TD de microprocesseurs - l'ADC

L'objectif du TD est d'étudier l'interfaçage avec le convertisseur analogique-numérique `_ADC1` inclu dans le microcontrôleur. Il s'agit d'un composant qui fournit une valeur numérique proportionnelle à la tension analogique en entrée.

$$N = 2^n \frac{V_e}{PE}$$

avec *PE* : pleine échelle (tension max d'entrée), *n* : résolution (configurable 6, 8, 10 ou 12 bits).

L'interface est définie dans les fichiers `lib/adc.h` et `lib/adc.c`. On en étudie la version simplifiée suivante :

```
#define ADC_MODE_12BITS          (0<<24)
#define ADC_MODE_10BITS          (1<<24)
#define ADC_MODE_8BITS           (2<<24)
#define ADC_MODE_6BITS           (3<<24)

#define ADC_MODE_SINGLE          (0)
#define ADC_MODE_CONTINUOUS      (1)
#define ADC_MODE_TRIGGERED       (2)

/* adc_init : intialize ADC */
int adc_init(ADC_t *adc, uint32_t mode, OnSample cb);

/* adc_channel_sample: sample the specified channel and generate an interrupt if a callblack
 * was specified on init, else operate in polling mode
 */
uint16_t adc_channel_sample(ADC_t *adc, uint32_t channel);
```

Le convertisseur peut prendre jusqu'à 16 canaux analogiques externes et 2 internes. Les potentiomètres sont connéctés aux broches PA0 et PA1 (schémas des cartes microcontrôleur et d'extension). Pour une tension d'alimentation de 3.3 V, la fréquence maximale de travail du convertisseur est de 30 MHz (voir la datasheet p. 110 – 12bit ADC characteristics).

1. Architecture et fonctionnalités

a. Analyser succintement les fonctionnalités du convertisseur à partir des parties 11.2 "ADC main features" et 11.3 "ADC functional description". Repérer, en particulier, sur la figure 31 p 212 les entrées analogiques, la différence de chemin entre les "regular channels" et les "injected channels".

b. Repérer les registres qui interviennent dans l'utilisation des "*regular channels*".

2. Initialisation de l'interface l'ADC

a. Indiquer le bit à configurer pour autoriser l'horloge de l'`_ADC1` et vérifier que les broches PA0 et PA1 sont associées aux canaux d'entrées 0 et 1.

b. L'horloge du bus APB2 est `sysclks.apb2_freq = 84 MHz`. En déduire la fréquence réelle de travail de l'ADC et la configuration CCR[17:16] du diviseur d'horloge.

c. Configuration du mode : *single* / *continuous* / *triggered*

La grille de configuration autorisée pour l'interface proposée est la suivante

	poll single	irq single	irq cont	irq triggered	commentaires
CR2[0]	1	1	1	1	ADC on
CR1[8]	1	1	1	1	convert input selected by SQR[ ] registers
CR2[1]	0	0	1	0	continuous conversion mode
CR1[11]	0	0	0	1	triggered conversion mode
callback	n	y	y	y	should there be a callback?
CR2[10]	1	1	1	1	EOC bit is set after each conversion
CR1[5]	0	1	1	1	generate an IRQ after each conversion

Compléter la fonction d'initialisation ci-dessous pour tenir compte de la configuration pour le mode *Single/polling*.

```
int adc_init(ADC_t *adc, uint32_t mode, OnSample cb)
{
        IRQn_tirq_number;
        uint32_t irq_priority;

        if (adc == _ADC1) {



                // configure pins
                io_configure(ADC1_GPIO_PORT, ADC1_GPIO_PINS, PIN_MODE_ANALOG, NULL);

                irq_number = ADC_IRQn;
                irq_priority = ADC1_IRQ_PRIORITY;
        } else { return -1; }

        for (int presc_cfg=0; presc_cfg<=3; presc_cfg++) {
                // set up ADCCLK prescaler config (2,4,6 or 8) so that Fadcclk < 30MHz
                if (sysclks.apb2_freq / (2*(presc_cfg+1)) < ADCCLK_MAX) {









                        // return ADCCLK
                        adc_freq = sysclks.apb2_freq / (2*(presc_cfg+1));
                        return (int)adc_freq;
                }
        }
        return -1;
}
```

3. Conversion d'un canal par polling

La fonction *adc_channel_sample*(*ADC_t *adc*, *uint32_t channel*) permet de convertir le canal passé en paramètre, soit en mode *Single/polling*, soit en mode *Single/Irq*. Dans le cadre du TD, on ne s'intéresse qu'au mode *Single/polling*. Dans ce mode, la fonction renvoie la valeur convertie.

a. Les registres SQR[] permettent d'indiquer le numéro du canal à convertir et le nombre de canaux à convertir. Identifier les parties des registres à modifier.

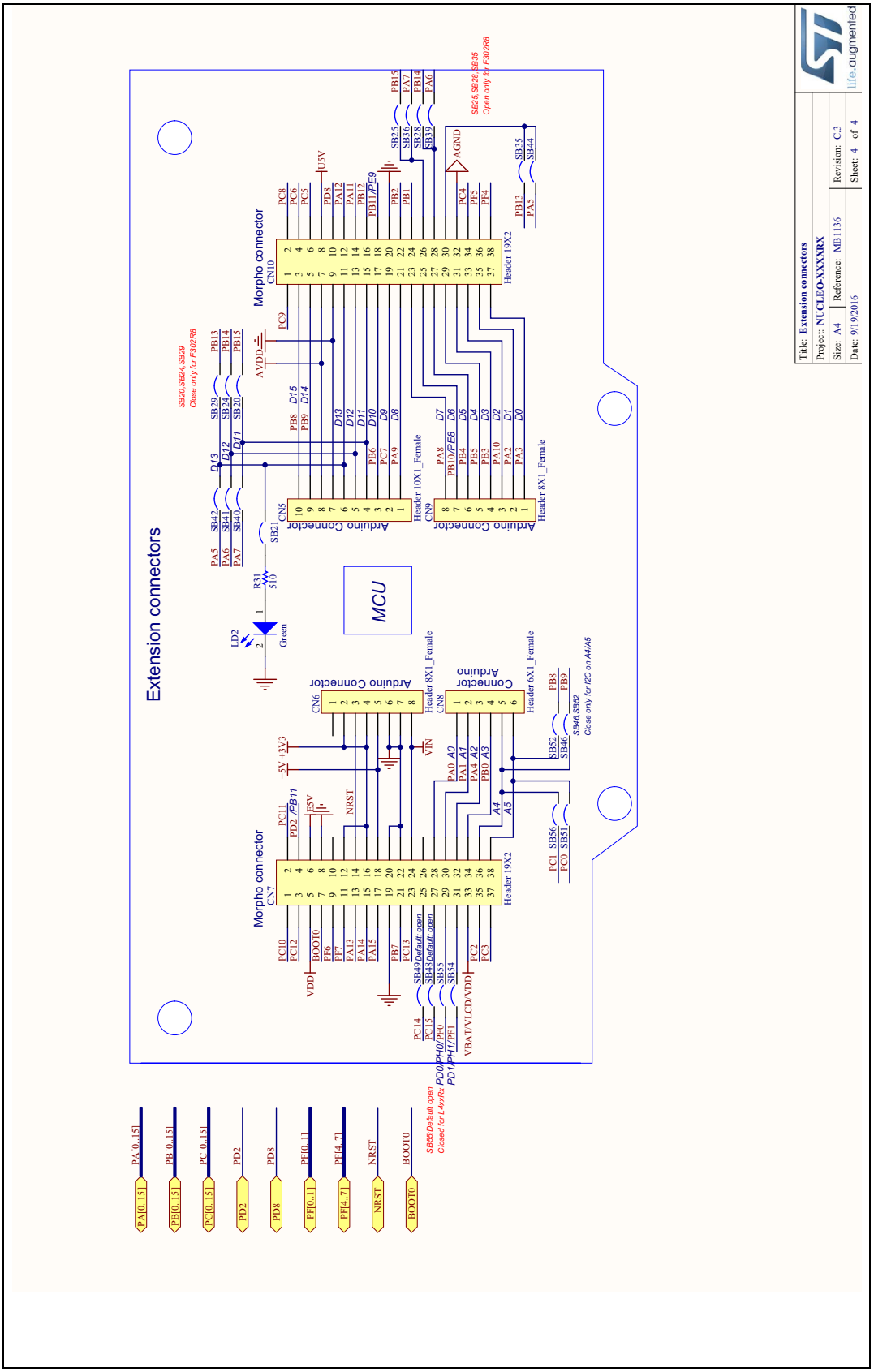
b. Indiquer comment démarrer une séquence de conversion.

c. Etant en polling, on s'appuie sur le registre d'état pour savoir quand la conversion est terminée. Indiquer le bit du registre d'état à tester.

d. Compléter le code ci-dessous pour gérer la conversion en mode polling

```
#define ADC_CHANNEL_TEMP                16
#define ADC_CHANNEL_VREFINT             17
#define ADC_CHANNEL_VBAT                18


uint16_t adc_channel_sample(ADC_t *adc, uint32_t channel)
{
        if (channel>18) return 0;

        n_to_convert = 1;

        /* special internal channels */
        if (channel == ADC_CHANNEL_TEMP) {
                ADC->CCR = (ADC->CCR & ~(3<<22)) | (2<<22);
        } else if (channel == ADC_CHANNEL_VREFINT) {
                ADC->CCR = (ADC->CCR & ~(3<<22)) | (2<<22);
        } else if (channel == ADC_CHANNEL_VBAT) {
                ADC->CCR = (ADC->CCR & ~(3<<22)) | (1<<22);
        }









}
```

