



Welcome to [E-XFL.COM](#)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	ARM® Cortex®-M4
Core Size	32-Bit Single-Core
Speed	80MHz
Connectivity	CANbus, I <sup>2</sup> C, IrDA, LINbus, MMC/SD, QSPI, SAI, SPI, SWPMI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, LCD, PWM, WDT
Number of I/O	52
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	1.71V ~ 3.6V
Data Converters	A/D 16x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-UFBGA, WLCSP
Supplier Device Package	64-WLCSP (3.14x3.13)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/stmicroelectronics/stm32l433rby6tr">https://www.e-xfl.com/product-detail/stmicroelectronics/stm32l433rby6tr</a>

33.3	WWDG functional description	961
33.3.1	Enabling the watchdog	962
33.3.2	Controlling the downcounter	962
33.3.3	Advanced watchdog interrupt feature	962
33.3.4	How to program the watchdog timeout	963
33.3.5	Debug mode	964
33.4	WWDG registers	964
33.4.1	Control register (WWDG_CR)	964
33.4.2	Configuration register (WWDG_CFR)	965
33.4.3	Status register (WWDG_SR)	965
33.4.4	WWDG register map	966
<b>34</b>	<b>Real-time clock (RTC)</b>	<b>967</b>
34.1	Introduction	967
34.2	RTC main features	968
34.3	RTC functional description	969
34.3.1	RTC block diagram	969
34.3.2	GPIOs controlled by the RTC	970
34.3.3	Clock and prescalers	972
34.3.4	Real-time clock and calendar	973
34.3.5	Programmable alarms	973
34.3.6	Periodic auto-wakeup	973
34.3.7	RTC initialization and configuration	974
34.3.8	Reading the calendar	976
34.3.9	Resetting the RTC	977
34.3.10	RTC synchronization	977
34.3.11	RTC reference clock detection	978
34.3.12	RTC smooth digital calibration	979
34.3.13	Time-stamp function	981
34.3.14	Tamper detection	981
34.3.15	Calibration clock output	983
34.3.16	Alarm output	984
34.4	RTC low-power modes	984
34.5	RTC interrupts	985
34.6	RTC registers	986
34.6.1	RTC time register (RTC_TR)	986

The Flash memory is organized as follows:

- A main memory block containing up to 256 pages of 2 Kbyte. Each page is made of 8 rows of 256 bytes.
- An Information block containing:
  - System memory from which the device boots in System memory boot mode. The area is reserved for use by STMicroelectronics and contains the boot loader that is used to reprogram the Flash memory through one of the following interfaces: USART1, USART2, USART3, USB (DFU), I2C1, I2C2, I2C3, SPI1, SPI2, SPI3. It is programmed by STMicroelectronics when the device is manufactured, and protected against spurious write/erase operations. For further details, please refer to the AN2606 available from [www.st.com](http://www.st.com).
  - 1 Kbyte (128 double word) OTP (one-time programmable) bytes for user data. The OTP data cannot be erased and can be written only once. If only one bit is at 0, the entire double word cannot be written anymore, even with the value 0x0000 0000 0000 0000.
  - Option bytes for user configuration.

The memory organization is based on a main area and an information block as shown in [Table 7](#).

**Table 7. Flash module - single bank organization**

Flash area	Flash memory addresses	Size (bytes)	Name
Main memory	0x0800 0000 - 0x0800 07FF	2 K	Page 0
	0x0800 0800 - 0x0800 0FFF	2 K	Page 1
	0x0800 1000 - 0x0800 17FF	2 K	Page 2
	0x0800 1800 - 0x0800 1FFF	2 K	Page 3
	-	-	-
	0x0802 F800 - 0x0802 FFFF	2 K	Page 63 <sup>(1)</sup>
	-	-	-
	0x0803 F800 - 0x0803 FFFF	2 K	Page 127 <sup>(2)</sup>
	-	-	-
	0x087F F800 - 0x087F FFFF	2 K	Page 255 <sup>(3)</sup>
Information block	0x1FFF 0000 - 0x1FFF 6FFF	28 K	System memory
	0x1FFF 7000 - 0x1FFF 73FF	1 K	OTP area
	0x1FFF 7800 - 0x1FFF 780F	16	Option bytes

1. Main Flash memory space of 128K devices is limited to sector 63.

2. Main Flash memory space of 256K devices is limited to sector 127.

3. Main Flash memory space of 512K devices is limited to sector 255.

Bit 4 **TIM6RST**: TIM6 timer reset  
 Set and cleared by software.  
 0: No effect  
 1: Reset TIM6

Bits 3:2 Reserved, must be kept at reset value.

Bit 1 **TIM3RST**<sup>(3)</sup>: TIM3 timer reset  
 Set and cleared by software.  
 0: No effect  
 1: Reset TIM3

Bit 0 **TIM2RST**: TIM2 timer reset  
 Set and cleared by software.  
 0: No effect  
 1: Reset TIM2

1. Available on STM32L4x2xx and STM32L4x3xx devices only.
2. Not available on STM32L432xx and STM32L442xx devices.
3. Available on STM32L45xxx and STM32L46xxx devices only.
4. Available on STM32L4x3xx devices only.
5. Available on STM32L43xxx and STM32L44xxx devices only.

### 6.4.13 APB1 peripheral reset register 2 (RCC\_APB1RSTR2)

Address offset: 0x3C

Reset value: 0x00000 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2 RST	Res.	Res.	SWP MI1 RST <sup>(1)</sup>	I2C4 RST (2)	LP UART1 RST
										rw			rw	rw	rw

1. Available on STM32L43xxx and STM32L44xxx devices only.
2. Available on STM32L45xxx and STM32L46xxx devices only.

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

Bit 5 **LPTIM2RST**: Low-power timer 2 reset  
 Set and cleared by software.  
 0: No effect  
 1: Reset LPTIM2

Bits 4:3 Reserved, must be kept at reset value.

