

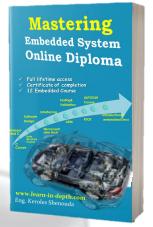
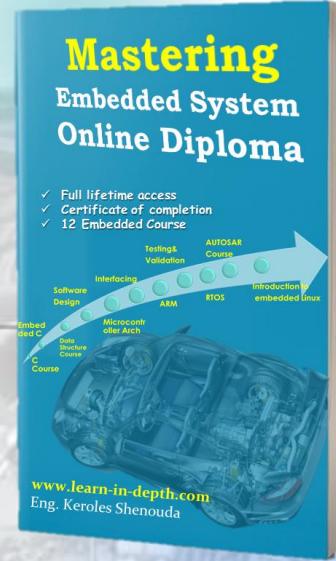
Mastering Embedded System

Online Diploma

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

Unit 7 (MCU Essential Peripherals) . lesson 3 GPIO Part3

- ▶ Big Picture
- ▶ Driver Development Sequence
- ▶ MCU Device Header
 - ▶ File LAYOUT
 - ▶ Specify Peripheral register by C Structure
 - ▶ Memory Address MAPE
- ▶ HOW to Create a Driver
 - ▶ Peripheral functional description
 - ▶ Driver APIs
 - ▶ Files LAYOUT



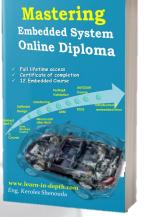
https://www.facebook.com/groups/embedded.system.KS/

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda



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



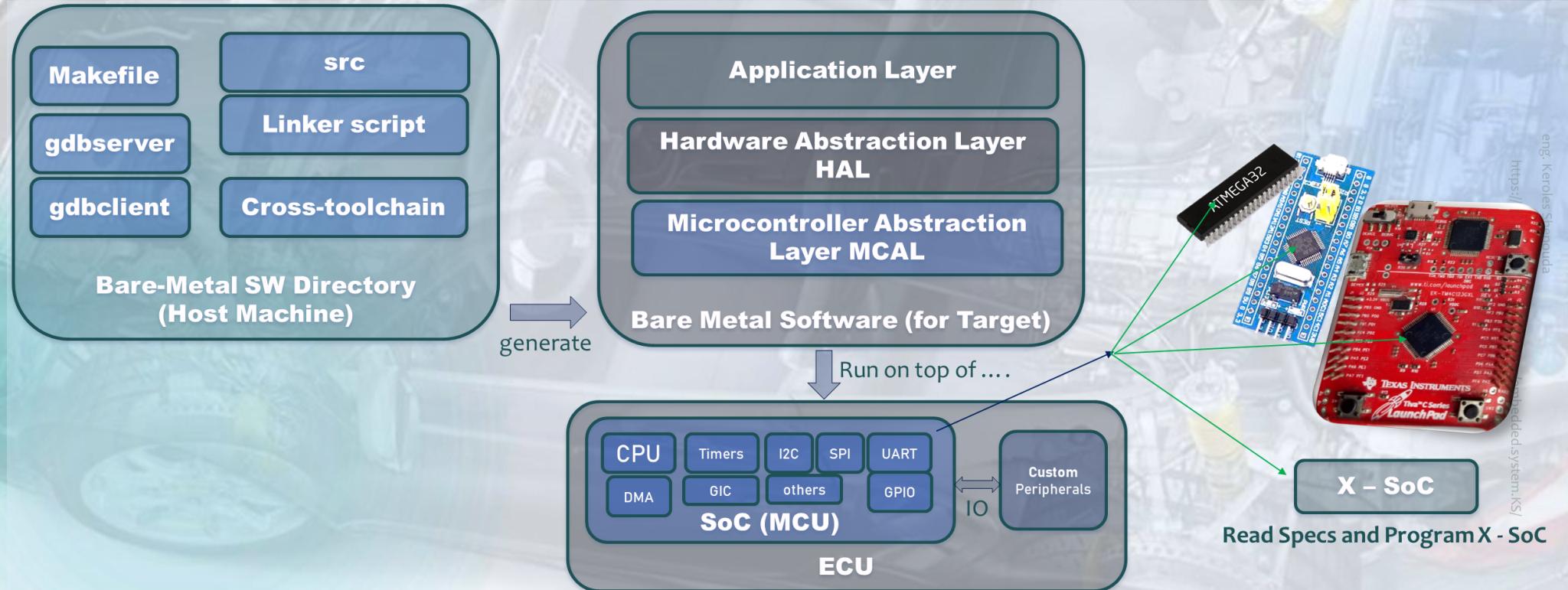
<https://www.facebook.com/groups/embedded.system.KS/>

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

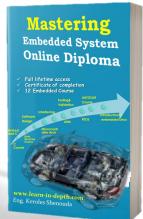
eng_Keroles Shenouda

2

Big Picture



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



3

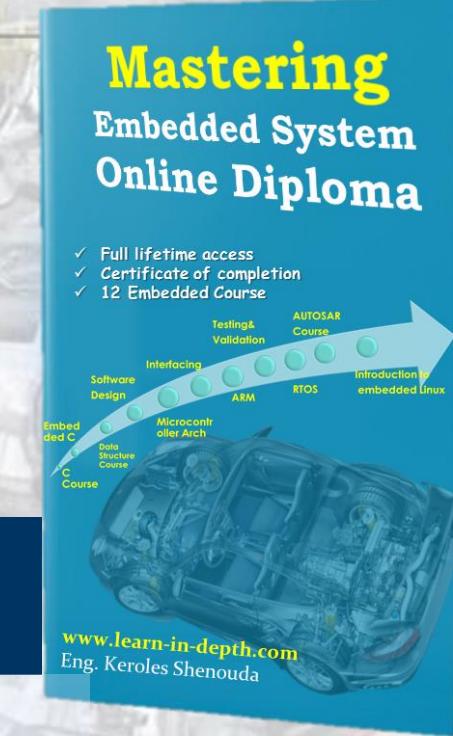
#LEARN_IN_DEPTH

#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

Driver Development Sequence

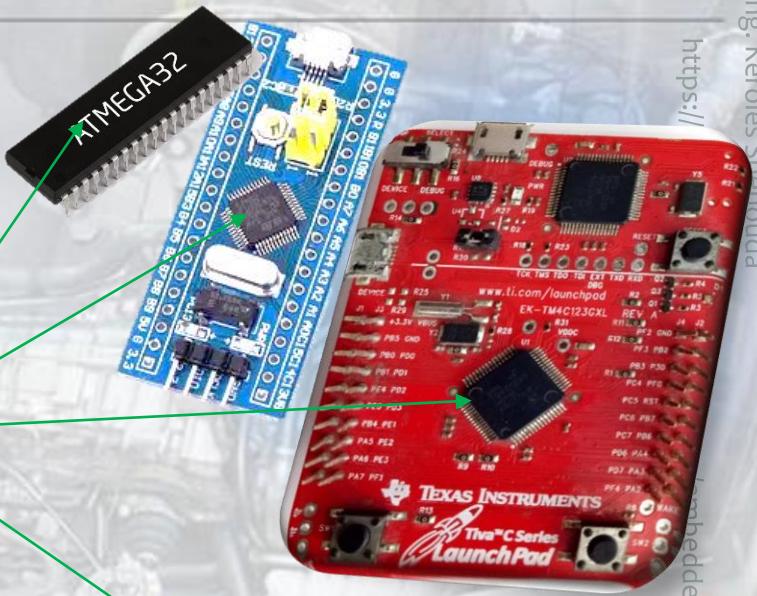
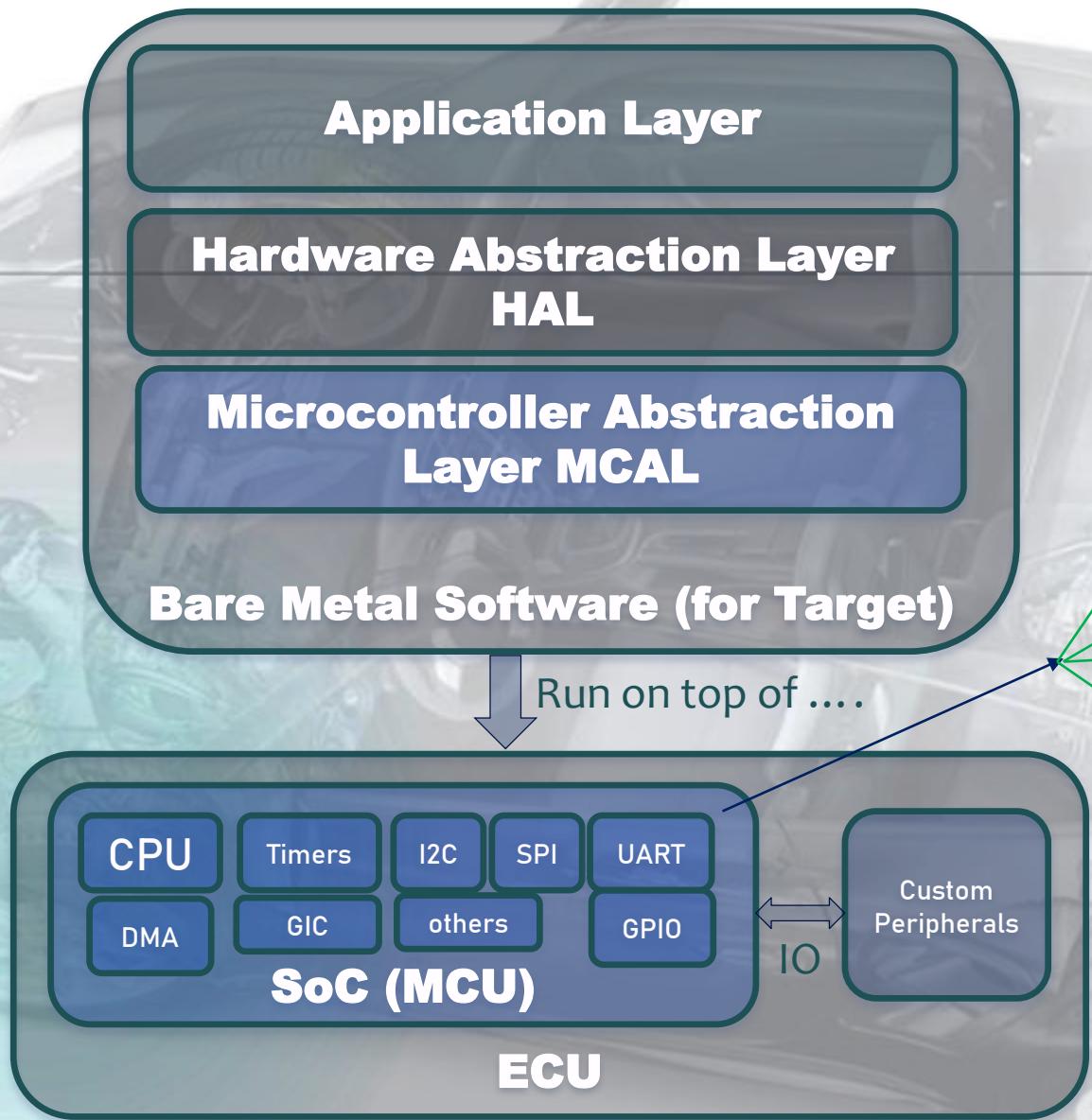


