AN3116
Application note

STM32™'s ADC modes and their applications

Introduction

STM32 microcontrollers have one of the most advanced ADCs on the microcontroller market. You could imagine a multitude of applications based on the STM32 ADC features. Some ADC modes are provided to simplify measurements and give efficient results in applications such as motor control.

This application note provides help for ADC users to understand some advanced modes offered in the STM32 microcontrollers, and to quick start development. Each of the described modes is provided with an example of application to better understand how to use them. Most modes come with a basic firmware to make it easier to understand the ADC configuration.

This application note is divided into two sections: independent modes and dual modes. The first section describes modes used with a single ADC. The second section describes modes that should be used with two ADCs (ADC1 and ADC2 working jointly).

This application note does not describe the ADC modes that result from the combination of other modes.

Contents

List of figures

1 Independent modes

1.1 Single-channel, single conversion mode

1.1.1 Description

This is the simplest ADC mode. In this mode, the ADC performs the single conversion (single sample) of a single channel x (refer to *Figure 1.*) and stops after completion of the conversion.

Note: *This application note is not delivered with a firmware example of this mode.*

Figure 1.   Single-channel, single conversion mode
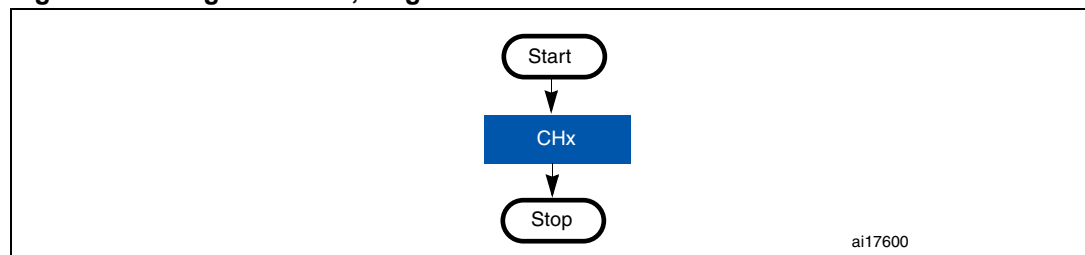
1.1.2 Example of application

This mode can be used for the measurement of a voltage level to decide if the system can be started or not. Measure the voltage level of the battery before starting the system: if the battery has a low level, the "low battery" message appears. In this case, do not start the system.

1.2 Multichannel (scan), single conversion mode

1.2.1 Description

This mode is used to convert some channels successively in independent mode. With the ADC sequencer, you can use this ADC mode to configure any sequence of up to 16 channels successively with different sampling times and in different orders. You can for example carry out the sequence shown in *Figure 2.* In this way, you do not have to stop the ADC during the conversion process in order to reconfigure the next channel with a different sampling time. This mode saves additional CPU load and heavy software development.

Figure 2.   ADC sequencer converting 7 channels with different configured sampling times

Note: *This application note is not delivered with a firmware example of this mode.*

Figure 3.   Multichannel, single conversion mode
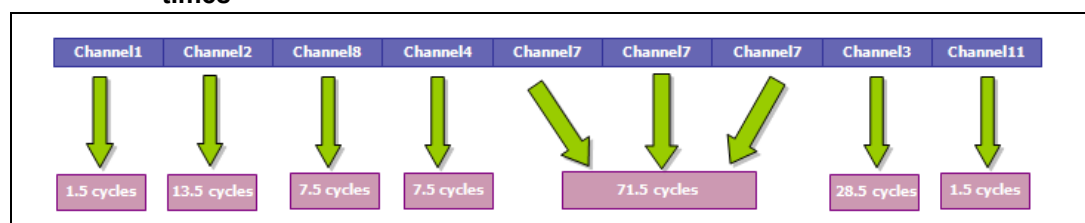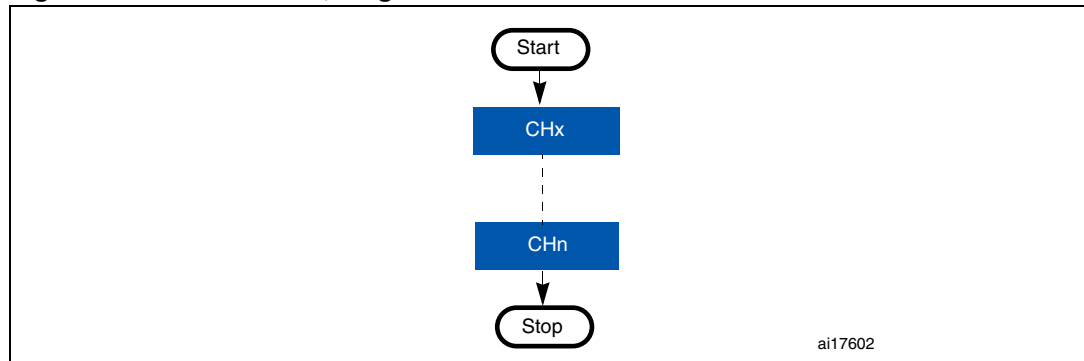

