11 General-purpose I/Os (GPIO)

11.1 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 registers (GPIOx_AFRH and GPIOx_AFRL).

11.2 GPIO main features

- Output states: push-pull or open drain + pull-up/down
- Output data from output data register (GPIOx_ODR) or peripheral (alternate function output)
- Speed selection for each I/O
- Input states: floating, pull-up/down, analog
- Input data to input data register (GPIOx_IDR) or peripheral (alternate function input)
- Bit set and reset register (GPIOx_BSRR) for bitwise write access to GPIOx_ODR
- Locking mechanism (GPIOx_LCKR) provided to freeze the port A, B, C, D, E, F, G, H on STM32F303xD/E and STM32F398xE, A, B and D I/O configuration in STM32F303xB/C and STM32F358xC devices and port A, B, C, D and F in STM32F303x6/8 and STM32F328x8 devices.
- Analog function
- Alternate function selection registers
- Fast toggle capable of changing every two clock cycles
- Highly flexible pin multiplexing allows the use of I/O pins as GPIOs or as one of several peripheral functions

11.3 GPIO functional description

Subject to the specific hardware characteristics of each I/O port listed in the datasheet, each port bit of the general-purpose I/O (GPIO) ports can be individually configured by software in several modes:

- Input floating
- Input pull-up
- Input-pull-down
- Analog
- Output open-drain with pull-up or pull-down capability
- Output push-pull with pull-up or pull-down capability
- Alternate function push-pull with pull-up or pull-down capability
- Alternate function open-drain with pull-up or pull-down capability

Each I/O port bit is freely programmable, however the I/O port registers have to be accessed as 32-bit words, half-words or bytes. The purpose of the GPIOx_BSRR register is
to allow atomic read/modify accesses to any of the GPIO\textsubscript{x} ODR registers. In this way, there is no risk of an IRQ occurring between the read and the modify access.

*Figure 40* and *Figure 41* show the basic structures of a standard and a 5-Volt tolerant I/O port bit, respectively. *Table 72* gives the possible port bit configurations.

1. \(V_{\text{DD_FT}}\) is a potential specific to 5-Volt tolerant I/Os and different from \(V_{\text{DD}}\).
### Table 72. Port bit configuration table(1)

<table>
<thead>
<tr>
<th>MODER(i) [1:0]</th>
<th>OTYPE(i)</th>
<th>OSPEEDR(i) [1:0]</th>
<th>PUPDR(i) [1:0]</th>
<th>I/O configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0</td>
<td></td>
<td>0 0</td>
<td>GP output PP</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>0 1</td>
<td>GP output PP + PU</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>1 0</td>
<td>GP output PP + PD</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>1 1</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>SPEED [1:0]</td>
<td>0 0</td>
<td>GP output OD</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0 1</td>
<td>GP output OD + PU</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>1 0</td>
<td>GP output OD + PD</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>1 1</td>
<td>Reserved (GP output OD)</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td></td>
<td>0 0</td>
<td>AF PP</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>0 1</td>
<td>AF PP + PU</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>1 0</td>
<td>AF PP + PD</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>1 1</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>SPEED [1:0]</td>
<td>0 0</td>
<td>AF OD</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>0 1</td>
<td>AF OD + PU</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>1 0</td>
<td>AF OD + PD</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td>1 1</td>
<td>Reserved</td>
</tr>
<tr>
<td>00</td>
<td>x</td>
<td>x x x</td>
<td>0 0</td>
<td>Input Floating</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x x x</td>
<td>0 1</td>
<td>Input PU</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x x x</td>
<td>1 0</td>
<td>Input PD</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x x x</td>
<td>1 1</td>
<td>Reserved (input floating)</td>
</tr>
<tr>
<td>11</td>
<td>x</td>
<td>x x x</td>
<td>0 0</td>
<td>Input/output Analog</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x x x</td>
<td>0 1</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x x x</td>
<td>1 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x x x</td>
<td>1 1</td>
<td></td>
</tr>
</tbody>
</table>

1. GP = general-purpose, PP = push-pull, PU = pull-up, PD = pull-down, OD = open-drain, AF = alternate function.
11.3.1 General-purpose I/O (GPIO)

During and just after reset, the alternate functions are not active and most of the I/O ports are configured in input floating mode.

