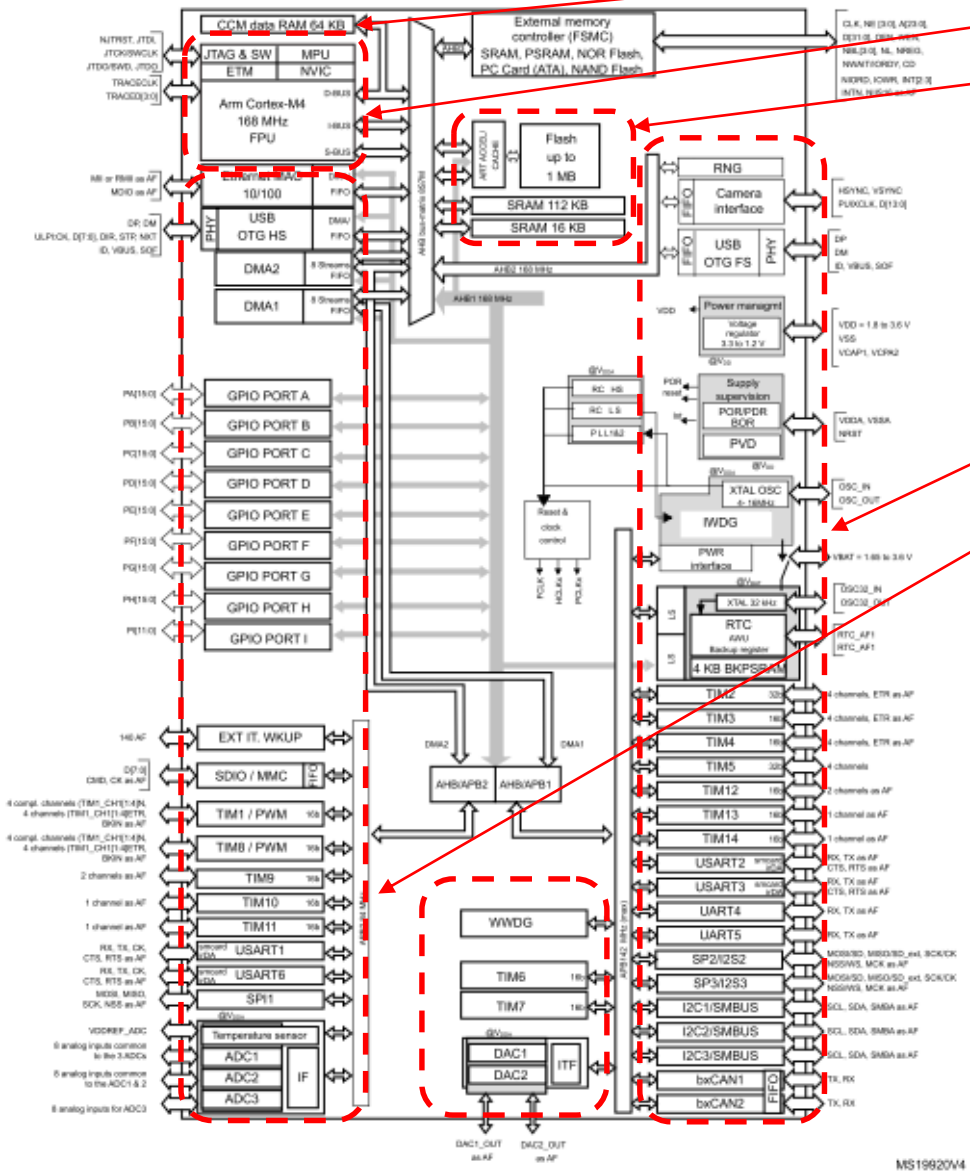
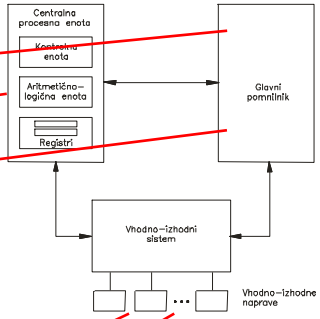
# *STM32F407 Discovery*

