

المبدلات التشابيهية الرقمية في متحكمات STM32

ADC in STM32

محتويات الجلسة:

- 1- مقدمة
- 2- المبدلات التشابيهية الرقمية في متحكمات STM32
- 3- أنماط عمليات التحويل ADC conversion modes
- 4- طرق قراءة المبدل التشابيهي الرقمي
- 5- الأخطاء الناتجة عن التحويل
- 6- التطبيق العملي.

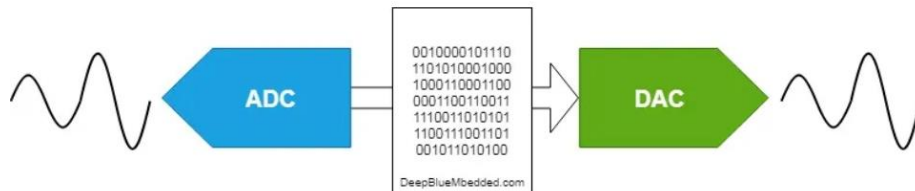
الأدوات اللازمة للجلسة:

- لوحة Nucleo-64bit
- كبل Type-A to Mini-B
- ليدات
- مفاتيح لحظية
- مقاومة متغيرة

مستوى الجلسة: متوسط

1. مقدمة:

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

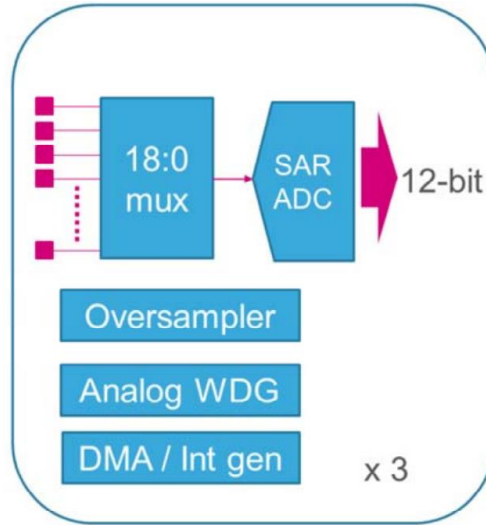


الشكل (1): المبدلات التشابيهية الرقمية

تعتبر المبدلات التشابيهية الرقمية من العناصر الالكترونية المكلفة مادياً خصوصاً عندما يكون لها معدل أخذ عينات عالي ودقة عالية، لذا يعتبر وجود الـ ADC في المتحكم المصغر من الإضافات الهامة والمفيدة جداً.

2. المبدلات التشابهيّة الرقمية في متحكمات STM32: Analog to digital converter(ADC):
تحتوي متحكمات STM32G0 على مبدل تشابهي رقمي وحيد من نوع Successive approximation (SAR) له المواصفات التالي:

- مبدل ADC وحيد بدقة 12 بت وما يقارب الـ 19 قناة للمبدل
- Oversampler
- بحد أعظمي لعملية أخذ العينات يصل إلى 2.5 Msamples/s
- لكل مبدل تشابهي رقمي ثلاث analog watchdog لمراقبة حالات الـ thresholds
- القدرة على توليد طلبات DMA
- القدرة على توليد مقاطعة interrupt
- سحب منخفض للتيار 18 Msamples/s @ 1 uA
- قابل للقدح بعدة طرق
- القدرة على إدارة البيانات القادمة ومن ثم تحويلها للـ CPU.



الشكل (2): المبدلات التشابهيّة الرقمية في متحكمات STM32

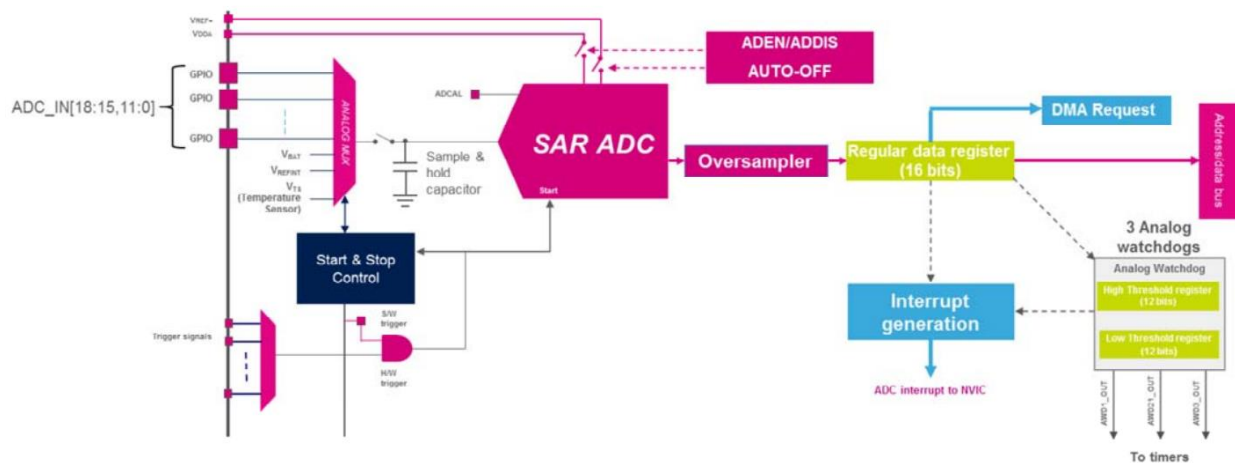
تسمح المبدلات التشابهيّة الرقمية للمتحكم STM32G0 باستقبال القيم التشابهيّة القادمة من الحساسات، حيث تقوم بتحويلها إلى القيم الرقمية المقابلة لها، فللمبدل ما يقارب الـ 19 قناة تحويل أي 19 دخل تشابهي للمتحكم بإمكانه استقبال القيم التشابهيّة من خلالها.

المبدل الموجود داخل المتحكم هو من نوع 12-bit successive approximation converter (SAR ADC) مع وجود وحدة أخذ عينات oversampling hardware إضافية تقوم بمعالجة البيانات مسبقاً للتخفيف عن المعالج الرئيسي، معدل أخذ العينات هو 2.5 ميغا عينة في الثانية وبدقة 12 بت وبإمكانها توليد طلبات مقاطعة أو DMA، حيث تم تصميم المبدل التشابهي الرقمي في المتحكم ليكون أدائه عالي مع سحب منخفض للتيار كي يكون مناسباً للتطبيقات منخفضة الطاقة ويكون قابل للقدح بعدة طرق وبحيث يتم إدارة البيانات كي يتم التخفيف عن الـ CPU، يوضح الجدول التالي أهم الخصائص للـ 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

