

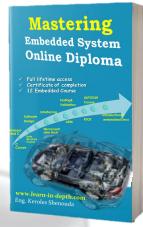
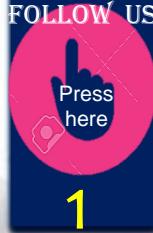
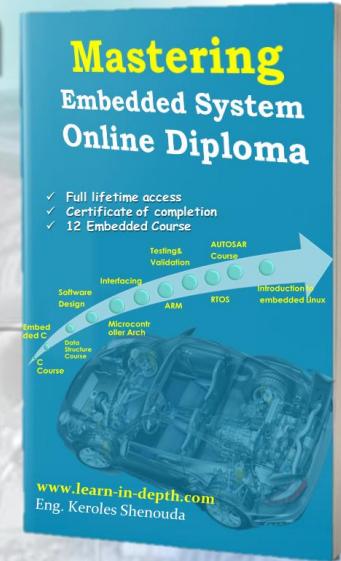
# Mastering Embedded System

## Online Diploma

- ✓ Full lifetime access
- ✓ Access on Android mobile and PC (Windows)
- ✓ Certificate of completion
- ✓ 12 Embedded Course

### Unit 7 (MCU Essential Peripherals) . lesson 2 GPIO Part2

- ▶ GPIO Pins and Alternate Functions
- ▶ Thinking Question
  - ▶ List all the Alt. connections with PortA.7 in TivaC
  - ▶ List all the Alt. connections with PortA.3 in BluePill Stm32F103XXX
- ▶ Alternate functions (AF) functional description
- ▶ Deep dive inside SoC to see PA.0 configured to be external interrupt
- ▶ Enable CANTX /RX ALT PINs
- ▶ AFIO registers
- ▶ EVENTOUT Cortex® output
- ▶ Software remapping of I/O alternate functions
- ▶ External interrupt configuration
- ▶ Driver Development Sequence
  - ▶ MCU Device Header
  - ▶ Specify the memory Map

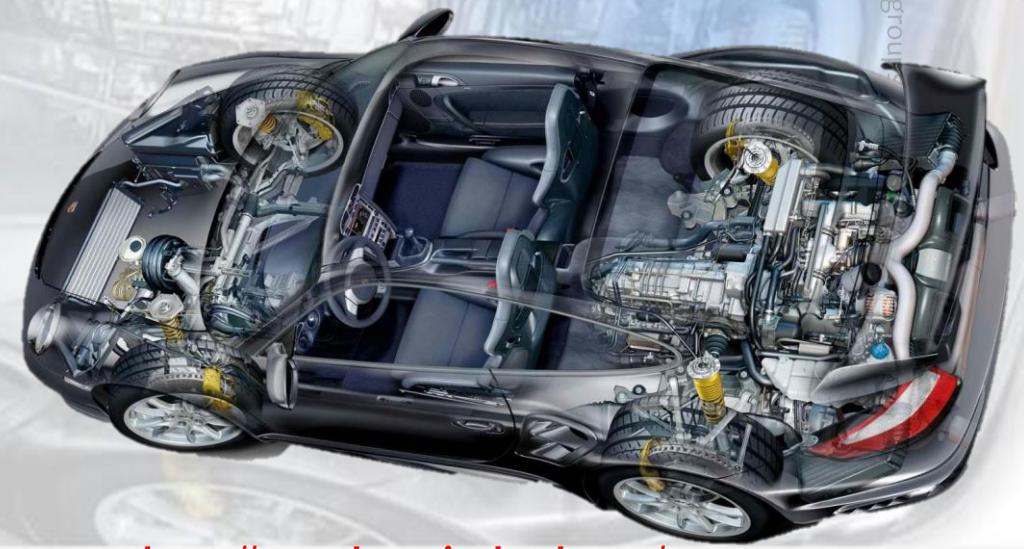


#LEARN\_IN\_DEPTH

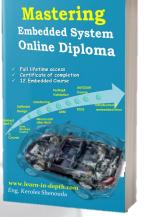
#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

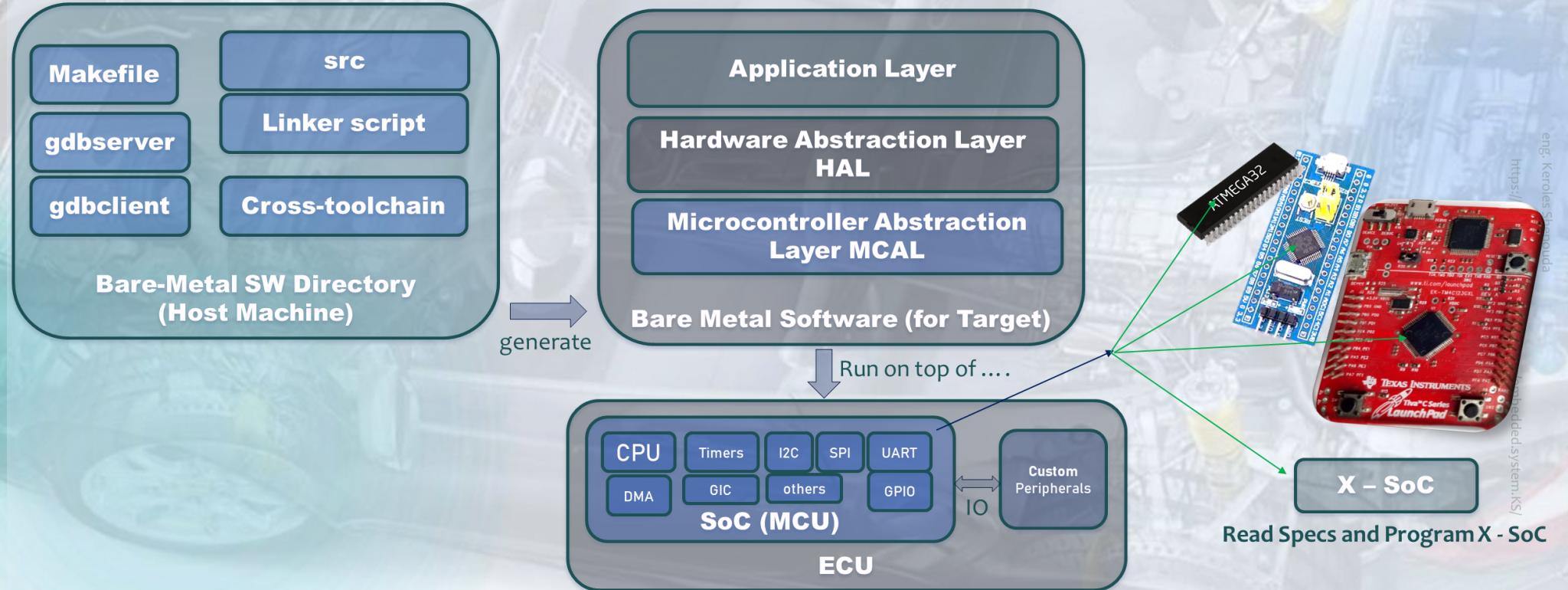


2

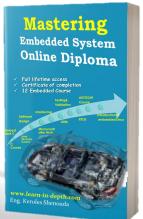
#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng\_Keroles Shenouda  
<https://www.facebook.com/groups/embedded.system.KS/>

# Big Picture



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

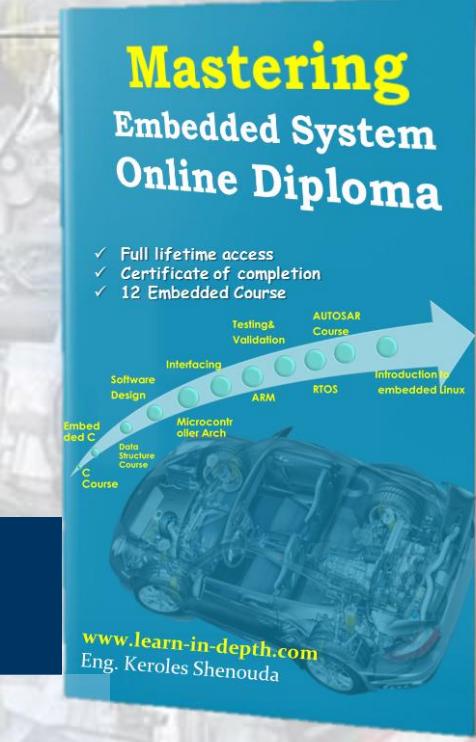


#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

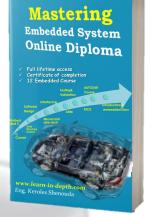
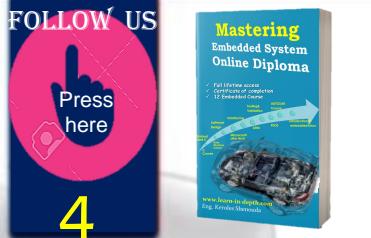
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



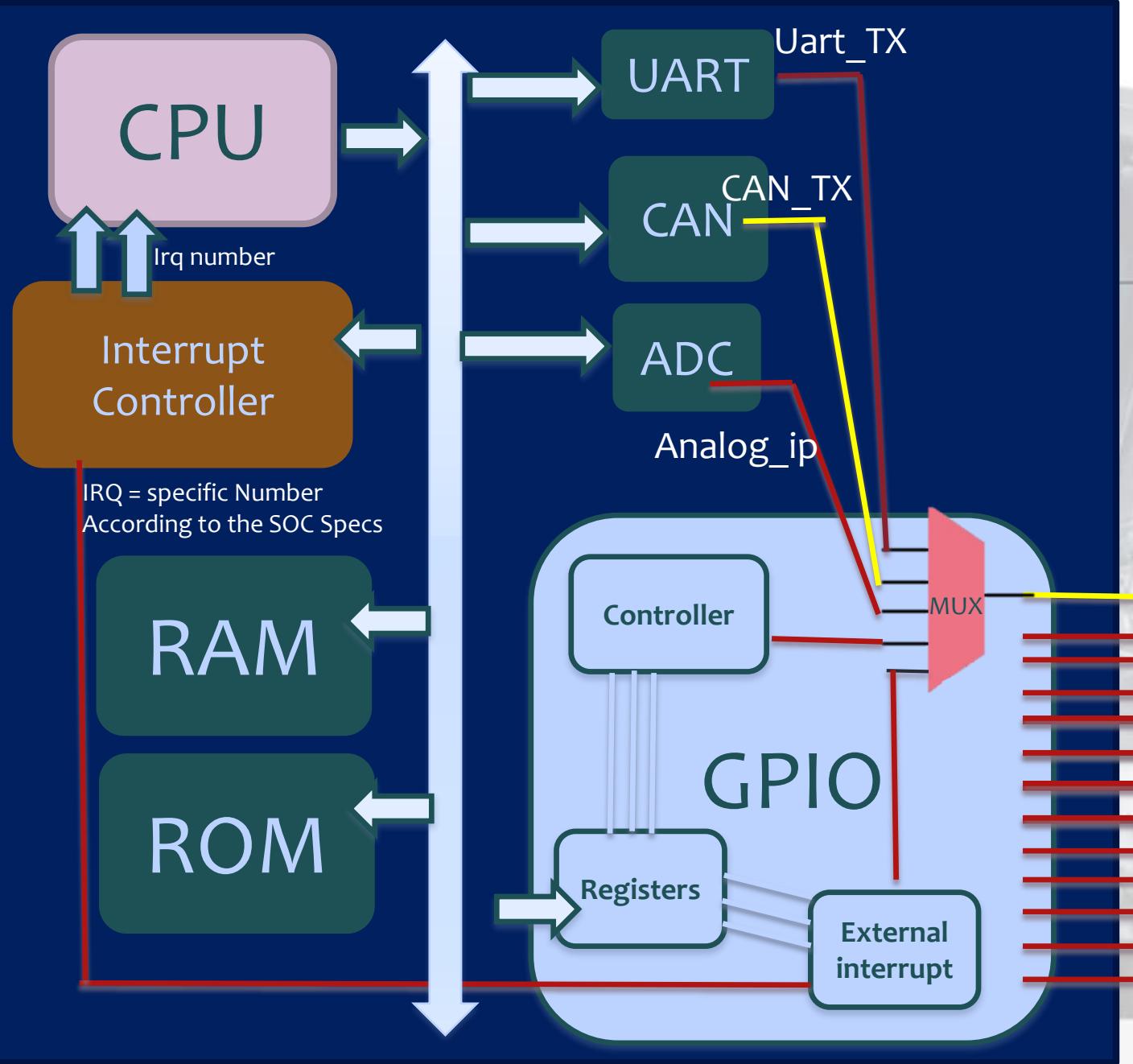
**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system



# Each PIN

Can be configured as

Alternative Signal  
Like in this example

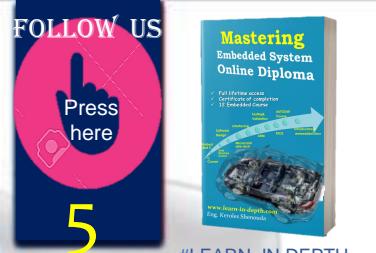
GPIO Signal

UART\_TX  
CAN\_TX  
Analog\_ip

PIN = CAN\_TX

You Can Configure PIN to  
be alternative Signal  
Then Write on the MUX  
register to choose the  
**CAN\_TX**

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

## Mastering Embedded System Online Diploma

- ✓ Full lifetime access
- ✓ Certificate of completion
- ✓ 12 Embedded Course



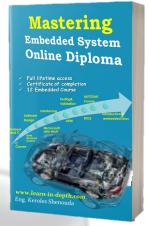
[www.learn-in-depth.com](http://www.learn-in-depth.com)  
Eng. Keroles Shenouda

**LEARN-IN-DEPTH**  
Be professional in  
embedded system

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

# Thinking Question

LIST ALL THE ALT. CONNECTIONS WITH PORTA.7 IN TIVAC



6

#LEARN\_IN\_DEPTH

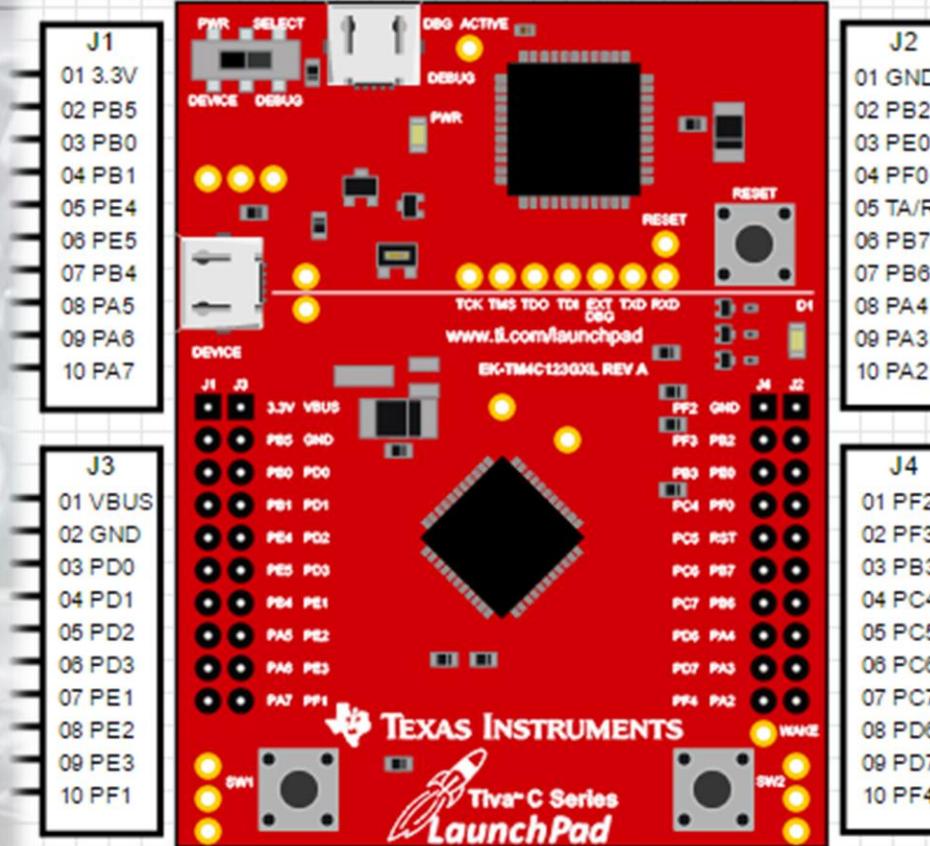
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