Figure 30. Extension connectors



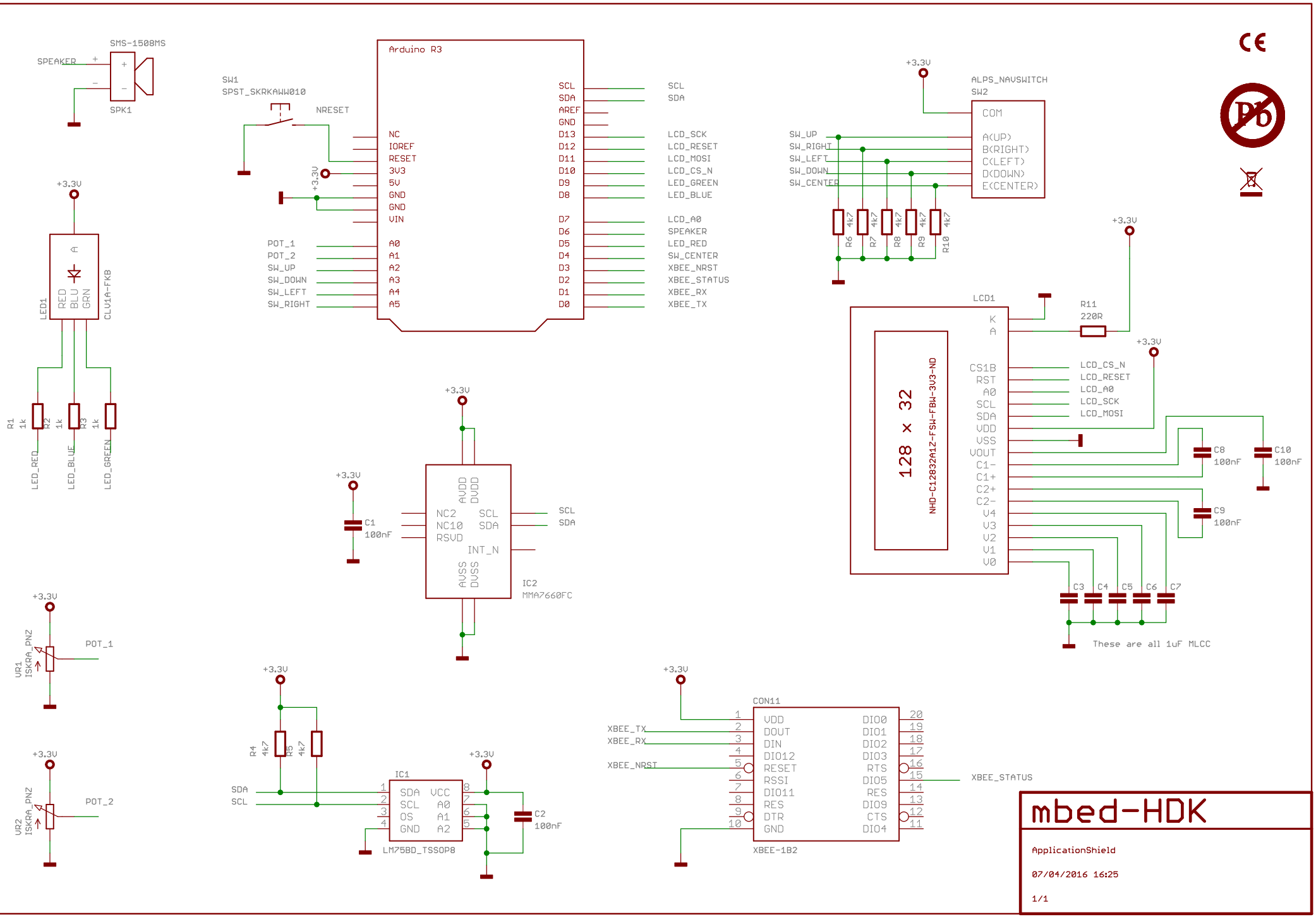

SMS-1508MS
SPEAKER
SPK1
Arduino R3
SW1
SPST_SKRKAWW010
NRESET
+3.3V
NC
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
SCL
SDA
AREF
GND
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
POT_1
POT_2
SW_UP
SW_DOWN
SW_LEFT
SW_RIGHT
LCD_SCK
LCD_RESET
LCD_MOSI
LCD_CS_N
LED_GREEN
LED_BLUE
LCD_A0
SPEAKER
LED_RED
SW_CENTER
XBEE_NRST
XBEE_STATUS
XBEE_RX
XBEE_TX
ALPS_NAVSWITCH
SW2
COM
A(UP)
B(RIGHT)
C(LEFT)
D(DOWN)
E(CENTER)
R6 4k7
R7 4k7
R8 4k7
R9 4k7
R10 4k7
LED1
RED
BLU
GRN
CLV1A-FKB
R1 1k
R2 1k
R3 1k
LED_RED
LED_BLUE
LED_GREEN
LCD1
128 x 32
NHD-C12832A1Z-FSW-FBW-3V3-ND
K
A
R11
220R
CS1B
RST
A0
SCL
SDA
VDD
VSS
VOUT
C1-
C1+
C2+
C2-
V4
V3
V2
V1
V0
C8 100nF
C10 100nF
C9 100nF
C3
C4
C5
C6
C7
These are all 1uF MLCC
C1 100nF
AVDD
DVDD
NC2
NC10
RSVD
INT_N
AVSS
DVSS
IC2
MMA7660FC
VR1
ISKRA_PNZ
POT_1
VR2
ISKRA_PNZ
POT_2
R4 4k7
R5 4k7
IC1
SDA
SCL
OS
GND
VCC
A0
A1
A2
LM75BD_TSSOP8
C2 100nF
CON11
VDD
DOUT
DIN
DIO12
RESET
RSSI
DIO11
RES
DTR
GND
DIO0
DIO1
DIO2
DIO3
RTS
DIO5
RES
DIO9
CTS
DIO4
XBEE-1B2
mbed-HDK
ApplicationShield
07/04/2016 16:25
1/1

Table 8. STM32F411xC/xE pin definitions (continued)

Pin number					Pin name (function after reset)(1)	Pin type	I/O structure	Notes	Alternate functions	Additional functions
UQFN48	LQFP64	WLCSP49	LQFP100	UFBGA100L						
10	14	F6	23	L2	PA0-WKUP	I/O	TC	(5)	TIM2_CH1/TIM2_ET, TIM5_CH1, USART2_CTS, EVENTOUT	ADC1_0, WKUP1
11	15	G7	24	M2	PA1	I/O	FT	-	TIM2_CH2, TIM5_CH2, SPI4_MOSI/I2S4_SD, USART2_RTS, EVENTOUT	ADC1_1
12	16	E5	25	K3	PA2	I/O	FT	-	TIM2_CH3, TIM5_CH3, TIM9_CH1, I2S2_CKIN, USART2_TX, EVENTOUT	ADC1_2
13	17	E4	26	L3	PA3	I/O	FT	-	TIM2_CH4, TIM5_CH4, TIM9_CH2, I2S2_MCK, USART2_RX, EVENTOUT	ADC1_3
-	18	-	27	-	VSS	S	-	-	-	-
-	-	-	-	E3	BYPASS_REG	S	-	-	-	-
-	19	-	28	-	VDD	I	FT	-	EVENTOUT	-
14	20	G6	29	M3	PA4	I/O	TC	-	SPI1_NSS/I2S1_WS, SPI3_NSS/I2S3_WS, USART2_CK, EVENTOUT	ADC1_4
15	21	F5	30	K4	PA5	I/O	TC	-	TIM2_CH1/TIM2_ET, SPI1_SCK/I2S1_CK, EVENTOUT	ADC1_5
16	22	F4	31	L4	PA6	I/O	FT	-	TIM1_BKIN, TIM3_CH1, SPI1_MISO, I2S2_MCK, SDIO_CMD, EVENTOUT	ADC1_6
17	23	F3	32	M4	PA7	I/O	FT	-	TIM1_CH1N, TIM3_CH2, SPI1_MOSI/I2S1_SD, EVENTOUT	ADC1_7

Figure 39. USB OTG FS timings: definition of data signal rise and fall time

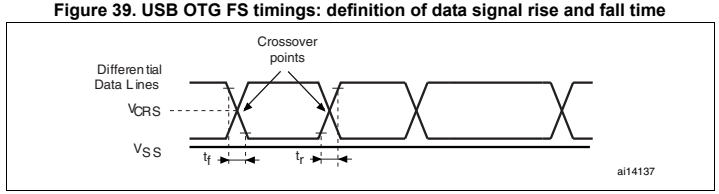


Table 64. USB OTG FS electrical characteristics(1)

Driver characteristics					
Symbol	Parameter	Conditions	Min	Max	Unit
t_r	Rise time(2)	C_L = 50 pF	4	20	ns
t_f	Fall time(2)	C_L = 50 pF	4	20	ns
t_{rfm}	Rise/ fall time matching	t_r/t_f	90	110	%
V_{CRS}	Output signal crossover voltage		1.3	2.0	V

1. Guaranteed by design, not tested in production.
2. Measured from 10% to 90% of the data signal. For more detailed informations, please refer to USB Specification - Chapter 7 (version 2.0).

6.3.20 12-bit ADC characteristics

Unless otherwise specified, the parameters given in *Table 65* are derived from tests performed under the ambient temperature, f_{PCLK2} frequency and V_{DDA} supply voltage conditions summarized in *Table 14*.