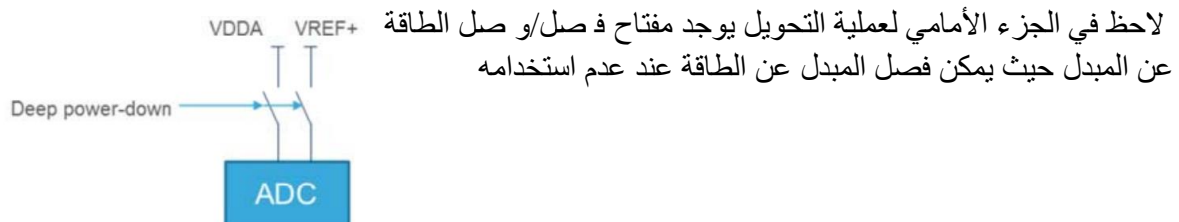
الجدول (1): أهم الخصائص للـ ADC الموجود في متحكمات STM32G0

للمبدل التشابهي الرقمي في متحكمات STM32G0 المخطط الصندوقي التالي:

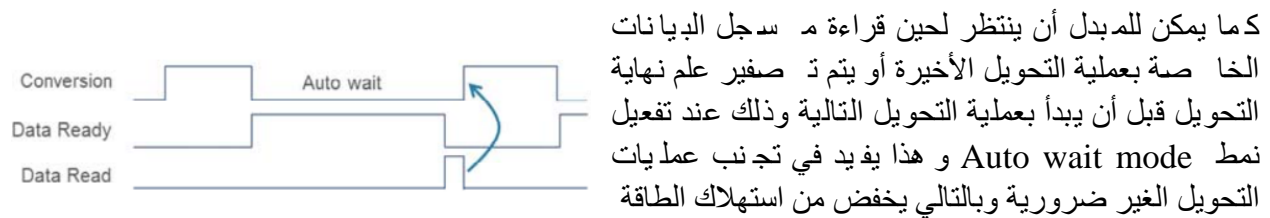


الشكل (3): المخطط الصندوقي للمبدل التشابهي الرقمي في متحكمات STM32G0

من المخطط الـ صندوقي الـ سابق نلاحظ أن الجزء الأيسر منه يحتوي الواجهة الأمامية لعملية التحويل والتي تشمل الناخب multiplexer ومكثف أخذ العينات، حيث يجب أن يكون زمن أخذ العينات كافي كي يتم شحن المكثف بقيمة جهد الدخل ، أما الجزء الأيمن من المخطط فيحتوي على الجزء الخلفي من عملية التحويل، حيث يتم تخزين العينات في م سجل البيانات data register والذي يمكن قراءته من خلال الكود أو يتم ترحيله إلى الذاكرة من خلال قناة الـ DMA، كما نلاحظ وجود ثلاث مراقبات تشابهية analog watchdog مهمتها مراقبة جهد الدخل التشابهي مع الحدود الدنيا والعليا المسموحة له ، وفي حال تجاوزه للحدود المسموحة يتم توليد مقاطعة وقذح وحدة المؤقت.

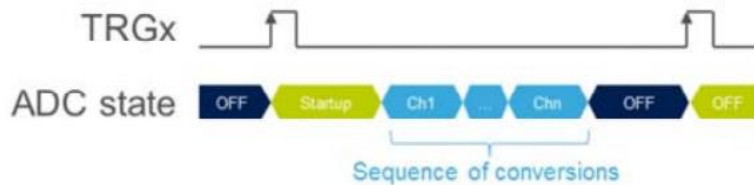


الشكل (4): مفتاح فصل/وصل الطاقة عن المبدل



الشكل(5):Auto wait mode

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



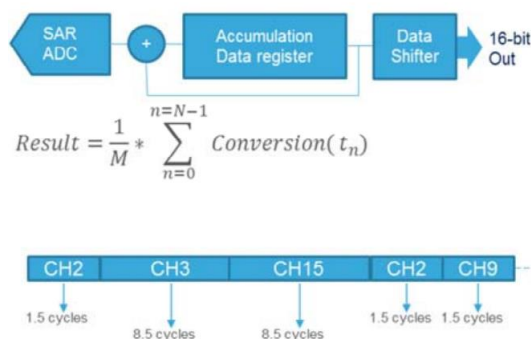
الشكل(6):Auto-off mode

كما يمكن دمج النمط auto wait mode مع النمط auto off mode حيث يتم فصل المبدل عن الطاقة خلال زمن الانتظار وإعادة وصله مرة أخرى عند قراءة م سجل البيانات ADC_DR ، مما يؤدي إلى تخفيض ملحوظ في استهلاك الطاقة خصوصاً مع التطبيقات ذات معدل أخذ العينات المنخفض نسبياً.

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

- 19 قناة تحويل وبترتيب تصاعدي أو تنازلي
- 8 قنوات للتحويل بالترتيب الذي يقوم المستخدم بتعريفه

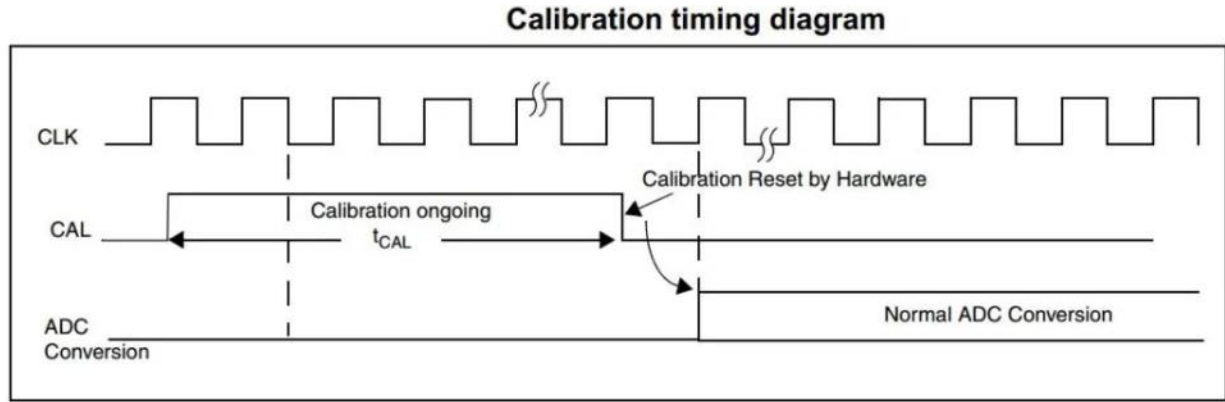
كما يمكن ضبط زمن أخذ العينات على قيمتين و تخصيص القيمة المناسبة لكل قناة من هاتين القيمتين.



الشكل(7):oversampling hardware

المعايرة الذاتية Self-calibration:

يوفر المبدل خا صية المعايرة الذاتية والتي تقلل ب شكل كبير الأخطاء الناتجة عن تغيرات مكثف ال شحن الداخلي internal capacitor، يوضح المخطط الزمني التالي المعايرة الذاتية:



الشكل(8):المخطط الزمني للمعايرة

توفر مكتبة HAL دالة ضمن ال - ADC APIs تقوم ببء عملية المعايرة والذي يو صى بعملها في بداية الكود بعد تهيئة ال- ADC عند تشغيل النظام.

