بسم الله الرحمن الرحيم

تفاصيل الاتصال التسلسلى بين الكومبيوتر والميكروكونترولر والمحول MAX232

من تأليف وكتابة

م/ فتح الله عبد العزيز

F.Abdelaziz

تجميع وإخراج الملف

م/ كريم عادل

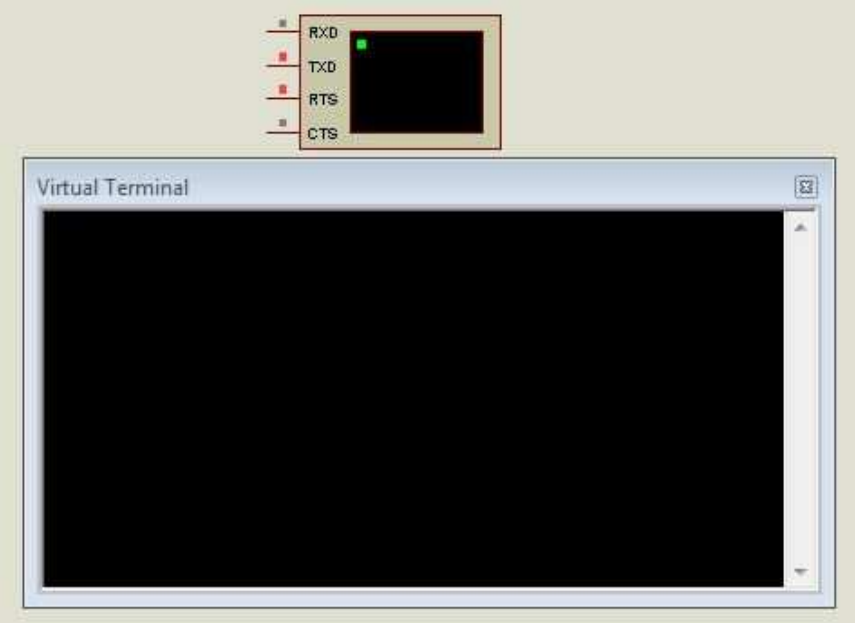
King5star

تفاصيل الاتصال التسلسلي بين الكومبيوتر والميكروكونترولر والمحول MAX232

أولا : الوحدة الطرفية التخيلية VIRTUAL TERMINAL فى برنامج بروتيس Proteus

مقدمة :

تمكنك محاكاة الوحدة الطرفية من استخدام لوحة المفاتيح وشاشة الكومبيوتر الشخصى PC لإرسال واستقبال البيانات التسلسلية الغير متزامنة نوع RS232 إلى ومن نظام يحاكى المعالج الدقيق .
وهى مفيدة بشكل خاص في تصحيح الأخطاء debugging حيث يمكنك استخدامها لعرض رسائل تتبع التصحيح والتي يتم إنشاؤها بواسطة البرنامج الذي تقوم بتطويره .



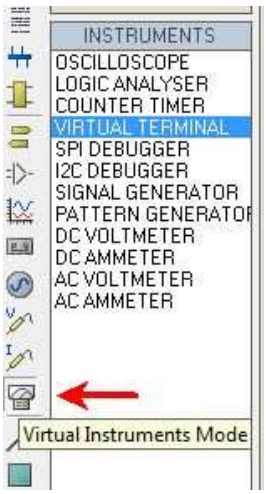
خواص الوحدة الطرفية التخيلية :

- **الوحدة ثنائية الاتجاه بالكامل bi-directional :تعرض البيانات المستقبلة كحروف نوع أسكى ASCII characters بينما ترسل الضغطات على المفاتيح كبيانات متسلسلة نوع أسكى .**
- **الربط interface التسلسلى للبيانات يتم بطريقة بسيطة بواسطة سلكين (خطين) : الخط RXD للبيانات المستقبلة والخط TXD للبيانات المرسلة .**
- **معدل البود (يعبر عن سرعة تبادل البيانات بعدد النبضات فى الثانية (Baud rate من ١١٠ إلى ٥٧،٦٠٠ .**
- **عدد خانات البيانات ٧ or 8 data bits .**
- **التكافؤ أو المساواة parity Odd, even , no" فردى – زوجى – بدون" .**
- **عدد خانات الإيقاف (النهاية) ٠، ١ ، ٢ stop bits .**
- **المصافحة البرمجية handshaking XON/XOFF software بالأضافة إلى المصافحة بالأجهزة handshaking hardware .**

استخدام الوحدة الطرفية *Terminal Using the Virtual:*

لإرفاق الوحدة الطرفية لنظام المحاكاة المستهدف :

١ـ اختار أيقونة المقاييس المتعددة *Multimeter* ومنها إلتقط (إجلب) الوحدة الطرفية التخيلية VIRTUAL TERMINAL وضعها فى المخطط .



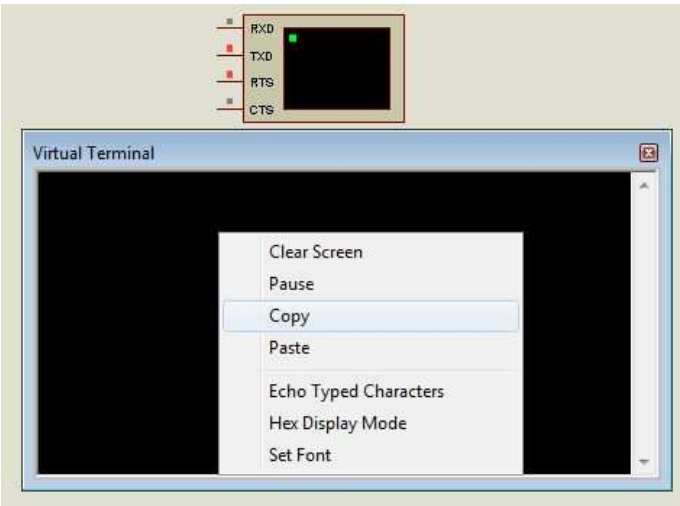
٢ـ وصل بأسلاك (بخطوط) كل من الأطراف RX و TX إلى كل من خطوط الإرسال والاستقبال للنظام المختبر . RX هو الدخل و TX هو الخرج .

٣ـ إذا كان النظام يستخدم المصافحة بالاجهزة hardware handshaking وصل الأطراف RTS و CTS بالخطوط الملائمة . RTSRequest to Send " " هو الخرج ويشير (يعطى إشارة) على أن الوحدة الطرفية جاهزة لاستقبال البيانات ، بينما CTS " Clear to Send " هو الدخل والذى يجب أن يكون مرتفعا (أو غيرموصل ويسمى عائم floating) قبل أن تقوم الوحدة الطرفية بالارسال .

٤ـ يتم تحرير عنصر الوحدة الطرفية لاختيار معدل البود rate baud وطول الكلمة و التكافؤ وباقى عناصرالتحكم فى سريان البيانات .

٥ـ إبدأ المحاكاة بالطريقة المعتادة . تعرض الوحدة الطرفية البيانات القادمة بمجرد ما تستقبلها ، لإرسال حرف إلى النظام تأكد وجود الوحدة الطرفية وانها فعالة عندئذ اكتب النص المطلوب بلوح مفاتيح الكومبيوتر الشخصى .

٦ـ بمجرد بدء المحاكاة تتاح وظائف إظافية عن طريق قوائم منبثقة والتى يمكن عرضها بالنقر بالزر الأيمن على النافذة المنبثقة . هذه القائمة تمكن من إيقاف العرض مؤقتا pause ونسخ ولصق النص إلى ومن الحافظة clipboard .

