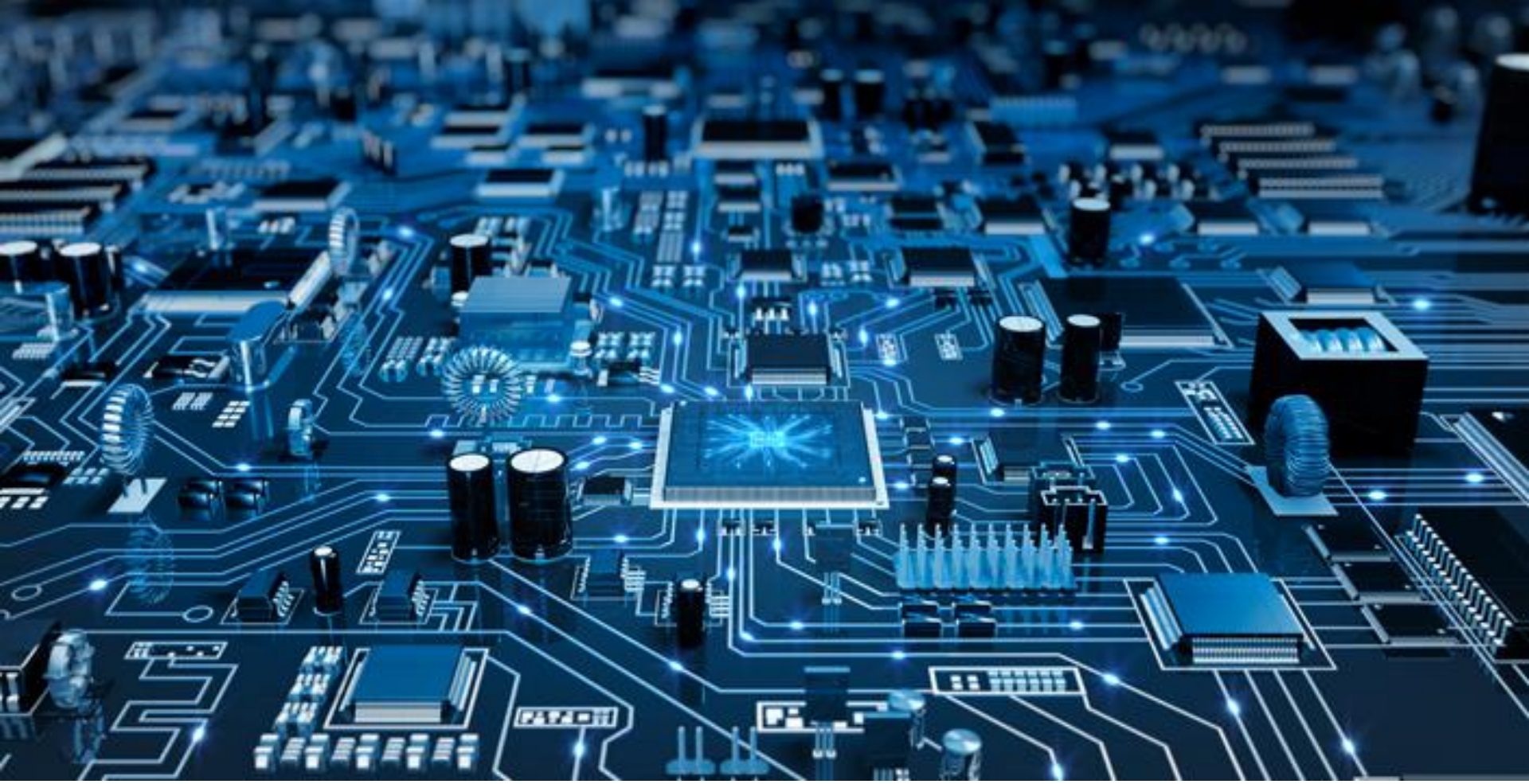


life.augmented

STM32 Microcontrollers Course

GPIO

Winter&Summer 2016

STM32F103RBT6

- General Purpose
- Performance Line
- 72 MHz maximum frequency,1.25 DMIPS/MHz
  - DMIPS/Mhz = 10^6 / (1757 * Number of Processor Clocks per Dhrystone loop)
  - MIPS = Million Instruction per second
- Single-cycle multiplication and hardware division
- 2.0 to 3.6 V application supply and I/Os
  - 5Volt Tolerant GPIOs
- 4-to-16 MHz crystal oscillator
- PLL for CPU clock
- Internal 8 MHz factory-trimmed RC
- Internal 40 kHz RC
- 7-channel DMA controller

STM32F103RBT6

Timers	General-purpose	4		
	Advanced-control	2		
	Basic	2		
Comm	SPI(I^2S)(3)	3(2)		
	I^2C	2		
	USART	5		
	USB	1		
	CAN	1		
	SDIO	1		
GPIOs		51	80	112
12-bit ADC Number of channels		3 16	3 16	3 21
12-bit DAC Number of channels		2 2		
CPU frequency		72 MHz		
Operating voltage		2.0 to 3.6 V		