المراقب التشابهي (AWD) The ADC Analog Watchdog:

عندما يكون جهد الدخل الـ تشابهي القادم على إحدى القنوات الـ تشابهية أقل من العتبة الدنيا أو أكبر من العتبة العليا المـ سموح بها عندها يتم تفعيل بت الحالة الخاص بالمراقب الـ تشابهي AWD status bit ، حيق يتم برمجة القيم الخاصة بالعتبات الدنيا و العليا من المسجلات ADC_HTR ، ADC_LTR ، ويمكن تفعيل المقاطعة باستخدام البت AWDIE bit الموجود في المسجل ADC_CR1.



الشكل(9):analog watchdog

سرعة التحويل Conversion speed:

يحتاج المبدل الـ تشابهي الرقمي على الأقل 1.5clock cycles لأخذ العينات و 12.5 clock cycles للتحويل من أجل دقة 12بت، بمعنى آخر ومن أجل تردد ال ساعة الأعظمي للمبدل 35MHZ يمكن أن ت صل سرعة أخذ العينات إلى 2.5mega samples/s ، ومن أجل الوصول إلى معدل أخذ عينات أعلى ممكن تخفيض الدقة إلى 10 أو 8 أو 6 بت كما هو موضح بالجدول التالي:

Resolution	$t_{\text{Conversion}}$
12 bits	12.5 Cycles
10 bits	10.5 Cycles
8 bits	8.5 Cycles
6 bits	6.5 Cycles

جدول(2): معدل أخذ العينات تبعاً لدقة التحويل

فمن أجل:

- دقة 12 بت: تحتاج عملية التحويل إلى 12.5Cycles ويكون معدل أخذ العينات 2.5Msamples/s
- دقة 10 بت: تحتاج عملية التحويل إلى 10.5Cycles ويكون معدل أخذ العينات 2.92Msamples/s
- دقة 8 بت: تحتاج عملية التحويل إلى 8.5Cycles ويكون معدل أخذ العينات 3.5Msamples/s
- دقة 6.5 بت: تحتاج عملية التحويل إلى 6.5Cycles ويكون معدل أخذ العينات 4.37Msamples/s

: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 ، ويمكن للمستخدم اختيار قيمتين من هذه القيم من خلال البتات bits SMP[2:0] ويتم تخزينها في المسجل ADC_SMPR واستخدامها لكل قناة من القنوات التشابهية، حيث يمكن أن يكون لكل قناة تشابهية معدل أخذ عينات مختلف.

يمكن حساب الزمن الكلي لعملية التحويل من خلال العلاقة التالية:

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

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

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

وبالتالي يكون تردد أخذ العينات:

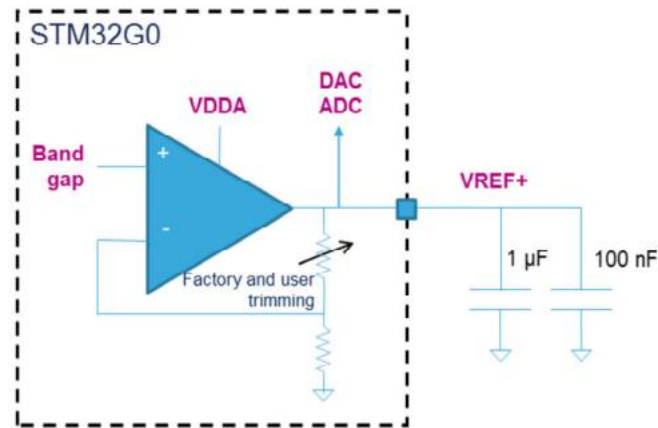
$$\text{Sampling rate} = 1/T_{\text{conv}}$$

ومن أجل المثال السابق يكون تردد أخذ العينات:

$$\text{Sampling rate} = 1000000 = 1 \text{ Ms/s}$$

الجهد المرجعي للمبدل التشابهي:

يوجد داخل متحكمات STM32 مولد جهد مرجعي مدمج بداخلها يقوم بتوليد جهد مرجعي ثابت ومستقر حتى عند تغذيته من بطارية ويمكن استخدامه مع المبدل التشابهي الرقمي ADC والمبدل الرقمي التشابهي DAC ، ويعطي في خرج قيمتين إما 2.5V أو 2.048V ، كما يمكنه أن يغذي أحمال خارجية باستقرار تيار لا يتجاوز الـ 4mA .



الشكل(10): مولد جهد مرجعي داخل متحكمات STM32

يمكن ضبط الجهد المرجعي للمبدل التناظري الرقمي ليكون خارجي أو داخلي من خلال البتات ENVR , HIZ bits الموجودين في المسجل VREFBUF_CSR وعلى المستخدم أن ينتظر حتى يتم تفعيل البت VRR والذي يعني أن الجهد المرجعي الخارجي قد وصل إلى القيمة المطلوبة.

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

الجدول(3): ضبط الجهد المرجعي للمبدل التناظري الرقمي

عند استخدام مولد الجهد المرجعي الداخلي يجب وصل مكثفات على القطب Vref+ وفي هذه الحالة لن تحتاج لوصل دائرة خارجية لتوليد الجهد المرجعي، كما يدعم مولد الجهد المرجعي الداخلي كما ذكرنا قيمتي جهد هي:

- 2.5V: وهذا يتطلب أن يكون الجهد على الطرف $VDDA \geq 2.4$
- 2.48V: وهذا يتطلب أن يكون الجهد على الطرف $VDDA \geq 2.8$

يتم تعريف أقطاب الجهد المرجعي للمتحكم من خلال ال - Datasheet ويفترض أن يتم توصيلها بمجالات معينة من الجهود 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}

الشكل (11): أقطاب الجهد المرجعي للمبدل التشابهي

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

$$V_{in} = ADC_{Res} * (reference\ voltage / 4096)$$

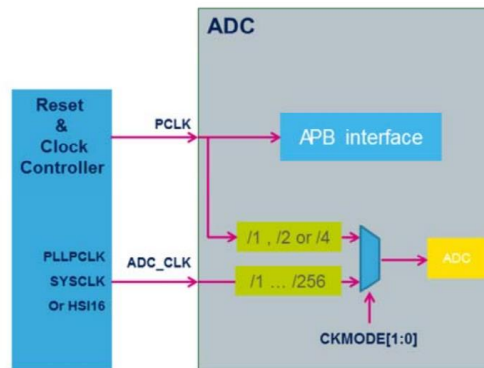
حيث :

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

اختيار مصدر الساعة للمبدل التشابهي:

يمكن اختيار مصدر الساعة للمبدل التشابهي الرقمي لتكون:

- إما APB clock: عندما يكون المطلوب أن يعمل النظام بالتزامن
- أو dedicated ADC clock: عندما يكون المطلوب أن تعمل الـ CPU بسرعة بطيئة وأن يعمل الـ ADC بسرعة عالية وبمعدل أخذ عينات عالي ، في هذه الحالة يقوم الـ ADC باستخدام PCLK , ADC_CLK

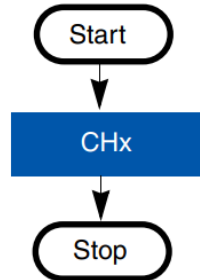


الشكل (12): ADC clocks

3. أنماط عمليات التحويل ADC conversion modes

يمكن للمبدل التشابهي الرقمي أن يعمل بعدة أنماط عمل هي :