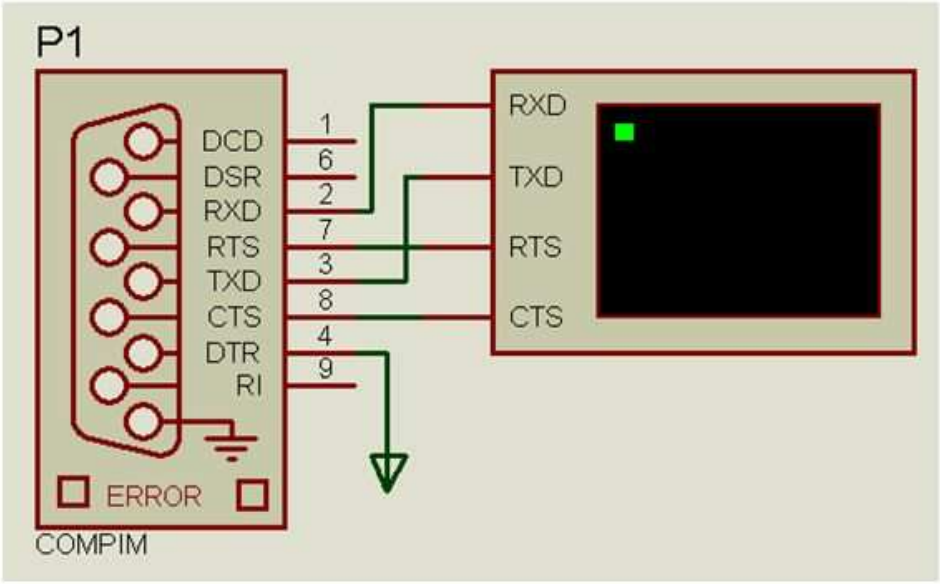


ملاحظات :

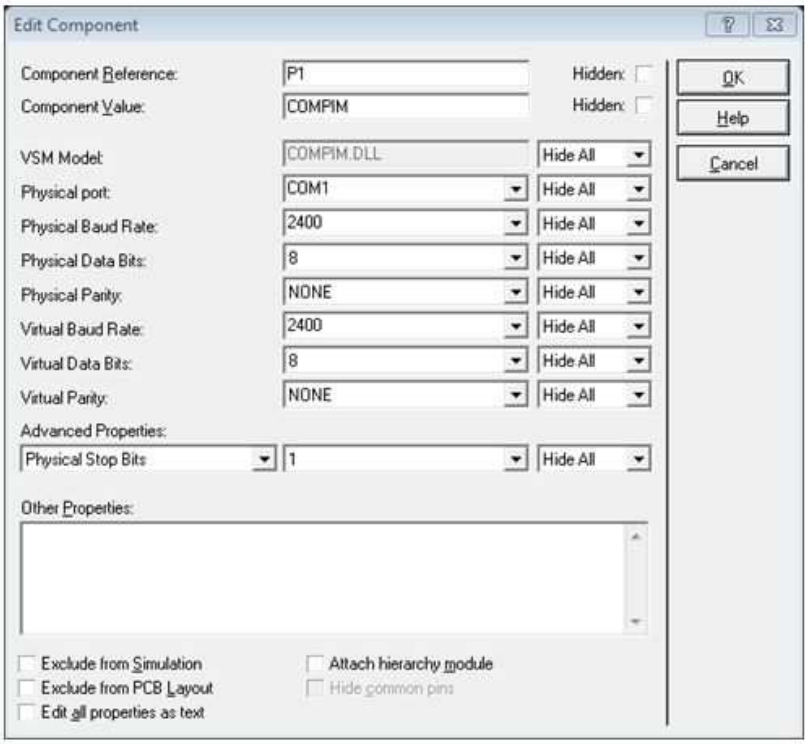
- **الوحدة الطرفية التخيلية هى نموذج رقمي محض وعلى هذا النحو لا تتطلب أي مستويات جهد خاص على أطرافها . الافتراض هو أنها سوف يتم توصيلها إلى النماذج الرقمية الأخرى بدلا من جانب الإخراج لأجهزة التعزيز والتحويل مثل MAX232.**
هذا سبب جزئى من عدم تقديم نماذج للدائرة المتكاملة MAX232 أو ما يماثلها والسبب الآخر هو أن توجيه الاشارات من خلال هذه النماذج سوف تتكبد تكلفة حسابية مرتفعة فى المحاكاة التماثلية دون الاستفادة على الإطلاق.
- **أطراف RX و TX فعالة فى الحالة المرتفعة . لذلك فإن حالة التسكع idling state تكون مرتفعة وخانة البدء start bit تكون منخفضة وخانة الإيقاف stop bit تكون مرتفعة . تظهر خانات البيانات كمنطق مرتفع للقيمة '١' ومنطق منخفض للقيمة '٠' . هذا متوافق مباشرة مع الاتصال التسلسلى UARTs الموجودة فى كثير من الميكروكونترولر ، وأيضا مع UARTs الخارجية مثل ٦٨٥٠ و ٨٢٥٠.**
- **الأطراف RTS و CTS أيضا فعالة فى الحالة المرتفعة .**

- **تدريب : إرسال نص .**

يتم إرسال سكريبت (سيناريو) النص حرف فى كل مرة بعد البدء ، ما لم CTS غير فعال .

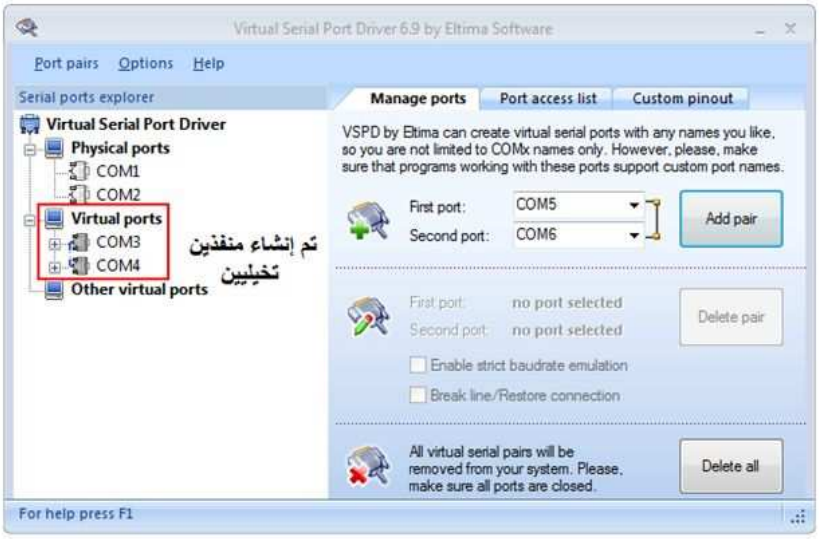


فى هذا التصميم يتم توصيل العنصر COM (port Physical Interface Model COMPIM) بالعنصر VTERM ليعطى مثال بسيط لإظهار طريقة استخدام COMPIM لإحضار بيانات العالم الحقيقى إلى عالم المحاكاة التخيلية .
للتجربة بهذه العينة سوف تحتاج إلى توصيل وحدة طرفية نوع RS232 أو كمبيوتر ليعمل عملها (مثال ذلك تشغيل برنامج HyperTerminal الذى يأتى ضمن نظام تشغيل النوافذ) أو مودم إلى منفذ COM خالى فى الكومبيوتر .
يتم تحرير العنصر COMPIM (بوضع المؤشر عليه والضغط على المفتاحين CTRL-E أو بالطرق المعروفة) ويتم ضبط منفذ COM ومعدل البود الطبيعى لعمل موائمة بين الجهاز المرتبط بالكومبيوتر . الضبط الافتراضى للمنفذ COM1 هو ٢٤٠٠ baud .



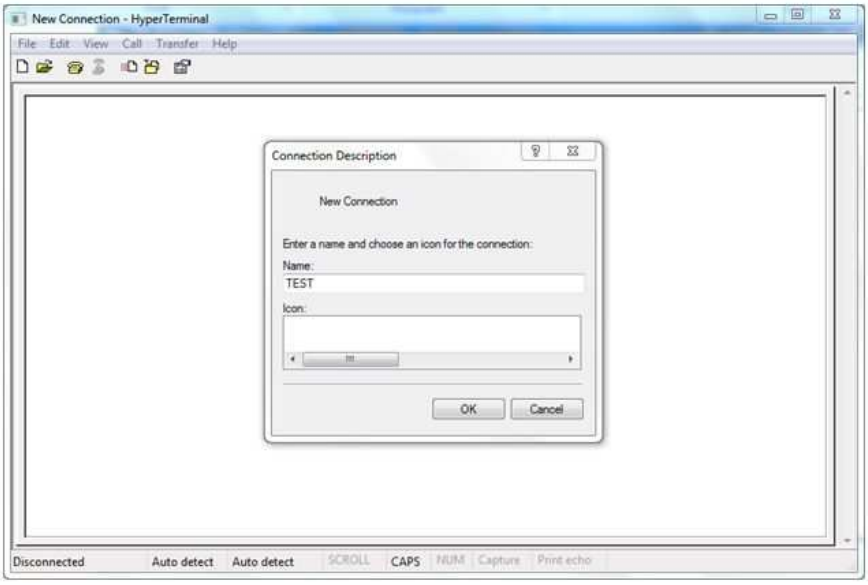
الآن يمكنك الضغط على زر play . الحروف التى تكتب على VTERM سوف ترسل إلى الجهاز التسلسلى الموجود بالمنفذ COM ، والحروف المستقبلة منه تظهر عائدة ألى VTERM . فى التطبيقات الحقيقية سوف يتم استبدال VTERM بالوحدة المركزية CPU أو بجهاز UART بحيث يكون النظام المحاكى يتفاعل مع الاجهزة hardware للعالم الحقيقى.

