

# General Purpose Input/Output (GPIO)

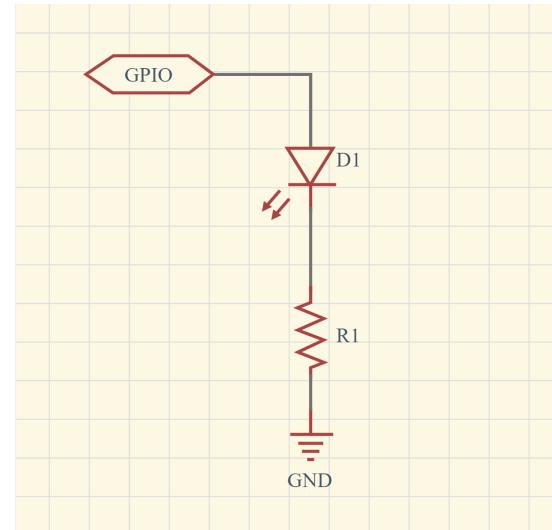
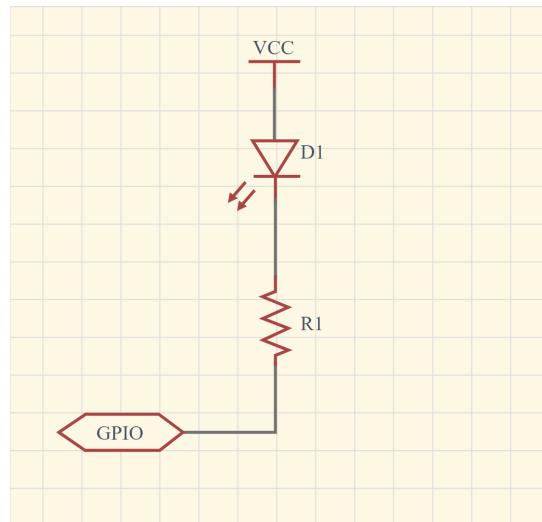
Microcontroller Application and Development

Sorayut Glomglome

$\pi$

# Digital Output

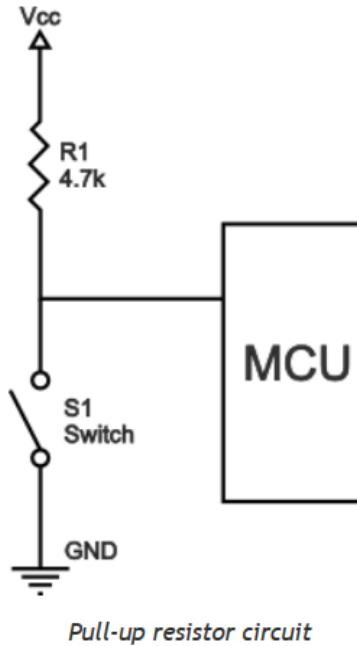
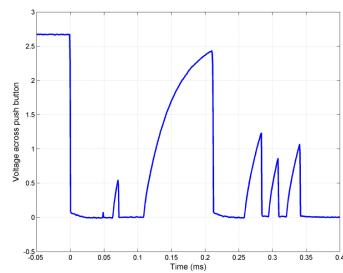
- LED
- Writing digital signal to external IC



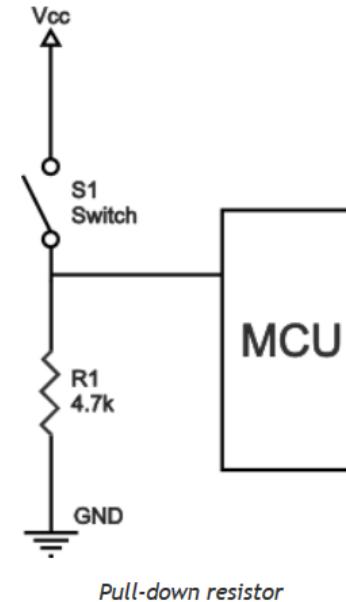
<https://blog.stratifylabs.co/device/2013-10-24-Using-LEDs-in-Embedded-Designs/>

# Digital Input

- Push Button
- Switch
- Reading signal from external IC



*Pull-up resistor circuit*



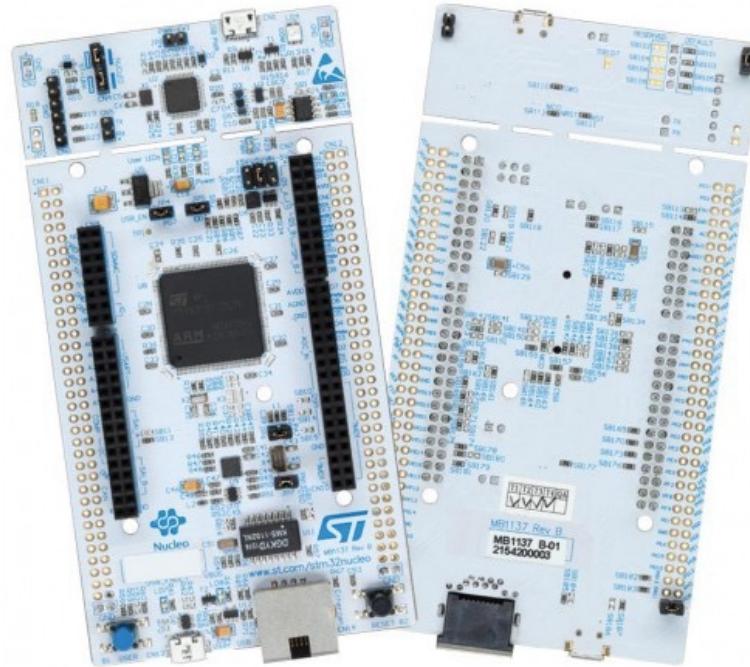
*Pull-down resistor*

[https://www.itbakery.net/2018/03/21/pull-up-pull-down-](https://www.itbakery.net/2018/03/21/pull-up-pull-down/)  
%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%AD%E0%B9%84%E0%B8%A3/

Bounce -> DeBounce

# NUCLEO-F767ZI

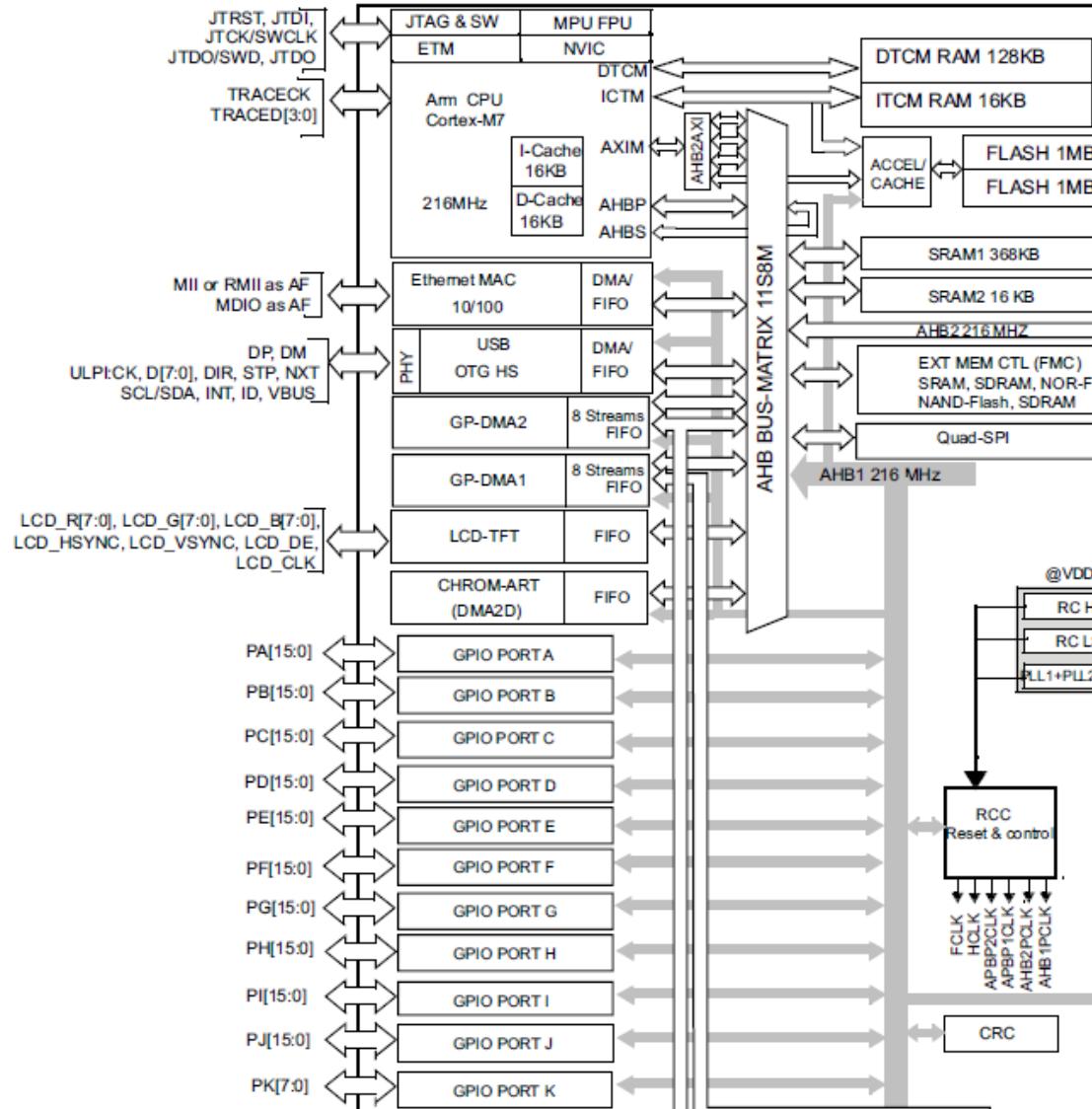
- บอร์ดสำหรับการทดลอง NUCLEO-F767ZI
- 3 user LED shared with Arduino™
- 1 user and 1 reset push-buttons
- Board expansion connectors:
  - Arduino™ Uno V3
  - ST morpho extension pin headers
- Flexible power-supply options
  - ST-LINK USB VBUS or external sources
  - On-board ST-LINK/V2-1 debugger/programmer with USB reenumeration capability. Three different interfaces supported on USB: mass storage, virtual COM port and debug port



