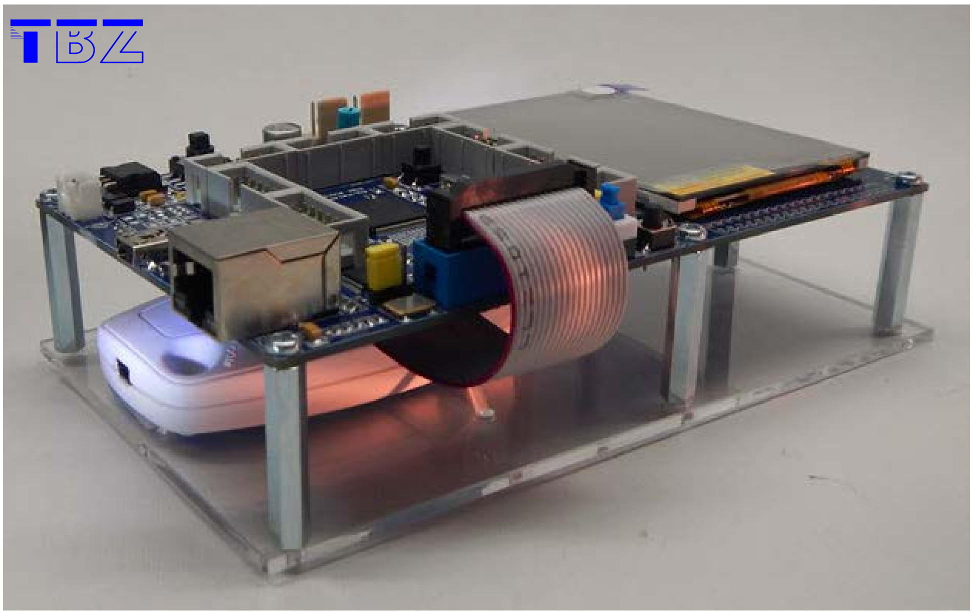
MCB32 APP
Anwendung GPIO mit MCB32



MCB32 - Embedded Programmierung

Version: **1722.00**

Bitte beachten. Diese Unterlagen können ohne Vorankündigung jederzeit angepasst, verbessert und erweitert werden. Wir bitten Sie Wünsche und auch Fehler zu melden. (info@mcb32.ch)

1 Inhaltsverzeichnis

3 Einleitung Beschreibung Port GPIOA … GPIOE

Der Mikrocontroller STM32F107VC verfügt über 5 IO-Ports mit 16Bits. Aber aufgepasst. Jeder Pin kann mehr als eine Funktion übernehmen. Das führt dazu, dass viele der Pins für die interne Hardwarestruktur verdrahtet sind und je nach Anwendung dann nicht mehr zur Verfügung stehen. Beispiel: wenn die SD-Karte im Einsatz ist, sind einige Pins von Port A dafür gebraucht worden und stehen dem Nutzer nicht mehr zur Verfügung.

3.1 Struktur eines GPIO Pins [1], [2]

Da wird es nun interessant. Wie bekannt haben heutige MCU auf jedem Pin mehrere Funktionen. Wenn wir nun aber nur die GPIO (General Purpose Input Output) ansehen stehen wir einer komplexen Hardware gegenüber.

1. Struktur eines Port Pins (x= A … E) [1]

Das folgende Blockschema [1] zeigt nun einen Port Pin mit seiner komplexen Schaltung. Die Funktion des Pins wird über die unten beschriebenen Registersätze ausgewählt resp. muss entsprechend programmiert werden. Die Library (CMSIS = Cortex Mikrocontroller Software Interface Standard) unterstützt den Benutzer aber mit entsprechenden Funktionen. [3]

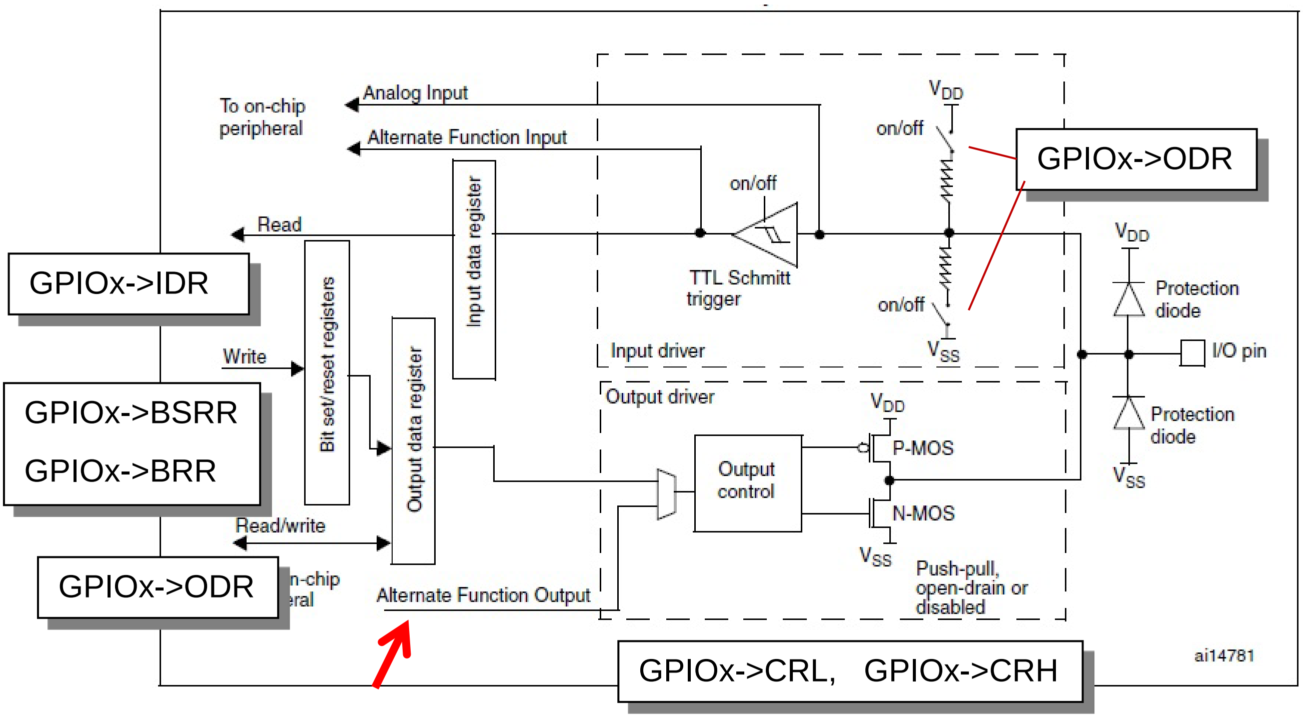


Abbildung 1: Basic structure of a five-volt tolerant I/O port bit.

Jeder Pin kann unabhängig über die Portregister individuell konfiguriert werden.:

- Input floaing
- Input pull-up
- Input pull-down
- Analog Input
- Output open-drain
- Output push-pull
- Alternate function push-pull (siehe Abb. 1)
- Alternate function open-drain (siehe Abb. 1)

In [4, p. 159ff] ist die Programmierung der Port im Detail beschrieben. Die folgende Tabelle ist wichtig um zu verstehen, wie die Eingänge und Ausgänge bezüglich Widerstand und Open-Drain in Abbildung 1 zu programmieren sind.

Im markierten Beispiel erhält ein Input einen Pull-Up oder Pull-Down Widerstand. Die Register CNF1 und CNF0 sind 1 und 0. Mode ist 00 infolge der Input-Programmierung. Via das ODR-Register kommt die Information ob der Pull-Up- oder Pull-Down Widerstand eingeschaltet ist. Siehe folgende

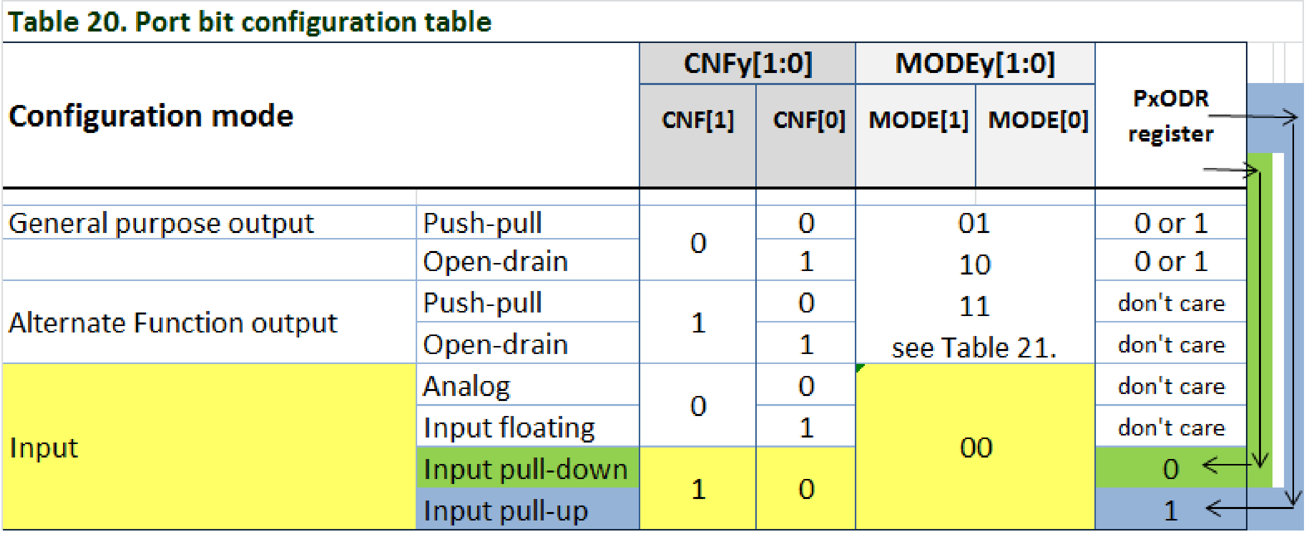


Table 20. Port bit configuration table

Configuration mode		CNFy[1:0]		MODEy[1:0]		PxODR register
		CNF[1]	CNF[0]	MODE[1]	MODE[0]	
General purpose output	Push-pull	0	0	01		0 or 1
	Open-drain		1	10		0 or 1
Alternate Function output	Push-pull	1	0	11		don't care
	Open-drain		1	see Table 21.		don't care
Input	Analog	0	0	00		don't care
	Input floating		1			don't care
	Input pull-down	1	0			0
	Input pull-up					1

Abbildung 2 (Table 20. Port Bit configuration table)

Zu beachten ist die Konfiguration des ODR Registers. Dort wird definiert ob der Pull-Up- oder der Pull-Down Widerstand eingeschaltet wird.

2. Beispiel mit einfachem Setup:

Aufgabe:

An Port A Bit 0…7 ist eine Schalterbox angeschlossen. Diese Schalter sollen an den LEDs an Port E Bit 8…15 angezeigt werden.
Die Ports sollen gemäss Datenblatt programmiert werden. Siehe dazu [4, p. 171] (1128).

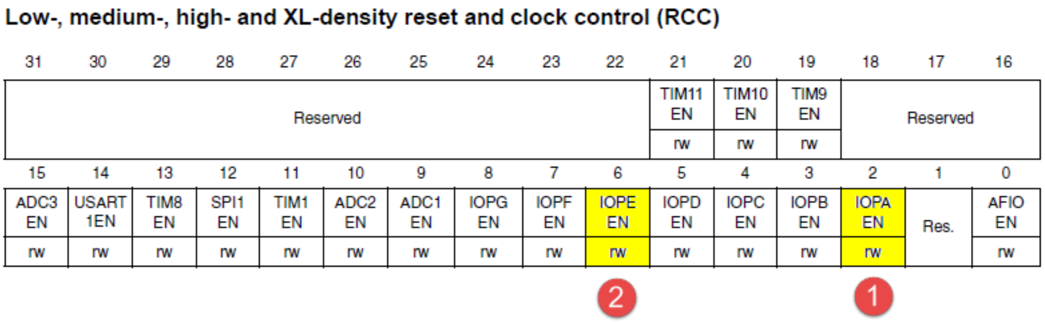
Als Idee schreiben wir zuerst eine Init Sequenz oder in unserem Fall eine lokale „system_init-Funktion“ in welcher das Setup der Ports vorgenommen wird.

Achtung: jeder Port benötigt einen eigenen Clock. Dieser Clock muss zuerst programmiert werden. [4, p. 112]. Ohne Clock liefern die Ports keine Antwort.

Also: wir müssen den Clock von Port A und Port E einschalten.

① RCC-> APB2EN |= 1<<2; schaltet den Clock Port A ein. usw.

Im Manual sieht das wie folgt aus: bei ① wird das Bit gesetzt und dadurch der Port A eingeschaltet. Bei (2) das Gleiche für Port E.



Low-, medium-, high- and XL-density reset and clock control (RCC)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TIM11 EN	TIM10 EN	TIM9 EN	Reserved		
										rw	rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3 EN	USART 1EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	IOPG EN	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw
									②				①		

Lösung:

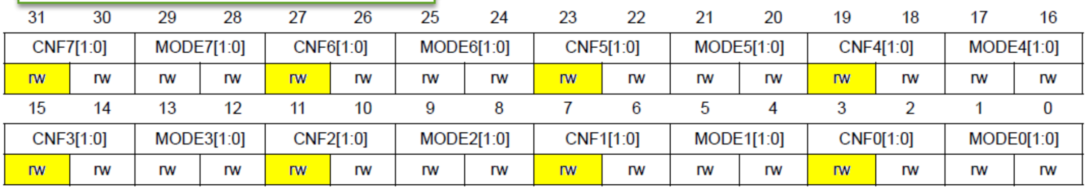
Der Code für unser „system_init“ sieht dann wie folgt aus.

```
void system_init()
 {
  RCC ->APB2ENR |= 1<<2;    // GPIOA Clock [1]
  RCC ->APB2ENR |= 1<<6;     // GPIOE Clock [2]
  GPIOA ->CRL = 0x88888888; // PA7..0 als Input. Nun definieren [3]
                            // ob wir Pull-Up oder Pull Down wollen.
  GPIOA ->ODR = 0x00FF;     // PA7..0 als Pull-Up  [4]
  GPIOE->CRH  = 0x11111111; // PE15-8 Output, Push Pull 10MHz  [5]
 }
```

Die beiden ersten Zeilen [1..2] sind nun klar. In der Zeile [3] sagen wir der Logik, dass Port A im Bereich von Bit 0…7 Eingang sein soll. [4, p. 171]. Der folgende Ausschnitt zeigt die gesetzten Bits im *„Port configuration register low (GPIOx_CRL) (x=A..G)“*

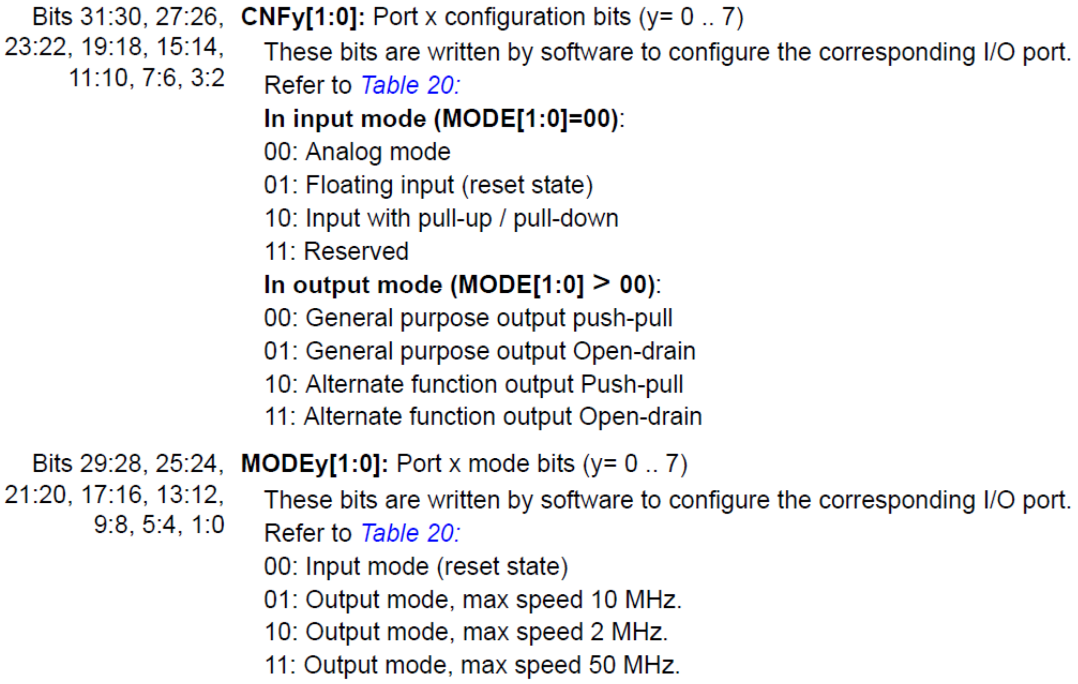
Auf der nächsten Seite sind die Bits des „Port configuration Registers“ näher erklärt.

GPIOA ->CRL = 0x88888888;



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]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	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]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Lösung ff:



Bits 31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2 **CNFy[1:0]:** Port x configuration bits (y= 0 .. 7)

These bits are written by software to configure the corresponding I/O port.
Refer to *Table 20:*

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, 21:20, 17:16, 13:12, 9:8, 5:4, 1:0 **MODEy[1:0]:** Port x mode bits (y= 0 .. 7)

These bits are written by software to configure the corresponding I/O port.
Refer to *Table 20:*
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.

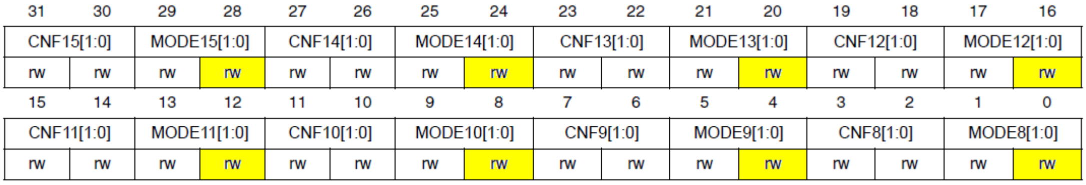
Mit dem Befehl „GPIOA ->ODR = 0x00FF“ wird der Eingang mit einem Pull-UP Widerstand beschaltet. ODRx=0 bewirkt PullDown und ODRx=1 PullUp.

Die einzelnen Bits und ihre Bedeutung sind oben über die Portx-configuration Bits CNFy[1:0] und die Port mode Bits MODEy[1:0] genauer spezifiziert.

Siehe auch Tabelle 20 in diesem Dokument. Diese Tabelle fasst die Zusammenhänge übersichtlich zusammen.
Das hier gesagte gilt auch für das High-Register via GPIOx-> CRH. Siehe unten.

Nun müssen wir noch dem Port E sagen, dass er im Output Modus mit 10MHz Clock und PushPull arbeiten soll. [4, p. 172]

GPIOE ->CRH = 0x1111 1111;



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Output - Register

Beschreiben `GPIOE->ODR = GPIOA->IDR<<8; // Lies Port A LOWER Byte…….`

BIT SET / RESET

REGISTER beschreiben

Fertig…?

Fast. Das eigentliche Programm fehlt noch.

Lösung:

```
/* Author: r. Malacarne    */

#include <stm32f10x.h>

void system_init()
 {
  RCC ->APB2ENR |= 1<<2;    // GPIOA Clock [1]
  RCC ->APB2ENR |= 1<<6;    // GPIOE Clock [2]
  GPIOA ->CRL = 0x88888888; // PA7..0 als Input. Nun definieren [3]
                            // ob wir Pull-Up oder Pull Down wollen.
  GPIOA ->ODR = 0x00FF;     // PA7..0 als Pullup  [4]
  GPIOE ->CRH = 0x11111111; // PE15-8 Output, Push Pull 10MHz  [5]
 }

/* Main                    */
int main(void){

system_init();                      // spezifischer System Init für
                                    // die aktuelle Applikation
while(1)
 {
  GPIOE->ODR = GPIOA->IDR<<8;       // Lies Port A LOWER Byte und
                                    // schiebe es nach links (8*). Weise
                                    // Resultat dann Port E zu.
 }
}
```

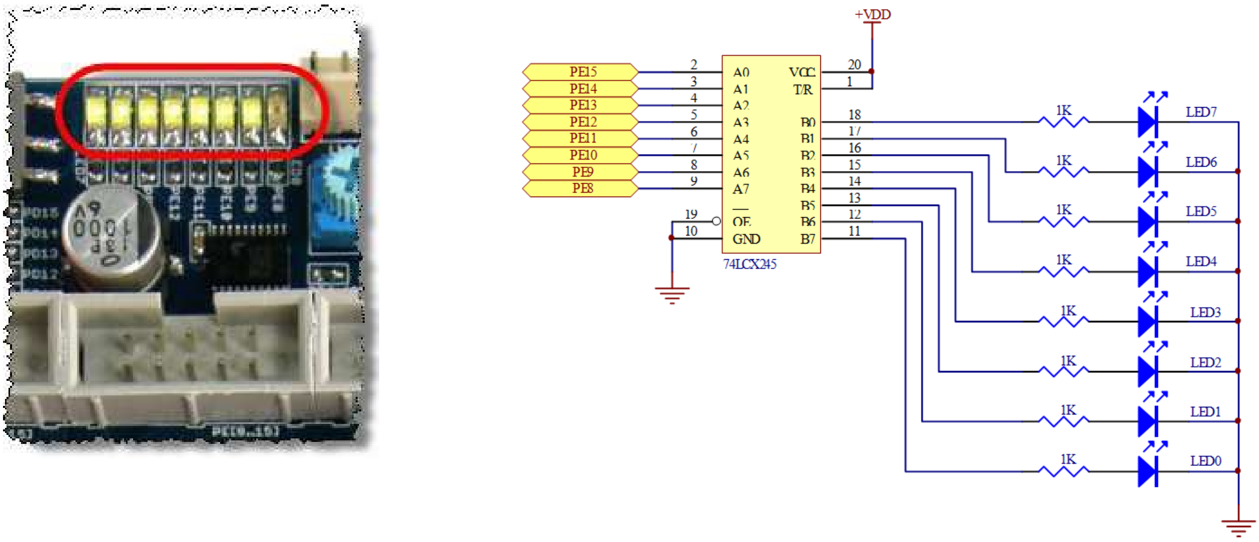
Wir lesen nun den Port A *(GPIOA->IDR)* komplett ein und schieben die Bits 8x nach links. Damit gelangen die PAL-Bits in den Bereich der High-Bits und weisen das Resultat dann dem Port E-High zu : `GPIOE->ODR = GPIOA->IDR<<8;` **Fertig**.