*Vhodno / izhodne naprave*

*GPIO Krmilnik*

# STM32F407VG



# *Delo na STM32F4 razvojnem sistemu*

Priključitev :

- **Mini USB** priklop na **krajši stranici**, svetita rdeči **LED** diodi

Poseben začetni projekt za *STM32F4* (e-učilnica) :

- ***dodajanje vsebine (template.s):***

'template.s - STM32CubeIDE

avigate Search Project Run Window Help

template.s

```
54
55 _start:
56     // Enable GPIOD Peripheral Clock (bit 3 in AHB1ENR register)
57     ldr r6, = RCC_AHB1ENR        // Load peripheral clock reg address to r6
58     ldr r5, [r6]                 // Read its content to r5
59     orr r5, #0x00000008           // Set bit 3 to enable GPIOD clock
60     str r5, [r6]                 // Store result in peripheral clock register
61
62     // Make GPIOD Pin12 as output pin (bits 25:24 in MODER register)
63     ldr r6, = GPIOD_MODER        // Load GPIOD MODER register address to r6
64     ldr r5, [r6]                 // Read its content to r5
65     bic r5, #0x3000000           // Clear bits 24, 25 for P12
66     orr r5, #0x01000000          // Write 01 to bits 24, 25 for P12
67     str r5, [r6]                 // Store result in GPIOD MODER register
68
69     // Set GPIOD Pin12 to 1 (bit 12 in ODR register)
70     ldr r6, = GPIOD_ODR          // Load GPIOD output data register
71     ldr r5, [r6]                 // Read its content to r5
72     orr r5, #0x1000               // write 1 to pin 12
73     str r5, [r6]                 // Store result in GPIOD output data register
74
75     // Set GPIOD Pin12 to 0 (bit 12 in ODR register)
76     ldr r6, = GPIOD_ODR          // Load GPIOD output data register
77     ldr r5, [r6]                 // Read its content to r5
78     bic r5, #0x1000              // write 0 to pin 12
79     str r5, [r6]                 // Store result in GPIOD output data register
80
81 loop:
82     nop                          // No operation. Do nothing.
83     b loop                       // Jump to loop
84
```

