

المتحكمات الصغيرة و النظم المضمنة

محاضرة 7

الاتصال التسلسلي

د. فادي متوج

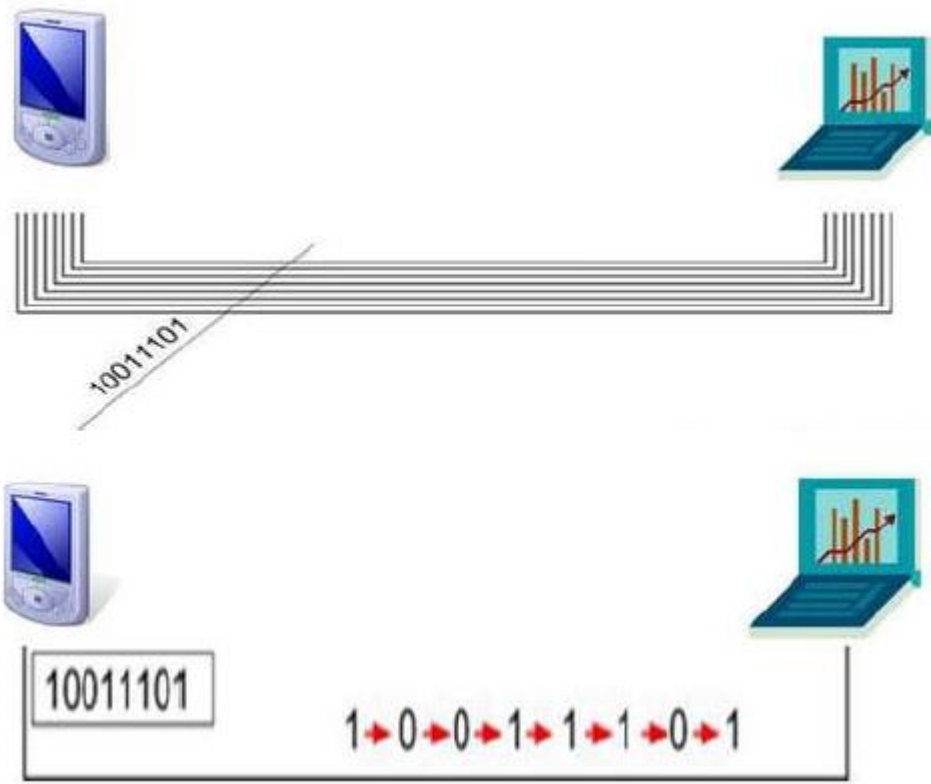
• الاتصال التسلسلي هو بروتوكول يستخدم لربط جهازين أو أكثر مثل توصيل الكمبيوتر بالميكرو أو توصيل الميكرو بميكرو آخر أو غير ذلك، وفيه يتم نقل الداتا بطريقة تسلسلية (Serial) على وصلة واحدة بين الجهازين يرسل عليها البت الأول يليه البت الثاني ... وهكذا، وهو عكس التوصيل التفرعي والذي فيه يتم نقل كامل البايت مرة واحدة على 8 وصلات (أسلاك)..



جامعة
المنارة
MANARA UNIVERSITY

الاتصال التسلسلي

Serial Communication



• وهي طريقة اتصال لنقل الداتا بين جهازين أو أكثر على أن يتم نقل الداتا بطريقة متتابعة بمعنى أن يتم إرسال البت الأول من البايت ثم بعده البت الثاني ثم الثالث ... وهكذا حتى يتم نقل البايت كاملاً،

• الشكل الاتي يوضح الفرق بين نقل الداتا بطريقة متتابعة (بالأسفل) ونقلها بالتوازي (بالأعلى).