4 Hardware auf dem Board

4.1 Schaltung / Circuit LED Display

Die 8 Leds sind via Port PE [8..15] angesteuert. Ein „Logik 1“ Signal schaltet die entsprechende LED ein. Eine „0“ schaltet sie ab. Siehe auch untenstehendes Bild

1. LED auf MCB32: Ort und Schema

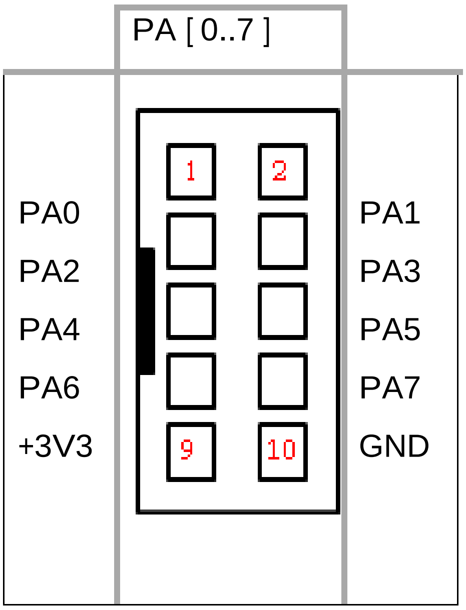


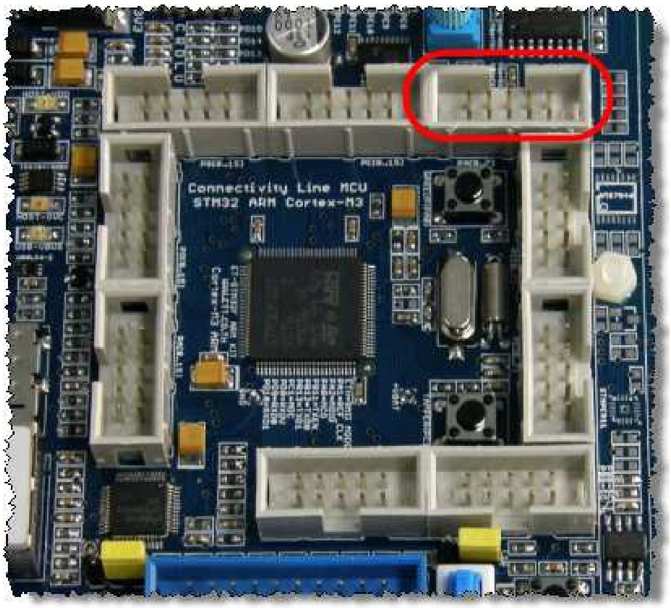
Die LED können via Port PE[8..15] (GPIO Output Port) geschaltet werden. Siehe dazu Code Beispiele.

4.2 Port PA Pin 0..7

Im folgenden Versuch wird Port PA benutzt. Der Port wird als Eingang geschaltet werden. Die anliegenden Signale werden dann an Port PE durchgeschaltet.

2. Steckerbelegung für 10pol. Stecker PA[0..7]





Die Belegung des 10poligen Steckers sieht wie oben abgebildet aus. Die roten Zahlen definieren die Adern des Flachbandkabels mit roter Ader = Pin1.

3. Original Belegung PA[0..7]

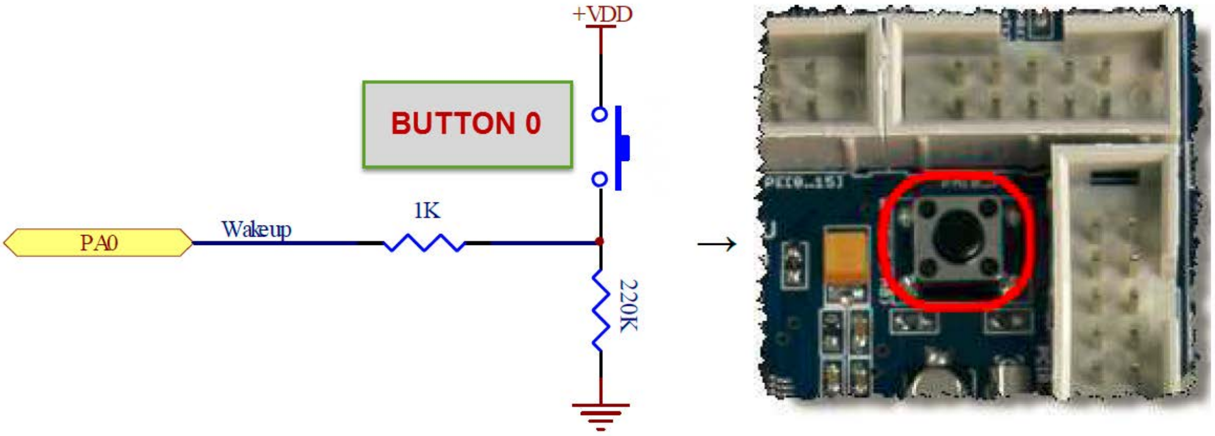
Die Original Pin-Belegung von Stecker PA[0...7] ist wie in der nebenstehenden Tabelle. Diese Einstellung wird im Versuch aber nicht benötigt.

Pin	Funktion	Module/Device	
PA0	Wakeup	Switch Wakeup	
PA1	RMII_REF_CLK	Ethernet LAN	
PA2	RMII_MDIO	Ethernet LAN	
PA3	-	-	
PA4	-	-	
PA5	SPI1_SCK	SD Card CLK	
PA6	SPI1_MISO	SD Card DAT0	
PA7	SPI1_MOSI	SD Card CMD	

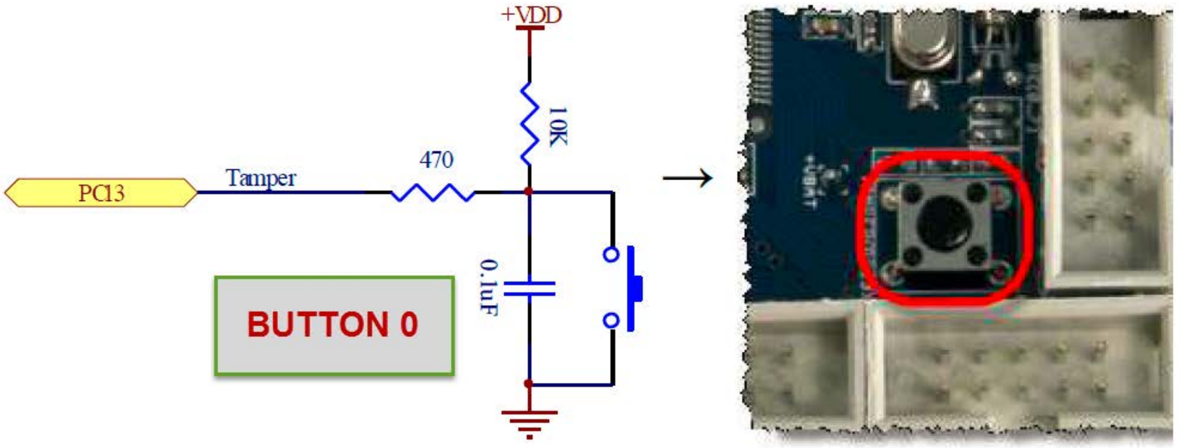
4.3 Anhang Anschlüsse am µC-Board MCB32

```
// In TouchP0P1.h definierte Pin-Bezeichnungen PA_0 .. PD_11, ohne Bezeichner wie Button .. !
char Button0     = PA_0;     // Bitwert 1/0, aktiv low, prellt wenig
char Button1     = PC_13;
char Stick       = PD_High;  // als Byte 0xF8 open, aktiv low, alle entprellt
char StickSelect = PD_15     // Bitwert  1/0;    Bytewert   0x80
char StickDown   = PD_14;    //          1/0;               0x40
char StickLeft   = PD_13;    //          1/0;               0x20
char StickUp     = PD_12;    //          1/0;               0x10
char StickRight  = PD_11;    //          1/0;               0x08
```

Button 0 (PA 0)

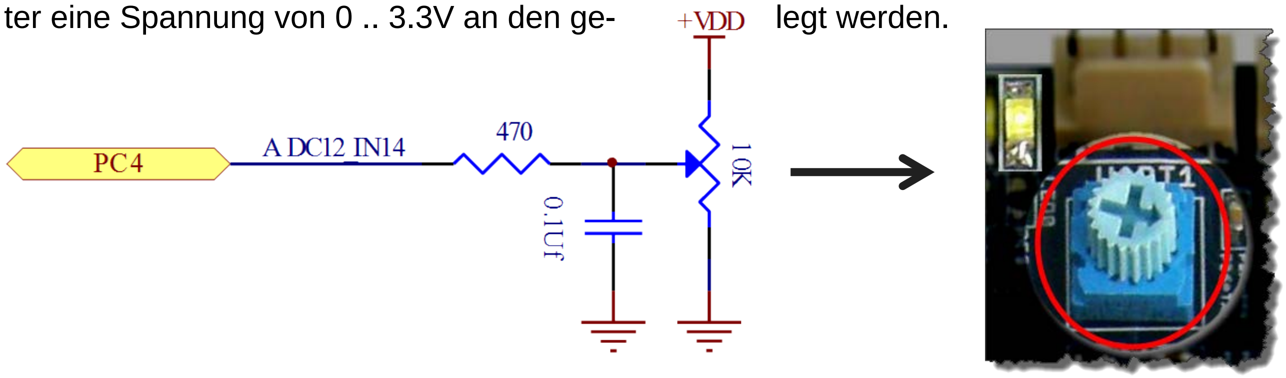


Button 1 (PC 13)

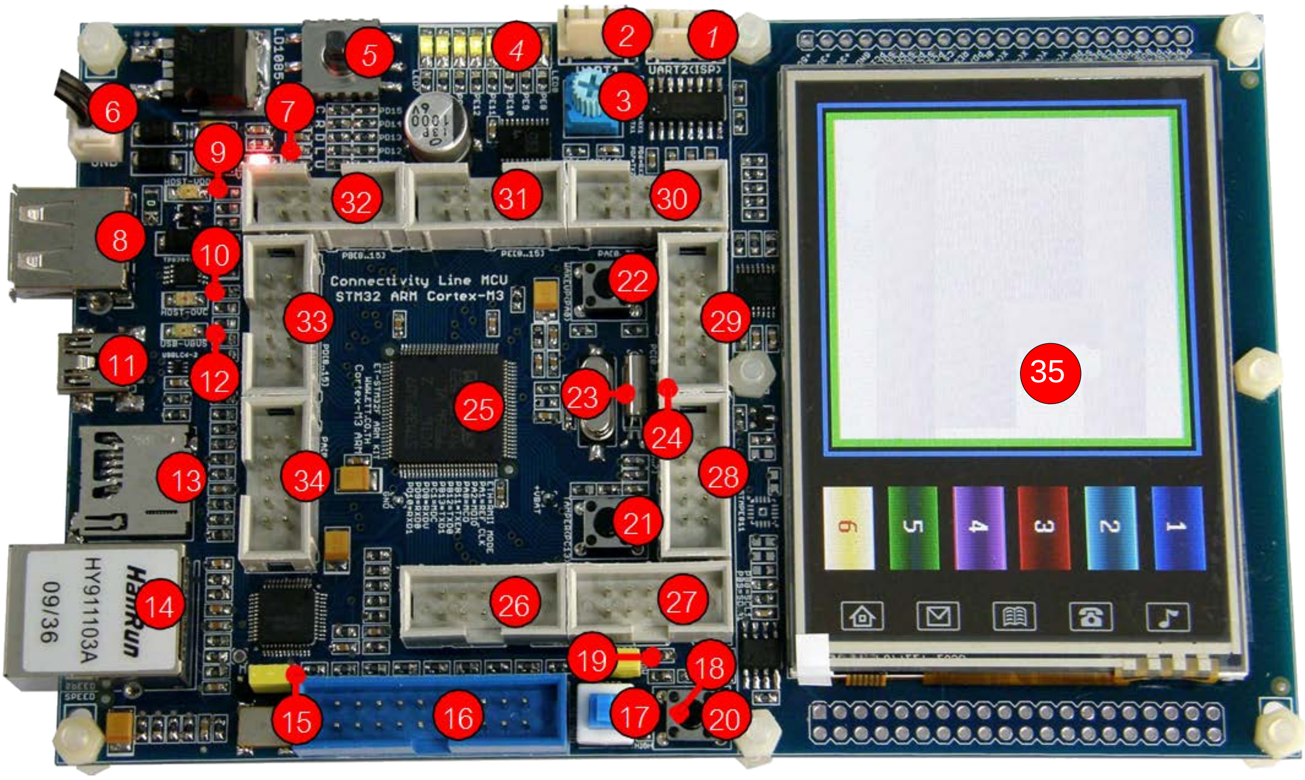


Potentiometer (PC4) // resp. P0_4 (Library)

Wenn der Port PC4 als Analog-Input (AD-Wandler) geschaltet ist kann mit dem Potentiometer eine Spannung von 0 .. 3.3V an den gelegt werden.



5 Anhang Übersicht Board



5.1 Tabelle mit Beschreibung der MCB32-Funktionen

#	Beschreibung der markierten Position (i)
1	RS232-2 (Uart2) Stecker. Empfängt HEX File für Bootloader
2	RS232-1 (Uart1) Stecker.
3	0…3,3Volt für A/D Wandler Test an (PC4/ADC14)
4	LED[0..7] angesteuert via PE[8..15].
5	5 Richtungs (Joy) Schalter
6	Stecker für +5Volt Speisung für das ganze Board
7	LED (+VDD: +3.3V) Status
8	USB Host
9	LED zeigt Status von USB-Host VDD.
10	LED zeigt Status von "Host Over Current"
11	Stecker "USB Device/OTG"
12	LED zeigt Status von "USB VBUS"

13	Micro SD Card Sockel
14	RJ45 Ethernet LAN.
15	Jumper(MCO/OSC) für die Auswahl des Clock-Signals für DP83848V.
16	JTAG Stecker für Real Time Debugging
17	Switch BOOT0 welcher zusammen mit Jumper BOOT1 den Boot-Modus der MCU bestimmt: Boot Loader (BOOT0=1, BOOT1=0) und Run (BOOT0=0,BOOT1=0).
18	LED zeigt Logic Status von BOOT0 = 1 (ON=Boot Loader, OFF=Run).
19	Jumper BOOT1(PB2); Ist im Normalfall auf LOW
20	Switch RESET.
21	Switch Tamper (PC13).
22	Switch Wakeup (PA0).
23	Quarz 25MHz für die Zeitbasis der MCU
24	Quarz 32.768KHz für die Zeitbasis der RTC (Real Time Clock)
25	MCU No.STM32F107VC T6
26	Stecker GPIO PD[0..7].
27	Stecker GPIO PB[0..7].
28	Stecker GPIO PE[0..7].
29	Stecker GPIO PC[0..7].
30	Stecker GPIO PA[0..7].
31	Stecker GPIO PE[8..15].
32	Stecker GPIO PB[8..15].
33	Stecker GPIO PD[8..15].
34	Stecker GPIO PA[8..15].
35	320x240 Dot TFT LCD mit Touch Screen Sensor. REV C oder REV D

8 Anhang Literaturverzeichnis und Links

[1] R. Weber, «General Purpos Input Output,» 2014.

[2] R. Weber, «Projektvorlagen (div) MCB32,» 2013ff.

[3] STM, «STM32F10x Standard Peripherals Firmware Library,» STM, 2010ff.

[4] ST, «ARM_STM_Reference manual_V2014_REV15,» ST, 2014.

[5] J. Yiu, The definitive Guide to ARM Cortex-M3 and M4 Processors, 3 Hrsg., Bd. 1, Elsevier, Hrsg., Oxford: Elsevier, 2014.

[6] R. Jesse, Arm Cortex M3 Mikrocontroller. Einstieg und Praxis, 1 Hrsg., www.mitp.de, Hrsg., Heidelberg: Hütigh Jehle Rehm GmbH, 2014.

[7] Diller, «System Timer,» 06 07 2014. [Online]. Available: http://www.diller-technologies.de/stm32.html#system_timer. [Zugriff am 06 07 2014].

[8] A. Limited, «DDI0337E_cortex_m3_r1p1_trm.pdf,» ARM Limited, http://infocenter.arm.com/help/topic/com.arm.doc.ddi0337e/DDI0337E_cortex_m3_r1p1_trm.pdf, 2005, 2006 ARM Limited.

[9] A. C. Group, «http://www.vr.ncue.edu.tw/esa/b1011/CMSIS_Core.htm,» 2007. [Online].

[10] E. Malacarne, *Glossar Malacarne,* V11 Hrsg., Rüti: Cityline AG, 2014.

[11] E. F. E. Schellenberg, «Programmieren im Fach HST,» TBZ, Zürich, 2010ff.

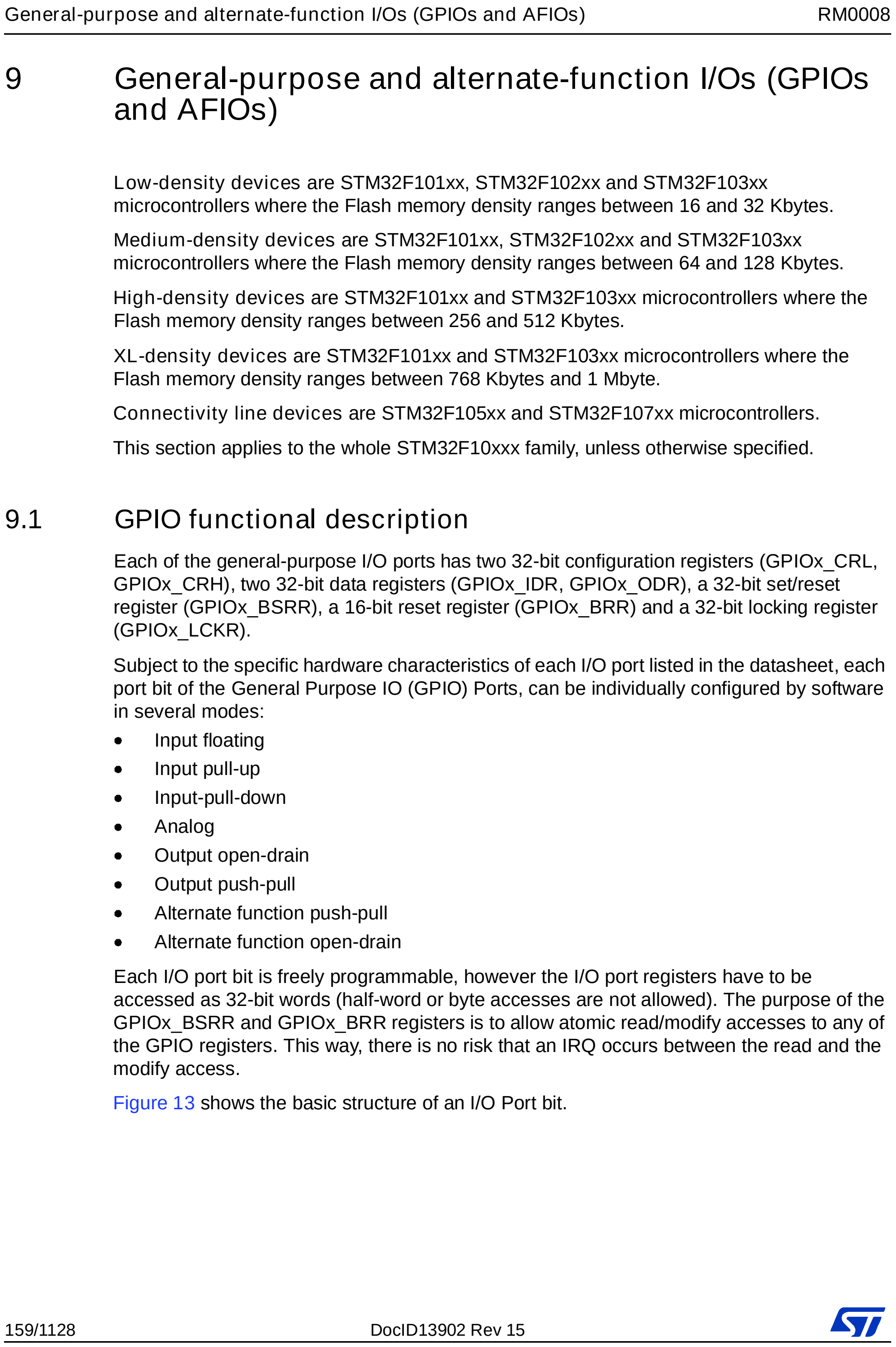
9 Anhang Wichtige Dokumente

Die folgende Liste zeigt auf die wichtigsten Dokumente welche im WEB zu finden sind. Beim Suchen lassen sich noch viele nützliche Links finden.

