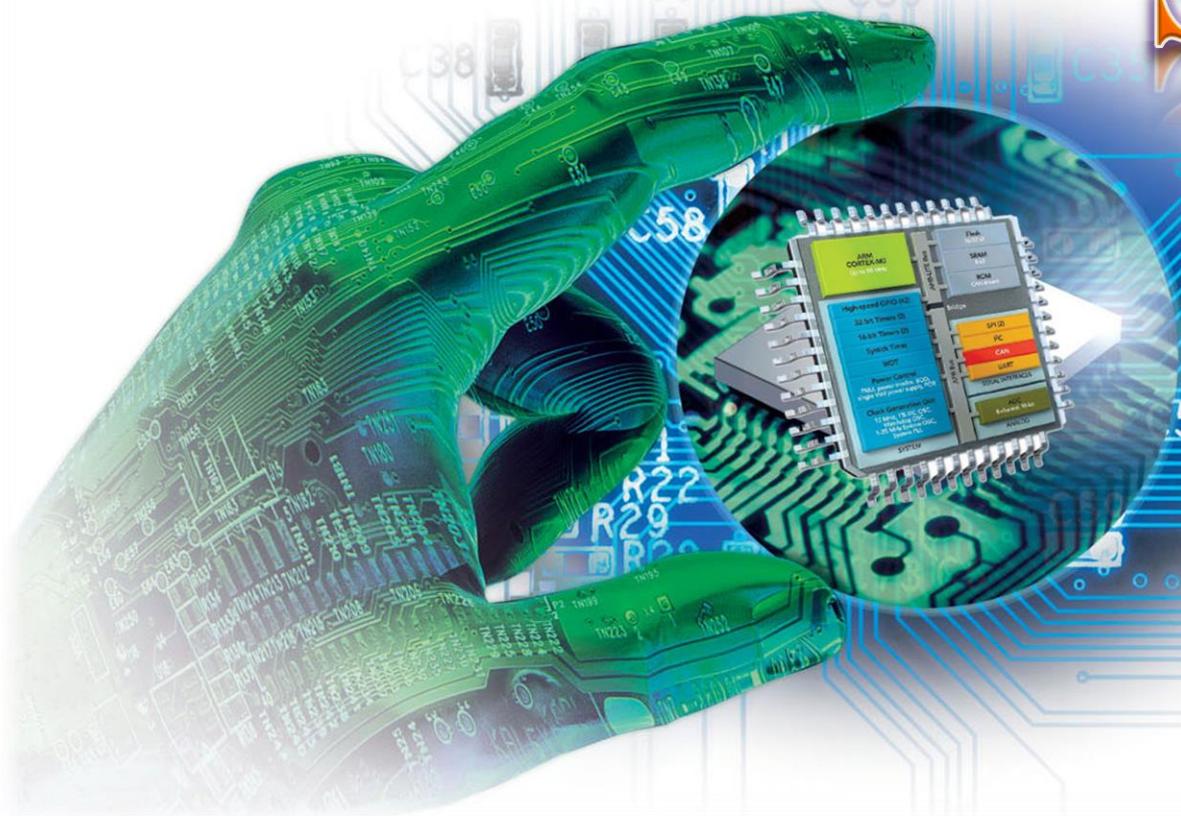


مُتَحَكِّمَات

STM32

6

5



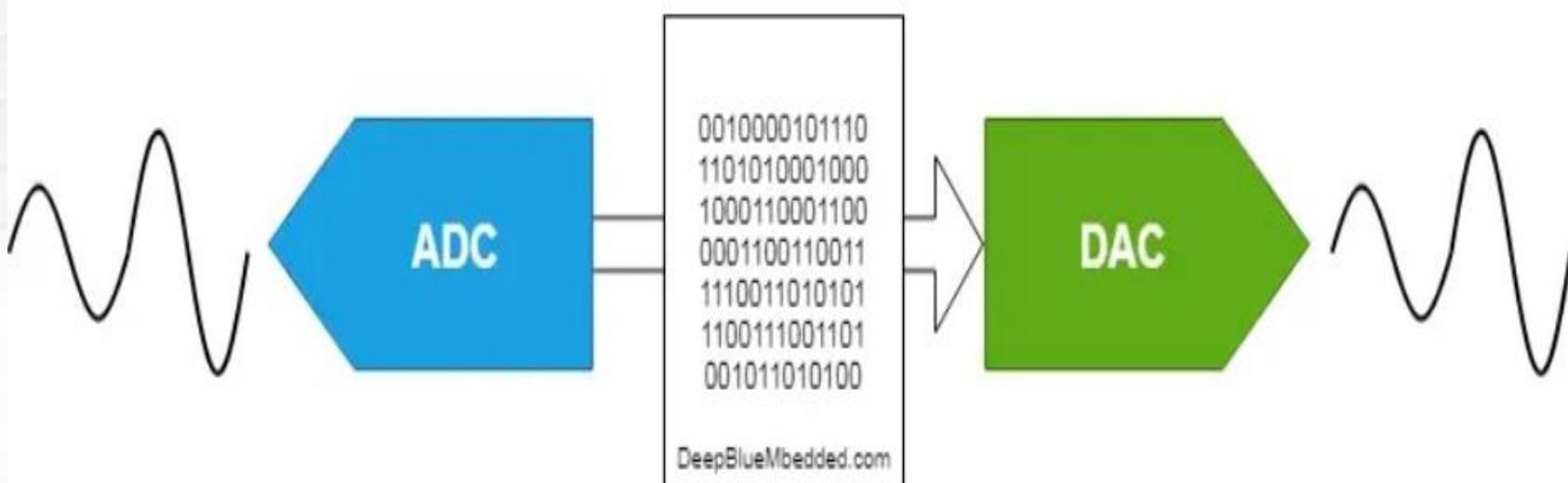
موضع عاًن المعاشرة:

- المبدلات التشابهية الرقمية في متحكمات STM32
- أنماط عمليات التحويل الرقمي modes
- طرق قراءة المبدل التشابهي الرقمي
- الأخطاء الناتجة عن التحويل
- التطبيق العملي

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

المبدلات التشابهية الرقمية عبارة عن دارات الكترونية تقوم بتحويل الجهد التشابهي على دخلها إلى قيمة رقمية بالنظام الثنائي مقابلة لمستوى الجهد، فبمجرد قذح المبدل التشابهي الرقمي يبدأ بأخذ العينات samples ويقوم بعملية تدعى التكميم ليقابل كل مستوى من الجهد بما يناسبه من القيم الرقمية

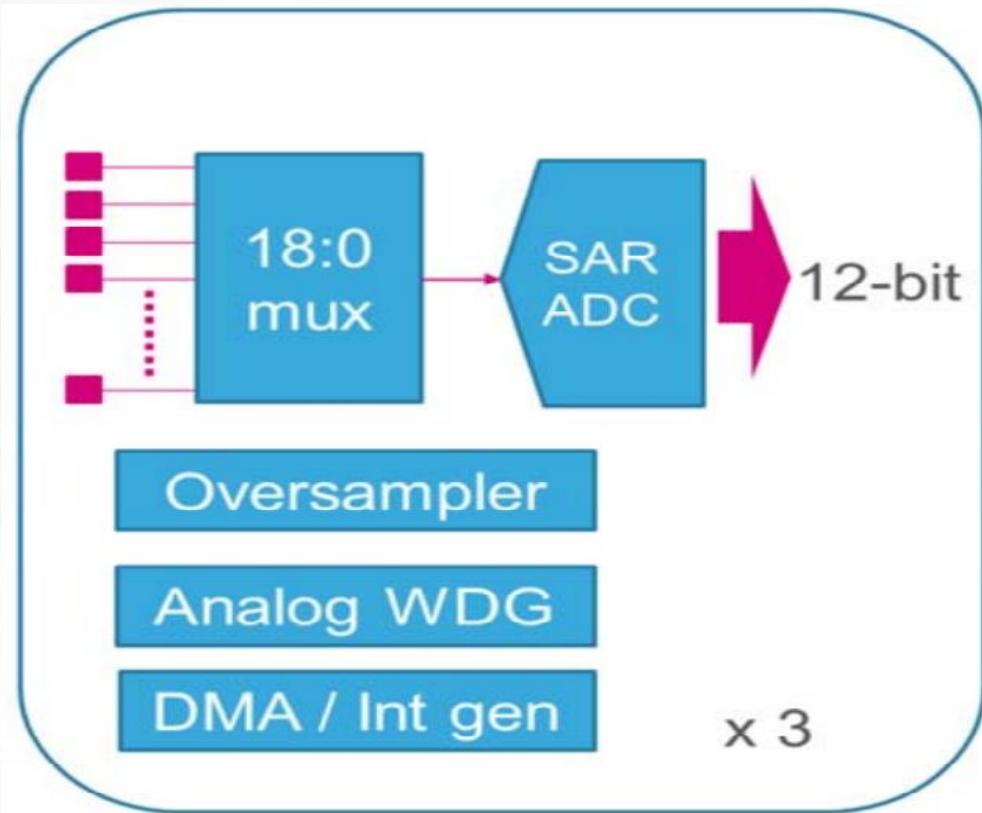


المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

تحتوي متحكمات STM32G0 على مبدل تشابهي رقمي وحيد من نوع Successive approximation ADC(SAR)