LEARN-IN-DEPTH
Be professional in
embedded system



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

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/>



4

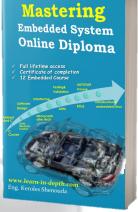
#LEARN_IN_DEPTH

#Be_professional_in
embedded_system

eng. Keroles Shenouda

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

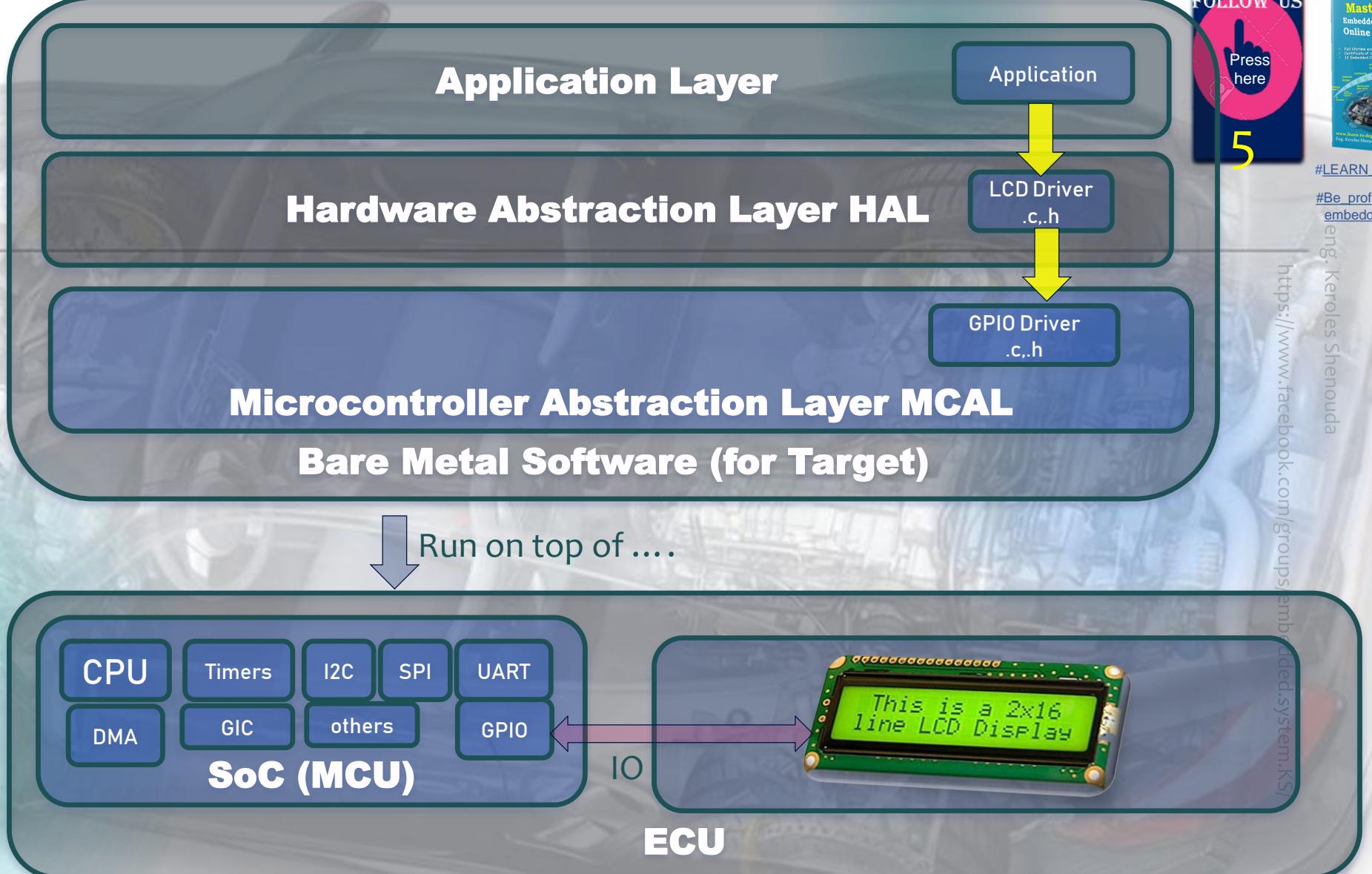




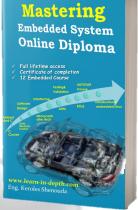
5

#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/>



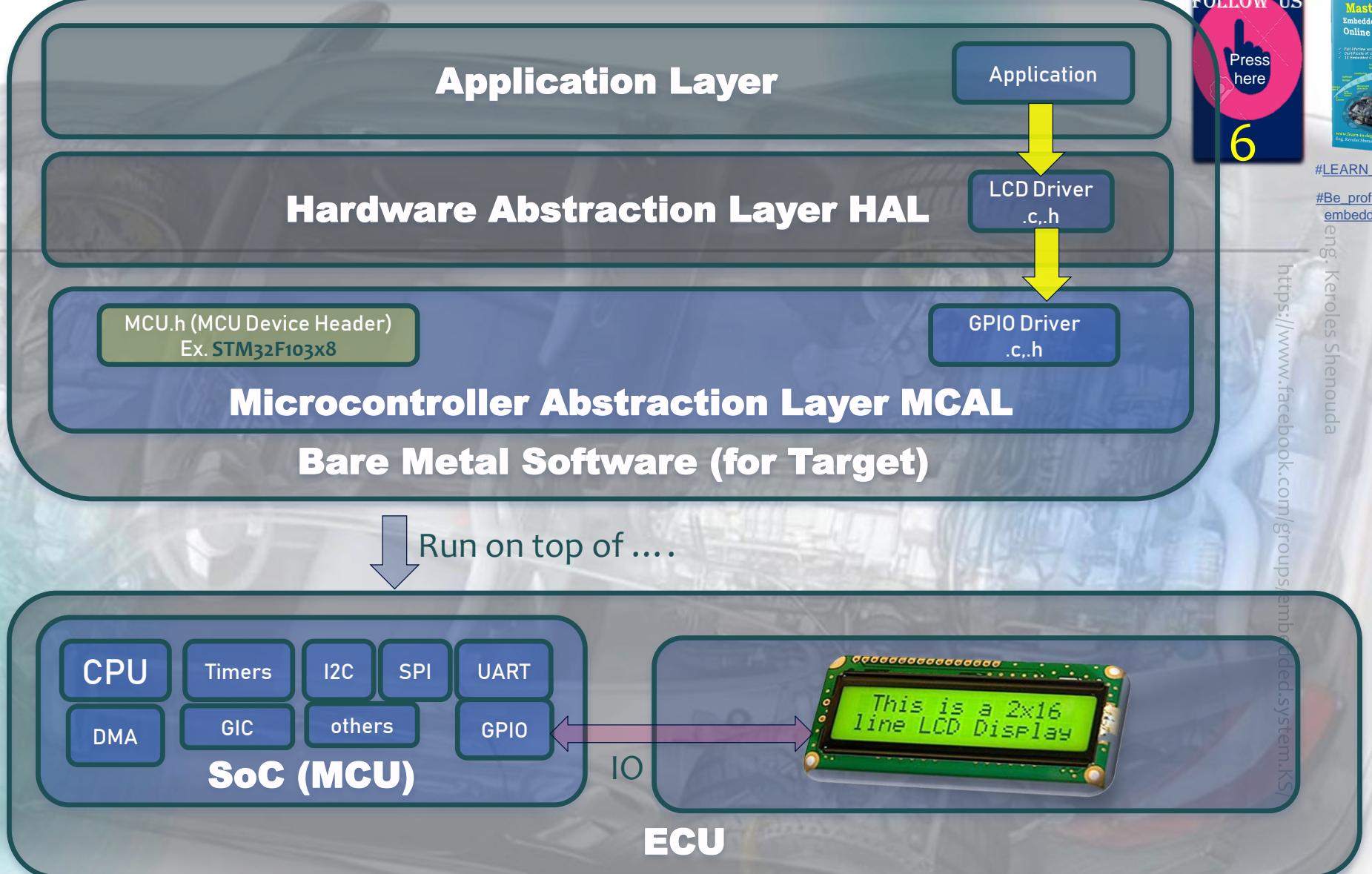
6

#LEARN_IN_DEPTH

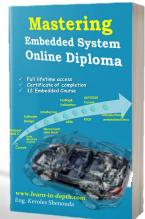
#Be_professional_in_embedded_system

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/>

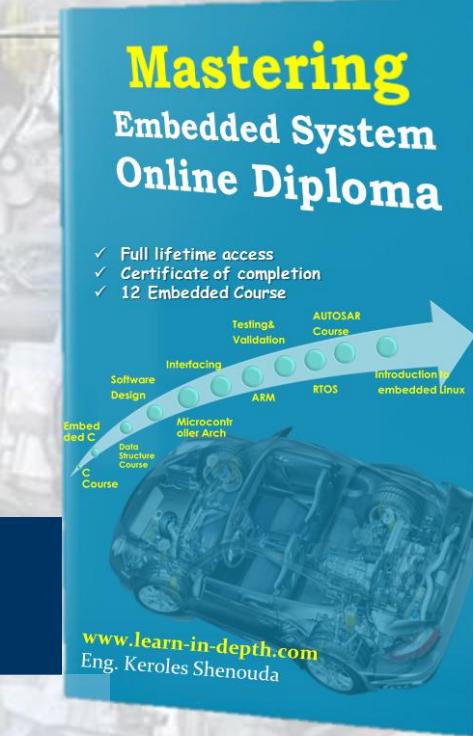


7

#LEARN_IN_DEPTH

#Be_professional_in_embedded_system

eng. Keroles Shenouda

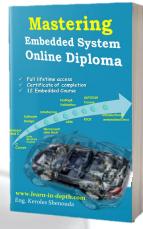
<https://www.facebook.com/groups/embedded.system.KS/>

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/>



