
Analog-to-Digital Converter (ADC)



Luigi Coppolino, Giovanni Mazzeo

Outline

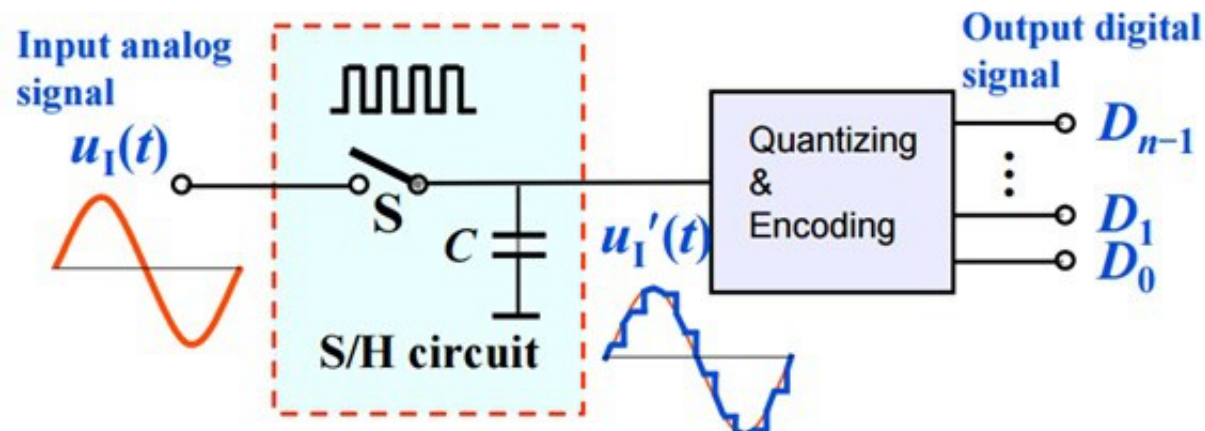
- Cos'è un ADC converter
- L'ADC Converter sulla STM32F3-Discovery
 - Schema
 - I modi di Funzionamento
 - Modello di Programmazione
 - Esercizio: Lettura Temperatura Interna al SoC
- Il DAC Converter sulla STM32F3-Discovery
 - Schema
 - I modi di Funzionamento
 - Modello di Programmazione
 - Esercizio

Analog-to-Digital Converter (ADC)

- Un **Analog-to-Digital Converter (ADC)** è una periferica fondamentale per ogni strumento di misura.
- Un ADC converte un segnale analogico con andamento continuo (ad es. una tensione) in una serie di valori discreti.
- Un ADC a partire da un voltaggio compreso tra 0 e V_{ref} di uno specifico input, genera un numero compreso tra 0 e $2^N - 1$ dove N è la sua risoluzione
- Quando si fa uso di un ADC è necessario definire la frequenza alla quale si vuole campionare il segnale di tensione in ingresso.
- Tale frequenza è chiamata **sampling rate** (frequenza di campionamento) del convertitore.

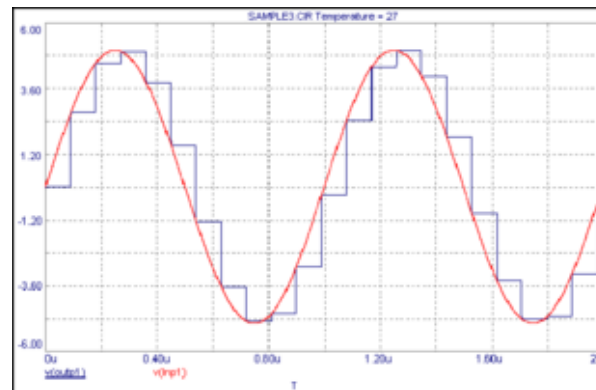
Come funziona un ADC

- Il funzionamento di un ADC si basa su due fasi:
 - S/H: Sampling and holding
 - Q/E: Quantizing and Encoding