هذه الأداة تحتاج منفذ COM و أن RS232 بالكومبيوتر يحتاج منفذCOM آخر لكي يتم التخاطبينهما . يمكن تنفيذ ذلك برمجيا عن طريق البرنامج المعروف باسم Driver Virtual Serial Ports (درايفر المنافذ المتسلسلة التخيلية) . بعد تنصيب البرنامج وتشغيله نقوم بإنشاء منفذين متسلسلين تخيليين يمكن عن طريقهما تحقيق الاتصال المطلوب كما فى الشكل حيث يتم إضافة زوج Add pair من المنافذ التخيلية ولتكن COM3 و COM4 .



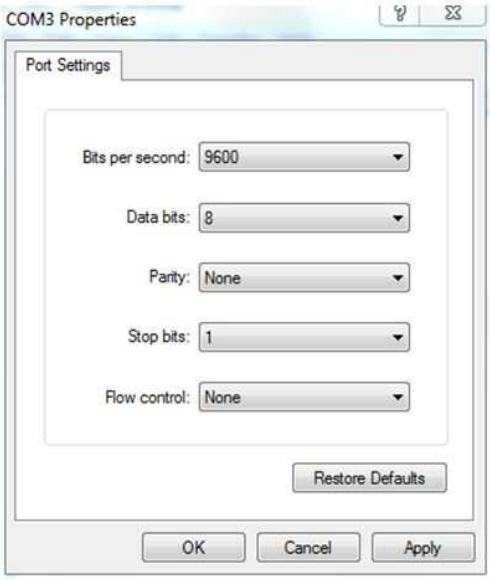
أحد المنفذين يمثل الكومبيوتر وليكن COM3 : ويتم تحديد ذلك فى برنامج HyperTerminal كما يلى :

- **تشغيل برنامج HyperTerminal وتسمية للتوصيل وليكن TEST**

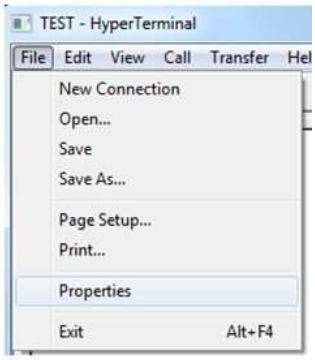


- **يلى ذلك تحديد المنفذ**

- ثم تحديد خصائص الاتصال

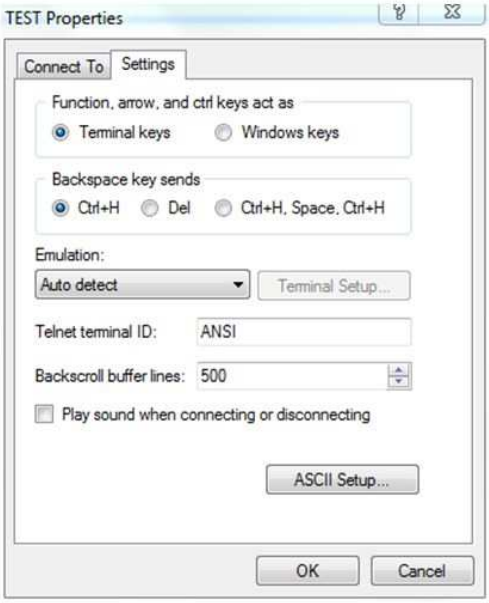


- لتحديد باقى الخواص من القائمة File اختار الخواص Properties

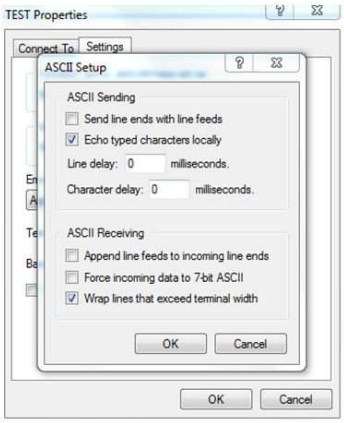


تظهر لك قائمة الخواص أنقر على مفتاح تحديدات Settings

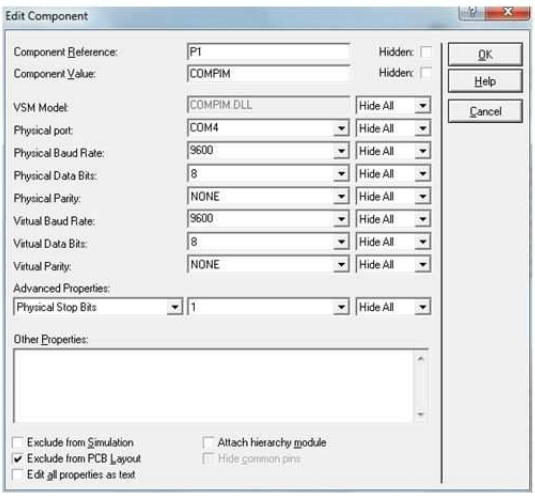
ثم انقر على مفتاح ASCII Setup



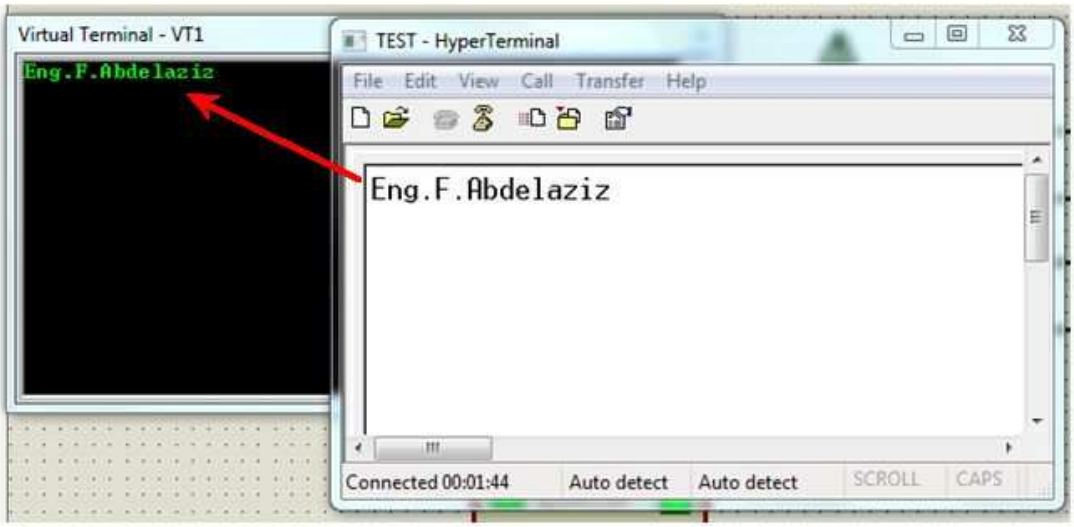
واختار (ضع علامة) Echo typed .. ليظهر (يعرض) صدى لما تكتبه محليا علاوة على الاستجابة الخارجية الناتجة من الاتصال .



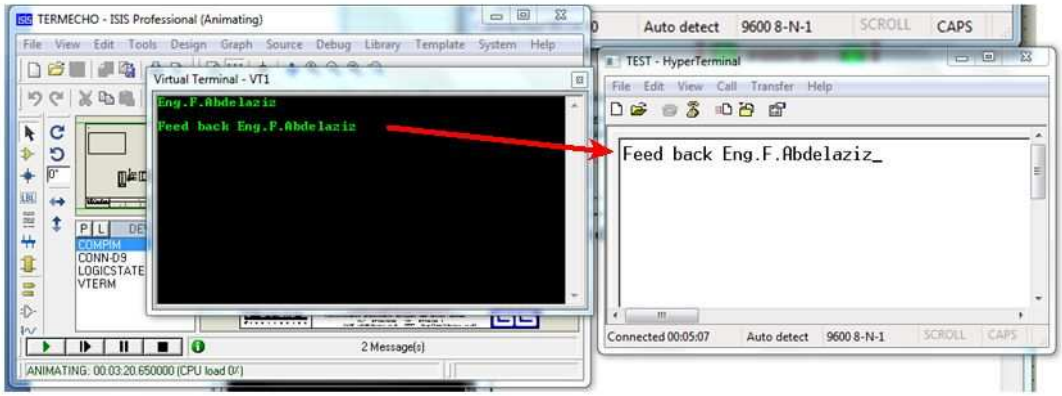
المنفذ الآخر COM4 ويمثل برنامج المحاكاة (الوحدة الطرفية): ويتم ذلك بتحرير خواص الوحدة الطرفية للتتلائم مع المنفذ COM3 حتى يتم الاتصال بينهما كما فى الشكل :



شغل برنامج المحاكاة وتأكد من تمام الاتصال بكتابة نص على نافذة برنامج HyperTerminal فتظهر لك النتيجة على الوحدة الطرقية



