برمجة المتحكمات المصغرة

التجارب العملية

الجلسة العاشرة

Programming

Embedded Systems Microcontroller

You Can Practice Microcontroller Programming Easily Now!

WALID BALID, Tuesday, December 15, 2009

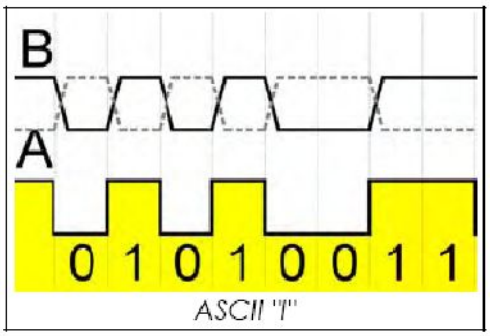
الغاية من التجربة:

دراسة بروتوكول الاتصال التسلسلي RS485 وتطبيقاته في أنظمة التحكم الرقمي.

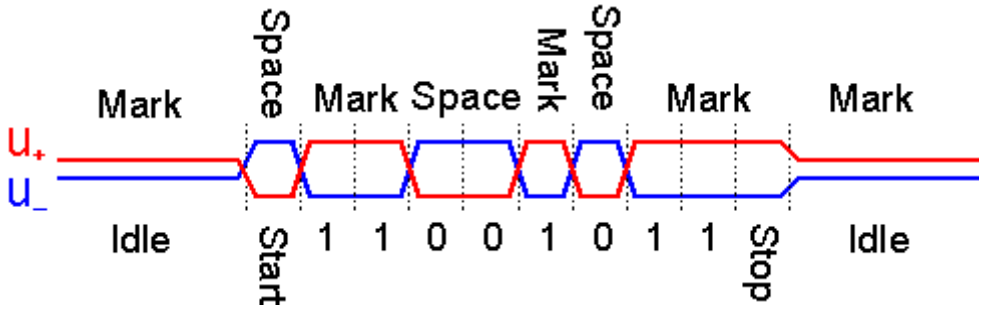
مبدأ عمل البروتوكول RS485:

يعتبر البروتوكول RS485 من أكثر بروتوكولات الاتصال التسلسلي متعدد النقاط انتشاراً في التطبيقات الصناعية في وقتنا الحاضر.

يعتمد هذا البروتوكول على مبدأ فرق الجهد التفاضلي بين خطي النقل (A,B)، حيث أنه في حال كون الجهد VA<VB تكون الحالة المنطقية على خط النقل "1" وفي حال كون الجهد VA>VB تكون الحالة المنطقية على خط النقل "0". في حال عدم وجود بيانات على خط النقل فإن الحالة المنطقية على الخط هي "1".



إن البروتوكول RS485 ذو ممانعة عالية للضجيج وذلك لأنه يعتمد على مبدأ الجهد التفاضلي على الخط لتعيين الحالة المنطقية لكل بت، فمثلاً إن وجود ضجيج على الخط A سوف يحدث ضجيجاً مماثلاً على الخط B، وباعتبار أن هذا البروتوكول يقارن فرق الجهد بين الخطين، فإن أي زيادة مماثلة أو نقصان لن يغير نتيجة المقارنة.



الشركات المصنعة لشرائح RS485:

الجدول التالي يبين الشركات الأساسية المصنعة لدرايفرات البروتوكول RS485.

ST	TI	MAXIM	LTC
ST485	SN75176	MAX485	LTC485

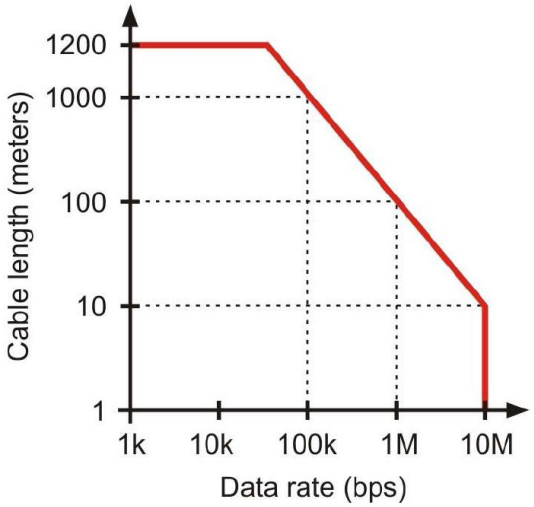
مواصفات البروتوكول RS485:

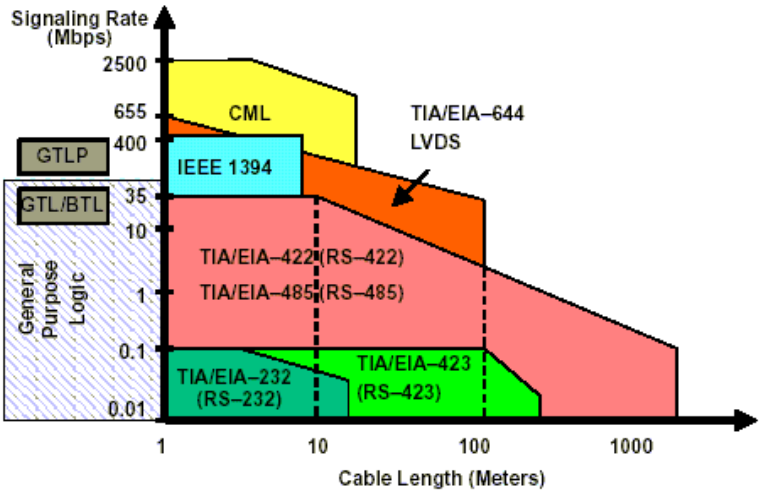
الجدول التالي يبين المواصفات التقنية للبروتوكول RS485 ومقارنة مع برتوكولات مشابهة.

SPECIFICATIONS		RS232	RS423	RS422	RS485
Mode of Operation		SINGLE-ENDED	SINGLE-ENDED	Differential	Differential
Total Number of Drivers and Receivers on One Line		**1**DRIVER **1**RECVR	**1**DRIVER **10**RECVR	**1**DRIVER **10**RECVR	**32**DRIVER **32**RECVR
Maximum Cable Length		50 FT	4000 FT	4000 FT	4000 FT
Maximum Data Rate		20kb/s	100kb/s	10Mb/s 100Kb/s	10Mb/s 100Kb/s
Maximum Driver Output Voltage		±25V	±6V	-0.25V to +6V	-7V to +12V
Driver Output Signal Level (Loaded Min.)	*Loaded*	±5V to ±15V	±3.6V	±2.0V	±1.5V
Driver Output Signal Level (Unloaded Max)	*Unloaded*	±25V	±6V	±6V	±6V
Driver Load Impedance (Ohms)		3k to 7k	>=450	100	54
Max. Driver Current in High Z State	*Power On*	N/A	N/A	N/A	±100uA
Max. Driver Current in High Z State	*Power Off*	±6mA @ ±2v	±100uA	±100uA	±100uA
Slew Rate (Max.)		30V/µS	Adjustable	N/A	N/A
Receiver Input Voltage Range		±15V	±12V	-10V to +10V	-7V to +12V
Receiver Input Sensitivity		±3V	±200mV	±200mV	±200mV
Receiver Input Resistance (Ohms)		3k to 7k	4k min	4k min	>=12k

العلاقة بين معدل النقل وطول خط النقل:

الشكل اليميني يوضح العلاقة بين معدل النقل وطول الكابل لبعض بروتوكولات الاتصال التسلسلي الصناعية، بينما يوضح الشكل اليساري العلاقة بين معدل النقل وطول الكابل للبروتوكول RS485.

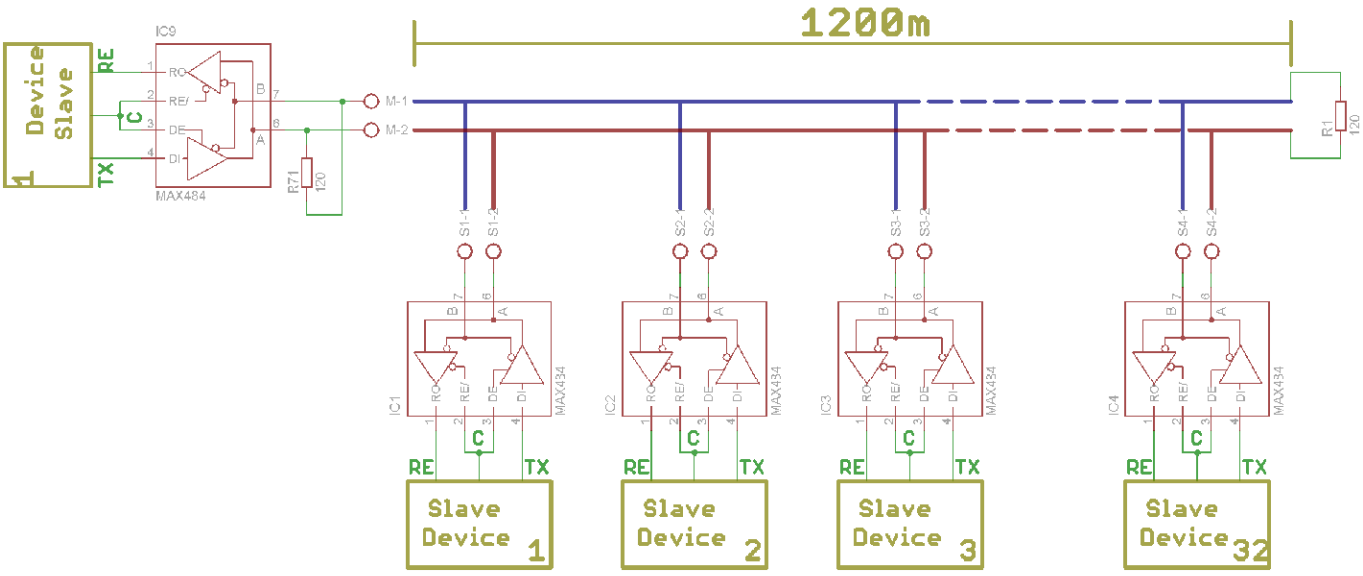




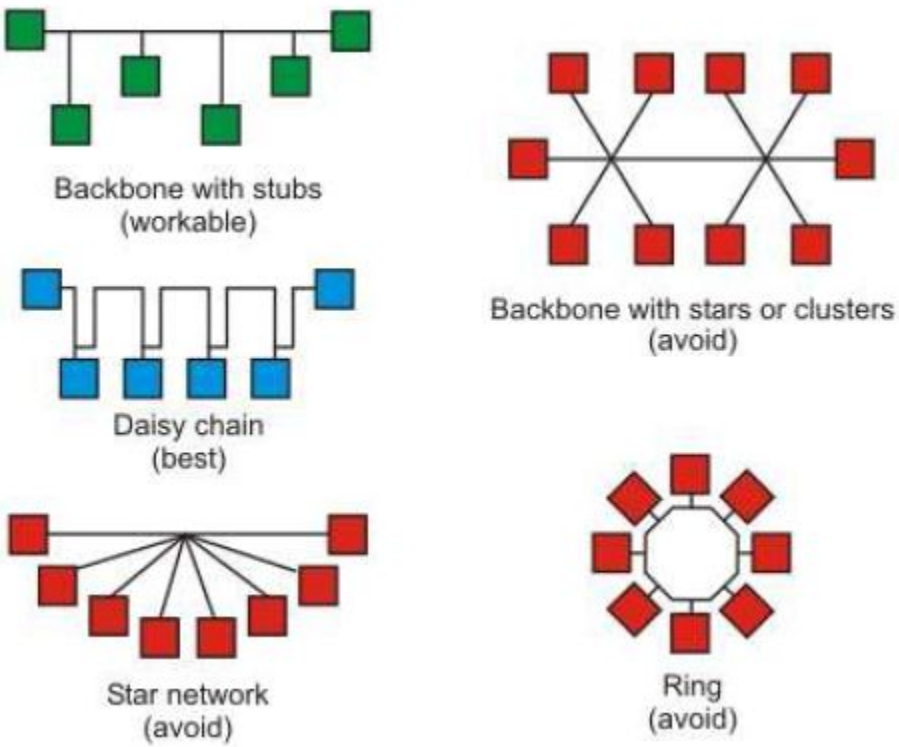
ملاحظة: بشكل عام فإنه بازدياد طول خط النقل يتم تخفيض معدل النقل لتخفيض الضجيج، ويمكن الحصول على مسافة نقل أعظمية عند معدلات نقل منخفضة.