The debug pins are in AF pull-up/pull-down after reset:
- PA15: JTDI in pull-up
- PA14: JTCK/SWCLK in pull-down
- PA13: JTMS/SWDIO in pull-up
- PB4: NJTRST in pull-up
- PB3: JTD0/TRACESWO

When the pin is configured as output, the value written to the output data register (GPIOx_ODR) is output on the I/O pin. It is possible to use the output driver in push-pull mode or open-drain mode (only the low level is driven, high level is HI-Z).

The input data register (GPIOx_IDR) captures the data present on the I/O pin at every AHB clock cycle.

All GPIO pins have weak internal pull-up and pull-down resistors, which can be activated or not depending on the value in the GPIOx_PUPDR register.

11.3.2 I/O pin alternate function multiplexer and mapping

The device I/O pins are connected to on-board peripherals/modules through a multiplexer that allows only one peripheral alternate function (AF) connected to an I/O pin at a time. In this way, there can be no conflict between peripherals available on the same I/O pin.

Each I/O pin has a multiplexer with up to sixteen alternate function inputs (AF0 to AF15) that can be configured through the GPIOx_AFRL (for pin 0 to 7) and GPIOx_AFRH (for pin 8 to 15) registers:
- After reset the multiplexer selection is alternate function 0 (AF0). The I/Os are configured in alternate function mode through GPIOx_MODER register.
- The specific alternate function assignments for each pin are detailed in the device datasheet.

In addition to this flexible I/O multiplexing architecture, each peripheral has alternate functions mapped onto different I/O pins to optimize the number of peripherals available in smaller packages.

To use an I/O in a given configuration, the user has to proceed as follows:
- **Debug function**: after each device reset these pins are assigned as alternate function pins immediately usable by the debugger host
- **GPIO**: configure the desired I/O as output, input or analog in the GPIOx_MODER register.
- **Peripheral alternate function**:
  - Connect the I/O to the desired AFx in one of the GPIOx_AFRL or GPIOx_AFRH register.
  - Select the type, pull-up/pull-down and output speed via the GPIOx_OTYPER, GPIOx_PUPDR and GPIOx_OSPEEDER registers, respectively.
– Configure the desired I/O as an alternate function in the GPIOx_MODER register.

**Additional functions:**
– For the ADC, DAC, OPAMP and COMP, configure the desired I/O in analog mode in the GPIOx_MODER register and configure the required function in the ADC, DAC, OPAMP, and COMP registers.
– For the additional functions like RTC, WKUPx and oscillators, configure the required function in the related RTC, PWR and RCC registers. These functions have priority over the configuration in the standard GPIO registers.

Refer to the "Alternate function mapping" table in the device datasheet for the detailed mapping of the alternate function I/O pins.

### 11.3.3 I/O port control registers

Each of the GPIO ports has four 32-bit memory-mapped control registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR) to configure up to 16 I/Os. The GPIOx_MODER register is used to select the I/O mode (input, output, AF, analog). The GPIOx_OTYPER and GPIOx_OSPEEDR registers are used to select the output type (push-pull or open-drain) and speed. The GPIOx_PUPDR register is used to select the pull-up/pull-down whatever the I/O direction.

### 11.3.4 I/O port data registers

Each GPIO has two 16-bit memory-mapped data registers: input and output data registers (GPIOx_IDR and GPIOx_ODR). GPIOx_ODR stores the data to be output, it is read/write accessible. The data input through the I/O are stored into the input data register (GPIOx_IDR), a read-only register.

See [Section 11.4.5: GPIO port input data register (GPIOx_IDR) (x = A to H)] and [Section 11.4.6: GPIO port output data register (GPIOx_ODR) (x = A to H)] for the register descriptions.

### 11.3.5 I/O data bitwise handling

The bit set reset register (GPIOx_BSRR) is a 32-bit register which allows the application to set and reset each individual bit in the output data register (GPIOx_ODR). The bit set reset register has twice the size of GPIOx_ODR.

