

دراسة وتطوير إمكانيات المتحكمات الصغيرة وشرائح Arduino في مجال تحصيل المعطيات

سوزي صالح *

علي سقور **

(تاريخ الإيداع ١٦ / ٤ / ٢٠١٩ . قُبل للنشر ٨ / ٩ / ٢٠١٩)

الملخص

تتضمن هذه المقالة دراسة تفصيلية حول قدرة اللغة البرمجية الخاصة بشرائح Arduino على التعامل مع الإشارات التشابهية والرقمية، وكذلك قدرة المتحكمات للعمل كمعالجات إشارات رقمية (DSP) وإشارات تشابهية عالية الدقة والسرعة، بالإضافة إلى تصميم نظام قياس باستخدام شريحة Arduino ومحولات تشابهية رقمية خارجية؛ لاختبار تردد أخذ العينات ودقة التحويل التشابهي الرقمي، واقتراح حلول برمجية بالنسبة إلى شرائح Arduino وحلول تقنية بالنسبة إلى المتحكمات لزيادة تردد أخذ العينات الرقمية والتشابهية. وقد دُرست إمكانية استخدام المحولات من نوع سيغما دلتا (SIGMA DELTA) مع المتحكمات وشرائح Arduino من ناحية تردد أخذ العينات والضجيج المرافق لها. تمت المحاكاة الإلكترونية باستخدام برنامج بروتوس PROTEUS وبرنامج أردوينو ARDUINO IDE لكتابة الكود البرمجي، بالإضافة إلى استخدام أداة راسم الإشارة التسلسلي Serial Oscilloscope. **الكلمات المفتاحية:** المحول التشابهي الرقمي (ADC)، معالج الإشارات الرقمية (DSP)، محولات سيغما ديلتا Sigma-Delta ADC، تردد أخذ العينات.

* أستاذ مساعد في قسم هندسة تكنولوجيا الاتصالات - كلية هندسة تكنولوجيا المعلومات والاتصالات - جامعة طرطوس - سورية.
** مهندس - ماجستير هندسة تكنولوجيا الإلكترونيات - قسم النظم الحاسوبية والإلكترونية - كلية هندسة تكنولوجيا المعلومات والاتصالات - جامعة طرطوس - سورية.

Studying and development of microcontrollers and Arduino capabilities in field of data acquisition

Susi Saleh *

Ali Sakor **

(Received 16 / 4 / 2019 . Accepted 8 / 9 / 2019)

ABSTRACT

This article includes a detailed study about the ability of the programming language of the Arduino chips to deal with analog and digital signals as well as the ability of microcontrollers to work as digital signal processors (DSP) and high resolution and speed analog signals, In addition to the design of a measurement system using the Arduino chip and external digital analog converters to test sampling frequency and digital analog conversion accuracy, and propose software solutions for Arduino and technical solutions for microcontrollers to increase digital and analog sampling frequency. The possibility of using SIGMA DELTA converters with microcontrollers and Arduino chips was studied in terms of sampling frequency and accompanying noise. The electronic simulation was performed using the PROTEUS software and the ARDUINO IDE program to write the code in addition to using Serial Oscilloscope.

Key words: Digital Analog Converter (ADC), Digital Signals Processor (DSP), Sigma-Delta ADC, Sampling Frequency.

*Assistant Professor – Faculty of Information and communication Technology – ICt Department – Tartous University, Syria.

**Engineer – Master of electronics technology engineer– Computer and Electronic System Department – Information and communication Technology Engineering – Tartous University, Syria.

مقدمة

إن المتطلب الأساسي للتعامل مع التطور التكنولوجي الحاصل هو تطوير قدرة الأجهزة الإلكترونية للتعامل مع هذا التطور، وأصبحت الحاجة ملحة لزيادة إمكانية المتحكمات المصغرة والشرائح المصممة مسبقاً لتلبي حاجة المطورين والباحثين، من دون الحاجة لشراء أجهزة ذات معالجات باهظة الثمن. إن الاهتمام بهذا الجانب سيؤدي لدمج مجالي الاتصالات والنظم الحاسوبية بشكل وثيق؛ حيث سيتمكن مستخدمو المتحكمات والشرائح المسبقة التصميم من التعامل مع الإشارات الرقمية والتشابهية ذات السرعات العالية. اهتمت العديد من الدراسات السابقة بتصميم أجهزة بإمكانيات بسيطة للتعامل مع الإشارات الكهربائية باستخدام شرائح Arduino وبعض البرمجيات لعرض النتائج [1][2][3]، وكذلك اهتم بعضها بتصميم أجهزة متطورة للتعامل مع الإشارات الكهربائية باستخدام معالجات مرتفعة التكلفة [13][14]. يقدم هذا العمل حلاً لتصميم أجهزة متطورة بتكلفة منخفضة، كتصميم رواسم إشارة بسرعات عالية ومحللات طيفية رقمية وأجهزة قياس عالية الدقة باستخدام شرائح المتحكمات ذات الإمكانيات المحدودة.

هدف البحث

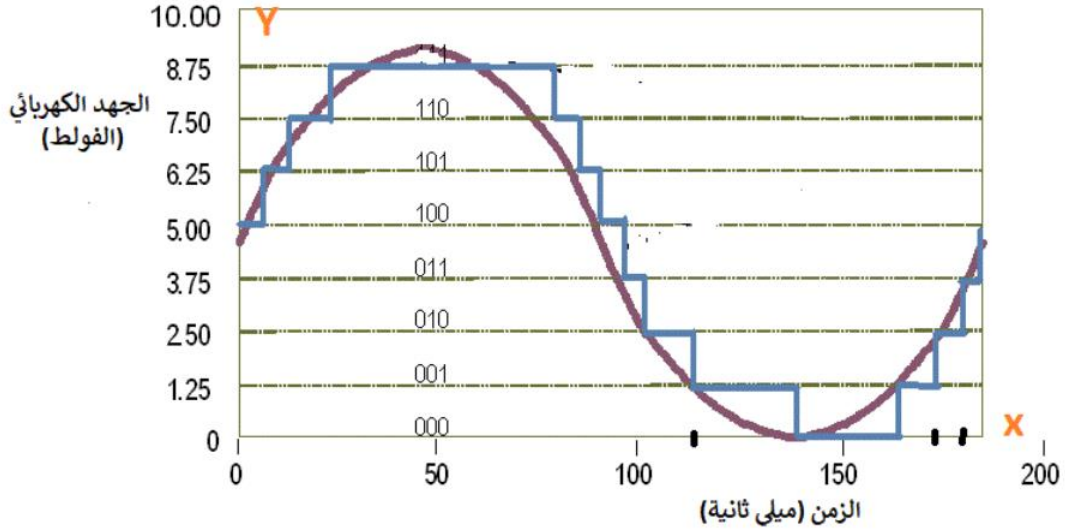
يقوم البحث على تصميم دائرة إلكترونية وتطويرها تسمح للمتحكمات وشرائح Arduino بالتواصل مع الإشارات الكهربائية فائقة السرعة والدقة، حيث سندرس الإمكانيات المتاحة لسرعة أخذ العينات ودقة القراءة بالنسبة إلى هذه الشرائح، وسنقوم باختبار المحولات التشابهية الرقمية من نوع sigma-delta ونقارنها بالمحولات الداخلية للمتحكمات، وكذلك سنختبر تأثير اللغة البرمجية الخاصة بـ Arduino ونقارنها بلغة المسجلات.

طرائق البحث ومواده

أنجز هذا البحث اعتماداً على دراسات ومراجع علمية حديثة وعديدة تختص في هذا المجال وقد أخذت نتائجها وتوصياتها بعين الاعتبار، كما تم استخدام برنامج أردوينو Arduino وبروتس Proteus وراسم الإشارة التسلسلي SerialOscilloscope [4][5][6] [12].