8

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

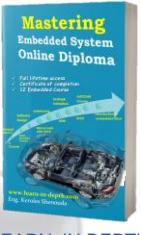
MCU Device Header

- ▶ 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/>

File layout

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



FOLLOW US

Press
here

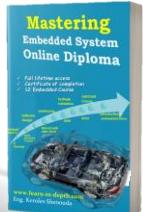
9

#LEARN IN DEPTH

#Be professional in embedded system



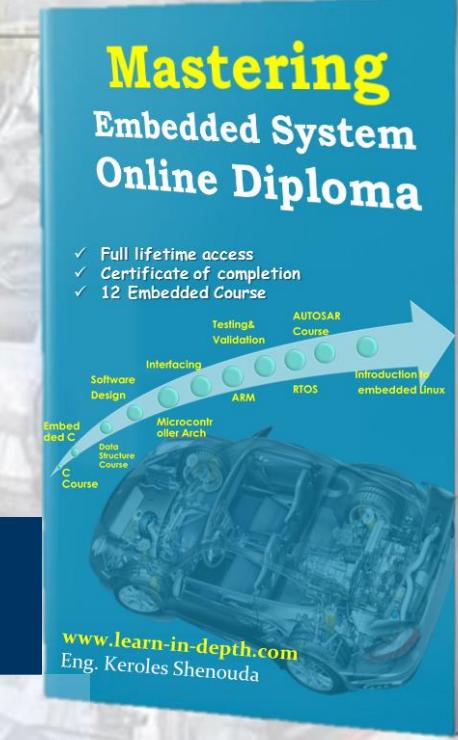
10



#LEARN_IN_DEPTH

#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

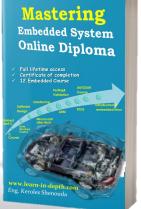
www.learn-in-depth.com
Eng. Keroles Shenouda

LEARN-IN-DEPTH
Be professional in
embedded system



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

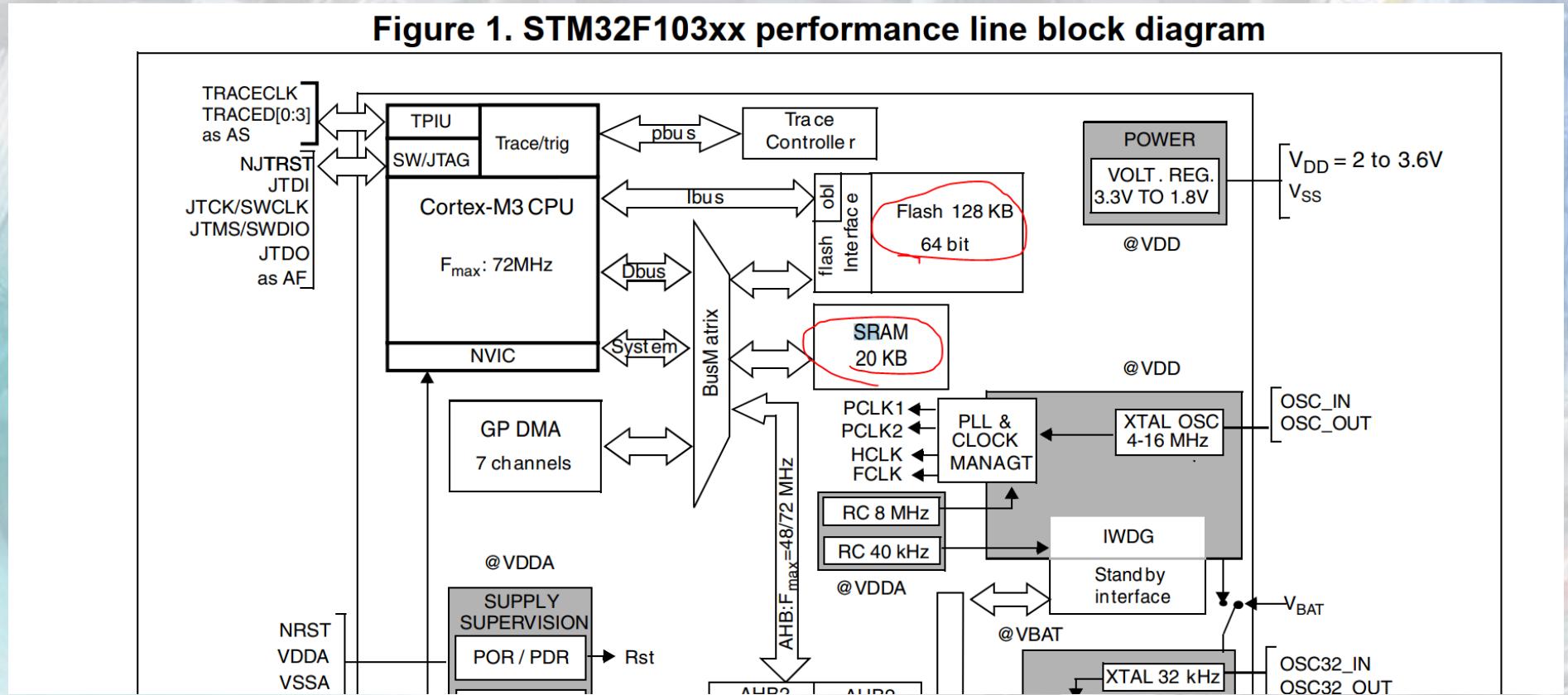
Specify the memory Map



11

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

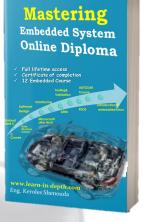
https://www.facebook.com/groups/embedded.system.KS/
eng. Keroles Shenouda



