AN4195
Application note

STM32F30x ADC modes and application

Introduction

This application note presents an overview of the ADC peripherals in the STM32F30x microcontroller families, and explains how to use the various modes and specific features of the ADC. Its purpose is to help ADC users to understand the advanced modes offered in STM32 microcontrollers, and to quick start development.

This document is divided into three sections:

- *Section 1: Comparison between ADC F1 family and F3 family* describes a brief comparison between the two ADCs of STM32 F1 and F3 family.
- *Section 2: Independent modes* describes modes used with a single ADC.
- *Section 3: Dual modes* describes modes that should be used with two ADCs (ADC1 and ADC2, or ADC3 and ADC4 working jointly).

Each mode described in these sections is provided with a typical configuration procedure to help understand how to use it.

Note: This application note does not describe the dual combined modes.

Table 1. Applicable products

Type	Part numbers
Microcontrollers	STM32F30x

Contents

1 Comparison between ADC F1 family and F3 family

1.1 STM32 ADC F3 family main features

The ADC F3 family presents several new features as:

- The queue of context for the injected conversion to allow fast context switching.
- Three Analog watchdogs per ADC allow very precise monitoring of the converted voltage of one, some or all selected channels.
- A programmable Single-ended or differential inputs.
- The delayed-conversion mode is used to ensure that no ADC data loss will occur whatever are the application requirements.
- The self calibration for single-ended and differential input.

1.1.1 Using the ADC in regular conversion

Calibration phase

Calibration is preliminary to all ADC operations. It removes the offset error, which may vary from one chip to another.

1. Ensure that ADVREGEN[1:0]=01, and that the ADC voltage regulator startup time has elapsed.
2. Ensure that ADEN=0.
3. Select the input mode for this calibration by setting ADCALDIF=0 (single-ended input) or ADCALDIF=1 (differential input).
4. Set ADCAL=1.
5. Wait until ADCAL=0.
6. Read the calibration factor from the ADC_CALFACT register.

If the ADC is supposed to convert both single-ended and differential inputs, two calibrations must be performed, one with ADCALDIF=0 and one with ADCALDIF=1.

1.1.2 ADC channel input mode

ADC channel inputs can be configured in single-ended or differential mode.

Single-ended mode

To configure the ADC channels to single-ended mode, first disable the ADC (ADCAL=0, JADSTART=0, JADSTP=0, ADSTART=0, ADSTP=0, ADDIS=0 and ADEN=0). Select single-ended input mode by programming DIFSEL[15:1] in the ADC_DIFSEL register.

Note: If ADC analog input Channel-i is configured in single-ended, the DIFSEL[i] should be kept cleared.

The analog voltage to be converted for channel “i” is the difference between the external voltage ADC_INi (positive input) and VREF- (negative input).

Differential mode

To configure the ADC channels to differential mode, first disable the ADC (ADCAL=0, JADSTART=0, JADSTP=0, ADSTART=0, ADSTP=0, ADDIS=0 and ADEN=0). Select differential input mode by programming DIFSEL[15:1] in the ADC_DIFSEL register.

Note: If ADC analog input Channel-i is configured in differential, DIFSEL[i] should be set to '1'.

The analog voltage to be converted for channel "i" is the difference between the external voltage ADC_INi (positive input) and ADC_INi+1 (negative input).

1.1.3 Auto-Delay controller

The Auto delayed-conversions are useful to simplify the software as well as to optimize performance of applications clocked at low frequency where there would be a risk to encounter an ADC overrun. When the autodelay feature is enabled (AUTDLY=1), a new conversion can start only if all the previous data of the same group has been treated:

- For the regular conversion group, once the ADC_DR register is read or the EOC flag is cleared.
- For the injected conversion group, once the JEOCS is cleared.

1.2 Difference between STM32ADC F1 and F3 family

The STM32F1 family and STM32F3 family embed the same type of the ADC, the successive approximation register (SAR) ADC.

But there are some difference in term of features, the following table present a comparison between the two ADCs.

Table 2. ADC F1 and ADC F3 families comparison

Feature	ADC F1 family	ADC F3 family
Max sampling rate	1 Msps	5 Msps
Resolution	12 bits	6/8/10/12 bits
Input range	[V_{REF-}, V_{REF+}]	[V_{REF-}, V_{REF+}]
ADC Clock	AHB clock	AHB clock / PLL clock
Input mode	Single-ended	Single-ended and differential
Number of channels	16 single-ended inputs	15 single ended/ differential inputs + 3 internal single-ended channels
Analog watchdog	1 AWDG	3 AWGD
Trigger sources for regular conversion	– Software – Embedded timers – External events	– Software – Embedded timers – External events
Trigger sources for injected conversion	– Software – Embedded timers – External events	– Software – Embedded timers – External events
Number of instances	1 or 2	4
Dual mode	Yes	Yes

Table 2. ADC F1 and ADC F3 families comparison (continued)

Feature	ADC F1 family	ADC F3 family
Queue of context	No	Yes
Autodelay controller	No	Yes

2 Independent modes

The STM32F30x ADC is a 12-bit ADC successive approximation analog-to-digital converter.

This ADC has up to 19 multiplexed channels, allowing measurement of up to 16 external sources and up to 3 internal sources. The various channels can be converted in single, multichannel, continuous or discontinuous mode. The result of conversion is stored in a left- or right-aligned 16-bit data register.

2.1 Regular conversion

Regular conversion can be performed for single-ended or differential inputs, in single or continuous mode, for single channel or multichannel with programmable sampling time for each channel.

The following sections present typical configuration procedures for each feature or mode.

2.1.1 Single or multi-channel configuration

In independent mode, the ADC channels can be used in a single or multi-channel manner. With the ADC sequencer, you can configure any sequence of up to 16 channels successively, with different sampling times and in different orders.