S/H: Sampling and Holding 1/2

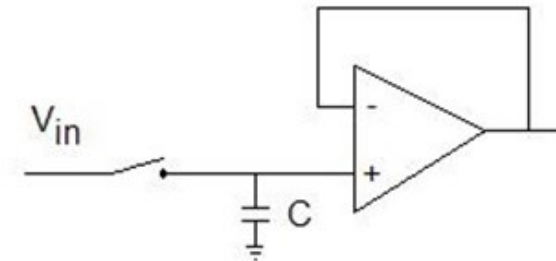
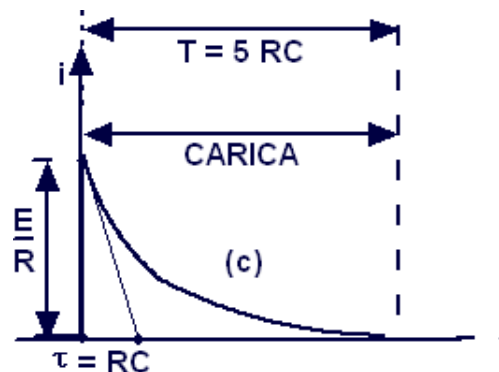
- Un segnale cambia continuamente nel tempo, per effettuare il campionamento è necessario mantenerlo costante per una certa durata di tempo in modo da generare un segnale a gradino che possa essere quantizzato dal Q/E
- Il *Sample and Hold circuit* quindi mantiene (hold) il segnale costante per la durata richiesta dal campionatore a digitalizzare il valore.



S/H: Sampling and Holding 2/2

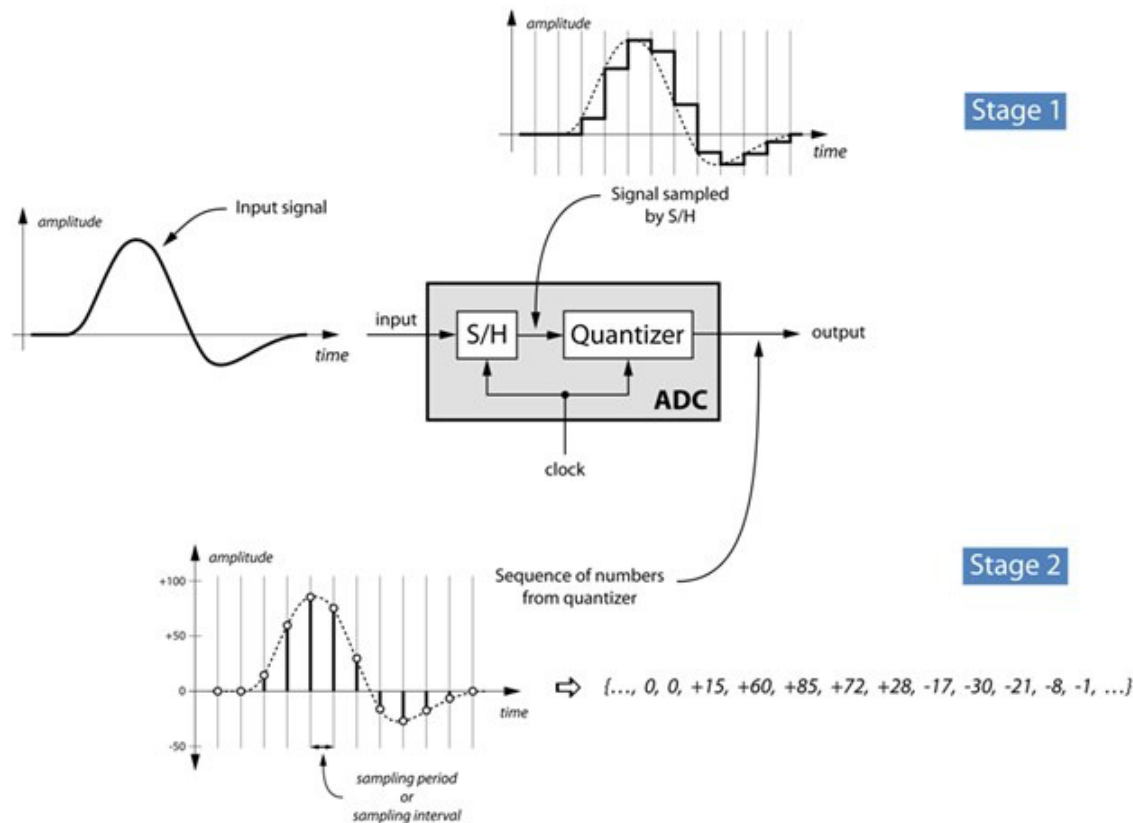
- Il circuito che permette il funzionamento del S/H è il seguente:
 - Lo switch è normalmente mantenuto aperto. Quando si vuole effettuare la misurazione lo si chiude. Il condensatore si carica e mantiene il segnale costante per una durata data dalla sua Legge di Scarica.
- Il condensatore si scarica con legge esponenziale:

$$i = \frac{V}{R} e^{-\frac{t}{RC}}$$



Q/E: Quantizing and Encoding 1/2

- In uscita dunque dal S/H ci sarà dunque un segnale analogico a gradino che deve essere quantizzato.