Table 65. ADC characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DDA}	Power supply	$V_{DDA} - V_{REF+} < 1.2$ V	1.7(1)	-	3.6	V
V_{REF+}	Positive reference voltage		1.7(1)	-	V_{DDA}	V
f_{ADC}	ADC clock frequency	V_{DDA} = 1.7(1) to 2.4 V	0.6	15	18	MHz
		V_{DDA} = 2.4 to 3.6 V	0.6	30	36	MHz
f_{TRIG}(2)	External trigger frequency	f_{ADC} = 30 MHz, 12-bit resolution	-	-	1764	kHz
			-	-	17	$1/f_{ADC}$
V_{AIN}	Conversion voltage range(3)		0 (V_{SSA} or V_{REF-} tied to ground)	-	V_{REF+}	V
R_{AIN}(2)	External input impedance	See *Equation 1* for details	-	-	50	kΩ
R_{ADC}(2)(4)	Sampling switch resistance		-	-	6	kΩ
C_{ADC}(2)	Internal sample and hold capacitor		-	4	7	pF

Table 65. ADC characteristics (continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{lat}(2)	Injection trigger conversion latency	f_{ADC} = 30 MHz	-	-	0.100	µs
			-	-	3(5)	$1/f_{ADC}$
t_{latr}(2)	Regular trigger conversion latency	f_{ADC} = 30 MHz	-	-	0.067	µs
			-	-	2(5)	$1/f_{ADC}$
t_S(2)	Sampling time	f_{ADC} = 30 MHz	0.100	-	16	µs
			3	-	480	$1/f_{ADC}$
t_{STAB}(2)	Power-up time		-	2	3	µs
t_{CONV}(2)	Total conversion time (including sampling time)	f_{ADC} = 30 MHz 12-bit resolution	0.50	-	16.40	µs
		f_{ADC} = 30 MHz 10-bit resolution	0.43	-	16.34	µs
		f_{ADC} = 30 MHz 8-bit resolution	0.37	-	16.27	µs
		f_{ADC} = 30 MHz 6-bit resolution	0.30	-	16.20	µs
		9 to 492 (t_S for sampling +n-bit resolution for successive approximation)				$1/f_{ADC}$
f_S(2)	Sampling rate (f_{ADC} = 30 MHz, and t_S = 3 ADC cycles)	12-bit resolution Single ADC	-	-	2	Msps
		12-bit resolution Interleave Dual ADC mode	-	-	3.75	Msps
		12-bit resolution Interleave Triple ADC mode	-	-	6	Msps
I_{VREF+}(2)	ADC V_{REF} DC current consumption in conversion mode		-	300	500	µA
I_{VDDA}(2)	ADC V_{DDA} DC current consumption in conversion mode		-	1.6	1.8	mA

1. V_{DDA} minimum value of 1.7 V is possible with the use of an external power supply supervisor (refer to *Section 3.15.2: Internal reset OFF*).
2. Guaranteed by characterization, not tested in production.
3. V_{REF+} is internally connected to V_{DDA} and V_{REF-} is internally connected to V_{SSA}.
4. R_{ADC} maximum value is given for V_{DD}=1.7 V, and minimum value for V_{DD}=3.3 V.
5. For external triggers, a delay of $1/f_{PCLK2}$ must be added to the latency specified in *Table 65*.

Equation 1: R_{AIN} max formula

$$R_{AIN} = \frac{(k-0,5)}{f_{ADC} \times C_{ADC} \times \ln(2^{N+2})} - R_{ADC}$$

The formula above (*Equation 1*) is used to determine the maximum external impedance allowed for an error below 1/4 of LSB. N = 12 (from 12-bit resolution) and k is the number of sampling periods defined in the ADC_SMPR1 register.

Table 66. ADC accuracy at f_{ADC} = 18 MHz(1)

Symbol	Parameter	Test conditions	Typ	Max(2)	Unit
ET	Total unadjusted error	f_{ADC} =18 MHz V_{DDA} = 1.7 to 3.6 V V_{REF} = 1.7 to 3.6 V $V_{DDA} - V_{REF}$ < 1.2 V	±3	±4	LSB
EO	Offset error		±2	±3	
EG	Gain error		±1	±3	
ED	Differential linearity error		±1	±2	
EL	Integral linearity error		±2	±3	

1. Better performance could be achieved in restricted V_{DD}, frequency and temperature ranges.
2. Guaranteed by characterization, not tested in production.

Table 67. ADC accuracy at f_{ADC} = 30 MHz(1)

Symbol	Parameter	Test conditions	Typ	Max(2)	Unit
ET	Total unadjusted error	f_{ADC} = 30 MHz, R_{AIN} < 10 kΩ, V_{DDA} = 2.4 to 3.6 V, V_{REF} = 1.7 to 3.6 V, $V_{DDA} - V_{REF}$ < 1.2 V	±2	±5	LSB
EO	Offset error		±1.5	±2.5	
EG	Gain error		±1.5	±4	
ED	Differential linearity error		±1	±2	
EL	Integral linearity error		±1.5	±3	

1. Better performance could be achieved in restricted V_{DD}, frequency and temperature ranges.
2. Guaranteed by characterization, not tested in production.

Table 68. ADC accuracy at f_{ADC} = 36 MHz(1)

Symbol	Parameter	Test conditions	Typ	Max(2)	Unit
ET	Total unadjusted error	f_{ADC} =36 MHz, V_{DDA} = 2.4 to 3.6 V, V_{REF} = 1.7 to 3.6 V $V_{DDA} - V_{REF}$ < 1.2 V	±4	±7	LSB
EO	Offset error		±2	±3	
EG	Gain error		±3	±6	
ED	Differential linearity error		±2	±3	
EL	Integral linearity error		±3	±6	

1. Better performance could be achieved in restricted V_{DD}, frequency and temperature ranges.
2. Guaranteed by characterization, not tested in production.

Table 69. ADC dynamic accuracy at f_{ADC} = 18 MHz - limited test conditions(1)

Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
ENOB	Effective number of bits	f_{ADC} =18 MHz $V_{DDA} = V_{REF+}$= 1.7 V Input Frequency = 20 KHz Temperature = 25 °C	10.3	10.4	-	bits
SINAD	Signal-to-noise and distortion ratio		64	64.2	-	dB
SNR	Signal-to-noise ratio		64	65	-	
THD	Total harmonic distortion		-	-72	-67	

1. Guaranteed by characterization, not tested in production.

Table 70. ADC dynamic accuracy at f_{ADC} = 36 MHz - limited test conditions(1)

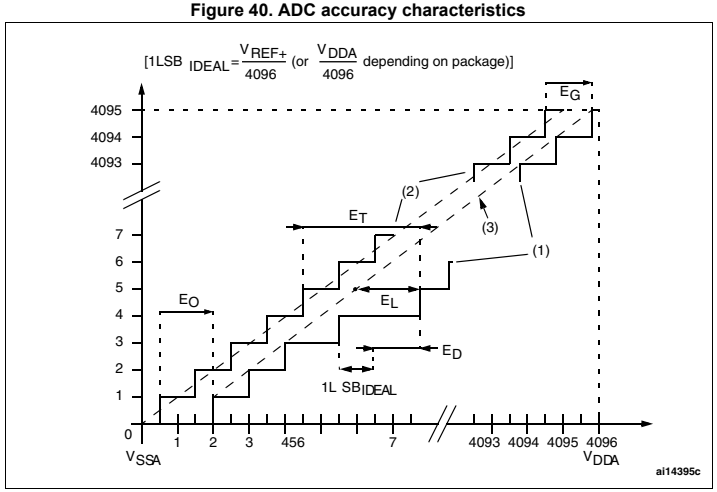
Symbol	Parameter	Test conditions	Min	Typ	Max	Unit
ENOB	Effective number of bits	f_{ADC} = 36 MHz $V_{DDA} = V_{REF+}$ = 3.3 V Input Frequency = 20 KHz Temperature = 25 °C	10.6	10.8	-	bits
SINAD	Signal-to noise and distortion ratio		66	67	-	dB
SNR	Signal-to noise ratio		64	68	-	
THD	Total harmonic distortion		-	-72	-70	

1. Guaranteed by characterization, not tested in production.

Note: ADC accuracy vs. negative injection current: injecting a negative current on any analog input pins should be avoided as this significantly reduces the accuracy of the conversion being performed on another analog input. It is recommended to add a Schottky diode (pin to ground) to analog pins which may potentially inject negative currents.

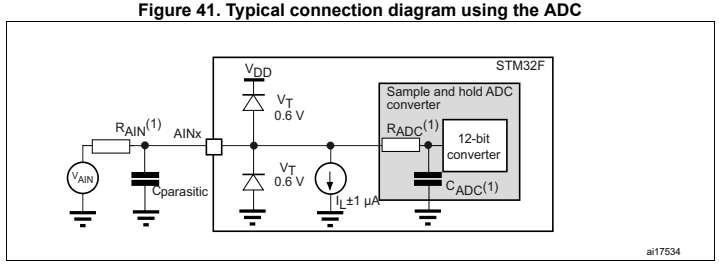
Any positive injection current within the limits specified for $I_{INJ(PIN)}$ and $\Sigma I_{INJ(PIN)}$ in *Section 6.3.16* does not affect the ADC accuracy.

Figure 40. ADC accuracy characteristics



1. See also *Table 67*.
2. Example of an actual transfer curve.
3. Ideal transfer curve.
4. End point correlation line.
5. E_T = Total Unadjusted Error: maximum deviation between the actual and the ideal transfer curves.
   EO = Offset Error: deviation between the first actual transition and the first ideal one.
   EG = Gain Error: deviation between the last ideal transition and the last actual one.
   ED = Differential Linearity Error: maximum deviation between actual steps and the ideal one.
   EL = Integral Linearity Error: maximum deviation between any actual transition and the end point correlation line.

Figure 41. Typical connection diagram using the ADC

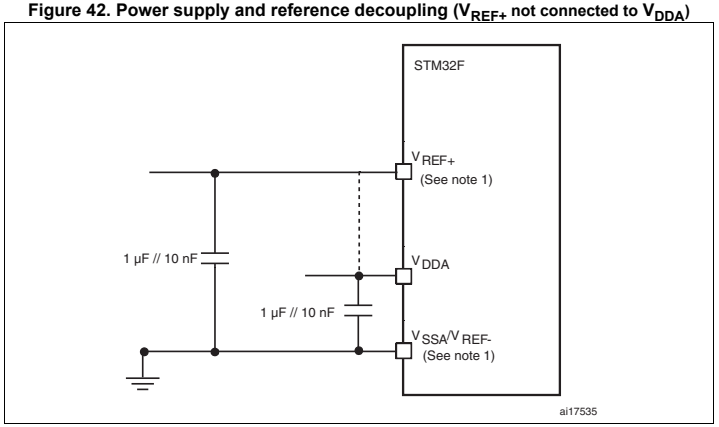


1. Refer to *Table 65* for the values of R_{AIN}, R_{ADC} and C_{ADC}.
2. $C_{parasitic}$ represents the capacitance of the PCB (dependent on soldering and PCB layout quality) plus the pad capacitance (roughly 5 pF). A high $C_{parasitic}$ value downgrades conversion accuracy. To remedy this, f_{ADC} should be reduced.

