



Studenckie Koło Naukowe
Politechniki Gdańskiej

Akademia ETI 2016

PREZENTACJA - MIKROKONTROLERY

Czym jest mikrokontroler

Mikrokontroler to wyspecjalizowany układ scalony, zawierający jednostkę centralną (mikroprocesor), pamięć RAM, zazwyczaj również pamięć programu i różne peryferia – układy wejść i wyjść, przetworniki cyfrowo-analogowe i analogowo-cyfrowe, kontrolery magistrali I2C, SPI i wiele innych, zależnie od producenta i modelu mikrokontrolera.

Wiele producentów elementów elektronicznych ma w swojej ofercie mikrokontrolery, na przykład: STMicroelectronics, Atmel, NXP, Microchip, Freescale. Dziś na warsztat weźmiemy mikrokontrolery z serii STM32 – układy firmy STMicroelectronics z rdzeniem ARM Cortex-M4.



ARM[®] Cortex[®]-M4

Nested Vectored
Interrupt Controller

Wake Up Interrupt
Controller Interface

CPU
(with DSP Extensions)

FPU

Code
Interface

Memory
Protection
Unit

SRAM &
Peripheral
Interface

Bus
Matrix

Data
Watchpoint

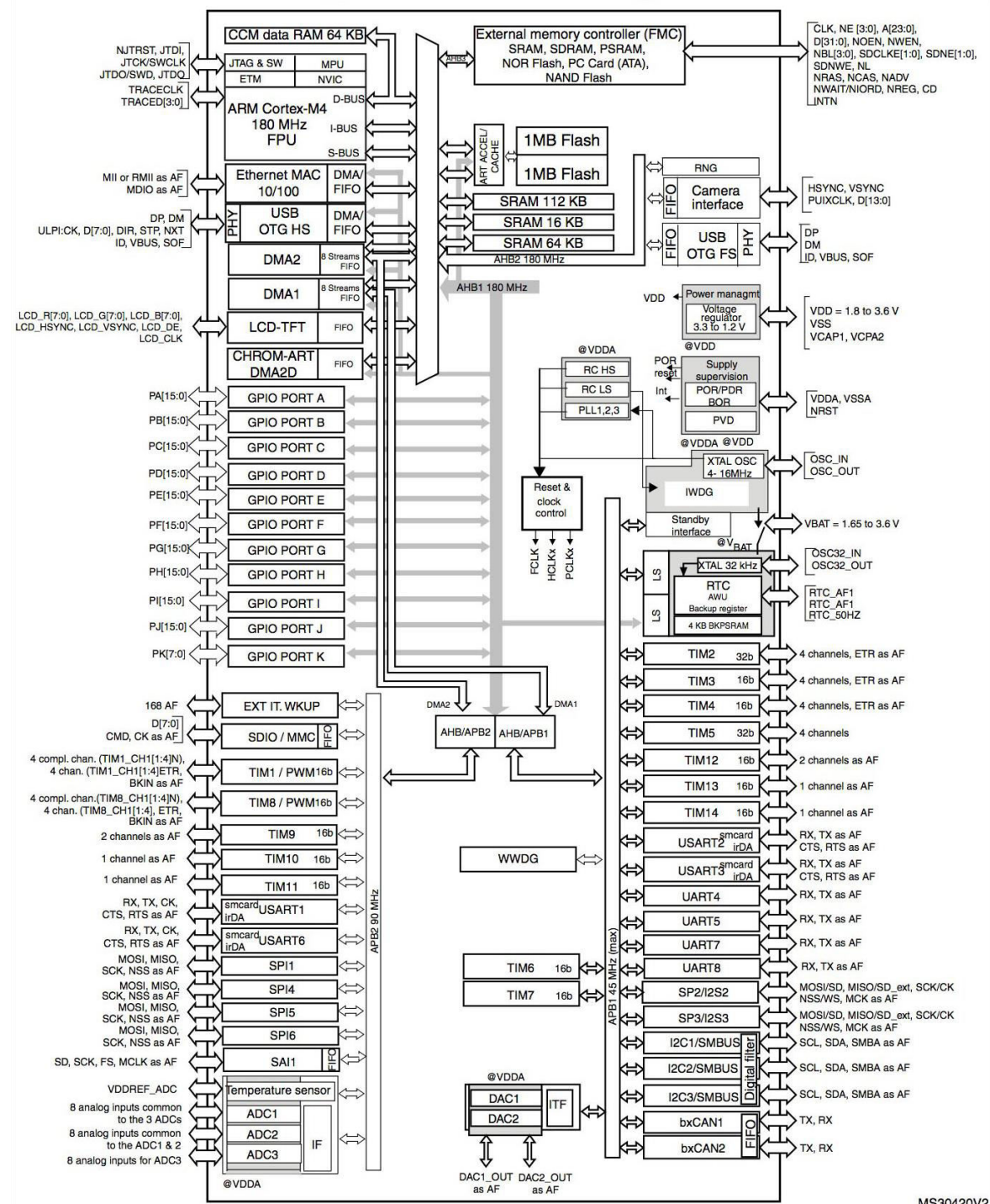
Flash Patch
& Breakpoint

ITM Trace

ETM Trace

Debug
Access
Port

Serial
Wire
Viewer,
Trace Port



Zastosowanie mikrokontrolerów

Mikrokontrolery mają zastosowanie w wielu przedmiotach codziennego użytku – pralki, lodówki, wagi kuchenne, samochody, bankomaty, systemy alarmowe itp. Układy sterujące pracą tych przedmiotów noszą miano „systemów wbudowanych”. Celem powstania systemów wbudowanych jest zapewnienie urządzeniom samodzielności i „inteligencji”.

Idealnym przykładem zastosowania systemu wbudowanego (a co za tym idzie mikrokontrolerów) są układy bezpieczeństwa w samochodzie. W jego wnętrzu znajdują się dziesiątki, jak nie setki czujników i sensorów – czujniki temperatury silnika, otoczenia, wnętrza kabiny, czujniki odległościowe do wspomagania parkowania, coraz popularniejsze stają się również kamery dzięki którym samochód może rozpoznawać znaki drogowe i przypominać kierowcy o ograniczeniu prędkości lub niebezpieczeństwie na drodze.