وبالعكس اكتب نص على الوحدة الطرفية فتظهر لك النتيجة على نافذة برنامج HyperTerminal

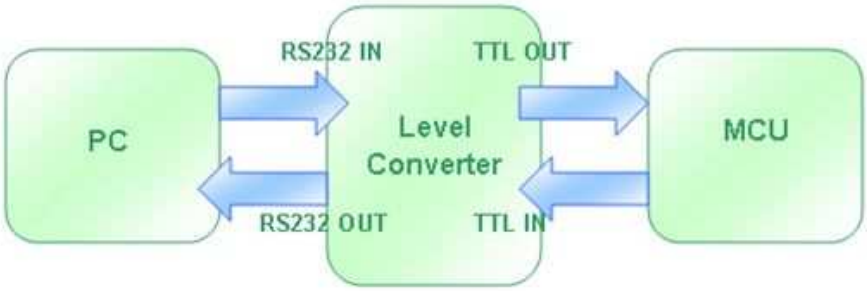


الربط بين الميكروكونترولر والكومبيوتر عن طريق المنفذ التسلسلى Port RS232-Serial والمحول MAX232

الربط بين الميكروكونترولر والكومبيوتر عن طريق المنفذ التسلسلى Port RS232-Serial

الرابط المرجع :
/http://extremeelectronics.co.in/avr-tutorials/rs232-communication-the-level-conversion

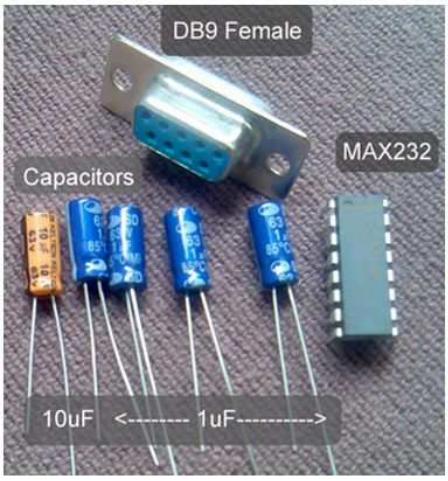
مستوى الإشارات المستخدمة فى الاتصالات التسلسلية مختلفة عن الإشارات المنطقية العادية ، لذلك للربط بين المنفذ التسلسلى للكومبيوتر والميكروكونترولر نحتاج إلى "محول مستويات" "Level converter" .
والهدف هو عمل هذا المحول .



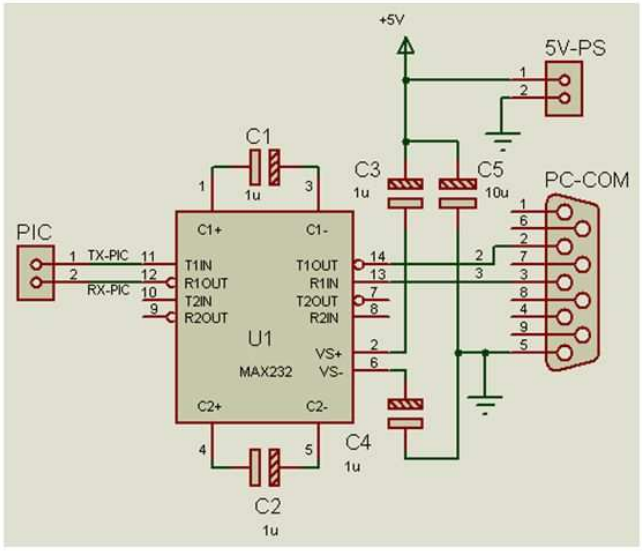
نظرا لأن الاتصال التسلسلى RS232 شائع الاستخدام يوجد دائرة متكاملة مصممة خصيصا لهذا الغرض وهى رقم *MAX232* . باستخدام مكثفات للشحن تقوم بتوليد جهود عالية موجبة (V١٢) وسالبة (-V١٢) .

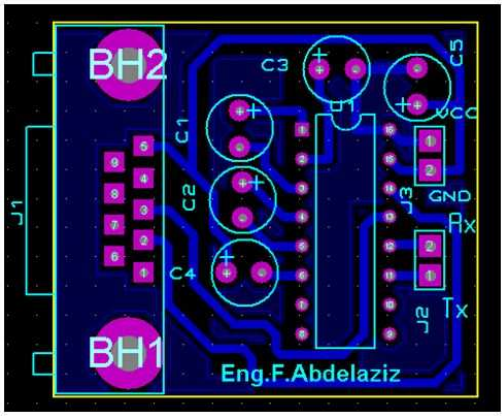
طريقة العمل :
المكونات المطلوبة

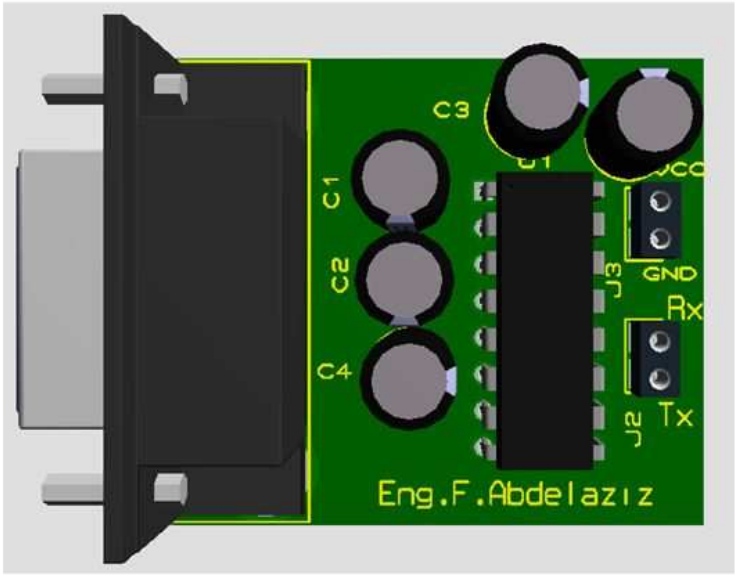
S.No	Item	Value	Qty
1	MAX232 IC		1
2	Capacitors	1uF	4
3		10uF	1
4	DB9 Female Connector		1
5	General Purpose PCB		1
6	Some Wires	-	-



الدائرة الكهربية :







تجميع الدائرة :

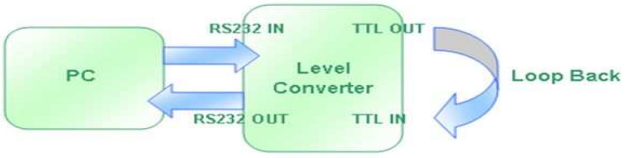
- **جمع المكونات على لوحة مطبوعة وفق الدائرة الكهربية . يتم أخذ سلكين يوصلان للتغذية(V٥) وسلكين يوصلان للخطين RX/TX بالميكروكونترولر .**
- **وصل السوكيت DB9 female بأسلاك طويلة للسماح بالتوصيل بالكومبيوتر ولكن يجب عدم زيادة الطول ويمكن أن يكون فى حدود واح ونصف متر إلى مترين .**
- **يجب الحذر عن توصيل الأسلاك بالأطراف المناسبة بالسوكيت DB9 وللمساعدة فإن السوكيت عليه أرقام الأطراف .**

الآن يمكن بسهولة التوصيل بالكومبيوتر عن طريق المنفذ التساسلى المسمى COM port.

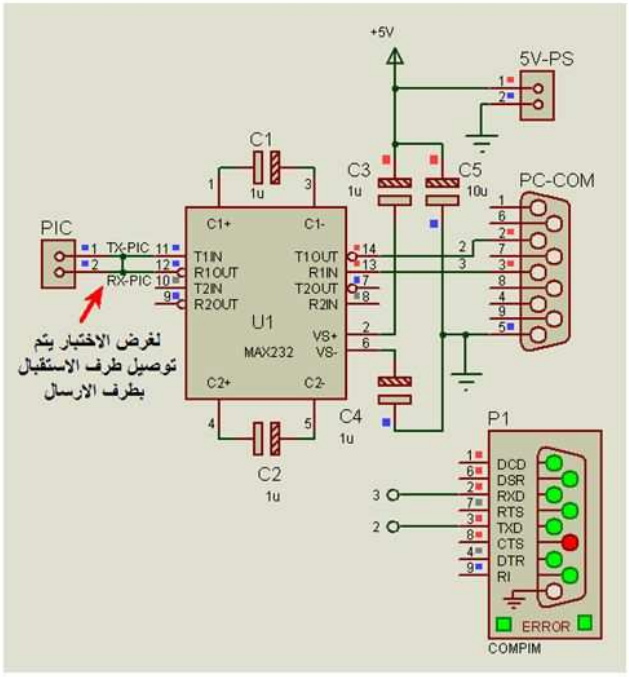
الاختبار : Testing

من المفضل التحقق من كل جزء على حدة . لذلك سوف نختبر المحول لنرى على يعمل جيدا .
للاختبار سوف نستخدم برنامج الاتصال الخاص بالنوافذ والمسمى Hyper-terminal والذى يمكن عن طريقة وبسرعة فتح المنافذ التسلسلية COM ports وإرسال واستقبال بيانات نصية . حتى الآن أنت لا تحتاج إلى الميكروكونترولر ولا لأى برمجة .

