PRIMENA MIKROKONTROLERA-MS1PMK

1. deo

2017

Nenad Jovičić

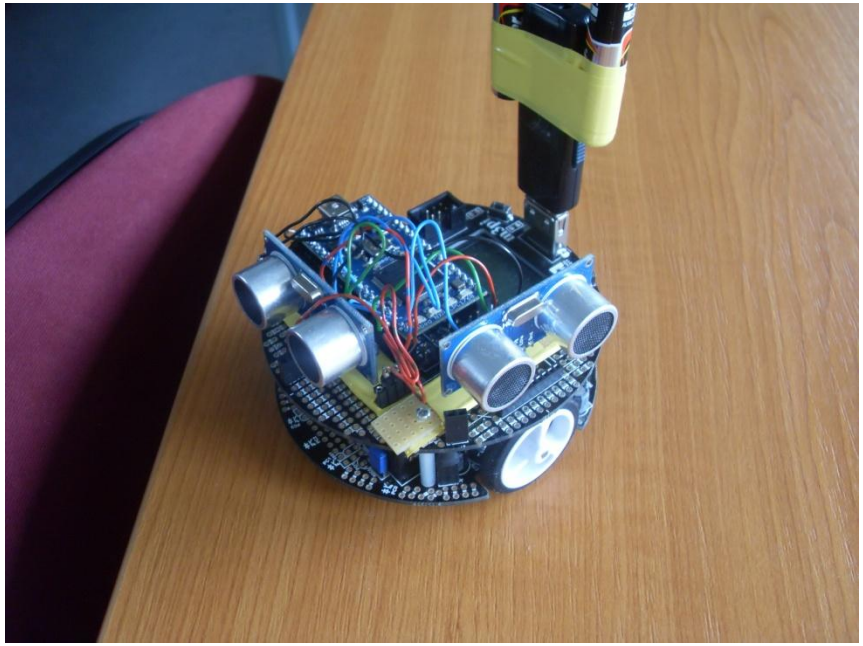
Organizacija kursa

- **Predavači**
  - Nenad Jovičić (nenad@etf.rs)
  - Marija Janković (meja@el.etf.rs)
- **Ocenjivanje (6–50^{+}, 7-60^{+}, …)**
  - 20% kolokvijum (krajem semestra)
  - 20% rad u laboratoriji tj. prisustvo na času
  - 60% projekat – demonstracija + diskusija – ovo je praktično ispit
- **Web**
  - tnt.etf.rs/~ms1pmk

Predmet kursa

- ARM Cortex Mikrokontroleri – uopštena priča
- Mbed.org – rapid prototyping pristup – kako da brzo napravimo prototip
- Klasično programiranje korišćenjem HAL drajvera koje obezbeđuje proizvođač mikrokontrolera – pravimo serijski proizvod
- Osvrt na niže i više nivoe – ARDUINO i RASPBERRY PI

Projekat



Projektni izveštaj

- Jednostavniji projekti se rade samostalno, a komlikovaniji grupno.

BLUETOOTH CONTROL OF AUTOMATED GUIDED VEHICLE USING MBED MICROCONTROLLER

Filip Vranić, Želimir Jovašević, Vladimir Petrović, Milana Milošević
University of Belgrade, School of Electrical Engineering, Belgrade, Serbia

INTRODUCTION

Vehicle control over Bluetooth interface is presented. Locomotion concept of the robot is very simple. If the right track moves forward or backward with equal speed as left track, vehicle will go straight forward or straight backward, respectively. If the right track has larger speed than the left one, robot will turn to the left. Same principle is used, when robot is turning to the right (left track speed > right track speed).

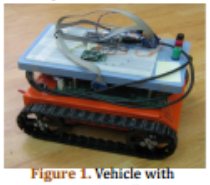


Figure 1. Vehicle with microcontroller

SYSTEM ARCHITECTURE

Radio control of track robot is implemented on mbed™ NXP LPC1768 microcontroller. Each track consists of two wheels, and each track is actuated with Hennkwell HG37D670WE12-052 DC motor with built-in encoders. Two Sharp GP2D12 infrared sensors have been incorporated in the robots design for proximity detection. Control and communication is achieved wirelessly via BlueMatic™ Bluetooth module.
Robot control consists of the instructions that control robot's actions and provide information regarding required tasks. Movement is achieved over forward/backward and left/right commands.

COMMUNICATION AND CONTROL

Commands are issued over keyboard (4 buttons). Combination of pressed buttons is coded into a string and transmitted over Bluetooth to the controller which drives robot's motors. Based on the received string and a current state of the system reference and control signals are determined and applied.
Motor control is conducted in open-loop mode, by setting the reference as control signal. Reference range varies from 0 to 100% (100% reference corresponds to 0.4 PWM duty cycle).
Software control loop is shown at Figure 2.

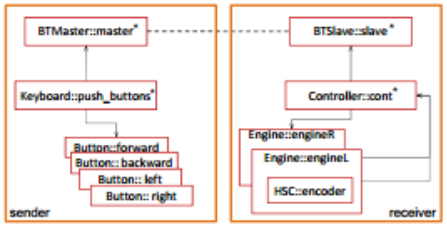


Figure2. UML Class diagram of software control loop

PROXIMITY SENSOR

Robot is equipped with infrared proximity sensor. Based on the readings of this sensor, when it reaches 20 or fewer centimeters, robot will stop and thus prevent collision with objects. Main defect of this solution is sensor's narrow FOV (field of view) which is only 6 degrees and prevents detection of narrow object such as chair and table legs etc. More adequate solution would probably be use of ultrasonic sensors which have up to 5 times wider FOV.

SYSTEM IDENTIFICATION

System identification was performed to model motor speed with PWM duty cycle of 40%. Method of multiple integrations was used and the results are given in Table 1.
Reference increase from 40% to 100% with 20% step and subsequent measurement and identification proved this system to be nonlinear. From this point speed control could be achieved with nonlinear controller or with linear control over small area near nominal value. Close-loop control should be considered to improve achieved results.

RESULTS

Resulting transfer functions of system identification are given in Table 1. Graphic results of identified system, speed measurements and speed reference are given at Figure 3.

Transfer function	Left track	Right track
Forward	3.3845 / (s + 3.433)	3.1801 / (s + 3.219)
Backward	3.1916 / (s + 3.304)	3.3809/ (s + 3.411)

Table 1. Identified transfer functions

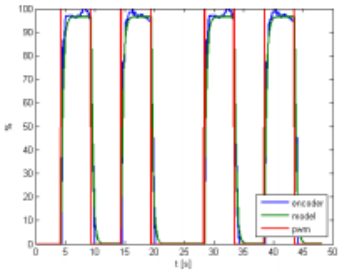


Figure 3. System identification. PWM control signal (red) and encoder reading (blue) serve as inputs into multiple integrators identification system. First order model (green) is a result of identification

Komunikacija...

- Molim sve koji će pratiti predmet da pošalje email na adresu nenad@etf.rs sa subject-om MS1PMK2016.

Literatura

- MBED.ORG
- ARM System-on-Chip Architecture, Steve Furber, Addison-Wesley, 2000.
- ARM System Developer's Guide: Designing and Optimizing System Software, *Andrew N. Sloss, Dominic Symes, Chris Wright and John Rayfield,* Elsevier, 2004.
- The Definitive Guide to the ARM Crotex-M3, *Joseph Yiu,* Elsevier, 2010.
- **The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors, Third Edition,** *Joseph Yiu,* Elsevier, 2013.

Literatura – ARM-ova dokumentacija

- ARMv7-M Architecture Reference Manual (ARM DDI 0403)
- ARM Cortex-M3 Integration and Implementation Manual (ARM DII 0240)
- ARM AMBA® 3 AHB-Lite Protocol (v1.0) (ARM IHI 0033)
- ARM AMBA™ 3 APB Protocol Specification (ARM IHI 0024)
- AMBA® 3 ATB Protocol Specification (ARM IHI 0032)
- ARM CoreSight™ Components Technical Reference Manual (ARM DDI 0314)
- ARM Debug Interface v5 Architecture Specification (ARM IHI 0031)
- ARM Embedded Trace Macrocell Architecture Specification (ARM IHI 0014).
- IEEE Standard *Test Access Port and Boundary-Scan Architecture* 1149.1-2001 (JTAG).
- ...

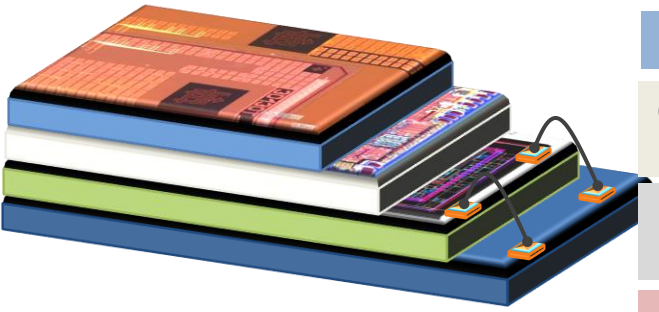
ARM

- Holding osnovan 1990 godine, kao zajedničko preduzeće kompanija Accorn, Apple i VLSI Technology.
- ARM ne proizvodi čipove već se bavi dizajnom arhitekture i procesora, tj. IP-a.
- Skoro svi vodeći proizvođači elektronskih komponenti otkupljuju licence od ARMa i na osnovu njih proizvode svoje varijante mikrokontrolera i mikroprocesora.
- Osnovna ideja je da procesorsko jezgro bude standardizovano (ARM), a da svaki proizvođač dodaje svoje specifične periferije.

ARM partneri

Aplikacije

Najmanji ARM računar



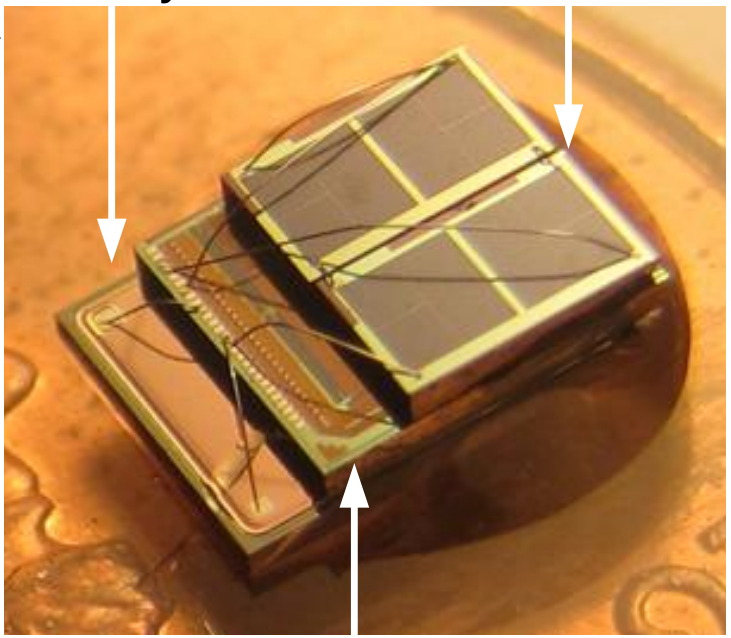
Wireless Sensor Network

Sensors, timers

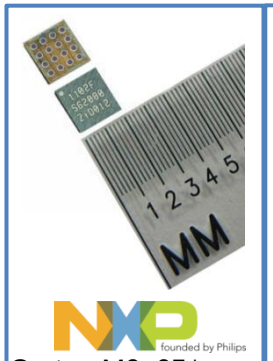
Cortex-M0 +16KB RAM 65nm
UWB Radio antenna

10 kB Storage memory ~3fW/bit

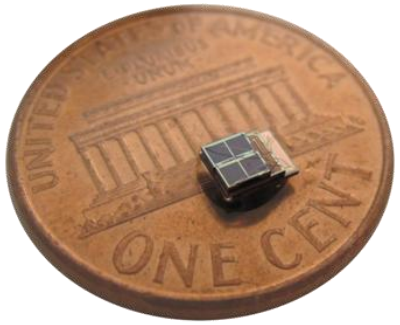
12µAh Li-ion Battery



Processor, SRAM and PMU



Cortex-M0; 65¢



University of Michigan

Wirelessly networked into large scale sensor arrays

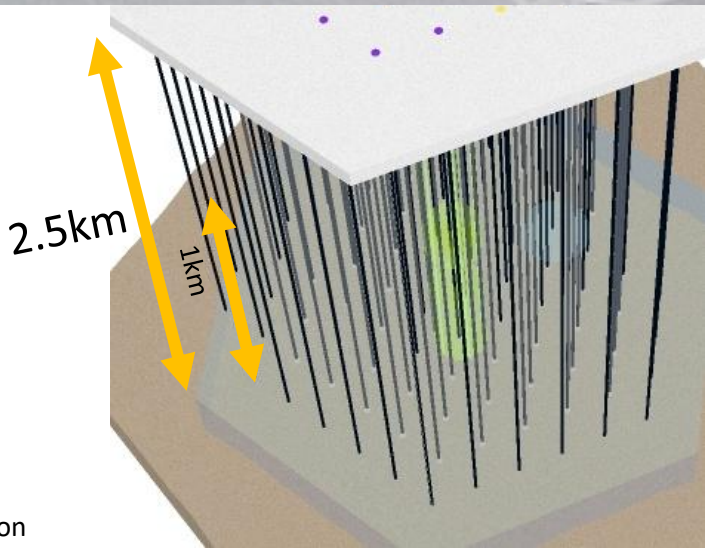
Najveći ARM računar

4200 ARM powered
Neutrino Detectors

70 bore holes 2.5km deep

60 detectors per string
starting 1.5km down

1km^3 of active telescope



Work supported by the National Science Foundation and University of Wisconsin-Madison

Udeo na tržištu 2012.

Markets for ARM in 2012

	Devices Shipped (Million of Units)	2012 Devices	Chips/ Device	TAM 2012 Chips	2012 ARM	2012 Share
Mobile	Smart Phone	730	3-5	2,500	2,200	90%
	Feature Phone	460	2-3	1,200	1,100	95%
	Low End Voice	730	1-2	730	700	95%
	Portable Media Players	130	1-3	250	220	90%
	Mobile Computing* (apps only)	400	1	400	160	40%
Home	Digital Camera	150	1-2	230	180	80%
	Digital TV & Set-top-box	420	1-2	640	290	45%
Enterprise	Desktop PCs & Servers (apps)	200	1	200	-	0%
	Networking	1,200	1-2	1,300	420	35%
	Printers	120	1	120	85	70%
	Hard Disk & Solid State Drives	700	1	700	620	90%
Embedded	Automotive	2,600	1	2,600	210	8%
	Smart Card	6,000	1	6,000	710	13%
	Microcontrollers	8,700	1	8,700	1,500	18%
	Others **	2,000	1	2,000	300	15%
	Total	**25,500**		**27,000**	**8,700**	**32%**

Year	Market Share
2007	17%
2008	20%
2009	22%
2010	25%
2011	29%
2012	32%

Source: Gartner, IDC, SIA, and ARM estimates

* Including tablets, netbooks and laptops

** Includes other applications not listed such as headsets, DVD, game consoles, etc

Planovi za 2017.

