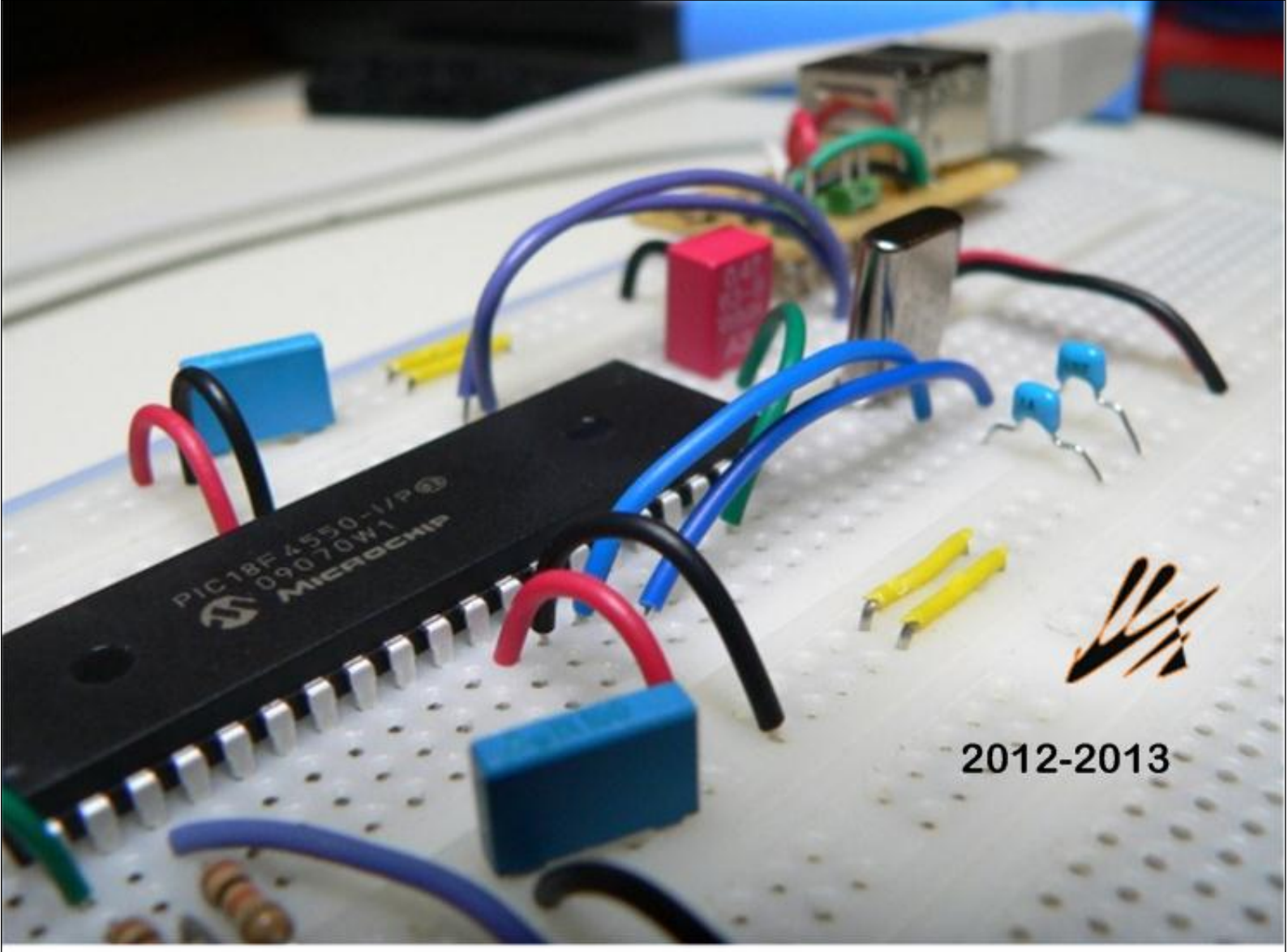


2012-2013

مشروع أعد لنيل إجازة في كلية الهندسة الكهربائية والإلكترونية

تطبيقات الاتصالات التسلسلية باستخدام المايكرو كونترولر

دراسة لبنية المتحكمات من عائلة PIC18F

إعداد الطالب :

مصطفى حاج عبد الرحمن

بإشراف : د. حليم علي

USART

PS2

USB

I2C

Contents

Architecture

Architecture

reset circuit

Power Supply

Clock Sources

Memory Organization

I/O Ports

A/D

TIMER0

PWM

Protocol

Ps/2

SIRC

SPI

I 2 C

One Wire

RS232

USB

جدول المحتويات

1.1. أهمية المتحكمات في تطبيقات الاتصالات ! 9
1.2. لماذا المتحكمات من عائلة PIC18 ؟ 9
1.3. عن الشركة (Introduction) : 9
1.4. الميزات الأساسية لعائلة PIC18 (basic features of PIC18): 10
..... 13
Microcontroller Architectures 13
1.1. بنية عائلة PICC18XX2 (Architecture PICC18XX2) : 13
1.2. جهد التغذية (Power Supply): 17
1.3. أنواع دارات التصفير (reset circuit) : 22
1.4. تصفير انخفاض الجهد (out reset-Brown) : 23
1.5. مؤقت المراقبة (timer Watchdog) : 23
1.6. مصادر الساعة (Clock Sources) : 24
1.6.1. الهزاز الخارجي في النمط EC 25
1.6.2. الهزاز الخارجي في النمط ECIO 26
1.6.3. النمط LP : 26
1.6.4. النمط XT : 26
1.6.5. النمط HS : 27
1.6.6. النمط RC : 27
1.6.7. النمط RCIO : 28
1.6.8. النمط HSPLL : 29
1.7. أنواع الذواكر (MEMORY): 31
1.7.1. Flash Memory 31
1.7.2. (RANDOM ACCESS MEMORY (RAM 31
1.7.3. (ELECTRICALLY ERASABLE PROGRAMMABLE ROM (EEPROM 31
1.8. تنظيم ذاكرة البرنامج (Program Memory Organization) : 32
1.9. تنظيم ذاكرة المعطيات (Data Memory Organization) : 33
1.10. مسجلات التهيئة (Configuration Registers): 35
المحيطيات 36
1.1. نوافذ الدخل / الخرج (I/O Ports) : 36

.1.2 المحول التشابهي الرقمي ADC : 45
.1.2.1 ميزات المحول التشابهي الرقمي ADC : 46
.1.2.2 أين يتم تخزين نتيجة التحويل وكيف ذلك ؟ 46
.1.2.3 دقة التحويل : 47
.1.2.4 خطوات التحويل : 52
.1.3 العداد/المؤقت TIMER0 : 53
.1.3.1 مبدأ العمل : 60
.1.3.2 عمله كمؤقت : 61
.1.3.3 عمله كعداد : 64
.1.4 الماسك / المقارن / تعديل PWM : 66
.1.4.1 نمط المقارنة Compare : 69
.1 Protocol The PS/2 Communication 72
.1.1 لوحات المفاتيحKeyboards : 72
.1.2 PS/2 Physical Interfaceتحقيق الواجهة 72
.1.3 البروتوكول The PS/2 Communication Protocol 73
.1 Protocol SIRC 79
.1.1 الاتصالات عن طريق الأشعة تحت الحمراء olProtoc Sony SIRC : 79
.1.2 الدارة العملية : 83
.1 (Synchronous Serial Port (MSSP 87
.1.1 SPI : 88
.1.2 الناقل (I 2 C) : 89
.1.2.1 اشارة حالة البدء والإيقاف : 92
.1.2.2 كيف يتم مخاطبة العبد المطلوب ؟ 92
.1.2.3 كيف نحصل على عنوان لتجهيزة ما : 93
.1 (Serial EEPROM (24LC512 95
.1.1 كتابة بايت الى الذاكرة : 95
.1.2 عملية القراءة 96
.1.2.1 قراءة العنوان الحالي Current address read signal 96
.1.2.2 القراءة العشوائية Random read operation 97
.1.2.3 القراءة المتسلسلة Sequential read operation: 98

.1.3 طريقة ربط ذاكرة خارجية مع المايكرو:..... 98

.1.4 تسلسل إجراء كتابة بايت واحد على الذاكرة : 100

.1.5 تسلسل إجراء كتابة أكثر من بايت على الذاكرة:..... 106

.1.6 طريقة القراءة من الذاكرة :..... 106

.1.6.1 الطريقة المباشرة :..... 106

.1 (I2C temperature sensor (DS1631 108

.1.1 عملية قياس درجة الحرارة TEMPERATURE MEASURING—OPERATION..... 111

.1.2.1 conversion mode-continuous: 111

.1.2.2 shot mode-one : 111

.1 Wire protocol–1..... 118

.1.1 "1 etirW" langis 118

.1.2 "0 etirW" langis 119

.1.3 "daeR" langis 119

.1.4 "ecneserP/teseR" langis 119

.1 232–Recommended Standard..... 121

.1.1 بروتوكول الاتصال التسلسلي (232-Recommended Standard) :..... 121

.1.2 نافذة (USART) :..... 127

.1 USB..... 132

.1.1 تقنية ارسال المعطيات : 133

.1.2 شعارات المسرى (Logos) :..... 134

.1.3 التركيب البنيوي للمسرى (opologyBus T):..... 134

.1.4 اصدارات البروتوكول :..... 136

.1.5 عرض اطار المعطيات..... 136

.1.6 أشكال الموصلات ((Connectors :..... 138

.1.7 اساسيات النقل (Transfer Basics):..... 139

.1.8 تحديد السرعة :..... 141

.1.9 صفوف التجهيزات (Device Classes):..... 142

.1.10 مصطلحات عامة :..... 144

الى من حُبُه قدر

مُحمد ﷺ

تمهيد :

يتلخص هذا المشروع " **تطبيقات الاتصالات التسلسلية باستخدام المايكرو كونترولر**" حول كيفية توظيف المتحكمات واستثمارها في مجال الاتصالات ، حيث لم يعد يخفى على أحد أن المتحكمات لم تعد مقتصرة على أعمال التحكم و الأتمتة لا سيما بعد أن شهدت تحسينات وإضافات نوعية تكاد تخرج المتحكم عن الغاية الأساسية المصمم من أجلها ، لذلك في هذا المشروع سنقوم بعون الله باستعراض وشرح لبنية المتحكمات عائلة PIC18F والتي تعتبر جيدة جدا من أجل التطبيقات المتوسطة وحتى الكبيرة فهي تلبي أغلب المتطلبات .

Mustafa haj abd alrahman

مراحل الدراسة والعمل (The Research on Time Line) :

تنقسم هذه المراحل الى ثلاث مراحل أساسية :

- الدراسة النظرية لبنية المتحكمات : في البداية توقفت عند دراسة البنية الداخلية لمتحكمات العائلة PIC18 والتي تعتبر مرحلة أساسية من أجل بقية المراحل حيث بمثابة حجر الأساس لبقية المراحل حتى العملية منها .
- الدراسة النظرية لأهم بروتوكولات الاتصالات التسلسلية :
  - ❖ وتتضمن هذه المرحلة دراسة بروتوكول الأشعة تحت الحمراء SIRC
  - ❖ دراسة بروتوكول PS2 المستخدم في الحواسيب
  - ❖ دراسة البروتوكول I2C
  - ❖ دراسة البروتوكول one wire
  - ❖ الدراسة النظرية للبروتوكول SPI وتطبيقاته
  - ❖ دراسة المعيار الهام RS232 .
  - ❖ وأخيرا أختم الدراسة النظرية بالبرتوكول الأهم USB2.0
- التنفيذ العملي للدارات :تم بناء الدارات على أربع مراحل متتالية
  - ❖ الأولى رسم الدارات ومحاكاتها عن طريق برنامج ISIS 7 Professional
  - ❖ الثانية رسم الدارات المطبوعة عن طريق شقيق ARES 7 Professional الذي يأتي ضمن نفس الحزمة .
  - ❖ الثالثة تتضمن كتابة الشيفرات للمتحكم PIC18F452 وذلك من خلال المترجم MikroC Pro 5.61
  - ❖ الرابعة كتابة التطبيقات بلغة عالية المستوى مثل C# ، Math lab .

الفصل الأول

Microcomputer Systems

1.1. أهمية المتحكمات في تطبيقات الاتصالات !

لمعرفة ذلك يكفي أن نعلم أن المتحكمات لم يعد يقتصر عملها الأساسي على قيادة مجموعة من الليدات أو قيادة محرك خطوي و إنما تم تزويدها بواجهات تحقق أكثر أنواع الاتصالات تعقيدا حيث أن المتحكم PIC18f4550 مزود بواجهة USB 0.2 وبعضها مزودة بواجهة TCP/IP ومنها ما هو مزود بواجهة ZIGbee وبعضها مزود بواجهة CAN .

1.2. لماذا المتحكمات من عائلة PIC18 ؟

بالرغم من الاستخدامات الواسعة لمتحكمات PIC16 الا أنها مقصرة بدعم بعض أنواع الاتصالات مثل USB ، CAN وتحقيق مثل هذه الاتصالات باستخدام pic16 لا يعتبر أمرا سهلا أبدا .

1.3. عن الشركة (Introduction) :

قامت شركة Microchip بتطوير عائلة جديدة من المتحكمات PIC18 من أجل التطبيقات المعقدة تقدم هذه العائلة ذاكرة برنامج من نوع flash ذات سعة من 8 حتى 128K وذاكرة معطيات من 256 حتى 4KB وتعمل ضمن مجال من 2 حتى 5volt وسرعة حتى 40Mhz .

1.4. الميزات الأساسية لعائلة PIC18 (basic features of PIC18):

77 تعليمة

متوافقة مع الكود المصدري لعائلة PIC16

تستطيع عنونة ذاكرة برنامج حتى 2Mbyte

تستطيع عنونة ذاكرة معطيات حتى 4Kbyte

امكانية التحكم بأولوية المقاطعات

عرض التعليمة 16-bit في حين أن عرض ممر المعطيات 8-bit

تحوي حتى 2 مؤقت/عداد بعرض 8-bit

تحوي حتى 3 مؤقت عداد بعرض 16-bit

تدعم حتى 4 أنواع من المقاطعات الخارجية

استهلاك تيار عالي يصل حتى 250mA

تحوي حتى 5 موديولات ماسك / مقارن / التعديل pwm

موديول للاتصال التسلسلي يدعم I2C ، SPI

تحوي حتى 2 موديول USART

محول تشابهي رقمي بعرض 10-bit

البرمجة عند الجهد المنخفض LVD

علما أن بعض المتحكمات من هذه العائلة مزودة بواجهات إضافية وهي :

واجهة ممر CAN

واجهة USB 0.2

واجهة LCD

واجهة ICP/IP

واجهة ZIGbee

واجهة للتحكم بالمحركات

والجدول التالي يقارن بين الميزات بين أقسام العائلة الأربعة :

Feature	PIC18F242	PIC18F252	PIC18F442	PIC18F452
Program memory (Bytes)	16K	32K	16K	32K
Data memory (Bytes)	768	1536	768	1536
EEPROM (Bytes)	256	256	256	256
I/O Ports	A,B,C	A,B,C	A,B,C,D,E	A,B,C,D,E
Timers	4	4	4	4
Interrupt sources	17	17	18	18
Capture/compare/PWM	2	2	2	2
Serial communication	MSSP USART	MSSP USART	MSSP USART	MSSP USART
A/D converter (10-bit)	5 channels	5 channels	8 channels	8 channels
Low-voltage detect	yes	yes	yes	yes
Brown-out reset	yes	yes	yes	yes
Packages	28-pin DIP 28-pin SOIC	28-pin DIP 28-pin SOIC	40-pin DIP 44-pin PLCC 44-pin TQFP	40-pin DIP 44-pin PLCC 44-pin TQFP

الجدول 1

The 18FXX2 microcontroller family

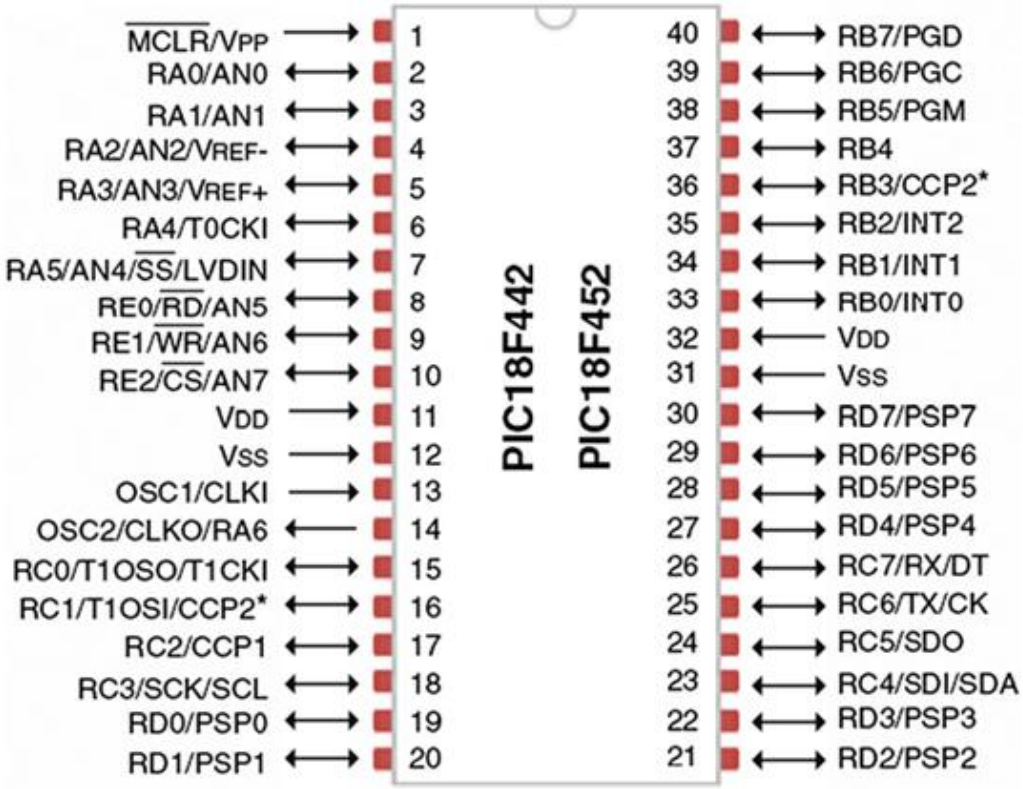
Device	Flash 8/16-Bit (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)	I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	SPI	Master I2C™	EAUSART	Comparators	Timer 8/16-Bit
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

الجدول 2

Architectures

1.1. بنية عائلة PICC18XX2 (Architecture PICC18XX2) :

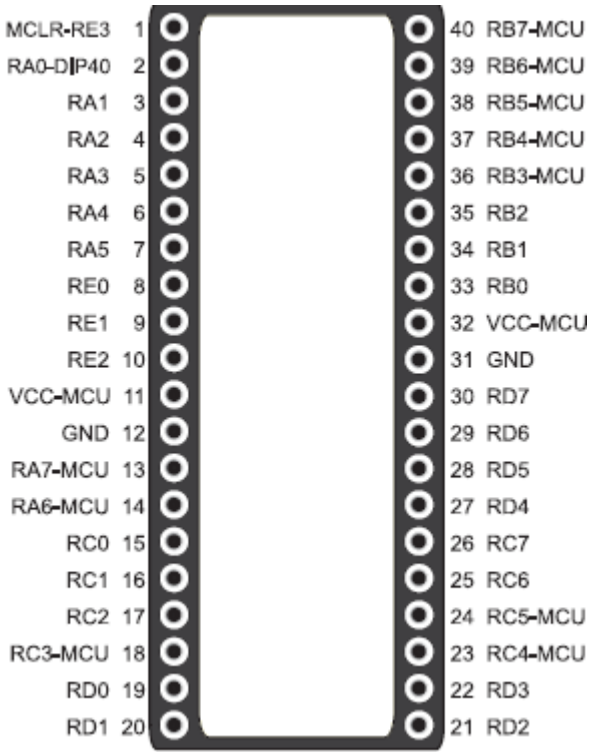
كما هو واضح في الجدول فإن عائلة PIC18FXX2 تحتوي أربعة أقسام . حيث أن متحكمات PIC18F2X2 هي ذات 28 رجل ، بينما PIC18F4X2 هي ذات 40 رجل بنية كلا المجموعتين بالغالب متطابقة ما عدا أن المتحكم الأطول لديه المزيد من أرجل الدخل/الخرج ومزيدا من القنوات التشابهية .في هذا البحث سوف نلقي الضوء على بنية المتحكم PIC18F452 بشيء من التفصيل ، وهي تكفي لفهم بقية العمليات على المتحكمات الأخرى من عائلة PIC18 .وظائف الأرجل للمتحكم PIC18F452 يبينها الشكل التالي ، هذا المتحكم لديه 40 رجل DIL .



الشكل 1

PIC18F452 microcontroller DIP pin configuration

أما الشكل فهو يظهر البلوكات الداخلية للمتحكم PIC18F452 ، حيث أن الCPU وتتألف من ALU8-bit ومسجل (WREG) الذي يدعى بالمراكم أيضا بعرض 8-bit ، كذلك ضارب هاردوير 8x8 حيث البايت العلوي و البايت السفلي لعملية الضرب تخزن في مسجلين كل منهما بعرض 8-bit يدعى PRODH ، PRODL .



الشكل 2

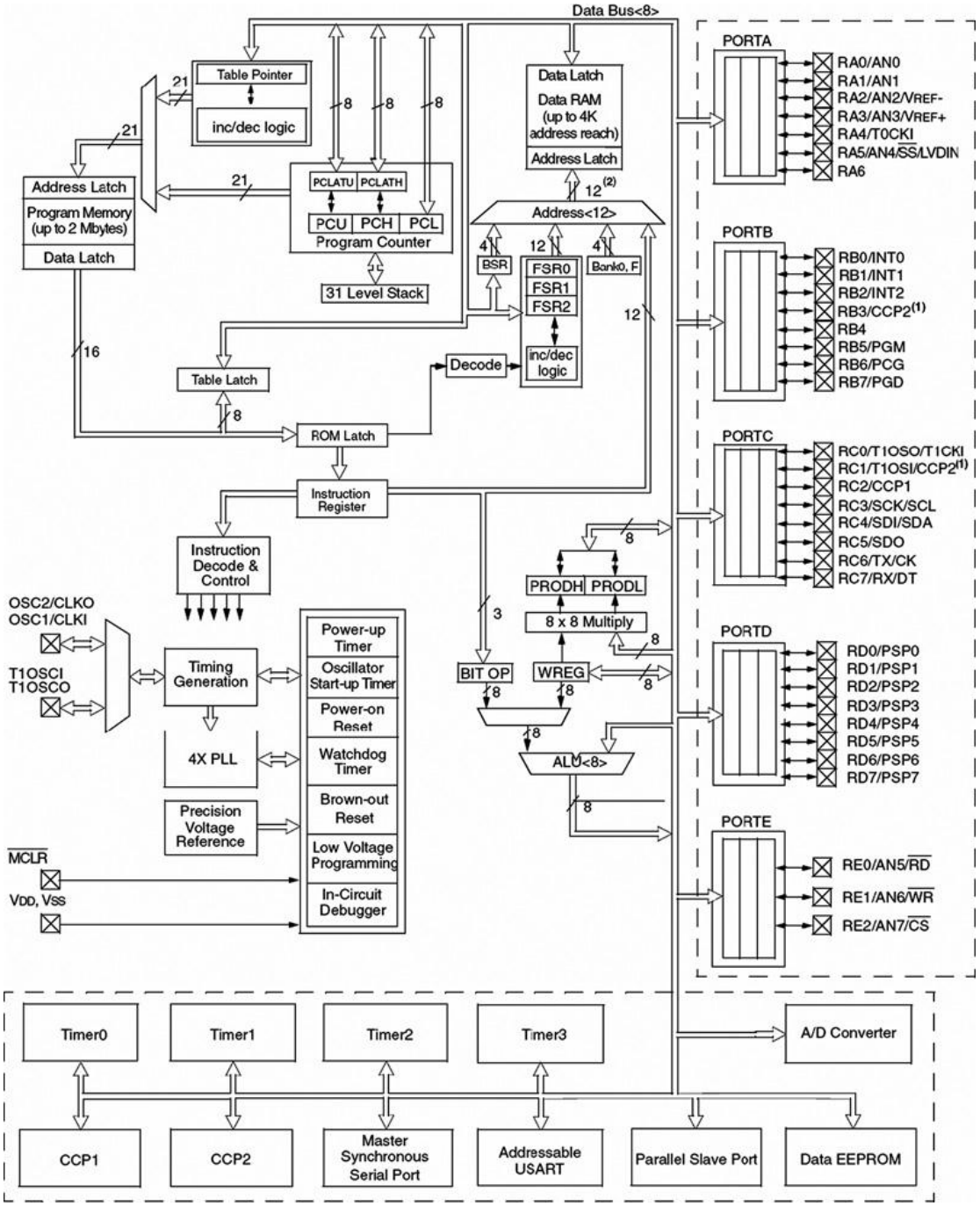
عداد البرنامج و ذاكرة البرنامج تظهر في القسم العلوي اليساري من الشكل عنوان ذاكرة البرنامج تتألف من 21bit قادرة على عنونة ذاكرة بحجم 2Mbyte ، المتحكم PIC18F452 لديه فقط 32K من ذاكرة البرنامج والتي تتطلب فقط 15-bit لعنونتها لذلك فإن البتات الستة الباقية تهمل ولا تستخدم .

ذاكرة البرنامج لديها مكدس وهو بعمق 31 مستوى ويستخدم بالعادة عنوان برنامج المقاطعة و عناوين البرامج الفرعية.

ذاكرة المعطيات ستظهر أعلى ومنتصف الشكل عرض ممر ذاكرة المعطيات 12-bit تسمح بعنونة 4kbyte من الذاكرة

ذاكرة المعطيات تتألف من مسجلات خاصة تدعى SFR ومسجلات للأغراض العامة وجميعها منتظمة في شيئ يدعى البنوك (بنوك الذاكرة)

الجزء السفلي من الشكل يظهر مسجلات المؤقتات/ العدادات ، الماسك/المقارن/تعديل عرض النبضة ، USART ، A/D ، الذاكرة EEPROM



الشكل 3

Block diagram of the PIC18F452 microcontroller

1.2. جهد التغذية (Power Supply):

كما هو واضح من الشكل التالي فإن المتحكم PIC18F452 في حال أردنا أن يعمل بالسرعة الكاملة أي عند 40Mhz يجب أن جهد التغذية يتراوح بين 4.2V الى 5.5V .الإصدار المنخفض التغذية من المتحكم PIC18F452 يدعى PIC18LF452 يمكن أن يعمل ضمن مجال الجهد من 2 الى 5.5V وللعمل عند الجهد المنخفض 2V فإن السرعة العظمى 4Mhz ويمكن العمل عند 40Mhz عند زيادة الجهد الى 4.2V .الذاكرة RAM تتطلب جهد 1.5V وسوف تفقد محتوياتها في حال انخفاض الجهد عن هذه القيمة ، في الغالب فإن معظم المتحكمات تعمل عند الجهد 5V والتي تتولد عن طريق منظم الجهد .

المتحكم PIC18F452 يضم:

- 4 مؤقت/عداد
- 2 موديول من مقارن/ماسك /تعديل عرض النبضة
- 2 موديول للاتصالات التسلسلية
- 8 قنوات للمحول التشابهي الرقمي 10-bit A/D
- ذاكرة EEPROM ذات سعة 256 bytes

دارة الهزاز متوضعة على الجانب اليساري من الشكل وتتألف من :

- Power-up timer
- Oscillator start-up timer
- Power-on reset
- Watchdog timer
- Brown-out reset
- Low-voltage programming
- In-circuit debugger
- PLL circuit
- Timing generation circuit

دارة PLL هي دارة جديدة في عائلة PIC18F وتقدم لنا خيار مضاعفة التردد وذلك لزيادة سرعة جميع العمليات ، هذه الدارة مهمة جدا ولاسيما عند استخدام هذا المتحكم من أجل تحقيق اتصال مع المسرى USB 0.2 كما سنرى لاحقا .

مؤقت المراقبة (watchdog) يمكن استخدامه من أجل اجبار المتحكم على عمل ريستارت في حال فشل البرنامج .

In-circuit debugger تتيح لنا تنقيح البرنامج أثناء عمله حيث تسمح لنا بتشخيص المعطيات داخل المسجلات مع إمكانية تعديلها وتفيد هذه الطريقة بتتبع الأخطاء و عند تطوير البرنامج .

نوافذ الدخل والخرج وهي تتوضع على الجانب اليميني من الشكل ، المتحكم PIC18F452 لديه 5 نوافذ تدعى PORTA, PORTB, PORTC, PORTD, PORTE كل بورت مكون من مجموعة من الأرجل وكل رجل لها وظيفة أو أكثر على سبيل المثال أرجل PRRTA يمكن أن تصبح أرجل دخل/خرج ويمكن أن تصبح قنوات تشابهية من أجل قياس الجهد التشابهي وتحويله الى قيم رقمية . أرجل البورت B (PORTB) يمكن أن تستخدم كأرجل دخل/خرج أو

أرجل مقاطعة دخل .