نظرية الاختبار تتلخص فى توصيل المخارج (RX/TX) معا بحيث أن أى بيانات تكتب إلى المنفذ COM port تخرج من الكومبيوتر و تدخل إلى دائرة MAX232حيث يحولها إلى مستويات العمل نوع TTL ثم تعود مرتدة وتدخل MAX232 ليحولها إلى مستويات RS232 للمنفذ التسلسلى وتدخل الى المنفذ COM port للكومبيوتر عائدة مرة أخرى .



مرة أخر سوف نستخدم برنامج المنافذ التخيلية السابق ذكره فى جعل الكومبيوتر يتصل عن طريق COM3 ووحدة MAX232 تتصل عن طريق COM4 بإضافة عنصر الاتصال التسلسلى المتاح فى برنامج بروتيس والمسمى COMPIM كما فى الشكل التالى :

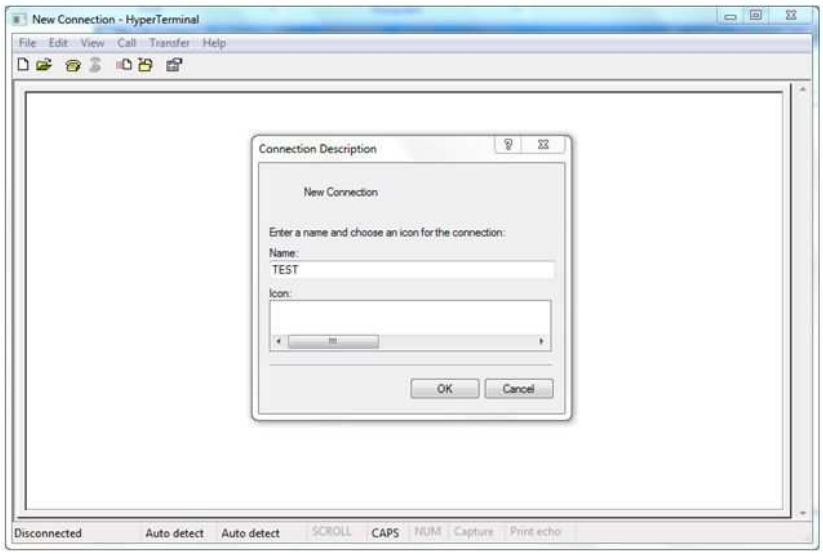


ونكرر ما سبق ذكره كتذكرة فى حينها :

يمكن تنفيذ ذلك برمجيا عن طريق البرنامج المعروف باسم Driver Virtual Serial Ports (درايفر المنافذ المتسلسلة التخيلية) . بعد تنصيب البرنامج وتشغيله نقوم بإنشاء منفذين متسلسلين تخيليين يمكن عن طريقهما تحقيق الاتصال المطلوب كما فى الشكل حيث يتم إضافة زوج Add pair من المنافذ التخيلية ولتكن COM3 و COM4 .

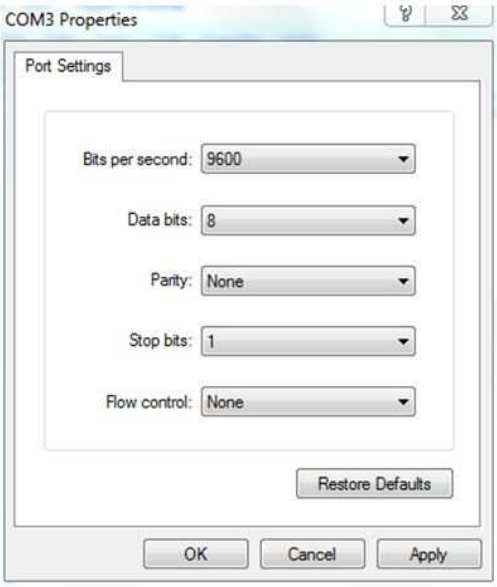
أحد المنفذين يمثل الكومبيوتر وليكن COM3 : ويتم تحديد ذلك فى برنامج HyperTerminal كما يلى :