توصيل شبكة RS485 باستخدام خطين:

يتم توصيل الشبكة بحيث يتم وصل جميع النقاط **A** مع بعضها وكذلك جميع النقاط **B**، ويتم وضع مقاومات تحميل طرفية في بداية ونهاية الخط كما في الشكل التالي:



ملاحظة: إن التفريعات الصادرة عن الخط الرئيسي يجب أن لا تتجاوز 15Meter، والأفضل أن تستخدم طريقة الوصل Daisy Chain بدلاً من طريقة الوصل Backbone with Stubs.



ملاحظة هامة: إن الاعتقاد الشائع بأن البروتوكول RS485 يحتاج إلى خطين فقط ولا يحتاج إلى خط تأريض هو اعتقاد خاطئ، فيجب توصيل خط التأريض بين الوحدات أو وصله إلى نقطة التأريض. الجدول التالي مثال عن الكابلات المستخدمة.

Belden P/N	Pairs	AWG	mm	Shield/drain wire	Imp	Cap
9841	1	24	0.6	Yes, 24AWG	120ohm	42pf/m
9842	2	24	0.6	Yes, 24AWG	120ohm	42pf/m
8132	2	28	0.4	Yes, 28AWG	120ohm	36pf/m

التحكم بشبكة RS485 باستخدام خطين:

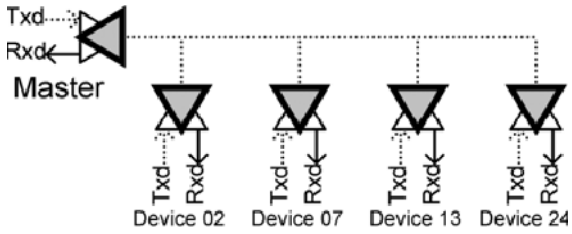
يتم التحكم بهذه الشبكة وفق مبدأين:

- PASSIVE DUPLEX CONTROL (*AUTOMATIC*)
- ACTIVE DUPLEX CONTROL (*RTS Pin*)

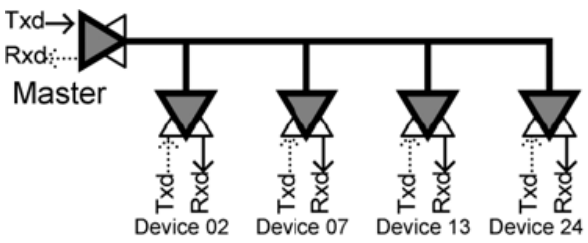
أولاً: نمط التحكم Passive Duplex: في هذه الحالة يستخدم خطين فقط (RO, DI) للتحكم بـ RS485 Driver ولا حاجة لقطب التحكم بالاتجاه (DE/RE)، حيث أن المستقبلات تفحص بشكل دائم خط النقل وعندما تستقبل أمرً من المرسل تقوم بالرد عليه، حيث أن التنسيق يتم برمجياً.

مبدأ العمل:

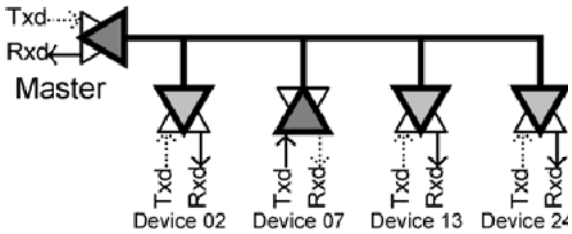
- **لا يوجد بيانات للإرسال** (No data to send): في هذه الحالة تكون الحالة المنطقية على خط النقل هي "1" وجميع الطرفيات الثانوية تكون في حالة انتظار استقبال بيانات من الوحدة الرئيسية للاستقبال (حالة توقف Idle).



- **إرسال أمر من الوحدة الرئيسية** (Master sends a request): تقوم الوحدة الرئيسية بإرسال أمر طلب على الناقل، فتكتشف الوحدات الثانوية بت بدء الإرسال (Start Bit) وتبدأ بقراءة البيانات الموجودة على الناقل، وعندما ينتهي الإرسال من قبل الوحدة الرئيسية ببت التوقف (Stop Bit)، تقوم الوحدات بتفحص الخطأ (Check sum) وفي حال حصوله لا يتم إرسال أي استجابة، وتعود جميع الوحدات إلى نمط الاستقبال ويصبح الناقل في حالة توقف (Idle).



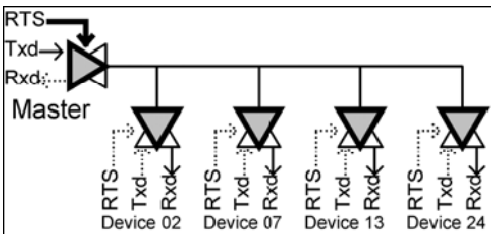
§ **إرسال رد من الوحدة الثانوية** (Slave sends a response): بفرض أن عنوان الجهاز الذي تم طلبه من قبل الوحدة الرئيسية هو add = 07 وبالتالي ستقوم الوحدة الثانوية السابعة بالرد وتتحول إلى وحدة إرسال بنفس الوقت الذي تكون فيه الوحدة الرئيسية في حالة استقبال وانتظار الرد، حالما تنتهي هذه الوحدة من الإرسال تعود إلى حالتها كمستقبل ويصبح خط النقل في حالة توقف من جديد.



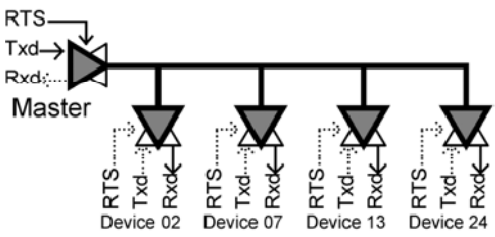
Ü **ثانياً: نمط التحكم Active Duplex:** في هذه الحالة يستخدم ثلاث خطوط للتحكم بشريحة RS485 Driver، حيث يستخدم الخط الثالث (RTS Control) كخط تحكم لتحديد اتجاه وحدة الإرسال/الاستقبال.

مبدأ العمل:

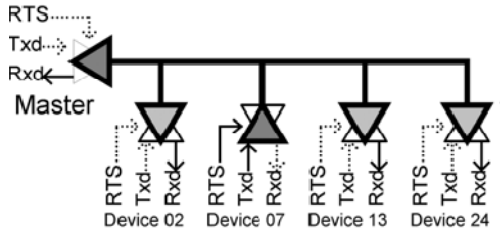
§ **لا يوجد بيانات للإرسال** (No data to send): في هذه الحالة تقوم الوحدة الرئيسية بتفعيل إشارة خط التحكم بالاتجاه الخاص بها (RTS) مما يجبر خط النقل إلى التحول إلى حالة التوقف (الحالة المنطقية على خط النقل هي "1"). كل إشارات التحكم بالاتجاه الخاصة الوحدات الثانوية تكون غير مفعلة والوحدات في حالة ترقب.



§ **إرسال أمر من الوحدة الرئيسية** (Master sends a request): تقوم الوحدة الرئيسية بإرسال أمر طلب على الناقل فتكتشف الوحدات الثانوية بت بدء الإرسال (Start Bit) وتبدأ بقراءة البيانات الموجودة على الناقل، وعندما ينتهي الإرسال من قبل الوحدة الرئيسية ببت التوقف (Stop Bit) تقوم الوحدات بتفحص الخطأ (Check sum) وفي حال حصوله لا يتم إرسال أي استجابة، وبعدها تقوم الوحدة الرئيسية بإلغاء تفعيل خط التحكم بالاتجاه وتعود جميع الوحدات إلى نمط الاستقبال ويصبح الناقل في حالة توقف من جديد (Idle).



إرسال رد من الوحدة الثانوية (Slave sends a response): بفرض أن عنوان الجهاز الذي تم طلبه من قبل الوحدة الرئيسية هو add = 07 وبالتالي ستقوم الوحدة الثانوية السابعة بتفعيل خط التحكم بالاتجاه الخاص بها وتتحول إلى وحدة إرسال بنفس الوقت الذي تكون فيه الوحدة الرئيسية في حالة استقبال وانتظار الرد، وحالما تنتهي هذه الوحدة من الإرسال تقوم بإلغاء تفعيل خط التحكم بالاتجاه الخاص بها وتعود إلى حالتها كمستقبل، حيث أن وحدة التحكم الرئيسية تقوم بإعادة تفعيل خط التحكم بالاتجاه الخاص بها حالما تدرك أنه تم انتهاء الإرسال بالتعرف على بت التوقف.



الميزات والمساوئ لبرتوكول الاتصال RS232:

المساوئ (Disadvantages)	المحاسن (Advantages)
× مناسب فقط من أجل الربط بين System-to-System أكثر من كونه قابلاً للربط بين Chip-2-Chip أو من أجل تطبيقات Chip-2-Sensor. × يحتاج إلى وحدة تبديل مستوى منطقي TTL<>RS485، بالإضافة إلى كابل مزدوج ملفوف ومقاومات طرفية مما يزيد من كلفة النظام.	ü بروتوكول اتصال شائع الاستخدام في كثير من التطبيقات ومعتمد من قبل العديد من الشركات. ü مسافة الاتصال طويلة جداً حوالي 1200 متر عند معدل إرسال 100kb/s. ü مناعة ضد الضجيج بسبب الجهد التفاضلي. ü سهل البناء والبرمجة ومتوفر برمجياً وكعيان صلب.

التخاطب باستخدام البنية البرمجية MODBUS وتطبيقاتها في الاتصالات التسلسلية RS485:

يعتبر التشكيل MODEBUS لرسالة البيانات أحد أهم البروتوكولات البرمجية المستخدمة في الاتصالات التسلسلية متعددة النقاط (Multi Master/Slave)، حيث أن هذا التشكيل لرسالة المعلومات المرسلة أو المستقبلة يتيح مرونة كبيرة في تحديد الجهاز المراد التخاطب معه وتعيين الوظيفة التي سيقوم بها، بالإضافة لكشف حدوث الأخطاء.

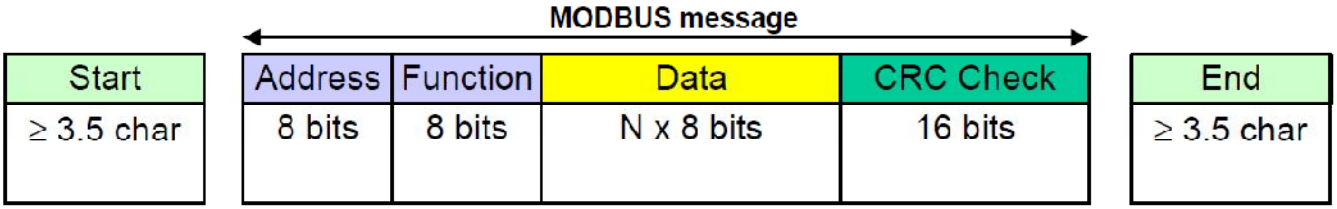
يوجد تشكيلين متقاربين إلى حد ما للبروتوكول البرمجي MODEBUS، أحدهما للقائد (Master Device) والآخر للمقاد (Slave Device).

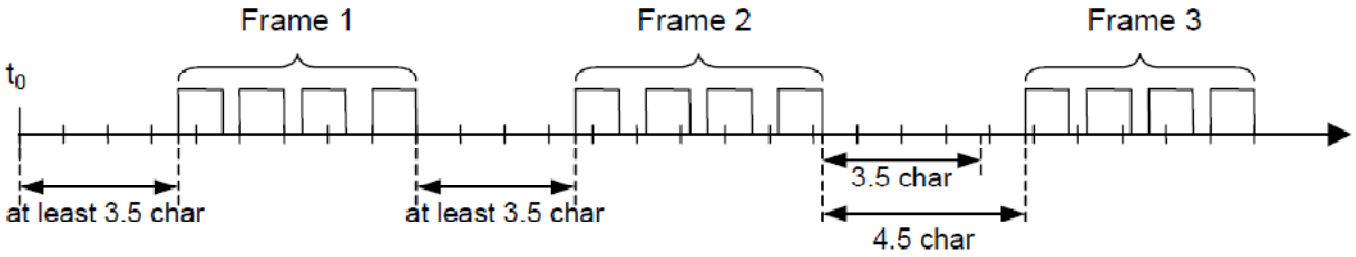
التشكيل MODEBUS لرسالة الاستعلام (Query) المرسلة من القائد (Master Device) للمقاد (Slave Device):

Slave Address	Function Code	Byte Count	Start Address	Data Stream	CRC CRC Low	CRC Hi
1 byte	1 byte	1 byte	2 byte	0 ~ 252 byte	1~2 byte	

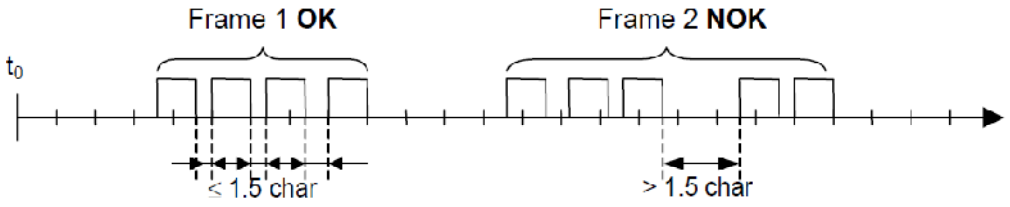
- Slave Address: يحدد عنوان الوحدة الطرفية الثانوية (1~252).
- Function code: يحدد الوظيفة المطلوبة من الإرسال.
- Byte Count: عدد البايتات التي سيتم إرسالها.
- Start Address: عنوان البايت الأول في ذاكرة المستقبل.
- Data Stream: وتحوي على حزمة البيانات المرسلة متضمنة العنوان ($255\ byte_{max}$).
- CRC (Cyclic Redundancy Check): كود تفحص خطأ الإرسال.