Markets for ARM in 2017

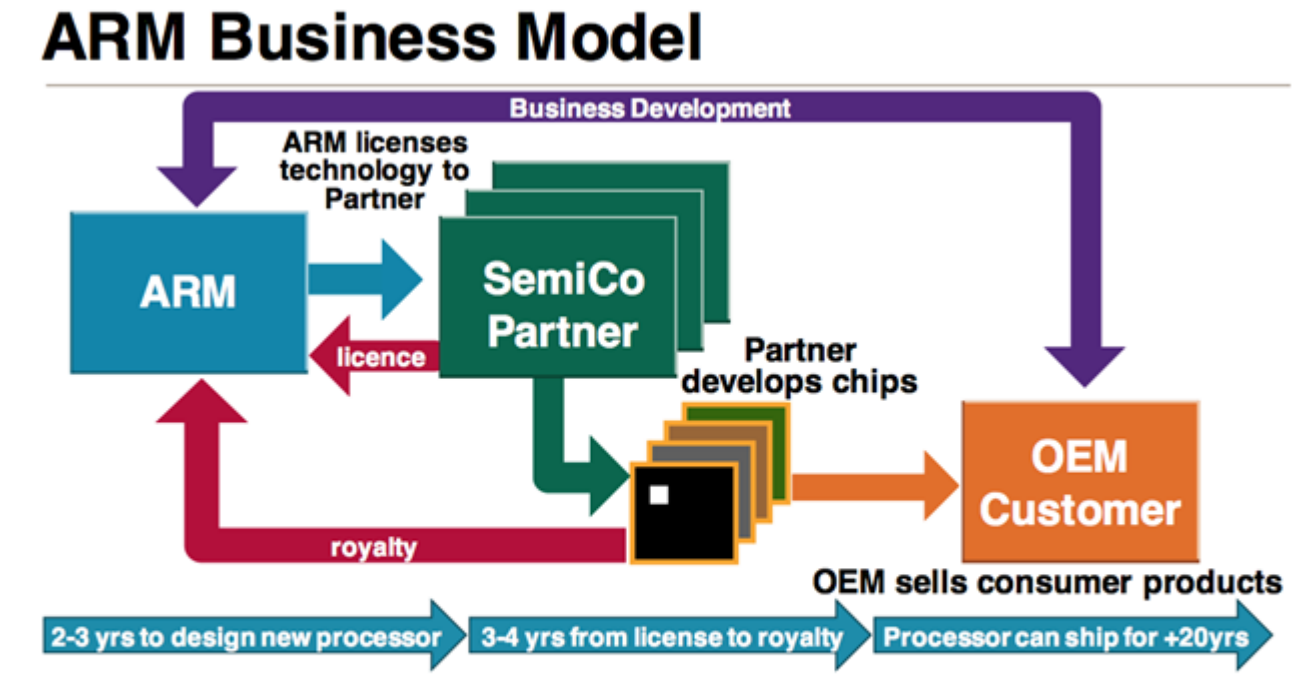
	Devices Shipped (Million of Units)	2017 Devices	Device CAGR	Chips/ Device	2017 Chips	Chip CAGR	Key Growth Areas for ARM
Mobile	Smart Phone	1,700	20%	3-5	6,800	20%	←
	Feature Phone	-	-	-	-	-	
	Low End Voice	710	-1%	1-2	1,400	15%	
	Portable Media Players	90	-10%	1-3	180	-5%	
	Mobile Computing* (apps only)	850	20%	1	850	20%	←
Home	Digital Camera	130	-5%	1-2	200	-5%	
	Digital TV & Set-top-box	600	10%	1-4	2,000	25%	←
Enterprise	Desktop PCs & Servers (apps)	200	Flat	1	200	Flat	
	Networking	1,500	5%	1-2	1,700	5%	←
	Printers	130	2%	1-3	130	2%	
	Hard Disk & Solid State Drives	1,100	10%	1	1,100	10%	
Embedded	Automotive	3,800	10%	1	3,800	10%	
	Smart Card	8,500	10%	1	8,500	10%	
	Microcontrollers	11,400	5%	1	11,400	5%	←
	Others **	3,000	10%	1-2	3,000	10%	
	Total	**34,000**	**5%**		**41,000**	10%	

Source: Gartner, IDC, SIA, and ARM estimates

* Including tablets, netbooks and laptops

** Includes other applications not listed such as headsets, DVD, game consoles, etc

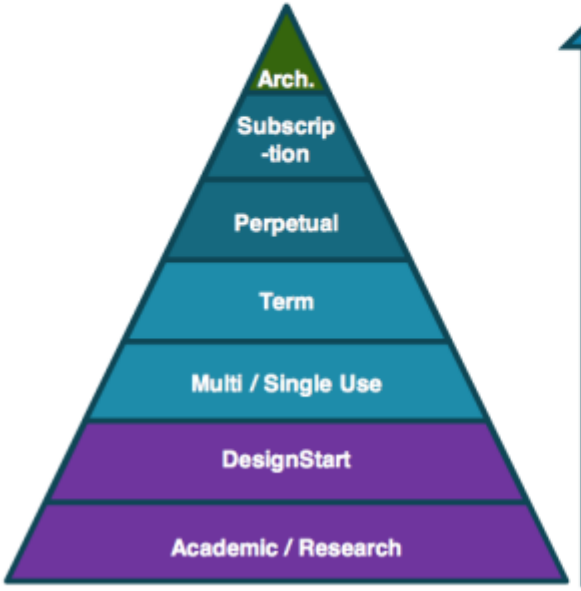
Kako ARM posluje?



Kakve sve licence postoje?

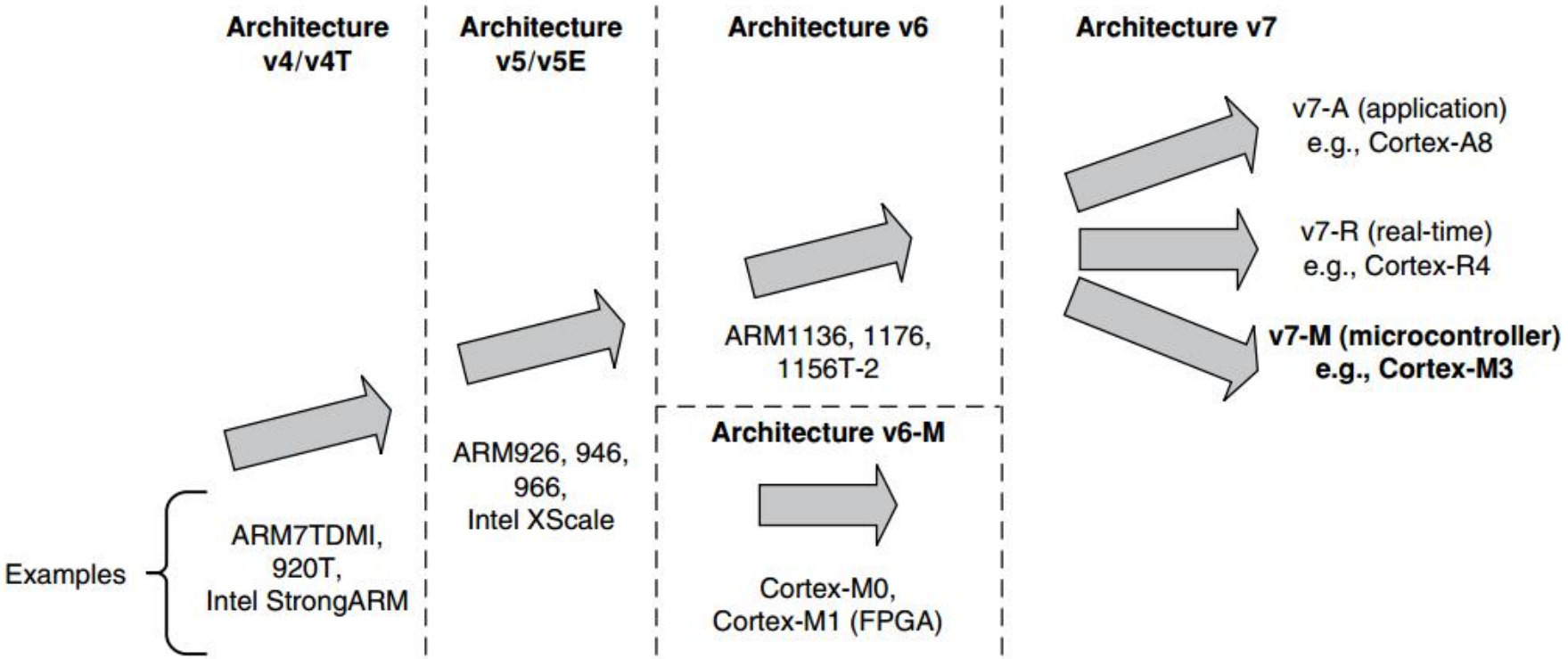
Broad Range of Licensing Options

- Innovation in the business model as well as technology
  - New partnerships for new types of companies



- Higher license fee
- Fewer use constraints
- Greater commercial and technical interaction
- Increasing internal investment by Partner
- Increasing ARM / Partner business knowledge

ARM – evolucija arhitektura



Cortex familije

ARM Cortex-**A** family (v7-A):

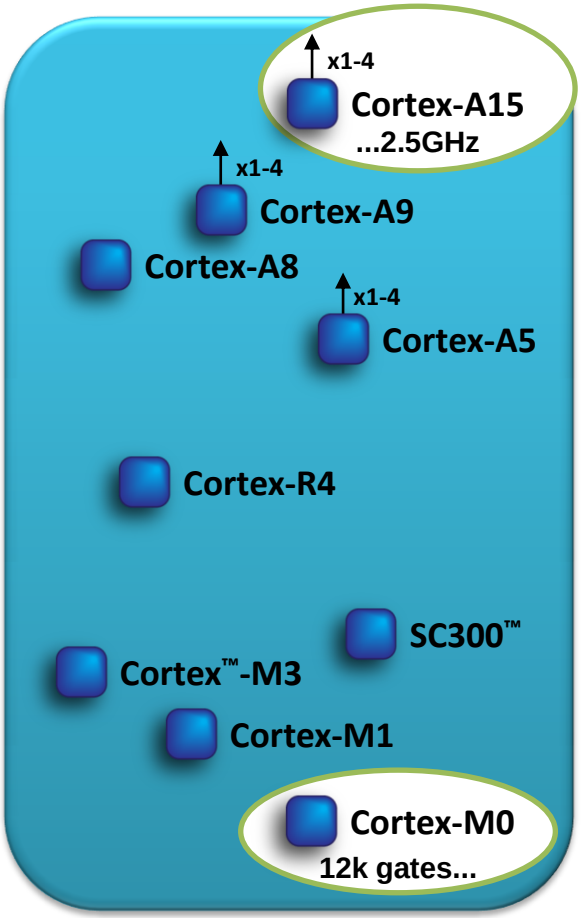
Applications processors for full OS and 3rd party applications

ARM Cortex-**R** family (v7-R):

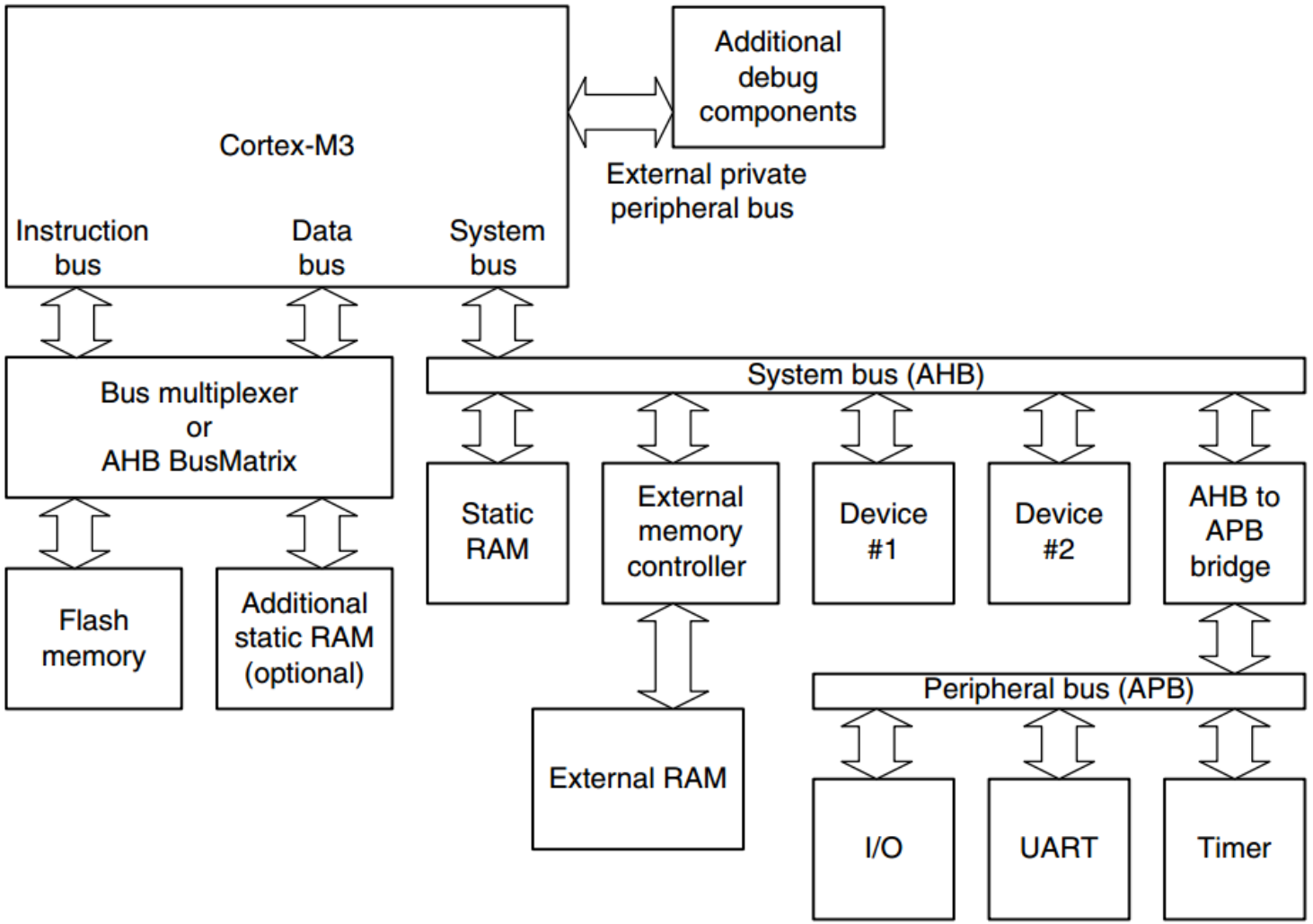
Embedded processors for real-time signal processing, control applications