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



12

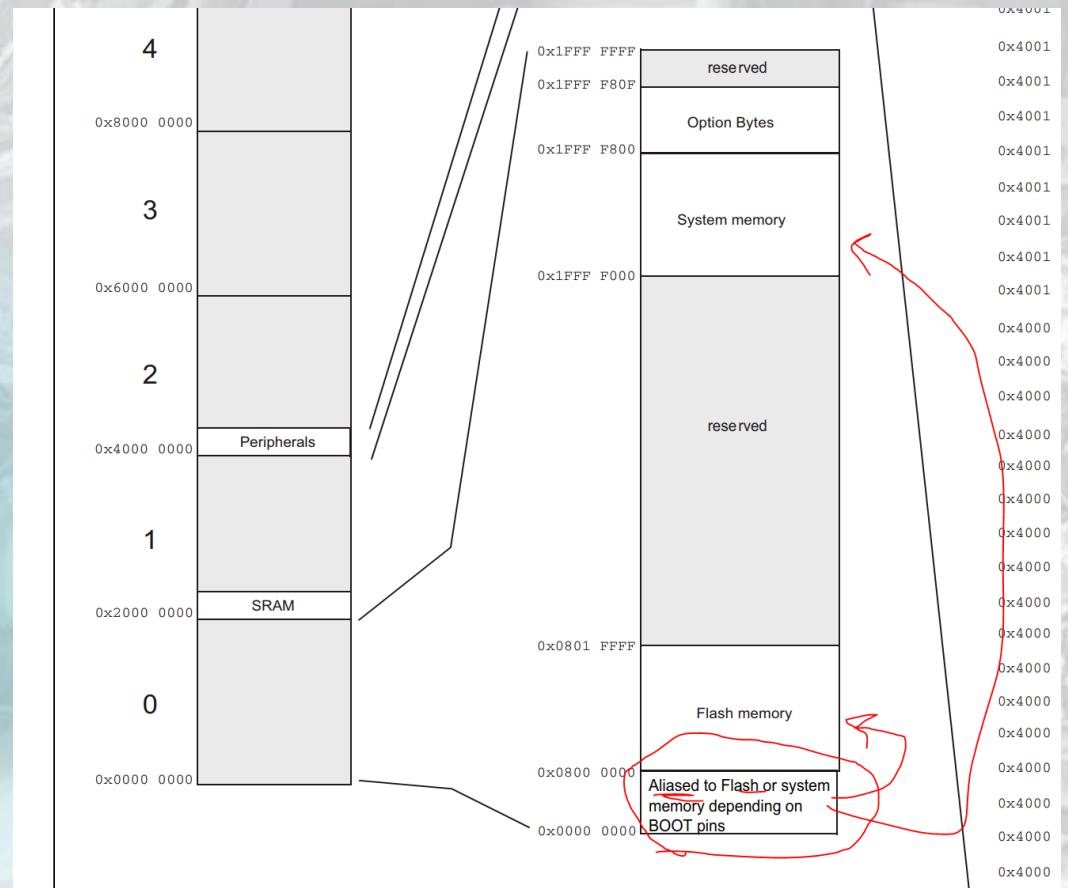


#LEARN_IN_DEPTH

#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/>



#LEARN_IN_DEPTH

#Be_professional_in
embedded_system

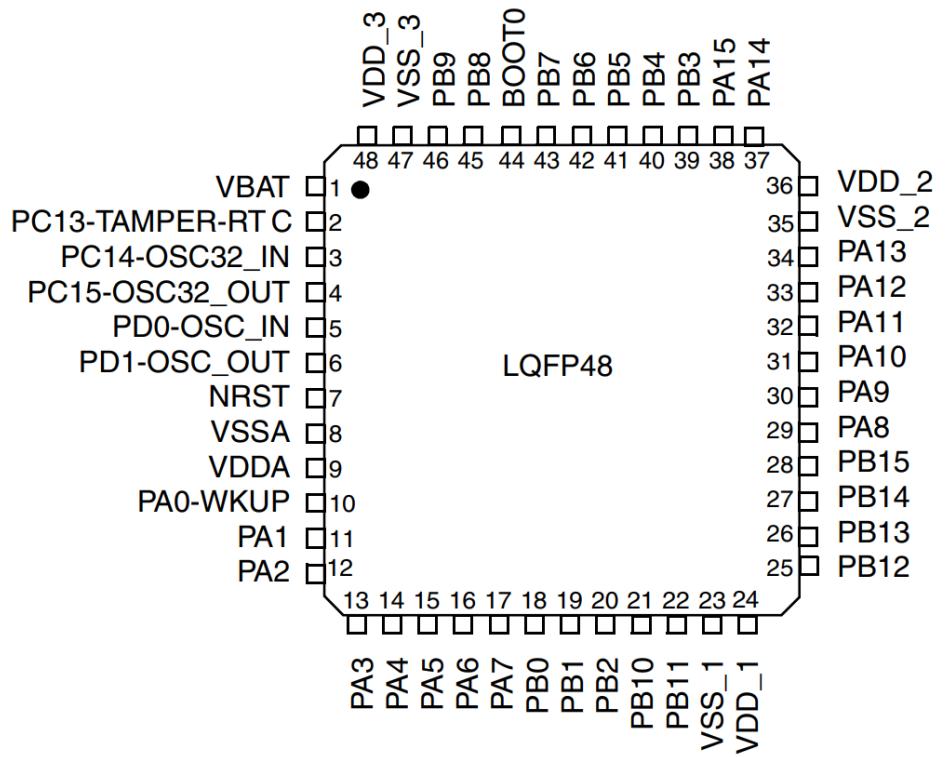
eng. Keroles Shenouda

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

```
STM32F103x8.h  stm32f103x6.h
1/* 
2 * STM32F103x8.h
3 *
4 * Created on: Apr 14, 2021
5 * Author: Keroles Shenouda
6 * Mastering Embedded System Online Diploma
7 */
8
9 #ifndef STM32F103X8_H_
10 #define STM32F103X8_H_
11
12//-----
13 //Base addresses for Memories
14 //-----
15 #define FLASH_Memory_BASE          0x08000000UL
16 #define System_Memory_BASE         0x1FFFFF000UL
17 #define SRAM_BASE                  0x20000000UL
18
19 #define Peripherals_BASE           0x40000000UL
20
21 #define Cortex_M3_Internal_Peripherals_BASE 0xE0000000UL
22
23
24 //-----*
25
26 #endif /* STM32F103X8_H_ */
27
```

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

Figure 8. STM32F103xx performance line LQFP48 pinout



```

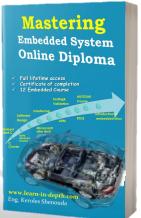
24 //-----
25 //Base addresses for AHB Peripherals
26 //-----
27 #define RCC_BASE (Peripherals_BASE + 0x00021000UL)
28
29 //-----
30 //Base addresses for APB2 Peripherals
31 //-----
32
33 //GPIO
34 //A,B fully included in LQFP48 Package
35 #define GPIOA_BASE (Peripherals_BASE + 0x00010800UL)
36 #define GPIOB_BASE (Peripherals_BASE + 0x00010C00UL)
37 //C,D Partially included in LQFP48 Package
38 #define GPIOC_BASE (Peripherals_BASE + 0x00011000UL)
39 #define GPIOD_BASE (Peripherals_BASE + 0x00011400UL)
40 //EP not included in LQFP48 Package
41 #define GPIOE_BASE (Peripherals_BASE + 0x00011800UL)
42 //-----
43
44 #define AFIO_BASE (Peripherals_BASE + 0x00010000UL)
45 #define EXTI_BASE (Peripherals_BASE + 0x00010400UL)
46 //-----
47 //Base addresses for APB1 Peripherals
48 //-----