- **تشغيل برنامج HyperTerminal وتسمية للتوصيل وليكن TEST**

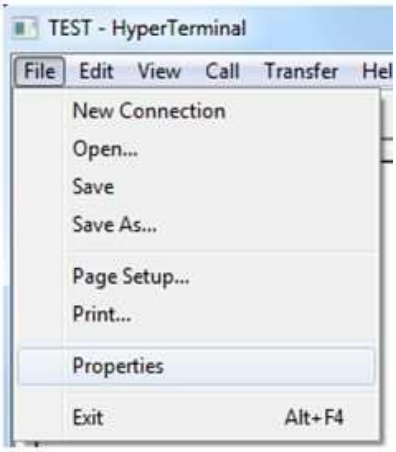


- **يلى ذلك تحديد المنفذ**

- **ثم تحديد خصائص الاتصال**

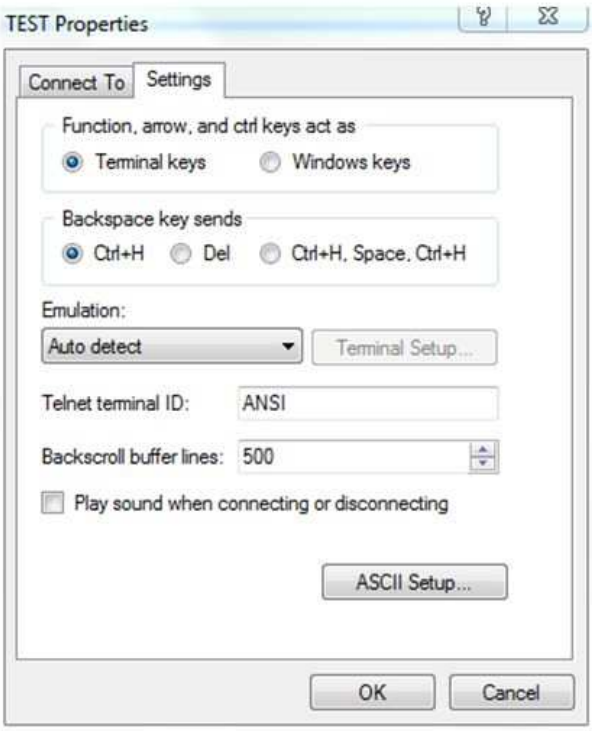


- **لتحديد باقى الخواص من القائمة File اختار الخواص Properties**

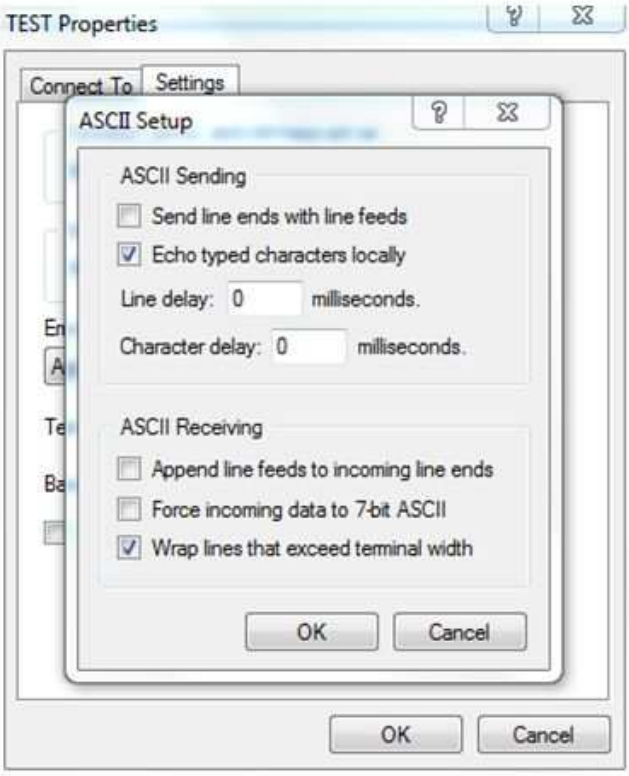


تظهر لك قائمة الخواص أنقر على مفتاح تحديدات Settings

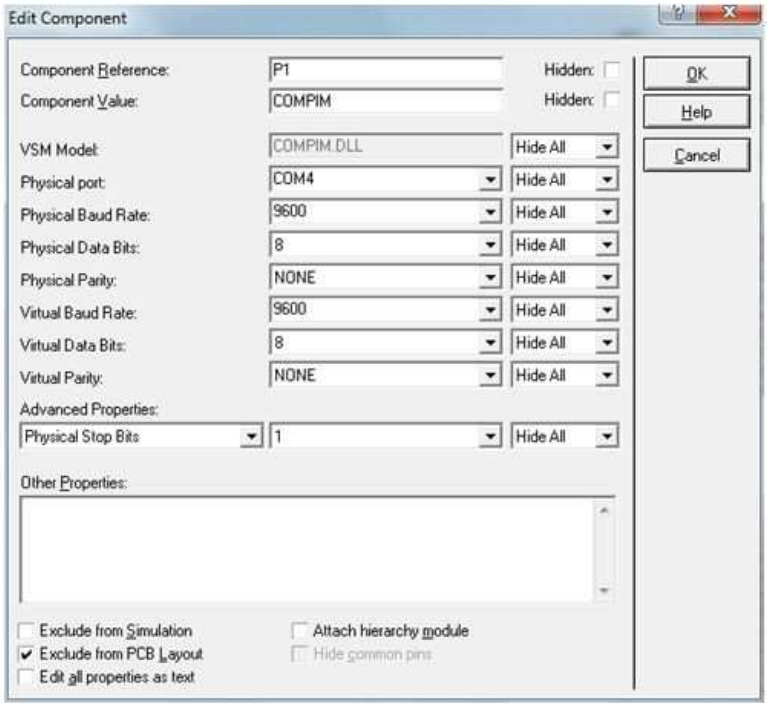
ثم انقر على مفتاح ASCII Setup



واختار (ضع علامة) Echo typed .. ليظهر (يعرض) صدى لما تكتبه محليا علاوة على الاستجابة الخارجية الناتجة من الاتصال .



المنفذ الآخر COM4 ويمثل برنامج المحاكاة (الوحدة الطرفية): ويتم ذلك بتحرير خواص الوحدة الطرفية للتتلائم مع المنفذ COM3 حتى يتم الاتصال بينهما كما فى الشكل :



شغل برنامج المحاكاة وتأكد من تمام الاتصال بكتابة نص على نافذة برنامج HyperTerminal فتظهر لك نتيجة نجاح تحقيق الاتصال كما يلى :

تدريب : الاتصال التسلسلى الغير متزامن Asynchronous serial communication

الوصف :

يمتلك الكثير من الميكروكونترولر PIC أجهزة hardware ارسال واستقبال عامة غير متزامنة USART ضمن بنائها الداخلى والتى تسمح بالاتصال على مدى واسع بالأجهزة التسلسلية كذاكرات الشرائح ووحدات العرض LCD والكومبيوتر الشخصى ..ألخ .

الوحدة (الموديول) USART لها نظامان للعمل :

النظام المتزامن synchronous: يحتاج لنبضات ساعة للتزامن بين المرسل والمستقبل .

النظام الغير متزامن asynchronous : لا يحتاج لنبضات ساعة للتزامن بين المرسل والمستقبل . وهذا النظام هو الأكثر شيوعا لذلك سوف يتم التركيز عليه ونقوم بإنشاء ربط للبيانات التسلسلية بين الميكروكونترولر PIC وبين الكومبيوتر الشخصى .

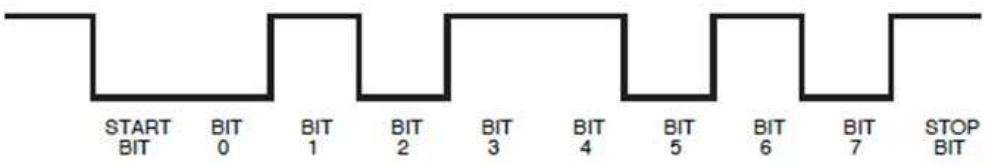
المعلومات النظرية المطلوبة :

تستخدم الاتصالات التسلسلية فى النظم المبنية على أساس الميكروكونترولر ويرجع ذلك في معظمه إلى ندرة أطراف المداخل / المخارج المتاحة . وبالاضافة الى إمكانية الاتصالات لمسافات طويلة فإن نقل البيانات تسلسليا هو الأكثر بساطة وفعالة من حيث التكلفة لأن توصيلات الأجهزة المطلوبة لربط البيانات يمكن أن تخفض إلى ثلاثة ، هى وصلة الإرسال Tx ووصلة الاستقبال Rx والأرضى المشترك Gnd .

هناك نوعان مختلفان من الاتصالات التسلسلية : النوع المتزامن والنوع الغير المتزامن. ويتمثل التحدي الرئيسي في وجود صلة (ربط) للبيانات التسلسلية هو الحفاظ على التزامن بين المرسل والمستقبل .
يستخدم الأسلوب الغير متزامن بروتوكول ذو خانة بدء bit start وخانة إيقاف bit stopللتزامن بين طرفى (نهايتى) الوصلة .
يتم إرسال كل بايت لحرف character byte في إطار يتألف من خانة بدء start bit يليها خانات الحرف bits character يليها (اختياريا) خانة التماثل (التطابق) parity bit ويتم الانتهاء بخانة توقف واحدة أو أكثر bits stop.
لابد من تهيئة (إعداد) كل من المرسل والمستقبل بنفس معدل البيانات وبنفس عدد خانات البيانات وبنفس عدد خانات التوقف.

في حالة الخمول(السكون) idle condition يكون خرج الارسال فى الحالة المنطقية المرتفعة .عندما يكون المرسل على استعداد لإرسال بايت الحرف فإنه يشير (يعطى إشارة) إلى المستقبل عن طريق سحب خط الارسال للحالة المنخفضة لفترة زمنية واحدة لنبضات الساعة . هذا هو بت (خانة) البدء start bit وهو يخبر المستقبل بأن القادم هو إطار بايت البيانات . يقوم المستقبل بقراءة عدد خانات الحرف المتوقع حسب البروتوكول المعتمد (والذى تم التهيئة له من قبل) حتى يتم سحب الخط إلى المنطق المرتفع بواسطة المرسل (يمثل بت توقف واحد أو أكثر) ويكون هذا هو نهاية الإطار. يتم تكرار العملية برمتها في كل مرة يكون المرسل جاهزا لإرسال بايت لحرف.
ويسمى هذا النوع من الارسال التسلسلى بالغير متوامن لأن المستقبل يقوم بعمل إعادة تزامن لنفسه مع المرسل في كل مرة يتم إرسال البيانات باستخدام بت البداية start bit. ومع ذلك، فإنه داخل كل إطار يكون الطرفين متزامنين .

من ناحية أخرى، فإنه فى الاتصال التسلسلى المتزامن يتم إرسال الحروف فى مجموعات أو كتل blocks مع عدم وجود خانات إطار تحيط بها . تتم مزامنة الارسال والاستقبال بخط منفصل لنبضات الساعة أو في بعض الحالات يتم تضمين إشارة نبضات الساعة مع الحروف المرسلة .
في كلا النوعين من الاتصالات التسلسلية،يعرف معدل البيانات المرسلة والمستقبلة بمعدل البود baud rate (عدد النبضات فى الثانية).
على سبيل المثال موديول USART الموجود داخل الميكروكونترولر PIC16F628A يعتمد كلا النوعين من الاتصالات التسلسلية وكنه أكثر ملائمة للطريقة الغير متزامن asynchronous .
فى النظام الغير متزامن يعمل الطرف RB2 كمخرج إرسال TX ويعمل الطرف RB1 كمدخل استقبال RX .
ترسل بايت البيانات كسلسلة نصية مكونة من ١٠ خانات : خانة البدء start biو ٨ خانات بيانات eight data bits وخانة إيقاف stop bit كمل فى الشكل التالى :

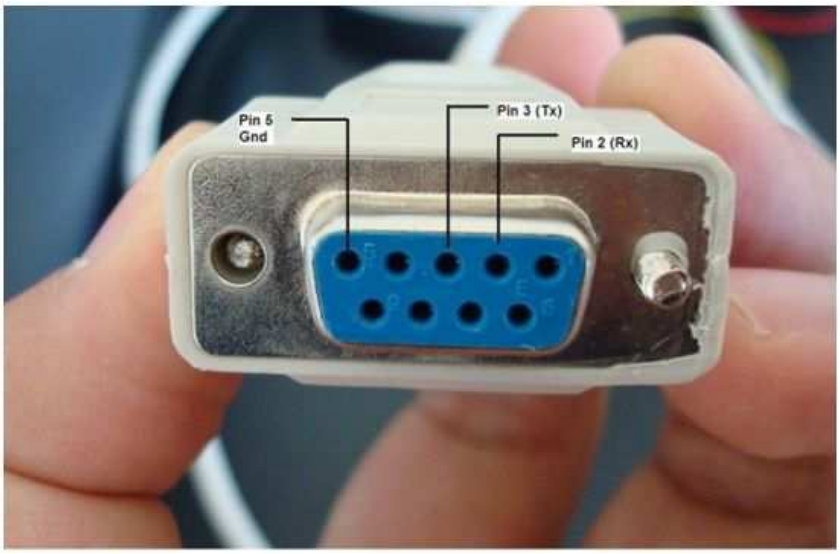


المنفذ التسلسلى بالكومبيوتر الشخصى (والمعروف باسم COM port) يستخدم النظام القياسى RS232-C للاتصال التسلسلى . والذى يحدد خواص الكهربائية والميكانيكية والإشارات والخطوات المتبعة للربط عن طريق الاتصال التسلسلى .