ARM Cortex-**M** family (v7-M):

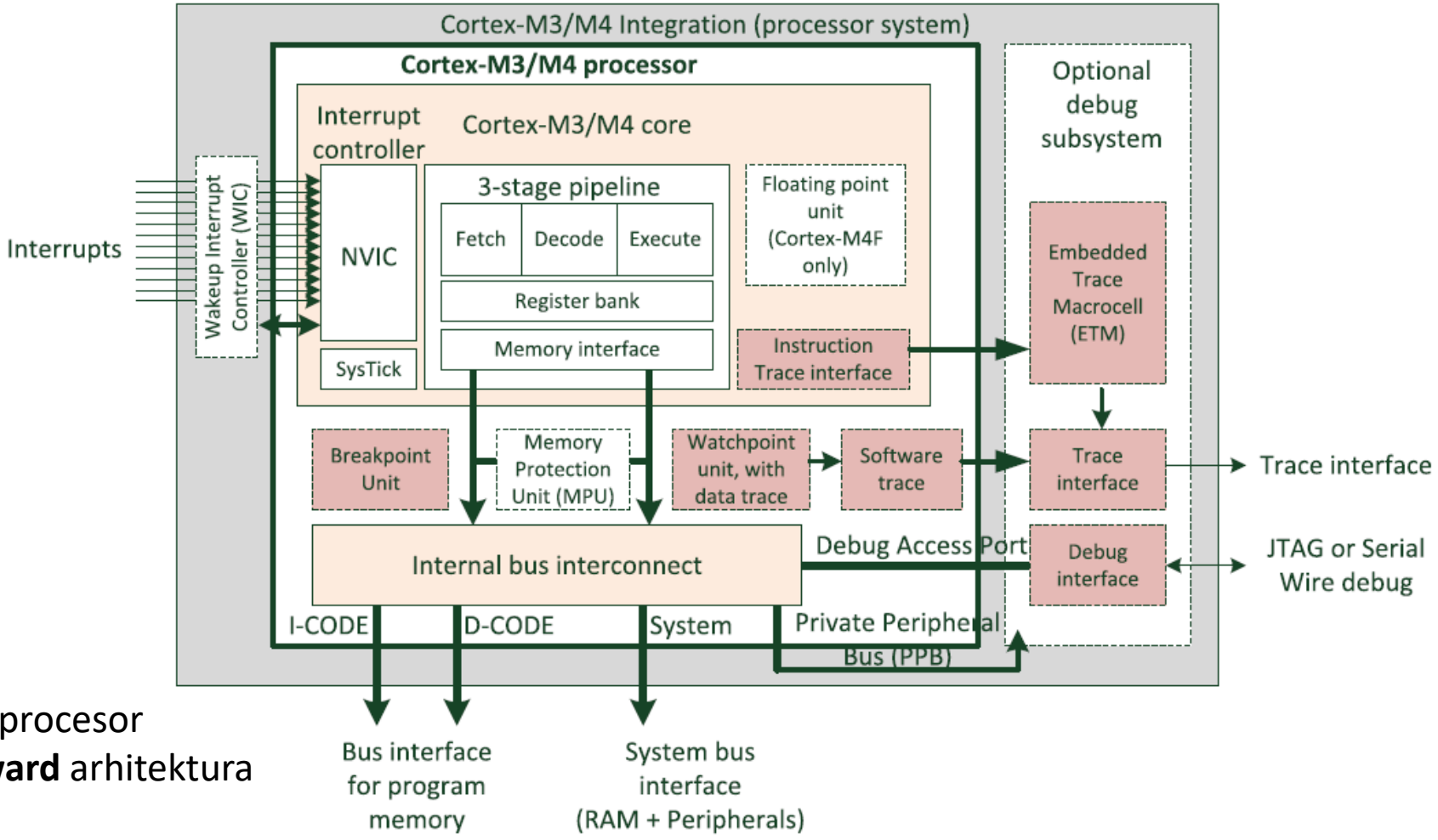
Microcontroller-oriented processors for MCU and SoC applications



Tipična arhitektura Cortex-Mx mikrokontrolera



Cortex-M3/M4 – procesori bazirani na ARMv7-M arhitekturi



RISC procesor
Harward arhitektura

ARM Cortex-M3/M4 tehnologije

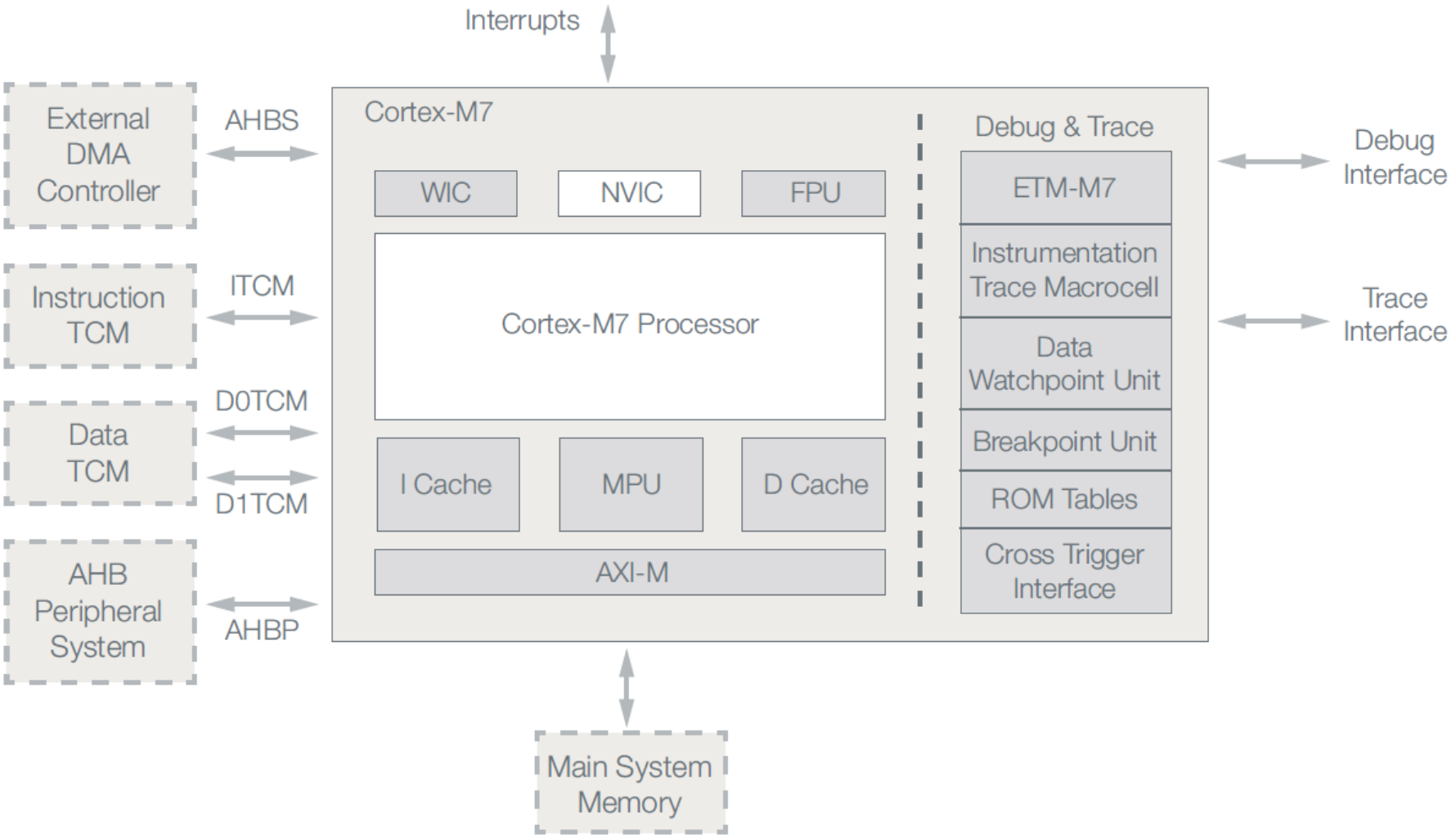
- Svaka od Cortex-M serije procesora je specifična, ali poseduje i sledeće zajedničke karakteristike:

RISC procesorsko jezgro	**Thumb-2® instukcijski set**
•32-bit CPU •Predvidiv rad •3-stepena protočna obrada	•Kompromis između 16/32-bitnih instrukcija •3x manja veličina koda nego kod 8-bitnih platformi •Bez negativnog uticaja na performanse
Modovi rada sa smanjenom potrošnjom	**Nested Vectored Interrupt Controller (NVIC)**
•Integrisani modovi rada •Više modova potrošnje •Softverska kontrola	•Malo kašnjenje servisiranja prekida •Bez potrebe za programiranjem u asembleru •Servisne rutine u C kodu
Alati i RTOS	**Podrška za debagovanje**
•Široka lepeza alata •Cortex Microcontroller Software Interface Standard •Software reuse podrška	• JTAG ili 2-pin Serial Wire Debug •Podrška za višeprocesorki rad •Debagovanje u realnom vremenu

Cortex-M3/M4 osnovne karakteristike

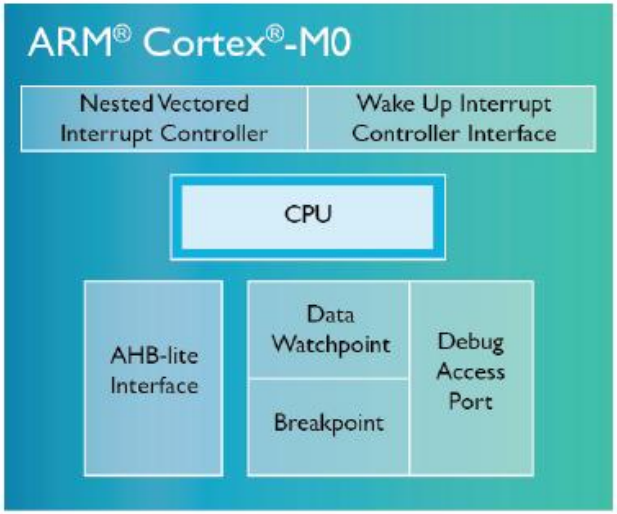
Arhitektura	ARMv7-M (Harvard)
Instrukcijski set	Thumb® / Thumb-2
Protočna obrada	3-stepena + spekulacija skoka
***Dhrystone* test**	1.25 DMIPS/MHz
MPU	Optional 8 region
Prekidi	NMI + 1 to 240 physical interrupts
Kašnjenje prekida	12 taktnih ciklusa
Kašnjenje gnježdenja prekida	6 taktnih ciklusa
Nivoi prioriteta prekida	8 do 256 nivoa prioriteta
Prekidni kontroler	Do 240 prekida
Redukcija potrošnje	Sleep/deep sleep modovi rada
Bit-operacije	Integrisane instrukcije (atomske)
Posebne instrukcije	HW deljenje (2-12 ciklusa) & množenje (32x32) 1 ciklus.
Podrška za debagovanje	Opciono JTAG & Serial-Wire Debug portovi.

Novo – Cortex-M7

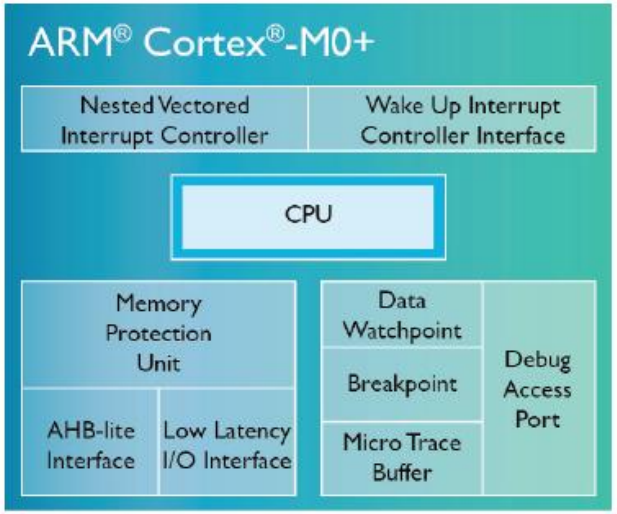


Cortex-M7 osnovne novine

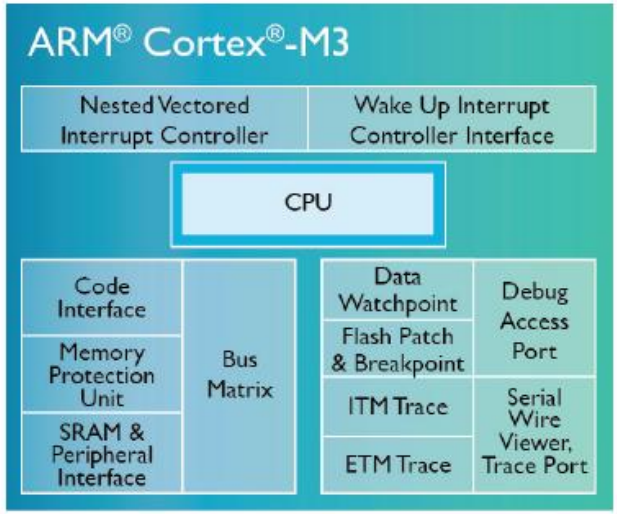
- 6-stage pipeline
- AXI (advanced extensible interface) magistrala
- Keš za podatke i instrukcije prema AXI magistrali
- 64-bitne magistrale
- ITCM - Instruction thightly coupled memory
- DTCM – Data thightly coupled memory
- I dalje podržan AHB interfejs zbog kompatibilnosti



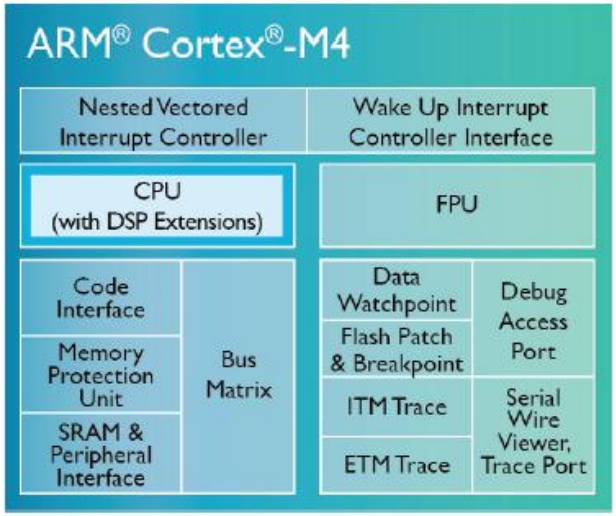
Lowest cost
Low power



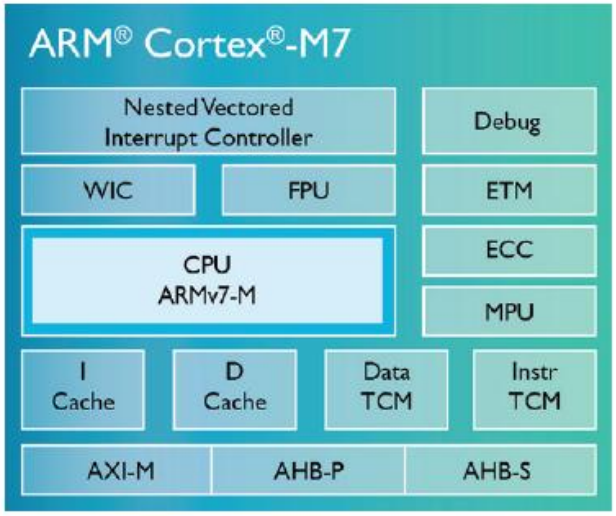
Lowest power
Outstanding energy efficiency