أنواع التصفير (reset sources) :

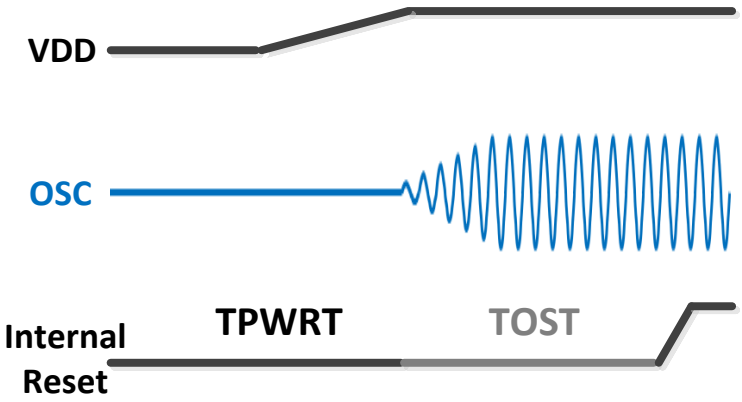
- التصفير عند وصل التغذية Reset during power on (POR)
- التصفير الخارجي باستخدام القطب reset during normal operation (MCLR)
- التصفير عن طريق مؤقت الحراسة Watchdog timer reset (WDT)
- التصفير عند انخفاض الجهد Brown-out reset (BOR)

التصفير حالة تسبب وبشكل مباشر أن المتحكم يقوم بايقاف جميع أعماله ويمسح مسجلاته . وبالنسبة لإشارة التصفير يمكن أن تكو خارجية (تطبيق منطق منخفض على MCLR) أو داخلية (عند وصل التغذية) .

هناك عدة أمور يقوم بها المتحكم عند وصول التغذية اليه ليتأكد من أن جميع داراته تقوم بالعمل على أكمل وجه

أولا يجب أن يكون هناك بعض التأخير وذلك حتى يستقر جهد التغذية ، كذلك حتى يستقر عمل الهزاز ، هناك مؤقتان مهمتهما هي تأمين التأخير وهما PWRT و المؤقت OST ، حيث أن المؤقت PWRT نستطيع تمكينه أو عدم تمكينه وذلك عند البرمجة

عندما يصل جهد التغذية بين 1.2 ، 1.7V فإن دارة تدعى power-up timer تقوم بتصفير المتحكم لمدة 72ms يعطي هذا الوقت تأخيرا كافيا حتى يسمح للجهد VDD بالوصول الى قيمته النهائية ، وحال انتهاء هذا الزمن تقوم دارة أخرى تدعى Oscillator start-up بتصفير المتحكم لمدة 1024 دورة آلة وهذا يسمح للهزاز الكريستالي بالوصول الى حالة الاستقرار وعند انتهاء هذه المدة (التي تدعى T reset) وحيث وطبعا جميع الأمور محققة فإن



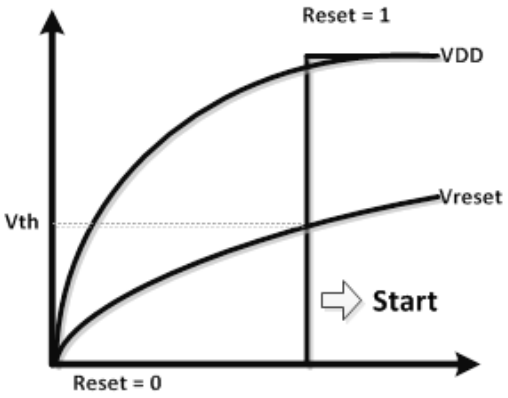
الشكل 4

المتحكم سيبدأ عمله ، الشكل السابق ليس دقيق وإنما يعطي فكرة عامة حيث أن هناك زمن ميت Tdead.

V_Rest هو الجهد المطب على الرجل MCLR ، Vth هد العتبة للرجل MCLR

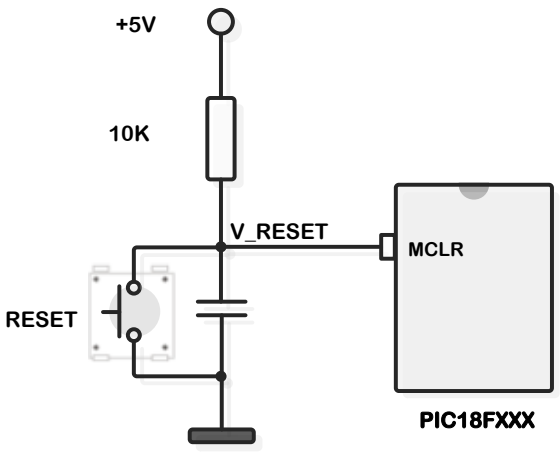
اذا كان V_rest < Vth فإن المتحكم يفهم أن القيمة هي جهد منخفض أي REST=0 .

اذا كان V_rest > Vth فإن المتحكم يفهم أن القيمة هي جهد منخفض أي REST=1 .



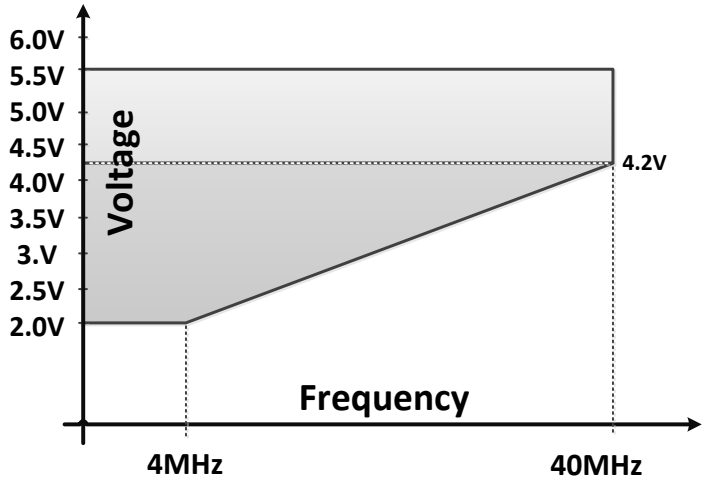
الشكل 5

المقاومة مع المكثف تشكل دارة RC ذات زمن شحن ثابت T= RC ، حيث أن المكثف سوف يشحن من خلال المقاومة R واذا كان زمن الشحن كبير بما فيه الكفاية T فإنه من خلال هذا الزمن سيكون الجهد V_Rest أقل من Vth (جهد العتبة) وبالتالي هذا يعطي وقت للجهد VDD حتى يصل الى حالته المستقرة (بهذه الطريقة يبقى المتحكم بحالة تصفير حتى يستقر جهد التغذية) .



الشكل 6

تتطلب المتحكمات PIC18F452 كما هو مبين بالشكل التالي جهد تغذية من 4.2V الى 5.5V عند عملها بالسرعة الكاملة أي عند التردد 40Mhz أما المتحكم PIC18LF452 (حيث يدل الحرف L على أنه منخفض التغذية) يمكن أن يعمل من 2V الى 5.5 وعندها فإن أعظم تردد هو 4Mhz وليعمل عند التردد 40Mhz يجب رفع الجهد الى 4.2V كما هو واضح من الشكل التالي .الذاكرة RAM تتطلب جهد 1.5V وسوف تفقد قيمتها في حال انخفاض الجهد عن هذه القيمة في الواقع فإن أغلب المتحكمات تعمل عند الجهد 5V والذي يمكن توليده بسهولة عن طريق منظم الجهد 5V .

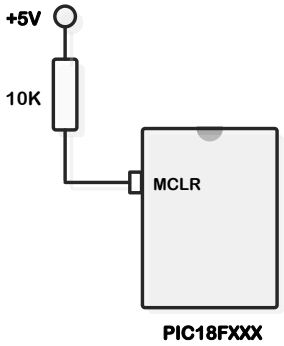


الشكل 7

Operation of PIC18LF452 at different vol

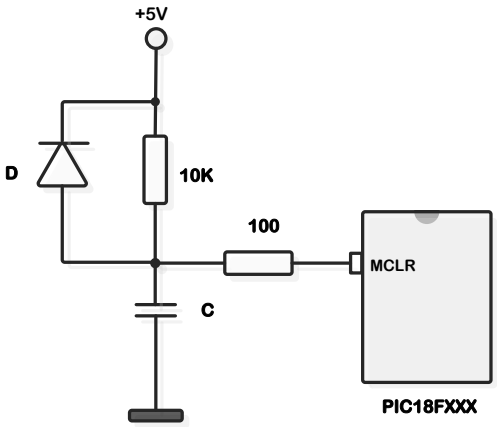
1.3. أنواع دارات التصفير (reset circuit) :

❖ Typical reset circuit



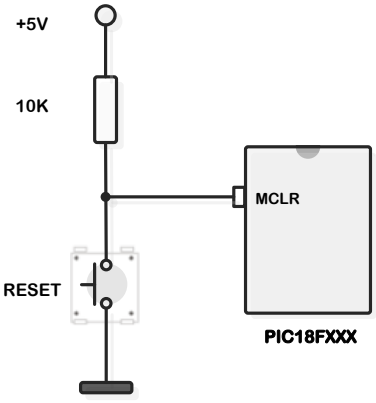
الشكل 8

❖ slow-rising voltages



الشكل 9

❖ External reset circuit



الشكل 10

1.4. تصفير انخفاض الجهد (Brown-out reset) :

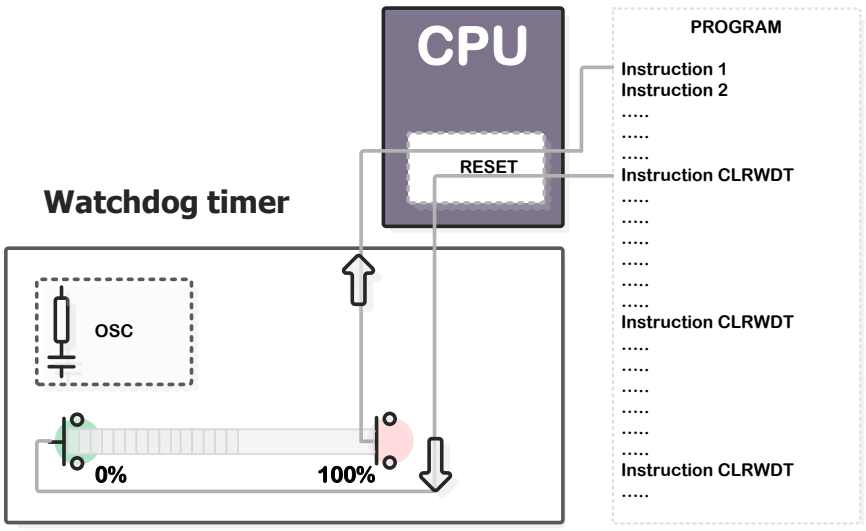
عندما ينخفض جهد التغذية بشكل بطيء (المثال النموذجي لهذه الحالة انخفاض جهد البطارية ، أو عند وصل حمل يستجر تيار كبير) الى قيمة محددة تدعى قيمة العتبة فإن دارة POR تقوم بتصفير الشريحة .

1.5. مؤقت المراقبة (Watchdog timer) :

تحوي عائلة PIC18F على مؤقت مراقبة يدعى WDT وهو يعمل عن طريق هزاز خاص RC منفصل عن الهزاز RC الداخلي الموصول مع OSC1 ، لذلك فإن هذا الهزاز سيعمل حتى في نمط النوم حيث يقوم هذا المؤقت بإيقاظ المتحكم ، نتحكم بهذا المؤقت عن طريق البت SWDTEN من المسجل WDTCON ، حيث لتمكين هذا المؤقت فإننا نضع SWDEN =1 ولحجب التمكين SWDEN=0 ، هناك أيضا مقسم سوف يفيد في مضاعفة زمن خرج المؤقت هذا المقسم يتم التحكم بعمله عن طريق مسجل GONFG2H وبشكل نموذجي فإن زمن خرج المؤقت هو 18ms من أجل نسبة تقسيم 1 .

في حال تم تمكين هذا المؤقت فإنه في كل مرة سيحاول أن يقوم بالعد حتى يصل الى قيمته النهائية (100%) وفي حال وصل اليها سيتم توليد اشارة تصفير ستؤدي الى بدء تنفيذ البرنامج من التعليمة الأولى ، الفكرة هي تفادي حدوث ذلك عن طريق تعليمة خاصة تقوم بتصفير المؤقت قبل أن يصل الى قيمته النهائية ، اذا علينا أن نصع أوامر تقوم بتصفير هذا المؤقت في الأماكن المناسبة من البرنامج وبالتالي فإن المؤقت لن يصل أبدا الى القيمة النهائية ولن يتم توليد اشارة تصفير للمتحكم .

الفائدة اذا من هذا المؤقت أنه ولسبب من الأسباب (نبضة كهربائية) أدت الى أن عداد البرنامج دخل في حلقة مفرغة لانهائية فإن التعليمة الخاصة بتصفير المؤقت لن تنفذ وبالتالي بعد مدة معينة سيصل المؤقت الى قيمته النهائية وسيقوم بتصفير المتحكم والعودة الى تنفيذ البرنامج بشكل طبيعي .



الشكل 11

1.6. مصادر الساعة (Clock Sources) :

حتى يقوم المتحكم بعمله فإنه بحاجة الى نبضات الساعة التي يمكن أن يكون مصدرها خارجي (كريستالة ، هزاز سيراميكي ، شبكة RC) أو داخلي (الهزاز الداخلي) وهي تزود المتحكم بالنبضات كمثابة القلب من الإنسان .

الهزاز الخارجي External Oscillator

وهذا الهزاز مبني داخل المتحكم وله قطبين يظهران للعالم الخارجي هما OSC1,OSC2 وهو يدعى هزاز خارجي رغم كونه مبنيا داخل المتحكم لأنه يعتمد على دارة خارجية تزوده بالنبضات عن طريق القطبين OSC1,OSC2 كما قلنا . الدارة الخارجية يمكن أن تكون كريستالة أو هزاز سيراميكي أو شبكة RC أو أي دارة أخرى حيث يمكن أن تكون متكم آخر .

الهزاز الخارجي يمكن أن يهيئ ليعمل مع عدة أنماط وهذا ما يمكن من تشغيله عند سرعات مختلفة باستخدام عناصر مختلفة لتوليد الترددات ، نمط التشغيل نستطيع تحديده عند كتابة البرنامج وذلك عن طريق برنامج PICFlash حيث من حقل Oscillator نختار النمط من القائمة المنسدلة وسيتم تلقائيا وضع البتات المقابلة بحسب الاختيار في البرنامج وستصبح جزء من عدة بايتات والتي تؤلف مع بعضها Configword حيث ستكتب مع البرنامج لاحقا الى ذاكرة المتحكم .

أثناء عملية البرمجة فإن هذه البايتات التي تدعى Configword ستكتب في ذاكرة البرنامج وتخزن في مسجلات خاصة غير متاحة للاستخدام أو التعديل وعلى هذا الأساس فإن المتحكم يعرف ما عليه فعله .

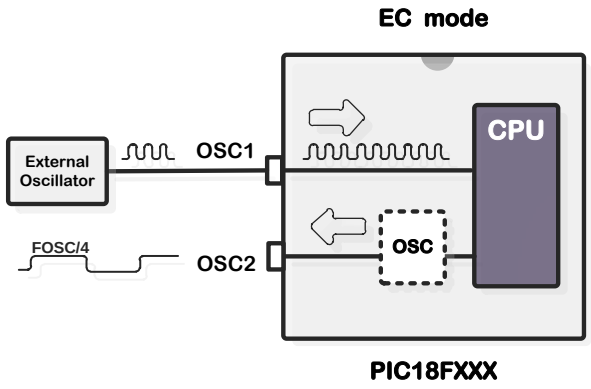
1.6.1. الهزاز الخارجي في النمط EC

في النمط EC فإن الهزاز الخارجي سيستخدم مصدرا خارجيا من أجل الحصول على نبضات الساعة حيث التردد الأعظمي في هذا النمط هو 20Mhz .

فوائد استخدام النمط EC

مصدر الساعة الخارجي سيتم وصله مع الرجل OSC1 وبالتالي فإن OSC2 ستقوم بإخراج نبضات FOSC/4 (أي كل 4 نبضات دخل على الرجل OSC1 سيتم إخراج نبضة على الرجل OSC2) هذه النبضات تستخدم لأغراض الفحص أو تستخدم كتردد ساعة لتجهيزات أخرى .

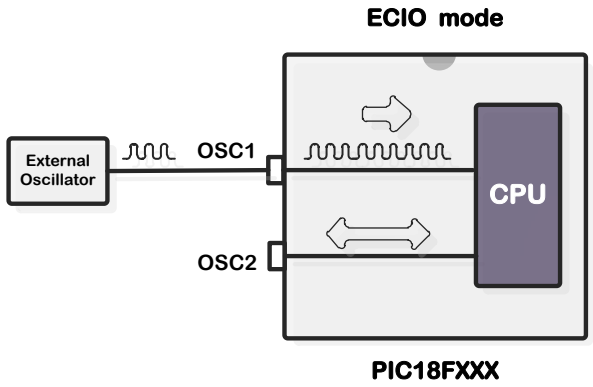
في هذا النمط فإن المتحكم سيبدأ عمله مباشرة بعد وصول التغذية ولن يكون هناك أي تأخير (من أجل استقرار التردد) .



الشكل 12

1.6.2. الهزاز الخارجي في النمط ECIO

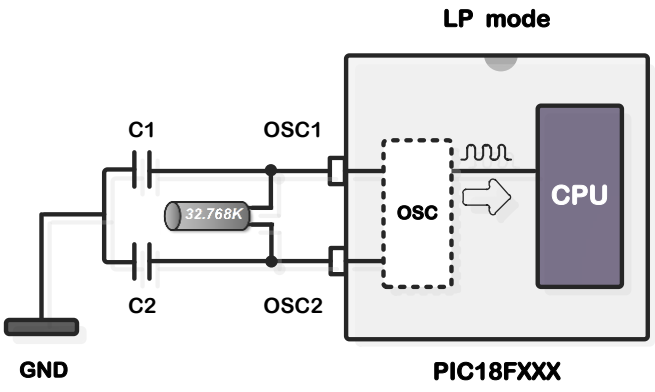
مشابه تماما للنمط EC باستثناء أن الرجل OSC2 يمكن أن تستخدم لوظائف الإدخال / الإخراج (سيكون لدينا رجل متاحة اسمها RA6).



الشكل 13

1.6.3. النمط LP :

(Low Power) ويستخدم فقط مع كريستالة الكوارتز ذات التردد المنخفض (32.768Khz) والمستخدمة عادة في الساعات ، حيث من السهل جدا تمييز هذا النوع فهو اسطواني الشكل ذو حجم صغير أما من ناحية استهلاك الطاقة فهو أقل من النمطين XT,HS



الشكل 14

1.6.4. النمط XT :

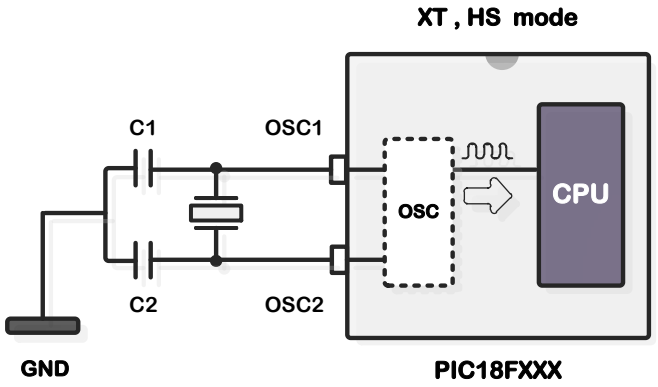
ويستخدم من أجل كريستالات الكوارتز ذات التردد المتوسط حتى 8Mhz وأما التيار المستهلك فهو الأوسط بين الأنماط الثلاثة (LP,XT,HS) .

ويستخدم مع الهزاز السيراميكي (مشابه للكريستالة ويتم توصيله الى المتحكم بنفس الطريقة ومختلف عن كريستال الكوارتز في أنه أرخص ثمنا واهتزازه أردئ بعض الشيء)

1.6.5. النمط HS :

نمط السرعة العالية يستخدم من أجل الكريستالات ذات التردد فوق 8Mhz و التيار المستهلك هو الأكبر بين الأنماط الثلاثة (LP,XT,HS).

ويستخدم مع الهزاز السيراميكي أيضا .

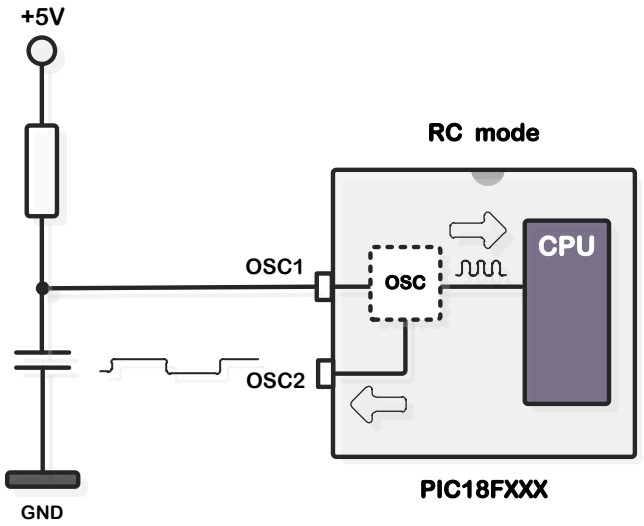


الشكل 15

هناك بالطبع العديد من فوائد استخدام العناصر ذات التردد المستقر مثل الكريستالات ولكنن في بعض الأحيان يمكن
أن يكون هذا غير ضروري . في مثل هذه الحالة فإن أرخص و أبسط حل هو استخدام شبكة RC .

1.6.6. النمط RC :

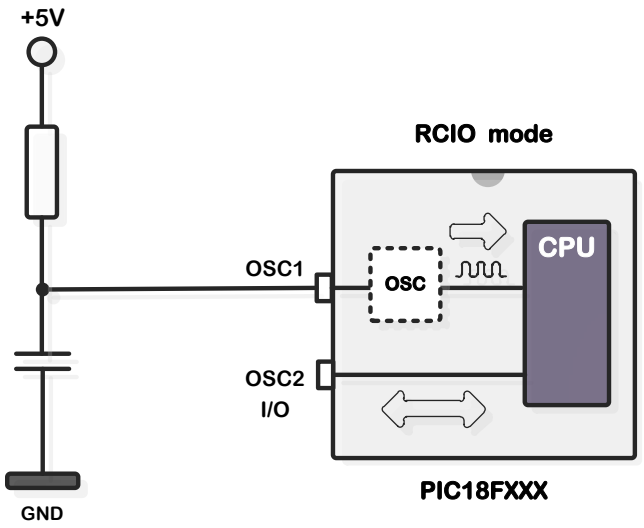
عند تهيئة الهزاز الخارجي ليعمل على النمط RC فإن الرجل OSC1 يجب أن يتم وصلها الى شبكة RC كما هو واضح بالشكل أما الرجل OSC2 فهي عبار عن خرج الهزاز لكن نسبة التقسيم هي 4 ويمكن أن تستخدم هذه النبضات للمزامنة أو الضبط أو أي شيء آخر .



الشكل 16

1.6.7. النمط RCIO :

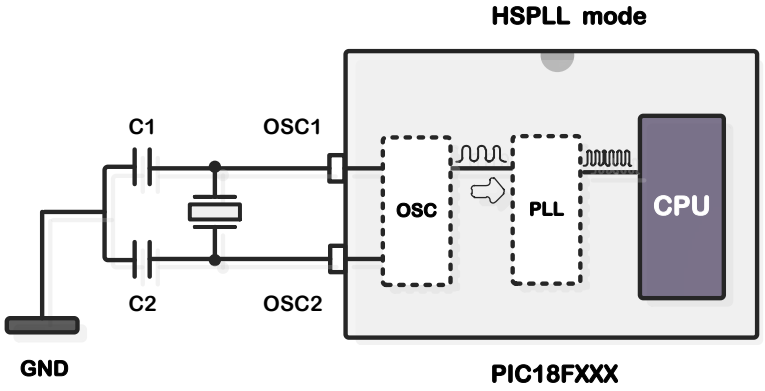
نفس الحالة السابقة نستخدم مع هذا النمط شبكة RC يتم وصلها الى الرجل OSC1 لكن الرجل OSC2 نستطيع استخدامها لوظائف الإدخال / الإخراج .



الشكل 17

1.6.8. النمط HSPLL :

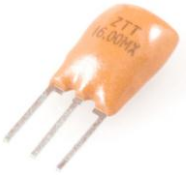
ويستخدم هذا النمط مع الكريستالة أو مه الهزاز السيراميكي ، هناك دارة تدعى PLL مبنية داخل المتحكم يمكن أن تستخدم لمضاعفة التردد بمقدار 4 ، على سبيل المثال من أجل كريستالة ذات تردد 10Mhz ، سيضرب ب4 وسيصبح تردد تنفيذ العمليات هو 40MHz ، وهذا النمط مهم جدا في تحقيق الاتصال مع المنفذ USB.



الشكل 18

الهزاز الداخلي : بعض المتحكمات من عائلة PIC18F لديها نمط للهزاز الداخلي (بالرغم من أن PIC18F452 ليس لديه) حيث في هذا النمط فإن القطبين OSC1 و OSC2 متاحة للأغراض العامة I/O (حيث سيكون متاح لدينا RA6 ، RA7) .

أو كـ (FOSC/4 ، RA7) تردد الهزاز الداخلي يمكن أن يكون من 32Khz حتى 8Mhz ونستطيع الاختيار بين الترددات عن طريق المسجل OSCCON والمسجل OSCTUNE والشكل التالي يظهر بتات الخاصة بمسجلات التحكم بالهزاز الداخلي .

مصدر نبضات الساعة	الأنماط التي يعمل معها	ميزات
External Oscillator	**EC mode** **ECIO mode**	
32.768 khz crystal quartz	**LP mode**	
crystal quartz	**XT mode** **HS mode**	
 **Ceramic Resonator**	**XT mode** **HS mode**	

1.7. أنواع الذواكر (MEMORY):

1.7.1 Flash Memory

هذه الذاكرة تم اختراعها في الثمانينيات في مختبرات INTEL وأصبحت خليفة UV EPROM ،عمليا محتويات هذه الذاكرة يمكن مسحها وإعادة كتابتها عددا غير محدد من المرات ، المتحكمات مع هذه الذاكرة العظيمة مناسبة جدا للأغراض التعليمية والتجريب الكثير ، وبسبب انتشارها الرهيب فإن المتحكمات تستخدمها اليوم ، اذا أردت شراء متحكم فإنك بعد هذا الشرح ستنظر الى نوعية ذاكرته . ملاحظة : كل المتحكمات التي نتعامل معها في هذا الكتاب ذاكرتها من النوع flash و من اسم المتحكم يمكنك معرفة نوعية الذاكرة كما سيأتي لاحقا .

1.7.2 RANDOM ACCESS MEMORY (RAM)

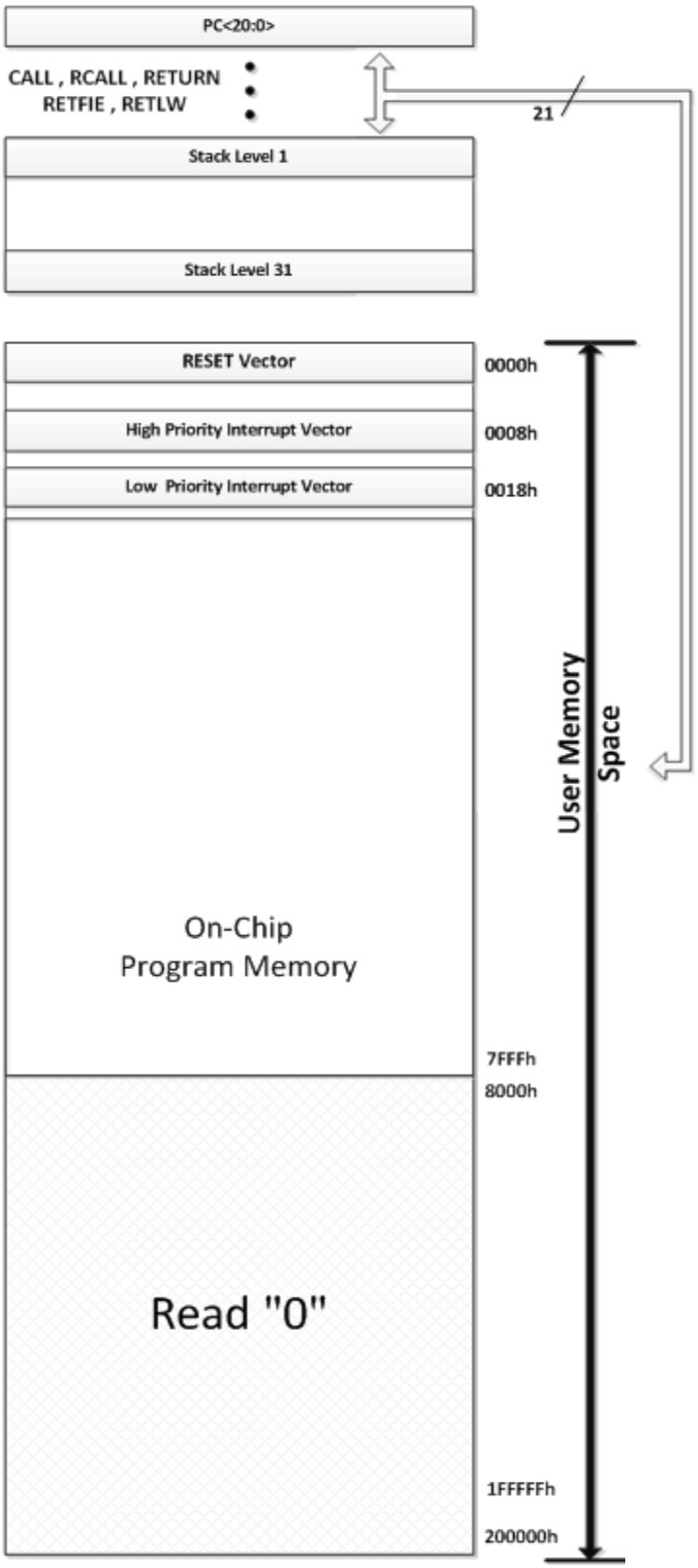
الذي نعرفه عن هذه الذاكرة أنه بمجرد أن التغذية قطعت فإن محتوياتها ستفقد ، لذلك ستستخدم من أجل التخزين المؤقت للبينات ،والبيانات الوسيطة مثلا اذا كان البرنامج ينفذ عملية جمع (أو أيا كان) بالإضافة لعملية طرح فإن ناتج العملية الأولى ستخزن مبدئيا في مسجلات الـRAM لاستخدامها مع العملية الأخرى .