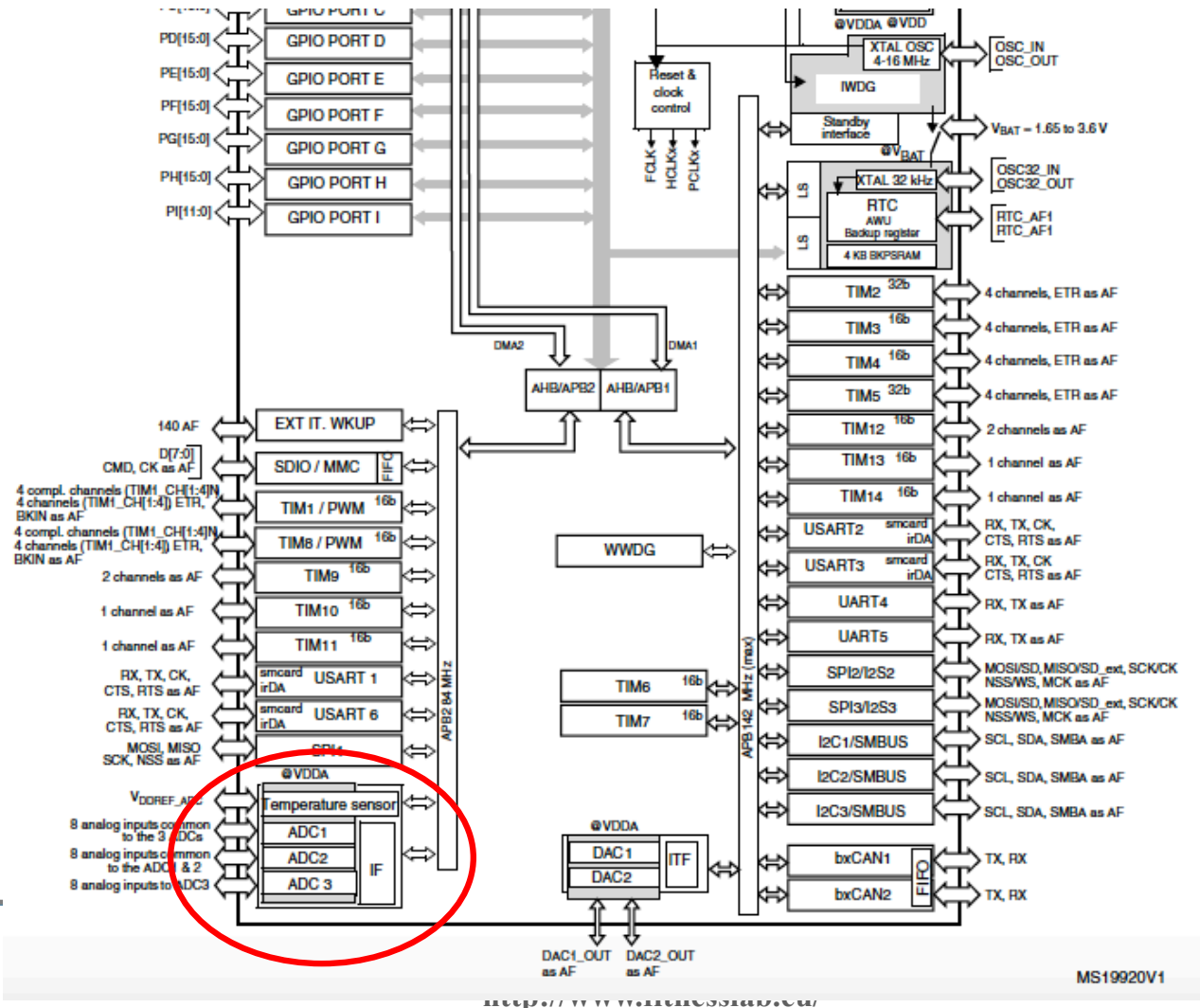


Q/E: Quantizing and Encoding 2/2

- Ad ogni livello di tensione è assegnato un valore numerico che dipende quindi dall'ampiezza del segnale in uscita dal S/H. Tale valore varia in un range dato dalla risoluzione del Quantizer (in potenza di 2^n).
- Dopo che tale valore numerico è stato definito sarà convertito dall'*Encoder* in formato binario su n bit dove n è la risoluzione del Q/E.
- Il valore ottenuto non sarà mai veramente accurato ma sempre un approssimazione di valori del mondo reale.
- Chiaramente, maggiore è la risoluzione maggiore sarà la qualità.