STM32 CubeIDE, STM32F4 (izbrana dokumentacij

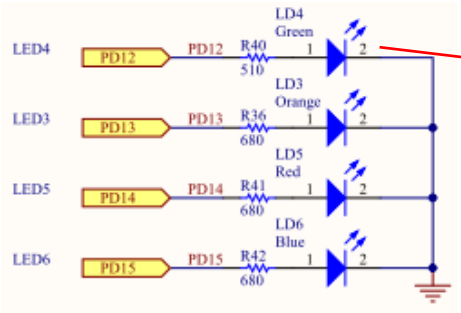
--------- Razvojni sistem ----------

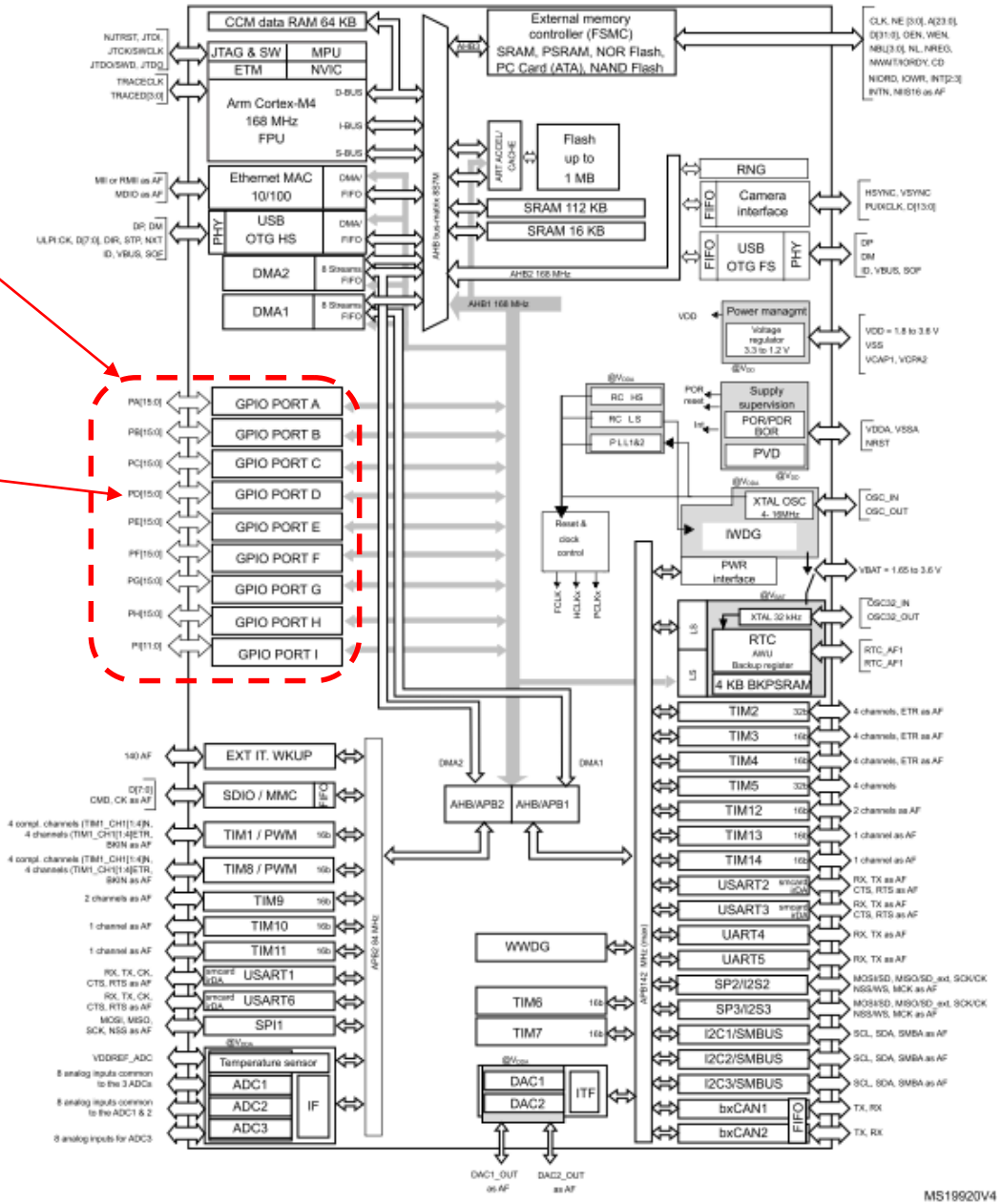
- STM32 CubeIDE
- ORLab-STM32 - GitHub repozitorij
- User Manual Discovery kit stm32f407vg Uploaded 8/11/21, 12.58
- DataSheet_stm32f407vg Uploaded 8/11/21, 12.56
- Reference Manual rm0090-stm32f407417 Uploaded 8/11/21, 12.57
- Programming_Manual_pm0214-stm32-cortexm4-mcus-and-mpu
- Arm Cortex-M4 Processor Datasheet Short Uploaded 29/10/21, 15.00

