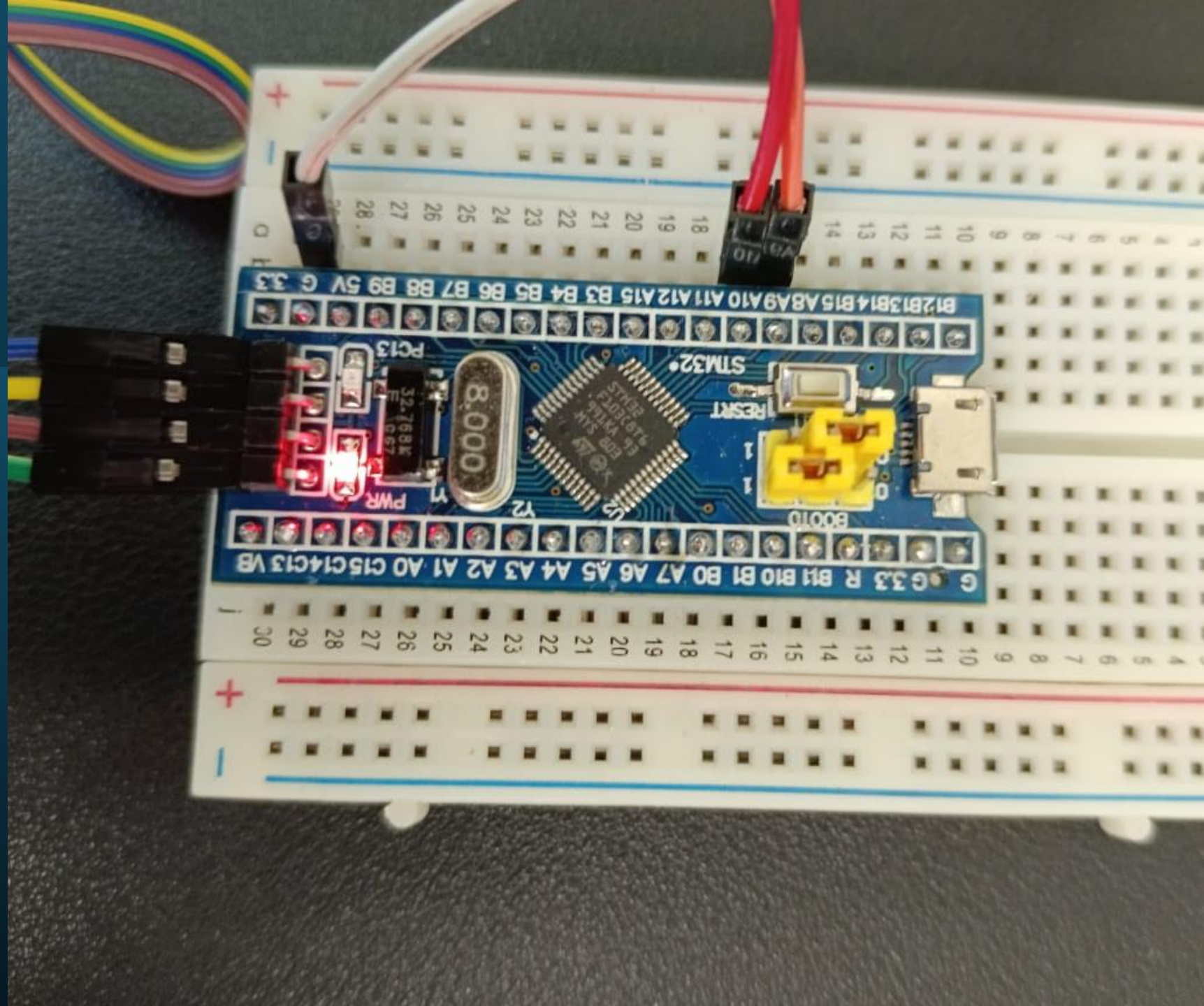
PM Ir Dr Tee Kian Sek
Faculty of Electrical and Electronic Engineering
Universiti Tun Hussein Onn Malaysia

Date: May 2024

Getting Start with STM32 using STM32CubeIDE and HAL



Instructors

PM Ir Dr Tee Kian Sek
https://community.uthm.edu.my/staff/people/tee

Dr Chew Chang Choon
https://community.uthm.edu.my/chewcc

Outlines

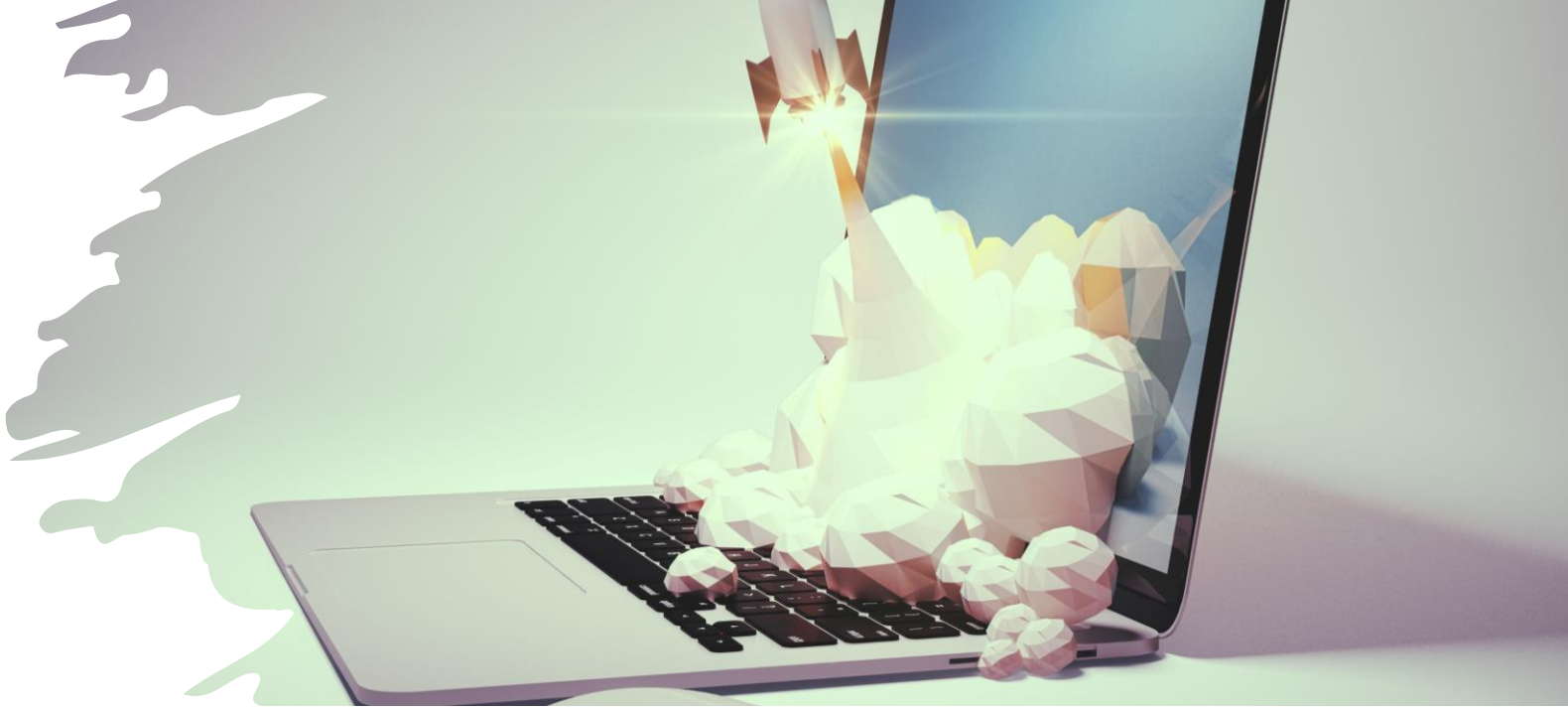
Cognition – Good to know!

- Objectives of this short course
- Arduino and STM32 – Learning scope
- Brief on ARM and STM32
- What is MCU?

This Short Course (Hands-on)

- The training kit
- The preparation
- First Project – very important
- Practices

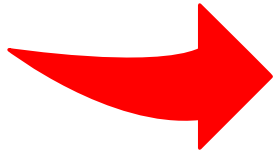
Objectives of this short course



Arduino and STM32 – Learning scopes

Arduino UNO

- ATmega328P
- Which company? Who cares?
- Datasheet – anyone read?
- IDE and libraries? Many from third parties.
- Libraries – who read them?



STM32

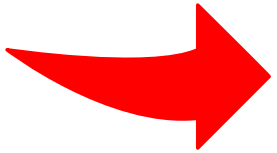
- STM32 MCU families
- ST
- Datasheet – we do care
- IDE and HAL.
- Copy customized .c and .h

All hardware engineers care about spec/datasheets on all MCUs and ICs.

Arduino and STM32 – Learning scopes

Arduino UNO

- Exploring registers?
- Is anyone trying to understand them?
- I2C/SPI electronic parts?
- Interrupts / Timer / PWM / UART / ADC ?
- Live debugging?



STM32

- We do care the registers
- We do care – Specification/datasheet of I2C/SPI electronic parts
- Registers for Interrupts / Timer / PWM / USART / DMA /USB / ADC / RTC
- Live debugging

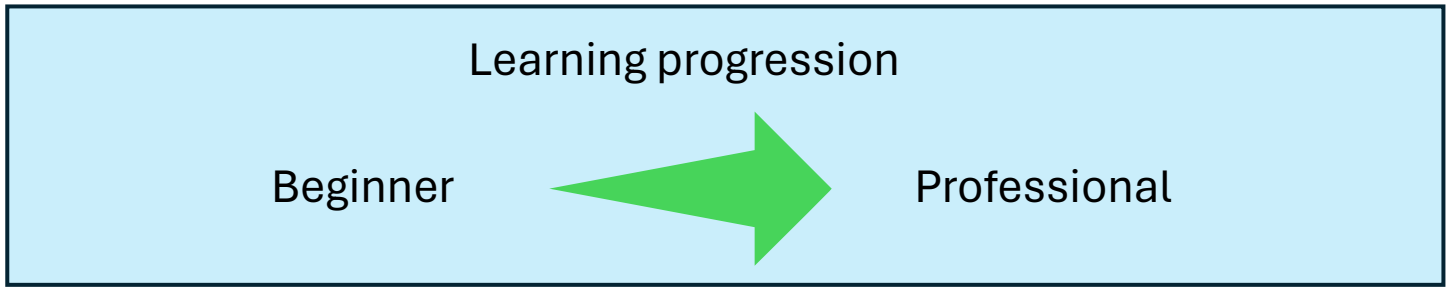
Arduino and STM32 – Learning scopes

Arduino UNO

- Anyone care?
- Production?
- Flashing ? duplicates, erase, password protected?

STM32

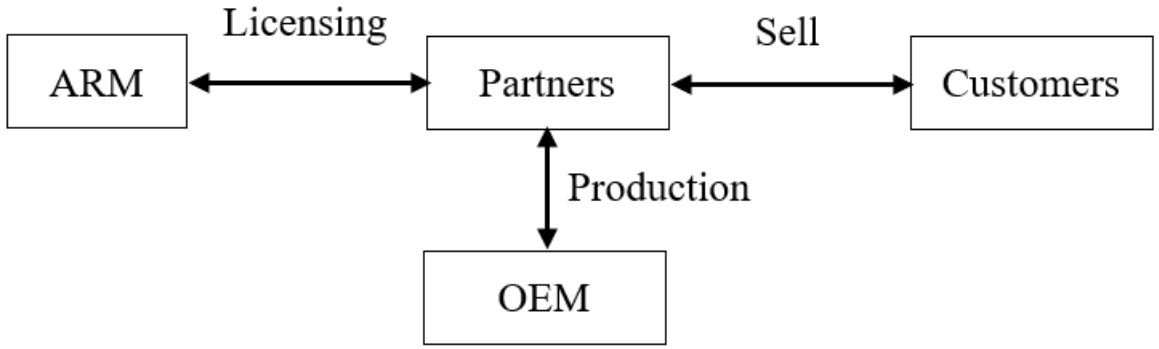
- Part of the exercises
- Tools for flashing



** ATmega328P could be flashed in production.*

Brief on ARM and STM32

ARM Licensing model (simplified)



Partners

1. Broadcom
2. Apple
3. ST
4. Microchip
5. Etc.

- ARM licenses IP to over 1,000 global partners (including Samsung, Apple, Microsoft).
- See modules