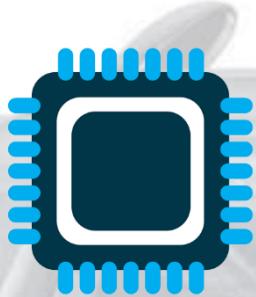
<https://www.facebook.com/groups/embedded.system.KS/>

# Thinking Question

List all the Alt. connections with PortA.7 in TivaC



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

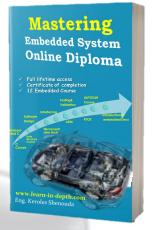


24

Tiva™ TM4C123GH6PM Microcontroller



7



#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

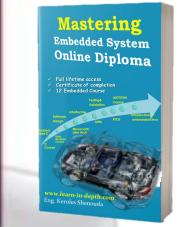
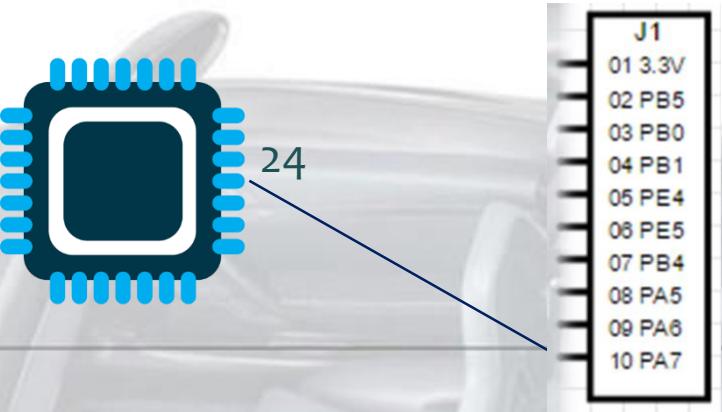
## 23.4 GPIO Pins and Alternate Functions

**Table 23-5. GPIO Pins and Alternate Functions**

IO	Pin	Analog Function	Digital Function (GPIOPCTL PMCx Bit Field Encoding) <sup>a</sup>											
			1	2	3	4	5	6	7	8	9	14	15	
PA0	17	-	U0Rx	-	-	-	-	-	-	CAN1Rx	-	-	-	
PA1	18	-	U0Tx	-	-	-	-	-	-	CAN1Tx	-	-	-	
PA2	19	-	-	SSI0Clk	-	-	-	-	-	-	-	-	-	
PA3	20	-	-	SSI0Fss	-	-	-	-	-	-	-	-	-	
PA4	21	-	-	SSI0Rx	-	-	-	-	-	-	-	-	-	
PA5	22	-	-	SSI0Tx	-	-	-	-	-	-	-	-	-	
PA6	23	-	-	-	I2C1SCL	-	M1PWM2	-	-	-	-	-	-	
PA7	24	-	-	-	I2C1SDA	-	M1PWM3	-	-	-	-	-	-	
PB0	45	USB0ID	U1Rx	-	-	-	-	-	T2CCP0	-	-	-	-	
PB1	46	USB0VBUS	U1Tx	-	-	-	-	-	T2CCP1	-	-	-	-	
PB2	47	-	-	-	I2C0SCL	-	-	-	T3CCP0	-	-	-	-	

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

# Board



#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

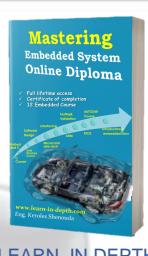
<https://www.facebook.com/groups/embedded.system.KS/>

Table 2-3. J1 Connector<sup>(1)</sup>

J1 Pin	GPIO	Analog Function	On-board Function	Tiva C Series MCU Pin	GPIOPCTL Register Setting										
					1	2	3	4	5	6	7	8	9	14	15
1.01					3.3 V										
1.02	PB5	AIN11	-	57	-	SSI2Fss	-	M0PWM3	-	-	T1CCP1	CAN0Tx	-	-	-
1.03	PB0	USB0ID	-	45	U1Rx	-	-	-	-	-	T2CCP0	-	-	-	
1.04	PB1	USB0VBUS	-	46	U1Tx	-	-	-	-	-	T2CCP1	-	-	-	
1.05	PE4	AIN9	-	59	U5Rx	-	I2C2SCL	M0PWM4	M1PWM2	-	-	CAN0Rx	-	-	-
1.06	PE5	AIN8	-	60	U5Tx	-	I2C2SDA	M0PWM5	M1PWM3	-	-	CAN0Tx	-	-	-
1.07	PB4	AIN10	-	58	-	SSI2Clk	-	M0PWM2	-	-	T1CCP0	CAN0Rx	-	-	-
1.08	PA5	-	-	22	-	SSI0Tx	-	-	-	-	-	-	-	-	-
1.09	PA6	-	-	23	-	-	I2C1SCL	-	M1PWM2	-	-	-	-	-	-
1.10	PA7	-	-	24	-	-	I2C1SDA	-	M1PWM3	-	-	-	-	-	-

<sup>(1)</sup> Shaded cells indicate configuration for compatibility with the MSP430 LaunchPad.

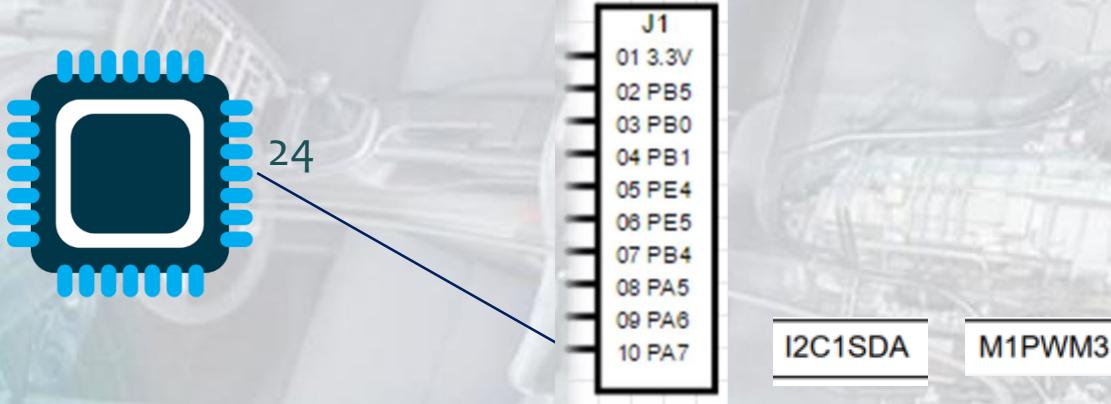
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

https://www.facebook.com/groups/embedded.system.KS/

eng. Keroles Shenouda



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

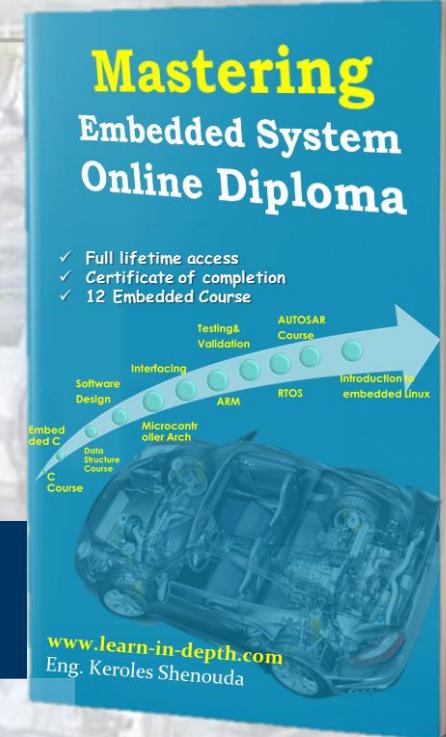


#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>



# Thinking Question

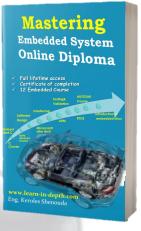
LIST ALL THE ALT. CONNECTIONS WITH PORTA.3 IN BLUEPILL

STM32F103XXX

**LEARN-IN-DEPTH**  
Be professional in  
embedded system



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



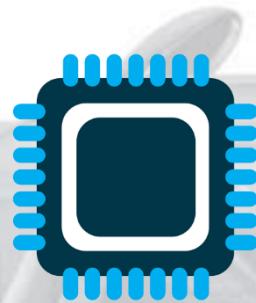
11

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# LQFP48



13

Pin name								Type <sup>(1)</sup>	I / O level <sup>(2)</sup>	Main function <sup>(3)</sup> (after reset)	Default	Remap	
G2	L2	10	G2	14	23	7	PA0-WKUP	I/O	-	PA0	WKUP/ USART2_CTS <sup>(9)</sup> / ADC12_IN0/ TIM2_CH1_ETR <sup>(9)</sup>	-	
H2	M2	11	H2	15	24	8	PA1	I/O	-	PA1	USART2_RTS <sup>(9)</sup> / ADC12_IN1/ TIM2_CH2 <sup>(9)</sup>	-	
J2	K3	12	F3	16	25	9	PA2	I/O	-	PA2	USART2_TX <sup>(9)</sup> / ADC12_IN2/ TIM2_CH3 <sup>(9)</sup>	-	
K2	L3	13	G3	17	26	10	PA3	I/O	-	PA3	USART2_RX <sup>(9)</sup> / ADC12_IN3/ TIM2_CH4 <sup>(9)</sup>	-	

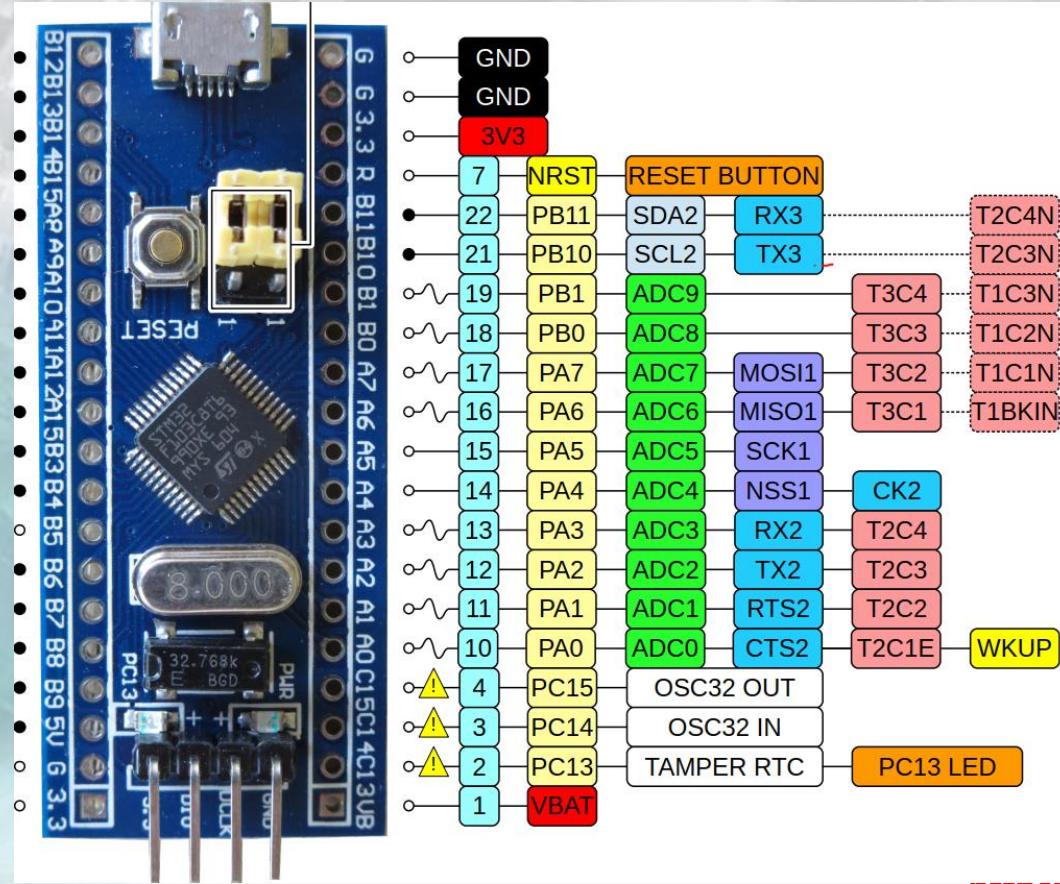
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



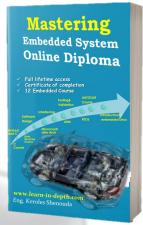
12

## #LEARN\_IN\_DEPTH

### #Be\_professional\_in\_embedded\_system



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

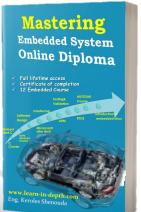


13

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system  
eng. Keroles Shenouda<https://www.facebook.com/groups/embedded.system.KS/>

- 9 General-purpose and alternate-function I/Os (GPIOs and AFIOs)
  - + ■ 9.1 GPIO functional description
  - + ■ 9.2 GPIO registers
  - + ■ 9.3 Alternate function I/O and debug configuration (AFIO)
  - ■ 9.4 AFIO registers
    - 9.4.1 Event control register (AFIO\_EVCR)
    - + ■ 9.4.3 External interrupt configuration register 1 (AFIO\_EXTICR1)
    - 9.4.4 External interrupt configuration register 2 (AFIO\_EXTICR2)
    - 9.4.5 External interrupt configuration register 3 (AFIO\_EXTICR3)
    - 9.4.6 External interrupt configuration register 4 (AFIO\_EXTICR4)
    - 9.4.7 AF remap and debug I/O configuration register2 (AFIO\_MAPR2)
  - ■ 9.5 GPIO and AFIO register maps
    - Table 59. GPIO register map and reset values
    - Table 60. AFIO register map and reset values