<https://www.st.com/en/evaluation-tools/nucleo-f767zi.html>

<https://os.mbed.com/platforms/ST-Nucleo-F767ZI/>

$\pi$

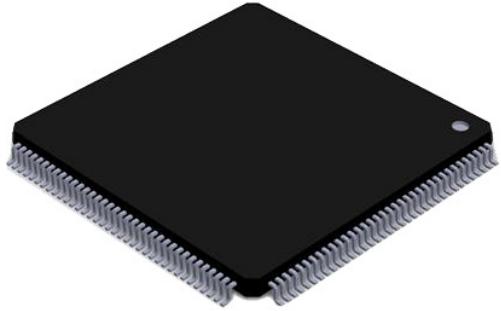


## STM32F767ZI & GPIO

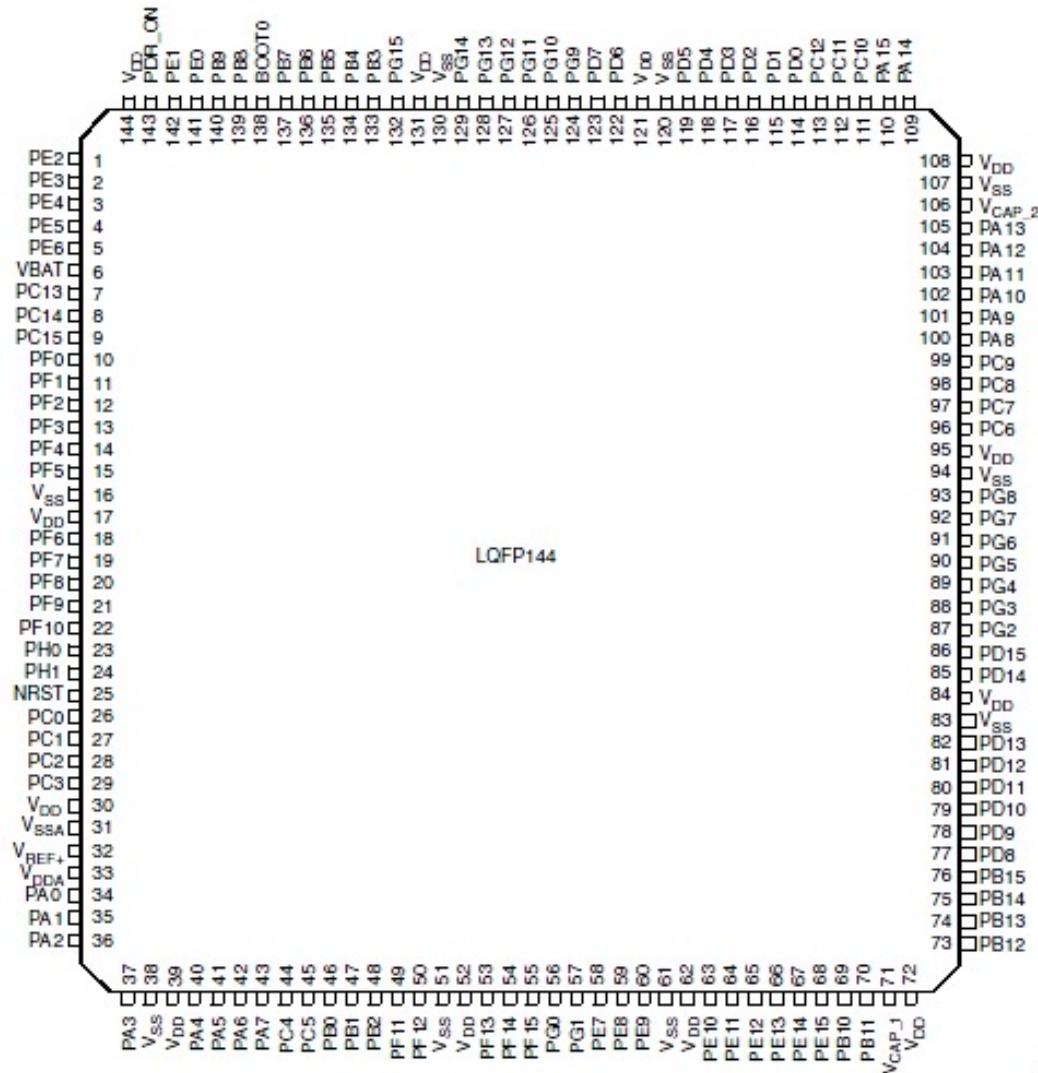
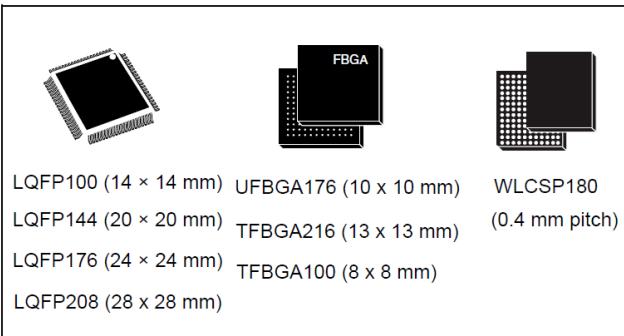
- › GPIO Port A B C D E F H I J K
- › Each port is 16 pins
- › PA[0] – PA[15]
- › PB[0] – PB[15]
- › ...
- › PJ[0] – PJ[15]
- › PK[0] – PK[7]

$\pi$

# LQFP144



LQFP: Low Profile Quad Flat Package



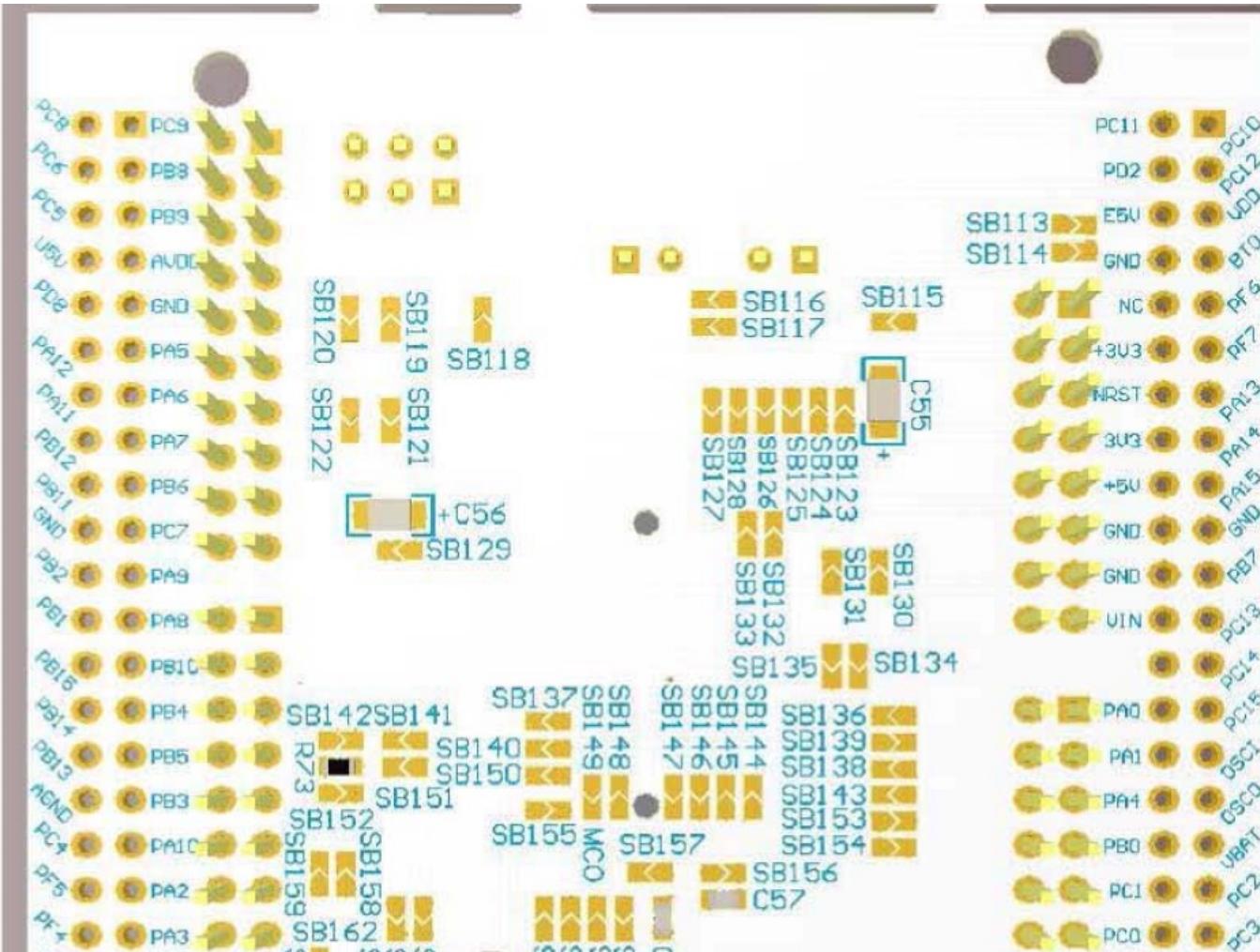
ai18496b

π

# Board Pin Out

		CN8				CN7	
NC	NC	1 2	D43	PC8	PC6	D16	D15 PB8
IOREF	IOREF	3 4	D44	PC9	PB15	D17	3 4 D14 PB9
RESET	RESET	5 6	D45	PC10	PB13	D18	5 6 AVDD AVDD
+3V3	+3V3	7 8	D46	PC11	PB12	D19	7 8 GND GND
+5V	+5V	9 10	D47	PC12	PA15	D20	9 10 D13 PA5
GND	GND	11 12	D48	PD2	PC7	D21	11 12 D12 PA6
GND	GND	13 14	D49	PG2	PB5	D22	13 14 D11 PA7
VIN	VIN	15 16	D50	PG3	PB3	D23	15 16 D10 PD14
PA3	A0	1 2	D51	PD7	PA4	D24	17 18 D9 PD15
PC0	A1	3 4	D52	PD6	PB4	D25	19 20 D8 PF12
PC3	A2	5 6	D53	PD5	AVDD	AVDD	1 2 D7 PF13
PF3	A3	7 8	D54	PD4	AGND	AGND	3 4 D6 PE9
PF5	A4	9 10	D55	PD3	GND	GND	5 6 D5 PE11
PF10	A5	11 12	GND	GND	PB1	A6	7 8 D4 PF14
NC	D72	13 14	D56	PE2	PC2	A7	9 10 D3 PE13
PA7	D71	15 16	D57	PE4	PF4	A8	11 12 D2 PF15
PF2	D70	17 18	D58	PE5	PB6	D26	13 14 D1 PG14
PF1	D69	19 20	D59	PE6	PB2	D27	15 16 D0 PG9
PF0	D68	21 22	D60	PE3	GND	GND	17 18 D42 PE8
GND	GND	23 24	D61	PF8	PD13	D28	19 20 D41 PE7
PD0	D67	25 26	D62	PF7	PD12	D29	21 22 GND GND
PD1	D66	27 28	D63	PF9	PD11	D30	23 24 D40 PE10
PG0	D65	29 30	D64	PG1	PE2	D31	25 26 D39 PE12
		CN9				CN10	