See the website for Licensing Model Options

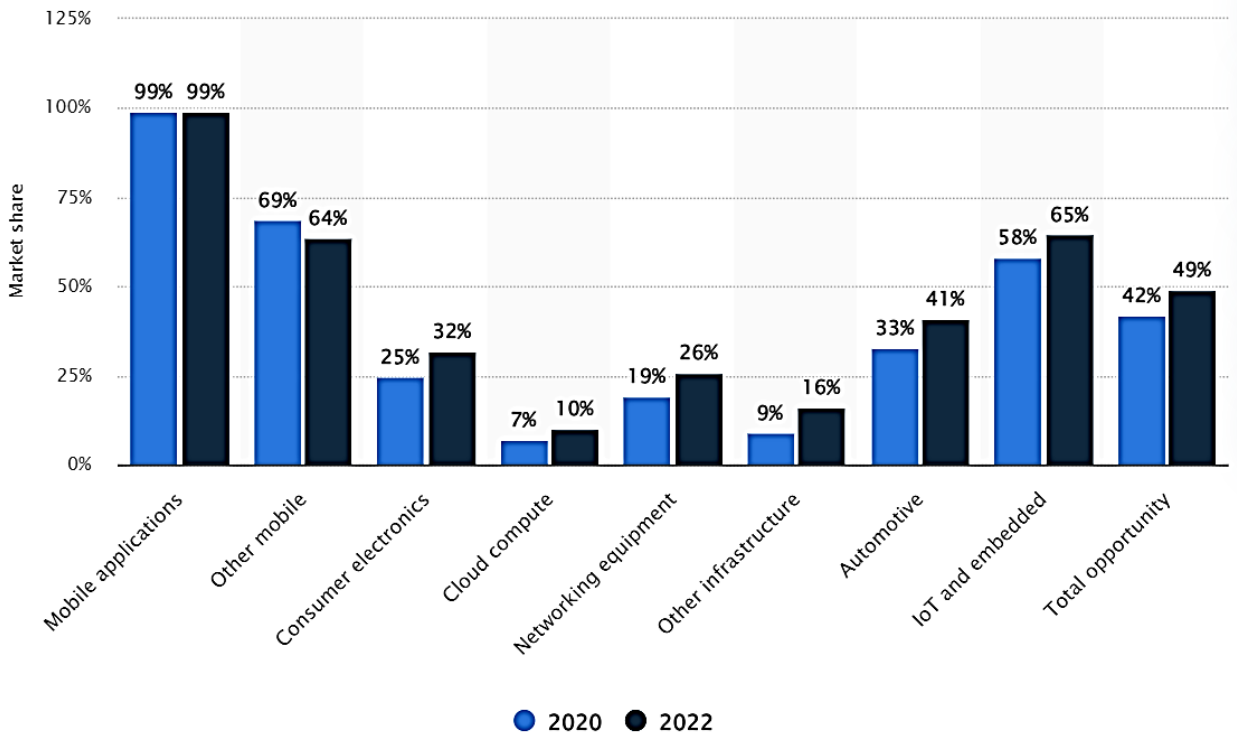
**** MCU, MP, GPU, , AI, IoT, etc.***

Brief on ARM and STM32

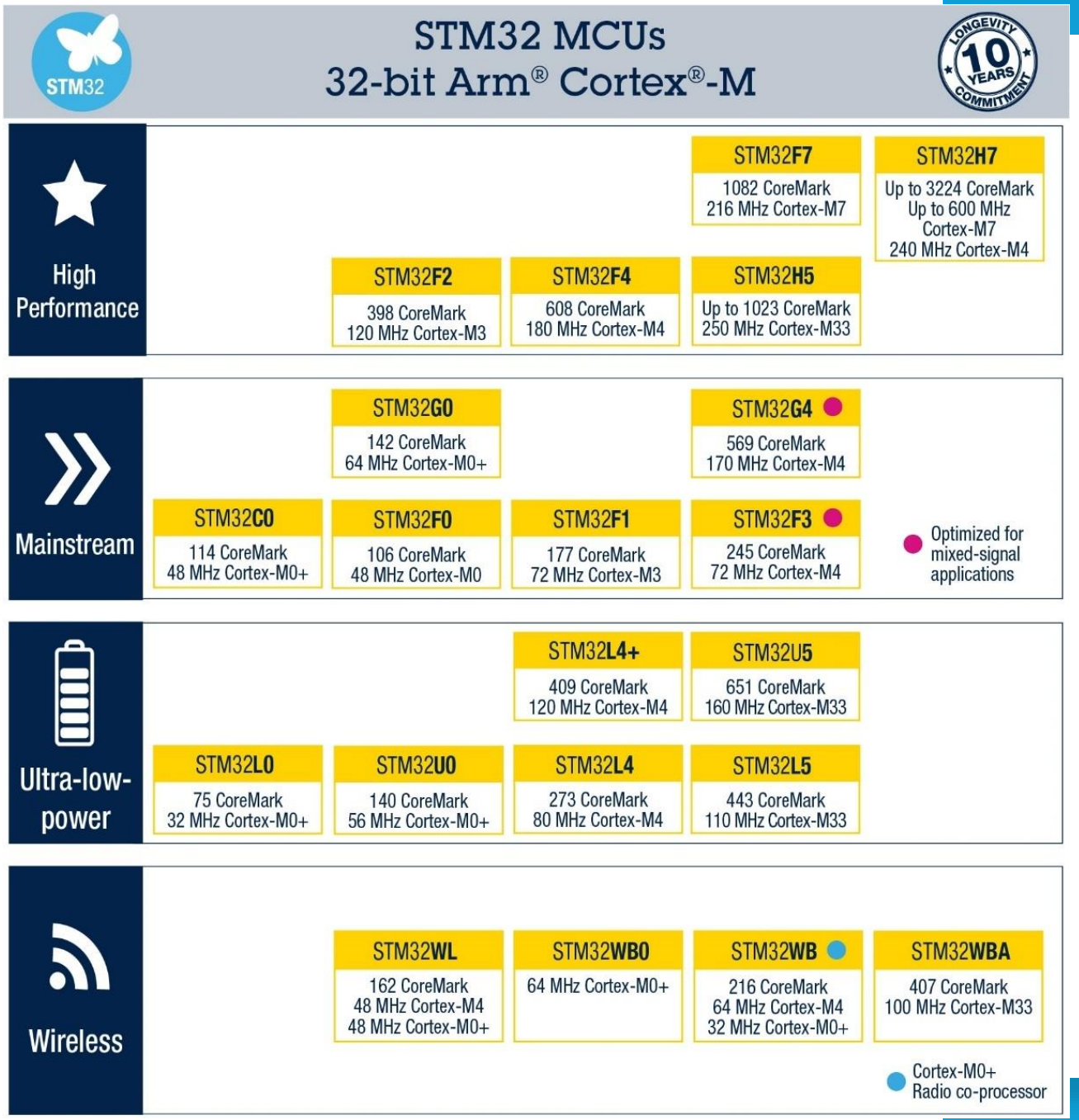
Popular

1. Low energy consumption, low cost, high performance
2. Support 16/32 instruction sets
3. Many partnerss
4. Rich ecosystem
5. Many more reasons...

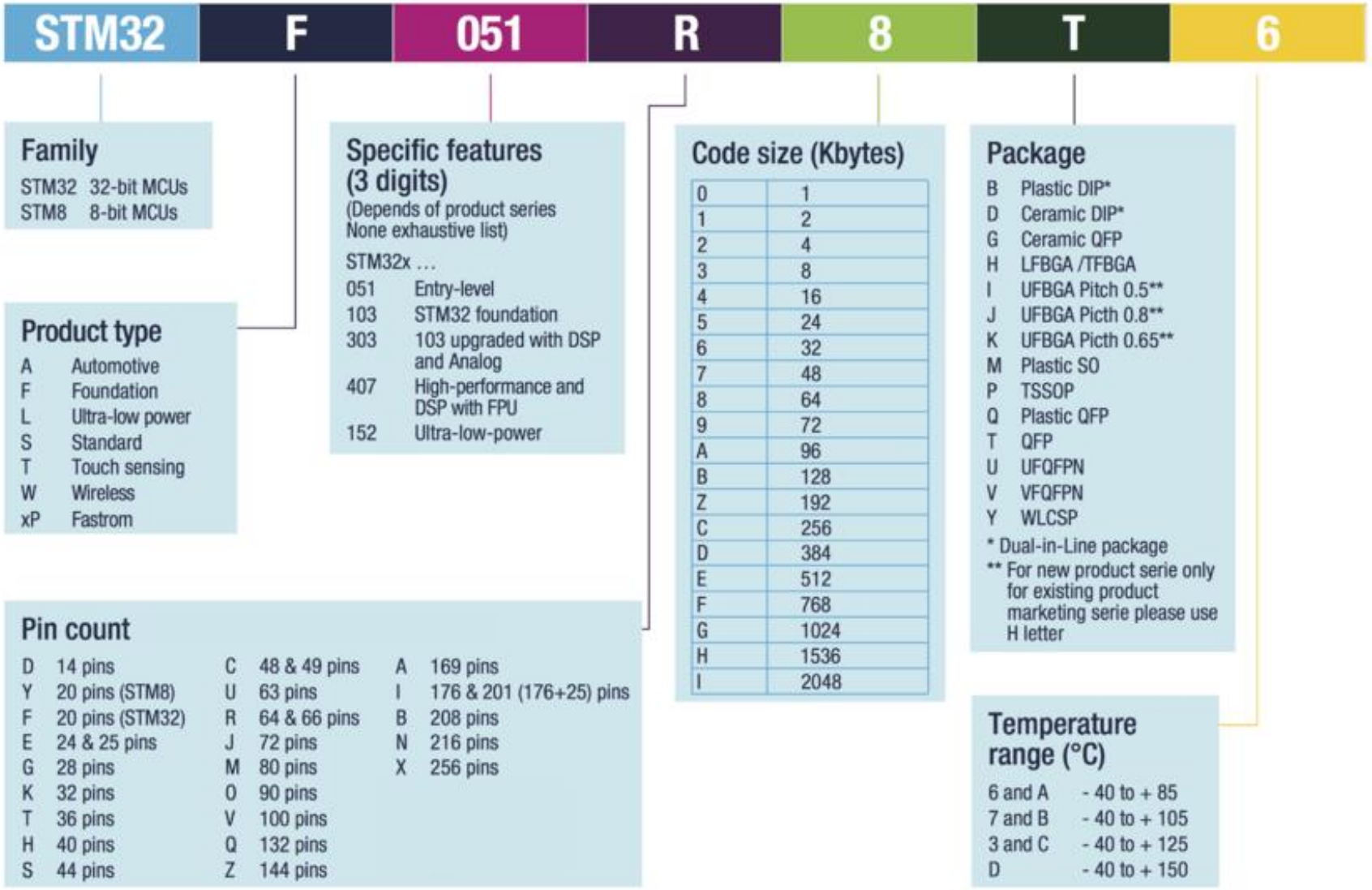
Market



ARM and STM32

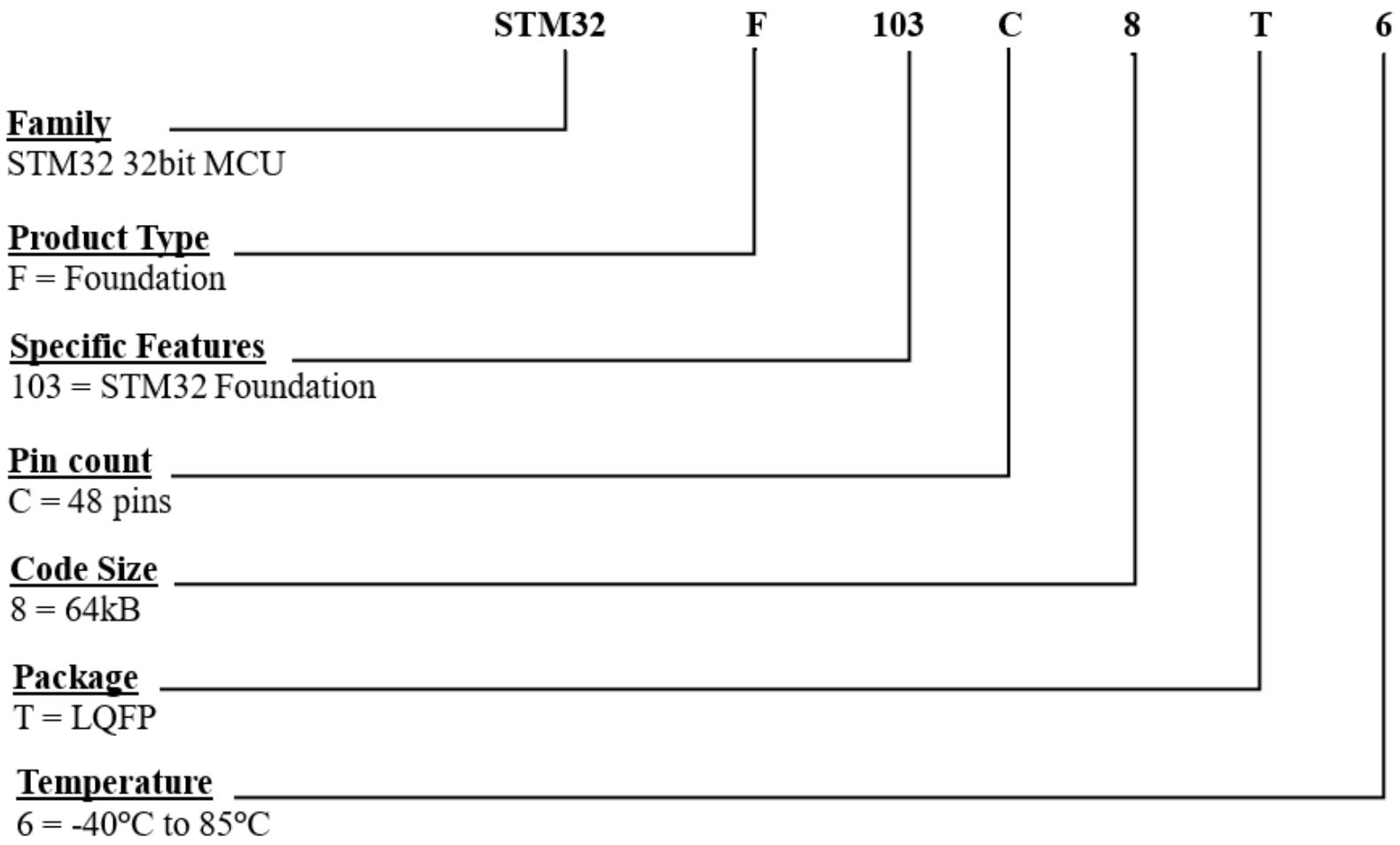


STM32 MCU


STM32 F 051 R 8 T 6
Family
STM32 32-bit MCUs
STM8 8-bit MCUs
Product type
A Automotive
F Foundation
L Ultra-low power
S Standard
T Touch sensing
W Wireless
xP Fastrom
Specific features (3 digits)
(Depends of product series None exhaustive list)
STM32x ...
051 Entry-level
103 STM32 foundation
303 103 upgraded with DSP and Analog
407 High-performance and DSP with FPU
152 Ultra-low-power
Code size (Kbytes)
0 1
1 2
2 4
3 8
4 16
5 24
6 32
7 48
8 64
9 72
A 96
B 128
Z 192
C 256
D 384
E 512
F 768
G 1024
H 1536
I 2048
Package
B Plastic DIP*
D Ceramic DIP*
G Ceramic QFP
H LFBGA /TFBGA
I UFBGA Pitch 0.5**
J UFBGA Picth 0.8**
K UFBGA Picth 0.65**
M Plastic SO
P TSSOP
Q Plastic QFP
T QFP
U UFQFPN
V VFQFPN
Y WLCSP
* Dual-in-Line package
** For new product serie only for existing product marketing serie please use H letter
Pin count
D 14 pins
Y 20 pins (STM8)
F 20 pins (STM32)
E 24 & 25 pins
G 28 pins
K 32 pins
T 36 pins
H 40 pins
S 44 pins
C 48 & 49 pins
U 63 pins
R 64 & 66 pins
J 72 pins
M 80 pins
O 90 pins
V 100 pins
Q 132 pins
Z 144 pins
A 169 pins
I 176 & 201 (176+25) pins
B 208 pins
N 216 pins
X 256 pins
Temperature range (°C)
6 and A - 40 to + 85
7 and B - 40 to + 105
3 and C - 40 to + 125
D - 40 to + 150

Example

STM32 F103C8T6



What is MCU?

- Quote from Wiki –

"A microcontroller (MC, UC, or µC) or microcontroller unit (MCU) is a small computer on **a single integrated circuit**. A microcontroller contains one or more CPUs (**processor cores**) along with **memory** and **programmable input/output peripherals**."

What is MCU?

Embedded System

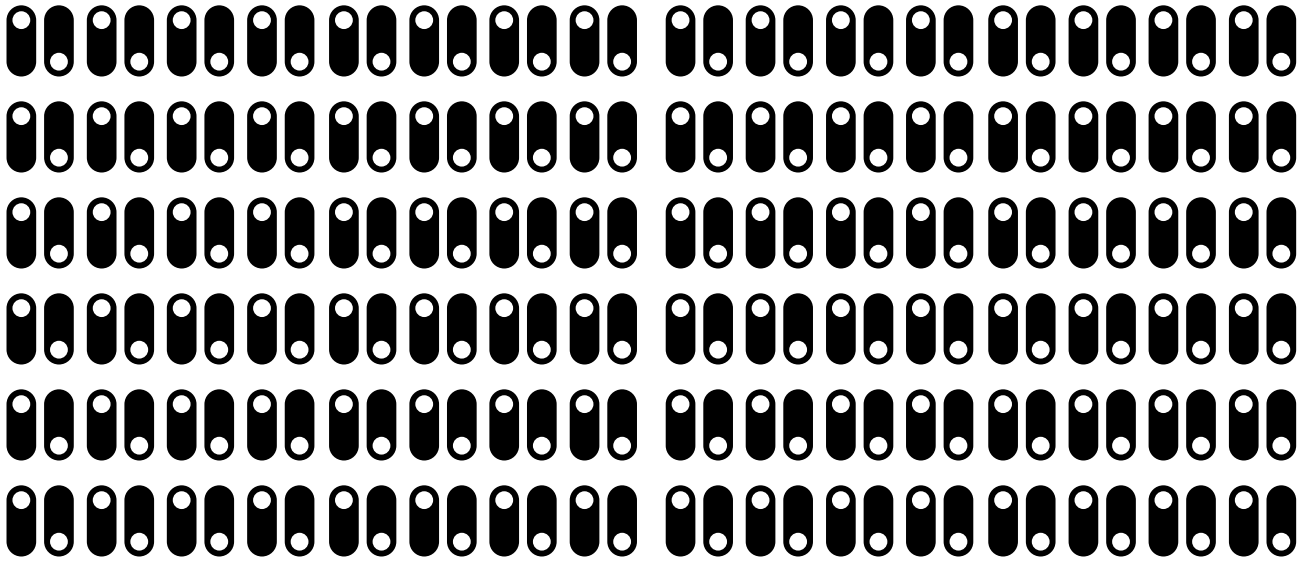
- Used in many applications
- home appliances, automotive systems, medical devices, and industrial control systems.
- consumer electronics products, such as gaming systems, digital cameras, and audio players.

