المفتاح البسيط للتعامل مع أنواع مختلفة من الميكروكنترولر

PIC Microcontroller

تأليف . م / أحمد سمير فايد

ahmad_s_f@hotmail.com

بسم الله الرحمن الرحيم

الحلقة الأولى

أحيانا ما يحتاج مستخدمي الميكروكنترولر إلى استخدام نوع آخر غير النوع الذي يستخدموه وذلك للحصول على ميزات إضافية مثل ذاكرة أكبر (RAM أو ROM) أو عدد أطراف أكبر أو خاصية أو مجموعة خواص موجودة في نوع دون آخر (USB – PWM – EEPROM – UART) إلخ..

ويفاجأ مبرمج الميكروكنترولر حينها أن بعض أطراف الميكروكنترولر لا تعمل بشكلها الطبيعي كما كانت في النوع الذي كان يستخدمه . هذه الأطراف غالبا ما تكون PORTA أي الأطراف A0,A1,...,A7 . وأحيانا تشترك معها أطراف أخرى , ولكن ماذا يعني هذا الكلام ؟ سأوضح الأمر بهذه القصة التي حكاها أحد الأشخاص :

قام الأخ فلان بشراء الميكروكنترولر PIC16F628A لأول مرة ولم يتعامل من قبل مع أي ميكروكنترولر آخر سوى PIC16F84A . وكان هدفه من شراء هذا الميكروكنترولر هو أن يستفيد من أهم خواصه (من وجهة نظره) وهي :

1- إمكانية تشغيل الميكروكنترولر دون توصيل مذبذب OSCILLATOR سواءا كان كريستال أو أي نوع آخر حيث يتمتع هذا النوع بإمكانية تشغيل المذبذب الداخلي .
2- استخدام خاصية الـ PWM الداخلية الموجودة داخل الميكروكنترولر بحيث يتحكم في سرعة موتور DC وفي نفس الوقت الذي يخرج فيه الميكروكنترولر هذه النبضات التي تتحكم في السرعة ينفذ الميكروكنترولر شيئا آخر مع هذه الخاصية .
3- أن له ذاكرة أكبر من الـ PIC16F84A سواءا الـ RAM أو الـ ROM .
4- أنه يحتوي على UART داخلي يمكنه من عمل تواصل بالحاسب الآلي باستخدام RS232 وإمكانية عمل مقاطعة عند استلام البيانات .

المهم , عندما قام فلان بشراء الميكروكنترولر تفاجأ من السعر .!! حيث وجد أن هذا الميكروكنترولر ذو الإمكانيات الرائعة مقارنة بـ PIC16F84a ليس بغالي الثمن بل وجده

بنفس السعر لـ PIC16f84a وعندما تجول أكثر في سوق الالكترونيات وذهب لمحل آخر وسأل عن سعر PIC16f628a وجد أنه أرخص شيئا يسيرا من PIC16f84a مع أنه أحدث , وذو إمكانيات أكبر هل تعلم السبب ؟؟ .... سأخبرك به في آخر هذا الكتيب إن شاء اللّه , ولكن دعنا الآن نعود للأخ فلان , فبعد أن اشترى الميكروكنترولر PIC16f628a قال سأقوم ببعض التجارب عليه كالتي كنت أفعلها مع PIC16f84a بحيث إذا استطعت تنفيذ نفس التجارب عليه فلن أشتري PIC16f84a بعد اليوم ..!! حيث يوجد ما هو أرخص وذو إمكانيات أكبر .

وبدأ بعدها فلان بكتابة أول برنامج للـ PIC16F628a حيث كتب البرنامج التالي :

```
void main()
{trisb=0;
while(1)
        { portb=0; delay_ms(500);
          Portb=0xff; delay_ms(500);
        }
}
```

وقام بتشغيل الدائرة في برنامج المحاكاة بروتس ISIS PROTEUS وعملت الدائرة بشكل ممتاز !! حيث أضاءت الليدات الموصلة بـ portb وانطفأت بزمن نصف ثانية بين الإضاءة والإطفاء . حينها قفز فلان من الفرحة !! وفعل مثل بعض الأجانب حيث قال بصوت مرتفع yeees yes !! فقال له صديقه يا أخي اعتز بلغتك وقل (أيواااا ..!!) أو (يا سلاااااام !!) أو (يا جماله ..!!) أو قل (يا حلاااااوة .. !!) فضحكوا جميعا على هذا الكلام ثم قال فلان لصديقه ما رأيك بهذه الليدات فقال له جميلة ولكن ثمانية رقم صغير اجعلهم 12 ليد البركة في الجماعة , فقال له بكل ثقة .. بسيطة ..!! وقال كل ما سنفعله أننا سنوصل أربع ليدات LEDs أخرى بالأطراف A0,A1,A2,A3 انظر سنكتب الكود التالي :

```
void main()
{trisb=0; trisa=0;
while(1)
        { portb=0; porta=0; delay_ms(500);
          Portb=0xff; porta=0xff; delay_ms(500);
        }
}
```

وعندما جاء دور المحاكاة وجد أن portb فقط هو من يعمل والليدات الموصلة بـ porta لا تعمل وحينها وقع في موقف محرج مع صاحبه (وقال بصوت منخفض يا للعار ..!!) , فتسرع قائلا و بدون علم لابد أن المشكلة في برنامج بروتس فقام باستخدام نسخة أخرى منه والنتيجة كانت كما هي , فقام بتجربة الدائرة في الواقع , وحصل على نفس النتيجة . غضب عندها فلان وأصابه بعض اليأس فقال له صاحبه لم لا تبحث عن الموضوع لربما هناك نقطة علمية لا نعرفها , وبعد البحث مرارا وتكرارا وجدوا الحل (والحل مكتوب هنا في هذا الدرس لكي يوفر عليك عناء البحث فالحمد للّه أولا وأخيرا) .

الحل هو أن نقوم بكتابة الكود التالي في بداية البرنامج CMCON=0x07; عندها يعمل porta بشكل طبيعي جدا كما كان في PIC16f84a. إذن الكود الذي جعل فلان ينجح في تجربته هو

```
void main() {  cmcon=0x07; trisb=0; trisa=0;
while(1)  { portb=0; porta=0; delay_ms(500);
          Portb=0xff; porta=0xff; delay_ms(500);
          }
}
```

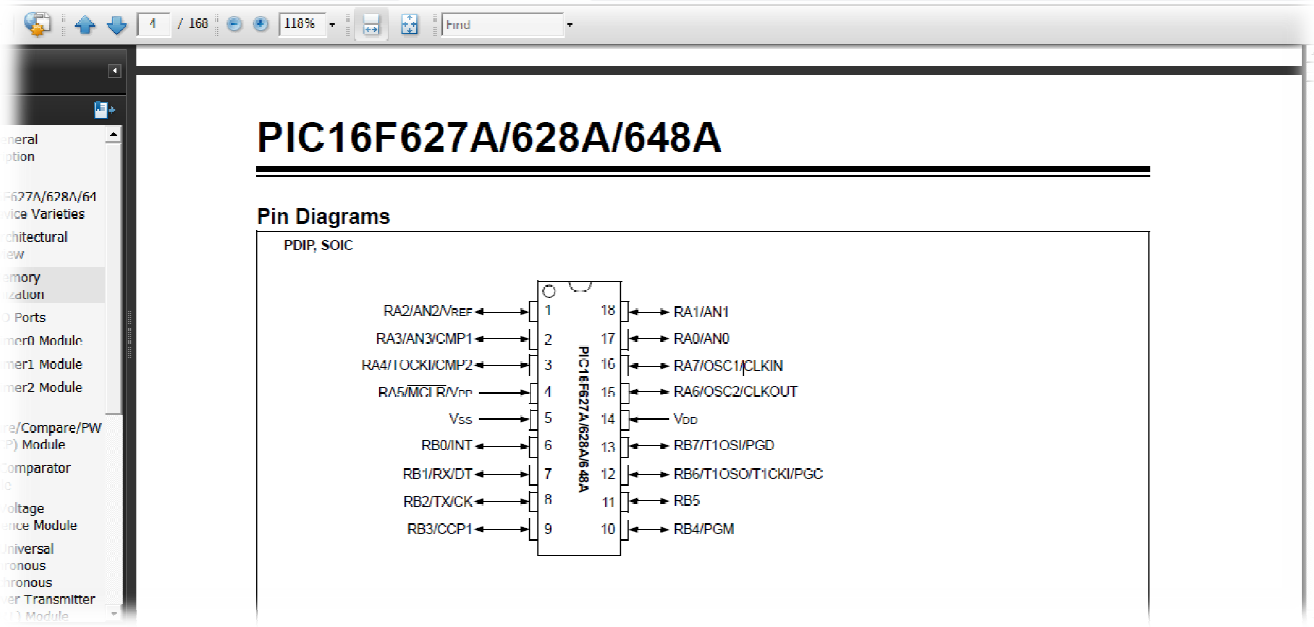
حينها لخص فلان تجربته بأنه إذا كان سيستخدم pic16f628a فإنه لابد أن يكتب هذا السطر في بداية البرنامج ;CMCON=0x07 . هذا إذا كان سيستخدم أطراف porta بشكل عادي أي digital أما إذا لم يستخدم هذه الأطراف فلا داعي لهذا الأمر مع أنه لو كتب هذا الأمر لن يضر .

عزيزي القارئ : كل التهاني لك ليس فقط لأنك ستستطيع استخدام نوع جيد من الميكروكنترولر الذي هو PIC16f628a ذو الإمكانيات الجيدة ولكن لأنك ستوفر بعض المال لأن هذا النوع من الـ PIC أرخص من PIC16f84a , وهذا المال يمكنك استغلاله وتحقيق بعض الأمنيات التي تحلم بها كشراء بعض المقاومات أو الليدات أو السويتشات . (مجرد مزحة إذا لم تعجبك اعتبرني لم أقلها ..!! وأتأسف على ذلك ..)

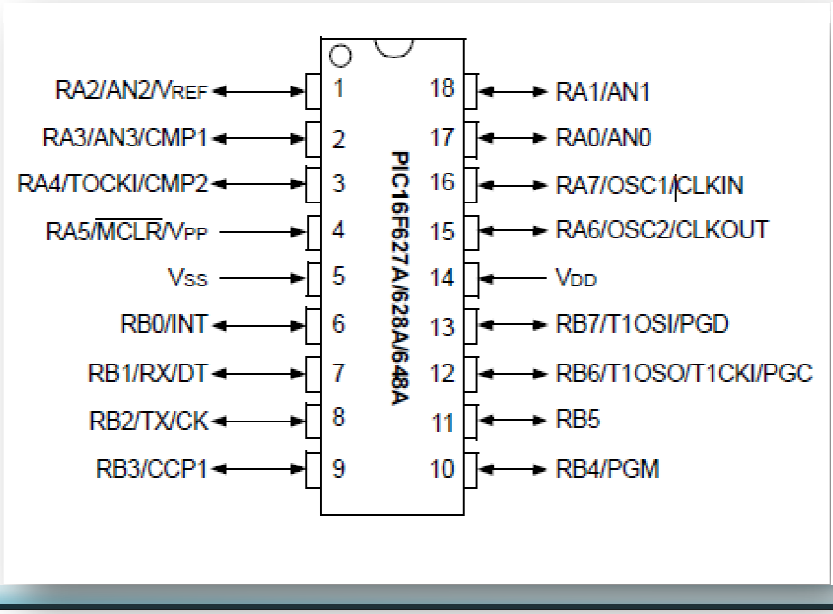
معلومة : قال الدكتور إبراهيم الفقي أن الأبحاث أثبتت أن الضحك يساعد على الفهم أكثر عشر مرات من المعلومة التي نأخذها دون أن نضحك خلال الدرس .

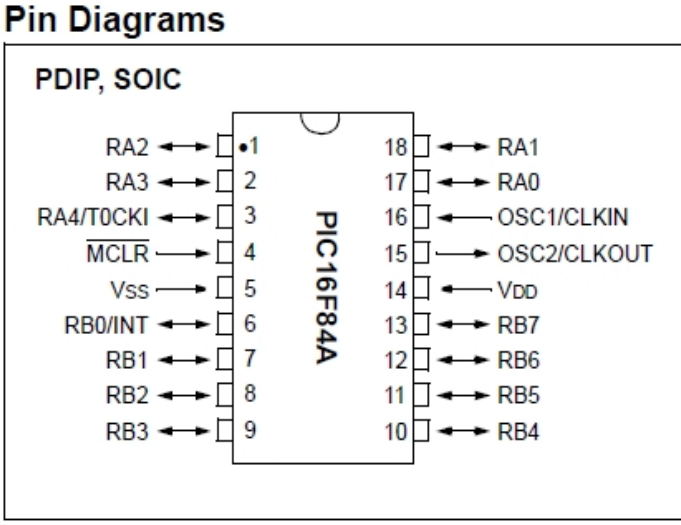
ولأننا نحترم المرأة ونقدرها كل التقدير فبدونها لا تسير حياتنا ولا يكون لها أي طعم , فالمرأة نصف المجتمع وتلد وتربي النصف الآخر , سنقوم بعد قليل بطرح قصة عن فلانة التي اشترت ميكروكنترولر PIC16877a وأجرت بعض التجارب ولكن قبل أن نذكر قصتها نريد أن نشير إلى إمكانيات مهمة موجودة في الميكروكنترولر PIC16f628a ..