في حال أريد إرسال حزمتي بيانات متلاحقة، فإنه يجب أن يفصل بينهما بزمن يسمى Inter-frame-time وهو يساوي $t_{3.5}$ بناءً على الشكل التالي:





في حال إرسال عدد كبير من البايتات (Data) فيجب أن لا تكون الفترة الفاصلة بين إرسال محرفين أكثر من $t_{1.5}$، وإلا فعلى المستقبل أن يلغي العملية ويشير إلى حصول خطأ!



إن كشف هذا التأخير أو تعيينه عند سرعات نقل عالية يصبح أمراً صعباً جداً على المعالج، لذلك يتم يتعين قيم افتراضية ثابتة وهي:

Inter-character time-out (t1.5) = 750µs
Inter-frame delay (t3.5) = 1.750ms

بافتراض أن معدل الإرسال لن يتجاوز 9600bps، كما أن عدد البايتات الأعظمي في كل حزمة لن يتجاوز 70_{Byte}، وبالتالي يمكن افتراض فترة تأخير زمني ثابتة بين حزم البيانات المتلاحقة، كما أن هذا الزمن هو زمن تعطيل الاستقبال للنافذة التسلسلية للوحدات الثانوية الغير معنية بالعنوان Slave address.

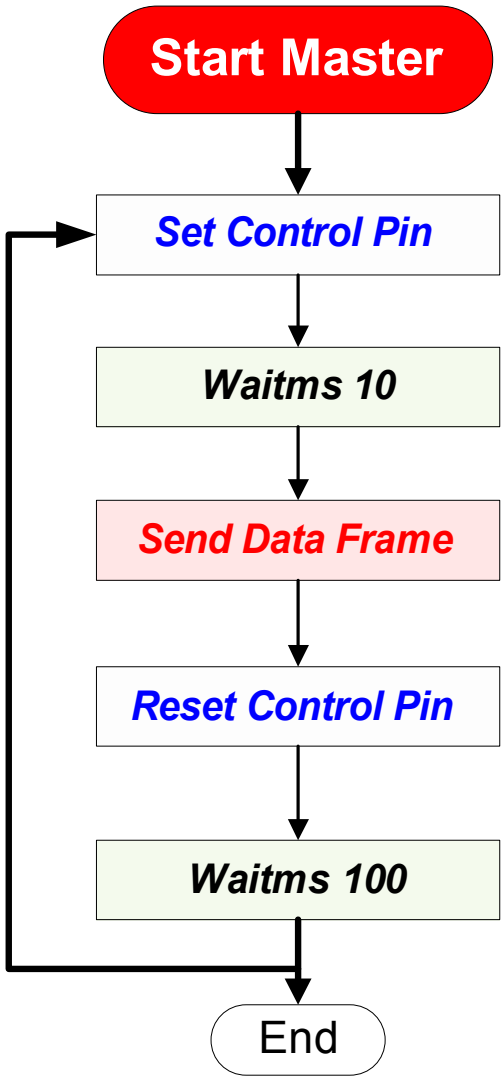
يمكن حساب زمن الإرسال تبعاً لمعدل نقل البيانات بالشكل التالي:

$$70_{Byte} \times 8_{Bit} = 560_{Bit}$$

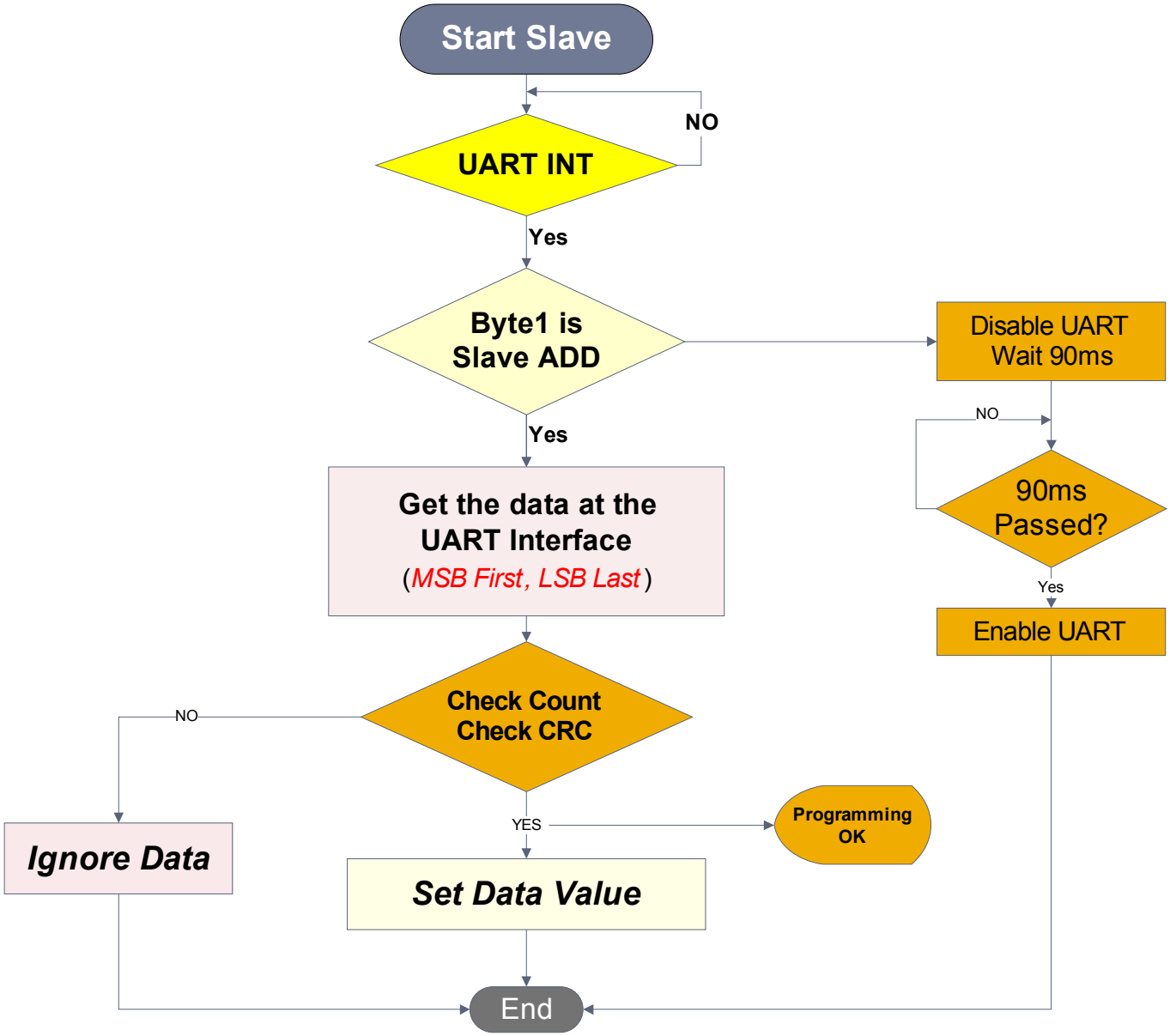
$$\frac{560}{9600} \times 1000 = \mathbf{58.3\ ms}$$

وبالتالي فإن الوحدات الثانوية الغير معنية يجب أن تعطل الاستقبال زمناً أكبر من 58ms وبعدها تعود لوضع الاستقبال، وأيضاً يجب على الوحدة الرئيسية القائدة أن لا ترسل أي حزمة جديدة إلا بعد انقضاء زمن الإرسال مضافاً إليه الزمن 1.75ms =t3.5.

الشكل التالي مخطط خوارزمية الإرسال من الوحدة الرئيسية.



الشكل التالي مخطط خوارزمية الاستقبال للوحدات الفرعية.



التشكيل MODEBUS لرسالة الاستجابة (Responce) المرسلة من المقاد (Slave) للقائد (Master):

بعد أن يقوم القائد بإرسال رسالة الاستعلام، يقوم المقاد المعني بالعنوان Slave بالرد على رسالة الاستعلام، وبالتالي فإن رسالة الرد تقتصر على تأكيد الرسالة أو ربما يتوجب على المقاد إرسال بيانات مطلوبة منه، وبالتالي فإن تشكيل رسالة الاستجابة يمكن أن يكون له عدة أشكال تبعاً لعمل النظام.

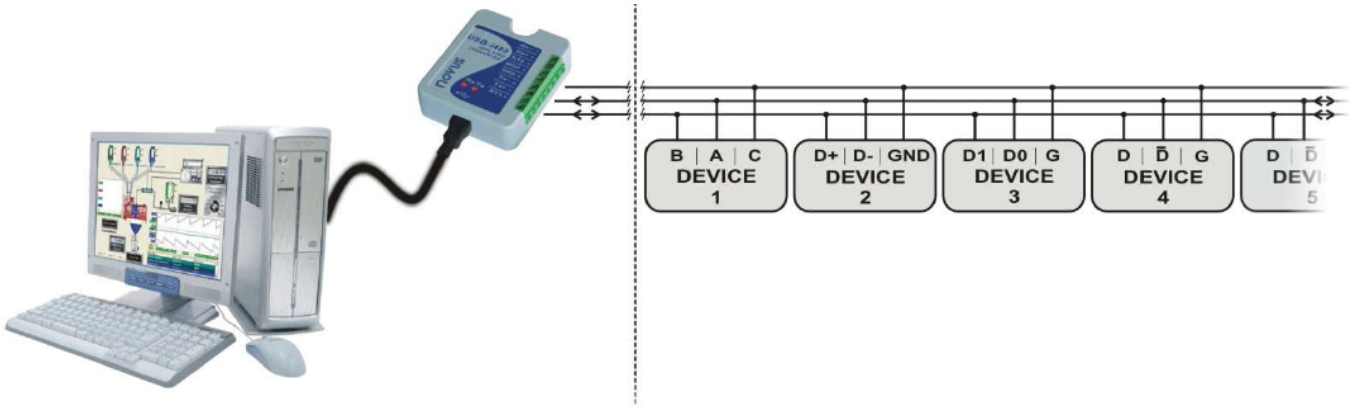
Slave Address	Function Code	Byte Count	CRC	
			CRC Low	CRC Hi
1 byte	1 byte	1 byte	1~2 byte	

مثلاً: في رسالة الرد أعلاه يقوم المقاد بإرسال عنوانه (Slave) والوظيفة التي استلمها وعدد البايتات التي تلقاها ويحسب قيمة الـ CRC.