What is MCU?

Registers – Nontechnical concepts of “configurable switches”

RCC, BUS, GPIO, USART, TIMER, etc....

switches



Many more....

Imagine we have a house with a wall of switches.
We know which switch to turn on a light!

What is MCU? – Register?

Example:

We want to blink an LED at PC13.

HAL is writing to a register Namely BSRR,

Set ON/OFF at GPIOC, pin 13

9.2.5 Port bit set/reset register (GPIOx_BSRR) (x=A..G)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:16 **BRy:** Port x Reset *bit y (y= 0 .. 15)*

These bits are write-only and can be accessed in Word mode only.

0: No action on the corresponding ODRx bit

1: Reset the corresponding ODRx bit

Note: If both BSx and BRx are set, BSx has priority.

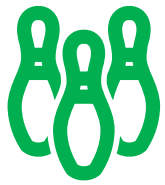
Bits 15:0 **BSy:** Port x *Set bit y (y= 0 .. 15)*

These bits are write-only and can be accessed in Word mode only.

0: No action on the corresponding ODRx bit

1: Set the corresponding ODRx bit

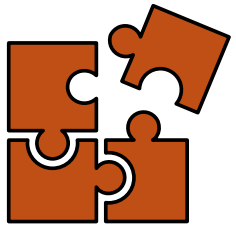
```
HAL_GPIO_WritePin(GPIOC,
GPIO_PIN_13, GPIO_PIN_RESET);
```



No worry! We are going to try this soon!

Refer to: rm0008- manual

What is HAL?



- The HAL driver layer provides a simple, generic multi-instance set of APIs (application programming interfaces) to interact with the upper layer (application, libraries and stacks).
- The HAL driver APIs are split into two categories: generic APIs, which provide common and generic functions for all the STM32 series and extension APIs, which include specific and customized functions for a given line or part number. The HAL drivers include a complete set of ready-to-use APIs that simplify the user application implementation

Refer to: UM1850 User manual - Description of STM32F1 HAL and low-layer drivers

Training Kit and MCU

The training kit – the outlook

- Pinouts
- Schematic

MCU – STM32F103C8T6

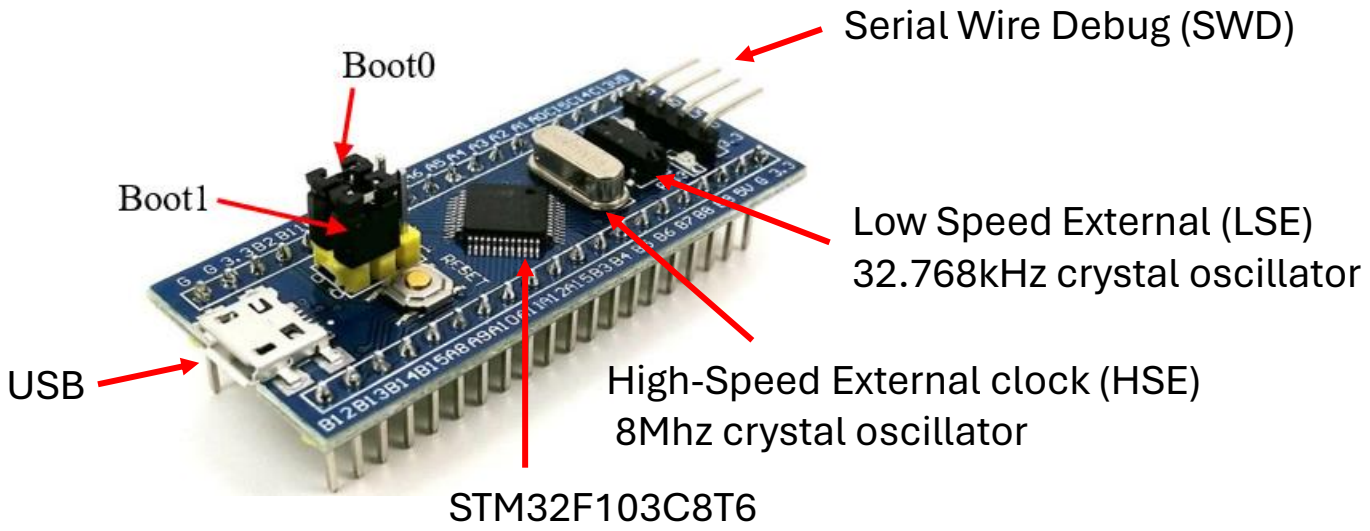
- Quick glimpse
- System Architecture
- Clock Tree
- GPIO

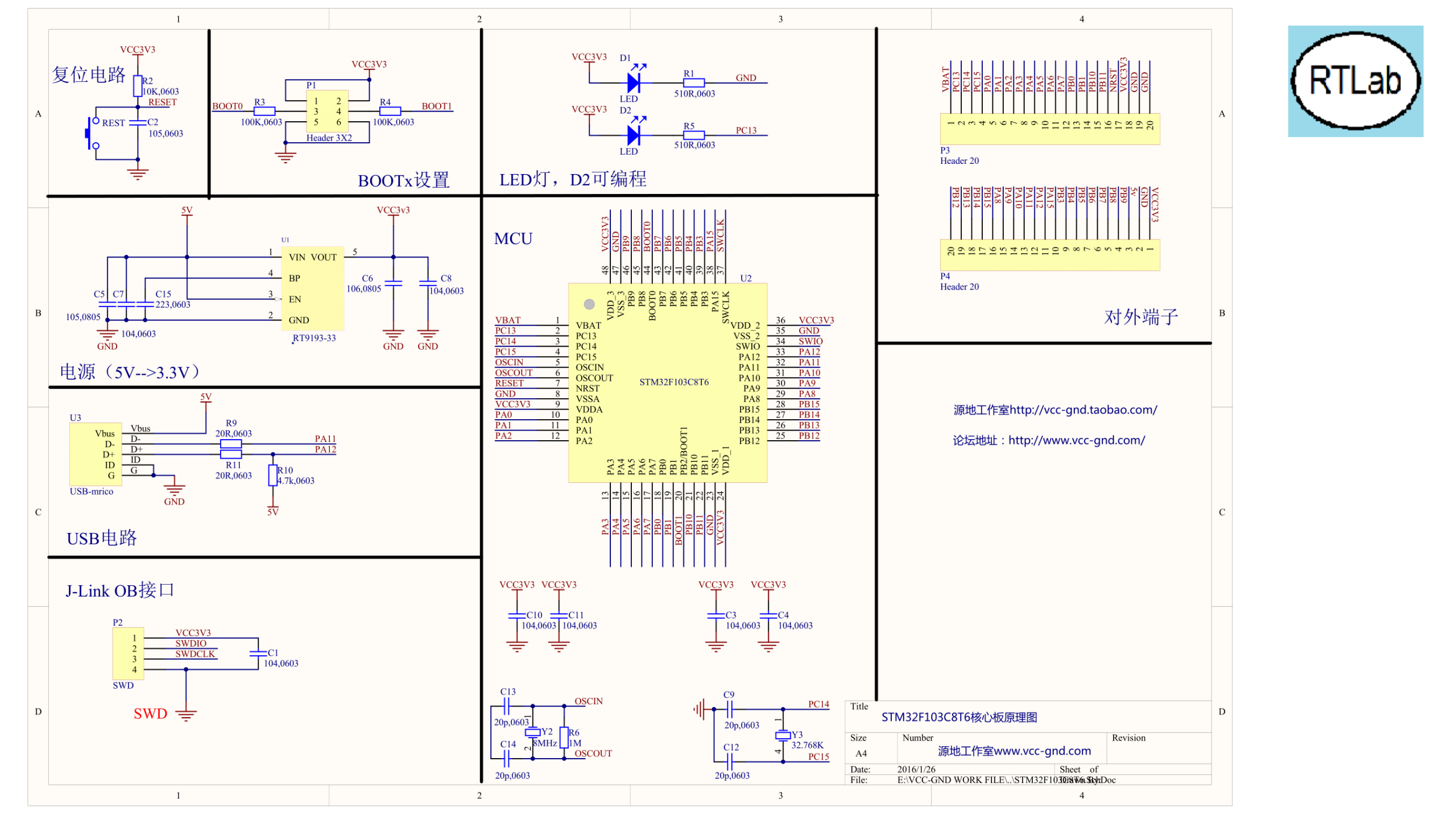
Many more on the datasheet and Manual as we explore further.

STM32F103C8T6 – datasheet

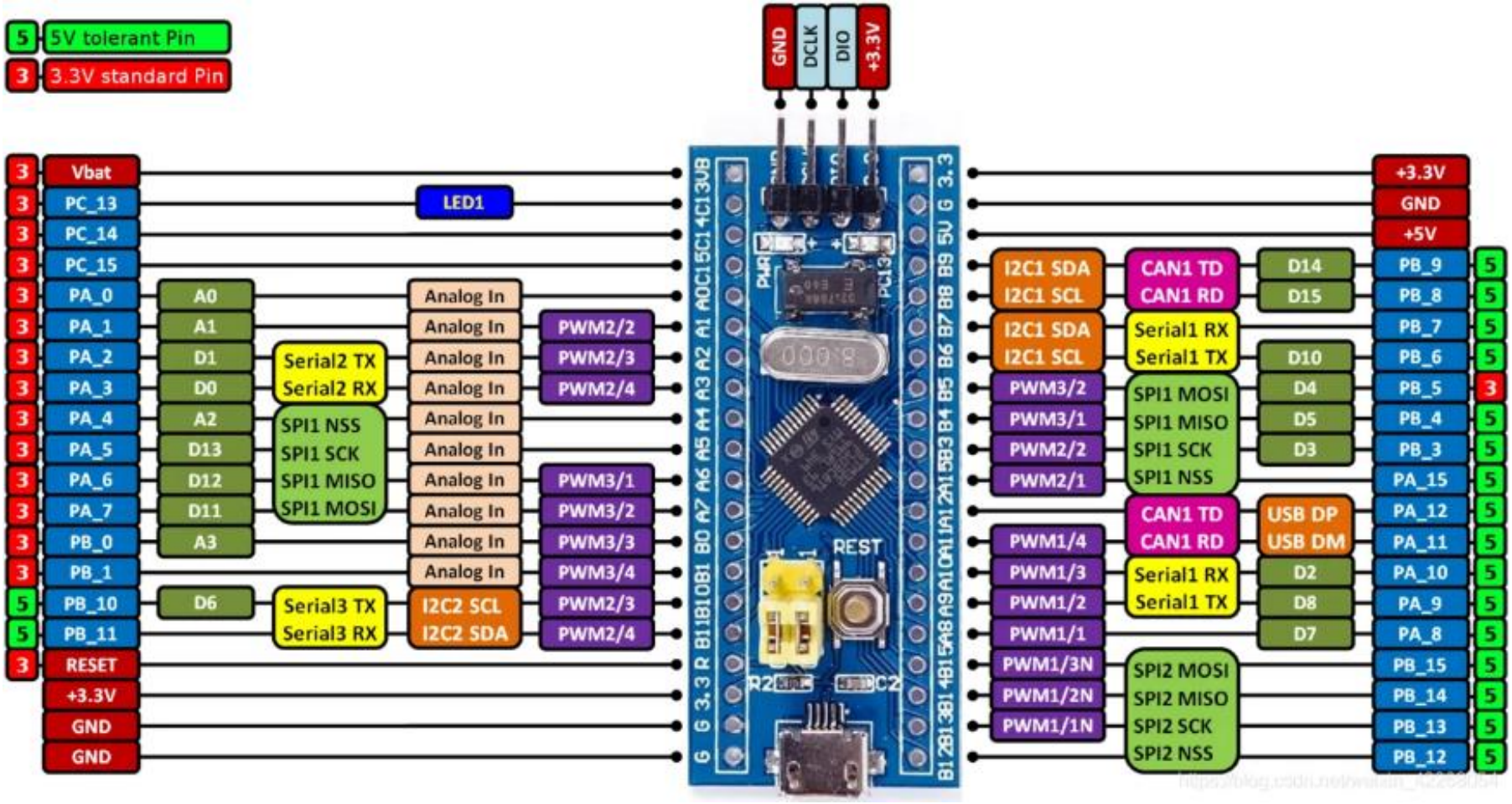
Table 9. Boot modes

Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Main Flash memory	Main Flash memory is selected as boot space
0	1	System memory	System memory is selected as boot space
1	1	Embedded SRAM	Embedded SRAM is selected as boot space




复位电路
VCC3V3
R2
10K,0603
RESET
REST
C2
105,0603
BOOT0
R3
100K,0603
P1
Header 3X2
R4
100K,0603
BOOT1
BOOTx设置
VCC3V3
D1
LED
R1
510R,0603
GND
VCC3V3
D2
LED
R5
510R,0603
PC13
LED灯，D2可编程
VBAT
PC13
PC14
PC15
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB10
PB11
NRST
VCC3V3
GND
GND
P3
Header 20
PB12
PB13
PB14
PB15
PA8
PA9
PA10
PA11
PA12
PA15
PB3
PB4
PB5
PB6
PB7
PB8
PB9
5v
GND
VCC3V3
P4
Header 20
对外端子
5V
U1
VIN VOUT
BP
EN
GND
RT9193-33
C5
C7
C15
223,0603
105,0805
104,0603
GND
VCC3v3
C6
106,0805
C8
104,0603
GND
GND
电源（5V-->3.3V）
MCU
U2
STM32F103C8T6
VBAT
PC13
PC14
PC15
OSCIN
OSCOUT
NRST
VSSA
VDDA
PA0
PA1
PA2
VDD_2
VSS_2
SWIO
PA12
PA11
PA10
PA9
PA8
PB15
PB14
PB13
PB12
源地工作室http://vcc-gnd.taobao.com/
论坛地址：http://www.vcc-gnd.com/
U3
Vbus
D-
D+
ID
G
USB-mrico
5V
R9
20R,0603
R11
20R,0603
PA11
PA12
R10
4.7k,0603
GND
5V
USB电路
J-Link OB接口
P2
SWD
VCC3V3
SWDIO
SWDCLK
C1
104,0603
SWD
C10
104,0603
C11
104,0603
C3
104,0603
C4
104,0603
C13
20p,0603
OSCIN
Y2
8MHz
R6
1M
C14
20p,0603
OSCOUT
C9
20p,0603
PC14
Y3
32.768K
C12
20p,0603
PC15
Title
STM32F103C8T6核心板原理图
Size
A4
Number
源地工作室www.vcc-gnd.com
Revision
Date:
2016/1/26
Sheet of
File:
E:\VCC-GND WORK FILE\..\STM32F103C8T6.SchDoc