1.2.2   **Example of application**

This mode can be used when starting a system depends 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.

1.3   Single-channel continuous conversion mode

1.3.1   Description

The single-channel continuous conversion mode converts a single channel continuously and indefinitely in regular channel conversion.

The continuous mode feature allows the ADC to work in the background. The ADC converts the channels continuously without any intervention from the CPU. Additionally, the DMA can be used in circular mode, thus reducing the CPU load.

Note: *An example of firmware is provided with this application note: SingleChannelContinuous. The example uses two methods: DMA and interrupts. To select either method, simply comment or uncomment* `#define USE_DMA_Transfer` *in the main.c file.*

Figure 4.   Single-channel, continuous conversion mode
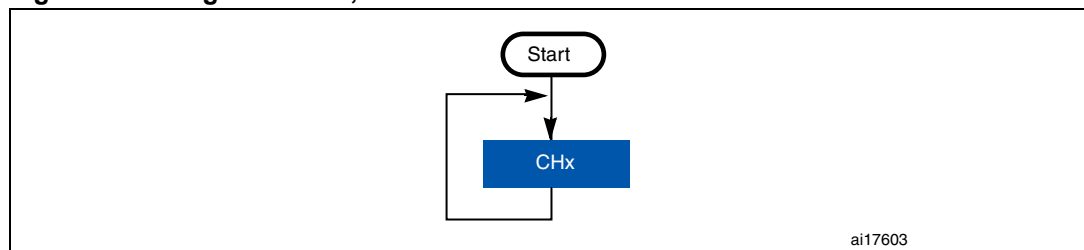
1.3.2 Example of applications

This ADC mode can be implemented to monitor a battery voltage, the measurement and regulation of an oven temperature, etc.

In the case of the oven temperature regulation, the temperature is read and compared to the temperature set by the user. When the oven temperature reaches the desired temperature, the heating resistor is powered off.

1.4 Multichannel (scan) continuous conversion mode

1.4.1 Description

The multichannel, or scan, continuous mode can be used to convert some channels successively with the ADC in independent mode. With the sequencer, you can configure any sequence of up to 16 channels successively with different sampling times and different orders. T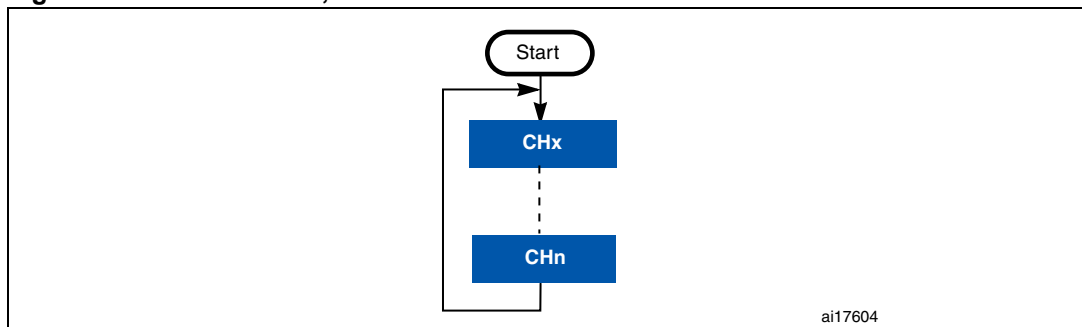his mode is similar to the multichannel single conversion mode except that it does not stop converting after the last channel of the sequence but it restarts the conversion sequence from the first channel and continues indefinitely.

