Linaro connect
San Francisco 2017

Deploy STM32 family on Zephyr

Erwan Gouriou

LITE
IoT/EMBEDDED

life.augmented

Zephyr™

Linaro
connect
San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Plan

- A glance at STM32 ecosystem
- STM32 porting on Zephyr
- Device tree concept applied to STM32 family
- What’s next for STM32 in Zephyr

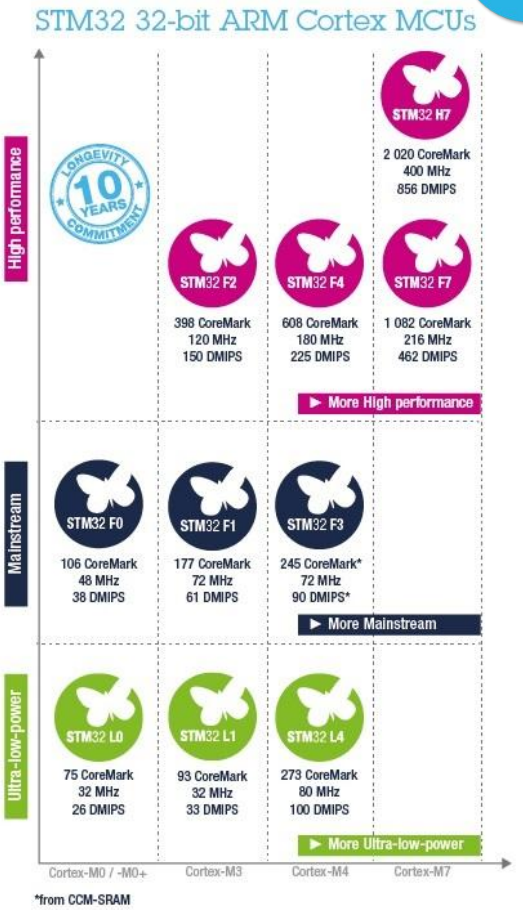
Linaro
connect
San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

A glance at STM32 Ecosystem

Quick outlook on STM32 SoC family

- Cortex-M based SoCs (M0 to M7)
- STM32 family encompasses today more than 700 SoC part numbers split in 10 series
- From Ultra Low Power (L: L0 >L4) to High Performance (F2 > H7)
- STM32 family keeps evolving (STM32H7) and more to come…
- http://www.st.com/en/microcontrollers/stm32-32-bit-arm-cortex-mcus.html



STM32 SoC naming convention

STM32 F 1 0 3 R B

Family
STM32 Cortex 32bit

Product type
- F Foundation
- L Ultra-low-power
- H High performance

Series:
- 0 Cortex-M0
- 1 Cortex-M3
- 2 Cortex-M3
- 3 Cortex-M4
- 4 Cortex-M4
- 7 Cortex-M7

Sub-Lines:
- 0 ---
- 1 ---
- 2 ---
- 3 --- Small die
- 4 ---
- 5 --- Medium die
- 6 ---
- 7 --- Big die
- 8 ---
- 9 --- High integration

A higher number means a richer configuration!

Lines :
- 0 Value
- 1 Access
- 2 USB
- 3 ---
- 4 ---
- 5 OTG
- 6 LCD/TFT+OTG
- 7 Ethernet
- 8 Regulator off
- 9 LCD TFT

Flash size (Kbytes) :

4	16	E	512
6	32	F	768
8	64	G	1024
B	128	H	1536
C	256	I	2048
D	384		

Pins:

F	20 pins	M	80 pins
G	28 pins	O	90 pins
K	32 pins	V	100 pins
T	36 pins	Q	132 pins
S	44 pins	Z	144
C	48 pins	I	176, 201 pins
R	64, 66 pins	B	208 pins
		N	216 pins

Linaro connect San Francisco 2017
ENGINEERS AND DEVICES WORKING TOGETHER

Focus on B-L475E-IOT01A (disco_l475_iot1)