Example: STM32 F 103 C 8 T 7 xxx

Device family

STM32 = ARM-based 32-bit microcontroller

Product type

F = general-purpose

Device subfamily

103 = performance line

Pin count

T = 36 pins
C = 48 pins
R = 64 pins
V = 100 pins

Flash memory size

8 = 64 Kbytes of Flash memory
B = 128 Kbytes of Flash memory

Package

H = BGA
I = UFBGA
T = LQFP
U = VFQFPN or UFQFPN

Temperature range

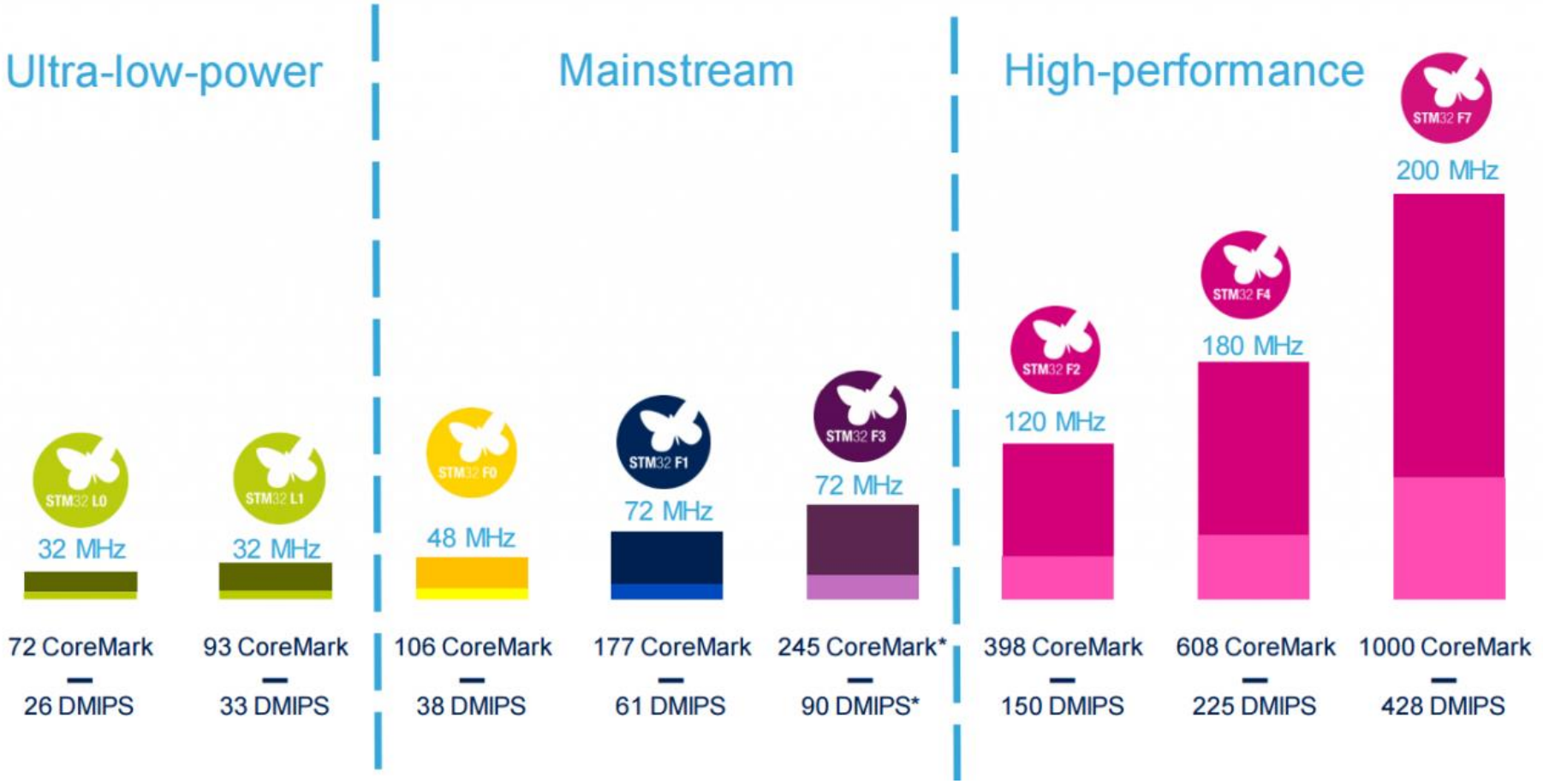
6 = Industrial temperature range, –40 to 85 °C.
7 = Industrial temperature range, –40 to 105 °C.