Performance efficiency
Feature rich connectivity

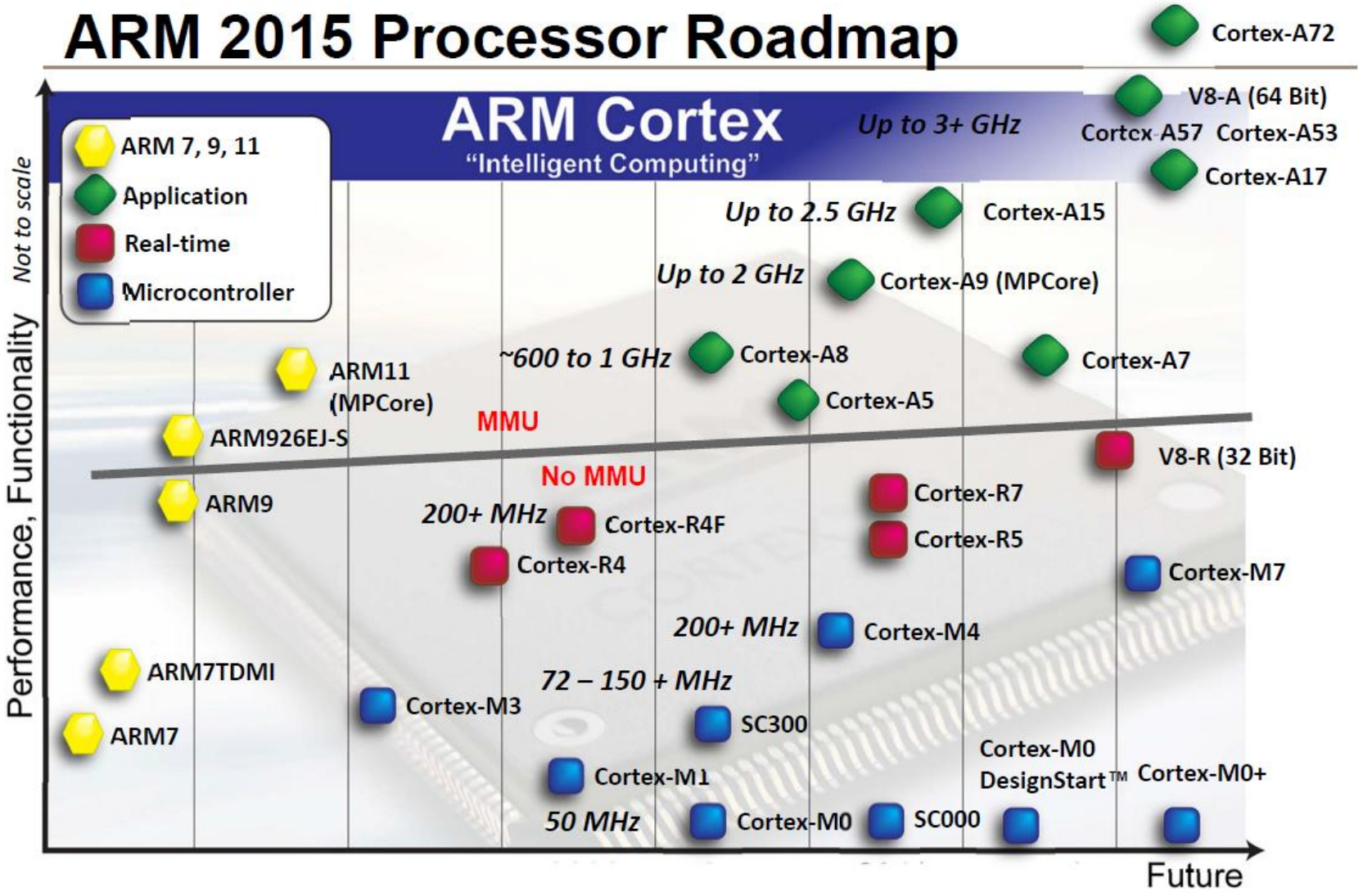


Digital Signal Control (DSC)
Processor with DSP
Accelerated SIMD
Floating point (FP)



Maximum DSC Performance
Flexible Memory System
Cache, TCM, AXI, ECC
Double & Single Precision FP

ARM 2015 Processor Roadmap



RISC-V

- Moguća alternativa
- Otvorena RISC platforma
- University of California, Berkeley.
- http://riscv.org/

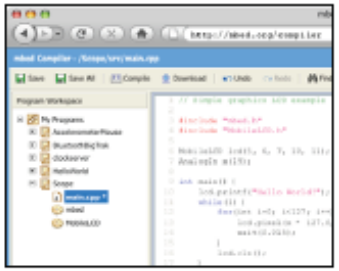
	ARM Cortex-A5	RISC-V Rocket	Ratio
Width (bits)	32	64	2x
Frequency	>1GHz	>1GHz	1x
Dhrystone Performance (Dmips/MHz)	1.57	1.72	1.1x
Area (no caches)	0.27mm^2	0.14mm^2	0.5x
Area (16 KB caches)	0.53mm^2	0.39mm^2	0.7x
Area Efficiency (Dmips/MHz/mm^2)	3.0	4.4	1.5x
Dynamic Power (mW/MHz)	<0.080	0.034	≥0.4x

mbed (http://mbed.org/)

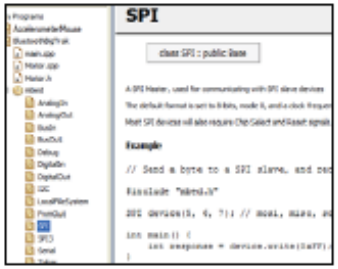
mbed

- **Getting Started and Rapid Prototyping with ARM MCUs**
  - Complete Targeted Hardware, Software and Web 2.0 Platform

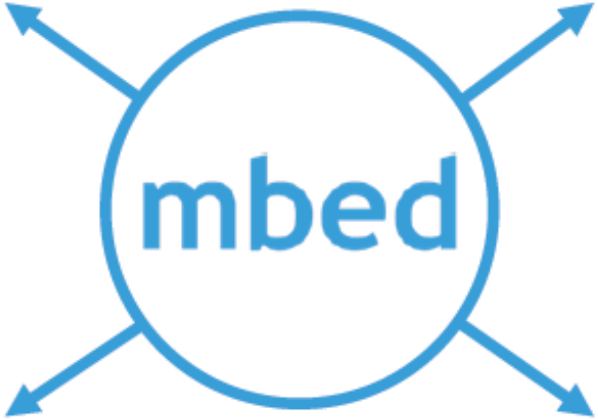
Dedicated Developer Web Platform



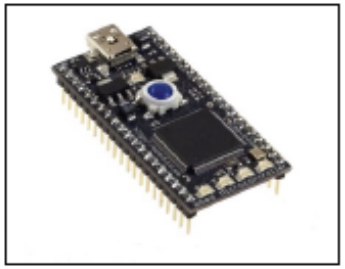
Lightweight Online Compiler



High-level Peripheral APIs



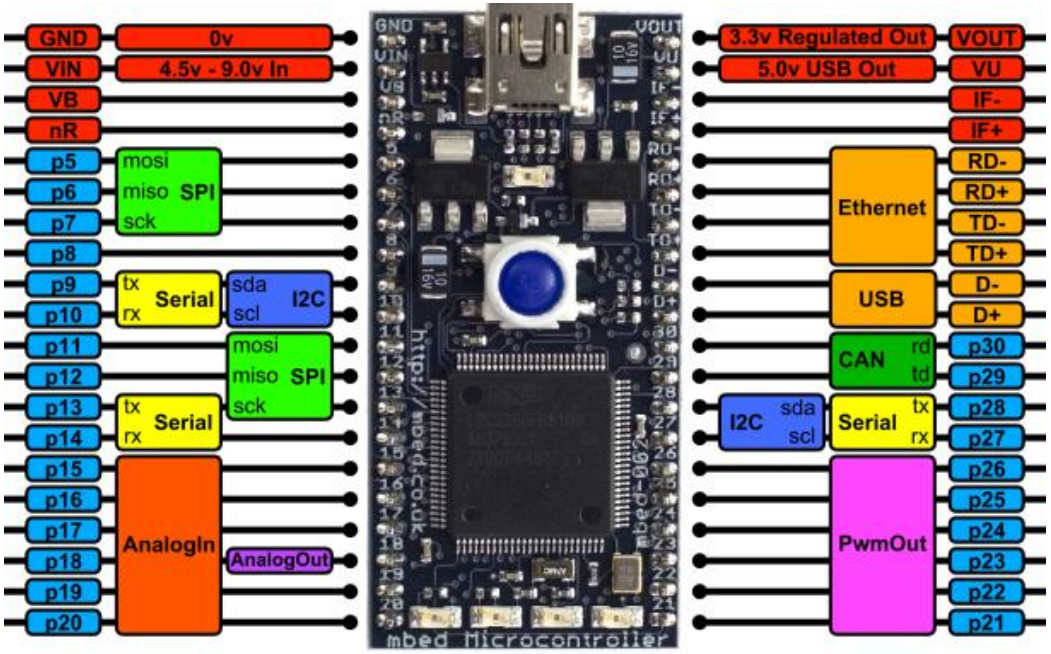
Rapid Prototyping for Microcontrollers



LPC Cortex-M MCU in a Prototyping Form-Factor

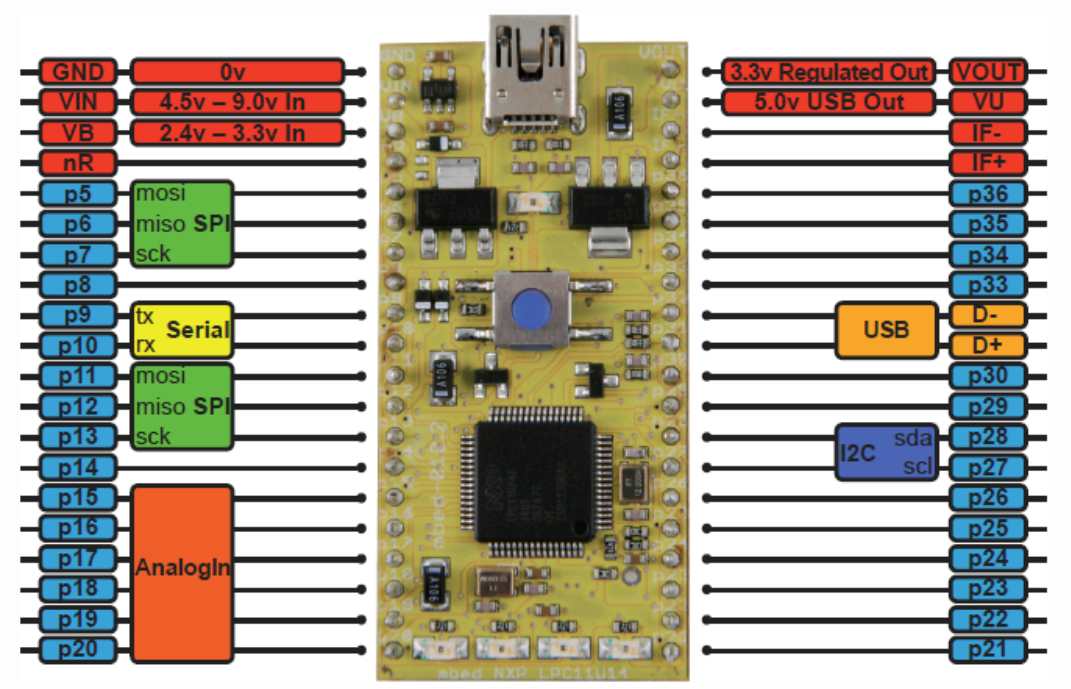
MBED – LPC1768

- MBED – LPC1768 je razvojna pločica namenjena za brzi razvoj aplikacija na ARM CORTEX M3 mikrokontroleru LPC1768.



MBED – LPC11U24

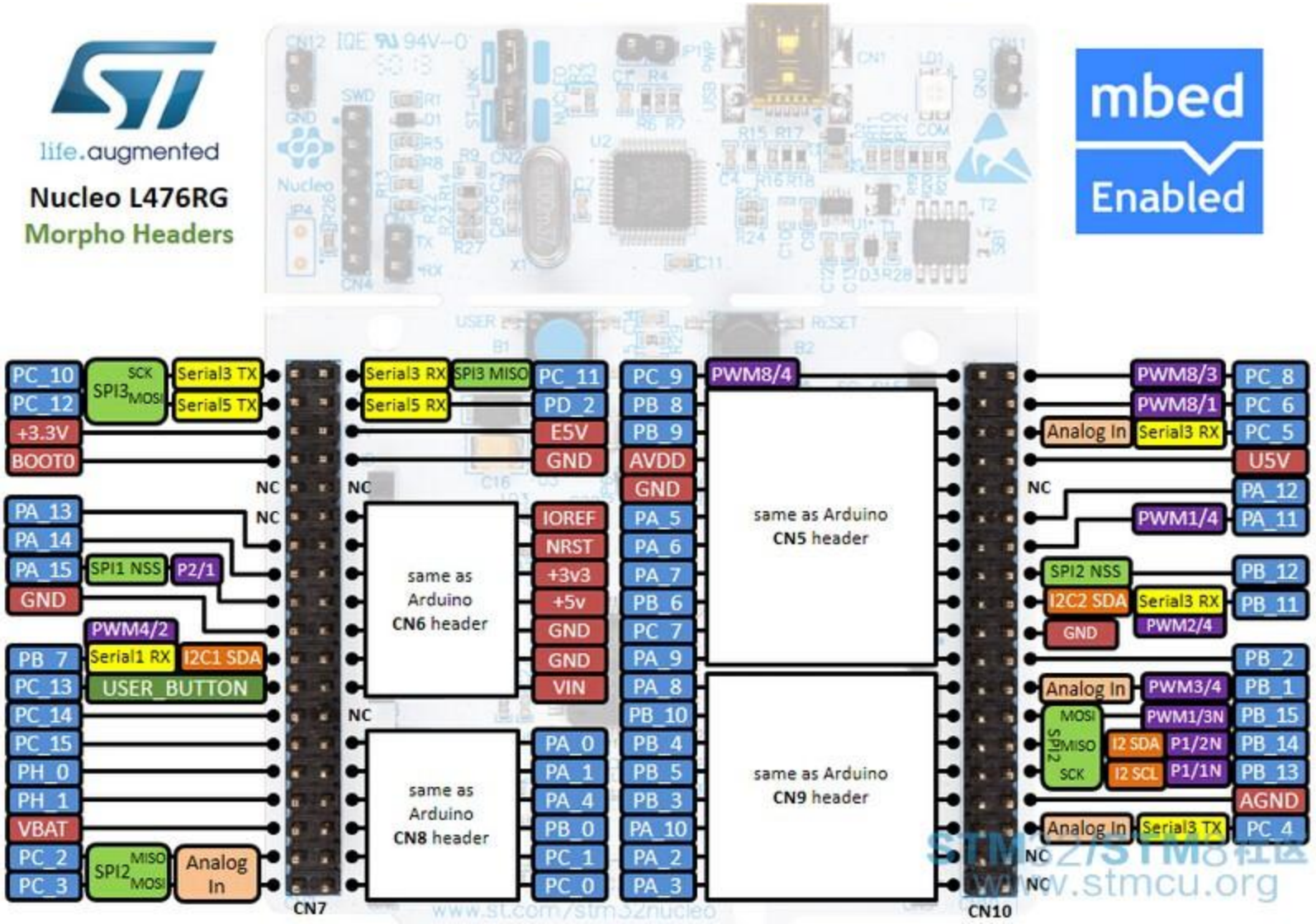
- MBED – LPC11U24 je razvojna pločica namenjena za brzi razvoj aplikacija na ARM CORTEX M0 mikrokontroleru LPC11U24.



