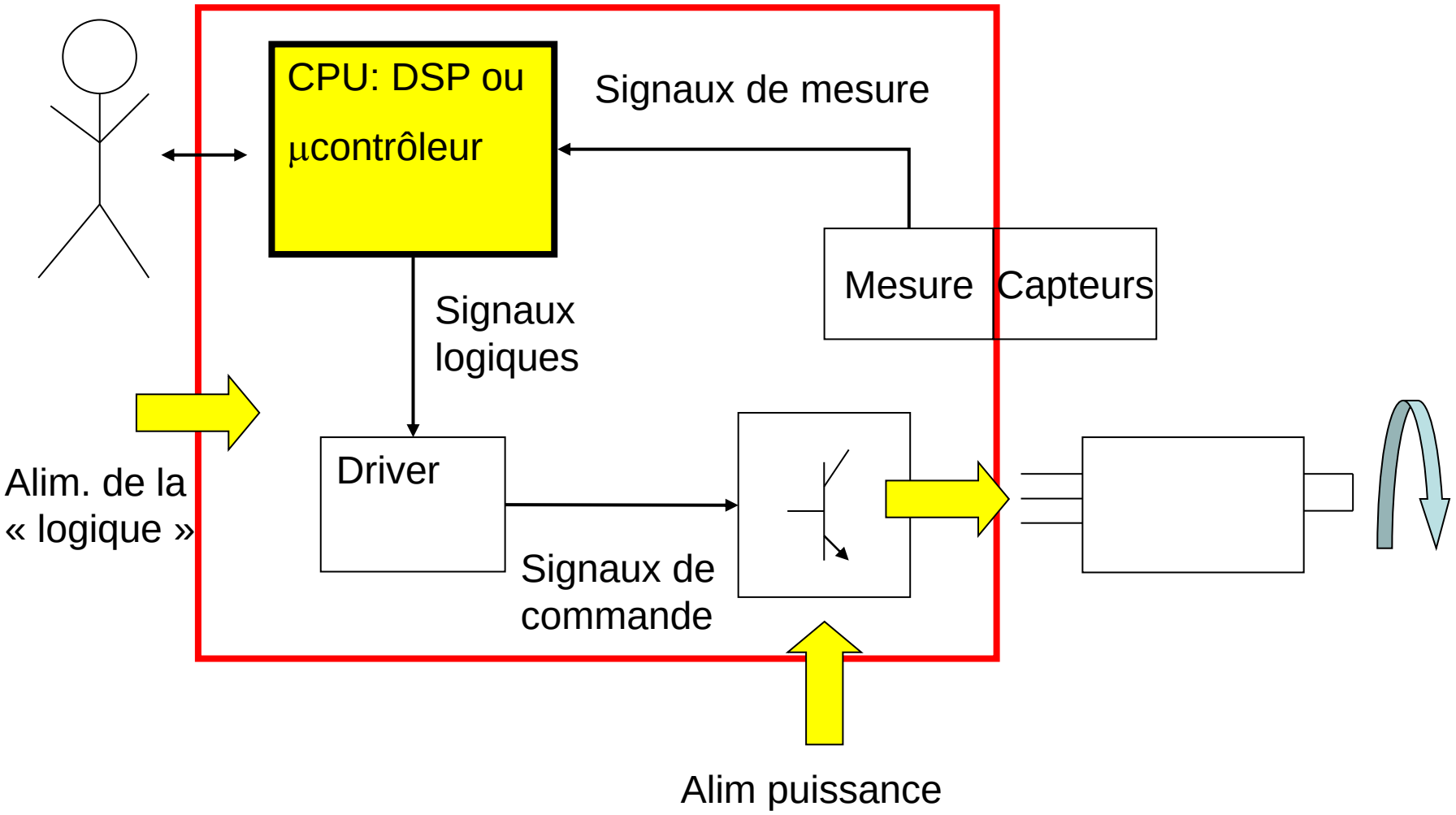
Commande embarquée de moteurs

1 Prise en main du Microcontrôleur

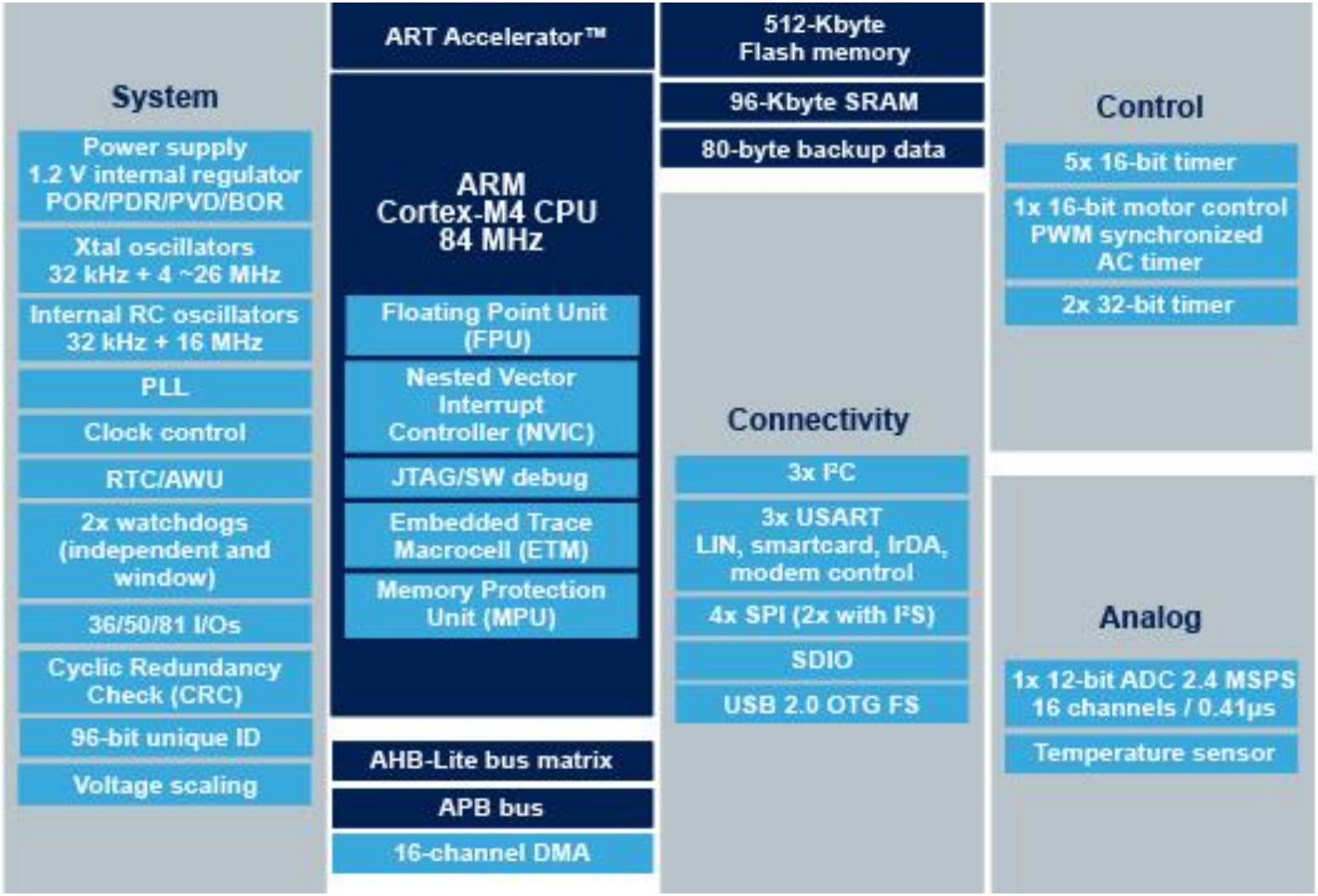
Périphériques

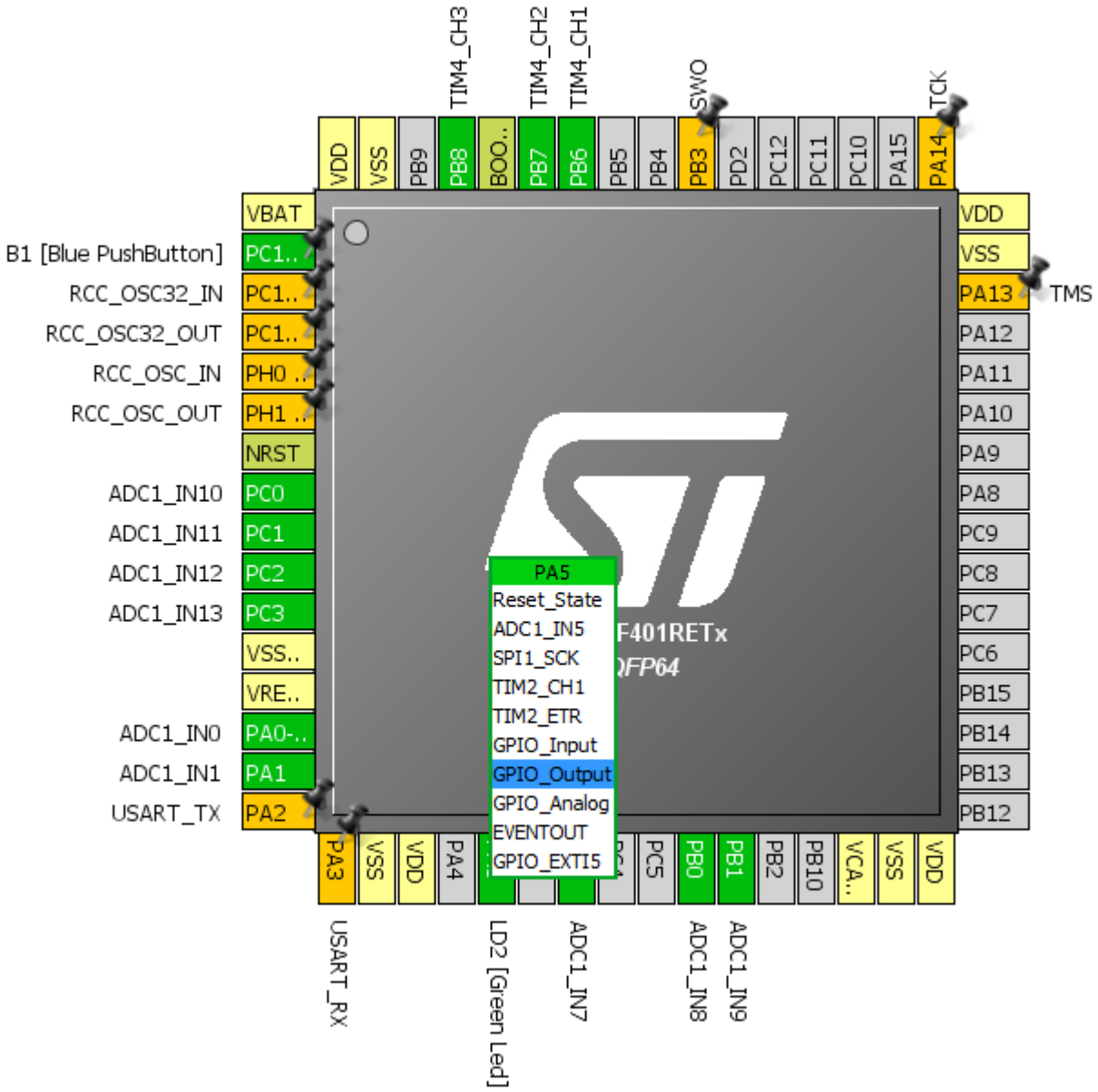
Christian Koechli