المنطق المرتفع فى RS232-C هو إشارة بجهد فى المدى -٣ V to -١٥ V (عمليا -١٢V) والمنطق المنخفض يكون فى المدى +٣ V to +15 V (عمليا +١٢V) . أى أنه لا يماثل المستويات المنطقية للميكروكونترولر PIC لأنه فى RS232-C الحالة المرتفعة سالبة والحالة المنطقية المنخفضة موجبة .
الجدول التالى يبين التوصيلات القياسية لنظام RS232-C للسوكيتات ٢٥-pin و٩-pin و RJ-45 .
هنا سوف نتناول فقط أقل توصيلات ممكنة بين الميكروكونترولر والكومبيوتر الشخصى باستخدام خطوط الإشارات TX, RX, GND فقط .

Definition of Common RS-232-C Lines

CONNECTOR					
DB-25	DB-9	RJ-45	FUNCTION	CODE NAME	DIRECTION
1		4	Ground	G	
2	3	6	Transmit data	TXD	Output
3	2	5	Receive data	RXD	Input
4	7	8	Request to send	RTS	Output
5	8	7	Clear to send	CTS	Input
6	6		Data set ready	DSR	Input
7	5		Chassis ground	G	
8	1	2	Carrier detect	CD	
20	4	3	Data terminal ready	DTR	Output
22	9	1	Ring indicator	RI	Input



سوف نستخدم النظام الغير متزامن لعمل اتصال مع المنفذ التسلسلى بالكومبيوتر الشخصى عن طريق RS232-C .
نظرا لأن الميكروكونترولر PIC16F628A يمتلك بالفعل ضمكن بنائه الداخلى الموديول USART والذى يعتمد النظام الغير متزامن للاتصال التسلسلى لذلك كل ما هو مطلوب وحدة خارجية لإزاحة المستوى لترجمة الإشارات ذات المستوى TTL إلى مستويات النظام RS232-C والعكس بالعكس .
يمكن تحقيق ذلك باستخدام الشريحة MAX232 المصنعة بمعرفة شركة Maxim .تحتاج الشريحة إلى قليل من المكثفات الخارجية لاستخدامها فى عملية مضخات الشحن الداخلية لكى تولد الجهود +١٢V و -١٢V المطلوبة للاتصال عن طريق RS232-C .
الطريقة البسيطة لارسال واستقبال بايتات من خلال المنفذ التسلسلى للكومبيوتر الشخصى باستخدام برنامج HyperTerminal الموجود ضمن نظام الوندوز عن طريق :

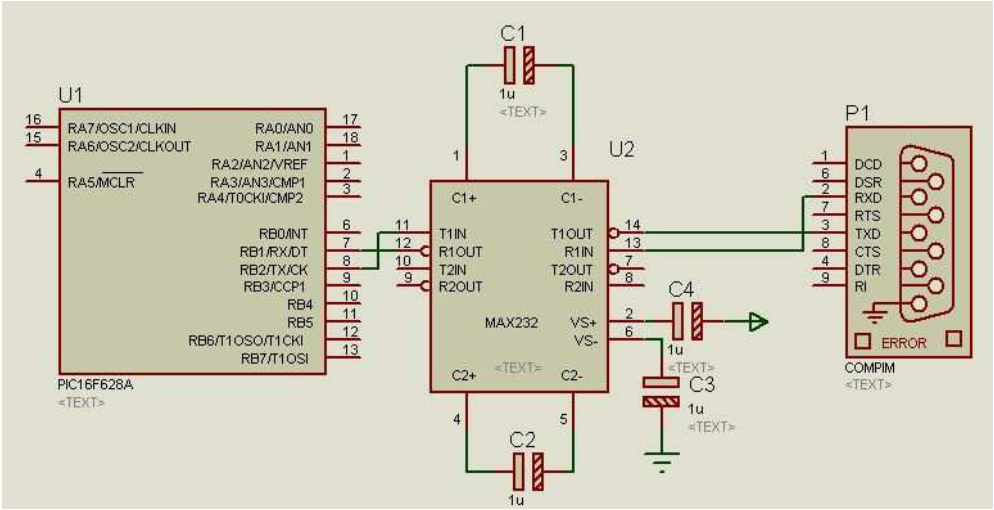
Communications→Hyperterminal → Menu → Programs → Accessories → Start

حيث يمكنك إنشاء اتصال مع المنفذ التسلسلى (COM1 على سبيل المثال) واختيار معدل البود baud rate وعدد الخانات number of bits وتحديد نوع التطابق parity ..ألخ .
عند إتمام الاتصال فأن أى حرف سوف تكتبه سوف يرسل (ككود أسكى) من خلال المنفذ التسلسلى . الحروف المستقبلة سوف يتم عرضها على الشاشة أيضا .

الدائرة الكهربية :

تتكون الدائرة من الميكروكونترولر PIC16F628A والشريحة MAX232 والتى تحتاج إلى ٤ مكثفات قيمة كل واحد منهم ١ µ F .

من جانب الكومبيوتر سوف يتم توصيل ثلاثة خطوط فقط (Tx, Rx, Ground) إلى المنفذ التسلسلى COM من خلال سوكيت ٩-pin .



البرنامج :

سوف نستخدم المترجم MikroC Pro for PIC . هذا المترجم يمدنا بمكتبة باسم UART والتى تعتمد نظام الاتصال التسلسلى الغير متزامن فى النظام المزدوج الكامل full duplex (الارسال والاستقبال فى نفس الوقت) . وهذا يؤدى إلى سهولة البرمجة بشكل كبير .

على سبيل المثال إذا رغبت فى تهيئة الموديول UART الموجود بالميكروكونترولر PIC16F628 للبدء بمعدل نبضات قدره ٩٦٠٠ نبضة فى الثانية فقط كل ما تحتاجه هو كتابة الدالة UART1_Init(9600) .

البرنامج التالى يقوم بإنشاء اتصال تسلسلى بالنظام الغير متزامن بين الميكروكونترولر PIC16F628A والكومبيوتر الشخصى .

يقوم الميكروكونترولر بارسال الرسالة 'in a Number Type' "أكتب عدد" والتى تعرض فى نافذة برنامج Hyperterminal .

عندما تقوم أى حرف من لوحة المفاتيح سوف يرسل إلى الميكروكونترولر من خلال المنفذ COM .

سوف يقوم الميكروكونترولر بقراءة الحرف ويرسله عائدا إلى الكومبيوتر ومن ثم يعرض مرة أخرى فى نافذة برنامج Hyperterminal .

يجب ضبط برنامج Hyperterminal لهذا البرنامج على :

Bits per second: 9600،
Data Bits: 8،
Parity: None،
Stop bits: 1،
Flow control: None.