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.

2.1.2 Single or continuous conversion mode

In single conversion mode, the ADC performs all channel conversions once. You can start this mode with the CONT bit at 0 either by setting the ADSTART bit in the ADC_CR register (for a regular channel), or through an external hardware trigger event.

In continuous conversion mode, when a software or hardware regular trigger event occurs, the ADC performs all the regular conversions of the channels once, and then automatically re-starts and continuously converts each conversion in the sequence. You can start this mode with the CONT bit at 1 either through an external trigger, or by setting the ADSTART bit in the ADC_CR register.

2.1.3 How to use the ADC in regular conversion

ADC mode and feature configuration

1. Select the conversion mode (single or continuous) by setting or resetting the CONT bit in the ADC_CFGR register.
2. Configure the sequence length and the order of the channels to be converted, by configuring the ADC_SQRx registers.
3. Configure the sampling time for each channel to be configured, by writing to the ADC_SMPRX registers.
4. If the ADC uses the external trigger to start the conversion, configure the trigger by programming the EXTEN and EXTSEL bits in the ADC_CFGR register.
5. If needed, configure the autodelay or the overrun mode features, or both, by setting the AUTDLY and/or the OVRMOD in the ADC_CFGR register.
6. Configure the ADC resolution through the RES bit in the ADC_CFGR register.

ADC start conversion

- Software-based regular ADC conversion starts immediately after the ADSTART bit is set.
- Hardware-based regular conversion starts after the ADSTART bit is set, and at the next active edge of the selected regular hardware trigger.

ADC read conversion data

There are two possible configurations for reading converted data:

- using EOC/EOS flags and interrupts
- using DMA

For the first configuration, the software must use the EOC flag and its associated interrupt to handle each data result. Each time a conversion is complete, EOC is set and the ADC_DR register can be read. OVRMOD should be configured to 0 to manage overrun events as an error.

In using DMA, when the DMA mode is enabled (DMAEN bit set to 1 in the ADC_CFGR register in single ADC mode), a DMA request is generated after each channel conversion. This allows converted data to be transferred from the ADC_DR register to the destination selected by the software.

Figure 1 presents the flowchart of the multichannel or single channel in continuous regular conversion mode.

Figure 1. Multichannel/Single, continuous mode

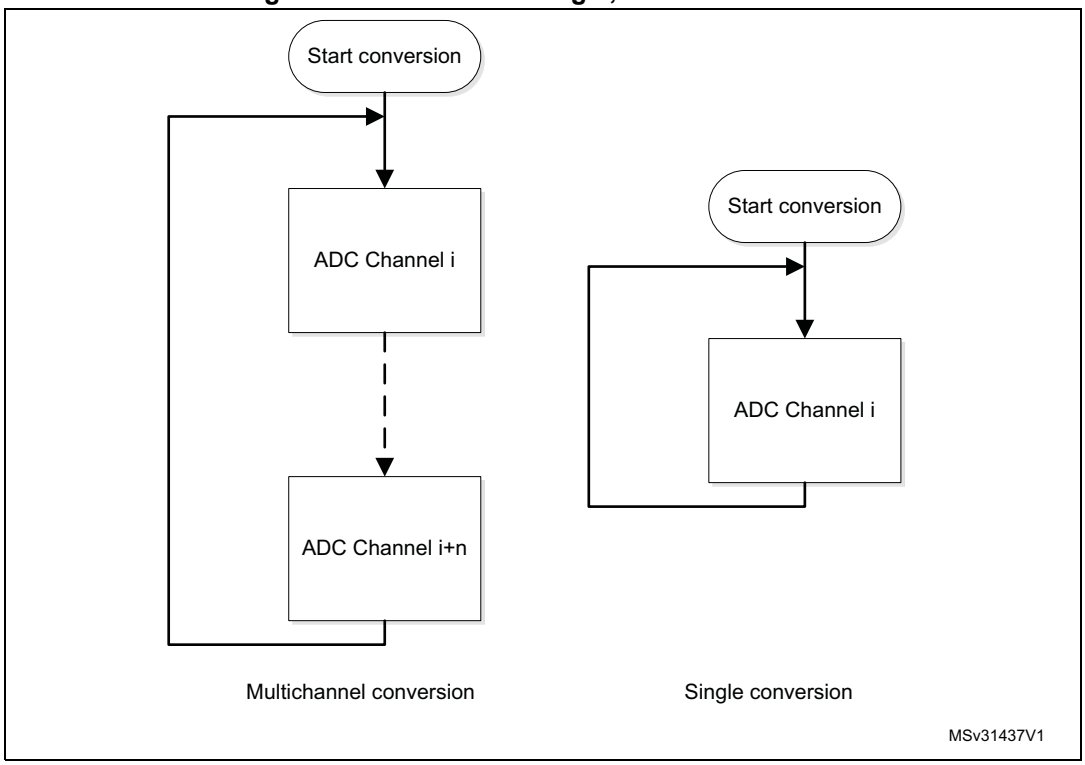
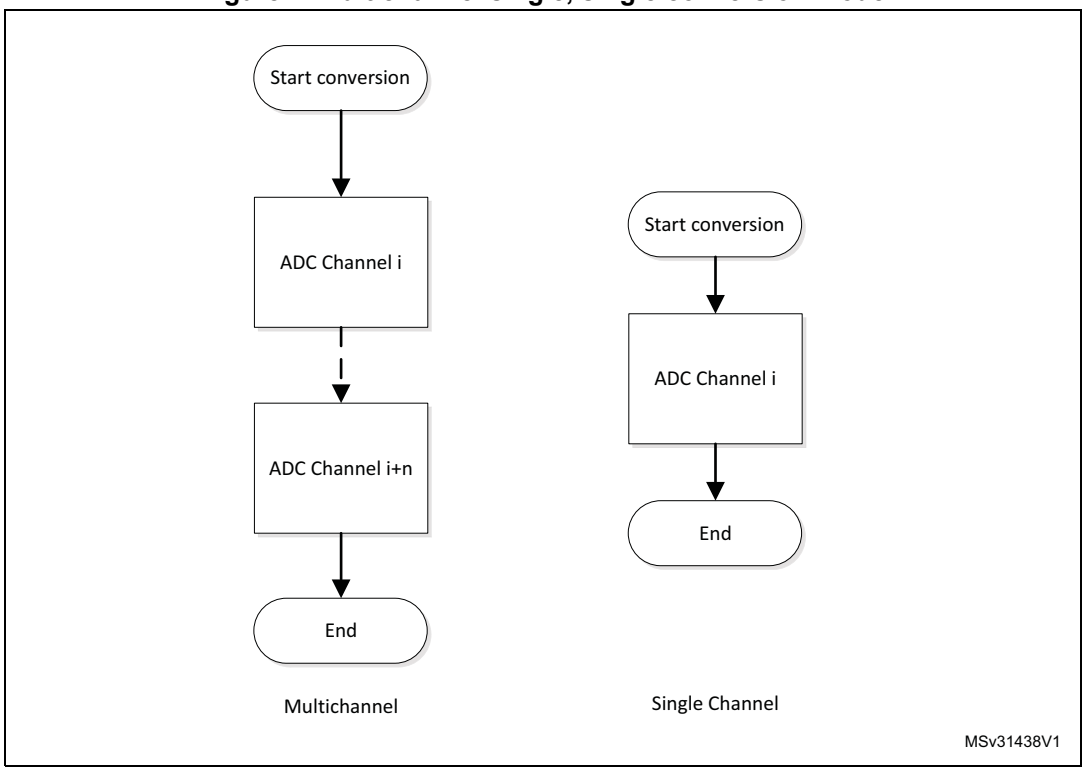