Options

xxx = programmed parts
TR = tape and real

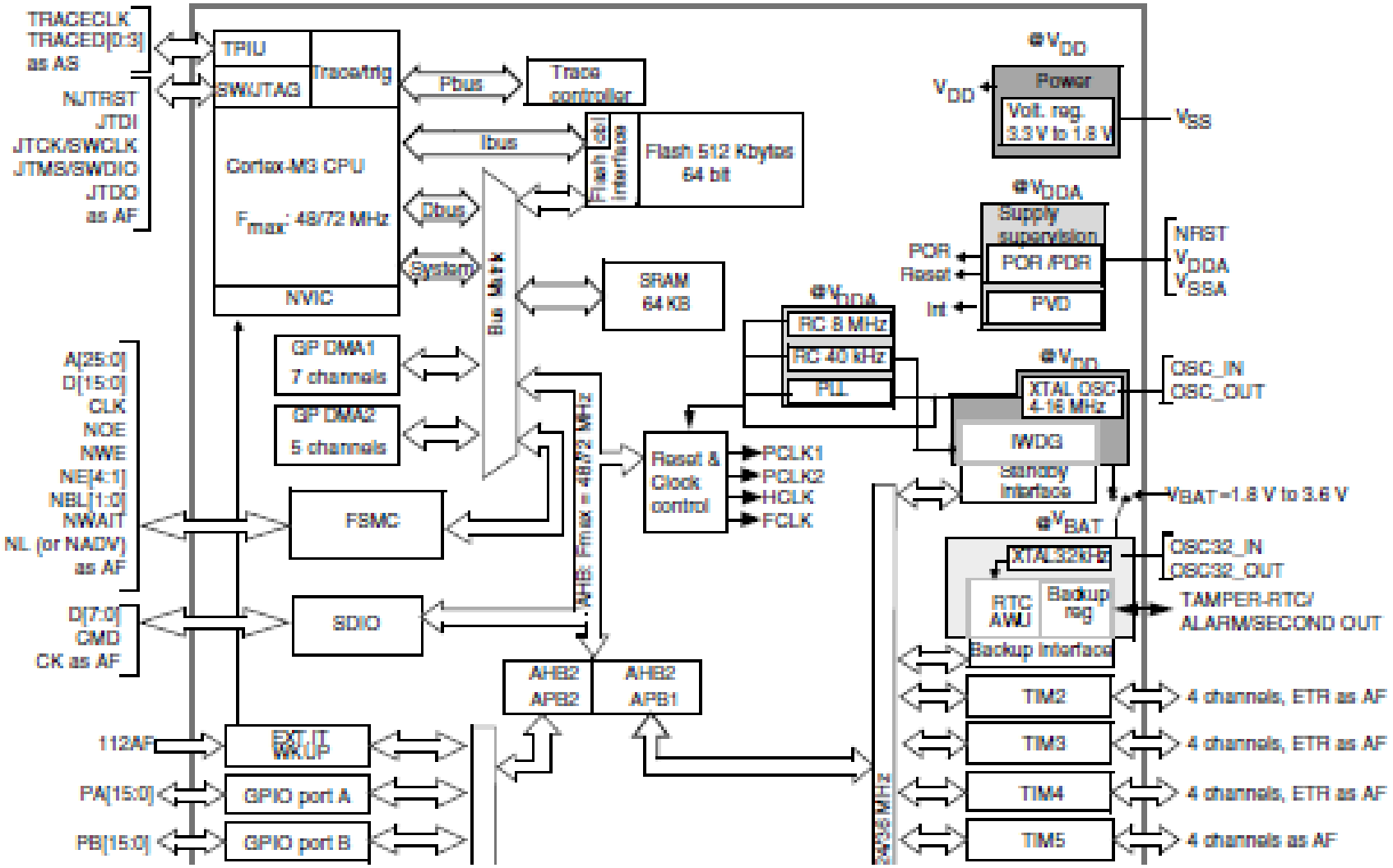
For a list of available options (speed, package, etc.) or for further information on any aspect of this device, please contact your nearest ST sales office.