--------- Cortex-M arhitektura, zbirnik ----------

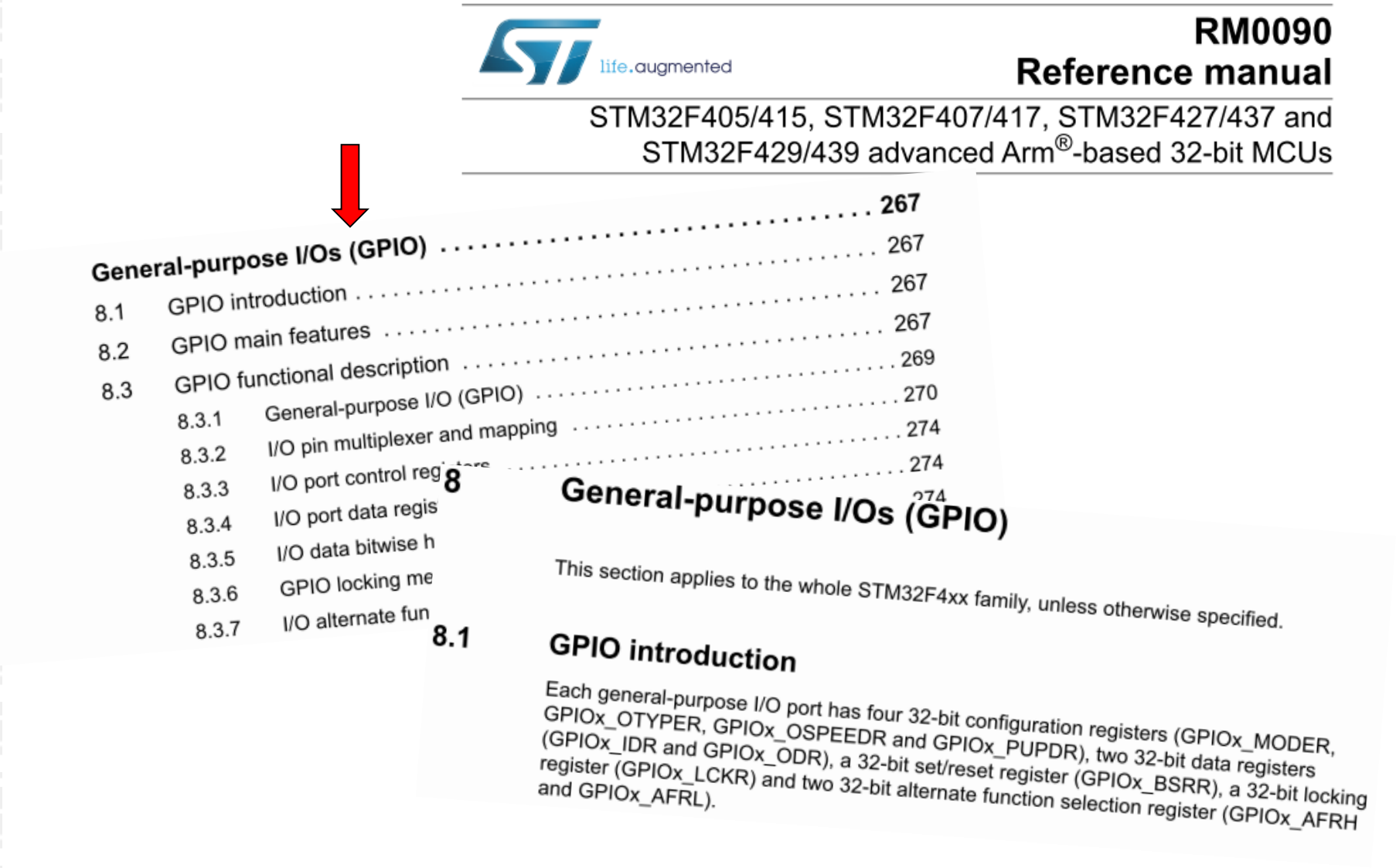
- ARM Cortex-M for Beginners ARM 2017 Uploaded 29/10/21, 14.50

# GPIO Krmilnik



# *Vir: Podatkovna listina Atmel SAM 9260*



life.augmented

**RM0090**
**Reference manual**

STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced Arm®-based 32-bit MCUs