Figure 2 presents the flowchart of the multichannel or single channel in single regular conversion mode.

Figure 2. Multichannel/single, single conversion mode



2.1.4 Application examples

The single or multichannel in single conversion mode can be used when starting a system depending on some parameters like knowing the coordinates of the arm's tip in a manipulator arm system. In this case, you have to read the position of each articulation in the manipulator arm system at power-on to determine the coordinates of the arm's tip.

This mode can also be used to make single measurements of multiple signal levels (voltage, pressure, temperature, etc.) to decide if the system can be started or not in order to protect the people and equipment.

It can likewise be used to convert signals coming from strain gauges to determine the directions and values of the different strains and deformations of an object.

The single or multichannel in continuous conversion mode can be used to monitor multiple voltages and temperatures in a multiple battery charger. The voltage and temperature of each battery are read during the charging process.

When the voltage or the temperature reaches the maximum level, the corresponding battery should be disconnected from the charger.

2.2 Injected conversion mode

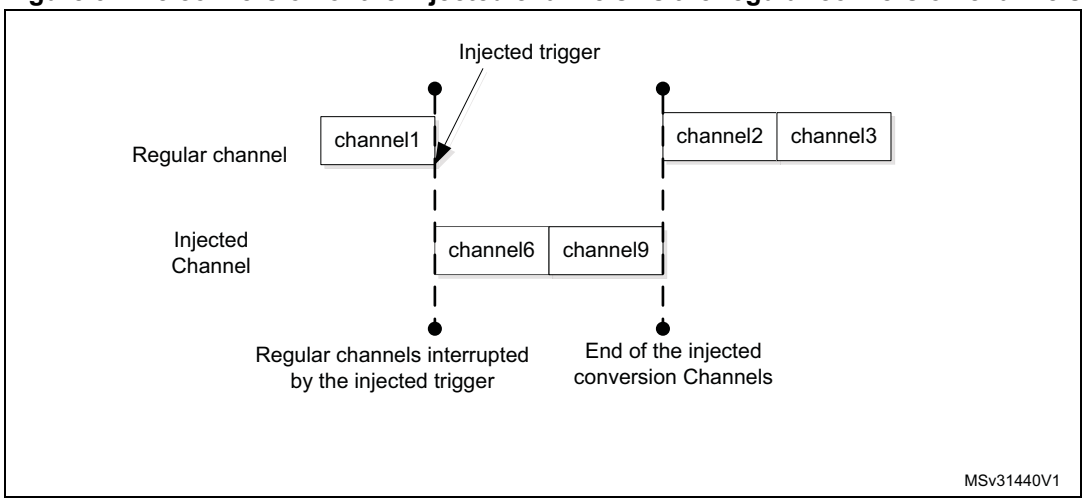
This mode is intended for use when conversion is triggered either by an external event or by software. The injected group has priority over the regular channel group. It interrupts conversion of the current channel in the regular channel group.

Figure 3 presents the behavior of the injected conversion vs regular sequence.

The user can choose between two modes of injected conversion:

- triggered injection mode
- auto-injection mode

Figure 3. The conversion of the Injected channels vs the regular conversion channels



2.2.1 Triggered injection mode

This mode is selected by resetting the JAUTO bit in the ADC_CFGR register.

If an external injected trigger occurs, or if the JADSTART bit in the ADC_CR register is set during the conversion of a regular group of channels, the current conversion is reset, and the injected channel sequence switches are launched. This means that all injected channels are converted once.

2.2.2 Auto-injection mode

This mode is selected by setting the JAUTO bit to ‘1’ in the ADC_CFGR register.

If this 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, as programmed in the ADC_SQRx and ADC_JSQR registers.

2.2.3 Queue of context for injected conversion

The STM32F30x ADC presents a new feature versus STM32F1 which is the queue of context for injected conversion.

A queue of context is used to anticipate up to two contexts for the next injected sequence of conversions.