ST Nucleo L476

- ARM Cortex-M4 CPU with FPU at 80MHz
- 1 MB Flash
- 128 kB RAM
- 3× 12-bit ADC 5 MSPS
- 2x 12-bit DAC
- Up to 16 timers
- Up to 3x I2C, 6x USARTs, 4x SPIs, CAN
- USB OTG 2.0 full-speed

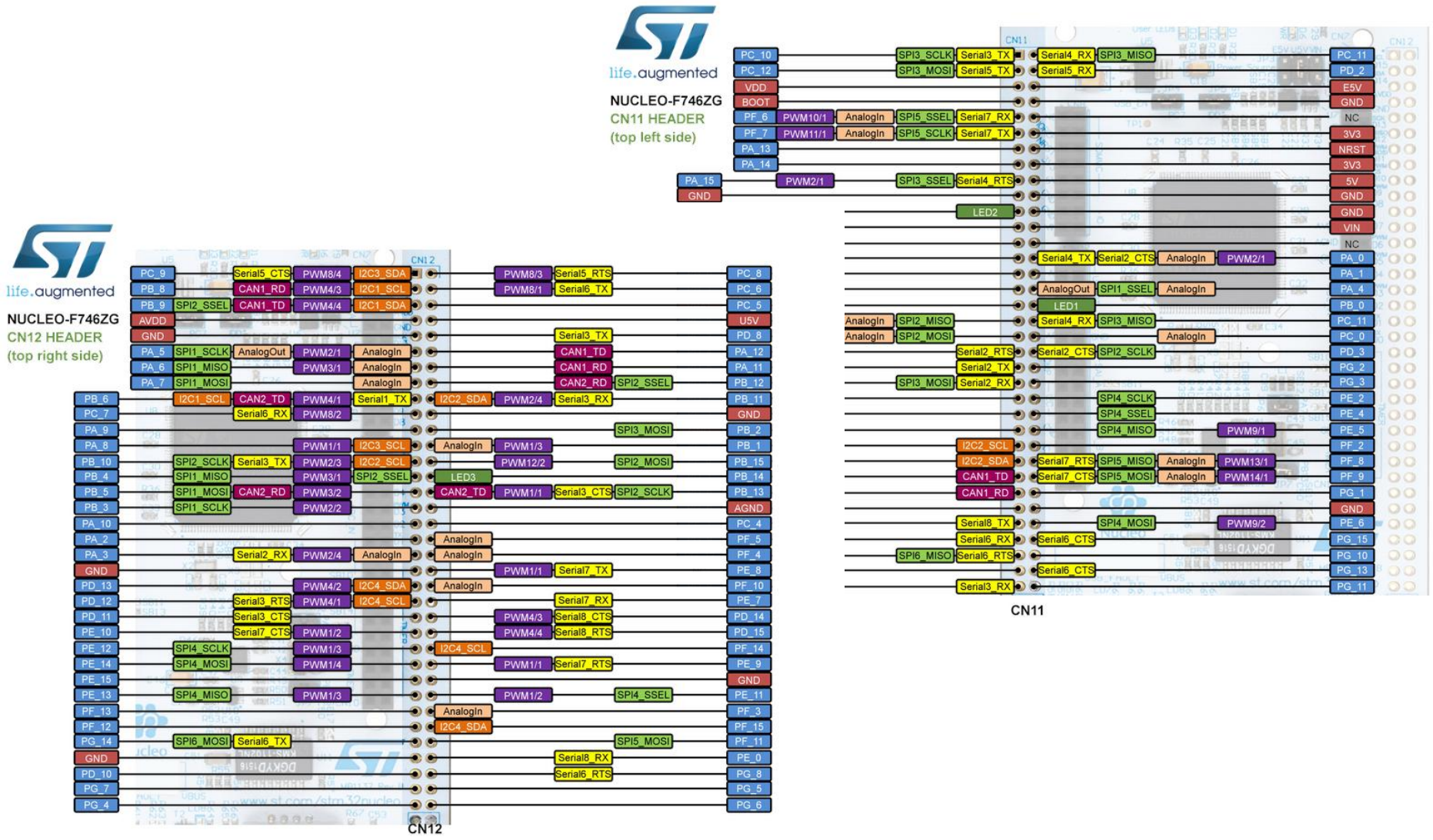
ST Nucleo L476



ST Nucleo F746

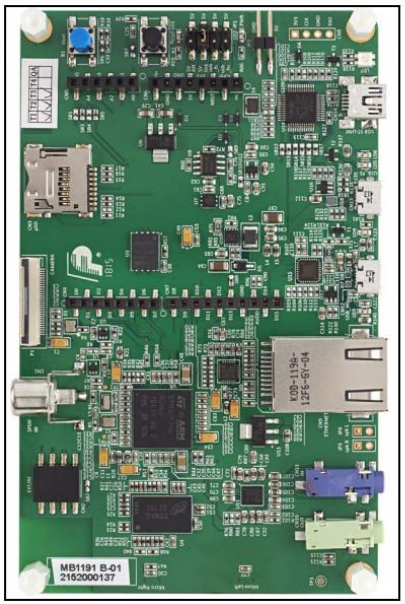
- ARM® 32-bit Cortex®-M7 CPU with FPU 216MHz
- 1MB of Flash memory
- SRAM: 320KB
- 3×12-bit, 2.4 MSPS ADC: up to 24 channels and 7.2 MSPS in triple interleaved mode
- 2×12-bit D/A converters
- Up to 18 timers
- Up to 4x I2C, 8x USARTs, 6x SPIs, 2xCAN
- USB 2.0 full-speed device/host/OTG controller with on-chip PHY
- USB 2.0 high-speed/full-speed
- 10/100 Ethernet MAC with dedicated DMA
- .....

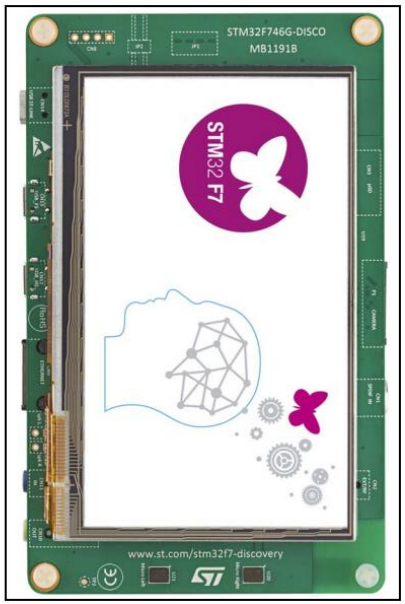
ST Nucleo F746



DISCO-F746NG

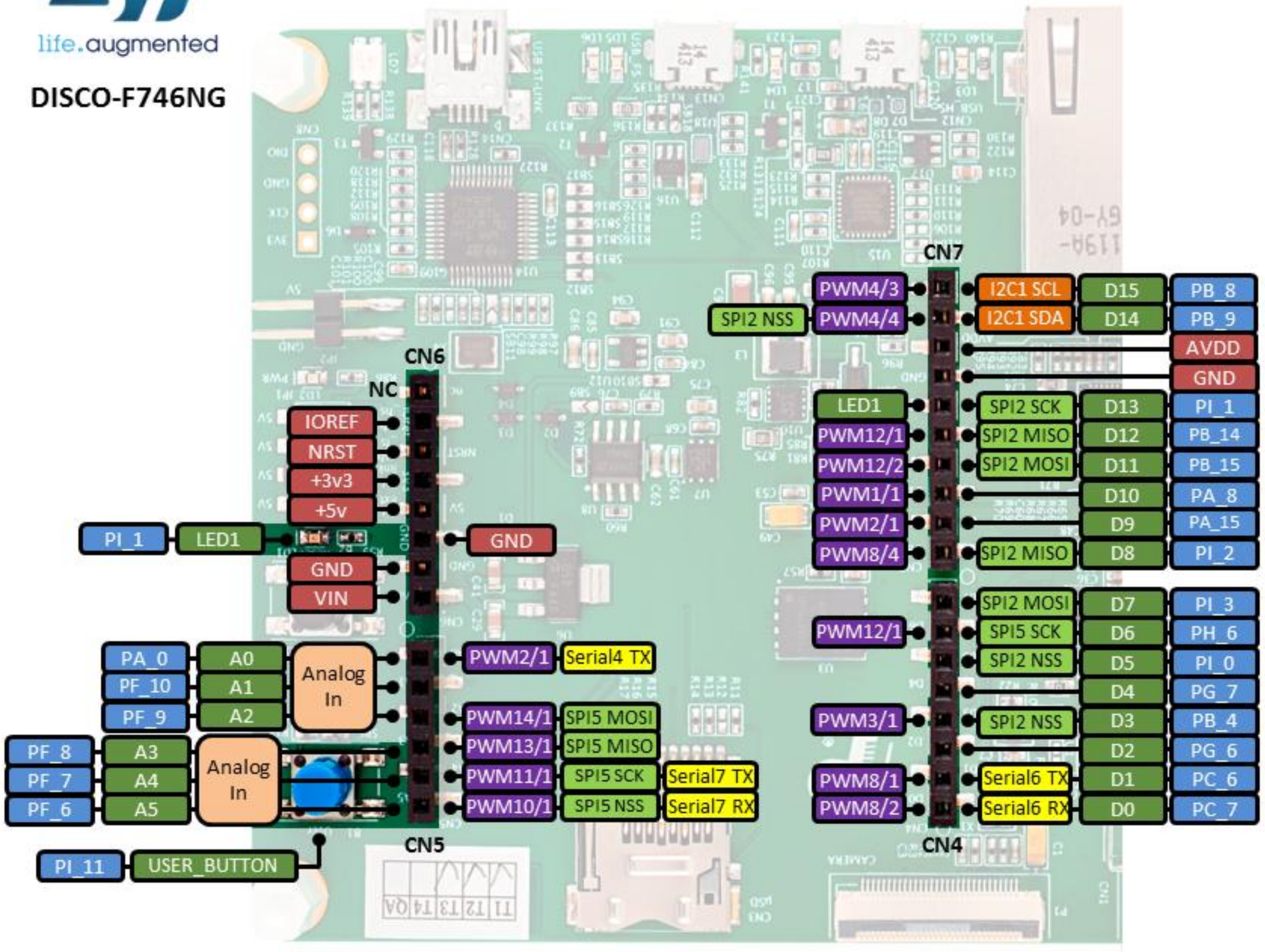
- ARM®32-bit Cortex®-M7 + FPU + Chrom-ART Accelerator
- 216 MHz max CPU frequency
- Ethernet 10/100Mb
- MicroSD card
- USART
- SAI audio DAC stereo with audio jack input and output
- MEMS digital microphones
- SDRAM
- Quad-SPI Flash memory
- 4.3-inch color LCD-TFT with a capacitive multi-touch panel





life.augmented

DISCO-F746NG



Programiranje

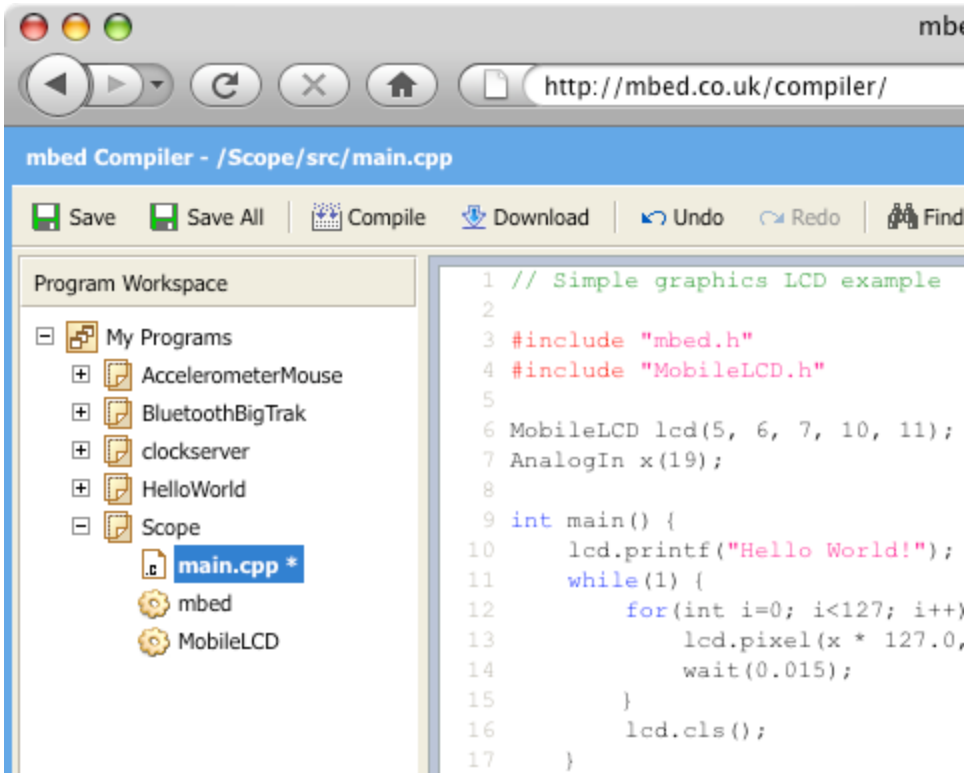
- Glavna prednost MBED arhitekture nad drugim arhitekturama je mogućnost brzog razvoja proof-of-concept aplikacija korišćenjem podrške na sajtu www.mbed.com
- Nakon logovanja sa korisničkim imenom dobijenim registrovanjem MBED-a, korisnik ima pristup web sadržajima vezanim za standardnu zvaničnu MBED biblioteku, ali i pristup bibliotekama i programima koje kreiraju drugi kosinici.
- MBED biblioteci se pristupa preko stranice Handbook, a ostalim bibliotekama preko stranice Cookbook.