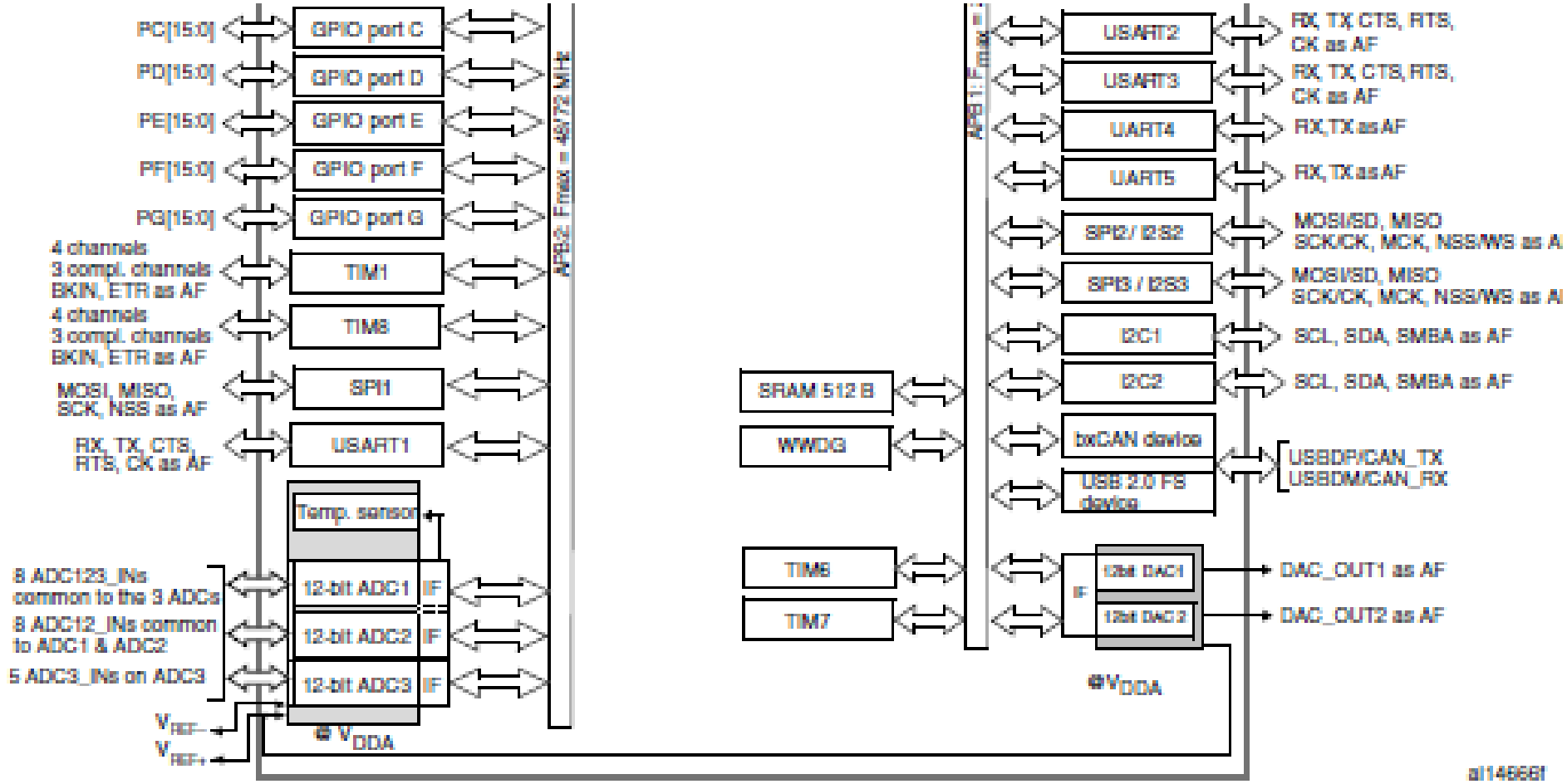
Device Overview



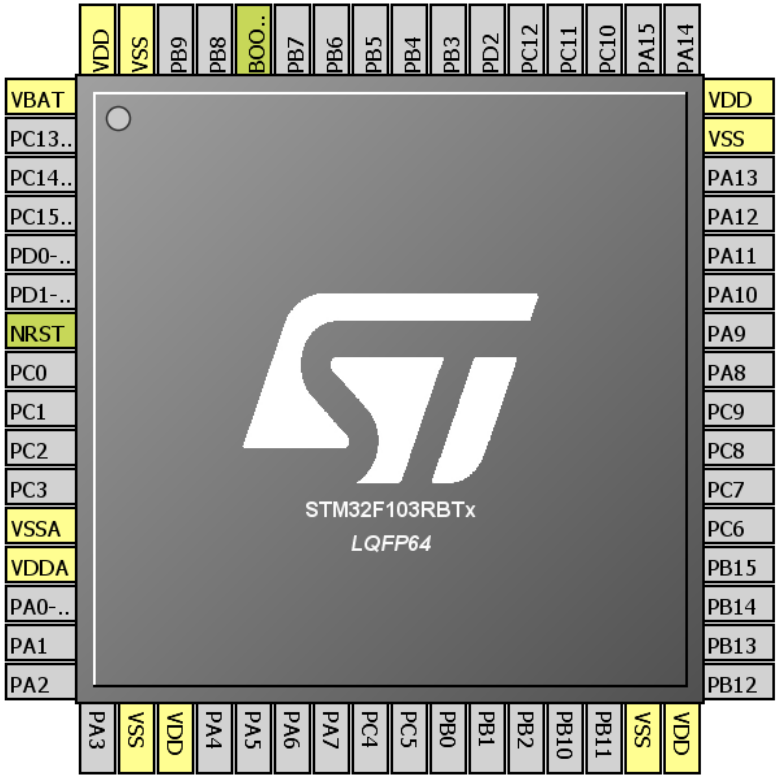
STM32 Official Shopping Centers

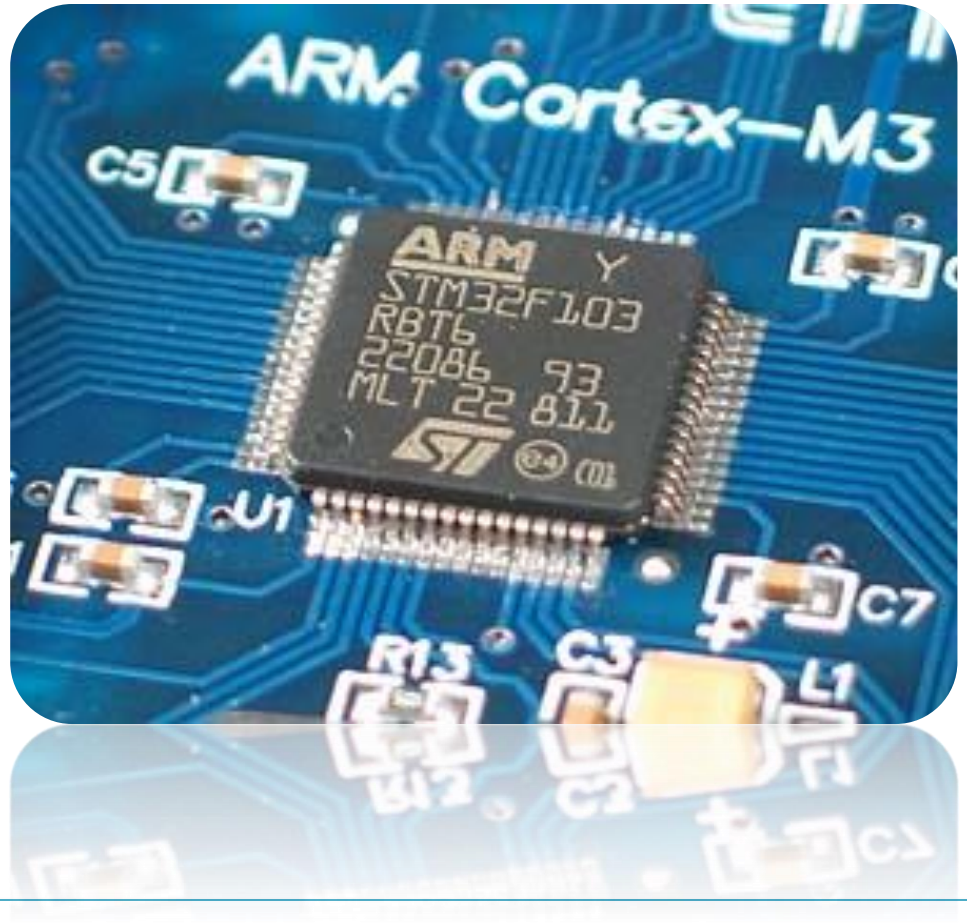
- For IC:
  - JavanElec.com
  - kavirelectronic.ir/eshop
- For Boards :
  - http://shop.aftabrayaneh.com
  - http://eshop.eca.ir


TRACECLK
TRACED[0:3]
as AS
NJTRST
JTDI
JTCK/SWCLK
JTMS/SWDIO
JTDO
as AF
TPIU
SWJ/JTAG
Trace/trig
Cortex-M3 CPU
Fmax: 48/72 MHz
NVIC
Pbus
Trace controller
Ibus
Dbus
System
Flash interface
Flash 512 Kbytes 64 bit
SRAM 64 KB
Bus Matrix
GP DMA1 7 channels
GP DMA2 5 channels
FSMC
SDIO
A[25:0]
D[15:0]
CLK
NOE
NWE
NE[4:1]
NBL[1:0]
NWAIT
NL (or NADV)
as AF
D[7:0]
CMD
CK as AF
AHB: Fmax = 48/72 MHz
Reset & Clock control
PCLK1
PCLK2
HCLK
FCLK
AHB2 APB2
AHB2 APB1
EXT.IT WKUP
112AF
GPIO port A
PA[15:0]
GPIO port B
PB[15:0]
@VDD
VDD
Power
Volt. reg. 3.3 V to 1.8 V
VSS
@VDDA
Supply supervision
POR/PDR
PVD
POR
Reset
Int
NRST
VDDA
VSSA
@VDDA
RC 8 MHz
RC 40 kHz
PLL
@VDD
XTAL OSC 4-16 MHz
OSC_IN
OSC_OUT
IWDG
Standby Interface
VBAT = 1.8 V to 3.6 V
@VBAT
XTAL32kHz
OSC32_IN