The context consists of:

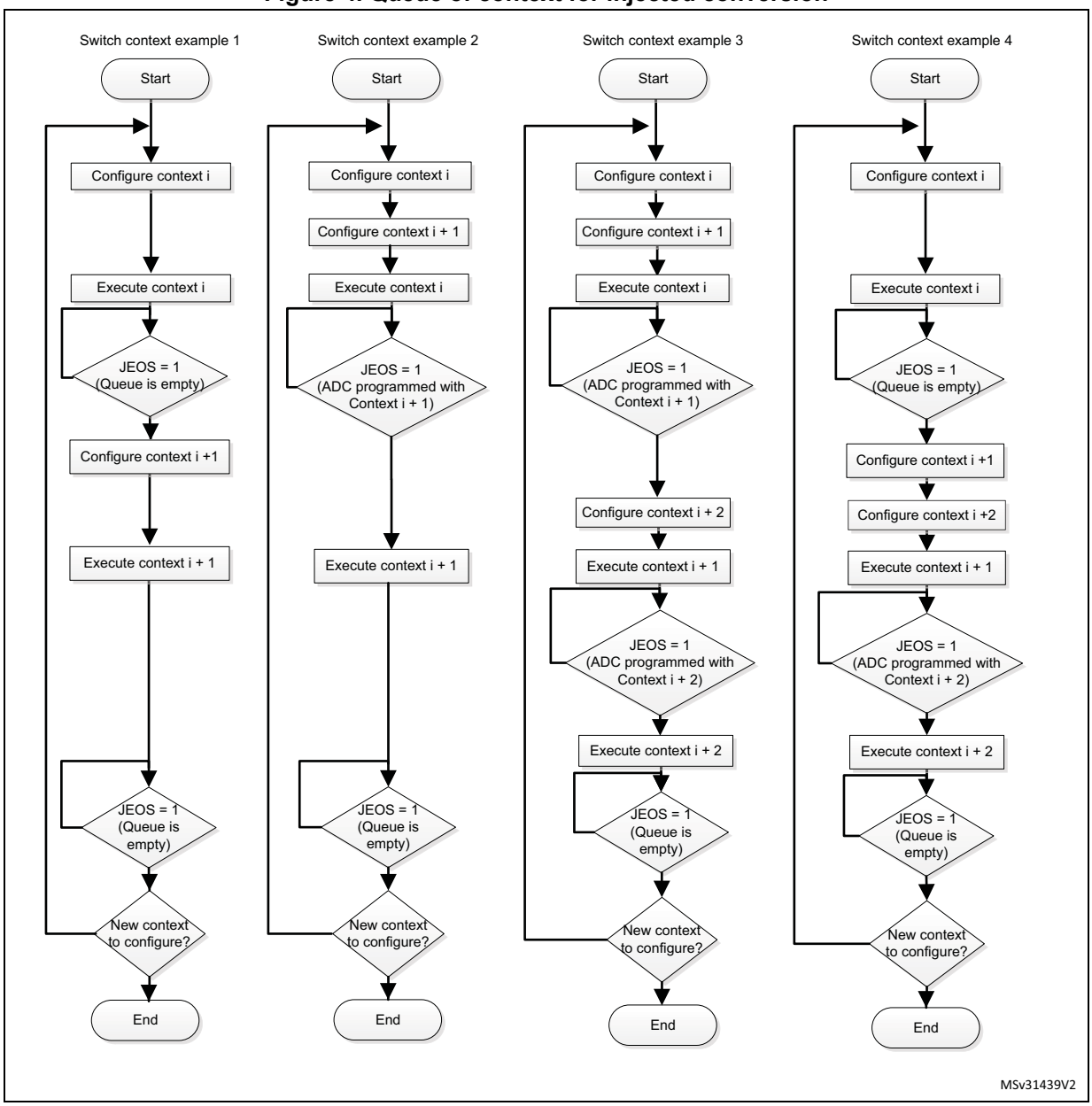
- Configuration of the injected triggers (bits JEXTEN[1:0] and JEXTSEL[3:0] in the ADC_JSQR register)
- Definition of the injected sequence (bits JSQx[4:0] and JL[1:0] in the ADC_JSQR register)

One of the two queue of context modes is selected by writing the JQM bit in the ADC_CFGR register:

- JSQR Mode 0: The queue is never empty, and maintains the last written configuration in JSQR.
- JSQR Mode 1: The queue can be empty. When this occurs, the software and hardware triggers of the injected sequence are both disabled internally just after completion of the last valid injected sequence.

The following figure presents the flowchart of examples of switch context sequence:

Figure 4. Queue of context for injected conversion



2.2.4 Using the ADC in injected conversion

Injected channel configuration

1. Configure the sequence length, the order of channels to be converted, the external hardware trigger and the edge of this trigger, by configuring the ADC_JSQR registers.
2. Configure the next context (if the queue of context will be used). Note that each write access to the ADC_JSQR register is considered as a new context.
3. Configure the sampling time for each channel to be configured, by writing to the ADC_SMPRX registers.
4. Select the JSQR mode by writing to the JQM bit in the ADC_CFGR register.
5. Set the JAUTO bit if the injected sequence should be started after the regular channels.
6. Configure the ADC resolution by writing to the RES bit in the ADC_CFGR register.

ADC start conversion

- Software-injected ADC conversion is started immediately after the JADSTART bit is set.
- Hardware-injected conversion is started after the JADSTART bit is set, and at the next active edge of the selected injected hardware trigger.
- If JAUTO=1, both the regular and injected conversion channels use the ADSTART bit to start the regular conversions, followed by the auto-injected conversions. This means that JADSTART must be kept cleared.

ADC read conversion data

At the end of each injected conversion channel (when the JEOC event occurs), the converted data result is stored in the corresponding 16-bit wide ADC_JDRx data register.