## 14 Cyclic redundancy check calculation unit (CRC)

### 14.1 Introduction

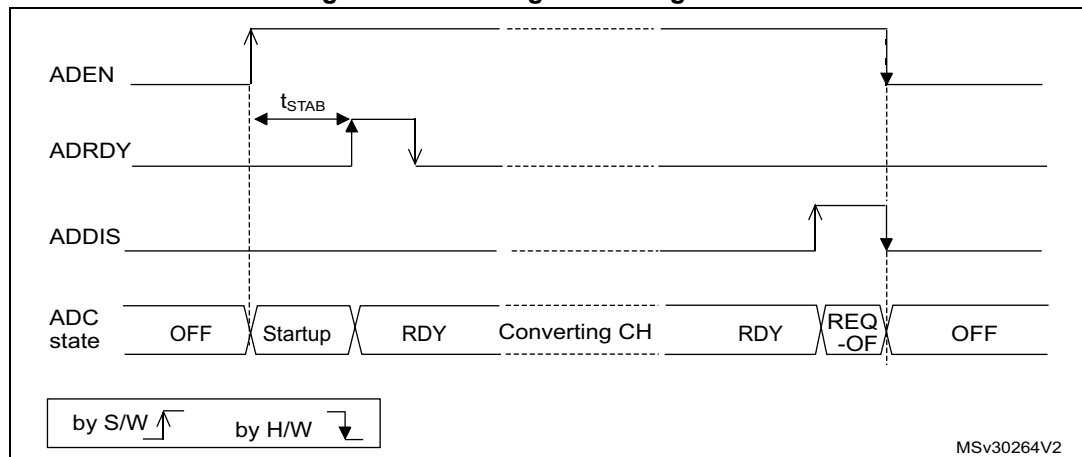
The CRC (cyclic redundancy check) calculation unit is used to get a CRC code from 8-, 16- or 32-bit data word and a generator polynomial.

Among other applications, CRC-based techniques are used to verify data transmission or storage integrity. In the scope of the functional safety standards, they offer a means of verifying the Flash memory integrity. The CRC calculation unit helps compute a signature of the software during runtime, to be compared with a reference signature generated at link time and stored at a given memory location.

### 14.2 CRC main features

- Uses CRC-32 (Ethernet) polynomial: 0x4C11DB7  
$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$
- Alternatively, uses fully programmable polynomial with programmable size (7, 8, 16, 32 bits)
- Handles 8-, 16-, 32-bit data size
- Programmable CRC initial value
- Single input/output 32-bit data register
- Input buffer to avoid bus stall during calculation
- CRC computation done in 4 AHB clock cycles (HCLK) for the 32-bit data size
- General-purpose 8-bit register (can be used for temporary storage)
- Reversibility option on I/O data

Figure 46. Enabling / Disabling the ADC



#### 16.4.10 Constraints when writing the ADC control bits

The software is allowed to write the RCC control bits to configure and enable the ADC clock (refer to RCC Section), the control bits DIFSEL in the ADC\_DIFSEL register and the control bits ADCAL and ADEN in the ADC\_CR register, only if the ADC is disabled (ADEN must be equal to 0).

The software is then allowed to write the control bits ADSTART, JADSTART and ADDIS of the ADC\_CR register only if the ADC is enabled and there is no pending request to disable the ADC (ADEN must be equal to 1 and ADDIS to 0).

For all the other control bits of the ADC\_CFGR, ADC\_SMPRx, ADC\_TRx, ADC\_SQRx, ADC\_JDRy, ADC\_OFRy, ADC\_OFCHR and ADC\_IER registers:

- For control bits related to configuration of regular conversions, the software is allowed to write them only if the ADC is enabled (ADEN=1) and if there is no regular conversion ongoing (ADSTART must be equal to 0).
- For control bits related to configuration of injected conversions, the software is allowed to write them only if the ADC is enabled (ADEN=1) and if there is no injected conversion ongoing (JADSTART must be equal to 0).

The software is allowed to write the control bits ADSTP or JADSTP of the ADC\_CR register only if the ADC is enabled and eventually converting and if there is no pending request to disable the ADC (ADSTART or JADSTART must be equal to 1 and ADDIS to 0).

The software can write the register ADC\_JSQR at any time, when the ADC is enabled (ADEN=1).

**Note:** *There is no hardware protection to prevent these forbidden write accesses and ADC behavior may become in an unknown state. To recover from this situation, the ADC must be disabled (clear ADEN=0 as well as all the bits of ADC\_CR register).*

## 16.4.26 Data management

### Data register, data alignment and offset (ADC\_DR, OFFSETy, OFFSETy\_CH, ALIGN)

#### Data and alignment

At the end of each regular conversion channel (when EOC event occurs), the result of the converted data is stored into the ADC\_DR data register which is 16 bits wide.

At the end of each injected conversion channel (when JEOP event occurs), the result of the converted data is stored into the corresponding ADC\_JDRy data register which is 16 bits wide.

The ALIGN bit in the ADC\_CFGR register selects the alignment of the data stored after conversion. Data can be right- or left-aligned as shown in [Figure 67](#), [Figure 68](#), [Figure 69](#) and [Figure 70](#).

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 69](#) and [Figure 70](#).

**Note:** *Left-alignment is not supported in oversampling mode. When ROVSE and/or JOVSE bit is set, the ALIGN bit value is ignored and the ADC only provides right-aligned data.*

#### Offset

An offset y (y=1,2,3,4) can be applied to a channel by setting the bit OFFSETy\_EN=1 into ADC\_OFRy register. The channel to which the offset will be applied is programmed into the bits OFFSETy\_CH[4:0] of ADC\_OFRy register. In this case, the converted value is decreased by the user-defined offset written in the bits OFFSETy[11:0]. The result may be a negative value so the read data is signed and the SEXT bit represents the extended sign value.