افتح الداتاشيت الخاصة بالميكروكنترولر PIC16f628a وقلب الصفحات الأولى إلى أن تجد رسمة الأطراف pin diagram



إذا قمنا بتكبير الرسمة ستبدو لنا كما في الشكل بأسفل الصفحة حيث نجد الميكروكنترولر مكتوب عليه PIC16f627a,PIC16f628a,PIC16f648 أي أن هذه الثلاثة أنواع لها نفس أسماء الأطراف . دقق النظر مرة أخرى ستلاحظ أن الأطراف التي نستطيع توصيل المذبذب بها هي الأطراف رقم 16 و 15 كيف عرفت ؟ لأنه مكتوب بجوارهم OSC1 , OSC2 . والآن قارن بين هذين الطرفين وبين نظائرهما في الـ PIC16f84a ماذا تلاحظ ؟؟





ستلاحظ أنه في PIC16f628a هذين الطرفين مكتوب بجوارهم A6,A7 أيضا . أي أننا نستطيع استخدامهم كأطراف دخل وخرج ولكن كيف أستخدمهم كأطراف دخل وخرج والكريستال موصل بهم ؟؟ لا يا عزيزي إننا سنستخدم المذبذب الداخلي الموجود في هذا الـ PIC وبالتالي سيكون هاذان الطرفان مهيأين لاستخدامهم كدخل أو خرج حسب الرغبة عن طريق TRISA .

أما الميكروكنترولر PIC16f84a فلا يوجد به مذبذب داخلي ولا يوجد A6,A7 حيث أن porta مكون من خمس أطراف فقط A0,A1,A2,A3,A4 .

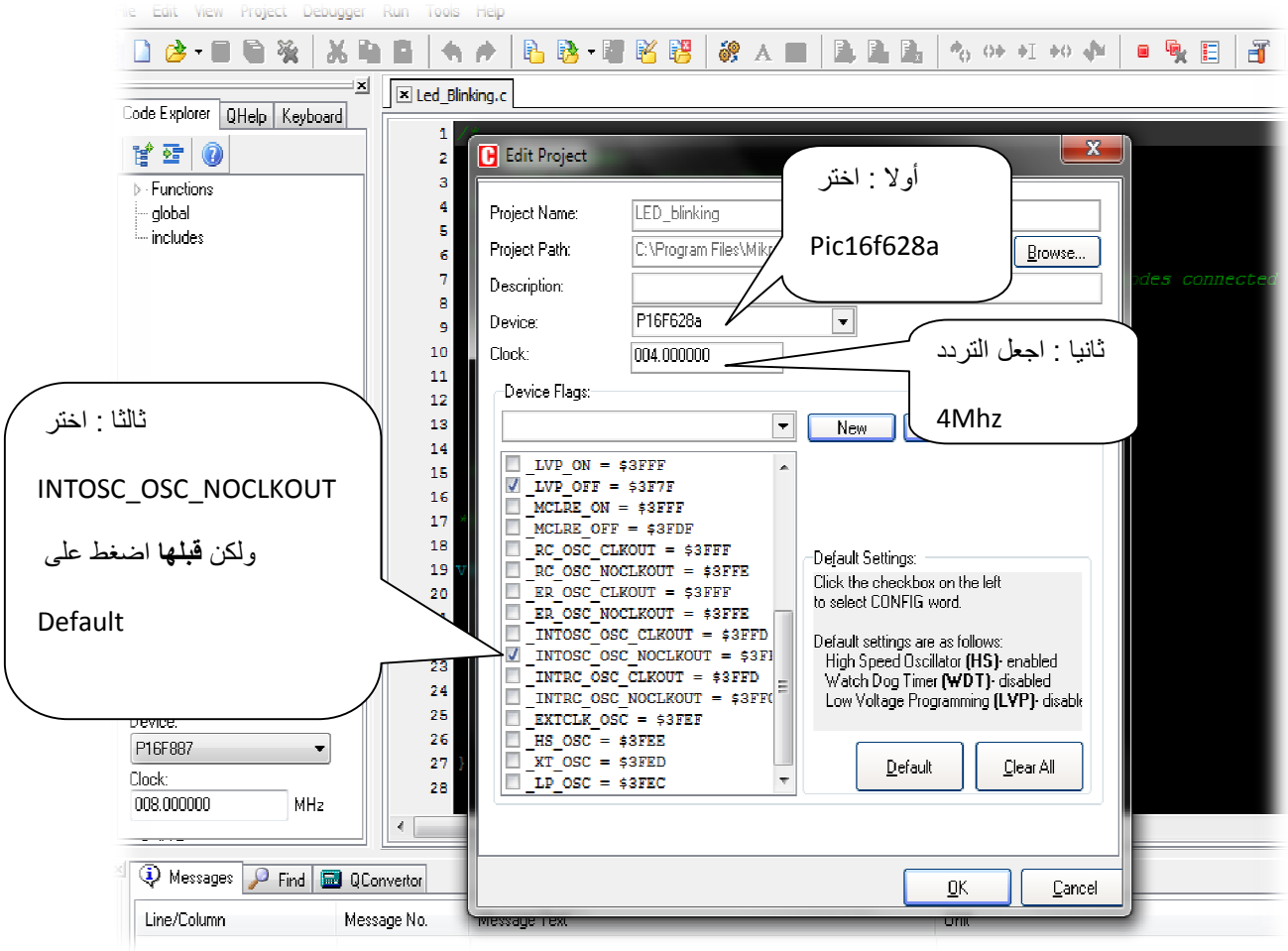
إذن هذه ميزة إضافية في PIC16f628a أنه يحتوي على أطراف دخل وخرج أكثر .

فرصة استثمارية : بما أننا سنستخدم المذبذب الداخلي في دوائرنا إذن لن نشتري كريستال ولن نشتري المكثفات الصغيرة التي توصل به بعد اليوم ..!! إذن فنلنقل جميعا أهلا بالتوفيـــر ..!!

إذا جنيت بعض الثروات من هذا التوفير فلا تنسانا من دعواتك .

كيف سنقوم باستخدام المذبذب الداخلي ؟

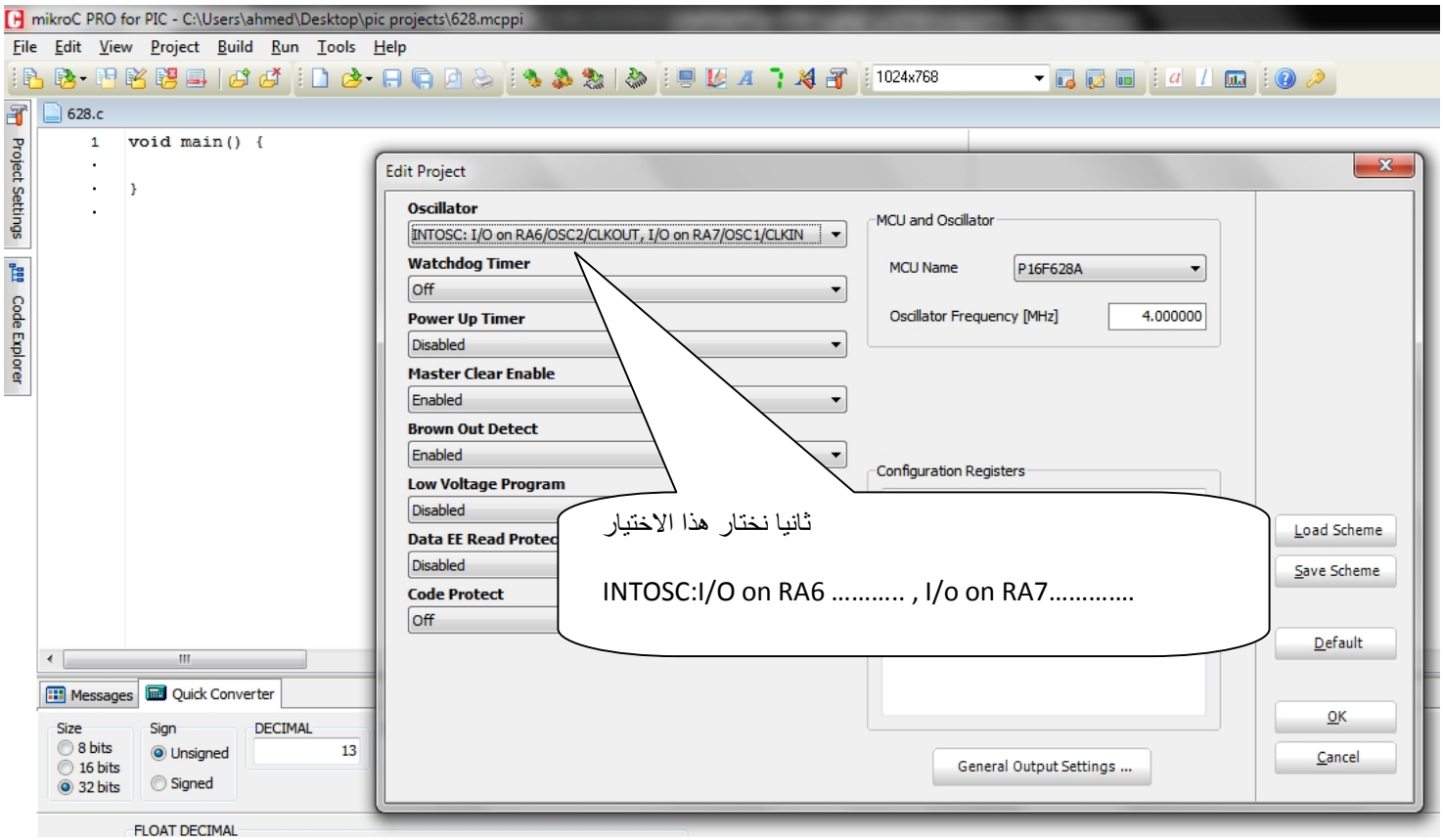
بكل بساطة في بداية عملك للمشروع قم باختيار الميكروكنترولر PIC16f628a ثم بعد ذلك اضغط على Default بعدها قم بحذف علامة الصح الموجودة على HS_OSC وضع علامة صح على INTOSC_OSC_NOCLKOUT كما بالشكل التالي :



إذا لم تضبط هذه الخاصية في بداية المشروع يمكنك ضبطها فيما بعد بالضغط على القائمة Project ثم تختار منها Edit project وستظهر لك نفس النافذة التي بالشكل السابق .

لاحظ أن لديك أربع اختيارات متشابهة كلهم يبدأوا بـ INTOSC_OSC نحن نختار الاختيار الثاني (باقي الاختيارات ربما نشير إليها في وقت لاحق) .

الصورة السابقة توضح كيفية استخدام المذبذب الداخلي باستخدام MikroC 8.2 وبطريقة مشابهة يمكننا تطبيق ذلك في الإصدار الأحدث من لغة MikroC مثل MikroC Pro v3.8 أو ما يضاهيه كما في الشكل التالي :