1- المحول التشابهي الرقمي

هو عبارة عن وحدة إلكترونية تقوم بتحويل الإشارات الكهربائية التشابهية إلى رقمية، حيث يكون دخلها عبارة عن إشارة تشابهية بعدد غير منته من القيم يتم تحويل كل قيمة إلى كلمة رقمية بـ N bit، وتزداد دقة المحول بزيادة عدد الخانات الثنائية التي تمثل الإشارة الرقمية، يبين الشكل (1) إشارة الخرج لمحول 3-bit عند تطبيق إشارة دخل تشابهية بمطال أعظمي 10V.



الشكل (1) إشارتي الدخل والخرج للمحول التشابهي الرقمي 3-bit

1-1- معدل أخذ العينات

تتم هذه العملية بأخذ عينات من الإشارة التماثلية كل فترة زمنية معينة وهي ما تسمى Sample rate (معدل العينات) ويقاس بعينة بالثانية (Sample per second)، وكلما زاد هذا المعدل كانت الإشارة قريبة إلى الإشارة الأصلية، ولكي نتمكن من استعادة الإشارة الأصلية من خلال هذه العينات ينبغي علينا تحقيق شرط نايكويست شانون الذي ينص: إن تردد أخذ العينات (F_s) يجب أن يكون على الأقل ضعف أكبر تردد تحتوي عليه الإشارة التشابهيّة، ويعطى بالعلاقة الآتية:

$$F_s \geq 2F_m \quad (1)$$

حيث:

F_s : تردد أخذ العينات

F_m : أكبر تردد تحتوي عليه الإشارة التماثلية.

1-2- دقة المحول Resolution

إن أهم مواصفات المحول التشابهي الرقمي ADC هي "الدقة أو القدرة على التحليل" (أقل كمية يمكن قياسها)، وتحدد مدى دقة قياس ADC لإشارات الدخل التشابهيّة.

على سبيل المثال إذا كان الجهد المرجعي لمحول ADC من 0V إلى 5V، فإن المحول 8-bit سوف يقسمه إلى 256 قسم (من 0 إلى 255)، لذلك يمكنه القياس بدقة تصل إلى 19.6mV كما يلي:

$$\frac{V_{ref}}{2^n - 1} = \frac{5}{255} = 19.6mV \quad (2)$$

في حين أن المحول 10-bit سوف يقسمه إلى 1024 قسم (أدنى نتيجة تحويل هي 0، وأقصى نتيجة تحويل هي 1023)، عندئذ تكون دقة القياس على النحو الآتي:

$$\frac{V_{ref}}{2^n - 1} = \frac{5}{1023} = 4.88mV \quad (3)$$

بالتالي نلاحظ أن المحول 8-bit لا يمكنه التفرقة بين 1mV و 18mV.

3-1- الجهد المرجعي Reference Voltage:

يحدد الجهد المرجعي الحد الأدنى والأقصى لمدى جهد الدخل التشابهي، حيث يمتلك المحول التشابهي الرقمي عادةً جهداً مرجعياً سالباً $-V_{ref}$ وجهداً مرجعياً موجباً $+V_{ref}$.
الجهد المرجعي السالب $-V_{ref}$ يحدد الحد الأدنى لجهد الدخل التشابهي، في حين أن الجهد المرجعي الموجب يحدد الحد الأكبر لجهد الدخل التشابهي.

مثلاً: إذا تم تطبيق إشارة قيمتها $-V_{ref}$ على دخل محمول تشابهي رقمي 10-bit، عندئذ تكون نتيجة التحويل 0 (000000000) وإذا تم تطبيق إشارة قيمتها $+V_{ref}$ فإن النتيجة سوف تكون 1023 (111111111).

2- محاكاة الدارات في بيئة proteus

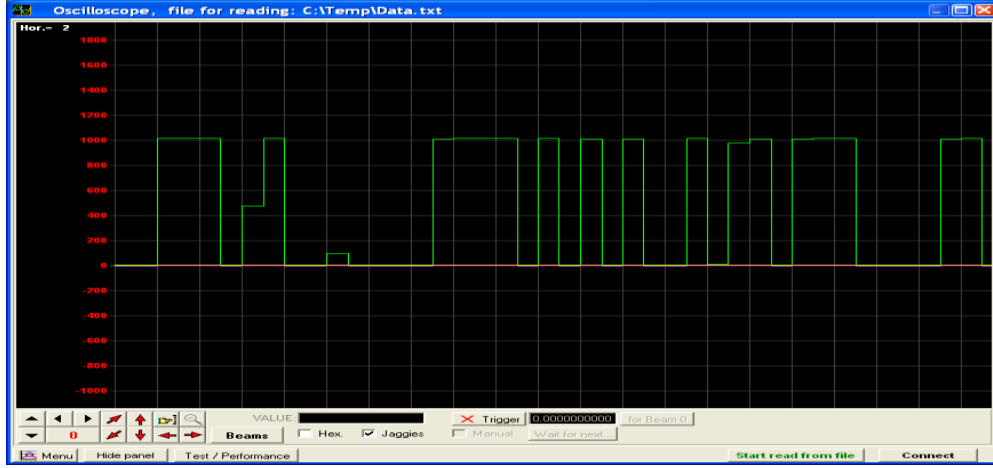
إن برنامج Proteus لنمذجة النظم الافتراضية Virtual System Modeling - المبين في الشكل (2) أداة تجمع طيفاً من الأدوات البرمجية في برامج محاكاة النظم الإلكترونية؛ لتضع بين يدي الطالب والمهندس والمحترف بيئة متكاملة تحوي كل ما يلزمه من أدوات لعملية محاكاة واقعية، فهو يجمع ما بين نظم SPICE لمحاكاة الدارات والعناصر الإلكترونية (يوضحها بشكل مقاطع صورية متحركة لسهولة التعامل)، ونماذج المعالجات الصغيرة لتسهيل مرحلة لاحقة من المحاكاة للنظم الإلكترونية المعتمدة على المتحكمات الصغيرة، لذلك سنقوم بتصميم الدارات اللازمة للأمثلة الواردة في هذه المقالة وكذلك أخذ النتائج باستخدام بيئة proteus.



الشكل (2) بيئة المحاكاة proteus

3- راسم الإشارة التسلسلي serial oscilloscope

يبين الشكل (3) أداة راسم الإشارة التسلسلي؛ حيث تقوم هذه الأداة بتحويل القيم الواردة عبر المنفذ التسلسلي إلى رسوم بيانية، وتمتلك العديد من المواصفات مثل رسم 9 إشارات على 3 منحنيات زمنية، والتحكم بالمقياس الزمني، والرسم بالزمن الحقيقي، حيث تقوم بربط شريحة الأردوينو مع هذه الأداة باستخدام وصلة USP ونرسل القيم التي نريد رسمها باستخدام الاتصال التسلسلي، وكذلك سنربط هذه الأداة مع بيئة proteus حيث سنستخدم هذه الأداة كراسم إشارة بالنسبة إلى القيم المقروءة من قبل شريحة الأردوينو لمقارنتها مع الإشارات الأصلية.



الشكل (3) أداة راسم الإشارة التسلسلي

النتائج والمناقشة:

1- تردد أخذ العينات

لاختبار قدرة شريحة أردوينو ميغا ذات المتحكم atmega2560 على التعامل مع الإشارات الخارجية تم حساب تردد أخذ العينات الرقمية والتشابهية، حيث تم كتابة كود برمجي (باستخدام لغة البرمجة الخاصة ب Arduino) لقراءة 100 عينة من إشارة دخل خارجية عبر حلقة تكرارية لا تحتوي إلا تعليمة قراءة القطب المطلوب وتخزينها في مصفوفة، ويقوم المتحكم بحساب زمن قراءة العينات ومن ثم حساب تردد أخذ العينة الواحدة، وكانت النتائج كما في الجدول (1).