L'ADC sulla STM32F4-Discovery

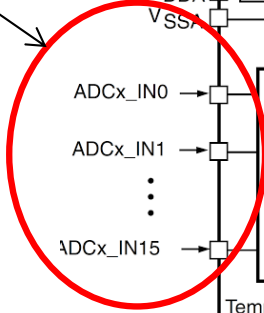
- Il Microcontrollore STM32F407 installato sulla scheda STM32F3-Discovery presenta al suo interno tre ADC a 12 bit, ognuno



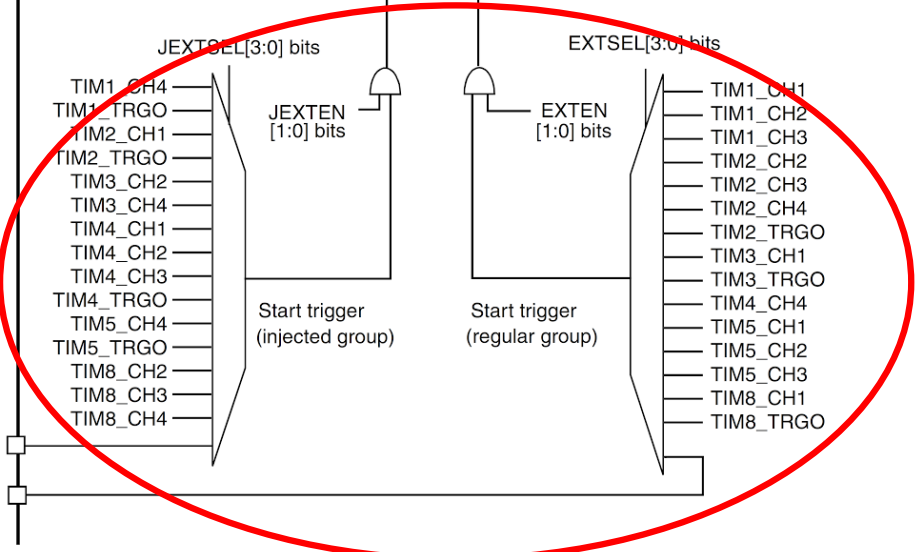
L'ADC sulla STM32F4-Discovery

- L'ADC presenta differenti linee di ingresso per i segnali provenienti dal mondo esterno. Tali linee di ingresso sono connesse a specifici GPIO (vedere lo user manual come fatto per i buttons) che devono essere configurati in **Analog Mode**.
- Inoltre l'ADC può essere configurato a funzionare con interrupt di modo da segnalare il processore quando una conversione è pronta.
- L'ADC può essere triggerato (ovvero abilitato ad eseguire la conversione) da timer TIMs in modo da effettuare conversioni con specifiche cadenze temporali.

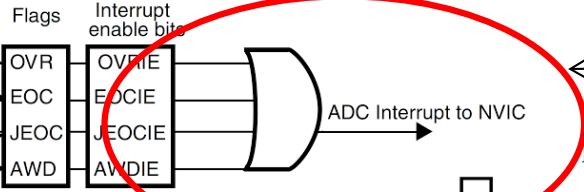
I segnali analogici in ingresso all'ADC



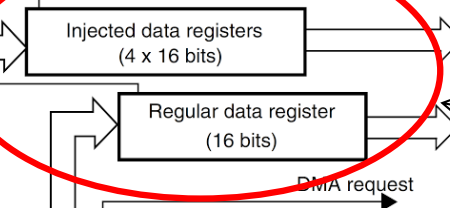
I Timer TIM che abilitano la conversione