W dużym uproszczeniu system wbudowany ma za zadanie pełnić kontrolę nad danym zjawiskiem lub funkcją urządzenia w zależności od obecnych warunków - tak więc alarm się włączy tylko jak wykryje włamanie, samochód włączy ABS tylko jeśli samochód zacznie wpadać w poślizg, zaś winda nie ruszy jeśli jest przeciążona.

Mikroprocesory ARM

ARM (Advanced RISC Machines) – firma projektująca mikroprocesory. Zajmuje się jedynie ich projektowaniem i sprzedawaniem licencji innym firmom, takim jak STMicroelectronics (STM32), Atmel (AT91SAM), NXP (LPC).

Obecna używana architektura to ARMv7, czyli mikrokontrolery z rdzeniem Cortex (za wyjątkiem rdzeni Cortex-M0, M0+ i M1 które mają architekturę ARMv6).

Rdzenie Cortex występują w 3 wariantach – A, M i R.

Cortex-A - *mikroprocesory* przeznaczone głównie do obsługi systemów operacyjnych. Ponieważ jest to sam procesor, do pracy wymaga jeszcze zewnętrznych peryferiów.

Cortex-M - *mikrokontrolery*, w tym STM32F429ZIT6 omawiany na warsztatach, poza mikroprocesorem posiada również wbudowane peryferia, ich typ i ilość zależna jest od modelu i złożoności układu.

Cortex-R - układy stosowane w systemach czasu rzeczywistego.

Rdzenie ARM Cortex

Nazwy rdzeni procesorów z serii M:

Cortex-M0/M0+ - najmniejszy i najprostszy mikroprocesor, tania 32-bitowa alternatywa dla mikrokontrolerów 8 i 16-bitowych. M0+ jest usprawnioną wersją M0 o zmniejszonym poborze energii i większej wydajności.

Cortex-M1 - mikroprocesor zaprojektowany do stosowania w układach FPGA.

Cortex-M3 - najbardziej popularna rodzina mikroprocesorów, stosowana m.in. w przemyśle motoryzacyjnym i urządzeniach do komunikacji bezprzewodowej.