الجدول (1) تردد أخذ العينات للمتحكم atmega2560 بلغة Arduino

تردد أخذ العينات التشابهية	تردد أخذ العينات الرقمية
8.92KHz	177.3KHz

نلاحظ بالتجربة العملية أن تردد أخذ العينات الرقمية 177.304KHz وهو أكبر من تردد أخذ العينات التشابهية 8.925KHz، بالتالي فإن أعلى تردد يمكن قياسه بشكل صحيح بحسب شرط نايكويست شانون هو:

$$F_m \leq \frac{1}{2} F_s \quad (4)$$

حيث:

F_m : أعلى تردد في إشارة الدخل.

F_s : تردد أخذ العينات.

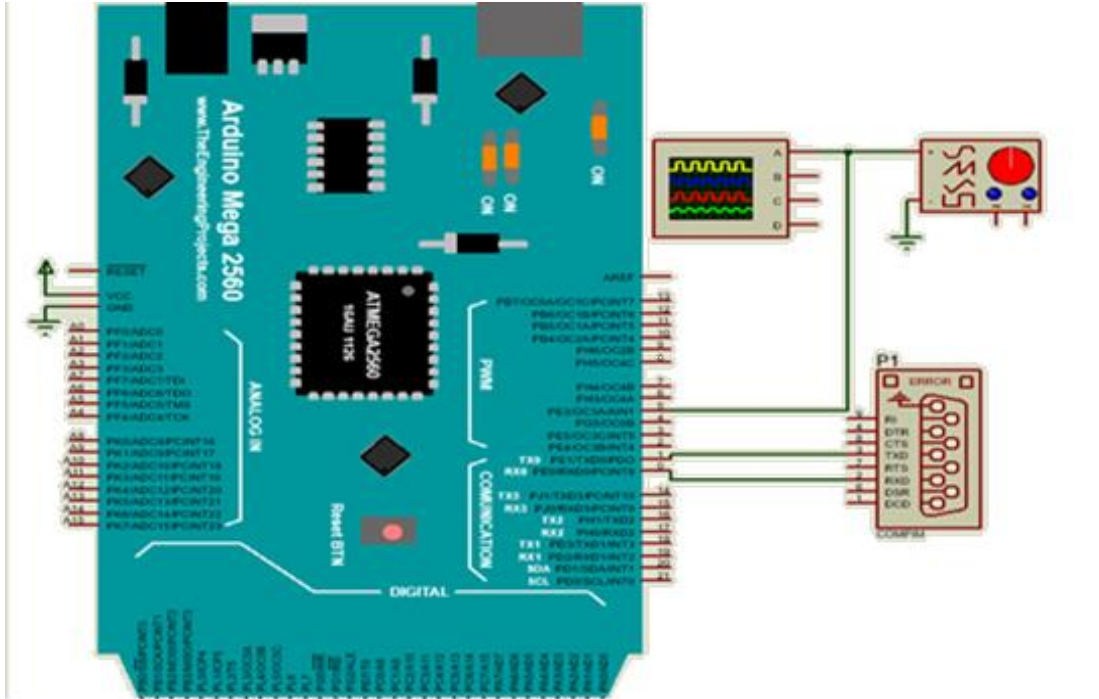
بالتعويض نجد أن أعلى تردد إشارة رقمية يمكن قراءته بشكل صحيح هو:

$$F_m \leq \frac{177.3}{2} = 88.65 \text{ KHz} \quad (5)$$

وأعلى تردد إشارة تشابهية يمكن قراءته بشكل صحيح هو:

$$F_m \leq \frac{8.92}{2} = 4.46 \text{ KHz} \quad (6)$$

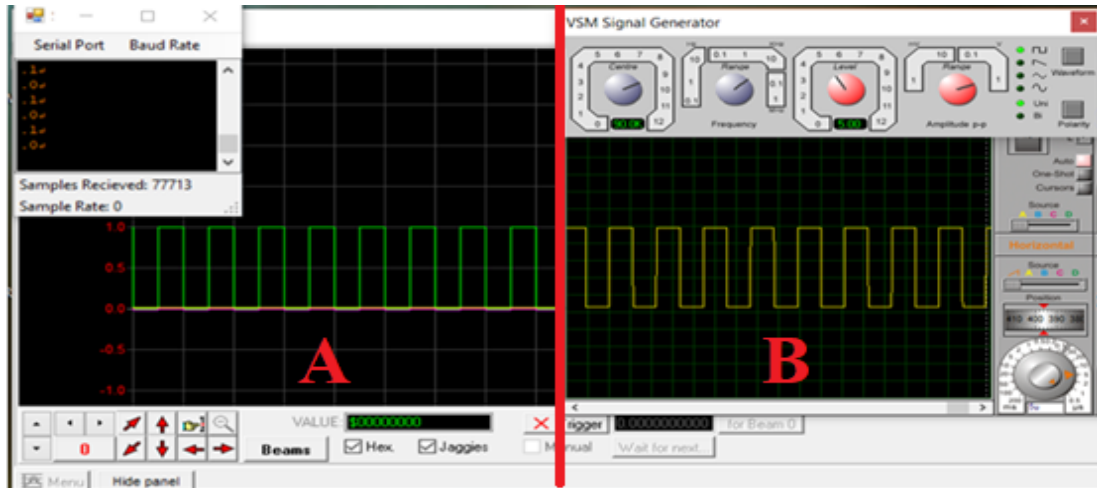
سنقارن النتائج العملية مع نتائج برنامج المحاكاة proteus لنختبر إمكانية الاعتماد على الأخير ونتأجه بالنسبة إلى البحث، تم إعادة قياس النتائج العملية السابقة باستخدام بيئة proteus، حيث جرى تصميم الدارة المبينة في الشكل (٤).



الشكل (4) دارة اختبار سرعة أخذ العينات الرقمية باستخدام بيئة proteus

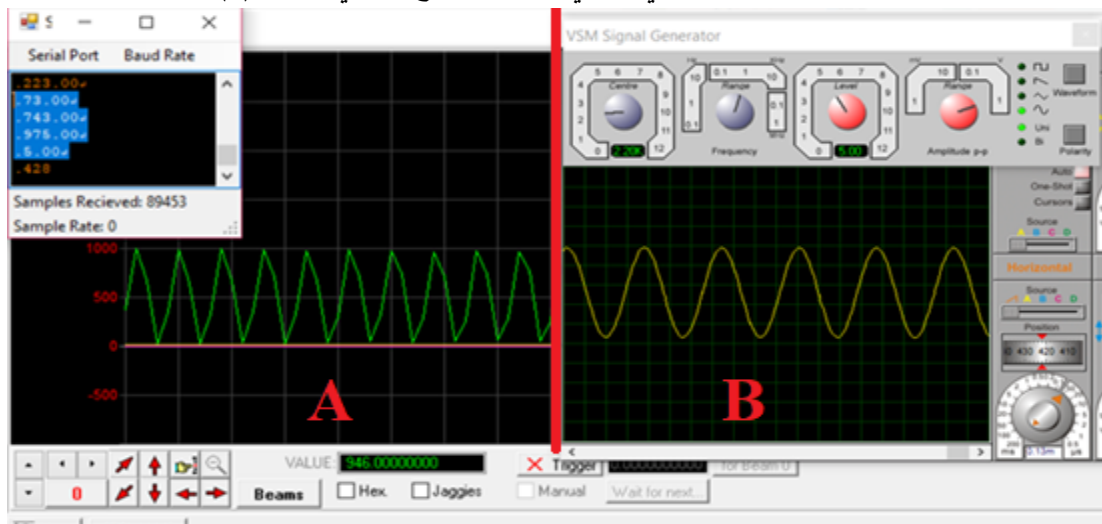
تتضمن هذه الدارة شريحة Arduino mega، ومولد إشارة بالإضافة إلى وصلة تسلسلية لعرض الإشارة التي تمكنت شريحة Arduino من قراءتها على برنامج serial oscilloscope.