OSC32_OUT
RTC AWU
Backup reg
TAMPER-RTC/ ALARM/SECOND OUT
Backup Interface
TIM2
TIM3
TIM4
TIM5
4 channels, ETR as AF
4 channels, ETR as AF
4 channels, ETR as AF
4 channels as AF
24/36 MHz


PC[15:0]
GPIO port C
PD[15:0]
GPIO port D
PE[15:0]
GPIO port E
PF[15:0]
GPIO port F
PG[15:0]
GPIO port G
APB2: Fmax = 48/72 MHz
4 channels
3 compl. channels
BKIN, ETR as AF
TIM1
4 channels
3 compl. channels
BKIN, ETR as AF
TIM8
MOSI, MISO,
SCK, NSS as AF
SPI1
RX, TX, CTS,
RTS, CK as AF
USART1
Temp. sensor
8 ADC123_INs
common to the 3 ADCs
8 ADC12_INs common
to ADC1 & ADC2
5 ADC3_INs on ADC3
12-bit ADC1
12-bit ADC2
12-bit ADC3
IF
VREF-
VREF+
VDDA
SRAM 512 B
WWDG
TIM6
TIM7
APB1: Fmax =
USART2
USART3
UART4
UART5
SPI2 / I2S2
SPI3 / I2S3
I2C1
I2C2
bxCAN device
USB 2.0 FS
device
12bit DAC1
12bit DAC2
RX, TX, CTS, RTS,
CK as AF
RX, TX, CTS, RTS,
CK as AF
RX,TX as AF
RX, TX as AF
MOSI/SD, MISO
SCK/CK, MCK, NSS/WS as A
MOSI/SD, MISO
SCK/CK, MCK, NSS/WS as AI
SCL, SDA, SMBA as AF
SCL, SDA, SMBA as AF
USBDP/CAN_TX
USBDM/CAN_RX
DAC_OUT1 as AF
DAC_OUT2 as AF
VDDA
ai14868f