1- Single-channel, single conversion mode: وهو نمط العمل الأسهل والأبسط مقارنة مع بقية الأنماط، حيث يقوم الـ ADC بعملية تحويل وحيدة (وعملية أخذ عينات وحيدة) ولقناة واحدة فقط ويتوقف عن العمل بمجرد انتهائه من عملية التحويل كما هو موضح بالشكل التالي:

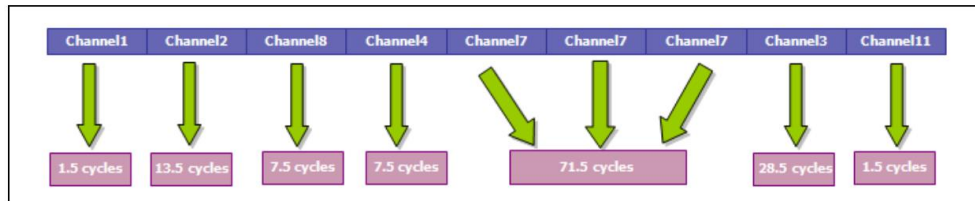


الشكل(13): Single-channel, single conversion mode

مثال : يـ ستخدم عادةً هذا النمط من العمل من أجل قياس مـ ستوى الجهد من أجل أخذ القرار كي يبدأ النظام أو لا، فإذا كان النظام يعمل بالبطارية، فقبل أن يعمل النظام يجب قياس مـ ستوى الجهد في البطارية كي يبدأ النظام أو لا، ففي حال كان مستوى البطارية منخفض عندها لن يبدأ النظام وستظهر رسالة "low battery".

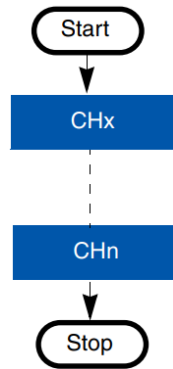
2- Multichannel (scan), single conversion mode

يستخدم هذا النمط من العمل لتحويل الجهود القادمة من عدة قنوات تشابهية ولمرة واحدة فقط، ويمكن اختيار ترتيب القنوات الذي ترغب به من خلال ADC sequencer وباستخدام زمن أخذ عينات مختلف لكل قناة من القنوات، يوضح الشكل التالي ترتيب معين للقنوات التشابهية وبأزمنة أخذ عينات مختلفة.



الشكل(14): Multichannel (scan), single conversion mode

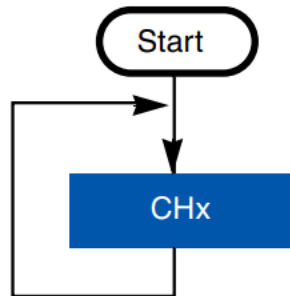
مثال: يمكن استخدام هذا النمط من العمل عند بدء نظام معين يتطلب فحص عدة بارامترات قبل أن يبدأ ، على سبيل المثال نظام الذراع المناور الذي يتطلب تحديد إحداثيات طرف الذراع والذي يتم من خلال معرفة إحداثيات كل مفصل من مفاصل الذراع، أيضاً يمكن استخدامه لإجراء مجموعة من القياسات اللازمة للأمان (كالجهد ، الضغط ، الحرارة ، وغيرها...) ولمرة واحدة فقط لأخذ القرار بتشغيل النظام أو لا.



الشكل(15):Multichannel (scan), single conversion mode

3- Single-channel continuous conversion mode

في هذا النمط من العمل يتم تحويل القيمة التشابهية القادمة على قناة واحدة إلى قيمة رقمية وتعاد عملية التحويل بشكل مستمر، ميزة النمط المستمر continuous mode أنها تسمح للمبدل أن يعمل في الخلفية بدون أي تدخل من الـ CPU، كما يمكن تفعيل نمط الـ DMA مما يخفف الحمل عن الـ CPU.

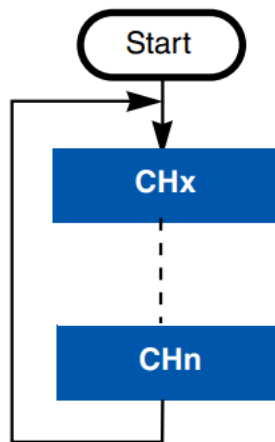


الشكل(16):Single-channel continuous conversion mode

مثال: يـستخدم هذا النمط لمراقبة جهد البطارية، أو لمراقبة درجة الحرارة وغيرها من التطبيقات، فمثلاً يتم قراءة درجة الحرارة بشكل مستمر ومقارنتها مع القيمة المرغوبة التي يتم إدخالها من قبل المستخدم وعند وصولها للقيمة المرغوبة يتم فصل التغذية عن السخان.

4- Multichannel (scan), continuous conversion mode

يـستخدم هذا النمط من العمل لتحويل الجهود القادمة من عدة قنوات تشابهية إلى قيم رقمية ويعيد العملية بشكل مستمر وذلك تبعاً لترتيب معين للقنوات حيث بإمكانك اختيار الترتيب المناسب للقنوات واختيار أزمدة أخذ عينات مناسبة لكل قناة من القنوات

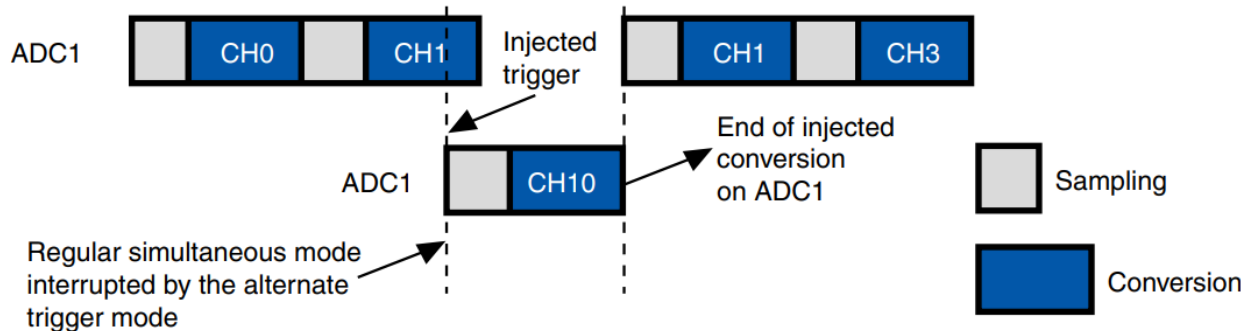


الشكل (17): Multichannel, continuous conversion mode

مثال: يمكن مراقبات جهود ودرجة حرارة عدة بطاريات أثناء عملية الشحن، وفي حال تجاوز الجهد أو الحرارة للنسب المسموحة يتم فصل الشاحن مباشرةً.

5- Injected conversion mode

يستخدم هذا النمط من العمل عندما تحتاج أن يتم قذح عملية التحويل التشابهي/رقمي من خلال حدث خارجي أو من خلال الكود، و للقناة التي تم حقنها (injected channel) أولوية أعلى من باقي القنوات النظامية (regular channels)، حيث يتم مقاطعة عملية التحويل الحالية وتحويل القناة التي تم قذحها بهذا النمط.