Logovanje na mbed.org

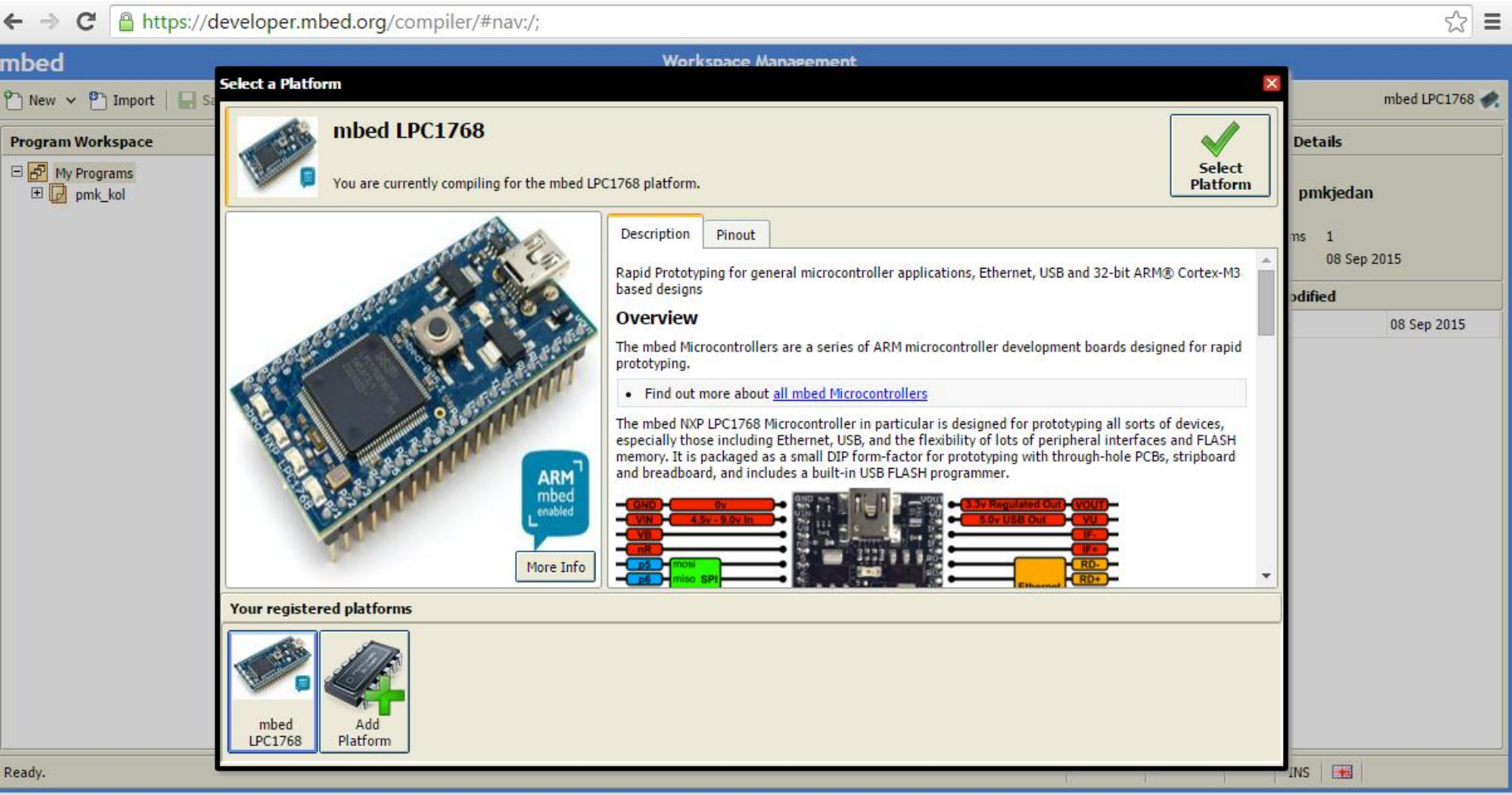
- username:
  - pmkjedan
  - pmkdva
  - pmktri
  - …..
  - pmkdeset
- password:
  - ms1pmk
- Svako može da kreira i svoj sopstveni nalog

Programiranje – pisanje programa

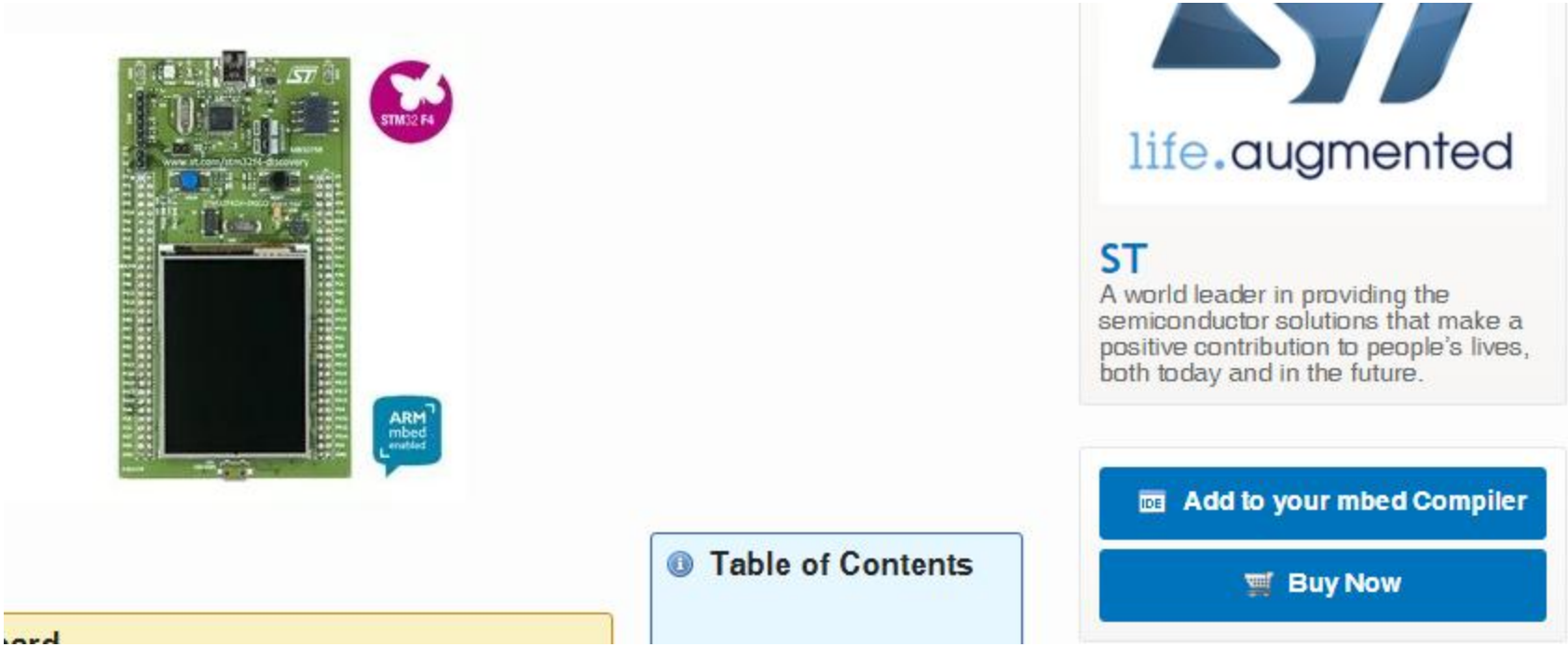
- Jedan od načina da se napiše program za MBED je korišćenje MBED online kompajlera. To naravno zahteva da se ulogujete na sajt www.mbed.org i kliknete na link compiler.
- Kompajler zna da bude spor i nekomforman za editovanje, ali prevođenje radi brzo i nema ograničenja u veličini fajla.