```

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



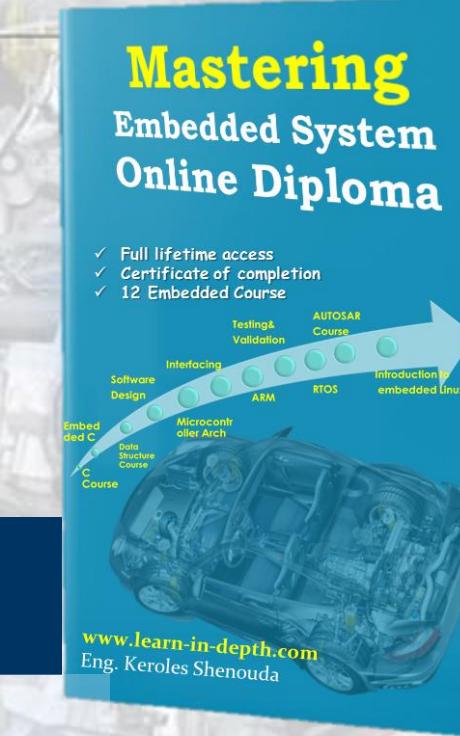
15



#LEARN_IN_DEPTH

#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

Specify Peripheral register by C Structure

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/>

16

```
46 //-----  
47 //Base addresses for APB1 Peripherals  
48 //-----  
49 |  
50 //-----  
51 //Peripheral register  
52 //-----  
53  
54  
55 //-----  
56 //Peripheral register: GPIO  
57 //-----  
58  
59  
60 //-----  
61  
62  
63 //-----  
64 //Peripheral register: EXTI  
65 //-----  
66  
67  
68 //-----  
69  
70  
71 //-----  
72 //Peripheral register: AFIO  
73 //-----  
74  
75  
76 //-----
```

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

Table 59. GPIO register map and reset values

Reset values

```
    volatile uint32_t CRL;
    volatile uint32_t CRH;
    volatile uint32_t IDR;
    volatile uint32_t ODR;
    volatile uint32_t BSRR;
    volatile uint32_t BRR;
    volatile uint32_t LCKR;
}
GPIO_TypeDef;
```

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



Table 60. AFIO register map and reset values

```
81 //-----*
82 //Peripheral register: AFIO
83 //-----*
84 typedef struct
85 {
86     volatile uint32_t EVCR;
87     volatile uint32_t MAPR;
88     volatile uint32_t EXTICR1;
89     volatile uint32_t EXTICR2;
90     volatile uint32_t EXTICR3;
91     volatile uint32_t EXTICR4;
92     uint32_t RESERVED0;
93     volatile uint32_t MAPR2;
94 } AFIO_TypeDef;
95
96 // * * * * *
```

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

10.3.7 EXTI register map

The following table gives the EXTI register map and the reset values. Bits 19 in all registers are used in connectivity line devices and reserved otherwise.

Table 64. External interrupt/event controller register map and reset values

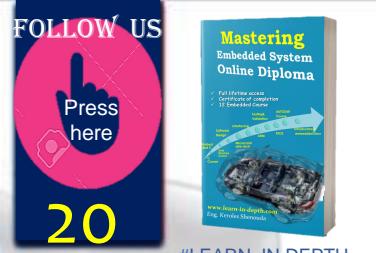
Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	EXTI_IMR																																
	Reset value																																
0x04	EXTI_EMR																																
	Reset value																																
0x08	EXTI_RTSR																																
	Reset value																																
0x0C	EXTI_FTSR																																
	Reset value																																
0x10	EXTI_SWIER																																
	Reset value																																
0x14	EXTI_PR																																
	Reset value																																

```

73④ //-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
74 //Peripheral register: EXTI
75 //-*-*-*-*-*-*-*-*-*-*-*-*-
76④ typedef struct
77 {
78     volatile uint32_t IMR;
79     volatile uint32_t EMR;
80     volatile uint32_t RTSR;
81     volatile uint32_t FTSR;
82     volatile uint32_t SWIER;
83     volatile uint32_t PR;
84 } EXTI_TypeDef;
85

```

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



#LEARN_IN_DEPTH

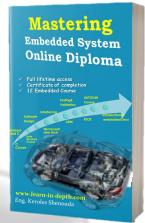
#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

```
130
131 //-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
132 //-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
133 //-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
134 //-*-*-*-*-*-*-*-*-*-*-
135 //Peripheral Instants:
136 //-*-*-*-*-*-*-*-*-*-*-
137
138
139 GPIO_TypeDef *GPIOA = (GPIO_TypeDef *) GPIOA_BASE ;
140
141 GPIO_TypeDef *GPIOA = GPIOA ;
142 #define GPIOA ((GPIO_TypeDef *)GPIOA_BASE)
143
144
145
146 |
```

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

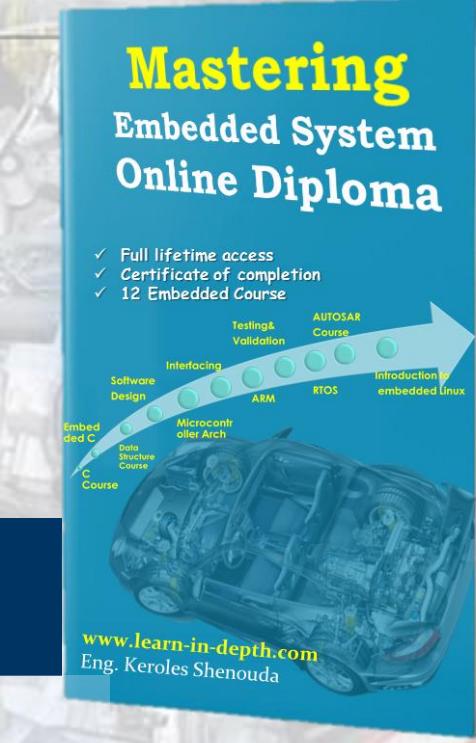


21

#LEARN_IN_DEPTH

#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

GPIO Driver

LEARN-IN-DEPTH
Be professional in
embedded system

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



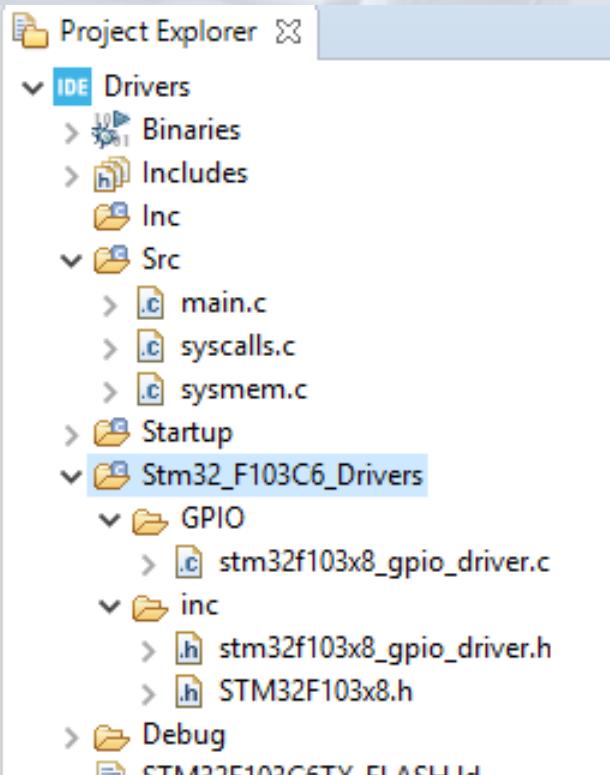
22

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

Folder Structure



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



23

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

Include the Drivers Headers

The screenshot shows a software interface for embedded system development. On the left is a file tree for a project named 'Drivers'. The 'Src' folder contains files like main.c, syscalls.c, and sysmem.c. Below it is a 'Stm32_F103C6_Drivers' folder containing 'GPIO' and 'inc' subfolders, and a 'STM32F103C6TX_FLASH.ld' file. The central part is a code editor showing a C file with the following content:

```
1 /*  
2 *  stm32f103x8_gpio_driver.c  
3 *  
4 *  Created on: Apr 28, 2022  
5 *  Author: Keroles Shenouda  
6 *  Mastering Embedded System  
7 *  www.learn-in-depth.com  
8 */  
9  
10 #ifndef STM32F103X8_GPIO_DRIVER_H_  
11 #define STM32F103X8_GPIO_DRIVER_H_  
12  
13 #include "STM32F103x8.h"  
14  
15  
16  
17  
18  
19 #endif /* STM32F103X8_GPIO_DRIVER_H_ */
```

To the right of the code editor is a 'Properties for Drivers' dialog box titled 'Settings'. It shows the 'Tool Settings' tab selected. Under 'Include paths (-I)', the path `./inc` and `"${workspace_loc:/$(ProjName)/Stm32_F103C6_Drivers/inc}"` are listed. The 'Include files (-include)' section is empty.

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

GPIO functional description



GPIO functional description

Each of the general-purpose I/O ports has two 32-bit configuration registers (`GPIOx_CRL`, `GPIOx_CRH`), two 32-bit data registers (`GPIOx_IDR`, `GPIOx_ODR`), a 32-bit set/reset register (`GPIOx_BSRR`), a 16-bit reset register (`GPIOx_BRR`) and a 32-bit locking register (`GPIOx_LCKR`).