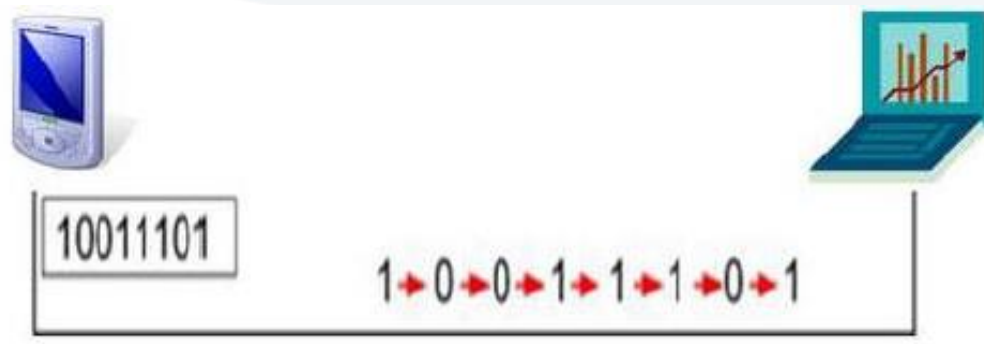


جامعة
المنارة

MANARA UNIVERSITY

الاتصال التسلسلي

Serial Communication



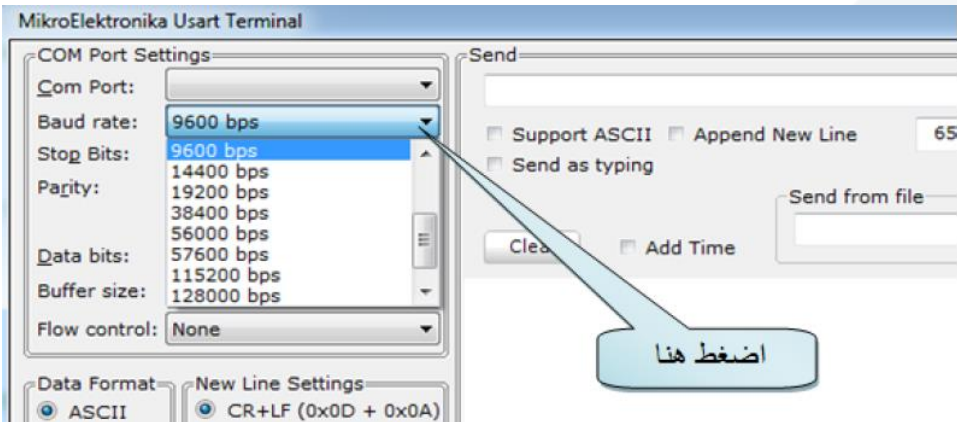
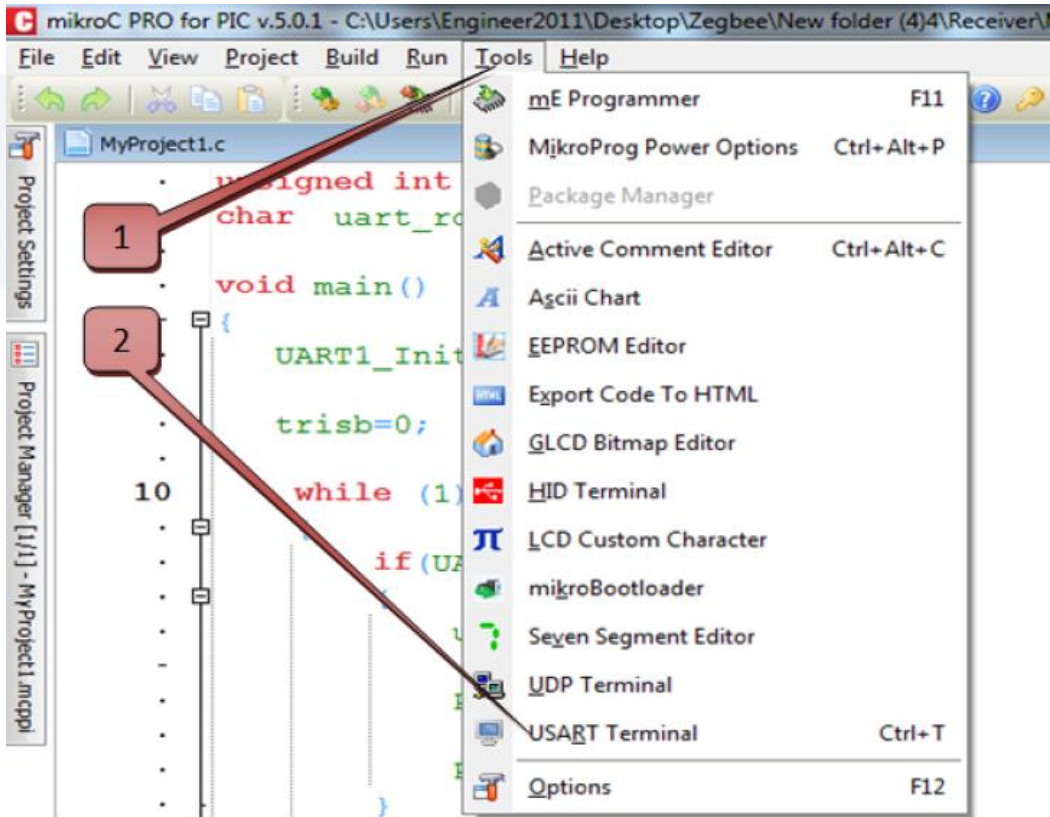
• وبالنظر إلى الشكل جانباً سنجد انه في حالة السيريال استخدمنا سلك (Line) واحد فقط، وهذا ينتج عنه مشكلة وهي أننا لا يمكننا الإرسال والاستقبال في نفس الوقت على سلك واحد، فإما أن نقوم بالإرسال أو أن نقوم بالاستقبال ...