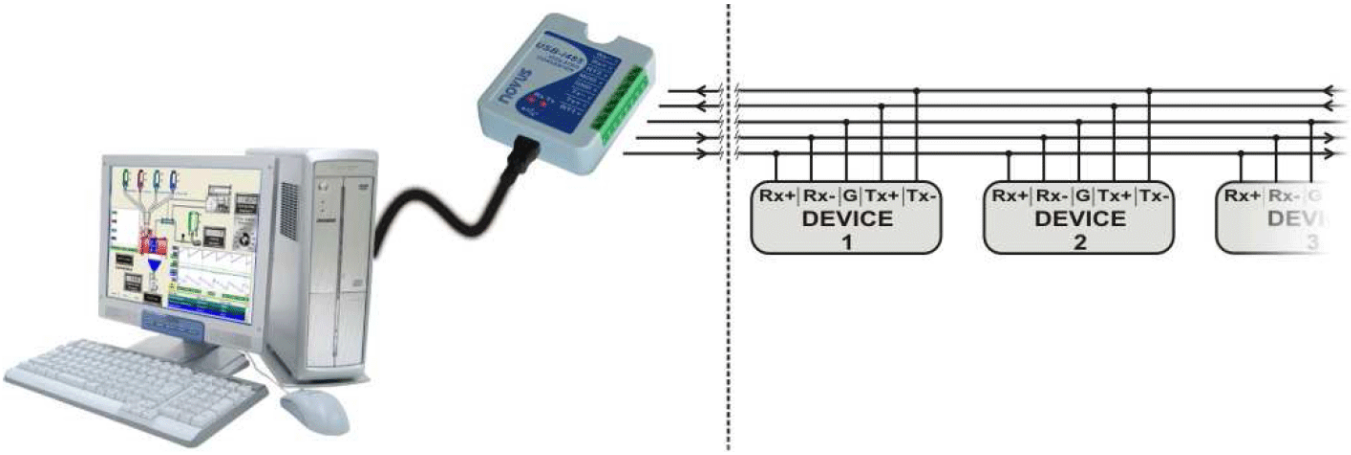
نمط الإرسال أحادي | ثنائي الاتجاه للبروتوكول RS485:

إن البروتوكول RS485 يمكن أن يعمل في نمطين: أحادي الاتجاه (Half-Duplex) وثنائي الاتجاه (Full-Duplex)، وقد تم شرح المبدأ في المحاضرة السابقة.

الشكل التالي توصيل شبكة أحادية الاتجاه.



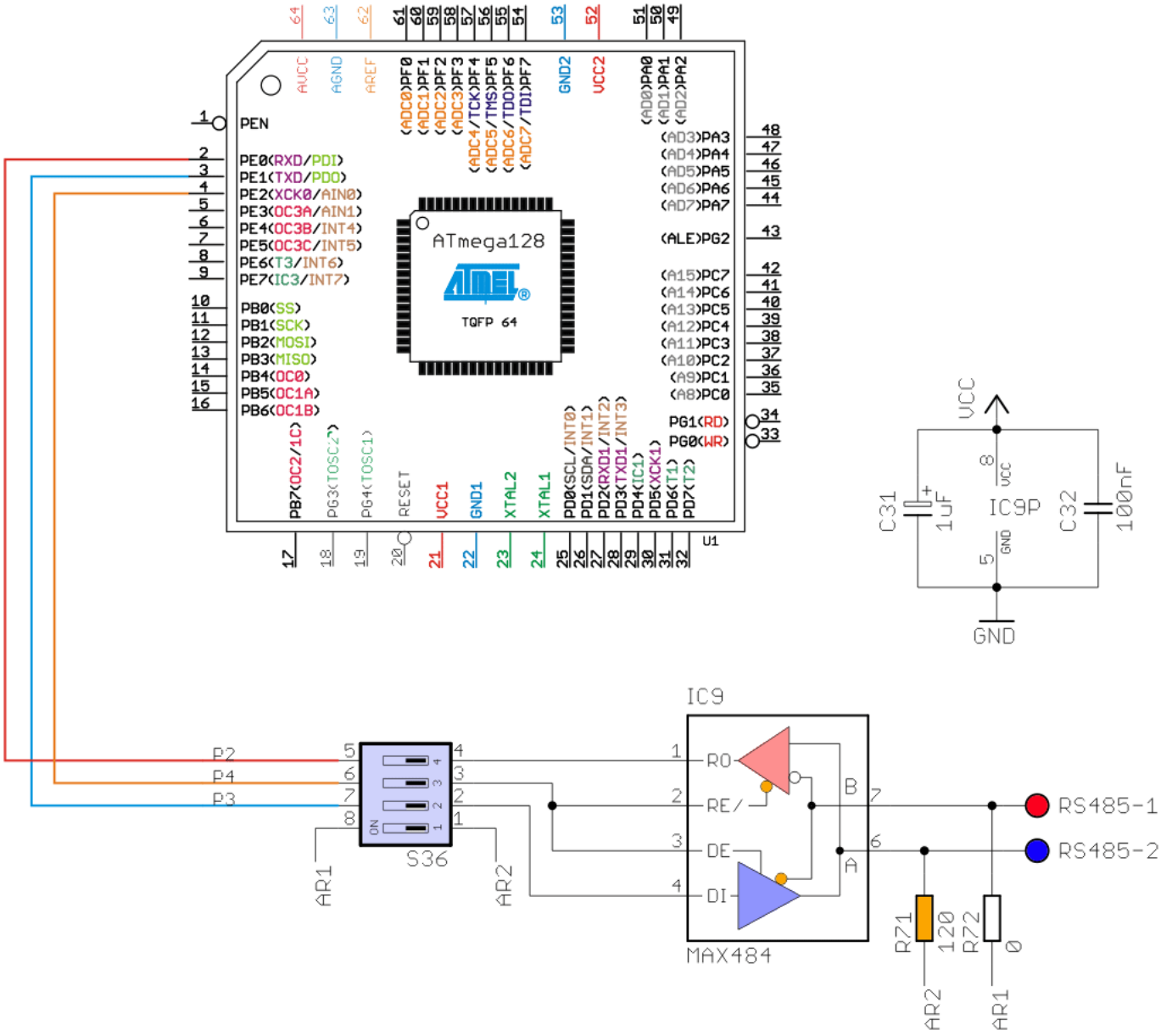
الشكل التالي توصيل شبكة ثنائية الاتجاه.



إن الشبكات RS485 أحادية الاتجاه أكثر شيوعاً، ولكن في بعض التطبيقات التي تتطلب تحكماً ومراقبةً للنظام في الزمن الحقيقي تستخدم الشبكات ثنائية الاتجاه.

مخطط دارة الاستقبال والإرسال عن طريق المتحكم المصغر (UART<>RS458):

الشكل التالي توصيل متحكم مصغر مع الدارة المتكاملة Max485.



التعليمات الجديدة:

شرح التعليمة	التعليمة
تعريف قطب التحكم بالاتجاه لدارة المحول RS485.	`Config Print0 = Porte.2 , Mode = Set`

ملاحظة: إن التعليمة السابقة (`Config Print0`) سوف تقوم بتطبيق "1" على القطب "DE" عند طلب الإرسال باستخدام التعليمة "DE"، ومن ثم تعيد حالة القطب المذكور إلى "0" فور انتهاء عملية الإرسال.

يمكن إجراء عملية التحكم على القطب "DE" بشكل يدوي والاستغناء على التعليمة السابقة وخصوصاً عندما يراد أن تكون الحالة المنطقية عائمة!

برنامج تشغيل الدارة كقائد (Microcontroller is Master, PC is Slave):

التوجيهات.

تعريف البوابة الموصل معها قطب التحكم DE.

تعريف المتحولات

```
$regfile = "m128def.dat"
$crystal = 8000000
$baud = 9600
'------------------
Config Print0 = Porte.2 , Mode = Set
Config Pine.2 = Output

Config Pine.4 = Input
Key1 Alias Pine.4
Porte.4 = 1
'------------------
Dim Msg As String * 100
'------------------
Do
   Debounce Key1 , 0 , Sendall , Sub
Loop
End
'------------------
Sendall:
   Print "This is RS485 Test Program"
   Print "RS458 Protocol Done Based-on MAX485"
   Print "University of Aleppo - Syria"
   Print "Faculty of Alectrical & Electronic Engineering"
   Print "Control Department"
   Print "Fourth Year Students"
   Print "Computer Aided Design Session"
   Print "Unfutunality This Was The Last Session Lab"
   Print "2nd Semester"
   Print "See You Next Semester Guys ;)"
   Print "Kindest Regards"
   Print "Walid BALID"
   Print ":) :) :) :) :) :) :) :) :) :) :) :)"
   Wait 1
Return
```

برنامج تشغيل الدارة كمقاد (Microcontroller is Slave, PC is Master):

التوجيهات.	`$regfile = "m128def.dat"`
	`$crystal = 8000000`
	`$baud = 9600`

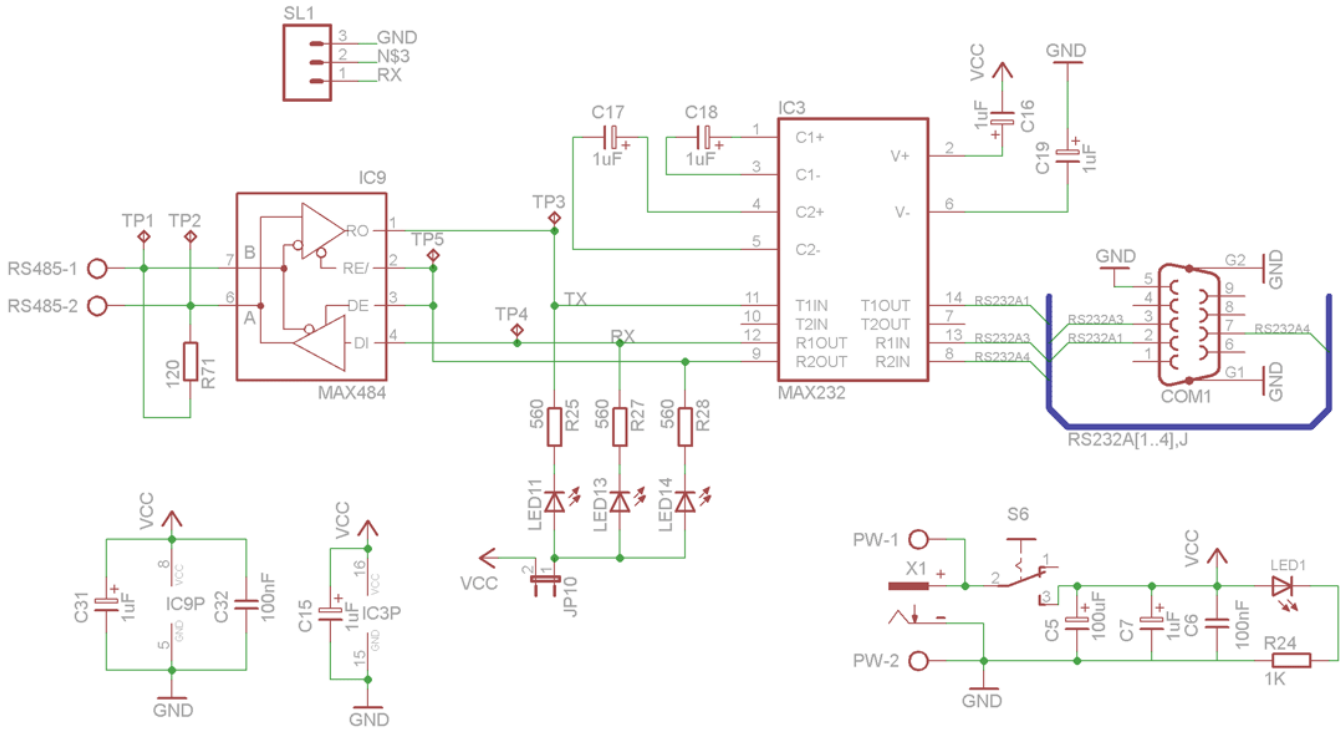
تعريف البوابة الموصل معها قطب التحكم DE.	`'-------------------------`
	`Config Porte.2 = Output`
	`Config Print = Porte.2 , Mode = Reset`

تعريف نافذة تسلسلية UART برمجية.	`Open "comd.3:9600,8,n,1" For Output As #1` `Open "comd.2:9600,8,n,1" For Input As #2`
تعريف مسجل بفر لدخل النافذة التسلسلية	`Config Serialin = Buffered , Size = 250` `Enable Interrupts`
تعريف المتحولات	`'-------------------------` `Dim Msg As String * 250`

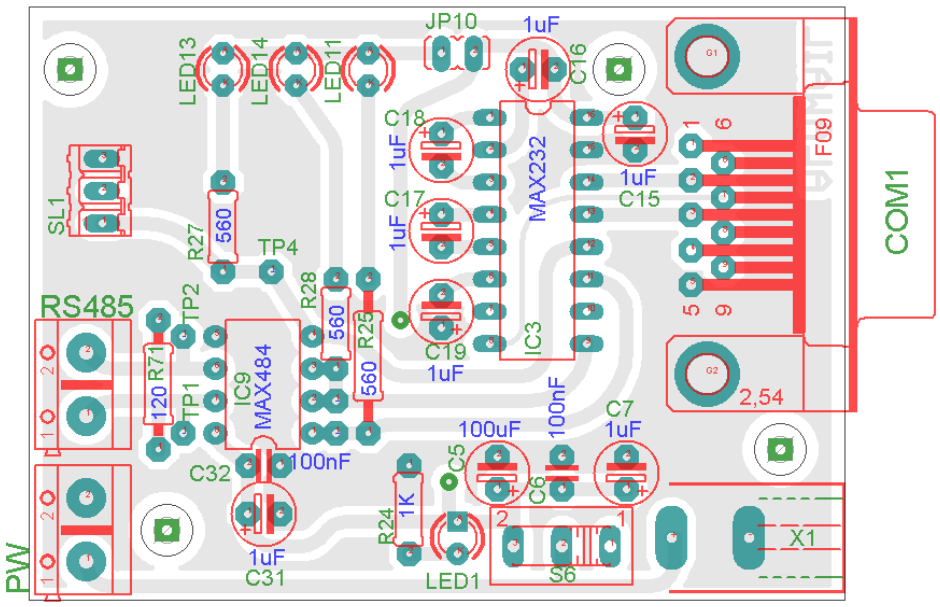
حلقة البرنامج الرئيسي يتم فيها:
فحص مسجل دخل النافذة UART الرئيسية.
قراءة وطباعة جميع البيانات الواردة على النافذة UART البرمجية.
	`'-------------------------`
	`Do`
	`  If Ischarwaiting() = 1 Then`
	`  Input Msg`
	`  Print #1 , Msg`
	`  End If`
	`Loop`
	`End`

مخطط دارة الاستقبال والإرسال عن طريق الحاسب (RS232<>RS458):

الشكل التالي المخطط النظري لدارة تحويل RS458<>RS232.