Subject to the specific hardware characteristics of each I/O port listed in the *datasheet*, each port bit of the General Purpose IO (GPIO) Ports, can be individually configured by software in several modes:

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

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



25

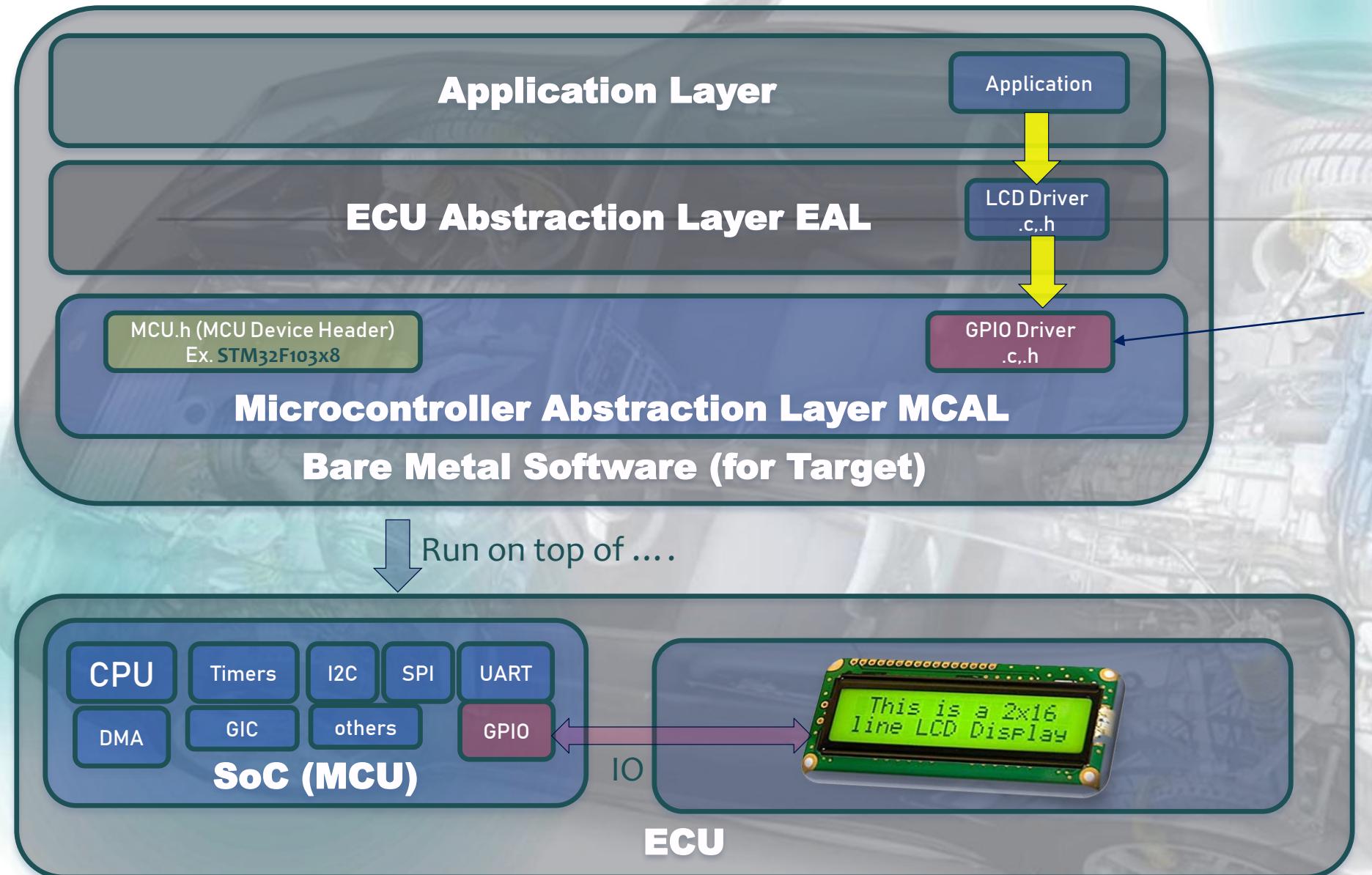
#LEARN_IN_DEPTH

#Be_professional_in_embedded_system

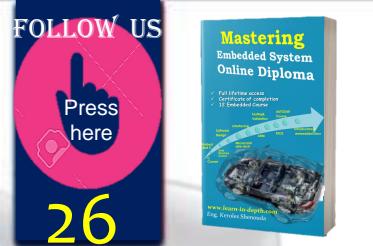
eng. Keroles Shenouda

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

We need a configuration feature
For every PIN



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

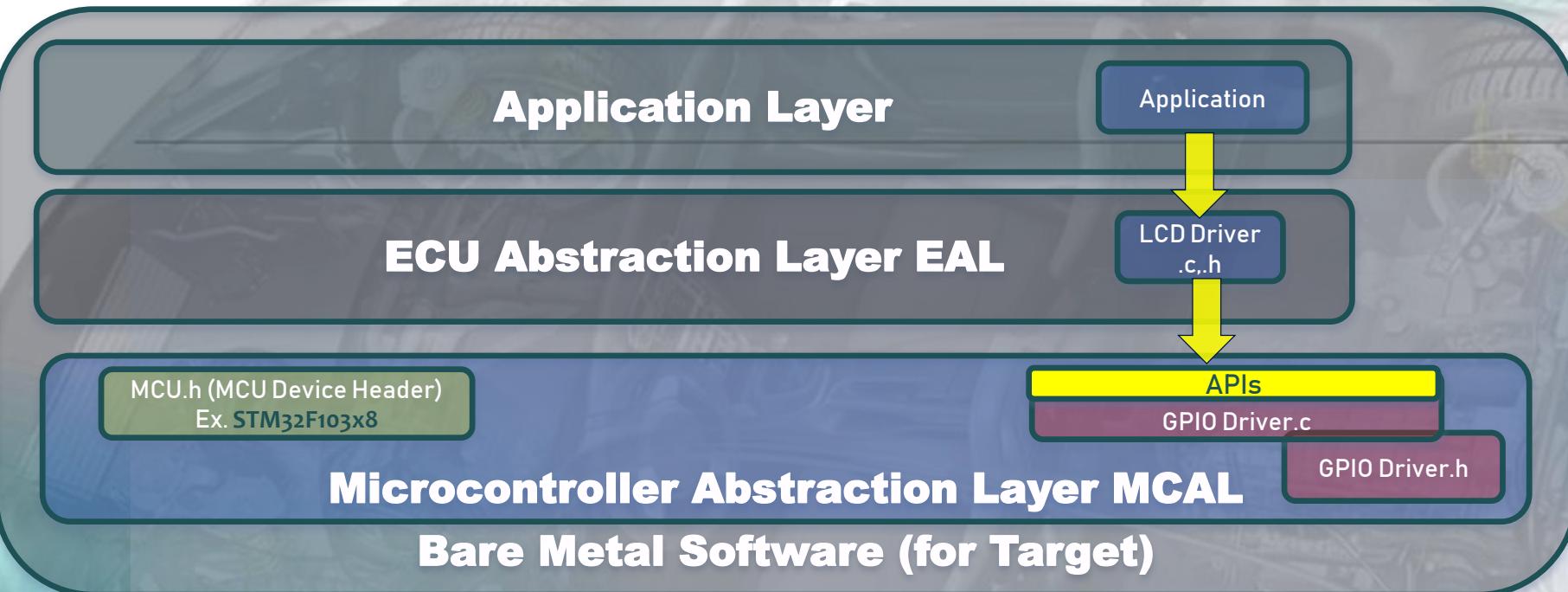


#LEARN_IN_DEPTH

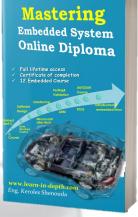
#Be_professional_in
embedded_system

eng. Keroles Shenouda

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



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

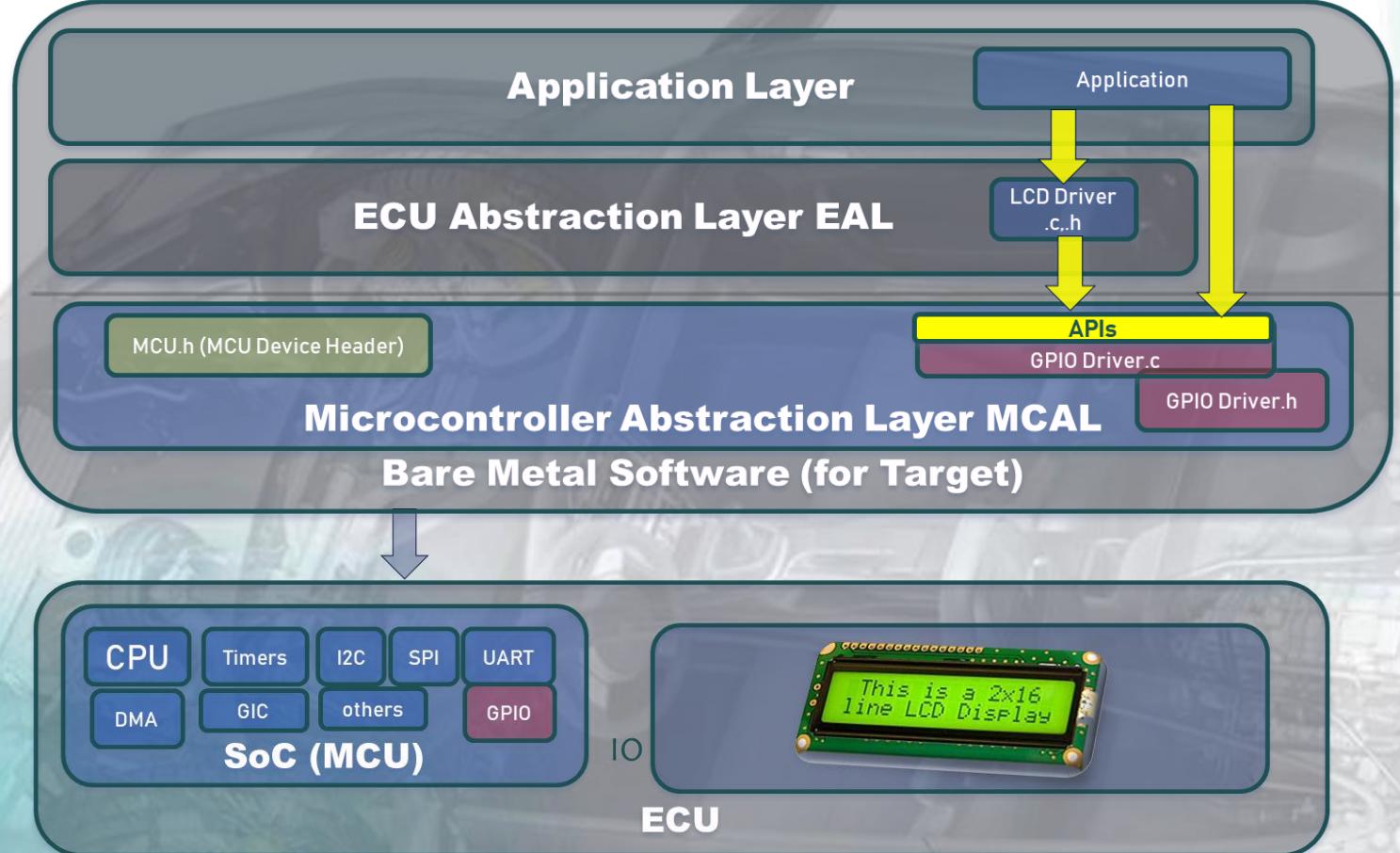


28

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

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



MCAL_GPIO_ReadPin

MCAL_GPIO_WritePin

MCAL_GPIO_Init

MCAL_GPIO_EXTI_IRQConfig

MCAL_GPIO_ReadPort

MCAL_GPIO_WritePort

MCAL_GPIO_DeInit

MCAL_GPIO_EXTI_IRQHandler

MCAL_GPIO_TogglePin

APIs GPIO Driver.c

GPIOx

PINx

Config Struct

Custom Data

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



29

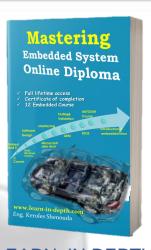
#LEARN_IN_DEPTH
#Be_professional_in_embedded_system<https://www.facebook.com/groups/embedded.system.KS/>

eng.Keroles Shenouda

File layout .h

```
//-----  
//Includes  
//-----  
  
//-----  
//User type definitions (structures)  
//-----  
  
//-----  
//Macros Configuration References  
//-----  
  
/*@ref Module REF NAME define  
 */  
  
/*  
* ======  
* APIs Supported by "MCAL GPIO DRIVER"  
* ======  
*/
```

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



30

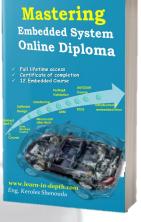
#LEARN_IN_DEPTH
#Be_professional_in_embedded_system<https://www.facebook.com/groups/embedded.system.KS/>

eng.Keroles Shenouda

File layout .c

```
//-----  
//Macros Configuration References  
//-----  
  
/*@ref Module_REF_NAME_define  
 */  
  
/*  
 * ======  
 * APIs Supported by "MCAL GPIO DRIVER"  
 * ======  
 */
```

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



31

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

```

typedef struct
{
    uint8_t GPIO_PinNumber ; // Specifies the GPIO pins to be configured.