بتطبيق إشارة مربعة بتردد 90KHz كما تظهر على راسم الإشارة الشكل (5) الجزء B تمكنت شريحة Arduino من أخذ 2 عينة لكل دور من الإشارة الرقمية، كما يوضح برنامج راسم الإشارة التسلسلي الجزء A من الشكل (٥).



الشكل (٥) إشارتي الدخل والخرج عند تردد 90KHz

تبين عملية القياس السابقة أنّ التردد الأعظمي الذي تمكنت شريحة Arduino من قراءته بشكل صحيح هو 90KHz وبمعدل 2 عينة لكل دور، أي معدل أخذ العينات الرقمية لشريحة Arduino حوالي 180 KSPS. وكذلك تم التأكد من القياسات التشابهية بالطريقة السابقة نفسها، بتطبيق إشارة تشابهية ذات مطال 5V وتردد 2.2KHz على أحد أقطاب المحول التشابهي الرقمي، وكانت النتائج كما في الشكل (٦).



الشكل (٦) إشارتي الدخل والخرج عند تردد 2.2KHz

تبين عملية القياس أنّ الشريحة تمكنت من تحصيل 4 عينات لكل دور من الإشارة ذات التردد 2.2Kz بالتالي فإن تردد أخذ العينات التشابهية لشريحة Arduino حوالي 8.8KSPS. نلاحظ أن نتائج بيئة المحاكاة قريبة جداً من نتائج القياسات العملية؛ بالتالي من الممكن الاستعانة بنتائج بيئة Proteus لإنجاز الدراسة البحثية.

٢- الحلول البرمجية المقترحة لزيادة تردد أخذ العينات لشريحة ARDUINO

٢-1- الإشارات الرقمية

تم استبدال تعليمات القراءة والكتابة الخاصة بلغة C لـ Arduino بتعليمات المسجلات الخاصة بالأقطاب مما يمكننا من القراءة أو الكتابة في مسجل كامل بتعليمية واحدة؛ أي ما يعادل ٨ أقطاب معاً وبأعلى سرعة ممكنة (إدخال

محتوى المسجل الذي يحتوي على القطب المراد قراءته إلى المعالج باستخدام تعليمة مسجلية)، وأظهر اختبار السرعة بالنسبة إلى الأقطاب الرقمية لشريحة Arduino Mega النتائج الآتية:

الجدول (٢) تردد أخذ العينات الرقمية لشريحة Arduino Mega2560 بلغة المسجلات

الشريحة	تردد أخذ العينات الرقمية
Arduino Mega	1.388 MHz

تبين القياسات العملية أنّ تردد أخذ العينات الرقمية أصبح 1.388MHz، أي تضاعف حوالي ٧,٨٣ ضعف، وأعلى تردد رقمي يمكن قياسه بحسب نايكويست هو الآتي:

$$F_m \leq \frac{1388888}{2} = 694.444 \text{ KHz} \quad (7)$$

لنقوم بتغيير برمجي بسيط حيث سنقرأ الأقطاب من دون استخدام حلقات تكرارية (for, while)، ونعيد عملية القياس السابقة وتظهر النتائج كما في الجدول (٣).

الجدول (٣) تردد أخذ العينات الرقمية لشريحة Mega2560 بلغة المسجلات دون حلقات تكرارية

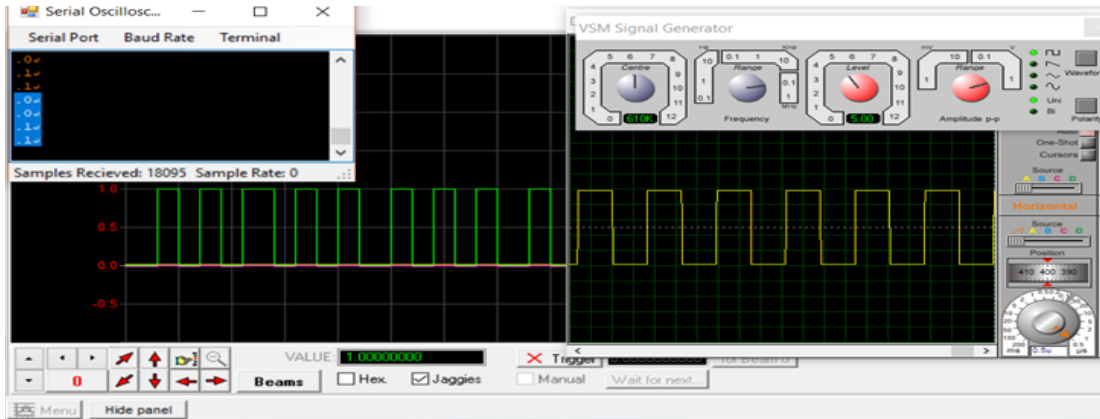
الشريحة	تردد أخذ العينات الرقمية دون استخدام الحلقات
Arduino Mega	2.5 MHz

نلاحظ أن تردد اخذ العينات أصبح 2.5MHz بالتالي أعلى تردد رقمي يمكن قياسه حسب شرط نايكويست هو:

$$F_m \leq \frac{2500000}{2} = 1.25 \text{ MHz} \quad (8)$$

لنتأكد من القياسات السابقة باستخدام برنامج proteus بتطبيق إشارة رقمية ذات تردد 610KHz كما في

الشكل (7).



الشكل (7) إشارتي الدخل والخرج عند تردد 610KHz

تبين من عملية القياس إن الشريحة تمكنت من تحصيل ٤ عينات لكل دور من الإشارة ذات التردد 610Kz بالتالي فإن تردد أخذ العينات الرقمية لشريحة Arduino حوالي 2.44MSPS. بالتالي إن استخدام تعليمات المسجلات الخاصة بالمتحكمات اختصر عدد دورات الساعة اللازمة لتنفيذ تعليمات قراءة الأقطاب الرقمية.

٢-٢- الإشارات التشابهيّة

سنستخدم تعليمات المسجلات كما سبق؛ ولكن ليس للقراءة لكن لتغيير إعدادات المسجل الخاص بالمحول التشابهي الرقمي، لأن التردد الخاص بالمحول يتم تقسيمه باستخدام معامل التقسيم (prescale factor) وتكون قيمة المعامل الافتراضية 128؛ وبما إن المذبذبة الخاصة بشريحة أردوينو هي 16MHz، بالتالي سيكون تردد العمل للمحول:

$$\frac{16\text{MHz}}{128} = 125\text{KHz} \quad (9)$$

ويحتاج المحول 13 دورة ساعة لكل عملية تحويل؛ بالتالي سيكون معدل التحويل كالآتي:

$$\frac{125\text{KHz}}{13} = 9615\text{Hz} \quad (10)$$

سنقوم بتقليل معامل التقسيم للحصول على سرعة أعلى للمحول، حيث سنغير محتوى المسجل ADCSRA لتغيير التردد الخاصة بالمحول، انظر الجدول (4) لمعرفة قيمة المعامل وتردد المحول المكافئ له [8]:

الجدول (4) معامل التقسيم وتردد عمل المحول للمتحكم atmega2560

معدل التحويل (KHz)	تردد عمل المحول (MHz)	ADPS2 ADPS1 ADPS0	معامل التقسيم
615	8	0 0 1	2
307	4	0 1 0	4
153	2	0 1 1	8
76.8	1	1 0 0	16
38.4	0.5	1 0 1	32
19.2	0.25	1 1 0	64
9.6	0.125	1 1 1	128

وضعنا قيمة المعامل 2 لنحصل على أعلى تردد عمل للمحول 8MHz، وكانت النتائج كما في الجدول (٥).