To each bit in GPIOx_ODR, correspond two control bits in GPIOx_BSRR: BS(i) and BR(i). When written to 1, bit BS(i) sets the corresponding ODR(i) bit. When written to 1, bit BR(i) resets the ODR(i) corresponding bit.

Writing any bit to 0 in GPIOx_BSRR does not have any effect on the corresponding bit in GPIOx_ODR. If there is an attempt to both set and reset a bit in GPIOx_BSRR, the set action takes priority.

Using the GPIOx_BSRR register to change the values of individual bits in GPIOx_ODR is a "one-shot" effect that does not lock the GPIOx_ODR bits. The GPIOx_ODR bits can always be accessed directly. The GPIOx_BSRR register provides a way of performing atomic bitwise handling.

There is no need for the software to disable interrupts when programming the GPIOx_ODR at bit level: it is possible to modify one or more bits in a single atomic AHB write access.
11.3.6 GPIO locking mechanism

It is possible to freeze the GPIO control registers by applying a specific write sequence to the GPIOx_LCKR register. The frozen registers are GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRL and GPIOx_AFRH.

To write the GPIOx_LCKR register, a specific write / read sequence has to be applied. When the right LOCK sequence is applied to bit 16 in this register, the value of LCKR[15:0] is used to lock the configuration of the I/Os (during the write sequence the LCKR[15:0] value must be the same). When the LOCK sequence has been applied to a port bit, the value of the port bit can no longer be modified until the next MCU reset or peripheral reset. Each GPIOx_LCKR bit freezes the corresponding bit in the control registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRL and GPIOx_AFRH).

The LOCK sequence (refer to Section 11.4.8: GPIO port configuration lock register (GPIOx_LCKR)) can only be performed using a word (32-bit long) access to the GPIOx_LCKR register due to the fact that GPIOx_LCKR bit 16 has to be set at the same time as the [15:0] bits.

For more details refer to LCKR register description in Section 11.4.8: GPIO port configuration lock register (GPIOx_LCKR).

11.3.7 I/O alternate function input/output

Two registers are provided to select one of the alternate function inputs/outputs available for each I/O. With these registers, the user can connect an alternate function to some other pin as required by the application.

This means that a number of possible peripheral functions are multiplexed on each GPIO using the GPIOx_AFRL and GPIOx_AFRH alternate function registers. The application can thus select any one of the possible functions for each I/O. The AF selection signal being common to the alternate function input and alternate function output, a single channel is selected for the alternate function input/output of a given I/O.

To know which functions are multiplexed on each GPIO pin refer to the device datasheet.

11.3.8 External interrupt/wake-up lines

All ports have external interrupt capability. To use external interrupt lines, the port must be configured in input mode.

Refer to Section 14.2: Extended interrupts and events controller (EXTI) and to Section 14.2.3: Wake-up event management.

11.3.9 Input configuration

When the I/O port is programmed as input:

- The output buffer is disabled
- The Schmitt trigger input is activated
- The pull-up and pull-down resistors are activated depending on the value in the GPIOx_PUPDR register
- The data present on the I/O pin are sampled into the input data register every AHB clock cycle
- A read access to the input data register provides the I/O state
Figure 42 shows the input configuration of the I/O port bit.

Figure 42. Input floating / pull up / pull down configurations

11.3.10 Output configuration

When the I/O port is programmed as output:
- The output buffer is enabled:
  - Open drain mode: a “0” in the output register activates the N-MOS whereas a “1” in the output register leaves the port in Hi-Z (the P-MOS is never activated)
  - Push-pull mode: a “0” in the output register activates the N-MOS whereas a “1” in the output register activates the P-MOS
- The Schmitt trigger input is activated
- The pull-up and pull-down resistors are activated depending on the value in the GPIOx_PUPDR register
- The data present on the I/O pin are sampled into the input data register every AHB clock cycle
- A read access to the input data register gets the I/O state
- A read access to the output data register gets the last written value
Figure 43 shows the output configuration of the I/O port bit.