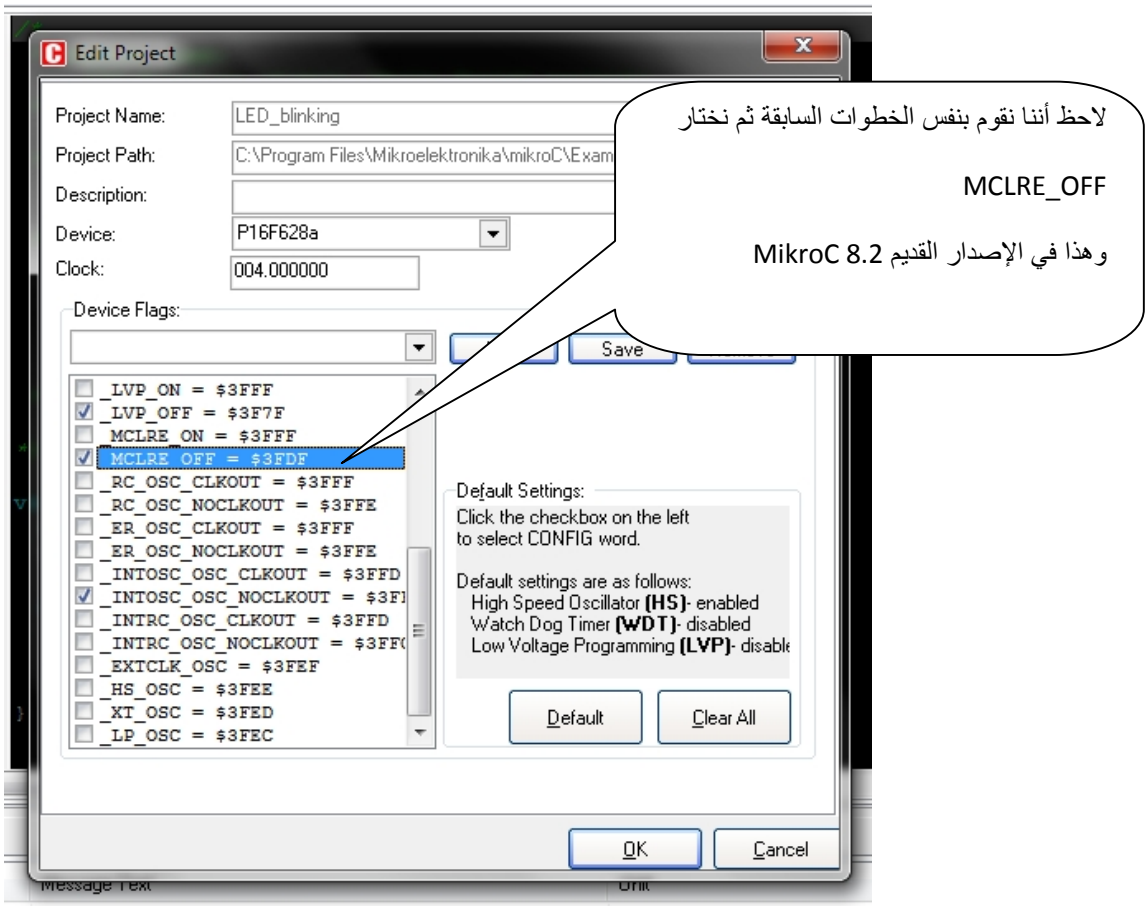


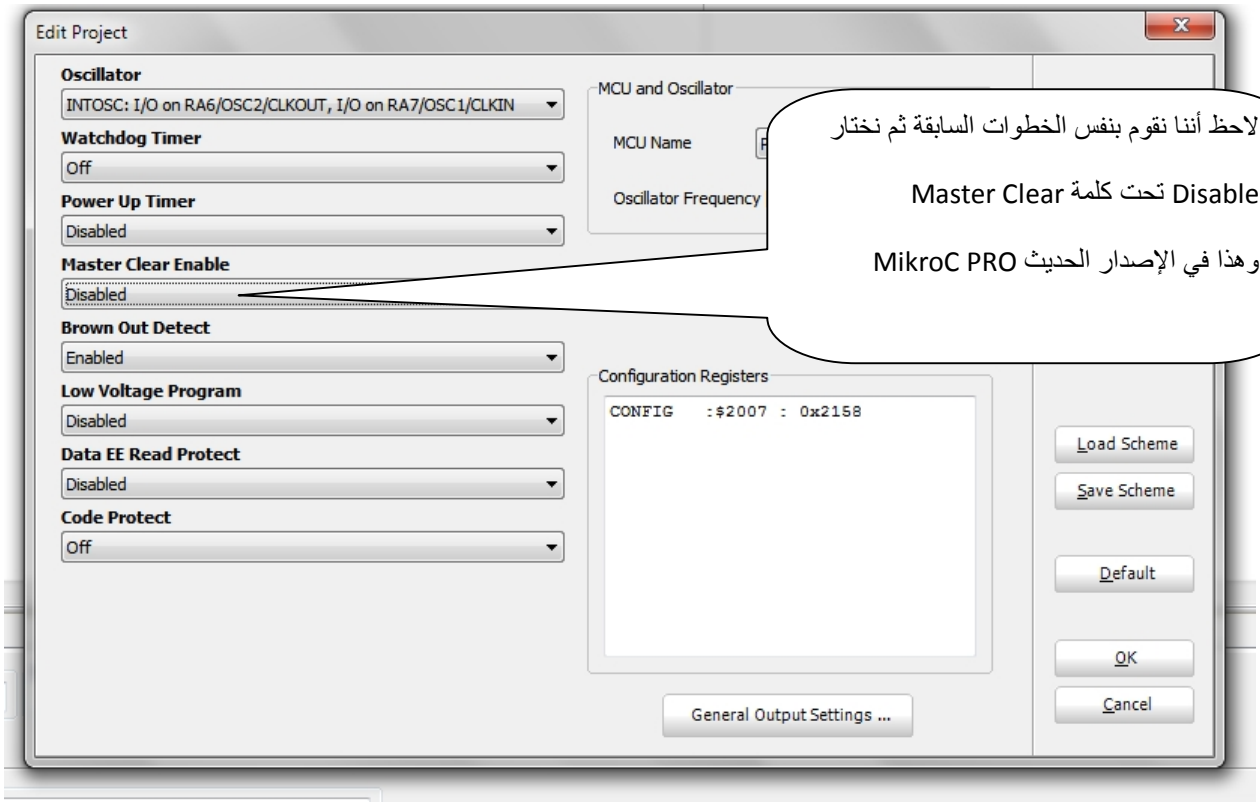
من قائمة project نختار edit project ثم نختار نوع الـ PIC والتردد ثم نضغط بالماوس على Default ثم نختار INTOSC: I/O on RA6/OSC2/CLKOUT , I/O on RA7/OSC2/CLKIN

ميزة أخرى أريد أن أخبرك بها , ولنتعرف على هذه الميزة سأذكرك بما كنا ننصح به دائما عندما كنا نستخدم PIC16f84a كنا نقول لا تنسى أن توصل مقاومة 10k بالطرف MCLR والطرف الآخر من المقاومة موصل بـ VDD (أي موجب خمسة فولت) لكي تعمل دائرتك بشكل سليم . إذا نظرت للطرف MCLR في الداتاشيت للـ PIC16f84a ستجد أنه الطرف رقم 4 ولن تجد أي شيء آخر مكتوب بجواره أما في الداتاشيت الخاصة بـ PIC16f628a فستجد أن هذا الطرف مكتوب بجواره A5 . وهل معنى هذا أنه يمكنني أن ألغي MCLR ولا أوصل المقاومة وأستخدم هذا الطرف كدخل أو خرج ؟ الإجابة , لا .. إنك تستطيع استخدامه كدخل فقط (هكذا صنعوه) .

ولاستخدام هذا الطرف كدخل لابد أن نختار من Edit Project الاختيار التالي :

MCLR_OFF والصورتان التاليتان توضحان عمل ذلك في MikroC 8.2 وفي MikroC PRO


Edit Project
Project Name:
LED_blinking
Project Path:
C:\Program Files\Mikroelektronika\mikroC\Exam
Description:
Device:
P16F628a
Clock:
004.000000
Device Flags:
Save
_LVP_ON = $3FFF
_LVP_OFF = $3F7F
_MCLRE_ON = $3FFF
_MCLRE_OFF = $3FDF
_RC_OSC_CLKOUT = $3FFF
_RC_OSC_NOCLKOUT = $3FFE
_ER_OSC_CLKOUT = $3FFF
_ER_OSC_NOCLKOUT = $3FFE
_INTOSC_OSC_CLKOUT = $3FFD
_INTOSC_OSC_NOCLKOUT = $3F
_INTRC_OSC_CLKOUT = $3FFD
_INTRC_OSC_NOCLKOUT = $3FF
_EXTCLK_OSC = $3FEF
_HS_OSC = $3FEE
_XT_OSC = $3FED
_LP_OSC = $3FEC
Default Settings:
Click the checkbox on the left
to select CONFIG word.
Default settings are as follows:
High Speed Oscillator (HS)- enabled
Watch Dog Timer (WDT)- disabled
Low Voltage Programming (LVP)- disabl
Default
Clear All
OK
Cancel
لاحظ أننا نقوم بنفس الخطوات السابقة ثم نختار
MCLRE_OFF
وهذا في الإصدار القديم MikroC 8.2


Edit Project
Oscillator
INTOSC: I/O on RA6/OSC2/CLKOUT, I/O on RA7/OSC1/CLKIN
Watchdog Timer
Off
Power Up Timer
Disabled
Master Clear Enable
Disabled
Brown Out Detect
Enabled
Low Voltage Program
Disabled
Data EE Read Protect
Disabled
Code Protect
Off
MCU and Oscillator
MCU Name
Oscillator Frequency
Configuration Registers
CONFIG :$2007 : 0x2158
General Output Settings ...
Load Scheme
Save Scheme
Default
OK
Cancel
لاحظ أننا نقوم بنفس الخطوات السابقة ثم نختار
Disable تحت كلمة Master Clear
وهذا في الإصدار الحديث MikroC PRO

واجب خفيف وبه بعض الذكاء والمهارة (1) : قم بعمل مشروع باستخدام pic16f628a يحتوي على أكبر عدد تستطيع تنفيذه من الليدات وسويتش واحد كلما تم الضغط عليه يضيء ليد مختلف من هذه الليدات .

(لا تستخدم أي دوائر متكاملة) .... إذا استطعت أن تجعلهم 14 ليد فأعلم أنك فهمت الدرس جيدا وإذا استطعت أن تجعلهم 15 ليد أو أكثر ودون استخدام أي دوائر متكاملة فأنت إما لديك معلومة جيدة أو فكر متميز .. !! صاحب أكبر عدد من الليدات هو الأكثر تميزا .)

والآن جاء دور قصة فلانة وحكايتها مع الـ PIC16f877a .

ولكن لنجعلها في الحلقة القادمة إن شاء اللّه .. من مسلسل متعة التعلم الذي يأتيكم طيلة شهر رمضان المبارك قناة أسرع طريق لاحتراف برمجة الميكروكنترولر.

علما أننا لم نصل بعد إلى المفتاح البسيط الذي أقصده , وسنتعرف عليه إن شاء اللّه ونتعلمه جيدا كي نستخدم أي نوع من أنواع الميكروكنترولر حتى ولو لم نذكره هنا في هذا الكتيب الصغير .

وكل عام وأنتم بخير ,, وكل عام وأنتم إلى اللّه أقرب ومن الذنوب أبعد .

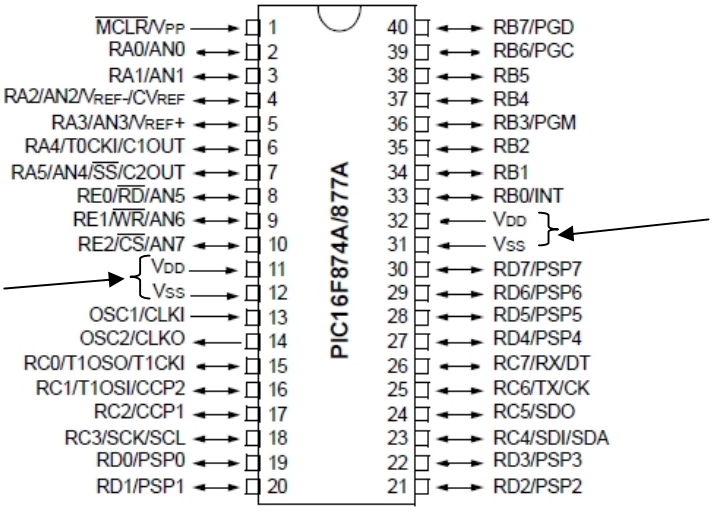
كل عام وأمتنا أنهض وأنهض .

دعواتكم ,,

الحلقة الثالثة :

اليوم نحكي لكم قصة فلانة , وتبدأ حكايتها باشتراكها في مسابقة في الالكترونيات تقدمها كلية الهندسة التي تدرس بها , وحينما قررت البدء استشارت دكتور لها في الجامعة والذي كان يدرسها مادة embedded system والتي كانوا يدرسوا فيها المايكروكنترولر , فقالت له يا دكتور أريد أن أقوم بعمل مشروع يفعل كذا وكذا وقالت له إنني أستخدم الـ pic16f84a فنصحها باستخدام pic16f877a كي تستطيع توصيل حساس الحرارة وحساس الضغط الذي في مشروعها حيث أن هذا الـ PIC يمتاز بخاصية قياس الفولت عن طريق ما يسمى بـ ANALOG TO DIGITAL CONVERTER وبالتالي يمكن قياس خرج هذه الحساسات , بالإضافة لحجم الذاكرة الأكبر وعدد الأطراف الضخم حيث أنه يحتوي على 33 طرف يمكن استخدامهم كدخل أو خرج (33 I/O pins).

فذهبت على الفور واشترت القطعة pic16f877a وعندما فتحت الداتا شيت لتعلم كيف ستوصل الأطراف فوجئت بشيء غريب حيرها بعض الشيء حيث وجدت أن هناك طرفين VDD وطرفين VSS كما في الشكل التالي :



احتارت حينها فلانة هل توصل الطرفين 32 و 31 أم توصل الطرفين 11 و 12 أم توصلهم جميعا , أم توصل طرف VDD الذي على اليمين وطرف VSS الذي على اليسار أم العكس , فقالت لنجرب , وكتبت برنامج يضيء الليدات الموصلة بـ portb كالذي كتبه فلان في الحلقة

الأولى ثم قامت بعملية البرمجة (الكتابة على الـ PIC أو الحرق) وقالت سأوصل الطرفين 31و32 فقط , فعملت الدائرة بشكل سليم , فرحت حينها وقالت لنجرب الآن ولنفصل الطرفين 31 و 32 ونجعل التغذية من 11 , 12 فقط فعملت الدائرة أيضا دون تغيير ملحوظ .