To read the converted data, both the JEOC flag/interrupt and the JEOS flag/interrupt can be used to access the corresponding register.

2.2.5 Application examples

The injected conversion mode can be used to synchronize the conversion of channels to an event. It is interesting in motor control applications where transistor switching generates noise that impacts ADC measurements and results in wrong conversions. Using a timer, the injected conversion mode can thus be implemented to delay the ADC measurements until after the transistor switching.
The queue of context is used to synchronize the conversion of channels to an event, each context uses a specific external trigger to convert a specific sequence of injected channels. A typical example of application of the Queue of context is the dual motor control.

3 Dual modes

In dual mode, two ADCs are used: an ADC master and an ADC slave. The beginning of conversion is triggered by the ADC master to the ADC slave. The converted data of both master and slave can be read in parallel by reading the multi-mode data register ADC_CDR. The status bits can be also read in parallel by reading the multi-mode status register ADC_CSR.

The STM32F30x ADC family presents several dual modes, each of which is explained independently in this document:

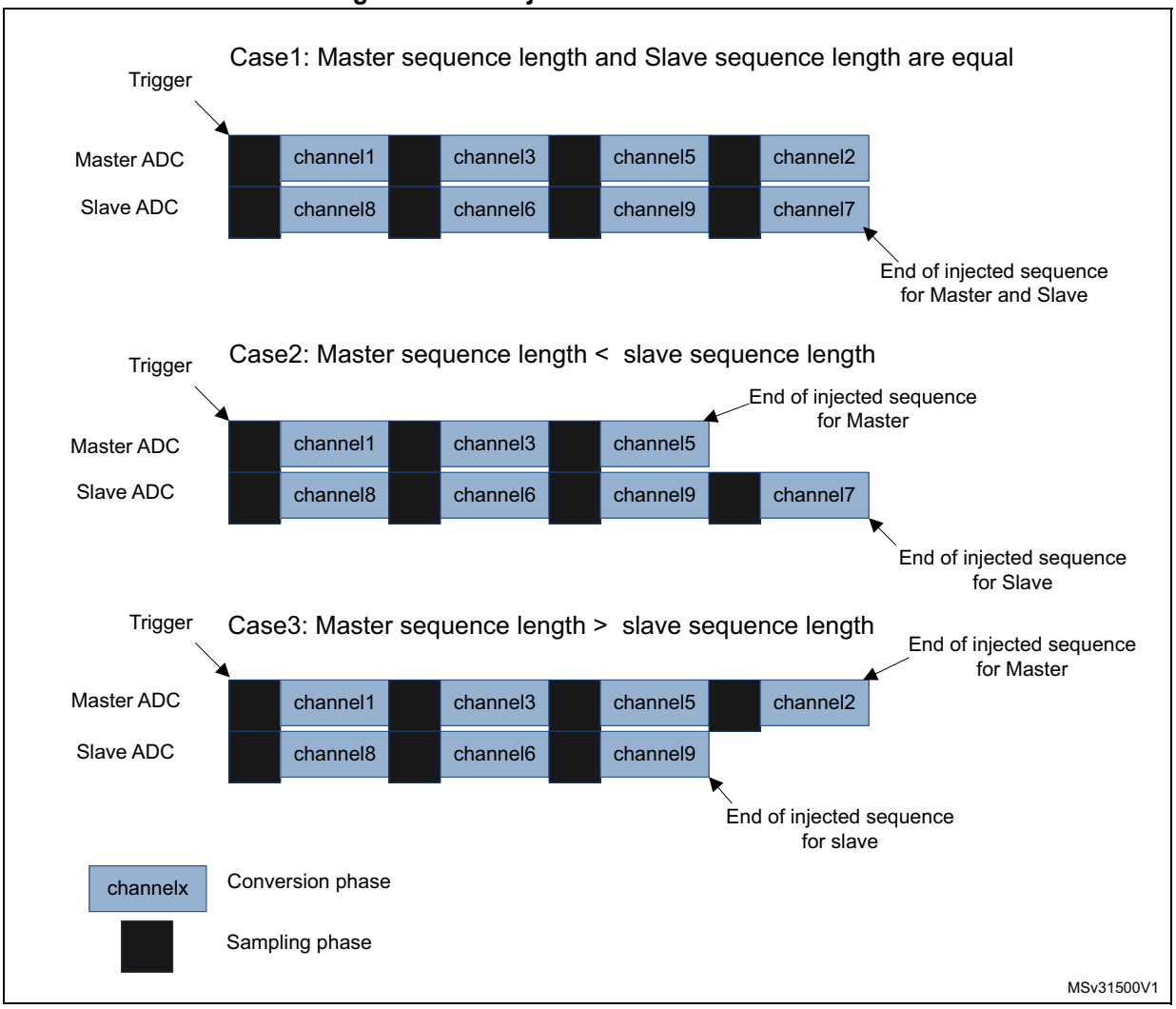
- **Injected simultaneous mode**: the master and slave injected channel groups are converted simultaneously after receiving an external trigger.
- **Regular simultaneous mode**: the master and slave regular channels are converted simultaneously.
- **Interleaved mode**: the master ADC starts immediately; then, after a programmed delay, the slave ADC starts.
- **Alternate trigger mode**: when the first trigger occurs, all injected master ADC channels in the group are converted; when the second trigger occurs, all injected slave ADC channels in the group are converted.

3.1 Dual injected simultaneous mode

For the injected simultaneous mode, the master and slave injected channel groups are converted simultaneously after receiving an external trigger. The master external trigger is used to convert the master and slave injected channel groups.

Figure 5 explains this mode.