- Datenblatt (STM32F107VC) Beschreibung des konkreten Chips für Pinbelegung etc.
- Reference Manual (STM32F107VC) (>1000Seiten in Englisch)
  Ausführliche Beschreibung der Module einer Familie. Unter Umständen sind nicht alle Module im eingesetzten Chip vorhanden – siehe Datenblatt.
- Programming Manual (Cortex-M3)
  Enthält beispielsweise Informationen zum Interrupt Controller (NVIC).

- Standard Peripheral Library (STM32F10x)
  Im Gegensatz zu anderen MCUs sollen die Register der STM32 nicht direkt angesprochen werden. Dafür dienen die Funktionen der Standard Peripheral Library.
  Sie ist auf http://www.st.com/ zusammen mit einer Dokumentation (Datei: stm32f10x_stdperiph_lib_um.chm) herunterladbar.

10 Anhang „STM 32 REF Manual" Auszug GPIO



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

9 General-purpose and alternate-function I/Os (GPIOs and AFIOs)

Low-density devices are STM32F101xx, STM32F102xx and STM32F103xx microcontrollers where the Flash memory density ranges between 16 and 32 Kbytes.

Medium-density devices are STM32F101xx, STM32F102xx and STM32F103xx microcontrollers where the Flash memory density ranges between 64 and 128 Kbytes.

High-density devices are STM32F101xx and STM32F103xx microcontrollers where the Flash memory density ranges between 256 and 512 Kbytes.

XL-density devices are STM32F101xx and STM32F103xx microcontrollers where the Flash memory density ranges between 768 Kbytes and 1 Mbyte.

Connectivity line devices are STM32F105xx and STM32F107xx microcontrollers.

This section applies to the whole STM32F10xxx family, unless otherwise specified.

9.1 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).

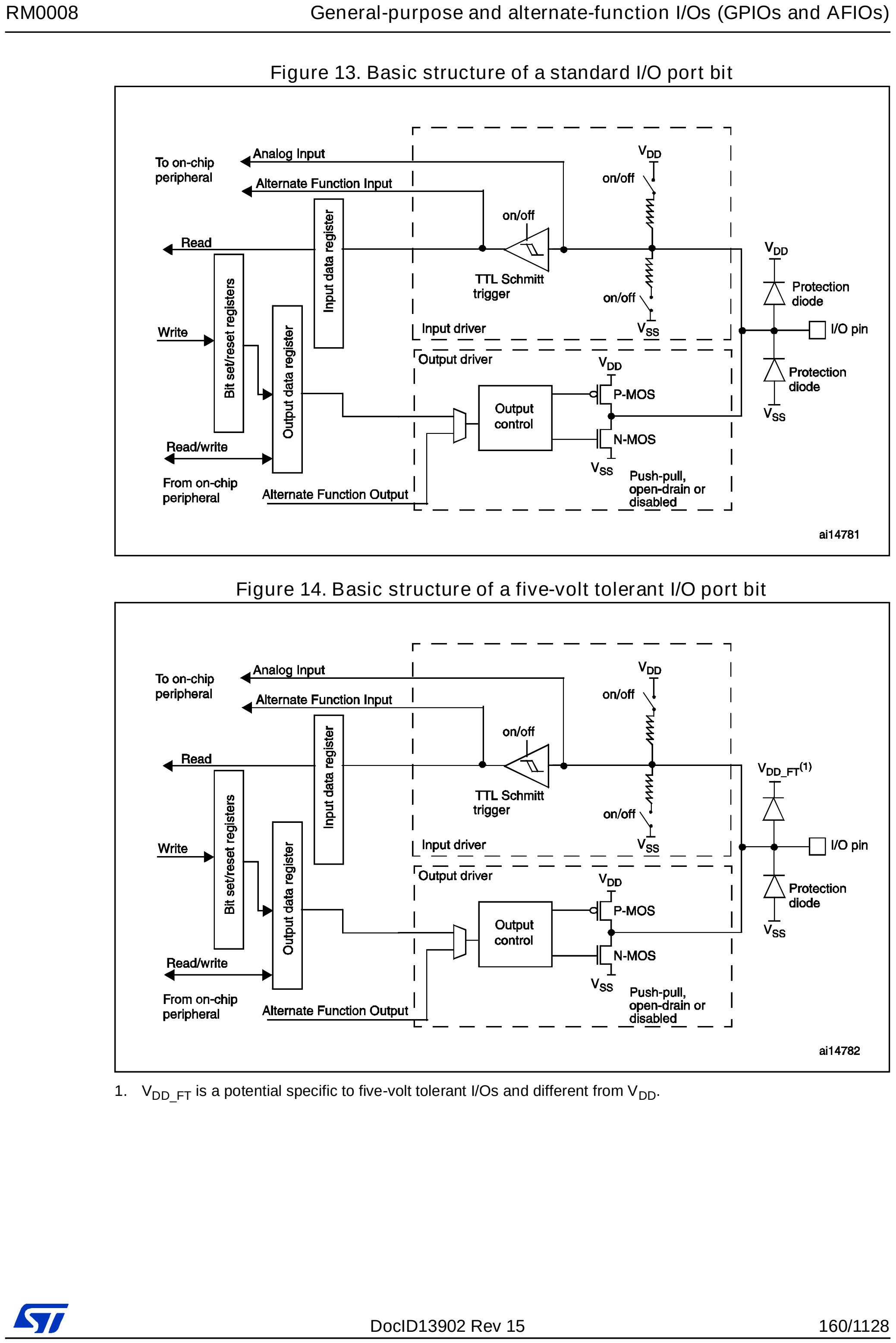
Subject to the specific hardware characteristics of each I/O port listed in the datasheet, each port bit of the General Purpose IO (GPIO) Ports, can be individually configured by software in several modes:

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

Each I/O port bit is freely programmable, however the I/O port registers have to be accessed as 32-bit words (half-word or byte accesses are not allowed). The purpose of the GPIOx_BSRR and GPIOx_BRR registers is to allow atomic read/modify accesses to any of the GPIO registers. This way, there is no risk that an IRQ occurs between the read and the modify access.

Figure 13 shows the basic structure of an I/O Port bit.

159/1128 DocID13902 Rev 15



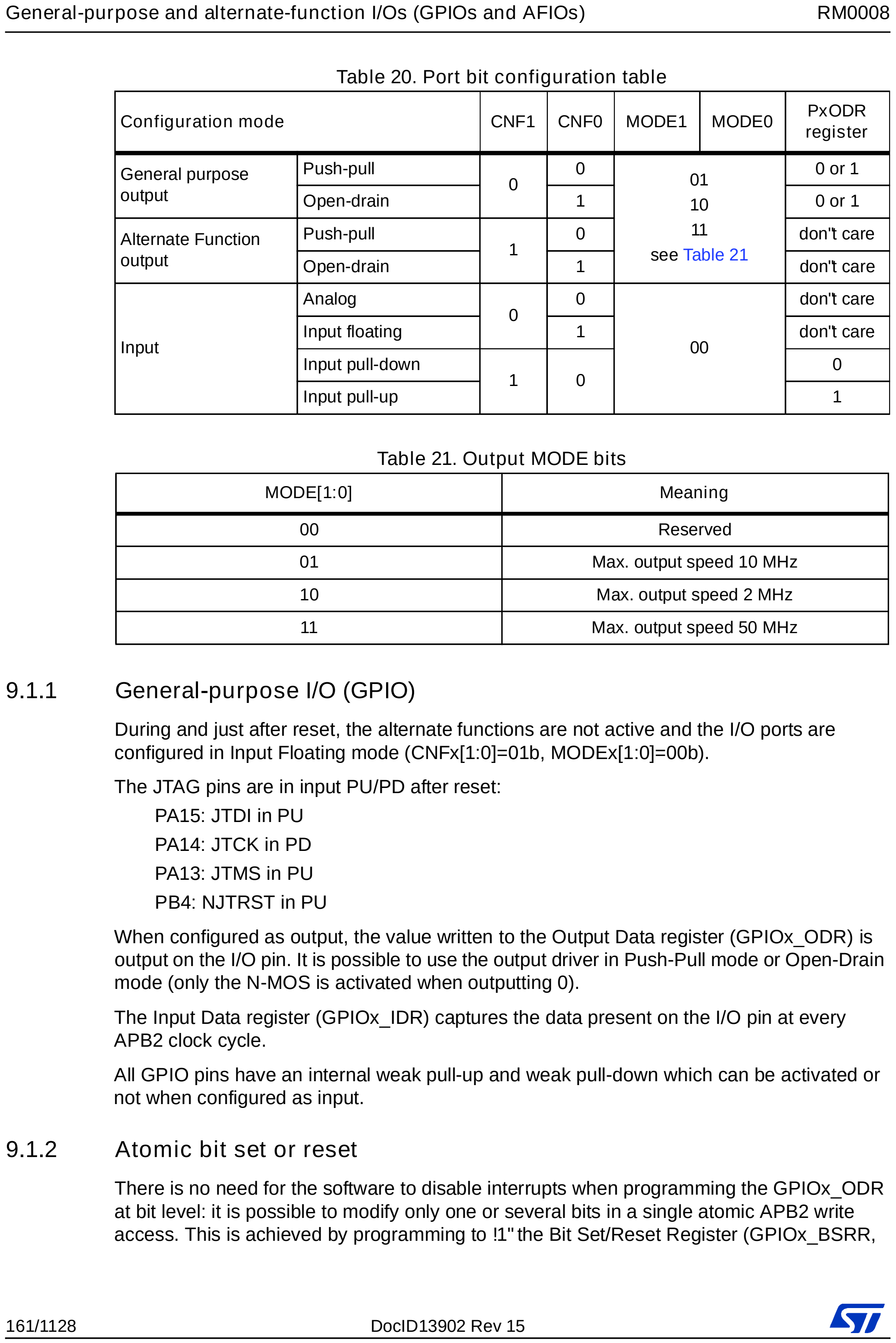
RM0008 General-purpose and alternate-function I/Os (GPIOs and AFIOs)

Figure 13. Basic structure of a standard I/O port bit

Figure 14. Basic structure of a five-volt tolerant I/O port bit

1. V_{DD_FT} is a potential specific to five-volt tolerant I/Os and different from V_{DD}.

DocID13902 Rev 15 160/1128



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

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 Table 21		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		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	Max. output speed 10 MHz
10	Max. output speed 2 MHz
11	Max. output speed 50 MHz

9.1.1 General-purpose I/O (GPIO)

During and just after reset, the alternate functions are not active and the I/O ports are configured in Input Floating mode (CNFx[1:0]=01b, MODEx[1:0]=00b).

The JTAG pins are in input PU/PD after reset:

PA15: JTDI in PU
PA14: JTCK in PD
PA13: JTMS in PU
PB4: NJTRST in PU

When 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 N-MOS is activated when outputting 0).

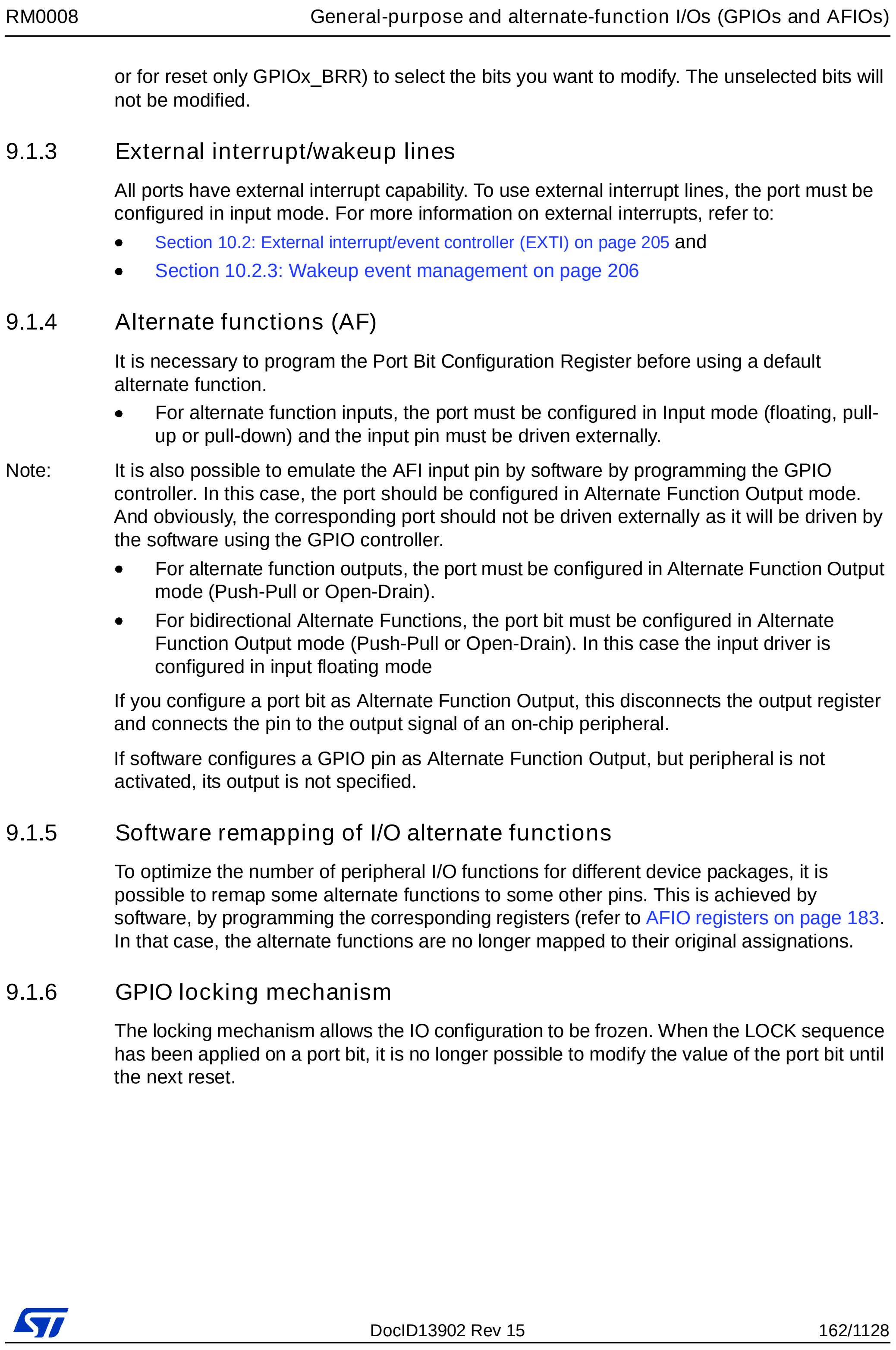
The Input Data register (GPIOx_IDR) captures the data present on the I/O pin at every APB2 clock cycle.

All GPIO pins have an internal weak pull-up and weak pull-down which can be activated or not when configured as input.

9.1.2 Atomic bit set or reset

There is no need for the software to disable interrupts when programming the GPIOx_ODR at bit level: it is possible to modify only one or several bits in a single atomic APB2 write access. This is achieved by programming to !1" the Bit Set/Reset Register (GPIOx_BSRR,

161/1128 DocID13902 Rev 15



or for reset only GPIOx_BRR) to select the bits you want to modify. The unselected bits will not be modified.

9.1.3 External interrupt/wakeup lines

All ports have external interrupt capability. To use external interrupt lines, the port must be configured in input mode. For more information on external interrupts, refer to:

- Section 10.2: External interrupt/event controller (EXTI) on page 205 and
- Section 10.2.3: Wakeup event management on page 206

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

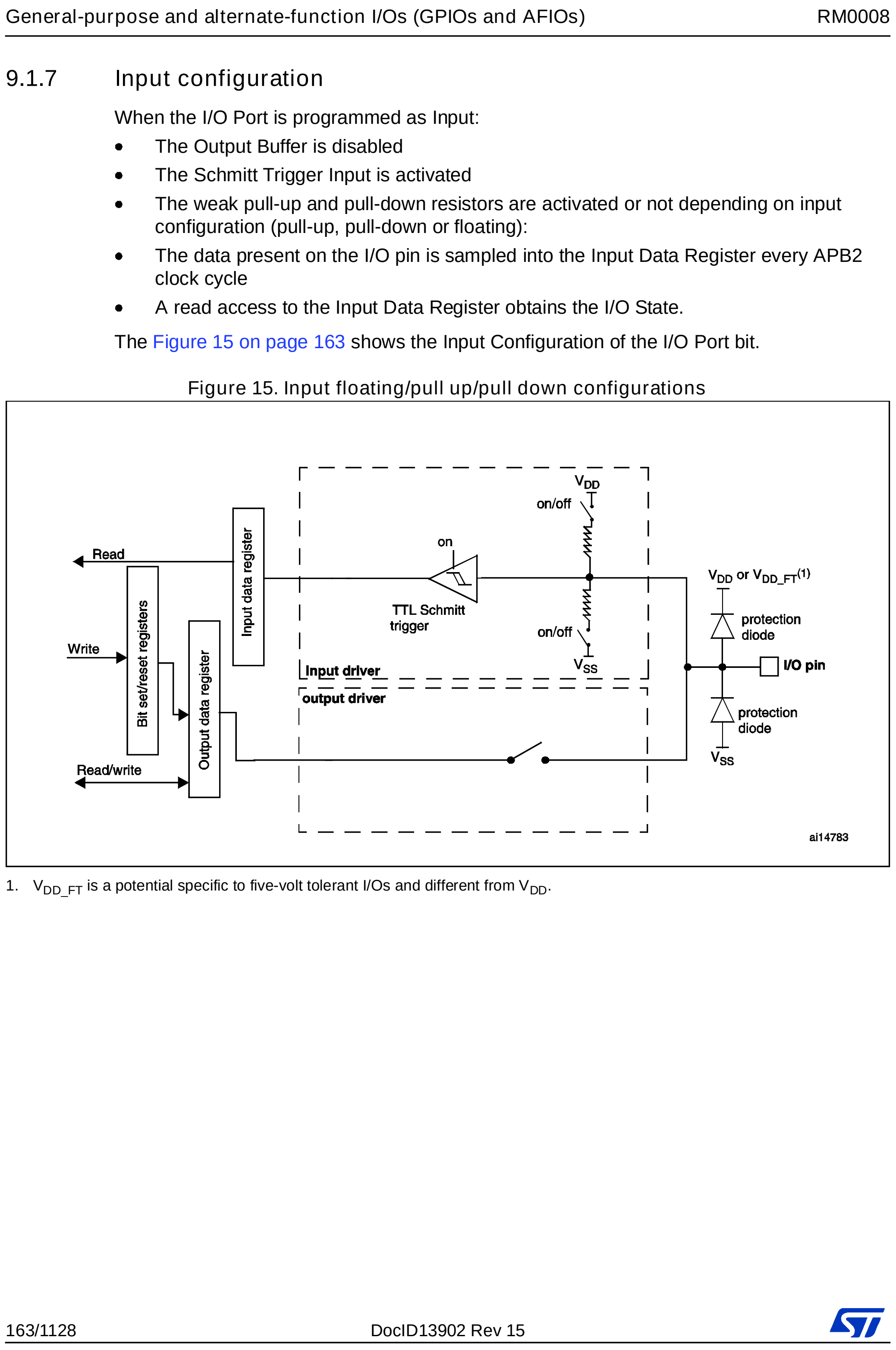
If software configures a GPIO pin as Alternate Function Output, but peripheral is not activated, its output is not specified.

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 on page 183. In that case, the alternate functions are no longer mapped to their original assignations.

9.1.6 GPIO locking mechanism

The locking mechanism allows the IO configuration to be frozen. When the LOCK sequence has been applied on a port bit, it is no longer possible to modify the value of the port bit until the next reset.