**Note:** *Offset correction is not supported in oversampling mode. When ROVSE and/or JOVSE bit is set, the value of the OFFSETy\_EN bit in ADC\_OFRy register is ignored (considered as reset).*

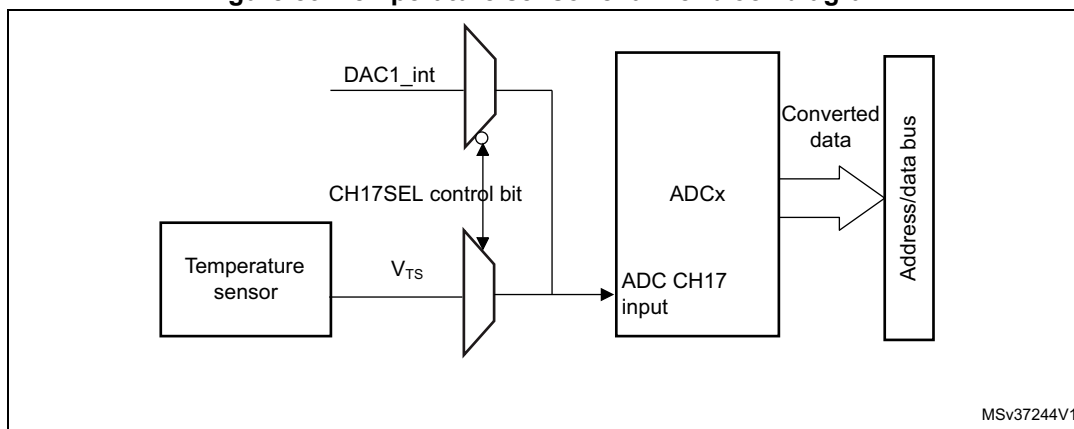
[Table 63](#) describes how the comparison is performed for all the possible resolutions for analog watchdog 1.

**Table 61. Offset computation versus data resolution**

Resolution (bits RES[1:0])	Subtraction between raw converted data and offset:		Result	Comments
	Raw converted Data, left aligned	Offset		
00: 12-bit	DATA[11:0]	OFFSET[11:0]	signed 12-bit data	-
01: 10-bit	DATA[11:2],00	OFFSET[11:0]	signed 10-bit data	The user must configure OFFSET[1:0] to "00"

Figure 89 shows the block diagram of the temperature sensor. The DAC1\_int signal and the temperature sensor share the same input.

**Figure 89. Temperature sensor channel block diagram**



1. The CH17SEL bit must be set to enable the conversion of the temperature sensor voltage  $V_{TS}$

### Reading the temperature

To use the sensor:

1. Select the ADC1\_IN17 input channels.
2. Program with the appropriate sampling time (refer to electrical characteristics section of the device datasheet).
3. Set the CH17SEL bit in the ADC\_CCR register to wake up the temperature sensor from power-down mode.
4. Start the ADC conversion.
5. Read the resulting  $V_{TS}$  data in the ADC data register.
6. Calculate the actual temperature using the following formula:

$$\text{Temperature (in } ^\circ\text{C)} = \frac{110\text{ }^\circ\text{C} - 30\text{ }^\circ\text{C}}{\text{TS\_CAL2} - \text{TS\_CAL1}} \times (\text{TS\_DATA} - \text{TS\_CAL1}) + 30\text{ }^\circ\text{C}$$

Where:

- TS\_CAL2 is the temperature sensor calibration value acquired at 110°C
  - TS\_CAL1 is the temperature sensor calibration value acquired at 30°C
  - TS\_DATA is the actual temperature sensor output value converted by ADC
- Refer to the device datasheet for more information about TS\_CAL1 and TS\_CAL2 calibration points.

**Note:** The sensor has a startup time after waking from power-down mode before it can output  $V_{TS}$  at the correct level. The ADC also has a startup time after power-on, so to minimize the delay, the ADEN and CH17SEL bits should be set at the same time.

#### 16.4.32 $V_{BAT}$ supply monitoring

The CH18SEL bit in the ADC\_CCR register is used to switch to the battery voltage. As the  $V_{BAT}$  voltage could be higher than  $V_{DDA}$ , to ensure the correct operation of the ADC, the  $V_{BAT}$  pin is internally connected to a bridge divider by 3. This bridge is automatically enabled when CH18\_SEL is set, to connect  $V_{BAT}/3$  to the ADC1\_IN18 input channels. As a



When a DAC channel1 trigger arrives, the DAC channel1 triangle counter, with the same triangle amplitude, is added to the DHR1 register and the sum is transferred into DAC\_DOR1 (three APB1 clock cycles later). The DAC channel1 triangle counter is then updated.

When a DAC channel2 trigger arrives, the DAC channel2 triangle counter, with the same triangle amplitude, is added to the DHR2 register and the sum is transferred into DAC\_DOR2 (three APB1 clock cycles later). The DAC channel2 triangle counter is then updated.

### Independent trigger with different triangle generation

To configure the DAC in this conversion mode, the following sequence is required:

- Set the two DAC channel trigger enable bits TEN1 and TEN2
- Configure different trigger sources by setting different values in the TSEL1[2:0] and TSEL2[2:0] bits
- Configure the two DAC channel WAVEx[1:0] bits as “1x” and set different maximum amplitude values in the MAMP1[3:0] and MAMP2[3:0] bits
- Load the dual DAC channel data into the desired DHR register (DAC\_DHR12RD, DAC\_DHR12LD or DAC\_DHR8RD)

When a DAC channel1 trigger arrives, the DAC channel1 triangle counter, with a triangle amplitude configured by MAMP1[3:0], is added to the DHR1 register and the sum is transferred into DAC\_DOR1 (three APB1 clock cycles later). The DAC channel1 triangle counter is then updated.