خطر في بالها فكرة بأن تقوم بتوصيل الطرف VDD الذي على اليمين (الطرف 32) وتوصل أيضا الطرف VSS الذي على اليسار (الطرف12) فلاحظت شيئا .....

أن الدائرة أيضا اشتغلت بشكل سليم .

حاولت فلانة بعدها التفكير هل يمكن أن تستفيد من هذه الخاصية في شيء ؟ ... وبعد تفكير قليل عرفت أن هذه الخاصية تسهل عليها تصميم الدوائر المطبوعة PCB حيث يمكن أن نوصل جهدي التغذية من اليمين أو اليسار على حسب الأسهل .

ثم قامت باستخدام الطرف VSS الذي على اليمين(31) وتوصيل الطرف VDD الذي على اليسار (11) ولكن يبدو أنها لم تحسن التوصيل , وهنا حدثت المفاجأة .. حيث لم تعمل الدائرة , بعد ثواني بسيطة وضعت يدها على الـ PIC فوجدت أن حرارته قد ارتفعت بشكل كبير ..!! فصلت فلانة حينها مصدر التغذية الكهربية عن الدائرة , وملأ الخوف قلبها وقالت آه ... آه .. ميزانية السنة ضاعت ... راحت الفلوس ... !! .. خرب الـ PIC ..!! , حيث أنها ظنت أن الـ PIC قد خرب وهو ثمنه مرتفع جدا (من 25 – 30 جنيه مصري تقريبا = من 5-6 دولار أمريكي) .

أسرعت فلانة إلى التليفون لتتصل بصديقتها " وفاء " وحكت لها القصة وهي حزينة فقالت لها وفاء : أنا ليس لي أي خبرة في الميكروكنترولر .. لا أستطيع أن أفيدك للأسف ..!!

فلانة : إذن وما الحل ..؟؟

وفاء : هل تذكرين صديقتنا في الكلية التي قامت بالحصول على جائزة أفضل مشروع السنة الماضية .

فلانة : هل تقصدين " آية" ؟ أم واحدة غيرها ؟

وفاء : نعم هي تلك البنت المؤدبة المتميزة , لقد سمعت أنها تساعد الجميع دائما بكل حب ودون أي تكبر مثل بعض الناس .. وتعطي المعلومة بسخاء دون كتمانها .. هل تريدين رقم هاتفها ؟

فلانة : ياريـــــت ..

بعدها قامت بفلانة بالاتصال بزميلتها آية وحكت لها القصة وأخبرتها أنها فور أن وجدت حرارة الـ PIC ارتفعت قامت على الفور بفصل الكهرباء عن الدائرة , فقالت لها آية .. لا تقلقي , هذا الخطأ يحدث غالبا إذا قمت بتوصيل جهد التغذية الموجب +5 فولت بالـ VSS بدلا من VDD وتوصيل جهد التغذية الأرضي بـ VDD بدلا من VSS أي أنك باختصار قمتي بعكس الأطراف وبالنسبة للـ PIC احتمال كبير أنه لم يتلف لا تخافي .. حاولي عكس الأطراف وجربي وأخبريني بالنتائج , على الفور نظرت فلانة إلى دائرتها ولاحظت أنها قد عكست التوصيلات , فقامت بتصحيحها فعملت الدائرة بشكل سليم ..

ثم أسرعت إلى الهاتف وقلبها يطير فرحا بأنها لم تخسر 30 جنيه , وأخبرت زميلتها آية بما حدث وقالت لها الحمد لله ..

ردت آية وقالت : إذن هاتي 10 جنيه بس ثمن النصيحة ... (أمزح أمزح لا تصدقي !!)

فلانة : لا والله أنت تستحقين أكثر من ذلك بكتير .

آية : إذن هاتي 50 جنيه ..

ضحكوا جميعا على هذه المزحة و قبل انهاء المكالمة قالت لها آية إذا كنت ستستخدمين PIC16f877a فيفضل أن تكتبي الأمر adcon1=0x07; أو adcon1=0x06; في بداية أي مشروع إلا إذا كان هذا المشروع ستستخدمين فيه حساسات من النوع analog حينها تغيريه على حسب الطرف الذي ستوصلين به الحساس , حيث أن هذا الأمر يجعلك تستخدمين porta,porte بشكل عادي جداً I/O كما كنت تستخدمينه مع PIC16f84a .

عزيزي القارئ : كل التهاني لقد أصبحت الآن لديك الإمكانية للتعامل مع نوع آخر من الـ PIC وهو Pic16f877a ذو الأطراف الكثيرة , كل ما عليك أن تفعله أن تكتب في بداية أي برنامج مايلي :

```
void main() {  adcon1=0x07;
.....
.....
}
```

مرت الأيام والأيام , وفي يوم ما أثناء تصفح فلانة للانترنت وقع تحت يدها كتيب بعنوان " المفتاح البسيط للتعامل مع أنواع مختلفة من الميكروكنترولر " والذي كان جزء من المسلسل العلمي الرمضاني "متعة الحياة " الذي قدمه المهندس / أحمد سمير فايد , في إحدى المنتديات العربية . حيث قرأت الحلقة الأولى والتي كانت تحكي عن قصة فلان مع الـ Pic16f628a .

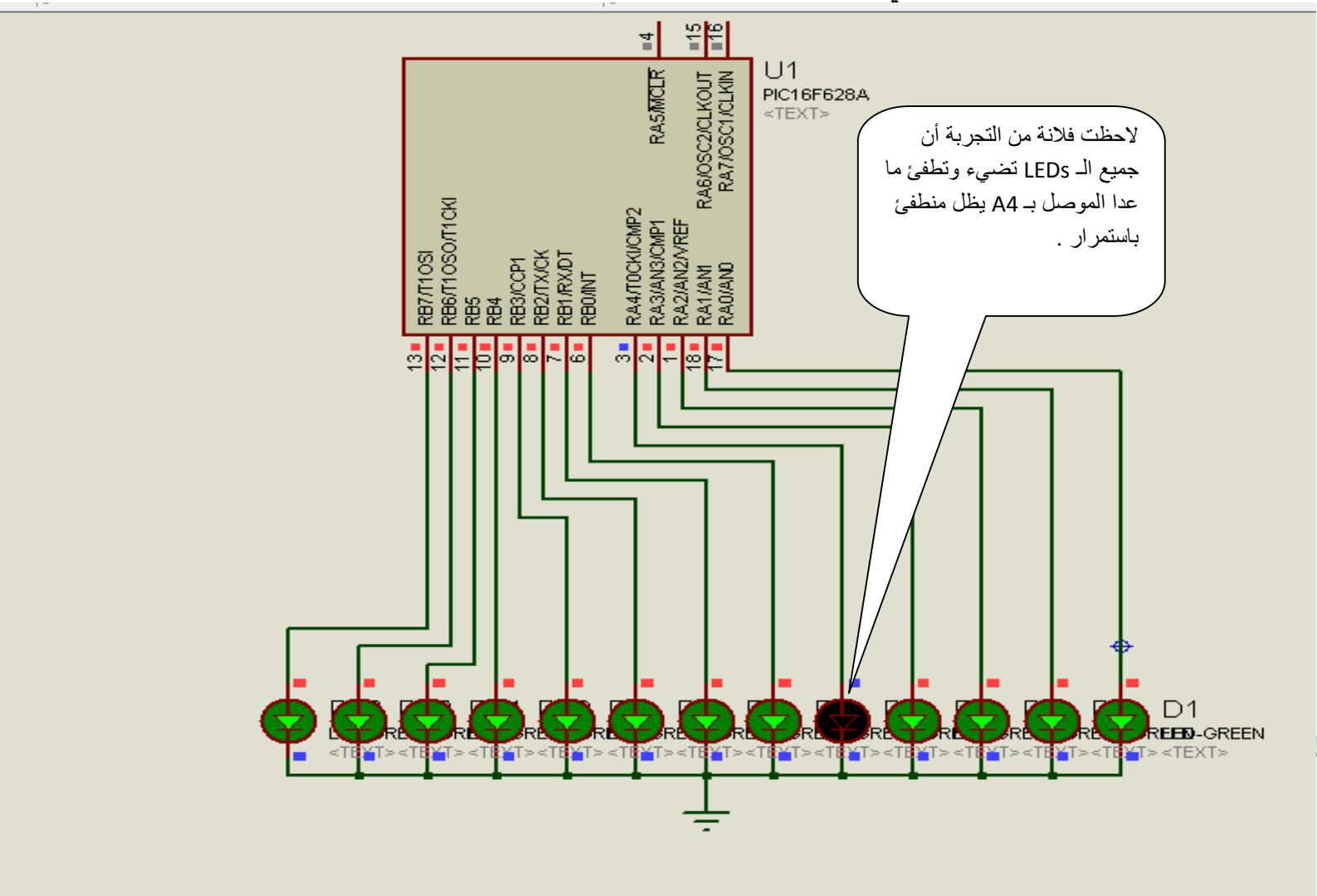
كانت فلانة متميزة فكريا , فلم تمر قصة فلان عليها مرور الكرام كما مرت على الكثيرين , فلم تأخذ المعلومة النهائية التي وصل إليها فلان بأن نكتب cmcon=0x07; في بداية البرنامج مع الـ PIC16f628a وفقط. بل قامت بتكرار نفس التجارب التي قام بها فلان , لترى هل كل ما حكي عن فلان صحيح أم لا ؟ . فهي تؤمن أن المعلومة العلمية التطبيقية نتأكد منها بالتطبيق والتجربة .

المهم ... قامت فلانة بتوصيل بعض الـ LEDs بالأطراف من A0 إلى A4 . كما قامت بتوصيل ثمانية LEDs بالأطراف من B0 إلى B7 .

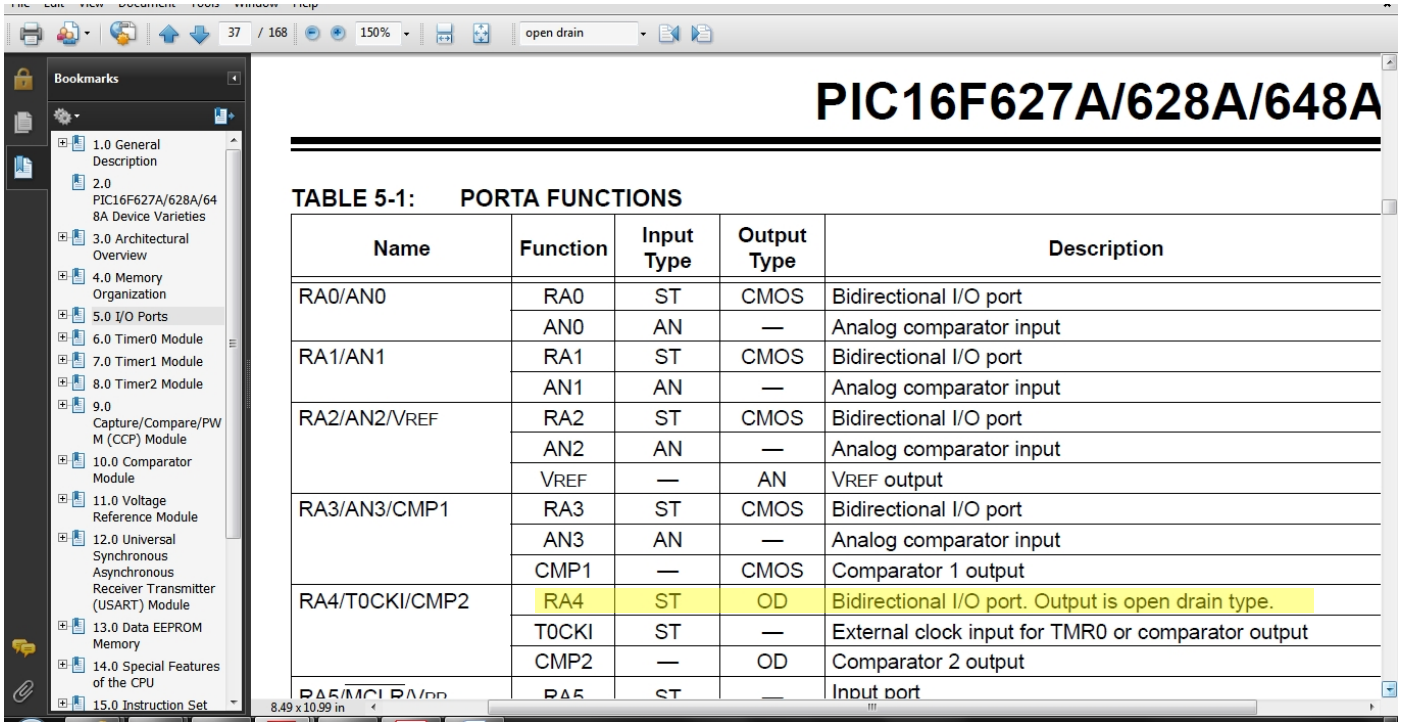
وكتبت الكود التالي :

```
void main() {
trisa=0;  trisb=0;
while(1)
 { porta=0; portb=0;
   Delay_ms(100);
   Porta=0xff; Portb=0xff;
   Delay_ms(100);
 }
}
```

ظنت فلانة أنها عندما تجرب هذه الدائرة أن الـ LEDs الموصلة بالأطراف A0-A4 لن تعمل على حسب قصة فلان . ولكنها فوجئت أن الـ LEDs الموصلة بـ بالأطراف من A0-A3 عملت بشكل سليم و الـ LED الوحيد الذي لم يعمل هو الموصل بـ A4 .