الشكل (18): Injected conversion mode

مثال: يستخدم هذا النمط من العمل في تطبيقات التحكم بالمحرك عندما ينتج عن تبديل الترانزستورات transistor switching ضجيج يؤدي إلى حدوث أخطاء في عمليات التحويل الخاصة بالمبدل التشابهي الرقمي، في هذه الحالة يستخدم مؤقت ويتم تفعيل نمط injected mode لتأخير عملية التبديل للـ ADC - لحين انتهاء الـ transistor switching

يوضح الجدول التالي المصادر المختلفة لقذح المبدل التشابهي الرقمي:

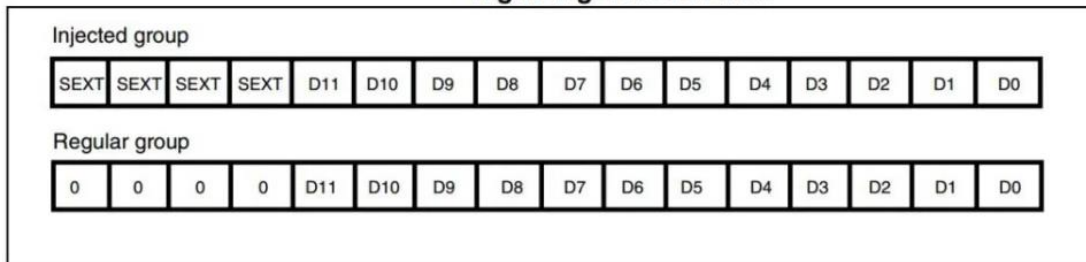
Source	Type	EXTSEL[2:0]
TIM1_CC1 event	Internal signal from on-chip timers	000
TIM1_CC2 event		001
TIM1_CC3 event		010
TIM2_CC2 event		011
TIM3_TRGO event		100
TIM4_CC4 event		101
EXTI line 11/TIM8_TRGO event ⁽¹⁾⁽²⁾	External pin/Internal signal from on-chip timers	110
SWSTART	Software control bit	111

الجدول (4): مصادر قذح الـ ADC

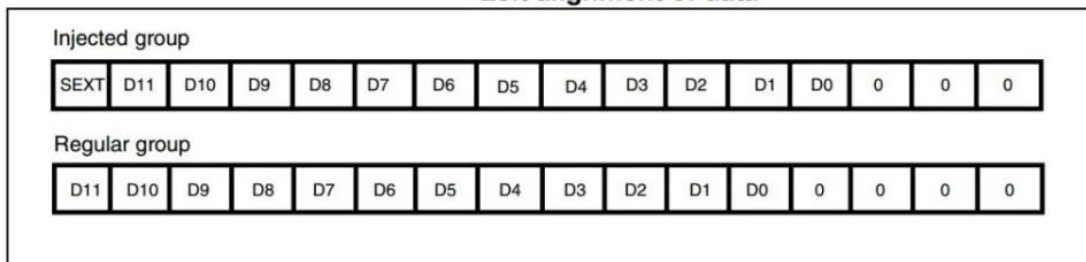
محاذاة البيانات ADC Result Data Alignment

يحدد بت ALIGN في الم سجل ADC_CR2 محاذاة البيانات المخزنة بعد عملية التحويل، حيث يمكن محاذاة البيانات إلى اليسار أو اليمين كما هو موضح في الرسم التخطيطي أدناه.

Right alignment of data



Left alignment of data



الشكل (19): Alignment of data

يتم تعريف الإزاحة offset من قبل المستخدم للبيانات المحولة القادمة من القنوات injected group من خلال المسجلات ADC_JOFRx ، لذا يمكن أن تكون قيم النتائج سالبة، ويعبر البت SEXT عن الإشارة، أما بالنسبة للـ regular group فلا يتم طرح أي إزاحة offset لذا يتم تمثيلها بـ 12 بت.

4. طرق قراءة المبدل التشابهي الرقمي ADC conversion modes

يوجد ثلاث طرق رئيسية لقراءة الـ ADC هي:

1- Polling method: تعتبر الطريقة الأسهل في كتابة الكود لقراءة القيمة القادمة من إحدى القنوات التشابهيّة، ولكنها ليست الأكثر فعالية، حيث علينا أن نبدأ بعملية التحويل وتنتظر حتى الانتهاء من عملية التحويل حينها يمكن للـ CPU استكمال تنفيذ الكود الرئيسي.

2- The interrupt Method: تعتبر هذه الطريقة طريقة فعالة لا تستخدم المبدل التشابهي الرقمي، حيث نقوم بقدرح المبدل فقط ويمكن للـ CPU أن تستكمل تنفيذ الكود والمهام المطلوبة منها حين انتهاء عملية التحويل عندها سيقوم المبدل بطلب مقاطعة وستتوجه الـ CPU لبرنامج خدمة المقاطعة وتحفظ نتائج عملية التحويل ليتم معالجته، وعلى الرغم من كونها طريقة فعالة، إلا أن طريقة المقاطعة يمكن أن تكون ضيف الكثير من الحمل إلى وحدة المعالجة المركزية وتسبب في تحميل وحدة المعالجة المركزية بشكل كبير جداً. خاصة عند إجراء العديد من التحويلات في الثانية، أيضاً وبسبب وجود عدة قنوات تشابهيّة فقد تحدث عدة مقاطعات في نفس الوقت مما يؤدي إلى حدوث تأخير في الاستجابة للمقاطعة (Latency) والتي يمكن تجنبها باستخدام الـ DMA.

يمكن للمبدل التشابهي الرقمي أن يولد 3 أنواع من المقاطعات موضحة في الجدول التالي:

ADC interrupts

Interrupt event	Event flag	Enable Control bit
End of conversion regular group	EOC	EOCIE
End of conversion injected group	JEOC	JEOCIE
Analog watchdog status bit is set	AWD	AWDIE

الجدول (5): ADC interrupts

3- DMA Method: وهي الطريقة الأكثر فعالية في استخدام المبدل التشابهي الرقمي وذلك لأن استخدام عدة قنوات تشابهيّة وبمعدلات عالية لأخذ العينات حيث يتم ترحيل نتائج التحويل إلى الذاكرة بدون تدخل الـ CPU مما يعني استغلال أفضل للوقت.

5. الأخطاء الناتجة عن التحويل:

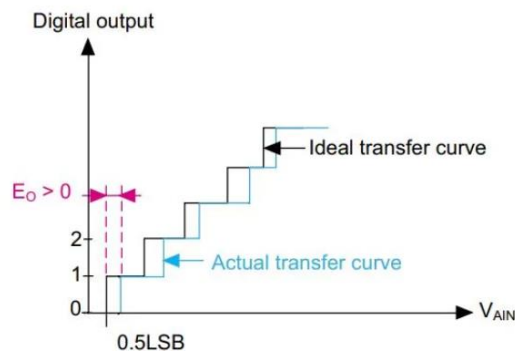
يمكن تقسيم الأخطاء إلى مجموعتين رئيسيتين:

1- أخطاء ناتجة عن المبدل نفسه

تقسم أيضاً إلى ثلاث مجموعات:

- ADC offset error: وهو الخطأ الناتج بين أول عملية تحويل فعلية وأول 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 0.5LSB والـ 1.5LSB عندها يكون الخرج الرقمي 1 منطقي، ويتم الإشارة إلى خطأ الإزاحة بـ Eo ويمكن معايرته من خلال الكود.