When a DAC channel2 trigger arrives, the DAC channel2 triangle counter, with a triangle amplitude configured by MAMP2[3:0], is added to the DHR2 register and the sum is transferred into DAC\_DOR2 (three APB1 clock cycles later). The DAC channel2 triangle counter is then updated.

### Simultaneous software start

To configure the DAC in this conversion mode, the following sequence is required:

- Load the dual DAC channel data to the desired DHR register (DAC\_DHR12RD, DAC\_DHR12LD or DAC\_DHR8RD)

In this configuration, one APB1 clock cycle later, the DHR1 and DHR2 registers are transferred into DAC\_DOR1 and DAC\_DOR2, respectively.

### Simultaneous trigger without wave generation

To configure the DAC in this conversion mode, the following sequence is required:

- Set the two DAC channel trigger enable bits TEN1 and TEN2
- Configure the same trigger source for both DAC channels by setting the same value in the TSEL1[2:0] and TSEL2[2:0] bits
- Load the dual DAC channel data to the desired DHR register (DAC\_DHR12RD, DAC\_DHR12LD or DAC\_DHR8RD)

When a trigger arrives, the DHR1 and DHR2 registers are transferred into DAC\_DOR1 and DAC\_DOR2, respectively (after three APB1 clock cycles).

Bits 13:12 **DATMPX[1:0]**: Input data multiplexer for channel y

0: Data to channel y are taken from external serial inputs as 1-bit values. DFSDM\_CHyDATINR register is write protected.

1: Reserved

2: Data to channel y are taken from internal DFSDM\_CHyDATINR register by direct CPU/DMA write. There can be written one or two 16-bit data samples according DATPACK[1:0] bit field setting.

3: Reserved

This value can be modified only when CHEN=0 (in DFSDM\_CHyCFGR1 register).

Bits 11:9 Reserved, must be kept at reset value.

Bit 8 **CHINSEL**: Channel inputs selection

0: Channel inputs are taken from pins of the same channel y.

1: Channel inputs are taken from pins of the following channel (channel (y+1) modulo 8).

This value can be modified only when CHEN=0 (in DFSDM\_CHyCFGR1 register).

Bit 7 **CHEN**: Channel y enable

0: Channel y disabled

1: Channel y enabled

If channel y is enabled, then serial data receiving is started according to the given channel setting.

Bit 6 **CKABEN**: Clock absence detector enable on channel y

0: Clock absence detector disabled on channel y

1: Clock absence detector enabled on channel y

Bit 5 **SCDEN**: Short-circuit detector enable on channel y

0: Input channel y will not be guarded by the short-circuit detector

1: Input channel y will be continuously guarded by the short-circuit detector

Bit 4 Reserved, must be kept at reset value.

Bits 3:2 **SPICKSEL[1:0]**: SPI clock select for channel y

0: clock coming from external CKINy input - sampling point according SITP[1:0]

1: clock coming from internal CKOUT output - sampling point according SITP[1:0]

2: clock coming from internal CKOUT - sampling point on each second CKOUT falling edge.

For connection to external  $\Sigma\Delta$  modulator which divides its clock input (from CKOUT) by 2 to generate its output serial communication clock (and this output clock change is active on each clock input rising edge).

3: clock coming from internal CKOUT output - sampling point on each second CKOUT rising edge.

For connection to external  $\Sigma\Delta$  modulator which divides its clock input (from CKOUT) by 2 to generate its output serial communication clock (and this output clock change is active on each clock input falling edge).

This value can be modified only when CHEN=0 (in DFSDM\_CHyCFGR1 register).

Bits 1:0 **SITP[1:0]**: Serial interface type for channel y

00: SPI with rising edge to strobe data

01: SPI with falling edge to strobe data

10: Manchester coded input on DATINy pin: rising edge = logic 0, falling edge = logic 1

11: Manchester coded input on DATINy pin: rising edge = logic 1, falling edge = logic 0

This value can only be modified when CHEN=0 (in DFSDM\_CHyCFGR1 register).

Bit 2 Reserved, must be kept at reset value.

Bit 1 **JSWSTART**: Start a conversion of the injected group of channels

0: Writing '0' has no effect.

1: Writing '1' makes a request to convert the channels in the injected conversion group, causing JCIP to become '1' at the same time. If JCIP=1 already, then writing to JSWSTART has no effect. Writing '1' has no effect if JSYNC=1.

This bit is always read as '0'.

Bit 0 **DFEN**: DFSDM\_FLTx enable

0: DFSDM\_FLTx is disabled. All conversions of given DFSDM\_FLTx are stopped immediately and all DFSDM\_FLTx functions are stopped.

1: DFSDM\_FLTx is enabled. If DFSDM\_FLTx is enabled, then DFSDM\_FLTx starts operating according to its setting.

Data which are cleared by setting DFEN=0:

–register DFSDM\_FLTxISR is set to the reset state

–register DFSDM\_FLTxAWSR is set to the reset state

## 21.8.2 DFSDM control register 2 (DFSDM\_FLTxCR2)

Address offset:  $0x104 + 0x80 * x$ ,  $x = 0 \dots 1$

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	EXCH[3:0]				Res.	CKAB IE	SCDIE	AWDIE	ROVR IE	JOVR IE	REOC IE	JEOC IE
				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

Bits 31:20 Reserved, must be kept at reset value.