• والحل البسيط لهذه المشكلة هو أن يتم وضع سلكين بين الجهازين يستخدم أحدهما للإرسال والآخر للاستقبال، وعندها سيكون التوصيل بين اثنين ميكروكترولر مثلاً كما بالشكل الآتي:

• نلاحظ في الشكل أن طرف الإرسال للجهاز الأول متصل على طرف الاستقبال للجهاز الثاني والعكس بالعكس في الطرفين الآخرين، وهذا منطقي إذ أنه في الوقت الذي يكون فيه الجهاز الأول يقوم بالإرسال عندئذ يقوم الجهاز الثاني بالاستقبال.

برنامج الميكرو سي

- لنقل الداتا بطريقة تتابعية لابد من تحديد مجموعة من العناصر أهمها سرعة النقل، والمقصود بالسرعة هنا هو عدد ال Bits التي سيتم إرسالها في كل ثانية، والتي لابد من تحديدها حيث إن كان المرسل يرسل البيانات بسرعة معينة والمستقبل يستقبل بسرعة أقل فسوف تضيع بيانات لن يستقبلها المستقبل، وبصفة عامة فإن عدم تطابق السرعة عند المرسل والمستقبل يؤدي إلى أن المستقبل لن يقرأ الداتا بالشكل الذي كما أرسلت له به، والسرعة يطلق عليها ال Baud rate
- السرعة محددة افتراضيا في أي جهاز هي 9600 بت في الثانية الواحدة، وبالطبع يمكننا تغيير هذه القيمة، ولكن لا يتم تحديد أي قيمة عشوائية فهي قيم قياسية محددة يمكننا تحديدها بالطريقة التالية:



فتظهر لنا النافذة الآتية ومنها
يمكن اختيار قيمة السرعة
من بين القيم المحددة كما يلي:

سنتناول الآن توابع الميكرو سي التي تستخدم مع ال Serial Communication

• التابع الأول

سنحتاج تابع لتحديد السرعة وهو كالآتي:

```
Uart1_Init(9600);
```

حيث يتم كتابة السرعة بين أقواس التابع كما هو مبين، ويتم كتابة هذا التابع داخل التابع الرئيسي.

• التابع الثاني

وهو تابع يستخدمه الجهاز لكي يقوم بإرسال البيانات للجهاز الآخر حيث يستخدم لإرسال حرف (أو بايت) فقط كالآتي:

```
Uart1_Write(Data);
```

حيث أن المتغير Data هو متغير لا تزيد قيمته عن 8 بت إذ أن هذا التابع مصمم على هذا الأساس، وسيتبين لاحقاً في هذه المحاضرة كيفية إرسال بيانات حجمها أكبر من 8 بت ...

سنتناول الآن توابع الميكرو سي التي تستخدم مع ال Serial Communication

• التابع الثالث

وهو تابع يستخدم لاستقبال البيانات التي تأتي إليه من الجهاز الآخر، حيث يستخدم لقراءة حرف واحد بالشكل التالي:

```
Uart1_Read ( ) ;
```

وحيث أن هذا التابع يرجع حرف فلا بد من تخصيصه لمتغير وليكن مثلا من النوع الحرفي يتم تخزين فيه ما يرجعه هذا التابع.

وهنا ملحوظة لابد من ذكرها وهي أنه عندما يقوم الجهاز الأول بإرسال بايت للجهاز الثاني فإن هذا البايتم تخزينه في Buffer إلى أن يتم قراءته، وبالتالي لضمان وجود داتا في ال Buffer قبل تابع القراءة لابد من اختبار وجود حرف في ال Buffer أولا قبل عملية القراءة وإن لم يكن به داتا لا نقوم بعملية القراءة من الأساس وهذا يتم من خلال الكود الآتي:

```
if (UART1_Data_Ready ())  
{  
  
    uart_rd = UART1_Read ();  
  
}
```

إذا كانت فيه داتا وصلت ينفذ
ما بداخل ال if

مشروع تطبيقي

- يحتوي المشروع على اثنين من الميكروكنترولر، الأول سيقوم بقراءة قيمة مقاومة (باعتبارها سينسور مثلا) ويقوم بإرسال هذه القيمة إلى الميكرو الثاني، ويقوم الميكرو الثاني بالتحكم في سرعة محرك على أساس القيمة التي استلمها من الميكرو الأول ...