Figure 5. Multichannel, continuous conversion mode

1.4.2 Example of application

This 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.

1.5 Injected conversion mode

1.5.1 Description

This mode is intended for use when conversion is triggered by an external event or by software.

The injected group has priority over the regular channel group. It interrupts the conversion of the current channel in the regular channel group.

Note: *An example of firmware is provided with this application note: Indep_InjectedGroup.*

Figure 6.   Injected conversion mode
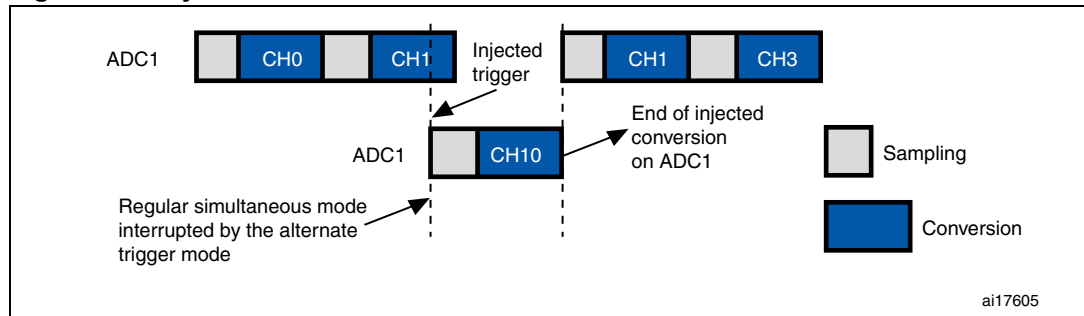
1.5.2   Example of application

This 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 to after the transistor switching.

2 Dual modes

Dual modes are available in STM32 microcontrollers that feature two ADCs: ADC1 master and ADC2 slave. ADC1 and ADC2 triggers are synchronized internally for regular and injected channel conversion. ADC1 and ADC2 work together.

In some devices, there are up to 3 ADCs: ADC1, ADC2 and ADC3. In this case ADC3 always works independently, and is not synchronized with the other ADCs.

Note: *Do not sample the same channel on the two ADCs at the same time, this would introduce conversion errors.*

2.1 Dual regular simultaneous mode

### 2.1.1 Description

The dual regular simultaneous ADC mode is used to perform two conversions simultaneously owing to the synchronization of ADC1 and ADC2. Each ADC converts a channel sequence (with scan enabled and the sequencer of each ADC configured) or converts a single channel (scan disabled). The conversion can be started with an external trigger or by software. In this mode, the conversion results of ADC1 and ADC2 are stored in ADC1's data register (32-bit format). *Figure 7* shows how ADC1 and ADC2 convert two sequences simultaneously. ADC1 converts a sequence of 16 channels successively: channel 15 to channel 0 and ADC2 converts a sequence of 16 channels successively: channel 0 to channel 15.

Note: *This application note does not provide a firmware example. However the STM32 firmware library provides an example of this mode at the following path:*
\Project\STM32F10x_StdPeriph_Examples\ADC\RegSimul_DualMode.