Bits 19:16 **AWDCH[3:0]**: Analog watchdog channel selection

These bits select the input channel to be guarded continuously by the analog watchdog

AWDCH[y] = 0: Analog watchdog is disabled on channel y

AWDCH[y] = 1: Analog watchdog is enabled on channel y

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

Bits 11:8 **EXCH[3:0]**: Extremes detector channel selection

These bits select the input channels to be taken by the Extremes detector

EXCH[y] = 0: Extremes detector does not accept data from channel y

EXCH[y] = 1: Extremes detector accepts data from channel y

Bit 7 Reserved, must be kept at reset value.

Bit 6 **CKABIE**: Clock absence interrupt enable

0: Detection of channel input clock absence interrupt is disabled

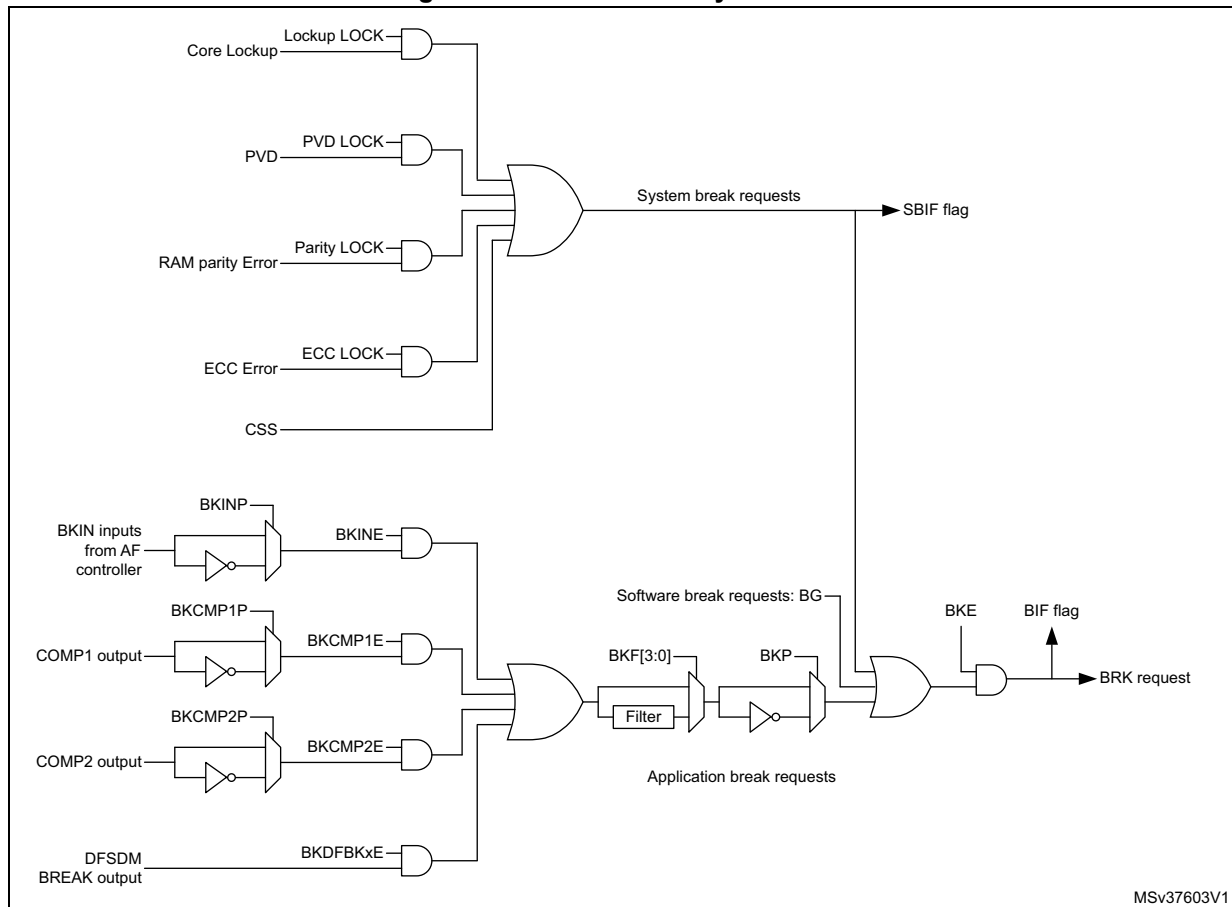
1: Detection of channel input clock absence interrupt is enabled

Please see the explanation of CKABF[3:0] in DFSDM\_FLTxISR.

*Note: CKABIE is present only in DFSDM\_FLT0CR2 register (filter x=0)*

All sources are ORed before entering the timer BRK inputs, as per [Figure 287](#) below.

**Figure 287. Break circuitry overview**



**Note:** An asynchronous (clockless) operation is only guaranteed when the programmable filter is disabled. If it is enabled, a fail safe clock mode (for example by using the internal PLL and/or the CSS) must be used to guarantee that break events are handled.

When a break occurs (selected level on the break input):

- The MOE bit is cleared asynchronously, putting the outputs in inactive state, idle state or even releasing the control to the AFIO controller (selected by the OSS1 bit). This feature functions even if the MCU oscillator is off.
- Each output channel is driven with the level programmed in the OISx bit in the TIMx\_CR2 register as soon as MOE=0. If OSS1=0, the timer releases the output control (taken over by the AFIO controller) else the enable output remains high.
- When complementary outputs are used:
  - The outputs are first put in reset state inactive state (depending on the polarity). This is done asynchronously so that it works even if no clock is provided to the timer.
  - If the timer clock is still present, then the dead-time generator is reactivated in order to drive the outputs with the level programmed in the OISx and OISxN bits after a dead-time. Even in this case, OCx and OCxN cannot be driven to their

Bit 10 **CC2DE**: Capture/Compare 2 DMA request enable

0: CC2 DMA request disabled  
1: CC2 DMA request enabled

Bit 9 **CC1DE**: Capture/Compare 1 DMA request enable

0: CC1 DMA request disabled  
1: CC1 DMA request enabled

Bit 8 **UDE**: Update DMA request enable

0: Update DMA request disabled  
1: Update DMA request enabled

Bit 7 **BIE**: Break interrupt enable

0: Break interrupt disabled  
1: Break interrupt enabled

Bit 6 **TIE**: Trigger interrupt enable

0: Trigger interrupt disabled  
1: Trigger interrupt enabled

Bit 5 **COMIE**: COM interrupt enable

0: COM interrupt disabled  
1: COM interrupt enabled

Bits 4:3 Reserved, must be kept at reset value.