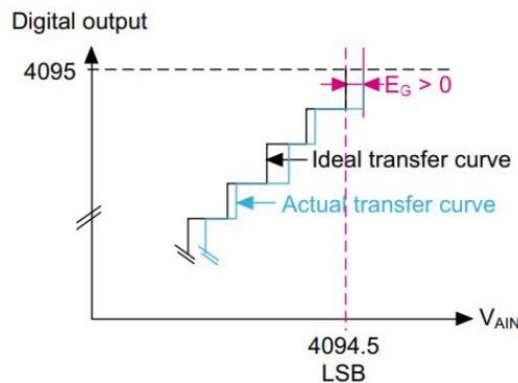
Positive offset error representation



الشكل(20): خطأ الإزاحة

- **خطأ الربح Gain error:** وهو الانحراف بين آخر تحويل فعلي وآخر تحويل مثالي ويعبر عنه بـ E_G حيث يتم آخر تحويل فعلي عند الانتقال بين 0xFFE و 0xFFFF ، مثالياً يجب أن يتم التحويل الأخير عندما يكون جهد الدخل التشابهي مساوياً لـ $V_{ref} - 0.5LSB$ ، فمن أجل جهد مرجعي $V_{ref}=3.3V$ فإن التحويل الأخير يجب أن يتم عند الجهد $3.299597V$ ، فإذا كان خرج المبدل 0xFFFF من أجل جهد دخل تشابهي أقل من القيمة المذكورة عندها يكون هناك خطأ ربح سلبي

Positive gain error representation



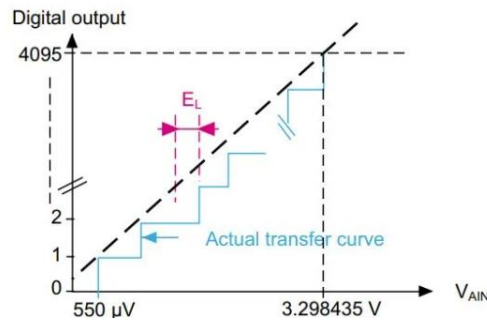
الشكل(21): خطأ الربح

يتم حساب خطأ الربح من خلال العلاقة التالية:

$$E_G = \text{Last actual transition} - \text{ideal transition}$$

- **Integral Linearity Error:** وهو الانحراف الأعظمي بين أي قيمة تحويل فعلي وبين الخط الواصل بين أول عملية تحويل فعلي وآخر عملية تحويل فعلي ويرمز له E_L

Integral linearity error representation



الشكل (22): Integral Linearity Error

2- أخطاء ناتجة عن البيئة المحيطة: أيضاً تقسم إلى عدة أنواع:

- ADC Reference Voltage Noise
- Analog Input Signal Noise
- ADC Dynamic Range Bad Matching
- Analog Signal Source Impedance (Resistance)
- Analog Signal Source Capacitance & Parasitics
- Injection Current Effect
- IO Pins Cross-Talking
- EMI-Induced Noise

6. التطبيق العملي: سنقوم في هذا التطبيق بالتحكم بشدة إضاءة ليد موصول على أحد أقطاب الـ PWM من

خلال مقاومة متغيرة موصولة على أحد أقطاب الدخل التشابهي

سنقوم بتنفيذ المشروع وفقاً للتسلسل التالي:

- ضبط تردد ساعة المتحكم على أعلى تردد وهو 64MHZ (لأننا نستعمل HSI)
- ضبط قطب الدخل التشابهي (CH7) في نمط التحويل لمرة واحدة Single conversion mode
- حيث سنربط مقاومة متغيرة معه.
- ضبط الـ timer2 في نمط PWM على القناة CH1 (وسنربط معه ليد).

سنقوم بتنفيذ المشروع بالطرق الثلاث DMA , interrupt , Polling، حيث سنقوم في البداية بقراءة القيمة القادمة من المقاومة المتغيرة الموجودة على القناة CH7 للمبدل التشابهي ومن ثم سنقوم بإسناد هذه القيمة لم سجل المؤقت CCR والذي من خلاله يتم تحديد دورة التشغيل dutycycle والتي تحدد شدة إضاءة الليد.

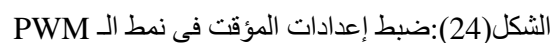
الطريقة الأولى: استخدام نمط الـ Polling:

سنقوم بضبط الإعدادات من خلال أداة CubeMx المدمجة داخل بيئة STM32CubeIDE وفقاً للخطوات التالية:

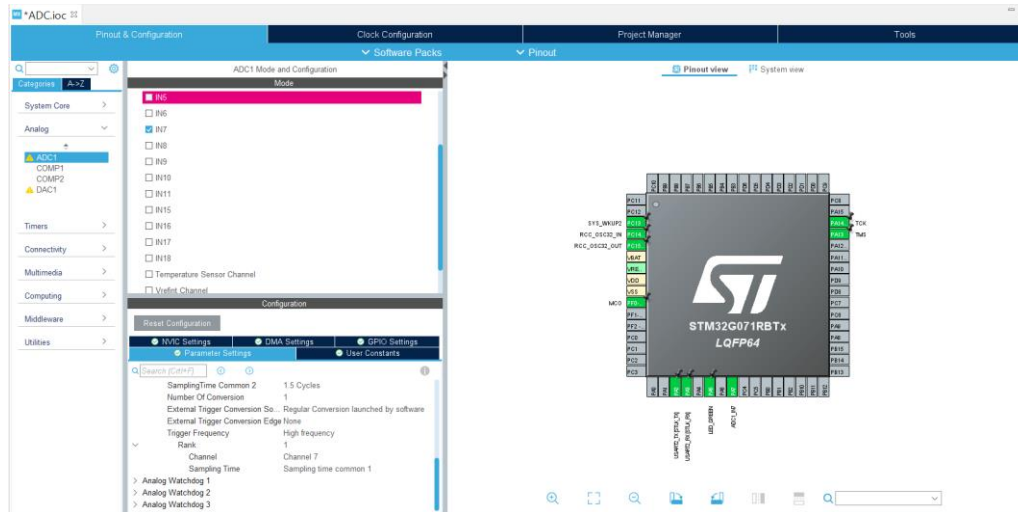
The screenshot shows the STM32Cube Configurator interface. The 'Target Selection' tab is active, displaying a search for 'STM32G0 Series'. The 'Board List' shows one item: 'NUCLEO-G071RB' by 'STMicroelectronics'. The 'Features' tab is also visible, showing a list of features for the selected board.