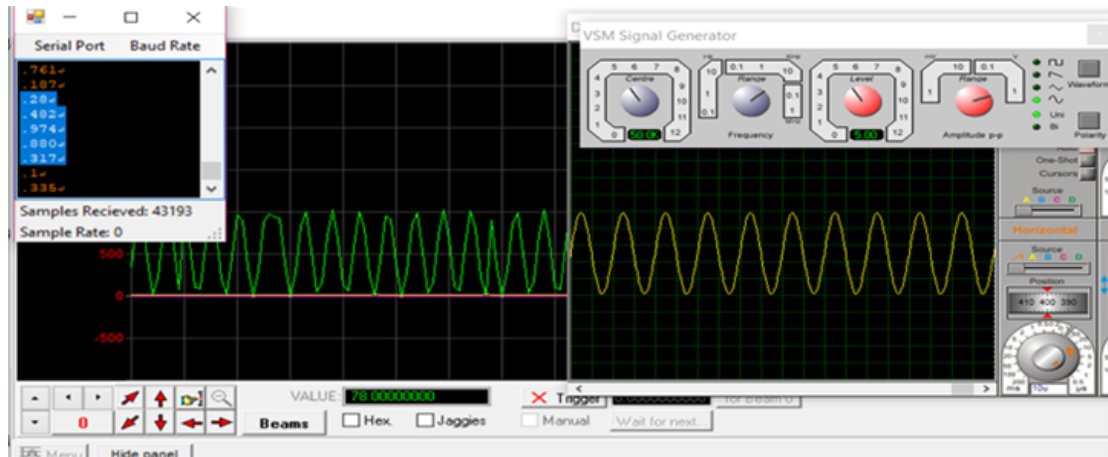
الجدول (٥) تردد أخذ العينات التشابهيّة للمتحكم ATMEGA2560 بلغة المسجلات

الشريحة	تردد أخذ العينات التشابهيّة
Arduino Mega	240.38 KHz

نلاحظ أن تردد أخذ العينات أصبح 240.38KHz، أي أكبر بـ 27 ضعفاً من تعليمة القراءة الخاصة بـ Arduino.

لنقم بالتأكد من القياسات السابقة باستخدام برنامج proteus بتطبيق إشارة تشابهيّة ذات مطال 5V بتردد

50KHz كما في الشكل (8).



الشكل (8) إشارتي الدخل والخرج عند تردد 50KHz

تبين من عملية القياس أن الشريحة تمكنت من تحصيل 5 عينات لكل دور من الإشارة ذات التردد 50Kz بالتالي فإن تردد أخذ العينات التشابهيية لشريحة Arduino حوالي 250SPS. مما سبق تبين أن لغة البرمجة عالية المستوى الخاصة بشرائح Arduino تقلل من قدرة شرائح المتحكمات للتعامل مع الإشارات، ويمكن حل هذه المشكلة برمجياً كما أسلفنا ذكره سابقاً. لنقم بتجربة بسيطة ونعدّل طريقة معالجة البيانات في المتحكم حيث سنقرأ قيمة المسجلات من قبل المعالج من دون تخزينها في الذاكرة.

سنختبر ذلك بالنسبة إلى الإشارة الرقمية والتشابهيية فتظهر النتائج كما في الجدول (6).

الجدول (6) تردد أخذ العينات الرقمية والتشابهيية للمتكم ATMEGA2560 من دون التخزين في الذاكرة

الشريحة	تردد أخذ العينات الرقمية	تردد أخذ العينات التشابهيية
Arduino Mega	12.5 MHz	257.731KHz

بالنسبة إلى القراءة الرقمية تمكنت الشريحة من قراءة مسجل كامل بطول 8-bit بتردد 12.5MHz، أما بالنسبة إلى القراءات التشابهيية فأعلى تردد أخذ عينات 257.731KHz وهذه الترددات الأعظمية التي لا يمكن للشريحة تجاوزها.

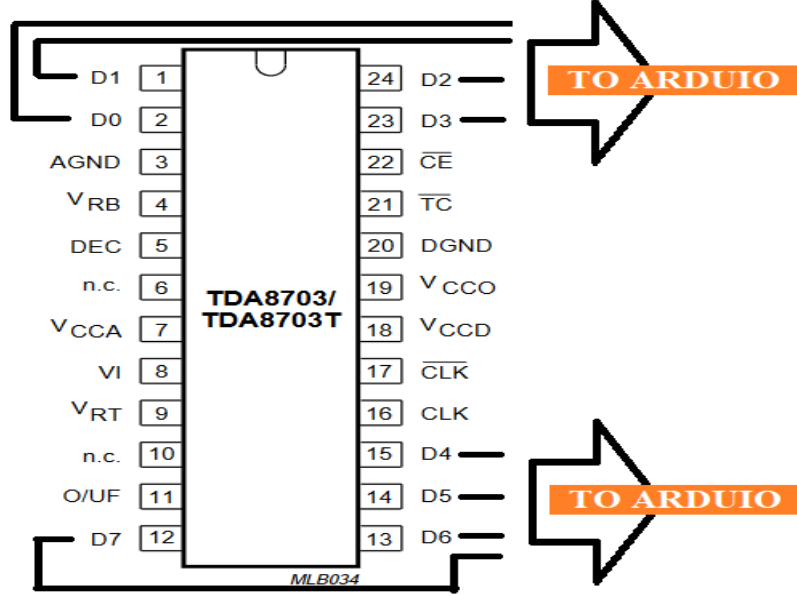
٣- الحلول التقنية المقترحة لتحسين إمكانيات المتحكمات الصغرية

سنقدم حلاً تقنيّة لزيادة سرعة التحويل التشابهي الرقمي ودقته؛ لتتمكن المتحكمات من التعامل مع الإشارات بفعالية أكثر.

3-1- زيادة سرعة التحويل

سنقوم باستخدام محول تشابهي رقمي خارجي، وجدنا سابقاً أنّ بإمكان شريحة المتحكم المستخدم في شرائح Arduino mega قراءة مجموعة أقطاب رقمية معاً (port)، بتردد أخذ عينات حتى 2.5MHz، بالتالي سنتمكن من قراءة n قطب لمحول تشابهي رقمي خارجي بهذا التردد، مما يعطي للمتكم سرعة قراءة عالية جداً للإشارات التشابهيية تفوق أعلى سرعة قراءة ممكنة للمحول التشابهي الرقمي الداخلي ب 10 أضعاف.

سنستخدم المحول التشابهي الرقمي TDA8703 (8 bits 40 MHz)، حيث تبلغ قيمة التردد الأعظمي لهذا المحول 40MHz، بالطبع عند توصيل خرج المحول الشكل (9) بشريحة المتحكم لن نتمكن من قراءة أكثر من 2.5MSPS لأنه أعلى تردد يستطيع المتحكم قراءته باستخدام المنافذ الرقمية.



الشكل (9) أقطاب الخرج للمحول التشابهي الرقمي 8-bit TDA8703

تعطى الدقة كما يلي:

$$1LSB = \frac{2^n - 1}{V_{ref}} = \frac{5}{2^8 - 1} = 0.019V \quad (11)$$

نلاحظ أن الدقة ستكون أقل من دقة المحول الداخلي للمتحكم؛ لأن عدد أقطاب الخرج للمحول 8 أقطاب، ولا نستطيع استخدام محول ب 10 أقطاب لأن طول المسجلات الخاصة بالأقطاب في المتحكم atmega2560 هي 8bit.