                           // This parameter can be a value of @ref GPIO_PINS_define

    uint8_t GPIO_MODE ; // Specifies the operating mode for the selected pins
                        // This parameter can be a value of @ref GPIO_MODE_define

    uint8_t GPIO_Output_Speed ; // Specifies the speed for the selected pins
                               // This parameter can be a value of @ref GPIO_SPEED_define
}GPIO_PinConfig_t;

40/* @ref GPIO_MODE_define
41 0: Analog mode
42 1: Floating input (reset state)
43 2: Input with pull-up
44 3: Input with pull-down
45 4: General purpose output push-pull
46 5: General purpose output Open-drain
47 6: Alternate function output Push-pull
48 7: Alternate function output Open-drain
49 8: Alternate function INPUT
50 */
51 #define GPIO_MODE_ANALOG 0x00000000u //Analog mode
52 #define GPIO_MODE_INPUT_FLO 0x00000001u //Floating input
53 #define GPIO_MODE_INPUT_PU 0x00000002u //Input with pull-up
54 #define GPIO_MODE_INPUT_PD 0x00000003u //Input with pull-down
55 #define GPIO_MODE_OUTPUT_PP 0x00000004u //General purpose output push-pull
56 #define GPIO_MODE_OUTPUT_OD 0x00000005u //General purpose output Open-drain
57 #define GPIO_MODE_OUTPUT_AF_PP 0x00000006u //Alternate function output Push-pull
58 #define GPIO_MODE_OUTPUT_AF_OD 0x00000007u //Alternate function output Open-drain
59 #define GPIO_MODE_OUTPUT_AF_INPUT 0x00000008u //Alternate function INPUT

62/* @ref GPIO_SPEED_define
63 01: Output mode, max speed 10 MHz.
64 10: Output mode, max speed 2 MHz.
65 11: Output mode, max speed 50 MHz.
66 */
67 #define GPIO_MODE_ANALOG 0x00000001u //Output mode, max speed 10 MHz
68 #define GPIO_MODE_INPUT_FLO 0x00000002u //Output mode, max speed 2 MHz
69 #define GPIO_MODE_INPUT_PU 0x00000003u //Output mode, max speed 50 MHz
71

```

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: Input with pull-up / pull-down
- 11: Reserved

In output mode (MODE[1:0] > 00):

- 00: General purpose output push-pull
- 01: General purpose output Open-drain
- 10: Alternate function output Push-pull
- 11: Alternate function output Open-drain

Bits 29:28, 25:24, **MODEy[1:0]:** Port x mode bits (y= 0 .. 7)
 21:20, 17:16, 13:12, These bits are written by software to configure the corresponding I/O port.
 9:8, 5:4, 1:0 Refer to [Table 20: Port bit configuration table](#).

- 00: Input mode (reset state)
- 01: Output mode, max speed 10 MHz.
- 10: Output mode, max speed 2 MHz.
- 11: Output mode, max speed 50 MHz.