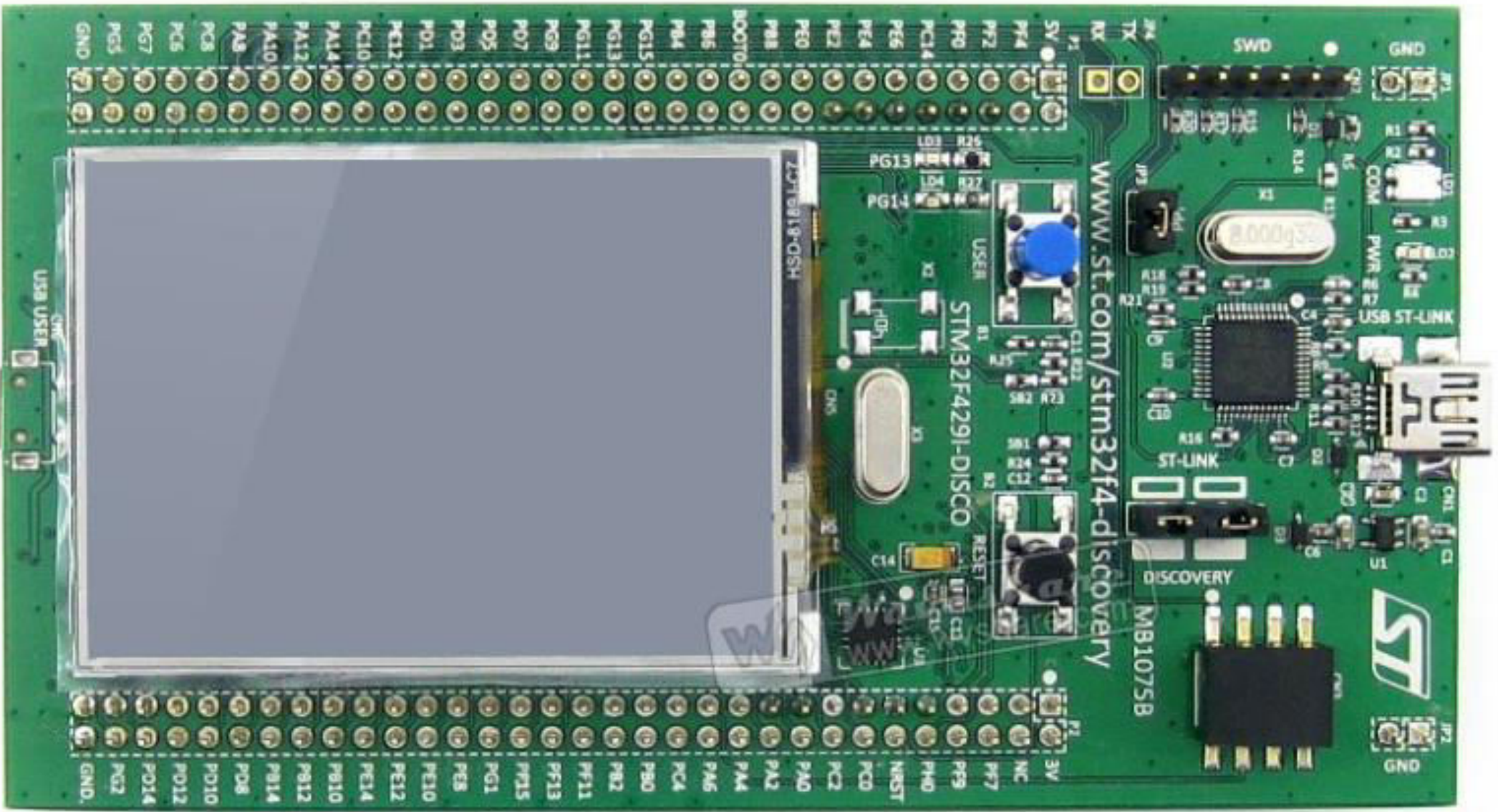
Cortex-M4 - mikroprocesor zaprojektowany głównie z myślą o sprawnej obsłudze dużej ilości sygnałów cyfrowych, stosowany w przemyśle motoryzacyjnym, energetycznym, automatyce przemysłowej i urządzeniach audio. Od Cortex-M3 różni się obecnością koprocatora FPU (Float-Point Unit) - układu wspomagającego procesor w obliczeniach, głównie zmiennoprzecinkowych. Używany przez nas STM32F429 jest mikrokontrolerem z rdzeniem M4.

Cortex-M7 - na chwilę obecną najwyższa rodzina procesorów w architekturze ARM. Używana do budowy skomplikowanych mikrokontrolerów i układów System On Chip.