1.7.3 ELECTRICALLY ERASABLE PROGRAMMABLE ROM (EEPROM)

محتويات الـEEPROM يمكن أن تغيير محتوياتها خلال تنفيذ البرنامج (مثل ال_RAM)، لكن تذكر أنها تحتفظ بمحتوياتها بعد انقطاع التغذية عنها (مثل الـ ROM) ، وهكذا فإن الذاكرة EEPROM تستخدم لتخزين القيم التي يجب أن تبقى محفوظة ، مثلا اذا قمت بتصميم قفل سري اذا مكنت المستخدم من ادخال كلمة السر ، ولكن لسوء الحظ اذا لم تقم بحفظ هذه الكلمة في الـEEPROM فإن كلمة السر ستضيع عند انقطاع التغذية

1.8. تنظيم ذاكرة البرنامج (Program Memory Organization) :

يبين الشكل مخطط ذاكرة البرنامج . جميع أفراد عائلة PIC18 لديهم عداد برنامج 21-bit يكفي لعنونة 2Mbyre من ذاكرة البرنامج . مساحة الذاكرة المستخدمة في المتحكم PIC18F452 هي من العنوان 00000H الى العنوان 7FFFH في حال محاولة الوصول وقراءة العناوين عير مستخدمة (غير الموجودة أصلا) سيكون نتيجة القراءة أصفار ، شعاع الترسيت أي نقطة بدء البرنامج بعد الترسيت هو عند العنوان 0000 . العنوان 0008H و العنوان 0018H محجوزان لشعاع مقاطعات الأولوية العالية ولشعاع مقاطعات الأولوية المنخفضة على الترتيب ، حيث أن برنامج خدمة المقاطعة يجب أن يكتب ليبدأ من أحد هذين الموقعين .متحكمات PIC18F لديها مكدس بعمق 31 مستوى يستخدم من أجل حفظ عناوين البرامج الفرعية أو برامج خدمة المقاطعة .علما أن المكدس ليس جزأ من ذاكرة البرنامج أو ذاكرة المعطيات . يتم التحكم بالمكدس عن طريق مؤشر المكدس 5-bit ويهيئ بعد الترسيت بالقيمة 00000 وعند استدعاء برنامج فرعي أو مقاطعة فإن مؤشر المكدس سوف يزداد



الشكل 19

1.9. تنظيم ذاكرة المعطيات (Data Memory Organization) :

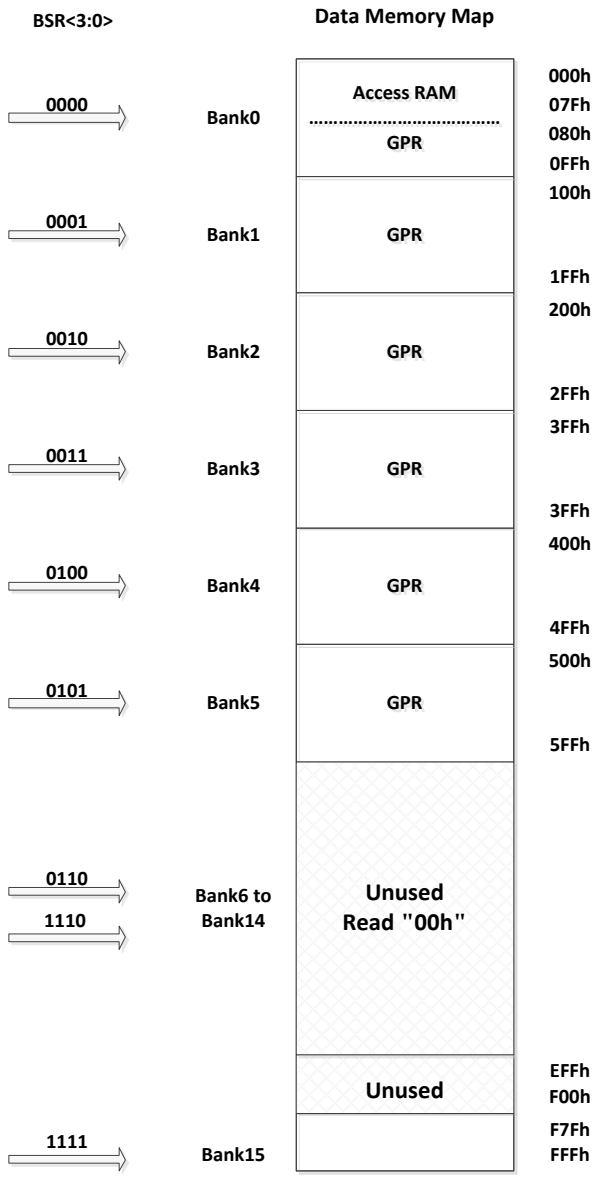
يظهر الشكل التالي مخطط ذاكرة المعطيات للمتحكم PIC18F452 . عرض ممر ذاكرة المعطيات 12-bit وهو قادر على عنونة ذاكرة تصل الى 4kbytes .تتألف الذاكرة من 16 بنك كل بنك بحجم 256 byte ، وفي هذا المتحكم فقط 6 بنوك متاحة ، لذلك فإن ذاكرة المعطيات للمتحكم PIC18f452 هي بحجك 1536bytes (6banks x 256 bytes) . التبديل بين البنوك يحدث بشكل اوتوماتيكي وذلك عند استخدام مترجم ذا لغة عالية المستوى ، لذلك ليس علينا أن نقلق حول كيفية الاختيار أو التبديل بين البنوك أثناء البرمجة

مسجلات الوظائف الخاصة SFR REGISTERS : كل متحكم لديه عددا من مسجلات الـ (SFR) والتي يتم تحديد وظائفها من قبل المصنع ، بايتاتها موصولة الى دارات داخلية في المايكرو مثل المؤقتات ، المبدل A/D ، الهزاز وغير ذلك ، هذا يعني أنها وبشكل مباشر تتحكم بالعمليات لهذه الدارات (اوامر لهذه الدارات).

يمكن تخيل عملها كمفاتيح تتحكم بالعمليات لدارة صغيرة داخل المتحكم ، المسجلات الخاصة هي تعمل تماما نفس عمل المفاتيح

بكلمات أخرى ، حالة بايتات المسجل تتغير أثناء تنفيذ البرنامج ، فتشغل دارات صغيرة في المتحكم ، هذه الدارات ومن خلال أرجل المتحكم تتحكم بالمحيطيات المستخدمة.

هذه المسجلات تحتل الجزء العلوي من البنك 0


BSR<3:0>
Data Memory Map
0000
Bank0
Access RAM
GPR
000h
07Fh
080h
0FFh
0001
Bank1
GPR
100h
1FFh
0010
Bank2
GPR
200h
2FFh
0011
Bank3
GPR
3FFh
3FFh
0100
Bank4
GPR
400h
4FFh
0101
Bank5
GPR
500h
5FFh
0110
1110
Bank6 to
Bank14
Unused
Read "00h"
Unused
EFFh
F00h
1111
Bank15
F7Fh
FFFh

الشكل 20

1.10.مسجلات التهيئة (Configuration Registers):

وهي عبارة عن مسجلات تأخذ قيمها عند برمجة ذاكرة البرنامج Flash عن طريق المبرمجة هذه المسجلات وهذه البتات لا يستطيع البرنامج تغيير قيمها مبينة في الجدول التالي :

Value	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	File Name	
--1- -111	FOSCO	FOSCI	FOSC2	-	-	OSCSEN	-	-	CONFIG1H	300001h
1111	PWRTEN	BOREN	BORVO	BORVI	-	-	-	-	CONFIG2L	300002h
---- 1111	WDTEN	WDTPSO	WDTPSI	WDTPS2	-	-	-	-	CONFIG2H	300003h
---- ---1	CCP2MX	-	-	-	-	-	-	-	CONFIG3H	30000Sh
1--- -1-1	STVREN	-	LVP	-	-	-	-	DEBUG	CONFIG4L	300006h
---- 1111	CPO	CPI	CP2	CP3	-	-	-	-	CONFIGSL	300008h
11-- ----	-	-	-	-	-	-	CPB	CPD	CONFIGSH	300009h
---- 1111	WRTO	WRTI	WRT2	WRT3	-	-	-	-	CONFIG6L	30000Ah
111- ----	-	-	-	-	-	WRTC	WRTB	WRTD	CONFIG6H	30000Bh
---- 1111	EBTRO	EBTRI	EBTR2	EBTR3	-	-	-	-	CONFIG7L	30000Ch
-1-- ----	-	-	-	-	-	-	EBTRB	-	CONFIGlH	30000Dh
(1)	REVO	REVI	REV2	REV3	REV4	DEVO	DEVI	DEV2	DEVIDI	3FFFFEh
00000100	DEV3	DEV4	DEVS	DEV6	DEV7	DEV8	DEV9	DEV10	DEVID2	3FFFFFh

الجدول 3

PIC18F452 configuration registers

الفصل الثالث

المحيطيات

1.1. نوافذ الدخل / الخرج (I/O Ports) :

لجعل المتحكم مفيدا ، يجب وصله مع عناصر الكترونية اضافية مثل الليدات ، المحركات ، المفاتيح .. جميع المتحكمات لديه مسجل أو أكثر يدعى بورت port متصل مع أرجل المتحكم .

لماذا الدخل / الخرج ؟ لأنه يمكنك تغيير وظيفة الرجل حيث يمكن أن تجعلها دخل أو تجعلها خرج وذلك حسب رغبتك ويتم ذلك برمجيا . مثلا تخيل أننا نقيادة ثلاث ليدات (on/off) حيث يضيئ هذه الليدات عند الضغط على مفتاح لحظي ، لذلك بعض الأرجل يجب أن تهيئ كخرج (لليدات) ورجل يجب أن تُهيئ دخل (الموصولة مع مفتاح لحظي) . يتم ذلك بشكل بسيط برمجيا ويمكن تبديل عمل الرجل من دخل الى خرج أو العكس في سياق التنفيذ .

احدى النقاط الهامة في input/output (I/O) هو التيار الأعظمي المستجر من المتحكم . بشكل عام التيار يكفي لتشغيل ليد أو تجهيزة تستهلك تيار منخفض من (10–20 mA) .

احدى النقاط الهامة هي أن الرجل يمكن أن يكون لديها مقاومة شد . هذه المقاومات تقوم بوصل الرجل الى الجهد الموجب للتغذية عن طريقها وذلك عند تهيئة الرجل كدخل أي عند وصلها الى مفتاح لحظي مثلا .

يتم التحكم بنوافذ المتحكم عن طريق مسجلات خاصة ،وهذا يعني أن كل بت من هذه المسجلات تحدد حالة رجل للمتحكم عند كتابة المنطق (1) في المسجل الخاص TRIS فإن رجل النافذة المقابلة ستهيئ كدخل وعندها يمكننا قراءة حالة الرجل التي يمكن أن تأخذ 0 أو 1 من مسجل PORT

الحالة الأخرى عند كتابة المنطق (0) في المسجل الخاص (TRIS) فإن رجل البورت المقابلة ستهيئ كخرج وسيكون جهد الخرج (0V or 5V) وذلك حسب القيمة التي سنضعها في المسجل port

لكن جميع المتحكمات على الأقل لديها نافذتين هما PORTA و PORTB ، يتم تسمية أرجل أي نافذة حسب الصيغة التالية RPn حيث P تدل على اسم النافذة ، n : رقم البت في النافذة ، مثلا أرجل النافذة PORTA تسمى من RA0 حتى RA7 . وأرجل النافذة PORTB تسمى RB0 حتى RB7 . وهكذا

للعمل مع النوافذ ربما سنتعامل مع العمليات التالية :

تحديد اتجاه النافذة (هل هي دخل أم خرج)

وضع قيمة الخرج

قراءة قيمة الدخل

وضع قيمة على الخرج ثم قراءة قيمة الخرج

العمليات الثلاثة الأولى موجودة للمتحكمات من عائلة PIC16 وعائلة PIC18 ، في بعض التطبيقات ربما نحتاج أن نرسل قيمة الى نافذة ما ومن ثم قراءة القيمة المخرجة (المرسلة) على النافذة ، بالنسبة لعائلة PIC16 لديه ضعف تصميمي من هذه الناحية حيث القيمة المقروءة من نافذة ما ربما تختلف عن القيمة المرسلة لها . بسبب أن عملية القراءة تقرأ القيمة الحقيقية على الرجل وهذه القيمة يمكن أن تتغير لسبب ما (تجهيزة خارجية موصولة مع هذه الرجل) . في عائلة PIC18F هناك مسجل مسك يدعى LATA بالنسبة للPORTA مهمته مسك القيمة المرسلة الى أرجل النافذة . قراءة قيمة نافذة ما PORT ستقرأ من مسجل المسك وهذه القيمة لن تتأثر بالتجهيزات الخارجية .

PORTA : في المتحكم PIC18F452 فإن البورت A يكون بعرض 7_Bits ولكل رجل من هذا البورت عدد من المهام بالإضافة الى الإدخال و الإخراج و الجدول التالي يبين وظيفة كل رجل من هذا البورت .

Pin	Description
RA0/AN0	
RA0	Digital I/O
AN0	Analog input 0
RA1/AN1	
RA1	Digital I/O
AN1	Analog input 1
RA2/AN2/VREF-	
RA2	Digital I/O
AN2	Analog input 2
VREF-	A/D reference voltage (low) input
RA3/AN3/VREF+	
RA3	Digital I/O
AN3	Analog input 3
VREF+	A/D reference voltage (high) input
RA4/T0CKI	
RA4	Digital I/O
T0CKI	Timer 0 external clock input
RA5/AN4/SS/LVDIN	
RA5	Digital I/O
AN4	Analog input 4
SS	SPI Slave Select input
RA6	Digital I/O

الجدول 4

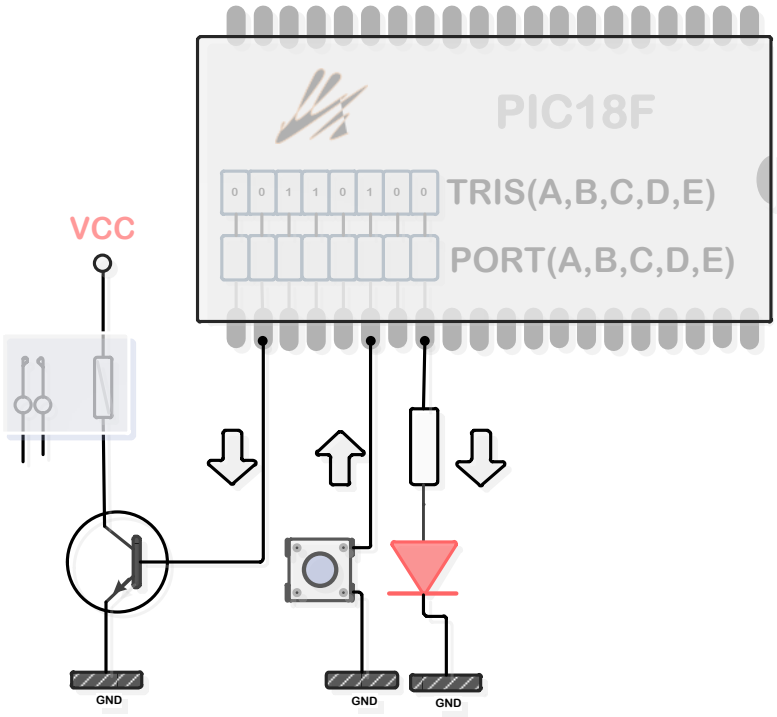
PIC18F452 PORTA pin functions

اذا النافذة PORTA لها ثلاثة مسجلات هي :

وهناك ثلاث مسجلات تتعلق بالنافذة PORTA A وهي

- Port data register—PORTA مسجل المعطيات
- Port direction register—TRISA مسجل الاتجاه
- Port latch register—LATA مسجل المسك .

المسجل PORTA هو مسجل المعطيات ، اي أن اي قيمة سيتم اسنادها الى المسجل ستنعكس على حالة الأرجل . المسجل TRISA وهو يحدد اتجاه ارجل البورت A هل هي دخل أم خرج ، فعند وضع واحد في بت ما من هذا المسجل فإن الرجل المقابلة له ستكون دخل وعندما نضع 0 في بت لآخر فإن الرجل المقابلة له ستكون خرج LATA هو مسجل مسك الخرج وهو يحوي نفس معطيات المسجل PORTA و الكتابة الى اي منهما تعني الكتابة الى الأخر . لكن القراءة من LATA لا تعبر عن القيمة الحقيقية للأرجل وإنما القيمة المرسلة .



الشكل 21

البتات من 0 ، 1 ، 2 ، 3 ، 5 من هذه النافذة تستخدم أيضاً من أجل الدخل التشابهي . بعد تصفير المتحكم ، هذه الأرجل مبرمجة كمداخل تشابهية أما RA4 ، RA6 مُهيئة كمداخل رقمية وأما لبرمجة المداخل التشابهية (0 ، 1 ، 2 ، 3 ، 5) لتصبح رقمية فإن المسجل المسؤول عن ذلك هو المسجل ADCON1 . كتابة القيمة 7 الى هذا المسجل يجعل كل أرجل هذا البورت رقمية .

الرجل RA4 مرتبطة مع المؤقت 0 (T0CKI) ، RA6 يمكن أن تستخدم من أجل وظائف الإدخال الإخراج العامة . أو كمدخل للساعة OSC2 ، أو كخرج للساعة FOSC/4 .

PORTB : الوظائف العامة لهذه النافذة مشابهة لوظائف PORTA . كل رجل من هذا البورت لديه مقاومة شد داخلية والتي يمكن أن تُمكّن عن طريق البت RBPU من المسجل INTCON2 ، مقاومات الشد هذه سوف يُلغى تمكينها في حالة تهيئة أرجل البورت كمخارج .مقاومات الشد ستسمح بوصل (تجهيزات دخل : مثل المفاتيح و الأزرار اللحظية) والتي ستوصل الى هذا البورت بدون مقاومات شد خارجية وهذا يقلل الكلفة لأن كمية العناصر والأسلاك المطلوبة تقل .

أرجل البورت من RB4 حتى RB7 يمكن أن تستخدم من أجل (مقاطعة تغير الدخل على أحد الأقطاب من RB4 حتى RB7) وأن أي تغير في الدخل على أي رجل من 4 حتى 7 ستسبب مقاطعة وسيفعّل علم لذلك ، من أجل تمكين هذه المقاطعة فإن البت المسؤول عن ذلك هو RBIE أما العلم الذي يدل على حدوث هذه المقاطعة فهو RBIF وهذين البتين موجودين في المسجل INTCON .

PORTC,PORTD, PORTE

بالإضافة الى البورت PORTA و البورت PORTB فإن المتحكم PIC18F452 لديه النافذة PORTC ، والنافذة PORTD وكل منهما بعرض 8-bit ، كذلك يحوي على النافذة PORTE وهي بعرض 3-bit وكل نافذة لديها مسجل المعطيات الخاص بها (مثلا النافذة PORTC فإن مسجل المعطيات الخاص اسمه TRISC ومسجل المسك اسمه LATC) والوظائف الأساسية مشابهة بالنسبة لجميع النوافذ .

Pin	Description
RB0/INT0	
RB0	Digital I/O
INT0	External interrupt 0
RB1/INT1	
RB1	Digital I/O
INT1	External interrupt 1
RB2/INT2	

RB2	Digital I/O
INT2	External interrupt 2
RB3/CCP2	
RB3	Digital I/O
CCP2	Capture 2 input, compare 2, and PWM2 output
RB4	Digital I/O, interrupt on change pin
RB5/PGM	
RB5	Digital I/O, interrupt on change pin
PGM	Low-voltage ICSP programming pin
RB6/PGC	
RB6	Digital I/O, interrupt on change pin
PGC	In-circuit debugger and ICSP programming pin
RB7/PGD	
RB7	Digital I/O, interrupt on change pin
PGD	In-circuit debugger and ICSP programming pin

الجدول 5

PIC18F452 PORTB pin functions

Pin	Description
RC0/T1OSO/T1CKI	
RC0	Digital I/O
T1OSO	Timer 1 oscillator output
T1CKI	Timer 1/Timer 3 external clock input
RC1/T1OSI/CCP2	
RC1	Digital I/O
T1OSI	Timer 1 oscillator input
CCP2	Capture 2 input, Compare 2 and PWM2 output

RC2/CCP1	
RC2	Digital I/O
CCP1	Capture 1 input, Compare 1 and PWM1 output
RC3/SCK/SCL	
RC3	Digital I/O
SCK	Synchronous serial clock input/output for SPI
SCL	Synchronous serial clock input/output for I2C
RC4/SDI/SDA	
RC4	Digital I/O
SDI	SPI data in
SDA	I2C data I/O
RC5/SDO	
RC5	Digital I/O
SDO	SPI data output
RC6/TX/CK	
RC6	Digital I/O
TX	USART transmit pin
CK	USART synchronous clock pin
RC7/RX/DT	
RC7	Digital I/O
RX	USART receive pin
DT	USART synchronous data pin

الجدول 6

PIC18F452 PORTC pin functions

Pin	Description
RD0/PSP0	
RD0	Digital I/O
PSP0	Parallel slave port bit 0
RD1/PSP1	
RD1	Digital I/O
PSP1	Parallel slave port bit 1
RD2/PSP2	
RD2	Digital I/O
PSP2	Parallel slave port bit 2
RD3/PSP3	
RD3	Digital I/O
PSP3	Parallel slave port bit 3
RD4/PSP4	
RD4	Digital I/O
PSP4	Parallel slave port bit 4
RD5/PSP5	
RD5	Digital I/O
PSP5	Parallel slave port bit 5
RD6/PSP6	
RD6	Digital I/O
PSP6	Parallel slave port bit 6
RD7/PSP7	
RD7	Digital I/O
PSP7	Parallel slave port bit 7

الجدول 7

PIC18F452 PORTD pin functions

Pin	Description
RE0/RD/AN5	
RE0	Digital I/O
RD	Parallel slave port read control pin
AN5	Analog input 5
/RE1/WR AN6	
RE1	Digital I/O
WR	Parallel slave port write control pin
AN6	Analog input 6
RE2/CS/AN7	
RE2	Digital I/O
CS	Parallel slave port CS
AN7	Analog input 7

الجدول 8

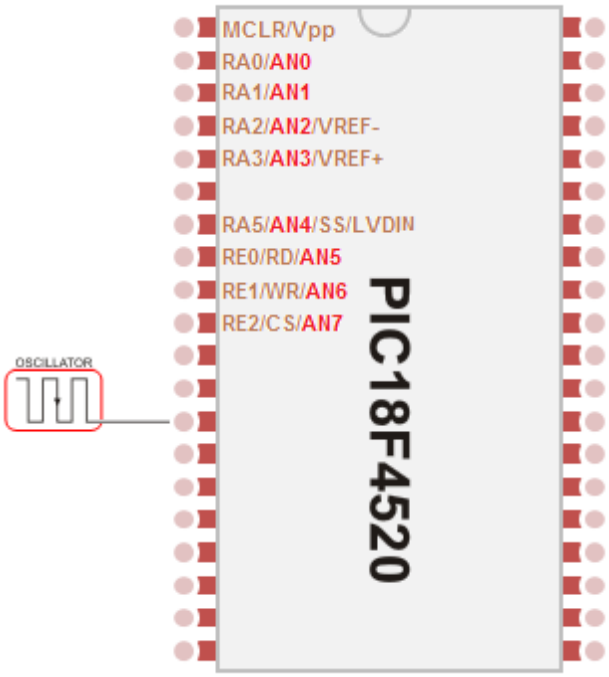
PIC18F452 PORTE pin functions

1.2. المحول التشابهي الرقمي ADC :

هو عبارة عن دارة ضمن المتحكم تقوم بتحويل الجهد التشابهي المطبق على ارجل المتحكم الى قيمة رقمية يمكن التعامل معها وتخزينها . ربما يبدو أن اسم المحول التشابهي الرقمي معقد لا أنه بسيط جدا

وعلى الرغم من توفر دارات متكاملة تقوم بعملية التحويل مثل ADC0804 الا ان تضمين مثل هذه الدارة في المتحكم تعتبر ميزة هامة جدا

الشكل 22



TRIS(A,B,C,D,E)							
1	0	1	1	0	0	1	1

الشكل 23

1.2.1. ميزات المحول التشابهي الرقمي ADC :

- نتيجة التحويل تكون بعرض 10 bits يتم تخزينها في المسجلين (ADRESL,ADRESH)
- دقة عملية التحويل يمكن أن تضبط عن طريق الجهد المرجعي

التحكم بعملية التحويل :

هناك أربع مسجلات يمكن من خلالها التحكم بعملية التحويل :

- ADRESH : يحوي عل البايت العلوي للنتيجة
- ADRESL : يحوي على البايت السفلي للنتيجة
- ADCON0 : مسجل التحكم 0
- ADCON1 : مسجل التحكم 1

المسجل ADCON0 يتحكم في موديول A/D مثل اختيار تردد الساعة بالتعاون مع المسجل ADCON1 ، اختيار القناة التشابهية ، بدء التحويل ، تشغيل وايقاف المحول التشابهي .

المسجل ADCON1 يتحكم بكيفية تخزين النتيجة ، اختيار الجهد المرجعي وجعل القناة كدخل تشابهي ،واختيار تردد الساعة بالتعاون مع المسجل ADCON0 .

تبدأ عملية التحويل عند وضع البت GO/DONE من المسجل ADCON0 .وعند انتهاء عملية التحويل سيصفر هذا البت أوتوماتيكيا . وكذلك يمكن تفعيل مقاطعة عن طريق السوفت وير .

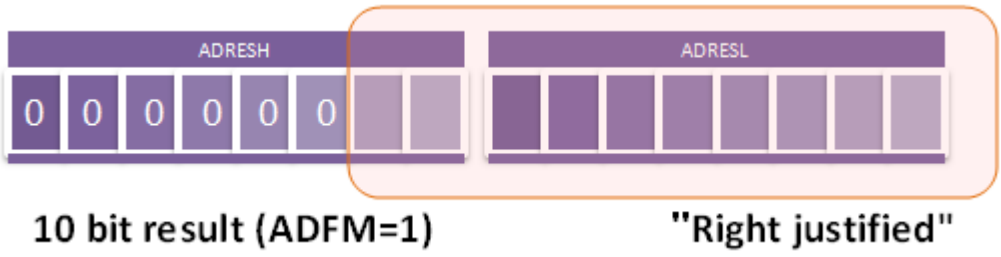
1.2.2. أين يتم تخزين نتيجة التحويل وكيف ذلك ؟

النتيجة بعد عملية التحويل تكون عبارة عن 10 bit ولن يكفي مسجل واحد لتخزينها لأن عرض المسجل 8 بت لذلك نستخدم مسجل آخر نخزن فيه البيتين الباقيين تخزن وهذين المسجلين هما (ADRESH ، ADRESL)

وهناك طريقتين للتخزين :

حالة اليمين : في هذه الحالة فإن المسجل ADRESL سيستخدم بالكامل بالإضافة الى بيتين من المسجل ADRESH

- حالة اليسار : في هذه الحالة فإن المسجل ADRESH سيستخدم بالكامل بالإضافة الى بيتين من المسجل ADRESL



- حالة اليسار : في هذه الحالة فإن المسجل ADRESH سيستخدم بالكامل بالإضافة الى بيتين من المسجل ADRESL

ويتم تحديد حالة التخزين عن طريق البت ADFM من المسجل ADCON1 .

في حال عدم استخدام المحول التشابهي الرقمي فإن هذين المسجلين سيستخدمان للأغراض العامة .