General PCB design guidelines

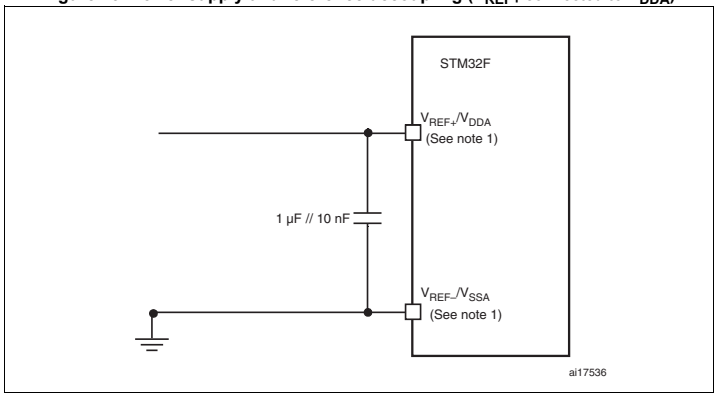
Power supply decoupling should be performed as shown in *Figure 42* or *Figure 43*, depending on whether V_{REF+} is connected to V_{DDA} or not. The 10 nF capacitors should be ceramic (good quality). They should be placed them as close as possible to the chip.

Figure 42. Power supply and reference decoupling (V_{REF+} not connected to V_{DDA})



1. V_{REF+} and V_{REF-} inputs are both available on UFBGA100. V_{REF+} is also available on LQFP100. When V_{REF+} and V_{REF-} are not available, they are internally connected to V_{DDA} and V_{SSA}.

Figure 43. Power supply and reference decoupling (V_{REF+} connected to V_{DDA})



1. V_{REF+} and V_{REF-} inputs are both available on UFBGA100. V_{REF+} is also available on LQFP100. When V_{REF+} and V_{REF-} are not available, they are internally connected to V_{DDA} and V_{SSA}.

6.3.21 Temperature sensor characteristics

Table 71. Temperature sensor characteristics

Symbol	Parameter	Min	Typ	Max	Unit
T_L(1)	V_{SENSE} linearity with temperature	-	±1	±2	°C
Avg_Slope(1)	Average slope	-	2.5	-	mV/°C
V_{25}(1)	Voltage at 25 °C	-	0.76	-	V
t_{START}(2)	Startup time	-	6	10	µs
T_{S_temp}(2)	ADC sampling time when reading the temperature (1 °C accuracy)	10	-	-	µs

1. Guaranteed by characterization, not tested in production.
2. Guaranteed by design, not tested in production.

Table 72. Temperature sensor calibration values

Symbol	Parameter	Memory address
TS_CAL1	TS ADC raw data acquired at temperature of 30 °C, V_{DDA}= 3.3 V	0x1FFF 7A2C - 0x1FFF 7A2D
TS_CAL2	TS ADC raw data acquired at temperature of 110 °C, V_{DDA}= 3.3 V	0x1FFF 7A2E - 0x1FFF 7A2F

6.3.22 V_{BAT} monitoring characteristics

Table 73. V_{BAT} monitoring characteristics

Symbol	Parameter	Min	Typ	Max	Unit
R	Resistor bridge for V_{BAT}	-	50	-	KΩ
Q	Ratio on V_{BAT} measurement	-	4	-	
Er(1)	Error on Q	–1	-	+1	%
T_{S_vbat}(2)(2)	ADC sampling time when reading the V_{BAT} 1 mV accuracy	5	-	-	µs

1. Guaranteed by design, not tested in production.
2. Shortest sampling time can be determined in the application by multiple iterations.

6.3.23 Embedded reference voltage

The parameters given in *Table 74* are derived from tests performed under ambient temperature and V_{DD} supply voltage conditions summarized in *Table 14*.

Table 74. Embedded internal reference voltage

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{REFINT}	Internal reference voltage	–40 °C < T_A < +105 °C	1.18	1.21	1.24	V
$T_{S\_vrefint}$(1)	ADC sampling time when reading the internal reference voltage	-	10	-	-	µs
V_{RERINT_s}(2)	Internal reference voltage spread over the temperature range	V_{DD} = 3V ± 10mV	-	3	5	mV

Table 74. Embedded internal reference voltage (continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T_{Coeff}(2)	Temperature coefficient	-	-	30	50	ppm/°C
t_{START}(2)	Startup time	-	-	6	10	µs

1. Shortest sampling time can be determined in the application by multiple iterations.
2. Guaranteed by design, not tested in production

Table 75. Internal reference voltage calibration values

Symbol	Parameter	Memory address
V_{REFIN_CAL}	Raw data acquired at temperature of 30 °C V_{DDA} = 3.3 V	0x1FFF 7A2A - 0x1FFF 7A2B

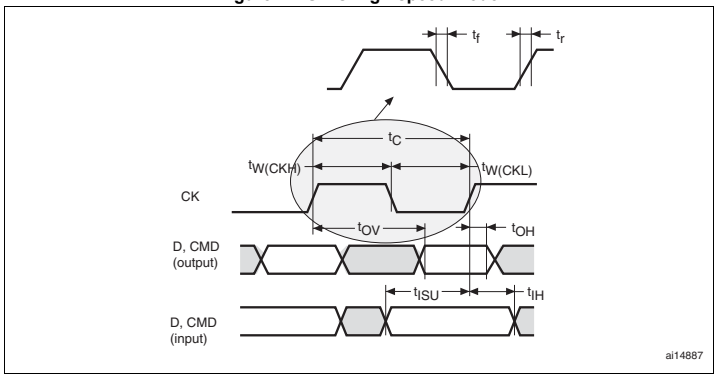
6.3.24 SD/SDIO MMC/eMMC card host interface (SDIO) characteristics

Unless otherwise specified, the parameters given in *Table 76* for the SDIO/MMC/eMMC interface are derived from tests performed under the ambient temperature, f_{PCLK2} frequency and V_{DD} supply voltage conditions summarized in *Table 14*, with the following configuration:

- Output speed is set to OSPEEDRy[1:0] = 10
- Capacitive load C = 30 pF (for eMMC C = 20 pF)
- Measurement points are done at CMOS levels: $0.5V_{DD}$

Refer to *Section 6.3.16: I/O port characteristics* for more details on the input/output characteristics.

Figure 44. SDIO high-speed mode



6.3.12 RCC APB2 peripheral clock enable register (RCC_APB2ENR)

Address offset: 0x44

Reset value: 0x0000 0000

Access: no wait state, word, half-word and byte access.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											SPI5EN	Reser-ved	TIM11 EN	TIM10 EN	TIM9 EN
											rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser-ved	SYSCFG EN	SPI4EN	SPI1 EN	SDIO EN	Reserved		ADC1 EN	Reserved			USART6 EN	USART1 EN	Reserved		TIM1 EN
	rw	rw	rw	rw			rw				rw	rw			rw

Bits 31:21 Reserved, must be kept at reset value.

Bit 20 **SPI5EN**:SPI5 clock enable
This bit is set and cleared by software
0: SPI5 clock disabled
1: SPI5 clock enabled

Bit 19 Reserved, must be kept at reset value.

Bit 18 **TIM11EN:** TIM11 clock enable
Set and cleared by software.
0: TIM11 clock disabled
1: TIM11 clock enabled

Bit 17 **TIM10EN:** TIM10 clock enable
Set and cleared by software.
0: TIM10 clock disabled
1: TIM10 clock enabled

Bit 16 **TIM9EN:** TIM9 clock enable
Set and cleared by software.
0: TIM9 clock disabled
1: TIM9 clock enabled

Bit 15 Reserved, must be kept at reset value.

Bit 14 **SYSCFGEN:** System configuration controller clock enable
Set and cleared by software.
0: System configuration controller clock disabled
1: System configuration controller clock enabled

Bit 13 **SPI4EN:** SPI4 clock enable
Set and reset by software.
0: SPI4 clock disabled
1: SPI4 clock enable

Bit 12 **SPI1EN:** SPI1 clock enable
Set and cleared by software.
0: SPI1 clock disabled
1: SPI1 clock enabled

Bit 11 **SDIOEN:** SDIO clock enable
Set and cleared by software.
0: SDIO module clock disabled
1: SDIO module clock enabled

Bit 8 **ADC1EN:** ADC1 clock enable
Set and cleared by software.
0: ADC1 clock disabled
1: ADC1 clock disabled

Bits 7:6 Reserved, must be kept at reset value.

Bit 5 **USART6EN:** USART6 clock enable
Set and cleared by software.
0: USART6 clock disabled
1: USART6 clock enabled

Bit 4 **USART1EN:** USART1 clock enable
Set and cleared by software.
0: USART1 clock disabled
1: USART1 clock enabled

Bits 3:1 Reserved, must be kept at reset value.

Bit 0 **TIM1EN:** TIM1 clock enable
Set and cleared by software.
0: TIM1 clock disabled
1: TIM1 clock enabled

11 Analog-to-digital converter (ADC)

ADC2 and ADC3 are not available in STM32F411xC/E.

11.1 ADC introduction

The 12-bit ADC is a successive approximation analog-to-digital converter. It has up to 19 multiplexed channels allowing it to measure signals from 16 external sources, two internal sources, and the V_{BAT} channel. The A/D conversion of the channels can be performed in single, continuous, scan or discontinuous mode. The result of the ADC is stored into a left- or right-aligned 16-bit data register.

The analog watchdog feature allows the application to detect if the input voltage goes beyond the user-defined, higher or lower thresholds.