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



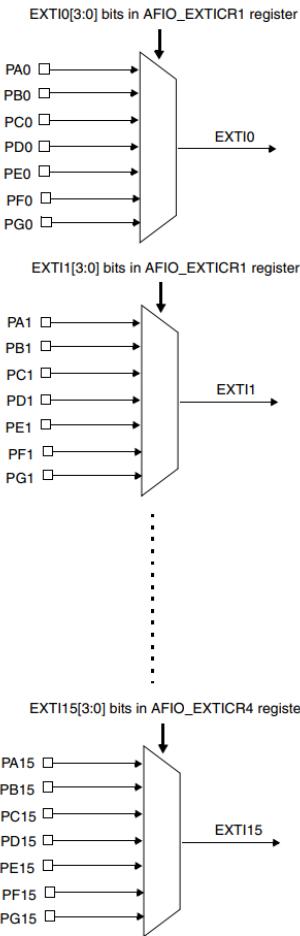
14

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

<https://www.facebook.com/groups/embedded.system.KS/>

# External interrupt/event line mapping

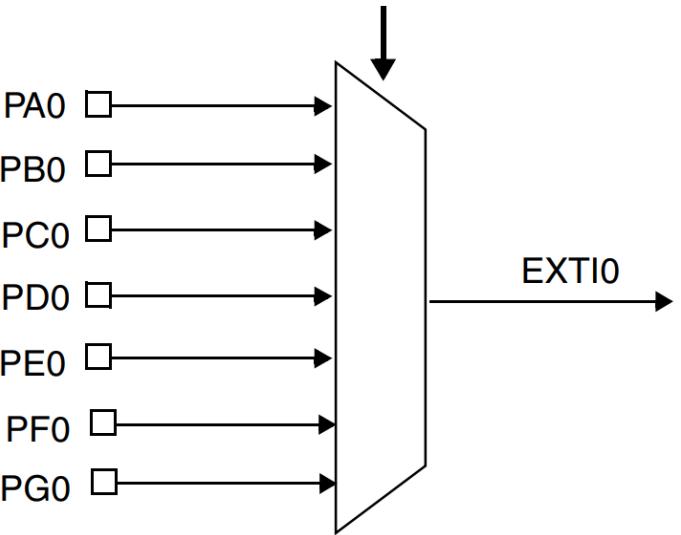
Figure 21. External interrupt/event GPIO mapping



## 10.2.5 External interrupt/event line mapping

The 112 GPIOs are connected to the 16 external interrupt/event lines in the following manner:

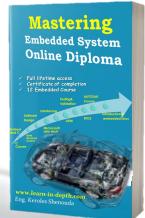
**EXTI0[3:0] bits in AFIO\_EXTICR1 register**



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



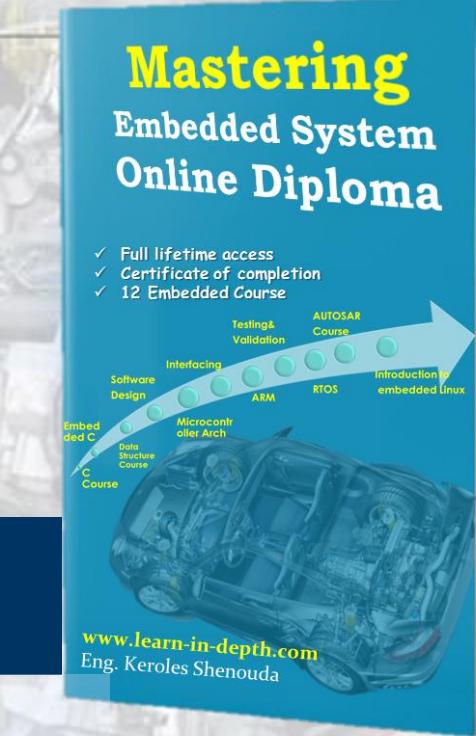
15



#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

#### 9.1.4 **Alternate functions (AF)**

It is necessary to program the Port Bit Configuration Register before using a default alternate function.

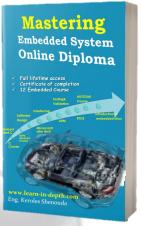
- **For alternate function inputs**, the port must be configured in Input mode (floating, pull-up or pull-down) and the input pin must be driven externally.

*Note:* It is also possible to emulate the AFI input pin by software by programming the GPIO controller. In this case, the port should be configured in **Alternate Function Output mode**. And obviously, the corresponding port should not be driven externally as it will be driven by the software using the GPIO controller.

- **For alternate function outputs**, the port must be configured in Alternate Function Output mode (Push-Pull or Open-Drain).
- **For bidirectional Alternate Functions**, the port bit must be configured in Alternate Function Output mode (Push-Pull or Open-Drain). In this case the input driver is configured in input floating mode

If a port bit is configured as Alternate Function Output, this disconnects the output register and connects the pin to the output signal of an on-chip peripheral.

If software configures a GPIO pin as Alternate Function Output, but peripheral is not activated, its output is not specified.

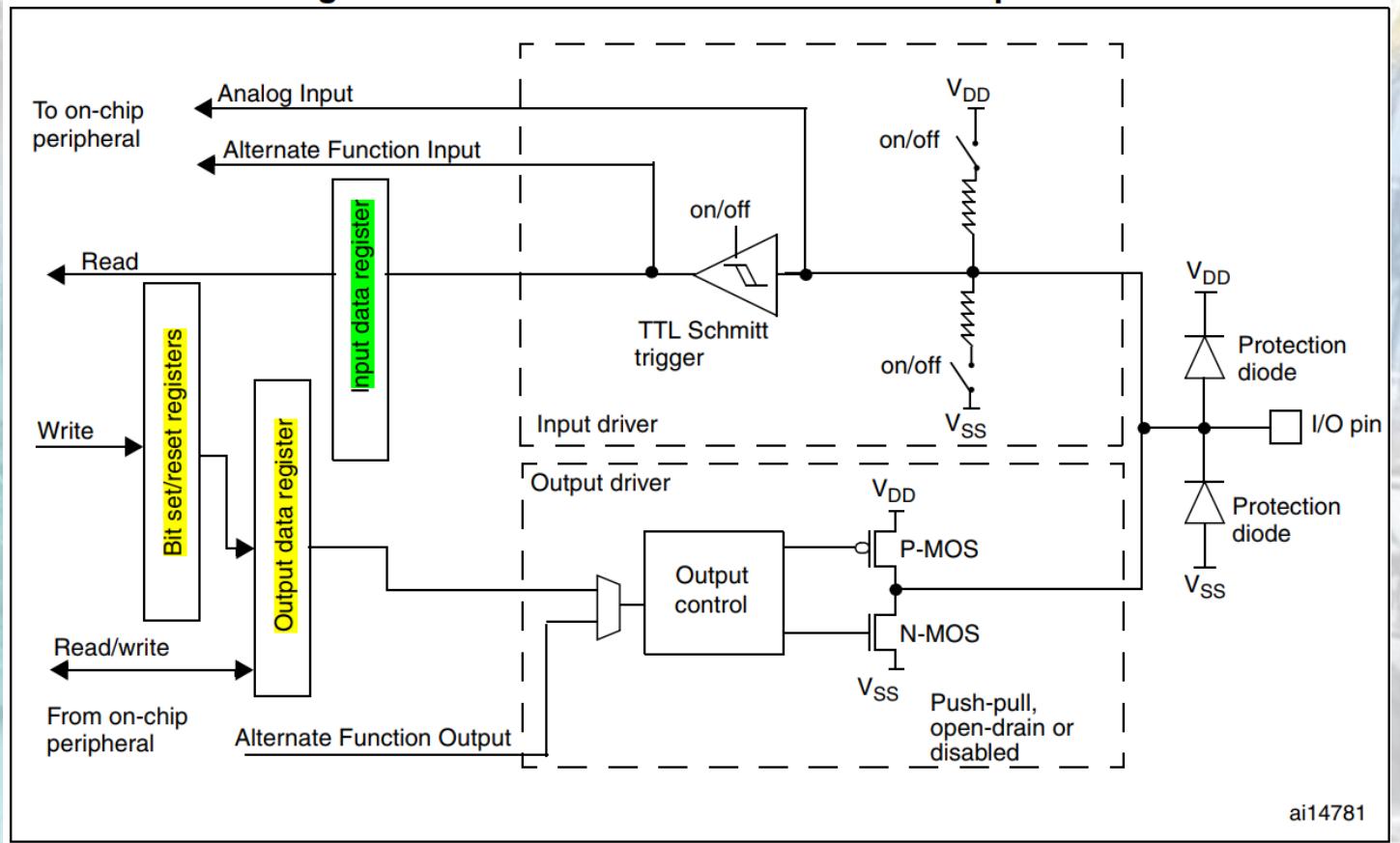


17

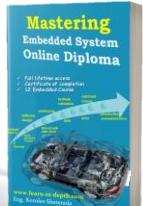
#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

<https://www.facebook.com/groups/embedded.system.KS/>

# Basic structure of a standard I/O port bit



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



18

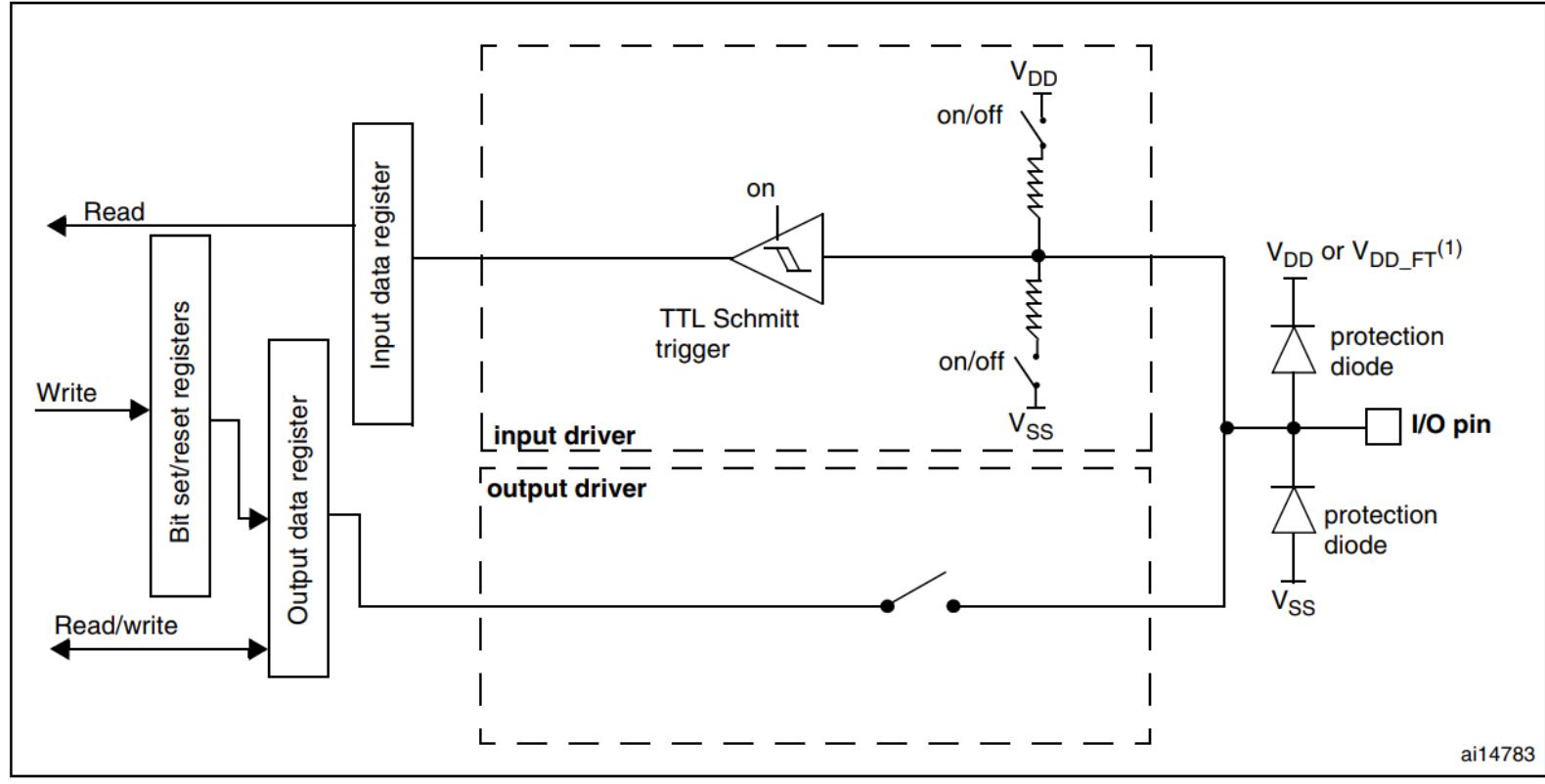
#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Input configuration

Figure 15. Input floating/pull up/pull down configurations



ai14783

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



19

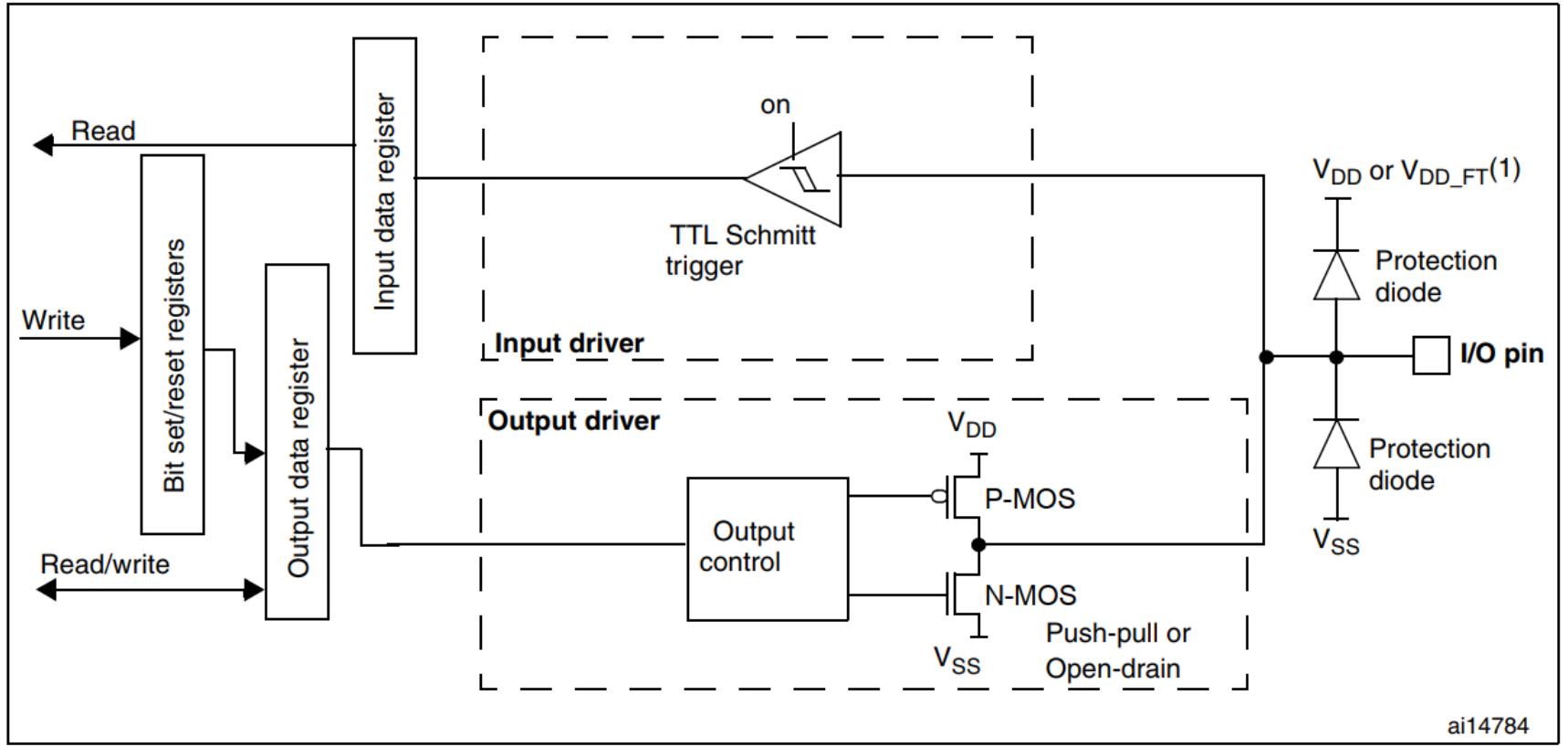
#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Output configuration