<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/>

```
104
105 *=====
106 *===== APIs Supported by "MCAL GPIO DRIVER"
107 *=====
108 *=====
109 */
110
111 void MCAL_GPIO_Init      (GPIO_TypeDef *GPIOx , GPIO_PinConfig_t* PinConfig);
112 void MCAL_GPIO_DeInit    (GPIO_TypeDef *GPIOx );
113
114 //READ  APIs
115 uint8_t  MCAL_GPIO_ReadPin   (GPIO_TypeDef *GPIOx , uint8_t PinNumber);
116 uint16_t MCAL_GPIO_ReadPort (GPIO_TypeDef *GPIOx );
117
118 //WRITE  APIs
119 void MCAL_GPIO_WritePort   (GPIO_TypeDef *GPIOx , uint8_t PinNumber, uint8_t Value);
120 void MCAL_GPIO_WritePin    (GPIO_TypeDef *GPIOx , uint8_t Value);
121
122 void MCAL_GPIO_TogglePin   (GPIO_TypeDef *GPIOx , uint8_t PinNumber);
123
124
125 void MCAL_GPIO_LockPin     (GPIO_TypeDef *GPIOx , uint8_t PinNumber);
126
```

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



33

#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

eng. Keroles Shenouda

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

Functional description

```
/**-----  
 * @Fn-  
 * @brief -  
 * @param [in] -  
 * @param [out] -  
 * @retval -  
 * Note-  
 */
```

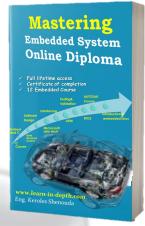
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Table 20. Port bit configuration table

Configuration mode		CNF1	CNF0	MODE1	MODE0	PxODR register
General purpose output	Push-pull	0	0	01 10 11	see <i>Table 21</i>	0 or 1
	Open-drain		1			0 or 1
Alternate Function output	Push-pull	1	0			Don't care
	Open-drain		1			Don't care
Input	Analog	0	0	00	see <i>Table 21</i>	Don't care
	Input floating		1			Don't care
	Input pull-down	1	0			0
	Input pull-up					1

Table 21. Output MODE bits

MODE[1:0]	Meaning
00	Reserved
01	Maximum output speed 10 MHz
10	Maximum output speed 2 MHz
11	Maximum output speed 50 MHz



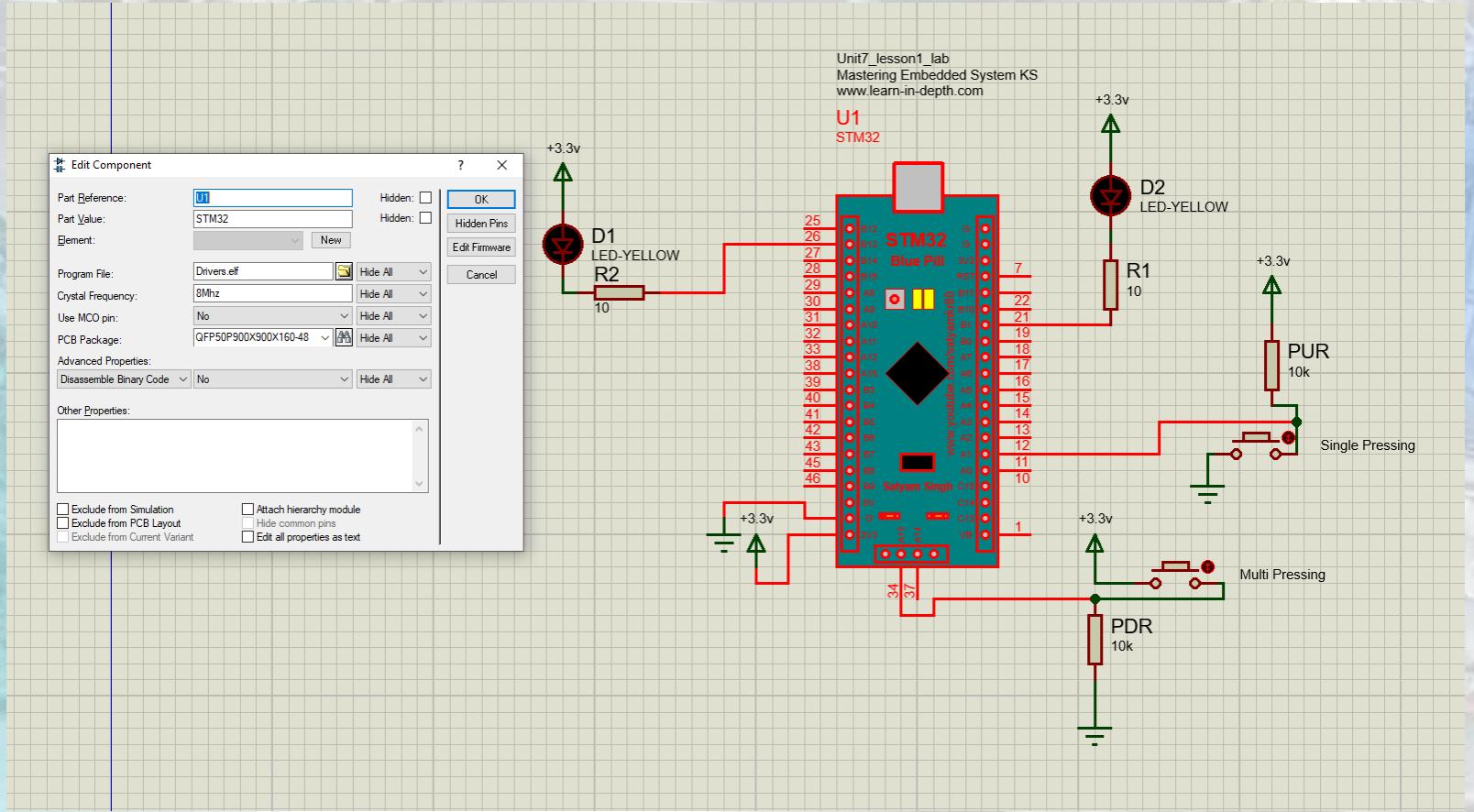
35

#LEARN_IN_DEPTH
#Be_professional_in
embedded_system

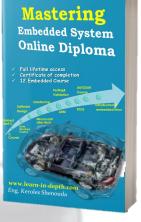
eng. Keroles Shenouda

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

Rewrite the unit7lesson1 lab by using the driver



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



36

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

```
43
44@ int main(void)
45 {
46
47     //Clock Enable
48
49     RCC_GPIOA_CLK_EN();
50     RCC_GPIOB_CLK_EN();
51
52     //GPIO init
53     GPIO_PinConfig_t PinCfg ;
54
55     //PA1 input HighZ
56     PinCfg.GPIO_PinNumber = GPIO_PIN_1 ;
57     PinCfg.GPIO_MODE = GPIO_MODE_INPUT_FLO ;
58     MCAL_GPIO_Init(GPIOA, &PinCfg) ;
59
60     //PB1 output push pull mode
61     //CNF  00: General purpose output push-pull
62     //01: Output mode, max speed 10 MHz.
63     PinCfg.GPIO_PinNumber = GPIO_PIN_1 ;
64     PinCfg.GPIO_MODE = GPIO_MODE_OUTPUT_PP ;
65     PinCfg.GPIO_Output_Speed = GPIO_SPEED_10M ;
66     MCAL_GPIO_Init(GPIOB, &PinCfg) ;
67
68     //PA13 input HighZ
69     PinCfg.GPIO_PinNumber = GPIO_PIN_13 ;
70     PinCfg.GPIO_MODE = GPIO_MODE_INPUT_FLO ;
71     MCAL_GPIO_Init(GPIOA, &PinCfg) ;
72
73     //PB13 output push pull mode
74     PinCfg.GPIO_PinNumber = GPIO_PIN_13 ;
75     PinCfg.GPIO_MODE = GPIO_MODE_OUTPUT_PP ;
76     PinCfg.GPIO_Output_Speed = GPIO_SPEED_10M ;
77     MCAL_GPIO_Init(GPIOB, &PinCfg) ;
78
79
80     /* Loop forever */
81     while(1)
82     {
83         //PA1 >>>Connected external PUR
84         if ( MCAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1) == 0 ) // press
85         {
86             MCAL_GPIO_TogglePin(GPIOB, GPIO_PIN_1);
87             while ( MCAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1) == 0 ) ;//Single press
88         }
89         //PA13 >>>Connected external PDR
90         if ( MCAL_GPIO_ReadPin(GPIOA, GPIO_PIN_13) == 1 ) // Multi pressing
91         {
92             MCAL_GPIO_TogglePin(GPIOB, GPIO_PIN_13);
93         }
94         my_wait(1) ;
95     }
96 }
97 }
```

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



37

#LEARN_IN_DEPTH

#Be_professional_in_embedded_system

References

- ▶ <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZmtlLnV0bS5teXxyaWR6dWFuLXMtd2Vic2I0ZXxneDo2ODU0NzIKM2JkOTg4MjRk>
- ▶ <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
- ▶ 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/>



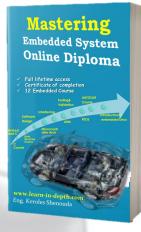
#LEARN_IN_DEPTH
#Be_professional_in_embedded_system

38

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
- ▶ 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/>



39

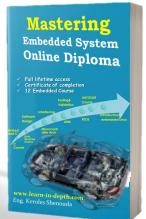
#LEARN_IN_DEPTH

#Be_professional_in
embedded_system

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/>



40

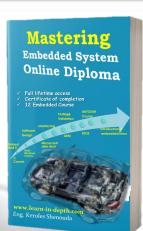
#LEARN_IN_DEPTH

#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/>



41

#LEARN_IN_DEPTH

#Be_professional_in
embedded_system

Thank You

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