11.2 ADC main features

- 12-bit, 10-bit, 8-bit or 6-bit configurable resolution
- Interrupt generation at the end of conversion, end of injected conversion, and in case of analog watchdog or overrun events
- Single and continuous conversion modes
- Scan mode for automatic conversion of channel 0 to channel ‘n’
- Data alignment with in-built data coherency
- Channel-wise programmable sampling time
- External trigger option with configurable polarity for both regular and injected conversions
- Discontinuous mode
- ADC supply requirements: 2.4 V to 3.6 V at full speed and down to 1.8 V at slower speed
- ADC input range: $V_{REF-} \leq V_{IN} \leq V_{REF+}$
- DMA request generation during regular channel conversion

Figure 31 shows the block diagram of the ADC.

Note: V_{REF-}, if available (depending on package), must be tied to V_{SSA}.

11.3 ADC functional description

Figure 31 shows a single ADC block diagram and *Table 39* gives the ADC pin description.

Figure 31. Single ADC block diagram

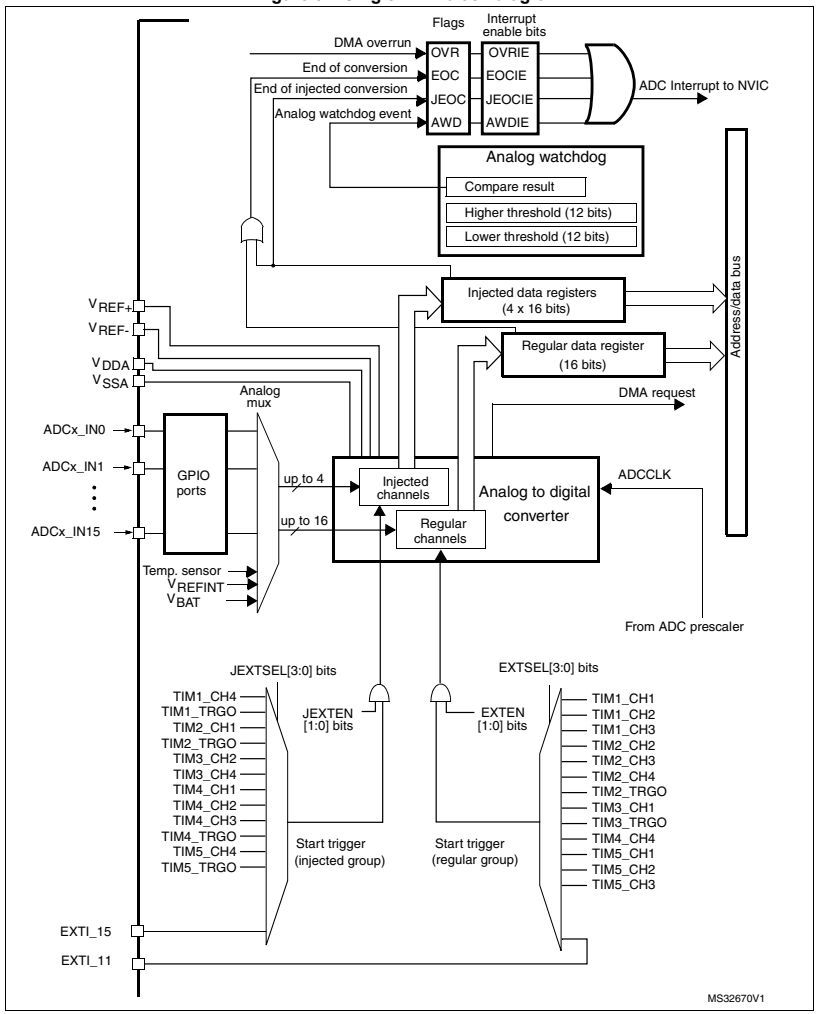


Table 39. ADC pins

Name	Signal type	Remarks
V_{REF+}	Input, analog reference positive	The higher/positive reference voltage for the ADC, 1.8 V ≤V_{REF+} ≤V_{DDA}
V_{DDA}	Input, analog supply	Analog power supply equal to V_{DD} and 2.4 V ≤V_{DDA} ≤V_{DD} (3.6 V) for full speed 1.8 V ≤V_{DDA} ≤V_{DD} (3.6 V) for reduced speed
$V_{REF–}$	Input, analog reference negative	The lower/negative reference voltage for the ADC, $V_{REF–}$ = V_{SSA}
V_{SSA}	Input, analog supply ground	Ground for analog power supply equal to V_{SS}
ADCx_IN[15:0]	Analog input signals	16 analog input channels

11.3.1 ADC on-off control

The ADC is powered on by setting the ADON bit in the ADC_CR2 register. When the ADON bit is set for the first time, it wakes up the ADC from the Power-down mode.

Conversion starts when either the SWSTART or the JSWSTART bit is set.

You can stop conversion and put the ADC in power down mode by clearing the ADON bit. In this mode the ADC consumes almost no power (only a few µA).

11.3.2 ADC clock

The ADC features two clock schemes:

- Clock for the analog circuitry: ADCCLK

  This clock is generated from the APB2 clock divided by a programmable prescaler that allows the ADC to work at f_{PCLK2}/2, /4, /6 or /8. Refer to the datasheets for the maximum value of ADCCLK.
- Clock for the digital interface (used for registers read/write access)

  This clock is equal to the APB2 clock. The digital interface clock can be enabled/disabled individually for each ADC through the RCC APB2 peripheral clock enable register (RCC_APB2ENR).

11.3.3 Channel selection

There are 16 multiplexed channels. It is possible to organize the conversions in two groups: regular and injected. A group consists of a sequence of conversions that can be done on any channel and in any order. For instance, it is possible to implement the conversion sequence in the following order: ADC_IN3, ADC_IN8, ADC_IN2, ADC_IN2, ADC_IN0, ADC_IN2, ADC_IN2, ADC_IN15.

- A **regular group** is composed of up to 16 conversions. The regular channels and their order in the conversion sequence must be selected in the ADC_SQRx registers. The total number of conversions in the regular group must be written in the L[3:0] bits in the ADC_SQR1 register.
- An **injected group** is composed of up to 4 conversions. The injected channels and their order in the conversion sequence must be selected in the ADC_JSQR register. The total number of conversions in the injected group must be written in the L[1:0] bits in the ADC_JSQR register.

If the ADC_SQRx or ADC_JSQR registers are modified during a conversion, the current conversion is reset and a new start pulse is sent to the ADC to convert the newly chosen group.

Temperature sensor, V_{REFINT} and V_{BAT} internal channels

- The temperature sensor is internally connected to ADC1_IN18 channel which is shared with VBAT. Only one conversion, temperature sensor or VBAT, must be selected at a time. When the temperature sensor and VBAT conversion are set simultaneously, only the VBAT conversion is performed.

  The internal reference voltage VREFINT is connected to ADC1_IN17.

The V_{BAT} channel (connected to channel ADC1_IN18) can also be converted as an injected or regular channel.

Note: The temperature sensor, V_{REFINT} and the V_{BAT} channel are available only on the master ADC1 peripheral.

11.3.4 Single conversion mode

In Single conversion mode the ADC does one conversion. This mode is started with the CONT bit at 0 by either:

- setting the SWSTART bit in the ADC_CR2 register (for a regular channel only)
- setting the JSWSTART bit (for an injected channel)
- external trigger (for a regular or injected channel)

Once the conversion of the selected channel is complete:

- If a regular channel was converted:
  - The converted data are stored into the 16-bit ADC_DR register
  - The EOC (end of conversion) flag is set
  - An interrupt is generated if the EOCIE bit is set
- If an injected channel was converted:
  - The converted data are stored into the 16-bit ADC_JDR1 register
  - The JEOC (end of conversion injected) flag is set
  - An interrupt is generated if the JEOCIE bit is set

Then the ADC stops.

11.3.5 Continuous conversion mode

In continuous conversion mode, the ADC starts a new conversion as soon as it finishes one. This mode is started with the CONT bit at 1 either by external trigger or by setting the SWSTRT bit in the ADC_CR2 register (for regular channels only).

After each conversion:

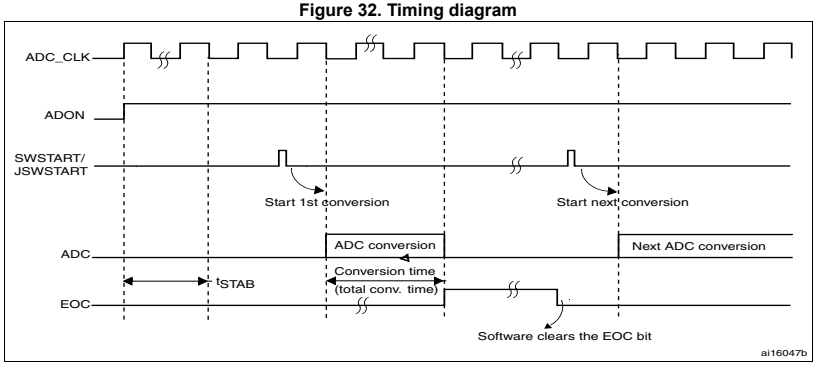
- If a regular group of channels was converted:
  - The last converted data are stored into the 16-bit ADC_DR register
  - The EOC (end of conversion) flag is set
  - An interrupt is generated if the EOCIE bit is set

Note: Injected channels cannot be converted continuously. The only exception is when an injected channel is configured to be converted automatically after regular channels in continuous mode (using JAUTO bit), refer to Auto-injection section).

11.3.6 Timing diagram

As shown in *Figure 32*, the ADC needs a stabilization time of t_{STAB} before it starts converting accurately. After the start of the ADC conversion and after 15 clock cycles, the EOC flag is set and the 16-bit ADC data register contains the result of the conversion.

Figure 32. Timing diagram



11.3.7 Analog watchdog

The AWD analog watchdog status bit is set if the analog voltage converted by the ADC is below a lower threshold or above a higher threshold. These thresholds are programmed in the 12 least significant bits of the ADC_HTR and ADC_LTR 16-bit registers. An interrupt can be enabled by using the AWDIE bit in the ADC_CR1 register.

The threshold value is independent of the alignment selected by the ALIGN bit in the ADC_CR2 register. The analog voltage is compared to the lower and higher thresholds before alignment.

Table 40 shows how the ADC_CR1 register should be configured to enable the analog watchdog on one or more channels.