**Figure 43. Output configuration**

11.3.11 **Alternate function configuration**

When the I/O port is programmed as alternate function:

- The output buffer can be configured in open-drain or push-pull mode
- The output buffer is driven by the signals coming from the peripheral (transmitter enable and data)
- The Schmitt trigger input is activated
- The weak pull-up and pull-down resistors are activated or not depending on the value in the GPIOx_PUPDR register
- The data present on the I/O pin are sampled into the input data register every AHB clock cycle
- A read access to the input data register gets the I/O state
**11.3.12 Analog configuration**

When the I/O port is programmed as analog configuration:
- The output buffer is disabled
- The Schmitt trigger input is deactivated, providing zero consumption for every analog value of the I/O pin. The output of the Schmitt trigger is forced to a constant value (0).
- The weak pull-up and pull-down resistors are disabled by hardware
- Read access to the input data register gets the value “0”

*Figure 45* shows the high-impedance, analog-input configuration of the I/O port bits.
11.3.13 Using the HSE or LSE oscillator pins as GPIOs

When the HSE or LSE oscillator is switched OFF (default state after reset), the related oscillator pins can be used as normal GPIOs.

When the HSE or LSE oscillator is switched ON (by setting the HSEON or LSEON bit in the RCC_CSR register) the oscillator takes control of its associated pins and the GPIO configuration of these pins has no effect.

When the oscillator is configured in a user external clock mode, only the pin is reserved for clock input and the OSC_OUT or OSC32_OUT pin can still be used as normal GPIO.

11.3.14 Using the GPIO pins in the RTC supply domain

The PC13/PC14/PC15 GPIO functionality is lost when the core supply domain is powered off (when the device enters Standby mode). In this case, if their GPIO configuration is not bypassed by the RTC configuration, these pins are set in an analog input mode.

For details about I/O control by the RTC, refer to Section 27.3: RTC functional description.
11.4 GPIO registers

For a summary of register bits, register address offsets and reset values, refer to Table 73.

The peripheral registers can be written in word, half word or byte mode.

11.4.1 GPIO port mode register (GPIOx_MODER) 
(x = A to H)

Address offset: 0x00

Reset value: 0xA800 0000 for port A

Reset value: 0x0000 0280 for port B

Reset value: 0x0000 0000 for other ports

<table>
<thead>
<tr>
<th>31</th>
<th>30</th>
<th>29</th>
<th>28</th>
<th>27</th>
<th>26</th>
<th>25</th>
<th>24</th>
<th>23</th>
<th>22</th>
<th>21</th>
<th>20</th>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Bits 31:0 MODER[15:0][1:0]: Port x configuration I/O pin y (y = 15 to 0)
These bits are written by software to configure the I/O mode.
00: Input mode (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode

Note: In STM32F303xB/xC and STM32F358x devices, bits 10 and 11 of GPIOF_MODER are reserved and must be kept at reset state.

11.4.2 GPIO port output type register (GPIOx_OTYPER) 
(x = A to H)

Address offset: 0x04

Reset value: 0x0000 0000

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

rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw rw
11.4.3 GPIO port output speed register (GPIOx_OSPEEDR) (x = A to H)

Address offset: 0x08

Reset value: 0x0C00 0000 (for port A)
Reset value: 0x0000 00C0 (for port B)
Reset value: 0x0000 0000 (for other ports)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
<tr>
<td>15:0</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
<tr>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
<tr>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
</tbody>
</table>

Bits 31:0 **OSPEEDR[15:0][1:0]**: Port x configuration I/O pin y (y = 15 to 0)
These bits are written by software to configure the I/O output speed.
- x0: Low speed
- 01: Medium speed
- 11: High speed

*Note: Refer to the device datasheet for the frequency specifications and the power supply and load conditions for each speed.*

11.4.4 GPIO port pull-up/pull-down register (GPIOx_PUPDR) (x = A to H)

Address offset: 0x0C

Reset value: 0x6400 0000 (for port A)
Reset value: 0x0000 0100 (for port B)
Reset value: 0x0000 0000 (for other ports)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
<tr>
<td>29:28</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
<tr>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
<tr>
<td></td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
</tbody>
</table>

Bits 31:0 **PUPDR[15:0][1:0]**: Port x configuration I/O pin y (y = 15 to 0)
These bits are written by software to configure the I/O pull-ups/pull-downs.

