней памяти (QSPI)

- Минусы:
  - Код исполняется медленно.
  - Сложность загрузки - нужна прошивка, например, через сеть (tftp).
  - Сложность отладки
- Плюсы:
  - Если код слишком большой, и его физически не “запихать” в SRAM, внешняя память может послужить альтернативой.

Запуск из SRAM на STM32F769I

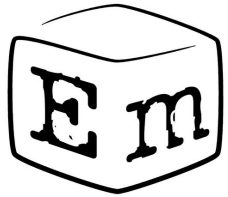
- Удалось разместить весь код и read-only данные в быструю память SRAM.
- Загрузчик не требуется, просто прошиваем плату и включаем.

Запуск из SRAM на STM32F769I

Итоги

- Запуск OpenCV и других больших библиотек возможен на микроконтроллерах.
- Да, библиотеки не имеют полного функционала как на десктопе, но его можно добавить в конфиг при необходимости.
- Узкое место оказывается не процессор, а память - нужно понимать что и в какой памяти находится (+ по возможности использовать DMA), грамотно управлять ресурсами.
- Кому интересно - попробуйте и на своих платформах :)

Контакты



Страница проекта

http://embox.github.io

Репозиторий проекта

https://github.com/embox/

embox-devel@googlegroups.com

embox-ru@googlegroups.com