جامعة
المنارة
MANARA UNIVERSITY

مشروع تطبيقي

• الميكرو الأول

- سيقوم بقراءة المقاومة المتغيرة وبالتالي نحتاج مودول ال ADC Interface ثم يقوم بإرسالها إلى الميكرو الثاني وبالتالي سنحتاج إلى ال Serial Interface ومن ذلك سنستخدم تابعي ال initialization الآتين داخل التابع الرئيسي:

```
ADC_Init();  
UART1_Init(9600);
```

مشروع تطبيقي



جامعة
المنارة

• الميكرو الأول

• وعندئذ سيكون البرنامج الذي سينفذه الميكرو الأول المرسل (Transmitter) كالآتي:

```
int x;
char y;
void main()
{
    UART1_Init(9600);
    ADC_init();

    while (1)
    {
        X = Adc_Read(0);
        y = X/4;
        UART1_Write(y);
    }
}
```

لتعريف متغير من النوعية الصحيحة لان ال ADC يقرأ القيمة في 10 بت

لتعريف متغير من النوعية الحرفية حجمه واحد بايت لان دالة الإرسال تقوم بإرسال 1 بايت

هذا الامر يستخدم لقراءة الجهد المتغير الناتج عن تغير قيمة المقاومة

هذا الامر يقسم القيمة التي قرأناها على 4 وسنعرف لماذا في الشرح

هذا الامر يستخدم لإرسال القيمة - بعد القسمة - للميكرو الثاني

- وهو برنامج سهل وبسيط لكن به سؤال يقول: لماذا قمنا بقسمة القيمة على 4 ... والإجابة تتلخص في أن ال ADC يقرأ القيمة في 10 بت وبالتالي أقصى قيمة يمكنه قراءتها هي 1023 ، ولكن التابع uart1_write يرسل واحد بايت فقط أي 8 بت فقط في كل مرة وأقصى قيمة لهذه ال 8 بت بالنظام الثنائي هي 255 ، وبالتالي لا يمكننا قراءة القيمة من ال ADC ثم إرسالها مباشرة، وبالتالي كانت القسمة على 4 حيث أنه لو كانت القيمة المقروءة هي القيمة العظمى 1023 ثم قسمناها على 4 فستكون قيمتها 255 تقريبا ولن تزيد عن ذلك، أي أن القيمة بعد القسمة أصبحت مخزنة في 8 بت وليس 10 كما كانت قبل القسمة وعندئذ يمكننا إرسالها...
- حتى هنا يكون الميكرو الأول قد قام بقراءة قيمة المقاومة المتغيرة وإرسالها والان جاء دور الميكرو الثاني ليقوم بوظيفته

• الميكرو الثاني

• يقوم الميكرو الثاني باستقبال القيمة المرسله له من الميكرو الأول من خلال ال Serial Interface وبالتالي سيقوم بقراءة القيمة من ال Buffer ثم يقوم على أساسها بتغيير سرعة المحرك وهذا ما جعلنا نحتاج إلى التابعين الآتين في بداية البرنامج:

```
UART1_Init(9600);  
PWM1_Init(500);
```

- ويمكن مراجعة كيفية التحكم في سرعة المحرك من المحاضرة 6 حتى يتسنى لك فهم المشروع
- جيداً، فقد كنا نتحكم في سرعة المحرك عن طريق التحكم في عرض النبضة باستخدام التابع:

```
PWM1_Set_Duty();
```

- وهذا التابع يرسل إليه قيمة من 0 إلى 255 كما سبق وشرحنا

مشروع تطبيقي



الميكرو الثاني

- وبالتالي سيكون البرنامج كالاتي:

```
char value;

void main()
{
    UART1_Init(9600);
    PWM1_Init(500);

    while (1)
    {
        if(UART1_Data_Ready())
        {
            value = UART1_Read();

            PWM1_Set_Duty(value);

            PWM1_Start();
        }
    }
}
```