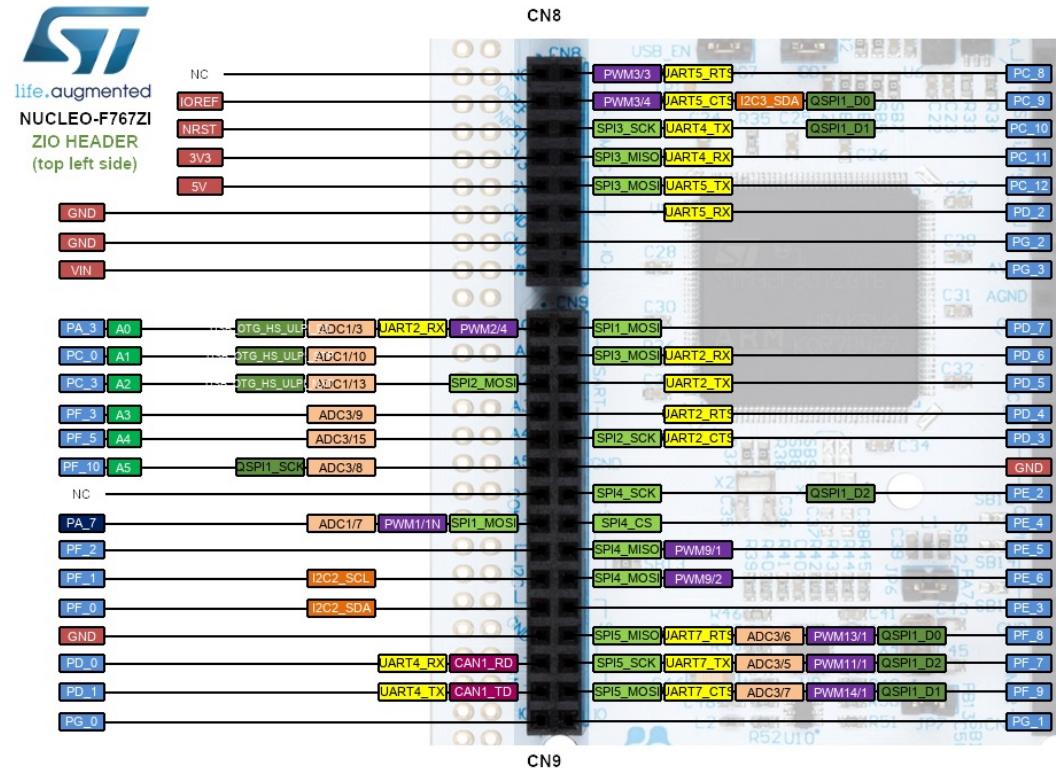
$\pi$



Bottom  
Layout

$\pi$

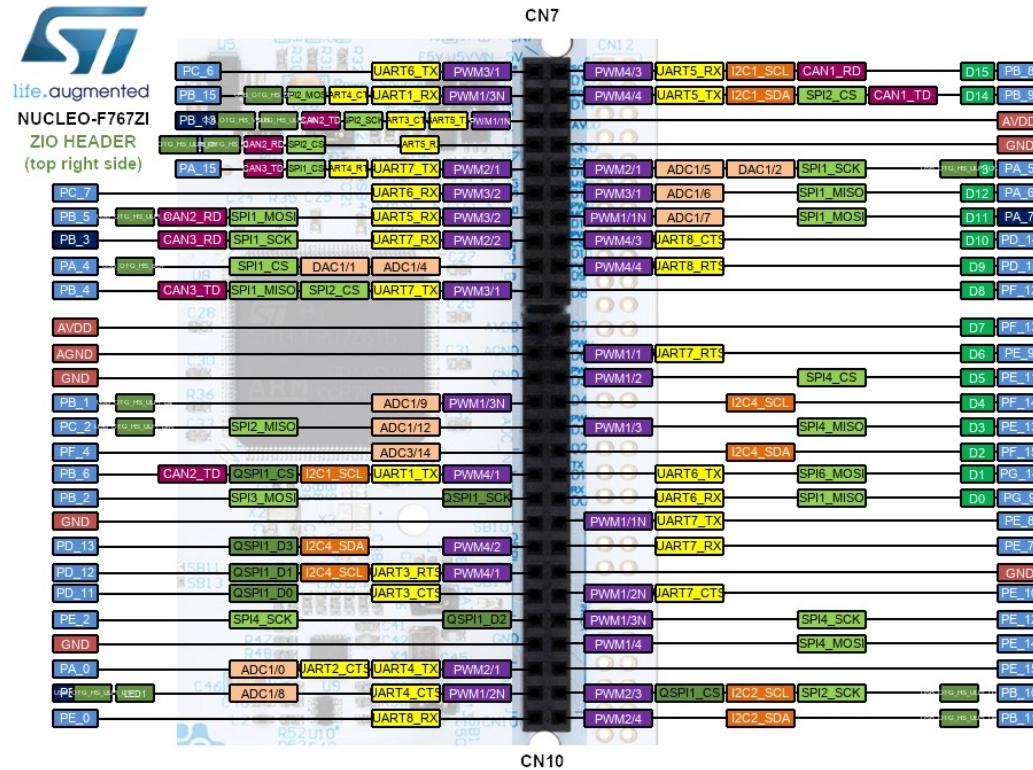
# Pinout



<https://os.mbed.com/platforms/ST-Nucleo-F767ZI/>

$\pi$

# Pinout



<https://os.mbed.com/platforms/ST-Nucleo-F767ZI/>

$\pi$

# Pin Mapping Table

Pin Number												Pin name (function after reset)	Pin type	I/O structure	Notes	Alternate functions	Additional functions						
STM32F765xx STM32F767xx						STM32F768Ax STM32F769xx																	
TFBGA100	LQFP100	LQFP144	UFBGA176	LQFP176	LQFP208	TFBGA216	WLCSP180 <sup>(1)</sup>	LQFP176	LQFP208	TFBGA216													
-	-	-	F3	15	15	F4	F12	15	15	F4	VDD	S	-	-	-	-	-						
-	-	10	E2	16	16	D2	G11	16	16	D2	PF0	I/O	FT	-	I2C2_SDA, FMC_A0, EVENTOUT	-							
-	-	11	H3	17	17	E2	G12	17	17	E2	PF1	I/O	FT	-	I2C2_SCL, FMC_A1, EVENTOUT	-							
-	-	12	H2	18	18	G2	G13	18	18	G2	PF2	I/O	FT	-	I2C2_SMBA, FMC_A2, EVENTOUT	-							
-	-	-	-	-	19	E3	NC	-	19	E3	PI12	I/O	FT	-	LCD_HSYNC, EVENTOUT	-							
-	-	-	-	-	20	G3	NC	-	20	G3	PI13	I/O	FT	-	LCD_VSYNC, EVENTOUT	-							
-	-	-	-	-	21	H3	NC	-	21	H3	PI14	I/O	FT	-	LCD_CLK, EVENTOUT	-							
-	-	13	J2	19	22	H2	H11	19	22	H2	PF3	I/O	FT	-	FMC_A3, EVENTOUT	ADC3_IN9							
-	-	14	J3	20	23	J2	H12	20	23	J2	PF4	I/O	FT	-	FMC_A4, EVENTOUT	ADC3_IN14							
-	-	15	K3	21	24	K3	H13	21	24	K3	PF5	I/O	FT	-	FMC_A5, EVENTOUT	ADC3_IN15							

# General Purpose Input Output

- Basic interfacing between MCU and real word
- Digital signal

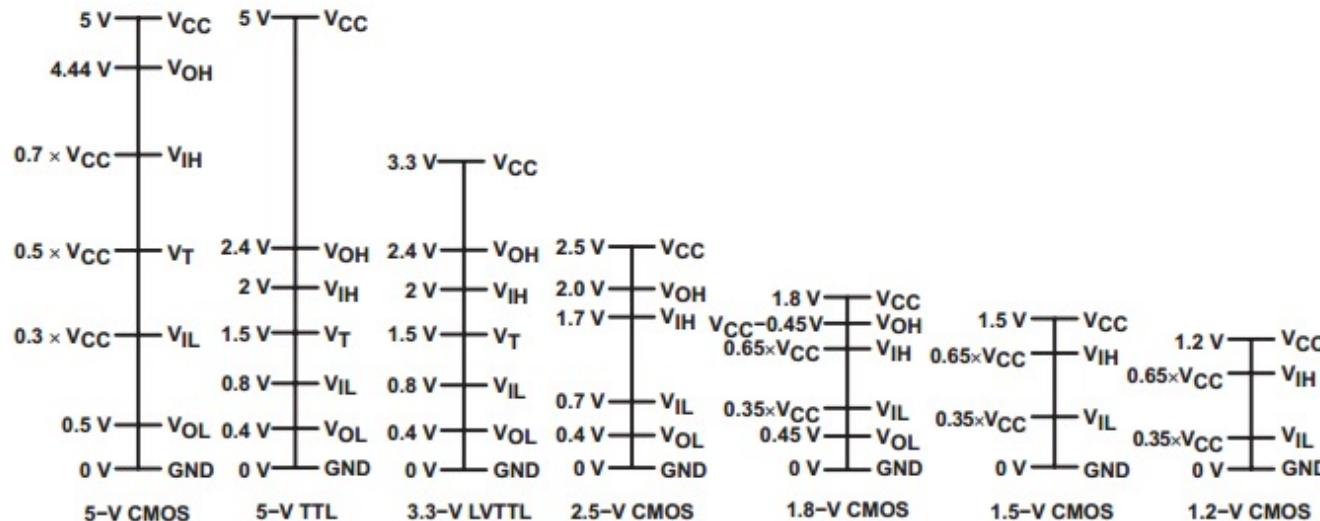


Figure 2. Digital Switching Levels

$\pi$

# Absolute Maximum Rating

Table 11. Voltage characteristics

Symbol	Ratings	Min	Max	Unit
$V_{DD-V_{SS}}$	External main supply voltage (including $V_{DDA}$ , $V_{DD}$ and $V_{BAT}$ ) <sup>(1)</sup>	-0.3	4.0	V
$V_{IN}$	Input voltage on FT pins <sup>(2)</sup>	$V_{SS}-0.3$	$V_{DD}+4.0$	V
	Input voltage on any other pin	$V_{SS}-0.3$	4.0	
	Input voltage for BOOT0	$V_{SS}$	9.0	
$ \Delta V_{DDx} $	Variations between different $V_{DD}$ power pins	-	50	mV
$ V_{SSx}-V_{SSL} $	Variations between all the different ground pins	-	50	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	see Section 6.3.14: Absolute maximum ratings (electrical sensitivity)		

Table 12. Current characteristics

Symbol	Ratings	Max.	Unit	
$\Sigma I_{VDD}$	Total current into sum of all $V_{DD_x}$ power lines (source) <sup>(1)</sup>	160	mA	
$\Sigma I_{VSS}$	Total current out of sum of all $V_{SS_x}$ ground lines (sink) <sup>(1)</sup>	-160		
$I_{VDD}$	Maximum current into each $V_{DD_x}$ power line (source) <sup>(1)</sup>	100		
$I_{VSS}$	Maximum current out of each $V_{SS_x}$ ground line (sink) <sup>(1)</sup>	-100		
$I_{IO}$	Output current sunk by any I/O and control pin	25		
	Output current sourced by any I/O and control pin	-25		
$\Sigma I_{IO}$	Total output current sunk by sum of all I/O and control pins <sup>(2)</sup>	120		
	Total output current sourced by sum of all I/Os and control pins <sup>(2)</sup>	-120		
$I_{INJ(PIN)}^{(3)}$	Injected current on FT pins <sup>(4)</sup>	-5/+0		
	Injected current on NRST and B pins <sup>(4)</sup>			
$\Sigma I_{INJ(PIN)}$	Total injected current (sum of all I/O and control pins) <sup>(5)</sup>	$\pm 25$		

$\pi$

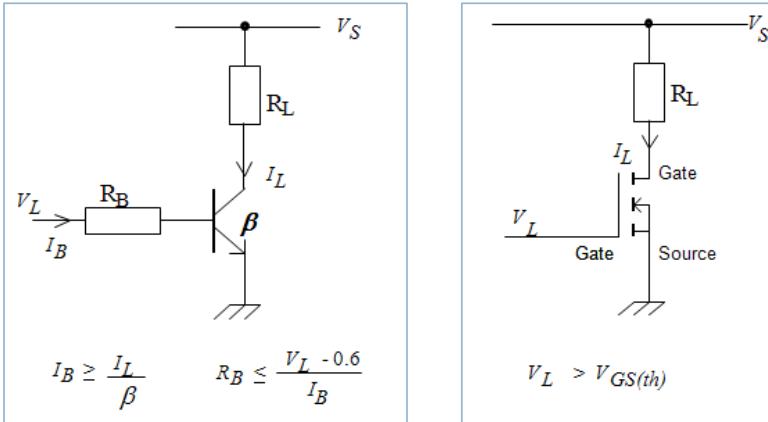
# Related Documents

The screenshot shows a Microsoft Teams interface with the 'General' channel selected. The 'Files' tab is active, displaying a list of related documents. The path 'Documents > General > Class Materials > Nucleo-F767' is shown at the top. The list includes six PDF files, all modified 'A few seconds ago' by 'sorayut glomglome'. The files are:

Name	Modified	Modified By
1. stm32f767zi datasheet.pdf	A few seconds ago	sorayut glomglome
2. rm0410 reference manual.pdf	A few seconds ago	sorayut glomglome
3. um1974 nucleo144 user manual.pdf	A few seconds ago	sorayut glomglome
4. MB1137 schematics.pdf	A few seconds ago	sorayut glomglome
5. um1905-description-of-stm32f7-hal-and...	A few seconds ago	sorayut glomglome
6. pm0253 programming manual.pdf	A few seconds ago	sorayut glomglome

# Working with Larger DC Loads

- MCU can drive simple DC loads directly with its digital I/O pins. (+/- 25 mA)
- MCU can't drive a load (e.g. motor) which needs more current than MCU port pin can supply.
- Need an interfacing circuit to draw current from a higher voltage.