USB OTG

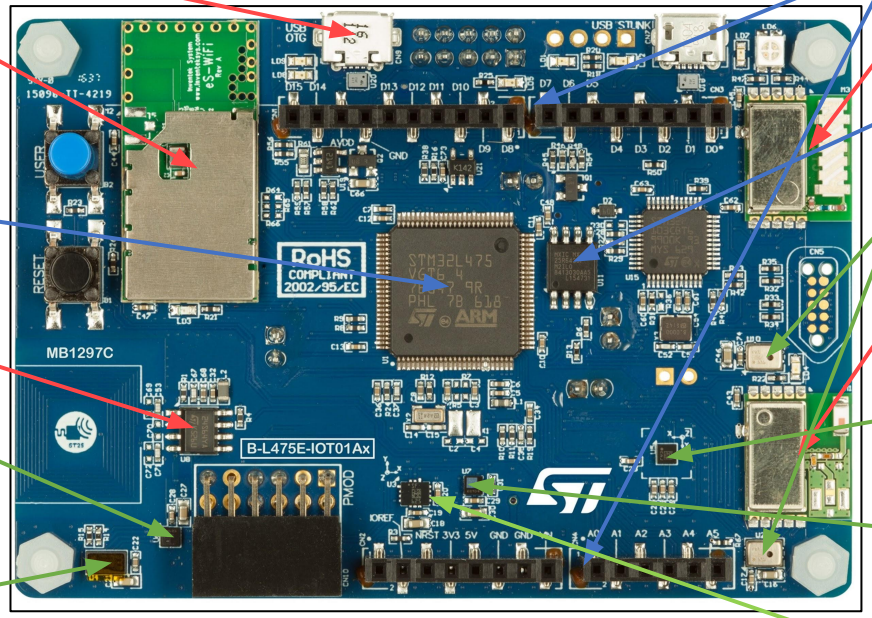
Inventek ISM43362-M3G: Wi-Fi Module

STM32L4 SoC: Low power, 80MHz 1Mbit flash 128 kRAM

M24SR: NFC Module

HTS221: Humidity and Temperature

VLX53L0X: Time of Flight and gesture recognition



Arduino V3 connector

SPSGRF:Sub-GHz Module

64MBit Quad-SPI Flash

MP34DT01: Omnidirectionnal mic (x2)