Figure 33. Analog watchdog's guarded area

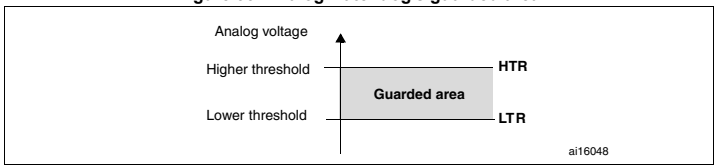


Table 40. Analog watchdog channel selection

Channels guarded by the analog watchdog	ADC_CR1 register control bits (x = don't care)		
	AWDSGL bit	AWDEN bit	JAWDEN bit
None	x	0	0
All injected channels	0	0	1
All regular channels	0	1	0
All regular and injected channels	0	1	1
Single(1) injected channel	1	0	1
Single(1) regular channel	1	1	0
Single (1) regular or injected channel	1	1	1

1. Selected by the AWDCH[4:0] bits

11.3.8 Scan mode

This mode is used to scan a group of analog channels.

The Scan mode is selected by setting the SCAN bit in the ADC_CR1 register. Once this bit has been set, the ADC scans all the channels selected in the ADC_SQRx registers (for regular channels) or in the ADC_JSQR register (for injected channels). A single conversion is performed for each channel of the group. After each end of conversion, the next channel in the group is converted automatically. If the CONT bit is set, regular channel conversion does not stop at the last selected channel in the group but continues again from the first selected channel.

If the DMA bit is set, the direct memory access (DMA) controller is used to transfer the data converted from the regular group of channels (stored in the ADC_DR register) to SRAM after each regular channel conversion.

The EOC bit is set in the ADC_SR register:

- At the end of each regular group sequence if the EOCS bit is cleared to 0
- At the end of each regular channel conversion if the EOCS bit is set to 1

The data converted from an injected channel are always stored into the ADC_JDRx registers.

11.3.9 Injected channel management

Triggered injection

To use triggered injection, the JAUTO bit must be cleared in the ADC_CR1 register.

1. Start the conversion of a group of regular channels either by external trigger or by setting the SWSTART bit in the ADC_CR2 register.
2. If an external injected trigger occurs or if the JSWSTART bit is set during the conversion of a regular group of channels, the current conversion is reset and the injected channel sequence switches to Scan-once mode.
3. Then, the regular conversion of the regular group of channels is resumed from the last interrupted regular conversion.
   If a regular event occurs during an injected conversion, the injected conversion is not

interrupted but the regular sequence is executed at the end of the injected sequence. *Figure 34* shows the corresponding timing diagram.

Note: When using triggered injection, one must ensure that the interval between trigger events is longer than the injection sequence. For instance, if the sequence length is 30 ADC clock cycles (that is two conversions with a sampling time of 3 clock periods), the minimum interval between triggers must be 31 ADC clock cycles.

Auto-injection

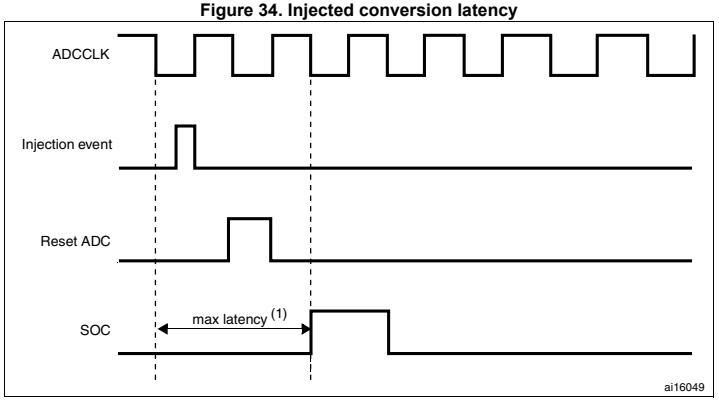
If the JAUTO bit is set, then the channels in the injected group are automatically converted after the regular group of channels. This can be used to convert a sequence of up to 20 conversions programmed in the ADC_SQRx and ADC_JSQR registers.

In this mode, external trigger on injected channels must be disabled.

If the CONT bit is also set in addition to the JAUTO bit, regular channels followed by injected channels are continuously converted.

Note: It is not possible to use both the auto-injected and discontinuous modes simultaneously.

Figure 34. Injected conversion latency



1. The maximum latency value can be found in the electrical characteristics of the STM32F411xC/E datasheets.

11.3.10 Discontinuous mode

Regular group

This mode is enabled by setting the DISCEN bit in the ADC_CR1 register. It can be used to convert a short sequence of n conversions (n ≤8) that is part of the sequence of conversions selected in the ADC_SQRx registers. The value of n is specified by writing to the DISCNUM[2:0] bits in the ADC_CR1 register.

When an external trigger occurs, it starts the next n conversions selected in the ADC_SQRx registers until all the conversions in the sequence are done. The total sequence length is defined by the L[3:0] bits in the ADC_SQR1 register.

Example:

- n = 3, channels to be converted = 0, 1, 2, 3, 6, 7, 9, 10
- 1st trigger: sequence converted 0, 1, 2. An EOC event is generated at each conversion.
- 2nd trigger: sequence converted 3, 6, 7. An EOC event is generated at each conversion
- 3rd trigger: sequence converted 9, 10.An EOC event is generated at each conversion
- 4th trigger: sequence converted 0, 1, 2. An EOC event is generated at each conversion

Note: When a regular group is converted in discontinuous mode, no rollover occurs.

When all subgroups are converted, the next trigger starts the conversion of the first subgroup. In the example above, the 4th trigger reconverts the channels 0, 1 and 2 in the 1st subgroup.

Injected group

This mode is enabled by setting the JDISCEN bit in the ADC_CR1 register. It can be used to convert the sequence selected in the ADC_JSQR register, channel by channel, after an external trigger event.

When an external trigger occurs, it starts the next channel conversions selected in the ADC_JSQR registers until all the conversions in the sequence are done. The total sequence length is defined by the JL[1:0] bits in the ADC_JSQR register.

Example:

n = 1, channels to be converted = 1, 2, 3
1st trigger: channel 1 converted
2nd trigger: channel 2 converted
3rd trigger: channel 3 converted and JEOC event generated
4th trigger: channel 1

Note: When all injected channels are converted, the next trigger starts the conversion of the first injected channel. In the example above, the 4th trigger reconverts the 1st injected channel 1.

It is not possible to use both the auto-injected and discontinuous modes simultaneously.

Discontinuous mode must not be set for regular and injected groups at the same time. Discontinuous mode must be enabled only for the conversion of one group.

11.4 Data alignment

The ALIGN bit in the ADC_CR2 register selects the alignment of the data stored after conversion. Data can be right- or left-aligned as shown in *Figure 35* and *Figure 36*.

The converted data value from the injected group of channels is decreased by the user-defined offset written in the ADC_JOFRx registers so the result can be a negative value. The SEXT bit represents the extended sign value.

For channels in a regular group, no offset is subtracted so only twelve bits are significant.

Figure 35. Right alignment of 12-bit data

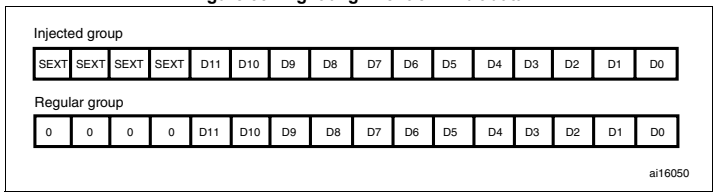
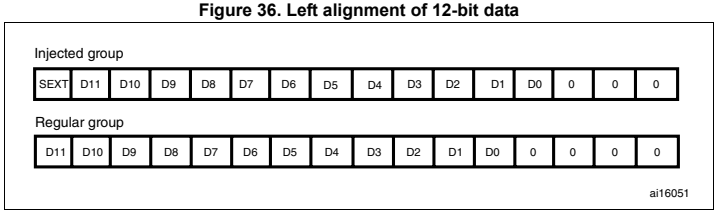
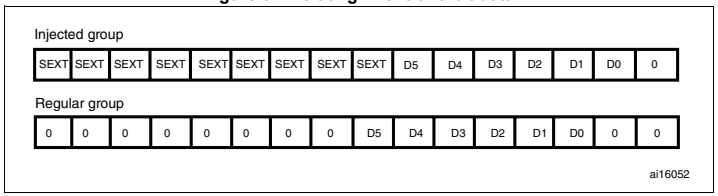


Figure 36. Left alignment of 12-bit data



Special case: when left-aligned, the data are aligned on a half-word basis except when the resolution is set to 6-bit. in that case, the data are aligned on a byte basis as shown in *Figure 37*.

Figure 37. Left alignment of 6-bit data



11.12 ADC registers

Refer to *Section 1.1: List of abbreviations for registers for registers* for a list of abbreviations used in register descriptions.

The peripheral registers must be written at word level (32 bits). Read accesses can be done by bytes (8 bits), half-words (16 bits) or words (32 bits).

11.12.1 ADC status register (ADC_SR)

Address offset: 0x00

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
Reserved										OVR	STRT	JSTRT	JEOC	EOC	AWD
										rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits 31:6 Reserved, must be kept at reset value.

Bit 5 **OVR:** Overrun
This bit is set by hardware when data are lost . It is cleared by software. Overrun detection is enabled only when DMA = 1 or EOCS = 1.
0: No overrun occurred
1: Overrun has occurred

Bit 4 **STRT:** Regular channel start flag
This bit is set by hardware when regular channel conversion starts. It is cleared by software.
0: No regular channel conversion started
1: Regular channel conversion has started

Bit 3 **JSTRT:** Injected channel start flag
This bit is set by hardware when injected group conversion starts. It is cleared by software.
0: No injected group conversion started
1: Injected group conversion has started

Bit 2 **JEOC:** Injected channel end of conversion
This bit is set by hardware at the end of the conversion of all injected channels in the group. It is cleared by software.
0: Conversion is not complete
1: Conversion complete

Bit 1 **EOC:** Regular channel end of conversion
This bit is set by hardware at the end of the conversion of a regular group of channels. It is cleared by software or by reading the ADC_DR register.
0: Conversion not complete (EOCS=0), or sequence of conversions not complete (EOCS=1)
1: Conversion complete (EOCS=0), or sequence of conversions complete (EOCS=1)

Bit 0 **AWD:** Analog watchdog flag
This bit is set by hardware when the converted voltage crosses the values programmed in the ADC_LTR and ADC_HTR registers. It is cleared by software.
0: No analog watchdog event occurred
1: Analog watchdog event occurred

11.12.2 ADC control register 1 (ADC_CR1)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					OVRIE	RES		AWDEN	JAWDEN	Reserved					
					rw	rw	rw	rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:27 Reserved, must be kept at reset value.