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

9.1.7 Input configuration

When the I/O Port is programmed as Input:

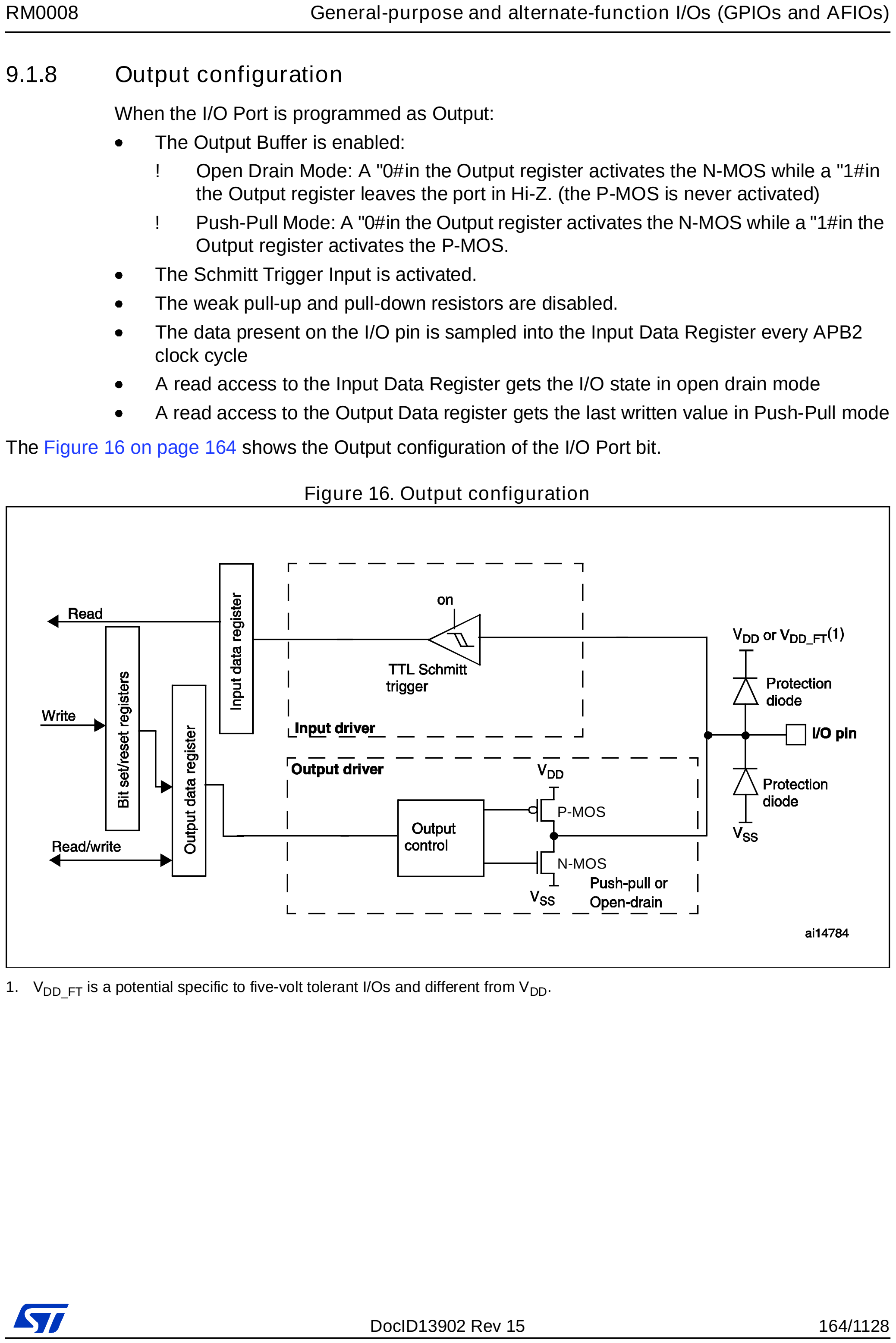
- The Output Buffer is disabled
- The Schmitt Trigger Input is activated
- The weak pull-up and pull-down resistors are activated or not depending on input configuration (pull-up, pull-down or floating):
- The data present on the I/O pin is sampled into the Input Data Register every APB2 clock cycle
- A read access to the Input Data Register obtains the I/O State.

The Figure 15 on page 163 shows the Input Configuration of the I/O Port bit.

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

1. V_{DD_FT} is a potential specific to five-volt tolerant I/Os and different from V_{DD}.

163/1128 DocID13902 Rev 15



RM0008 General-purpose and alternate-function I/Os (GPIOs and AFIOs)

9.1.8 Output configuration

When the I/O Port is programmed as Output:

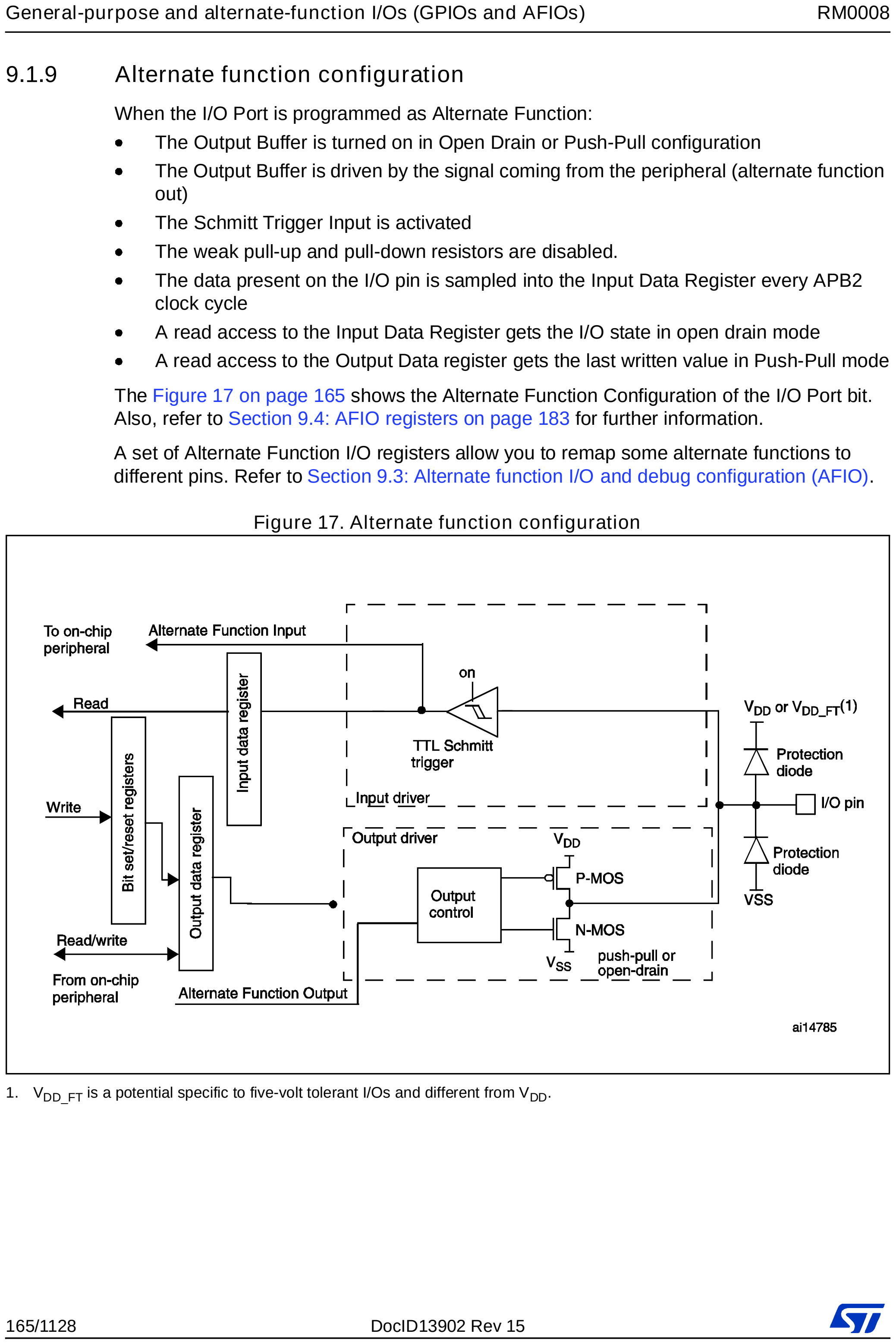
- The Output Buffer is enabled:
  - ! Open Drain Mode: A "0#in the Output register activates the N-MOS while a "1#in the Output register leaves the port in Hi-Z. (the P-MOS is never activated)
  - ! Push-Pull Mode: A "0#in the Output register activates the N-MOS while a "1#in the Output register activates the P-MOS.
- The Schmitt Trigger Input is activated.
- The weak pull-up and pull-down resistors are disabled.
- The data present on the I/O pin is sampled into the Input Data Register every APB2 clock cycle
- A read access to the Input Data Register gets the I/O state in open drain mode
- A read access to the Output Data register gets the last written value in Push-Pull mode

The Figure 16 on page 164 shows the Output configuration of the I/O Port bit.

Figure 16. Output configuration

1. V_{DD_FT} is a potential specific to five-volt tolerant I/Os and different from V_{DD}.

DocID13902 Rev 15 164/1128



9.1.9 Alternate function configuration

When the I/O Port is programmed as Alternate Function:

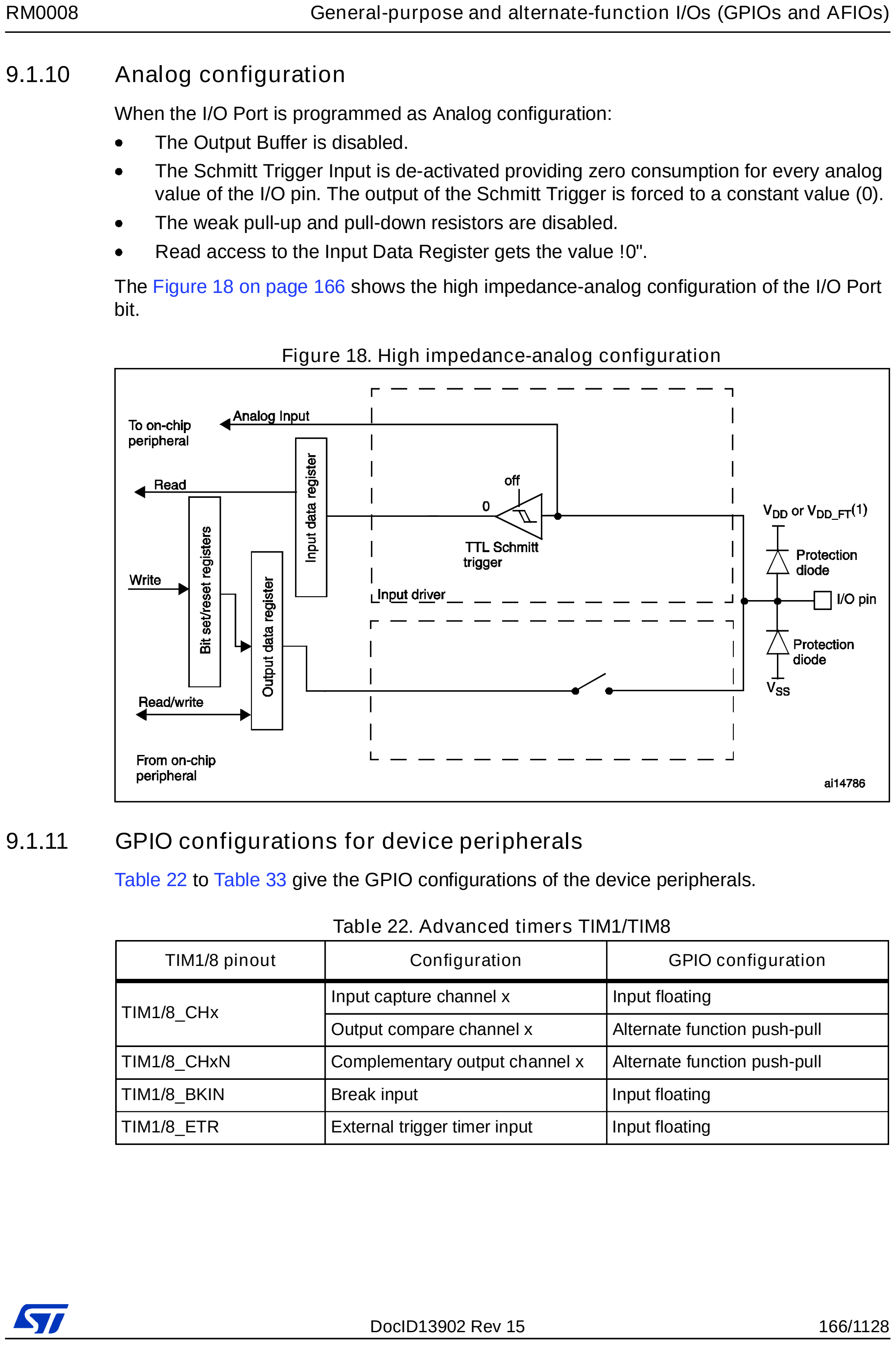
- The Output Buffer is turned on in Open Drain or Push-Pull configuration
- The Output Buffer is driven by the signal coming from the peripheral (alternate function out)
- The Schmitt Trigger Input is activated
- The weak pull-up and pull-down resistors are disabled.
- The data present on the I/O pin is sampled into the Input Data Register every APB2 clock cycle
- A read access to the Input Data Register gets the I/O state in open drain mode
- A read access to the Output Data register gets the last written value in Push-Pull mode

The Figure 17 on page 165 shows the Alternate Function Configuration of the I/O Port bit. Also, refer to Section 9.4: AFIO registers on page 183 for further information.

A set of Alternate Function I/O registers allow you to remap some alternate functions to different pins. Refer to Section 9.3: Alternate function I/O and debug configuration (AFIO).

Figure 17. Alternate function configuration

1. V_{DD_FT} is a potential specific to five-volt tolerant I/Os and different from V_{DD}.



9.1.10 Analog configuration

When the I/O Port is programmed as Analog configuration:

- The Output Buffer is disabled.
- The Schmitt Trigger Input is de-activated 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.
- Read access to the Input Data Register gets the value !0".

The Figure 18 on page 166 shows the high impedance-analog configuration of the I/O Port bit.

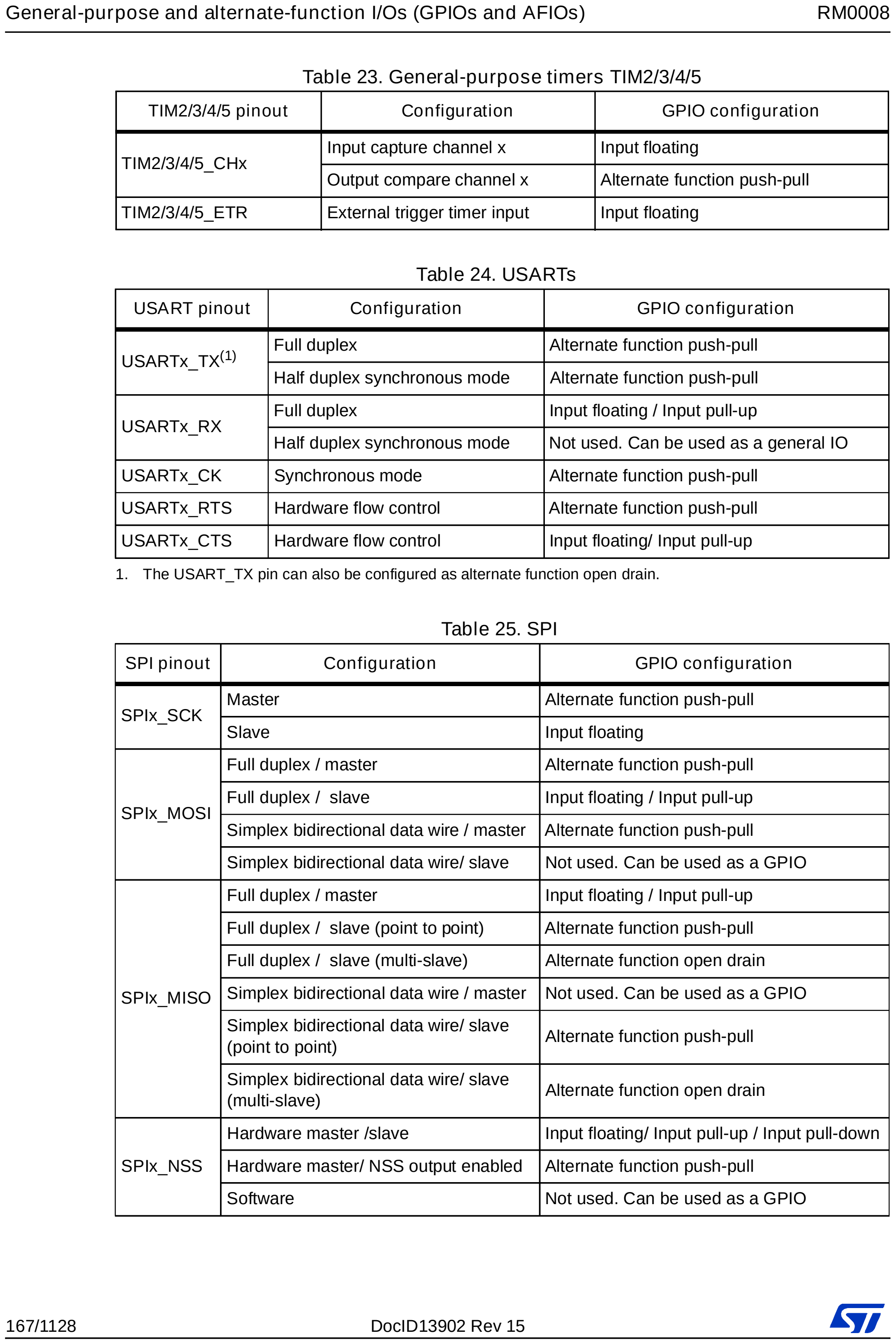
Figure 18. High impedance-analog configuration

9.1.11 GPIO configurations for device peripherals

Table 22 to Table 33 give the GPIO configurations of the device peripherals.

Table 22. Advanced timers TIM1/TIM8

TIM1/8 pinout	Configuration	GPIO configuration
TIM1/8_CHx	Input capture channel x	Input floating
	Output compare channel x	Alternate function push-pull
TIM1/8_CHxN	Complementary output channel x	Alternate function push-pull
TIM1/8_BKIN	Break input	Input floating
TIM1/8_ETR	External trigger timer input	Input floating



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

Table 23. General-purpose timers TIM2/3/4/5

TIM2/3/4/5 pinout	Configuration	GPIO configuration
TIM2/3/4/5_CHx	Input capture channel x	Input floating
	Output compare channel x	Alternate function push-pull
TIM2/3/4/5_ETR	External trigger timer input	Input floating

Table 24. USARTs

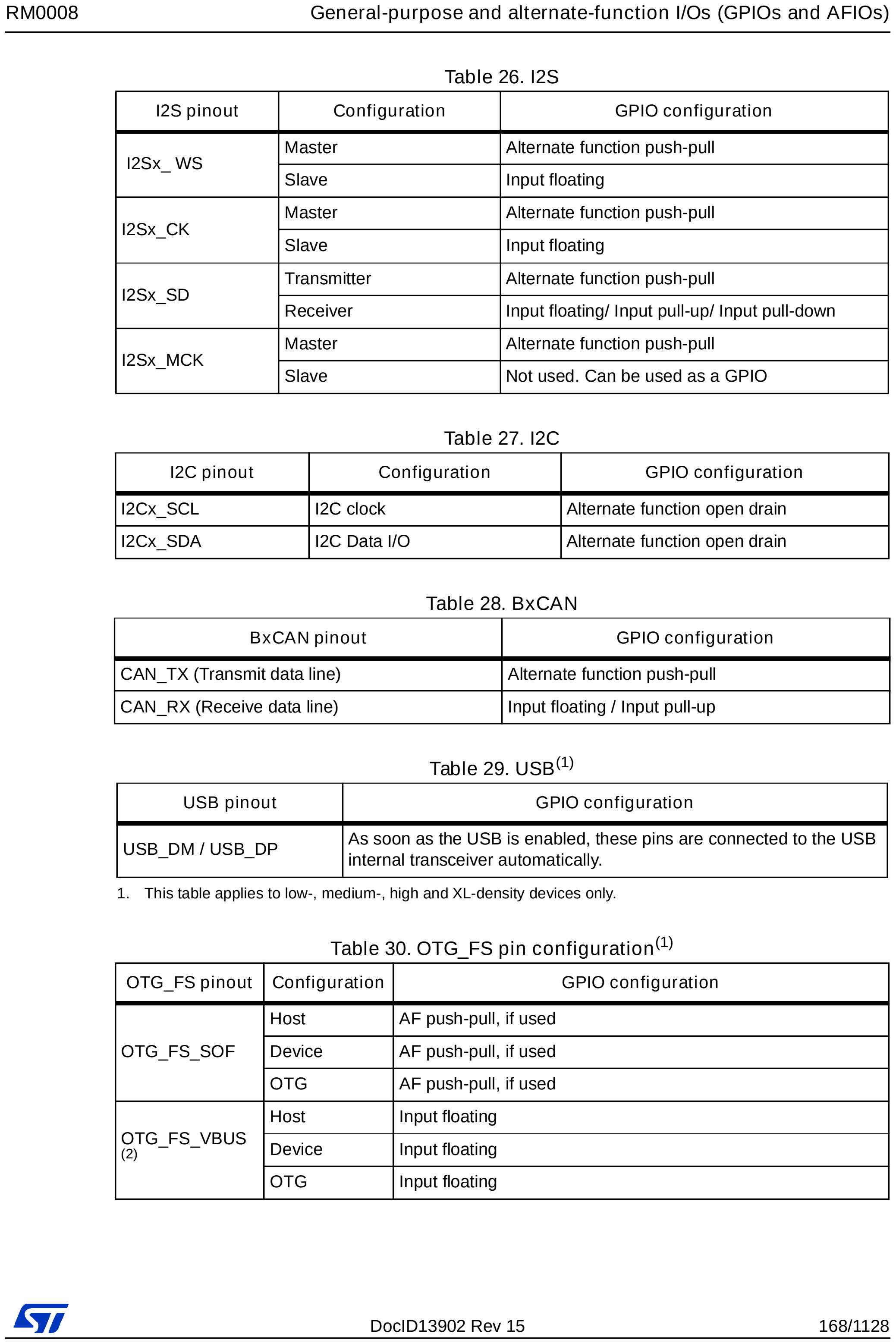
USART pinout	Configuration	GPIO configuration
USARTx_TX[1]	Full duplex	Alternate function push-pull
	Half duplex synchronous mode	Alternate function push-pull
USARTx_RX	Full duplex	Input floating / Input pull-up
	Half duplex synchronous mode	Not used. Can be used as a general IO
USARTx_CK	Synchronous mode	Alternate function push-pull
USARTx_RTS	Hardware flow control	Alternate function push-pull
USARTx_CTS	Hardware flow control	Input floating/ Input pull-up