3-2- زيادة السرعة الفائقة

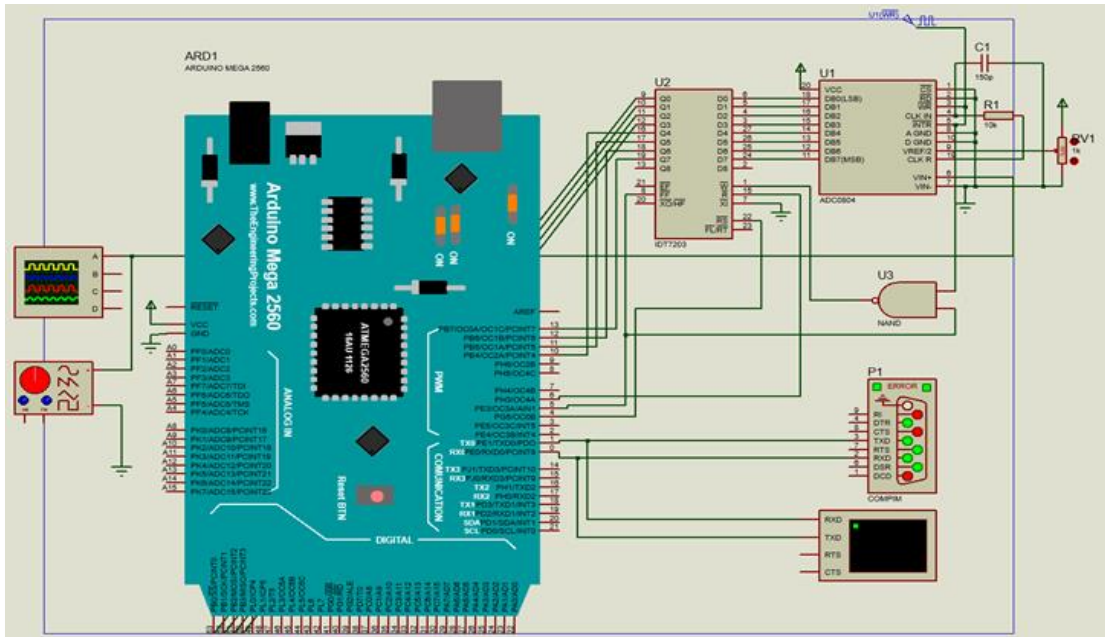
علينا أن نجد طريقة بحيث نأخذ العينات بتردد سريع من المحول الخارجي، وتجميعها لفترة بحيث يتمكن المتحكم من أخذها بسرعة أبطأ (أي تقسيم التردد الحقيقي للإشارة)، سنقترح التعديل كما يوضح الشكل (10) لحل المشكلة:



الشكل (10) المخطط الصندوقي لطريقة حل مشكلة فرق السرعة بين المتحكم والمحول الخارجي

بإضافة مرحلة وسيطة بين المتحكم والمحول التشابهي الرقمي تقوم بدور الذاكرة كاش في الحاسب التقليدي؛ حيث تقوم بأخذ العينات بتردد سريع من المحول وحفظها ثم تقديمها للمتحكم بسرعة أبطأ، بالتالي سنتمكن من حل فرق السرعة بين المتحكم والمحول الخارجي السريع ولكن سيكون ذلك حساب قراءة العينات في الزمن الحقيقي.

توضح الدارة في الشكل (11) طريقة توصيل الذاكرة (IDT7203) كمرحلة وسيطة بين شريحة Arduino mega ميغا والمحول التناظري الرقمي ADC0804



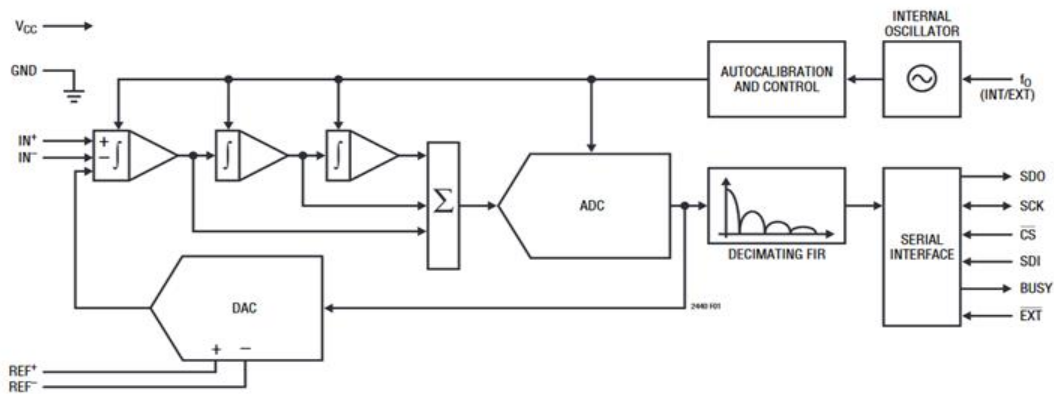
الشكل (11) ربط الذاكرة IDT7203 مع المحول ADC0804 وشريحة Arduino

بالاعتماد على الحلول السابقة سنتمكن من تحقيق سرعات تحصيل معطيات عالية جداً، مما يمهد الطريق لتصنيع أجهزة قياس عالية السرعة باستخدام المتحكمات رخيصة الثمن ومنخفضة الإمكانيات، ولكن لم ينته الأمر هنا لأننا نحتاج لزيادة الدقة، بالإضافة إلى السرعة لذلك سنقدم حلاً لزيادة دقة التحويل التناظري الرقمي.

3-3- زيادة دقة التحويل

سنستخدم محولات تشابهية رقمية خارجية من نوع sigma delta حيث تتميز هذه المحولات بدقة التحويل العالية (16 bit-24 bit) ويكون الخرج بشكل تسلسلي يعتمد بروتوكول I2C حيث سنستخدم المحول LTC2440 [11].

يتميز هذا المحول بإمكانية تحديد تردد عينات الخرج من خلال القطب sdi الذي يمكننا من الاختيار بين 6.9Hz أو 880Hz، ويحتوي في جزء الخرج على مرشح لتقليل الضجيج وأيضاً يتميز بوجود مذبذب داخلي كما في الشكل (12).



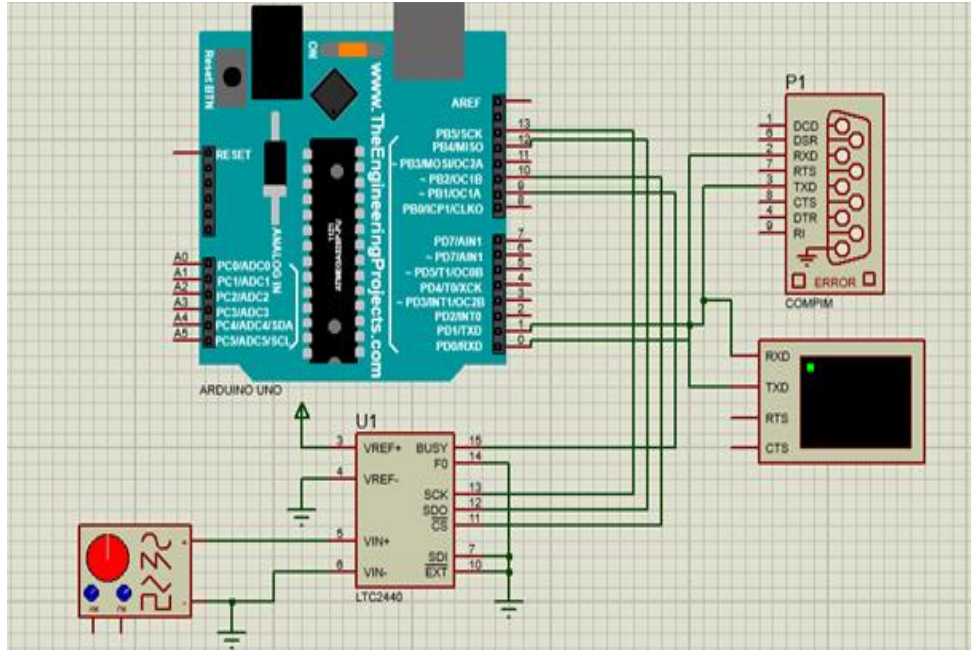
الشكل (12) البنية الداخلية للمحول LTC2440