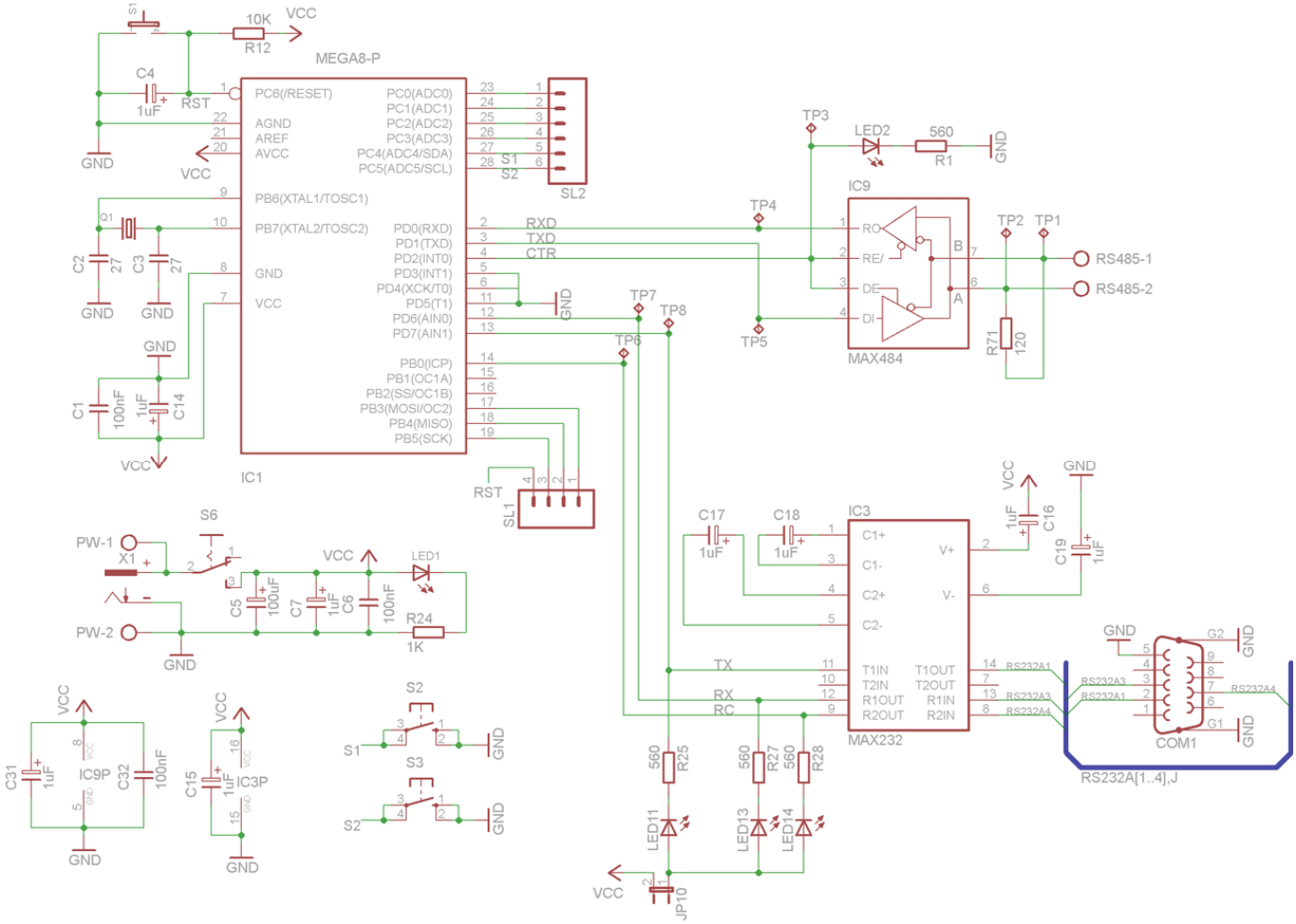


الشكل التالي مخطط الدارة المطبوعة وتوزع العناصر لدارة تحويل RS458<>RS232.

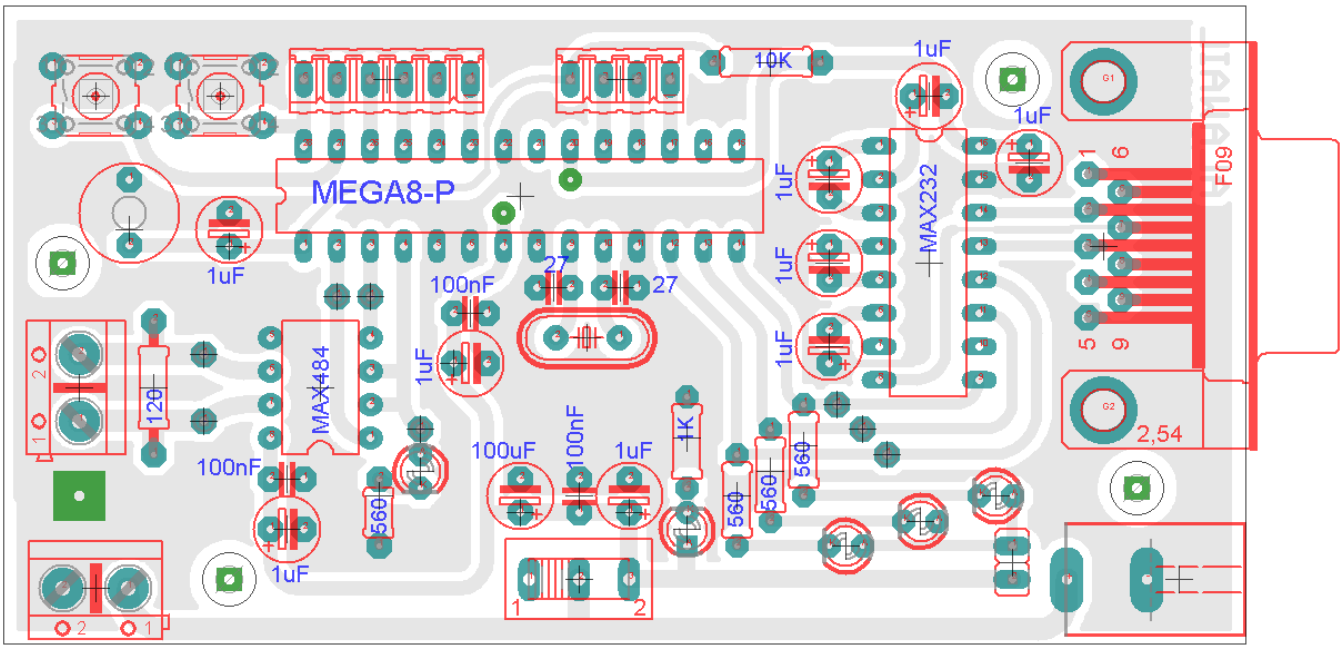


مخطط لوحة اختبار لدارة استقبال وإرسال عن طريق الحاسب ومتحكم (RS232<>UART<>RS458):

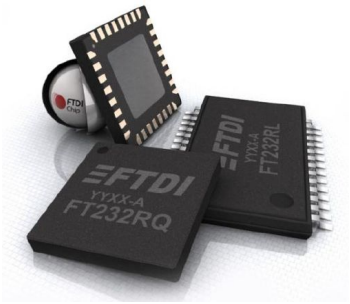
الشكل التالي المخطط النظري لدارة تحويل RS232<>UART<>RS458. في هذه الدارة يتعامل المتحكم مع الشريحة Max485 الموصولة مع النافذة UART الأساسية، ويتعامل بنفس الوقت مع الشريحة Max232 الموصولة مع نافذة UART برمجية، وبالتالي يمكن للمتحكم أن يكون وسيطاً بين البروتوكولين.



الشكل التالي مخطط الدارة المطبوعة وتوزع العناصر لدارة التحويل RS232<>UART<>RS458.



التجربة التاسعة والعشرون: البروتوكول USB Exp.29:Serial Communications USB



الغاية من التجربة:

برمجة واستخدام بروتوكول الاتصال التسلسلي USB في أنظمة التحكم الرقمي.

شرح عمل الدارة:

إن بنية البروتوكول USB شديدة التعقيد، لذلك لن نناقش بالتفصيل في هذه العجالة مبدأ عمل البروتوكول USB وإنما سوف نقدم إحدى طرق استثمار المنفذ لأغراض الربط مع الحاسب.

ميزات منفذ الاتصالات التسلسلي USB:

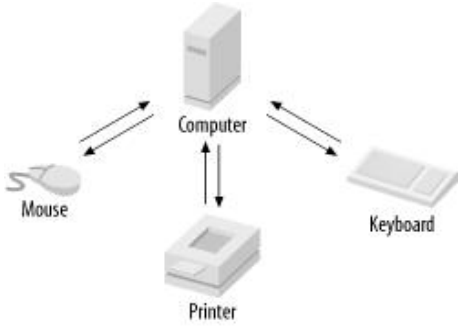
- السرعة العالية التي تصل إلى 480Mb/s.
- دعمه لتقنية Plug & Play حيث يمكن وصل الجهاز دون الحاجة لإعادة إقلاع الجهاز.
- عدد أقل من الخطوط، لأن التقنية تسلسلية تستخدم فقط أربع خطوط.
- إمكانية ربط عدة طرفيات حتى 127 طرفية على نفس الممر.

معايير البروتوكول USB:

يوجد هناك عدة معايير للبروتوكول USB: USB 3.0 & USB 2.0 & USB 1.1 & USB 1.0 تدعم أربع سرعات:

- السرعة المنخفضة Low Speed بسرعة 1.5 Mbits/s في الإصدار USB 1.0.
- السرعة الكاملة Full Speed بسرعة 12 Mbits/s في الإصدار USB 1.1.
- السرعة العالية High Speed بسرعة 480 Mbits/s في الإصدار USB 2.0.
- الجيل الجديد بسرعات عالية جداً 4800 Mbits/s في الإصدار USB 3.0.