SPBTLE-RF: BT Module

LIS3MDL: 3-Axis magnetometer

LPS22HB: Barometer

LMS6DSL: 3D accelerometer and gyroscope

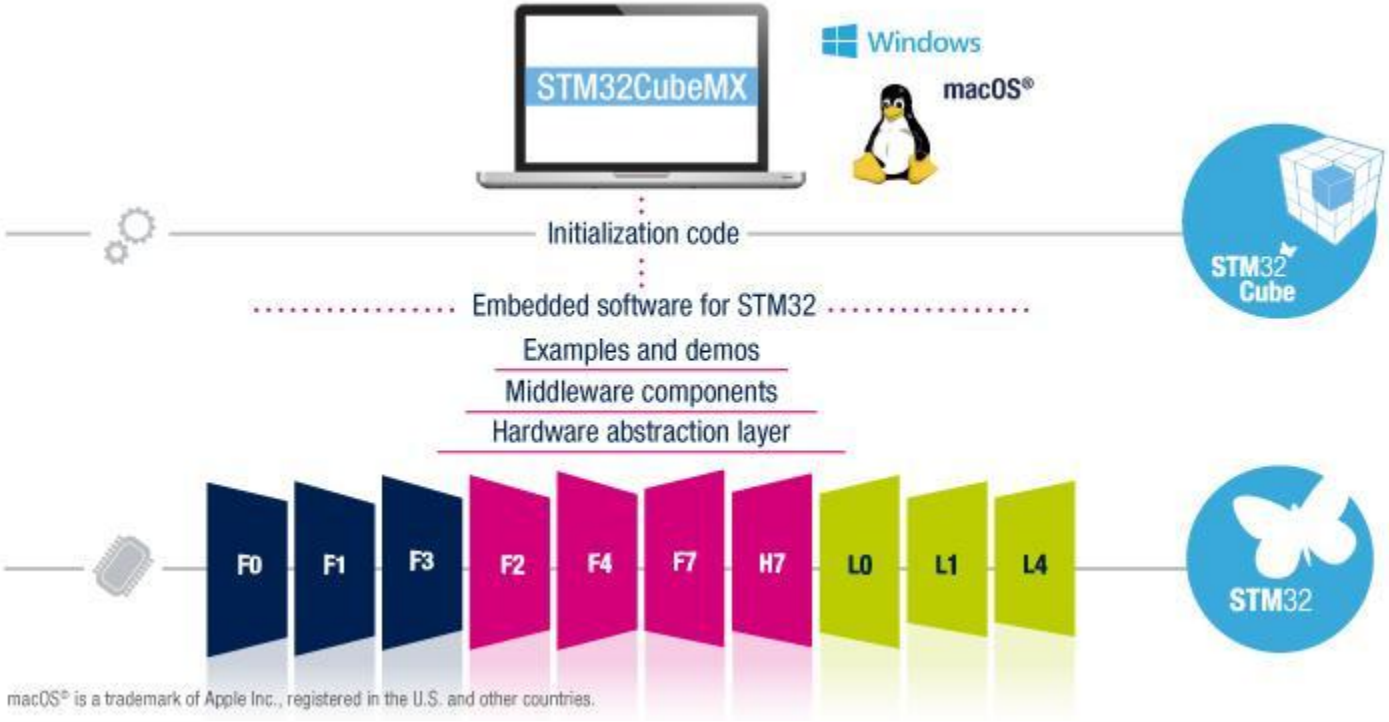
MCU

Sensing

Connectivity

Status:
Upstreamed
In progress

STM32Cube: Embedded Software for STM32



http://www.st.com/en/embedded-software/stm32cube-embedded-software.html

Main STM32Cube benefit: Abstraction

- Usual prejudices on HAL:
  - Size, sub optimization
  - coding style…
- Usual pro-HAL arguments are:
  - Avoid re-inventing the wheel, allow to save time…
  - Thoroughly maintained and validated, help developing more mature drivers
- **Abstraction capability breaks silos between series and re-inforces cooperation**
  - Focussed competencies: more users and more reviewers on a single driver
  - More minds: more performant drivers
  - Reduce work duplication: faster increase of driver support
  - Less code: maintenance effort is reduced.

STM32Cube APIs: HAL vs LL

- HAL (Hardware Abstraction Layers):
  - High level and functional abstraction
  - 100% coverage of all peripherals
  - Good match for complex drivers such as USB, Ethernet, ..
  - Higher portability => bigger software footprints, more time running adaptation code
- LL (Low-Layers):
  - Light-weight, optimized, expert oriented set of APIs designed for both performance and runtime efficiency
  - Register level access library. Do not use global variables. Atomic operations
  - Fit for most zephyr drivers and allows to build stm32 generic lightweight drivers.
  - Perfect fit for “simple IPs” drivers
- HAL/LL could be used together in a single driver

Linaro connect
San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

STM32 porting on Zephyr

STM32 SoC porting: Minimize code duplication

- Code organized per series
- SoC specific info (number of IRQ, SoC reference) is stored under Kconfig.defconfig.stm32yyyxx
- New “st_stm32/common” folder introduced to factorize cross-series code (MPU code for now)

- Introduction of device tree will further optimize SoC specific code

`arch/arm/soc/st_stm32`

- `common` (new! More code to be put here)
- `stm32f1`
  - `Kconfig.defconfig.stm32f103xx`
  - `Kconfig.defconfig.stm32f107xc`
  - …
- ….
- `stm32l4`