Figure 16. Output configuration

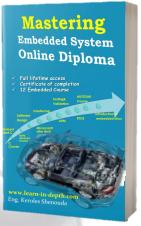


ai14784

<https://www.learn-in-depth.com/><https://www.facebook.com/groups/embedded.system.KS/>



20



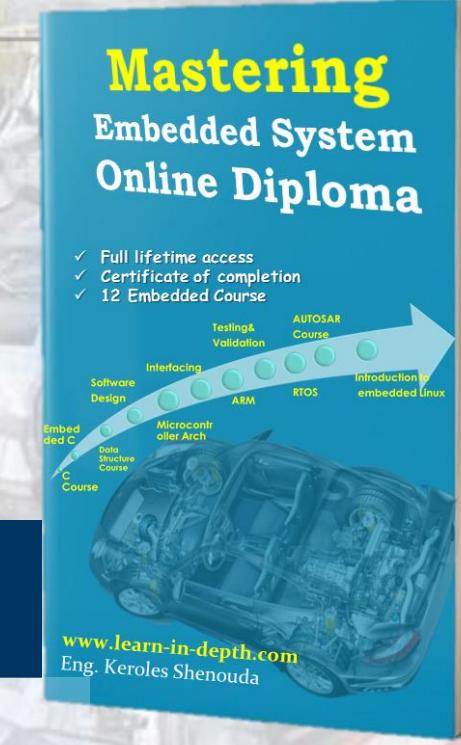
#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Deep dive inside SoC to see PA.0 configured to be external interrupt



**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**



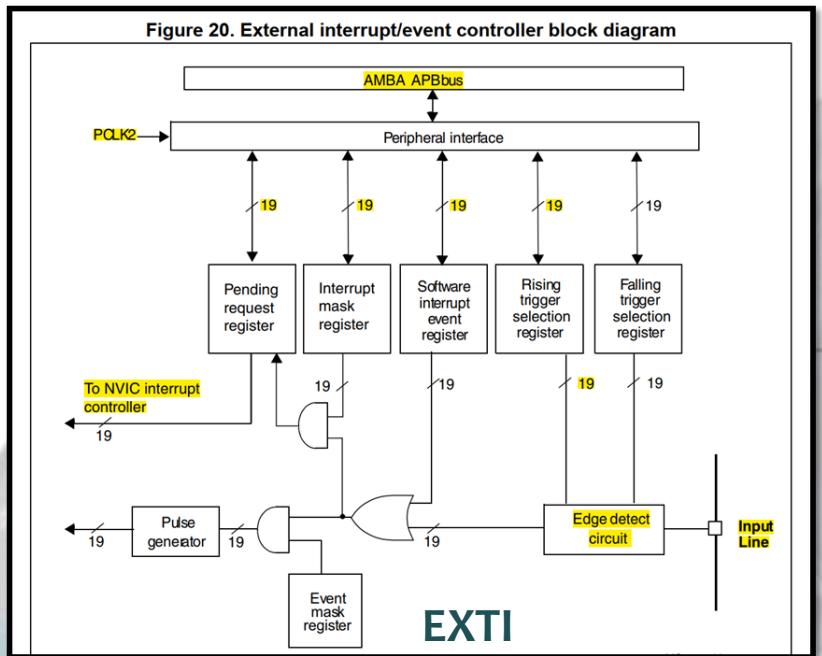
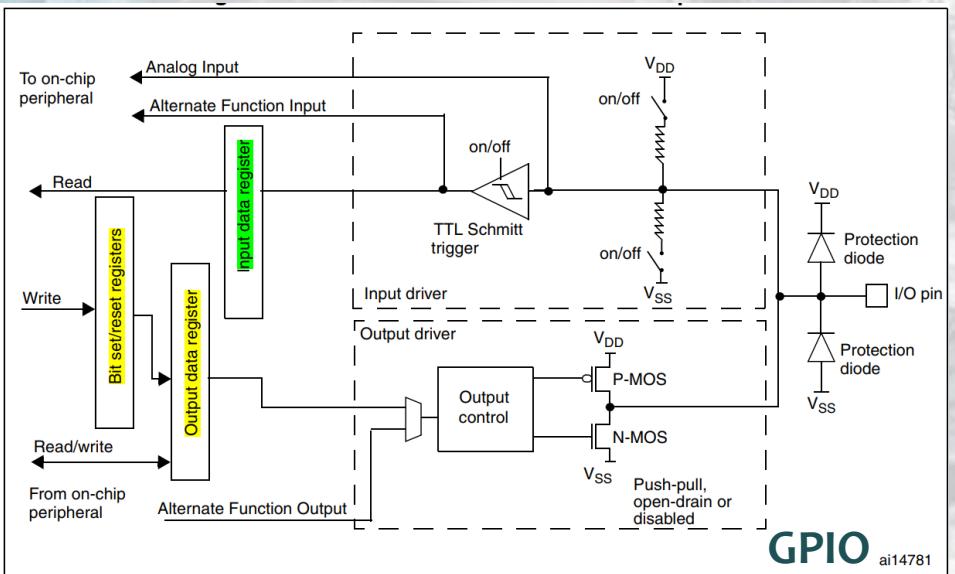
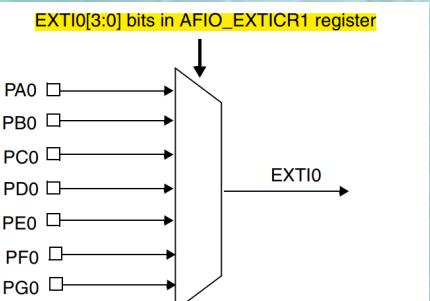
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



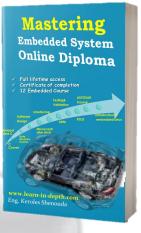
Press here

#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

<https://www.facebook.com/groups/embedded.system.KS/>
**EXTI**

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



22

#LEARN\_IN\_DEPTH  
#Be\_professional\_in  
embedded\_system<https://www.facebook.com/groups/embedded.system.KS/>

```
--  
61 }  
62 GPIOA_Pin0_Input_init()  
63 {  
64     GPIOA_CRL |= (1<<2) ;  
65 }  
66 GPIOA_Pin13_Output_init()  
67 {  
68     GPIOA_CRH    &= 0xFF0FFFFF;  
69     GPIOA_CRH    |= 0x00200000;  
70 }  
71 int main(void)  
72 {  
73  
74     clock_init ();  
75     GPIOA_Pin0_Input_init ();  
76     GPIOA_Pin13_Output_init();  
77  
78     //Select PortA For EXTI0  
79     AFIO_EXTICR1 = 0x0 ;  
80  
81     //Rising trigger enabled  
82     EXTI_RTSR |= (1<<0) ;  
83  
84     //Enable Mask EXTI0  
85     EXTI_IMR |= (1<<0) ;  
86  
87     //Enable NVIC IRQ 6 (EXTI0)  
88     NVIC_ISER0 |= (1<<6) ;  
89  
90     while(1) ;  
91 }  
92 }  
93 void EXTI0_IRQHandler(void)  
94 {  
95     //Toggle Led|  
96     GPIOA_ODR ^= (1<<13) ;  
97     //Clear the EXTI0 Pending Request  
98     EXTI_PR |= (1<<0) ;  
--
```

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

```

61 }
62 GPIOA_Pin0_Input_init()
63 {
64     GPIOA_CRL |= (1<<2) ;
65 }
66 GPIOA_Pin13_Output_init()
67 {
68     GPIOA_CRH &= 0xFF0FFFFF;
69     GPIOA_CRH |= 0x00200000;
70 }
71 int main(void)
72 {
73
74     clock_init ();
75     GPIOA_Pin0_Input_init ();
76     GPIOA_Pin13_Output_init();
77
78 //Select PortA For EXTI0
79 AFIO_EXTICR1 = 0x0 ;
80
81 //Rising trigger enabled
82 EXTI_RTSR |= (1<<0) ;
83
84 //Enable Mask EXTI0
85 EXTI_IMR |= (1<<0) ;
86
87 //Enable NVIC IRQ 6 (EXTI0)
88 NVIC_ISER0 |= (1<<6) ;
89
90 while(1) ;
91
92 }
93 void EXTI0_IRQHandler(void)
94 {
95     //Toggle Led
96     GPIOA_ODR ^= (1<<13) ;
97     //Clear the EXTI0 Pending Request
98     EXTI_PR |= (1<<0) ;

```

## 9.2.1 Port configuration register low (GPIOx\_CRL) (x=A..G)

Address offset: 0x00

Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]	MODE7[1:0]	CNF6[1:0]	MODE6[1:0]	CNF5[1:0]	MODE5[1:0]	CNF4[1:0]	MODE4[1:0]	CNF3[1:0]	MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]	MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]	CNF3[1:0]	MODE3[1:0]	CNF2[1:0]	MODE2[1:0]	CNF1[1:0]	MODE1[1:0]	CNF0[1:0]	MODE0[1:0]
rw	rw														

Bits 31:30, 27:26, CNFy[1:0]: Port x configuration bits (y= 0 .. 7)

23:22, 19:18, 15:14, These bits are written by software to configure the corresponding I/O port.  
11:10, 7:6, 3:2 Refer to [Table 20: Port bit configuration table](#).

### In input mode (MODE[1:0]=00):

00: Analog mode

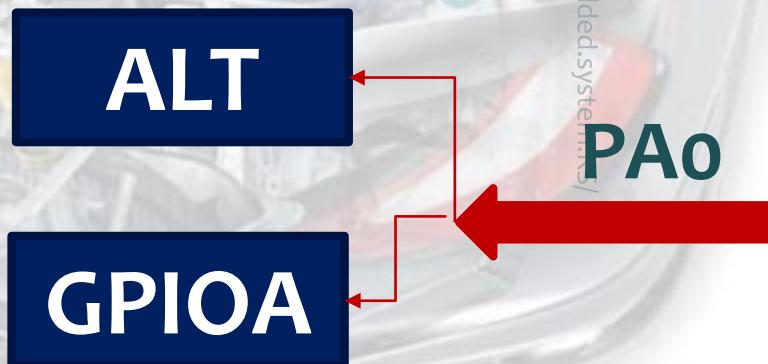
01: Floating input (reset state)

10: Low level (reset state)

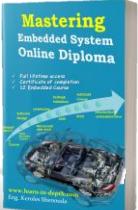
## 9.1.4 Alternate functions (AF)

It is necessary to program the Port Bit Configuration Register before using a default alternate function.

- For alternate function inputs, the port must be configured in Input mode (floating, pull-up or pull-down) and the input pin must be driven externally.



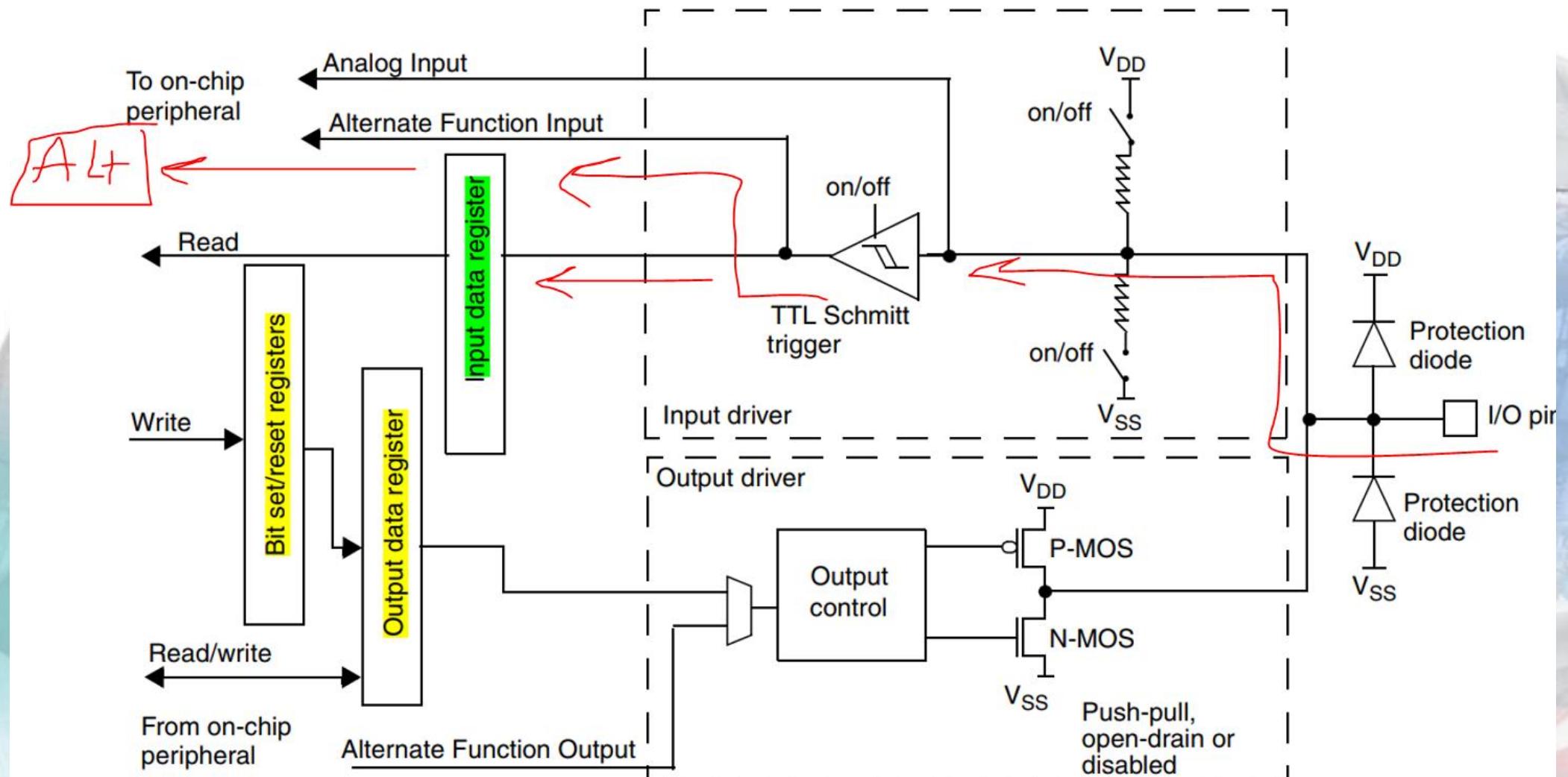
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



24

#LEARN\_IN\_DEPTH  
#Be\_professional\_in  
embedded\_system

<https://www.facebook.com/groups/embedded.system.KS/>



ai14781

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

```

61 }
62 GPIOA_Pin0_Input_init()
63 {
64     GPIOA_CRL |= (1<<2) ;
65 }
66 GPIOA_Pin13_Output_init()
67 {
68     GPIOA_CRH &= 0xFF0FFFFF;
69     GPIOA_CRH |= 0x00200000;
70 }
71 int main(void)
72 {
73     clock_init ();
74     GPIOA_Pin0_Input_init ();
75     GPIOA_Pin13_Output_init();
76
77 //Select PortA For EXTI0
78 AFIO_EXTICR1 = 0x0 ;
79
80 //Rising trigger enabled
81 EXTI_RTSR |= (1<<0) ;
82
83 //Enable Mask EXTI0
84 EXTI_IMR |= (1<<0) ;
85
86 //Enable NVIC IRQ 6 (EXTI0)
87 NVIC_ISER0 |= (1<<6) ;
88
89 while(1) ;
90
91 }
92 void EXTI0_IRQHandler(void)
93 {
94     //Toggle Led
95     GPIOA_ODR ^= (1<<13) ;
96     //Clear the EXTI0 Pending Request
97     EXTI_PR |= (1<<0) ;
98