- 00: No pull-ups/pull-downs
- 01: Pull-up
- 10: Pull-down
- 11: Pull-up and pull-down enabled
11.4.5 GPIO port input data register (GPIOx_IDR) 
(x = A to H)

Address offset: 0x10
Reset value: 0x0000 XXXX

Bits 31:0 PUPDR[15:0][1:0]: Port x configuration I/O pin y (y = 15 to 0)
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

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

Bits 15:0 IDR[15:0]: Port x input data I/O pin y (y = 15 to 0)
These bits are read-only. They contain the input value of the corresponding I/O port.

11.4.6 GPIO port output data register (GPIOx_ODR) 
(x = A to H)

Address offset: 0x14
Reset value: 0x0000 0000

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

Bits 15:0 ODR[15:0]: Port output data I/O pin y (y = 15 to 0)
These bits can be read and written by software.

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

Address offset: 0x18
Reset value: 0x0000 0000

Bits 31:16 **BR[15:0]**: Port x reset I/O pin y (y = 15 to 0)
These bits are write-only. A read to these bits returns the value 0x0000.
0: No action on the corresponding ODRx bit
1: Resets the corresponding ODRx bit

Note: If both BSx and BRx are set, BSx has priority.

Bits 15:0 **BS[15:0]**: Port x set I/O pin y (y = 15 to 0)
These bits are write-only. A read to these bits returns the value 0x0000.
0: No action on the corresponding ODRx bit
1: Sets the corresponding ODRx bit

11.4.8 GPIO port configuration lock register (GPIOx_LCKR)

x = A, B and D in STM32F303xB/C and STM32F358xC devices, x = A, B, C, D and F in
STM32F303x6/8 and STM32F328x8 devices and x = A, B, C, D, E, F, G, H in
STM32F303xD/E devices.

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

Note: A specific write sequence is used to write to the GPIOx_LCKR register. Only word access
(32-bit long) is allowed during this locking sequence.

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

Address offset: 0x1C
Reset value: 0x0000 0000
11.4.9 GPIO alternate function low register (GPIOx_AFRL) (x = A to H)

Address offset: 0x20
Reset value: 0x0000 0000

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

Bit 16 **LCKK**: Lock key

This bit can be read any time. It can only be modified using the lock key write sequence.
0: Port configuration lock key not active
1: Port configuration lock key active. The GPIOx_LCKR register is locked until the next MCU reset or peripheral reset.

LOCK key write sequence:
- WR LCKR[16] = 1 + LCKR[15:0]
- WR LCKR[16] = 0 + LCKR[15:0]
- WR LCKR[16] = 1 + LCKR[15:0]
- RD LCKR

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

Note: During the LOCK key write sequence, the value of LCK[15:0] must not change.
Any error in the lock sequence aborts the lock.
After the first lock sequence on any bit of the port, any read access on the LCKK bit returns 1 until the next MCU reset or peripheral reset.

Bits 15:0 **LCK[15:0]**: Port x lock I/O pin y (y = 15 to 0)

These bits are read/write but can only be written when the LCKK bit is 0.
0: Port configuration not locked
1: Port configuration locked

**11.4.9 GPIO alternate function low register (GPIOx_AFRL) (x = A to H)**

Address offset: 0x20
Reset value: 0x0000 0000

<table>
<thead>
<tr>
<th>31</th>
<th>30</th>
<th>29</th>
<th>28</th>
<th>27</th>
<th>26</th>
<th>25</th>
<th>24</th>
<th>23</th>
<th>22</th>
<th>21</th>
<th>20</th>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFR3[3:0]</td>
<td>AFR2[3:0]</td>
<td>AFR1[3:0]</td>
<td>AFR0[3:0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
</tr>
</tbody>
</table>

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

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

AFRy selection:
- 0000: AF0
- 0001: AF1
- 0010: AF2
- 0011: AF3
- 0100: AF4
- 0101: AF5
- 0110: AF6
- 0111: AF7
- 1000: AF8
- 1001: AF9
- 1010: AF10
- 1011: AF11
- 1100: AF12
- 1101: AF13
- 1110: AF14
- 1111: AF15
11.4.10  GPIO alternate function high register (GPIOx_AFRH)  
(x = A to H)