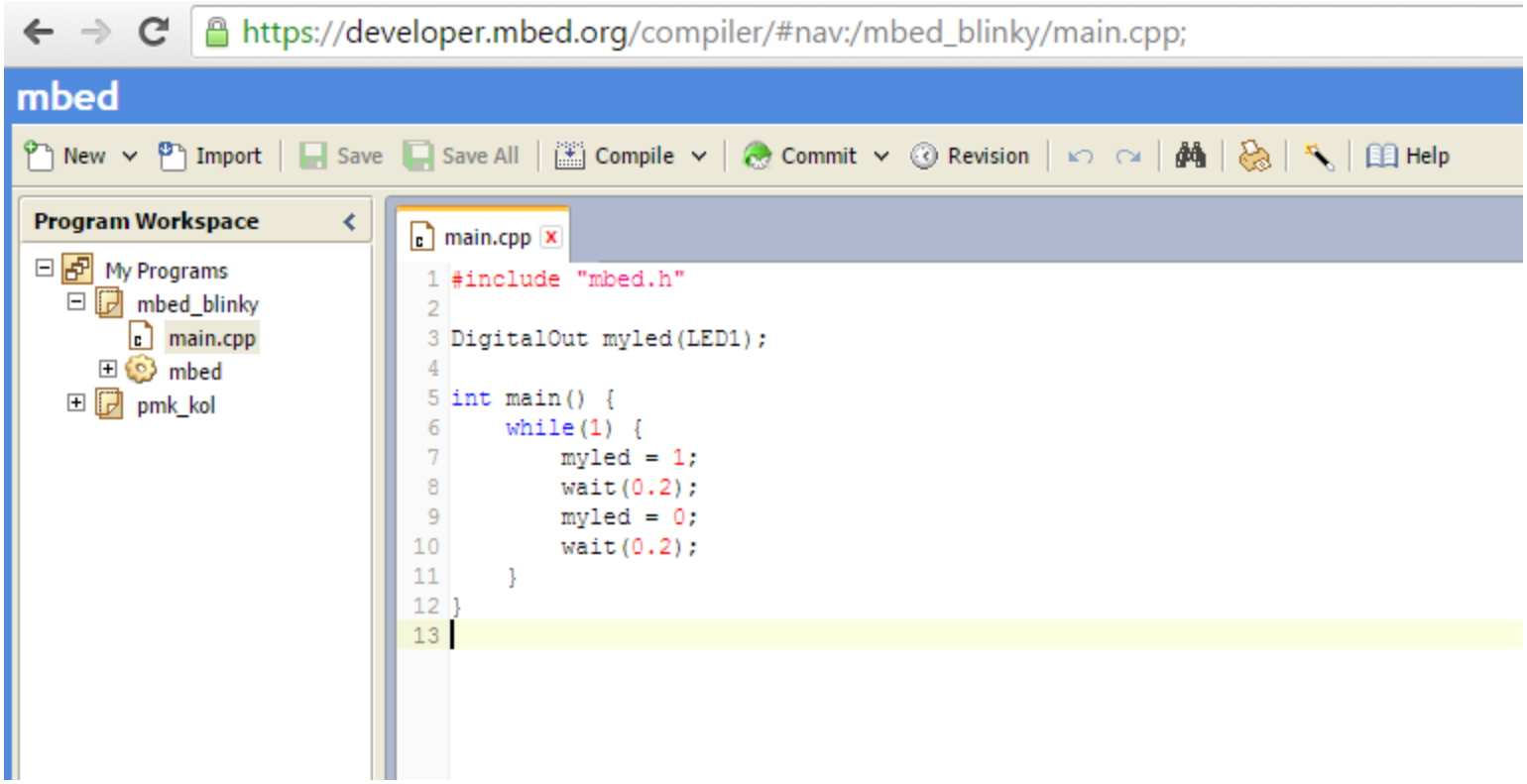
Dodavanje nove platforme



Dodavanje nove platforme

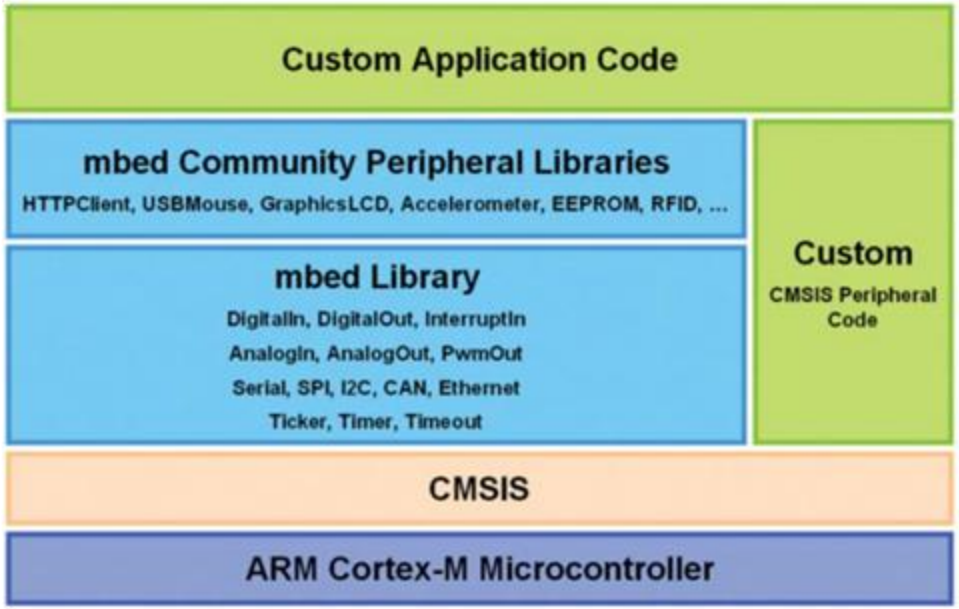


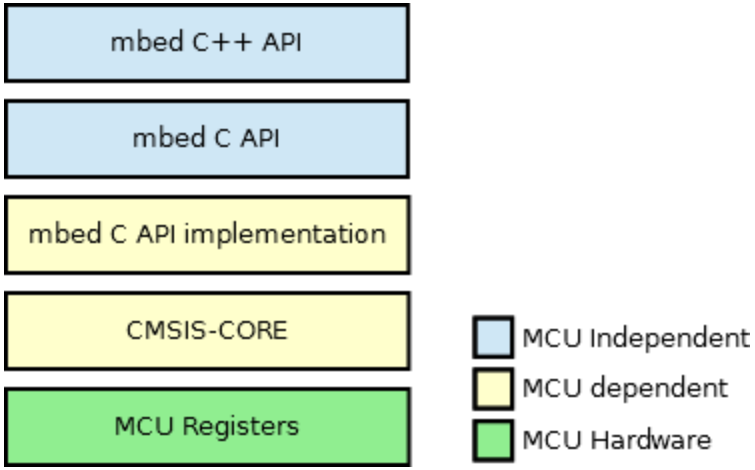
Kreiranje novog programa



Arhitektura MBED softvera

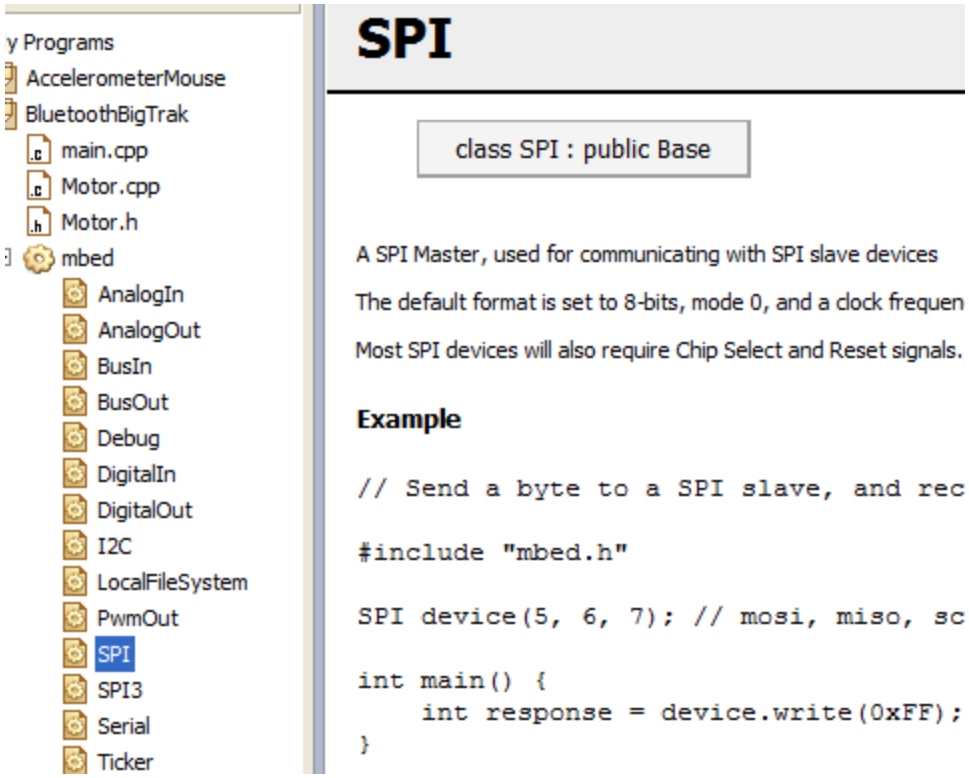
- MBED se oslanja na CMSIS standard koji definiše Low Level pristup resursima mikrokontrolera, dok MBED biblioteke obezbeđuju visoki aplikativni nivo apstrakcije.



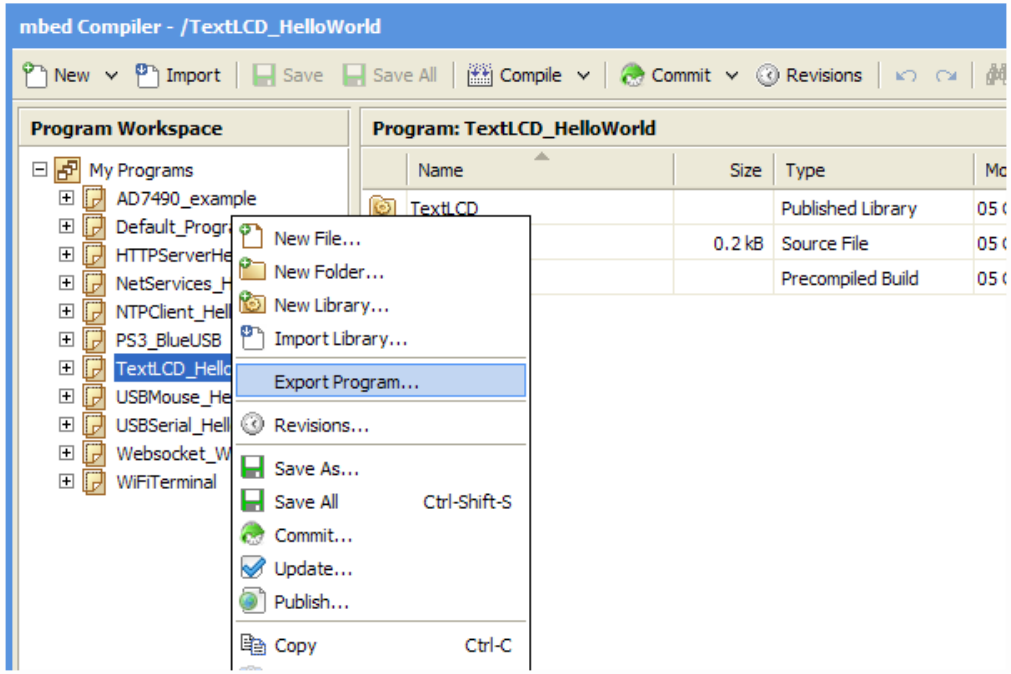


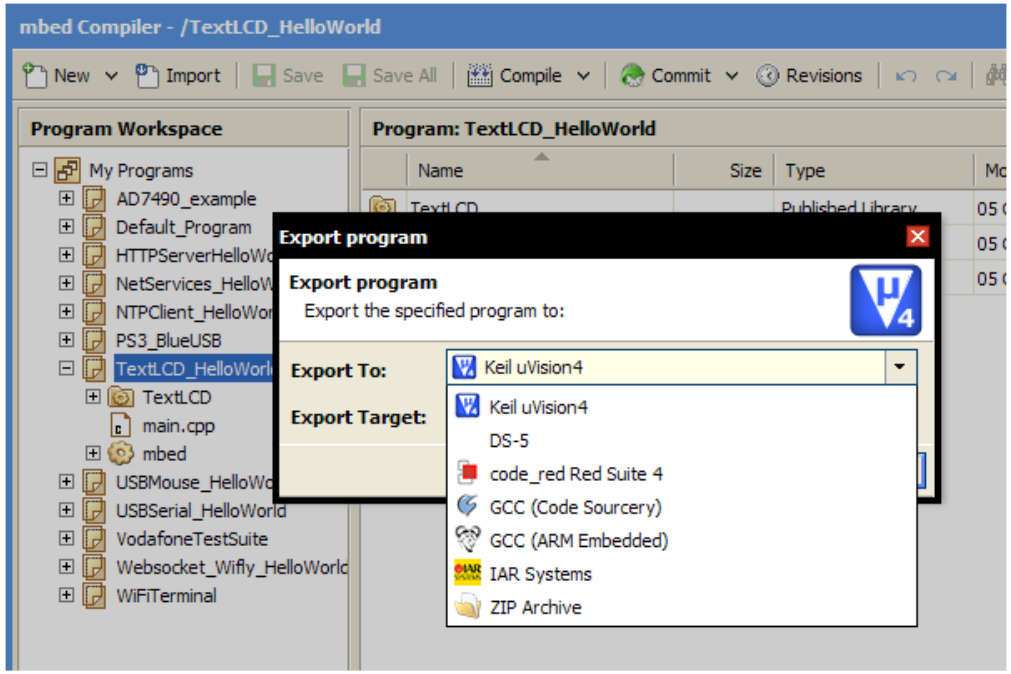
Programiranje – biblioteke

- Osnovna biblioteka MBED kompajlera nosi naziv MBED.xx, i header za pristup njenim funkcijama i konstantama nosi naziv mbed.h.
- Biblioteka MBED se sastoji iz mnogo podbiblioteka i svaka od njih pseduje svoj .h fajl.
- Svaki program koji koristi bilo koju MBED funkciju ili konstatnu treba na početku da ima direktivu #include “mbed.h”.
- Za razliku od klasičnih kompajlera koji dozvoljavaju čitanje .h fajlova online compiler to ne dozvoljava.
- NA sreću na sajtu se može naći hello_world verzija projekta prilagođena KEIL kompajleru u kojoj se nalaze i svi .h fajlovi.



Korišćenje offline kompajlera

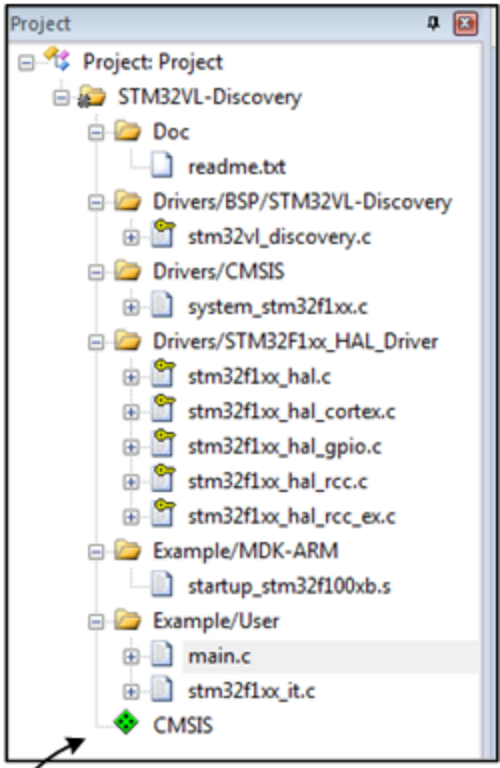




Pravimo serijski proizvod

- Možemo da koristimo i MBED softver (?)
- Svakako moramo da razumemo celu strukturu projekta.
- Ako je u pitanju zaista ozbiljan proizvod preporuka je osloniti se na najnovije HAL drajvere proizvođača.
- Mi ćemo to da ispitamo na primeru NUCLEO razvojnog sistema kompanije ST Microelectronics

STM CUBE CMSIS Struktura projekta



Startup fajl – definiše ga proizvođač kompajlera

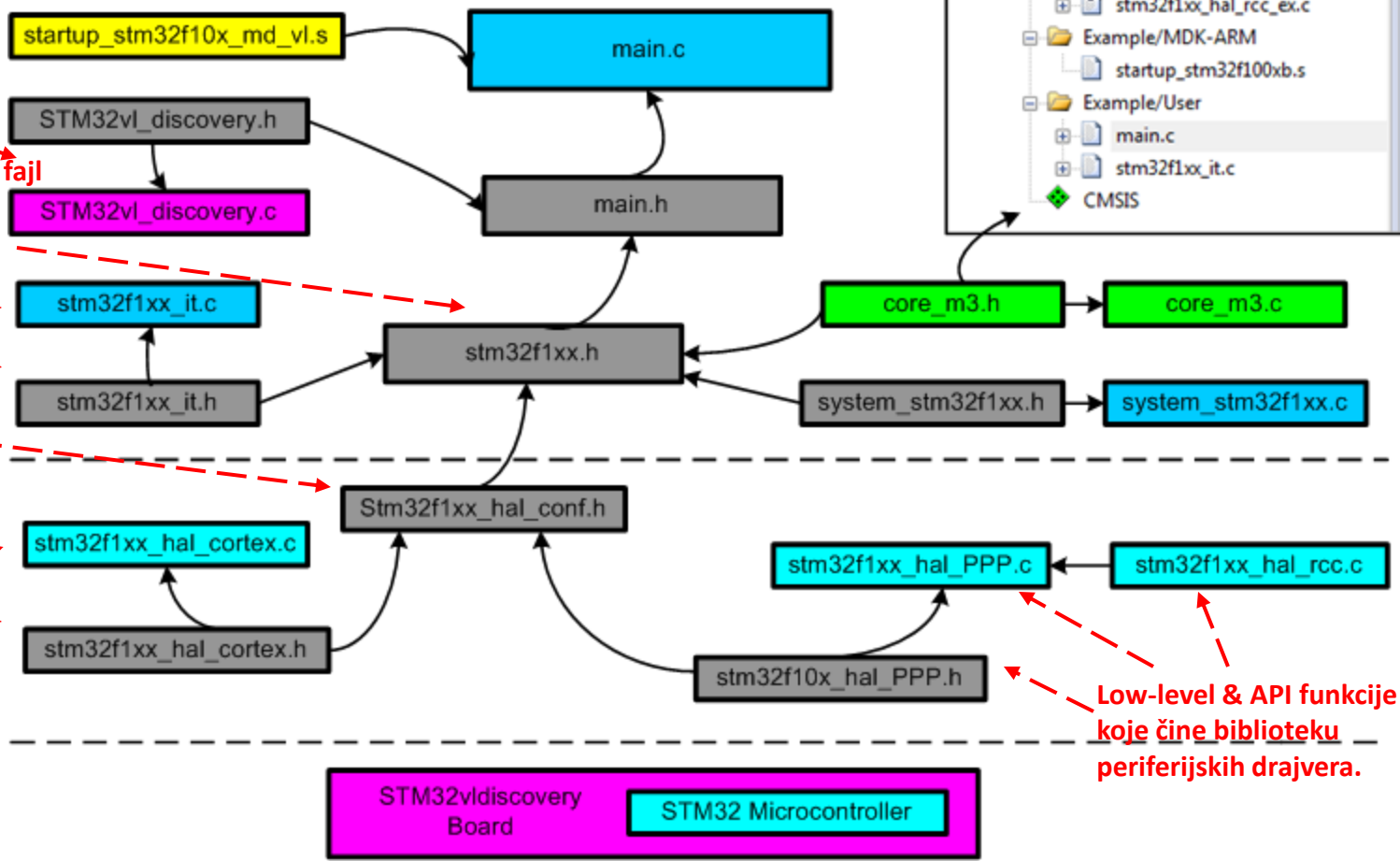
BSP

Osnovni konfiguracioni heder fajl za odabranu familiju mikrokontrolera

Cortex-M3 prekidi i izuzetci

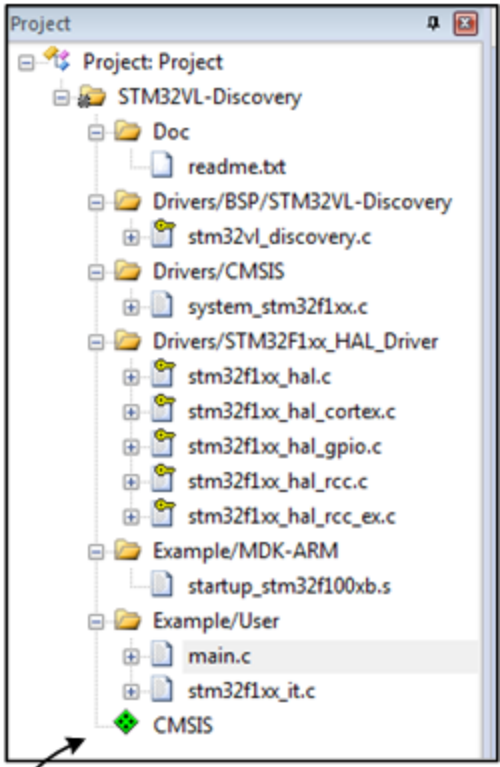
Peripheral header file
Ovaj fajl prilagodjava korisnik

NVIC i SysTick drivers



Low-level & API funkcije koje čine biblioteku periferijskih drajvera.