الخطوة الثانية: ضبط إعدادات المؤقت ليعمل في نمط PWM

نمط إشارة الـ PWM

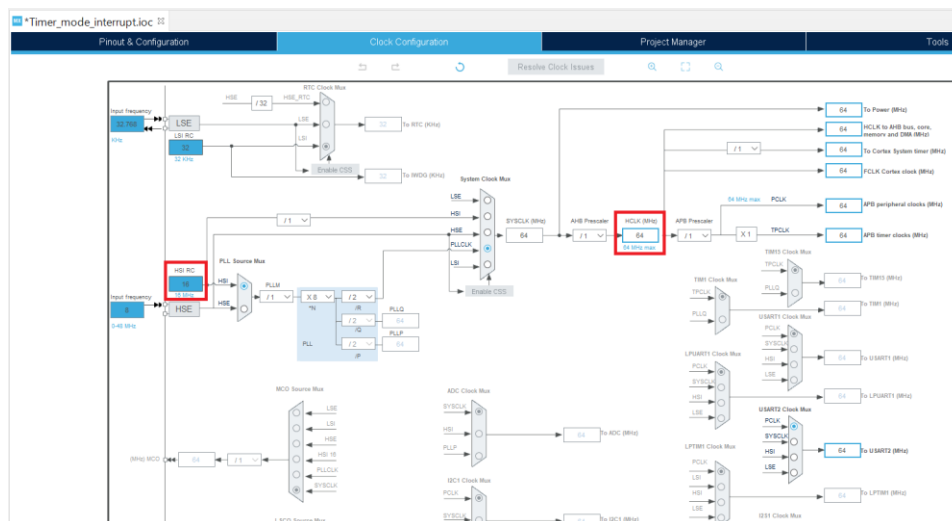


الخطوة الثالثة: قم بضبط إعدادات المبدل التشابهي الرقمي ADC على القناة التشابهية السابعة CH7 بالشكل التالي:



الشكل (25): ضبط إعدادات المبدل التشابهي الرقمي

الخطوة الرابعة: ضبط تردد ساعة المتحكم



الشكل (26): ضبط تردد ساعة المتحكم

الخطوة الخامسة: توليد الكود بناءً على الإعدادات التي تم ضبطها من خلال ctrl+s

يصبح الكود النهائي بالشكل التالي:

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

```
ADC_HandleTypeDef hadc1;
TIM_HandleTypeDef htim2;
```

تعريف المبدل التشابهي ADC1
تعريف المؤقت TIM2

```
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);
static void MX_TIM2_Init(void);
```

```
int main(void)
```

```
{
    uint16_t AD_RES = 0;

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

    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 Perihperal & TimeOut = 1mSec
        HAL_ADC_PollForConversion(&hadc1, 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);
    }
}
```

التصريح عن الدوال المسؤولة عن تهيئة ساعة المتحكم ، أقطاب الدخل/الخرج ، تهيئة المبدل التشابهي الرقمي ADC1 ، تهيئة المؤقت TIM2

التصريح عن متحول AD_RES لنخزن فيه لاحقاً القيمة الناتجة عن المحول التشابهي الرقمي

استدعاء الدوال المسؤولة عن تهيئة ساعة المتحكم ، أقطاب الدخل/الخرج ، تهيئة المبدل التشابهي الرقمي ADC1 ، تهيئة المؤقت TIM2

بدء عمل المؤقت في نمط الـ PWM وعلى القناة الأولى

بدء المعايرة الذاتية للمبدل التشابهي الرقمي adc1

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

طلب عملية تحويل من المبدل التشابهي الرقمي

إسناد ناتج عملية التحويل إلى المتغير AD_RES

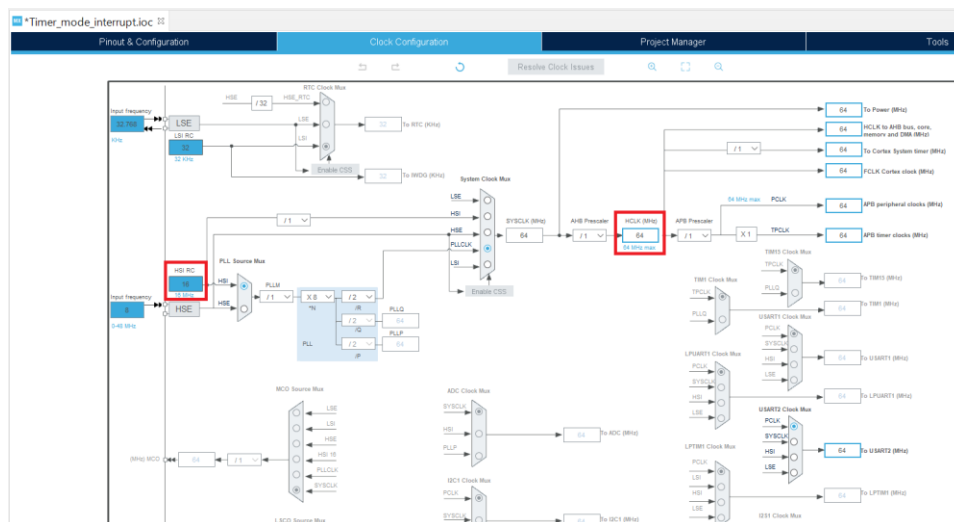
إسناد قيمة المتغير AD_RES التي تعبر عن قيمة المقاومة المتغيرة ، إلى المسجل CCR1 من المؤقت TIM2 ، المسؤول عن دورة التشغيل dutycycle إجراء تأخير زمني بمقدار 1sec

الطريقة الثانية: استخدام نمط الـ Interrupt:

سنقوم بضبط الإعدادات من خلال أداة CubeMx المدمجة داخل بيئة STM32CubeIDE وفقاً للخطوات التالية:

الخطوة الأولى والثانية هي نفس الخطوات الموجودة في الطريقة الأولى

الخطوة الرابعة: ضبط تردد ساعة المتحكم



يصبح الكود النهائي بالشكل التالي:

```
#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);
```

```
int main(void)
```

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

```
    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_IT(&hadc1);
        // Update The PWM Duty Cycle With Latest ADC
        Conversion Result
```

```
        TIM2->CCR1 = (AD_RES<<4);
        HAL_Delay(1);
```

```
    }
}
```

التصريح عن متحول عام (لأننا سنستخدمه في الدالة الرئيسية وفي برنامج خدمة المقاطعة) AD_RES لنخزن فيه لاحقاً القيمة الناتجة عن المحول التشابهي الرقمي

تعريف المبدل التشابهي ADC1
تعريف المؤقت TIM2

التصريح عن الدوال المسؤولة عن تهيئة ساعة المتحكم ، أقطاب الدخل/الخرج ، تهيئة المبدل التشابهي الرقمي ADC1 ، تهيئة المؤقت TIM2

تهيئة مكتبة HAL
استدعاء الدوال المسؤولة عن تهيئة ساعة المتحكم ، أقطاب الدخل/الخرج ، تهيئة المبدل التشابهي الرقمي ADC1 ، تهيئة المؤقت TIM2