1. The USART_TX pin can also be configured as alternate function open drain.

Table 25. SPI

SPI pinout	Configuration	GPIO configuration
SPIx_SCK	Master	Alternate function push-pull
	Slave	Input floating
SPIx_MOSI	Full duplex / master	Alternate function push-pull
	Full duplex / slave	Input floating / Input pull-up
	Simplex bidirectional data wire / master	Alternate function push-pull
	Simplex bidirectional data wire/ slave	Not used. Can be used as a GPIO
SPIx_MISO	Full duplex / master	Input floating / Input pull-up
	Full duplex / slave (point to point)	Alternate function push-pull
	Full duplex / slave (multi-slave)	Alternate function open drain
	Simplex bidirectional data wire / master	Not used. Can be used as a GPIO
	Simplex bidirectional data wire/ slave (point to point)	Alternate function push-pull
	Simplex bidirectional data wire/ slave (multi-slave)	Alternate function open drain
SPIx_NSS	Hardware master /slave	Input floating/ Input pull-up / Input pull-down
	Hardware master/ NSS output enabled	Alternate function push-pull
	Software	Not used. Can be used as a GPIO

167/1128 DocID13902 Rev 15



RM0008 General-purpose and alternate-function I/Os (GPIOs and AFIOs)

Table 26. I2S

I2S pinout	Configuration	GPIO configuration
I2Sx_ WS	Master	Alternate function push-pull
	Slave	Input floating
I2Sx_CK	Master	Alternate function push-pull
	Slave	Input floating
I2Sx_SD	Transmitter	Alternate function push-pull
	Receiver	Input floating/ Input pull-up/ Input pull-down
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

Table 29. USB[1]

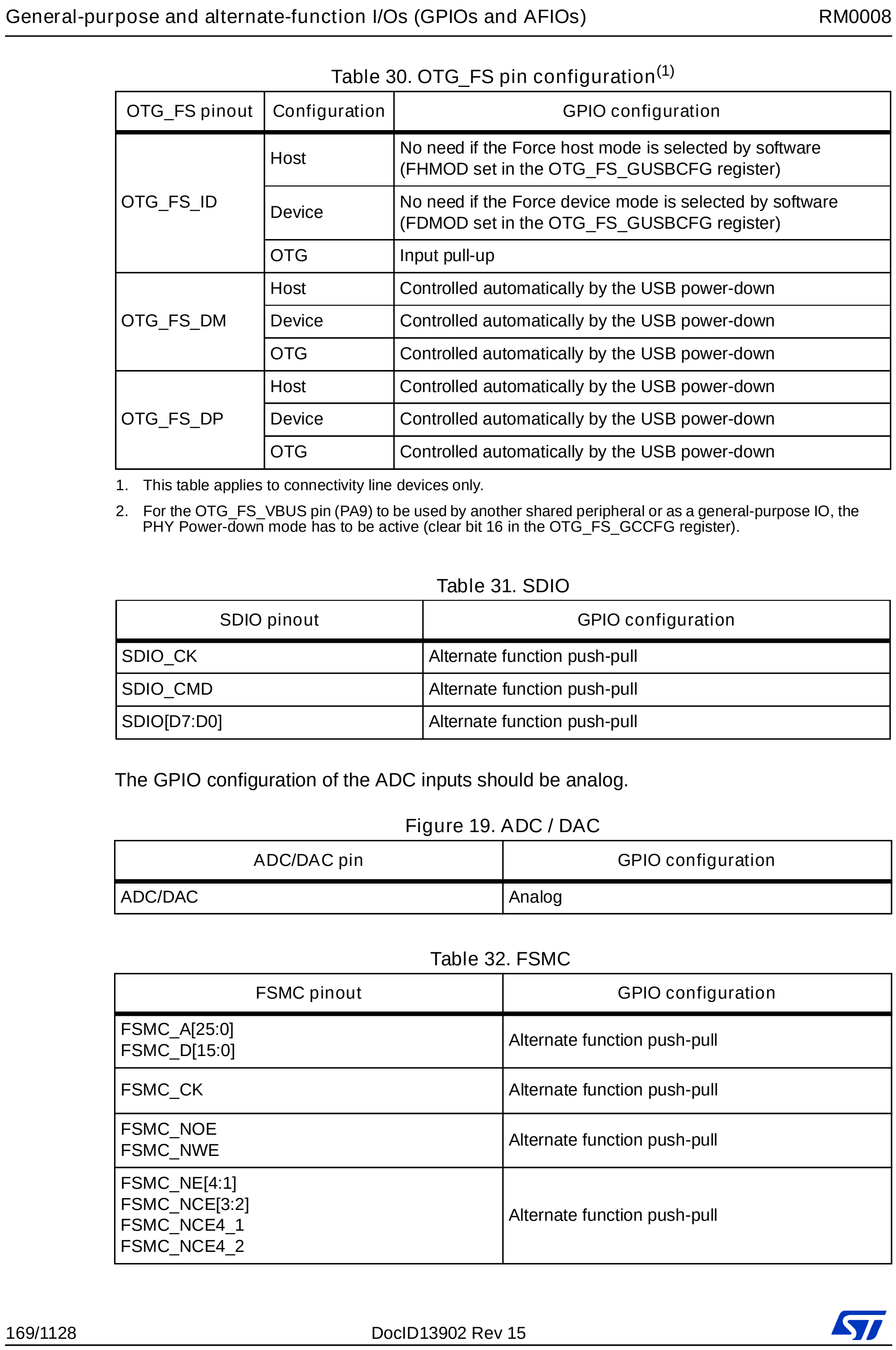
USB pinout	GPIO configuration
USB_DM / USB_DP	As soon as the USB is enabled, these pins are connected to the USB internal transceiver automatically.

[1] This table applies to low-, medium-, high and XL-density devices only.

Table 30. OTG_FS pin configuration[1]

OTG_FS pinout	Configuration	GPIO configuration
OTG_FS_SOF	Host	AF push-pull, if used
	Device	AF push-pull, if used
	OTG	AF push-pull, if used
OTG_FS_VBUS [2]	Host	Input floating
	Device	Input floating
	OTG	Input floating

DocID13902 Rev 15 168/1128



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

Table 30. OTG_FS pin configuration[1]

OTG_FS pinout	Configuration	GPIO configuration
OTG_FS_ID	Host	No need if the Force host mode is selected by software (FHMOD set in the OTG_FS_GUSBCFG register)
	Device	No need if the Force device mode is selected by software (FDMOD set in the OTG_FS_GUSBCFG register)
	OTG	Input pull-up
OTG_FS_DM	Host	Controlled automatically by the USB power-down
	Device	Controlled automatically by the USB power-down
	OTG	Controlled automatically by the USB power-down
OTG_FS_DP	Host	Controlled automatically by the USB power-down
	Device	Controlled automatically by the USB power-down
	OTG	Controlled automatically by the USB power-down

1. This table applies to connectivity line devices only.
2. For the OTG_FS_VBUS pin (PA9) to be used by another shared peripheral or as a general-purpose IO, the PHY Power-down mode has to be active (clear bit 16 in the OTG_FS_GCCFG register).

Table 31. SDIO

SDIO pinout	GPIO configuration
SDIO_CK	Alternate function push-pull
SDIO_CMD	Alternate function push-pull
SDIO[D7:D0]	Alternate function push-pull

The GPIO configuration of the ADC inputs should be analog.

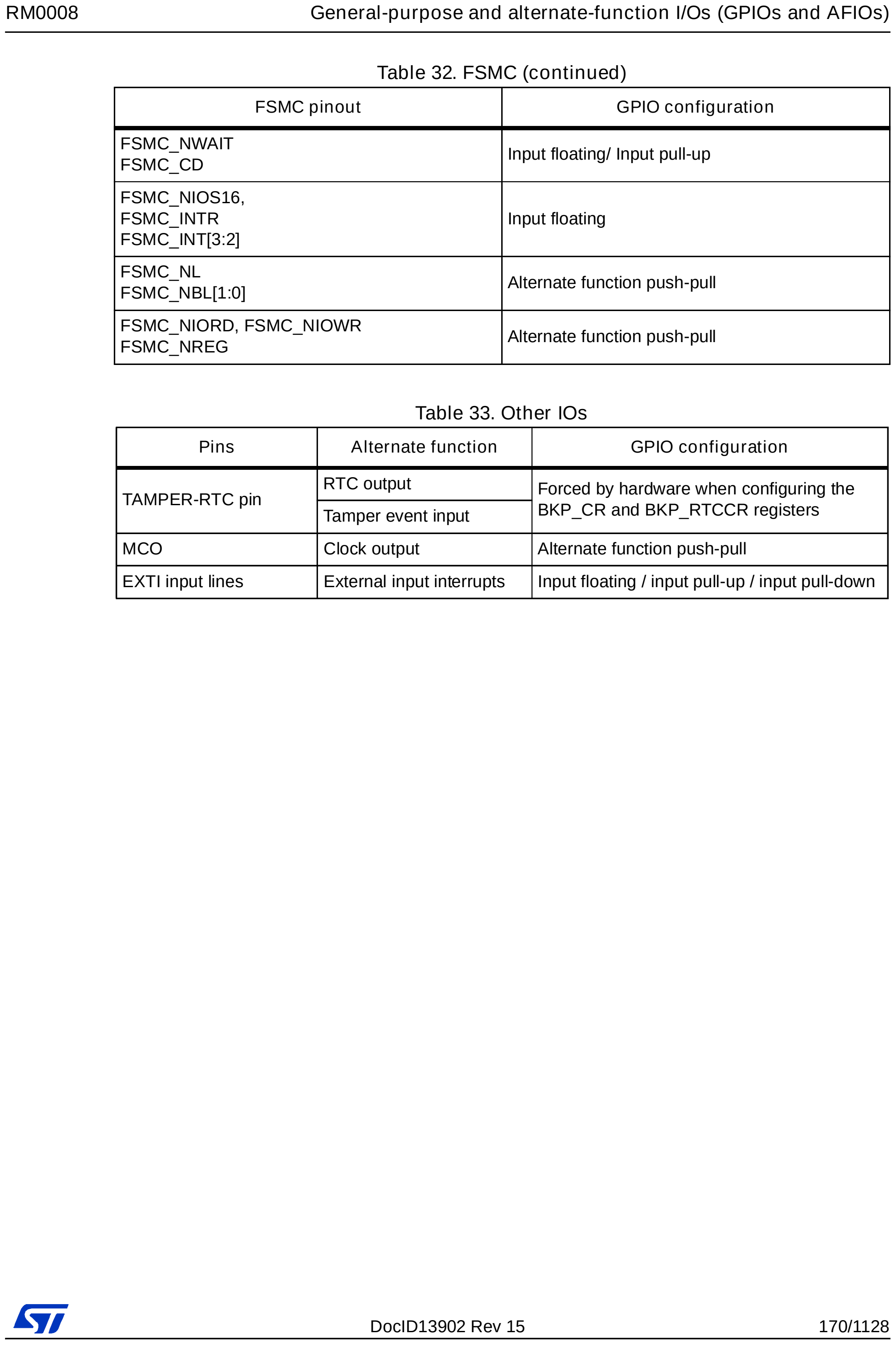
Figure 19. ADC / DAC

ADC/DAC pin	GPIO configuration
ADC/DAC	Analog

Table 32. FSMC

FSMC pinout	GPIO configuration
FSMC_A[25:0] FSMC_D[15:0]	Alternate function push-pull
FSMC_CK	Alternate function push-pull
FSMC_NOE FSMC_NWE	Alternate function push-pull
FSMC_NE[4:1] FSMC_NCE[3:2] FSMC_NCE4_1 FSMC_NCE4_2	Alternate function push-pull

169/1128 DocID13902 Rev 15



RM0008 General-purpose and alternate-function I/Os (GPIOs and AFIOs)

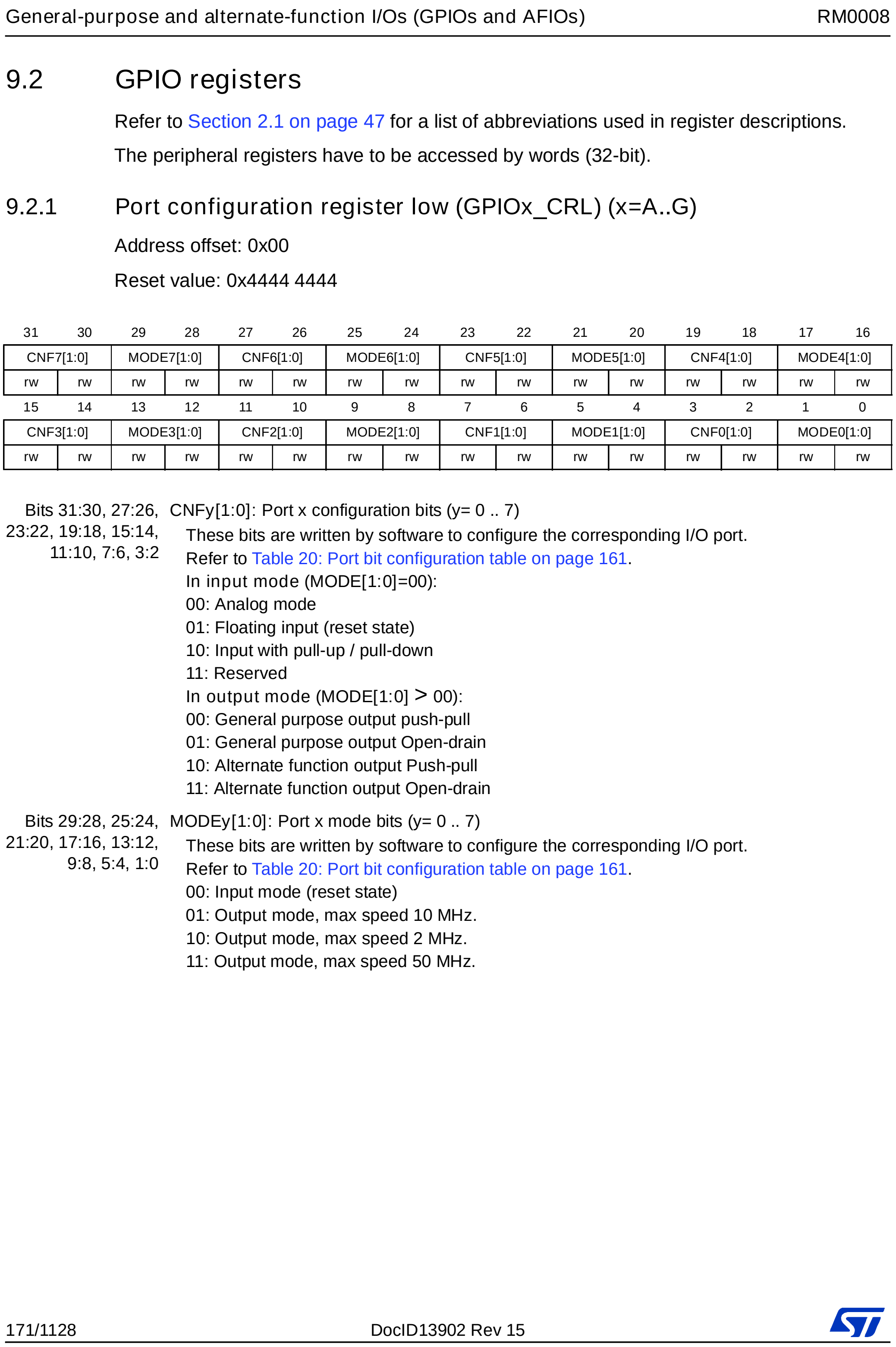
Table 32. FSMC (continued)

FSMC pinout	GPIO configuration
FSMC_NWAIT FSMC_CD	Input floating/ Input pull-up
FSMC_NIOS16, FSMC_INTR FSMC_INT[3:2]	Input floating
FSMC_NL FSMC_NBL[1:0]	Alternate function push-pull
FSMC_NIORD, FSMC_NIOWR FSMC_NREG	Alternate function push-pull

Table 33. Other IOs

Pins	Alternate function	GPIO configuration
TAMPER-RTC pin	RTC output	Forced by hardware when configuring the BKP_CR and BKP_RTCCR registers
	Tamper event input	
MCO	Clock output	Alternate function push-pull
EXTI input lines	External input interrupts	Input floating / input pull-up / input pull-down

DocID13902 Rev 15 170/1128



9.2 GPIO registers

Refer to Section 2.1 on page 47 for a list of abbreviations used in register descriptions.

The peripheral registers have to be accessed by words (32-bit).

9.2.1 Port configuration register low (GPIOx_CRL) (x=A..G)

Address offset: 0x00

Reset value: 0x4444 4444

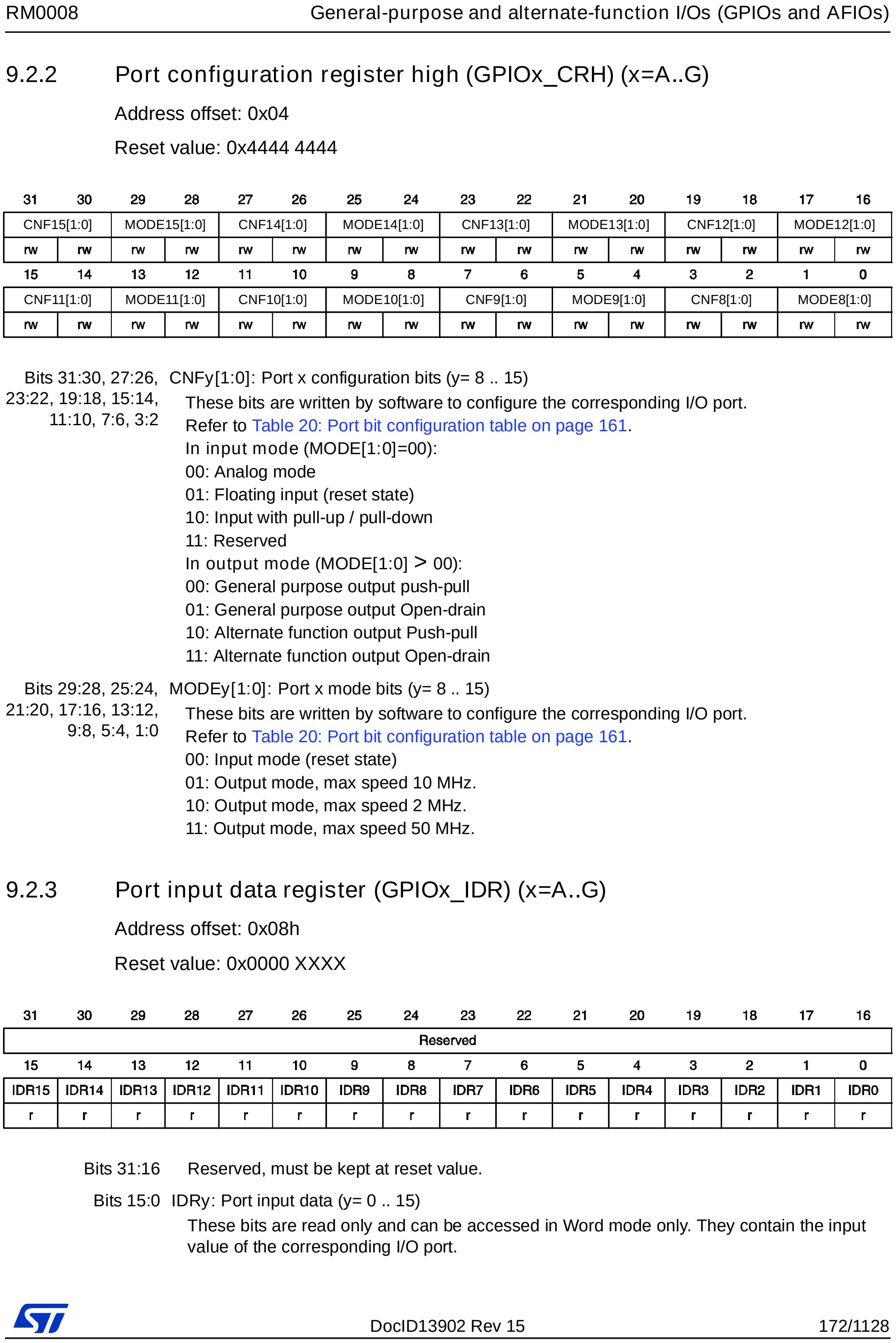
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]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

Bits 31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2 CNFy[1:0]: Port x configuration bits (y= 0 .. 7)

These bits are written by software to configure the corresponding I/O port.
Refer to Table 20: Port bit configuration table on page 161.
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, 21:20, 17:16, 13:12, 9:8, 5:4, 1:0 MODEy[1:0]: Port x mode bits (y= 0 .. 7)

These bits are written by software to configure the corresponding I/O port.
Refer to Table 20: Port bit configuration table on page 161.
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.



RM0008 General-purpose and alternate-function I/Os (GPIOs and AFIOs)

9.2.2 Port configuration register high (GPIOx_CRH) (x=A..G)

Address offset: 0x04

Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30, 27:26, 23:22, 19:18, 15:14, 11:10, 7:6, 3:2 CNFy[1:0]: Port x configuration bits (y= 8 .. 15)

These bits are written by software to configure the corresponding I/O port.
Refer to Table 20: Port bit configuration table on page 161.
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, 21:20, 17:16, 13:12, 9:8, 5:4, 1:0 MODEy[1:0]: Port x mode bits (y= 8 .. 15)

These bits are written by software to configure the corresponding I/O port.
Refer to Table 20: Port bit configuration table on page 161.
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.