Figure 7.   Dual regular simultaneous mode
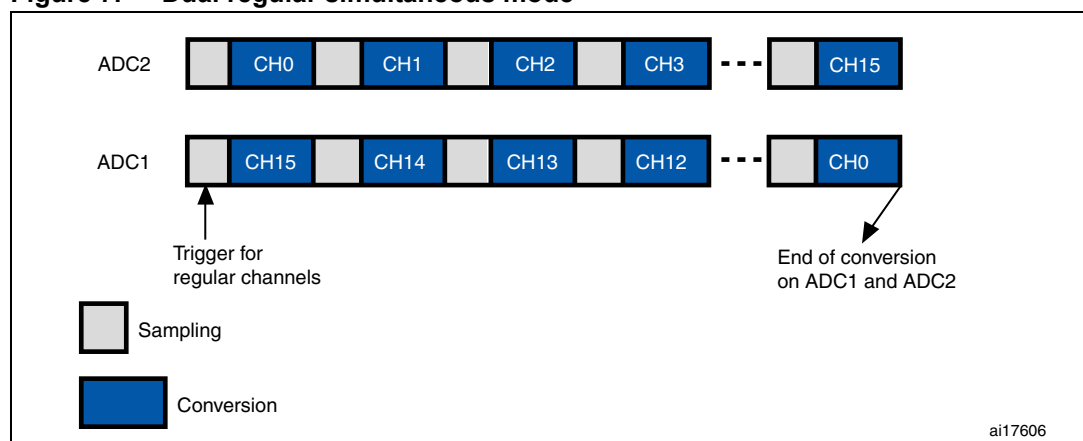
2.1.2 Example of application

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.

Figure 8 shows how to measure a power using the two ADCs in dual regular simultaneous mode.

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).

Figure 8.   Simultaneous measurement of voltage and current
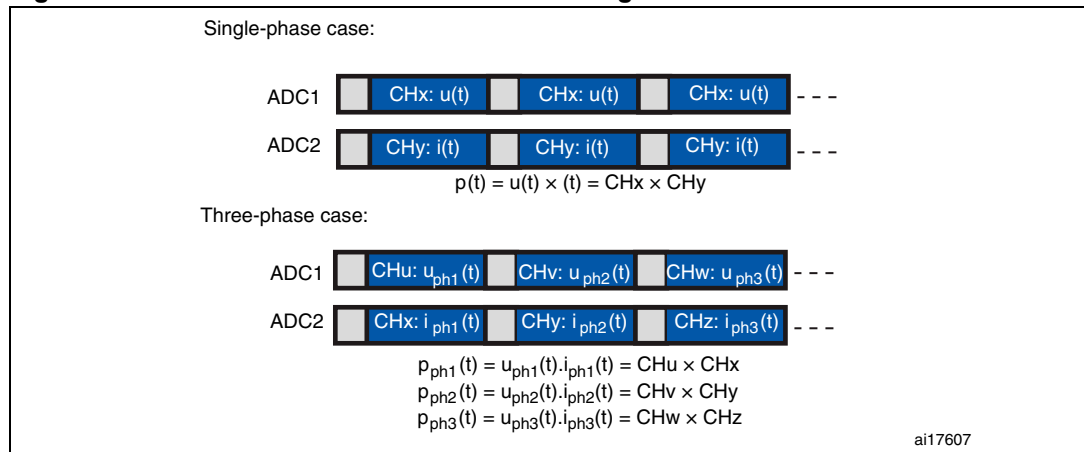
2.2   Dual fast interleaved mode

2.2.1   Description

The dual fast interleaved ADC mode is intended for the conversion of one channel. ADC1 and ADC2 convert the selected channel alternately with a period of 7 ADC clock cycles. This means that the channel is converted every 7 clock cycles. Each ADC converts the channel every 14 ADC clock cycles. With a 14 MHz ADC clock, it is thus possible to reach 2 Msamples per second: 14 MHz/7 = 2 MHz (sampling frequency). The conversion can be started by external trigger or by software and the conversion results of ADC1 and ADC2 are stored into ADC1 data register (32-bit format).

The maximum allowed sampling time is 7 ADC clock cycles to avoid the overlap between the ADC1 and ADC2 sampling phases in the event that they convert the same channel. This means that the only allowed sampling time is 1.5 cycles.