Composants de l'électronique de commande



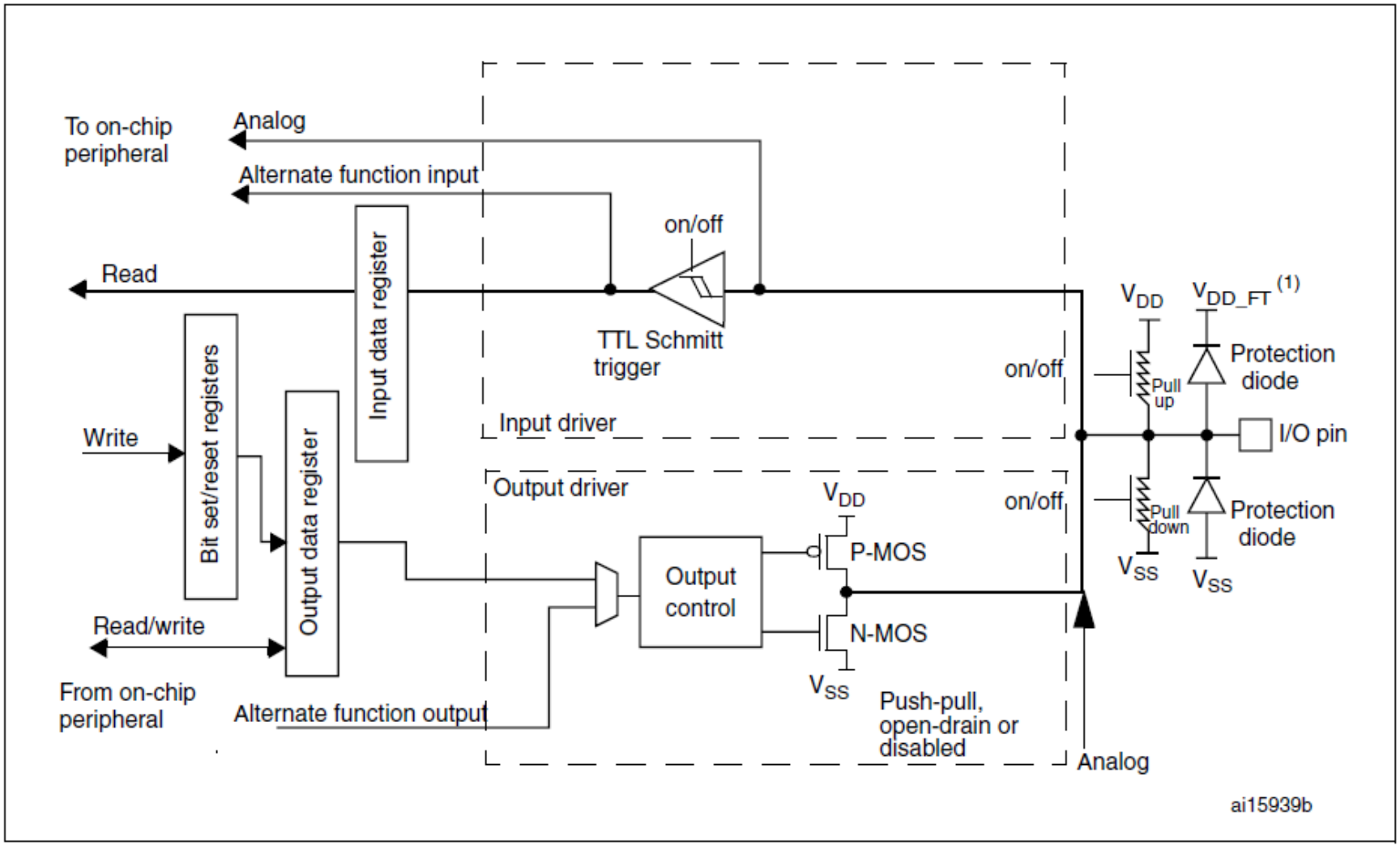
STM32F401RE




VDD
VSS
PB9
PB8
BOO..
PB7
PB6
PB5
PB4
PB3
PD2
PC12
PC11
PC10
PA15
PA14
TIM4_CH3
TIM4_CH2
TIM4_CH1
SWO
TCK
VBAT
PC1..
PC1..
PC1..
PH0 ..
PH1 ..
NRST
PC0
PC1
PC2
PC3
VSS..
VRE..
PA0-..
PA1
PA2
B1 [Blue PushButton]
RCC_OSC32_IN
RCC_OSC32_OUT
RCC_OSC_IN
RCC_OSC_OUT
ADC1_IN10
ADC1_IN11
ADC1_IN12
ADC1_IN13
ADC1_IN0
ADC1_IN1
USART_TX
VDD
VSS
PA13
PA12
PA11
PA10
PA9
PA8
PC9
PC8
PC7
PC6
PB15
PB14
PB13
PB12
TMS
PA5
Reset_State
ADC1_IN5
SPI1_SCK
TIM2_CH1
TIM2_ETR
GPIO_Input
GPIO_Output
GPIO_Analog
EVENTOUT
GPIO_EXTI5
F401RETx
QFP64
PA3
VSS
VDD
PA4
PC5
PB0
PB1
PB2
PB10
VSS
VDD
USART_RX
LD2 [Green Led]
ADC1_IN7
ADC1_IN8
ADC1_IN9