STM32F429i-Discovery

Zestaw uruchomieniowy produkcji STMicroelectronics. Znajdują się na nim:

- mikrokontroler STM32F429ZIT6U
- programator + debugger ST-Link/V2
- 114 wyprowadzeń GPIO (złącza goldpin 2.54mm)
- wyświetlacz LCD QVGA 262144 kolorów
- SDRAM 64Mbit
- 3-osiowy cyfrowy żyroskop
- microswitch (PA0, zewnętrzny pulldown i debouncing) i 2 diody LED (PG13, PG14)
- User USB
- wyprowadzenie do pomiaru pobieranego prądu
- złącze SWD do podłączenia zewnętrznego programatora
- kwarc 8MHz, źródło taktowania zegara HSE
- diody sygnalizujące zasilanie, podłączenie/pracę programatora i debuggera, pracę interfejsu USB



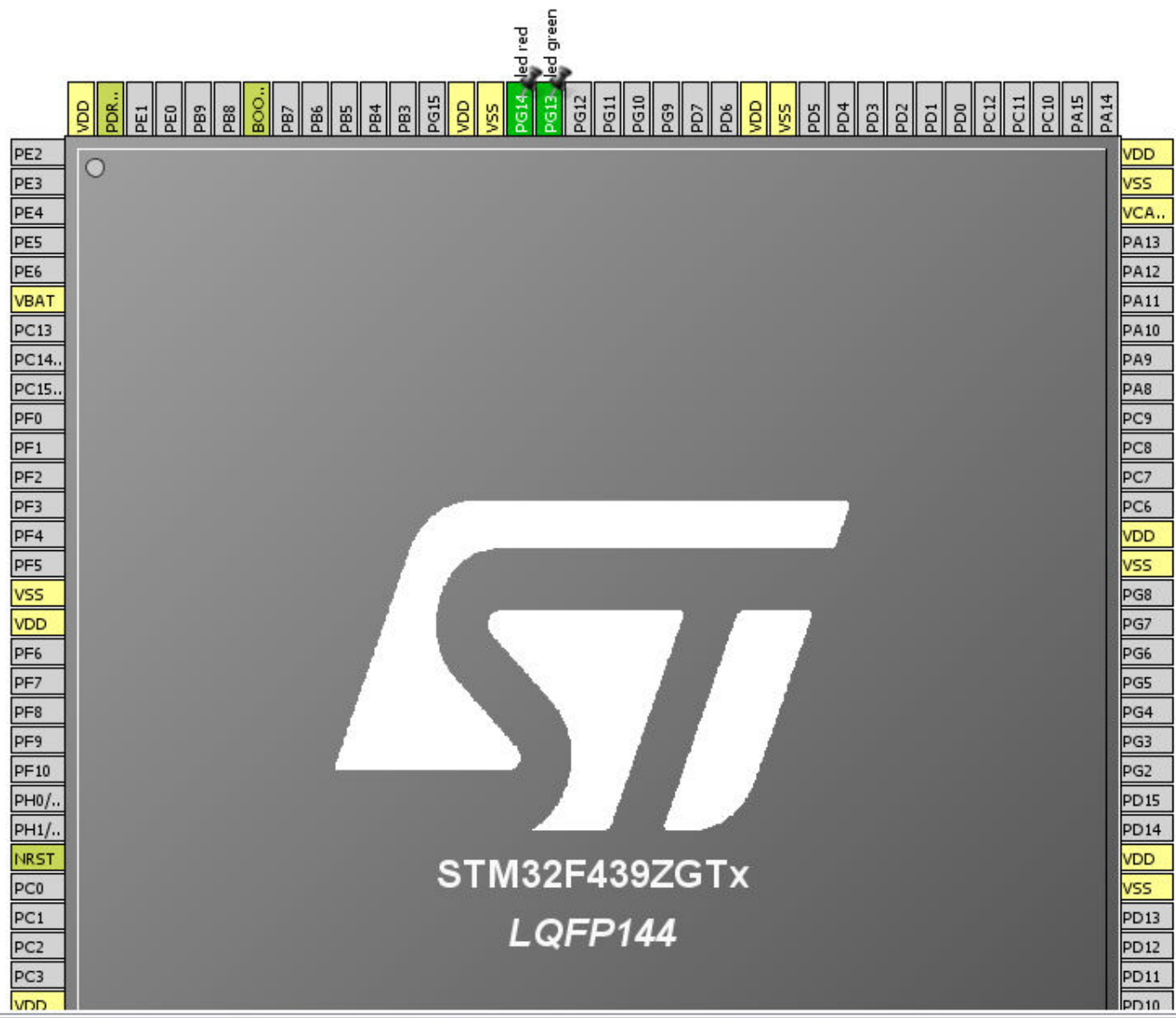
STM32CubeMX

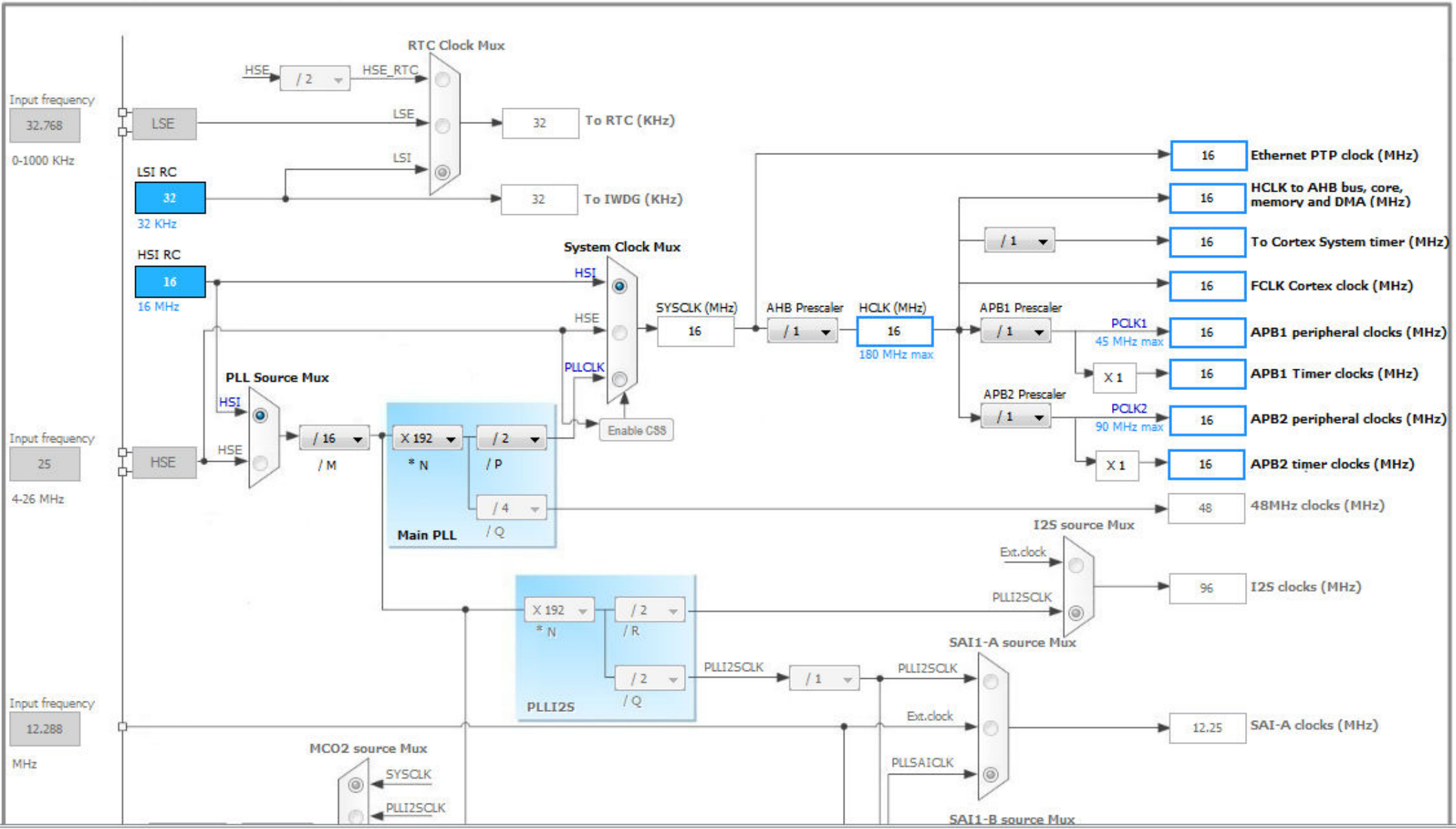
Aby prawidłowo zaprogramować mikrokontroler należy prawidłowo skonfigurować używane w projekcie peryferia – porty wejść i wyjść, zegar taktujący układ, timery, przetworniki itd. Mikrokontrolery z serii STM32 są bardzo złożonymi i zaawansowanymi układami. Posiadają wiele rozmaitych peryferiów, konfiguracja każdego z nich jest czasochłonna i wymaga dużej uwagi – źle skonfigurowane nie będą działać prawidłowo, a znalezienie błędu może zająć dużo czasu.