Bit 2 **CC2IE**: Capture/Compare 2 interrupt enable

0: CC2 interrupt disabled  
1: CC2 interrupt enabled

Bit 1 **CC1IE**: Capture/Compare 1 interrupt enable

0: CC1 interrupt disabled  
1: CC1 interrupt enabled

Bit 0 **UIE**: Update interrupt enable

0: Update interrupt disabled  
1: Update interrupt enabled

### 28.5.5 TIM15 status register (TIM15\_SR)

Address offset: 0x10

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0		rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0

Bits 15:11 Reserved, must be kept at reset value.

Bit 10 **CC2OF**: Capture/Compare 2 overcapture flag

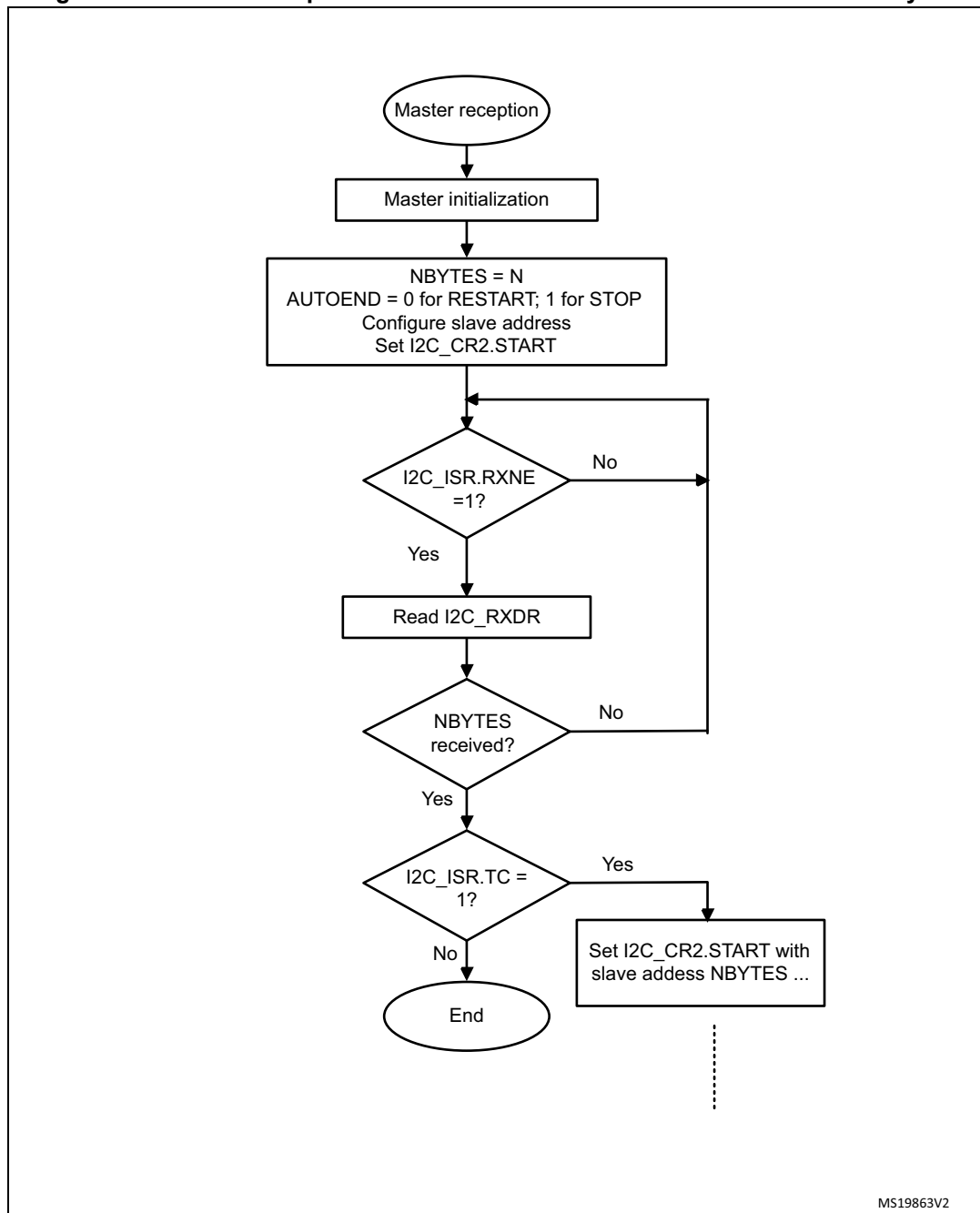
Refer to CC1OF description

Bit 9 **CC1OF**: Capture/Compare 1 overcapture flag

This flag is set by hardware only when the corresponding channel is configured in input capture mode. It is cleared by software by writing it to '0'.