Exemple GPIO

Figure 16. Basic structure of a five-volt tolerant I/O port bit



Memory Map 1

Table 1. STM32F401xB/C and STM32F401xD/E register boundary addresses

Boundary address	Peripheral	Bus	Register map
0x5000 0000 - 0x5003 FFFF	USB OTG FS	AHB2	*Section 22.16.6: OTG_FS register map on page 746*
0x4002 6400 - 0x4002 67FF	DMA2	AHB1	*Section 9.5.11: DMA register map on page 197*
0x4002 6000 - 0x4002 63FF	DMA1		
0x4002 3C00 - 0x4002 3FFF	Flash interface register		*Section 3.8: Flash interface registers on page 60*
0x4002 3800 - 0x4002 3BFF	RCC		*Section 6.3.22: RCC register map on page 136*
0x4002 3000 - 0x4002 33FF	CRC		*Section 4.4.4: CRC register map on page 70*
0x4002 1C00 - 0x4002 1FFF	GPIOH		*Section 8.4.11: GPIO register map on page 162*
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		

Dans le fichier stm32f401xe.h

```
281 typedef struct
282 {
283   __IO uint32_t MODER;    /*!< GPIO port mode register,               Address offset: 0x00      */
284   __IO uint32_t OTYPER;   /*!< GPIO port output type register,        Address offset: 0x04      */
285   __IO uint32_t OSPEEDR;  /*!< GPIO port output speed register,       Address offset: 0x08      */
286   __IO uint32_t PUPDR;    /*!< GPIO port pull-up/pull-down register,  Address offset: 0x0C      */
287   __IO uint32_t IDR;      /*!< GPIO port input data register,         Address offset: 0x10      */
288   __IO uint32_t ODR;      /*!< GPIO port output data register,        Address offset: 0x14      */
289   __IO uint32_t BSRR;     /*!< GPIO port bit set/reset register,      Address offset: 0x18      */
290   __IO uint32_t LCKR;     /*!< GPIO port configuration lock register, Address offset: 0x1C      */
291   __IO uint32_t AFR[2];   /*!< GPIO alternate function registers,     Address offset: 0x20-0x24 */
292 } GPIO_TypeDef;
```

```
796 #define GPIOA                ((GPIO_TypeDef *) GPIOA_BASE)
797 #define GPIOB                ((GPIO_TypeDef *) GPIOB_BASE)
798 #define GPIOC                ((GPIO_TypeDef *) GPIOC_BASE)
799 #define GPIOD                ((GPIO_TypeDef *) GPIOD_BASE)
800 #define GPIOE                ((GPIO_TypeDef *) GPIOE_BASE)
801 #define GPIOH                ((GPIO_TypeDef *) GPIOH_BASE)
```

Registre

8.4.6 GPIO port output data register (GPIOx_ODR) (x = A..E and H)

Address offset: 0x14

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
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