لعمل تهيئة للمودبولات

لقراءة القيمة وتخزينها في المتغير value

نعطي القيمة الى هذه الدالة

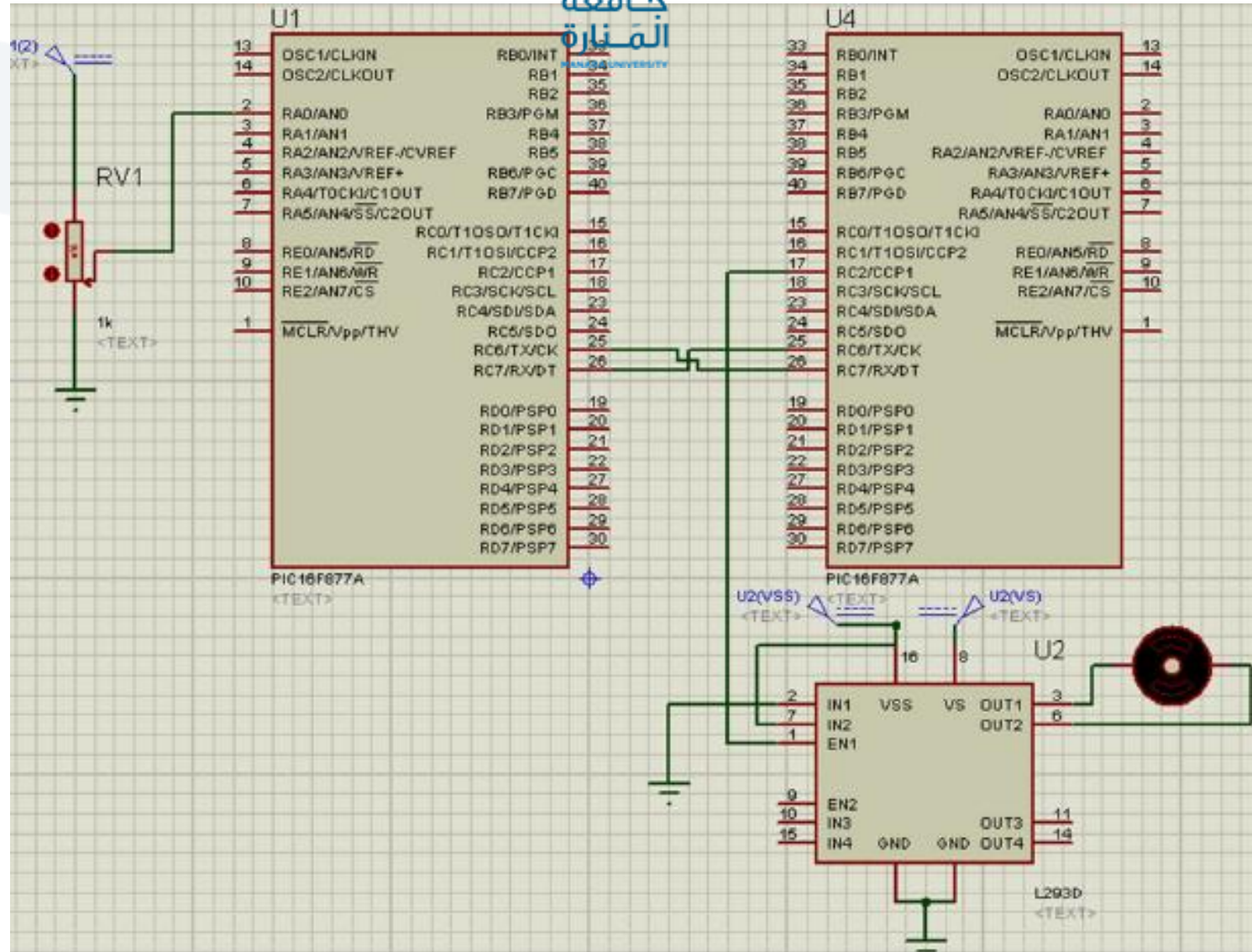
- نلاحظ أن القيمة التي تم استقبالها تتراوح من 0 إلى 255 لأننا قسمناها على 4 عند المرسل، و التابع set_duty يأخذ قيما من 0 إلى 255 ، وبالتالي أخذنا القيمة وأعطيناها مباشرة للتابع ...

مشروع تطبيقي



جامعة المنارة

المحاكاة:





جامعة
المنارة
MANARA UNIVERSITY

مشروع تطبيقي

الهاردوير