لنقوم بحساب دقة الخطوة، مع العلم أن مجال الدخل للمحول $[-2.5V, 2.5V]$:

$$LSB = \frac{2 * V_{ref}}{2^N - 1} = \frac{2 * 2.5}{2^{24} - 1} = 0.00000298 V \quad (12)$$

سيضيف هذا المحول إمكانات مميزة للمتحكمات من ناحية دقة القياس، لنقم بتصميم دائرة لاختبار المحول مع

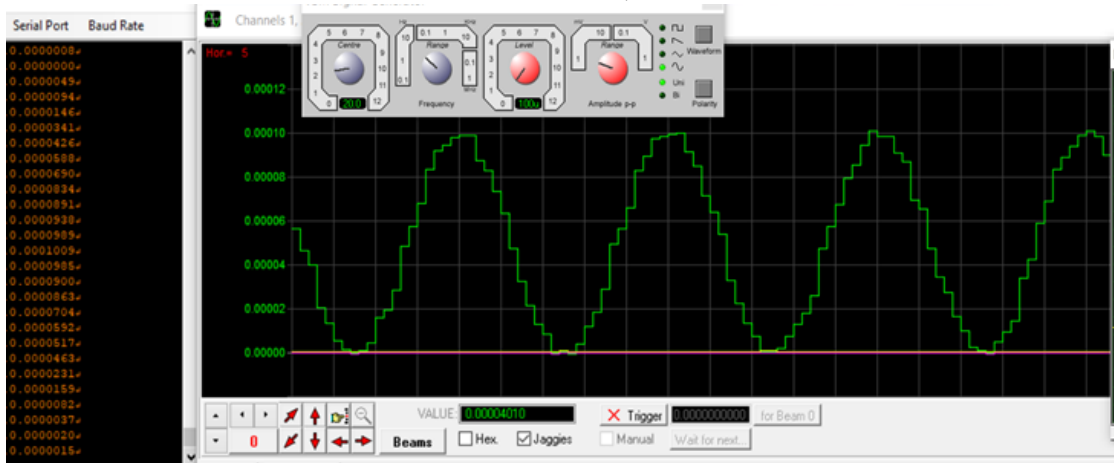
المتحكم ونختبر دقة القياس كما في الشكل (١٣).



الشكل (13) ربط المحول LTC2440 مع ARDUINO

نطبق جهداً منخفضاً على القطب sdi ليصبح تردد خرج المحول 880Hz، ونقوم بتطبيق إشارة جيبية بمطال

100uV وتردد 20Hz لنرى إشارة الخرج كما في الشكل (١٤).



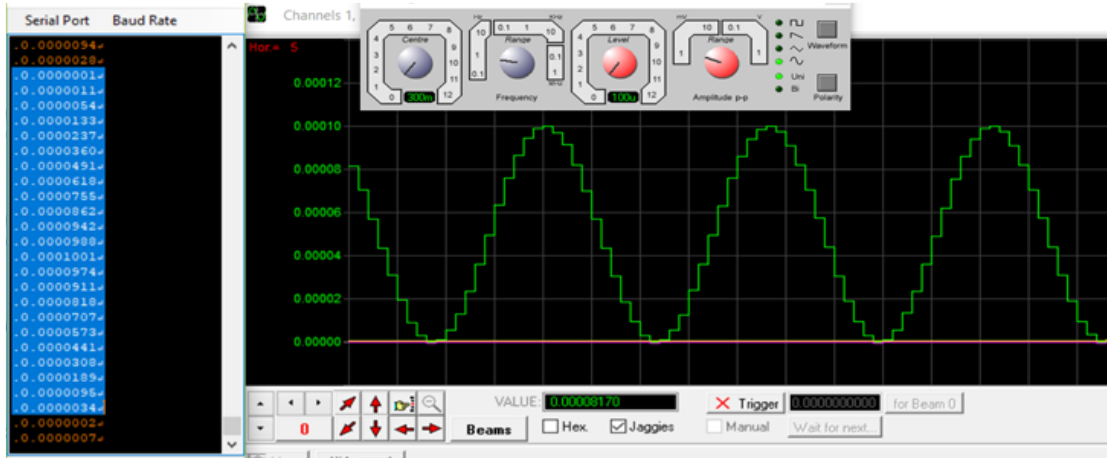
الشكل (14) إشارة الخرج للمحول LTC2440 عند معدل 880Hz

تبين القياسات السابقة أن المحول تمكن من قراءة 27 عينة من إشارة ذات تردد 20Hz أي بمعدل أخذ عينات

يعادل 540SPS.

نطبق جهداً مرتفعاً على القطب sdi ليصبح تردد خرج المحول 6.9Hz، ونقوم بتطبيق إشارة جيبية بمطال

100uV وتردد 1Hz لنرى إشارة الخرج كما في الشكل (١٥).

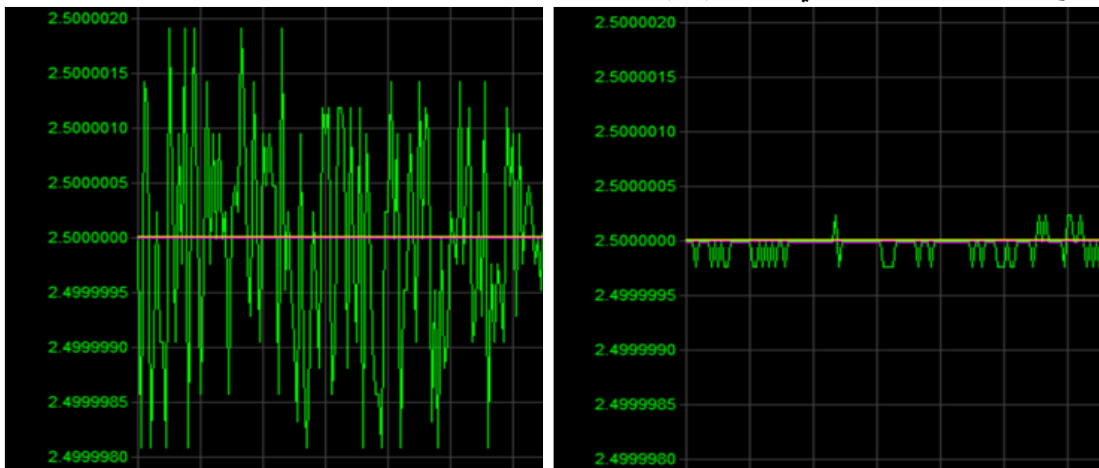


الشكل (15) إشارة الخرج للمحول LTC2440 عند معدل 6.9Hz

تبين القياسات أن المحول تمكن من قراءة 23 عينة من إشارة ذات تردد 0.3Hz أي بمعدل أخذ عينات يعادل 6.9SPS.

٤-دراسة الضجيج:

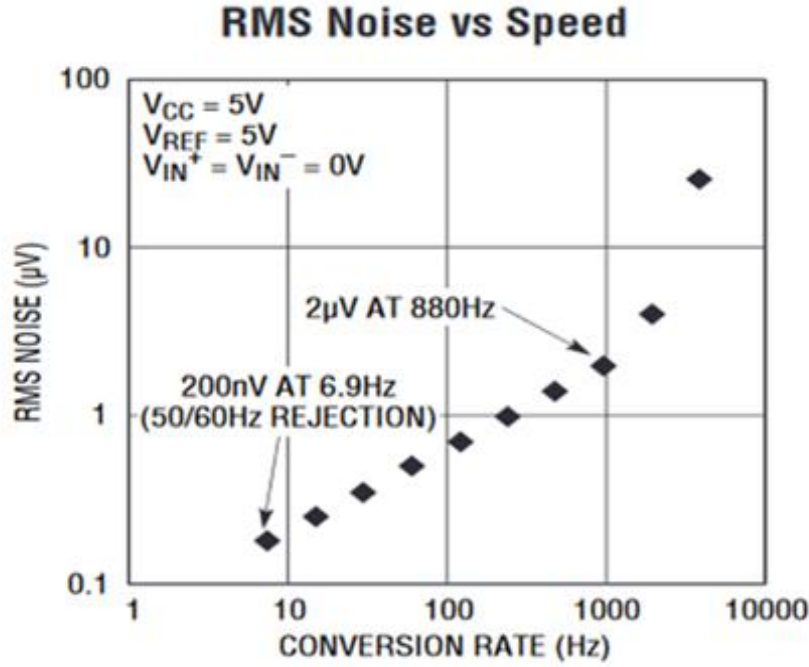
سنقوم بدراسة الضجيج الناتج عن قراءة المحول السابق، حيث نطبق جهداً مستمراً بمطال 2.5V وندرس إشارة الخرج للمحول بالحالتين كما في الشكل (١٦).