1.2.3. دقة التحويل :

كما قلنا فإن المحول بعرض 10 بت لذلك أكبر قيمة يمكن ان للنتيجة هي 1023

1023=

result							
1	1	1	1	1	1	1	1

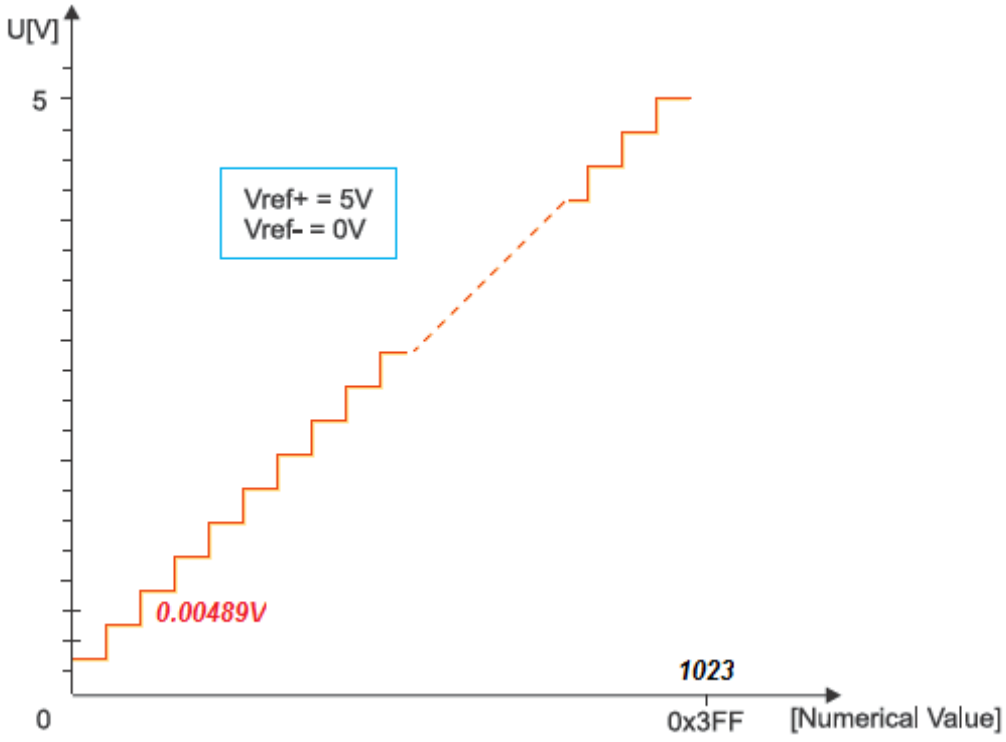
في حال كان الجهد المرجعي +5V فإن خطوة التحويل $\frac{5}{1023} = 0.00489V$

واذا كان جهد الدخل 1V فإن نتيجة التحويل ستكون $\frac{1}{0.00489} = 205$

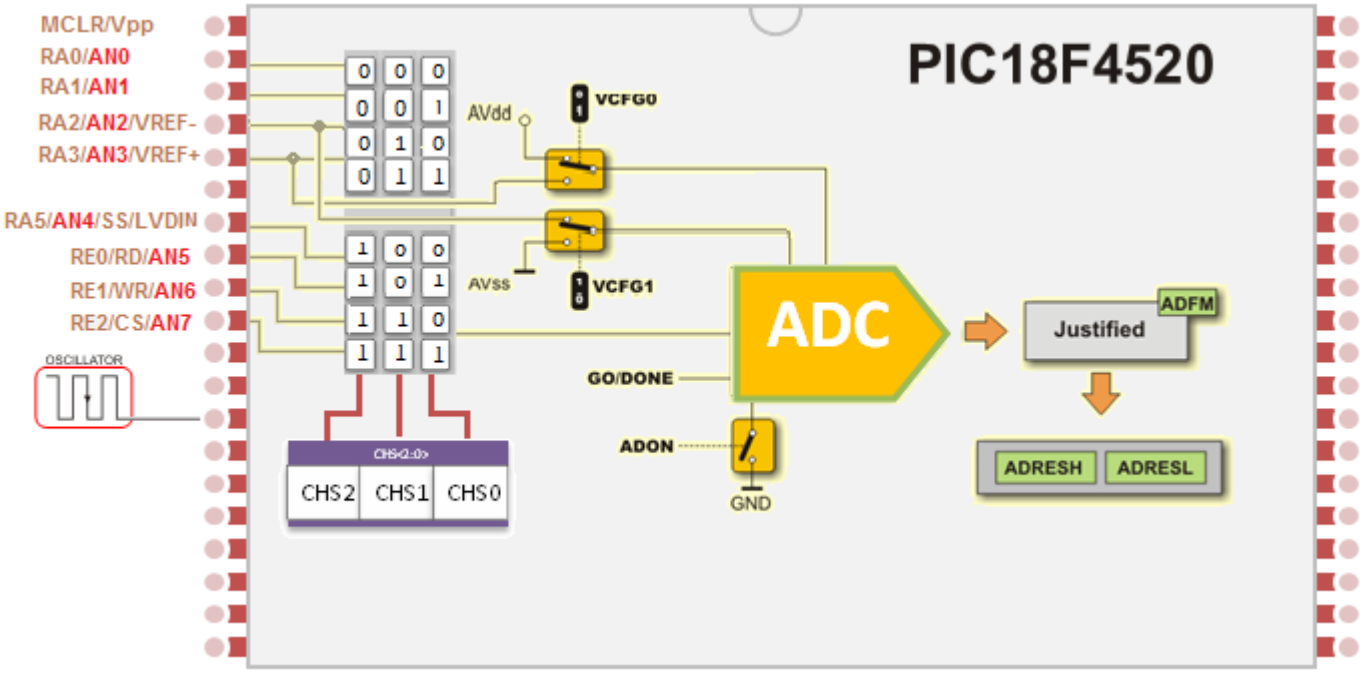
واذا كان جهد الدخل 5V فإن نتيجة التحويل ستكون $\frac{5}{0.00489} = 1023$

واذا كان جهد الدخل 3V فإن نتيجة التحويل ستكون $\frac{3}{0.00489} = 613$

في حال كان الجهد المرجعي +3V فإن خطوة التحويل $\frac{3}{1023} = xxxxV$



الشكل 24



الشكل 25

مسجل التحكم بعمل المحول A/D

ADCON0							
ADCS1	ADCS0	CHS2	CHS2	CHS2	GO/DONE	-----	ADON

اختيار تردد ساعة المحولADCS1, ADCS0 :

<ADCS2> ADCON1	<ADCS1: ADCS0> ADCON0	CLOCK CONVERSION
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	RC *
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	RC *

CHS3-CHS0 : اختيار القناة أو الرجل التي سيطبق الجهد التشابهي عليها .

CHS2	CHS1	CHS0	CHANNEL	PIN
0	0	0	0	RA0/AN0
0	0	1	1	RA1/AN1
0	1	0	2	RA2/AN2
0	1	1	3	RA3/AN3
1	0	0	4	RA5/AN4
1	0	1	5	RE0/AN5
1	1	0	6	RE1/AN6
1	1	1	7	RE2/AN7

ملاحظة : متحكمات PIC18F2X2 لا تملك ثمان قنوات تشابهية

GO/DONE : بت يحدد حالة عملية التحويل

1 : عملية التحويل قيد التنفيذ

0 : انتهاء عملية التحويل (هذا البت يتم تصفيرة أوتوماتيكيا عند انتهاء عملية التحويل)

ADON - A/D On bit : تمكين المحول

1 : تمكين المحول التشابهي الرقمي A/D

0 : الغاء تمكين المحول التشابهي الرقمي A/D (أي أن المحول في حالة OFF ولا يستهلك تيار)

ADCON1							
ADFM	ADCS2	----	----	PCFG3	PCFG2	PCFG1	PCFG0

ADFM : اختيار كيفية تخزين النتيجة

1 : حالة اليمين سيتم قراءة الست بتات الأكثر أهمية كصفر

0 : حالة اليسار سيتم قراءة الست بتات الأقل أهمية كصفر

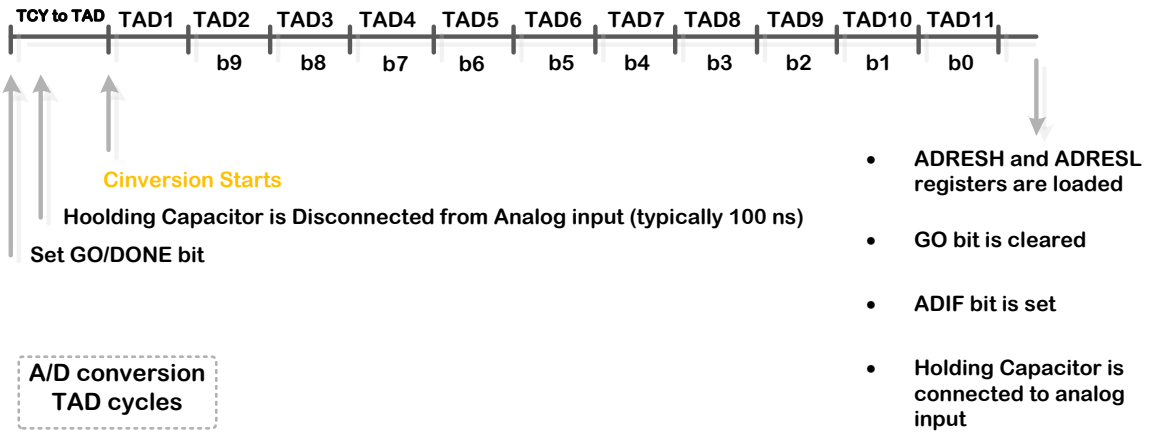
PCFG3:PCFG0 : التحكم بحالة أرجل البورت A,E هل هي تشابهية أم رقمية

PCFG<3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	Vss	**8/0**
0001	A	A	A	A	VREF+	A	A	A	AN3	Vss	**7/1**
0010	D	D	D	A	A	A	A	A	VDD	Vss	**5/0**

0011	D	D	D	A	VREF+	A	A	A	AN3	Vss	**4/1**
0100	D	D	D	D	A	D	A	A	VDD	Vss	**3/0**
0101	D	D	D	D	VREF+	D	A	A	AN3	Vss	**2/1**
011x	D	D	D	D	D	D	D	D	----	---	**0/0**
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	**6/2**
1001	D	D	A	A	A	A	A	A	VDD	Vss	**6/0**
1010	D	D	A	A	VREF+	A	A	A	AN3	Vss	**5/1**
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	**4/2**
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	**3/2**
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	**2/2**
1110	D	D	D	D	D	D	D	A	VDD	VSS	**1/0**
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	**1/2**

- **A: ANALOG INPUT**
- **D: DIGITAL I/O**

من أجل عملية تحويل صحيحة يجب أن نتأكد أن زمن تحويل بت واحد من النتيجة يجب الا يقل عن 1.6U والجدول التالي يعطينا تردد الساعة المطلوبة من أجل عملية التحويل وذلك من أجل ترددات مختلفة للمتحكم .



الشكل 26

حيث الزمن الذي يحتاجه من أجل تحويل بت واحد يعرف كـ TAD وهو يساوي كما قلنا 1.6us أما الزمن الكلي فهو أكبر قليلا من 11TAD

Operation	ADCS2:ADCS0	Maximum microcontroller frequency
2 Tosc	000	1.25 MHz
4 Tosc	100	2.50 MHz
8 Tosc	001	5.0 MHz
16 Tosc	101	10.0 MHz
32 Tosc	010	20.0 MHz
64 Tosc	110	40.0 MHz
RC	011	-

1.2.4. خطوات التحويل :

المحول A/D الموجود في شريحة PIC18F452 لديه ثمانية قنوات تشابهية

نقوم بتهيئة الرجل لتصبح تشابهية واختيار الجهد المرجعي

نقوم بوضع الرجل التي تمثل القناة التشابهية كدخل وذلك من المسجل TRISA أو TRISE .

من المسجل ADCON0 نختار احدى القنوات التشابهية

من المسجل ADCON0 و المسجل ADCON1 نختار تردد ساعة التحويل

من المسجل ADCON0 نشغل موديول المحول (لا يعني بدء عملية التحويل)

نهيئ مقاطعة المحول (اذا اردت ذلك)

وضع الخانة GO/DONE (جعل قيمتها واحد) لبدء عملية التحويل

ننتظر حتى يتم تصفير الخانة GO/DONE أو حتى تتولد مقاطعة

نقرأ نتيجة التحويل من المسجلين ADRESH,ADRESL .

نعيد هذه الخطوات من أجل عملية تحويل جديدة.

1.3. العداد/المؤقت TIMER0 :

المؤقت/العداد :وهو عبارة عن سجل ذو 8 بتات أو 16-bit يقوم بزيادة قيمته عندما يتلقى نبضة (أو عدة نبضات في حال استخدام المقسم) وعدما يصل الى قيمته النهائية فإن مقاطعة يمكن أن تتولد . سنضرب عدة أمثلة عن استخدامات المؤقت /العداد وذلك حتى نبين أهميته !

مثال عن المؤقت : إجرائية معينه (عدة أسطر برمجية) أريد تنفيذها كل خمس ثواني مثلا بشكل منتظم فإنه من المفيد استخدام مقاطعة هذا المؤقت حيث أنه يوقف تنفيذ البرنامج الحالي وينفذ برنامج خدمة المقاطعة ثم يعود بعد أن ينتهي لينفذ البرنامج الرئيسي.

ربما استخدام التأخير الزمني ()delay قد يحل هذه المسألة لكن هناك فرق جوهري بين التأخير واستخدام المؤقت Timer0

التأخير ()delay سوف يشغل المعالج أي أثناء التأخير لا يمكن تنفيذ الا تعليمة التأخير ، مثلا (delay(6sec أي أثناء الست ثواني لا يمكننا تنفيذ اي تعليمة

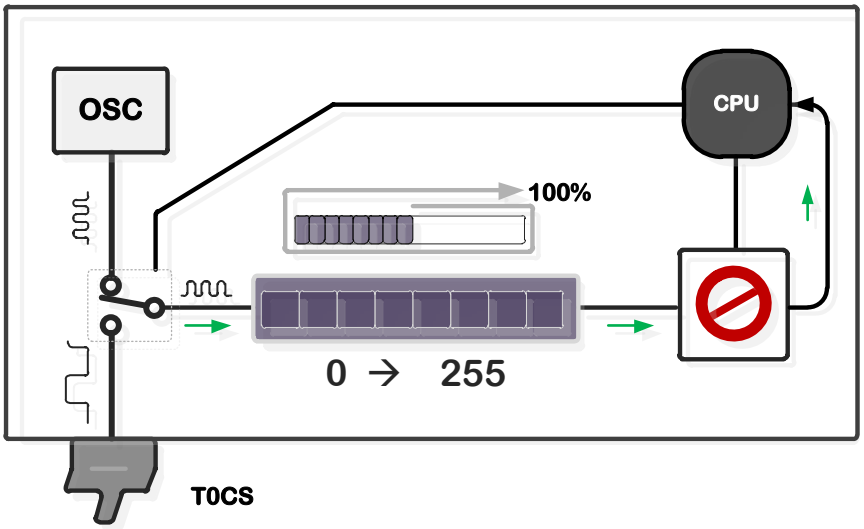
المؤقت Timer0 لا يشغل المعالج وإنما يعمل بشكل مستقل وعند حدوث طفحان يقوم بتنفيذ برنامج المقاطعة وبعد انتهاء تنفيذ برنامج المقاطعة يعود لتنفيذ البرنامج الرئيسي .

التأخير لا يستخدم لتوليد زمن حقيقي

المؤقت Timer0 يمكن أن يستخدم لتوليد زمن حقيقي .

مثال عن العداد : نريد كتابة برنامج يقوم بعد منتج ما حيث يقوم الحساس بتوليد نبضة عند مرور كل عبوة من المنتج الحل الأمثل هو استخدام Timer0 كعداد ، ربما نستطيع كتابة برنامج يقوم باستخدام متغير Cnt++ يزداد عند كل نبضة ، لكن نعيد الفكرة وهي أننا قمنا بشغل المعالج ولن نستطيع تنفيذ أي عمل آخر .

يبين الشكل التالي هذا المؤقت العداد :



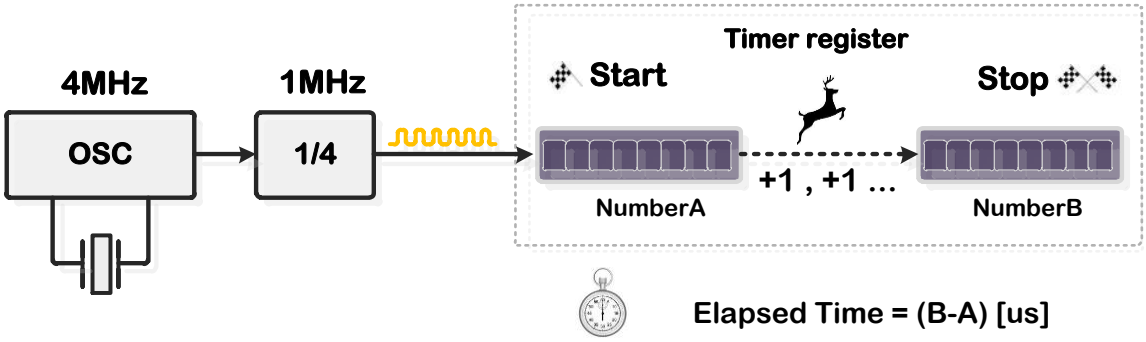
الشكل 27

وهذه النبضات إما أن تكون داخلية نبضات مشتقة من الكريستالة (هنا يعمل كمؤقت)أو خارجية يتلقاها المتحكم عن طريق الرجل عن طريق الرجل RA4 (هنا يعمل كعداد لهذه النبضات الخارجية)

لهذه العملية فإن المتحكم يستخدم كريستالة الكوارتز ، هناك عدة أسباب لهذا الاختيار ، هذا الهزاز دقيق جدا ومستقر ، لذلك فإن النبضات ستتولد دائما بنفس العرض ، والتي تجعل من هذه النبضات مناسبة جدا من أجل عمليات قياس الزمن ، مثل هذه الهزازات تستخدم في ساعات اليد . اذا كنت تريد قياس الزمن بين حادثتين معينتين فيكفيك أن تعد النبضات المولدة بين هاتين الحادثتين . هذا بالضبط مبدأ عمل المؤقت .

اذا استخدم المؤقت هزاز الكوارتز الداخلي فإنه يمكنه قياس الزمن بين حادثتين (اذا كان قيمة المسجل هي T1 لحظة بدء القياس و t2 هي لحظة انتهاء القياس فإن الزمن نحصل عليه عن طريق عملية الطرح T2-T1)
اذا استخدمت المسجلات النبضات من مصدر خارجي فإن المؤقت سيتحول الى عداد

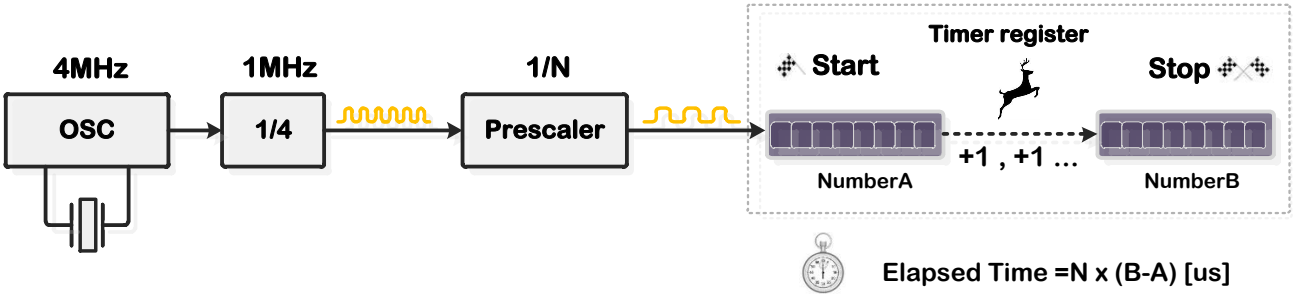
مثال : قيمة المسجل ستزداد مليون مرة بالثانية وذلك من أجل تردد كوارتز 4MHz .



الشكل 28

من السهل قياس فاصل زمني حتى 256 ميكرو ثانية باستخدام الطريقة أعلاه لأنها أكبر قيمة يمكن لمسجل أن يخزنها هذا التقييد يمكننا التغلب عليه بسهولة بعدة طرق ، مثلا استخدام هزاز أبطأ ، مسجل بعرض أكبر ، عن طريق المقسم أو المقاطعات ، الطريقتين الأوليتين ضعيفتين وغير مستخدمتين عمليا لذلك سنستخدم المقسم أو المقاطعات

استخدام المقسم مع المؤقت :المقسم هو دارة الكترونية تستخدم لتقليل التردد بعامل محدد ، المقسم يقول من أجل أن أولد نبضة في الخرج فإنه يجب أن تتولد 1,2,4 أو أكثر على الدخل معظم المتحكمات لديها مقسم أو أكثر مبنية داخليا ، نسبة التقسيم يمكن تغييرها برمجيا . المقسم يتم استخدامه عندما نريد قياس زمن طويل ، اذا تمت مشاركة المقسم بين المؤقت و مؤقت watchdog فإنه لا يمكن استخدامه بنفس الوقت .



الشكل 29

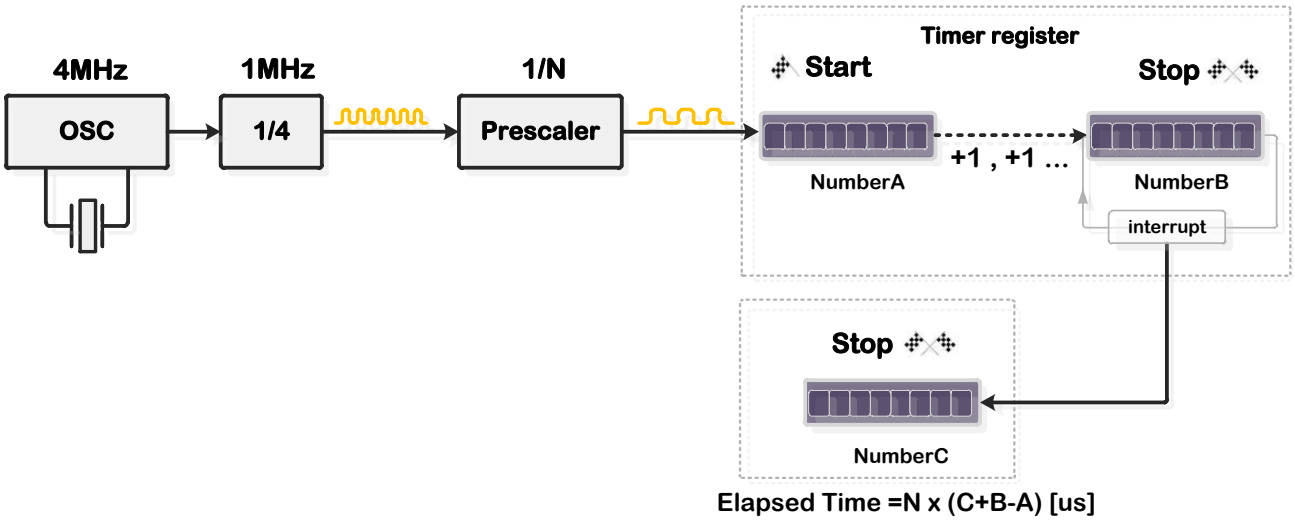
اذا كان المسجل يتألف من ثماني بتات أكبر رقم يمكن تخزينه هو 255 ، أما بالنسبة لمسجل بعرض 16 بت فإن أكبر رقم يمكن تخزينه 65.535

وفي حال تم تجاوز هذا الرقم فإنه سيتم تصفير المؤقت بشكل اتوماتيكي وسيبدأ العد من جديد من الصفر ، هذه الحالة تدعى الطفحان overflow .

في حال تمكين المقاطعة عن طريق البرنامج ، فإن الطفحان يمكن أن يولد مقاطعة والتي تمكننا من فعل اشياء جديدة

مثلا : يمكننا حساب الزمن لثواني أو دقائق أو حتى أيام

أهم مافي الأمر أن المؤقت يعمل أثناء العمل الطبيعي للمتحكم بشكل مستقل لكنه ينبهه عند حدوث مقاطعة ، ويمكنه من قراءة قيمته أو تصفيرها أو تحميل قيمة ما اليه



الشكل 30

المسجل T0CON :

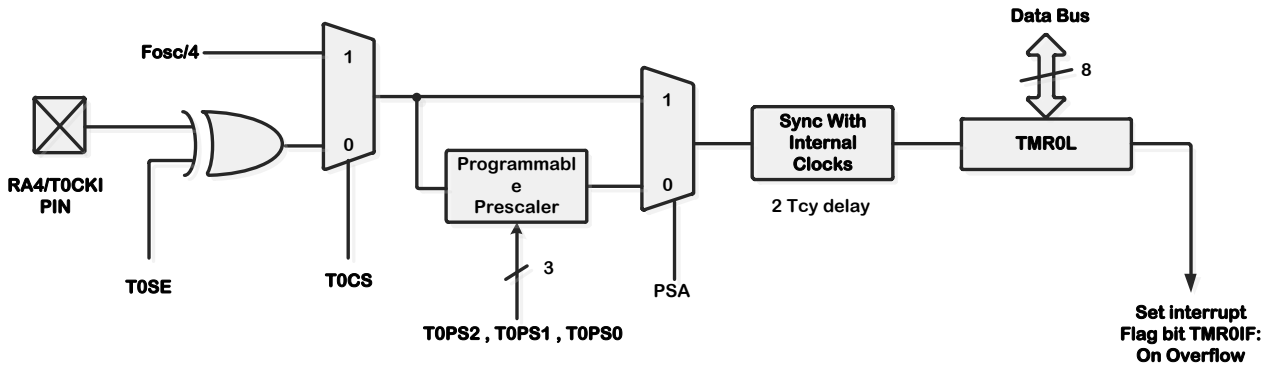
T0CON							
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

bit	Description
TMR0ON: bit 7	**بت التحكم بتشغيل وايقاف المؤقت** 1 =تمكين المؤقت 0=ايقاف المؤقت
T08BIT:bit 6	**اختيار نمط 8 بت أو 16 بت للمؤقت** 1=سيتم اختيار النمط 8 bit 0=سيتم اختيار النمط 16 bit
T0CS:bit 5	**بت يحدد مصدر الساعة للمؤقت** 1=نبضات الساعة خارجية مطبقة على الرجل RA4/T0CKI (عداد) 0=نبضات الساعة داخلية CLKO (مؤقت)
T0SE:bit 4	**بت يحدد الجبهة التي عندها سيتم زيادة قي مسجل المؤقت** 1=سيتم زيادة السجل عند كل جبهة هابطة على الرجل RA4/T0CKI (عداد) 0= سيتم زيادة السجل عند كل جبهة صاعدة على الرجل RA4/T0CKI (عداد) نلاحظ أن هذا البت يخص العداد فقط
PSA:bit 3	**بت اختيار المقسم** 1=المؤقت لا يستخدم المقسم (أي مثلا في حالة العداد نسبة 1:1 أي عند ورود نبضة على الرجل RA4 سيتم زيادة المسجل بمقدار واحد) 0=المؤقت يستخدم المقسم (مثلا كل ثماني نبضات خارجية يتم المسجل واحد كما اسلفنا)
T0PS2 : T0PS0:bit 2-0	**البتات التي تحدد نسبة التقسيم المطلوبة** 111= نسبة التقسيم هي 1/256

110= نسبة التقسيم هي 1/128	
101= نسبة التقسيم هي 1/64	
100= نسبة التقسيم هي 1/32	
011= نسبة التقسيم هي 1/16	
010= نسبة التقسيم هي 1/8	
001= نسبة التقسيم هي 1/4	
000= نسبة التقسيم هي 1/2	

الجدول 9

الشكل التالي يبين مخطط هذا المؤقت :



الشكل 31

مثال (Example) :

T0CON							
1	1	1	1	0	0	1	0

تمكين الـ Timer0 .

نمط 255 بت : أي تحدث مقاطعة عندما ينتقل العداد من 255 الى الصفر وليس عندما يصل الى الـ255 .

يتلقى النبضات من الرجل RA4 .

يتم الزيادة عند الجبهة الهابطة .

استخدام نسبة تقسيم .

نسبة التقسيم هي 8 : حسب الجدول أعلاه أي عندما ترد 8 نبضات على الرجل RA4 يزداد العداد بمقدار واحد .

مثال :

T0CON							
1	1	0	0	0	0	1	0

تمكين الـTimer0.

نمط 255 بت : أي تحدث مقاطعة عندما ينتقل العداد من 255 الى الصفر وليس عندما يصل الى الـ255(0xFF to 0x00) .

يتلقى النبضات من الكريستالة.

لا توثر في حال كان واحد أم صفر .

استخدام نسبة تقسيم .

نسبة التقسيم هي 8 : حسب الجدول أعلاه أي عندما ترد 8 نبضات من الكريستالة الى المتحكم يزداد السجل بمقدار واحد .

مثال :(عدم استخدام مجال القسمة)

T0CON							
1	1	1	0	1	0	1	0

البت الرابع يساوي الواحد يعني أننا ألغينا التقسيم معنى هذا أن السجل سوف يزداد مع كل نبضة للكريستالة أي أن نسبة التقسيم هي 1:1 ، نفس الأمر لو كان مصدر النبضات هو الرجل RA4 .

البت الرابع يساوي الواحد يعني أننا ألغينا التقسيم معنى هذا أن السجل سوف يزداد مع كل نبضة للكريستالة أي أن نسبة التقسيم هي 1:1

1.3.1. مبدأ العمل :

نحمل رقم ما(رقم ابتدائي) الى المؤقت(نقصد المسجل داخلي 8 بت) فيعد هذا المؤقت بدءا من هذا الرقم حتى يصل الى القيمة(255في نمط 8بت أو الى القيمة 65535 في نمط 16 بت) عندها يولد المؤقت مقاطعة. حيث يقوم برنامج خدمة المقاطعة بإعادة تحميل نفس الرقم الابتدائي الى المؤقت ليقوم مرة اخرى بالعد من هذه القيمة الابتدائية وصولا الى الرقم (255في نمط 8بت أو الى الرقم 65535 في نمط 16 بت) عندها يولد مقاطعة تقوم بإعادة تحميل الرقم الابتدائي الى المؤقت وهكذا ...تتكرر هذه العملية .