بنية ممر USB:



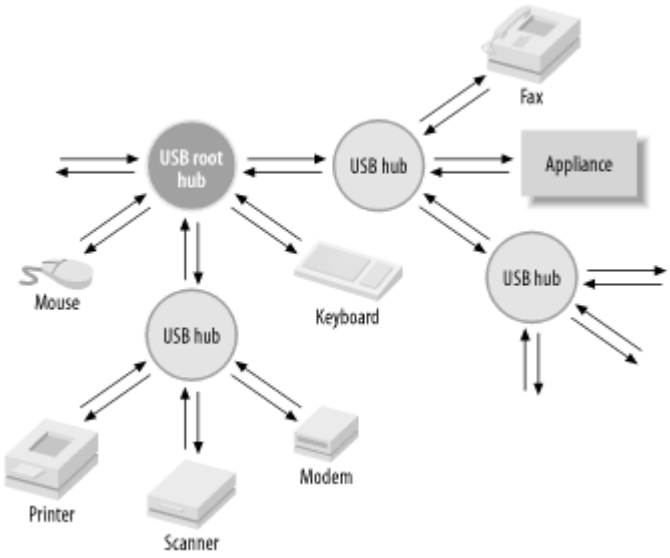
يتصف ممر USB بأنه "Host Controlled" أي يوجد هناك جهاز مضيف واحد على الممر (وهو عبارة عن جهاز يتحكم بالممر وببقية الأجهزة الموصولة معه أثناء العمل) يقع على عاتقه عدة مسؤوليات مثل تهيئة عمليات النقل وتوزيع عرض الحزمة، ولا تسمح تقنية USB بوجود أكثر

من جهاز مضيف واحد على الممر، إلا أنه صدر معيار جديد يدعى"On-The-Go" يسمح بإمكانية وجود جهازين يتم التفاوض فيما بينهما لتبادل دور الجهاز المضيف.

يتمتع ممر USB ببنية نجمية مركزها الجهاز المضيف، تتصف هذه البنية بعدة مزايا مثل:

- إمكانية مراقبة تغذية كل طرفية على حدى.
- إمكانية إطفاء الطرفية في حال استجرارها لتيار كبير بدون أن تتأثر بقية الطرفيات.

لوصل أكثر من جهاز إلى منفذ USB واحد نقوم بوصل تلك الأجهزة إلى موزع HUB والذي بدوره يوصل إلى هذا المنفذ، ويتوفر حالياً العديد من الطرفيات التي تحوي على موزعات مدمجة فيها مثل الشاشات و لوحات المفاتيح.



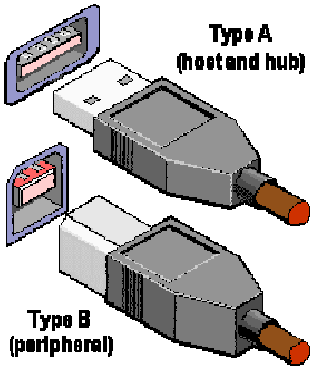
تقنية Plug& Play:

يدعم ممر USB تقنية Plug & Play مع إمكانية تحميل وإزالة ديناميكية لبرنامج القيادة، حيث يقوم المستخدم بوصل الطرفية على ممر USB فيقوم الجهاز المضيف بكشف اتصال هذه الطرفية، و يقوم بالبحث عن برنامج القيادة المناسب لهذا الجهاز ومن ثم يقوم بتنصيبه. إن حجز موارد المقاطعة وعناوين المنافذ يتم بشكل آلي، وكل هذا بدون الحاجة لإعادة إقلاع الحاسب.

بمجرد أن يقوم المستخدم بإزالة هذه فإن الجهاز المضيف سيكشف غياب هذه الطرفية ويزيل برنامج التعريف الخاص بها. تتم عملية تحميل برنامج القيادة المناسب باستخدام رقمي VID, PID .

يدعم ممر USB عدة أنواع من أنماط النقل يتميز كل نمط عن الآخر بإمكانية كشف الأخطاء و تصحيحها، وعرض الحزمة المخصص، و زمن الوصول فيما إذا كان ثابت أم لا.

المواصفات الميكانيكية والكهربائية لمنفذ USB:

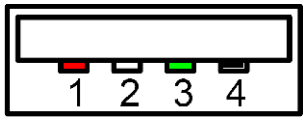
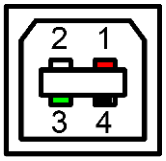


تحدد معايير USB نوعين من المآخذ :

- مأخذ نوع A: يوجد على الحاسب المضيف.
- مأخذ نوع B: يوجد على الطرفية.

يمنع هذا الاختلاف في الشكل بين المأخذين من وصل طرفيتين أو جهازي حاسب مع بعضهما البعض.

يستخدم ممر USB أربع خطوط: خطي تغذية GND & +5v، خطي معطيات D+ & D-.

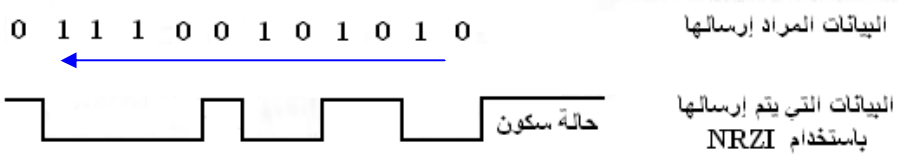
1 VCC (أحمر) 2 D- (أبيض) 3 D+ (أخضر) 4 GND (أسود)		

يستخدم ممر USB الخطين D+&D- لإرسال البيانات بشكل تفاضلي، فمثلاً لإرسال '1' منطقي بشكل تفاضلي على الممر يجب وضع '1' منطقي على القطب D+، '0' منطقي على القطب D- وذلك في حال السرعة الكاملة والمنخفضة للممر، وينعكس هذا التشفير في حال السرعة العالية.

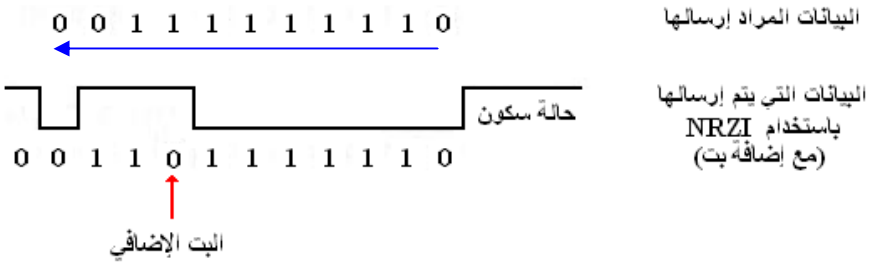
تقنية التشفير NRZI (Non Return to Zero Invert):

يستخدم ممر USB تقنية NRZI لتشفير البيانات المرسلة على الممر للحماية من الضجيج و للمحافظة على التواقت بين المرسل والمستقبل

تعتمد هذه التقنية على تغيير حالة الممر من '0' إلى '1' أو بالعكس عندما يراد إرسال '0' أما إذا كان البت المراد إرساله هو '1' فتبقى حالة الممر كما هي.



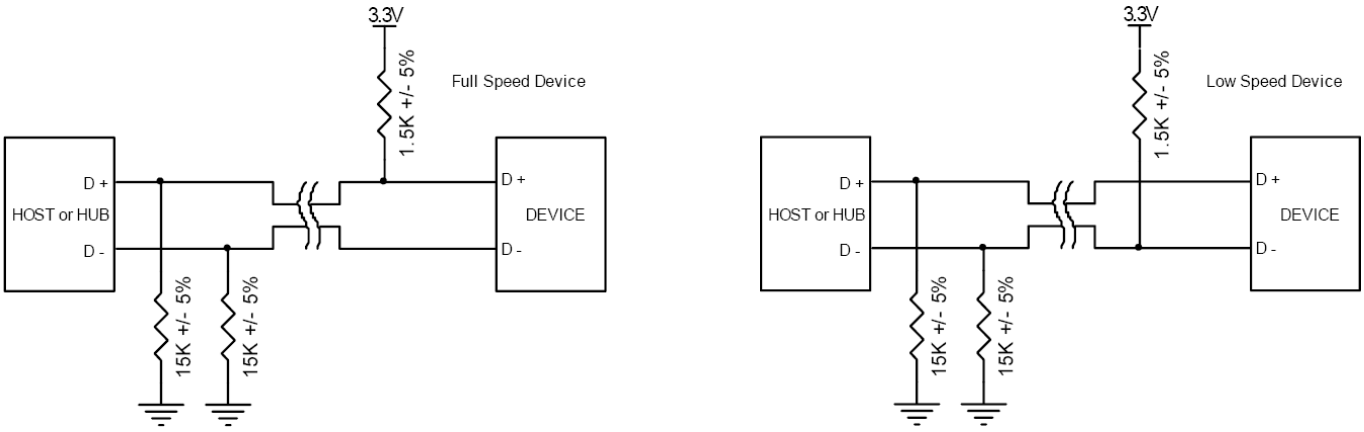
عندما يراد إرسال أكثر من ستة خانات بقيمة '1' منطقي فإنه يتم الفصل بين كل ستة من هذه الخانات بخانة إضافية تحمل القيمة صفر، و ذلك لضمان استمرار عملية التزامن بين المرسل و المستقبل.



تحديد سرعة الطرفية:

يتمكن الحاسب المضيف من تحديد سرعة الطرفية المربوطة بممر USB عن طريق مقاومة شد تربط إلى أحد قطبي البيانات D+ D- في الطرفية، ونستنتج الحالات التالية:

1. الطرفية تدعم السرعة المنخفضة: في هذه الحالة تربط هذه المقاومة إلى القطب D-.
2. الطرفية تدعم السرعة الكاملة: في هذه الحالة تربط هذه المقاومة إلى القطب D+.
3. الطرفية تدعم السرعة العالية: تكون طريقة الربط مماثلة لحالة السرعة الكاملة.



بنية البروتوكول USB:

يحوي برتوكول USB على العديد من المصطلحات الجديدة المرتبطة به، ولذلك لابد لنا من التعرف على هذه المصطلحات قبل الخوض في شرح بروتوكول ممر USB.

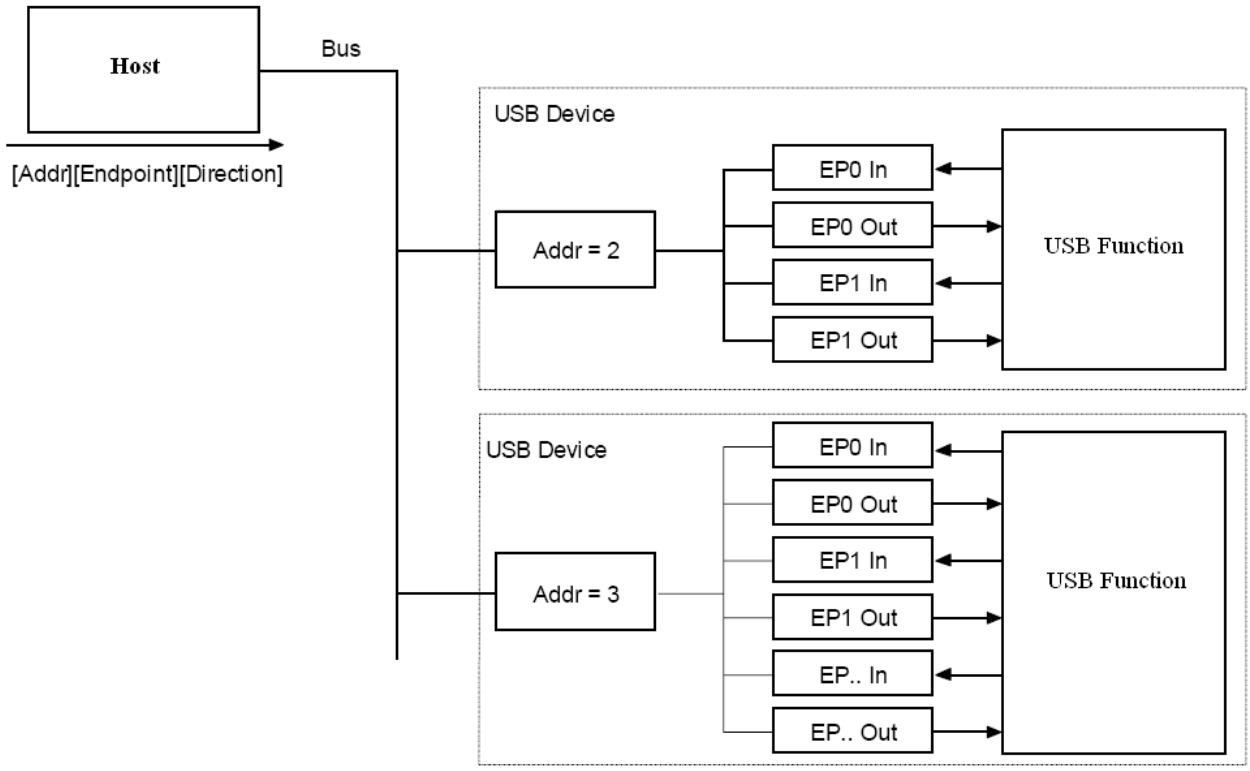
جهاز USB (USB Device): هو تعبير عام قد يشير إلى وحدة طرفية أو حاسب مضيف أو موزع أو إلى دارة متحكم USB في المضيف (Host Controller IC) ولتجنب هذه العمومية تم استخدام مصطلح (USB Function) حيث يقوم المضيف بتخصيص عنوان فريد لكل جهاز USB أثناء عملية التهيئة.