8

## 8 General-purpose I/Os (GPIO)

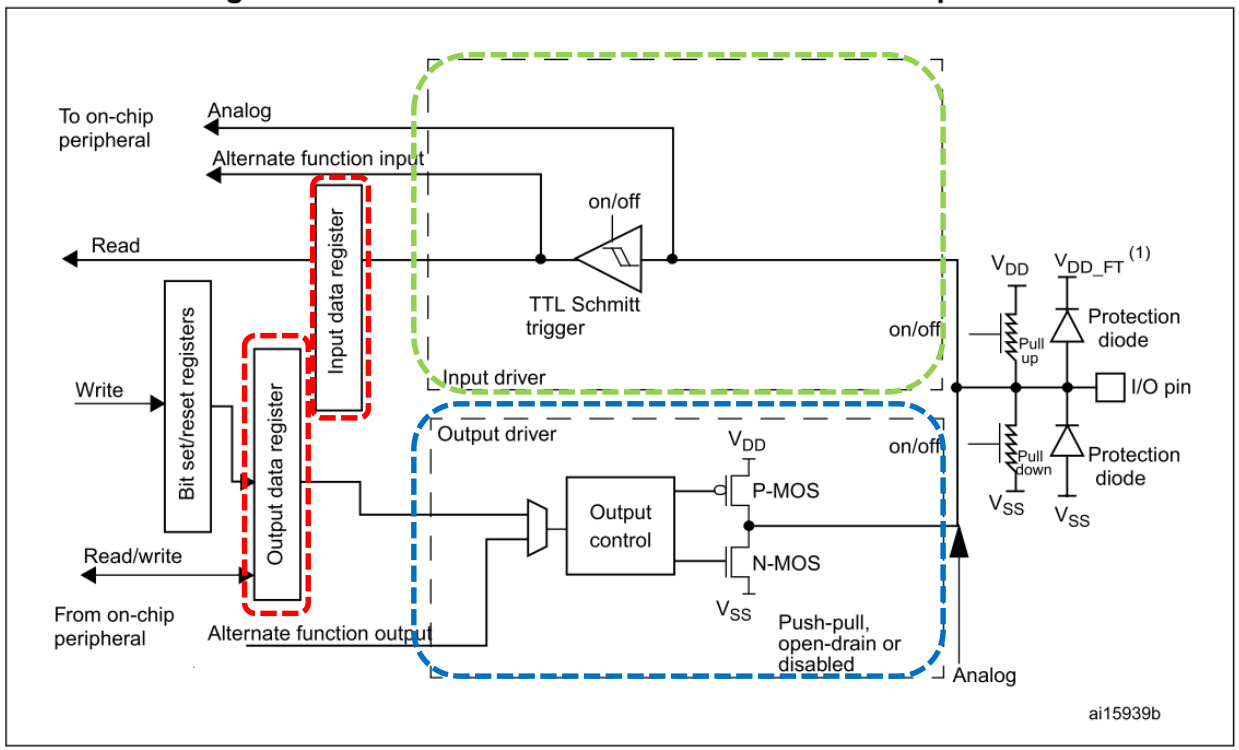
This section applies to the whole STM32F4xx family, unless otherwise specified.

### 8.1 GPIO introduction

Each general-purpose I/O port has four 32-bit configuration registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR), two 32-bit data registers (GPIOx_IDR and GPIOx_ODR), a 32-bit set/reset register (GPIOx_BSRR), a 32-bit locking register (GPIOx_LCKR) and two 32-bit alternate function selection register (GPIOx_AFRH and GPIOx_AFRL).

# GPIO krmilnik – vhod/izhod

Figure 25. Basic structure of a five-volt tolerant I/O port bit




## Osnovni registri za GPIO priključke :

**RCC_AHBxENR : vklop urinega signala (enote) : Port D: RCC_AHB1ENR($b_3$=1 .. Port D Enable)**

**MODER (Mode Register): 00: Input (reset) / 01: General purpose output mode**

**OTYPER (Output TYPE Register): 0: Output push-pull (reset) / 1: Output open-drain**

**OSPEEDR (Output SPEED Register): 00 – Low speed (reset) .. 11: Very high speed**

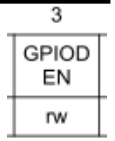
**PUPDR (Pull Up/Down Register): 00 – No pull (reset) .. 01: Pull-Up .. 10: Pull-Down**

**IDR (Input Data Register): stanje vhoda 1 / 0**

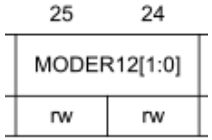
**ODR (Output Data Register): stanje izhoda 1 / 0**

# GPIO krmilnik – izhod *(Registri za nastavitve delovanja)*

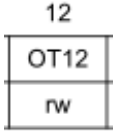
**RCC_AHB1ENR(Peripheral Clock Register):** Bit 3 **GPIODEN:** IO port D clock enable
This bit is set and cleared by software.
0: IO port D clock disabled
1: IO port D clock enabled




**MODER (Mode Register):** Bits 2y:2y+1 **MODERy[1:0]:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O direction mode.
00: Input (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode

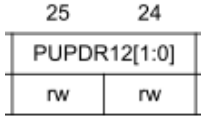



**OTYPER (Output TYPE Register):**
Bits 15:0 **OTy:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the output type of the I/O port.
0: Output push-pull (reset state)
1: Output open-drain