نستفيد من الرقم الابتدائي بتحديد طول الدور (الزمن الذي يستغرقه العداد ليعد من الرقم الابتدائي الى القيمة 255)

فكلما كان الرقم الابتدائي صغيرا كان الدور كبيرا

وكلما كان الرقم الابتدائي كبيرا كان الدور صغيرا

1.3.2. عمله كمؤقت :

المؤقت يحتاج الى تردد (نبضات) لكي يعمل وبما أن تردد الكريستالة يكون غير مناسب عادة لذلك يستعمل ترددات داخلية مختلفة مشتقة من تردد الكريستالة (ocs/2 أو ocs/4 أو ocs/8 أو ocs/16 أو ocs/32 أو ocs/128 أو ocs/256).

ocs/4 : عندما يرد الى المتحكم أربع نبضات من الكريستالة تزداد قيمة الـTimer0 بمقدار واحد .

ocs/8 : عندما يرد الى المتحكم ثماني نبضات من الكريستالة تزداد قيمة الـTimer0 بمقدار واحد وهكذا.

حيث ocs :تردد الكريستالة

Overflow time =4 xTOSC x Prescaler x (256 _ TMR0)

where

Overflow time is in µs : الزمن الازم ليعد من القيمة الابتدائية الى الرقم255

T=1/F : دور الكريستالة وهو مقلوب التردد TOSC is the oscillator period in µs

Prescaler is the Prescaler (2 أو 4 أو 8 أو 16 أو32 او64 أو 128 أو 256) :نسبة التقسيم

TMR0 is the value loaded into TMR0 register :القيمة الابتدائية التي تم تحميلها الى العداد

مثال لنفرض أن تردد الكريستالة 4MHz ونسبة التقسيم 32 والقيمة الابتدائية بالعشري هي 100

يحسب دور العد كما يلي :

4MHZ clock has a period; T = 1/F = 0.25µs

Overflow time = 4 xTOSC x Prescaler x (256 _ TMR0)

Overflow time = 4 x 0.25 x 32 (256 −100)=4992µs

اي أنه بعد كل 4.992ms ستحدث مقاطعة اذا كانت timer interrupt و global interrupts ممكنة

من العلاقة السابقة يمكننا استنتاج هذه العلاقة التي سنعتمد عليها دائما للحصول على الدور المرغوب

$$TMR0 = 256 - \frac{\text{Overflow time}}{4 \text{ x TOSC x Prescaler}}$$

مثال : نريد أن تحدث المقاطعة كل 500µs

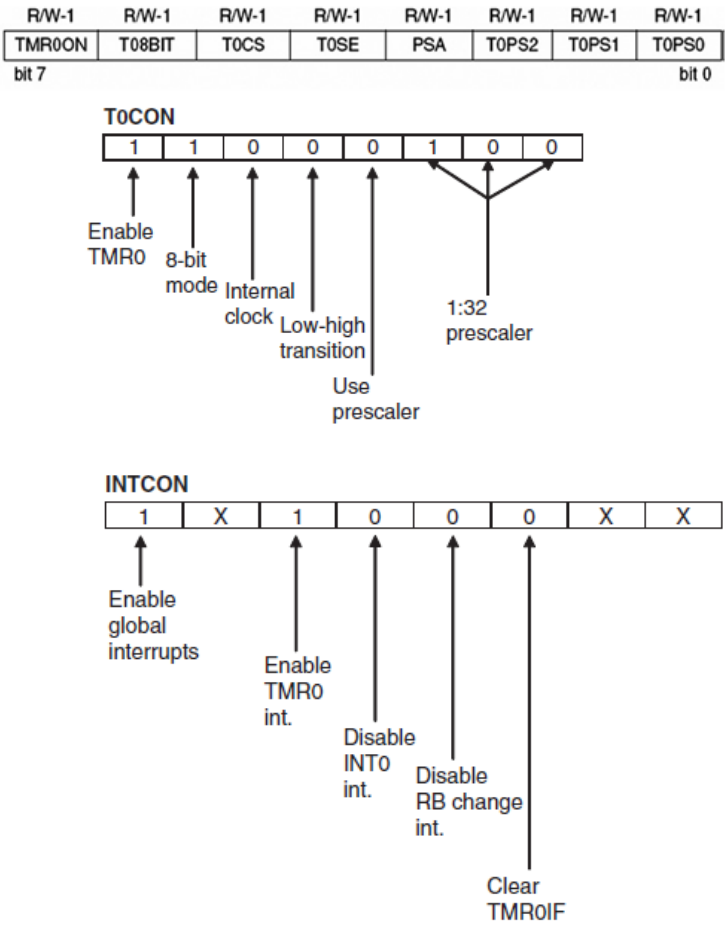
وتردد الساعة (الكريستالة) هو 4MHZ

نسبة التقسيم 8 وذلك بجعل T0PS0: T0PS2=010

( )

$$TMR0 = 256 - \frac{500}{4 \text{ x } 0.25 \text{ x } 8} = 193.5$$

أي أننا اذا أردنا دورا قدره 500µs فإننا نحمل القيمة الابتدائية 193 الى العداد TMR0



مثال :

نريد أن تحدث مقاطعة كل 160µs (delaytime)

تردد الكريستالة 3.58MHz

نسبة التقسيم 1:1

احسب القيمة الابتدائية (Initial)التي سنضعها في المسجل TMR0 .

$$TMR0 = 256 - \frac{\text{Overflow time}}{4 \text{ x TOSC x Prescaler}}$$

$$TMR0 = 256 - \frac{160\mu s}{4 \text{ x } (\frac{1}{3.58}) \text{ x } 1} = 112.8$$

نقوم بتقريب النتيجة لأقرب رقم وهو 113

إذا يجب تحميل القيمة 113 الى المسجل TMR0.

مثال :

نريد أن تحدث مقاطعة كل 5ms (delaytime)

تردد الكريستالة 4MHz

نسبة التقسيم 1:1

$$TMR0 = 256 - \frac{\text{Overflow time}}{4 \text{ x TOSC x Prescaler}}$$

$$TMR0 = 256 - \frac{5*10^{-3}s}{4 \text{ x } 0.25*10^{-6} \text{ x } 1} = -4744$$

نلاحظ أن القيمة الناتجة سالبة أي أنها خاطئة

نستنتج أنه من أجل زمن تأخير كبير يحتم علينا استخدام نسب التقسيم لنجرب نسبة التقسيم 1:2 ستكون النتيجة خاطئة وكذلك نسبة التقسيم 1:8 .....

حتى تكون القيمة صحيحة يجب أن يكون المقدار

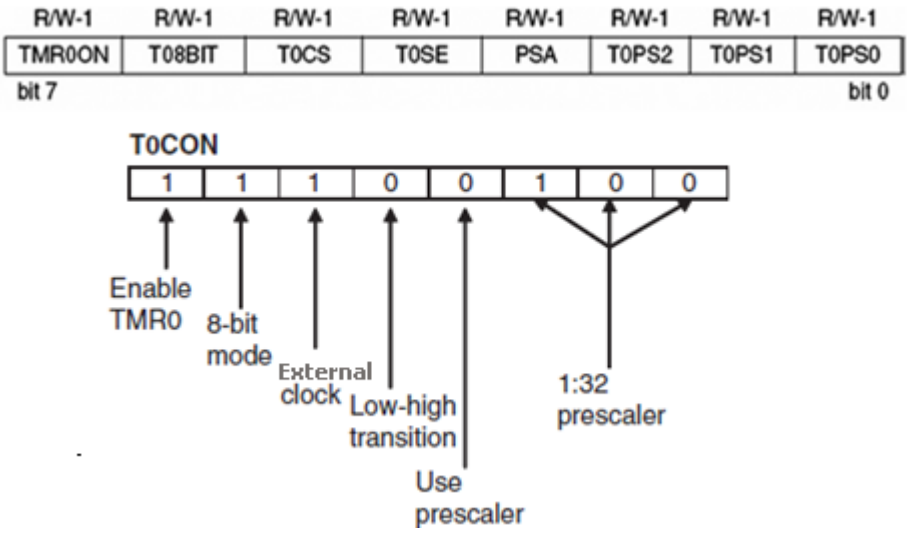
$$\text{ledelay cycle} = \frac{\text{Overflow time}}{4 \text{ x TOSC x Prescaler}} \text{ أكبر من } 256$$

لذلك نستخدم نسبة التقسيم 32

$$\text{TMR0} = 256 - \frac{5*10^{-3}\text{s}}{4 \text{ x } 0.25*10^{-6} \text{ x } 32} = 99.75$$

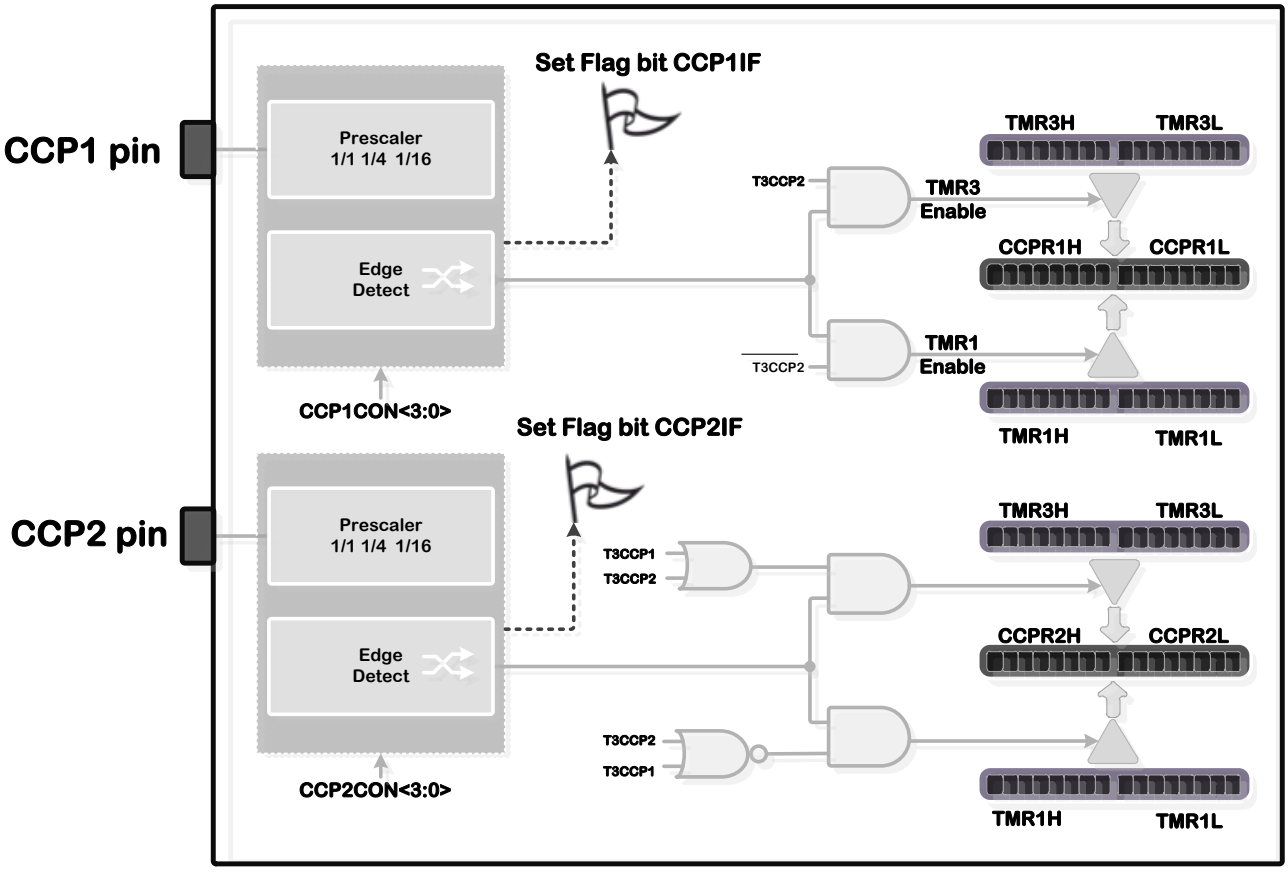
ومنه نقرب الى أقرب قيمة وهي 100 .

1.3.3. عمله كعداد :



TOCS=1 :مصدر الإشارة خارجي

T0PS2:T0PS0=100 : كل 32 نبضة خارجية فإن الTimer0 سيزداد بمقدار واحد



Capture mode of operation 1

1.4. الماسك / المقارن / تعديل PWM :

يمتلك المتحكم PIC18F452 موديولين CCP كل موديول يقدم ثلاث وظائف ويستخدم المؤقتات لتحقيق عمله ،كل موديول لديه مسجلين كل مسجل بعرض 8-bit

Pulse Width Modulation (PWM) using PIC CCP module

تعديل عرض النبضة هو تقنية تستخدم للتحكم بكمية الطاقة الموجهة إلى الحمل عن طريق الوصل والفصل المتكرر . فترة الوصل ON تعرف باسم DUTY CYCLE وبذلك فإن القيمة الوسطية للجهد المستمر يمكن أن تتغير بتغيير الـ DUTY CYCLE

DUTY CYCLE : يمكن أن تتغير بين الـ 0 (signal is always off)والـ 1 (signal is constantly on)

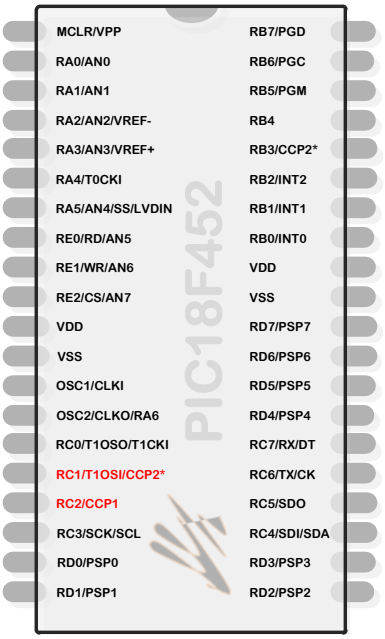
افترض أن إشارة قيمتها 5VOLT في حالة ON وقيمتها0 VOLT في حالة OFF

بتغيير الـ DUTY CYCLE أي فترة الوصل

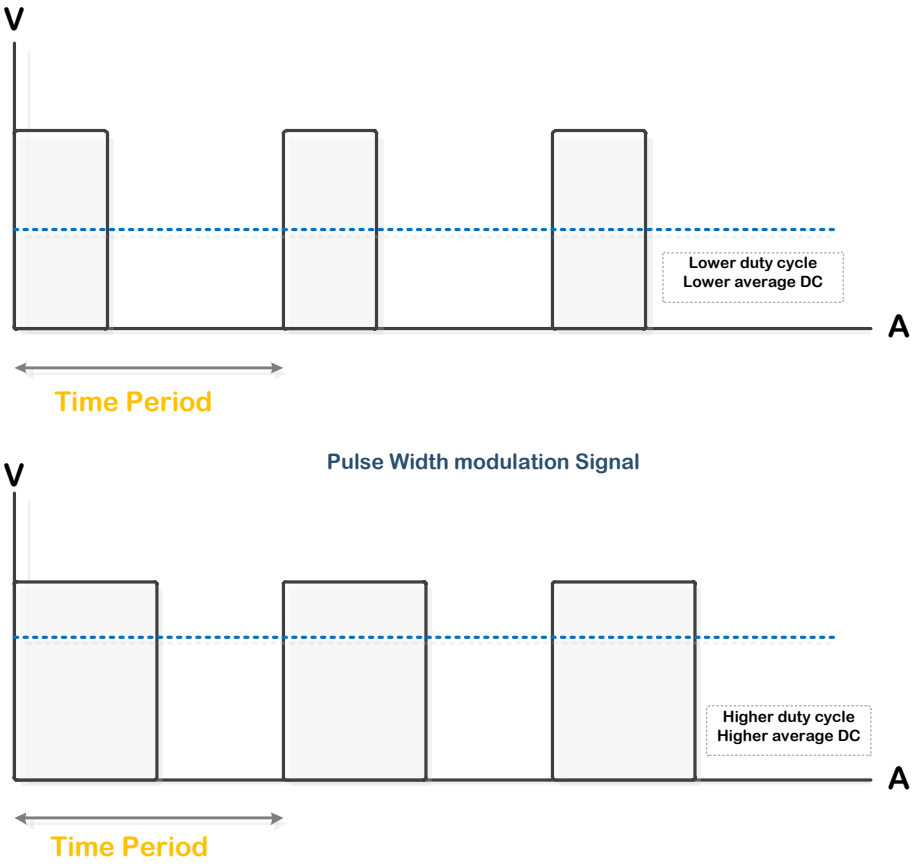
يمكننا الحصول على أي جهد بين الـ 0و 5VOLT هذه الطريقة شائعة جدا وتستخدم كثيرا في التحكم بسرعة المحركات المستمرة وكذلك التحكم بشدة الإضاءة DC motors and brightness of lamps

هذا الفصل سوف يتحدث عن كيفية توليد إشارة PWM باستخدام المايكرو لإرسال اشارة التحكم باستخدام IR-LED ويتم توليد الإشارة عن طريق هاردوير داخلي في المايكرو يدعى الموديول:
Capture/Compare/PWM (CCP) ونستطيع التعامل معه عن طريق الرجل RB3 .

هناك بعض المتحكمات لها أكثر من موديول للـ pwm نستطيع معرفة الأرجل التي يمكن أن نتعامل من خلالها مع هذا الموديول عن طريق وجود اسم الموديول بجانب الرجل كما في الصورة التالية فإن هذا المتحكم يملك موديولين pwm هما CCP1,CCP2 كما في الشكل التالي :



الشكل 32



الشكل 33

موديول Capture/Compare/PWM (CCP) في المايكرو متعدد الاستخدام ويملك ثلاث مسجلات :

CCP CONTROL REGISTER مسجل التحكم

CCP HIGH BYTE مسجل الـCCP العلوي

CCP LOW BYTE مسجل الـCCP السفلي المسجل العلوي مع السفلي يمكن اعتبارهما كمسجل 16 بت

الـ Capture/Compare/PWM (CCP) في المايكرو متعدد الاستخدام ويعمل بثلاث أنماط :

نمط المسك Capture : عند حدوث حدث EVENTS ما على القطب CCPX فإن قيمة الـTIMER1 وهي عبارة عن 16 بت يتم تخزينها في المسجل CCP ذو الـ16 بت (العلوي والسفلي) .وهذا الحدث يمكن أن يكون إما جبهة هابطة أو جبهة صاعدة أو أربع جبهات صاعدة أو 16 جبهة صاعدة .

CCP1CON – CCP OPERATION REGISTER (ADDRESS: 17h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

bit 7-6 **Unimplemented**: Read as '0'

bit 5-4 **CCP1X:CCP1Y**: PWM Least Significant bits

Capture Mode
Unused

Compare Mode
Unused

PWM Mode
These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

bit 3-0 **CCP1M<3:0>**: CCPx Mode Select bits

0000 = Capture/Compare/PWM off (resets CCP1 module)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCP1IF bit is set)
1001 = Compare mode, clear output on match (CCP1IF bit is set)
1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)
1011 = Compare mode, trigger special event (CCP1IF bit is set; CCP1 resets TMR1
11xx = PWM mode

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

الرجل RC2/CCP1 في المايكرو PIC16F877A يجب أن تهيئ كـINPUT والـTIMER1 يجب أن يكون

إما في نمط Timer mode أو نمط synchronized Counter mode و نمط المسك Capture يمكن أن يولد مقاطعة interrupt

1.4.1. نمط المقارنة Compare :

تتم مقارنة قيمة الـ CCPR(العلوي +السفلى) مع الـ TIMER1 وعندما تتم المطابقة فإن الرجل CCP1 سوف تصبح HIGH ، أو LOW ، أولا تتأثر أبدا وذلك حسب الإعدادات الموجودة في المسجل CCP1CON .

الرجل RC2/CCP1 في المايكرو PIC16F877A يجب أن تهيئ كـ OUTPUT .عن طريق المسجل TRASC. سيتم شرح نمط الـ CAPTURE و COPPARE بتفصيل أكبر في فصول لاحقة .

1.4.2. نمط الـ PWM :

نمط تعديل عرض النبضة ، الوسيط بين الموديول الداخلي والعالم الخارجي هي الرجل RC2/CCP1

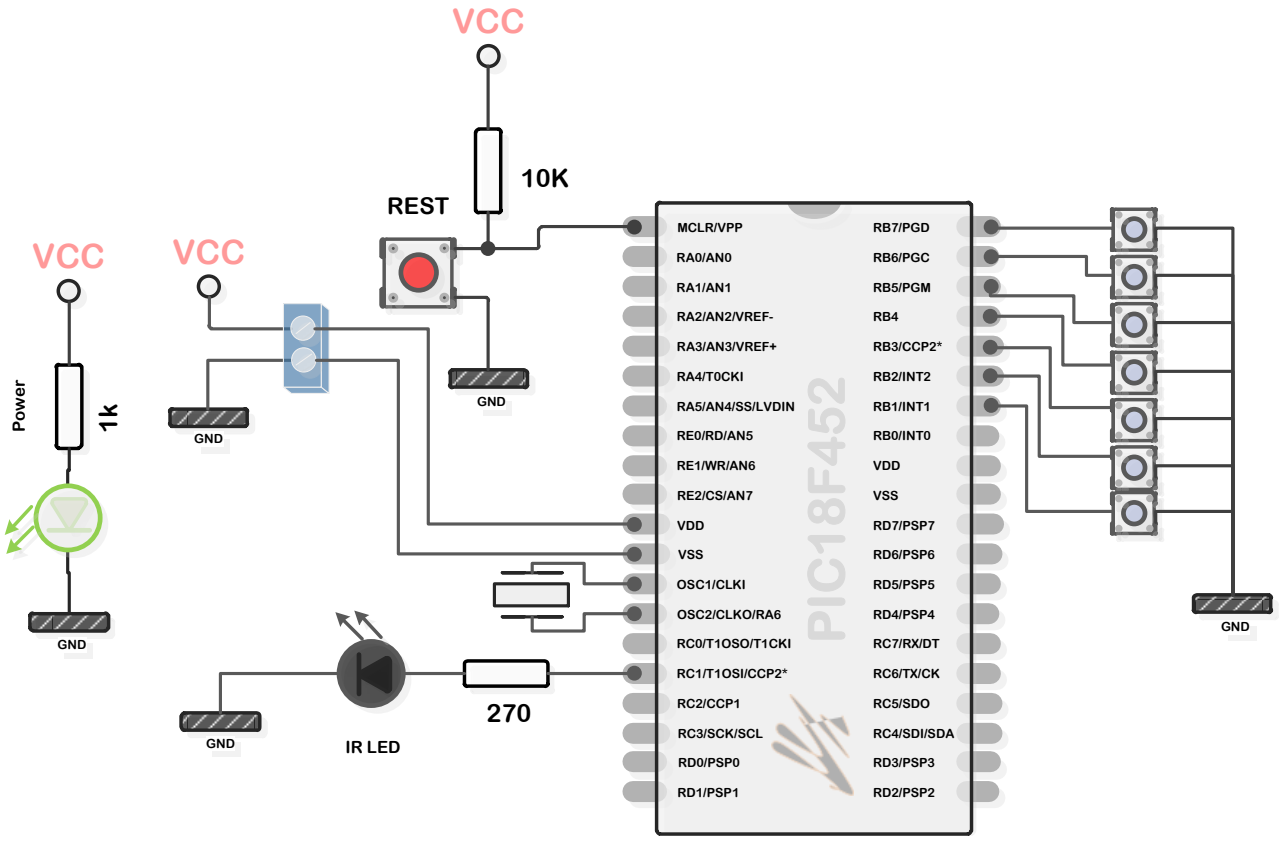
مخطط الدارة بسيط جدا يحوي على عدة مفاتيح لحظية موصولة الى PB0→7 المعرفة كمداخل أما الليد فموصول الى الموديول RB3/CCP1 من خلال مقاومة 330 أوم .

يقدم مكتبة MikroC Pro من الإجراءات التي تسهل التعامل مع الموديول CCP وهذه التوابع

PWM1_Init(const long frequency)

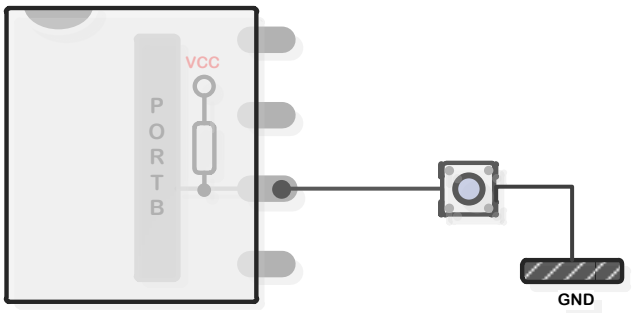
PWM1_Set_Duty(unsigned short duty_ratio)

PWM1_Start(void), and PWM_Stop(void)



الشكل 34

لتبسيط الدارة استخدمنا مقاومات الشد الداخلية يوضح الشكل التالي التوصيل الداخلي لمقاومة شد مع قطب من النافذة B .



الشكل 35

This Page intentionally left blank

الفصل الرابع

The PS/2 Communication Protocol

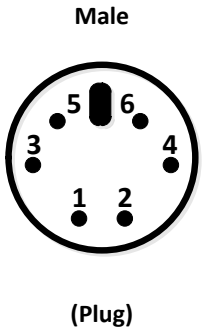
1.1. لوحات المفاتيح Keyboards :

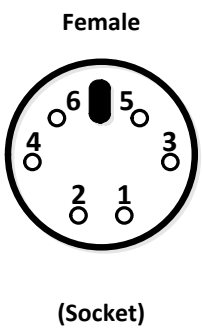
تعتبر لوحات المفاتيح من أكثر وحدات الإدخال شيوعا فهي تؤمن ادخال الاحرف بطريقة سهلة وتستخدم لذلك بروتوكول يدعى PS2 ولحسن الحظ فإن ربطها مع المتحكمات يعطي مزيدا من القوة للتطبيقات ومزيدا من المرونة .

1.2. تحقيق الواجهة PS/2 Physical Interface

تستخدم لوحات المفاتيح موصلات مكونة ست أرجل (mini–DIN) كماه هو واضح بالشكل

التالي:





6-pin mini-DIN (PS/2):
1-Data
2-NC
3-Ground
4-Vcc (+5V)
5-Clck
6-NC

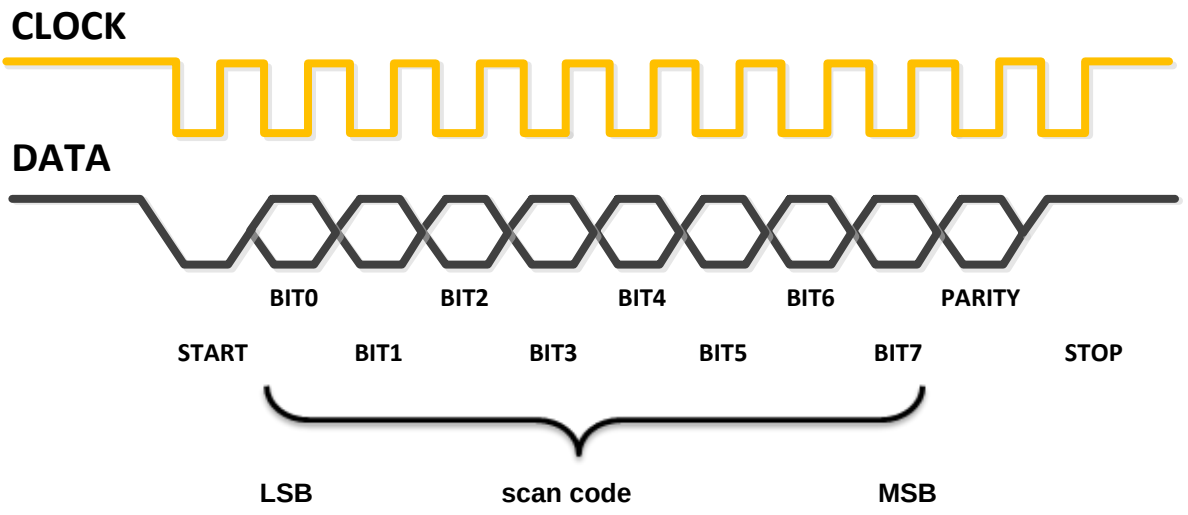
الشكل 36

بالنسبة للمضيف عليه تقديم جهد تغذية و عبارة عن 5V أما استهلاك الطاقة فهو يعتمد على نوع لوحة المفاتيح ويتراوح بين 100 mA , 50 .خط المعطيات والساعة يجب وصلهما مع مقاومات شد (1–10 k ohm) ،

1.3. البروتوكول The PS/2 Communication Protocol

في حالة الخمول أو الحالة الطبيعية يكو كلا خطي المعطيات والتزامن على المنطق العالي بسبب مقاومات الشد (open-collector). عند هذه الحالة فقط فإن لوحة المفاتيح تسمح بإرسال المعطيات

.



الشكل 37

Keyboard-to-host communication waveform.

a. 1 start bit. This is always 0.

b. 8 data bits, least significant bit first.

c. 1 parity bit (odd parity).