Note: *1   It is highly recommended to use DMA instead of interrupts to avoid the loss of data.*

2   An example of firmware is provided with this application note: Dual_FastInterleaved.

Figure 9.   Dual fast interleaved mode
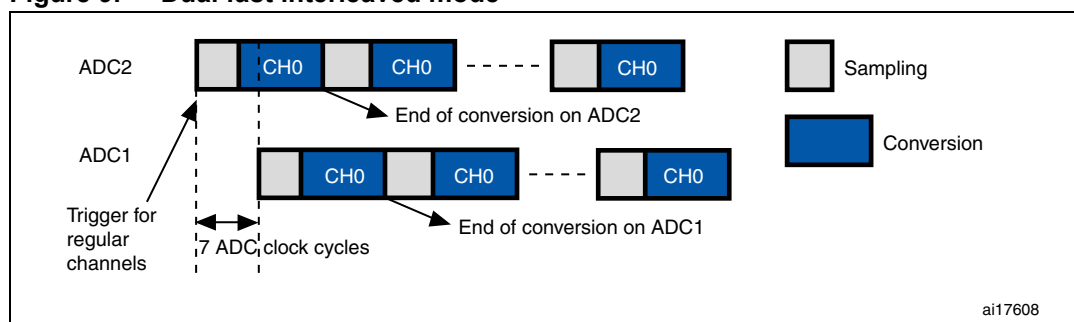
2.2.2   Example of application

This mode is used to speed up the sampling rate of the ADC when 2 Msamples/second are needed with 1.5 cycle of sampling time. ADC1 and ADC2 convert the same channel alternately to reduce the conversion time. While ADC1 samples channel CHx, ADC2 converts the previous sample.

For example if a signal to be converted has a maximum frequency of 800 kHz, 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 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 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 fast interleaved ADC mode, 2 Msamples/second can be achieved with 1.5 cycles of sampling time.

## 2.3   Dual slow interleaved mode

2.3.1   Description

The dual slow interleaved ADC mode is intended for the conversion of one channel. ADC1 and ADC2 convert the selected channel alternately with a period of 14 ADC clock cycles. The channel is thus converted every 14 clock cycles. Each ADC converts the channel every 28 ADC clock cycles. The conversion can be started by external trigger or by software and the conversion results of ADC1 and ADC2 are stored into ADC1's data register (32-bit format).

The maximum allowed sampling time is 14 ADC clock cycles to avoid any overlap with the next conversion. This means that the only allowed sampling times are 1.5, 7.5 and 13.5 cycles.

Continuous conversion should not be used in this mode since the ADCs would continuously convert the selected regular channel automatically (the CONT bit should not be set in this mode).

Note: *1*   *It is highly recommended to use DMA instead of interrupts to avoid the loss of data.*

          2   *An example of firmware is provided with this application note: Dual_SlowInterleaved.*

Figure 10.   Dual slow interleaved mode
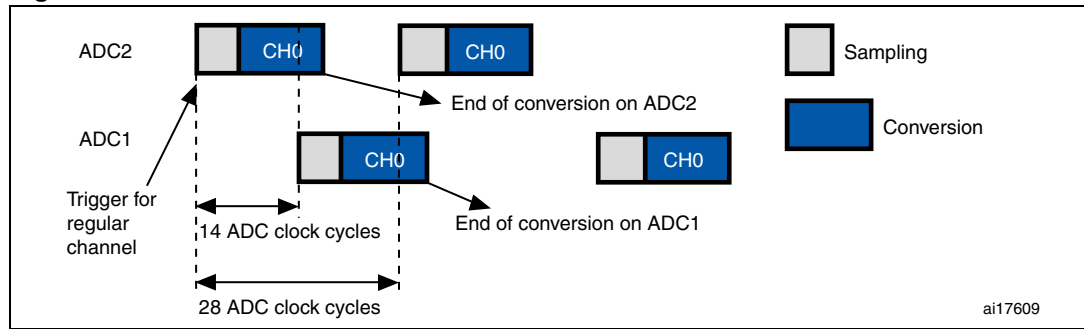
2.3.2 Example of application

Let us assume that the signal to be converted has a maximum frequency (f_{in}) of 500 kHz with an impedance R_{AIN} = 10 kΩ. The minimum sampling rate should be 1 Msamples/second (in accordance with Shannon-Nyquist criteria).