```

#### 9.4.3 External interrupt configuration register 1 (AFIO\_EXTICR1)

Address offset: 0x08

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTIx[3:0]				EXTIx[3:0]				EXTIx[3:0]				EXTIx[3:0]			
rw	rw	rw	rw												

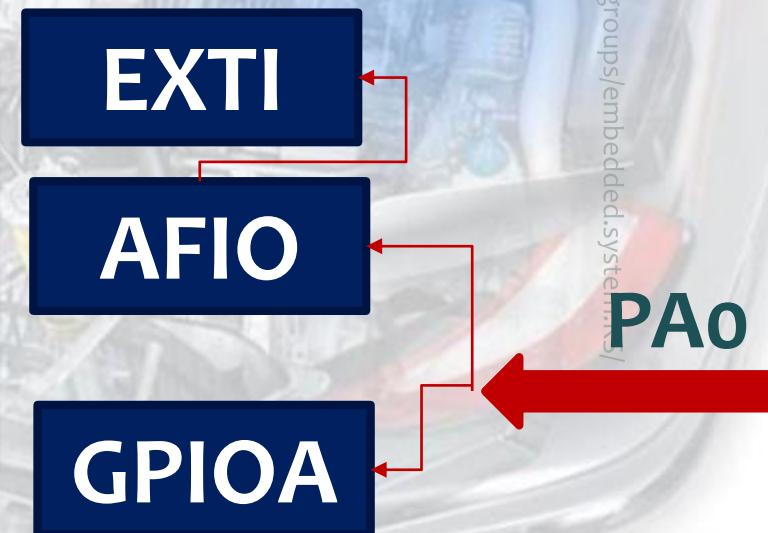
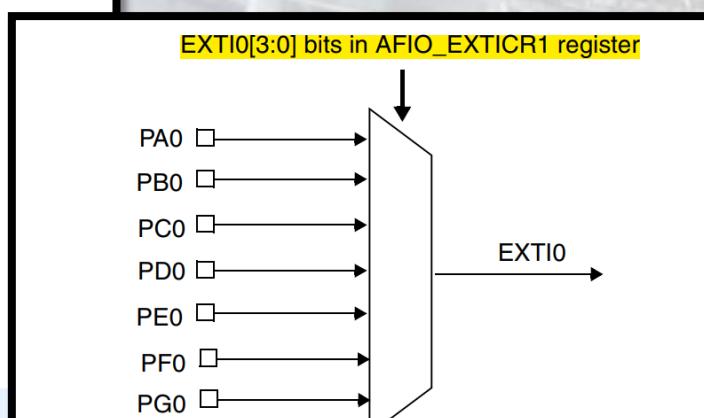
Bits 31:16 Reserved

Bits 15:0 **EXTIx[3:0]**: EXTI x configuration (x= 0 to 3)

These bits are written by software to select the source input for EXTIx external interrupt.

Refer to [Section 10.2.5: External interrupt/event line mapping](#)

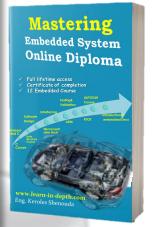
- 0000: PA[x] pin
- 0001: PB[x] pin
- 0010: PC[x] pin
- 0011: PD[x] pin
- 0100: PE[x] pin
- 0101: PF[x] pin
- 0110: PG[x] pin



<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>

Figure 20. External interrupt/event controller block diagram

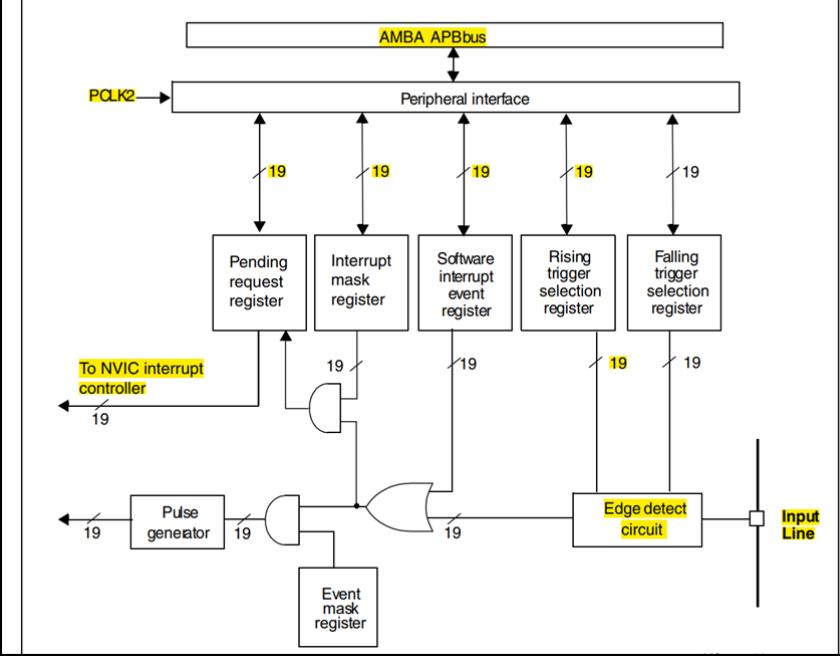


LEARN IN DEPTH  
Be professional in  
embedded system

Eng. Keroles Shenouda

com/groups/embedded.systems/

PAo



```

61 }
62 GPIOA_Pin0_Input_init()
63 {
64     GPIOA_CRL |= (1<<2) ;
65 }
66 GPIOA_Pin13_Output_init()
67 {
68     GPIOA_CRH &= 0xFF0FFFFF;
69     GPIOA_CRH |= 0x00200000;
70 }
71 int main(void)
72 {
73
74     clock_init ();
75     GPIOA_Pin0_Input_init ();
76     GPIOA_Pin13_Output_init();
77
78 //Select PortA For EXTI0
79 AFIO_EXTICR1 = 0x0 ;
80
81 //Rising trigger enabled
82 EXTI_RTSR |= (1<<0) ;
83
84 //Enable Mask EXTI0
85 EXTI_IMR |= (1<<0) ;
86
87 //Enable NVIC IRQ 6 (EXTI0)
88 NVIC_ISER0 |= (1<<6) ;
89
90 while(1) ;
91
92 }
93 void EXTI0_IRQHandler(void)
94 {
95     //Toggle Led
96     GPIOA_ODR ^= (1<<13) ;
97     //Clear the EXTI0 Pending Request
98     EXTI_PR |= (1<<0) ;

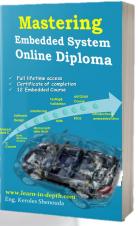
```

EXTI

AFIO

GPIOA

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



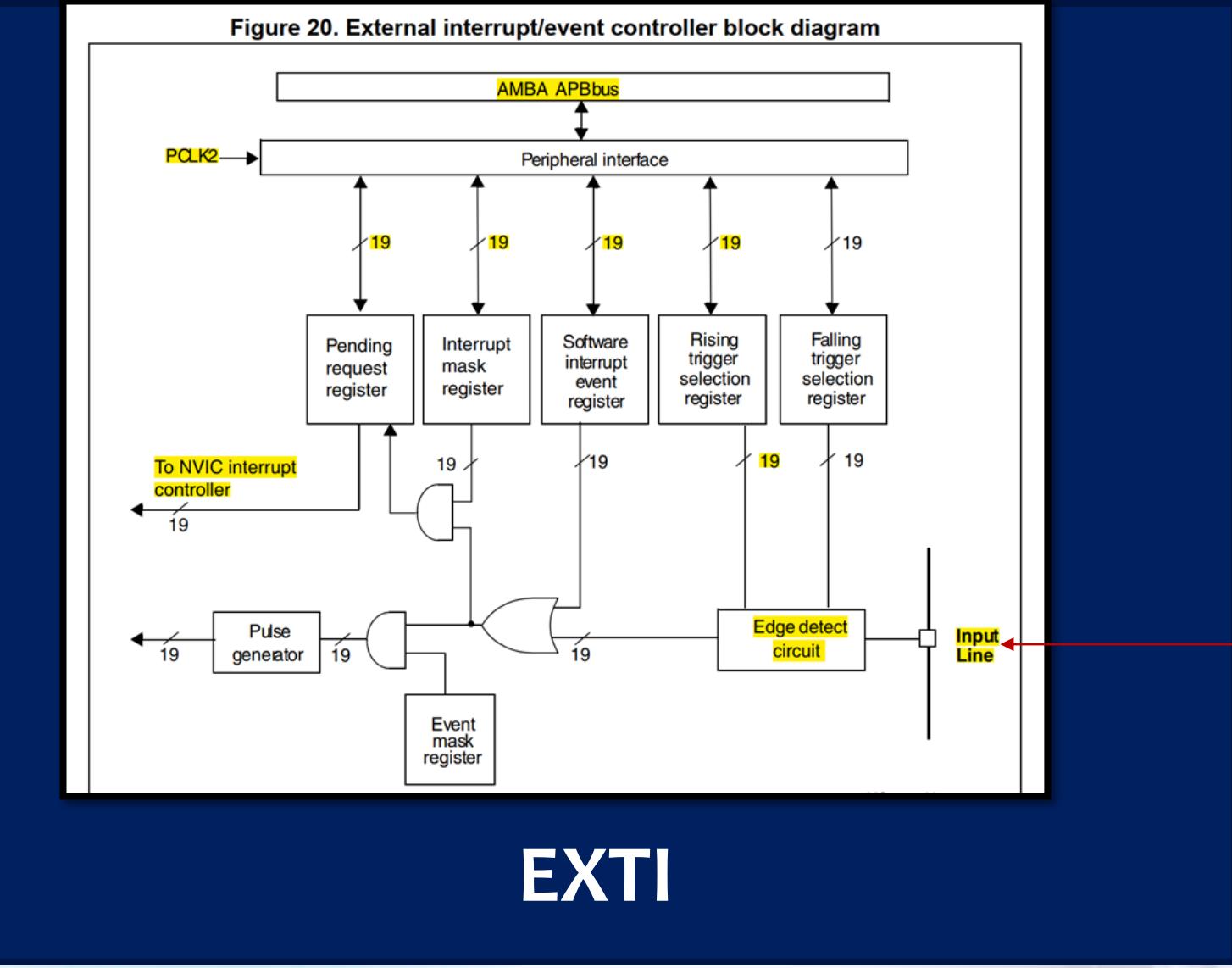
27

#LEARN\_IN\_DEPTH

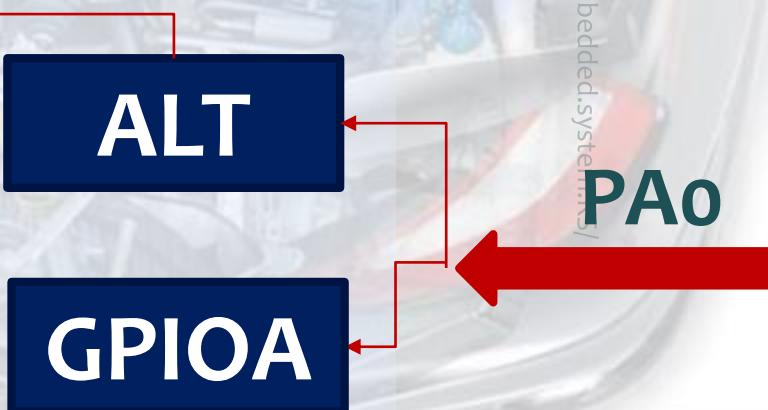
#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.systems/>



EXTI



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

```

61 }
62 GPIOA_Pin0_Input_init()
63 {
64     GPIOA_CRL |= (1<<2) ;
65 }
66 GPIOA_Pin13_Output_init()
67 {
68     GPIOA_CRH &= 0xFF0FFFFF;
69     GPIOA_CRH |= 0x00200000;
70 }
71 int main(void)
72 {
73
74     clock_init ();
75     GPIOA_Pin0_Input_init ();
76     GPIOA_Pin13_Output_init();
77
78     //Select PortA For EXTI0
79     AFIO_EXTICR1 = 0x0 ;
80
81     //Rising trigger enabled
82     EXTI_RTSR |= (1<<0) ;
83
84     //Enable Mask EXTI0
85     EXTI_IMR |= (1<<0) ;
86
87     //Enable NVIC IRQ 6 (EXTI0)
88     NVIC_ISER0 |= (1<<6) ;
89
90     while(1) ;
91
92 }
93 void EXTI0_IRQHandler(void)
94 {
95     //Toggle Led
96     GPIOA_ODR ^= (1<<13) ;
97     //Clear the EXTI0 Pending Request
98     EXTI_PR |= (1<<0) ;

```

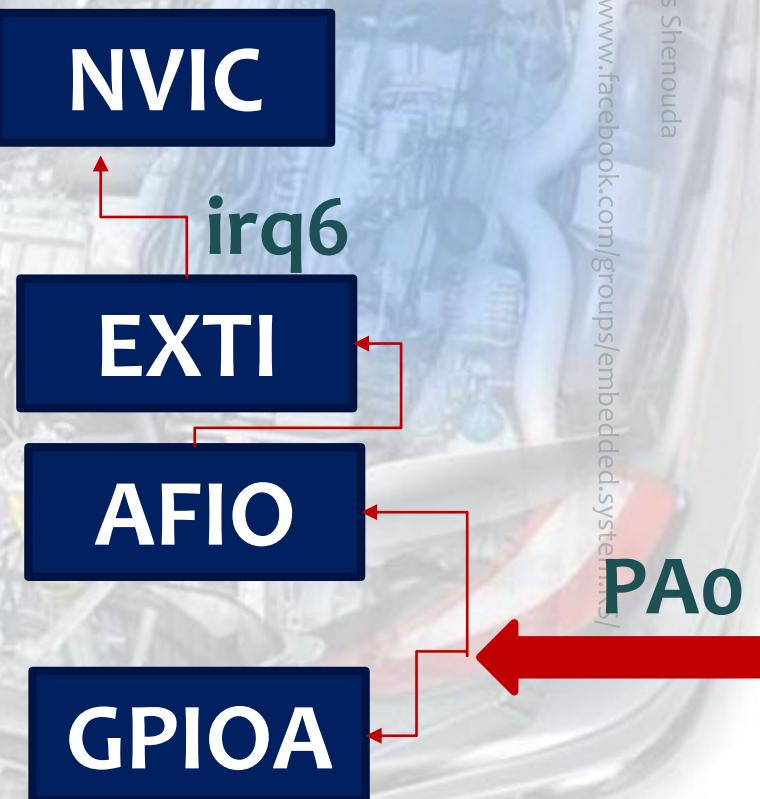
# IVT

3	10	settable	RTC	RTC global interrupt	0x0000_004C
4	11	settable	FLASH	Flash global interrupt	0x0000_0050
5	12	settable	RCC	RCC global interrupt	0x0000_0054
6	13	settable	EXTI0	EXTI Line0 interrupt	0x0000_0058
7	14	settable	EXTI1	EXTI Line1 interrupt	0x0000_005C