USB Function: عبارة عن مصطلح يشير إلى جهاز USB يقوم بوظيفة معينة مثل الطابعة أو الماسح الضوئي.

النقطة النهائية (Endpoint): يحوي USB Functionعلى عدد من المسجلات التي تمثل صلة الوصل بين الجهاز المضيف وبين USB Function، يدعى كل مسجل من هذه المسجلات بالنقطة النهائية (Endpoint)، و تكون هذه النقاط إما كمصدر للبيانات (Out) أو كمتلقي لها (In) وذلك من وجهة نظر المضيف.

تملك كل نقطة نهائية عنوان فريد ضمن جهاز الـ USB نفسه، و هذا العنوان هو عبارة عن رقم هذه النقطة ونوعها، واختصاراً نرمز للنقطة النهائية الثانية مثلاً من النوع متلقي بالرمز EP2 In.

إن كل عملية اتصال تنشأ بين USB Function والمضيف تكون عملياً بين أحد النقاط النهائية والمضيف كما هو موضح في الشكل التالي:



على سبيل المثال يتمكن المضيف من الوصول إلى النقطة النهائية EP1 In في جهاز USB1 بإرسال عنوان جهاز USB1 ورقم النقطة النهائية ونوعها كما يلي:

[Addr : 2] [Endpoint : EP1] [In]

Pipe: يصطلح على تسمية قناة الاتصال بين أحد النقاط النهائية والمضيف بـ Pipe والذي يمتلك عدة خواص منها: عرض الحزمة المخصص له، نوع البيانات التي ستنقل عليه (تحكم، مقاطعة، كتلية)، اتجاه حركة البيانات، حجم رزمة البيانات الأعظمي. فمثلاً: Pipe الافتراضي هو Pipe0 ثنائي الاتجاه ويتألف من: EP0 In & EP0 Out وبيانات النقل هي للتحكم.

أنواع النقاط النهائية:

يرتبط مفهوم نوع النقاط النهائية بمفهوم نوع النقل الذي يتم عبر Pipe لذلك فإن أنواع النقاط النهائية هي نفسها أنواع النقل والتي تقسم إلى:

- عمليات النقل الخاصة بالتحكم (Control Transfers).

- عمليات النقل الخاصة بالمقاطعة (Interrupt Transfers).
- عمليات النقل التي تتم بشكل دوري وبزمن ثابت (Isochronous Transfers).
- عمليات النقل الخاصة بنقل الكتل (Bulk Transfers).

طريقة عمل البروتوكول USB:

يتبادل منفذ USB البيانات من خلال إطارات (Frame)، لكل إطار فترة زمنية ثابتة تتغير بحسب نوع المنفذ (USB1.1, USB2.0) والطرفية المستخدمة، فمثلاً: يكون طول الإطار (1ms) في حالة كانت السرعة المستخدمة هي سرعة منخفضة أو كاملة، في حين يكون عند السرعة العالية (125µs).

يحوي كل إطار عدد من عمليات تداول البيانات (Transaction) وكل عملية تداول تتألف من مجموعة من رزم البيانات.

تقسم هذه الرزم إلى الأنواع التالية:

Token Packet: ترسل في بداية كل عملية تداول للبيانات وتحوي معلومات عن عملية التداول المراد إجرائها وتقسم إلى أربعة أنواع:

In Token: تقوم هذه الرزمة بإخبار الطرفية بأن المضيف يريد قراءة بيانات، وتتألف من خمسة حقول:

1. Synchronization: تتألف من ثمانية بتات وتوجد في بداية كل أنواع الرزم، وتستخدم لمواقتة المرسل مع المستقبل.
2. PID (Packet Identity): يتألف من ثمانية بتات تستخدم للدلالة على نوع الرزمة المرسلة، وتكون ثاني أربع خانات هي متممة لأول أربع خانات وذلك للتأكد من أن البيانات صحيحة.
3. Address: يحوي هذا الحقل عنوان الطرفية المطلوبة ويتألف من سبع خانات مما يسمح بعنونة 128جهاز.
4. Endpoint: يتألف من أربع خانات تدل على رقم النقطة النهائية المراد مخاطبتها.
5. CRC (Cyclic Redundancy Check): بطول 5 بت من أجل كشف الأخطاء.

Out Token: تقوم هذه الرزمة بإخبار الطرفية بأن المضيف يريد إرسال بيانات، وتتألف من نفس حقول الرزمة السابقة (In Token).

Setup Token: تستخدم هذه الرزمة للدلالة على بداية عملية تهيئة Control Transfer وهي تتألف من نفس حقول رزمة In Token.

Start of frame Token : يقوم المضيف بإرسال هذه الرزمة في بداية كل إطار أي كل 1ms±500ns ، وتحتوي هذه الرزمة على رقم الإطار ضمن حقل (Frame number) المؤلف من 11 خانة.

In	SYNC 0x01	PID 0x96	Address 7 bits	End point 4 bits	CRC 5 bits
Out	SYNC 0x01	PID 0x1E	Address 7 bits	End point 4 bits	CRC 5 bits
Setup	SYNC 0x01	PID 0xD2	Address 7 bits	End point 4 bits	CRC 5 bits
Start of frame	SYNC 0x01	PID 0x5A	Frame number 11 bits		CRC 5 bits

رزمة البيانات (Data Packet): وتقسم إلى نوعين:

1. رزمة بيانات من نوع Data0.
2. رزمة بيانات من نوع Data1.

لكلا نوعي رزم البيانات حقل بيانات بطول 1024bit وحقل CRC بطول 16bit وذلك بسبب كبر هذه الرزمة.

Data 0	SYNC 0x01	PID 0x3C	Data 0-1023 bits	CRC 16 bits
Data 1	SYNC 0x01	PID 0xB4	Data 0-1023 bits	CRC 16 bits

رزمة المصافحة (Handshake Packet): ولها ثلاثة أنواع:

Acknowledge: وتدل على أن عملية التداول قد تمت بنجاح وأنه تم استقبال البيانات بشكل صحيح.

Not acknowledge: ترسل هذه الرزمة للدلالة على أن الجهاز مشغول لا يمكنه بشكل مؤقت إرسال أو استقبال البيانات، وترسل أيضا للمضيف خلال عمليات التداول الخاصة بالمقاطعة لإخباره بعدم وجود بيانات.

Stall: تدل هذه الرزمة على أن النقطة النهائية في حالة توقف بسبب مشكلة ما وتتطلب تدخل المضيف أو أن الأمر المرسل غير مدعوم.

حلول التطوير باستخدام منفذ الاتصالات التسلسلي USB:

تعتبر تقنية USB في الوقت الحالي من التقنيات المعقدة حيث أن تضمين منفذ USB في النظام الإلكتروني وكتابة برنامج القيادة الخاص به على الحاسب أمر شديدة التعقيد، وذلك لأنه يتوجب على المصمم تحقيق أمرين:

1. تصميم عتاد الكتروني (Hardware) يحقق معايير البروتوكول USB.
2. كتابة برنامج التعريف الخاص بقيادة هذا العتاد.

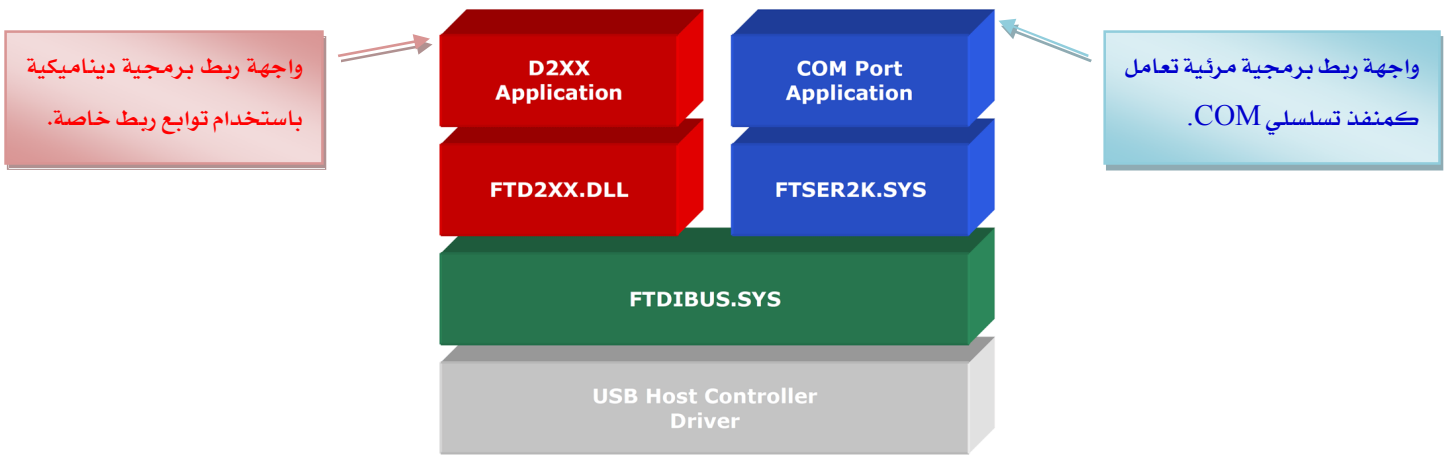
لذلك وبسبب الطلب المتزايد على هذه التقنية واقتحامها للسوق العالمية فإن هنالك الكثير من الشركات التي وفرت على المصممين عناء تصميم العتاد الالكتروني لينصّب اهتمامهم على كتابة برامج القيادة، لذلك كل ما يتوجب على المصمم هو الإطلاع على معايير USB بغرض فهم كيفية التعامل مع هذا العتاد الالكتروني.

تقدم بعض الشركات حلولاً للتعامل مع المنفذ USB باستخدام شرائح متكاملة تقوم على تحويل البروتوكول USB إلى نافذة تسلسلية UART تمكن المستخدم من توصيل المتحكم المصغر بشكل مباشرة مع هذه النافذة، بالإضافة إلى ذلك توفر هذه الشرائح حلولاً برمجية من خلال مكتبات ربط ديناميكية من أجل ربط نظام مع الحاسب عن طريق البروتوكول USB ومعالجة بارامترات النظام أو إرسال أوامر التحكم إلى النظام.

من أشهر وأكثر الشرائح انتشاراً واستخداماً هي الدارة المتكاملة FT232 التي هي عبارة عن دارة تحويل USB<>UART التي تنتجها شركة **FTDI**.

إن عملية تحويل البروتوكول USB تم بنائها في داخل هذه الشريحة ككيان صلب (Hardware) دون الحاجة إلى برمجة الشريحة، حيث تؤمن هذه الشريحة واجهتي ربط ديناميكي للتعامل برمجياً مع المنفذ باستخدام توابع خاصة وجاهزة موجودة في مكتبات الربط الديناميكي للشريحة دون الحاجة إلى بناء البروتوكول USB بشكل برمجي من البداية أو حتى فهم مبدأ عمله.

إن واجتهي الربط (D2XX driver & VCP driver) التي تؤمنها هذه الشريحة هي على الشكل التالي:



فيما يلي جدول مقارنة بين واجتهي الربط (D2XX driver & VCP driver):

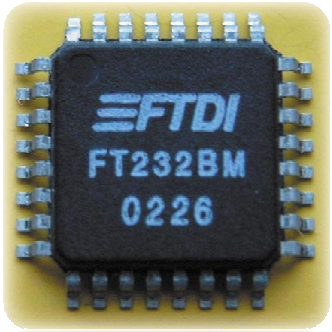
	VCP Driver	D2XX.DLL Driver
بساطة البرنامج	برنامج بسيط	برنامج معقد
السرعة	سرعة ثابتة لا يمكن تغيرها 300 KB/s	سرعة قابلة للتغيير تصل إلى 3MB
التحكم بالشريحة	لا يمكن التحكم بالشريحة	تحكم كامل ومباشر بالشريحة

VCP (Virtual Com Port): يعرف منفذ USB كمنفذ Com تسلسلي إضافي، مما يسمح لنا بالتخاطب مع منفذ USB كمنفذ Com معياري.

D2XX.DLL: يسمح هذا التعريف بالوصول المباشر إلى كامل مميزات هذه الشريحة عن طريق أوامر موجودة ضمن مكتبة ربط ديناميكيةDLL.