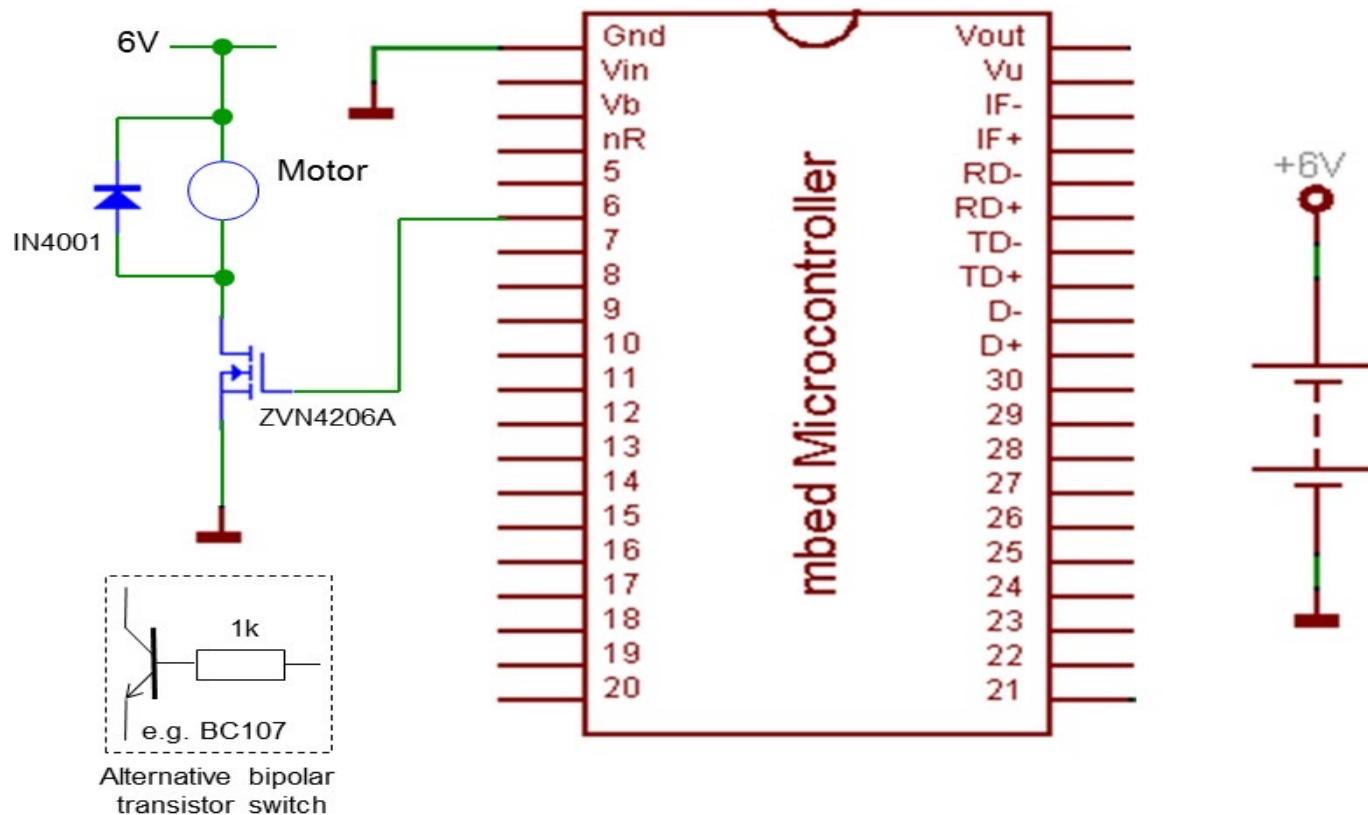
- A good switching transistor for small DC loads is the ZVN4206A.
- The maximum VGS threshold value, shown as 3 V.
- MOSFET will respond to the 3.3V Logic 1 output level of MCU.

Characteristic	ZVN4206A
Maximum Drain-Source Voltage $V_{DS}$	60V
Maximum Gate-Source Threshold $V_{GS(th)}$	3V
Maximum Drain-Source Resistance when 'On'. $R_{DS(on)}$	1.5Ω
Maximum Continuous Drain current $I_D$	600mA
Maximum Power Dissipation	0.7W
Input Capacitance	100pF

Table 3.5: Characteristics of the ZVN4206A n-channel MOSFET

$\pi$

# DC motor with Flying Diode



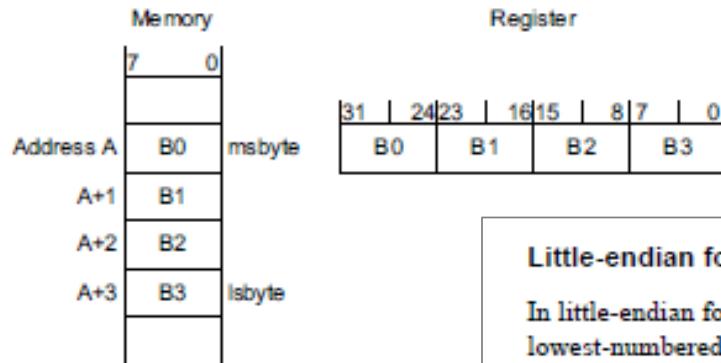
# Memory Organization

- Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space
- Bytes are coded in memory in *little endian* format
  - The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte, the most significant
- Addressable memory space is divided into 8 main blocks, each of 512 MB

# Big-endian and Little-endian format

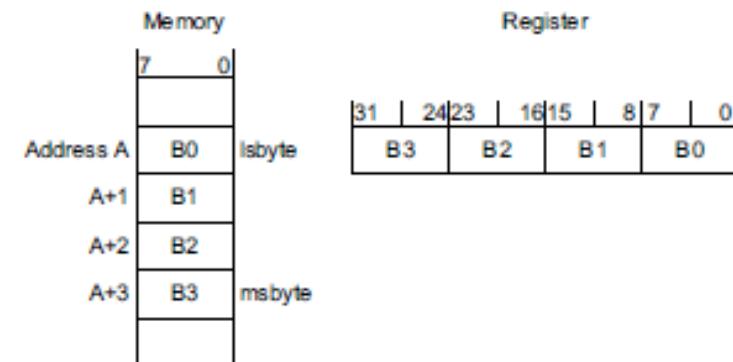
## Byte-invariant big-endian format

In byte-invariant big-endian format, the processor stores the most significant byte of a word at the lowest-numbered byte, and the least significant byte at the highest-numbered byte. For example:



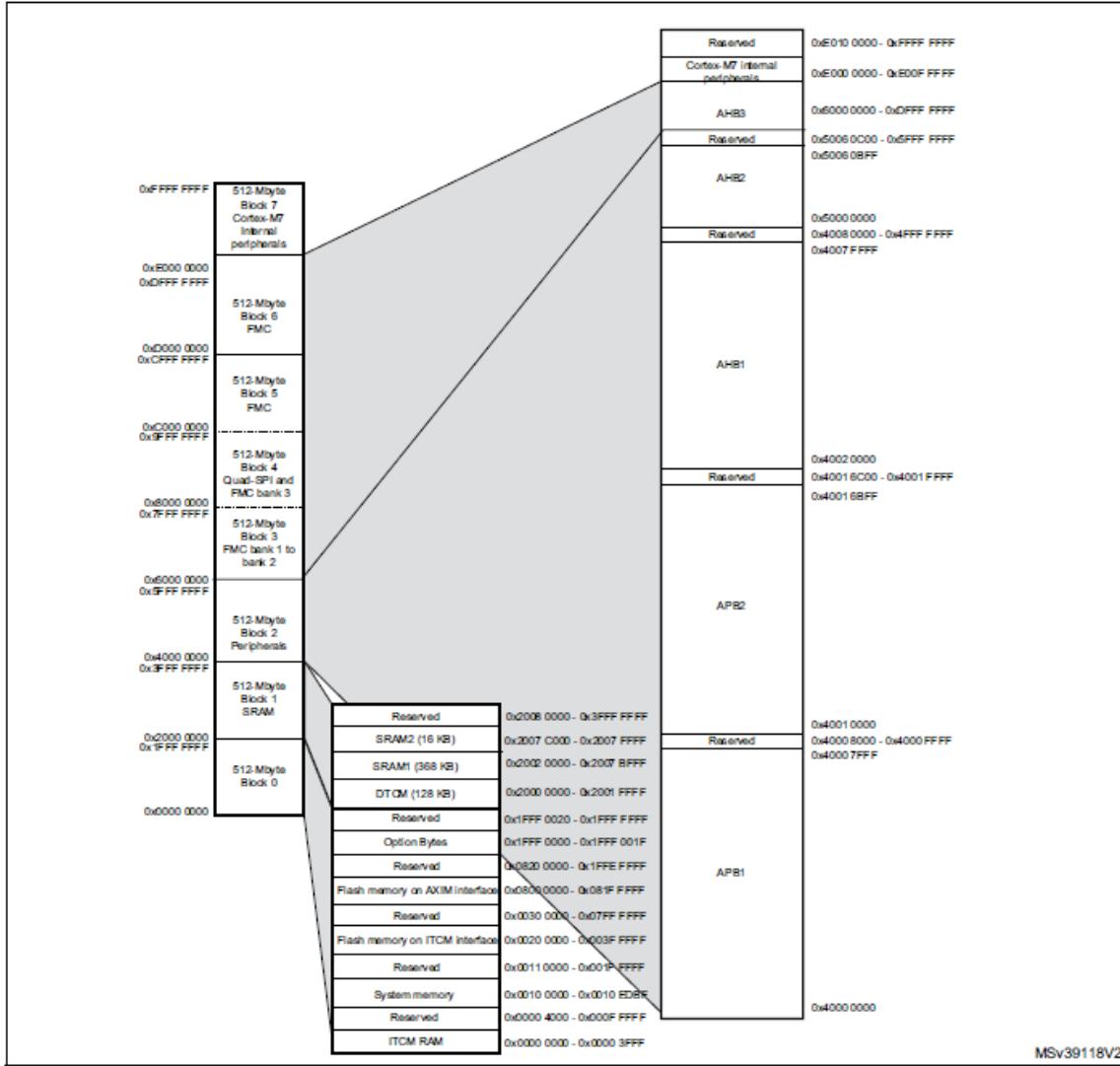
## Little-endian format

In little-endian format, the processor stores the least significant byte of a word at the lowest-numbered byte, and the most significant byte at the highest-numbered byte. For example:



π

Figure 22. Memory map



# Memory Map

$\pi$

# Memory Map

Bus	Boundary address	Peripheral
-	0xE00F FFFF - 0xFFFF FFFF	Reserved
Cortex-M7	0xE000 0000 - 0xE00F FFFF	Cortex-M7 internal peripherals
AHB1	0x4002 3C00 - 0x4002 3FFF	Flash interface register
	0x4002 3800 - 0x4002 3BFF	RCC
	0X4002 3400 - 0X4002 37FF	Reserved
	0x4002 3000 - 0x4002 33FF	CRC
	0x4002 2C00 - 0x4002 2FFF	Reserved
	0x4002 2800 - 0x4002 2BFF	GPIOK
	0x4002 2400 - 0x4002 27FF	GPIOJ
	0x4002 2000 - 0x4002 23FF	GPIOI
	0x4002 1C00 - 0x4002 1FFF	GPIOH
	0x4002 1800 - 0x4002 1BFF	GPIOG
	0x4002 1400 - 0x4002 17FF	GPIOF
	0x4002 1000 - 0x4002 13FF	GPIOE
	0X4002 0C00 - 0x4002 0FFF	GPIOD
	0x4002 0800 - 0x4002 0BFF	GPIOC
	0x4002 0400 - 0x4002 07FF	GPIOB
	0x4002 0000 - 0x4002 03FF	GPIOA