Address offset: 0x24  
Reset value: 0x0000 0000

<table>
<thead>
<tr>
<th>31</th>
<th>30</th>
<th>29</th>
<th>28</th>
<th>27</th>
<th>26</th>
<th>25</th>
<th>24</th>
<th>23</th>
<th>22</th>
<th>21</th>
<th>20</th>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td>rw</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bits 31:0  AFRy[3:0]: Alternate function selection for port x pin y (y = 8..15)  
These bits are written by software to configure alternate function I/Os

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

11.4.11  GPIO port bit reset register (GPIOx_BRR) (x = A to H)

Address offset: 0x28  
Reset value: 0x0000 0000

<table>
<thead>
<tr>
<th>31</th>
<th>30</th>
<th>29</th>
<th>28</th>
<th>27</th>
<th>26</th>
<th>25</th>
<th>24</th>
<th>23</th>
<th>22</th>
<th>21</th>
<th>20</th>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR15</td>
<td>BR14</td>
<td>BR13</td>
<td>BR12</td>
<td>BR11</td>
<td>BR10</td>
<td>BR9</td>
<td>BR8</td>
<td>BR7</td>
<td>BR6</td>
<td>BR5</td>
<td>BR4</td>
<td>BR3</td>
<td>BR2</td>
<td>BR1</td>
<td>BR0</td>
</tr>
<tr>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
<td>w</td>
</tr>
</tbody>
</table>

Bits 31:16  Reserved, must be kept at reset value.  
Bits 15:0  BR[15:0]: Port x reset IO pin y (y = 15 to 0)  
These bits are write-only. A read to these bits returns the value 0x0000.  
0: No action on the corresponding ODx bit  
1: Reset the corresponding ODx bit
### 11.4.12 GPIO register map

The following table gives the GPIO register map and reset values.

| Offset | Register name         |   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   | GPIOA_MODER           | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | 1010100000000000 |
| 0x00   | GPIOB_MODER           | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | 0000000000000000 |
| 0x00   | GPIOx_MODER           | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | MODER | 0000000000000000 |
| 0x04   | GPIOx_OTYPER          | Res.  | Res.  | Res.  | Res.  | Res.  | Res.  | Res.  | OT15  | OT14  | OT13  | OT12  | OT11  | OT10  | OT09  | OT08  | OT07  | OT06  | OT05  | OT04  | OT03  | OT02  | OT01  | OT00  | 0000000000000000 |
| 0x08   | GPIOA_OSPEEDR         | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | 0000110000000000 |
| 0x08   | GPIOB_OSPEEDR         | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | 0000000000000000 |
| 0x08   | GPIOx_OSPEEDR         | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | OSPEED | 0000000000000000 |
| 0x0C   | GPIOA_PUPDR           | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | PUPDR | 0111000000000000 |
### Table 73. GPIO register map and reset values (continued)

| Offset | Register name          | 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  |
|--------|------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0x0C   | GPIOB_PUPDR            |    |    |    |    |    |    |    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
|        | Reset value            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0x10   | GPIOx_IDR (where x = A..H) | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
|        | Reset value            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0x14   | GPIOx_ODR (where x = A..H) | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
|        | Reset value            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0x18   | GPIOx_BSRR (where x = A..H) | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
|        | Reset value            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0x1C   | GPIOx_LCKR (where x = see (1)) | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
|        | Reset value            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0x20   | GPIOx_AFRL (where x = C..H) | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
|        | Reset value            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0x24   | GPIOx_AFRH (where x = A..H) | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
|        | Reset value            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0x28   | GPIOx_BRR (where x = A..H) | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
|        | Reset value            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

1. A, B and D in STM32F303xB/C and STM32F358xC, A, B, C, D and F in STM32F303x6/8 and STM32F328x8, and A, B, C, D, E, F, G, H in STM32F303xD/E.

Refer to Section 3.2 on page 53 for the register boundary addresses.