البرنامج :

كود:

```
/*
Hardware UART
MCU: PIC16F628A
External 4MHz Crystal, MCLR Enabled, PWRT Enabled, WDT OFF
*/
void newline(){
UART1_Write(13); // Carriage Return
UART1_Write(10); // Line Feed
}
void main() {
unsigned char MyError, Temp;
CMCON = 7;       // Disable Comparators
TRISB = 0b00000010;
UART1_Init(9600);
Delay_ms(100);
UART1_Write_Text("Testing UART! ");
newline();
do {
UART1_Write_Text("Type in a Number: ");
while(!UART1_Data_Ready());
Temp = UART1_Read();
newline();
UART1_Write_Text("You entered: ");
UART1_Write(Temp);
newline();
} while(1);
}  // End main()
```

التعليق على البرنامج :/* Hardware UART:MCU PIC16F628AExternal 4MHz Crystal, MCLR Enabled, PWRT Enabled, WDT OFF*/

١- الإعلان عن دالة لبدء الطباعة من أول سطر جديد على خطوتين : عودة المؤشر إلى بداية السطر ثم النتقال إلى سطر جديد بغرض تسهيل فهم وقراءة البرنامج .void newline(){ UART1_Write(13); // Carriage Return UART1_Write(10); // Line Feed}

٢- بداية الدالة الرئيسية void main(){

٣- الإعلان عن المتغيرات unsigned char MyError، Temp؛

٤- تهيئة منافذ الميكروكونترولر PIC16F628 :- عدم تفعيل موديول المقارنات بغرض جعل جميع الأطراف كمداخل / مخارج رقمية فقط .- تهيئة المنفذ PORTB لكى تعمل جميع أطرافه كمخارج فيما عدا الطرف RB1 فيعمل كمدخل وهو طرف دخل الاستقبال Rx . CMCON = 7; // Disable Comparators TRISB = b00000010؛

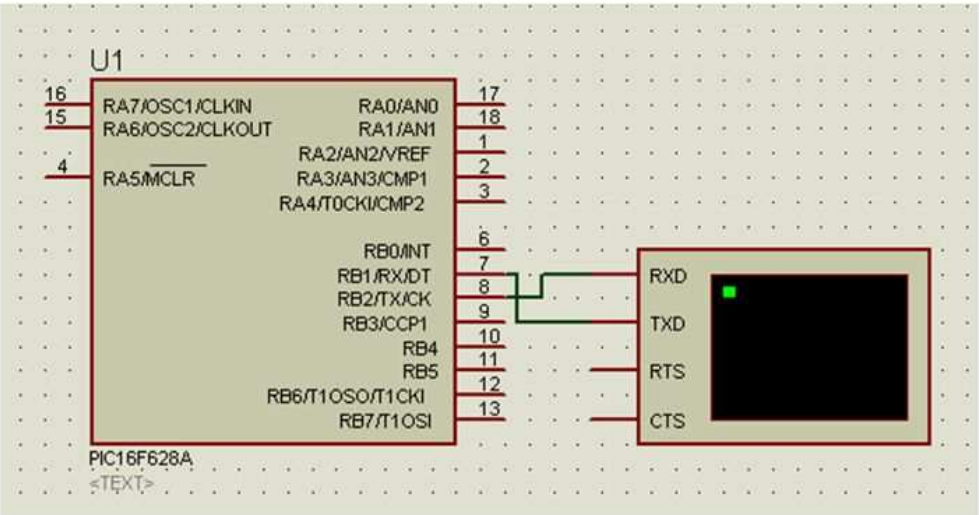
٥- بدء تهيئة موديول UART1 بمعدل بود ٩٦٠٠ ثم تأخير زمنى ١٠٠ ملى ثانية حتى تستقر التهيئة . UART1_Init(9600)؛ Delay_ms(100)؛

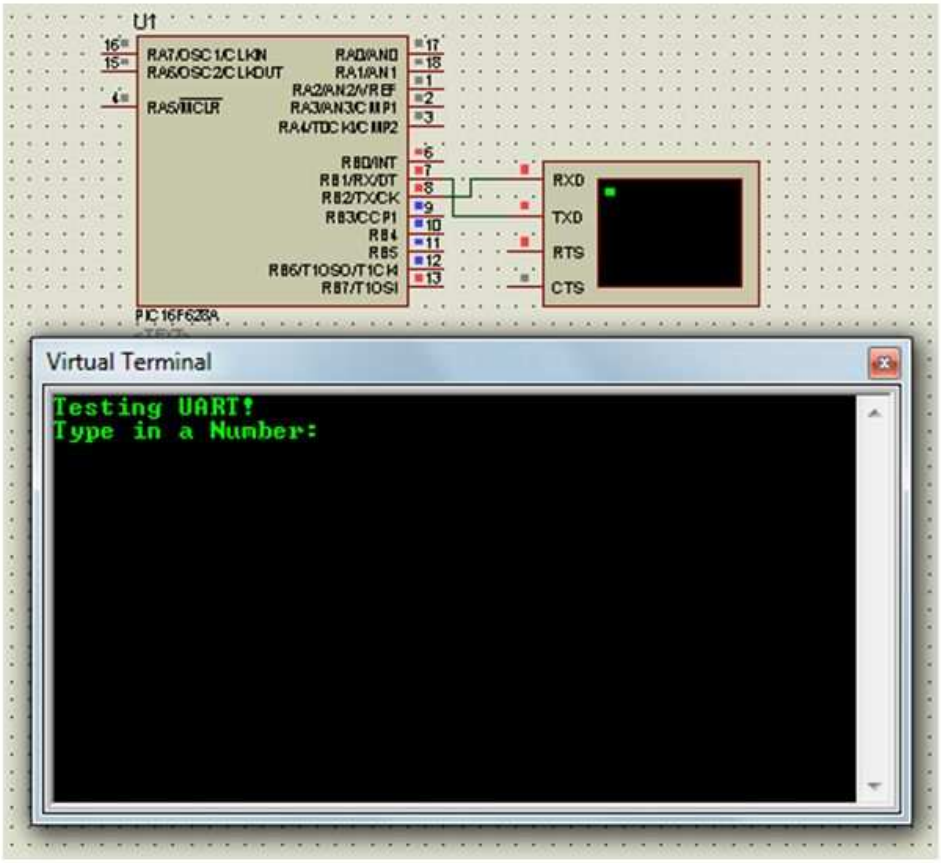
٦- كتابة (إرسال) سلسلة نصية تنص على UART Testing! ثم استدعاء دالة "سطر جديد " التى تم إنشائها سابقا . UART1_Write_Text("Testing UART! ")؛ newline()؛

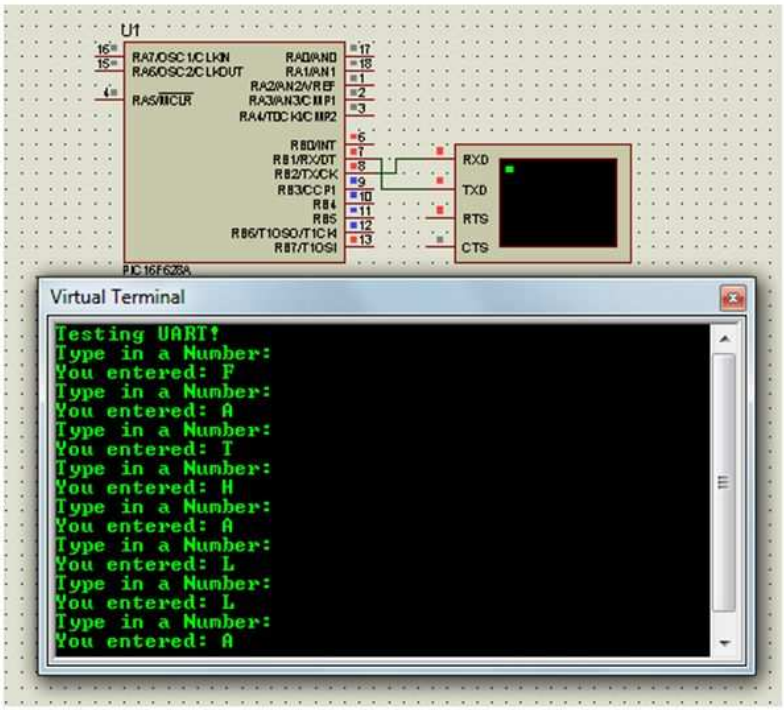
٧- الدخول إلى حلقة do{.... While....} وتحتوى على :- كتابة (إرسال) سلسلة نصية تنص على Type in a Number: "أكتب عدد : " .- اختبار جاهزية UART لقراءة البيانات فإن كانت البيانات جاهزة تعود الدالة UART1_Data_Ready() بواحد وبعد التأثير عليها بمؤثر المعكوس "!" تكون النتيجة بصفر وتصبح عبارة while غير محققة false ويخرج البرنامج من هذه الحلقة لينفذ ما بعدها وهو قراءة البايت الجاهز للاستقبال ونسخة وحفظه فى المتغير المسمى Temp ومرة ثانية استدعاء دالة سطر جديد بعد كتابة Number Type in a: . do { UART1_Write_Text("Type in a Number: ")؛ while(!UART1_Data_Ready())؛ Temp = UART1_Read()؛ newline()؛- كتابة (إرسال) السلسلة النصية You entered: "لقد قمت بإدخال : " ويليها كتابة ما تم قراءته (استقباله)وحفظه فى المتغير Temp وهو الرقم الذى أدخلته . مرة أخرى يتم استدعاء دالة سطر جديد UART1_Write_Text("You entered: ")؛ UART1_Write(Temp)؛ newline()؛-

تكرار ما سبق باستخدام حلقة غير منتهية . } 1)while(؛{ // main End()

أولا التعامل مع الوحدة الطرفية ببرنامج بروتيس للتأكد من البرنامج مباشرة دون الحاجة إلى المحول MAX232 .







ثانيا : التعامل من خلال محاكاة العنصر COMPIM وبرنامج terminal Hyper والمنافذ التخيلية كما سبق