STM32F103C8T6 – pins



Source: Downloaded online

STM32F103C8T6 - pin definition
LQFP 48 pins

RTLab

Find me!
STM32F103C8T6 -pin definition.xlsx

STM32F103C8T6 - pin definition
LQFP 48 pins

					Alternate functions	
Pin No	Pin Name	Type	IO Level	Main Function	Default	Remap
1	VBAT	S		VBAT		
2	PC13-TAMPER-RTC	I/O		PC13	TAMPER-RTC	
3	PC14-OSC32_IN	I/O		PC14	OSC32_IN	
4	PC15-OSC32_OUT	I/O		PC15	OSC32_OUT	
5	OSC_IN	I		OSC_IN		
6	OSC_OUT	O		OSC_OUT		
7	NRST	I/O		NRST		
8	VSSA	S		VSSA		
9	VDDA	S		VDDA		
10	**PA0-WKUP**	I/O		PA0	WKUP/USART2_CTS/ADC12_IN0/TIM2_CH1_ETR	
11	**PA1**	I/O		PA1	USART2_RTS/ADC12_IN1/TIM2_CH2	
12	**PA2**	I/O		PA2	USART2_TX/ADC12_IN2/TIM2_CH3	
13	**PA3**	I/O		PA3	USART2_RX/ADC12_IN3/TIM2_CH4	
14	**PA4**	I/O		PA4	SPI1_NSS/USART2_CK/ADC12_IN4	

Refer to the datasheet [Medium-density STM32F103xx pin definitions]

STM32F103C8T6 – Features and Peripherals

Term	Description	Term	Description
NVIC	Nested Vectored Interrupt Controller	CAN	CAN Comm.
SysTick	the Cortex® System Timer	USB	USB Comm.
RCC	Reset and Clock Control	RTC	Real-time clock
GPIO	General-purpose I/O	CRC	Cyclic Redundancy Check
AFIO	Alternate-function I/O	PWR	Power Control
EXTI	External interrupt/event controller	BKP	Backup registers
TIM	Timer	IWDG	Independent watchdog
ADC	Analog-to-Digital Converter	WWDG	Window watchdog
DMA	Direct memory access	DAC	Digital-to-analog converter
USART	USART Comm.	SDIO	SD Interface
I2C	I2C Comm.	FSMC	Flexible static memory controller
SPI	SPI Comm.	USB OTG	USB

STM32F103C8T6 – Important References

Documents

- Datasheets
- RM0008 Reference manual

Find me!

Websites

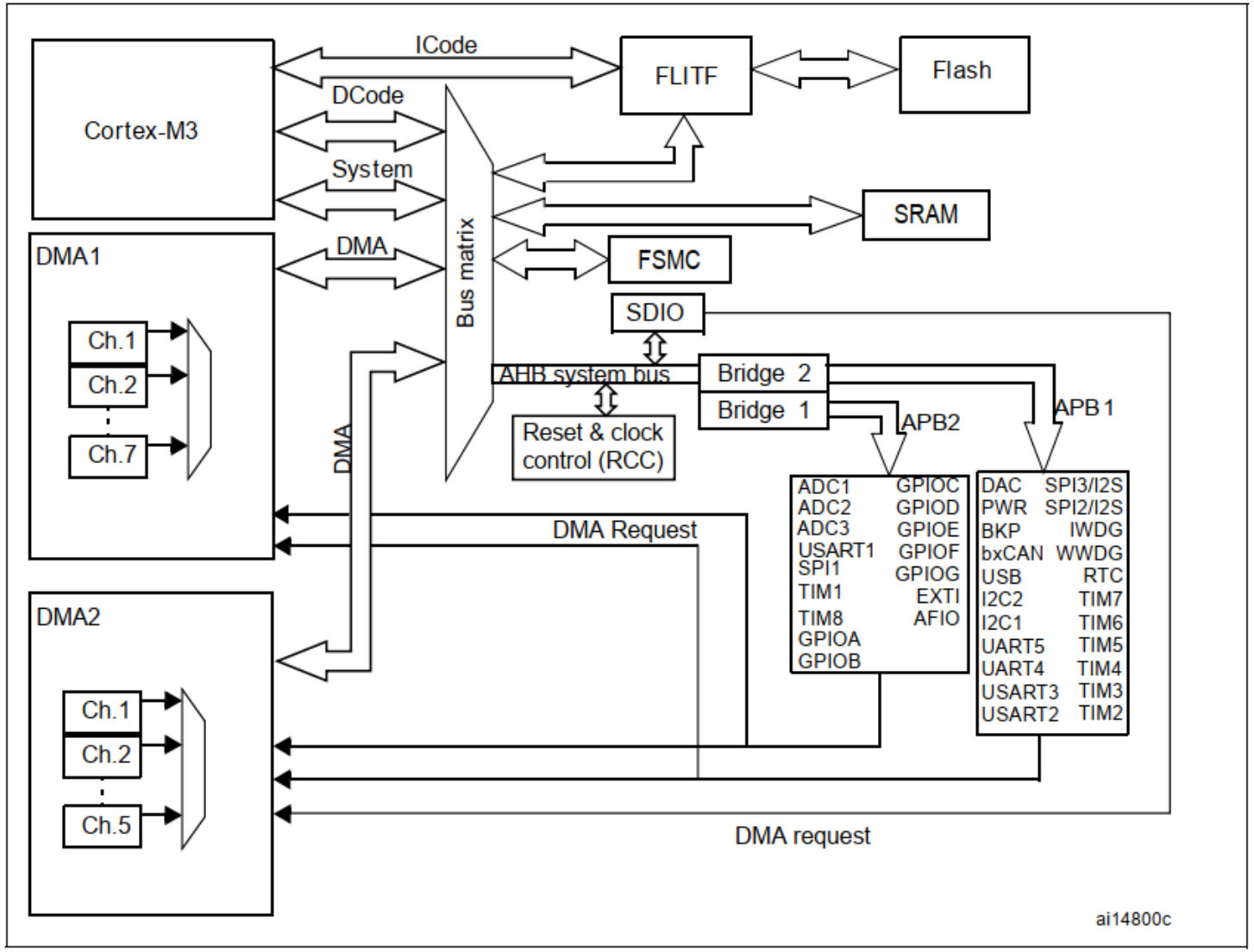
- Official site - ST

https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html

- Many learning sites…

System architecture

Figure 1. System architecture (low-, medium-, XL-density devices)



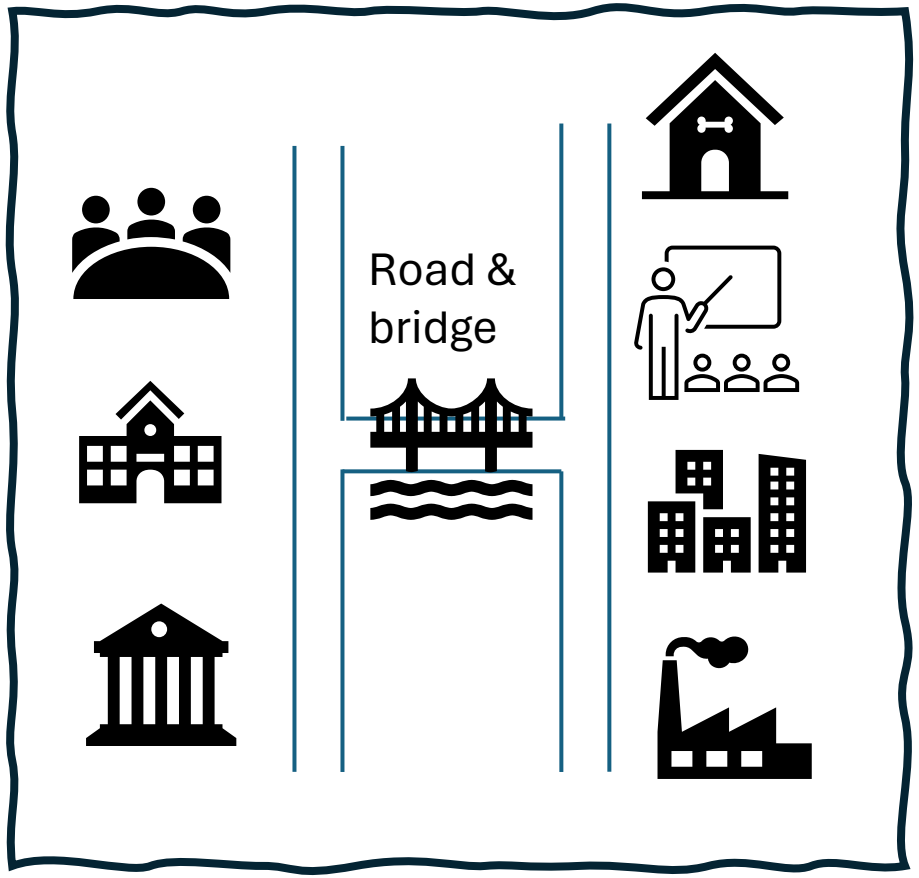
Advanced High-performance **Bus** (**AHB**)
Advanced Peripheral **Bus** (**APB**)

What is this?

System architecture

Simplified idea of “Town Plan”

<u>Administrators</u>
Council, town hall, bank, etc.

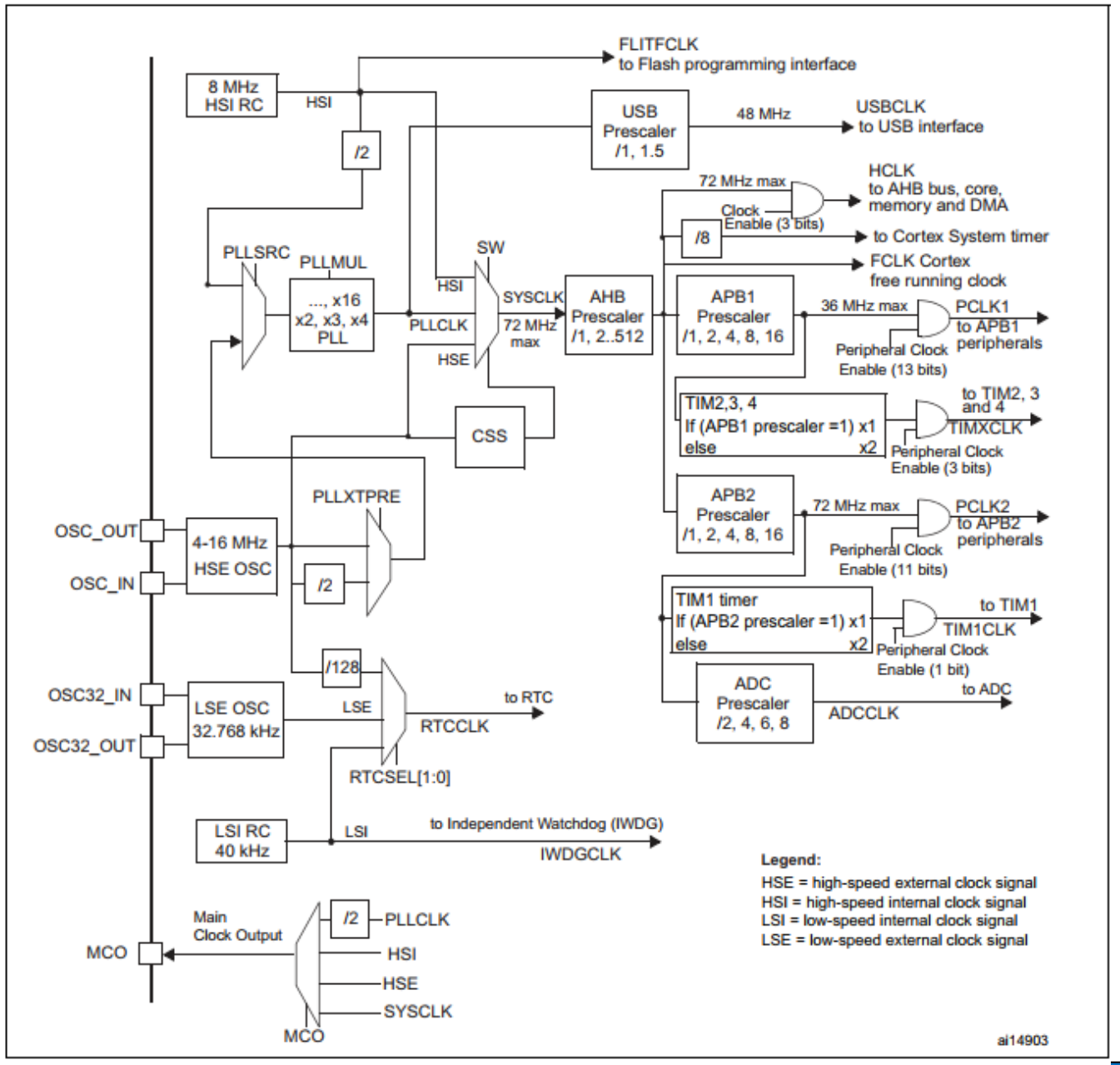


House, factory, commercial buildings, school, etc.

Clock Tree

Let me explain!

RTLab



High-Speed Internal clock (HSI) RC oscillator
High-Speed External clock (HSE) crystal oscillator
Phase-locked loop or phase lock loop (PLL) clock
Low Speed External (LSE)
Low Speed Internal (LSI)

STM32-F103C8T6 (See Schematic)
Crystal oscillator – 8MHz and 32.768kHz

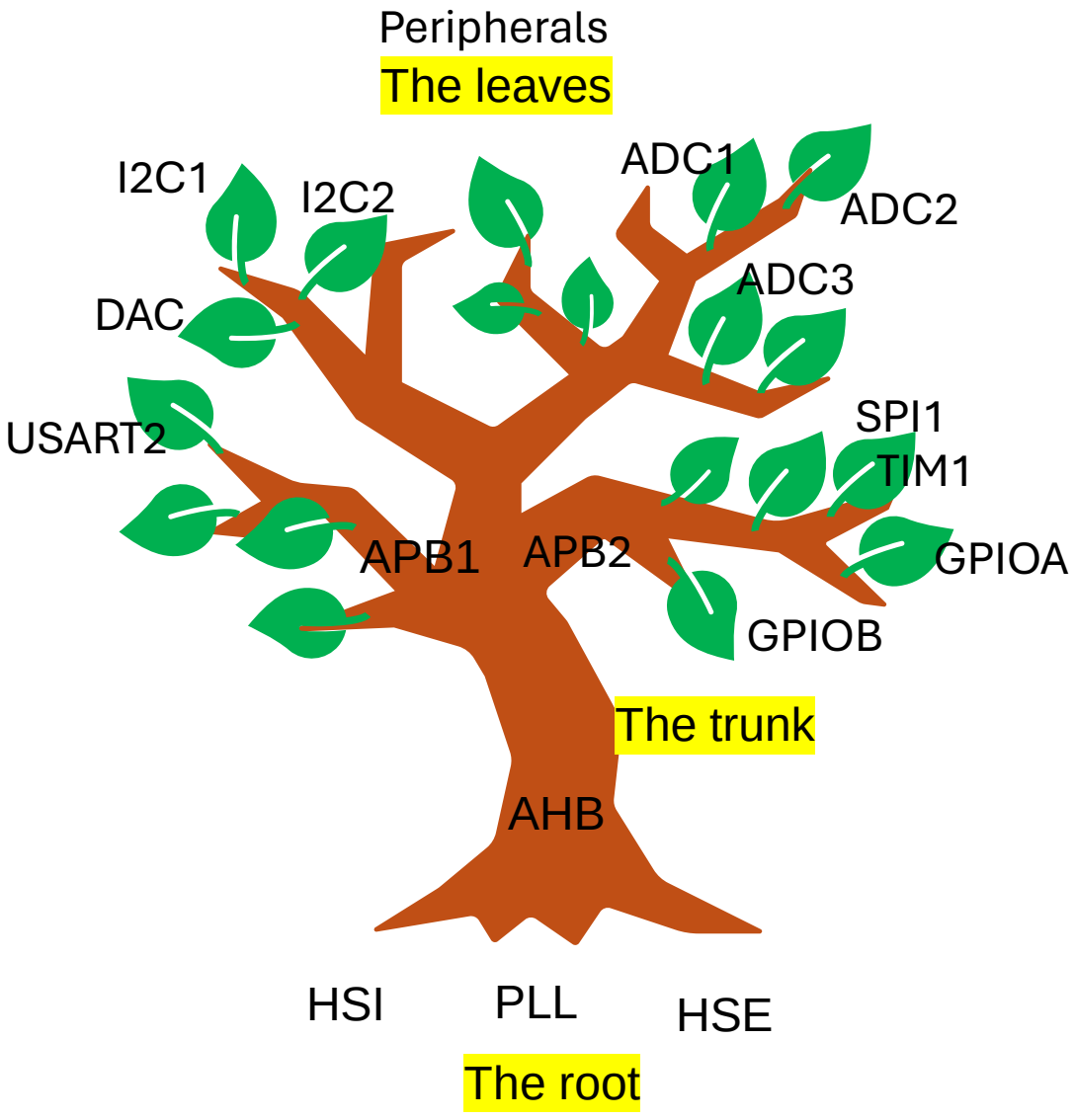
What is this?