التالي:



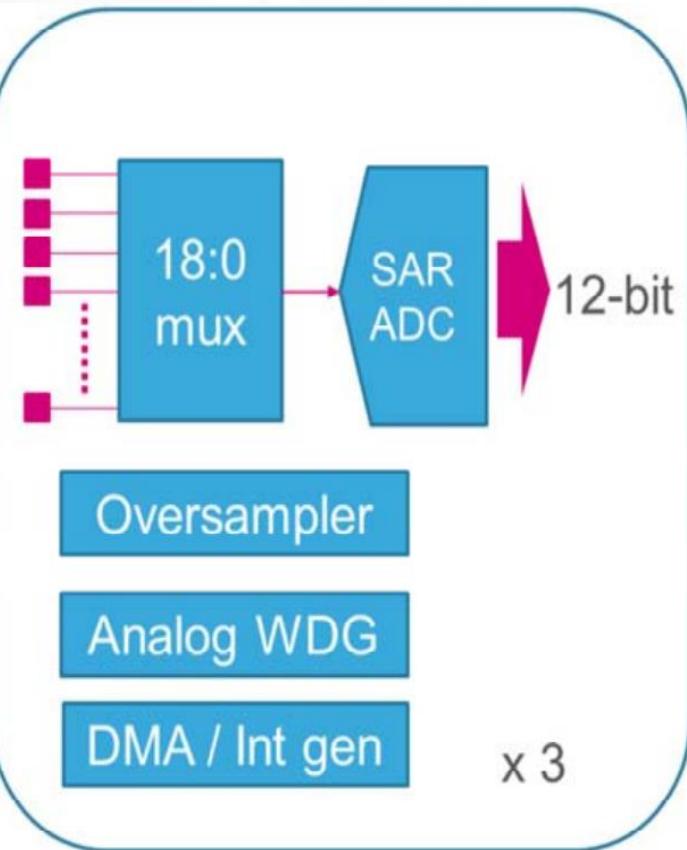
- مبدل ADC وحيد بدقة 12 بت وما يقارب الـ 19 قناة للمبدل
- Oversampler
- بحد أقصى لعملية أخذ العينات يصل إلى Msamples/s 2.5

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

تحتوي متحكمات STM32G0 على مبدل تشابهي رقمي وحيد من نوع Successive approximation ADC(SAR)

التالي:



لكل مبدل تشابهي رقمي ثلاثة

مراقبات تشابهية

analog watchdogs

لمراقبة حالات الـ

thresholds Oversampler

القدرة على توليد طلبات DMA

القدرة على توليد مقاطعة interrupt

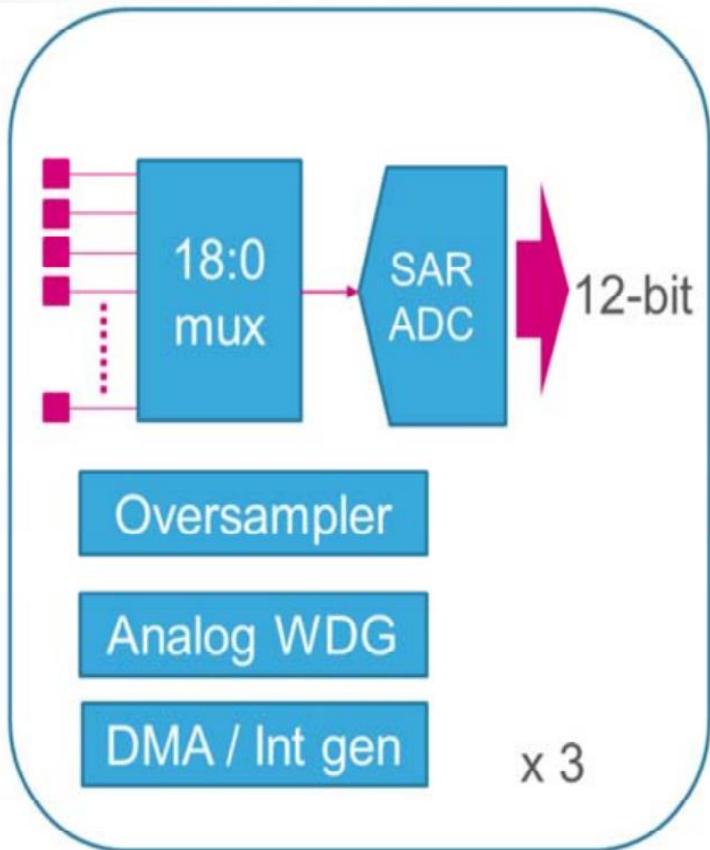
سحب منخفض للتيار 1181 Msamples/s

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

تحتوي متحكمات STM32G0 على مبدل تشابهي رقمي وحيد من نوع Successive approximation ADC(SAR)

التالي:



- قابل للقذح بعدة طرق
- القدرة على إدارة البيانات القادمة ومن ثم تحويلها للأ. CPU.
- تسمح المبدلات التشابهية الرقمية للمتحكم STM32G0 باستقبال القيم التشابهية القادمة من الحساسات، حيث تقوم بتحويلها إلى القيم الرقمية المقابلة لها

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

يوضح الجدول التالي أهم الخصائص لـ ADC الموجود في متحكمات STM32G0

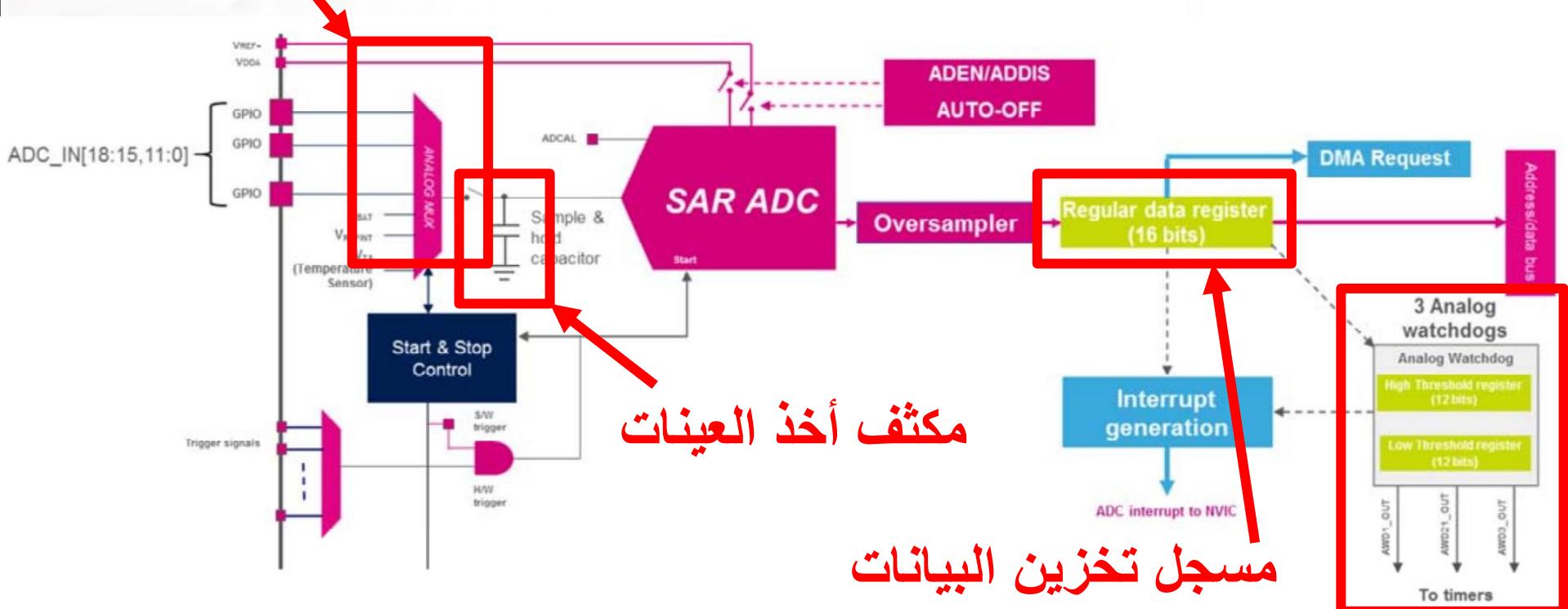
Features	Description
Input channel	Up to 16 external (GPIOs) and 3 internal channels
Type of conversion	12-bit successive approximation
Conversion time	400 ns, 2.5 Msamples/s (when $f_{ADC_CLK} = 35\text{ MHz}$, 12 bits)
Functional mode	Single, Continuous, Scan, and Discontinuous
Triggers	Software or external trigger (Timers & IOs)
Special functions	Analog watchdogs, Hardware oversampling, and Self-calibration
Data processing	Interrupt generation and DMA requests
Low-power modes	Wait, Auto-off, and Power-down