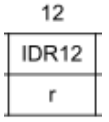



**OSPEEDR (Output SPEED Register):** Bits 2y:2y+1 **OSPEEDRy[1:0]:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O output speed.
00: Low speed
01: Medium speed
10: High speed
11: Very high speed

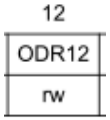

**PUPDR (Pull Up/Down Register):** Bits 2y:2y+1 **PUPDRy[1:0]:** Port x configuration bits (y = 0..15)
These bits are written by software to configure the I/O pull-up or pull-down
00: No pull-up, pull-down
01: Pull-up
10: Pull-down
11: Reserved

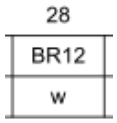



**IDR (Input Data Register): stanje vhoda 1 / 0**
Bits 15:0 **IDRy**: Port input data (y = 0..15)
These bits are read-only and can be accessed in word mode only. They contain the input value of the corresponding I/O port.




**ODR (Output Data Register): stanje izhoda 1 / 0**
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..I/J/K).*

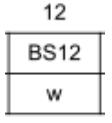



Bits 31:16 **BRy:** Port x reset bit y (y = 0..15)
1: Resets the corresponding ODRx bit

Bits 15:0 **BSy:** Port x set bit y (y= 0..15)
1: Sets the corresponding ODRx bit

0: No action on the corresponding ODRx bit



# GPIO krmilnik – krmiljenje izhodov

**Potrebni koraki za krmiljenje izhoda:**

1. RCC_AHB1ENR(Peripheral Clock Register): $b_3$=1 .. Port D Enable
2. **MODER (Mode Register): 01: General purpose output mode**
3. Default vrednosti že ustrezne v registrih :
   **OTYPER (Output TYPE Register): 0: Output push-pull (reset)**
   **OSPEEDR (Output SPEED Register): 00 – Low speed (reset)**
   **PUPDR (Pull Up/Down Register): 00 – No pull (reset)**
4. določi stanje izhoda s pisanjem v ODR ali BSRR (nastavljamo na 1/0)

**Naslovi registrov:**

```
// RCC   base address is 0x40023800
//   AHB1ENR register offset is 0x30
.equ     RCC_AHB1ENR,   0x40023830 // RCC AHB1 peripheral clock reg (page 180)

// GPIOD base address is 0x40020C00
//   MODER register offset is 0x00
//   ODR   register offset is 0x14
.equ     GPIOD_MODER,   0x40020C00 // GPIOD port mode register (page 281)
.equ     GPIOD_ODR,     0x40020C14 // GPIOD output data register (page 283)
```

# CubeIDE – Registers okno

| Name | Value |
|---|---|
| General Registers | |
| r0 | 0x0 |
| r1 | 0x0 |
| r2 | 0x0 |
| r3 | 0x0 |
| r4 | 0x0 |
| r5 | 0x1000 |
| r6 | 0x40020c14 |
| r7 | 0x0 |
| r8 | 0x0 |
| r9 | 0x0 |
| r10 | 0x0 |
| r11 | 0x0 |
| r12 | 0x0 |
| sp | 0x20020000 |
| lr | 0xffffffff |
| pc | 0x800002a |
| xpsr | 0x41000000 |

# CubeIDE – SFR okno

| Register | Address | Value |
|---|---|---|
| GPIOF | | |
| GPIOE | | |
| GPIOD | | |
| MODER | 0x40020c00 | 0x1000000 |
| OTYPER | 0x40020c04 | 0x0 |
| OSPEEDR | 0x40020c08 | 0x0 |
| PUPDR | 0x40020c0c | 0x0 |
| IDR | 0x40020c10 | 0x14ac |
| ODR | 0x40020c14 | 0x1000 |
| BSRR | 0x40020c18 | |
| LCKR | 0x40020c1c | 0x0 |
| AFRL | 0x40020c20 | 0x0 |
| AFRH | 0x40020c24 | 0x0 |
| GPIOC | | |
| GPIOJ | | |
| GPIOK | | |
| GPIOB | | |
| GPIOA | | |
| SYSCFG | | |
| SPI1 | | |
| SPI2 | | |
| SPI3 | | |