حاولت فلانة عدة محاولات لكن دون جدوى جربت أن تكتب الأمر CMCON=0x07; فلم يجدي نفعا . فتحت الداتا شيت datasheet وأخذت تقرأ وتقرأ فوجدت أن كل الأطراف عندما يذكر وظيفتها يكتب أن من ضمن هذه الوظائف أنها يمكن استخدامها كدخل وخرج Bidirectional I/O . ما عدا الطرف A4 مكتوب بجواره نفس الجملة + open drain فلم تفهم ما المقصود بها

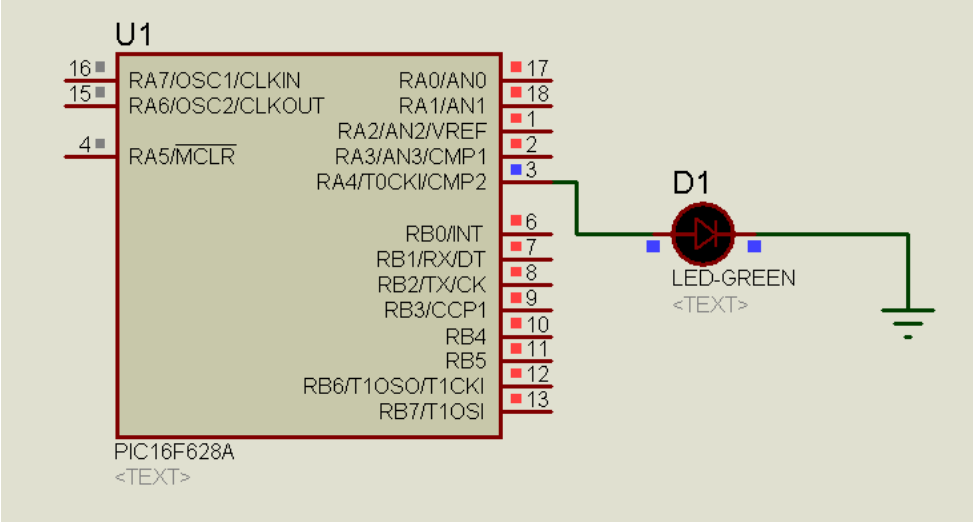


PIC16F627A/628A/648A

TABLE 5-1: PORTA FUNCTIONS

Name	Function	Input Type	Output Type	Description
RA0/AN0	RA0	ST	CMOS	Bidirectional I/O port
	AN0	AN	—	Analog comparator input
RA1/AN1	RA1	ST	CMOS	Bidirectional I/O port
	AN1	AN	—	Analog comparator input
RA2/AN2/VREF	RA2	ST	CMOS	Bidirectional I/O port
	AN2	AN	—	Analog comparator input
	VREF	—	AN	VREF output
RA3/AN3/CMP1	RA3	ST	CMOS	Bidirectional I/O port
	AN3	AN	—	Analog comparator input
	CMP1	—	CMOS	Comparator 1 output
RA4/T0CKI/CMP2	RA4	ST	OD	Bidirectional I/O port. Output is open drain type.
	T0CKI	ST	—	External clock input for TMR0 or comparator output
	CMP2	—	OD	Comparator 2 output
RA5/MCLR/VPP	RA5	ST	—	Input port

قررت أن تقوم ببعض التجارب على الطرف A4 دون الأطراف الأخرى , فقامت بتوصيله كما بالشكل :



وقامت بكتابة الكود التالي :

```
void main() {

trisa.f4=0;

porta.f4=1;

}
```

هذا الكود من المفترض أن يجعل الـ LED الموصل بـ A4 يعمل . ولكن عند التجربة لم يعمل الـ LED . وفشلت المحاولة .

ولكن لم يؤثر هذا الفشل على همتها لأنها تؤمن بمقولة تقول

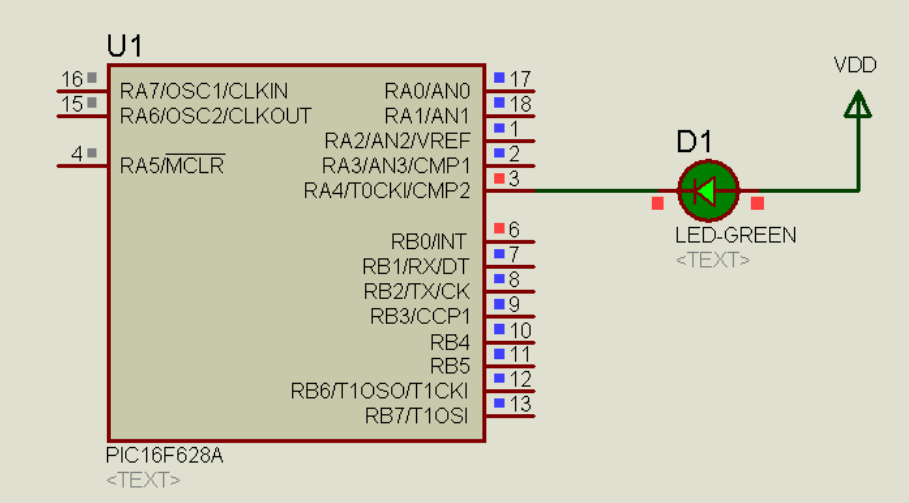
The difference between successful and unsuccessful is that the successful have failed many times more than non-successful

" الفرق بين الناجحين وغير الناجحين أن الناجحين فشلوا عدد من المرات أكثر من غير الناجحين"

قامت بقياس فرق الجهد بين هذا الطرف والأرضي (والطبيعي أن يكون بخمسة فولت لأن A4=1) ولكنه لم يكن كذلك .

فخطر ببالها فكرة جديدة بأن تغير طريقة توصيل الـ LED حيث تقوم بتوصيله بالشكل التالي

وبهذه الطريقة من المفترض أن يعمل الـ LED إذا كان A4 يساوى صفر ولا يعمل إذا كان غير الصفر .



حدثت المفاجأة وعمل الـ LED بشكل سليم . قامت بقياس فرق الجهد بين هذا الطرف وبين الجهد الموجب فوجدت أنه خمسة فولت وهذا يعني أن هذا الطرف جهده أصبح صفر .

إذن الطرف A4 يمكن أن يكون صفر ولا يمكن أن يكون واحد في التوصيلات العادية .

وعلى الفور , قامت بنفس التجربة في الواقع باستخدام LED شفاف 5mm يضيء لون أحمر هكذا تقول عليه (وبعض الناس يقولون عليه ليد ليزر) المهم هذا النوع من الـ LEDs يحتاج إلى تيار أكبر من الـ LEDs العادية حيث أنه يستهلك 20 ملي أمبير أو أكثر .

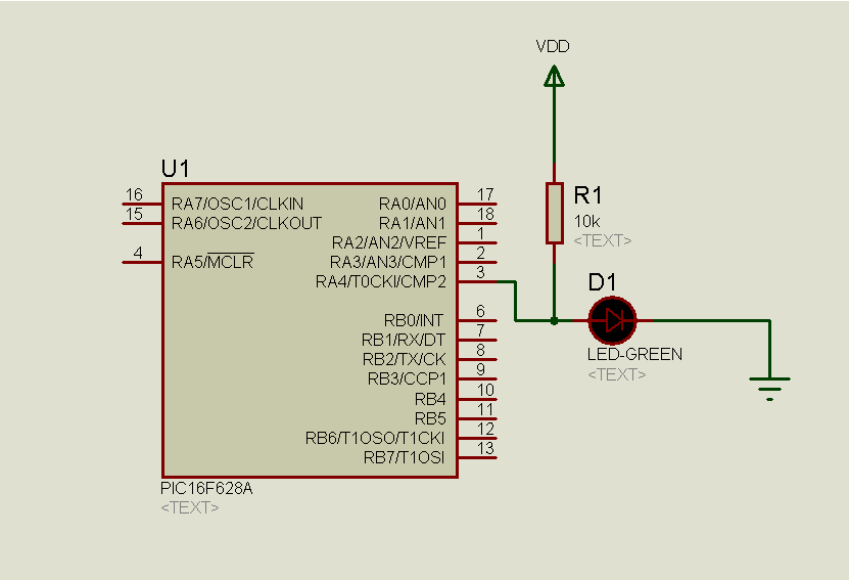
نجحت التجربة , فقامت بعدها بتوصيل هذا الليد بدون مقاومات لاحظت أنه يعمل أيضا , قررت استبدال هذا الـ Led بآخر أصغر 3mm يستهلك تيار أقل ونجحت التجربة بدون مقاومات وباستخدام مقاومات .

قامت فلانة بإخبار إحدى صديقاتها , عن الموضوع وعن هذه النتيجة الطيبة التي وصلت إليها , فقالت لها صديقتها أنها عندما كانت تتصفح إحدى المنتديات العربية وجدت المهندس المغربي الرائع / زكي شاكر . وفقه اللّه يتكلم عن هذه النقطة ويقول أن هذا الطرف من النوع open drain , فردت فلانة نعم قرأت في الداتاشيت أن هذا الطرف Open drain ولكن لم أفهم ما هو المقصود بذلك ؟

قالت صديقتها : أن المهندس زكي يقول أنه يجب توصيل هذا الطرف A4 بمقاومة متصلة بجهد التغذية vdd حينها يستطيع هذا الطرف أن يكون واحد أو أن يكون صفر . وهذه الخاصية توجد في أنواع مختلفة من الدوائر المتكاملة حيث يجب توصيل هذه المقاومة في حالة أن طرف الدائرة المتكاملة open drain . كما تكلم عن هذه النقطة أحد المتابعين للمسلسل الرمضاني المهندس الاسكنداني المبدع والرائع والصاعد إلى التميز غير العادي king5star (كريم)

قاطعتها فلانة , قائلة .. نعم .. نعم .. عرفت الآن لماذا قال المهندس أحمد سمير فايد في كتبه أنه يفضل جعل A4 دخل وليس خرج , لكي لا نقع في هذا الموقف ... سأذهب وأجرب ما تقولين .

قامت بعدها فلانة بتوصيل الدائرة كما قالت لها صديقتها , كما بالشكل التالي :



وجربتها في برنامج المحاكاة فلم ترى الـ LED يضيء , فقامت بتجربتها في الواقع فوجدت أن الـ LED يضيء إضاءة خافتة , قامت بتقليل المقاومة من 10 كيلو إلى واحد كيلو فوجدت أن الإضاءة ممتازة والـLED يعمل , فقامت بتقليل المقاومة أيضا في برنامج المحاكاة فعمل الـ LED .

وبهذا انتهت مشكلة الطرف A4 للأبد .. إما بعكس الليد وتشغيله إذا كان الطرف يساوي صفر أو توصيل مقاومة واحد كيلو مثلا بجهد التغذية كما بالشكل السابق .

ولكن فلانة الطموحة لا يزال هناك غموض عندها في نقطة هامة وهي ما فائدة كتابتنا للأمر CMCON=0x07; ؟؟

نتابع باقي قصتها في حلقات قادمة إن شاء اللّه ...

==========

الواجب العملي ... محاولة الحصول على الفائدة الحقيقية لهذا الأمر .. أم أنه فقط ننكتبه لأننا لا نستخدم خاصية المقارن الموجودة في هذا الميكرو أي لعدم تشغيل شيء لا نستخدمه ؟ ؟؟

الواجب الثاني : محاولة عمل أشكال متحركة بالـ LEDs باستخدام الطريقة التي شرحناها في الحلقة الثانية carlieplexing . دائرة متحركة مثلا .. اظهار حرف أو حرفين مثلا .. أي فكرة ؟ واجث في الانترنت عن هذه الكلمة لترى الكثير من الإبداعات ..!! .... أتمنى أن لا تنسونا من دعواتكم , وأن تكثر الهمة في الحل حتى نسرع أكثر في الحلقات . ولتشغيل أكثر من LED في نفس الوقت سنشغل ليد ثم نطفيه ثم نشغل الآخر ثم نطفيه وهكذا إلى أن ننتهي من جميع الـ LEDs المراد تشغيلها وكل ذلك يتم بسرعة رهيبة بالميكرو ثانية . مثلا ..