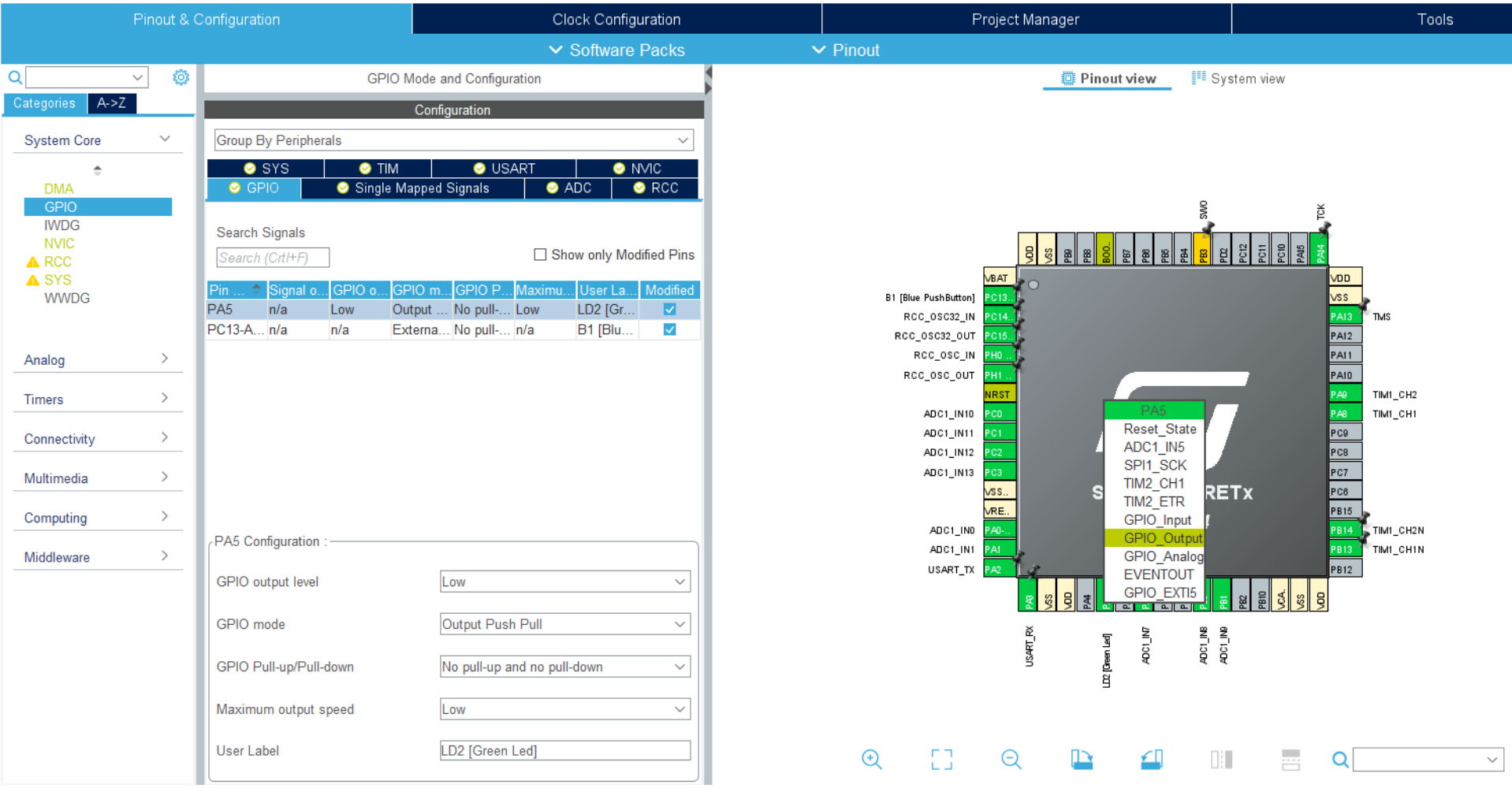
Bits 31:16 Reserved, must be kept at reset value.

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

These bits can be read and written by software.

Note: For atomic bit set/reset, the ODR bits can be individually set and reset by writing to the GPIOx_BSRR register (x = A..E and H).

Configuration par CubeMX



HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET);

Librairie

```
void HAL_GPIO_WritePin(
GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState)
{
  /* Check the parameters */
  assert_param(IS_GPIO_PIN(GPIO_Pin));
  assert_param(IS_GPIO_PIN_ACTION(PinState));

  if(PinState != GPIO_PIN_RESET)
  {
    GPIOx->BSRR = GPIO_Pin;
  }
  else
  {
    GPIOx->BSRR = (uint32_t)GPIO_Pin << 16U;
  }
}
```

8.4.7 GPIO port bit set/reset register (GPIOx_BSRR) (x = A..E and H)

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 and can be accessed in word, half-word or byte mode. 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 and can be accessed in word, half-word or byte mode. A read to these bits returns the value 0x0000.

0: No action on the corresponding ODRx bit

1: Sets the corresponding ODRx bit

Périphériques importants

- GPIO
- Timers + PWM+Capture/Compare/QEP
- Convertisseur AD
- DMA
- Gestionnaire d’interruptions
- Watchdog timer
- Communication