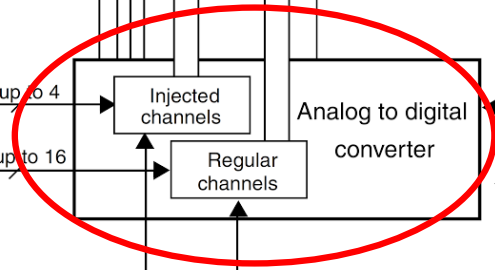
La linea di interruzione che permette di segnalare che una conversione è pronta



i registri dove vengono mantenuti i valori delle conversioni



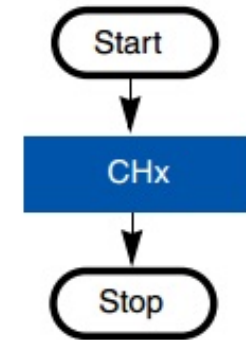
Il core dell'ADC dove ci saranno le due unità: S/H e Q/E



ADC Modi di Funzionamento 1/4

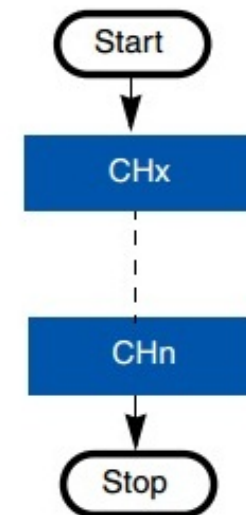
➤ Single Channel Single Conversion

Es. Lettura del valore di tensione di alimentazione prima dell'avvio del dispositivo



➤ Multiple Channel Single Conversion

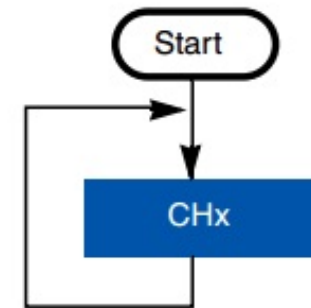
Es. Lettura del valore di posizione di ciascun asse di un braccio meccanico durante la fase di inizializzazione



ADC Modi di Funzionamento 2/4

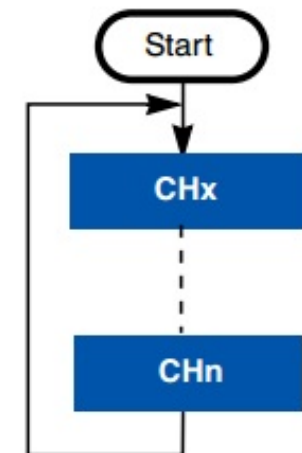
➤ Single Ch. Continuous Conversion

Es. Misura di una temperatura per la regolazione di un forno



➤ Multiple Ch. Continuous Conversion

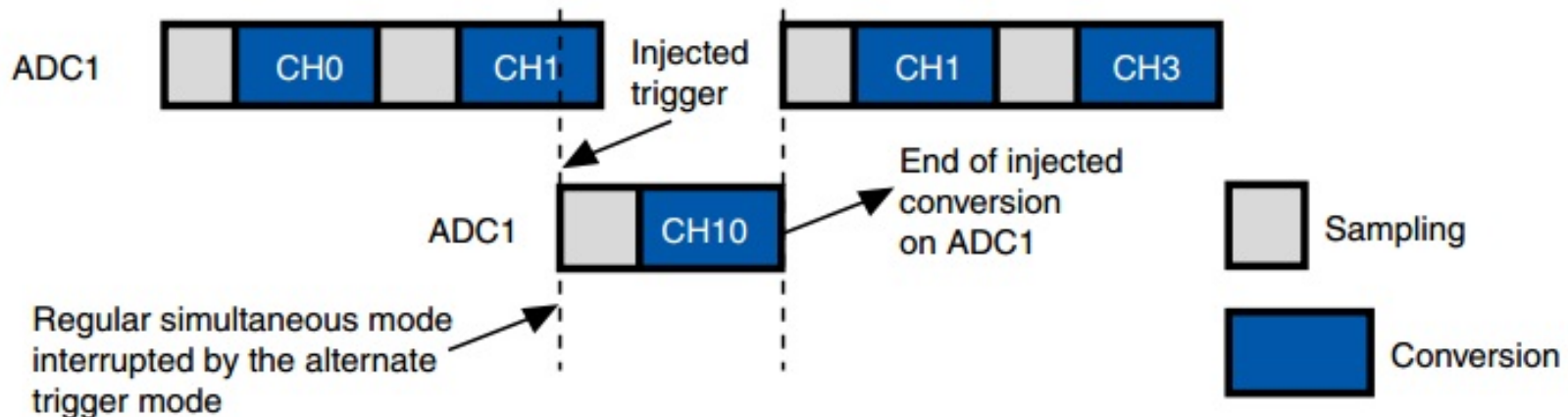
Es. Lettura del valore di carica e/o temperatura di ogni cella in un battery pack