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

للمبدل التشابهي الرقمي في متحكمات STM32G0 المخطط الصندوقي
التالي:

multiplexer



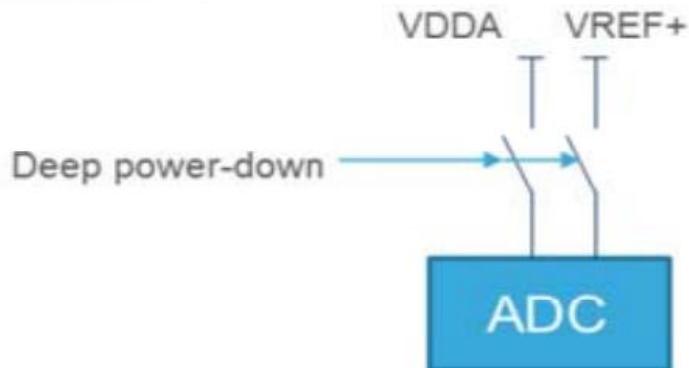
مكثف أخذ العينات

مسجل تخزين البيانات

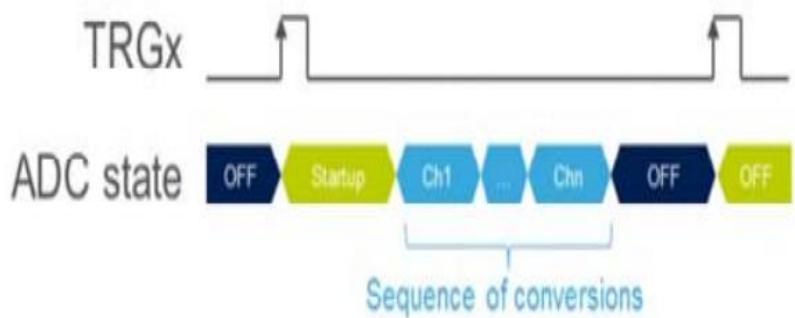
مراقبات تشابهية

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32



لاحظ في الجزء الأمامي لعملية التحويل يوجد مفتاح فصل/وصل الطاقة عن المبدل حيث يمكن فصل المبدل عن الطاقة عند عدم استخدامه



للمبدل ميزة إدارة الطاقة والتي تقوم بفصل الطاقة آلياً عن المبدل عند عدم وجود عملية تحويل ويتم إيقاظه أيضاً آلياً عند بدء عملية التحويل من خلال قدحه عبر الكود أو من خلال الهاردويير ، وذلك عند تفعيل نمط Auto-off mode، ويتم إضافة زمن startup time تلقائياً بين لحظة البدء وبدء عملية التحويل

المبدلات التشابهية الرقمية في متحكمات STM32

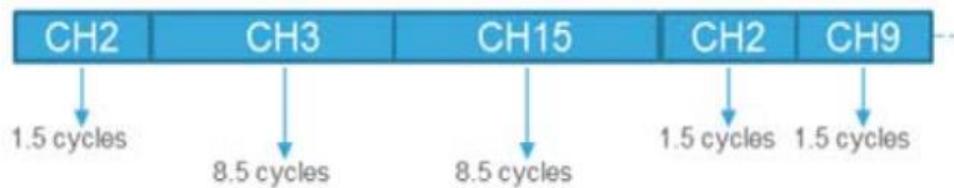
ADC in STM32

يحتوي المبدل على وحدة أخذ عينات oversampling hardware والتي تقوم بتجميع العينات ثم تقوم بإعادة تقسيمها دون مساعدة إلى CPU، حيث بإمكانها تجميع من 2 حتى 256 عينة ثم إزاحتها نحو اليمين، مما يسمح للمستخدم

باستخدام:



$$Result = \frac{1}{M} * \sum_{n=0}^{n=N-1} Conversion(t_n)$$



العينات على قيمتين و تخصيص القيمة المناسبة لكل قناة من هاتين

□ يمكن ضبط زمن أخذ

□ 19 قناة تحويل وترتيب تصاعدي أو تنازلي

□ 8 قنوات للتحويل بالترتيب

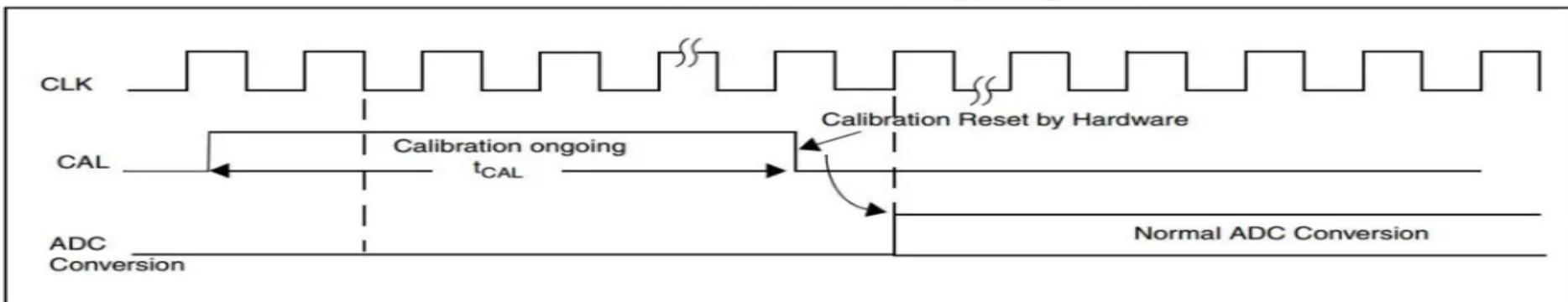
الذى يقوم المستخدم بتعريفه

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

المعايرة الذاتية:

يوفر المبدل خاصية المعايرة الذاتية والتي تقلل بشكل كبير الأخطاء الناتجة عن تغيرات مكثف الشحن الداخلي internal capacitor يوضح المخطط الزمني التالي المعايرة الذاتية:



توفر مكتبة HAL دالة ضمن الـ ADC APIs تقوم ببدء عملية المعايرة والذي يوصى بعملها في بداية الكود بعد تهيئة الـ ADC عند تشغيل النظام

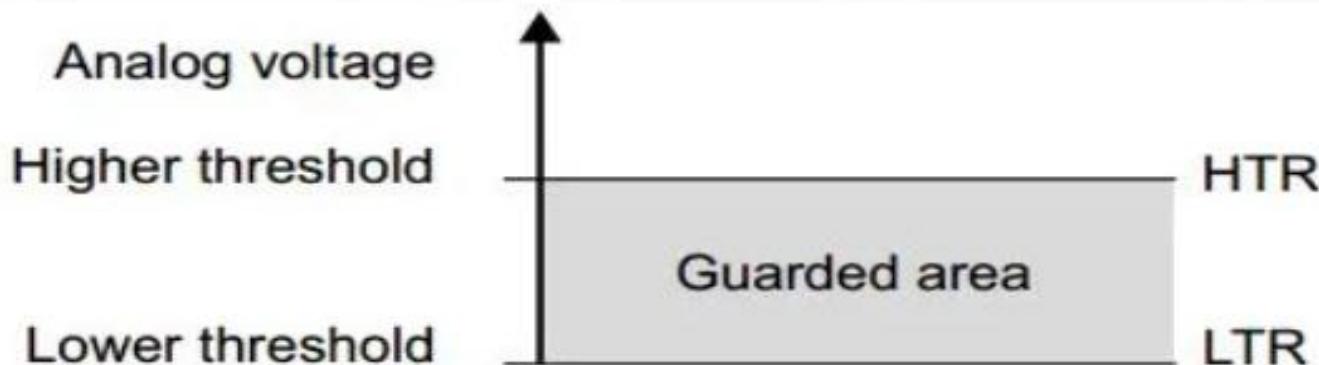
المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

المراقب الشابهي (AWD):

عندما يكون جهد الدخل التشابهي القادر على إحدى القنوات التشابهية أقل من العتبة الدنيا أو أكبر من العتبة العليا المسموح بها عندها يتم تفعيل بit الحالة الخاصة بالمراقب الشابهي AWD status bit