9.2.3 Port input data register (GPIOx_IDR) (x=A..G)

Address offset: 0x08h

Reset value: 0x0000 XXXX

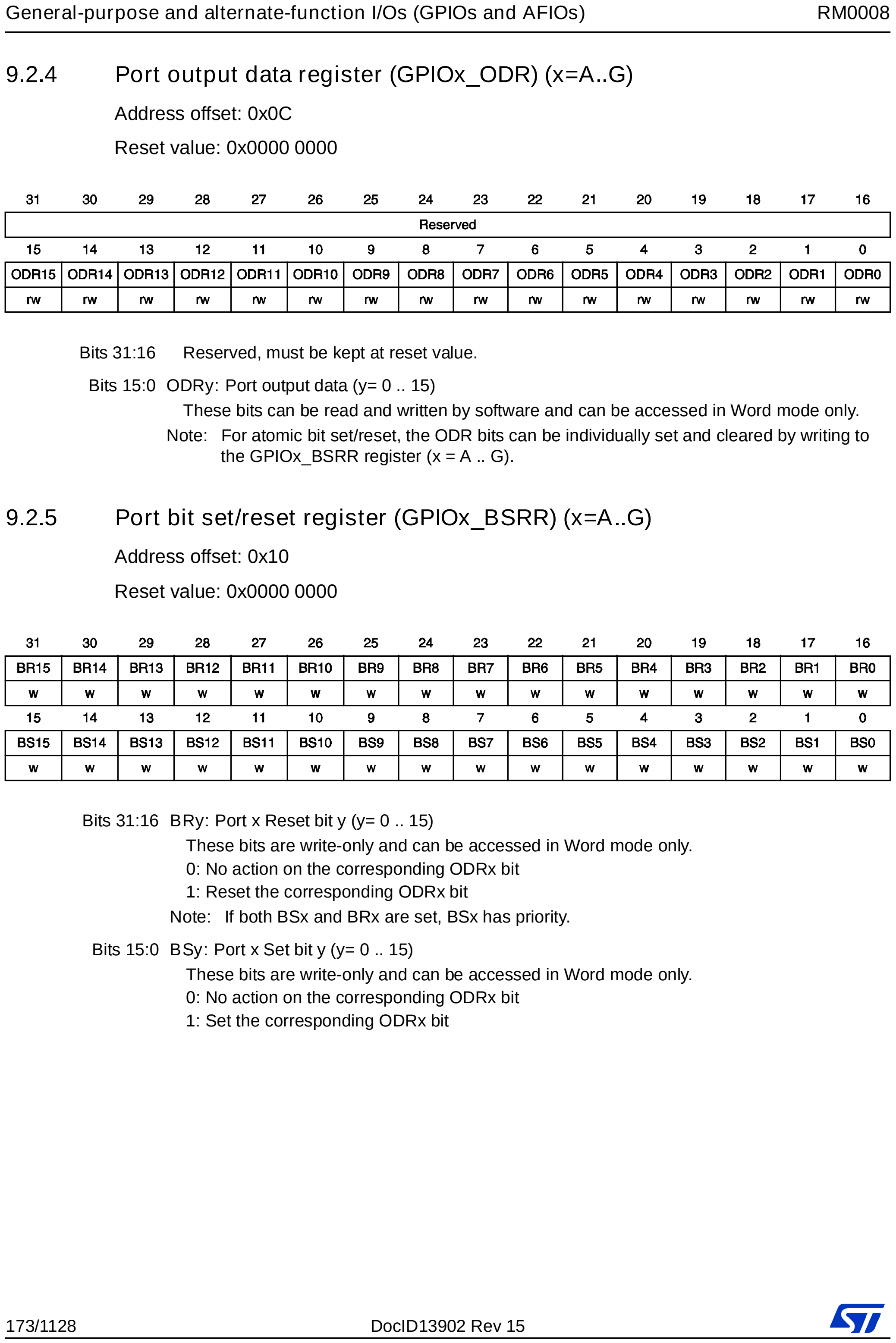
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

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

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.

DocID13902 Rev 15 172/1128



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

9.2.4 Port output data register (GPIOx_ODR) (x=A..G)

Address offset: 0x0C

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 and can be accessed in Word mode only.

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

9.2.5 Port bit set/reset register (GPIOx_BSRR) (x=A..G)

Address offset: 0x10

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)

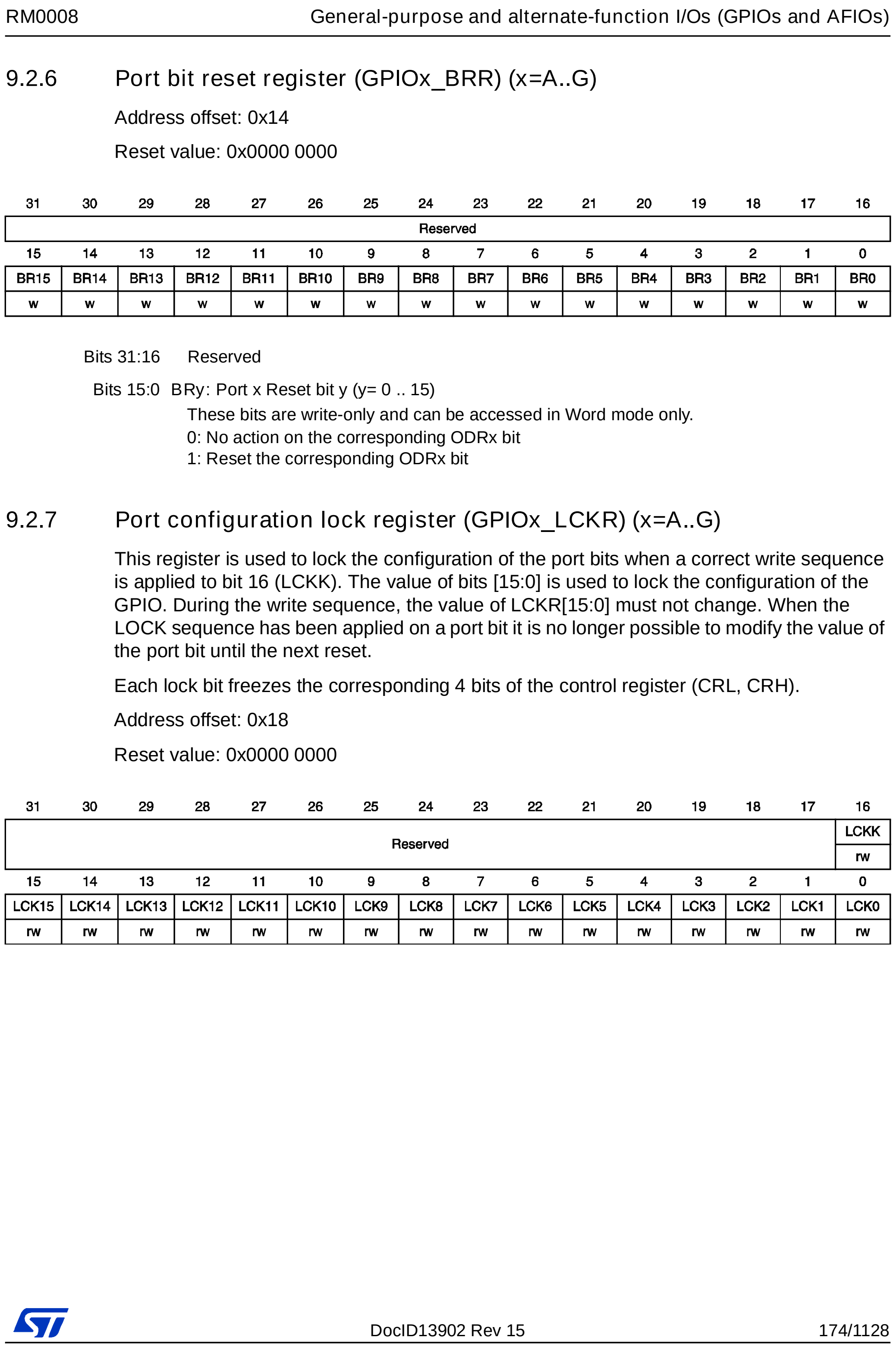
These bits are write-only and can be accessed in Word mode only.
0: No action on the corresponding ODRx bit
1: Reset the corresponding ODRx bit

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 mode only.
0: No action on the corresponding ODRx bit
1: Set the corresponding ODRx bit

173/1128 DocID13902 Rev 15



9.2.6 Port bit reset register (GPIOx_BRR) (x=A..G)

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
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

Bits 31:16 Reserved

Bits 15:0 BRy: Port x Reset bit y (y= 0 .. 15)
These bits are write-only and can be accessed in Word mode only.
0: No action on the corresponding ODRx bit
1: Reset the corresponding ODRx bit

9.2.7 Port configuration lock register (GPIOx_LCKR) (x=A..G)

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 it is no longer possible to modify the value of the port bit until the next reset.

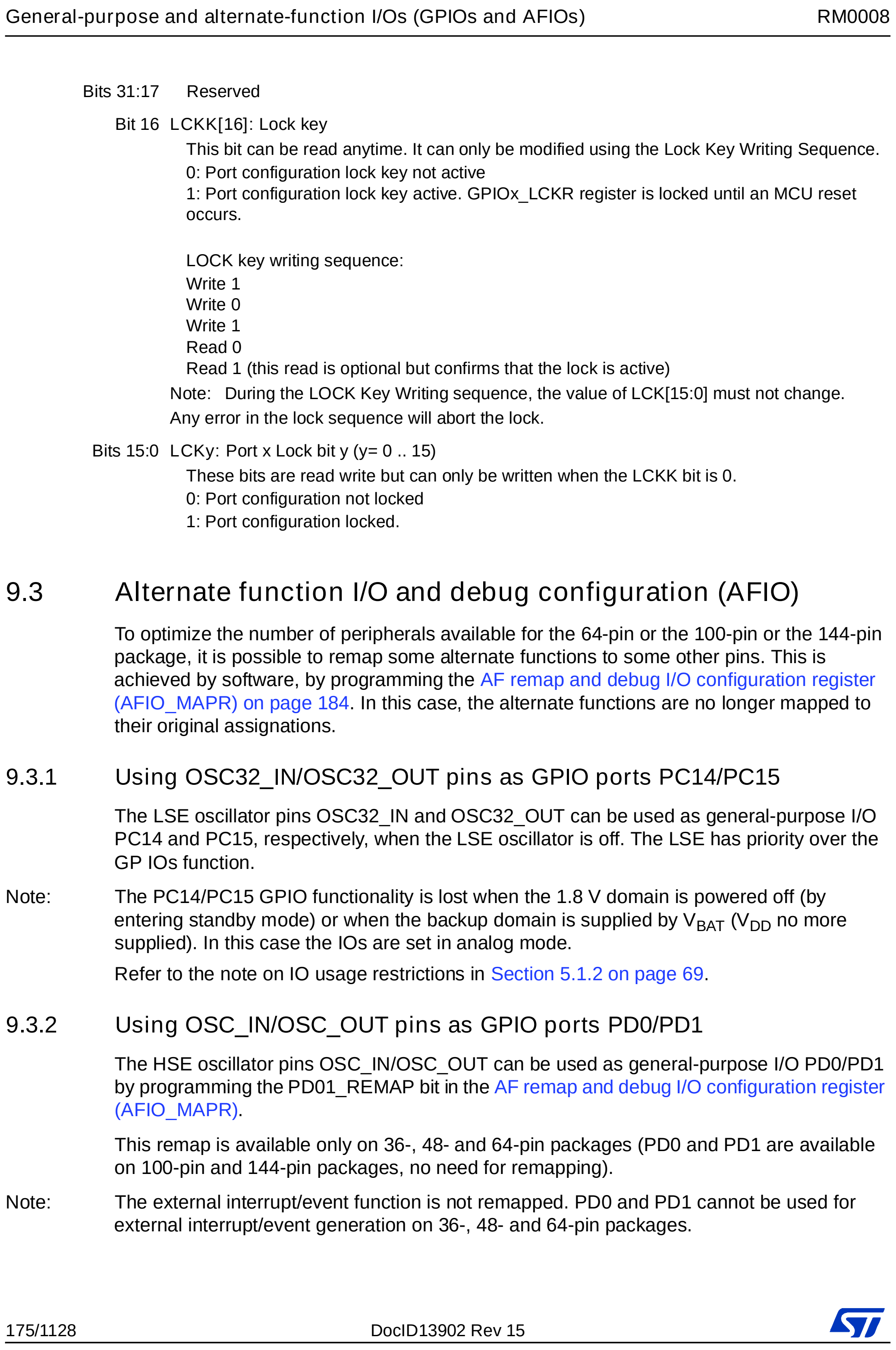
Each lock bit freezes the corresponding 4 bits of the control register (CRL, CRH).

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LCKK
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

Bits 31:17 Reserved

Bit 16 LCKK[16]: Lock key

This bit can be read anytime. It can only be modified using the Lock Key Writing Sequence.
0: Port configuration lock key not active
1: Port configuration lock key active. GPIOx_LCKR register is locked until an MCU reset occurs.

LOCK key writing sequence:
Write 1
Write 0
Write 1
Read 0
Read 1 (this read is optional but confirms that the lock is active)

Note: During the LOCK Key Writing sequence, the value of LCK[15:0] must not change.
Any error in the lock sequence will abort the lock.

Bits 15:0 LCKy: Port x Lock bit y (y= 0 .. 15)

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.

9.3 Alternate function I/O and debug configuration (AFIO)

To optimize the number of peripherals available for the 64-pin or the 100-pin or the 144-pin package, it is possible to remap some alternate functions to some other pins. This is achieved by software, by programming the AF remap and debug I/O configuration register (AFIO_MAPR) on page 184. In this case, the alternate functions are no longer mapped to their original assignations.

9.3.1 Using OSC32_IN/OSC32_OUT pins as GPIO ports PC14/PC15

The LSE oscillator pins OSC32_IN and OSC32_OUT can be used as general-purpose I/O PC14 and PC15, respectively, when the LSE oscillator is off. The LSE has priority over the GP IOs function.

Note: The PC14/PC15 GPIO functionality is lost when the 1.8 V domain is powered off (by entering standby mode) or when the backup domain is supplied by V_{BAT} (V_{DD} no more supplied). In this case the IOs are set in analog mode.

Refer to the note on IO usage restrictions in Section 5.1.2 on page 69.

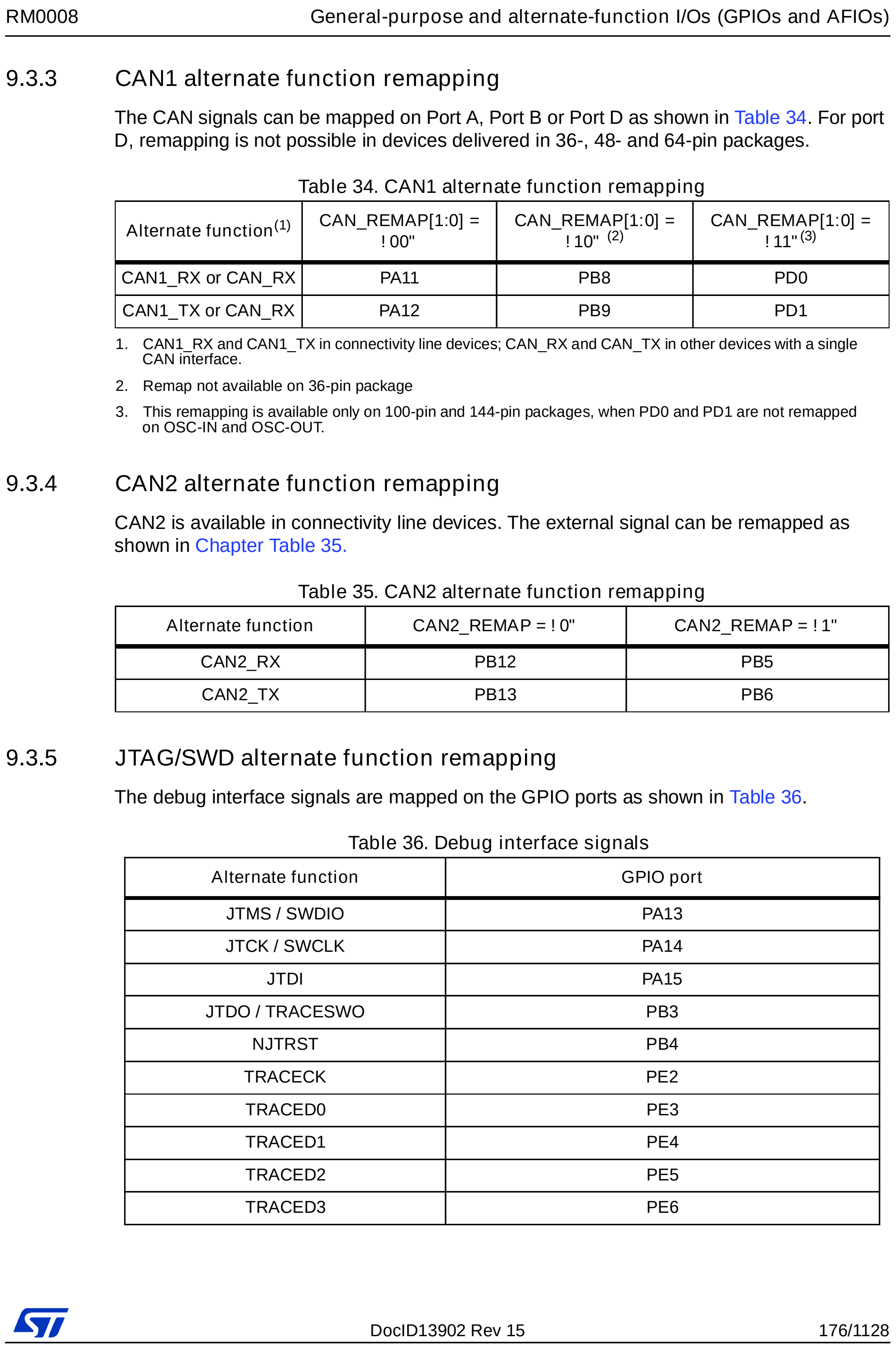
9.3.2 Using OSC_IN/OSC_OUT pins as GPIO ports PD0/PD1

The HSE oscillator pins OSC_IN/OSC_OUT can be used as general-purpose I/O PD0/PD1 by programming the PD01_REMAP bit in the AF remap and debug I/O configuration register (AFIO_MAPR).

This remap is available only on 36-, 48- and 64-pin packages (PD0 and PD1 are available on 100-pin and 144-pin packages, no need for remapping).

Note: The external interrupt/event function is not remapped. PD0 and PD1 cannot be used for external interrupt/event generation on 36-, 48- and 64-pin packages.

175/1128 DocID13902 Rev 15



9.3.3 CAN1 alternate function remapping

The CAN signals can be mapped on Port A, Port B or Port D as shown in Table 34. For port D, remapping is not possible in devices delivered in 36-, 48- and 64-pin packages.

Table 34. CAN1 alternate function remapping

Alternate function[1]	CAN_REMAP[1:0] = ! 00"	CAN_REMAP[1:0] = ! 10" [2]	CAN_REMAP[1:0] = ! 11" [3]
CAN1_RX or CAN_RX	PA11	PB8	PD0
CAN1_TX or CAN_RX	PA12	PB9	PD1

1. CAN1_RX and CAN1_TX in connectivity line devices; CAN_RX and CAN_TX in other devices with a single CAN interface.
2. Remap not available on 36-pin package
3. This remapping is available only on 100-pin and 144-pin packages, when PD0 and PD1 are not remapped on OSC-IN and OSC-OUT.

9.3.4 CAN2 alternate function remapping

CAN2 is available in connectivity line devices. The external signal can be remapped as shown in Chapter Table 35.

Table 35. CAN2 alternate function remapping

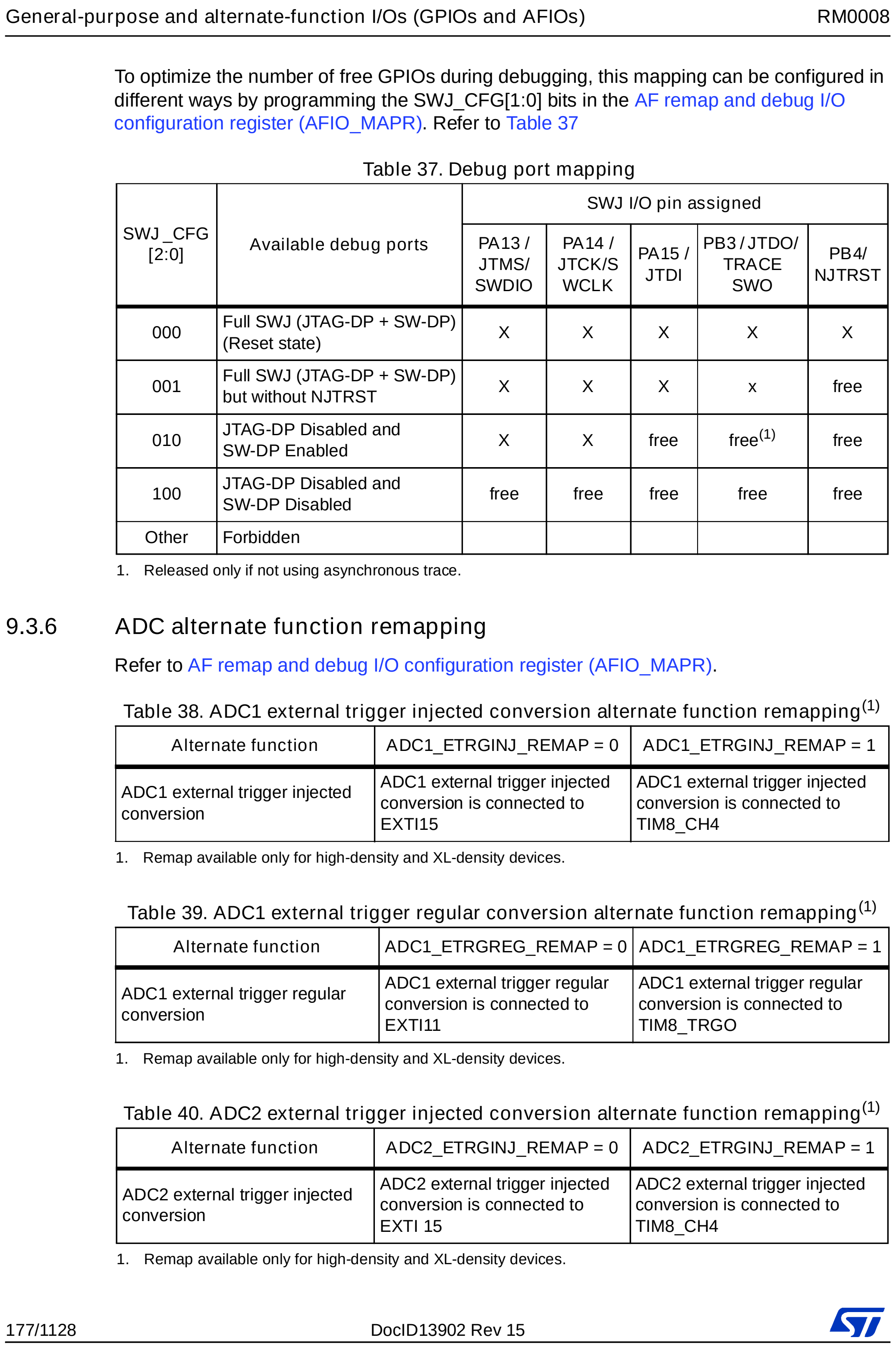
Alternate function	CAN2_REMAP = ! 0"	CAN2_REMAP = ! 1"
CAN2_RX	PB12	PB5
CAN2_TX	PB13	PB6

9.3.5 JTAG/SWD alternate function remapping

The debug interface signals are mapped on the GPIO ports as shown in Table 36.