STM32Cube: Integration in Zephyr

- One package per family, stored under ext/hal/st/stm32cube
- Each package divided in 2 parts:
  - HAL and LL APIs (stm32cube/drivers)
  - CMSIS file (stm32cube/soc. Eg: stm32f103xe.h): sets of defines and registers structures
    - **Tip**: CMSIS files are useful to check SoC supported features and SoCs comparison
- Maintained and updated at each Zephyr DV
- STM32Cube, as an external library, should not be modified
- In case of genuine STM32Cube bug:
  - Report to STM32Cube Zephyr maintainer (issue will be raised in ST bugtracker)
  - Fix issue in a separate PR.
  - Reported issues are taken into account / Don’t hesitate to report issues!

How to: Introduce new LL/HAL driver

- HAL or LL:
  - Driver complexity?
  - Acceptable footprint?
  - Supported features?
  - Validation effort?
- Adapt code structure to IP diversity:
  - Maximize code reuse
    - Use CMSIS defines
    - Take benefit from porting APIs
  - Preserve readability

Small variations within family:

`spi_ll_stm32.c` (use of #ifdef's)

2 different IPs in whole family (v1/v2):

`i2c_ll_stm32.c`

`i2c_ll_stm32_v1.c` (I2C V1: F1/F2/F4/L1)

`i2c_ll_stm32_v2.c` (I2C V2: F0/F3/F7/L4)

Important heterogeneity:

`stm32_ll_clock.c`

`stm32f1x_ll_clock.c` (F1 series)

`stm32f3x_ll_clock.c` (F3 series)

...

- Lot of HAL/LL examples in stm32cube packages
- Community support on https://community.st.com

Linaro
connect
San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Device tree concept applied to STM32 family

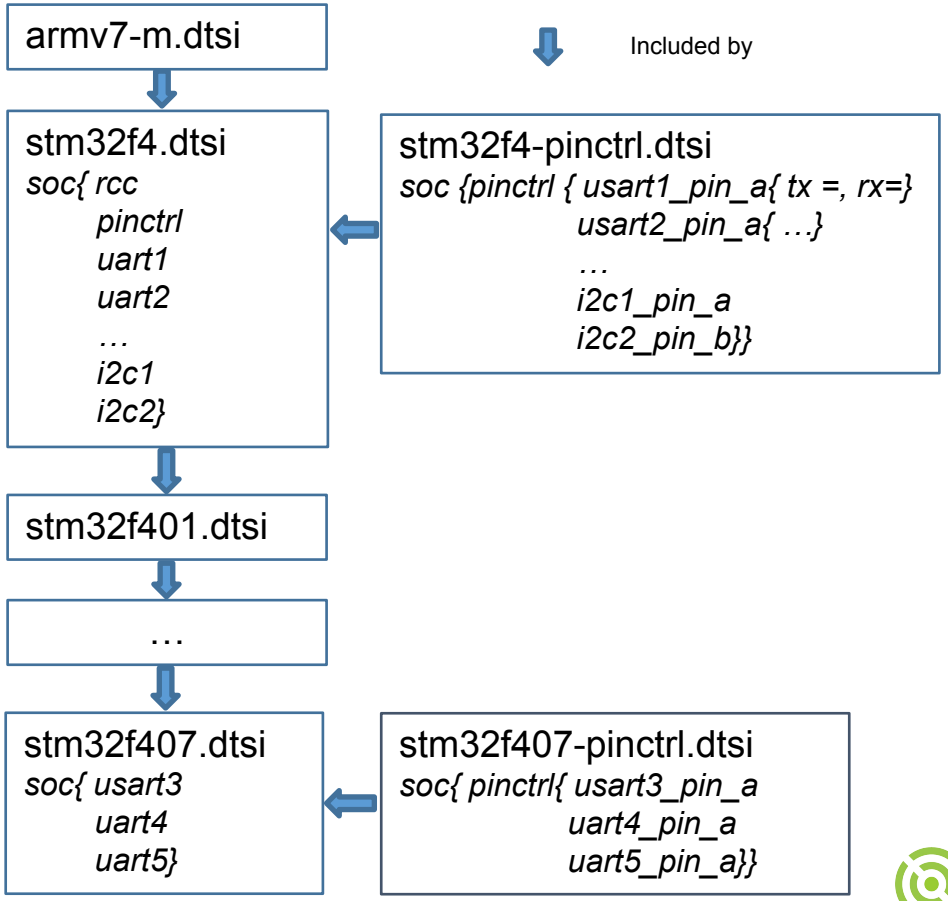
STM32 device tree: Zephyr vs Linux (1/2)