$\pi$

# Memory Map

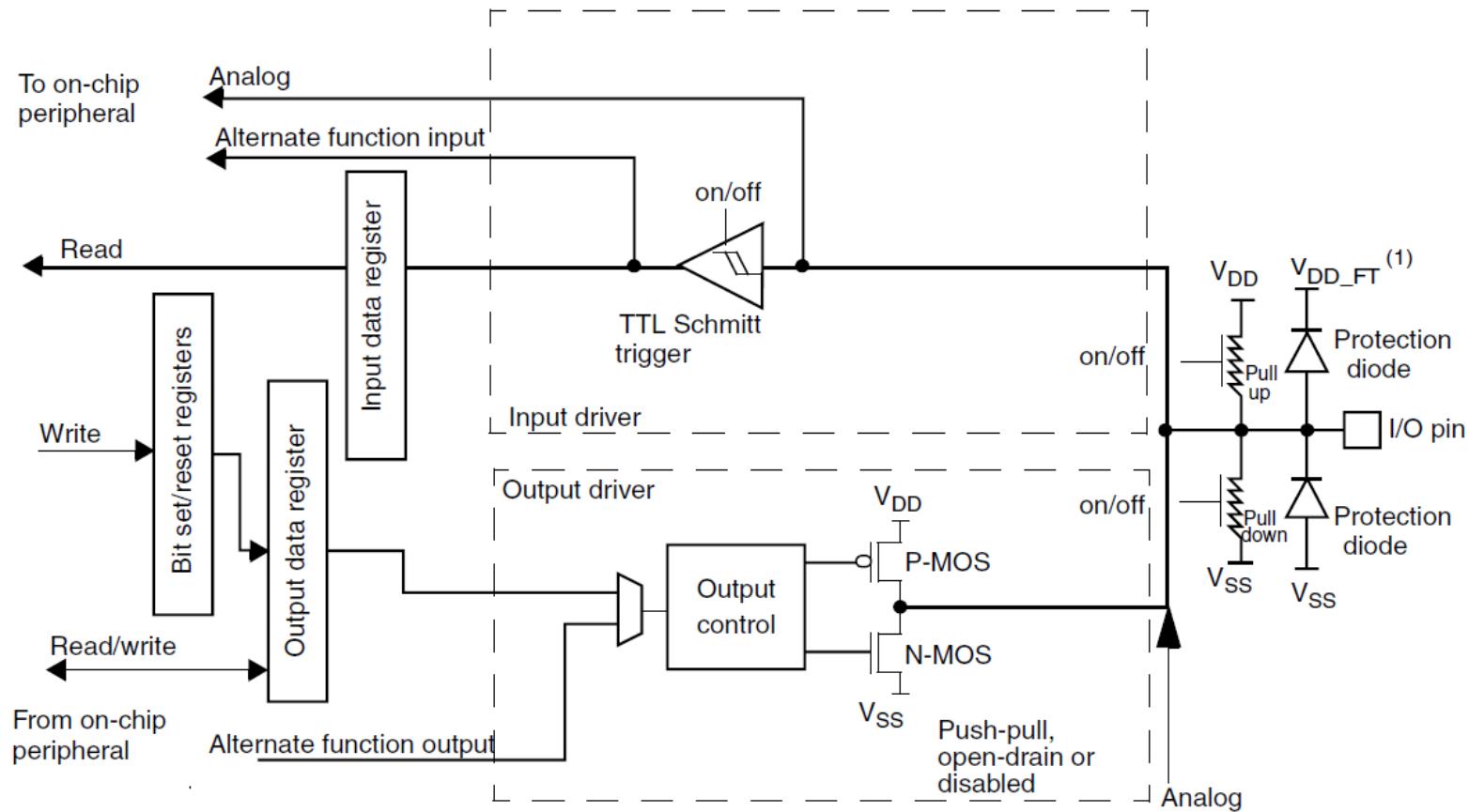
APB1

0x4000 6000 - 0x4000 63FF	I2C4
0x4000 5C00 - 0x4000 5FFF	I2C3
0x4000 5800 - 0x4000 5BFF	I2C2
0x4000 5400 - 0x4000 57FF	I2C1
0x4000 5000 - 0x4000 53FF	UART5
0x4000 4C00 - 0x4000 4FFF	UART4
0x4000 4800 - 0x4000 4BFF	USART3
0x4000 4400 - 0x4000 47FF	USART2
0x4000 4000 - 0x4000 43FF	SPDIFRX
0x4000 3C00 - 0x4000 3FFF	SPI3 / I2S3
0x4000 3800 - 0x4000 3BFF	SPI2 / I2S2
0x4000 3400 - 0x4000 37FF	CAN3
0x4000 3000 - 0x4000 33FF	IWDG
0x4000 2C00 - 0x4000 2FFF	WWDG
0x4000 2800 - 0x4000 2BFF	RTC & BKP Registers
0x4000 2400 - 0x4000 27FF	LPTIM1
0x4000 2000 - 0x4000 23FF	TIM14

# General-Purpose I/Os (GPIOs)

- Each of the general-purpose I/O ports has:
  - Four 32-bit configuration registers (GPIOx\_MODER, GPIOx\_OTYPER, GPIOx\_OSPEEDR and GPIOx\_PUPDR)
  - Two 32-bit data registers (GPIOx\_IDR and GPIOx\_ODR)
  - 32-bit set/reset register (GPIOx\_BSRR)
  - 32-bit locking register (GPIOx\_LCKR)
  - Two 32-bit alternate function (GPIOx\_AFRH and GPIOx\_AFRL)
- Each port bit of GPIOs can be individually configured by software in several modes:
  - Input floating
  - Input pull-up
  - Input-pull-down
  - Analog
  - Output open-drain with pull-up or pull-down capability
  - Output push-pull with pull-up or pull-down capability
  - Alternate function push-pull with pull-up or pull-down capability
  - Alternate function open-drain with pull-up or pull-down capability

# Basic Structure of a GPIO Bit (5V tolerance)



# GPIO Operation

- During and just after reset, the alternate functions are not active and the I/O ports are configured in **Input Floating** mode ( $\text{MODERx}[1:0]=00\text{b}$ ,  $\text{PUPDRx}[1:0]=00\text{b}$ )
- When configured as output, the value written to the Output Data register ( $\text{GPIOx\_ODR}$ ) is output on the I/O pin.  
It is possible to use the output driver in Push-Pull mode or Open-Drain mode (only the N-MOS is activated when outputting 0).
- The Input Data register ( $\text{GPIOx\_IDR}$ ) captures the data present on the I/O pin at every AHB1 clock cycle
- All GPIO pins have an internal weak pull-up and weak pull-down which can be activated or not when configured as input

# GPIO Atomic Bit Set or Reset

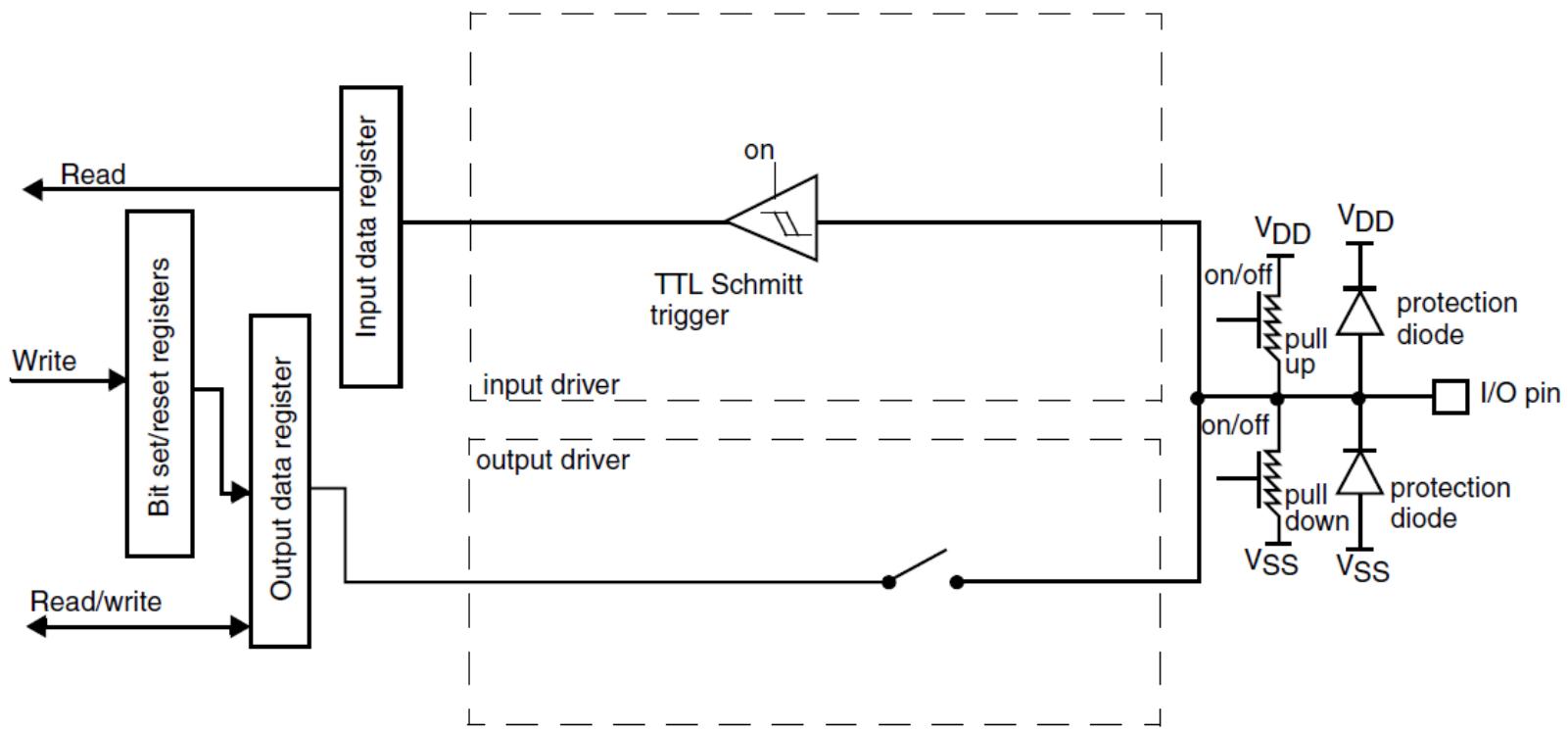
- Atomic Read/Modify access
  - No interruption in the middle to cause errors
- Atomic operations ensure that the desired change is not interrupted resulting in partial set/reset of GPIOs
- There is no need for the software to disable interrupts when programming the GPIOx\_ODR at bit level:
- It is possible to modify only one or several bits in a single atomic AHB1 write access
- This is achieved by programming to ‘1’ the Bit Set/Reset Register (GPIOx\_BSRR, or for reset only GPIOx\_BRR) to select the bits you want to modify.