الشكل (b-16) الضجيج عند معدل خرج 880Hz

الشكل (a-16) الضجيج عند معدل خرج 6.9Hz

الضجيج يختلف بحسب تردد الخرج للمحول، تبين أنه يصل حتى 2uV في حال استخدام معدل 880Hz و 200nV عند معدل 6.9Hz، ويمكننا تقليل الضجيج عند المعدل العالي باستخدام طريقة أخذ المتوسط لكل مجموعة عينات معاً، ولكن سيقبل تردد الخرج بحسب عدد العينات. ويوضح المنحني الشكل (17) علاقة تردد الخرج بالضجيج.



الشكل (17) علاقة تردد الخرج بالضجيج

الاستنتاجات والتوصيات:

وجدنا أن إضافة مرحلة وسيطة بين المحول التناظري الرقمي الخارجي عالي السرعة والمتحكم منخفض السرعة تؤدي لزيادة كفاءة المتحكم بالنسبة إلى التعامل مع الإشارات الكهربائية بشكل كبير، وكذلك وجدنا أن تخزين القيم في الذاكرة في أثناء المعالجة يستهلك وقتاً كبيراً نسبياً، ولحل هذه المشكلة اقترحنا تصميم محول تناظري رقمي يمتلك ذاكرة FIFO مضمنة داخله، حيث صممنا شريحة إلكترونية تحتوي محولاً تناظرياً رقمياً عالي السرعة مع ذاكرة FIFO الشكل (11)، لتصبح هذه الشريحة وسيطاً لاتصال المتحكم مع الإشارات التناظرية والرقمية الخارجية عالية السرعة والدقة.

في النهاية سنضع عدة نقاط دليوية يستخدمها الباحثون في هذا المجال:

- عند استخدام شرائح Arduino (ذات متحكمات AVR) تسمح البيئة البرمجية Arduino IDE بتعدد أخذ عينات رقمية أعظمي 177KHz، وتعدد أخذ عينات تناظري حوالي 8.9KHz.
- عند استخدام لغة المسجلات نستطيع الحصول على تردد أخذ عينات رقمية 1.38MHz، وعند تغيير معامل التقسيم الخاص بالمحول التناظري الرقمي الداخلي للمتحكم من 128 إلى 2 أصبح تردد أخذ العينات التناظرية 200KHz.
- استخدام الحلقة التكرارية for قلل تردد أخذ العينات الرقمية إلى 2.5MHz، وكذلك تردد أخذ العينات التناظرية إلى 240KHz.
- عملية تخزين القيمة في الذاكرة SRAM يقلل كثيراً من تردد أخذ العينات بسبب زيادة دورات الساعة اللازمة للكتابة، وعند الاكتفاء بالقراءة من قبل المعالج دون الكتابة في الذاكرة نحصل على تردد أخذ عينات رقمية حتى 12.5MHz، وتعدد أخذ عينات تناظرية حتى 257KHz.

- من الممكن استخدام شرائح Arduino والمتحكمات لتصميم رواسم الإشارة والمحللات الطيفية بسرعة ودقة عالية جداً باستخدام الحل الذي اقترحنه في هذه المقالة بإضافة مرحلة تخزين بين المتحكم والمحول التشابهي الرقمي الخارجي السريع أو محولات Sigma-Delta.
- الضجيج في المحولات التشابهيية الرقمية sigma-delta ناتج عن الاتصال التسلسلي بين المتحكم والمحول وليس خطأ من المحول، ويزداد الضجيج بزيادة تردد عينات الخرج للمحول.

المراجع

- [1] Elfasi, A., Shawesh, M. A., Shanab, W. T., & Thabet, A. K. *Oscilloscope using Arduino interface LabVIEW*. (2017) International Conference on Green Energy Conversion Systems (GECS). doi:10.1109/gecs.2017.8066129.
- [2] Silva, S., Soares, S., Valente, A., & Marcelino, S. T. “*Digital sound processing using arduino and MATLAB*”. (2015) Science and Information Conference (SAI). doi:10.1109/sai.2015.7237295
- [3] Ishtiak Ahmed Karim. “*A low cost portable oscilloscope based on Arduino and GLCD*”. (2014) International Conference on Informatics, Electronics & Vision (ICIEV). doi:10.1109/iciev.2014.6850686.
- [4] Pittini, R., Zhang, Z., & Andersen, M. A. E. “*An interface board for developing control loops in power electronics based on microcontrollers and DSPs Cores -Arduino /ChipKit /dsPIC /DSP /TI Piccolo*”. (2013) IEEE 14th Workshop on Control and Modeling for Power Electronics (COMPEL). doi:10.1109/compel.2013.6626423.
- [5] Pereira, N. S. A. “*Measuring the RC time constant with Arduino*”. (2016) Physics Education, 51(6), 065007. doi:10.1088/0031-9120/51/6/065007.
- [6] Ali, A. S., Zanzinger, Z., Debose, D., & Stephens, B. “*Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection*”. (2016) Building and Environment, 100, 114–126. doi:10.1016/j.buildenv.2016.02.010.
- [7] Atmel, “*ATmega640/V-1280/V-1281/V-2560/V-2561/V[DATASHEET]-page5*” https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf, (2014).
- [8] Atmel, “*ATmega640/V-1280/V-1281/V-2560/V-2561/V[DATASHEET]-page285*” https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf, (2014).
- [9] ADC “*TDA87038-bit high-speed analog-to-digital converter [DATASHEET]-page4*” <https://pdf1.alldatasheet.com/datasheet-pdf/view/19534/PHILIPS/TDA8703.html>, (1996).
- [10] ADC “*IDT7203/7204/7205/7206/7207/7208 CMOS ASYNCHRONOUS FIFO [DATASHEET]-page1*” <https://www.idt.com/document/dst/7203-7208-datasheet>, (2017).
- [11] ADC “*LTC2440 24-Bit High Speed Differential $\Delta\Sigma$ ADC with Selectable Speed/Resolution [DATASHEET]-page9*” <https://www.analog.com/media/en/technical-documentation/data-sheets/2440fe.pdf>, (2017).
- [12] Muller, L., Mohammed, M., & Kimball, J. W. “*Using the Arduino Uno to teach digital control of power electronics*”. (2015) IEEE 16th Workshop on Control and Modeling for Power Electronics (COMPEL). doi:10.1109/compel.2015.7236487.
- [13] Khan, S. A., Agarwala, A. K., Shahani, D. T., & Alam, M. M. “*Advance Oscilloscope Triggering*”. (2007). IEEE Transactions on Instrumentation and Measurement, 56(3), 944–953. doi:10.1109/tim.2007.894198.

[14] Khan, S. A., Agrwala, A. K., & Shahani, D. T. (n.d.). “ *Implementation of Advance Oscilloscope Triggering Scheme on FPGA*“(2005) IEEE Instrumentation and Measurement Technology Conference Proceedings. doi:10.1109/imtc.2005.1604147.