Z pomocą przychodzi program CubeMX, udostępniany bezpłatnie przez producenta. Posiada czytelny, przyjazny użytkownikowi interfejs graficzny, główną jego zaletą jest łatwość obsługi – każde peryferium można włączyć i ustawić za pomocą kilku kliknięć, nie trzeba czytać kilkusetstronicowej noty katalogowej mikrokontrolera żeby móc zacząć z nim pracę.

Po wygenerowaniu projektu tworzy się folder zawierający wszystkie niezbędne pliki – biblioteki, pliki projektu oraz szablon kodu źródłowego, w którym znajdują się funkcje włączające i konfigurujące potrzebne nam funkcje i peryferia – nie musimy robić tego ręcznie.

- Configuration**
- MiddleWares
 - FATFS
 - FREERTOS
 - LWIP
 - USB_DEVICE
 - USB_HOST
 - Peripherals
 - ADC1
 - ADC2
 - ADC3
 - CAN1
 - CAN2
 - CRC
 - DAC
 - DCMI
 - DMA2D
 - ETH
 - FMC
 - I2C1
 - I2C2
 - I2C3
 - I2S2
 - I2S3
 - IWDG
 - LTDC
 - RCC
 - RNG
 - RTC
 - SAI1
 - SDIO
 - SPI1
 - SPI2