STM32CubeIDE
Try to read together *.ioc | Clock Configuration

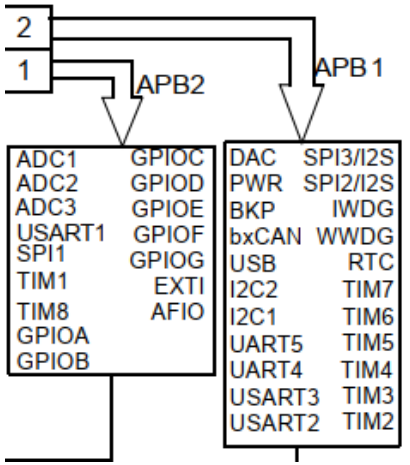
Clock Tree

Let me explain!

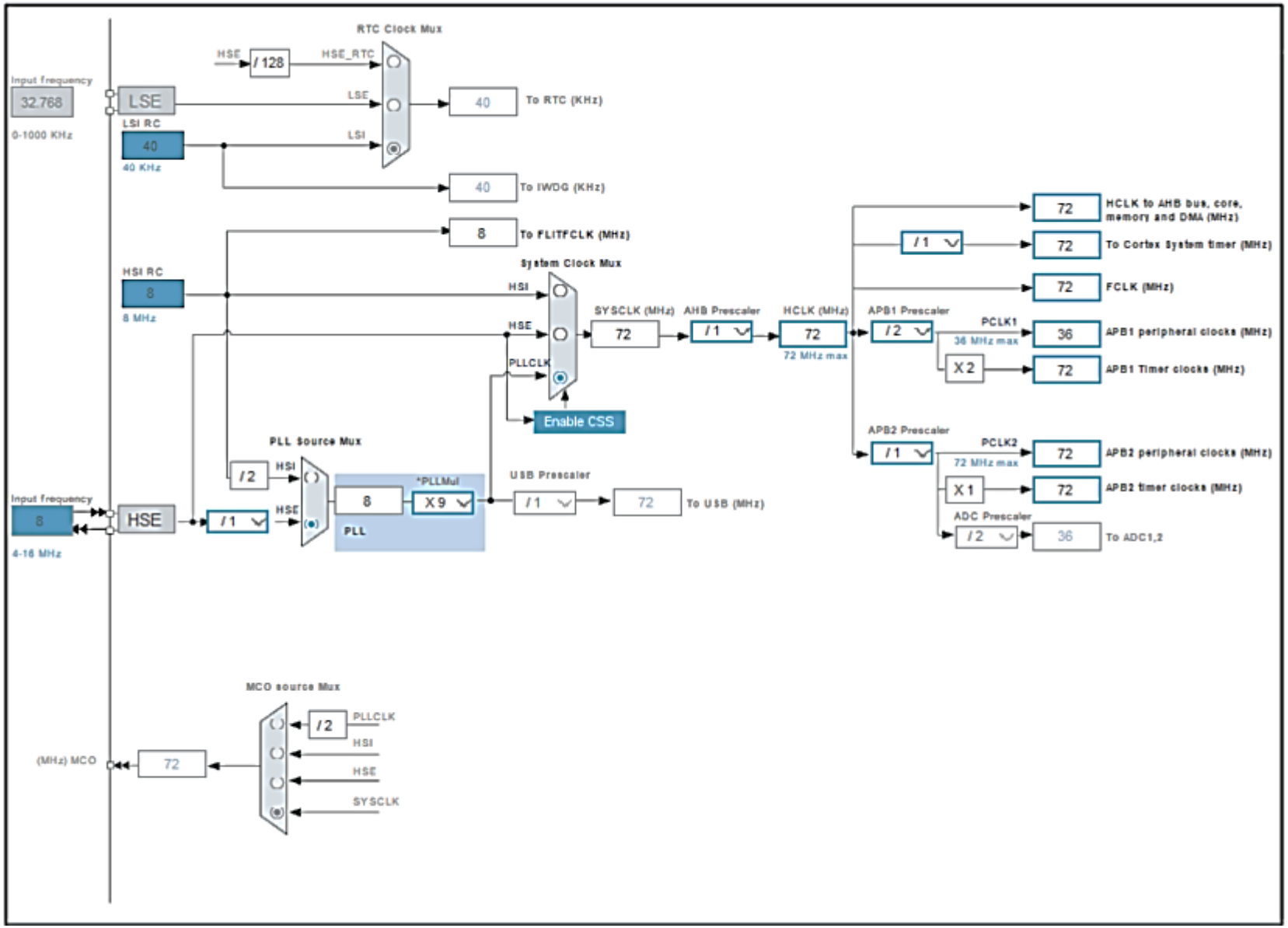
RTLab



Simplified idea of a “tree”



STM32CubeIDE
Try to read together *.ioc | Clock Configuration



Let me explain!

RTLab

Clock Tree

STM32CubeIDE

Try to read together *.ioc | Clock Configuration

GPIO

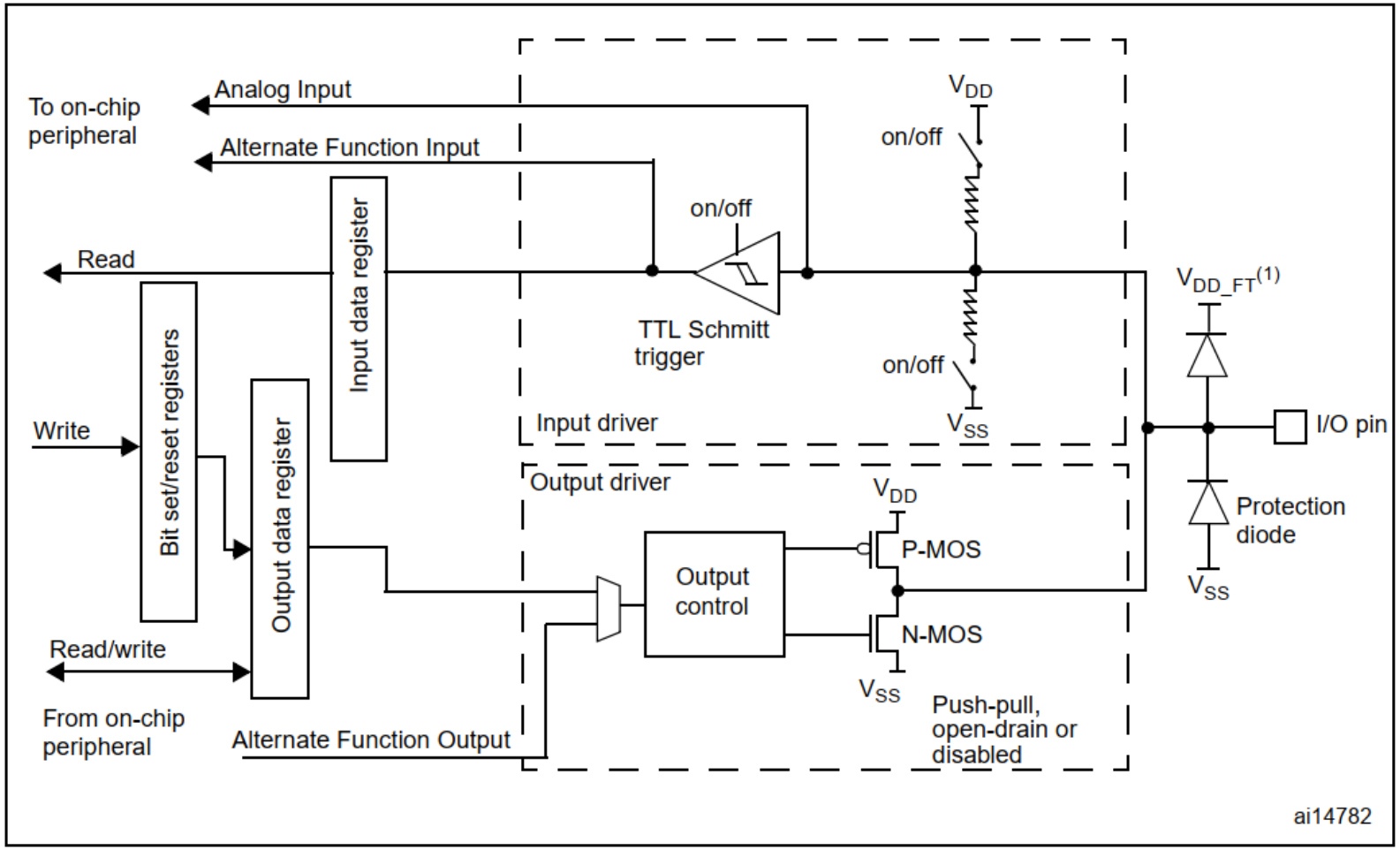
Let me explain!

RTLab

General Purpose IO (GPIO) Ports, can be individually configured by software in several modes:

- Input floating
- Input pull-up
- Input-pull-down
- Analog
- Output open-drain
- Output push-pull
- Alternate function push-pull
- Alternate function open-drain

Figure 14. Basic structure of a 5-Volt tolerant I/O port bit



1. V_{DD_FT} is a potential specific to 5-Volt tolerant I/Os, and different from V_{DD}.

GPIO

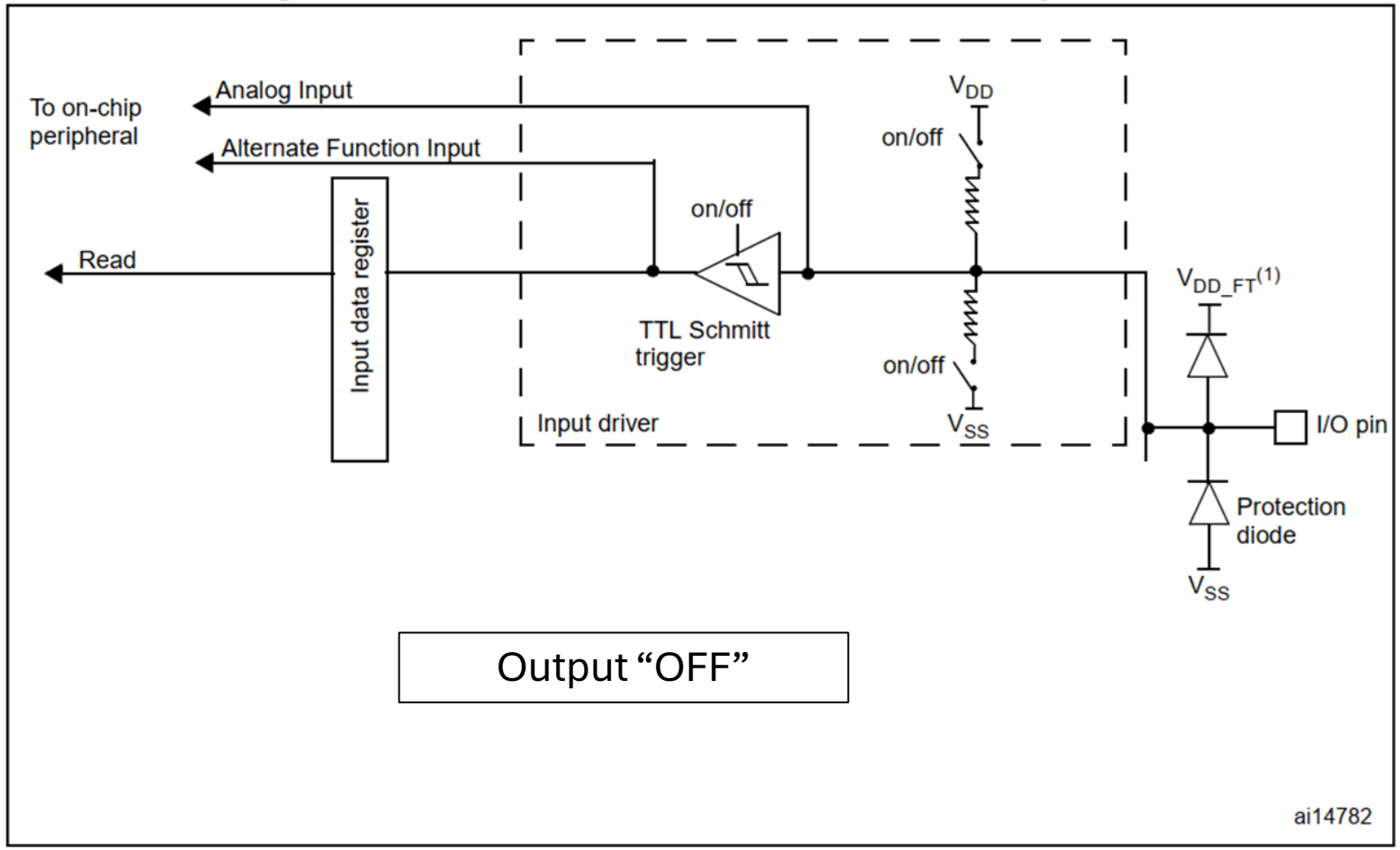
Let me explain!