Lakše je sa MBED-om



Startup fajl – definiše ga proizvođač kompajlera

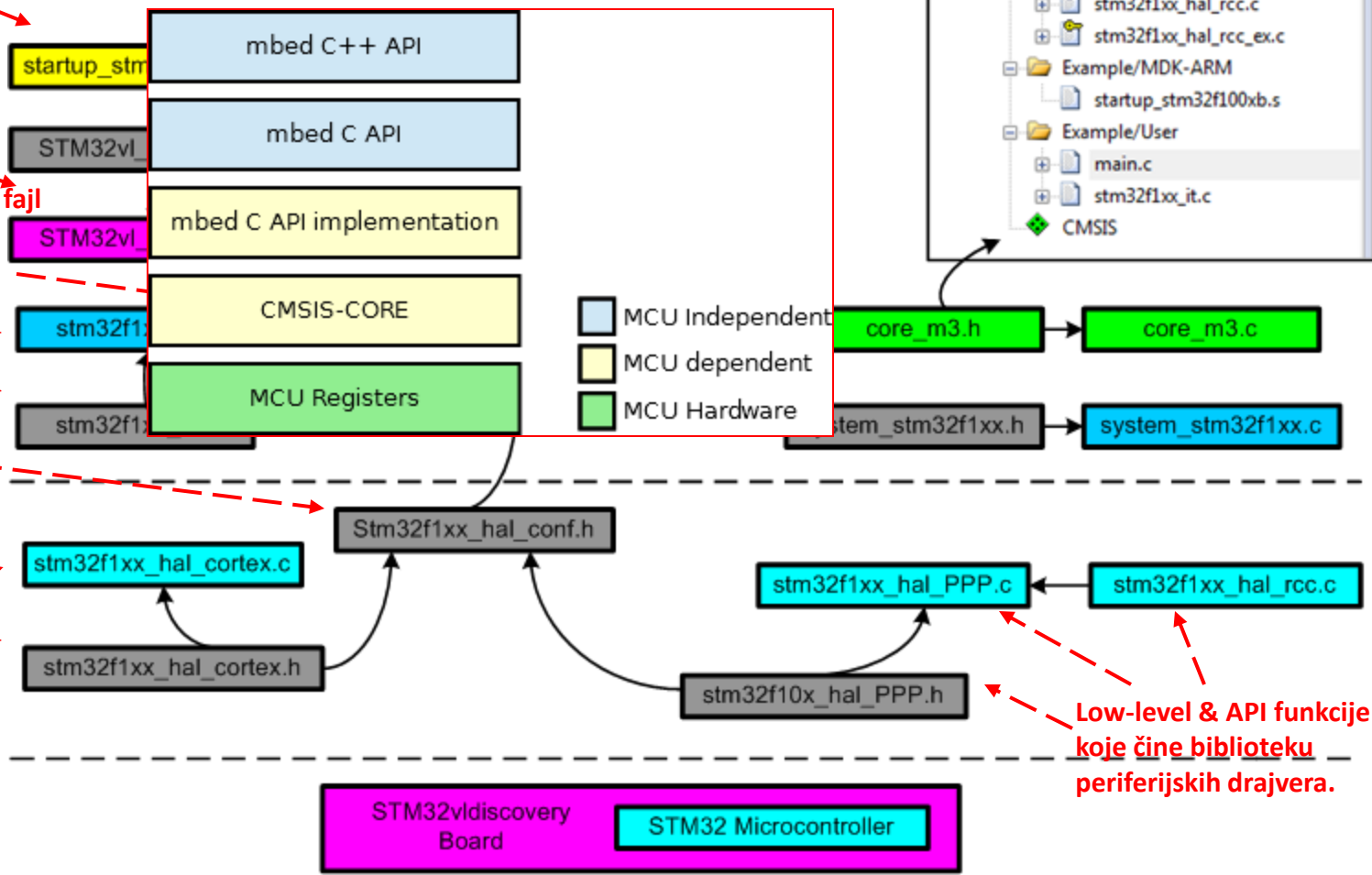
BSP

Osnovni konfiguracioni heder fajl za odabranu familiju mikrokontrolera

Cortex-M3 prekidi i izuzetci

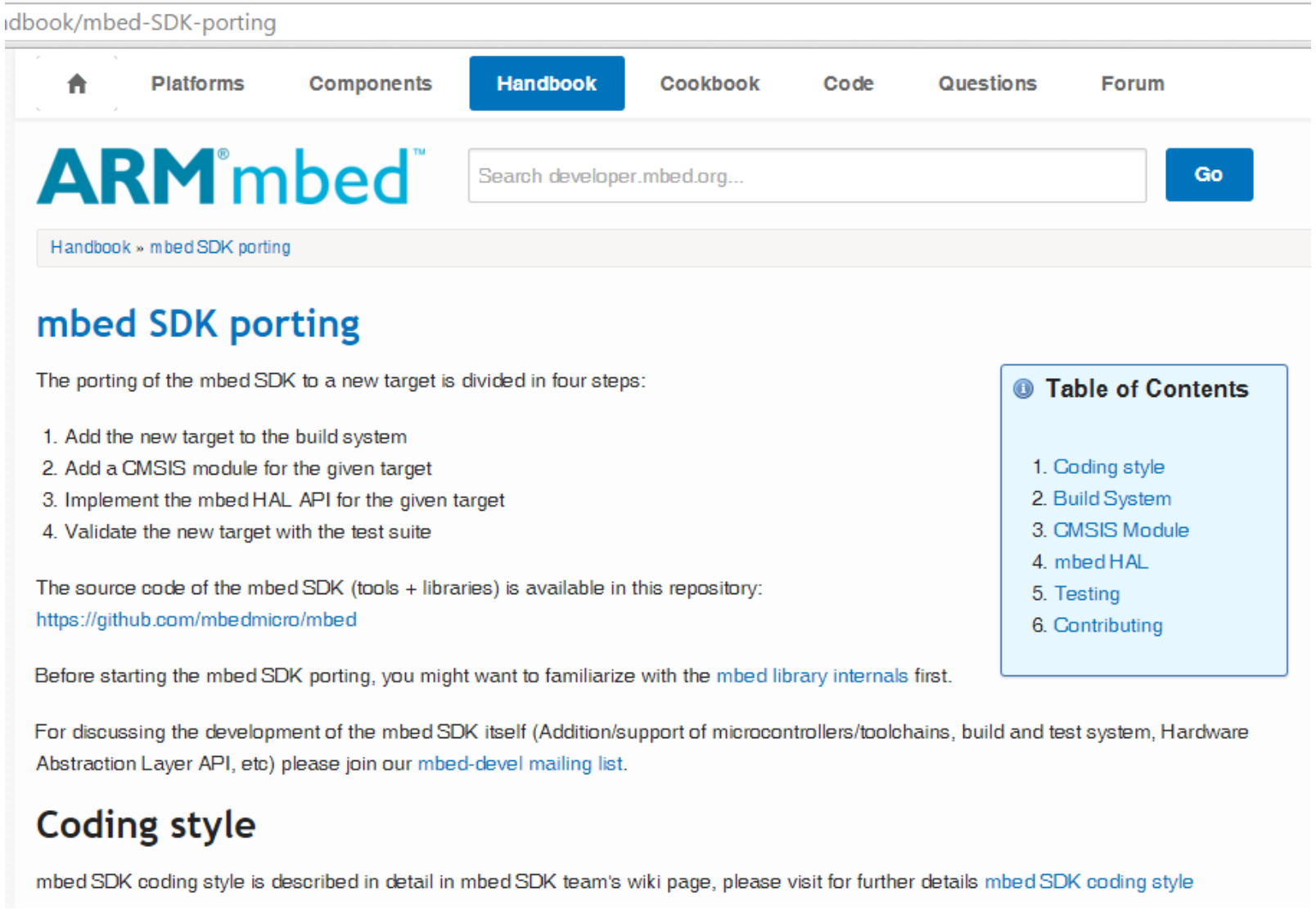
Peripheral header file
Ovaj fajl prilagodjava korisnik

NVIC i SysTick drivers



Low-level & API funkcije koje čine biblioteku periferijskih drajvera.

A šta ako pravimo malu seriju i najbitnije je vreme?



dbook/mbed-SDK-porting

Platforms Components Handbook Cookbook Code Questions Forum

ARM mbed

Search developer.mbed.org... Go

Handbook » mbed SDK porting

mbed SDK porting

The porting of the mbed SDK to a new target is divided in four steps:

1. Add the new target to the build system
2. Add a CMSIS module for the given target
3. Implement the mbed HAL API for the given target
4. Validate the new target with the test suite

The source code of the mbed SDK (tools + libraries) is available in this repository:
https://github.com/mbedmicro/mbed

Before starting the mbed SDK porting, you might want to familiarize with the mbed library internals first.

For discussing the development of the mbed SDK itself (Addition/support of microcontrollers/toolchains, build and test system, Hardware Abstraction Layer API, etc) please join our mbed-devel mailing list.

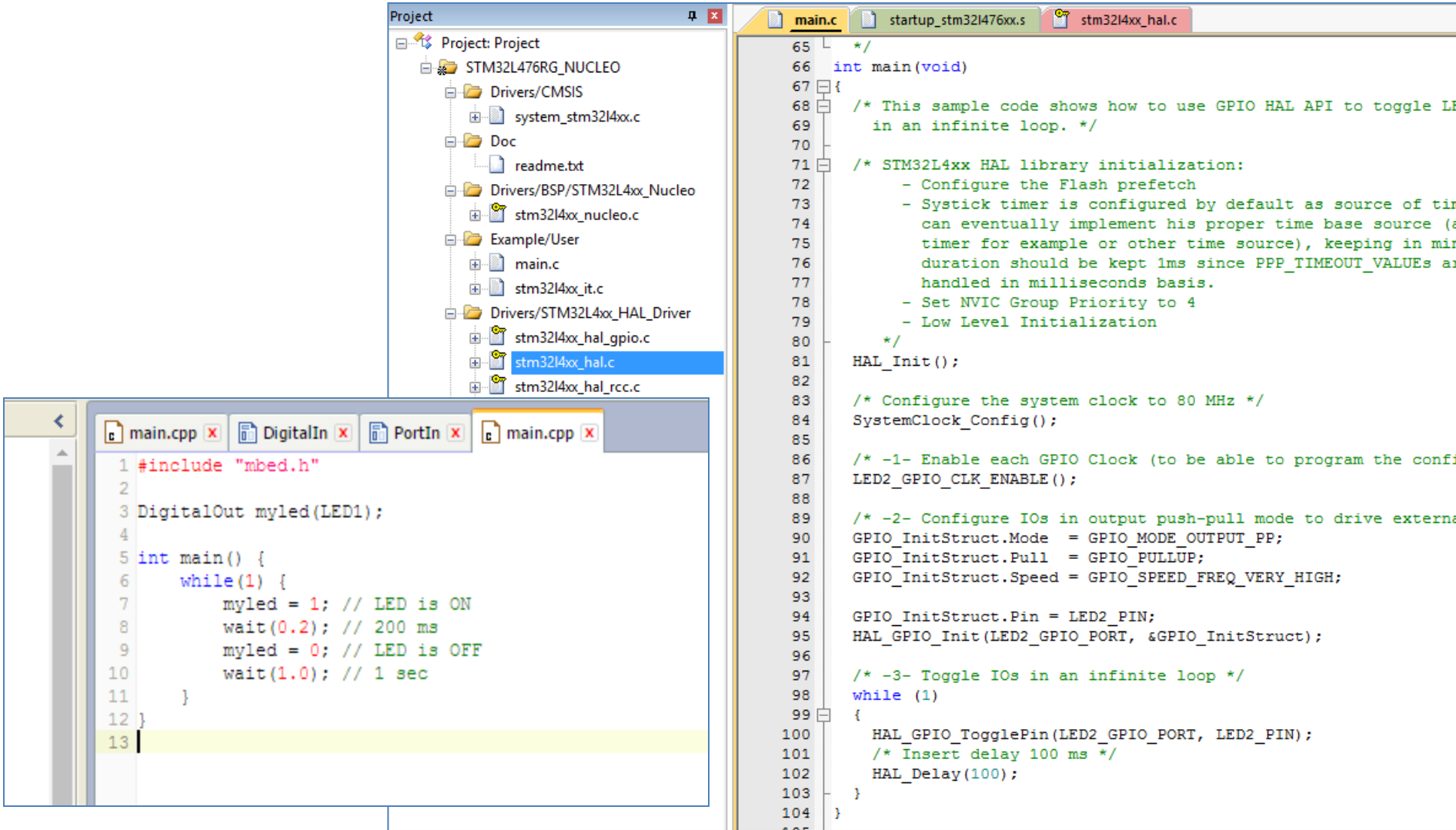
Table of Contents

1. Coding style
2. Build System
3. CMSIS Module
4. mbed HAL
5. Testing
6. Contributing

Coding style

mbed SDK coding style is described in detail in mbed SDK team's wiki page, please visit for further details mbed SDK coding style

MBED vs CUBE


Project
Project: Project
STM32L476RG_NUCLEO
Drivers/CMSIS
system_stm32l4xx.c
Doc
readme.txt
Drivers/BSP/STM32L4xx_Nucleo
stm32l4xx_nucleo.c
Example/User
main.c
stm32l4xx_it.c
Drivers/STM32L4xx_HAL_Driver
stm32l4xx_hal_gpio.c
stm32l4xx_hal.c
stm32l4xx_hal_rcc.c
main.cpp
DigitalIn
PortIn
main.cpp
#include "mbed.h"
DigitalOut myled(LED1);
int main() {
while(1) {
myled = 1; // LED is ON
wait(0.2); // 200 ms
myled = 0; // LED is OFF
wait(1.0); // 1 sec
}
}
main.c
startup_stm32l476xx.s
stm32l4xx_hal.c
*/
int main(void)
{
/* This sample code shows how to use GPIO HAL API to toggle LE
in an infinite loop. */
/* STM32L4xx HAL library initialization:
- Configure the Flash prefetch
- Systick timer is configured by default as source of tim
can eventually implement his proper time base source (a
timer for example or other time source), keeping in mir
duration should be kept 1ms since PPP_TIMEOUT_VALUEs ar
handled in milliseconds basis.
- Set NVIC Group Priority to 4
- Low Level Initialization
*/
HAL_Init();
/* Configure the system clock to 80 MHz */
SystemClock_Config();
/* -1- Enable each GPIO Clock (to be able to program the confi
LED2_GPIO_CLK_ENABLE();
/* -2- Configure IOs in output push-pull mode to drive externa
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_PULLUP;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
GPIO_InitStruct.Pin = LED2_PIN;
HAL_GPIO_Init(LED2_GPIO_PORT, &GPIO_InitStruct);
/* -3- Toggle IOs in an infinite loop */
while (1)
{
HAL_GPIO_TogglePin(LED2_GPIO_PORT, LED2_PIN);
/* Insert delay 100 ms */
HAL_Delay(100);
}
}