ADC Modi di Funzionamento 3/4

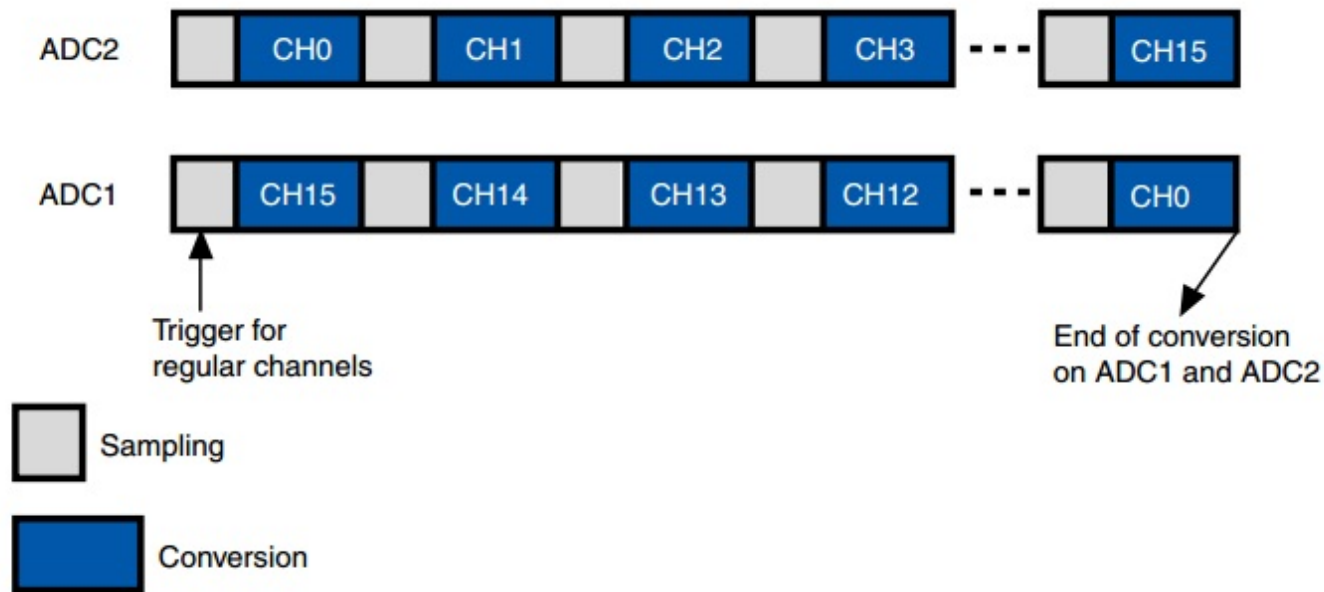
➤ Injected Conversion

- Usata se la conversione è attivata da un evento esterno: non coinvolge la CPU
- Ha priorità sul Regular Group: interrompe la conversione attualmente in atto su tale gruppo



ADC Modi di Funzionamento 4/4

➤ Dual regular simultaneous mode



Es. Lettura di tensione e corrente per il calcolo della potenza istantanea

Modello di Programmazione

- Al solito per iniziare a programmare un ADC dobbiamo definirne il suo modello di programmazione
- Oltre a definire una delle modalità di funzionamento viste prima è necessario definire:
 - Quale ADC (ADC1, ADC2,..) e quale channel si intende usare.
 - Il GPIO associato all'ingresso della specifica linea dell'ADC e quindi configurarlo (importante impostare Analog Mode)
 - La risoluzione (12 bit ad esempio)
 - Se fare uso di un trigger esterno che abiliti l'ADC
 - Il numero di conversioni che si desidera effettuare
 - Se fare uso di interrupts per ricevere un evento quando una misura è “pronta”
 - altro