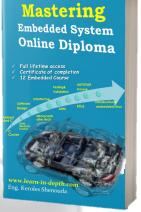
28

https://www.facebook.com/groups/embedded.systems/  
Eng. Keroles Shenouda

#LEARN\_IN\_DEPTH  
#Be\_professional\_in  
embedded\_system



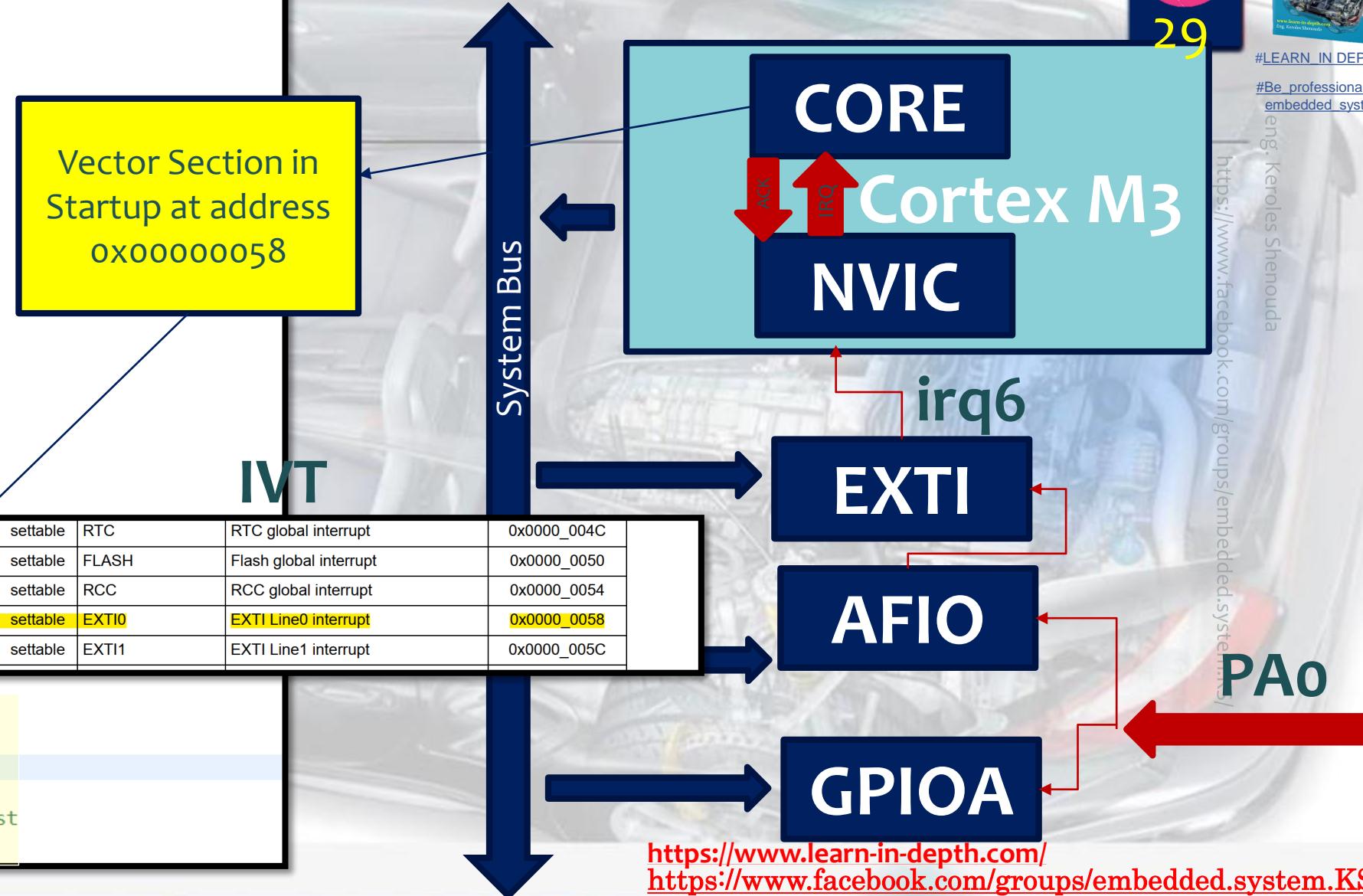
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

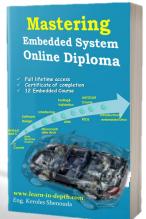


```

61 }
62 GPIOA_Pin0_Input_init()
63 {
64     GPIOA_CRL |= (1<<2) ;
65 }
66 GPIOA_Pin13_Output_init()
67 {
68     GPIOA_CRH &= 0xFF0FFFFF;
69     GPIOA_CRH |= 0x00200000;
70 }
71 int main(void)
72 {
73
74     clock_init ();
75     GPIOA_Pin0_Input_init ();
76     GPIOA_Pin13_Output_init();
77
78     //Select PortA For EXTI0
79     AFIO_EXTICR1 = 0x0 ;
80
81     //Rising trigger enabled
82     EXTI_RTSR |= (1<<0) ;
83
84     //Enable Mask EXTI0
85     EXTI_IMR |= (1<<0) ;
86
87     //Enable NVIC IRQ 6 (
88     NVIC_ISER0 |= (1<<6)
89
90     while(1) ;
91
92 }
93 void EXTI0_IRQHandler(void)
94 {
95     //Toggle Led
96     GPIOA_ODR ^= (1<<13) ;
97     //Clear the EXTI0 Pending Request
98     EXTI_PR |= (1<<0) ;

```



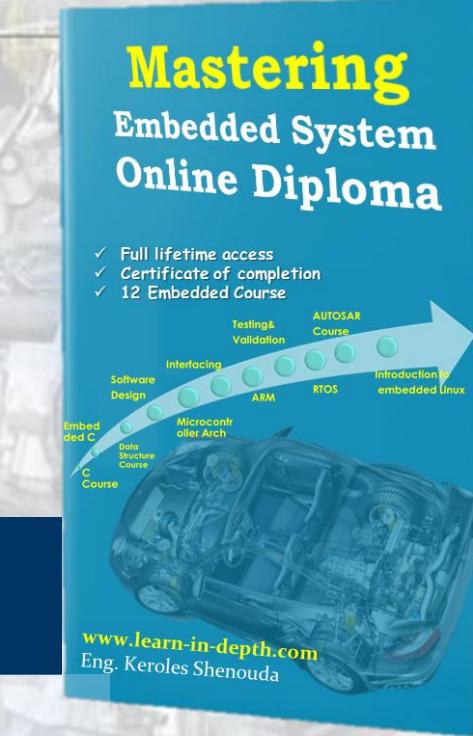


30

#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

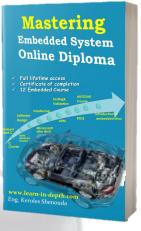
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

# Enable CANTX /RX



31

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

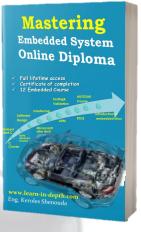
<https://www.facebook.com/groups/embedded.system.KS/>

# According to the Specs

Table 5. Medium-density STM32F103xx pin definitions (continued)

Pins							Pin name	Type <sup>(1)</sup> I/O Level <sup>(2)</sup>	Main function <sup>(3)</sup> (after reset)	Alternate functions <sup>(4)</sup>	
LFBGA100	UFBG100	LQFP48/UQFPN48	TFBGA64	LQFP64	LqFP100	VQFPN36				Default	Remap
C10	B12	32	C8	44	70	23	PA11	I/O FT	PA11	USART1_CTS/ CANRX <sup>(9)</sup> / USBDM/ TIM1_CH4 <sup>(9)</sup>	-
B10	A12	33	B8	45	71	24	PA12	I/O FT	PA12	USART1_RTS/ CANTX <sup>(9)</sup> / /USBDP TIM1_ETR <sup>(9)</sup>	-

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



32

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# The recommended Configuration

**General-purpose and alternate-function I/Os (GPIOs and AFIOs)****RM0008****Table 26. I2S (continued)**

I2S pinout	Configuration	GPIO configuration
I2Sx_MCK	Master	Alternate function push-pull
	Slave	Not used. Can be used as a GPIO

**Table 27. I2C**

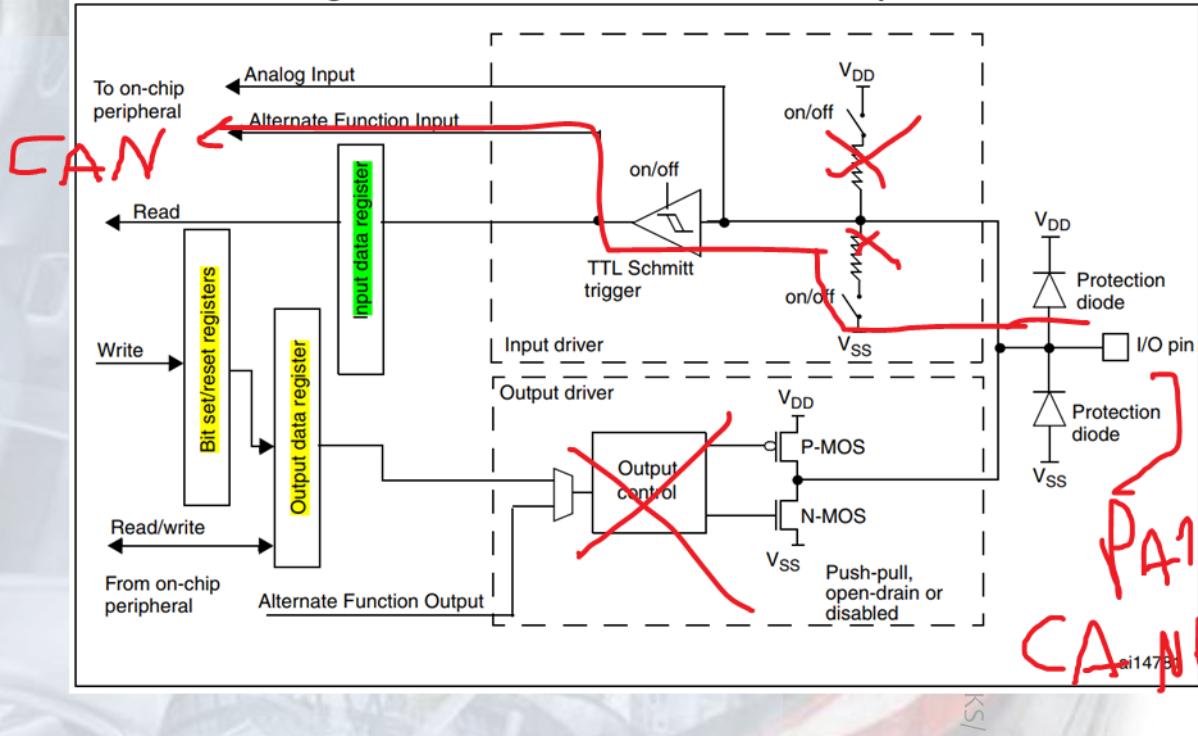
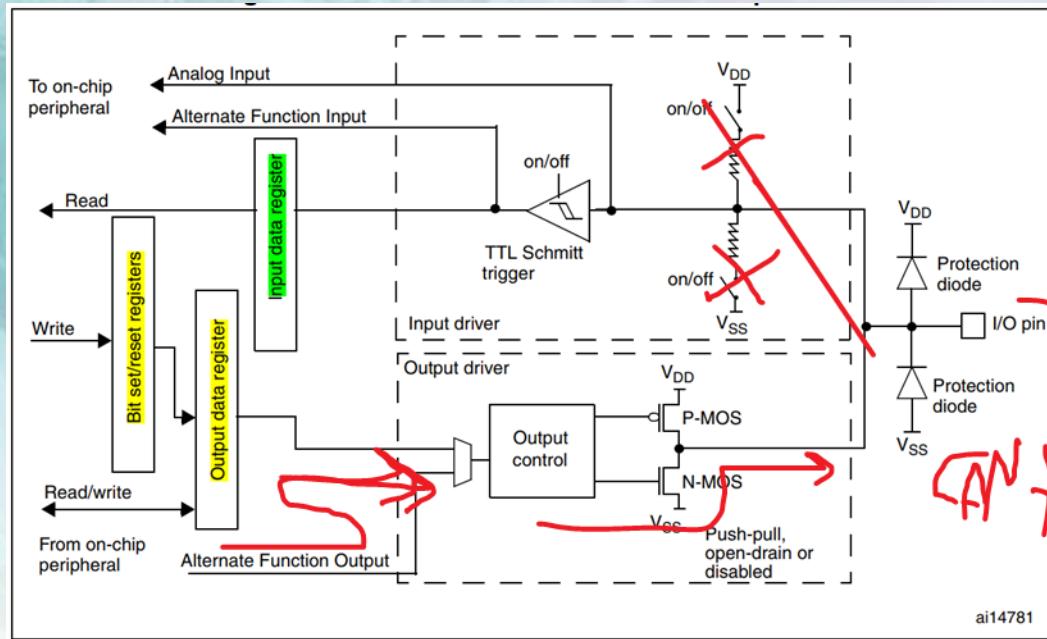
I2C pinout	Configuration	GPIO configuration
I2Cx_SCL	I2C clock	Alternate function open drain
I2Cx_SDA	I2C Data I/O	Alternate function open drain

**Table 28. bxCAN**

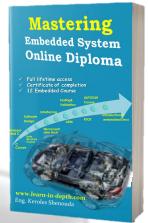
BxCAN pinout	GPIO configuration
CAN_TX (Transmit data line)	Alternate function push-pull
CAN_RX (Receive data line)	Input floating / Input pull-up

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

- ▶ So we will Configure PA11 Input floating
- ▶ PA12 Alternative function push pull
- ▶ Then Enable CAN Controller



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

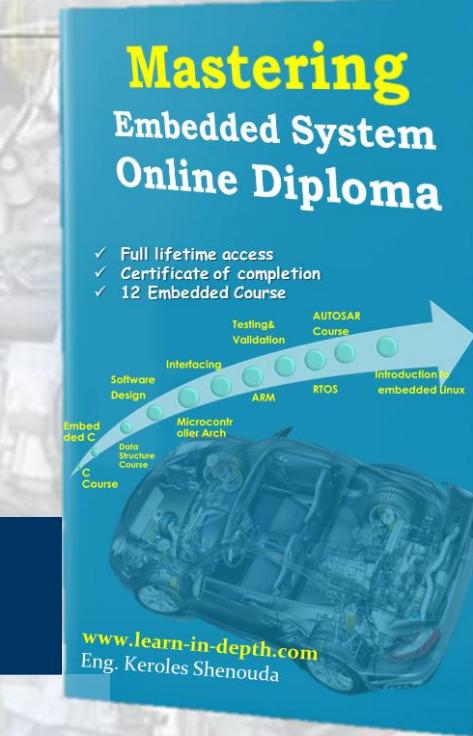


34

#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

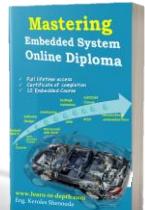
<https://www.facebook.com/groups/embedded.system.KS/>

35

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

Table 60. AFIO register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	PIN[3:0]			
0x00	AFIO_EVCR																																				
	Reset value																																				
0x04	AFIO_MAPR low-, medium-, high- and XL-density devices																																				
	Reset value																																				
0x04	AFIO_MAPR connectivity line devices																																				
	Reset value																																				
0x04	AFIO_MAPR																																				
	Reset value																																				
0x08	AFIO_EXTICR1																																				
	Reset value																																				
0x0C	AFIO_EXTICR2																																				
	Reset value																																				
0x10	AFIO_EXTICR3																																				
	Reset value																																				
0x14	AFIO_EXTICR4																																				
	Reset value																																				
0x1C	AFIO_MAPR2																																				
	Reset value																																				

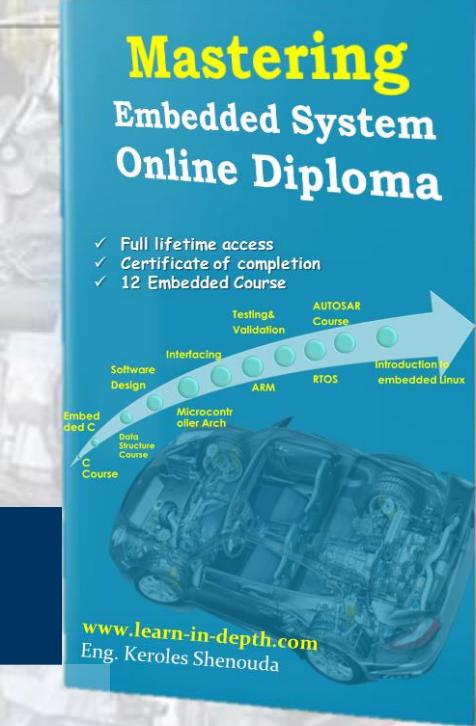


36

#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# EVENTOUT Cortex® output

**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



37

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# EVENTOUT Cortex® output

## 9.4.1 Event control register (AFIO\_EVCR)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								EVOE	PORT[2:0]			PIN[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:8 Reserved

Bit 7 **EVOE**: Event output enable

Set and cleared by software. When set the EVENTOUT Cortex® output is connected to the I/O selected by the PORT[2:0] and PIN[3:0] bits.