General Purpose I/O



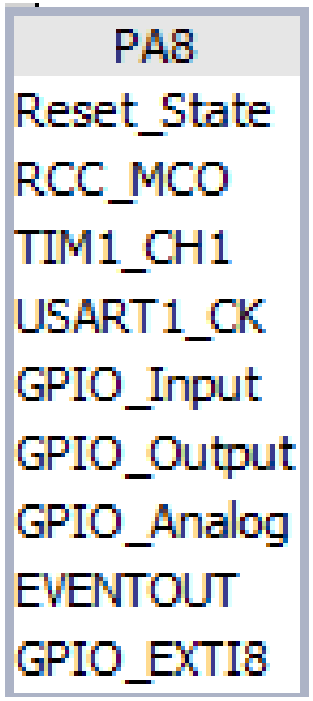


General Purpose I/O

- GPIO Modes :
  - Reset State (Input Float)
  - GPIO Output
    - Push-Pull
    - Open-Drain (AND Functionality)
    - When we have to use Push/Pull or Open Drain
  - GPIO Input
    - Float
    - Pull-Up
    - Pull-Down
  - Alternative
    - Push-Pull
    - Open Drain

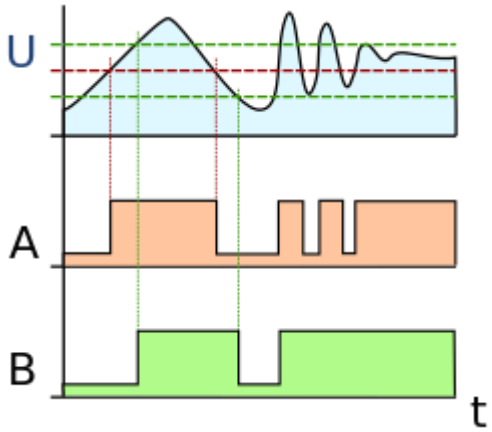
General Purpose I/O

- GPIO Modes :
  - GPIO Analog
    - ADC vs Analog
      - Analog : Put GPIO in low power mode
        - » By turning off pull-up/down resistor and Schmitt triggers
      - ADC : Analog + Conversion capability
  - Alternative Function
    - SPI, I2C, Timer I/O, UART
  - EventOut
    - SEV Assembly



Schmitt Triggers

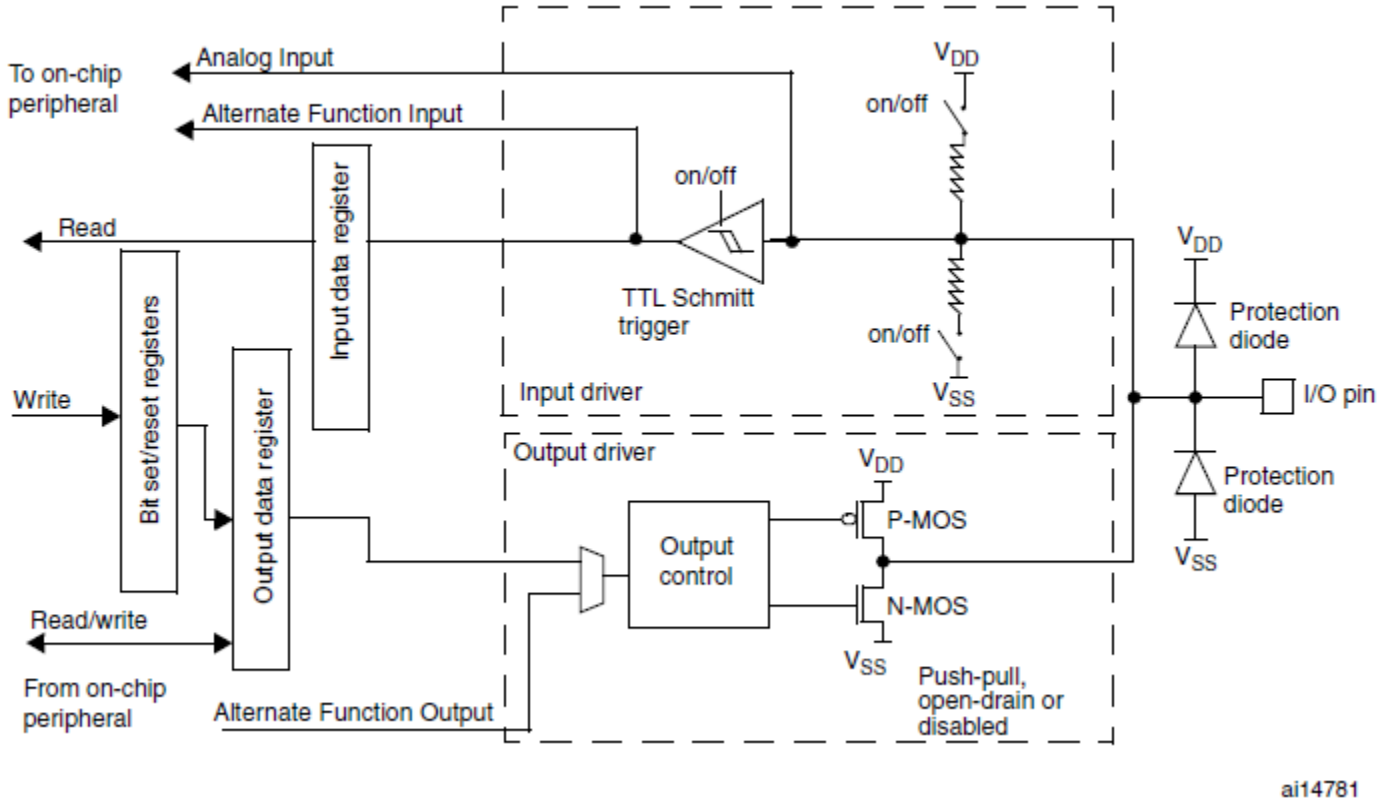
- a electronic circuit in which the output increases to a steady maximum when the input rises above a certain threshold, and decreases almost to zero when the input voltage falls below another threshold. Schmitt triggers tend to remove noise especially mechanical bounces on input pins.