Analog watchdog guarded area



المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

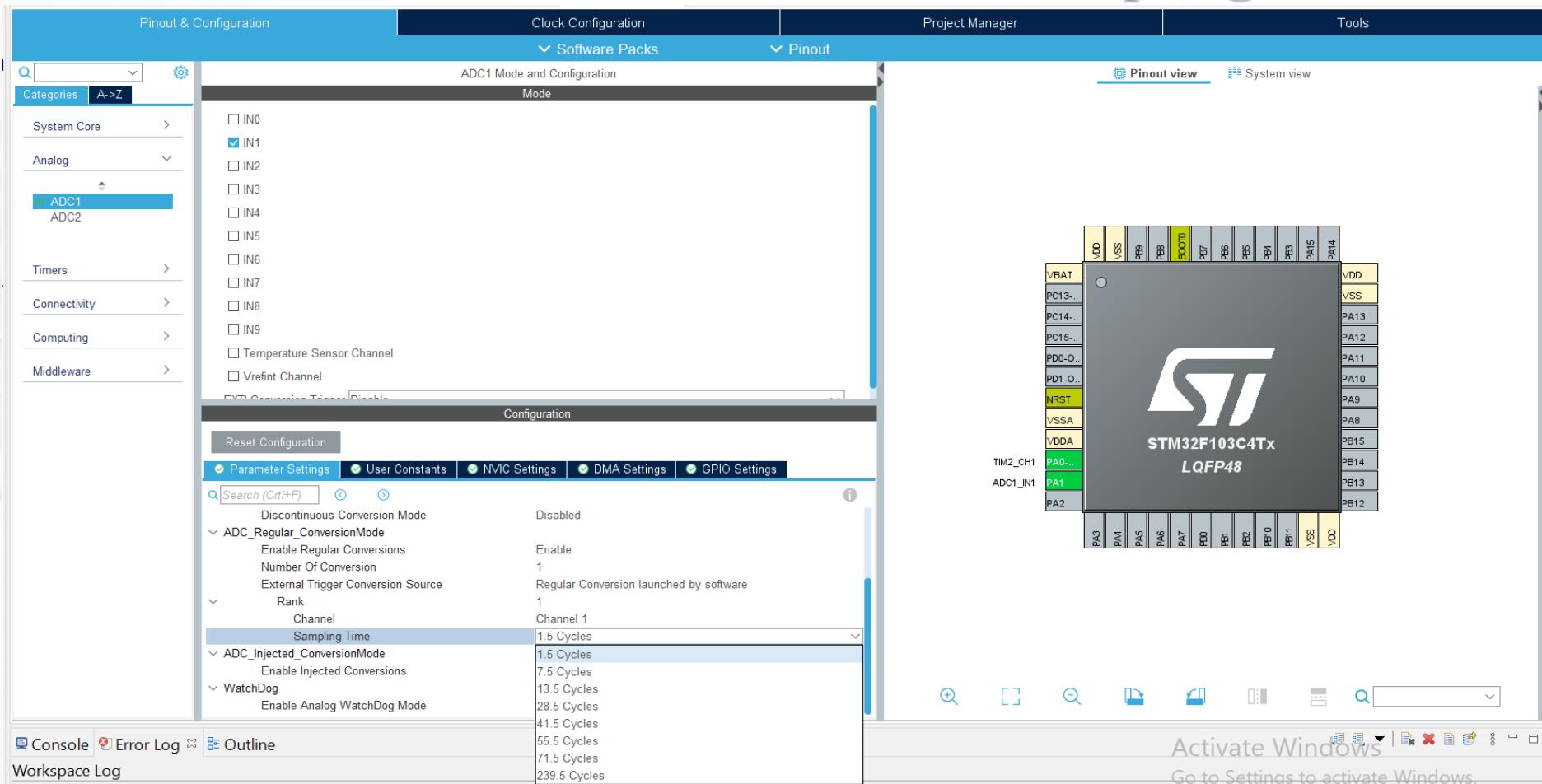
:Sampling times

يتم أخذ عينات الجهد التشابهي خلال عدد معين من دورات الساعة، حيث أزمنة أخذ العينات المتوفرة 1.5cycles ، 3.5cycles ، 7.5cycles ، 12.5cycles ، 19.5cycles ، 39.5cycles ، 79.5cycles ، 160.5cycles و يتم تخزينها في المسجل SMP[2:0] bits و استخدامها لكل قناة من القنوات التشابهية، حيث يمكن أن يكون لكل قناة تشابهية معدل أخذ عينات مختلف.

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

:Sampling times 



The screenshot shows the STM32CubeMX software interface. On the left, the navigation tree highlights the 'ADC1' and 'ADC2' nodes under the 'Analog' category. The main workspace displays the 'Pinout & Configuration' tab, specifically the 'ADC1 Mode and Configuration' section. In this section, the 'IN1' checkbox is selected, indicating it is configured as an analog input. The 'Sampling Time' dropdown is set to '1.5 Cycles'. To the right of the configuration window, a detailed pinout diagram for the STM32F103C4TX LQFP48 package is shown, with pins labeled from PA0 to PA14 and PB0 to PB15. The VDD and VSS pins are also indicated.

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

:Sampling times

يمكن حساب الزمن الكلي لعملية التحويل من خلال العلاقة التالية:

$$T_{conv} = Sampling\ time + 12.5\ cycles$$

مثال: من أجل ADCCLK=14MHZ، و زمن أخذ عينات يكون الزمن الكلي للتحويل: 1.5 cycles

$$T_{conv} = 1.5 + 12.5\ cycles = 14cycles = 1\mu sec$$

وبالتالي يكون تردد أخذ العينات:

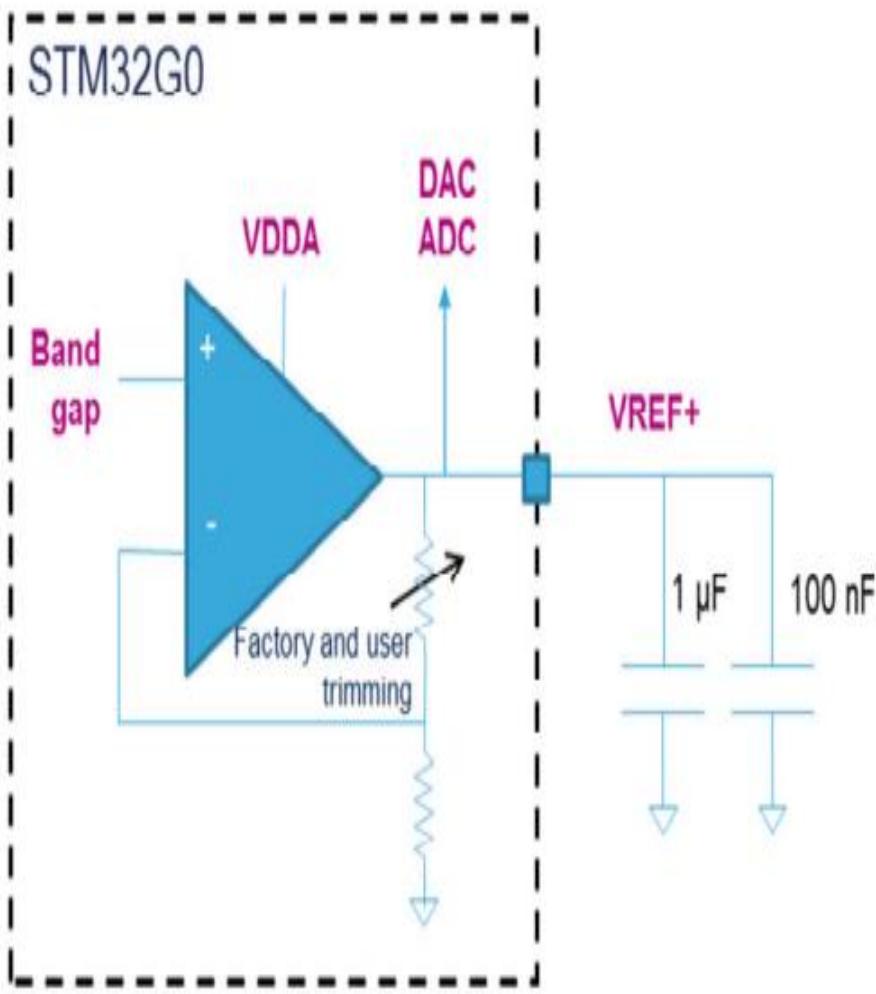
$$Sampling\ rate = 1/T_{conv}$$

ومن أجل المثال السابق يكون تردد أخذ العينات:

$$Sampling\ rate = 1000000 = 1Ms/ses$$

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32



□ **الجهد المرجعي للمبدل التشابهي:**
يوجد داخل متحكمات STM32 مولد جهد مرجعي مدمج بداخلها يقوم بتوليد جهد مرجعي ثابت ومستقر حتى عند تغذيته من بطارية و يمكن استخدامه مع المبدل التشابهي الرقمي ADC والمبدل الرقمي التشابهي DAC، و يعطي في خرجه قيمتين إما 2.5 V أو 2.048 V، كما يمكنه أن يغذي أحمال خارجية باستجرار تيار لا يتجاوز الـ 4 mA

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

□ الجهد المرجعي للمبدل التشابهي:

يمكن ضبط الجهد المرجعي للمبدل التشابهي الرقمي ليكون خارجي أو داخلي من خلال البات **ENVR** ، **HIZ** bits الموجودين في المسجل و على المستخدم أن ينتظر حتى يتم تفعيل البت **VREFBUF_CSR** والذي يعني أن الجهد المرجعي الخارجي قد وصل إلى القيمة المطلوبة **VRR**