Esercizio: Leggere La Temperatura Interna della Scheda 1/4

- Leggere la temperatura del sensore interno alla scheda facendo uso delle interrupt
- La STM32F3 presenta un sensore di temperatura interna del SoC che è connesso all'ADC. Come ogni sensore, questo dà in uscita un segnale analogico che deve essere digitalizzato.
- Per svolgere l'esercizio i passi da seguire sono i seguenti:
 - Prima di iniziare a programmare dobbiamo capire come configurare l'ADC facendo uso dei manuali:
 - Lo **user manual sulle API del HAL** è utile per capire quali funzioni dobbiamo utilizzare per inizializzare l'ADC.
 - Dobbiamo capire, in seguito, a quale ADC e a quale linea il sensore di temperatura è connesso. Vedere il **reference manual**

Esercizio: Leggere La Temperatura Interna della Scheda 2/4

- In seguito dobbiamo:
 - Chiaramente abilitare il clock
 - Abilitare le linee di interrupt connesse all'ADC
 - Configurare l'ADC
 - Configurare il channel relativo al Temperature Sensor
 - Definire l' Handler che deve gestire le interrupts e la relativa funzione di Callback
 - All'interno della callback leggere il valore in uscita dal convertitore
 - Infine, effettuare la conversione del dato grezzo nel valore di temperatura applicando la caratteristica lineare del sensore

Esercizio: Leggere La Temperatura Interna della Scheda 3/4

How to use this driver

1. Initialize the ADC low level resources by implementing the HAL_ADC_MspInit():
 - a. Enable the ADC interface clock using `__HAL_RCC_ADC_CLK_ENABLE()`
 - b. ADC pins configuration
 - Enable the clock for the ADC GPIOs using the following function:
`HAL_RCC_GPIOx_CLK_ENABLE()`
 - **Configuration of ADC, groups regular/injected, channels parameters**
 - c. In
 -
 - 1. Configure the ADC parameters (resolution, data alignment, ...) and regular group parameters (conversion trigger, sequencer, ...) using function `HAL_ADC_Init()`.
 - d. In
 -
 - 2. Configure the channels for regular group parameters (channel number, channel rank into sequencer, ..., into regular group) using function `HAL_ADC_ConfigChannel()`.
 - 3. Optionally, configure the injected group parameters (conversion trigger, sequencer, ..., of injected group) and the channels for injected group parameters (channel number, channel rank into sequencer, ..., into injected group) using function `HAL_ADCEx_InjectedConfigChannel()`.
 - 4. Optionally, configure the analog watchdog parameters (channels monitored, thresholds, ...) using function `HAL_ADC_AnalogWDGConfig()`.
 - 5. Optionally, for devices with several ADC instances: configure the multimode parameters using function `HAL_ADCEx_MultiModeConfigChannel()`.

Esercizio: Leggere La Temperatura Interna della Scheda 4/4

Reading the temperature

To use the sensor:

3. Select ADC1_IN16 or ADC1_IN18 input channel.
4. Select a sampling time greater than the minimum sampling time specified in the datasheet.
5. Set the TSVREFE bit in the ADC_CCR register to wake up the temperature sensor from power down mode
6. Start the ADC conversion by setting the SWSTART bit (or by external trigger)
7. Read the resulting V_{SENSE} data in the ADC data register
8. Calculate the temperature using the following formula:

$$\text{Temperature (in } ^\circ\text{C)} = \{(V_{SENSE} - V_{25}) / \text{Avg_Slope}\} + 25$$

Where:

- $V_{25} = V_{SENSE}$ value for 25°C
- Avg_Slope = average slope of the temperature vs. V_{SENSE} curve (given in $\text{mV}/^\circ\text{C}$ or $\mu\text{V}/^\circ\text{C}$)