Bits 6:4 **PORT[2:0]**: Port selection

Set and cleared by software. Select the port used to output the Cortex® EVENTOUT signal.

*Note: The EVENTOUT signal output capability is not extended to ports PF and PG.*

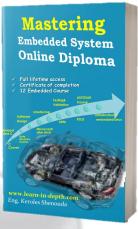
- 000: PA selected
- 001: PB selected
- 010: PC selected
- 011: PD selected
- 100: PE selected

Bits 3:0 **PIN[3:0]**: Pin selection (x = A .. E)

Set and cleared by software. Select the pin used to output the Cortex® EVENTOUT signal.

- 0000: Px0 selected
- 0001: Px1 selected
- 0010: Px2 selected
- 0011: Px3 selected
- ...
- 1111: Px15 selected

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

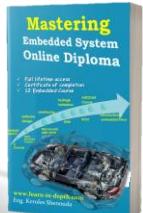
eng. Keroles Shenouda

38

# EVENTOUT Cortex® output

- ▶ it allow the TXEV (transmit event) signal from the Cortex-M3 processor to be output to the port pin.
- ▶ This signal is asserted (pulsed) when the SEV (Send Event) instruction is executed.
- ▶ This can be used to connect to RXEV (Receive Event) signal of other Cortex-M3 processor(s) to wait up from WFE (Wait For Event) sleep.
- ▶ On STM32, you can configure the event input to be connected from EXTI controller via the Event Mask Register (8.3.2), and various other EXTI registers to define the event generation.

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



39

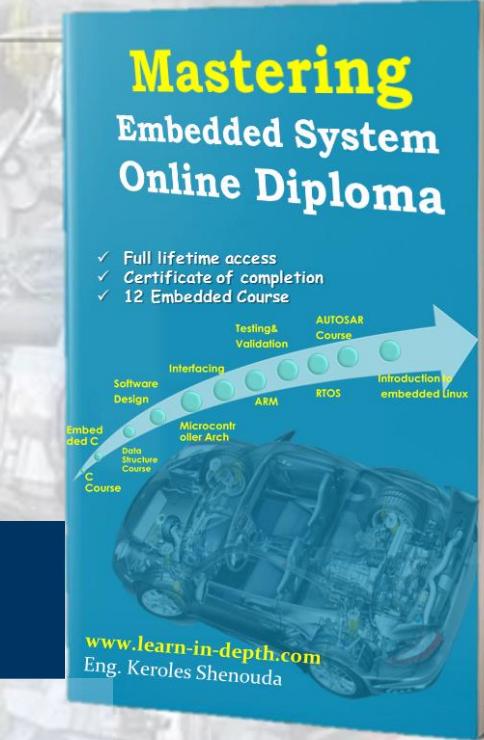
#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

## Software remapping of I/O alternate functions



**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

40

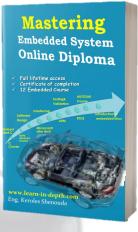
# Software remapping of I/O alternate functions

## 9.1.5

### Software remapping of I/O alternate functions

To optimize the number of peripheral I/O functions for different device packages, it is possible to remap some alternate functions to some other pins. This is achieved by software, by programming the corresponding registers (refer to [AFIO registers](#)). In that case, the alternate functions are no longer mapped to their original assignations.

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



41

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# Software remapping of I/O alternate functions

LFBGA100	UFBG100	LQFP48/UQFPN48	TFBGA64	LQFP64	LQFP100	VFQFPN36	Pin name	Type <sup>(1)</sup> I/O Level <sup>(2)</sup>	Main function <sup>(3)</sup> (after reset)	Default	Remap
G5	L6	20	G6	28	37	17	PB2	I/O FT	PB2/BOOT1	-	-
H5	M7	-	-	-	38	-	PE7	I/O FT	PE7	-	TIM1_ETR
J5	L7	-	-	-	39	-	PE8	I/O FT	PE8	-	TIM1_CH1N
K5	M8	-	-	-	40	-	PE9	I/O FT	PE9	-	TIM1_CH1
G6	L8	-	-	-	41	-	PE10	I/O FT	PE10	-	TIM1_CH2N
H6	M9	-	-	-	42	-	PE11	I/O FT	PE11	-	TIM1_CH2
J6	L9	-	-	-	43	-	PE12	I/O FT	PE12	-	TIM1_CH3N

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

#### 9.4.2 AF remap and debug I/O configuration register (AFIO\_MAPR)

Address offset: 0x04

Reset value: 0x0000 0000

**Memory map and bit definitions for low-, medium- high- and XL-density devices:**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				SWJ_CFG[2:0]			Reserved				ADC2_E TRGREG _REMAP	ADC2_E TRGINJ_ REMAP	ADC1_E TRGREG _REMAP	ADC1_E TRGINJ_ REMAP	TIM5CH4 _IREMAP
				w	w	w					rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD01_ REMAP	CAN_REMAP [1:0]		TIM4_ REMAP	TIM3_REMAP [1:0]		TIM2_ REMAP [1:0]	TIM1_ REMAP [1:0]		USART3_ REMAP[1:0]	USART2_ REMAP	USART1_ REMAP	I2C1_ REMAP	SPI1_ REMAP		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

LFBGA100	UFBG100	LQFP48/UFBQFPN	TFBGA64	LQFP64	LQFP100	VFQFN36	Pin name	Type <sup>(1)</sup> I/O Level <sup>(2)</sup>	Main function <sup>(3)</sup> (after reset)	Default	Remap
G5	L6	20	G6	28	37	17	PB2	I/O FT	PB2/BOOT1	-	-
H5	M7	-	-	-	38	-	PE7	I/O FT	PE7	-	TIM1_ETR
J5	L7	-	-	-	39	-	PE8	I/O FT	PE8	-	TIM1_CH1N
K5	M8	-	-	-	40	-	PE9	I/O FT	PE9	-	TIM1_CH1

#### Bits 7:6 TIM1\_REMAP[1:0]: TIM1 remapping

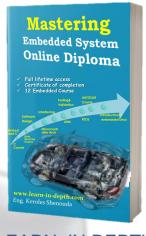
These bits are set and cleared by software. They control the mapping of TIM1 channels 1 to 4, 1N to 3N, external trigger (ETR) and Break input (BKIN) on the GPIO ports.

00: No remap (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PB12, CH1N/PB13, CH2N/PB14, CH3N/PB15)

01: Partial remap (ETR/PA12, CH1/PA8, CH2/PA9, CH3/PA10, CH4/PA11, BKIN/PA6, CH1N/PA7, CH2N/PB0, CH3N/PB1)

10: not used

11: Full remap (ETR/PE7, CH1/PE9, CH2/PE11, CH3/PE13, CH4/PE14, BKIN/PE15, CH1N/PE8, CH2N/PE10, CH3N/PE12)



43

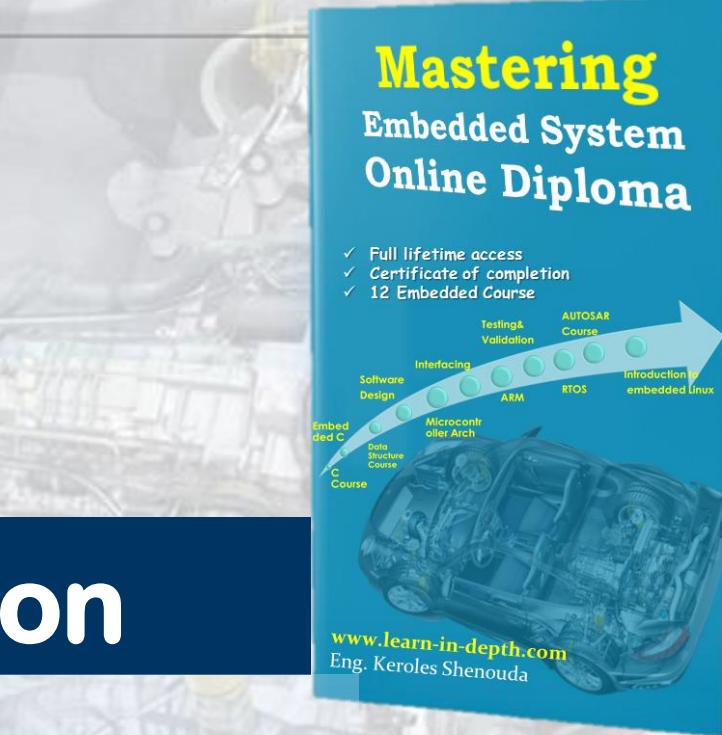
#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# External interrupt configuration



**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



44

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

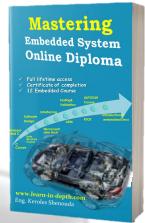
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# External interrupt configuration

		Reserved	EXTI3[3:0]	EXTI2[3:0]	EXTI1[3:0]	EXTI0[3:0]
0x08	AFIO_EXTICR1		0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0x0C	AFIO_EXTICR2	Reserved	EXTI7[3:0]	EXTI6[3:0]	EXTI5[3:0]	EXTI4[3:0]
			0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0x10	AFIO_EXTICR3	Reserved	EXTI11[3:0]	EXTI10[3:0]	EXTI9[3:0]	EXTI8[3:0]
			0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0x14	AFIO_EXTICR4	Reserved	EXTI15[3:0]	EXTI14[3:0]	EXTI13[3:0]	EXTI12[3:0]
			0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



45

#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

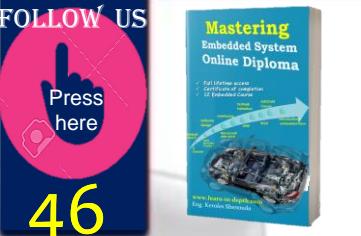
# Driver Development Sequence



**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



Press here

46

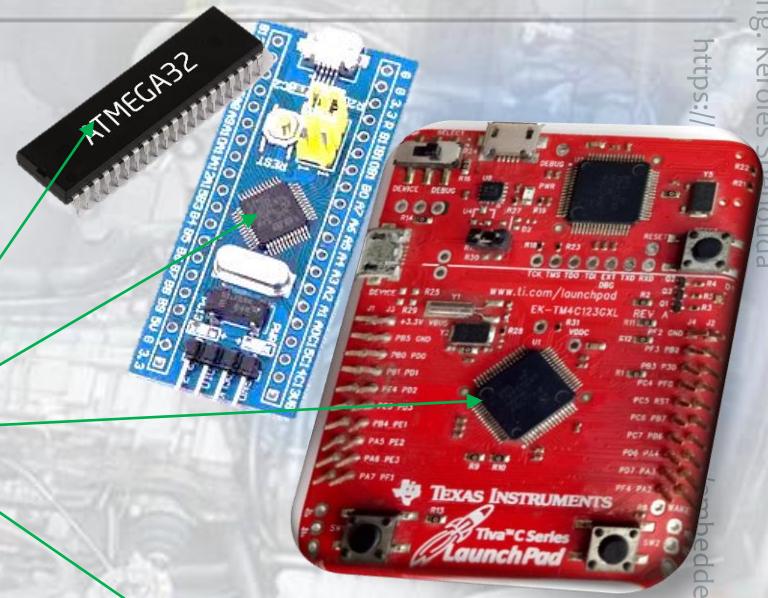
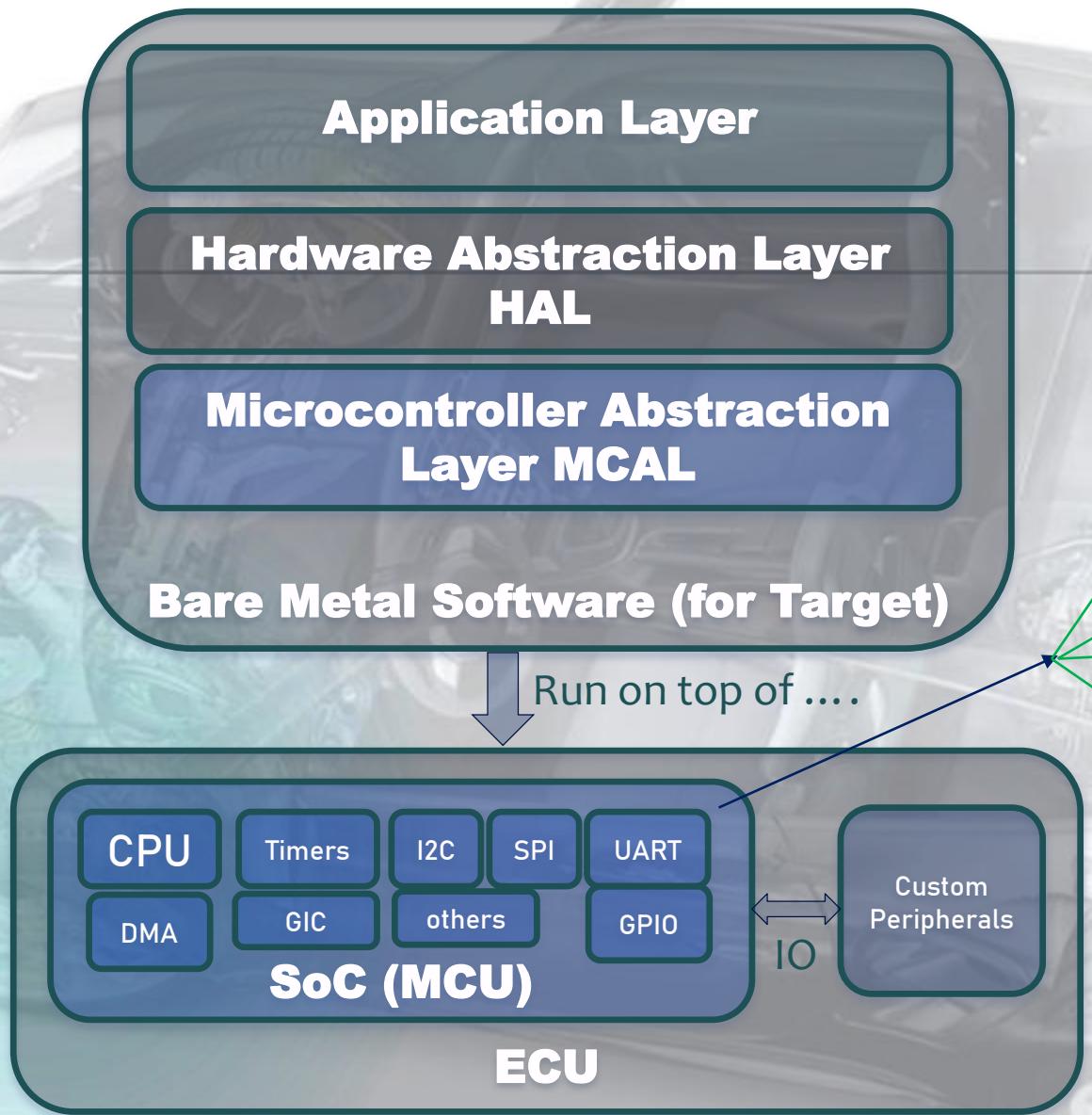
#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.learn-in-depth.com/>