Bit 26 **OVRIE:** Overrun interrupt enable
This bit is set and cleared by software to enable/disable the Overrun interrupt.
0: Overrun interrupt disabled
1: Overrun interrupt enabled. An interrupt is generated when the OVR bit is set.

Bits 25:24 **RES[1:0]:** Resolution
These bits are written by software to select the resolution of the conversion.
00: 12-bit (15 ADCCLK cycles)
01: 10-bit (13 ADCCLK cycles)
10: 8-bit (11 ADCCLK cycles)
11: 6-bit (9 ADCCLK cycles)

Bit 23 **AWDEN:** Analog watchdog enable on regular channels
This bit is set and cleared by software.
0: Analog watchdog disabled on regular channels
1: Analog watchdog enabled on regular channels

Bit 22 **JAWDEN:** Analog watchdog enable on injected channels
This bit is set and cleared by software.
0: Analog watchdog disabled on injected channels
1: Analog watchdog enabled on injected channels

Bits 21:16 Reserved, must be kept at reset value.

Bits 15:13 **DISCNUM[2:0]:** Discontinuous mode channel count
These bits are written by software to define the number of regular channels to be converted in discontinuous mode, after receiving an external trigger.
000: 1 channel
001: 2 channels
...
111: 8 channels

Bit 12 **JDISCEN:** Discontinuous mode on injected channels
This bit is set and cleared by software to enable/disable discontinuous mode on the injected channels of a group.
0: Discontinuous mode on injected channels disabled
1: Discontinuous mode on injected channels enabled

Bit 11 **DISCEN:** Discontinuous mode on regular channels
This bit is set and cleared by software to enable/disable Discontinuous mode on regular channels.
0: Discontinuous mode on regular channels disabled
1: Discontinuous mode on regular channels enabled

Bit 10 **JAUTO:** Automatic injected group conversion
This bit is set and cleared by software to enable/disable automatic injected group conversion after regular group conversion.
0: Automatic injected group conversion disabled
1: Automatic injected group conversion enabled

Bit 9 **AWDSGL:** Enable the watchdog on a single channel in scan mode
This bit is set and cleared by software to enable/disable the analog watchdog on the channel identified by the AWDCH[4:0] bits.
0: Analog watchdog enabled on all channels
1: Analog watchdog enabled on a single channel

Bit 8 **SCAN:** Scan mode
This bit is set and cleared by software to enable/disable the Scan mode. In Scan mode, the inputs selected through the ADC_SQRx or ADC_JSQRx registers are converted.
0: Scan mode disabled
1: Scan mode enabled

Note: An EOC interrupt is generated if the EOCIE bit is set:
- *At the end of each regular group sequence if the EOCS bit is cleared to 0*
- *At the end of each regular channel conversion if the EOCS bit is set to 1*

Note: A JEOC interrupt is generated only on the end of conversion of the last channel if the JEOCIE bit is set.

Bit 7 **JEOCIE:** Interrupt enable for injected channels
This bit is set and cleared by software to enable/disable the end of conversion interrupt for injected channels.
0: JEOC interrupt disabled
1: JEOC interrupt enabled. An interrupt is generated when the JEOC bit is set.

Bit 6 **AWDIE:** Analog watchdog interrupt enable
This bit is set and cleared by software to enable/disable the analog watchdog interrupt.
0: Analog watchdog interrupt disabled
1: Analog watchdog interrupt enabled

Bit 5 **EOCIE:** Interrupt enable for EOC
This bit is set and cleared by software to enable/disable the end of conversion interrupt.
0: EOC interrupt disabled
1: EOC interrupt enabled. An interrupt is generated when the EOC bit is set.

Bits 4:0 **AWDCH[4:0]:** Analog watchdog channel select bits
These bits are set and cleared by software. They select the input channel to be guarded by the analog watchdog.

Note: 00000: ADC analog input Channel0
00001: ADC analog input Channel1
...
01111: ADC analog input Channel15
10000: ADC analog input Channel16
Other values reserved

11.12.3 ADC control register 2 (ADC_CR2)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
reserved	SWSTART	EXTEN		EXTSEL[3:0]				reserved	JSWSTART	JEXTEN		JEXTSEL[3:0]			
	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
reserved				ALIGN	EOCS	DDS	DMA	Reserved						CONT	ADON
				rw	rw	rw	rw							rw	rw

Bit 31 Reserved, must be kept at reset value.

Bit 30 **SWSTART:** Start conversion of regular channels

This bit is set by software to start conversion and cleared by hardware as soon as the conversion starts.
0: Reset state
1: Starts conversion of regular channels

Note: This bit can be set only when ADON = 1 otherwise no conversion is launched.

Bits 29:28 **EXTEN:** External trigger enable for regular channels

These bits are set and cleared by software to select the external trigger polarity and enable the trigger of a regular group.
00: Trigger detection disabled
01: Trigger detection on the rising edge
10: Trigger detection on the falling edge
11: Trigger detection on both the rising and falling edges

Bits 27:24 **EXTSEL[3:0]:** External event select for regular group

These bits select the external event used to trigger the start of conversion of a regular group:
0000: Timer 1 CC1 event
0001: Timer 1 CC2 event
0010: Timer 1 CC3 event
0011: Timer 2 CC2 event
0100: Timer 2 CC3 event
0101: Timer 2 CC4 event
0110: Timer 2 TRGO event
0111: Timer 3 CC1 event
1000: Timer 3 TRGO event
1001: Timer 4 CC4 event
1010: Timer 5 CC1 event
1011: Timer 5 CC2 event
1100: Timer 5 CC3 event
1101: Reserved
1110: Reserved
1111: EXTI line11

Bit 23 Reserved, must be kept at reset value.

Bit 22 **JSWSTART:** Start conversion of injected channels

This bit is set by software and cleared by hardware as soon as the conversion starts.
0: Reset state
1: Starts conversion of injected channels

Note: This bit can be set only when ADON = 1 otherwise no conversion is launched.

Bits 21:20 **JEXTEN:** External trigger enable for injected channels

These bits are set and cleared by software to select the external trigger polarity and enable the trigger of an injected group.
00: Trigger detection disabled
01: Trigger detection on the rising edge
10: Trigger detection on the falling edge
11: Trigger detection on both the rising and falling edges

Bits 19:16 **JEXTSEL[3:0]:** External event select for injected group

These bits select the external event used to trigger the start of conversion of an injected group.
0000: Timer 1 CC4 event
0001: Timer 1 TRGO event
0010: Timer 2 CC1 event
0011: Timer 2 TRGO event
0100: Timer 3 CC2 event
0101: Timer 3 CC4 event
0110: Timer 4 CC1 event
0111: Timer 4 CC2 event
1000: Timer 4 CC3 event
1001: Timer 4 TRGO event
1010: Timer 5 CC4 event
1011: Timer 5 TRGO event
1100: Reserved
1101: Reserved
1110: Reserved
1111: EXTI line15

Bits 15:12 Reserved, must be kept at reset value.

Bit 11 **ALIGN:** Data alignment

This bit is set and cleared by software. Refer to *Figure 35* and *Figure 36*.
0: Right alignment
1: Left alignment

Bit 10 **EOCS:** End of conversion selection

This bit is set and cleared by software.
0: The EOC bit is set at the end of each sequence of regular conversions. Overrun detection is enabled only if DMA=1.
1: The EOC bit is set at the end of each regular conversion. Overrun detection is enabled.

Bit 9 **DDS:** DMA disable selection (for single ADC mode)

This bit is set and cleared by software.
0: No new DMA request is issued after the last transfer (as configured in the DMA controller)
1: DMA requests are issued as long as data are converted and DMA=1

Bit 8 **DMA:** Direct memory access mode (for single ADC mode)

This bit is set and cleared by software. Refer to the DMA controller chapter for more details.
0: DMA mode disabled
1: DMA mode enabled

Bits 7:2 Reserved, must be kept at reset value.

Bit 1 **CONT:** Continuous conversion
This bit is set and cleared by software. If it is set, conversion takes place continuously until it is cleared.
0: Single conversion mode
1: Continuous conversion mode

Bit 0 **ADON:** A/D Converter ON / OFF
This bit is set and cleared by software.
Note: 0: Disable ADC conversion and go to power down mode
1: Enable ADC

11.12.4 ADC sample time register 1 (ADC_SMPR1)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
					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
SMP15_0	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31: 27 Reserved, must be kept at reset value.

Bits 26:0 **SMPx[2:0]:** Channel x sampling time selection
These bits are written by software to select the sampling time individually for each channel. During sampling cycles, the channel selection bits must remain unchanged.
Note: 000: 3 cycles
001: 15 cycles
010: 28 cycles
011: 56 cycles
100: 84 cycles
101: 112 cycles
110: 144 cycles
111: 480 cycles

11.12.5 ADC sample time register 2 (ADC_SMPR2)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		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
SMP 5_0	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30 Reserved, must be kept at reset value.

Bits 29:0 **SMPx[2:0]:** Channel x sampling time selection
These bits are written by software to select the sampling time individually for each channel. During sample cycles, the channel selection bits must remain unchanged.
Note: 000: 3 cycles
001: 15 cycles
010: 28 cycles
011: 56 cycles
100: 84 cycles
101: 112 cycles
110: 144 cycles
111: 480 cycles

11.12.6 ADC injected channel data offset register x (ADC_JOFRx) (x=1..4)

Address offset: 0x14-0x20

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
Reserved				JOFFSETx[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:12 Reserved, must be kept at reset value.