With one ADC, the only configuration is f_{ADC} = 14 MHz and the sampling time T_s = 1.5 cycles ($T_s = t_s \times f_{ADC}$, refer to the datasheet of the STM32 product you are using for more details on the sampling time vs. the source impedance).

In this case, the maximum allowed impedance of the source (R_{AIN}max) is 1.2 kΩ. The conversion will not be accurate since $R_{AIN} > R_{AINmax}$.

Solution

Use the ADC in dual slow interleaved mode. Each ADC sampling should be configured to have a minimum sampling rate of 500 ksamples/second with f_{ADC} = 14 MHz.

$$T_{TotalRate} = T_s + T_{conv} = t_{TotalRate} \times f_{ADC} = \frac{f_{ADC}}{F_{sRate}} \text{, where:}$$

– $T_{TotalRate}$ is the total conversion time in ADC clock cycles.
– $t_{TotalRate}$ is the sum of the sampling time t_s ($t_s = T_s/f_{ADC}$) and the conversion time t_{conv} ($t_{conv} = T_{conv}/f_{ADC}$)
– T_{conv} is a constant equal to 12.5 cycles

T_s is the sampling time given in ADC cycles, it is equal to:

$$T_s = \frac{f_{ADC}}{F_{sRate}} - 12.5 = \frac{14 \times 10^6}{500 \times 10^3} - 12.5 = 15.5 \text{ cycles , where } F_{sRate} \text{ is the sampling rate frequency.}$$

Since T_s = 15.5 cycles is not available for the configuration (for more details about the different available T_s values, refer to the "12-bit ADC characteristics" section in the datasheet of the STM32 product you are using). The nearest available value is T_s = 13.5. To know whether this value meets Shannon-Nyquist criteria, we need to calculate the sampling rate with T_s = 13.5 and f_{ADC} = 14 MHz:

$T_{TotalRate}$ = 13.5 + 12.5 = 26 cycles
$F_{sRate} = f_{ADC}/T_{TotalRate} = 14.10^6 / 26 = 538$ kHz per ADC

Since the conversion is made with two ADCs (ADC1 + ADC2) working alternately, the sampling rate frequency is: $F_{sRate} = 538 \times 2 = 1076$ kHz $> 2 \times f_{in}$, and so the Shannon-Nyquist criteria is respected.

With T_s = 13.5 cycles, the maximum allowed impedance of the source (R_{AIN}) is 19 kΩ. Since the source has an impedance R_{AIN} = 10 kΩ < 19 kΩ the impedance condition is respected. With a single ADC and with T_s = 1.5 (corresponding to 1.2 kΩ), the impedance condition is not satisfied. By using two ADCs, we have thus extended the impedance margin from 1.2 kΩ to 19 kΩ.

With the dual slow interleaved ADC mode, the 1 Msamples/second sampling rate is achieved with higher input impedances.
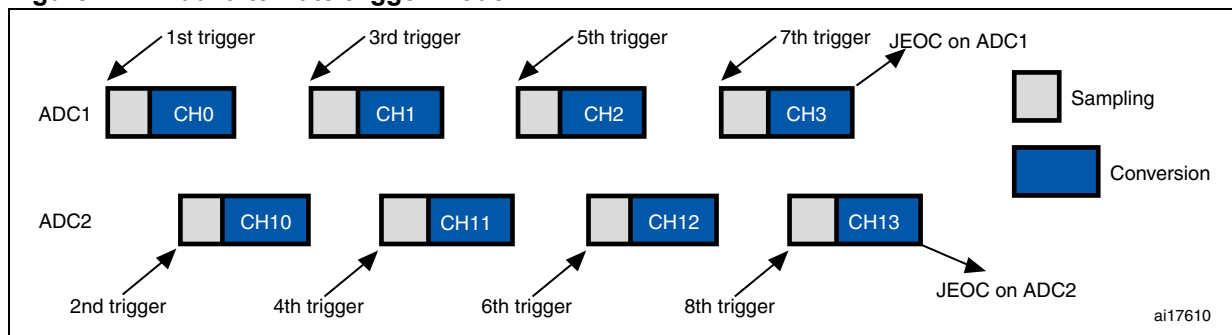
2.4 Dual alternate trigger mode