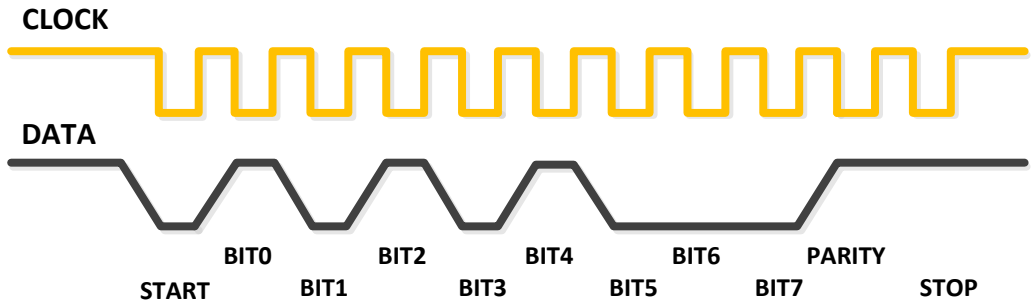
d. 1 stop bit. This is always 1.

clock	تؤمن نبضات الساعة من أجل التزامن ، نبضات الساعة ثنائية الإتجاه من المضيف الى لوحة المفاتيح و العكس بالعكس
Data	المعطيات ويكون اتجاهها من لوحة المفاتيح الى المضيف وهي تكون على شكل بايت واحد 8-bit (byte) ويدعى scan code وهو يعبر عن المفتاح الذي تم ضغطه . المعطيات اتجاهها من المضيف الى لوحة المفاتيح : عبارة عن أوامر لتهيئة اللوحة يمليها المضيف . في هذا الفصل سنتناول فقط الاتجاه من اللوحة الى المضيف

الجدول 10

يعتبر هذا البروتوكول من الاتصالات التسلسلية المتزامنة، لكنه يشبه البروتوكول المتزامن بسب نبضات الساعة ، ويشبه غير المتزامن بسبب start, a stop, and a parity bit حيث تقوم هذه البتات بحصر أو كبسلة 8- bit من المعطيات

Parity هذا البت يكون واحد اذا كان عدد الواحدات فردي ويكون صفر اذا كان عدد الواحدات زوجي ويستخدم من أجل كشف الأخطاء



الشكل 38

Scan code for the "Q" key (15h)

الشكل 39

تسلسل الأحداث :

ضغط المفتاح (key pressed) : سيتم ارسال 11-Bit

بت البداية start bit ، بتات المفتاح 8-bit scan code ، خانة الفردية odd parity bit ، بت التوقف stop bit.

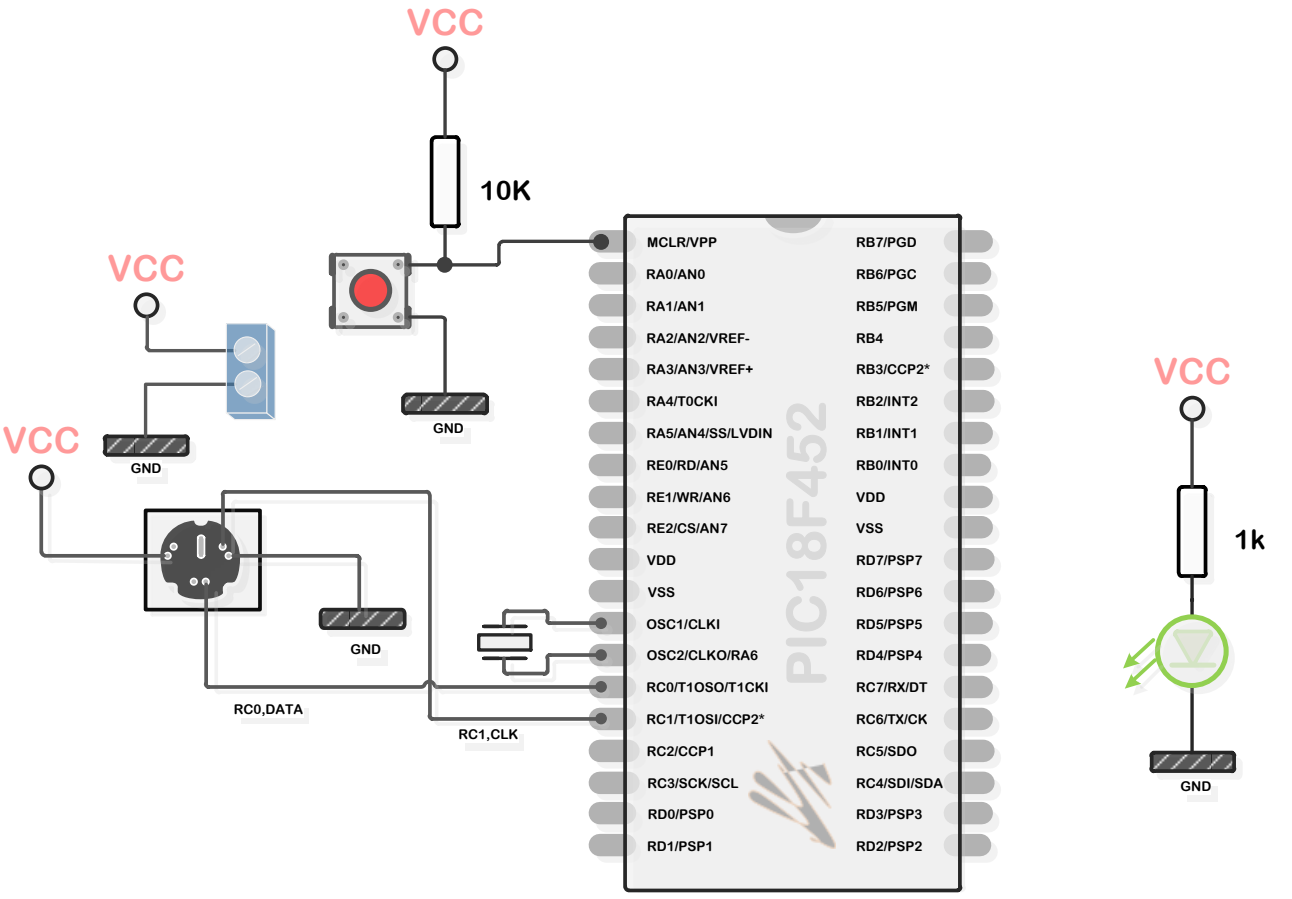
تحرير المفتاح (key released) : سيتم ارسال 11-bit مرتين

الأولى يتم ارسال كود مسح FO = scan code

الثانية يتم ارسال كود المسح الخاص بالمفتاح .

أما في حال استمرار الضغط (key held down) : سيتم ارسال كود المسح الخاص بالمفتاح (11-bit) كل 100ms .

سرعة الإرسال تتراوح بين 10-16.7 kHz حيث يقوم معالج لوحة المفاتيح بمسح مصفوفة المفاتيح واذا وجد أنه ثمت مفتاح قد ضغط سيقوم بإرسال شيفرة هذا المفتاح الى المتحكم


VCC
10K
VCC
VCC
GND
GND
GND
RC0,DATA
RC1,CLK
MCLR/VPP
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS/LVDIN
RE0/RD/AN5
RE1/WR/AN6
RE2/CS/AN7
VDD
VSS
OSC1/CLKI
OSC2/CLKO/RA6
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2*
RC2/CCP1
RC3/SCK/SCL
RD0/PSP0
RD1/PSP1
RB7/PGD
RB6/PGC
RB5/PGM
RB4
RB3/CCP2*
RB2/INT2
RB1/INT1
RB0/INT0
VDD
VSS
RD7/PSP7
RD6/PSP6
RD5/PSP5
RD4/PSP4
RC7/RX/DT
RC6/TX/CK
RC5/SDO
RC4/SDI/SDA
RD3/PSP3
RD2/PSP2
PIC18F452
VCC
1k
GND

الشكل 40

PS2: in mikroC pro 5.61

```
/*
 * Project name:
     PS2
 * Copyright:
      mustafa haj abd alrhmaan

 * Description:

 * Test configuration:
     MCU:             PIC18F452
     Oscillator:      HS, 16.0000 MHz
     Ext. Modules:    -
     SW:              mikroC PRO V5.61
 * NOTES:TYPE LEDx : x:1--→8 , TYPE : hi , TYPE :doctor ,
 This is all command

*/
// LCD module connections
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;

sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;

sbit PS2_Data            at RC0_bit;
sbit PS2_Clock           at RC1_bit;
sbit PS2_Data_Direction  at TRISC0_bit;
sbit PS2_Clock_Direction at TRISC1_bit;


unsigned short  special=0,keydata, down=0 ,i=0;
unsigned short READ[5];
void main() {
 TRISD  = 0x00;
 TRISC.F4  = 0x00;TRISC.F5  = 0x00;
 PORTD=0x00;
 TRISB = 0;

  Lcd_Init();                        // Initialize LCD
  Lcd_Cmd(_LCD_CLEAR);               // Clear display
  Lcd_Cmd(_LCD_CURSOR_OFF);          // Cursor off

  Delay ms(100);
  UART1_Init(19200);                 // Initialize UART module at 19200 bps
  Ps2_Config();                      // Init PS/2 Keyboard

 Delay_ms(100);     // Wait for keyboard to finish
 Delay ms(100);
 while (1) {
      if (Ps2_Key_Read(&keydata, &special, &down)) {
       PORTC.F4=0x01;
       Delay ms(100);
        PORTC.F4=0x00;
      if (down && (keydata == 16)) {// Backspace
        // ...do something with a backspace...  LCD SHIFT RIGHT
```

```
                Lcd Chr(1, i, 32);
                READ[i]=32;
          // ...do something with a backspace...
                i--;

      }
      else if (down && (keydata == 13)) {// Enter
                //_____________ ____________________
                  Lcd Chr(2, 1, READ[0]);Lcd Chr(2, 2, READ[1]);
                  Lcd Chr(2, 3, READ[2]);Lcd Chr(2, 4, READ[3]);Lcd Out(2, 5, "               ");
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==49){ PORTD=0x02;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==50){ PORTD=0x01;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==51){ PORTD=0x04;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==52){ PORTD=0x08;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==53){ PORTD=0x10;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==54){ PORTD=0x20;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==55){ PORTD=0x40;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==56){ PORTD=0x80;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==114){ PORTD=0xF0;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==108){ PORTD=0x0F;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==97){ PORTD=0xFF;}
                  if(READ[0]==108&&READ[1]==101&&READ[2]==100&&READ[3]==102){ PORTD=0x00;}
                  if(READ[0]==104&&READ[1]==105){Lcd_Out(2, 1, "Hello!              "); }
                  if(READ[0]==100&&READ[1]==111&&READ[2]==99&&READ[3]==116){Lcd_Out(2, 1, "D.Haleem
Ali"); }

      }         //_____________________________________
      else if (down && !special && keydata) {
        Lcd_Out(2, 5, "               ");
        READ[i]=keydata;


        Lcd Chr(1, i+1, keydata);
        i++;
      }
      else if (down && (keydata == 34)) {// Escape
                //_____________________________________
              PORTD=0x00;
              Lcd Cmd( LCD CLEAR);
              READ[0]=0;
              i=0;
      }         //_____________________________________
    }
    Delay_ms(10);     // debounce

  }
}
```

الفصل الخامس

SIRC Protocol

1.1. الاتصالات عن طريق الأشعة تحت الحمراء Sony SIRC Protocol :

تستخدم تقنية التحكم عن بعد الأشعة تحت الحمراء ، وهذه التقنية منتشرة جدا ومنذ فترة طويلة ، كل شركة مصنعة تستخدم بروتوكول خاص بها على سبيل المثال :

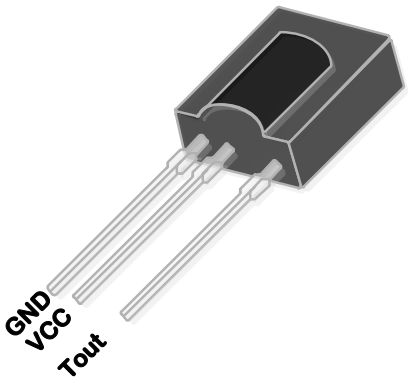
RC80 : هو بروتوكول تستخدمه شركة Panasonic

RC5 : هو بروتوكول تستخدمه شركة Philips .

SIRC : تستخدمه شركة Sony ويعتبر بسيط جدا .

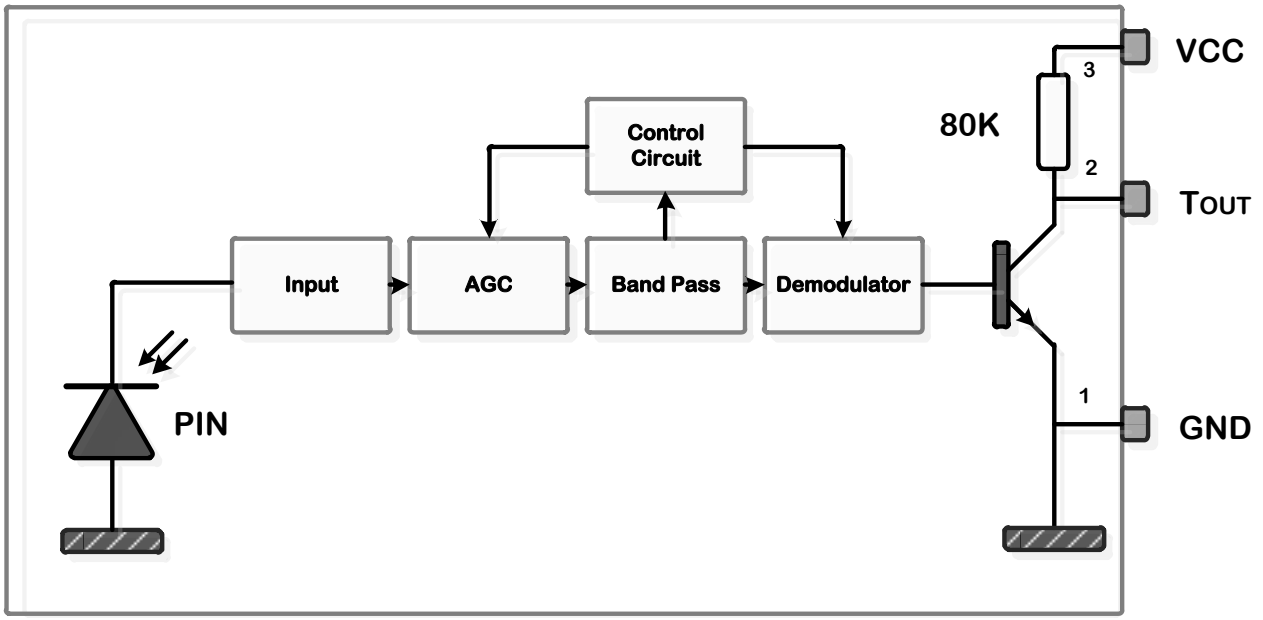
لا يمكن ارسال الأشعة تحت الحمراء من المرسل الى المستقبل بصيغتها البدائية وإنما يجب تعديلها قبل الارسال وذلك من أجل الغاء تأثيرات الضوء الموجود في المحيط الصادر عن طريق الشمس أو أي منبع يولد أشعه تحت حمراء .

يتم تعديل الإشارة قبل ارسالها عن طريق حامل ذو تردد مركزي يتراوح بين 32K الى 56K . في حالتنا هذه أي من أجل بروتوكول Sony فإن التردد المركزي للحامل هو 40khz وهذا يعني بالضرورة أننا بحاجة الى مستقبل يستطيع استقبال الأشعة تحت الحمراء المعدلة بالحامل 40k ومن ثم يقوم بتحويلها الى منطق الـ TTL من أجل معالجتها من قبل الـ PIC .



هناك العديد من مستقبلات الأشعة تحت الحمراء ، لكل واحدة منها تردد مركزي محدد وتكون عند هذا التردد أكثر حساسية .

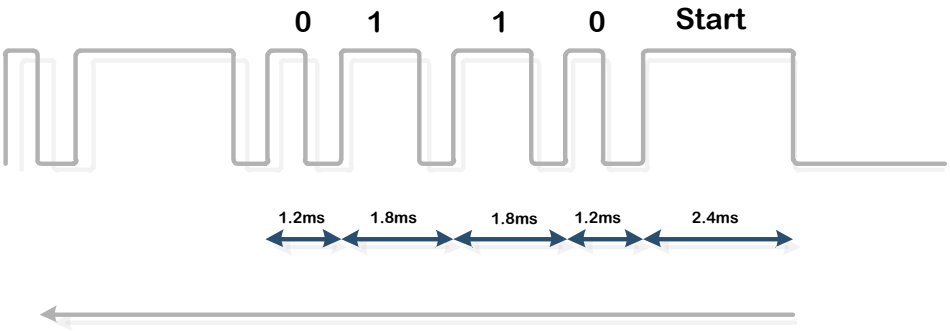
الشكل التالي يبين المخطط الصندوقي لمستقبل الأشعة تحت الحمراء



الخرج (Rout) يكون بالحالة الطبيعة high كما نرى من المخطط الصندوقي وذلك بسبب مقاومة الشد 80K و التي تبقي الخرج على المنطق العالي عندما لا يكون هناك أي اشارة مطبقة على المستقبل ، أما عند تطبيق اشارة تحت حمراء فإن الخرج سيصبح LOW .

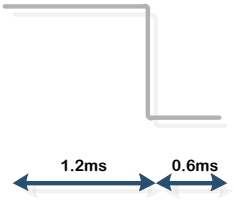
بروتوكول Sony أو ما يسمى SIRC (Serial Infra-Red Control) : هو عبارة عن بروتوكول يستخدم الأشعة تحت الحمراء حيث أن المرسل يقوم بتعديل الإشارة باستخدام صيغة من تعديل عرض النبضة PWM قبل ارسالها .

البروتوكول الأكثر انتشارا يستخدم 12-bit لكن هناك اصدارات تستخدم 15-bit و أخرى تستخدم 20-bit والشكل التالي يوضح قطار من النبضات 12-bit :

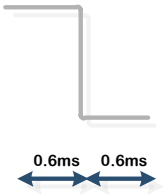


الشكل 41

حي الواحد منطقي يمثل كما يلي :



أما الصفر منطقي فيمثل كما يلي :



حيث صيغة الإرسال على الشكل التالي :

Header + command code (7-bit) + Device code (5-bit)

command code (7-bit) : وتمثل كود المفتاح الذي تم ضغطه

Device code (5-bit) : وتحدد هل الجهاز هو TV أو VCR أو CD_player .....

شكل الجهاز RM-870 المستخدم وهو يعتمد على بروتوكول SIRC .

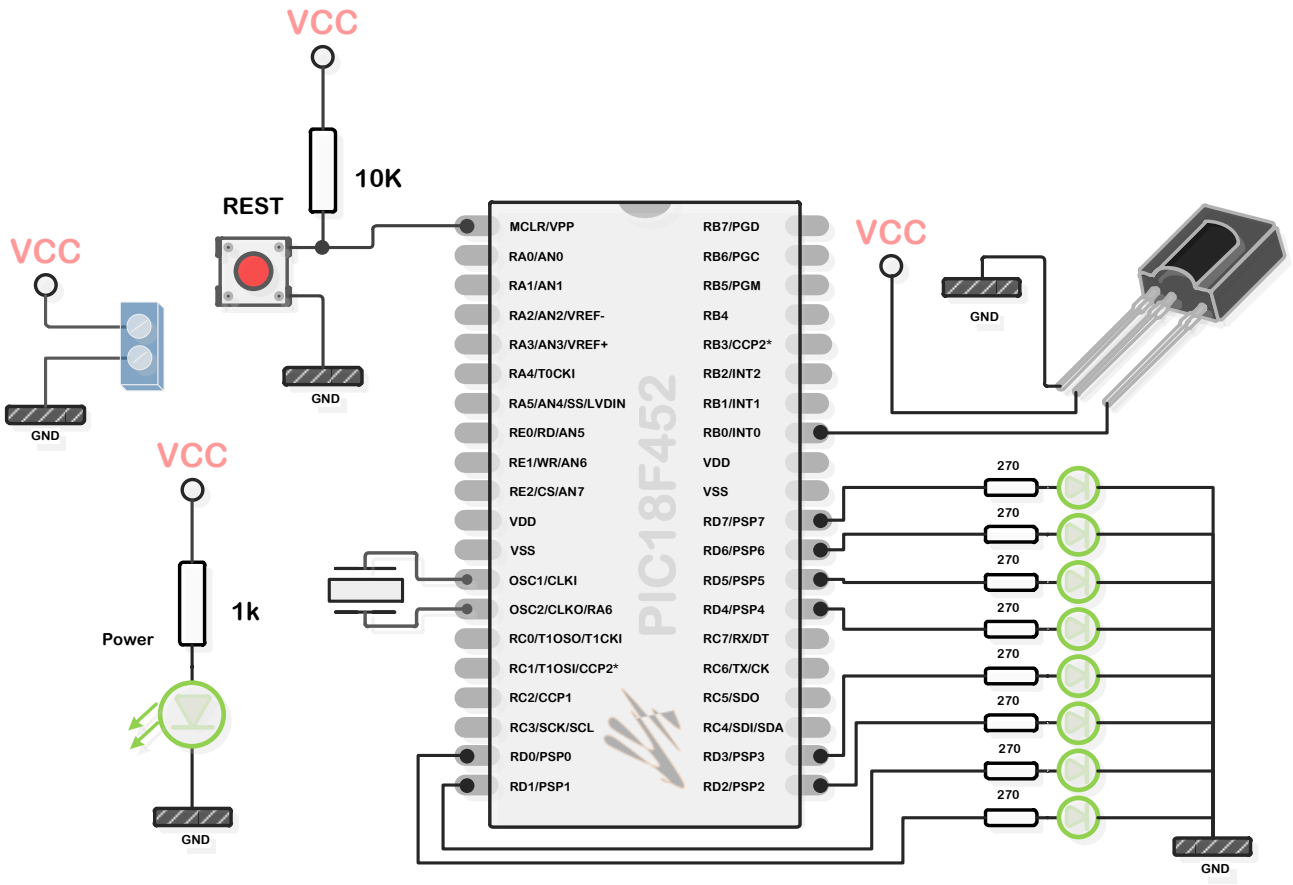
الشكل 42

الجدول التالي يبن وظيفة كل مفتاح من مفاتيح جهاز التحكم RM-870 مع الكود لكل مفتاح :

n.n	KEY	2nd FUNC	Code	n.n	KEY	2nd FUNC	Code
1	MUTING		14	**16**	-/--		1D
2	DISPLAY	REVEAL	3A	**17**	0		09
3	POWER		15	**18**		TEXT CLR	2B
4		TEXT	3F	**19**	S+	■	74
5	VIDEO	HOLD	25	**20**	SELECT	■	7C
6	TV		38	**21**	A/B	ENLARGE	17
7	1		00	**22**	-	■	75
8	2		01	**23**	PIC MODE	■	61
9	3		02	**24**	WAKE UP	INDEX	3C
10	4		03	**25**	SLEEP		36
11	5		04	**26**	VOLUME +		12
12	6		05	**27**	VOLUME -		13
13	7		06	**28**	CH +		10
14	8		07	**29**	CH -		11
15	9		08				

الجدول 11

1.2. الدارة العملية :



الشكل 43

SONY SIRC decoding : in mikroC pro 5.61

```
/*
 * Project name:
     SONY SIRC decoding
 * Copyright:
      Nicholas Sirirak
      updated by :
      mustafa haj abd alrhmaan
      for RM-870 remote control
 * Description:

 * Test configuration:
     MCU:             PIC18F452
     Dev.Board:       -
     Oscillator:      HS, 16.0000 MHz
     Ext. Modules:    -
     SW:              mikroC PRO V5.61
 * NOTES:

*/
#define  LED0  LATD.F0
#define  LED1  LATD.F1
#define  LED2  LATD.F2
#define  LED3  LATD.F3
#define  LED4  LATD.F4
#define  LED5  LATD.F5
#define  LED6  LATD.F6
#define  LED7  LATD.F7
unsigned mstx[] = {1,3,7,15,31,63,127,255};
unsigned mst = 0;
unsigned counter = 0;
unsigned input_data, bit_count;

enum {
        Idle,
        Start_bit,
        Capture_bit
};

char Current_state = Idle;
char got_data = 0;
char Command_code, Device_code;

void interrupt(){
     if(INTCON.INT0IF){
        switch (Current_state){
           case Idle:
                      INTCON2.INTEDG0 = 1;        //interrupt on rising edge.
                      counter = 0;
                      Current_state = Start_bit;
                      break;
           //found the rising edge, check lenght for 2.4ms
           case Start_bit:
                      //correct signal, move on to next state
                      if(counter == 4) {
                         counter = 0;
```

```
                        bit_count = 0;
                        input_data = 0;
                        Current_state = Capture_bit;
                        } else {
                           //fault signal, reset to Idle
                           Current_state = Idle;
                        }
                     break;
            case Capture_bit:
                     //check plus length for 0 or 1
                     if(counter == 2){
                        input_data >>= 1;          // add 0 to received data
                        bit_count++;
                     }else {
                        if(counter == 3){
                           input_data >>= 1;
                           input_data |= 0x8000;      //add 1 to received data
                           bit_count++;
                        } else {
                           //error occurs, reset to Idle state
                           INTCON2.INTEDG0 = 0;    //interrupt on falling edge.
                           Current_state = Idle;
                        }
                      }
                     //compleat 12 bit
                     if(bit_count >= 12){
                        got_data = 1;
                        input_data >>= 4;
                        INTCON2.INTEDG0 = 0;       //interrupt on falling edge.
                        Current_state = Idle;
                     }
                     counter = 0;
                     break;
             default: Current_state = Idle;
         }
      INTCON.INT0IF = 0;        //clear interrupt flag.
      }
      if(PIR1.TMR2IF){
         counter++;
         if(counter > 5) {
             Current_state = Idle;
             counter = 0;
             INTCON2.INTEDG0 = 0;         //interrupt on falling edge.
         }
         PIR1.TMR2IF = 0;           //clear interrupt flag
      }
}

//*****************************************************************************
// MAIN  MAIN  MAIN MAIN
//*****************************************************************************
void main() {
      TRISC = 0;                    //portc is output
      TRISD = 0;                    //portb is output

//*****************************************************************************
// RB0 interrupt set up
//*****************************************************************************
      INTCON.INT0IE = 1;           //enable RB0 interrupt.
```

```
     INTCON2.INTEDG0 = 0;         //interrupt on falling edge.
     TRISB.F0 = 1;                //RB0 = input.
     ADCON1 = 0x0F;               //all digital I/O
//*****************************************************************************
// Timer2 interrupt set up, interrupt every 600us
//*****************************************************************************
     T2CON = 2;                   //timer off, prescaler 1:16
     PR2 = 155;                   //preload timer2 comparator value.
     TMR2 = 0;                    //reset value timer2
     PIR1.TMR2IF = 0;             //clear interrupt flag.
     PIE1.TMR2IE = 1;             //enable timer2 interrupt.
     IPR1.TMR2IP = 1;             //timer2 interrupt high priority
//*****************************************************************************
// Global interrupt enable
//*****************************************************************************
     INTCON.PEIE = 1;             //enable interrupt
     INTCON.GIE = 1;              //enable global interrupt

     T2CON.TMR2ON = 1;            //timer2 is on
   // PORTD=0x00;
   // PORTD=~PORTD;
     while(1){
        // if(PORTB.F0){
        //  PORTD=~PORTD;
        //  }
          if(got_data){
            Command_code = input_data & 0x7F;
            Device_code = input_data >> 7;
            got_data = 0;
            if(Device_code == 1){
                  TMR2 = 0;                    //reset value timer2
                  if (mst>7){mst=0;}       //vol+-
               switch (Command_code){

                  case 0x00: LED0 = ~LED0; break;
                  case 0x01: LED1 = ~LED1; break;
                  case 0x02: LED2 = ~LED2; break;
                  case 0x03: LED3 = ~LED3; break;
                  case 0x04: LED4 = ~LED4; break;
                  case 0x05: LED5 = ~LED5; break;
                  case 0x06 :LED6 = ~LED6; break;
                  case 0x07: LED7 = ~LED7; break;
                  case 0x15: PORTD=0xFF; break;          // power
                  case 0x3A: PORTD=0x00; break;          //display

                  case 0x12: PORTD=mstx[mst];mst++; break;          //vol+
                  case 0x13: PORTD=mstx[mst];mst--; break;          //vol-

               }
               PIE1.TMR2IE = 0;             //disable timer2 interrupt.
               Delay_ms(100);

               PIE1.TMR2IE = 1;             //enable timer2 interrupt.
            }
          }
     }

}
```

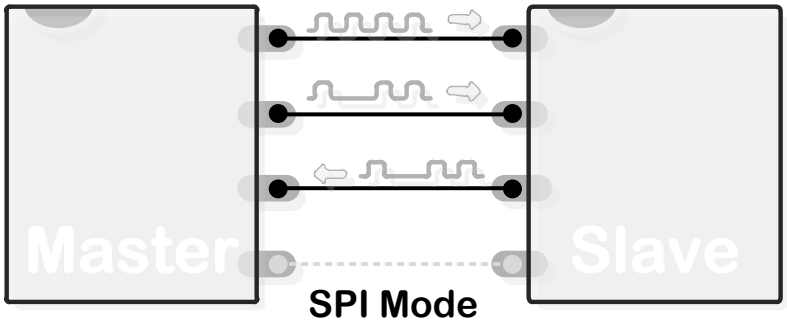
الفصل السادس

Synchronous Serial Port (MSSP)

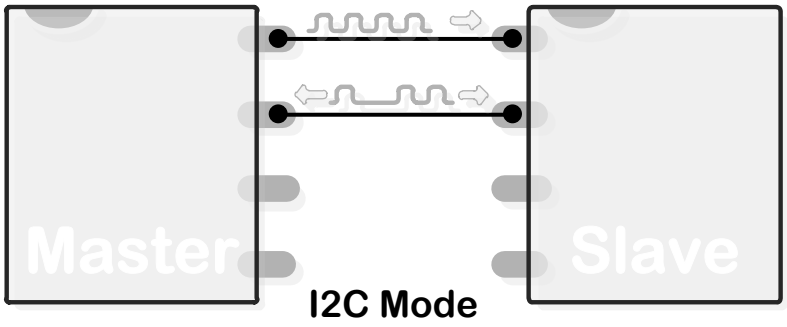
نستطيع عن طريق هذا الموديول تحقيق نوعين من الاتصالات التسلسلية SPI ، I2C .