Bits 11:0 **JOFFSETx[11:0]:** Data offset for injected channel x
These bits are written by software to define the offset to be subtracted from the raw converted data when converting injected channels. The conversion result can be read from in the ADC_JDRx registers.

11.12.7 ADC watchdog higher threshold register (ADC_HTR)

Address offset: 0x24

Reset value: 0x0000 0FFF

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
Reserved				HT[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:12 Reserved, must be kept at reset value.

Bits 11:0 **HT[11:0]:** Analog watchdog higher threshold
These bits are written by software to define the higher threshold for the analog watchdog.

Note: The software can write to these registers when an ADC conversion is ongoing. The programmed value will be effective when the next conversion is complete. Writing to this register is performed with a write delay that can create uncertainty on the effective time at which the new value is programmed.

11.12.8 ADC watchdog lower threshold register (ADC_LTR)

Address offset: 0x28

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
Reserved				LT[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:12 Reserved, must be kept at reset value.

Bits 11:0 **LT[11:0]:** Analog watchdog lower threshold
These bits are written by software to define the lower threshold for the analog watchdog.

Note: The software can write to these registers when an ADC conversion is ongoing. The programmed value will be effective when the next conversion is complete. Writing to this register is performed with a write delay that can create uncertainty on the effective time at which the new value is programmed.

11.12.9 ADC regular sequence register 1 (ADC_SQR1)

Address offset: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								L[3:0]				SQ16[4:1]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16_0	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

Bits 31:24 Reserved, must be kept at reset value.

Bits 23:20 **L[3:0]:** Regular channel sequence length
These bits are written by software to define the total number of conversions in the regular channel conversion sequence.
0000: 1 conversion
0001: 2 conversions
...
1111: 16 conversions

Bits 19:15 **SQ16[4:0]:** 16th conversion in regular sequence
These bits are written by software with the channel number (0..18) assigned as the 16th in the conversion sequence.

Bits 14:10 **SQ15[4:0]:** 15th conversion in regular sequence

Bits 9:5 **SQ14[4:0]:** 14th conversion in regular sequence

Bits 4:0 **SQ13[4:0]:** 13th conversion in regular sequence

11.12.10 ADC regular sequence register 2 (ADC_SQR2)

Address offset: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			
		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
SQ10_0	SQ9[4:0]					SQ8[4:0]					SQ7[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30 Reserved, must be kept at reset value.

Bits 29:26 **SQ12[4:0]:** 12th conversion in regular sequence
These bits are written by software with the channel number (0..18) assigned as the 12th in the sequence to be converted.

Bits 24:20 **SQ11[4:0]:** 11th conversion in regular sequence

Bits 19:15 **SQ10[4:0]:** 10th conversion in regular sequence

Bits 14:10 **SQ9[4:0]:** 9th conversion in regular sequence

Bits 9:5 **SQ8[4:0]:** 8th conversion in regular sequence

Bits 4:0 **SQ7[4:0]:** 7th conversion in regular sequence

11.12.11 ADC regular sequence register 3 (ADC_SQR3)

Address offset: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			
		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
SQ4_0	SQ3[4:0]					SQ2[4:0]					SQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30 Reserved, must be kept at reset value.

Bits 29:25 **SQ6[4:0]:** 6th conversion in regular sequence
These bits are written by software with the channel number (0..18) assigned as the 6th in the sequence to be converted.

Bits 24:20 **SQ5[4:0]:** 5th conversion in regular sequence

Bits 19:15 **SQ4[4:0]:** 4th conversion in regular sequence

Bits 14:10 **SQ3[4:0]:** 3rd conversion in regular sequence

Bits 9:5 **SQ2[4:0]:** 2nd conversion in regular sequence

Bits 4:0 **SQ1[4:0]:** 1st conversion in regular sequence

11.12.12 ADC injected sequence register (ADC_JSQR)

Address offset: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										JL[1:0]		JSQ4[4:1]			
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4[0]	JSQ3[4:0]					JSQ2[4:0]					JSQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:22 Reserved, must be kept at reset value.

Bits 21:20 **JL[1:0]:** Injected sequence length

These bits are written by software to define the total number of conversions in the injected channel conversion sequence.
00: 1 conversion
01: 2 conversions
10: 3 conversions
11: 4 conversions

Bits 19:15 **JSQ4[4:0]:** 4th conversion in injected sequence (when JL[1:0]=3, see note below)

These bits are written by software with the channel number (0..18) assigned as the 4th in the sequence to be converted.

Bits 14:10 **JSQ3[4:0]:** 3rd conversion in injected sequence (when JL[1:0]=3, see note below)

Bits 9:5 **JSQ2[4:0]:** 2nd conversion in injected sequence (when JL[1:0]=3, see note below)

Bits 4:0 **JSQ1[4:0]:** 1st conversion in injected sequence (when JL[1:0]=3, see note below)

Note: When JL[1:0]=3 (4 injected conversions in the sequencer), the ADC converts the channels in the following order: JSQ1[4:0], JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0].

When JL=2 (3 injected conversions in the sequencer), the ADC converts the channels in the following order: JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0].

When JL=1 (2 injected conversions in the sequencer), the ADC converts the channels in starting from JSQ3[4:0], and then JSQ4[4:0].

When JL=0 (1 injected conversion in the sequencer), the ADC converts only JSQ4[4:0] channel.

11.12.13 ADC injected data register x (ADC_JDRx) (x= 1..4)

Address offset: 0x3C - 0x48

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
JDATA[15:0]															
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 **JDATA[15:0]:** Injected data

These bits are read-only. They contain the conversion result from injected channel x. The data are left -or right-aligned as shown in *Figure 35* and *Figure 36*.

11.12.14 ADC regular data register (ADC_DR)

Address offset: 0x4C

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
DATA[15:0]															
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 **DATA[15:0]:** Regular data

These bits are read-only. They contain the conversion result from the regular channels. The data are left- or right-aligned as shown in *Figure 35* and *Figure 36*.

11.12.15 ADC common control register (ADC_CCR)

Address offset: 0x04 (this offset address is relative to ADC1 base address + 0x300)

Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits 31:24 Reserved, must be kept at reset value.

Bit 23 **TSVREFE:** Temperature sensor and V_{REFINT} enable
This bit is set and cleared by software to enable/disable the temperature sensor and the V_{REFINT} channel.
0: Temperature sensor and V_{REFINT} channel disabled
1: Temperature sensor and V_{REFINT} channel enabled
Note: VBATE must be disabled when TSVREFE is set. If both bits are set, only the VBAT conversion is performed.

Bit 22 **VBATE:** V_{BAT} enable
This bit is set and cleared by software to enable/disable the V_{BAT} channel.
0: V_{BAT} channel disabled
1: V_{BAT} channel enabled

Bits 21:18 Reserved, must be kept at reset value.

Bits 17:16 **ADCPRE:** ADC prescaler
Set and cleared by software to select the frequency of the clock to the ADC. .
Note: 00: PCLK2 divided by 2
01: PCLK2 divided by 4
10: PCLK2 divided by 6
11: PCLK2 divided by 8

Bits 15:0 Reserved, must be kept at reset value.

11.12.16 ADC register map

The following table summarizes the ADC registers.

Table 45. ADC global register map

Offset	Register
0x000 - 0x04C	ADC1
0x050 - 0x0FC	Reserved
0x100 - 0x14C	Reserved
0x118 - 0x1FC	Reserved
0x200 - 0x24C	Reserved
0x250 - 0x2FC	Reserved
0x300 - 0x308	Common registers

Table 46. ADC register map and reset values for each ADC

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	**ADC_SR**	Reserved																										OVR	STRT	JSTRT	JEOC	EOC	AWD
	Reset value																											0	0	0	0	0	0
0x04	**ADC_CR1**	Reserved					OVRIE	RES[1:0]		AWDEN	JAWDEN	Reserved						DISC NUM [2:0]			JDISCEN	DISCEN	JAUTO	AWD SGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]					
	Reset value						0	0	0	0	0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**ADC_CR2**	Reserved	SWSTART	EXTEN[1:0]		EXTSEL [3:0]				Reserved	JSWSTART	JEXTEN[1:0]		JEXTSEL [3:0]				Reserved				ALIGN	EOCS	DDS	DMA	Reserved						CONT	ADON
	Reset value		0	0	0	0	0	0	0		0	0	0	0	0	0	0					0	0		0							0	0
0x0C	**ADC_SMPR1**	Sample time bits SMPx_x																															
	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
0x10	**ADC_SMPR2**	Sample time bits SMPx_x																															
	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	**ADC_JOFR1**	Reserved																				JOFFSET1[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x18	**ADC_JOFR2**	Reserved																				JOFFSET2[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**ADC_JOFR3**	Reserved																				JOFFSET3[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x20	**ADC_JOFR4**	Reserved																				JOFFSET4[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x24	**ADC_HTR**	Reserved																				HT[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x28	**ADC_LTR**	Reserved																				LT[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**ADC_SQR1**	Reserved								L[3:0]					Regular channel sequence SQx_x bits																		
	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
0x30	**ADC_SQR2**	Reserved	Regular channel sequence SQx_x bits																														
	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
0x34	**ADC_SQR3**	Reserved	Regular channel sequence SQx_x bits																														
	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
0x38	**ADC_JSQR**	Reserved										JL[1:0]		Injected channel sequence JSQx_x bits																			
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**ADC_JDR1**	Reserved																JDATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**ADC_JDR2**	Reserved																JDATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**ADC_JDR3**	Reserved																JDATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**ADC_JDR4**	Reserved																JDATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	**ADC_DR**	Reserved																Regular DATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 47. ADC register map and reset values (common ADC registers)

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	**ADC_CCR**	Reserved								TSVREFE	VBATE	Reserved				ADCPRE[1:0]		Reserved															
	Reset value									0	0					0	0																

Refer to *Section 2.3: Memory map* for the register boundary addresses.