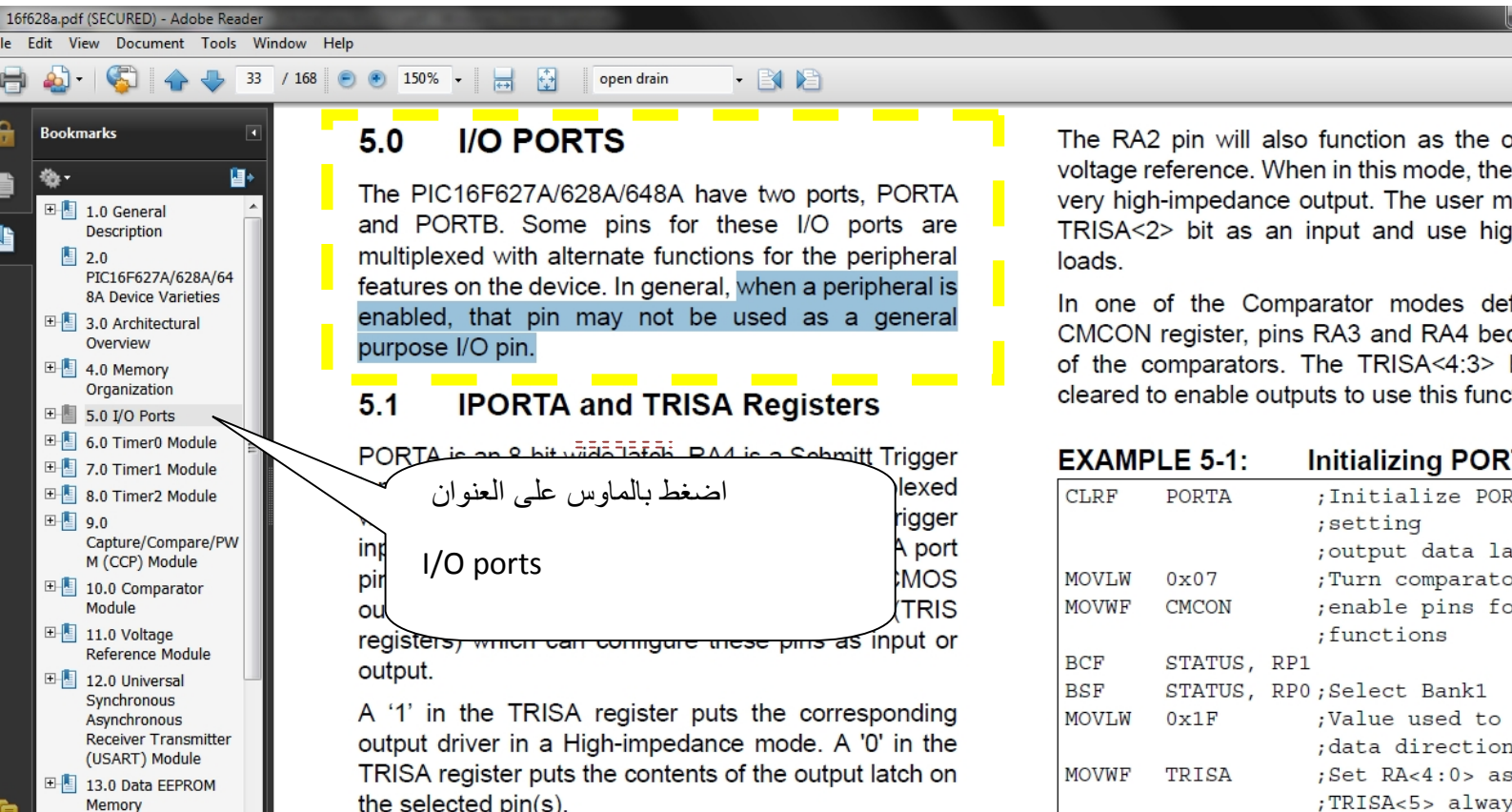
الحلقة الرابعة :

نكمل اليوم قصة فلانة , في الحلقة الماضية قامت هي ببعض التجارب ولم تجد فرقا في كتابتها للأمر adcon1=0x07 مع الـ pic16f877a أو كتابة الأمر cmcon=0x07 مع الـ pic16f628a حيث أننا استطعنا أن نضيء ونطفئ الليدات بدون كتابة هذه الأوامر واكتشفنا أن قصة فلان لم تكن سليمة مئة بالمئة ....

العجيب أن فلان سمع عن قصة فلانة !! فقام بنفس التجارب التي قامت هي بها مع pic16f628a ومع pic16f877a ووجد أن قصتها سليمة مئة بالمئة فأصبح في حيرة من أمره , فقرر أن يتصل بصديقه الخبير في علم الميكروكنترولر و CNC المهندس / محمد المليجي .

فلان : ما الفائدة الحقيقية من كتابتنا للأمر adcon1=0x07 وهل أكتبه أم لا ؟ حيث أنني لم أرى أي فرق في كتابته ؟ فوظيفة البرنامج لم تتغير

محمد المليجي : إذا قمت بقراءة الداتاشيت ستجد أنه مكتوب أن بعض الأطراف كأطراف porta لها خواص وإمكانيات إضافية (يسمونها بالانجليزية peripheral features) هذه الخواص والمميزات إذا كانت مفعلة enable (وهي تكون تلقائيا مفعلة في بعض أنواع الميكرو) ربما لا تستخدم هذه الأطراف كدخل أو خرج.. قم بفتح الداتاشيت ثم العنوان I/O ports وستجد هذه المعلومة :



فلان : ولكن من خلال التجارب التي قمت بها لم أجد أي فرق ؟

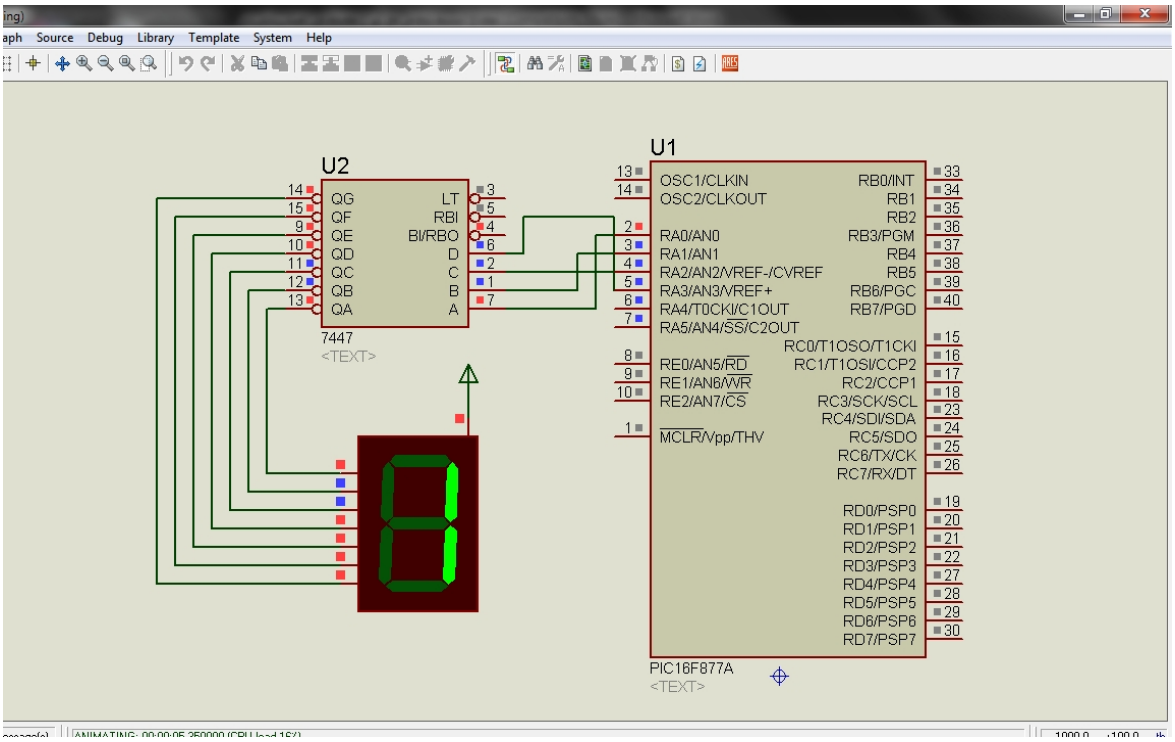
محمد المليجي : أريد أن أقول لك أنه ليس في كل الأحيان ستعمل معك هذه الأطراف بشكل سليم ربما في التجارب التي قمت بها أنت صادفت أن الأمر adcon1=0x07 لم يؤثر . ولكن هناك بعض الأمثلة التي ستجعل هذه الأطراف لا تعمل بشكل سليم كما كنت تستخدمها مع Pic16f84a .

فلان : هل لك أن تضرب لي مثال ؟ فأريد أن أرى فائدة حقيقية للأمر adcon=0x07; وإلا سأرمي كتاب "المفتاح البسيط لتعلم أنواع مختلفة من الميكروكنترولر" بل ربما أحرقه , حيث أنه لن تصبح له أي فائدة؟

محمد المليجي : يا أخي لا تقل هذا الكلام إنه كتاب ممتاز وممتع حيث يوصلك لطريقة التعامل مع أنواع مختلفة من الميكروكنترولر بدون أي مشاكل ويعطيك لمحات سريعة عن فوائد بعض الأنواع عن الأخرى . اسمع اخي .. سأخبرك بمثال يجعل أطراف porta لا تعمل بشكل سليم إلا إذا أضفت الأمر adcon1=0x07; أو adcon1=0x06; . اكتب الكود التالي :

```
void main() {
trisa=0;
for(;;)
   {
       Porta++;
       Delay_ms(1000);
      if(porta==10)porta=0;
    }
}
```

وقم بتوصيل أربعة أطراف من porta بـ (BCD to 7segment decoder) مثل الدائرة المتكاملة 7447 . وقم بتوصيلها مع سفن سيجمنت 7segment .



عند تشغيل الدائرة ستلاحظ أن الرقم الظاهر على السفن سيجمنت هو واحد وهو ثابت لا يتغير , ومن المفترض أن يظهر الرقم واحد ثم اثنين ثم ثلاثة .. وهكذا إلى تسعة ثم يبدأ العد من جديد . حيث أنك لو قمت بكتابة نفس الكود مع portb ستجد أنه يعمل أما مع porta سيقف العد عند الواحد .

والآن قم بإضافة الأمر adcon1=0x07; في بداية الكود وستعمل معك الدائرة بشكل سليم .

قام بعدها فلان بالتأكد من هذا الكلام فوجد أنه صحيح مئة بالمئة .

وأخبره محمد المليجي أنه ربما يكتب نفس الكود السابق ولكن pic16f628a فيعمل بشكل سليم , ولكن دائما وأبدا لكي لا تقع في مشكلة ما مع هذه الأطراف عليك أن تجعل هذه الأطراف digital إما بكتابة الأمر adcon1=0x07; مع pic16f877a أو بكتابة ansel=0; مع pic16f882 .. وهكذا ؟ وأقل فائدة تستفيدها أن تغلق الأشياء التي لا تستخدمها .

هنا قاطعه فلان : ما هذا إني أرى أسماء أخرى غير adcon1 وغير cmcon مع أنواع أخرى من الميكروكنترولر فكيف أعرف هذه المعلومات بحيث أتمكن من استخدام أي ميكروكنترولر بدون مشاكل ؟

محمد المليجي : الحل بسيط جدا قم بقراءة باقي الدروس في كتاب " المفتاح البسيط للتعامل مع أنواع مختلفة من الميكروكنترولر " .

فلان : إن شاء اللّه سأقرأه .. شكرا لك.