0: No overcapture has been detected

1: The counter value has been captured in TIMx\_CCR1 register while CC1IF flag was already set

Figure 336. Transfer sequence flowchart for I2C master receiver for  $N \leq 255$  bytes

Bit 2 **RXIE**: RX Interrupt enable

0: Receive (RXNE) interrupt disabled

1: Receive (RXNE) interrupt enabled

Bit 1 **TXIE**: TX Interrupt enable

0: Transmit (TXIS) interrupt disabled

1: Transmit (TXIS) interrupt enabled

Bit 0 **PE**: Peripheral enable

0: Peripheral disable

1: Peripheral enable

*Note: When PE=0, the I2C SCL and SDA lines are released. Internal state machines and status bits are put back to their reset value. When cleared, PE must be kept low for at least 3 APB clock cycles.*

### 35.7.2 Control register 2 (I2C\_CR2)

Address offset: 0x04

Reset value: 0x0000 0000

Access: No wait states, except if a write access occurs while a write access to this register is ongoing. In this case, wait states are inserted in the second write access until the previous one is completed. The latency of the second write access can be up to  $2 \times \text{PCLK1} + 6 \times \text{I2CCLK}$ .

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTO END	RE LOAD	NBYTES[7:0]							
					rs	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD 10R	ADD10	RD_ WRN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw									

Bits 31:9 Reserved, must be kept at reset value.

Bits 8:0 **TDR[8:0]**: Transmit data value

Contains the data character to be transmitted.

The TDR register provides the parallel interface between the internal bus and the output shift register (see [Figure 347](#)).

When transmitting with the parity enabled (PCE bit set to 1 in the LPUART\_CR1 register), the value written in the MSB (bit 7 or bit 8 depending on the data length) has no effect because it is replaced by the parity.

*Note: This register must be written only when TXE=1.*



## 37.7.10 LPUART register map

The table below gives the [LPUART](#) register map and reset values.

Table 179. LPUART 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	LPUART_CR1	Res.	Res.	Res.	M1	Res.	Res.	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	DEDT4	DEDT3	DEDT2	DEDT1	DEDT0	Res.	CMIE	MME	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE				
	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	LPUART_CR2	ADD[7:4]				ADD[3:0]				Res.	Res.	Res.	Res.	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	Res.	STOP[1:0]				Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.			
	Reset value	0	0	0	0	0	0	0	0					0	0	0	0	0	0		0	0									0						
0x08	LPUART_CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCESM	WUFIE	WUS[1:0]		Res.	Res.	Res.	Res.	DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE				
	Reset value									0	0	0	0					0	0	0	0		0	0	0	0	0			0			0				
0x0C	LPUART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[19:0]																								
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x10-0x14	Reserved																																				
0x18	LPUART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				
	Reset value																													0	0	0					
0x1C	LPUART_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	Res.	Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE			
	Reset value										0	0	0	0	0	0	0						0	0		1	1	0	0	0	0	0	0				
0x20	LPUART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ORECF	NCF	FE CF	PE CF					
	Reset value												0			0									0			0		0	0	0	0				
0x24	LPUART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]												
	Reset value																								X	X	X	X	X	X	X	X	X				
0x28	LPUART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]											
	Reset value																								X	X	X	X	X	X	X	X	X				

Refer to [Section 2.2 on page 62](#) for the register boundary addresses.

Bits 31:2 Reserved, must be kept at reset value

Bit 1 **SWP\_CLASS**: SWP class selection

This bit is used to select the SWP class (refer to [Section 40.3.2: SWP initialization and activation](#)).

0: Class C: SWPMI\_IO uses directly VDD voltage to operate in class C.

This configuration must be selected when VDD is in the range [1.62 V to 1.98 V]

1: Class B: SWPMI\_IO uses an internal voltage regulator to operate in class B.

This configuration must be selected when VDD is in the range [2.70 V to 3.30 V]

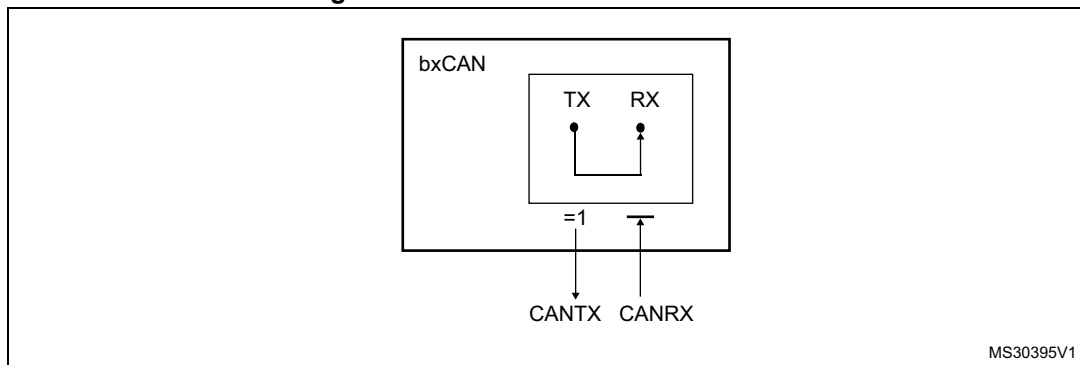
Bit 0 **SWP\_TBYP**: SWP transceiver bypass

This bit is used to bypass the internal transceiver (SWPMI\_IO), and connect an external transceiver.

0: Internal transceiver is enabled. The external interface for SWPMI is SWPMI\_IO (SWPMI\_RX, SWPMI\_TX and SWPMI\_SUSPEND signals are not available on GPIOs)

1: Internal transceiver is disabled. SWPMI\_RX, SWPMI\_TX and SWPMI\_SUSPEND signals are available as alternate function on GPIOs. This configuration is selected to connect an external transceiver

Figure 445. bxCAN in combined mode



MS30395V1

## 42.6 Behavior in debug mode

When the microcontroller enters the debug mode (Cortex®-M4 core halted), the bxCAN continues to work normally or stops, depending on:

- the DBF bit in CAN\_MCR. For more details, refer to [Section 42.9.2: CAN control and status registers](#).