RTLab

Input

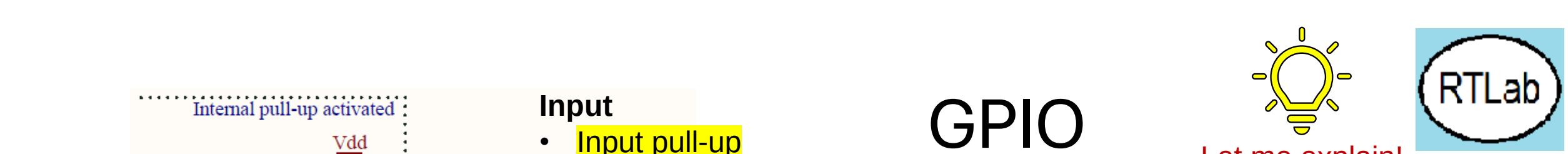
- Input floating
- Input pull-up
- Input-pull-down
- Analog

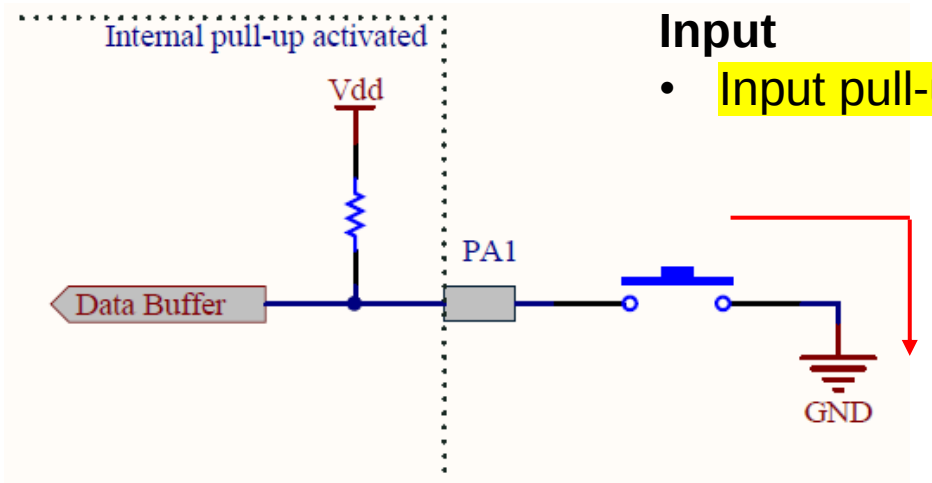
Figure 14. Basic structure of a 5-Volt tolerant I/O port bit



1. V_{DD_FT} is a potential specific to 5-Volt tolerant I/Os, and different from V_{DD}.

GPIO

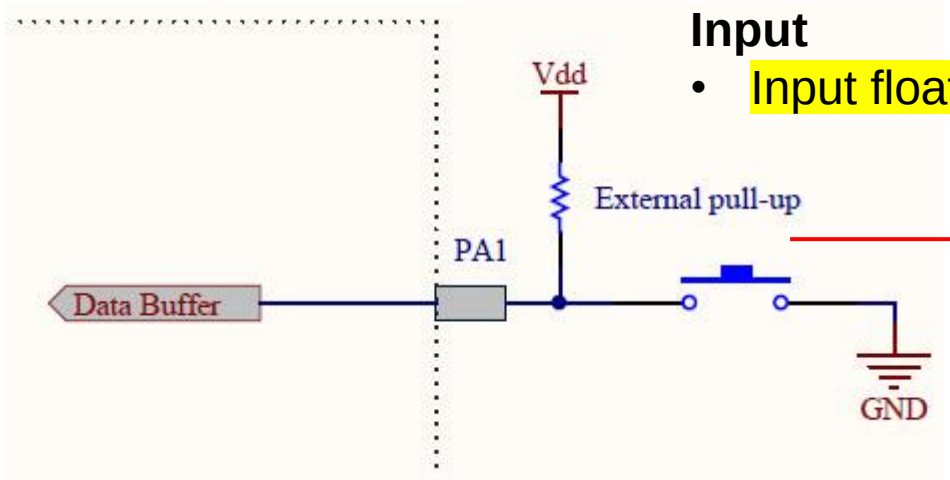




Input

- Input pull-up

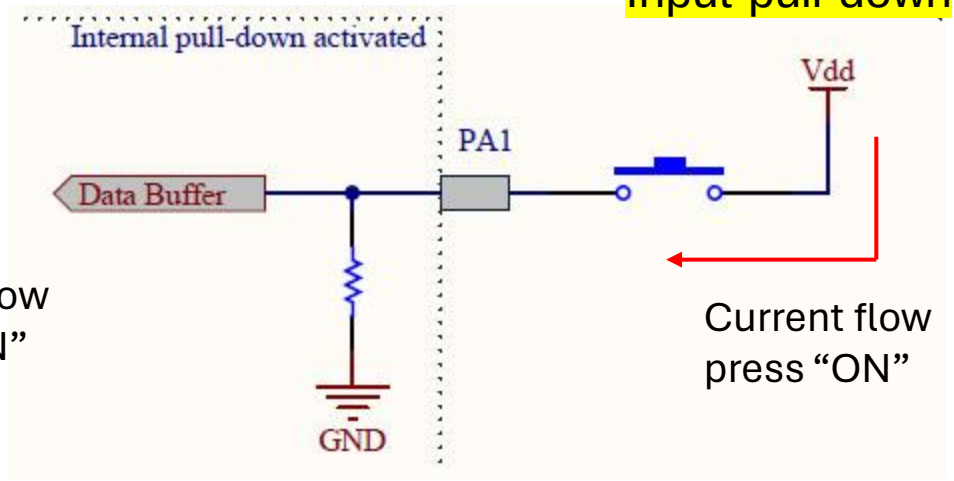
Current flow
press “ON”



Input

- Input floating

Current flow
press “ON”



Input

- Input-pull-down

Current flow
press “ON”

GPIO

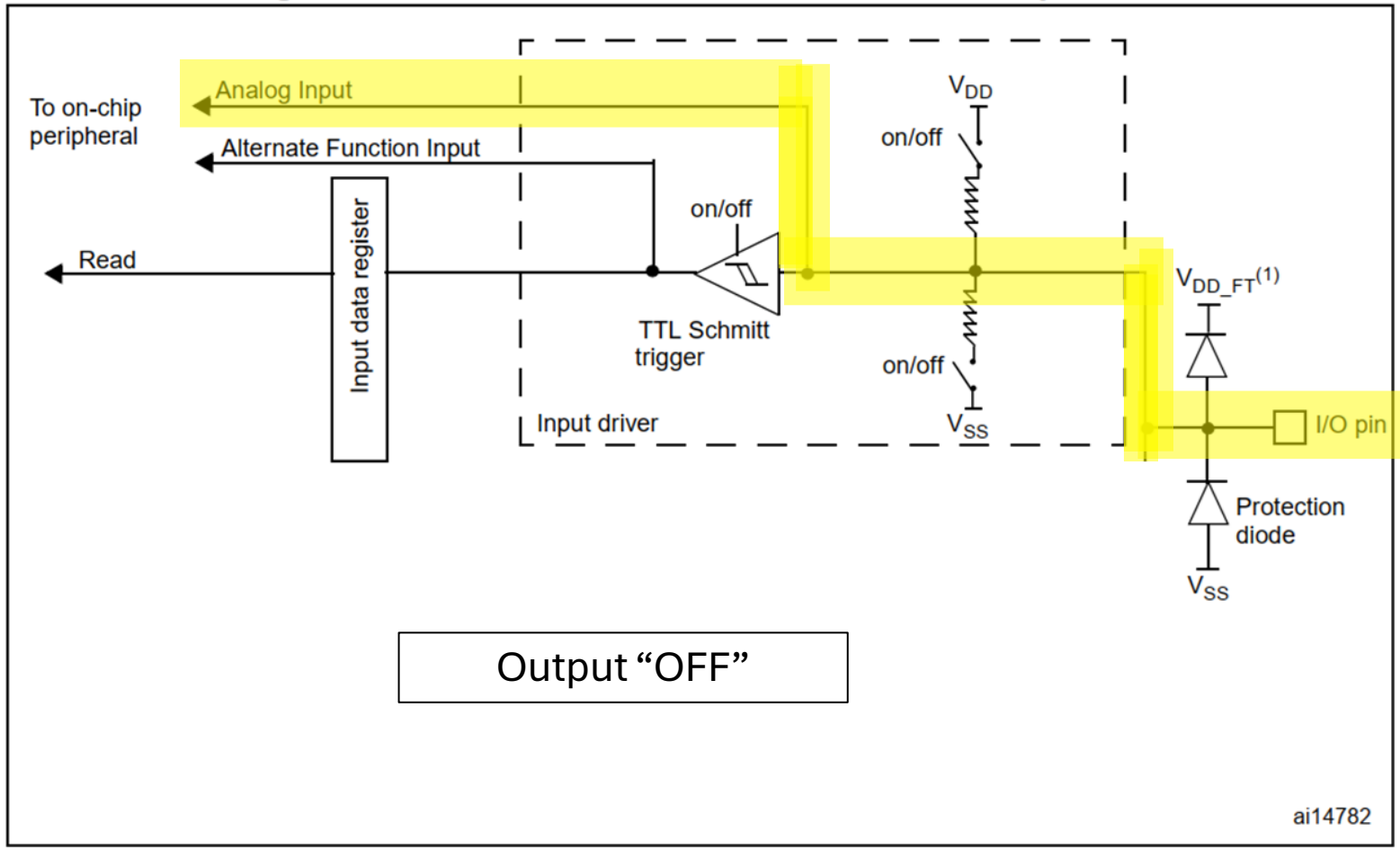
Let me explain!

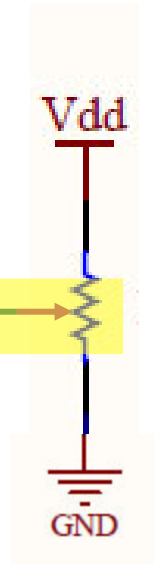
RTLab

Input

- Analog

Figure 14. Basic structure of a 5-Volt tolerant I/O port bit





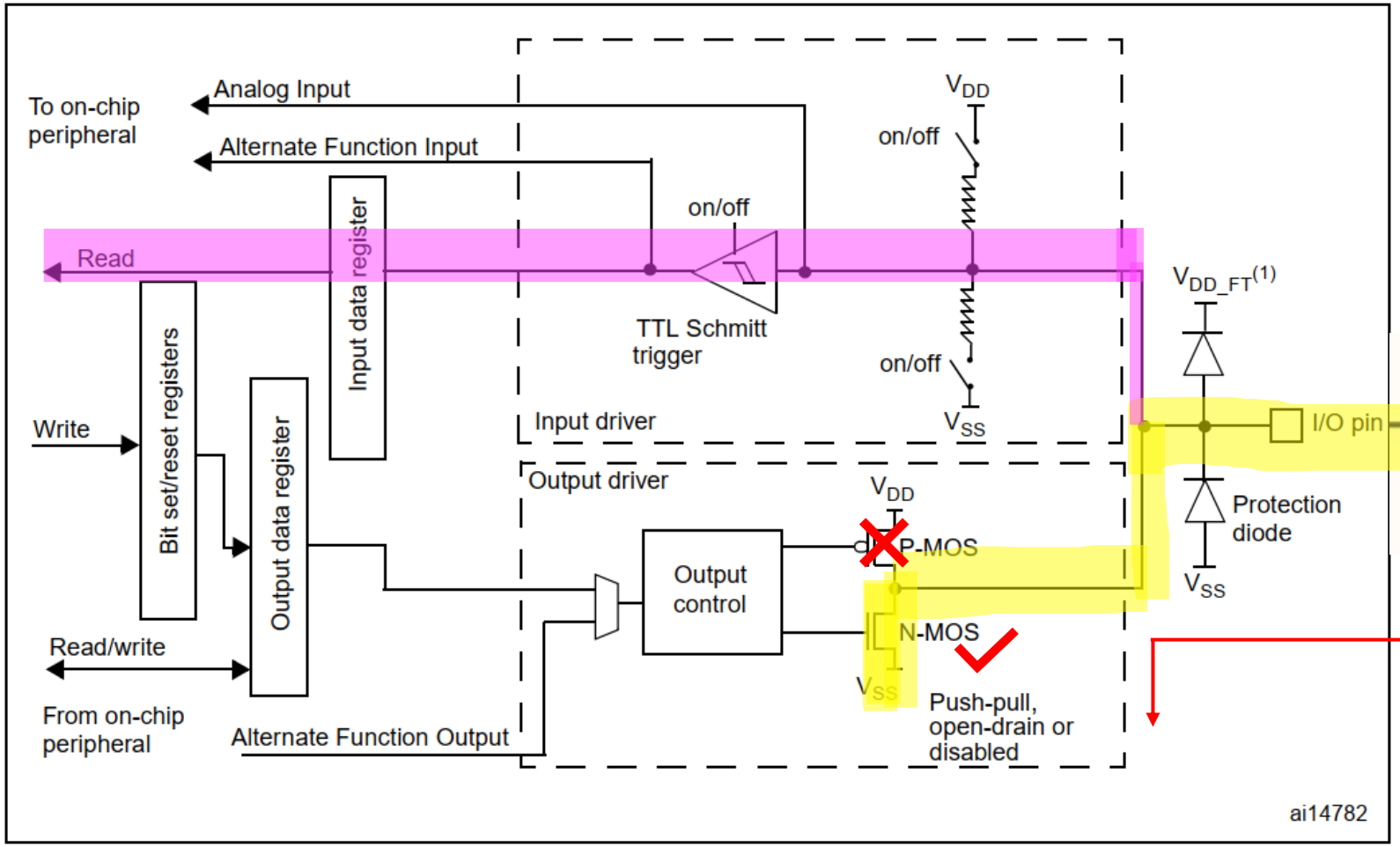
Output “OFF”

1. V_{DD_FT} is a potential specific to 5-Volt tolerant I/Os, and different from V_{DD}.

GPIO

Let me explain!

Figure 14. Basic structure of a 5-Volt tolerant I/O port bit



1. V_{DD_FT} is a potential specific to 5-Volt tolerant I/Os, and different from V_{DD}.

Output,

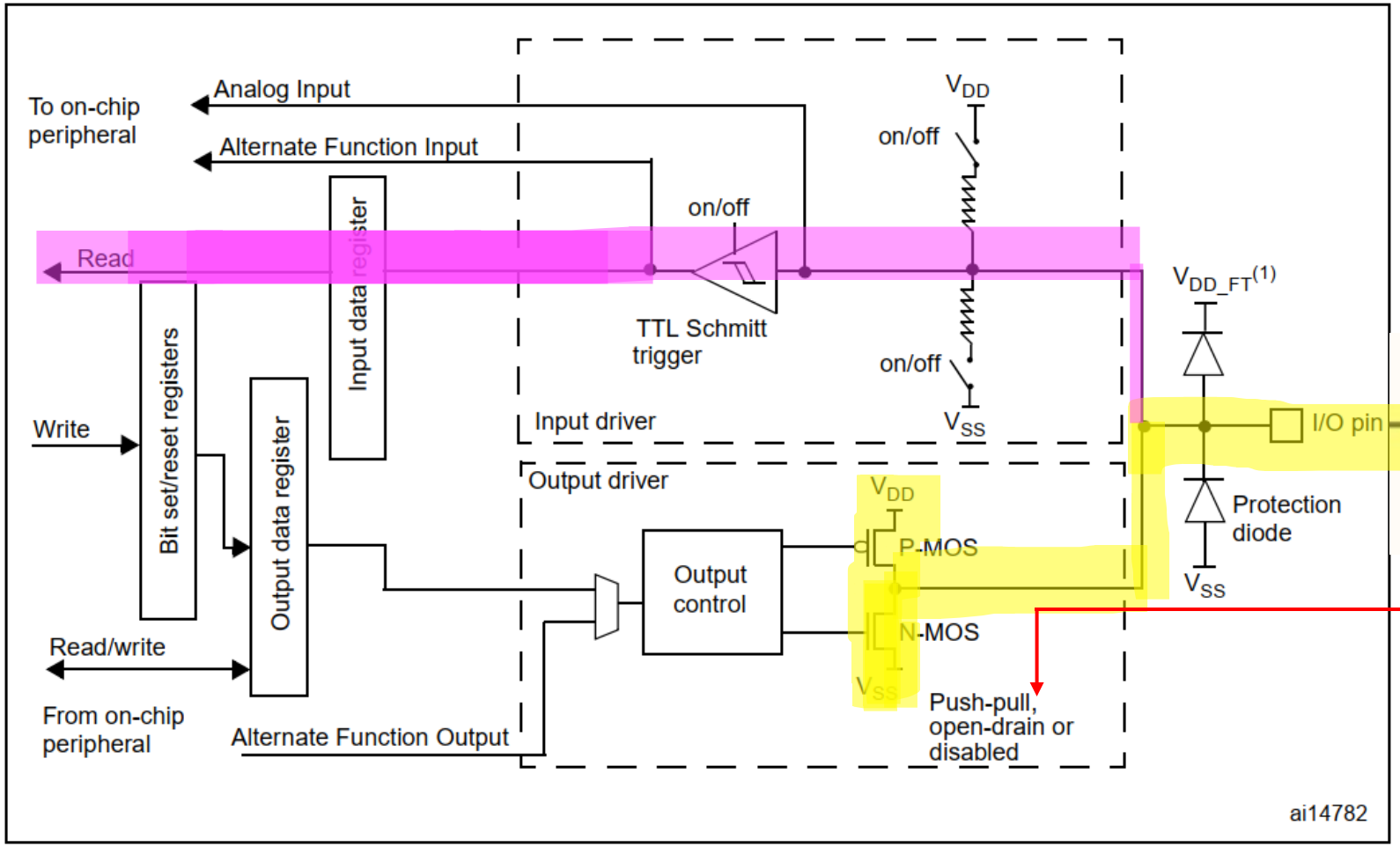
- Output open-drain

Current flow
press “ON”

GPIO

Let me explain!