- STM32 dts files deployed in Zephyr and Linux in parallel
- No dependency today
- Long term goal: Common repo
- Respective evolutions monitored and controlled
  - Divergence points are listed
  - Regular alignments done
  - Linux kernel STM32 machine maintainer is reviewer of STM32 Zephyr dts PRs
- Today, alignments are only done in one way:
  - Linux => Zephyr
- Once device tree will be stabilized in Zephyr:
  - Linux <=> Zephyr

STM32 device tree: Zephyr vs Linux (2/2)

- Know Issues (and current solutions):
  - **Set of SoCs**:
    - More SoCs in Zephyr (were Linux won't fit)
    - Inheritance model should foresee this.
    - Current answer is to use generic stm32XX.dtsi files to store this minimum configuration
  - **Properties**:
    - Different drivers > Different properties
    - Current solution: Use of string prefix "zephyr," (Eg: "zephyr,bt-uart")
  - **Bindings**:
    - Different drivers > Different bindings
    - Current solution: Use same paths/same file names but different #define
- Upstream procedures for common repo are yet to be discussed

STM32 device tree organization



ENGINEERS
AND DEVICES
WORKING
TOGETHER

What's next for STM32 in Zephyr

- Keep extending STM32 drivers support
- Complete STM32 code factorization (still some duplicated code between series)
- Expand connectivity support
  - Add Wi-Fi support on Disco_L475_IOT
  - Add Sub-GHZ
  - NFC…
  - USB

Linaro
connect
San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Back up sildes

STM32 prototyping boards and shields

STM32 Nucleo development boards	Discovery kits	Evaluation boards	STM32 Nucleo expansion boards	Third-party boards
Flexible prototyping	Key feature prototyping	Full feature evaluation	Add-on functionalities	From full evaluation to open hardware

Status on STM32 presence in Zephyr (17'09)

- 4 series present (F1, F3, F4, L4)
- 21 SoCs ported
- 22 Boards available

Legend:

Available

Issues reported

Z: Zephyr native
LL: Based on STM32CUBE LL
HAL: Based on STM32CUBE HAL

	GPIO/ Pinmux	RCC	UART	I2C	SPI	RNG	PWM	Flash	DMA	EXTI	ETH
stm32f1 4 socs 4 boards	Z	LL	HAL/LL	LL	LL	LL	HAL	NA	NA	Z	NA
stm32f3 3 socs 3 boards	Z	LL	HAL/LL	LL	LL	LL	HAL	Z	NA	Z	NA
stm32f4 10 socs 11 boards	Z	LL	HAL/LL	LL	LL	LL	HAL	Z	Z	Z	HAL
stm32l4 4 socs 4 boards	Z	LL	HAL/LL	LL	LL	LL	HAL	Z	NA	Z	NA

Linaro
connect
San Francisco 2017

Zephyr STM32 Misc

- BT available with X-Nucleo shield or embedded on Disco IoT Board
  - Activate with CONFIG_BT_SPI_BLUENRG

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Thank You

Contact: erwan.gouriou@linaro.org

#SFO17

SFO17 keynotes and videos on: connect.linaro.org

For further information: www.linaro.org