Table 36. Debug interface signals

Alternate function	GPIO port
JTMS / SWDIO	PA13
JTCK / SWCLK	PA14
JTDI	PA15
JTDO / TRACESWO	PB3
NJTRST	PB4
TRACECK	PE2
TRACED0	PE3
TRACED1	PE4
TRACED2	PE5
TRACED3	PE6



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

To optimize the number of free GPIOs during debugging, this mapping can be configured in different ways by programming the SWJ_CFG[1:0] bits in the AF remap and debug I/O configuration register (AFIO_MAPR). Refer to Table 37

Table 37. Debug port mapping

SWJ _CFG [2:0]	Available debug ports	SWJ I/O pin assigned				
		PA13 / JTMS/ SWDIO	PA14 / JTCK/S WCLK	PA15 / JTDI	PB3 / JTDO/ TRACE SWO	PB4/ NJTRST
000	Full SWJ (JTAG-DP + SW-DP) (Reset state)	X	X	X	X	X
001	Full SWJ (JTAG-DP + SW-DP) but without NJTRST	X	X	X	x	free
010	JTAG-DP Disabled and SW-DP Enabled	X	X	free	free(1)	free
100	JTAG-DP Disabled and SW-DP Disabled	free	free	free	free	free
Other	Forbidden					

1. Released only if not using asynchronous trace.

9.3.6 ADC alternate function remapping

Refer to AF remap and debug I/O configuration register (AFIO_MAPR).

Table 38. ADC1 external trigger injected conversion alternate function remapping(1)

Alternate function	ADC1_ETRGINJ_REMAP = 0	ADC1_ETRGINJ_REMAP = 1
ADC1 external trigger injected conversion	ADC1 external trigger injected conversion is connected to EXTI15	ADC1 external trigger injected conversion is connected to TIM8_CH4

1. Remap available only for high-density and XL-density devices.

Table 39. ADC1 external trigger regular conversion alternate function remapping(1)

Alternate function	ADC1_ETRGREG_REMAP = 0	ADC1_ETRGREG_REMAP = 1
ADC1 external trigger regular conversion	ADC1 external trigger regular conversion is connected to EXTI11	ADC1 external trigger regular conversion is connected to TIM8_TRGO

1. Remap available only for high-density and XL-density devices.

Table 40. ADC2 external trigger injected conversion alternate function remapping(1)

Alternate function	ADC2_ETRGINJ_REMAP = 0	ADC2_ETRGINJ_REMAP = 1
ADC2 external trigger injected conversion	ADC2 external trigger injected conversion is connected to EXTI 15	ADC2 external trigger injected conversion is connected to TIM8_CH4

1. Remap available only for high-density and XL-density devices.

177/1128 DocID13902 Rev 15

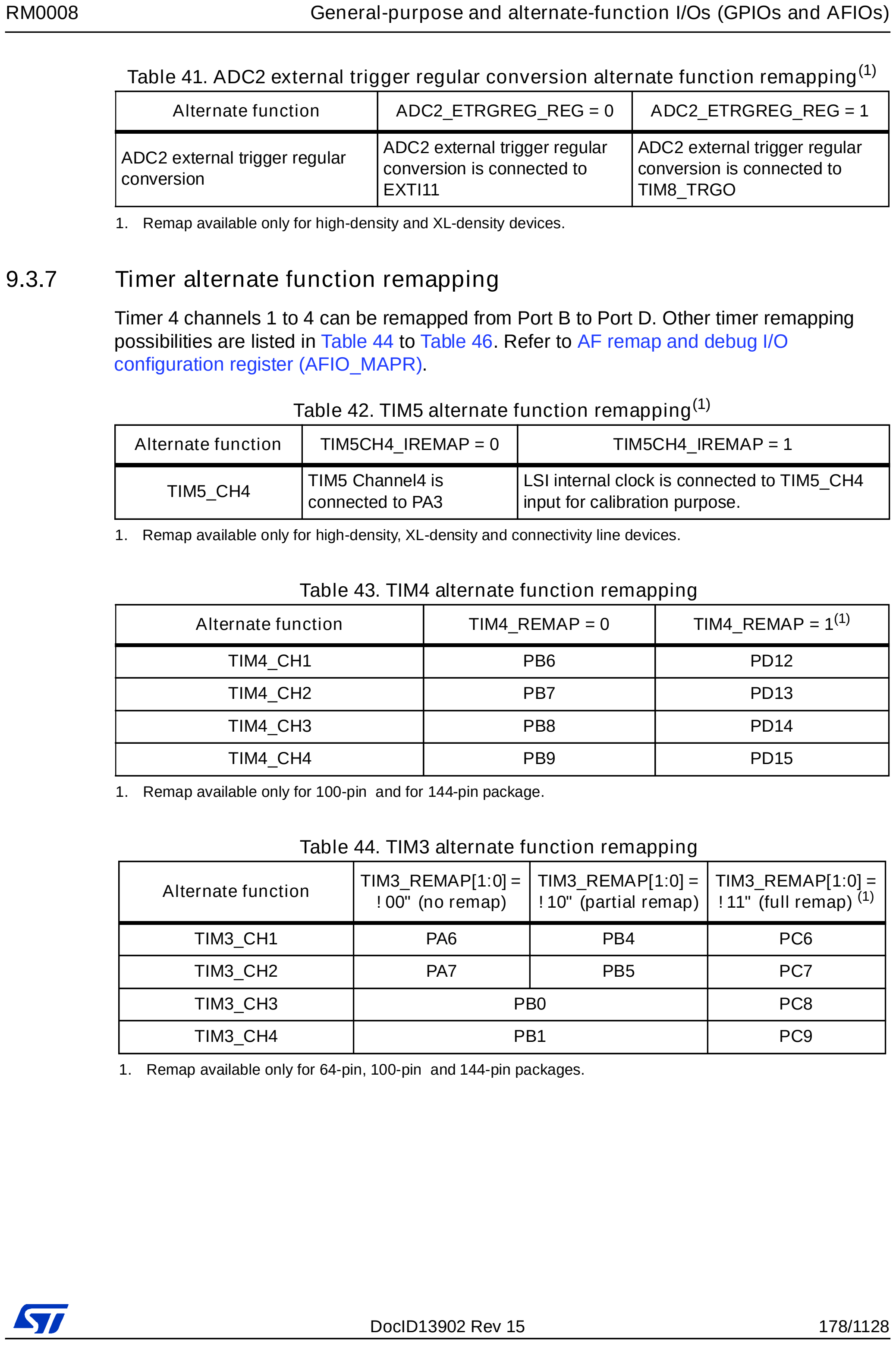


Table 41. ADC2 external trigger regular conversion alternate function remapping[1]

Alternate function	ADC2_ETRGREG_REG = 0	ADC2_ETRGREG_REG = 1
ADC2 external trigger regular conversion	ADC2 external trigger regular conversion is connected to EXTI11	ADC2 external trigger regular conversion is connected to TIM8_TRGO

1. Remap available only for high-density and XL-density devices.

9.3.7 Timer alternate function remapping

Timer 4 channels 1 to 4 can be remapped from Port B to Port D. Other timer remapping possibilities are listed in Table 44 to Table 46. Refer to AF remap and debug I/O configuration register (AFIO_MAPR).

Table 42. TIM5 alternate function remapping[1]

Alternate function	TIM5CH4_IREMAP = 0	TIM5CH4_IREMAP = 1
TIM5_CH4	TIM5 Channel4 is connected to PA3	LSI internal clock is connected to TIM5_CH4 input for calibration purpose.

1. Remap available only for high-density, XL-density and connectivity line devices.

Table 43. TIM4 alternate function remapping

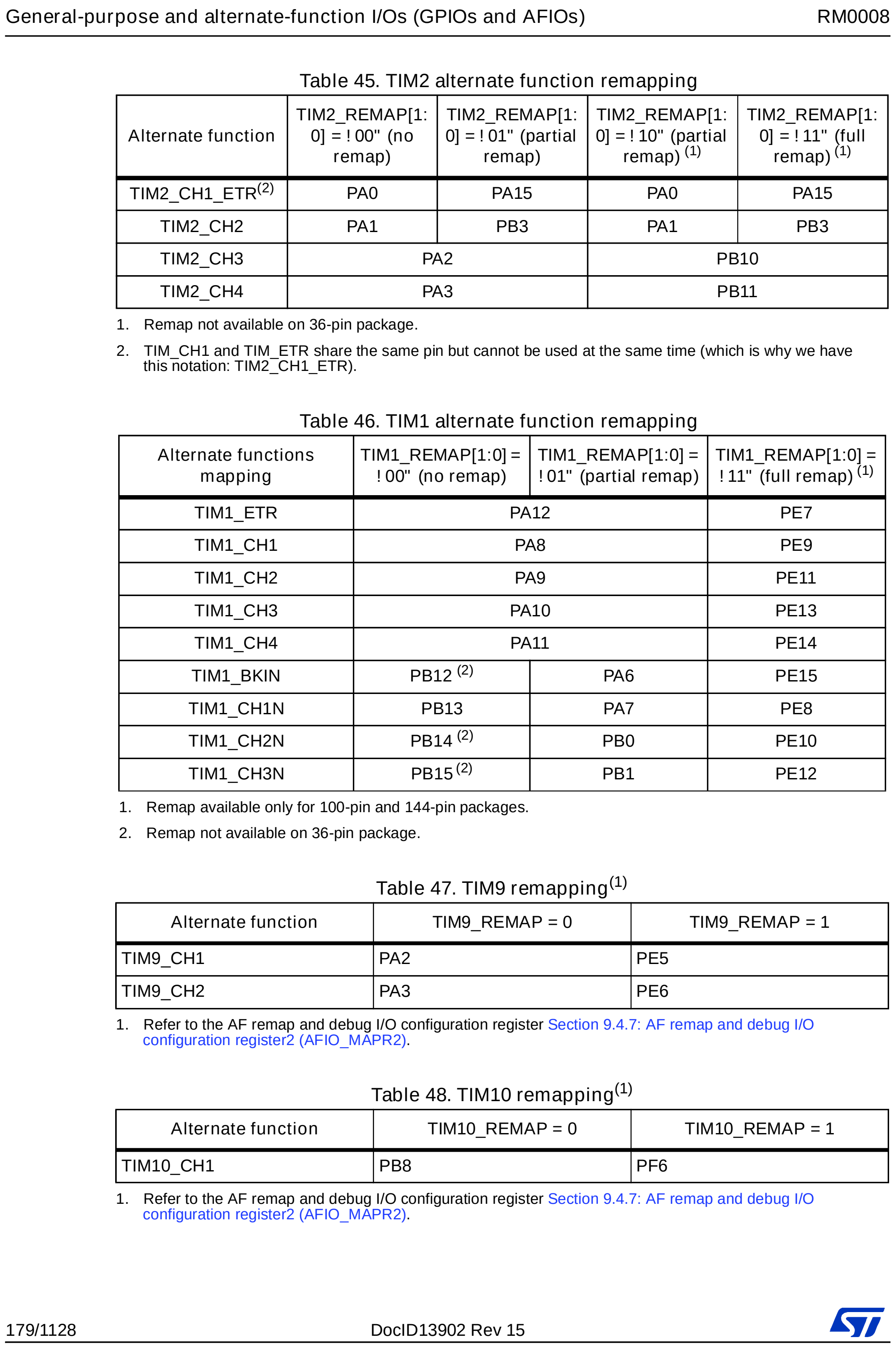
Alternate function	TIM4_REMAP = 0	TIM4_REMAP = 1[1]
TIM4_CH1	PB6	PD12
TIM4_CH2	PB7	PD13
TIM4_CH3	PB8	PD14
TIM4_CH4	PB9	PD15

1. Remap available only for 100-pin and for 144-pin package.

Table 44. TIM3 alternate function remapping

Alternate function	TIM3_REMAP[1:0] = ! 00" (no remap)	TIM3_REMAP[1:0] = ! 10" (partial remap)	TIM3_REMAP[1:0] = ! 11" (full remap) [1]
TIM3_CH1	PA6	PB4	PC6
TIM3_CH2	PA7	PB5	PC7
TIM3_CH3	PB0		PC8
TIM3_CH4	PB1		PC9

1. Remap available only for 64-pin, 100-pin and 144-pin packages.



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

Table 45. TIM2 alternate function remapping

Alternate function	TIM2_REMAP[1:0] = ! 00" (no remap)	TIM2_REMAP[1:0] = ! 01" (partial remap)	TIM2_REMAP[1:0] = ! 10" (partial remap) [1]	TIM2_REMAP[1:0] = ! 11" (full remap) [1]
TIM2_CH1_ETR[2]	PA0	PA15	PA0	PA15
TIM2_CH2	PA1	PB3	PA1	PB3
TIM2_CH3	PA2		PB10	
TIM2_CH4	PA3		PB11	

1. Remap not available on 36-pin package.
2. TIM_CH1 and TIM_ETR share the same pin but cannot be used at the same time (which is why we have this notation: TIM2_CH1_ETR).

Table 46. TIM1 alternate function remapping

Alternate functions mapping	TIM1_REMAP[1:0] = ! 00" (no remap)	TIM1_REMAP[1:0] = ! 01" (partial remap)	TIM1_REMAP[1:0] = ! 11" (full remap) [1]
TIM1_ETR	PA12		PE7
TIM1_CH1	PA8		PE9
TIM1_CH2	PA9		PE11
TIM1_CH3	PA10		PE13
TIM1_CH4	PA11		PE14
TIM1_BKIN	PB12 [2]	PA6	PE15
TIM1_CH1N	PB13	PA7	PE8
TIM1_CH2N	PB14 [2]	PB0	PE10
TIM1_CH3N	PB15 [2]	PB1	PE12

1. Remap available only for 100-pin and 144-pin packages.
2. Remap not available on 36-pin package.

Table 47. TIM9 remapping[1]

Alternate function	TIM9_REMAP = 0	TIM9_REMAP = 1
TIM9_CH1	PA2	PE5
TIM9_CH2	PA3	PE6

1. Refer to the AF remap and debug I/O configuration register Section 9.4.7: AF remap and debug I/O configuration register2 (AFIO_MAPR2).

Table 48. TIM10 remapping[1]

Alternate function	TIM10_REMAP = 0	TIM10_REMAP = 1
TIM10_CH1	PB8	PF6

1. Refer to the AF remap and debug I/O configuration register Section 9.4.7: AF remap and debug I/O configuration register2 (AFIO_MAPR2).

179/1128 DocID13902 Rev 15

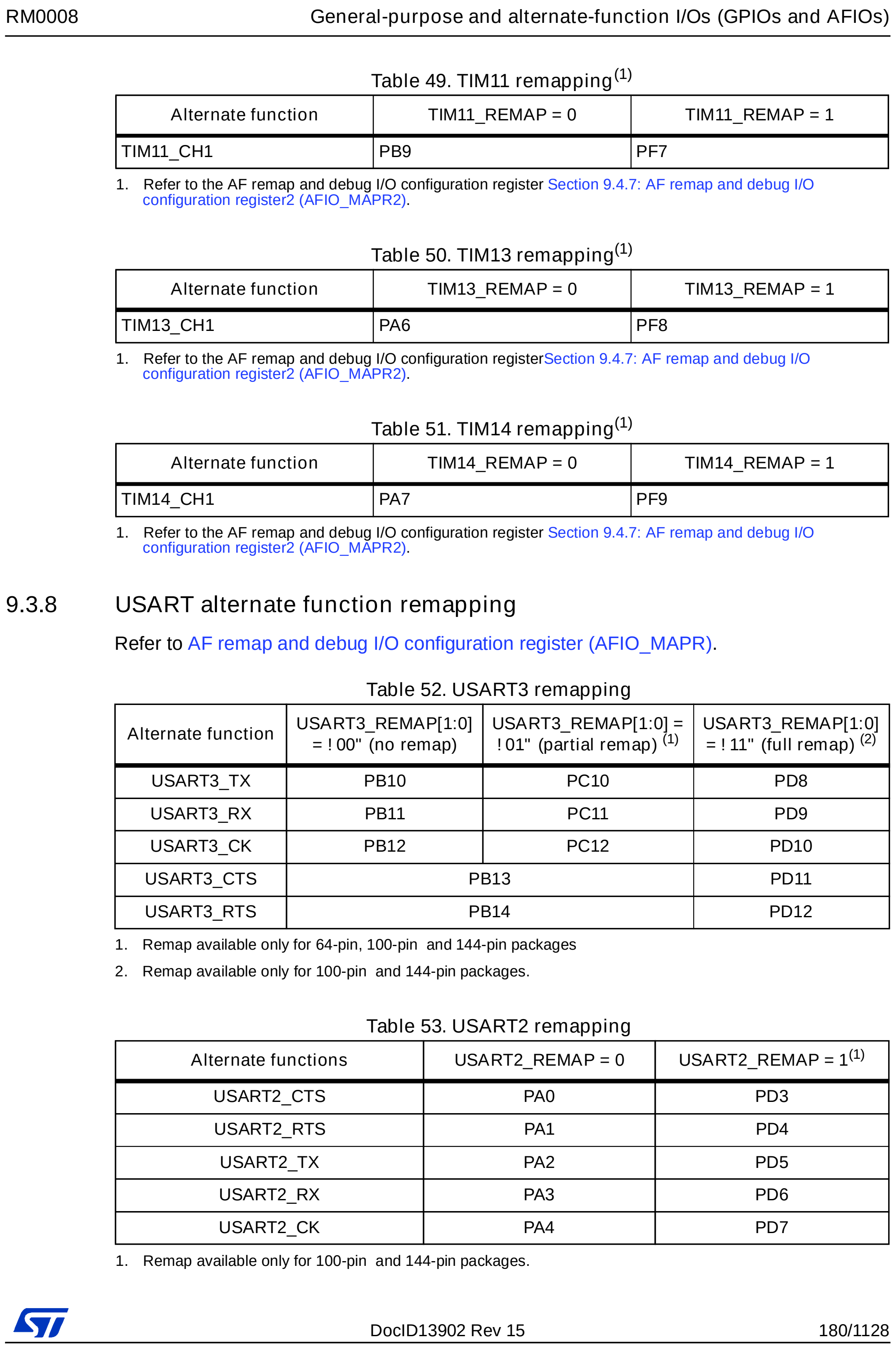


Table 49. TIM11 remapping[1]

Alternate function	TIM11_REMAP = 0	TIM11_REMAP = 1
TIM11_CH1	PB9	PF7

1. Refer to the AF remap and debug I/O configuration register Section 9.4.7: AF remap and debug I/O configuration register2 (AFIO_MAPR2).

Table 50. TIM13 remapping[1]

Alternate function	TIM13_REMAP = 0	TIM13_REMAP = 1
TIM13_CH1	PA6	PF8

1. Refer to the AF remap and debug I/O configuration registerSection 9.4.7: AF remap and debug I/O configuration register2 (AFIO_MAPR2).

Table 51. TIM14 remapping[1]

Alternate function	TIM14_REMAP = 0	TIM14_REMAP = 1
TIM14_CH1	PA7	PF9

1. Refer to the AF remap and debug I/O configuration register Section 9.4.7: AF remap and debug I/O configuration register2 (AFIO_MAPR2).

9.3.8 USART alternate function remapping

Refer to AF remap and debug I/O configuration register (AFIO_MAPR).

Table 52. USART3 remapping

Alternate function	USART3_REMAP[1:0] = ! 00" (no remap)	USART3_REMAP[1:0] = ! 01" (partial remap) [1]	USART3_REMAP[1:0] = ! 11" (full remap) [2]
USART3_TX	PB10	PC10	PD8
USART3_RX	PB11	PC11	PD9
USART3_CK	PB12	PC12	PD10
USART3_CTS	PB13		PD11
USART3_RTS	PB14		PD12

1. Remap available only for 64-pin, 100-pin and 144-pin packages
2. Remap available only for 100-pin and 144-pin packages.