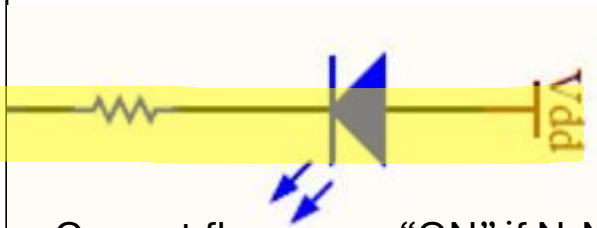
Figure 14. Basic structure of a 5-Volt tolerant I/O port bit



1. V_{DD_FT} is a potential specific to 5-Volt tolerant I/Os, and different from V_{DD}.

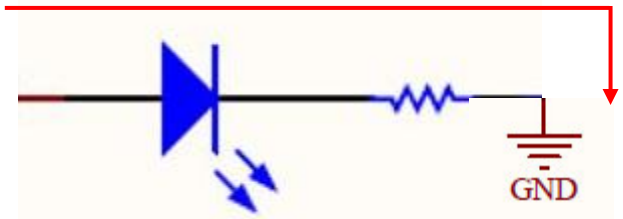
Output,

- Output push-pull



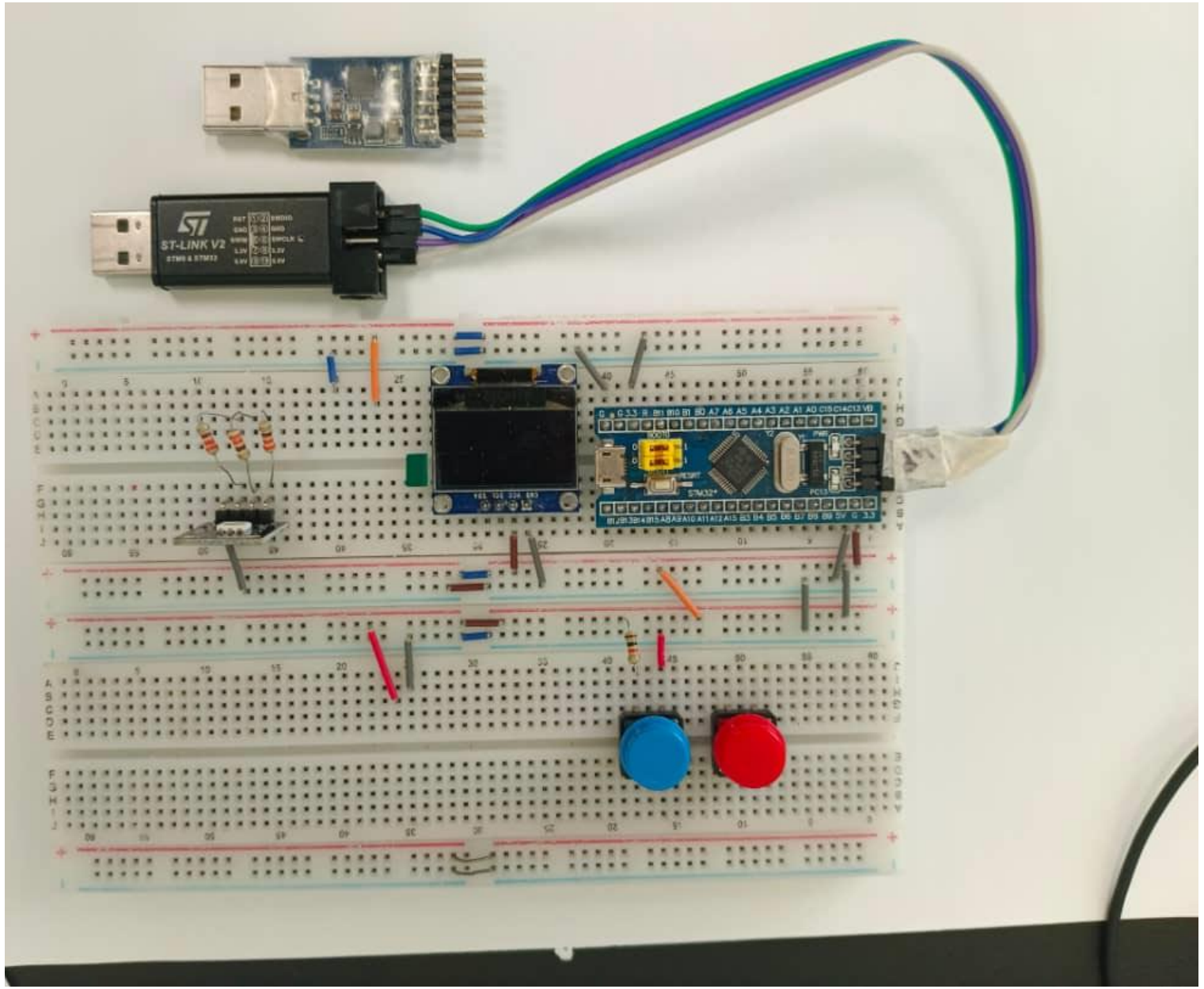
Current flow press “ON” if N-MOS “ON” and P-MOS “OFF”

Current flow press “ON” if P-MOS “ON” and N-MOS “OFF”



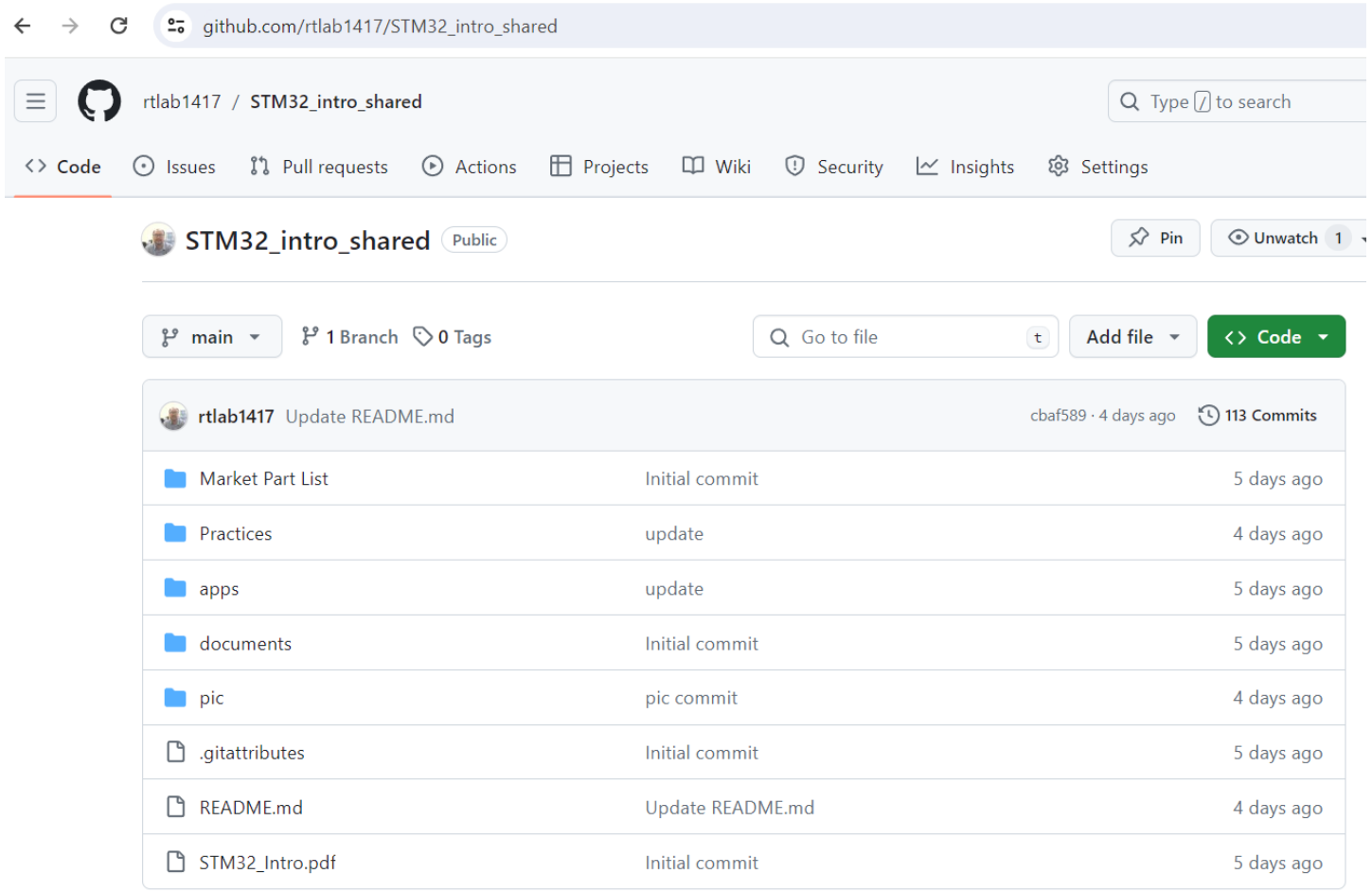
Can also wired this way!

The Training Kit



The Training Kit

Shared information – https://github.com/rtlab1417/STM32_intro_shared.git



Programming Software

1. **STM32CubeIDE - free - explored in this training.**
2. Keil MDK - paid service
3. IAR Embedded Workbench – paid service
4. Arduino IDE - free
5. PlatformIO - free
6. Matlab - Hardware support needed.
7. Etc.

1. Bare metal programming
   a. Call the registers directly and manually
2. Standard peripheral library
   a. Provided by ST
3. **Hardware Abstract Layer (HAL)**
   a. **ST provides HAL for its MCU family.**
   b. **This is implemented in this short course.**

Programming Software - Language

C Language

- STM32CubeIDE deploy C-language for HAL and coding.
- Having a basic understanding of the C Language can be very useful.
- Some C elements: Variables, Data type, Operators, Loops, Struct, Pointer, Function, type cast, etc.
- *Note:* ***C not equal to C++ however C could be implemented in C++***

Installing STM32CubeIDE

- Preparation – installed before attending the short course
- See module

- Remember where is your workspace.
- For example:
- D:\STM32CubeIDE\workspace\

First Project – looking silly yet very important

RTLab

The motivation

- To be confirmed that the MCU is communicating with ST-Link
- To be confirmed that STM32CubeIDE is working

First Project

Repeated procedures for all projects

1. Create New Project
2. Target Selection
3. *.ioc
   - Pinout & Configuration
   - Clock Configuration
   - Project Managers
   - Tools
4. Save and auto-generate project – template
5. Edit the code
6. RUN or DEBUG