ENVR	HIZ	Configuration
0	0	VREF buffer OFF VREF+ pin pulled-down to VSSA
0	1	External voltage reference mode (default): <ul style="list-style-type: none">• VREF buffer OFF• VREF+ pin floating
1	0	Internal voltage reference mode: <ul style="list-style-type: none">• VREF buffer ON• VREF+ pin connected to the VREF buffer output
1	1	Hold mode: <ul style="list-style-type: none">• VREF buffer ON• VREF+ pin floating. The voltage is held with an external capacitor

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

□ الجهد المرجعي للمبدل التشابهي:

عند استخدام مولد الجهد المرجعي الداخلي يجب وصل مكثفات على القطب V_{ref+} وفي هذه الحالة لن تحتاج لوصل دارة خارجية لتوليد الجهد المرجعي، كما يدعم مولد الجهد المرجعي الداخلي قيمتي جهد هي:

- $V_{DDA} \geq 2.4V$: وهذا يتطلب أن يكون الجهد على الطرف

- $V_{DDA} \geq 2.48V$: وهذا يتطلب أن يكون الجهد على الطرف

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

الجهد المرجعي للمبدل التشابهي:

يتم تعريف أقطاب الجهد المرجعي للمتحكم من خلال الـ Datasheet ويفترض أن يتم توصيلها بمجالات معينة من الجهد كما هو موضح بالجدول التالي:

Name	Signal type	Remarks
V_{REF+}	Input, analog reference positive	The higher/positive reference voltage for the ADC, $2.4 \text{ V} \leq V_{REF+} \leq V_{DDA}$
$V_{DDA}^{(1)}$	Input, analog supply	Analog power supply equal to V_{DD} and $2.4 \text{ V} \leq V_{DDA} \leq 3.6 \text{ V}$
V_{REF-}	Input, analog reference negative	The lower/negative reference voltage for the ADC, $V_{REF-} = V_{SSA}$
$V_{SSA}^{(1)}$	Input, analog supply ground	Ground for analog power supply equal to V_{ss}

المبدلات التشابهية الرقمية في متحكمات STM32

ADC in STM32

الجهد المرجعي للمبدل التشابهي:

يتم حساب جهد الدخل التشابهي الناتج عن عملية التحويل من خلال العلاقة التالية:

$$V_{in} = \text{ADC_Res} * (\text{reference voltage} / 4096)$$

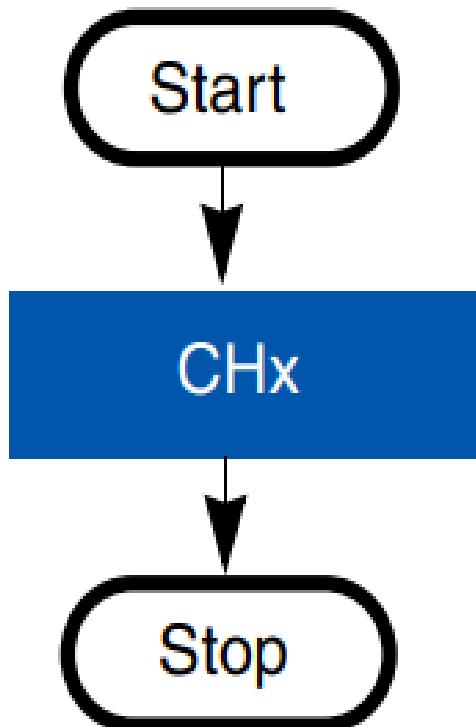
حيث :

$$\text{reference voltage} = (V_{ref+}) - (V_{ref-})$$

أنماط عمليات التحويل ADC conversion modes

يمكن للمبدل التشابهـي الرقمـي أن يعـمل بعـدة أنـماط عـمل هي :

□ **Single-channel, single conversion mode**: وهو نـمط العـمل الأـسـهل والأـبـسـط مـقارـنةً مع بـقـية الـأـنـماـط، حيث يـقـوم الـADC بـعـملـيـة تحـوـيل وـحـيدـة (وـعـملـيـة أـخـذ عـيـنـات وـحـيدـة) ولـقـاتـة وـاحـدـة فـقـط ويـتـوقف عن الـعـمل بـمـجـرـد اـنـتـهـائـه من عـملـيـة التـحـوـيل

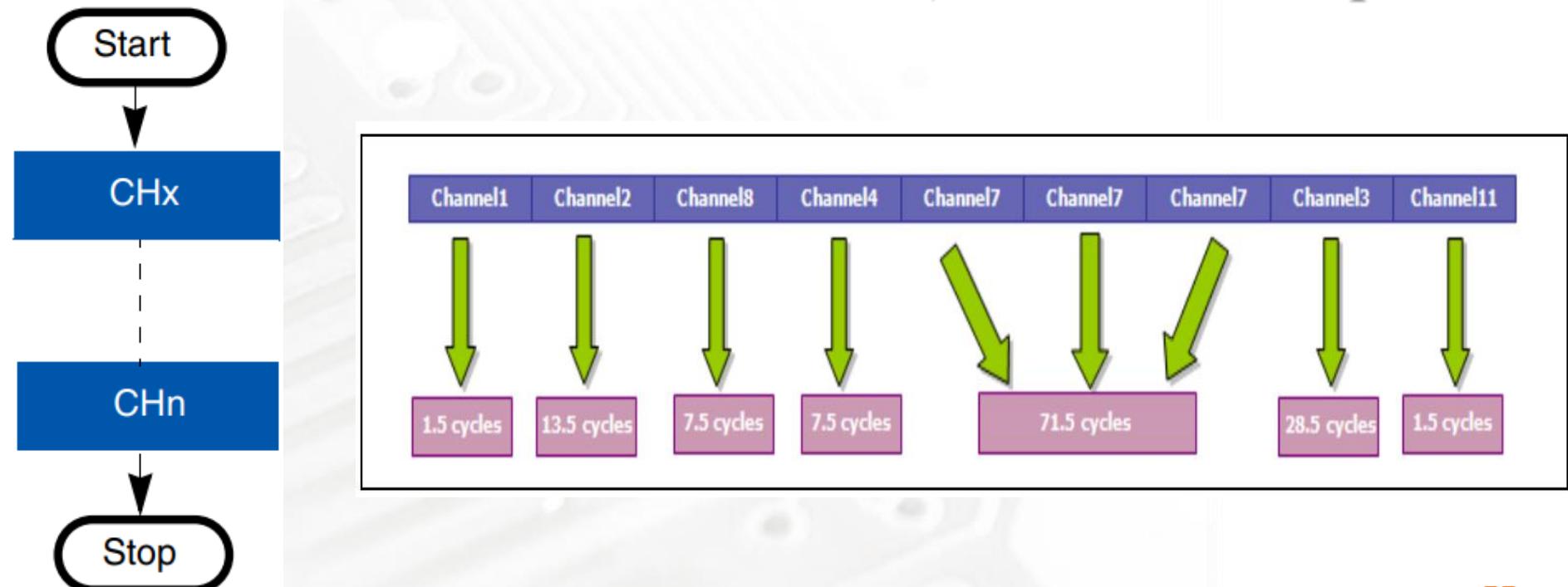


أنماط عمليات التحويل ADC conversion modes

يمكن للمبدل التشابهى الرقمي أن يعمل بعدة أنماط عمل هي :

:Multichannel (scan) single conversion mode

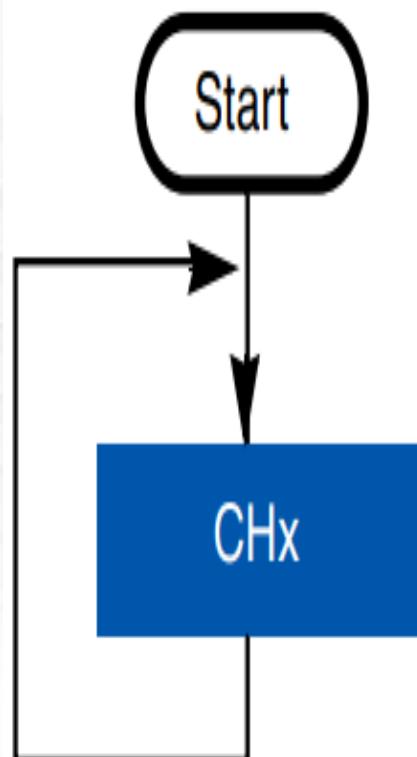
يستخدم هذا النمط من العمل لتحويل الجهود القادمة من عدة قنوات تشابهية ولمرة واحدة فقط، ويمكن اختيار ترتيب القنوات الذي ترغب به من خلال و باستخدام زمن أخذ عينات مختلف لكل قناة من القنوات **ADC sequencer**



أنماط عمليات التحويل ADC conversion modes

يمكن للمبدل التشابهـي الرقمـي أن يعـمل بعـدة أنـماط عـمل هـي :