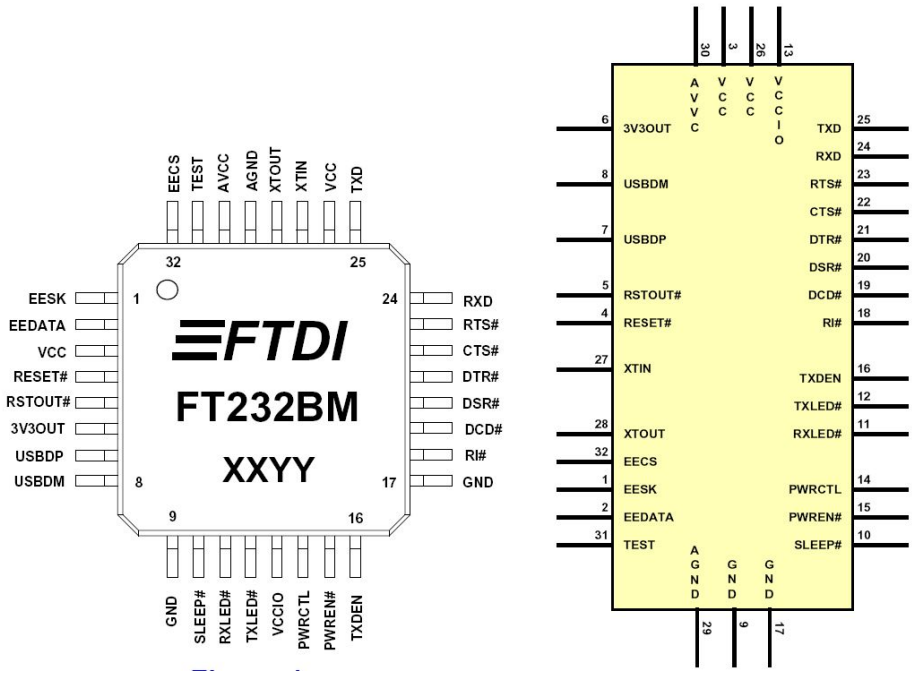
الشريحة FT232BM:

- توفر الشركة الصانعة برنامج القيادة لهذه الشريحة بشكل مجاني متوافق مع معظم أنظمة التشغيل.
- تقدم شركة FTDI برنامجي قيادة لشرائحها (VCP & D2XX.DLL).
- متوافقة مع المعيارين USB1.1, USB2.0.
- تدعم هذه الشريحة ملائمة كاملة لنظم الاتصالات التسلسلية.
- سرعة اتصال 300kb~2Mb بحسب نوع برنامج القيادة.
- ذاكرة استقبال وسيطية من نوع FIFO بطول 384 بايت.
- ذاكرة إرسال وسيطية من نوع FIFO بطول 128 بايت.
- رقمي VID, PID ورقم تسلسلي للمنتج ووصف لهذا الجهاز.
- توفر العديد من المقالات التقنية من الشركة المصنعة تقدم معلومات مفصلة عن طرق استخدام هذه الشريحة.

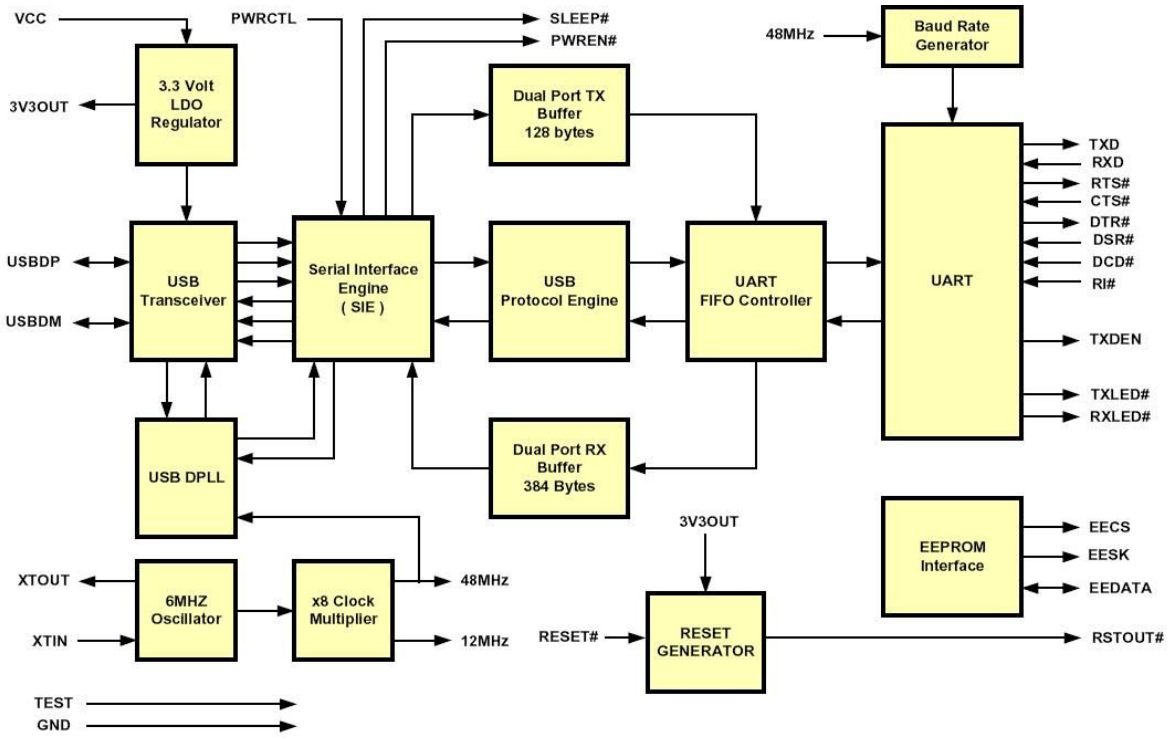


تلعب هذه الشريحة دور الملائم بين منفذ USB وبين النظام حيث تقوم باستقبال بيانات منفذ USB وتستخلص منها البيانات المطلوبة، كما تقوم بإرسال البيانات من المتحكم بشكلها التسلسلي إلى منفذ USB بعد إضافة الحقول اللازمة لتحقيق بروتوكول USB.

توزع أقطاب الشريحة:



المخطط العام لبنية هذه الشريحة مع العناصر الأساسية :



تمتلك هذه الشريحة 11 قطب للوصل مع النظام بحيث تؤمن تبادل البيانات مع الحاسب باستخدام بروتوكول المصافحة.

عند وصل الشريحة مع الحاسب وبعد أن يقوم نظام التشغيل بتحميل برنامج القيادة لها، تقوم هذه الشريحة بإعطاء صفر منطقي على القطب #PWREN وبالتالي يمكن استخدام هذا القطب لتشغيل الدارة الخارجية عند وصل النظام بالحاسب.

تدخل الشريحة في نمط الطاقة المنخفضة (Sleep mode) إذا لم تكن هناك عملية تبادل بيانات لمدة 3ms (أي بطول ثلاث إطارات)، في هذه الحالة تصبح حالة القطب "Sleep="1 ، وبالتالي فإن ربط هذا القطب مع المتحكم بطريقة مناسبة يمكن أن يدخله في نمط الطاقة التحتية أيضاً.

تمتلك هذه الشريحة قطب Wake up يسمح للنظام بإخراج الحاسب من الوضع الاحتياطي عند ورود جبهة صاعدة وذلك في حال كان النظام في حالة وضع احتياطي، أما إذا لم يكن كذلك فإن هذه الجبهة الصاعدة تؤدي إلى إرسال البيانات الموجودة في ذاكرة الاستقبال إلى الحاسب.

يتم تحقيق المتطلبات الخاصة لبرتوكول التخاطب مع منفذ USB في الشريحة من قبل وحدة الملائمة التسلسلية SIE (Serial Interface Engine) التي تقوم بالمهام التالية:

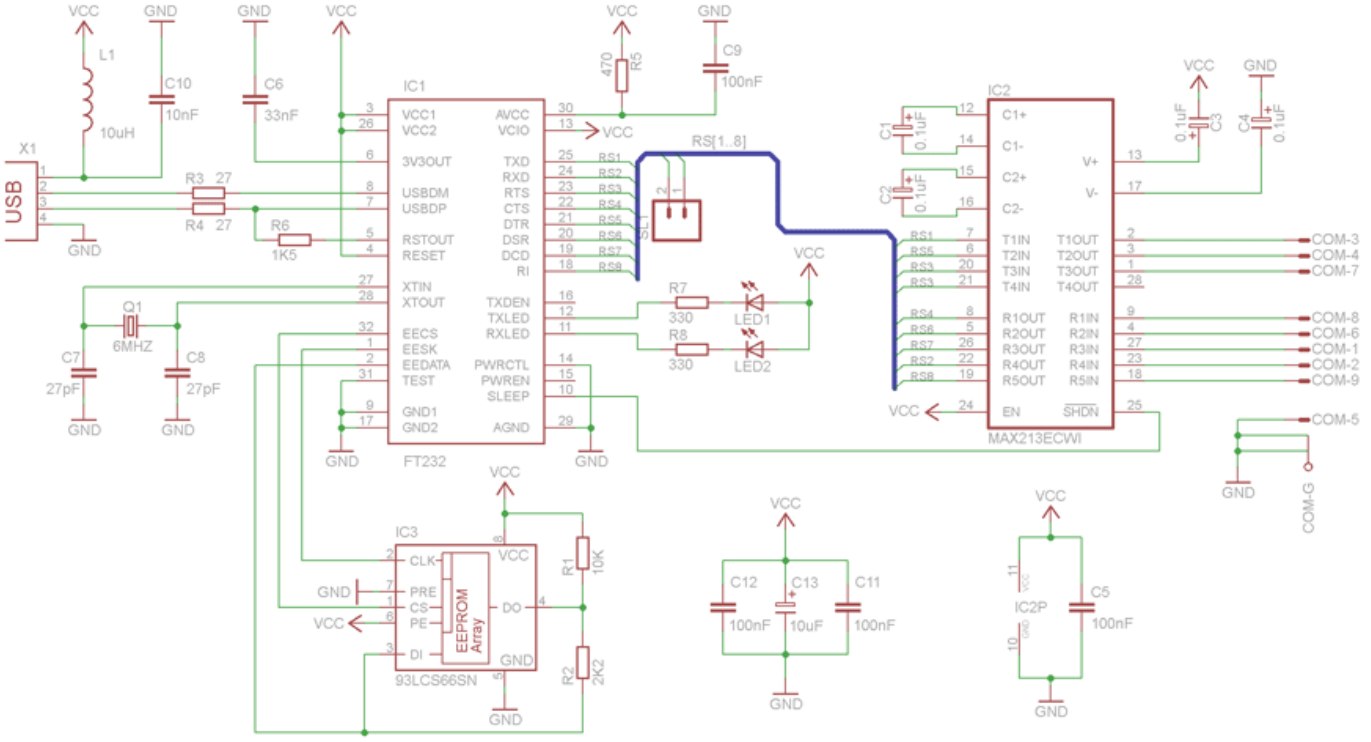
- كشف الرزم المستقبلة، وإرسال الرزم من وحدة بروتوكول USB إلى مرسل USB.
- فحص و توليد قيم CRC.
- كشف و توليد إشارات Start Of Packet, End Of Packet, Resume, Reset.
- تشفير وفك تشفير البيانات على الممر بحسب تقنية NRZI.
- فك تشفير وتوليد معرفات الرزم (PID).

تقوم وحدة (USB Transceiver) بملائمة الشريحة مع الممر وفقاً للشروط الكهربائية المحققة للمعايير USB1.1 ,USB2.0، بما في ذلك تعريف سرعة الشريحة على الممر باستخدام مقاومة شد على القطب D+ لأن سرعة الشريحة تصنف كسرعة كاملة.

دارة ملائمة الشريحة FT232BM:

تحتاج هذه الشريحة تردد 48MHz تحصل عليه بواسطة ضارب داخلي لتردد 6MHz هزاز كريستالي خارجي.
يُخزَّن كلا رقمي PID, VID والرقم التسلسلي في ذاكرة نوع EEPROM خارجية، كما يخزن أيضاً معلومات أخرى عن صنف الجهاز والتيار الأعظمي الذي يحتاجه ومعلومات أخرى.

الشكل التالي المخطط التصميمي لدارة التحويل RS232<>UART<>USB.



الشكل التالي مخطط الدارة المطبوعة وتوزع العناصر لدارة التحويل RS232<>UART<>USB.