Figure 5. Dual injected simultaneous mode



To use this mode, perform the following steps:

1. Configure the ADC master (channels to be converted, sampling times, etc.).
2. Configure the ADC slave (channels to be converted, sampling times, etc.).
3. Configure the ADC master external trigger to be used for triggering the injected conversion for both master and slave.
4. Select the dual mode by setting MULTI[4:0] to '00101' to select the injected simultaneous mode.
5. Select the ADC clock mode by setting the CKMODE[1:0] bits to the desired value.

The converted data results can be read separately from the:

- master ADC_JDR registers, using the JEOS interrupt and flag
- slave ADC_JDR registers, using the JEOS interrupt and flag

Note: If the durations of the master injected sequence and the slave injected sequence are equal, it is possible for the software to enable only one of the two JEOS interrupts (for example, the master JEOS), and to read both converted data results from the master ADC_JDRx and slave ADC_JDRx registers.

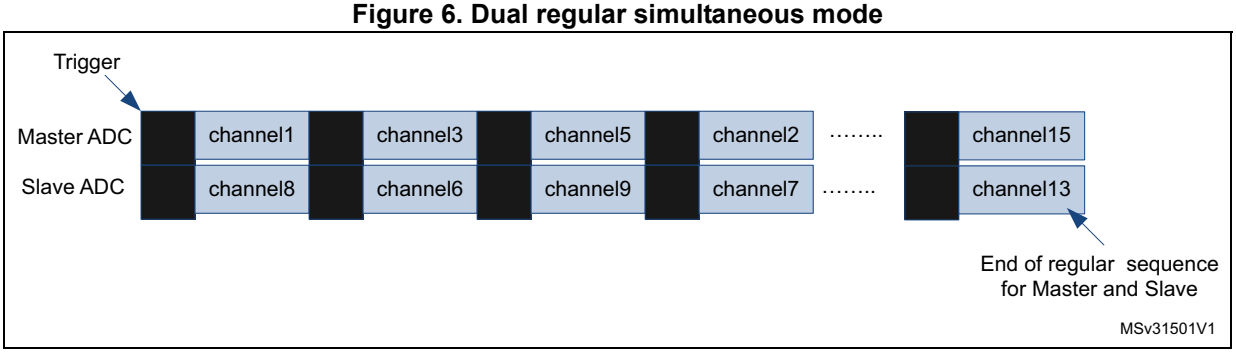
3.2 Dual regular simultaneous mode

For the regular simultaneous mode, the master and slave regular channel groups are converted simultaneously after receiving an external trigger. The master external trigger is used to convert the master and slave channel groups.

Note: Injected conversion is not supported.

Figure 6 explains this mode.

Figure 6. Dual regular simultaneous mode



To use this mode, perform the following steps:

1. Configure the ADC master (channels to be converted, sampling times, etc.).
2. Configure the ADC slave (channels to be converted, sampling times, etc.).
3. Configure the ADC master external trigger to be used to trigger the regular conversion for both master and slave.
4. Select the dual mode by setting MULTI[4:0] to ‘00110’ to select the regular simultaneous mode.
5. Select the ADC clock mode by setting the CKMODE[1:0] bits to the desired value.

To read the converted data, two configurations are possible: use of the EOC/EOS flags and interrupts, or use of the DMA.

3.2.1 Using the EOC/EOS flags and interrupts

In this configuration, the EOC interrupt or flag of the master is used to read the ADC master conversion sequence from the ADC_DR register. Similarly, the EOC interrupt or flag of the slave is used to read the slave-converted data from the ADC_DR register.

If the duration of the master regular sequence is equal to the duration of that of the slave, it is possible for the software to enable only one of the two EOC interrupts (for example, the master EOC), and to read both converted data results from the Common Data register ADC_CDR.

3.2.2 Using the DMA

When using the DMA, there are two possible cases:

- Use of two separate DMA channels for master and slave.

  Each ADC (in this case, the MDMA[1:0]) must be kept cleared. The first DMA channel is used to read the master ADC converted data from ADC_DR, and the DMA requests are generated at each EOC event of the master ADC.

  The second DMA channel is used to read the slave ADC converted data from ADC_DR, and the DMA requests are generated at each EOC event of the slave ADC.

- Use of the dual mode DMA.

  a) Configure MDMA[1:0]=0b10 or 0b11, depending on the resolution.

  b) A single DMA channel is used: the one belonging to the master. Configure the DMA master ADC channel to read the common ADC register ADC_CDR.

  A single DMA request is generated each time both master and slave EOC events have occurred. At that time, the slave ADC converted data is available in the upper half-word of the ADC_CDR 32-bit register, and the master ADC converted data is available in the lower half-word of the ADC_CCR register.

3.2.3 Application example

The dual regular simultaneous mode can be used in applications where two signals should be sampled and converted at the same time. For example, to measure and plot the single phase or three-phase instantaneous electrical power: $p_N(t) = u_N(t) \times i_N(t)$.

In this case, the voltage and current should be measured simultaneously and then the instantaneous power, which is the product of $u_N(t)$ and $i_N(t)$, should be computed.

To measure a single-phase power, ADC1 and ADC2 are used with two channels (1 channel for the voltage and 1 channel for the current).

To measure a three-phase power, ADC1 and ADC2 are used with 6 channels (3 channels for the voltage and 3 channels for the current).