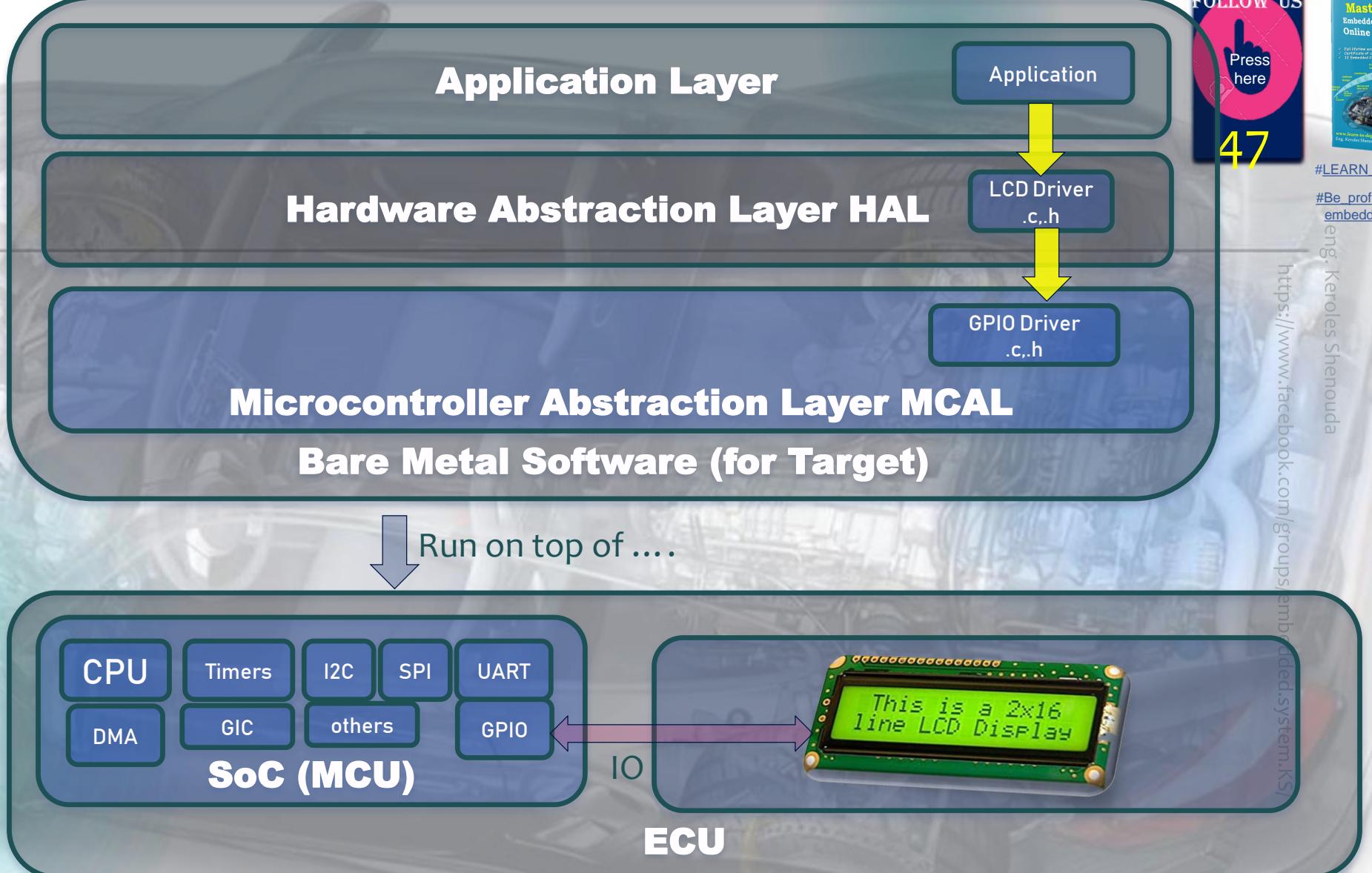
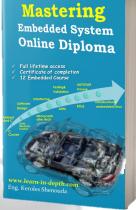
# SW Layers (Generic Shape)



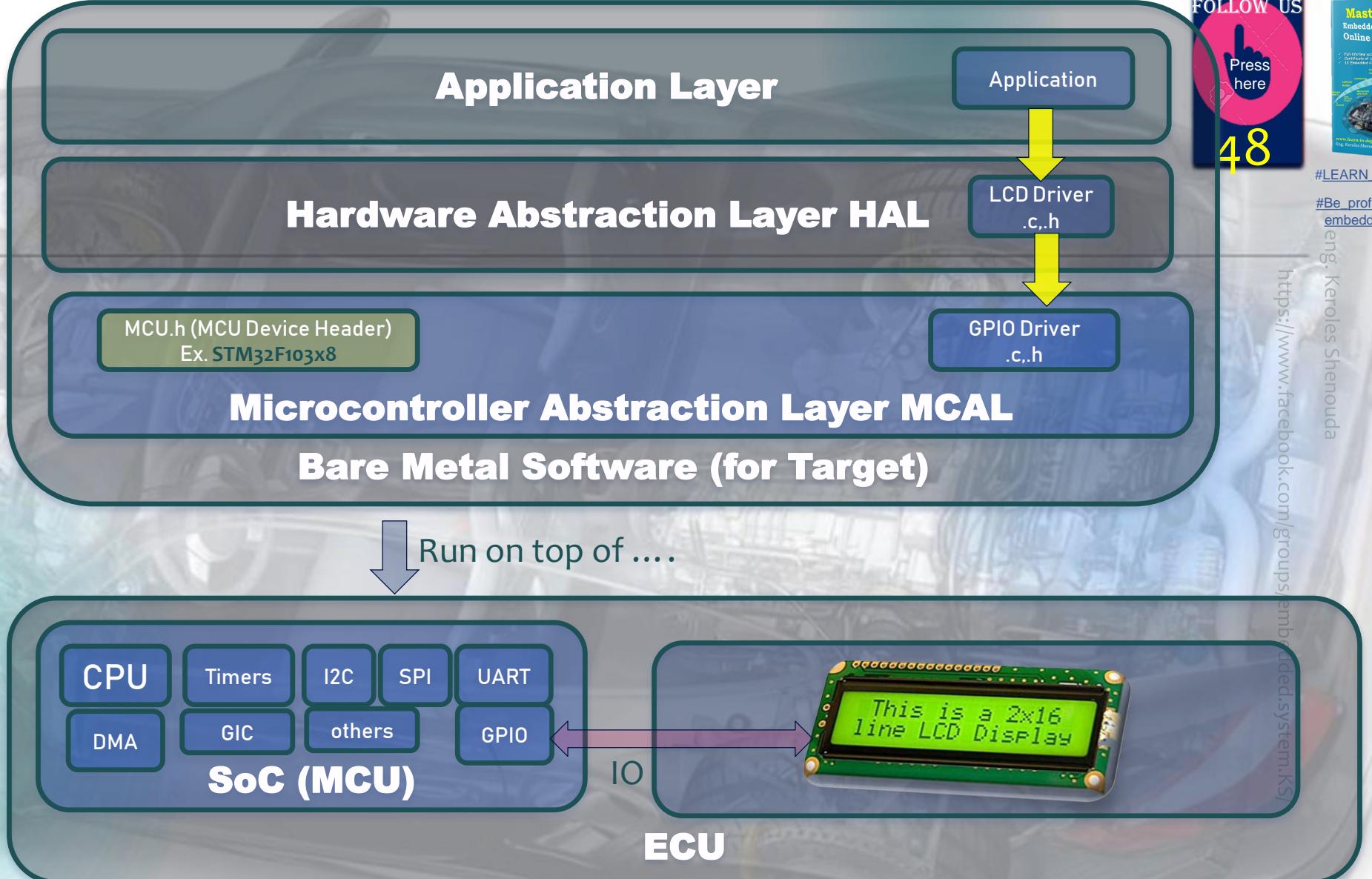
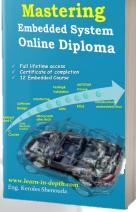
**X - SoC**

Read Specs and Program X - SoC

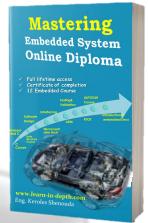
<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



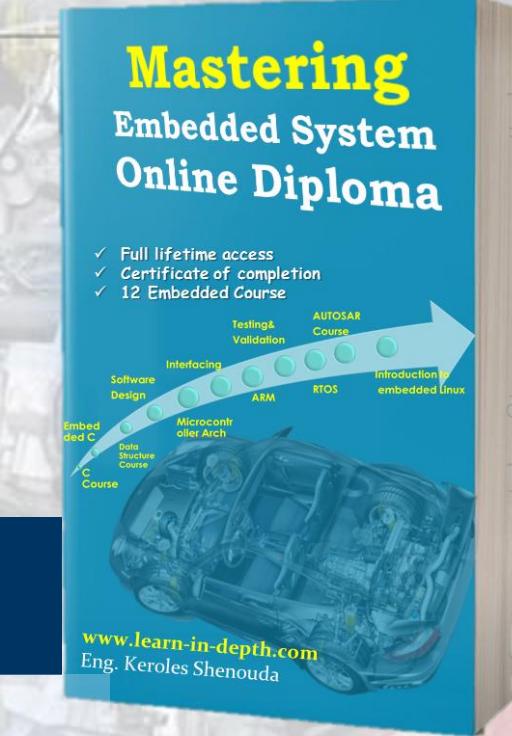
49

#LEARN\_IN\_DEPTH

#Be\_professional\_in  
embedded\_system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

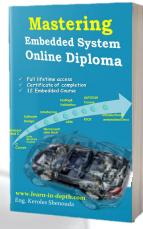


# MCU Device Header

**LEARN-IN-DEPTH**  
Be professional in  
**embedded system**



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



50

#LEARN\_IN\_DEPTH  
#Be\_professional\_in\_embedded\_system

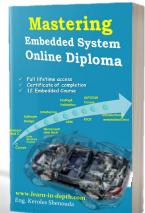
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

# MCU Device Header

- ▶ It is specific for each SoC/MCU
- ▶ It contains
  - ▶ Base address for each Module inside SoC
  - ▶ Clock Management Macros
  - ▶ IRQ Definitions
  - ▶ Peripheral Registers definition structure
  - ▶ Other useful Microcontroller configuration macros

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



62

#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

# References

- ▶ <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZmtlLnVObS5teXxyaWR6dWFuLXMtd2Vic2I0ZXxneDo2ODU0NzIKM2JkOTg4MjRk>
- ▶ <http://www.avrprojects.net/index.php/avr-projects/sensors/38-humidity-and-temperature-sensor-dht11?showall=&start=1>
- ▶ <http://www.cse.wustl.edu/~lu/cse467s/slides/dsp.pdf>
- ▶ <http://www.avr-tutorials.com/>
- ▶ Microprocessor: ATmega32 (SEE3223-10)  
<http://ridzuan.fke.utm.my/microprocessor-atmega32-see3223-10>
- ▶ <http://circuitdigest.com/article/what-is-the-difference-between-microprocessor-and-microcontroller>
- ▶ [http://cs4hs.cs.pub.ro/wiki/roboticsisfun/chapter2/ch2\\_7\\_programming\\_a\\_microcontroller](http://cs4hs.cs.pub.ro/wiki/roboticsisfun/chapter2/ch2_7_programming_a_microcontroller)
- ▶ Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C Dr. Yifeng Zhu Third edition June 2018

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

# References

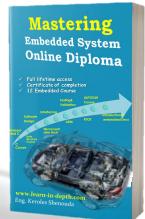
- ▶ <http://techdifferences.com/difference-between-interrupt-and-polling-in-os.html>
- ▶ [http://www.bogotobogo.com/Embedded/hardware\\_interrupt\\_software\\_interrupt\\_latency\\_irq\\_vs\\_fiq.php](http://www.bogotobogo.com/Embedded/hardware_interrupt_software_interrupt_latency_irq_vs_fiq.php)
- ▶ Preventing Interrupt Overload Presented by Jiyong Park Seoul National University, Korea 2005. 2. 22. John Regehr, Usit Duogsaa, School of Computing, University.
- ▶ First Steps Embedded Systems Byte Craft Limited reference
- ▶ COMPUTER ORGANIZATION AND ARCHITECTURE DESIGNING FOR PERFORMANCE EIGHTH EDITION William Stallings
- ▶ Getting Started with the Tiva™ TM4C123G LaunchPad Workshop

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>

# References

- ▶ Tiva™ TM4C123GH6PM Microcontroller DATA SHEET
- ▶ Interrupts and Exceptions COMS W6998 Spring 2010
- ▶ THE AVR MICROCONTROLLER. AND EMBEDDED SYSTEMS Using Assembly and C. Muhammad Ali Mazidi.

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



65

#LEARN\_IN\_DEPTH

#Be\_professional\_in\_embedded\_system

# References

- ▶ <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZmtILnV0bS5teXxyaWR6dWFuLXMfd2Vic2I0ZXxneDo2ODU0Nzlkm2JkOTg4MjRk>
- ▶ <http://www.avrprojects.net/index.php/avr-projects/sensors/38-humidity-and-temperature-sensor-dht11?showall=&start=1>
- ▶ <http://www.cse.wustl.edu/~lu/cse467s/slides/dsp.pdf>
- ▶ <http://www.avr-tutorials.com/>
- ▶ Microprocessor: ATmega32 (SEE3223-10)  
<http://ridzuan.fke.utm.my/microprocessor-atmega32-see3223-10>
- ▶ <http://circuitdigest.com/article/what-is-the-difference-between-microprocessor-and-microcontroller>
- ▶ AVR Microcontroller and Embedded Systems: Using Assembly and C (Pearson Custom Electronics Technology) 1st Edition  
<https://www.amazon.com/AVR-Microcontroller-Embedded-Systems-Electronics/dp/0138003319>

<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.learn-in-depth.com/>  
<https://www.facebook.com/groups/embedded.system.KS/>