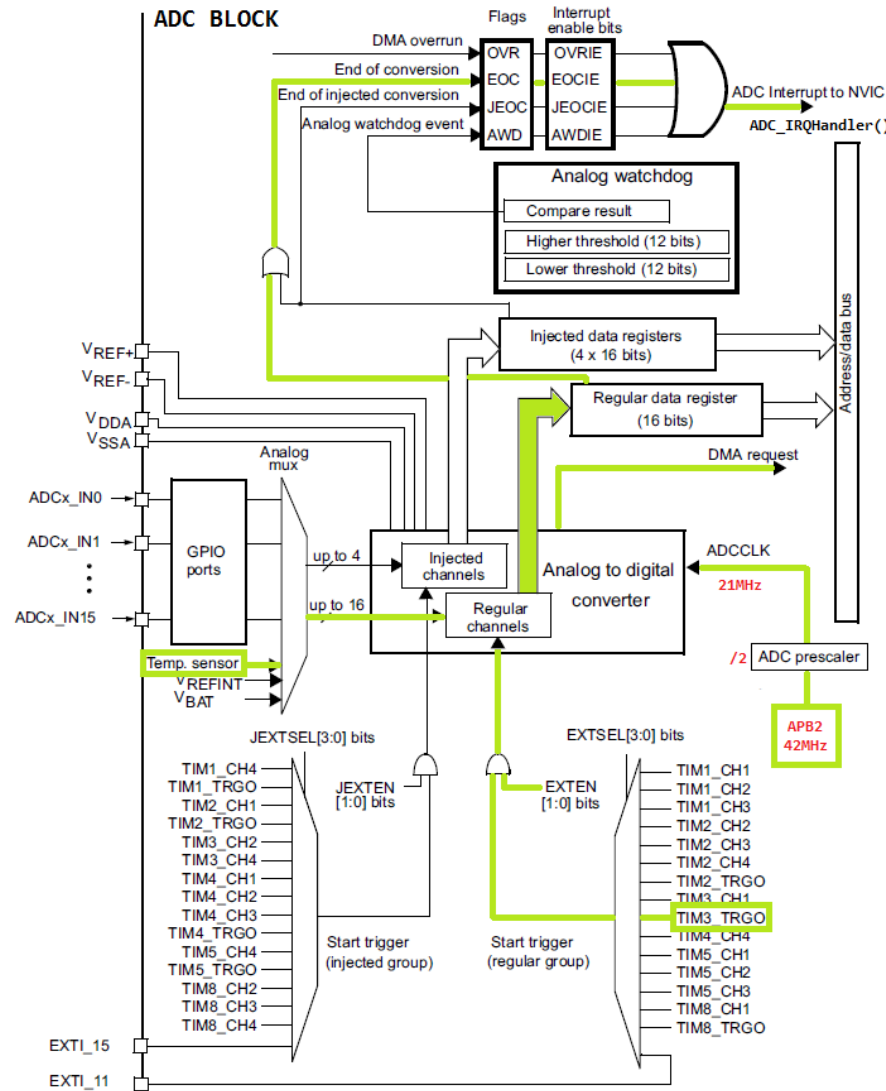
Refer to the datasheet's electrical characteristics section for the actual values of V_{25} and Avg_Slope.

~~The ...~~

Conversione del Valore letto dall'ADC

- Il dato letto in uscita dall'ADC è proporzionale al riferimento che si sta usando. Come visto prima sarà un valore compreso tra 0 e 2^N-1 dove N è la risoluzione dell'ADC.
- Per ottenere il valore in V bisogna moltiplicare per la tensione di riferimento V_{ref} (nel nostro caso 3.3V) e dividere per la risoluzione.
 - $Result_V = (V_{ref} / 2^N-1) * Result\ from\ ADC$
- Una volta ottenuto il valore di tensione sarà possibile applicare la caratteristica lineare del sensore per ottenere il valore di temperatura in C°

Schema ADC



This is what hardware is doing in the diagram:

- Temp. sensor is ON
- ADC is ON
- TIM3 triggers at 2kHz
- ADC converts the voltage and stores it in the regular data register
- ADC makes a DMA request
- DMA moves data to memory
- ADC raises EOC flag which generates an ADC interrupt
- ADC IRQ handler is called at 2kHz toggling a pin at 1kHz (indirectly testing my calculations)

Software will do the following:

- Calibrating the sensor
- Averaging N ADC samples (VSENSE is the average value)
- Calculating the absolute temperature
- Calculating temperature variations

$$\text{Temp (in } ^\circ\text{C)} = \frac{(\text{VSENSE} - \text{V25})}{\text{Avg_Slope}} + 25$$

- (The rest is just for fun)
- Setting some threshold temperatures
 - If the measured temp. is below/above the threshold values, some LEDs light up :)

Digital-to-Analog Converter (DAC)

- Il DAC, al contrario dell'ADC, genera un segnale analogico a partire da una parola digitale.
- Può essere utilizzato in svariate applicazioni, come ad esempio per la generazione di segnali audio a partire dalla sequenza di bit del file audio che si vuole riprodurre.