□ **Single-channel continuous conversion mode**: في هذا النـمـط من العـمل يـتم تحـويل الـقيـمة التـشاـبـهـيـة الـقادـمـة عـلـى قـنـاة وـاحـدة إـلـى قـيـمة رـقـمـيـة وـتـعـاد عـمـلـيـة التـحـولـيـل بـشـكـل مـسـتـمر

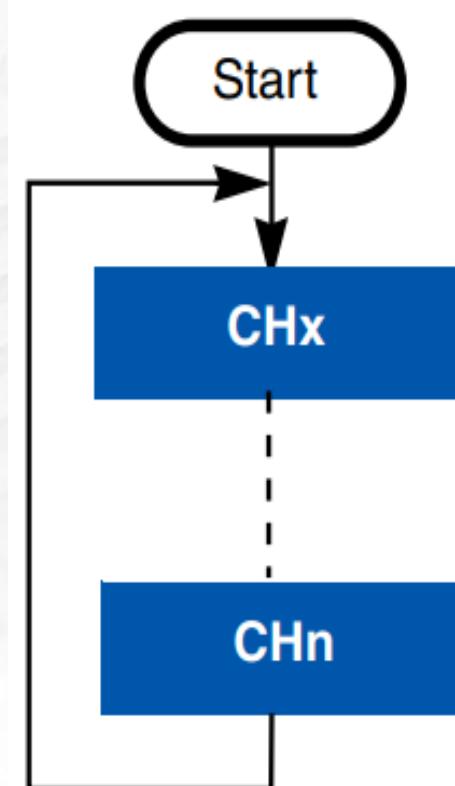


أنماط عمليات التحويل ADC conversion modes

يمكن للمبدل التشابهـي الرقمـي أن يعـمل بعـدة أنـماط عـمل هـي :

:Multichannel (scan) , continuous conversion mode

يـستخدم هـذا النـمط مـن العـمل لـتحويل الجـهود الـقادمة مـن عـدة قـنوات تـشابهـية إـلـى قـيم رـقمـية وـيعـيد العمـلـية بـشـكـل مـسـتـمر وـذـلـك تـبعـاً لـتـرتـيـب مـعـين لـلـقـنـوات



طرق قراءة المبدل التشابهي الرقمي ADC conversion modes

يوجد ثلاث طرق رئيسية لقراءة الـ ADC هي:

Polling Method: تعتبر الطريقة الأسهل في كتابة الكود لقراءة القيمة القادمة من إحدى القنوات التشابهية، ولكنها ليست الأكثر فعالية ، حيث علينا أن نبدأ بعملية التحويل وتتوقف الـ CPU عن تنفيذ الكود وتنظر لحين الانتهاء من عملية التحويل حينها يمكن للـ CPU استكمال تنفيذ الكود الرئيسي

The interrupt Method: تعتبر هذه الطريقة طريقة فعالة لاستخدام المبدل التشابهي الرقمي ، حيث يقوم بقدر المبدل فقط و يمكن للـ CPU أن تستكمل تنفيذ الكود والمهام المطلوبة منها لحين انتهاء عملية التحويل عنها سيقوم المبدل بطلب مقاطعة وستتوجه الـ CPU لبرنامج خدمة المقاطعة

DMA Method: وهي الطريقة الأكثر فعالية في استخدام المبدل التشابهي الرقمي

الأخطاء الناتجة عن التحويل

قد تنتج عن عملية التحويل التشابهي عدة أخطاء سنذكر نوعين منها هما الأكثر شيوعاً:

وهو الخطأ الناتج بين أول عملية تحويل فعلية و أول عملية تحويل متوقعة ، حيث تتم أول عملية تحويل عندما يتغير خرج المبدل من 0 إلى 1 ، يتم ذلك بالحالة المثالية عندما يتغير الدخل التشابهي بين الـ 0.5 LSB والـ 1.5 LSB عددها يكون الخرج الرقمي 1 منطقي، ويتم الإشارة إلى خطأ الإزاحة ب E_o ويمكن معايرته من خلال الكود

Positive offset error representation

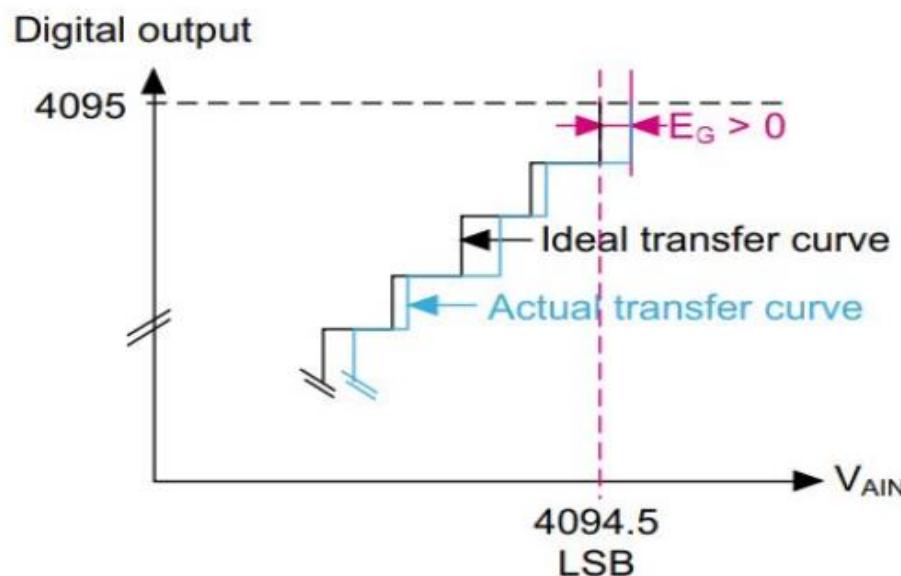


الأخطاء الناتجة عن التحويل

قد تنتج عن عملية التحويل التشابهى عدة أخطاء سنذكر نوعين منها هما الأكثر شيوعاً:

خطأ الربح Gain error : وهو الانحراف بين آخر تحويل فعلى وأخر تحويل مثالي ويعبر عنه بـ E_G حيث يتم آخر تحويل فعلى عند الانتقال بين 0xFFFF والـ 0xFFE

Positive gain error representation



دوال مكتبة HAL المستخدمة للتعامل مع المبدل التشابهي الرقمي في وضع الـ Polling

- لبدء المعايرة الذاتية للمبدل التشابهي الرقمي :adc1

HAL_ADCEx_Calibration_Start(&hadc1);

- إعطاء إشارة البدء للمبدل التشابهي الرقمي ليكون جاهز فيما بعد لإجراء عمليات التحويل

HAL_ADC_Start(&hadc1);

- لطلب عملية تحويل من المبدل التشابهي الرقمي

HAL_ADC_PollForConversion(&hadc1, 1);

- إسناد ناتج عملية التحويل إلى المتغير **AD_RES**

AD_RES = HAL_ADC_GetValue(&hadc1);

دوال مكتبة HAL المستخدمة للتعامل مع المبدل التشابهي الرقمي في وضع الـ Interrupt

- لبدء المعايرة الذاتية للمبدل التشابهي الرقمي :adc1

HAL_ADCEx_Calibration_Start(&hadc1);

- إعطاء إشارة البدء للمبدل التشابهي الرقمي ليكون جاهز فيما بعد لإجراء عمليات التحويل

HAL_ADC_Start_IT(&hadc1);

- إسناد ناتج عملية التحويل إلى المتغير AD_RES

AD_RES = HAL_ADC_GetValue(&hadc1);