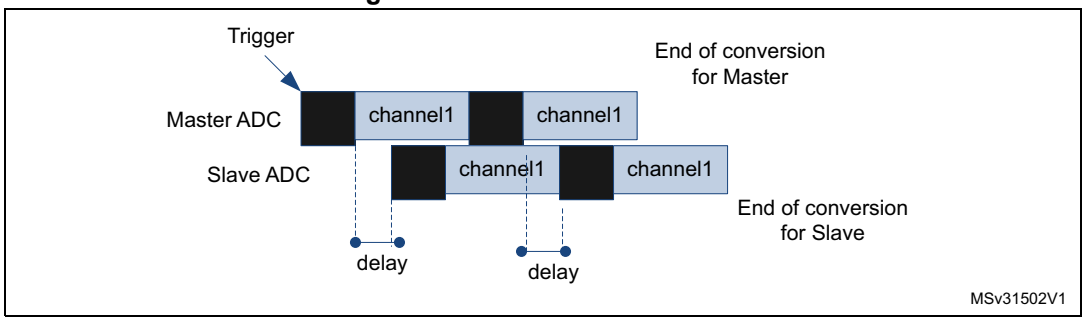
3.3 Dual interleaved mode

The Dual interleave mode can only be used with a regular group (usually one channel), and cannot used for injected conversion. The external trigger source comes from the regular channel multiplexer of the master ADC.

Once an external trigger occurs, the master ADC starts immediately; the slave ADC starts after a programmable delay of several ADC clock cycles after the sampling phase of the master ADC is complete.

Figure 7 explains this mode.

Figure 7. Dual interleaved mode



To use this mode, perform the following steps:

1. Configure the ADC master (channels to be converted, sampling times, etc.).
2. Configure the ADC slave (channels to be converted, sampling times, etc.).
3. Configure the ADC master external trigger to be used to trigger the regular conversion for both master and slave.
4. Select the dual mode by setting MULTI[4:0] to ‘00111’ to select the interleave mode.
5. Select the ADC clock mode by setting the CKMODE[1:0] bits to the desired value.
6. Program the delay which separates two conversions in interleaved mode, by writing the DELAY bits in the ADC_CCR register.

To read the converted data, two configurations are possible: use of the EOC/EOS flags and interrupts, or use of the DMA.

3.3.1 Using the EOC/EOS flags and interrupts

In this configuration, a master EOC interrupt is generated (if EOCIE is enabled) at the end of each conversion event (EOC) on the master ADC, and software can read the ADC_DR of the master ADC.

Similarly, a slave EOC interrupt is generated (if EOCIE is enabled) at the end of each conversion event (EOC) on the slave ADC, and software can read the ADC_DR of the slave ADC.

3.3.2 Using the DMA

When using the DMA, there are two possible cases:

- Use of two separate DMA channels for master and slave.

  Each ADC (in this case, the MDMA[1:0]) must be kept cleared. The first DMA channel is used to read the master ADC converted data from ADC_DR, and the DMA requests are generated at each EOC event of the master ADC. The second DMA channel is used to read the slave ADC converted data from ADC_DR, and the DMA requests are generated at each EOC event of the slave ADC.
- Use of the dual mode DMA.

  a) Configure MDMA[1:0]=0b10 or 0b11, depending on the resolution.

  b) A single DMA channel is used: the one belonging to the master. Configure the DMA master ADC channel to read the common ADC register ADC_CDR.

  A single DMA request is generated each time both master and slave EOC events have occurred. At that time, the slave ADC converted data is available in the upper half-word

of the ADC_CDR 32-bit register, and the master ADC converted data is available in the lower half-word of the ADC_CCR register.

How to configure the interleave mode to reach the maximum conversion rate

The sampling ADC rate is calculate as follows:

t_{ADC} = TSMPL + TSAR.

To have the maximum conversion rate, the minimum sampling time should be used (1.5 cycles) and if the ADC is used with the maximum resolution, the equation will be:

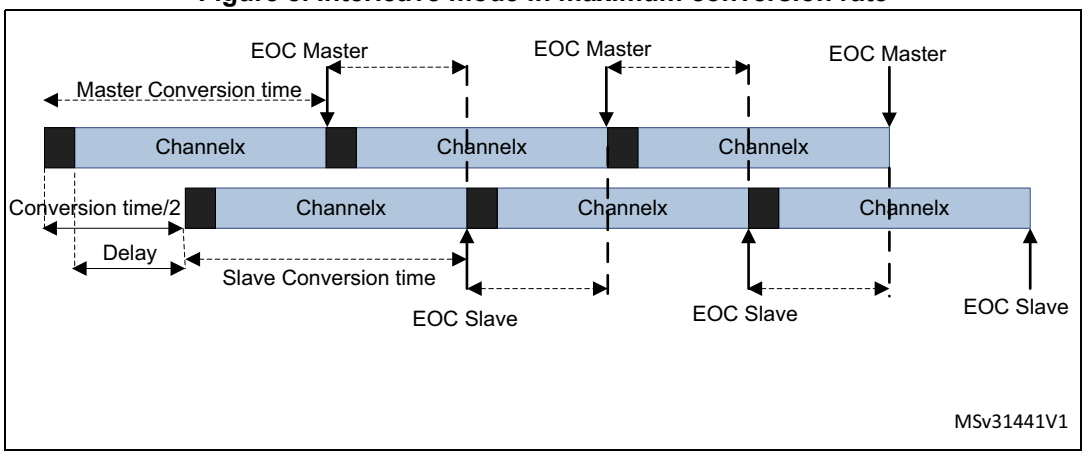
t_{ADC} = [1.5 | min + 12.5 | 12bit] x t_{ADC_CLK}

with t_{ADC_CLK} = 72 MHz, the t_{ADC} is 5.1 Msample/s, this rate can achieve the double, 10.2 Msamples/s, when the conversion of ADC Channelx is done in interleaved mode (the master and the slave convert the same Channelx with a specific delay value):

- Sampling time 1.5 cycles (the minimum possible sampling time)
- The time between the start of the Master conversion and the start of the slave conversion should be fixed to t_{ADC}/2.

In this case the Channelx will be converted each t_{ADC}/2 time, as explained in the following figure.

Figure 8. Interleave mode in maximum conversion rate



If the sampling time = 1.5 cycles and 12 bit resolution is used:

The Master conversion time = t_{ADC} = 14 cycles

The Slave conversion time = t_{ADC} = 14 cycles

The channelx is converted each (Master/Slave conversion time) / 2 = 7 cycles, so in this case the ADC conversion rate is 10.2 Msamples/s.