## 42.7 bxCAN functional description

### 42.7.1 Transmission handling

In order to transmit a message, the application must select one **empty** transmit mailbox, set up the identifier, the data length code (DLC) and the data before requesting the transmission by setting the corresponding TXRQ bit in the CAN\_TlRxR register. Once the mailbox has left **empty** state, the software no longer has write access to the mailbox registers. Immediately after the TXRQ bit has been set, the mailbox enters **pending** state and waits to become the highest priority mailbox, see *Transmit Priority*. As soon as the mailbox has the highest priority it will be **scheduled** for transmission. The transmission of the message of the scheduled mailbox will start (enter **transmit** state) when the CAN bus becomes idle. Once the mailbox has been successfully transmitted, it will become **empty** again. The hardware indicates a successful transmission by setting the RQCP and TXOK bits in the CAN\_TSR register.

If the transmission fails, the cause is indicated by the ALST bit in the CAN\_TSR register in case of an Arbitration Lost, and/or the TERR bit, in case of transmission error detection.

#### Transmit priority

By identifier

When more than one transmit mailbox is pending, the transmission order is given by the identifier of the message stored in the mailbox. The message with the lowest identifier value has the highest priority according to the arbitration of the CAN protocol. If the identifier values are equal, the lower mailbox number will be scheduled first.

By transmit request order

The transmit mailboxes can be configured as a transmit FIFO by setting the TXFP bit in the CAN\_MCR register. In this mode the priority order is given by the transmit request order.

Table 246. JTAG debug port data registers (continued)

IR(3:0)	Data register	Details
1011	APACC [35 bits]	<p>Access port access register</p> <p>Initiates an access port and allows access to an access port register.</p> <ul style="list-style-type: none"> <li>When transferring data IN: <ul style="list-style-type: none"> <li>Bits 34:3 = DATA[31:0] = 32-bit data to shift in for a write request</li> <li>Bits 2:1 = A[3:2] = 2-bit address (sub-address AP registers).</li> <li>Bit 0 = RnW= Read request (1) or write request (0).</li> </ul> </li> <li>When transferring data OUT: <ul style="list-style-type: none"> <li>Bits 34:3 = DATA[31:0] = 32-bit data which is read following a read request</li> <li>Bits 2:0 = ACK[2:0] = 3-bit Acknowledge: <ul style="list-style-type: none"> <li>010 = OK/FAULT</li> <li>001 = WAIT</li> <li>OTHER = reserved</li> </ul> </li> </ul> </li> </ul> <p>There are many AP Registers (see AHB-AP) addressed as the combination of:</p> <ul style="list-style-type: none"> <li>The shifted value A[3:2]</li> <li>The current value of the DP SELECT register</li> </ul>
1000	ABORT [35 bits]	<p>Abort register</p> <ul style="list-style-type: none"> <li>Bits 31:1 = Reserved</li> <li>Bit 0 = DAPABORT: write 1 to generate a DAP abort.</li> </ul>

Table 247. 32-bit debug port registers addressed through the shifted value A[3:2]

Address	A(3:2) value	Description
0x0	00	Reserved, must be kept at reset value.
0x4	01	<p>DP CTRL/STAT register. Used to:</p> <ul style="list-style-type: none"> <li>Request a system or debug power-up</li> <li>Configure the transfer operation for AP accesses</li> <li>Control the pushed compare and pushed verify operations</li> <li>Read some status flags (overrun, power-up acknowledges)</li> </ul>
0x8	10	<p>DP SELECT register. Used to select the current access port and the active 4-words register window.</p> <ul style="list-style-type: none"> <li>Bits 31:24: APSEL: select the current AP</li> <li>Bits 23:8: reserved</li> <li>Bits 7:4: APBANKSEL: select the active 4-words register window on the current AP</li> <li>Bits 3:0: reserved</li> </ul>
0xC	11	<p>DP RDBUFF register: Used to allow the debugger to get the final result after a sequence of operations (without requesting new JTAG-DP operation)</p>

IDCODE read or CTRL/STAT read or ABORT write which are accepted even if the write buffer is full.

- Because of the asynchronous clock domains SWCLK and HCLK, two extra SWCLK cycles are needed after a write transaction (after the parity bit) to make the write effective internally. These cycles should be applied while driving the line low (IDLE state)  
This is particularly important when writing the CTRL/STAT for a power-up request. If the next transaction (requiring a power-up) occurs immediately, it will fail.

#### 44.8.5 SW-DP registers

Access to these registers are initiated when APnDP=0

**Table 251. SW-DP registers**

A(3:2)	R/W	CTRLSEL bit of SELECT register	Register	Notes
00	Read	-	IDCODE	The manufacturer code is not set to ST code <b>0x2BA01477</b> (identifies the SW-DP)
00	Write	-	ABORT	-
01	Read/Write	0	DP-CTRL/STAT	Purpose is to: <ul style="list-style-type: none"> <li>– request a system or debug power-up</li> <li>– configure the transfer operation for AP accesses</li> <li>– control the pushed compare and pushed verify operations.</li> <li>– read some status flags (overrun, power-up acknowledges)</li> </ul>
01	Read/Write	1	WIRE CONTROL	Purpose is to configure the physical serial port protocol (like the duration of the turnaround time)
10	Read	-	READ RESEND	Enables recovery of the read data from a corrupted debugger transfer, without repeating the original AP transfer.
10	Write	-	SELECT	The purpose is to select the current access port and the active 4-words register window
11	Read/Write	-	READ BUFFER	This read buffer is useful because AP accesses are posted (the result of a read AP request is available on the next AP transaction). This read buffer captures data from the AP, presented as the result of a previous read, without initiating a new transaction