SPI : يعتبر الأبسط والأسرع ويستخدم عنوان Hardware .

I2C : أكثر تعقيدا مع عنوان Software .

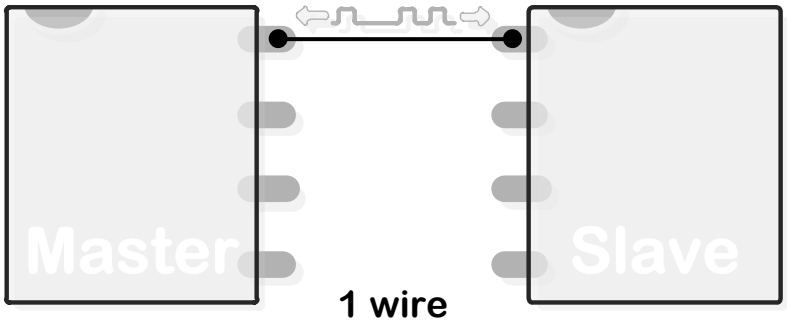


الشكل 44



الشكل 45

وسنقوم بالمقارنة مع one wire protocol



الشكل 46

1.1. SPI :

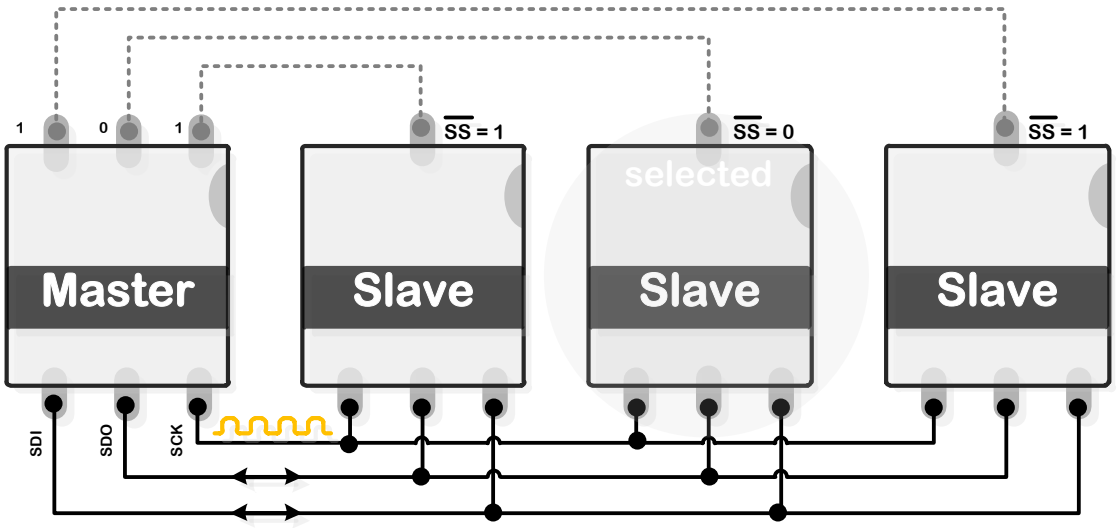
وهو عبارة عن بروتوكول اتصال تسلسلي يستخدم لإنشاء اتصال بين الشرائح يسمح هذا النمط بإرسال واستقبال 8bit من المعطيات بنفس الوقت ويتطلب لتحقيقه ثلاث خطوط :

SDO : معطيات الخرج

SDI : معطيات الدخل

SCK : من أجل الساعة ويقوم القائد بتوليدها

SS# : اختيار التابع (ولا يعتبر خط لأنه يمكننا وصله الى VCC مباشرة في حال وجود طرفية وحيدة وبالتالي لا نحتاج الى قطب رابع من المتحكم) أي يمكن ان يستخدمه المتحكم من أجل اختيار طرفية حال وجود أكثر من طرفية وتبادل المعطيات معها.



الشكل 47

Single synchronous master with multiple slaves

1.2. الناقل (I 2 C) :

وهو عبارة عن بروتوكول اتصال تسلسلي يستخدم لإنشاء اتصال بين الشرائح لمسافات قصيرة على نفس البورد، و يتطلب لتحقيقه خطين فقط و يشبه في عمله شبكة محلية بسيطة لكن في هذه الحالة هي شبكة بين الشرائح التي تدعم هذا النوع من البروتوكولات.

تم اختراعه بواسطة شركة Philips 1980 وكان الهدف الأساسي هو تأمين اتصال سهل بين معالجات Philips المختلفة التي تكون على نفس البورد في التلفزيون .أما اليوم أصبح استخدامه واسع جدا لتأمين الاتصال ذو السرعة المنخفضة بين المتحكم الرئيسي مع المحيطيات (EEPROM، الحساسات الرقمية ، شاشات LCD) .

وبشكل مماثل للبروتوكول SPI فإن المعطيات ترسل بشكل متزامن وثنائي الاتجاه باستخدام قطبين فقط من أقطاب المتحكم هما (SDA) و القطب (SLC) .

وبالتالي سيكون لدينا :

خط الساعة SCL (serial clock line) يعطي نبضات التزامن الضرورية من أجل نقل المعطيات

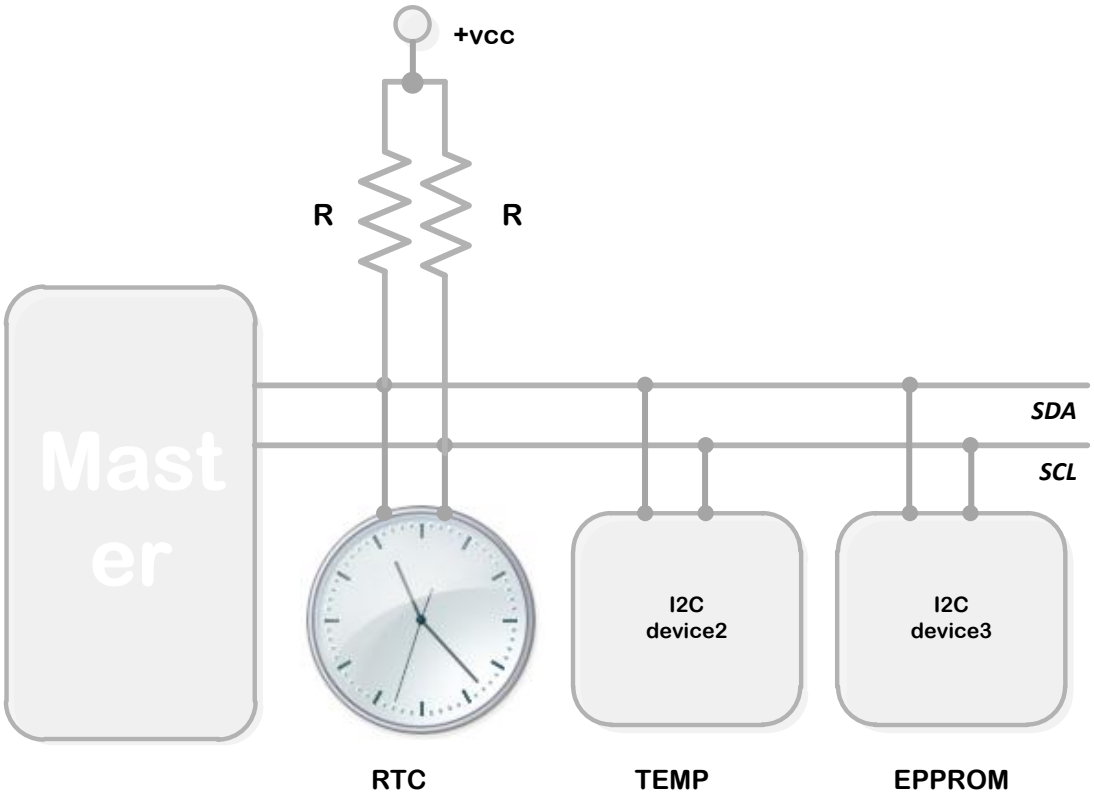
خط المعطيات SDA (serial data) المعطيات لابد أن ترسل عبر هذا الخط من أي تجهيزة الى اي تجهيزة اخرى .التجهيزات الموصولة على الخط I2C إما أن تكون Masters أو Slaves .

السيد فقط هو الذي يقوم بتهيئة المعطيات من اجل الإرسال أما العبد فيستجيب للسيد وحسب.

من المحتمل أن يكون هناك اكثر من سيد على خط واحد ولكن واحد فقط يكون فعال أي أن متحكم واحد تنتهي اليه مقاليد القيادة ، بالنسبة للقطب SCL دائما يكون مقاد بواسطة السيد.

في هذا البحث ، سأقوم بتناول حالة سيد وحيد والسيد في هذه الحالة هو المتحكم PIC18F2550 .

الحالة الأولى : سيد واحد وثلاثة عبيد ، العبيد كما قلنا لا يمكن أن تهيئ المعطيات أبدا لكن يمكنهم ارسال المعطيات عبر الممر I2C ، ودائما يتحكم بعملهم السيد .



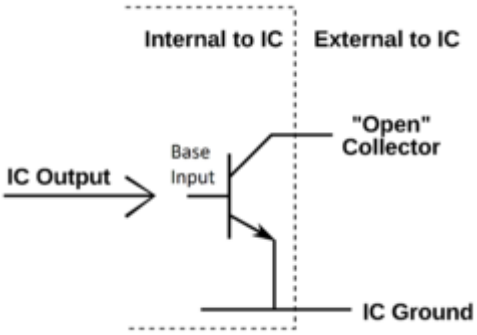
الشكل 48

كلا الخطين SDA , SCL يعتبران Open Drain ولذلك يتم وصلهما مع الجهد التغذية الموجب من خلال مقاومات شد 1.

هذا يعني أن التجهيزات على الناقل I2C يمكنها فقط أن تسحب الخط نحو المنطق المنخفض ولا يمكنها جعله على المنطق العالي .اي ان الخط هو دائما على المنطق العالي بسبب مقاومات الشد الى أن تأتي احدى التجهيزات وتجره الى المنطق المنخفض .لهذا السبب فإن مقاومات الشد مهمة جدا .

أنماط سرعة النقل عبر الخط I2C :

- في نمط لنقل القياسي حتى 100 Kbps
- نمط النقل السريع 400 Kbps
- في نمط السرعة العالية 3.4 Mbps



1 المصب المفتوح : وهو نوع شائع لأرجل الدارات المتكاملة ، فبدلا من أن يكون الخرج –مازلنا ضمن الدارة المتكاملة– جهد أو تيار محدد فإن اشارة الخرج تطبق على قاعدة ترانزستور NPN أما مجمع هذا الترانزستور يظهر للعالم الخارجي على شكل رجل للدارة المتكاملة ، أما الباعث فيكون موصول داخليا مع الأرضي

في الشكل أعلاه قاعدة الترانزستور تم تسميتها "IC Output" وهو الخرج الداخلي من منطق الدارة المتكاملة الى قاعدة الترانزستور أما من وجهة نظر الترانزستور فهي عبارة عن دخل والتي تتحكم بفتح واغلاق الترانزستور ، خرج الدارة المتكاملة هو عبارة عن المجمع فيكون عمل الترانزستور كأنه واجهة وصل بين العالم الخارجي ومنطق الدارة المتكاملة ، الخرج سيكون بالضرورة كدارة مفتوحة (غير موصول) أو موصول مع الأرضي . الخرج سيوصل مع مقاومات شد والتي ستجعل الخرج واحد منطقي عندما يكون الترانزستور في حالة قطع OFF ، أما عندما يصبح الترانزستور في حالة ON فإن الخرج سيجبر أن يصبح صفر منطقي بسبب وصله مع الأرضي

تطبيقات : لأن مقاومات الشد هي مقاومات خارجية ولا تتطلب أن تتصل مع نفس جهد الشريحة فيمكن وصلها مع جهد أعلى أو أخفض لذلك في بعض الأحيان تستخدم هذه الطريقة كواجهة وصل بين عوائل مختلفة من الشرائح التي تعمل عند سويات جهد مختلفة ،فائدة أخرى أنه يمكن وصل أكثر من مجمع (خرج) الى خط واحد .

1.2.1. اشارة حالة البدء والإيقاف :

قبل أن تتم أي عملية نقل على الخط ، يجب ارسال اشارة حالة البدء تحدد بواسطة السيد ليعلم كل العبيد أن هناك معطيات سيتم ارسالها على الناقل ، كنتيجة لذلك جميع العبيد سوف تصغي الى المعطيات التسلسلية كتعليمات .

اشارة حالة البدء تتم عن طريق السيد وذلك بسحب الخط SDA الى المنطق المنخفض ، يتبع ذلك سحب SCL الى المنطق المنخفض

أما عند الانتهاء فإن السيد يقوم بإرسال اشارة الإيقاف ليعلم الجميع بأن عملية النقل قد تمت من اجل تحرير الناقل ليعود الى حالته الطبيعية المنطق العالي ، وأما اشارة الإيقاف هي تحرير الخط SCL يتبعه تحرير الخط SDA .

الشكل 49

ملاحظة :عندما يتم تحرير أي خط فإنه سيعود الى المنطق العالي بسب مقاومات الشد.

1.2.2. كيف يتم مخاطبة العبد المطلوب ؟

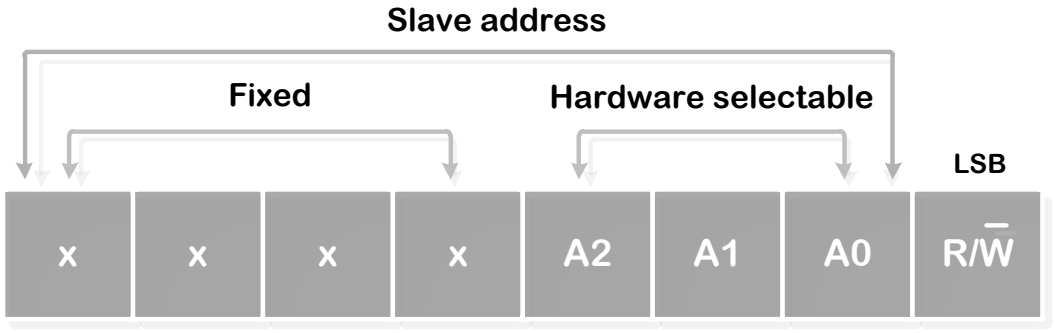
كل تجهيزة موصولة مع الناقل تكون معنونة بواسطة عنوان فريد مكون من سبع بتات أو عشر بتات ، عشر بتات ليس شائعا كثيرا لذلك لن نناقشه هنا .

البايت الأول الذي يتم ارساله بعد حالة البدء يعرف باسم بايت التحكم Control byte البتات السبعة الأولى(MSB) منه تحدد عنوان التجهيزة المطلوبة بينما البت الثامن (LSB) يحدد اتجاه المعطيات (R/W)

عندما يكون "صفر " في هذا البت فهذا يعني أن السيد سيقوم بكتابه (ارسال) معطيات الى عبد محدد ، وعندما يكون "واحد " فهذا يعني أن السيد سيقرأ (يستقبل) معطيات من عبد .

1.2.3. كيف نحصل على عنوان لتجهيزة ما :

بشكل قياسي أول اربع بتات تكون ثابتة ، أما البتات الثلاثة التالية يتم تحديدهم بواسطة الهاردوير hardware أي بواسطة ثلاثة أرجل من الشريحة (A0, A1, and A2) . و هذا يعطي امكانية تشغيل ثماني عبيد على نفس الناقل فمن أجل المنطق المنخفض يتم وصل هذه الأرجل الى الأرضي GND أما من أجل المنطق العالي يتم وصل هذه الأرجل الى VCC .

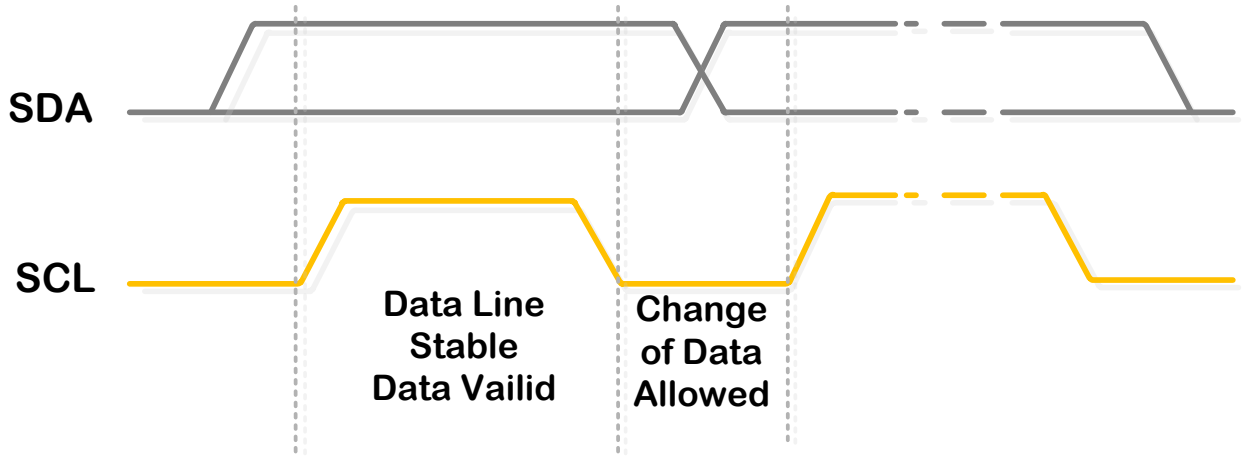


الشكل 50

عندما يتم ارسال بايت التحكم كل تجهيزة ستقارن البتات السبعة الأولى من بايت التحكم مع عنوانها ، وفي حال حدوث تطابق ستعتبر نفسها المقصودة وستستقبل وترسل ، وذلك حسب بت الاتجاه (البت الثامن) .

المعطيات سترسل على الخط SDA بدأ من البت الأكثر أهمية بينما خط الساعة سيولد نبضات التزامن المعطيات المرسلة عبر خط المعطيات ستعتبر مقبولة وصحيحة عندما يكون SLC على المنطق العالي لذلك فإن المعطيات يجب أن تكون مستقرة خلال فترة المنطق العالي للساعة

لذلك فإن حالة الواحد أو الصفر يمكن أن تتغير فقط عندما تكون الساعة على المنطق المنخفض



الشكل 51

Data is valid only during the High condition of SCL

مثلا : مقاطعة داخلية ، يمكن أن تمسك الخط SCL على المنطق المنخفض لتجبر السيد على الانتظار ارسال المعطيات يستكمل عندما يكون العبد مستعدا لاستقبال بايت آخر من المعطيات ولذلك يحرر خط الساعة .

ارسال المعطيات دائما ينتهي بإشارة حالة الايقاف المولدة من السيد على كل حال اذا كان السيد لا يزال يرغب باتصال على الخط ، يمكنه اعادة تولد اشارة حالة البدء . وعنونة تجهيزة اخرى بدون توليد اشارة حالة الايقاف .

بشكل عام ، المستقبل الذي تم عنونته سيقوم بتوليد نبضة عُلم (ACK).

نبضة عُلم ستأخذ مكانها بعد البت المعطيات الثامن لأي عملية ارسال ، بعد ذلك مباشرة فإن المرسل سيحرر خط المعطيات ليسمح للمستقبل بإرسال هذه النبضة . المستقبل سيقوم بدوره و سيقود خط المعطيات الى الصفر او المنطق المنخفض ليعلم بأن البايت قد وصل .

اذا لم يقم المستقبل بقيادة خط المعطيات الى الصفر ، معنى هذا أن البايت لم يصل وأن العملية احبطت .

اذا تم ارسال بايت التحكم (slave address + R/W bit) فإنه فقط التجهيزة المطلوبة ستستجيب وترسل نبضة عُلم. الموديول : MSSP (Synchronous Serial Port) الموجود في المتحكم PIC18F2550) يسمح بتحقيق الاتصال I2C من خلال قطبين من أقطابه.

تفاصيل التوابع لن نناقشها هنا لهذا الموديول وذلك لأن المترجم MikroC PRO يقدم لنا مكتبة خاصة للتعامل مع هذا الموديول لتحقيق الاتصال I2C .

Serial EEPROM (24LC512)

من نوع EEPROM من شركة ماكروشيب مع سعة (K x 8 512 Kbits64) هي عبارة عن ذاكرة تدعم اتصال مع I2C

مخطط يُظهر أرجل الدارة حيث أول أربع بتات من البتات السبعة الخاصة بالعنوان '1010' البتات الثلاثة التالية ، يتم تهيئتها بواسطة الأقطاب A0, A1, and A2 pins.

A0 :على المنطق العالي

A1-A2:على المنطق المنخفض

ستكون عنوان هذه الذاكرة على الشكل التالي '1010001'

هذا سيسمح لثماني تجهيزات بشكل أعظمي من نفس العنوان (النوع) لأن تتصل مع هذا الناقل .

1.1. كتابة بايت الى الذاكرة :

من أجل كتابة بايت ، فإن ذلك يتطلب بايتين للعنونة أي لتحدد واحد من 65536 موقع في الـEEPROM السيد سيرسل هذين البايتين بعد أن يرسل بايت التحكم .

الذاكرة تستجيب بأن سترسل نبضة عُلم بعد تلقيها لكل بايت بعد ذلك يقوم السيد بإرسال بايت لتم كتابته في الذاكرة وعند وصول هذه المعطيات ، الذاكرة ترسل نبضة عُلم . بعد ذلك ينهي السيد الاتصال بإشارة حالة الإيقاف .

لتقليل زمن دور كتابة تقدم الذاكرة ميزة صفحات الكتابة ، وهذا يسمح بشكل آني لكتابة حتى 128موقع متجاور . لكتابة صفحة كاملة تتم العملية بنفس طريقة كتابة بايت واحد لكن دون لكن بدون توليد اشارة حالة الإيقاف

السيد يقوم بإرسال حتى 127 بايت اضافي ستخزن بشكل مؤقت في صفحة البفر وسيتم كتابتها الى الذاكرة بعد ان يرسل السيد اشارة حالة الإيقاف .

1.2. عملية القراءة

يتم تهيئتها بنفس طريقة عملية الكتابة مع استثناء وحيد وهو بت الـ (R/W) من بايت التحكم سيكون 1 .

الذاكرة تسمح بثلاثة انواع من عمليات قراءة

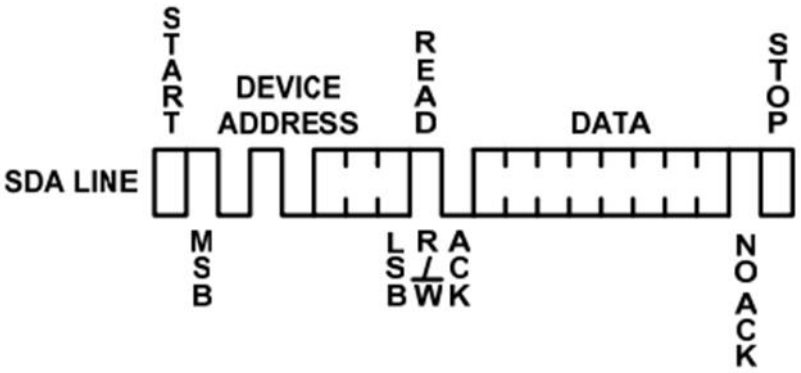
- ❖ قراءة العنوان الحالي Current address read signal
- ❖ القراءة العشوائية Random read operation
- ❖ القراءة المتسلسلة

1.2.1. قراءة العنوان الحالي Current address read signal

داخليا EEPROM تحوي عداد العنوان ، الذي يبقي(يحتفظ ب)على عنوان اخر موقع ذاكري تم الولوج اليه ، يزداد هذا العداد بمقدار 1 .

لذلك ، اذا كان العنوان السابق الذي ولج هو N ، فإن العنوان التالي للقراءة سيقرأ من الموقع N+1 .

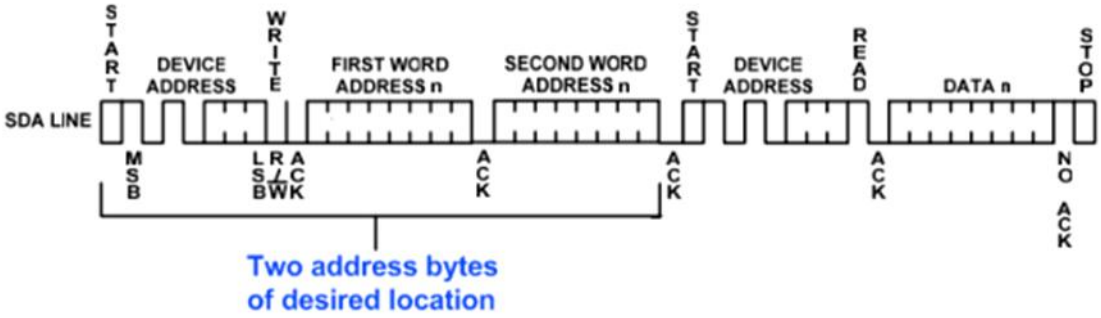
عند وصول بايت التحكم وعندما يكون البت 1 الخاص بالقراءة والكتابة ، الذاكرة تولد نبضة عُلم وترسل معطيات عنوانها الحالي السيد سوف يرسل نبضة لاعُلم (NO ACK) ثم سيرسل حالة ايقاف ، والذاكرة ستوقف الإرسال



Current address read signal

1.2.2. القراءة العشوائية Random read operation

الأسلوب العشوائي للقراءة يسمح للسيد بالوصول الى اي موقع ذاكرة لتحقيق هذا النوع من عمليات القراءة ، فإنه في البداية يجب أن يرسل كلمة العنوان ، يتم هذا بإرسال كلمة العنوان الى الذاكرة بنفس طريقة عملية الكتابة أي كما لو أننا نريد كتابة بايت ما (R/W bit set to '0') ،و بعد أن يتم ارسال كلمة العنوان السيد يعيد ارسال حالة البدء بعد نبضة العُلم ؟ هذه تنهي عملية الكتابة . بعد ذلك يقوم السيد بتوليد بايت التحكم مرة ثانية

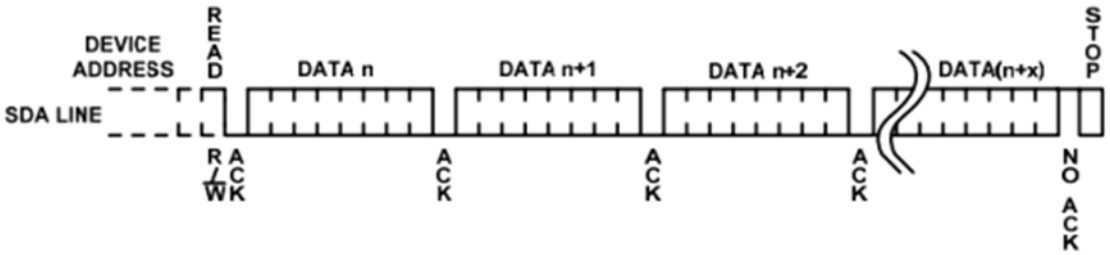


Random read operation

لكن سيكون (R/W bit set to a one) في هذه المرة ، السيد سوف لا يولد نبضة علم ، بعدها سيولد اشارة حالة لإيقاف والذي سيؤدي بالذاكرة الى ان توقف الإرسال.

1.2.3. القراءة المتسلسلة Sequential read operation:

القراءة المتسلسلة يتم تهيئتها اما بالعنوان الحالي أو بعنوان عشوائي بعد استقبال البايت الأول من الذاكرة ، السيد يولد نبضة علم بدلا من حالة الإيقاف تستخدم في حالة العنوان الحالي أو القراءة العشوائية، هذه العلم تدفع الذاكرة لإرسال كلمة العنوان المتسلسل التالي ، يتبع البايت الأخير المرسل الى المتحكم السيد ، السيد لن يرسل نبضة العلم وانما سيولد حالة الإيقاف

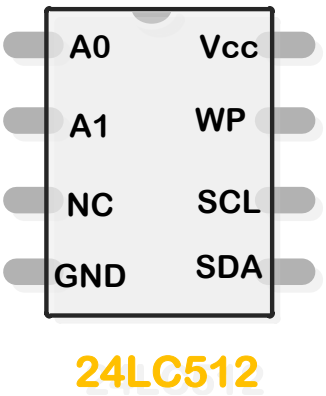


1.3. طريقة ربط ذاكرة خارجية مع المايكرو:

أولا :شكل الذاكرة وبنيتها الداخلية :الذاكرة تتألف من صفحة واحدة أو أكثر من صفحة وكل صفحة تتألف من 16 بايت يمكن أن تخزن ضمنها المعلومات

أقطاب الذاكرة :

SCL :قطب الساعة التسلسلية



الشكل 52

عند الجبهة الصاعدة يتم ادخال المعطيات

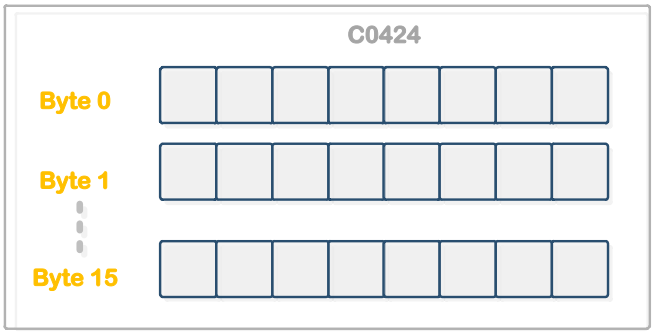
عند الجبهة الهابطة يتم قراءة المعطيات

خط التزامن مشترك بين الجميع .