As it shown it the figure, the Delay = (Master/slave conversion time) - sampling phase, so the delay to be used in the interleaved mode to have the maximum rate is:

Delay = 5 cycles, so the DELAY[3:0] bits in ADCx_CCR should be set to 4.

3.3.3 Application example

For example if a signal to be converted has a maximum frequency of 4MHz, the sampling rate frequency should be higher than or equal to twice the frequency of the signal to be

converted (in accordance with Shannon-Nyquist criteria). Since the maximum sampling rate is 5.1 Msample/second with one ADC, the criteria cannot be met. This is solved by using the dual fast interleaved ADC mode. In this way, the sampling rate frequency becomes 10.2 Msample/second since the two ADCs (ADC1 and ADC2) work alternately and sample the signal at equal periods (7 ADC cycles).
With the dual interleaved ADC mode, 10.2 Msamples/second can be achieved with 1.5 cycles of sampling time.

3.4 Dual alternate trigger mode

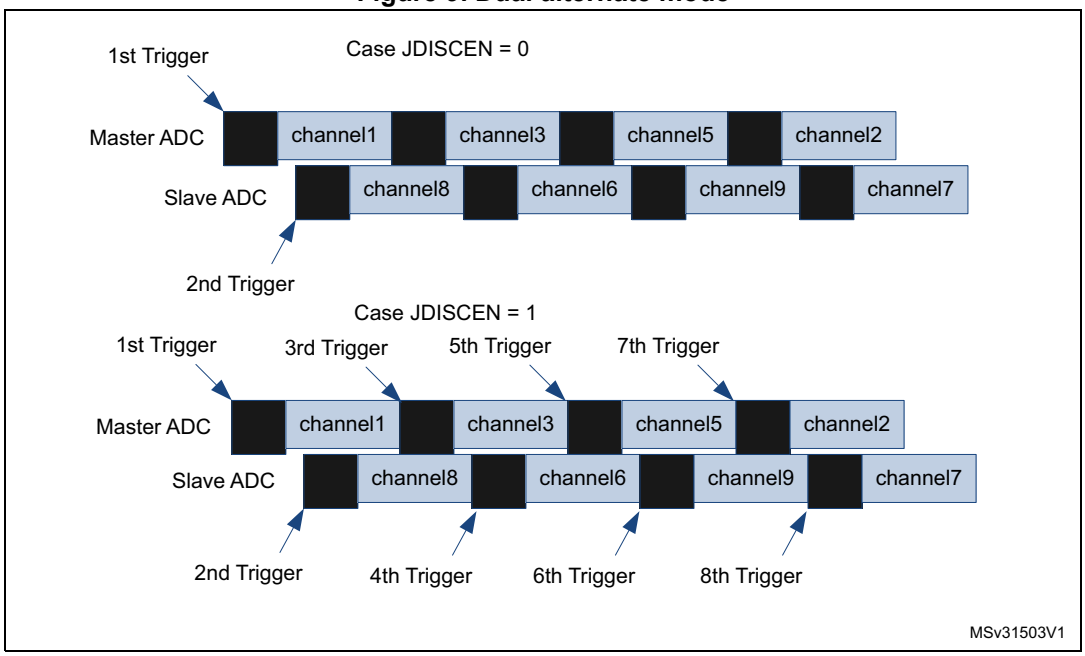
The dual alternate mode can be used only with injected channel groups, and cannot be used for regular conversion. The external trigger source comes from the injected channel multiplexer of the master ADC.

When the first external trigger occurs, the master ADC injected channels are converted. When the second external trigger occurs, the slave ADC injected channels are converted, and so on.

If another external trigger occurs after conversion of all injected channels in the group, then the alternate trigger process restarts by converting the injected channels of the master ADC in the group.

Figure 9 explains this mode.

Figure 9. Dual alternate mode



To use this mode, perform the following steps:

1. Configure the ADC master (channels to be converted, sampling times, etc.).
2. Configure the ADC slave (channels to be converted, sampling times, etc.).
3. Configure the ADC master external trigger to be used to trigger the injected conversion for both master and slave.
4. Select the dual mode by setting MULTI[4:0] to ‘01001’ to select the alternate trigger mode.
5. Select the ADC clock mode by setting the CKMODE[1:0] bits to the desired value.

The JEOS flags and interrupts for master and slave are used to read the converted data.

If enabled, a JEOS interrupt is generated after conversion of all injected channels of the master ADC in the group. The converted data can then be read from the ADC_JDRx master registers.

If enabled, a JEOS interrupt is generated after conversion of all injected channels of the slave ADC in the group. The converted data can then be read from the ADC_JDRx slave registers.

3.4.1 Application example

The dual alternate trigger mode makes it possible to have sampling points as close as possible to each other (down to 1.5 ADC cycles). This is interesting for instance in motor control applications, where a single-shunt sensor is used for three-phase current reading. In some cases, the PWM duty cycle of the power stage has to be limited to maintain a minimum time slot for two consecutive conversions. If the sampling points are as close as possible, the PWM duty cycle is maximized, which increases the voltage applied to the motor.

4 Conclusion

This application note explains certain specific ADC modes, and provides typical configuration procedures to help with understanding the presented modes.

The STM32's ADC has several modes which are intended for advanced conversion processes, so as to attain efficient conversion results in applications such as motor control.

DMA is a major feature, and its use is recommended whenever possible in order to avoid sample loss, and to release the CPU load.

The queue of context is a new feature for injected channels embedded in the F3 ADC family to anticipate up to 2 contexts for the next injected sequence of conversions.

5 Revision history

Table 3. Document revision history

Date	Revision	Changes
04-Mar-2013	1	Initial release.