2.4.1 Description

The dual alternate trigger ADC mode can be used only on an injected channel group. In this mode, ADC1 and ADC2 alternately convert the injected channels on the same external trigger. When the first trigger occurs, all injected group channels in ADC1 are converted. When the second trigger happens, all injected group channels in ADC2 are converted and so on. The maximum number of injected channels in the group is 4 for each ADC.

Note: *An example of firmware is provided with this application note: Dual_AlternateTrigger.*

Figure 11. Dual alternate trigger mode

2.4.2 Example of application

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.

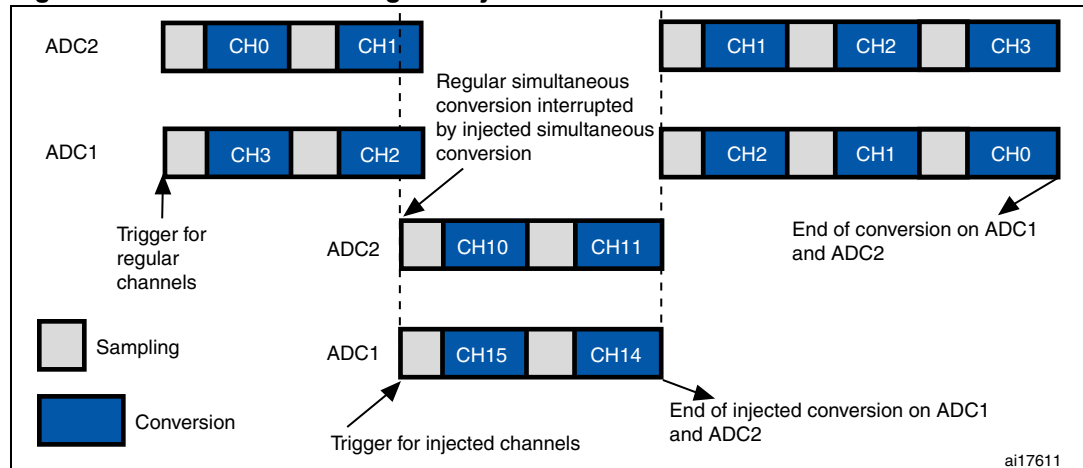## 2.5 Dual combined regular/injected simultaneous mode

2.5.1 Description

The dual combined regular/injected simultaneous ADC mode is a regular simultaneous mode that allows injection. The injected channels are also converted simultaneously.

In this mode, you should convert sequences with the same length or ensure that the interval between triggers is longer than the longest of the 2 sequences. Otherwise, the ADC with the shortest sequence may restart while the ADC with the longest sequence is completing the previous conversions.

Note: *An example of firmware is provided with this application note: Dual_Combined_Reg_Inj_Sumultaneous. The example uses two methods: DMA and interrupts. To select either method, simply comment or uncomment #define USE_DMA_Transfer in the main.c file.*

Figure 12. Dual combined regular/injected simultaneous mode

2.5.2 Example of application

The dual combined (regular/injected) simultaneous mode can be used to read 3-phase currents at definite times. This mode is interesting when you read the currents of two of the phases and you determine the current of the third phase by extrapolation since there is a relationship between the currents of the 3 phases.

2.6 Dual combined: injected simultaneous + interleaved mode

2.6.1 Description

This mode is the combination of the dual interleaved mode (fast or slow) and the dual injected simultaneous mode. When the regular channel is triggered, the dual interleaved conversion starts: ADC2 makes the first conversion, then ADC1 converts the channel and so on. When the injected channel is triggered, it interrupts the interleaved channel conversion and the two ADCs (ADC1 and ADC2) start converting the group of injected channels. When the two ADCs finish the injected conversion, they resume the conversion of the channel configured in interleaved mode.