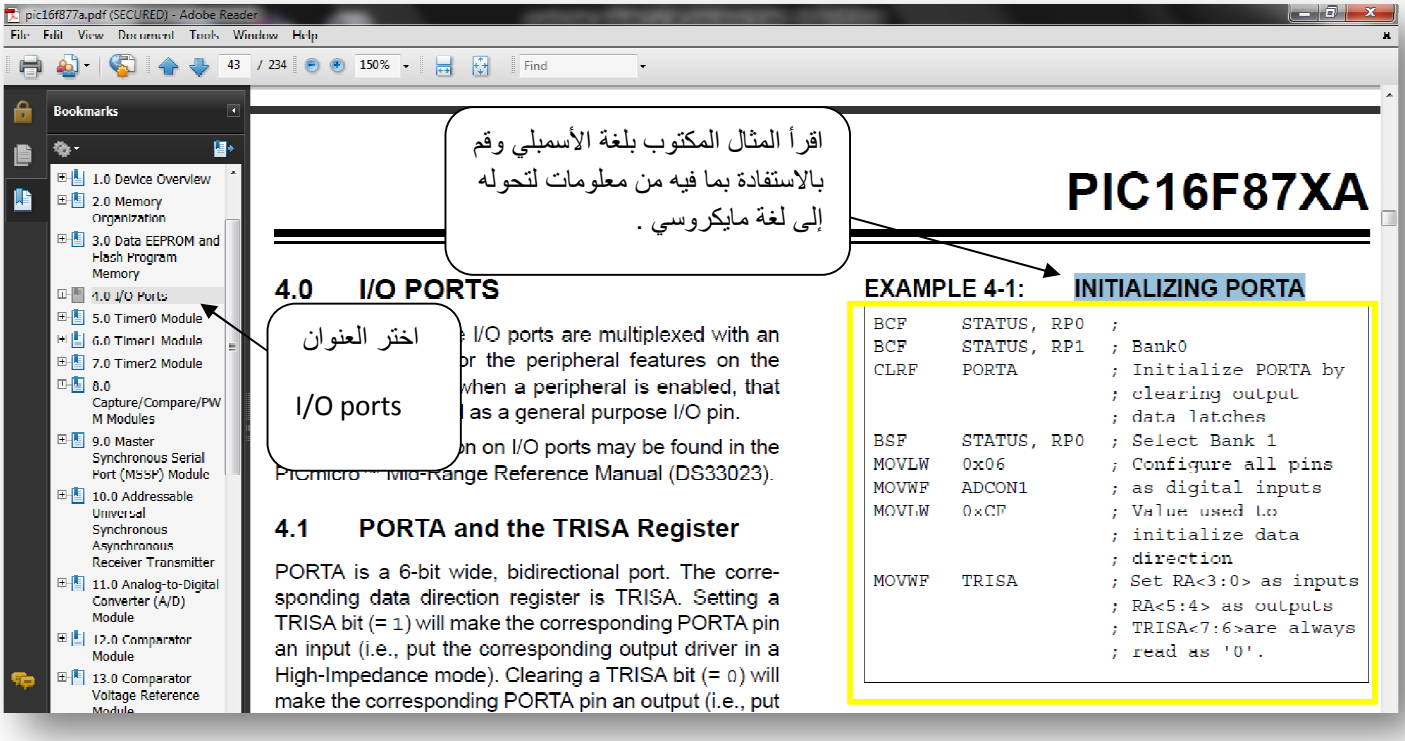
الطريقة السهلة التي تجعلك تبدأ مع أي PIC جديد عليك , هي أن تتبع الخطوات التالية :

1- افتح الداتا شيت datasheet الخاصة بهذا الـ PIC .

2- اختر العنوان I/O ports .

3- اقرأ المثال المكتوب بعنوان **INITIALIZING PORTA**

4- اكتب الكود المقابل بلغة مايكروسي بدلا من الأسمبلي .. كما في الصور التالية :



طريقة سهلة للحصول على الأوامر المطلوبة بلغة مايكروسي من المثال المكتوب بالأسمبلي :

1- انظر إلى المثال المكتوب بلغة الأسمبلي ولا تلتفت لأي سطر مكتوب فيه status لأنه لا يهمنا في لغة مايكروسي . أيضا لا تلتفت إلى اي سطر مكتوب فيه port أو tris فلا يهمنا كثيرا حيث أننا نكتب الأوامر المقابلة لها بالسي على حسب البرنامج المطلوب .

2- عندما تجد الأمر MOVLW وبجواره رقم معين فكأننا نقول للميكروكنترولر خزن هذا الرقم في عقلك .

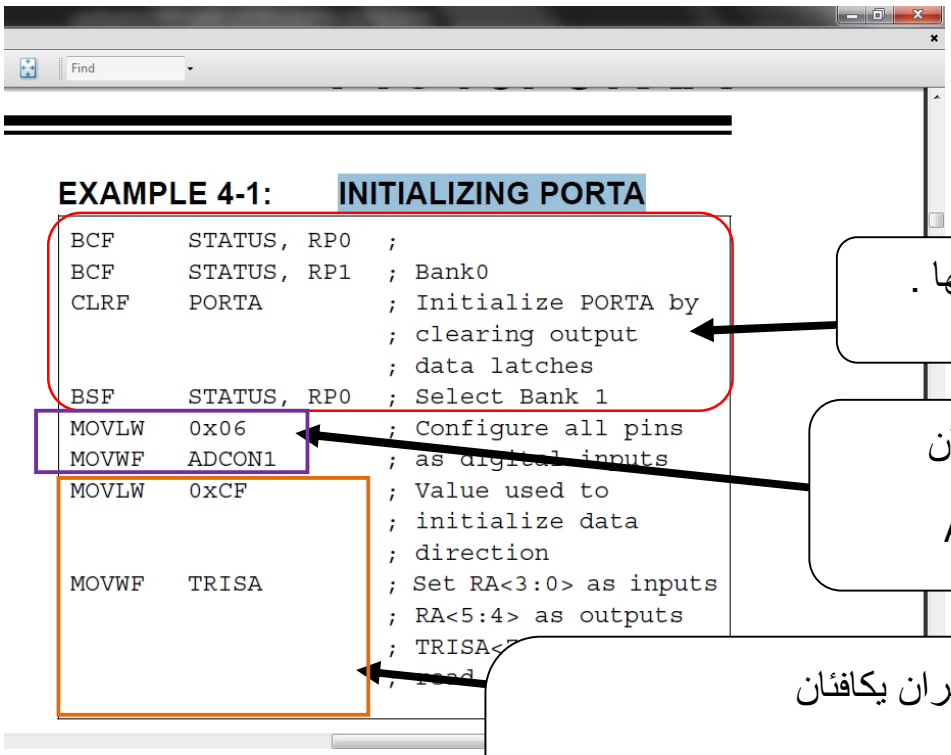
3- عندما تجد الأمر MOVWF وبجواره اسم مسجل (ريجستر REGISTER) فإننا نقول للميكرو اجعل هذا المسجل يساوي الرقم الذي سجلته في عقلك منذ قليل .

4- عندما تجد الأمر CLRF وبجواره اسم مسجل فهذا يعني أن قيمة هذا المسجل تساوي صفر . مثال : لو وجدت الأمر CLRF ANSEL هذا يحول إلى السي هكذا ANSEL=0; .

5- و بالهناء والشفاء . الآن PORTA سيعمل معك بدون مشاكل

والآن لنطبق هذه الطريقة مع بعض أنواع الميكروكنترولر من النوع pic :

Pic16f877a



EXAMPLE 4-1: INITIALIZING PORTA

```
BCF     STATUS, RP0   ;
BCF     STATUS, RP1   ; Bank0
CLRF    PORTA         ; Initialize PORTA by
                      ; clearing output
                      ; data latches
BSF     STATUS, RP0   ; Select Bank 1
MOVLW   0x06          ; Configure all pins
MOVWF   ADCON1        ; as digital inputs
MOVLW   0xCF          ; Value used to
                      ; initialize data
                      ; direction
MOVWF   TRISA         ; Set RA<3:0> as inputs
                      ; RA<5:4> as outputs
                      ; TRISA<
                      ; read
```

هذه الأوامر لن نلتفت إليها .

هذان الأمران يكافئان

Adcon1=0x06;

هذان الأمران يكافئان

Trisa=0xcf;

وبالطبع الأمر tris يمكننا تغييره كما نشاء على حسب الكود

لذا فلن نلتفت إليهم

الخلاصة : عند استخدام pic16f877a فإننا نكتب في البداية ADCON1=0X06;

Pic16f628A

EXAMPLE 5-1: Initializing PORTA

```
CLRF    PORTA         ;Initialize PORTA by
                      ;setting
                      ;output data latches
MOVLW   0x07          ;Turn comparators off and
MOVWF   CMCON         ;enable pins for I/O
                      ;functions
BCF     STATUS, RP1
BSF     STATUS, RP0 ;Select Ban
MOVLW   0x1F          ;Value used
                      ;data direc
MOVWF   TRISA         ;Set RA<4:0
                      ;TRISA<5> a
                      ;read as '1
                      ;TRISA<7:6>
                      ;depend on oscillator
                      ;mode
```

هذان الأمران يكافئان

CMCON=0X07;

أما باقي الأوامر لا تهمنا كما تعلم

Pic16f887/883/886/884/882

EXAMPLE 3-1: INITIALIZING PORTA

```
BANKSEL PORTA       ;
CLRF    PORTA       ;Init PORTA
BANKSEL ANSEL       ;
CLRF    ANSEL       ;digital I/O
BCF     STATUS,RP1  ;Bank 1
BANKSEL TRISA       ;
MOVLW   0Ch         ;Set RA<3:2> as inputs
MOVWF   TRISA       ;and set RA<5:4,1:0>
                    ;as outputs
```

هذا الأمر هو ما يهمنا حيث نحوله للغة السي كما يلي

ANSEL = 0;

أما باقي الأوامر لا تهمنا كما تعلم

لا تلتفت إلى أي شيء يكتب بجواره

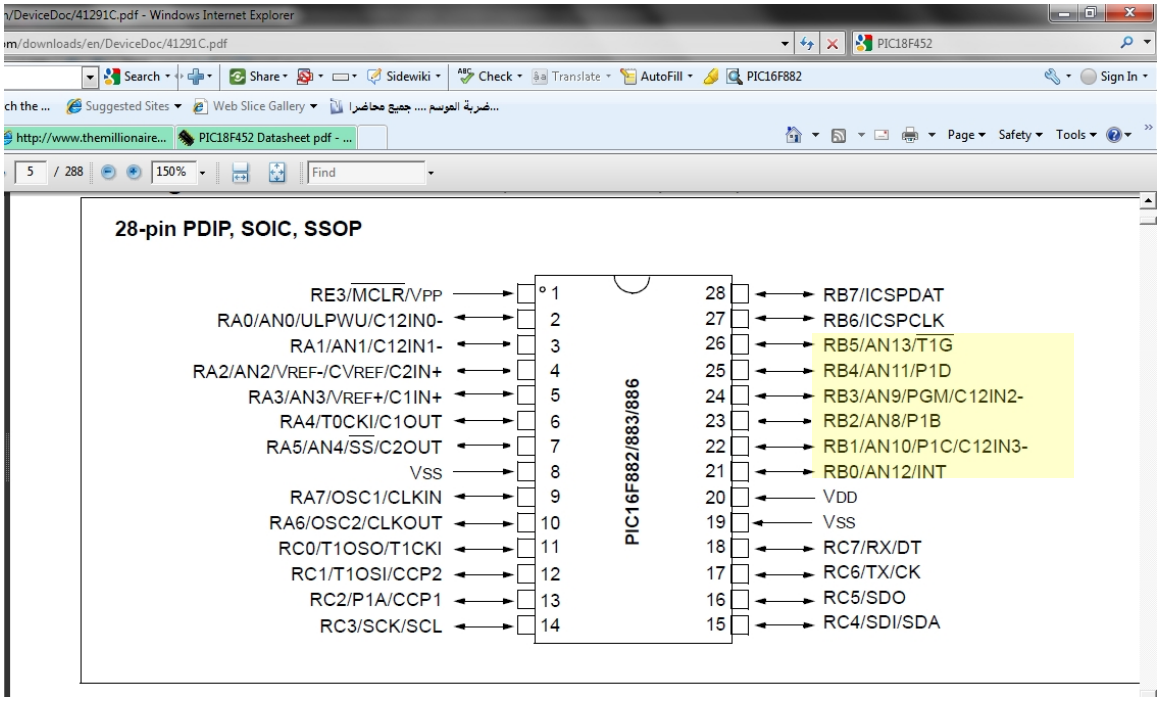
BANKSEL

ملحوظة : لكي نعلم أي الأطراف يمكن استخدامه كتماثلي ANALOG أو رقمي DIGITAL نلاحظ في PIN DIAGRAM أنه مكتوب بجواره مثلا AN0 أو AN1 .. وهكذا . وهذه الأطراف تكون PORTA غالبا . ولكن في بعض الأحيان يكون PORTB أيضا مكتوب بجواره AN9 مثلا أو اي رقم آخر .. وهنا لا نكتفي بالطريقة السابقة بل نكتب أيضا ANSELH=0; وهذا الكلام يطبق مع PIC16F882/884/887/883 ومع بعض الأنواع الأخرى . فتكون النتيجة النهائية أننا سنكتب الأمرين

```
ANSEL=0;
ANSELH=0;
```