بدء عمل المؤقت في نمط الـ PWM وعلى القناة الأولى

بدء المعايرة الذاتية للمبدل التشابهي الرقمي adc1

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

إسناد قيمة المتغير AD_RES التي تعبر عن قيمة المقاومة المتغيرة ، إلى المسجل CCR1 من المؤقت TIM2 ، المسؤول عن دورة التشغيل dutycycle
إجراء تأخير زمني بمقدار 1sec

```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef*
hadc)
{

// Read & Update The ADC Result
AD_RES = HAL_ADC_GetValue(&hadc1);
}
```

برنامج خدمة المقاطعة والذي يتم استدعاؤه عند إتمام عملية التحويل التشابهي الرقمي

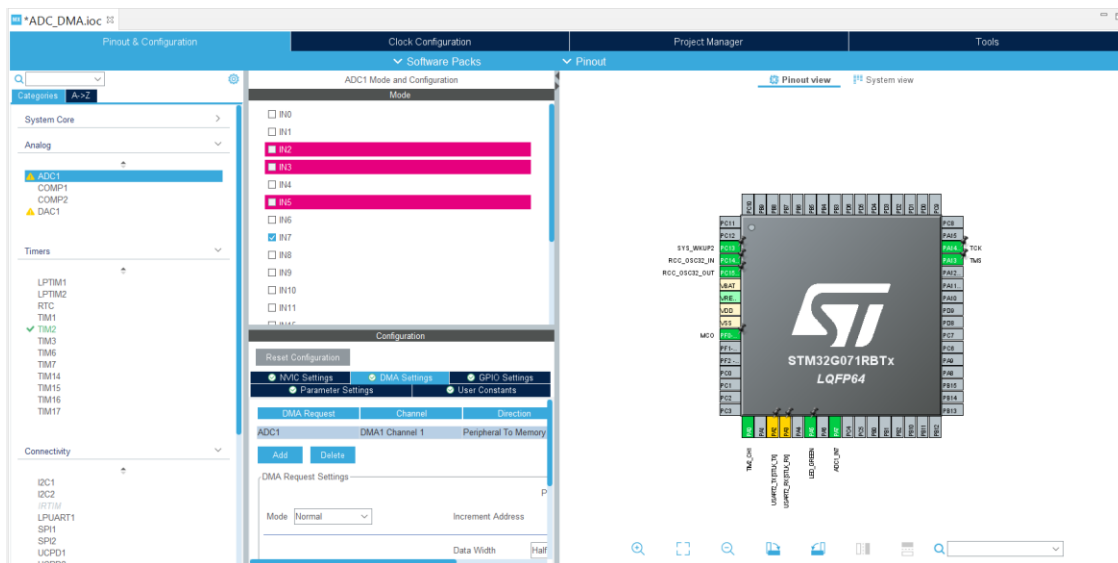
إسناد ناتج عملية التحويل إلى المتغير AD_RES (نحصل على ناتج التحويل التشابهي الرقمي من برنامج خدمة المقاطعة)

الطريقة الثالثة: استخدام نمط الـ DMA:

سنقوم بضبط الإعدادات من خلال أداة CubeMx المدمجة داخل بيئة STM32CubeIDE وفقاً للخطوات التالية:

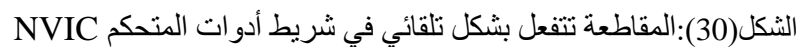
الخطوة الأولى والثانية هي نفس الخطوات الموجودة في الطريقة الأولى

الخطوة الثالثة: ضبط إعدادات المبدل التشابهي بنفس الإعدادات السابقة ، فقط سنفعّل نمط الـ DMA من خلال إضافة قناة إلى DMA ونلاحظ أن المقاطعة تتفعل بشكل تلقائي في المتحكم NVIC

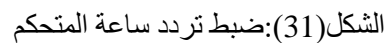


الشكل (29): إضافة قناة إلى DMA

نلاحظ أن المقاطعة تتفعل بشكل تلقائي في شريط أدوات المتحكم NVIC



الخطوة الرابعة: ضبط تردد ساعة المتحكم



الخطوة الخامسة: توليد الكود بناءً على الإعدادات التي تم ضبطها من خلال **ctrl+s**

يصبح الكود النهائي بالشكل التالي:

التصريح عن متحول عام (لأننا
سنستخدمه في الدالة الرئيسية وفي
برنامج خدمة المقاطعة) AD_RES
لنخزن فيه لاحقاً القيمة الناتجة عن
المحول التشابهي الرقمي

<pre> ADC_HandleTypeDef hadc1; DMA_HandleTypeDef hdma_adc1; TIM_HandleTypeDef htim2; void SystemClock_Config(void); static void MX_GPIO_Init(void); static void MX_DMA_Init(void); static void MX_ADC1_Init(void); static void MX_TIM2_Init(void); int main(void) { HAL_Init(); SystemClock_Config(); MX_GPIO_Init(); MX_DMA_Init(); MX_ADC1_Init(); MX_TIM2_Init(); 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 // Pass (The ADC Instance, Result Buffer Address, Buffer Length) HAL_ADC_Start_DMA(&hadc1, &AD_RES, 1); HAL_Delay(1); } } void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc) { // Conversion Complete & DMA Transfer Complete As Well // So The AD_RES Is Now Updated & Let's Move IT To The PWM CCR1 </pre>	<p>تعريف المبدل التشابهي ADC1 تعريف نمط الـ DMA للمبدل التشابهي ADC1</p> <p>تعريف المؤقت TIM2</p> <p>التصريح عن الدوال المسؤولة عن تهيئة ساعة المتحكم ، أقطاب الدخل/ الخرج ، تهيئة الـ DMA ، تهيئة المبدل التشابهي الرقمي ADC1 ، تهيئة المؤقت TIM2</p> <p>تهيئة مكتبة HAL استدعاء الدوال المسؤولة عن تهيئة ساعة المتحكم ، أقطاب الدخل/الخرج ، استدعاء دالة تهيئة الـ DMA ، تهيئة المبدل التشابهي الرقمي ADC1 ، تهيئة المؤقت TIM2</p> <p>بدء عمل المؤقت في نمط الـ PWM وعلى القناة الأولى</p> <p>بدء المعايرة الذاتية للمبدل التشابهي الرقمي adc1</p> <p>إعطاء إشارة البدء للمبدل التشابهي الرقمي في نمط الـ DMA مع تخزين القيمة الناتجة عن عملية التحويل في المتغير AD_RES إجراء تأخير زمني بمقدار 1sec</p> <p>برنامج خدمة المقاطعة والذي يتم استدعاؤه عند إتمام عملية التحويل التشابهي الرقمي</p> <p>إسناد قيمة المتغير AD_RES التي تعبر عن قيمة المقاومة المتغيرة ، إلى</p>
--	---

<pre>// Update The PWM Duty Cycle With Latest ADC Conversion Result TIM2->CCR1 = (AD_RES<<4); }</pre>	<p>المسجل CCR1 من الموقت TIM2 ، المسؤول عن دورة التشغيل dutycycle</p>
--	---