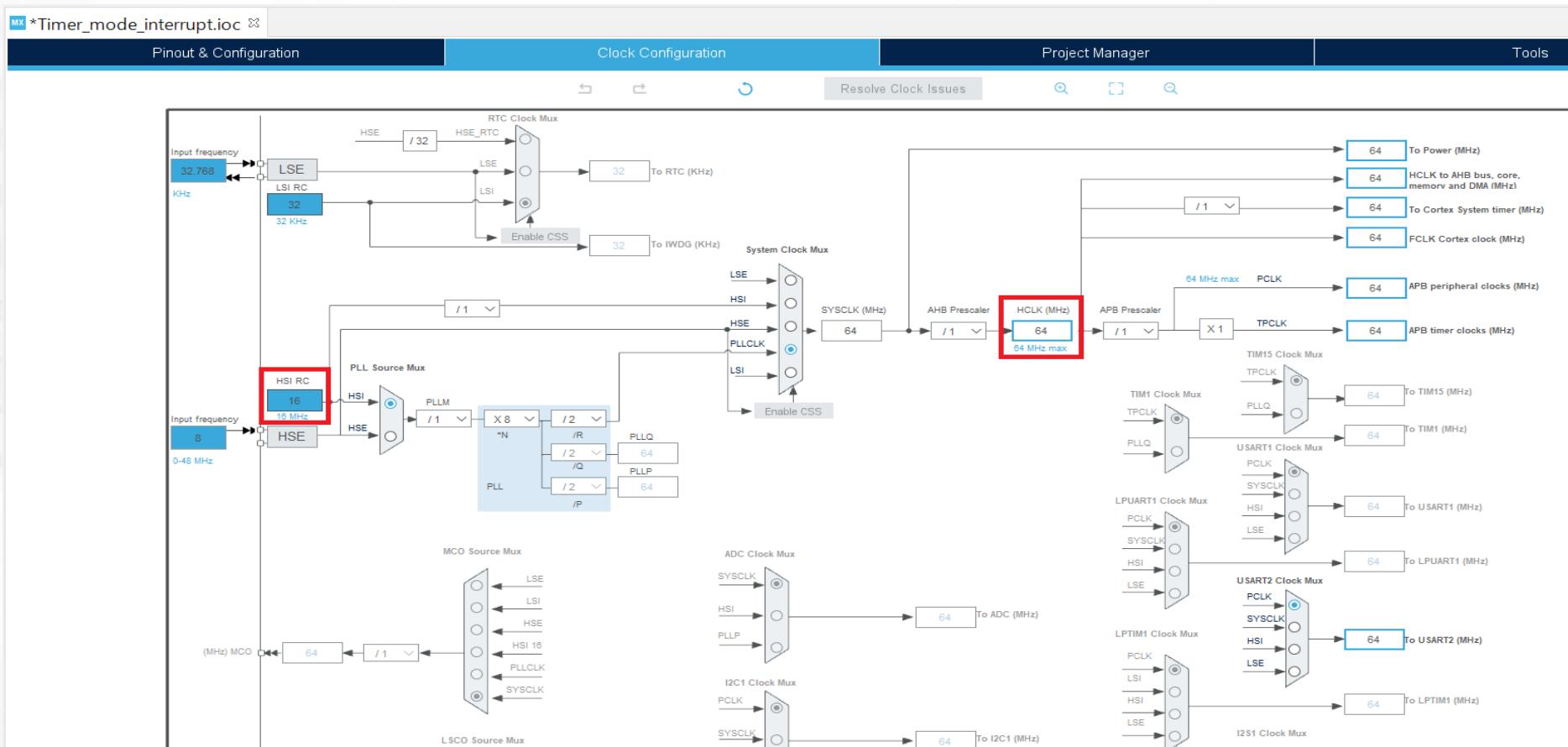
التطبيق 1: التحكم بشده إصاءه ليد موصول على أحد أقطاب الـ PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل التشابهي باستخدام نمط Polling

سنقوم بتنفيذ المشروع وفقاً للتسلسل التالي:

- ضبط تردد ساعة المتحكم
- ضبط قطب الدخل التشابهي CH7 في نمط التحويل لمرة واحدة Single حيث سرّبط مقاومة متغيرة معه.
- ضبط الـ timer2 في نمط PWM على القناة CH1 وسرّبط معه ليد.

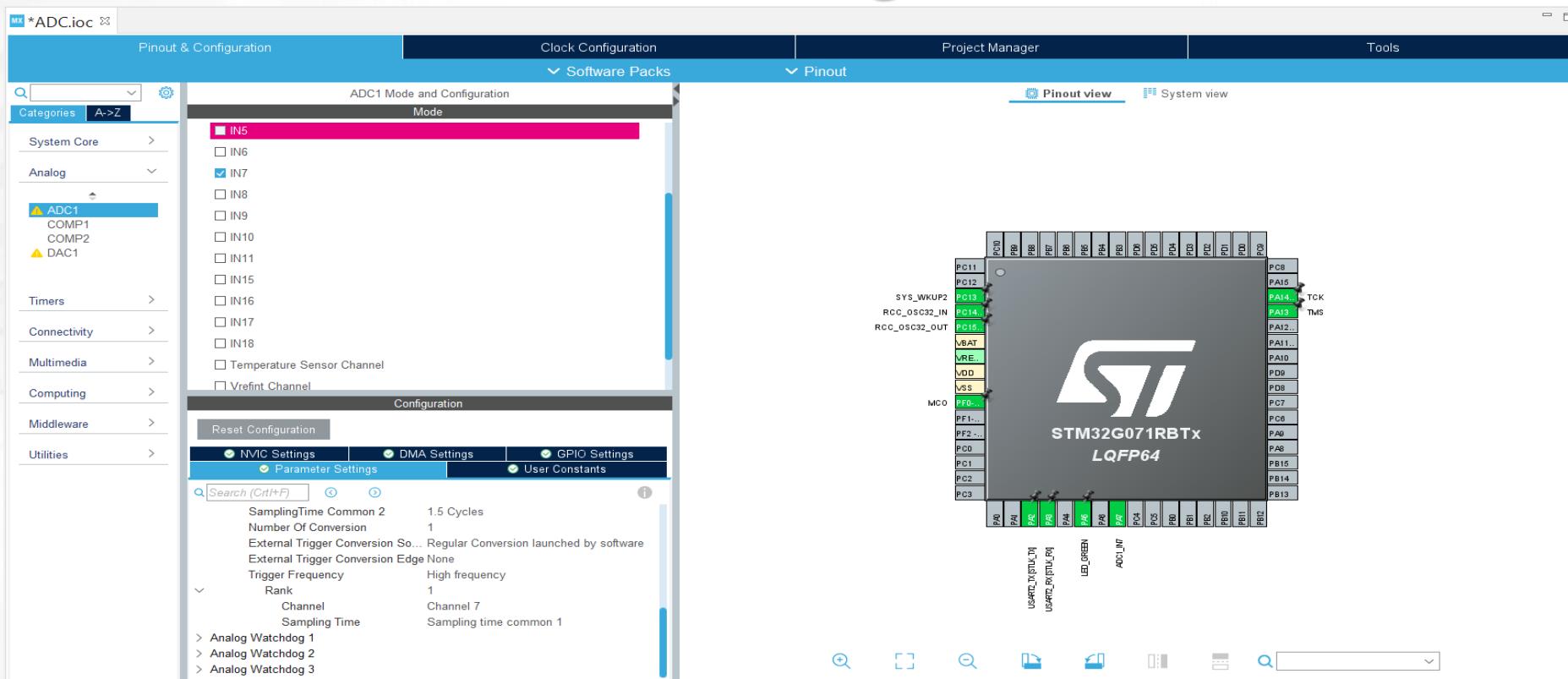
التطبيق 1: التحكم بشدة إصاءه ليد موصول على أحد أقطاب الـ PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل التشابهي باستخدام نمط Polling

ضبط تردد ساعة المتحكم -



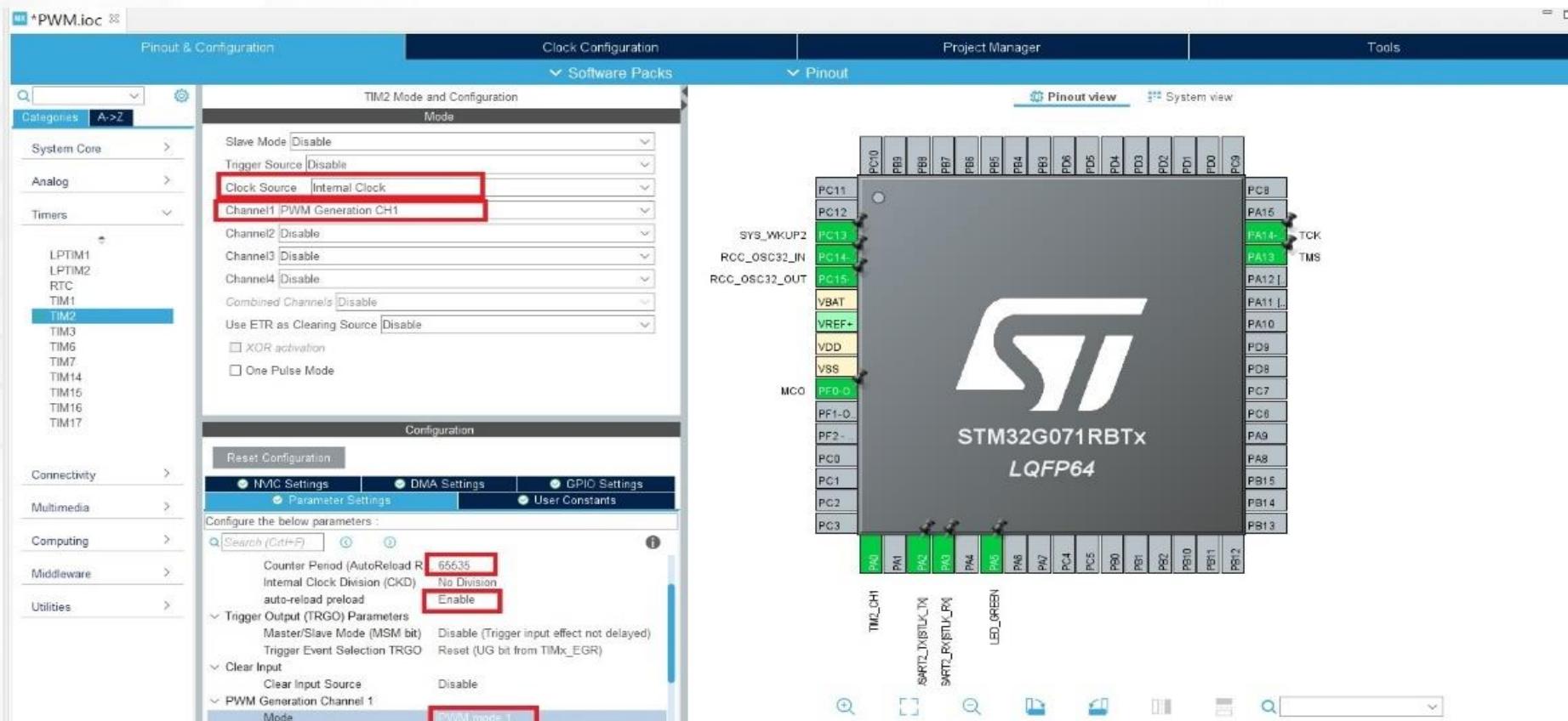
التطبيق 1: التحكم بشده إصاءه ليد موصول على أحد اقطاب الـ PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل الشابهي باستخدام نمط Polling

- ضبط قطب الدخل الشابهي CH7 في نمط التحويل لمرة واحدة حيث سرّبط مقاومة متغيرة معه.