Unselected bits will not be modified

# Input Configuration

- When the I/O Port is programmed as Input:
  - The Output Buffer is disabled
  - The Schmitt Trigger Input is activated
  - The weak pull-up and pull-down resistors are activated or not depending on input configuration (pull-up, pull-down or floating)
  - The data present on the I/O pin is sampled into the Input Data Register every AHB1 clock cycle
  - A read access to the Input Data Register obtains the I/O State

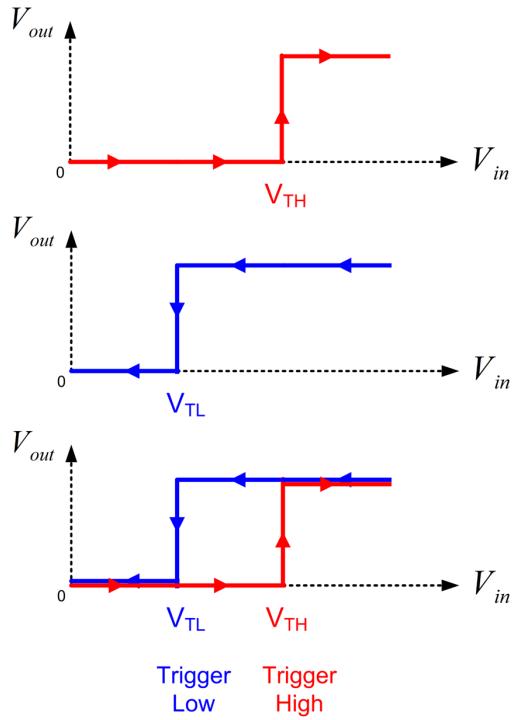
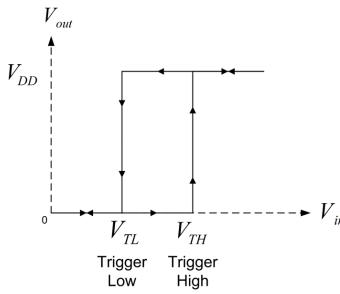
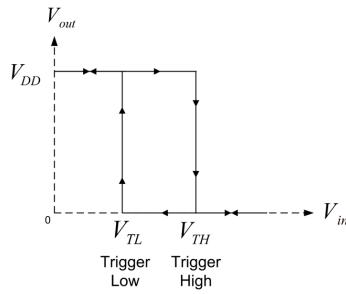
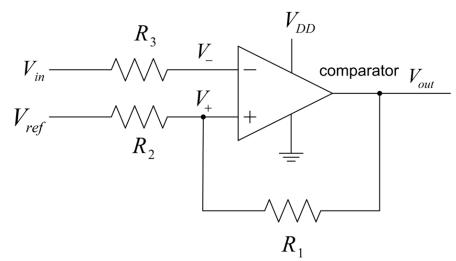
# Input Configuration



sil5040b

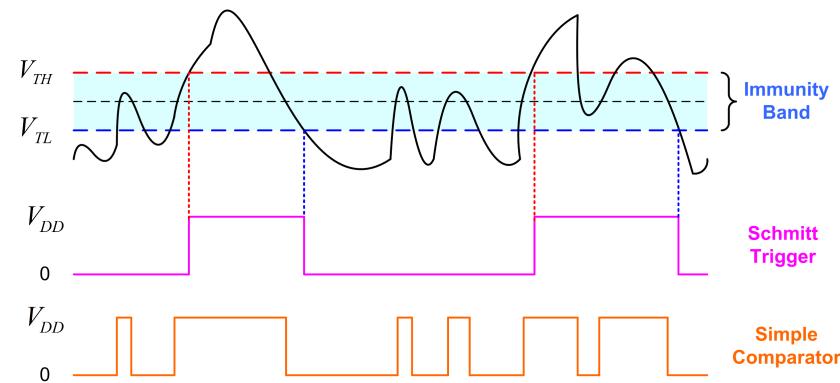
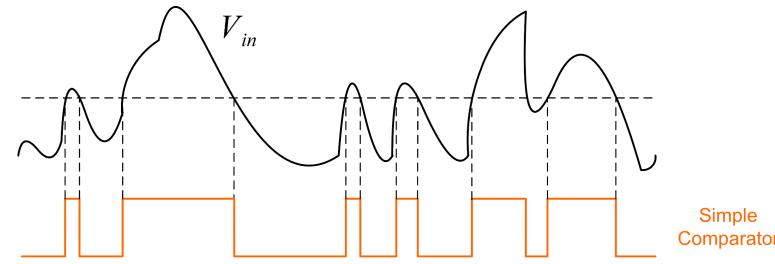
$\pi$

# Schmitt Trigger



$\pi$

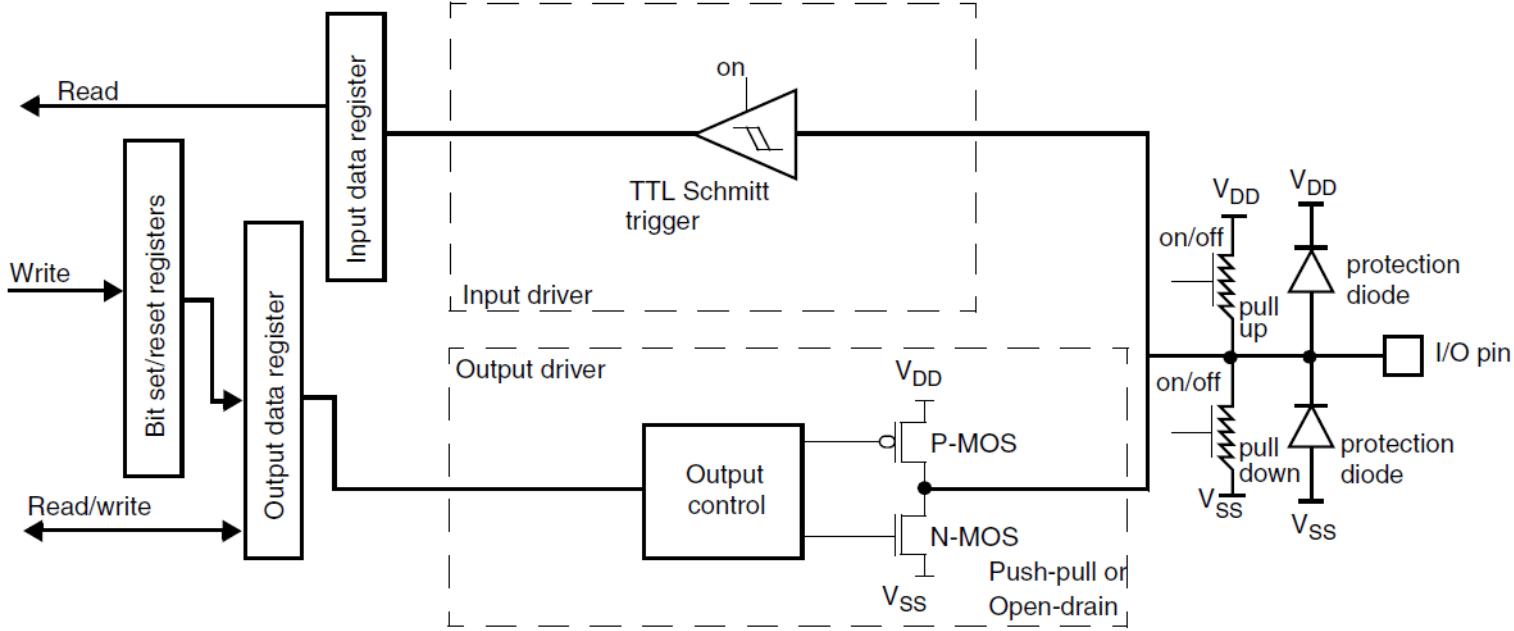
# Scmitt Trigger Output



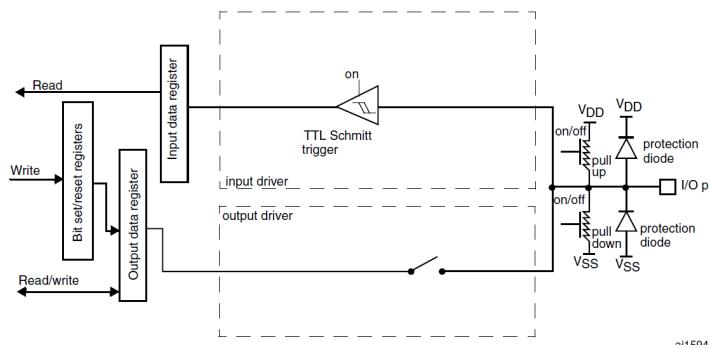
# Output Configuration

- When the I/O Port is programmed as Output:
  - The Output Buffer is enabled:
    - Open Drain Mode: A “0” in the Output register activates the N-MOS while a “1” in the Output register leaves the port in Hi-Z. (the P-MOS is never activated)
    - Push-Pull Mode: A “0” in the Output register activates the N-MOS while a “1” in the Output register activates the P-MOS
  - The Schmitt Trigger Input is activated.
  - The weak pull-up and pull-down resistors are disabled.
  - The data present on the I/O pin is sampled into the Input Data Register every AHB1 clock cycle
  - Read access to Input Data Register gets the I/O state
  - Read access to Output Data register gets last written value

$\pi$

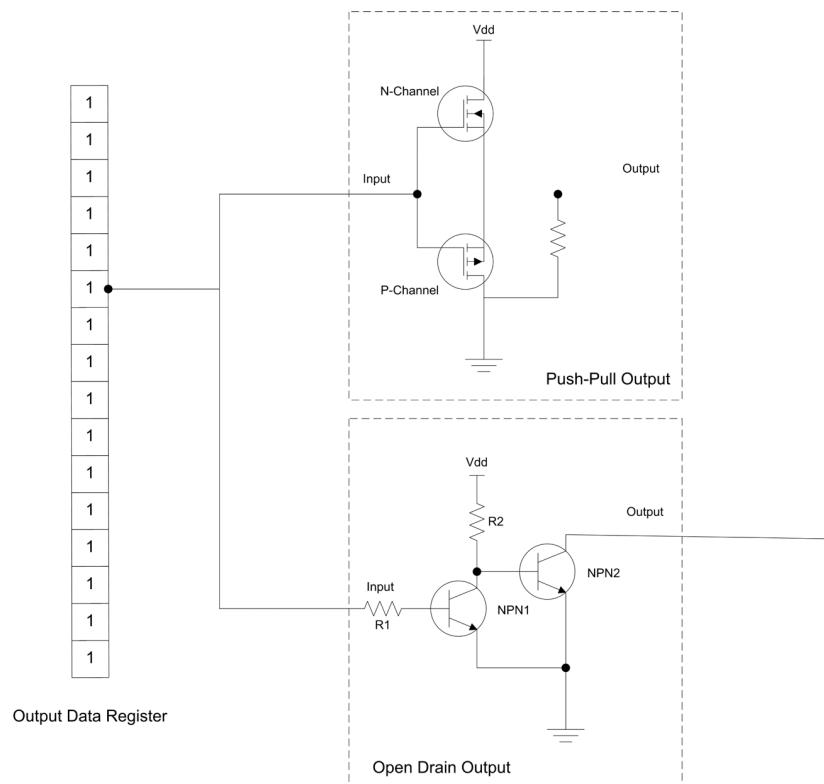


Compare to input configuration

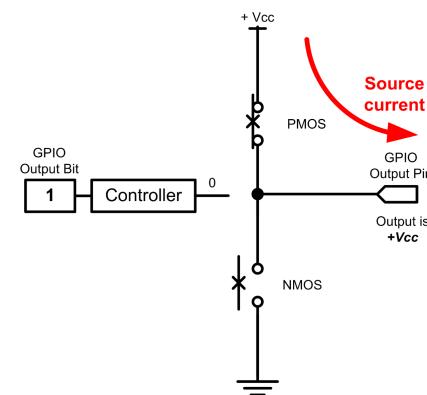
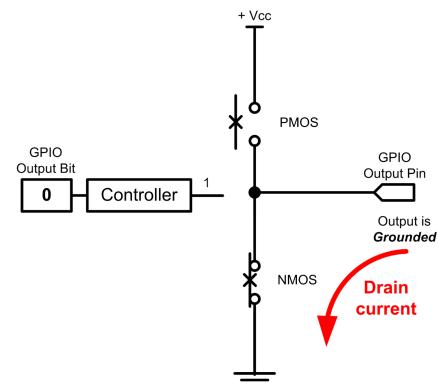
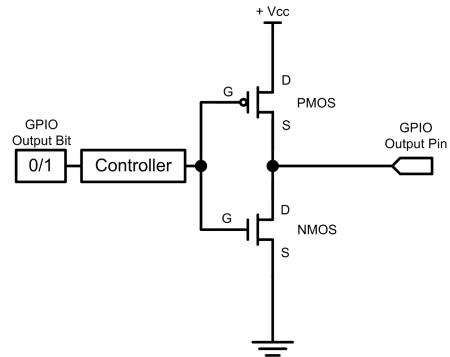