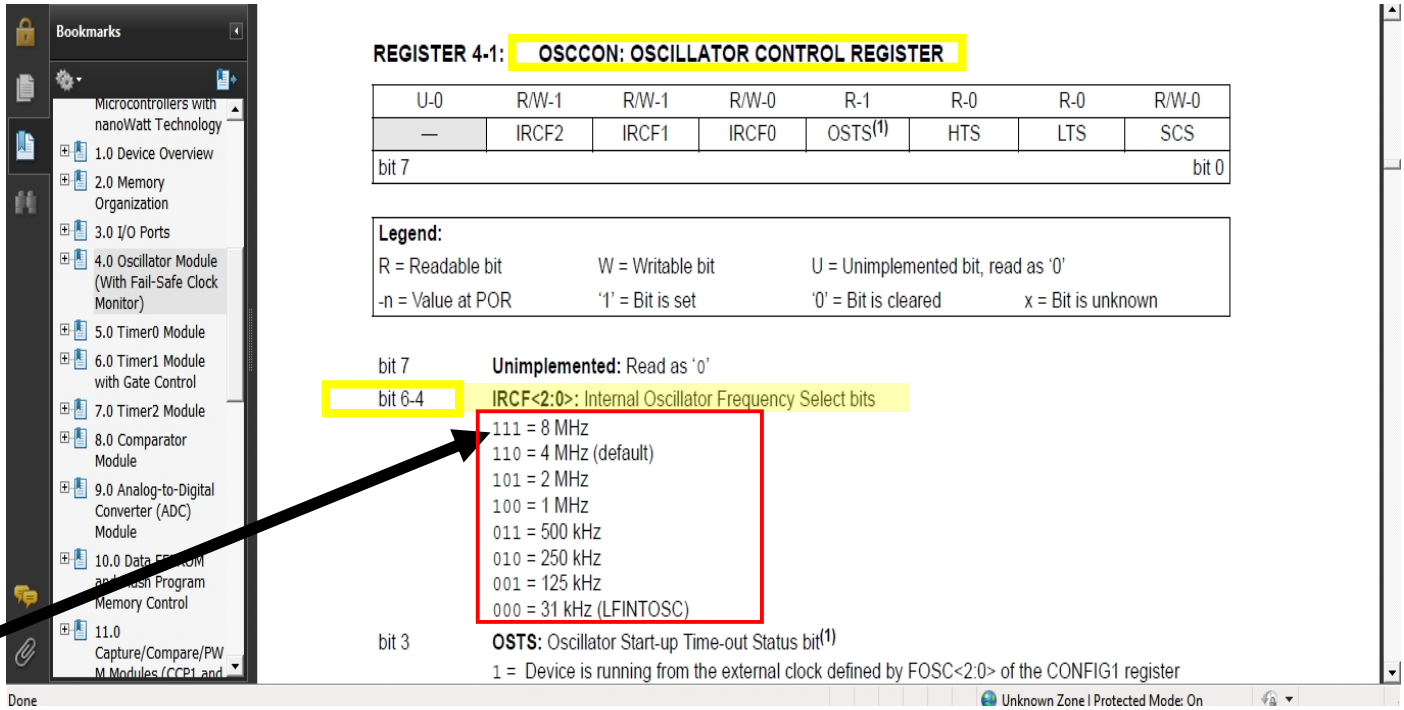
في بداية الكود .

هذه الأنواع رخيصة وأفضل بكثير من PIC16F877A من نواحي كثيرة .



من ميزات هذه الأنواع أن لديها مذبذب داخلي , يرجك من سعر الكريستال ومن مشاكله المختلفة وهذا المذبذب الداخلي له عدة سرعات . وأنت تعلم أنه كلما زادت سرعة المذبذب زادت سرعة تنفيذ الأوامر . وتعلم أيضا أنه إذا أردت أن تستخدم الأمر DELAY_US فإنك تحتاج إلى سرعة عالية للمذبذب ولا ينبغي أبدا اختيار سرعة منخفضة كـ 4 ميجا مثلا .

نرجع إلى موضوعنا وهو أنه يمكنك تحديد السرعة التي تريدها للمذبذب من خلال المسجل OSCCON وهو اختصار لكلمة OSCILLATOR CONTROL افتح الداتاشيت الصفحة 64 ستجد ما يلي :



ستجد أن السرعات المتاحة هي 8 ميجا و 4 ميجا و 2 ميجا و 1 ميجا و 500 كيلو و 250 كيلو و 125 كيلو و 31 كيلو كما في المربع الأحمر .

ويمكنك اختيار السرعة من خلال ضبطك لهذا المسجل للـ BITs من 4 إلى 6 .

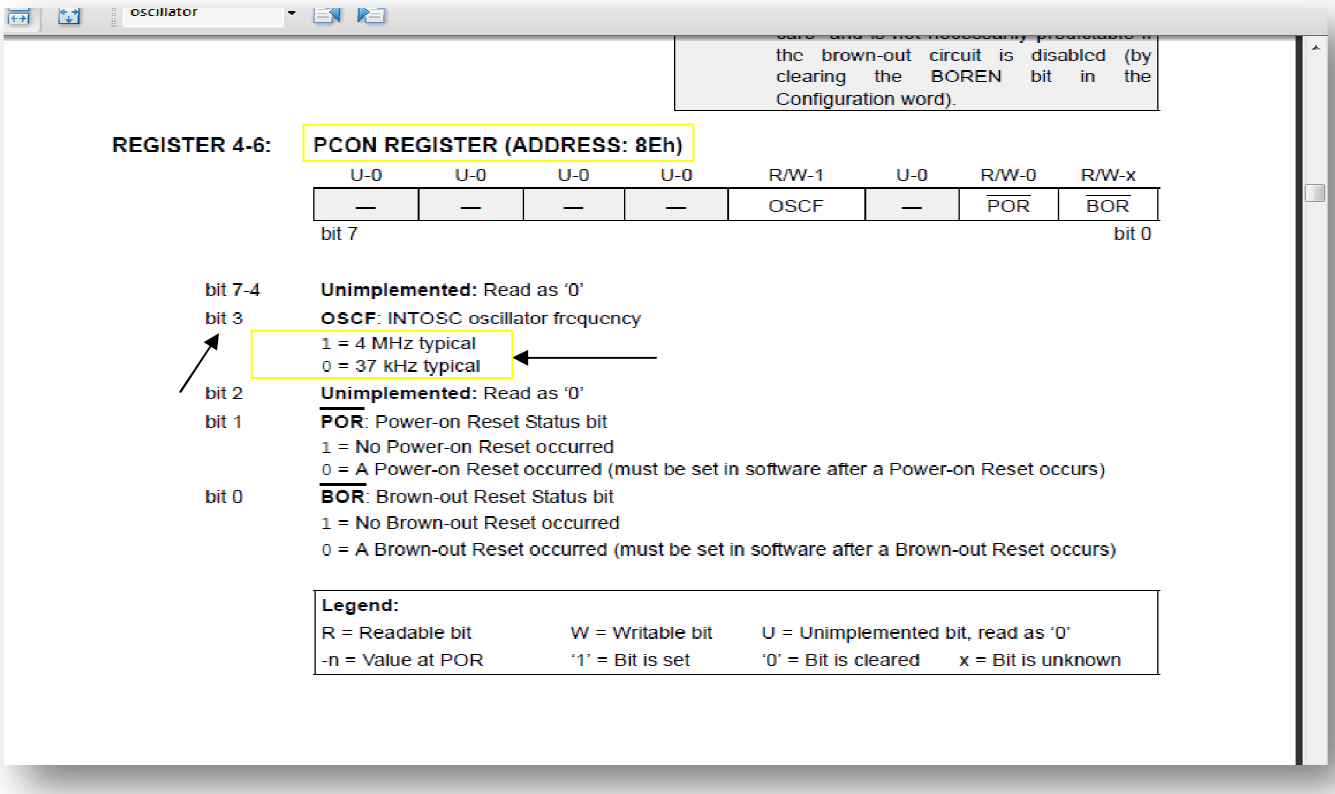
فمثلا لاختيار السرعة 8 ميجا ستجعل الـ bit الرابع يساوي واحد وكذلك الخامس والسادس .

فنكتب الكود التالي في بداية البرنامج `osccon.f4=1;osccon.f5=1;osccon.f6=1;`

أما لو أردنا جعل التردد 2MHz فإننا سنجعل الـbit الرابع يساوي واحد وكذلك السادس أما الخامس نجعله بصفر بالكود التالي `osccon.f4=1;osccon.f5=1;osccon.f6=1;`

وهكذا في باقي السرعات على حسب البيانات التي في الداتا شيت في المربع المشار إليه باللون الأحمر .

بما أننا تطرقنا إلى موضوع المذبذب الداخلي فإنه من الجدير بالذكر أن نقول أن pic16f628a يحتوي على مذبذب داخلي تردده يمكن أن يكون 4MHz أو 37KHz . ولاختيار السرعة يمكننا استخدام المسجل PCON . حيث أنه هنا لا يوجد OSCCON في هذا النوع .



thc brown-out circuit is disablcd (by clearing the BOREN bit in the Configuration word).

REGISTER 4-6: PCON REGISTER (ADDRESS: 8Eh)

U-0	U-0	U-0	U-0	R/W-1	U-0	R/W-0	R/W-x
—	—	—	—	OSCF	—	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **OSCF**: INTOSC oscillator frequency
1 = 4 MHz typical
0 = 37 kHz typical

bit 2 **Unimplemented:** Read as '0'

bit 1 **$\overline{POR}$**: Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$**: Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

لجعل التردد يساوي 4MHz نجعل الـ Bit3 من PCON يساوي واحد فنكتب الكود التالي :

```
PCON.f3=1;
```

ولجعل التردد يساوي 37MHz نكتب الكود التالي :

```
PCON.f3=0;
```

PIC18F2455/2550/4455/455

EXAMPLE 10-1: INITIALIZING PORTA

```
CLRF    PORTA   ; Initialize PORTA by
                ; clearing output
                ; data latches
CLRF    LATA    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0Fh     ; Configure A/D
MOVWF   ADCON1  ; for digital inputs
MOVLW   07h     ; Configure comparators
MOVWF   CMCON   ; for digital input
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISA   ; Set RA<3:0> as inputs
                ; RA<5:4> as outputs
```

هذا ما يهمنا فقط واستثني الباقي

حيث أنه عند التعامل مع هذه الأنواع نكتب في بداية الكود

ADCON1=0X0F;

CMCON=0X07;

هذه الأنواع من الميكروكنترولر تدعم خاصية استخدام USB بحيث أن دائرتك الالكترونية توصلها بالحاسب بالمنفذ USB ويتم تبادل البيانات والأوامر . كما أن لها ميزات أخرى كثيرة .

وبنفس المفتاح يمكنك استخدام أنواع مختلفة من الميكروكنترولر

ولكن أريد أن أقول أن مفتاحك الحقيقي والذي ترجع إليه دائما عند التعامل مع نوع جديد أن تقرأ الدتاشيت datasheet الخاصة به , وما أشرت إليه هو فقط أداة مساعدة تبسط لك الأمور لكن باقي الخفايا وباقي المعلومات الأخرى تجدها في الـ datasheet فحاول دائما الرجوع إليها وقراءتها ومحاولة فهمها فستفيدك كثيرا .

مأساة حقيقية

هل تعلم عزيزي القارئ أن هناك أنواع كثيرة من الميكروكنترولر تصدرها شركات مختلفة مثل microchip والتي تصدر النوع PIC و dsPIC وغيره , وشركة ATMEL تصدر النوع AVR وأيضا 8051 و ATtiny .. وهناك شركات أخرى كثيرة , وهكذا كل شركة تصدر أنواع مختلفة والتنافس بينهم شديد فكل شركة تصدر إصدار أحدث وبإمكانيات أعلى يكون في حسبانها أيضا السعر فتحاول أن تجعل السعر أقل ما يمكن .

ولقد أخبرتكم أن الأنواع الأحدث أرخص من الأنواع الأقدم لسببين السبب الأول هو ماذكرته منذ قليل من تنافس بين الشركات ونحوه , والسبب الثاني أن الشركات الموزعة (المستوردين) يشترون كميات كبيرة من النوع الواحد عندما تصنعه الشركة المصنعة , فإذا قامت الشركة فيما بعد بتقليل السعر فسيخسر هؤلاء الموزعين الكثير من المال لأن لديهم كميات كبيرة من النوع القديم , لذلك غالبا لا تقوم الشركة المصنعة بتنزيل السعر وتعوض ذلك بأن تجعل النوع الأحدث أرخص , وما كان يحزنني أن كثير من الناس يبتعد عن الأنواع الحديثة هذه كل البعد (لا أدري خوفا منها أم إيمانا بالمثل الذي يقول " اللي تعرفه أحسن من اللي ما تعرفوش") ومن حاول منهم استخدام هذه الأنواع الحديثة واجهته بعض المشاكل مع porta مثلا فقام بتخويف الآخرين ويقول بكل جهل " هذا النوع من الميكرو به عيوب وسيء " وهنا يقع بائعوا العناصر الالكترونية في مشكلة أنهم يطلب منهم من الزبائن الأنواع القديمة فقط والأنواع الحديثة عند شراءها لا يطلبها أحد منهم .ومن يريد مواكبة التطور من المتعلمين يضطر إلى أن يشتري من الخارج عن طريق الانترنت .

فحاولت أن أقدم هذا الكتيب الصغير لعل وعسى أن تحل هذه المشكلة ونتطور أكثر ولا نتخوف أبدا من الأنواع الحديثة ونوفر بعض المال أيضا بشراءنا للأنواع الأخرى الأحدث .

إن استفدت من هذا الكتيب فلا تنساني من دعواتك .

وأرجو أن تنشر هذا الكتيب في مختلف المواقع كي تعم الفائدة , وتنال الثواب العظيم .

إن أردت التبرع المالي لدعم المؤلف للقيام بمثل هذه الأعمال الخيرية من كتب علمية وفيديوهات ودورات راسلني على ايميلي ahmad_s_f@hotmail.com لمعرفة كيفية التبرع

واللّه الموفق , ودمتم في حفظ اللّه

أخوكم / م . أحمد سمير فايد