Note: *This application note is not delivered with a firmware example of this mode.*

Figure 13.   Dual combined: injected simultaneous + interleaved mode
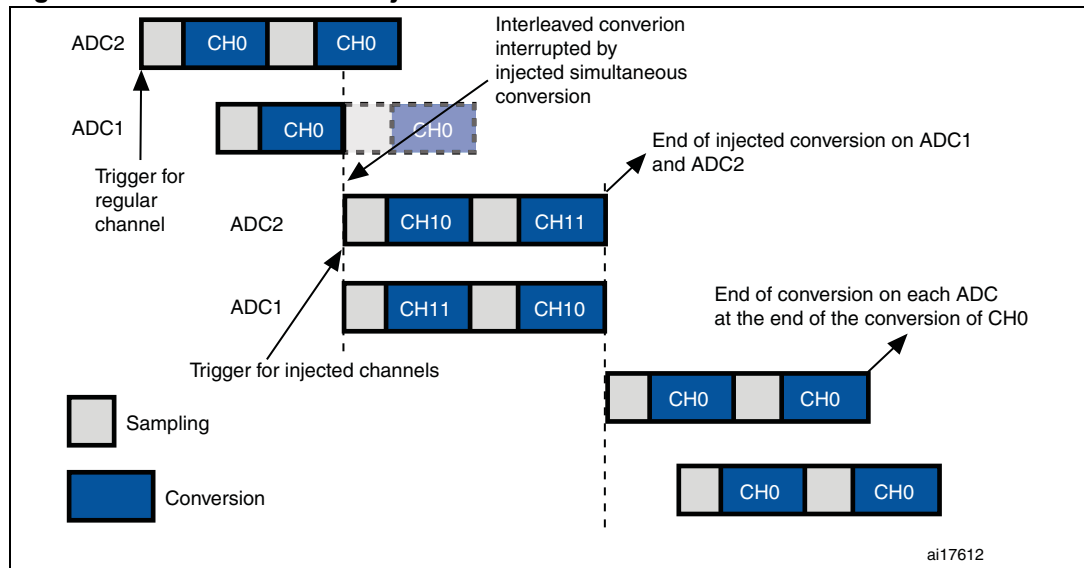
2.6.2   Example of application

You can use this dual combined mode in a UPS system (uninterruptible power supply). The battery voltage is monitored with the ADC watchdog converted in dual interleaved mode. The ADC also monitors the power consumed by the load by measuring the voltage and the current in dual injected simultaneous mode.

Figure 14.   UPS system
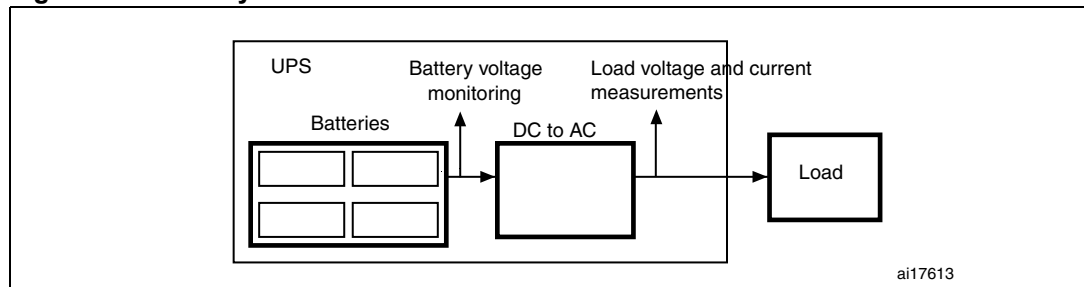
3 Conclusion

This application note explains some ADC modes, and provides examples of applications to make it easier to understand the presented modes. Most examples come with firmware to simplify the ADC configuration and speed up development. With the provided firmware examples, you can modify the code a little to switch from one mode to another.

The STM32's ADC has several modes 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 when possible to avoid the loss of samples and release the CPU load.

4   Revision history

Table 1.   Document revision history

Date	Revision	Changes
02-Mar-2010	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2010 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com