$\pi$

# Output Data Register



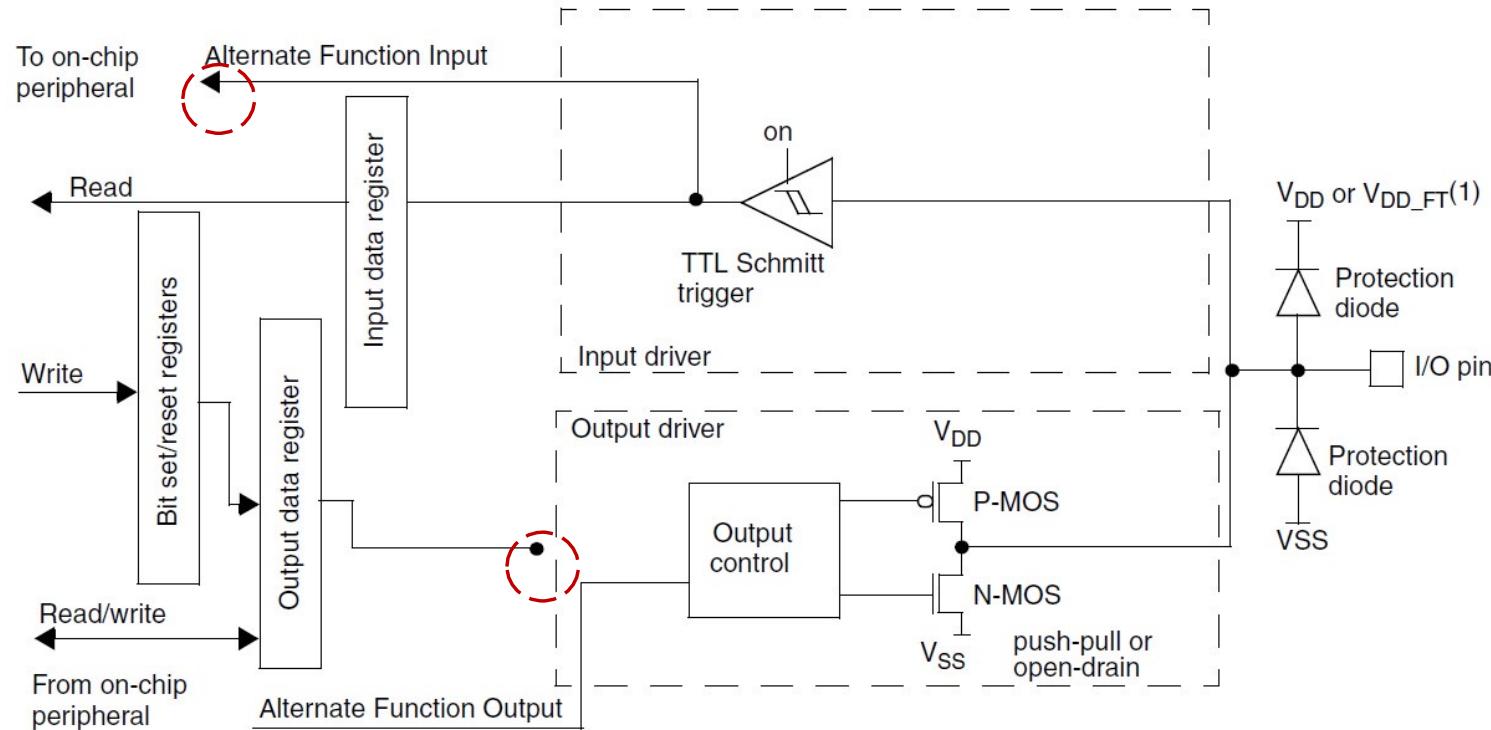
# Output Push Pull



# Alternate Function Configuration

- When the I/O Port is programmed as Alternate Function:
  - The Output Buffer is turned on in Open Drain or Push-Pull configuration
  - The Output Buffer is driven by the signal coming from the peripheral (alternate function out)
  - The Schmitt Trigger Input is activated
  - The weak pull-up and pull-down resistors are disabled
  - The data present on the I/O pin is sampled into the Input Data Register every AHB1 clock cycle
  - A read access to the Input Data Register gets the I/O state in open drain mode
  - A read access to the Output Data register gets the last written value in Push-Pull mode

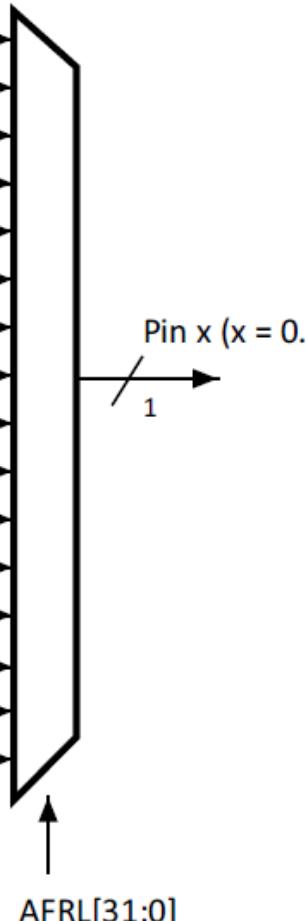
# Alternate Function Configuration



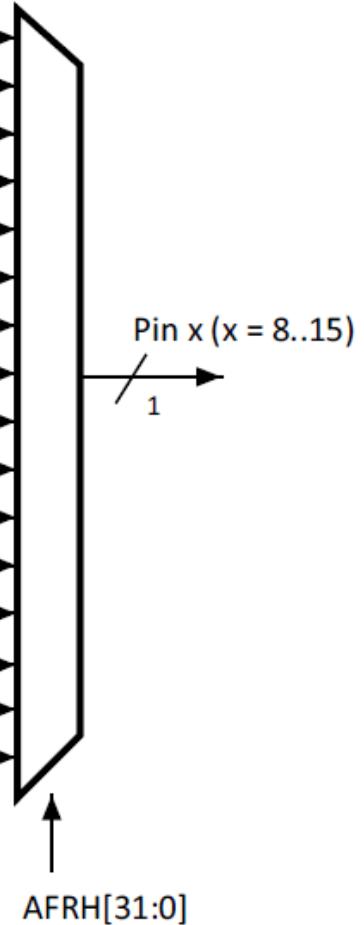
$\pi$

# Alternate Function Configuration

AF0 (system)  
AF1 (TIM1/TIM2)  
AF2 (TIM3..5)  
AF3 (TIM9..11)  
AF4 (I2C1..3)  
AF5 (SPI1..4)  
AF6 (SPI3..5)  
AF7 (USART1..2)  
AF8 (USART6)  
AF9 (I2C2..3)  
AF10 (OTG\_FS)  
AF11  
AF12 (SDIO)  
AF13  
AF14  
AF15 (EVENTOUT)



AF0 (system)  
AF1 (TIM1/TIM2)  
AF2 (TIM3..5)  
AF3 (TIM9..11)  
AF4 (I2C1..3)  
AF5 (SPI1..4)  
AF6 (SPI3..5)  
AF7 (USART1..2)  
AF8 (USART6)  
AF9 (I2C2..3)  
AF10 (OTG\_FS)  
AF11  
AF12 (SDIO)  
AF13  
AF14  
AF15 (EVENTOUT)

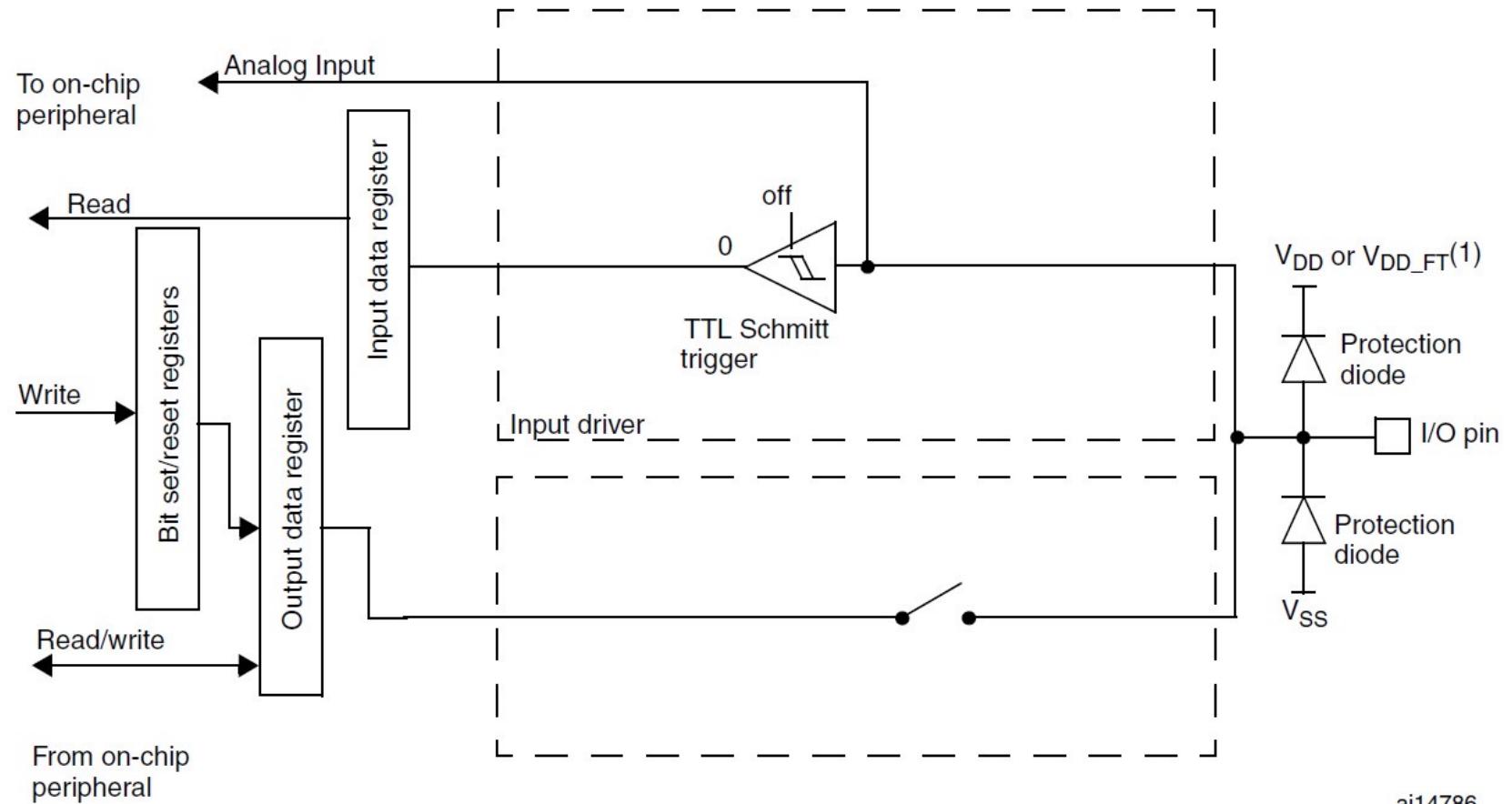


# Analog Configuration

- When the I/O Port is programmed as Analog configuration:
  - The Output Buffer is disabled.
  - The Schmitt Trigger Input is de-activated providing zero consumption for every analog value of the I/O pin.
  - The output of the Schmitt Trigger is forced to a constant value (0).
  - The weak pull-up and pull-down resistors are disabled.
  - Read access to the Input Data Register gets the value “0”.