First Project

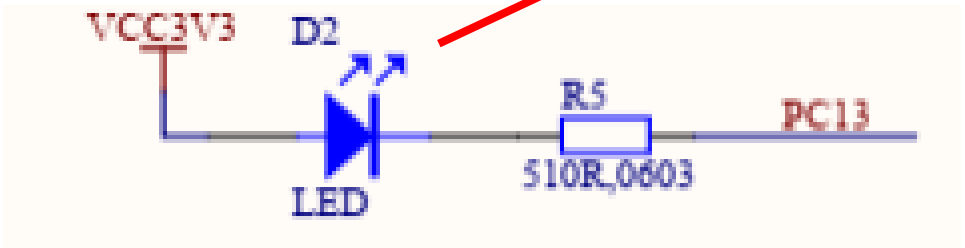
No external connection

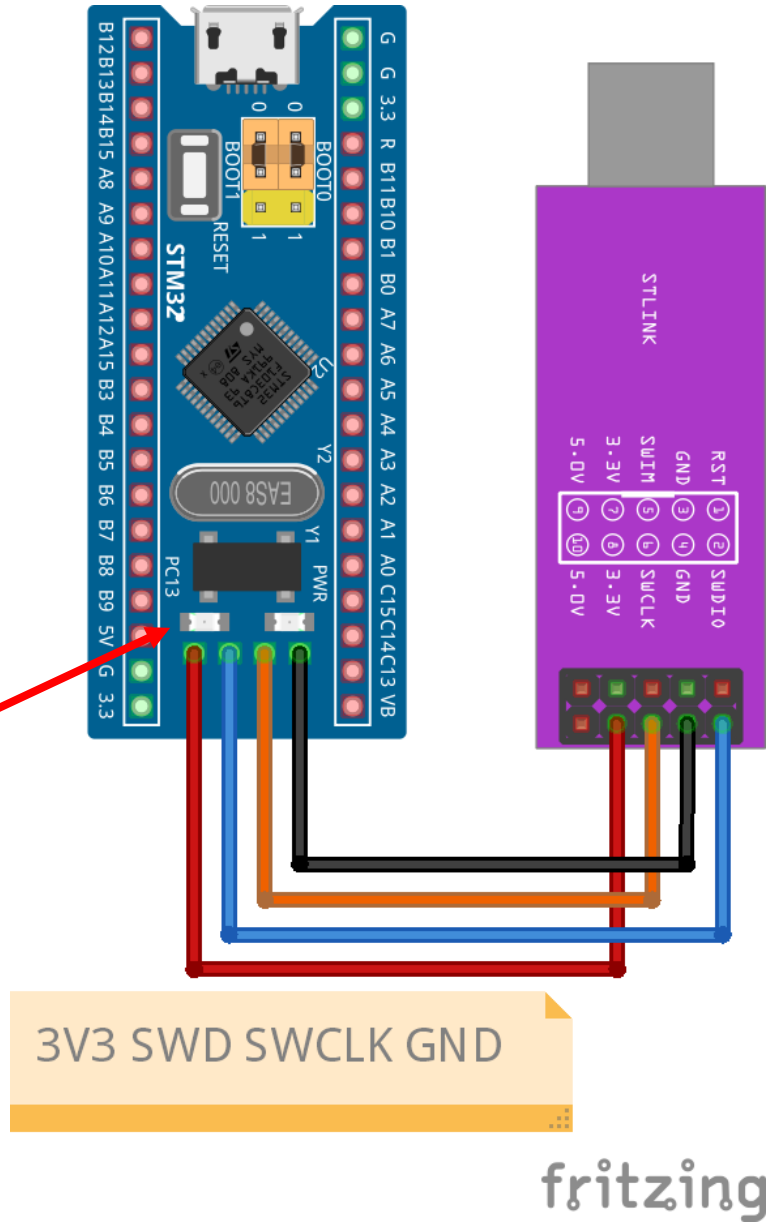
No coding needed

Just configure GPIO

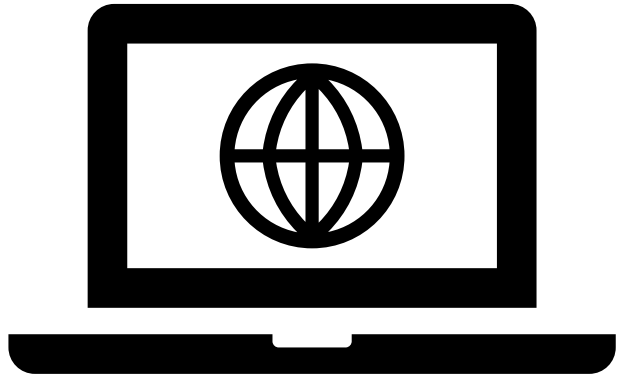
Lit up built-in LED – PC13

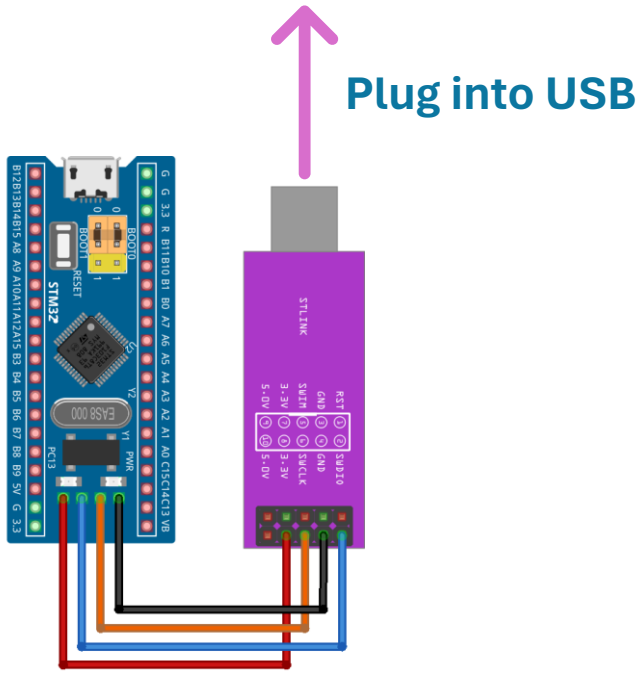
See module





First Project





Reminder:

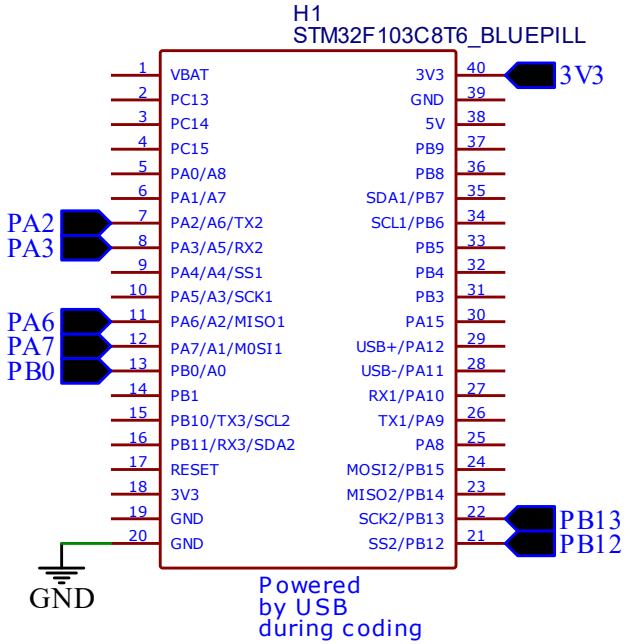
- **USB type-A**
  - **Detected and listed in Device Manager**
- **Keep online**
  - **Update ST-Link firmware**
  - **Update app.**

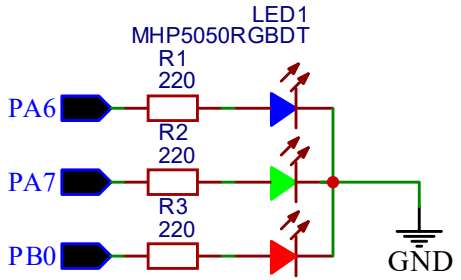
List of Practices

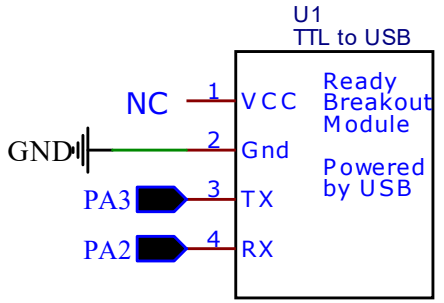
1. GPIO – LED
2. GPIO – LEDs Blink
3. GPIO – LED Buttons
4. GPIO – EXTI
5. UART in Polling Mode
6. UART With Interrupt
7. UART With DMA

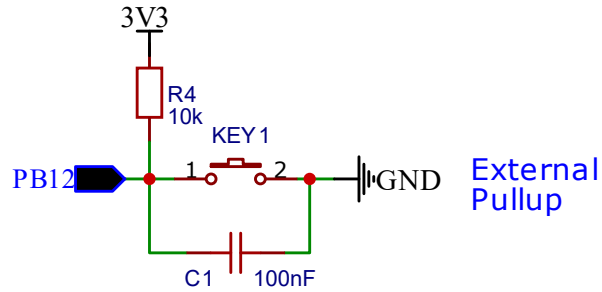
- More to go…

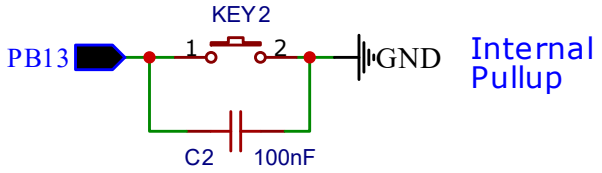
Schematic - Practice 1 to 7



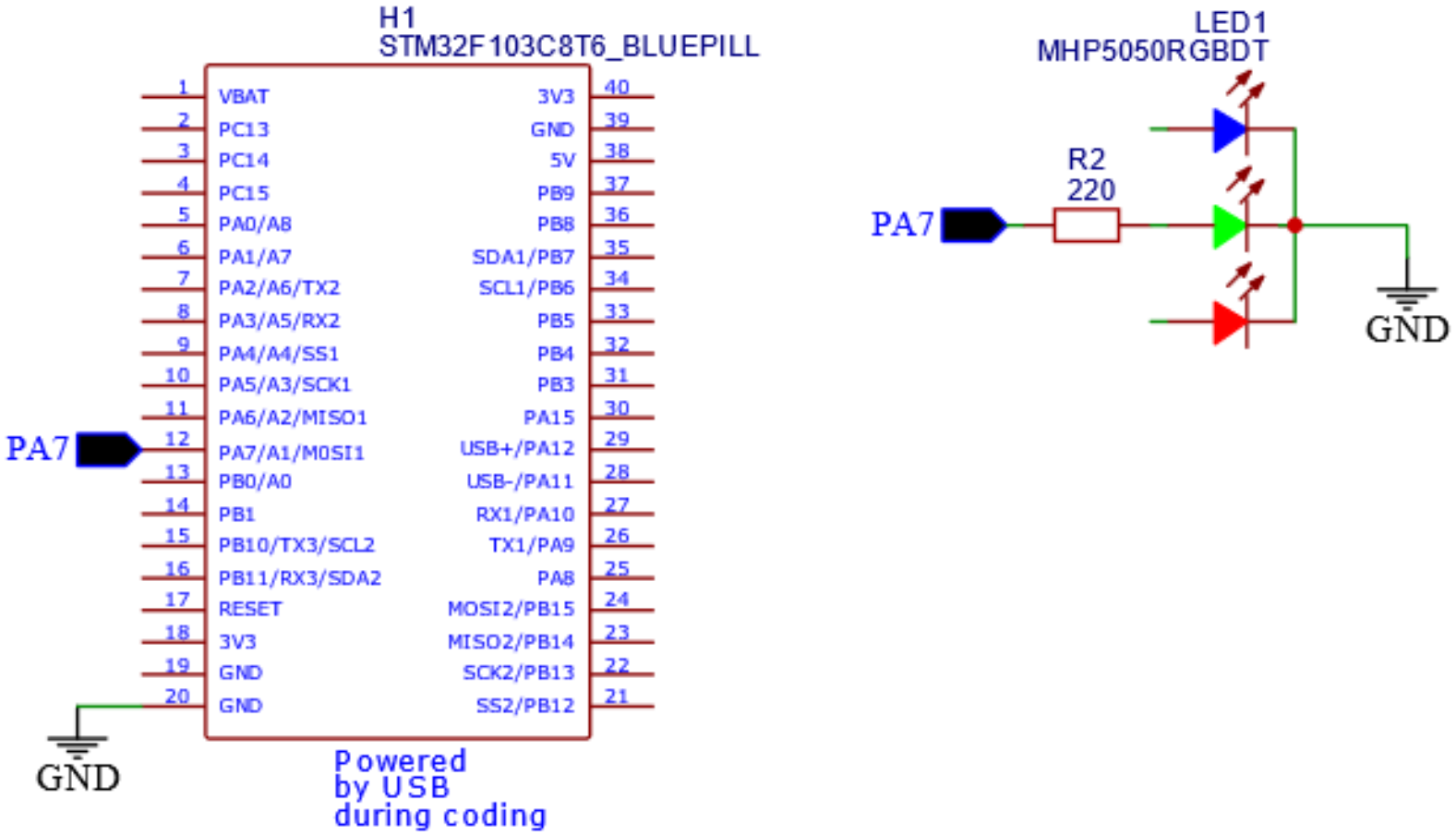








Practice 1 – GPIO -LED



Practice 1 – GPIO -LED

Pin	Mode	User Label
PA7	GPIO_Output, Low, Output Push Pull, No pull-up and no pull-down	LED_GREEN

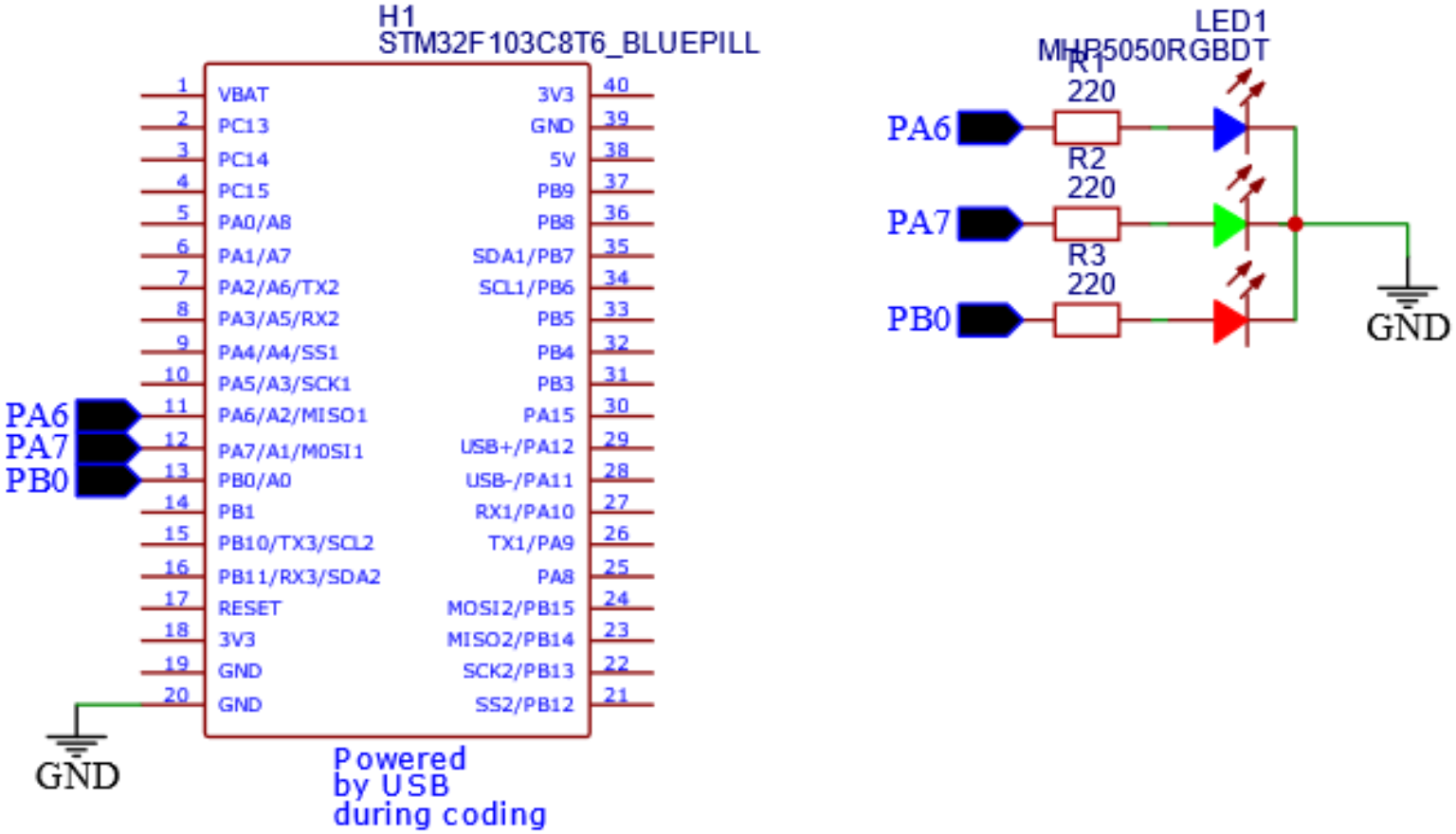
Practice 1 – GPIO -LED

Exploring GPIO

- See module

- Exploring the IDE and file structure
- main.c
- HAL and code accordingly – see module and demonstration

Practice 2 – GPIO –LEDs Blink



Practice 2 – GPIO –LEDs Blink

Pin	Mode	User Label
PA6	GPIO_Output, Low, Output Push Pull, No pull-up and no pull-down	LED_BLUE
PA7	GPIO_Output, Low, Output Push Pull, No pull-up and no pull-down	LED_GREEN
PB0	GPIO_Output, Low, Output Push Pull, No pull-up and no pull-down	LED_RED

Requirement:

	State0	State1	State2	State3	State4	State5
LED_BLUE	0	0	1	1	1	1
LED_GREEN	0	1	1	1	0	0
LED_RED	1	1	1	1	0	1

Practice 2 – GPIO –LEDs Blink

- See module
- Update main.c

- Have fun.
- Change new delay time.

- Challenge
- What if we have a new requirement? Design your own!

Practice 2 – GPIO –LEDs Blink