التطبيق 1: التحكم بـ شدّه إصاعيّة لـ LED موصول على أحد أقطاب الـ PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل الشابهي باستخدام نمط Polling

- ضبط الـ timer2 في نمط PWM على القناة CH1 وسربط معه LED.



التطبيق 1: التحكم بـشدة إصاءه لـيد موصول على أحد أقطاب آلة PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل التشابهي باستخدام نمط Polling

```
#include "main.h"
```

```
ADC_HandleTypeDef hadc1;
```

```
TIM_HandleTypeDef htim2;
```

```
void SystemClock_Config(void);
```

```
static void MX_GPIO_Init(void);
```

```
static void MX_ADC1_Init(void);
```

```
static void MX_TIM2_Init(void);
```

التطبيق 1: التحكم بـشدة إصاءه لـيد موصول على أحد أقطاب آلة PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل التشابهي باستخدام نمط Polling

```
int main(void)
{
    uint16_t AD_RES = 0;

    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_ADC1_Init();
    MX_TIM2_Init();
```

التطبيق 1: التحكم بـشدة إصاءه لـيد موصول على أحد أقطاب آلة PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل التشابهي باستخدام نمط Polling

```
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);  
// Calibrate The ADC On Power-Up For Better  
Accuracy
```

```
HAL_ADCEx_Calibration_Start(&hadc1);  
while (1)  
{  
    // Start ADC Conversion  
    HAL_ADC_Start(&hadc1);  
    // Poll ADC1 Peripheral & TimeOut = 1mSec  
    HAL_ADC_PollForConversion(&hadc1, 1);
```

التطبيق 1: التحكم بـشدة إصاءه لـيد موصول على أحد أقطاب آلة PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل التشابهي باستخدام نمط Polling

```
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);  
// Read The ADC Conversion Result & Map It To PWM  
DutyCycle
```

```
AD_RES = HAL_ADC_GetValue(&hadc1);
```

```
TIM2->CCR1 = (AD_RES<<4);
```

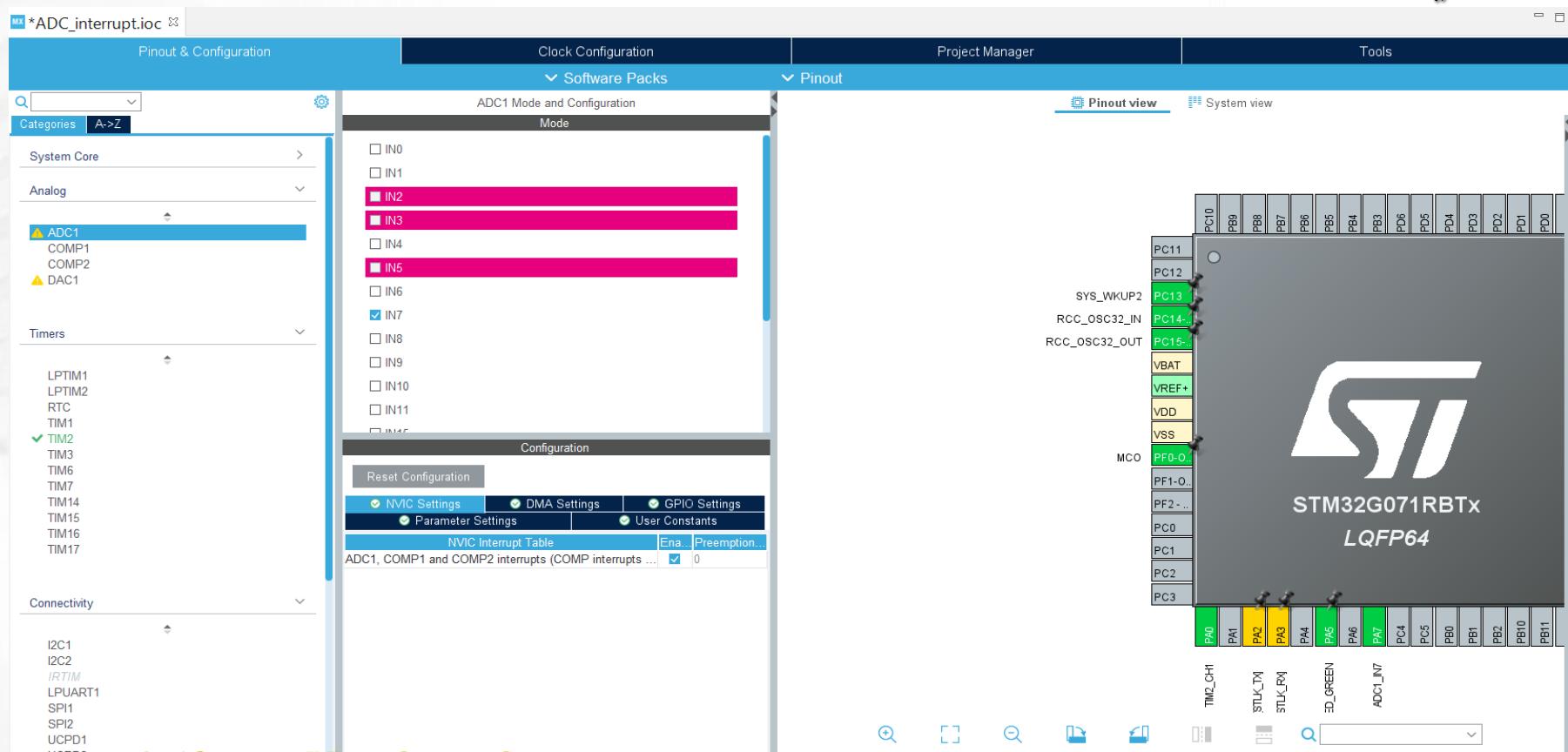
```
HAL_Delay(1);
```

```
}
```

```
}
```

التطبيق 2: التحكم بشده إصاءه ليد موصول على أحد اقطاب الـ PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل الشابهي باستخدام نمط Interrupt

سنقوم بإعادة الإعدادات السابقة بالإضافة إلى تفعيل مقاطعة المبدل الشابهي الرقمي



التطبيق 2: التحكم بـشدة إصاءه لـPin موصول على أحد أقطاب AD من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل PWM التشابهي باستخدام نمط Interrupt

```
#include "main.h"

uint16_t AD_RES = 0;
ADC_HandleTypeDef hadc1;
TIM_HandleTypeDef htim2;
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);
static void MX_TIM2_Init(void);
```

التطبيق 2: التحكم بـشدة إصاءه لـLED موصول على أحد أقطاب AD من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل PWM التشابهي باستخدام نمط Interrupt

```
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_ADC1_Init();
    MX_TIM2_Init();
    HAL_TIM_PWM_Start(&htim2,
TIM_CHANNEL_1);
```

التطبيق 2: التحكم بـشدة إصاءه لـيد موصول على أحد أقطاب آلة PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل الشابهي باستخدام نمط Interrupt

```
HAL_ADCEx_Calibration_Start(&hadc1);  
while (1)  
{  
    // Start ADC Conversion  
    HAL_ADC_Start_IT(&hadc1);  
    // Update The PWM Duty Cycle With Latest ADC  
    // Conversion Result  
    TIM2->CCR1 = (AD_RES<<4);  
    HAL_Delay(1);  
}  
}
```

التطبيق 2: التحكم بـشدة إصاءه لـيد موصول على أحد أقطاب الـ
PWM من خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل
التشابهي باستخدام نمط Interrupt

```
Void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef*  
hadc)  
{  
    // Read & Update The ADC Result  
    AD_RES = HAL_ADC_GetValue(&hadc1);  
}
```

Thank you for listening