Table 53. USART2 remapping

Alternate functions	USART2_REMAP = 0	USART2_REMAP = 1[1]
USART2_CTS	PA0	PD3
USART2_RTS	PA1	PD4
USART2_TX	PA2	PD5
USART2_RX	PA3	PD6
USART2_CK	PA4	PD7

1. Remap available only for 100-pin and 144-pin packages.

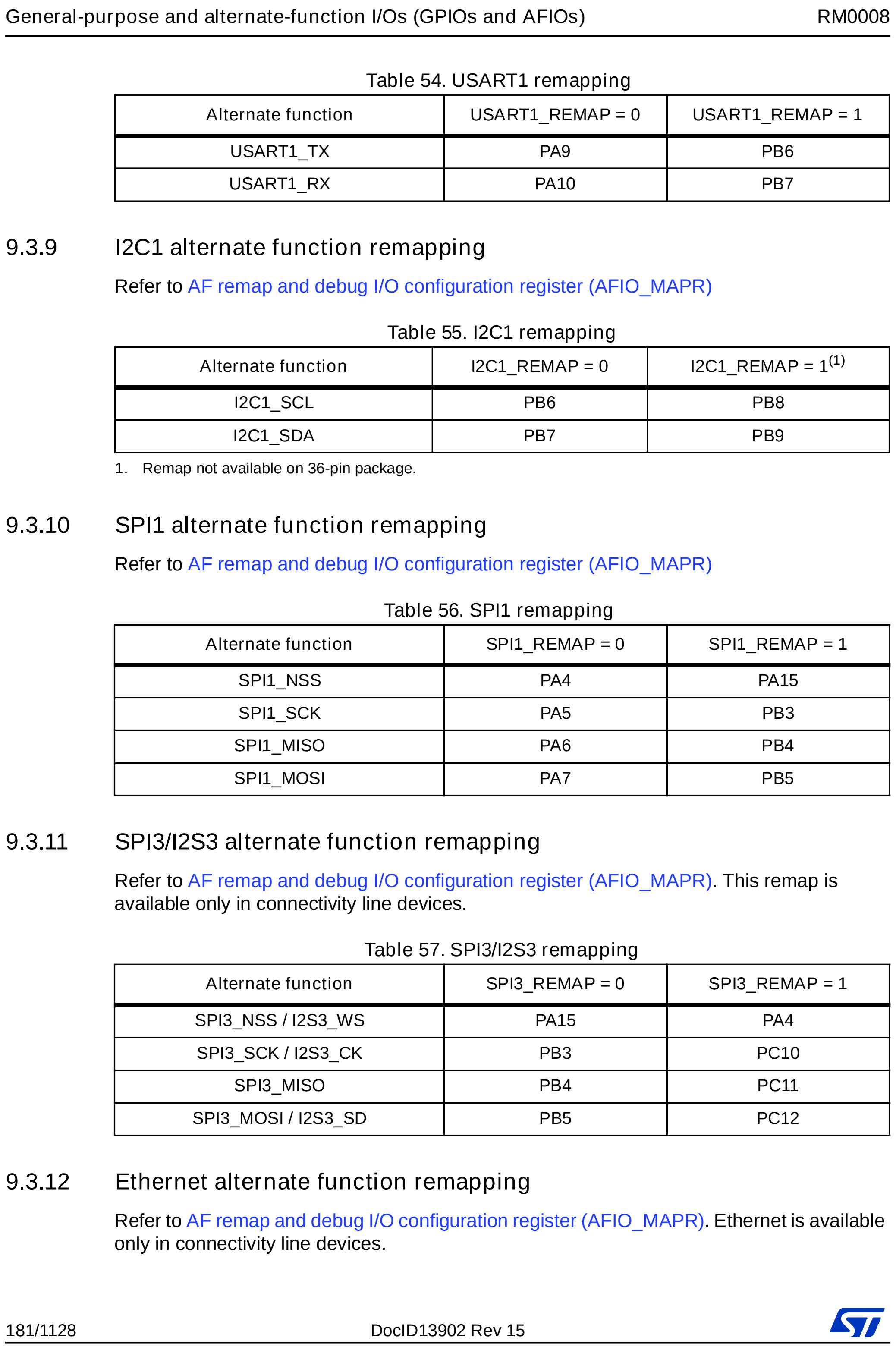


Table 54. USART1 remapping

Alternate function	USART1_REMAP = 0	USART1_REMAP = 1
USART1_TX	PA9	PB6
USART1_RX	PA10	PB7

9.3.9 I2C1 alternate function remapping

Refer to AF remap and debug I/O configuration register (AFIO_MAPR)

Table 55. I2C1 remapping

Alternate function	I2C1_REMAP = 0	I2C1_REMAP = 1[1]
I2C1_SCL	PB6	PB8
I2C1_SDA	PB7	PB9

[1] Remap not available on 36-pin package.

9.3.10 SPI1 alternate function remapping

Refer to AF remap and debug I/O configuration register (AFIO_MAPR)

Table 56. SPI1 remapping

Alternate function	SPI1_REMAP = 0	SPI1_REMAP = 1
SPI1_NSS	PA4	PA15
SPI1_SCK	PA5	PB3
SPI1_MISO	PA6	PB4
SPI1_MOSI	PA7	PB5

9.3.11 SPI3/I2S3 alternate function remapping

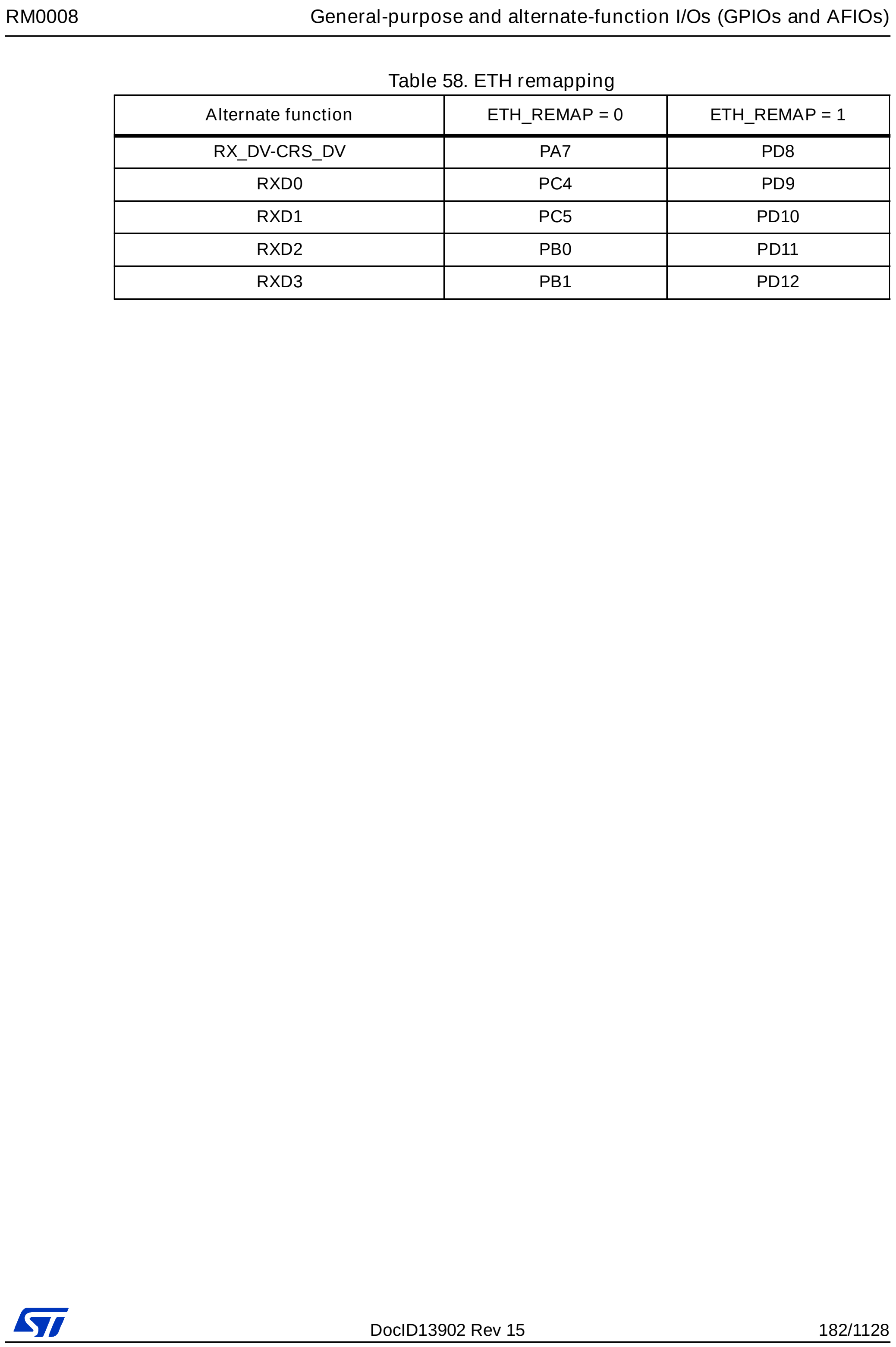
Refer to AF remap and debug I/O configuration register (AFIO_MAPR). This remap is available only in connectivity line devices.

Table 57. SPI3/I2S3 remapping

Alternate function	SPI3_REMAP = 0	SPI3_REMAP = 1
SPI3_NSS / I2S3_WS	PA15	PA4
SPI3_SCK / I2S3_CK	PB3	PC10
SPI3_MISO	PB4	PC11
SPI3_MOSI / I2S3_SD	PB5	PC12

9.3.12 Ethernet alternate function remapping

Refer to AF remap and debug I/O configuration register (AFIO_MAPR). Ethernet is available only in connectivity line devices.

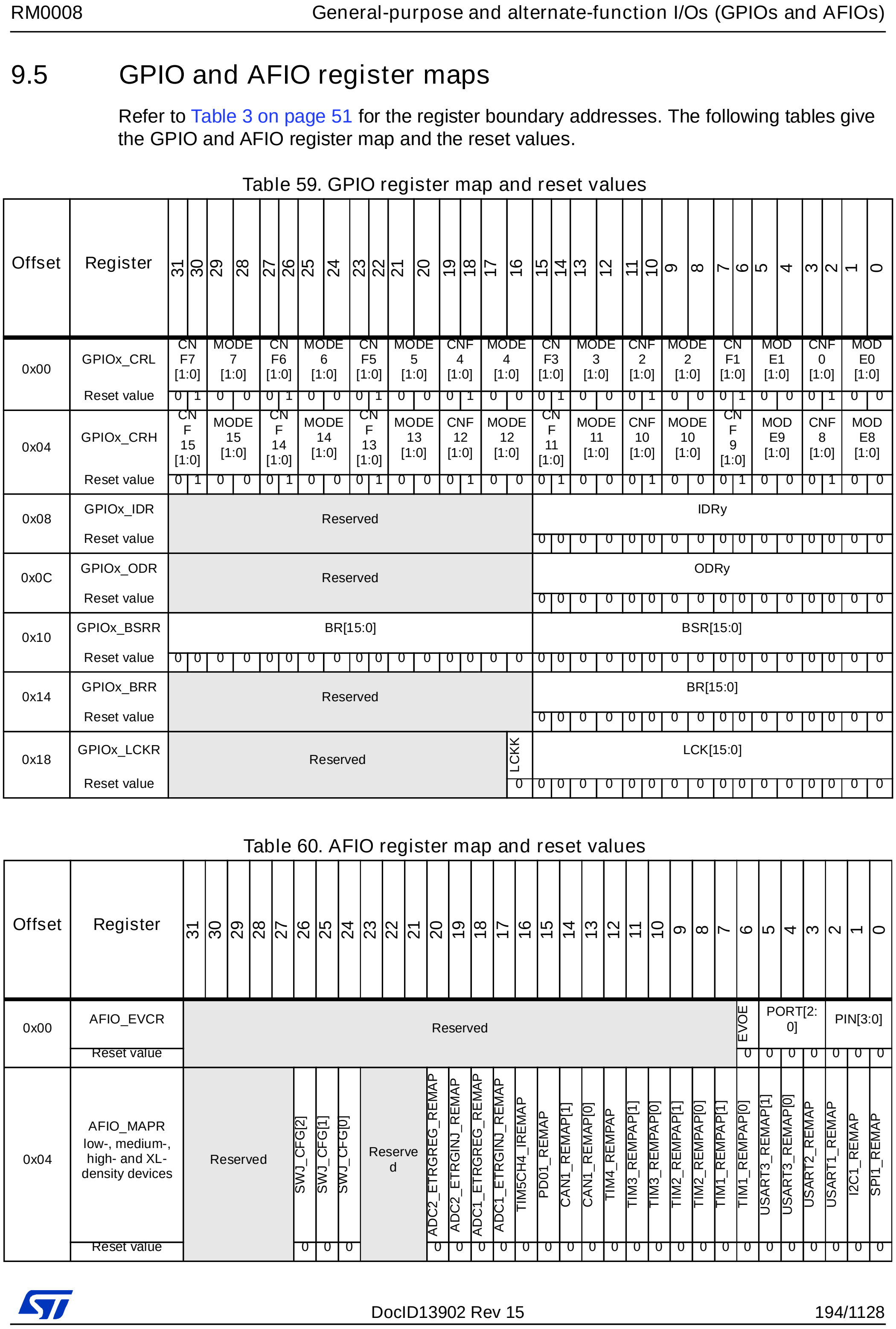


RM0008 General-purpose and alternate-function I/Os (GPIOs and AFIOs)

Table 58. ETH remapping

Alternate function	ETH_REMAP = 0	ETH_REMAP = 1
RX_DV-CRS_DV	PA7	PD8
RXD0	PC4	PD9
RXD1	PC5	PD10
RXD2	PB0	PD11
RXD3	PB1	PD12

DocID13902 Rev 15 182/1128



9.5 GPIO and AFIO register maps

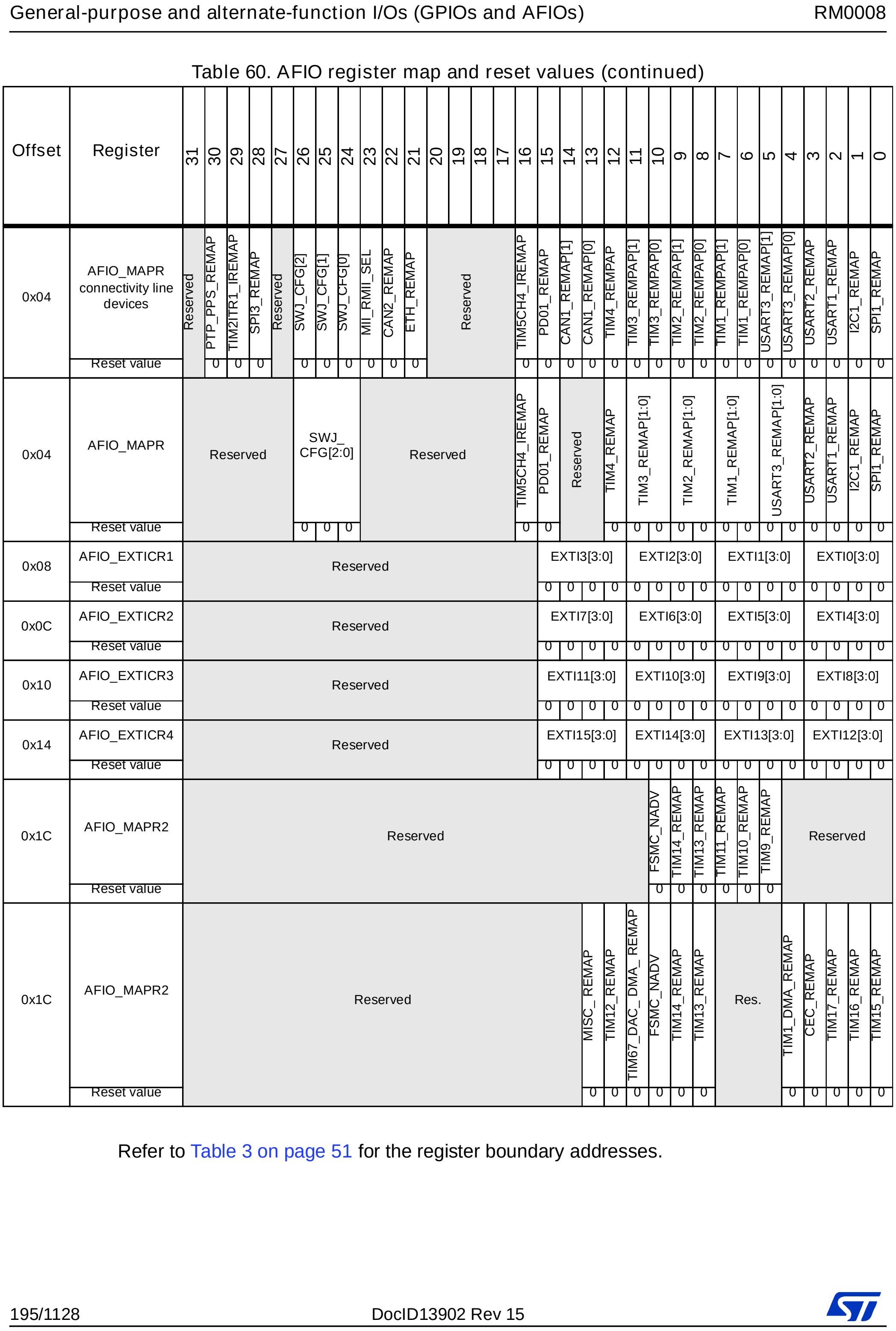
Refer to Table 3 on page 51 for the register boundary addresses. The following tables give the GPIO and AFIO register map and the reset values.

Table 59. GPIO 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	GPIOx_CRL	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]	
	Reset value	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0x04	GPIOx_CRH	CNF15 [1:0]		MODE15 [1:0]		CNF14 [1:0]		MODE14 [1:0]		CNF13 [1:0]		MODE13 [1:0]		CNF12 [1:0]		MODE12 [1:0]		CNF11 [1:0]		MODE11 [1:0]		CNF10 [1:0]		MODE10 [1:0]		CNF9 [1:0]		MODE9 [1:0]		CNF8 [1:0]		MODE8 [1:0]	
	Reset value	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
0x08	GPIOx_IDR	Reserved																IDRy															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	GPIOx_ODR	Reserved																ODRy															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	GPIOx_BSRR	BR[15:0]																BSR[15:0]															
	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	0	0	0
0x14	GPIOx_BRR	Reserved																BR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_LCKR	Reserved															LCKK	LCK[15:0]															
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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
0x00	AFIO_EVCR	Reserved																									EVOE	PORT[2:0]			PIN[3:0]		
	Reset value																										0	0	0	0	0	0	0
0x04	AFIO_MAPR low-, medium-, high- and XL-density devices	Reserved					SWJ_CFG[2]	SWJ_CFG[1]	SWJ_CFG[0]	Reserved			ADC2_ETRGREG_REMAP	ADC2_ETRGINJ_REMAP	ADC1_ETRGREG_REMAP	ADC1_ETRGINJ_REMAP	TIM5CH4_IREMAP	PD01_REMAP	CAN1_REMAP[1]	CAN1_REMAP[0]	TIM4_REMPAP	TIM3_REMPAP[1]	TIM3_REMPAP[0]	TIM2_REMPAP[1]	TIM2_REMPAP[0]	TIM1_REMPAP[1]	TIM1_REMPAP[0]	USART3_REMAP[1]	USART3_REMAP[0]	USART2_REMAP	USART1_REMAP	I2C1_REMAP	SPI1_REMAP
	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



General-purpose and alternate-function I/Os (GPIOs and AFIOs) RM0008

Table 60. AFIO register map and reset values (continued)

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
0x04	AFIO_MAPR connectivity line devices	Reserved	PTP_PPS_REMAP	TIM2ITR1_IREMAP	SPI3_REMAP	Reserved	SWJ_CFG[2]	SWJ_CFG[1]	SWJ_CFG[0]	MII_RMII_SEL	CAN2_REMAP	ETH_REMAP	Reserved				TIM5CH4_IREMAP	PD01_REMAP	CAN1_REMAP[1]	CAN1_REMAP[0]	TIM4_REMPAP	TIM3_REMPAP[1]	TIM3_REMPAP[0]	TIM2_REMPAP[1]	TIM2_REMPAP[0]	TIM1_REMPAP[1]	TIM1_REMPAP[0]	USART3_REMAP[1]	USART3_REMAP[0]	USART2_REMAP	USART1_REMAP	I2C1_REMAP	SPI1_REMAP
	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
0x04	AFIO_MAPR	Reserved					SWJ_ CFG[2:0]			Reserved							TIM5CH4_IREMAP	PD01_REMAP	Reserved		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
	Reset value						0	0	0								0	0			0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	AFIO_EXTICR1	Reserved																EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
	Reset value																	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]			
	Reset value																	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]			
	Reset value																	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]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	AFIO_MAPR2	Reserved																					FSMC_NADV	TIM14_REMAP	TIM13_REMAP	TIM11_REMAP	TIM10_REMAP	TIM9_REMAP	Reserved				
	Reset value																						0	0	0	0	0	0					
0x1C	AFIO_MAPR2	Reserved																		MISC_ REMAP	TIM12_REMAP	TIM67_DAC_ DMA_ REMAP	FSMC_NADV	TIM14_REMAP	TIM13_REMAP	Res.			TIM1_DMA_REMAP	CEC_REMAP	TIM17_REMAP	TIM16_REMAP	TIM15_REMAP
	Reset value																			0	0	0	0	0	0				0	0	0	0	0

Refer to Table 3 on page 51 for the register boundary addresses.

195/1128 DocID13902 Rev 15