$\pi$

# Analog I/O Configuration



ai14786

38

$\pi$

# GPIO port mode register

## 6.4.1 GPIO port mode register (GPIOx\_MODER) (x =A..K)

Address offset:0x00

Reset values:

- 0xA800 0000 for port A
- 0x0000 0280 for port B
- 0x0000 0000 for other ports

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw										

Bits 2y+1:2y MODERy[1:0]: Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O mode.

- 00: Input mode (reset state)
- 01: General purpose output mode
- 10: Alternate function mode
- 11: Analog mode

# GPIO port output type register

## 6.4.2 GPIO port output type register (GPIOx\_OTYPER) (x = A..K)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw															

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **OTy**: Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O output type.

0: Output push-pull (reset state)

1: Output open-drain

# GPIO port output speed register

## 6.4.3 GPIO port output speed register (GPIOx\_OSPEEDR) (x = A..K)

Address offset: 0x08

Reset value:

- 0x0C00 0000 for port A
- 0x0000 00C0 for port B
- 0x0000 0000 for other ports

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
rw	rw	rw	rw	rw	rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7 [1:0]		OSPEEDR6 [1:0]		OSPEEDR5 [1:0]		OSPEEDR4 [1:0]		OSPEEDR3 [1:0]		OSPEEDR2 [1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
rw	rw	rw	rw	rw	rw										

Bits 2y+1:2y **OSPEEDR<sub>y</sub>[1:0]**: Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O output speed.

- 00: Low speed
- 01: Medium speed
- 10: High speed
- 11: Very high speed

# GPIO port pull-up/pull-down register

## 6.4.4 GPIO port pull-up/pull-down register (GPIOx\_PUPDR) (x = A..K)

Address offset: 0x0C

Reset values:

- 0x6400 0000 for port A
- 0x0000 0100 for port B
- 0x0000 0000 for other ports

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]	PUPDR14[1:0]	PUPDR13[1:0]	PUPDR12[1:0]	PUPDR11[1:0]	PUPDR10[1:0]	PUPDR9[1:0]	PUPDR8[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]	PUPDR6[1:0]	PUPDR5[1:0]	PUPDR4[1:0]	PUPDR3[1:0]	PUPDR2[1:0]	PUPDR1[1:0]	PUPDR0[1:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 2y+1:2y **PUPDR<sub>y</sub>[1:0]**: Port x configuration bits (y = 0..15)

These bits are written by software to configure the I/O pull-up or pull-down

- 00: No pull-up, pull-down
- 01: Pull-up
- 10: Pull-down
- 11: Reserved

# GPIO port input data register

## 6.4.5 GPIO port input data register (GPIOx\_IDR) (x = A..K)

Address offset: 0x10

Reset value: 0x0000 XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **IDRy**: Port input data bit (y = 0..15)

These bits are read-only. They contain the input value of the corresponding I/O port.

# GPIO port output data register

## 6.4.6 GPIO port output data register (GPIOx\_ODR) (x = A..K)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:16 Reserved, must be kept at reset value.

Bits 15:0 **ODRy**: Port output data bit (y = 0..15)

# GPIO port bit set/reset register

## 6.4.7 GPIO port bit set/reset register (GPIOx\_BSRR) (x = A..K)

Address offset: 0x18

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. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Resets 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. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Sets the corresponding ODRx bit

# GPIO port configuration lock register

## 6.4.8 GPIO port configuration lock register (GPIOx\_LCKR) (x = A..K)

This register is used to lock the configuration of the port bits when a correct write sequence is applied to bit 16 (LCKK). The value of bits [15:0] is used to lock the configuration of the GPIO. During the write sequence, the value of LCKR[15:0] must not change. When the LOCK sequence has been applied on a port bit, the value of this port bit can no longer be modified until the next MCU reset or peripheral reset.

Each lock bit freezes a specific configuration register (control and alternate function registers).

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	LCKK
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

# GPIO port configuration lock register

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **LCKK:** Lock key

This bit can be read any time. It can only be modified using the lock key write sequence.

0: Port configuration lock key not active

1: Port configuration lock key active. The GPIOx\_LCKR register is locked until the next MCU reset or peripheral reset.

LOCK key write sequence:

WR LCKR[16] = '1' + LCKR[15:0]

WR LCKR[16] = '0' + LCKR[15:0]

WR LCKR[16] = '1' + LCKR[15:0]

RD LCKR

RD LCKR[16] = '1' (this read operation is optional but it confirms that the lock is active)

Bits 15:0 **LCKy:** Port x lock bit y (y= 0..15)

These bits are read/write but can only be written when the LCKK bit is '0'.

0: Port configuration not locked

1: Port configuration locked

# GPIO alternate function low register

## 6.4.9 GPIO alternate function low register (GPIOx\_AFRL) (x = A..K)

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
rw	rw	rw	rw												

Bits 31:0 **AFRy[3:0]**: Alternate function selection for port x pin y (y = 0..7)

These bits are written by software to configure alternate function I/Os

AFSELy selection:

0000: AF0	1000: AF8
0001: AF1	1001: AF9
0010: AF2	1010: AF10
0011: AF3	1011: AF11
0100: AF4	1100: AF12
0101: AF5	1101: AF13
0110: AF6	1110: AF14
0111: AF7	1111: AF15

# GPIO alternate function high register

## 6.4.10 GPIO alternate function high register (GPIOx\_AFRH) (x = A..J)

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
rw	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
rw	rw	rw	rw												

Bits 31:0 **AFRy[3:0]**: Alternate function selection for port x pin y (y = 8..15)

These bits are written by software to configure alternate function I/Os

AFSELy selection:

0000: AF0	1000: AF8
0001: AF1	1001: AF9
0010: AF2	1010: AF10
0011: AF3	1011: AF11
0100: AF4	1100: AF12
0101: AF5	1101: AF13
0110: AF6	1110: AF14
0111: AF7	1111: AF15

# GPIO Bit Configuration Table

Table 24. Port bit configuration table<sup>(1)</sup>

MODE(i) [1:0]	OTYPER(i)	OSPEED(i) [1:0]	PUPD(i) [1:0]		I/O configuration	
01	0	SPEED [1:0]	0	0	GP output	PP
	0		0	1	GP output	PP + PU
	0		1	0	GP output	PP + PD
	0		1	1	Reserved	
	1		0	0	GP output	OD
	1		0	1	GP output	OD + PU
	1		1	0	GP output	OD + PD
	1		1	1	Reserved (GP output OD)	

$\pi$

MODE(i) [1:0]	OTYPER(i)	OSPEED(i) [1:0]	PUPD(i) [1:0]		I/O configuration	
10	0	SPEED [1:0]	0	0	AF	PP
	0		0	1	AF	PP + PU
	0		1	0	AF	PP + PD
	0		1	1	Reserved	
	1		0	0	AF	OD
	1		0	1	AF	OD + PU
	1		1	0	AF	OD + PD
	1		1	1	Reserved	
	x	x	x	0	0	Input
00	x	x	x	0	1	PU
	x	x	x	1	0	PD
	x	x	x	1	1	Reserved (input floating)
	x	x	x	0	0	Input/output
11	x	x	x	0	1	Reserved
	x	x	x	1	0	
	x	x	x	1	1	

# Accessing GPIOA/GPIOx

```
#define PERIPH_BASE ((uint32_t)0x40000000U)
```

```
#define AHB1PERIPH_BASE (PERIPH_BASE + 0x00020000U)
```

```
#define GPIOA_BASE (AHB1PERIPH_BASE + 0x0000U)
```

```
#define GPIOA ((GPIO_TypeDef *) GPIOA_BASE)
```



# Stm32f767xx.h

```
typedef struct
{
    __IO uint32_t MODER;      /*!< GPIO port mode register,          Address offset: 0x00 */
    __IO uint32_t OTYPER;     /*!< GPIO port output type register, Address offset: 0x04 */
    __IO uint32_t OSPEEDR;    /*!< GPIO port output speed register, Address offset: 0x08 */
    __IO uint32_t PUPDR;      /*!< GPIO port pull-up/pull-down register, Address offset: 0x0C */
    __IO uint32_t IDR;        /*!< GPIO port input data register,   Address offset: 0x10 */
    __IO uint32_t ODR;        /*!< GPIO port output data register,  Address offset: 0x14 */
    __IO uint32_t BSRR;       /*!< GPIO port bit set/reset register, Address offset: 0x18 */
    __IO uint32_t LCKR;       /*!< GPIO port configuration lock register, Address offset: 0x1C */
    __IO uint32_t AFR[2];     /*!< GPIO alternate function registers, Address offset: 0x20-0x24 */
} GPIO_TypeDef;
```



# GPIO Initialization

```
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct;
    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOB_CLK_ENABLE();
    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0|GPIO_PIN_14|GPIO_PIN_7, GPIO_PIN_RESET);
    /*Configure GPIO pins : PB0 PB14 PB7 */
    GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_14|GPIO_PIN_7;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
}
```

# GPIO Operations

## HAL\_GPIO\_ReadPin

Function Name	<b>GPIO_PinState HAL_GPIO_ReadPin ( <i>GPIO_TypeDef</i> * GPIOx, uint16_t GPIO_Pin)</b>
Function Description	Reads the specified input port pin.
Parameters	<ul style="list-style-type: none"><li>• <b>GPIOx</b> : where x can be (A..K) to select the GPIO peripheral for STM32F429X device or x can be (A..I) to select the GPIO peripheral for STM32F40XX and STM32F427X devices.</li><li>• <b>GPIO_Pin</b> : specifies the port bit to read. This parameter can be GPIO_PIN_x where x can be (0..15).</li></ul>
Return values	<ul style="list-style-type: none"><li>• <b>The input port pin value.</b></li></ul>
Notes	<ul style="list-style-type: none"><li>• None.</li></ul>



## HAL\_GPIO\_WritePin

Function Name	<code>void HAL_GPIO_WritePin ( <i>GPIO_TypeDef</i> * GPIOx, uint16_t GPIO_Pin, <i>GPIO_PinState</i> PinState)</code>
Function Description	Sets or clears the selected data port bit.
Parameters	<ul style="list-style-type: none"><li>• <b>GPIOx</b> : where x can be (A..K) to select the GPIO peripheral for STM32F429X device or x can be (A..I) to select the GPIO peripheral for STM32F40XX and STM32F427X devices.</li><li>• <b>GPIO_Pin</b> : specifies the port bit to be written. This parameter can be one of <i>GPIO_PIN_x</i> where x can be (0..15).</li><li>• <b>PinState</b> : specifies the value to be written to the selected bit. This parameter can be one of the <i>GPIO_PinState</i> enum values: <i>GPIO_BIT_RESET</i>: to clear the port pin <i>GPIO_BIT_SET</i>: to set the port pin</li></ul>
Return values	<ul style="list-style-type: none"><li>• None.</li></ul>
Notes	<ul style="list-style-type: none"><li>• This function uses <i>GPIOx_BSRR</i> register to allow atomic read/modify accesses. In this way, there is no risk of an IRQ occurring between the read and the modify access.</li></ul>



## HAL\_GPIO\_TogglePin

Function Name	<code>void HAL_GPIO_TogglePin ( <i>GPIO_TypeDef</i> * GPIOx, uint16_t GPIO_Pin)</code>
Function Description	Toggles the specified GPIO pins.
Parameters	<ul style="list-style-type: none"><li>• <b>GPIOx</b> : Where x can be (A..K) to select the GPIO peripheral for STM32F429X device or x can be (A..I) to select the GPIO peripheral for STM32F40XX and STM32F427X devices.</li><li>• <b>GPIO_Pin</b> : Specifies the pins to be toggled.</li></ul>
Return values	<ul style="list-style-type: none"><li>• None.</li></ul>
Notes	<ul style="list-style-type: none"><li>• None.</li></ul>