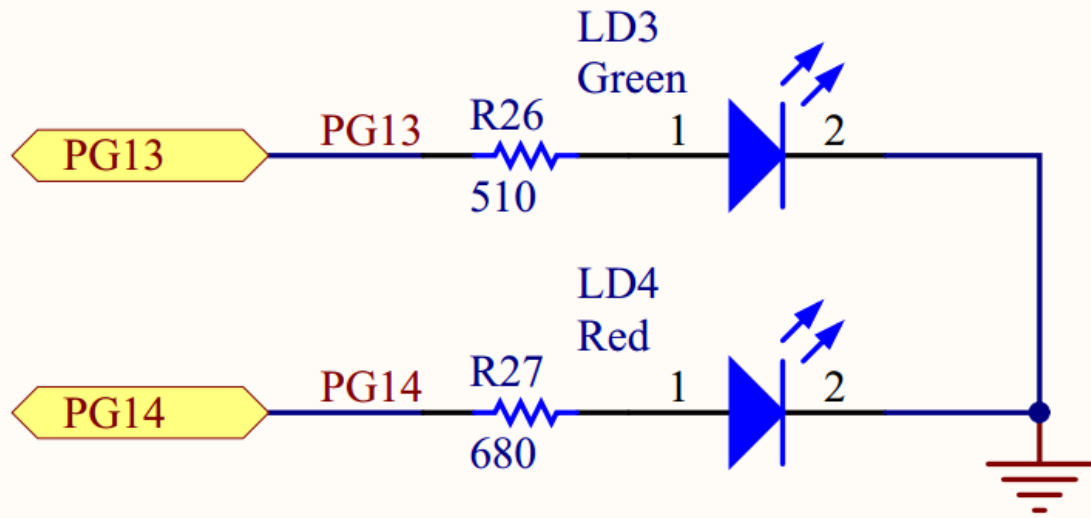




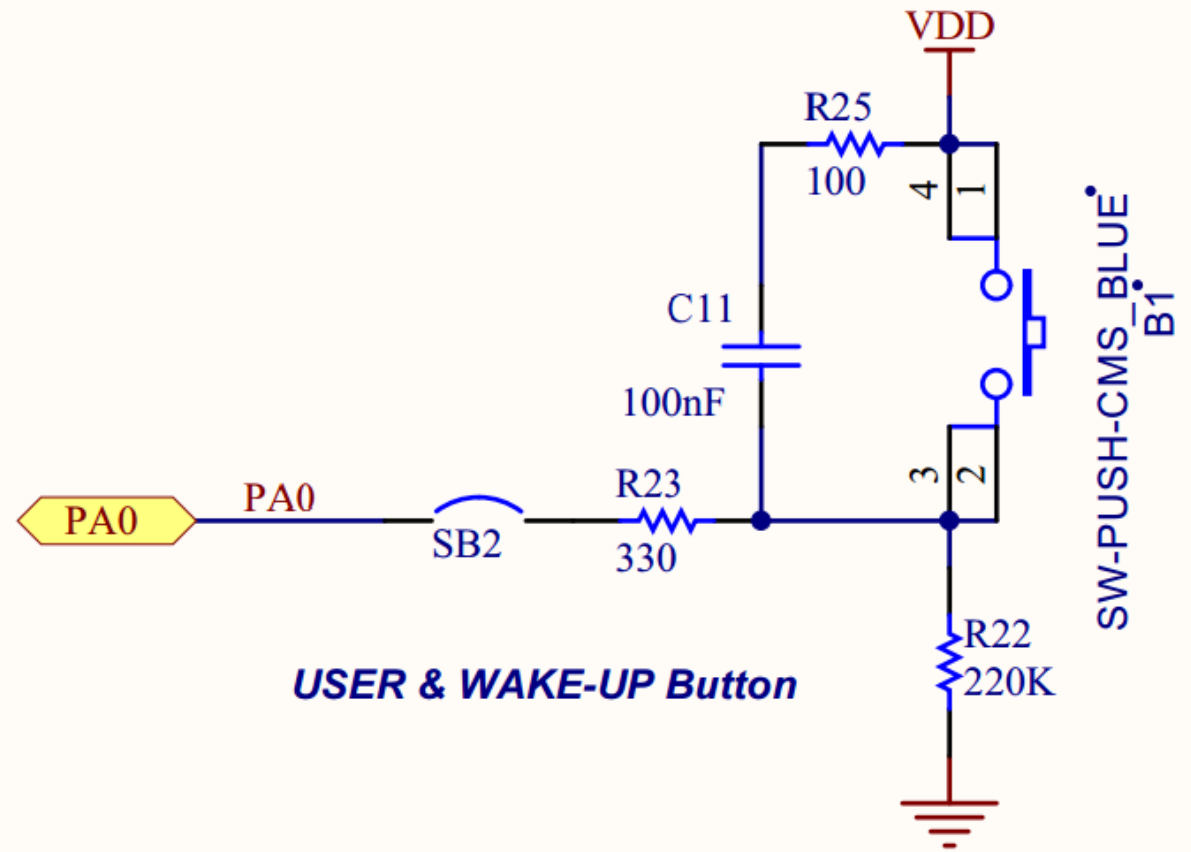
Wejścia i wyjścia cyfrowe

Większość mikrokontrolerów wyposażona jest w układy wejść i wyjść. Są to specjalne wyprowadzenia mikrokontrolera za pomocą których można komunikować się z otoczeniem. Służą one do ustawiania i sprawdzania stanu logicznego na wyprowadzeniu. W elektronice stan logiczny związany jest z napięciem, tak więc stan niski (logiczne 0) to napięcie 0V, zaś stan wysoki (logiczne 1) to napięcie wyższe – zazwyczaj takie samo jak napięcie zasilania mikrokontrolera, w przypadku STM32 będzie to 3.3V. W trakcie zajęć ćwiczeniowych wykorzystamy wyjścia do sterowania diodami – stan wysoki będzie odpowiadał diodzie zapalanej, stan niski zgaszonej.

Wejście cyfrowe operuje na tych samych poziomach logicznych (0 – 0V, 1 – 3.3V), z tym że nie może zmieniać stanu logicznego, może jedynie go sprawdzać. Jeżeli pin będący wejściem będzie podłączony do masy odczytanie w programie jego stanu da nam wynik zero. Jeśli zaś będzie podpięty do plusa zasilania otrzymamy wynik jeden. Najprostszym zastosowaniem pinu wejścia jest obsługa przycisku. Na płytce STM32F429i-Discovery znajduje się przycisk który w stanie spoczynku zwiera wejście PA0 do masy, dając niski stan logiczny, wciśnięcie go zwiera wejście do plusa i zmienia ten stan na wysoki.



LEDs



USER & WAKE-UP Button