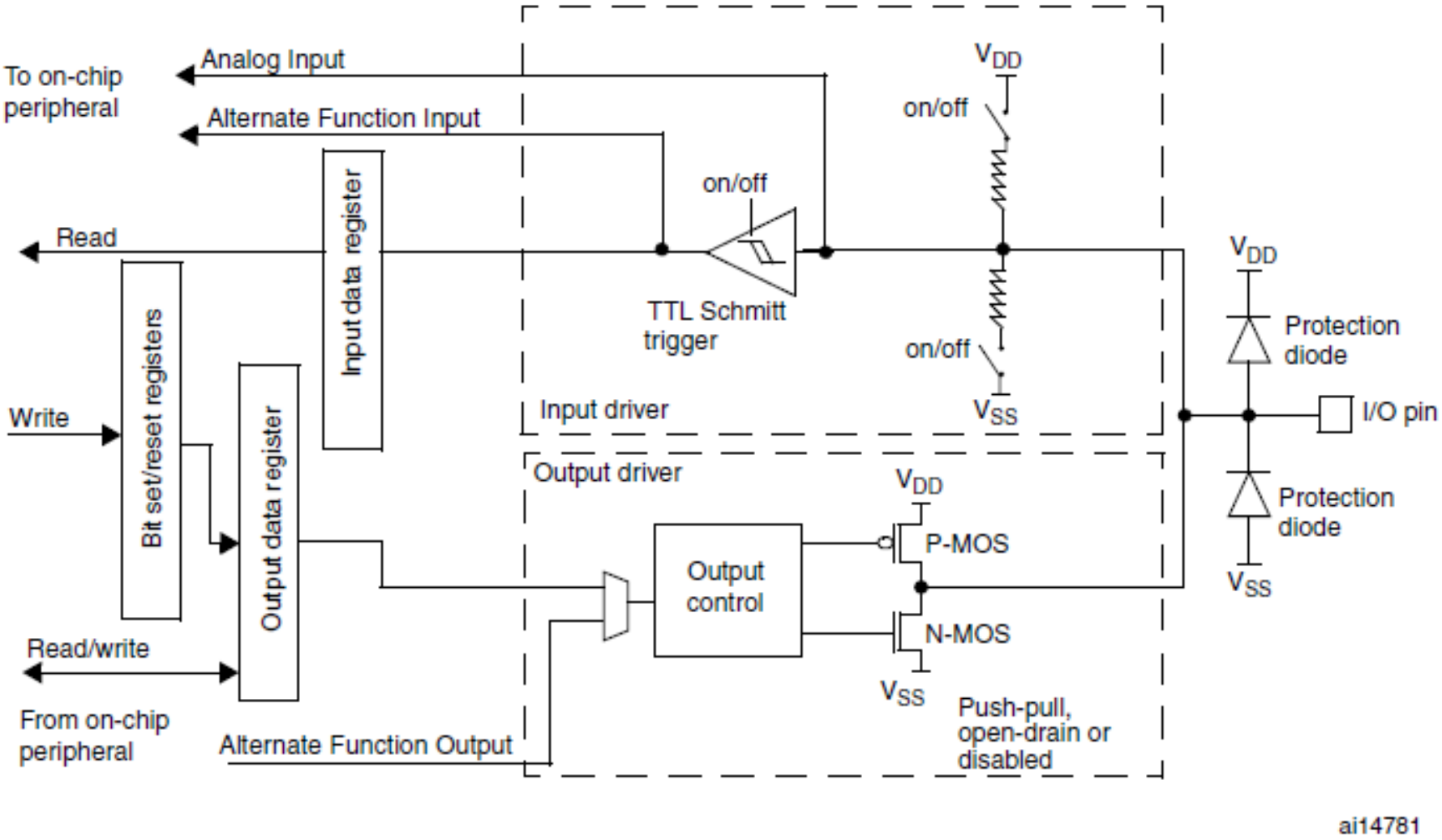


U : Input Signal
A : Comparator
B : Schmitt Trigger

I/O Different Configurations

- Input Configuration
- Output Configuration

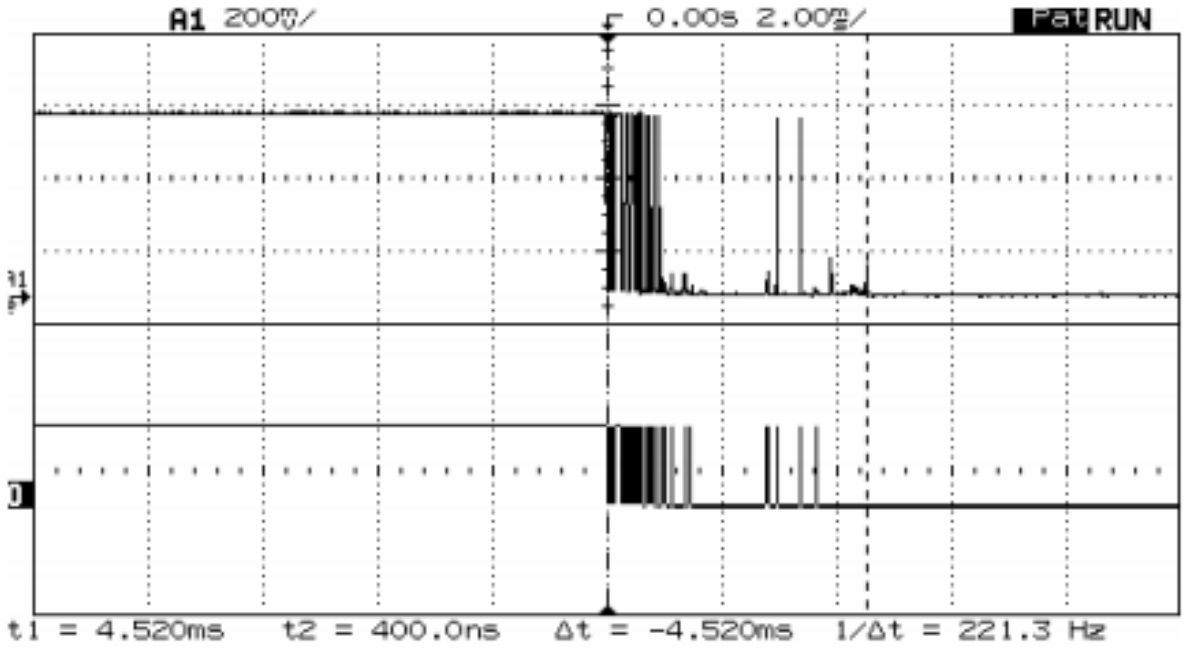



To on-chip peripheral
Analog Input
Alternate Function Input
Read
Write
Read/write
From on-chip peripheral
Alternate Function Output
Bit set/reset registers
Input data register
Output data register
on/off
TTL Schmitt trigger
Input driver
V_{DD}
on/off
on/off
V_{SS}
Output driver
V_{DD}
P-MOS
Output control
N-MOS
V_{SS}
Push-pull, open-drain or disabled
V_{DD}
Protection diode
I/O pin
Protection diode
V_{SS}
ai14781

General Purpose I/O

- GPIO Speed
  - Low Speed 2Mhz
  - Medium Speed 10Mhz
  - High Speed 50Mhz
- Software I/O Remapping

GPIO Input Debouncing



Bouncing showcase

GPIO Input Debouncing Using RC

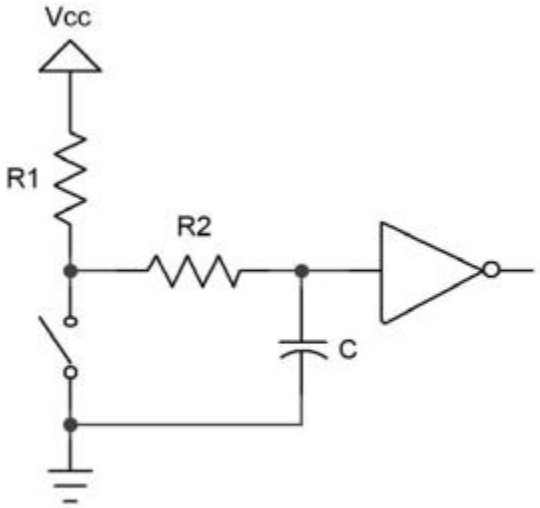


Figure 2: An RC debouncer

GPIO Input Debouncing Using Software Solutions

```
while (1)
{
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
    if((HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_11) == 1))
    {
        //Do something
    }
}
/* USER CODE END 3 */
```

```
/* USER CODE BEGIN 3 */
    if((HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_11) == 1))
    {
        HAL_Delay(100);
        if((HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_11) == 1))
        {
            //Do something
        }
    }
}
/* USER CODE END 3 */
```

```
/* USER CODE BEGIN 3 */
    if((HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_11) == 1))
    {
        HAL_Delay(100);
        if((HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_11) == 1))
        {
            //Do something

            while(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_11) == 1){}
        }
    }
} /* while( 1 ) */
/* USER CODE END 3 */
```

More Advanced Methods?

STM32 HAL GPIO Functions

- GPIO Read Pin – Reading a **input** or **output** pin state
  - GPIO_PinState **HAL_GPIO_ReadPin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)**
- GPIO Write Pin – Writing a value on **output** Pin
  - void **HAL_GPIO_WritePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState)**
- GPIO Toggle Pin – Toggle an **output** Pin
  - void **HAL_GPIO_WritePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState)**
- GPIO Lock Pin – Lock a pin until next reset
  - HAL_StatusTypeDef **HAL_GPIO_LockPin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)**