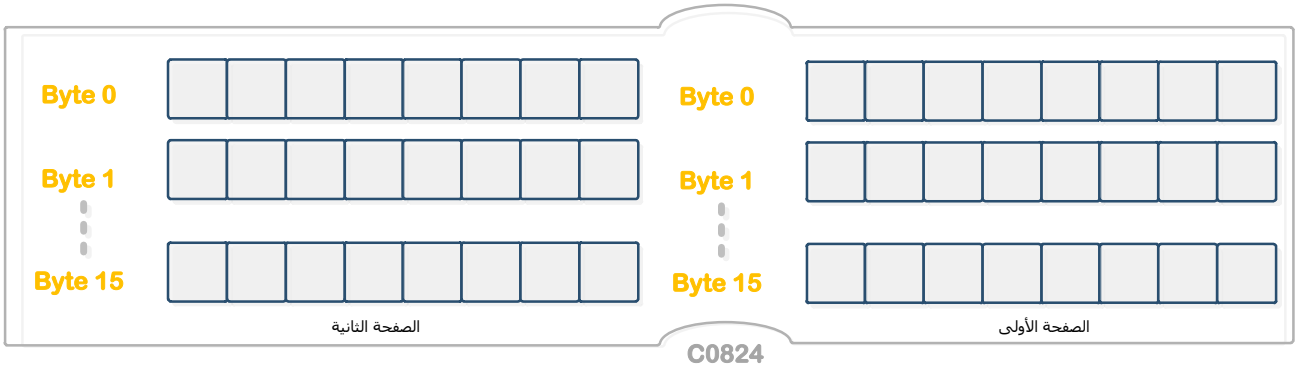
SDA :قطب المعطيات التسلسلي

هو عبارة عن قطب ثنائي الاتجاه يستقبل المعطيات من المايكرو وبعد كل بايت يستقبله يرسل بت اسمه Acknowledge تم تمثيله ACK وهو عبارة عن صفر منطقي يدل على أنه تم استقبال البايت بنجاح .

سنرى كيف أن للذاكرة الأولى صفحة وللثانية صفحتين و كأنها دفتر :



الشكل 53



الشكل 54

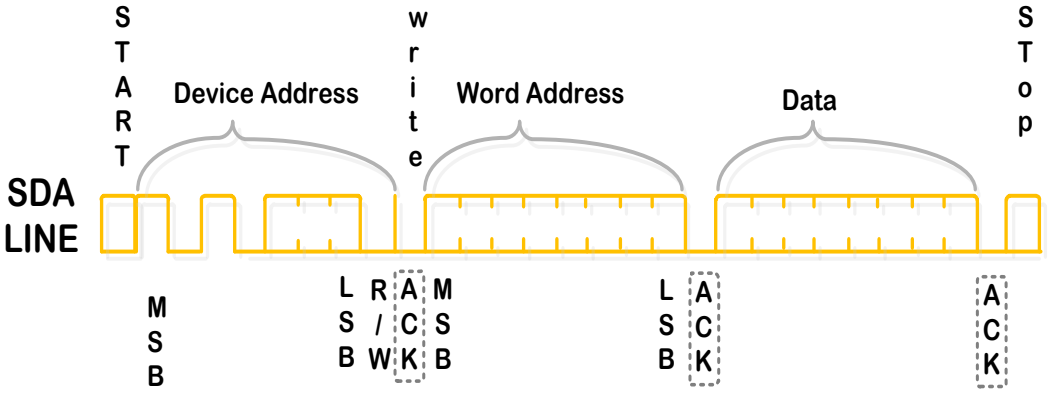
سنقوم في هذه الشرح باستخدام الذاكرة 24C02 & PIC18F452

تستخدم خطين نقل(غير القطبين GND, VCC) فقط. الأول للمعطيات والثاني للclock .

ملاحظة :لا يمكن أن يكون هناك أكثر من سيد ويمكن أن يكون هناك أكثر من عبد ، لا يستطيع السيد أن يتخاطب مع كل العبيد بنفس الوقت ، العبيد لا يمكن أن تتخاطب مع بعضها أبدا.

1.4. تسلسل إجراء كتابة بايت واحد على الذاكرة :

المايكرو هو السيد و الذاكرة هي العبد .



الشكل 55

على خط المعطيات تحدث الدراما التالية من أجل كتابة بت واحد :

المايكرو يرسل : بت البدء START

المايكرو يرسل : بايت عنوان الشريحة

الذاكرة ترسل : صفر منطقي لتدل على أنها استلمته .

المايكرو يرسل :كلمة العنوان

الذاكرة ترسل : صفر منطقي.

المايكرو يرسل : المعطيات .

الذاكرة ترسل : صفر منطقي

المايكرو يرسل : بت البدء STOP نهاية العملية

DEVICE ADDRESS : عناوين الشرائح

هو عبارة عن بايت يتم ارساله الى الذاكرة ليحدد هل ستتم عملية قراءة أو كتابة ، ولتحديد الصفحة المطلوبة .

1K/2K	1	0	1	0	A2	A1	A0	R/W
4K	1	0	1	0	A2	A1	P0	R/W
8K	1	0	1	0	A2	P1	P0	R/W
16K	1	0	1	0	P2	P1	P0	R/W

الشكل 56

نبدأ بالبت الأول R/W :إذا أردنا الكتابة نضع صفر في هذا البت

أما إذا أردنا القراءة من الذاكرة نضع واحد في هذا البت .

AX :عنونة الشرائح في حال استخدام عدة ذواكر تستخدم لتحديد الشريحة المطلوب التعامل معها.

P0,P1,P2 :الذاكرة تكون مقسمة الى صفحات

P0 : الصفحة رقم صفر في الذاكرة هذا بالنسبة لذاكر ة C0424 ولا يوجد بها الا صفحة واحدة للكتابة .

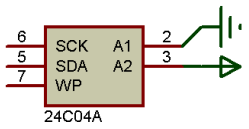
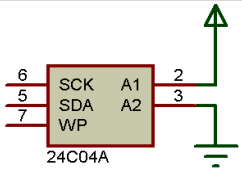
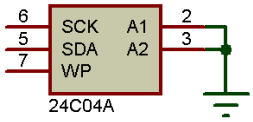
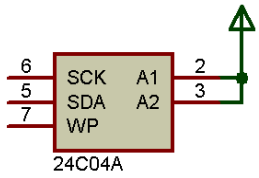
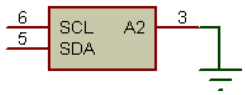
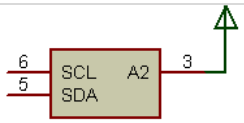
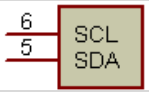
و P1 : معناها الصفحة الثانية بالذاكرة C0824 لأنها تحتوي على صفحتين الكتابة P0 & P1 .

P2 : الصفحة الثالثة بالنسبة للذاكرة C01624 لأنها تحوي على ثلاث صفحات .

أما البايتات الأربعة الباقية فهي ثابتة 1010 وتساوي الى A بالست عشري

مما سبق يمكننا صياغة الجدول التالي وهناك احتمالات اخرى كثيرة :

من أجل الكتابة على الذاكرة

نوع الذاكرة	DEVICE ADDRESS	تمثيل البايت ثنائيا مع الرسم Software	تعليمة MikroC Hardware
4K	A4	1 0 1 0 0 1 0 للقراءة 1 أو 0 حسب الصفحة كما نلاحظ في الصورة المجاورة A1 موصولة الى الأرضي لذلك تم وضع 0 في البت A1 في الصورة أعلاه في البت المظلل . أي يجب أن يتطابق الهاردوير مع السوفت وير	
4K	A2	MSB LSB 4K 1 0 1 0 0 0 1 0 كتابه على الصفحة الأولى والوحيدة	
4K			
4K			
8K	A2	MSB LSB 4K 1 0 1 0 0 0 1 0 كتابه على الصفحة الأولى	
8K	A4	8K 1 0 1 0 0 1 0 0 كتابه على الصفحة الثانية	
16K	A2	16K 1 0 1 0 0 0 1 0	
16K	A4	16K 1 0 1 0 0 1 0 0	
16K	A8	16K 1 0 1 0 1 0 0 0	

الجدول 12

من الجدول السابق لدينا ثلاث خانات A0,A1,A2 وبذلك لدينا ثمان احتمالات بدء من 000 انتهاء بـ 111 ومنه نستنتج أنه يمكننا وصل ثمانية ذواكر 2K مع المايكرو.

وبالنسبة لذواكر 4K لديها خطي عنونة ومنه فإن عدد الاحتمالات هي أربعة (00,01,10,11) أي يمكن وصل أربع شرائح فقط.

بالنسبة لـ 8K يمكن وصل ذاكرتين مع المايكرو .

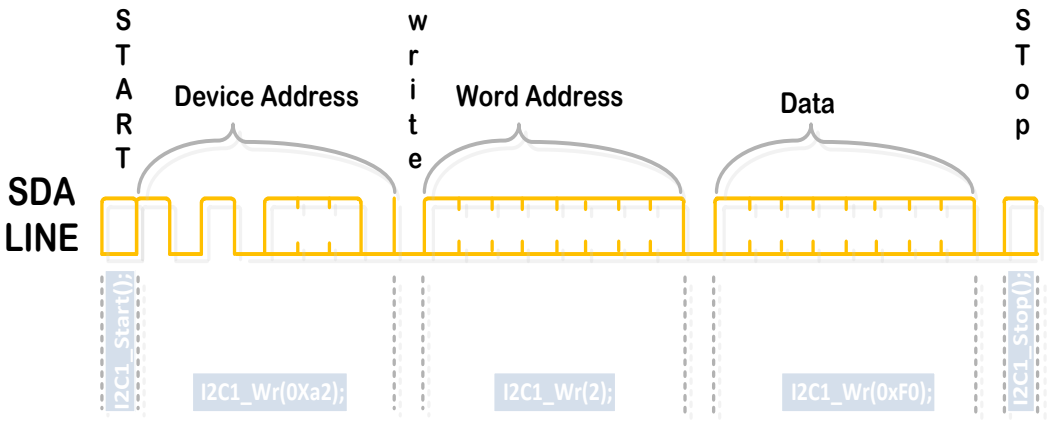
بالنسبة لـ16K يمكن وصل ذاكرة واحدة فقط .

WORD ADDRESS :كلمة العنوان

لكل حجرة ذاكرية عنوان يعبر عنه برقم ست عشري

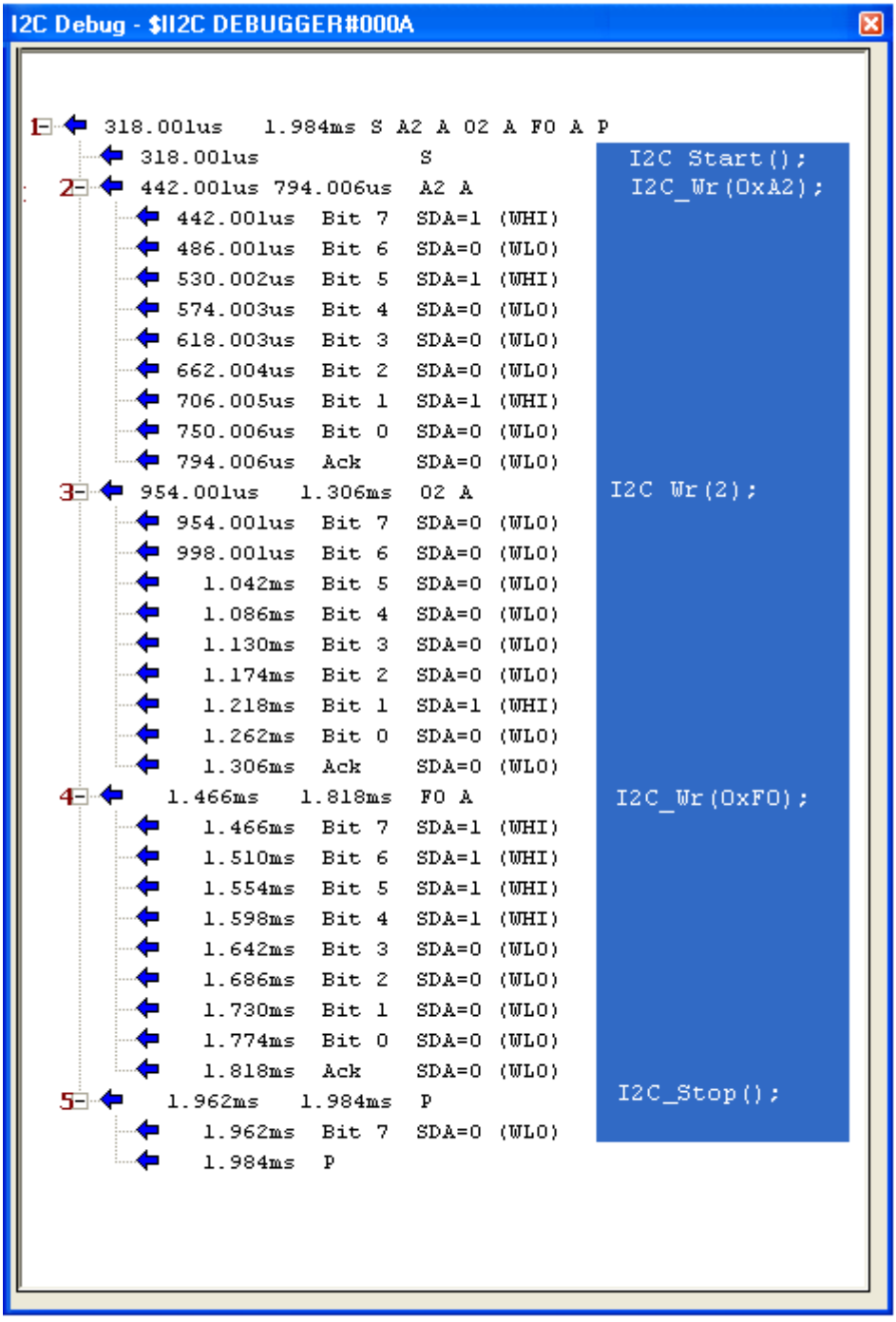
لذلك و للوصول الى حجرة معينة يجب ارسال عنوانها وبذلك نخبر الذاكرة (يا ذاكرة نريد أن نتعامل مع الحجرة ذات الرقم كذا)

DATA : المعطيات المراد كتابتها على الذاكرة



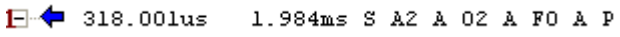
الشكل 57

وهذه طريقة التحليل بالبروتس :



الشكل 58

السطر الأول من الصورة :



وهويشرح العملية من الألف الى الياء

زمن البدء : 318.001us ، زمن الإنتهاء: 1.98ms

بت البداية : S ،

DEVICE ADDRESS: A2 ،

Acknowledge: A ،

WORD ADDRESS:02 ،

Acknowledge:A ،

بايت المعطيات : F0 ،

Acknowledge:A ،

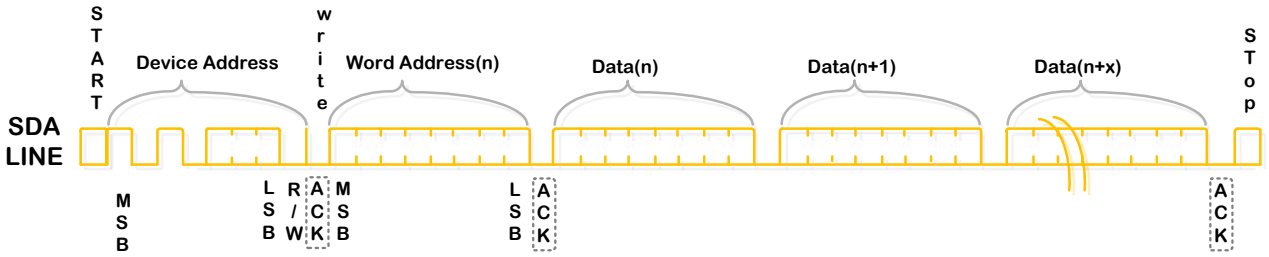
STOP : P

وبقية السطور هي تفصيل العملية وقد تم الشرح سابقا

1.5. تسلسل إجراء كتابة أكثر من بايت على الذاكرة:

يمكننا بهذه الطريقة ملئ صفحة كاملة أي كتابة 16 بايت كحد أعظمي قبل أن نرسل نبضة التوقف وفي حال أننا تجاهلنا نبضة التوقف وأرسانا البايت السابع عشر فسيقوم هذا البايت الدخيل بالاستقرار مكان البايت الأول وبالمثل البايت الثامن عشر سيستقر مكان البايت الثاني وهكذا .

إذا يكتفي بهذه الطريقة أن نرسل عنوان أول بايت Word Address نريد أن نخزن فيه بايت المعطيات .



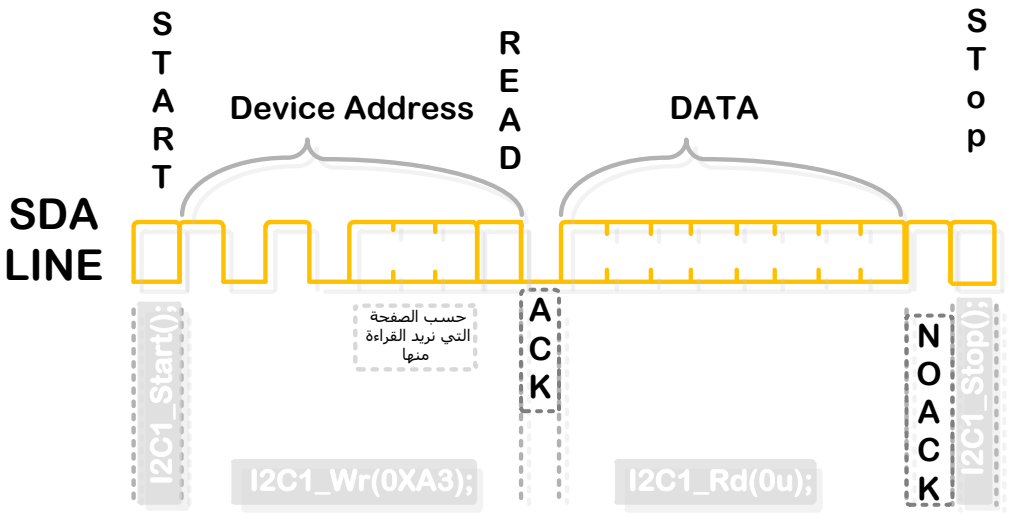
الشكل 59

1.6. طريقة القراءة من الذاكرة :

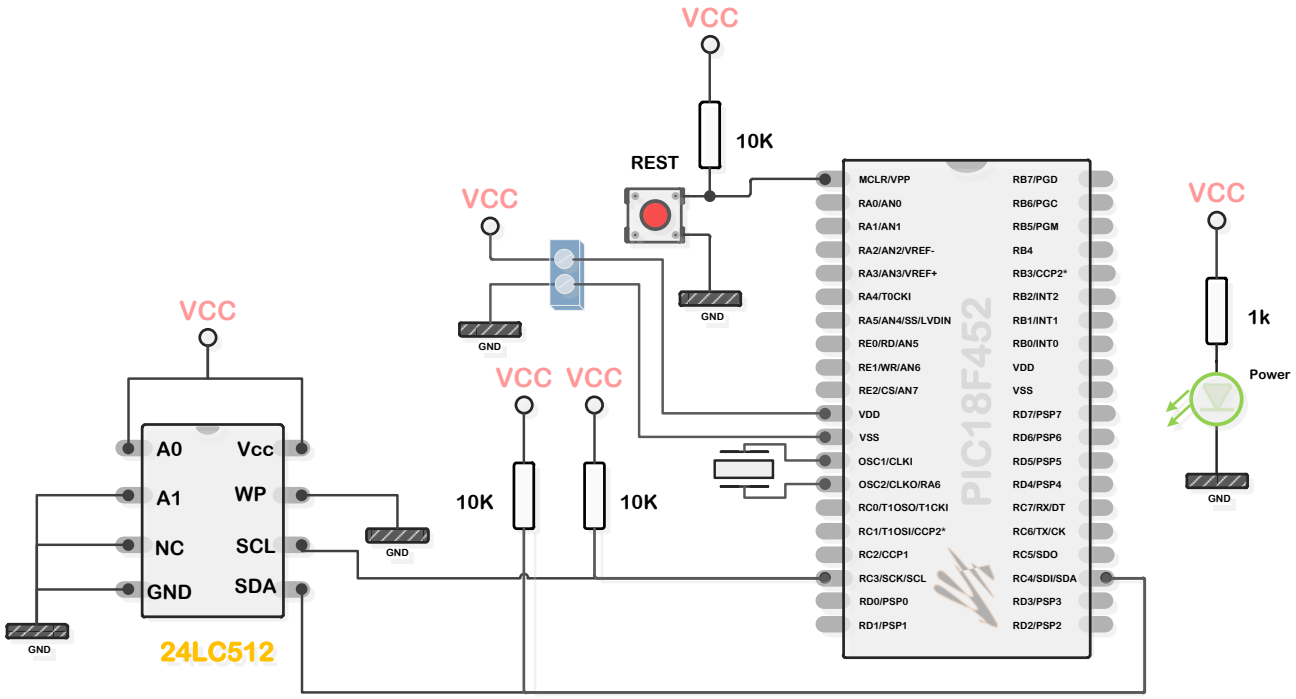
في الواقع يوجد ثلاثة طرق

1.6.1. الطريقة المباشرة :

لنفترض أن شخصا ما سيقوم بقراءة كتاب عدة مرات فإنه سيقرأ من أول كلمة في الكتاب حتى أخر كلمة ثم يعود الى اول كلمة لكنه ربما يضطر لترك القراءة عندها يقوم بوضع خط صغير تحت الكلمة التي وصل اليها ليتمكن من مواصلة القراءة فيما بعد ، مغزى هذا أن في الذاكرة يوجد مؤشر داخلي يشير الى البايت الذي تم التعامل معه آخر مرة.



الشكل 60



الشكل 61

I2C temperature sensor (DS1631)

DS1631 : وهو حساس حرارة رقمي مصنع من قبل شركة Dallas لأنصاف النواقل (حاليا MAXIM) خرج الحساس عبارة عن عدد بعرض 9 bit-، 10-bit، 11-bit، 12-bit حسب الاختيار .

DS1631 : ذو دقة قياس ±0.5°C في المجال من 0°C الى +70°C

DS1731 : ذو دقة قياس ±1°C وذلك ضمن المجال(+85°C -10°C)

يقوم بعملية القياس مباشرة عند وصول التغذية

مجال العمل من -55°C الى +125°C (-67°F to +257°F)

عملية القياس لا تحتاج اي مكونات اضافية

الخرج سيكون بعرض 9 ,10,11,12 بت حسب الاختيار

مجال جهد تغذية عريض يتراوح من (+2.7V to +5.5V)

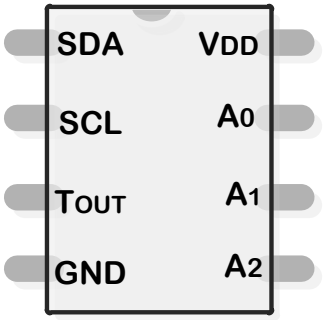
الزمن الأعظمي لعملية التحويل 750ms

اعدادات الحساس يمكن حفظها بذاكرة غير متطايرة(NV)

درجة الحرارة يمكن قراءتها كتابتها باستخدام سلكين فقط SDA , SCL

متاح مع µSOP Packages أو 150mil SO package كما في الشكل التالي :

جميعها له ثماني أرجل



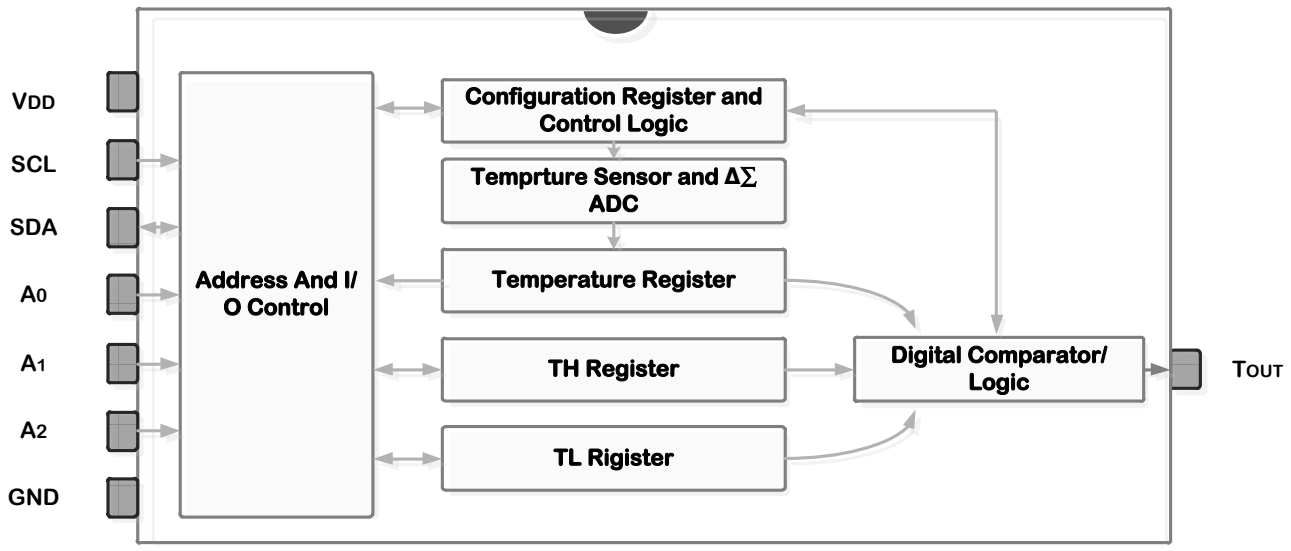
الشكل 62

في الراوترات والسوييتشات

في الخليوي

الأجهزة المحمولة

DS1713/DS1613		
Pin #	**Name**	**Description**
1	SDA	رجل المعطيات من أجل الناقل I2C
2	SCL	دخل الساعة من أجل الناقل I2C
3	TOUT	Thermostat Output Pin. Push-Pull
4	GND	الأرضي
5	A2	رجل من أجل العنوان
6	A1	رجل من أجل العنوان
7	A0	رجل من أجل العنوان
8	VDD	جهد التغذية +2.7V to +5.5V



الشكل 63

من المخطط نلاحظ وجود المقارن الذي يقارن TR مع TH,TL .

من المخطط السابق نلاحظ وجود أربع مسجلات في هذا الحساس:

REGISTER SUMMARY			
REGISTER NAME (USER ACCESS)	**SIZE (BYTES)**	**MEMORY TYPE**	**REGISTER CONTENTS AND POWER-UP STATE**
Temperature (Read Only)	2	SRAM	Measured temperature in two's complement .format Power-up state: -60ºC (1100 0100 0000 0000)
TH (Read/Write)	2	EEPROM	Upper alarm trip point in two's complement format. Factory state: 15ºC (0000 1111 0000 0000)
TL (Read/Write)	2	EEPROM	Lower alarm trip point in two's complement .format Factory state: 10ºC (0000 1010 0000 0000)
Configuration (Various bits are	1	SRAM, EEPROM	Configuration and status information. Unsigned data.

Read/Write and Read Only—See Table 5)			6 MSbs = SRAM 2 LSbs (POL and 1SHOT bits) = EEPROM Power-up state: 100011XX (XX = user defined)

الجدول 13

1.1. عملية قياس درجة الحرارة OPERATION—MEASURING TEMPERATURE

يقوم الحساس بقياس درجة الحرارة ثم عن طريق المحول (ADC) يتم تحويل درجة الحرارة الى قيمة رقمية وهذه القيمة يمكن ان تكون بعرض 9 أو 10 أو 11 أو 12–bit وهذا يسمح بالتحكم بدقة القياس (هناك مسجل يسمح لنا باختيار عرض درجة الحرارة 9 أو 10 أو 11 أو 12–bit)

- ❖ الحساسان ,DS1631 و DS1731 سيبقيان في حالة الخمول ولن يقوما بعملية التحويل الا بعد تلقيهما أمر بدء التحويل من السيد (ويدعى هذا الأمر بـ T command)
- ❖ بينما الحساس DS1631A سيبدأ مباشرة بالتحويل عند وصول التغذية وسيعمل بالنمط 1SHOT.

1.2. أنماط العمل للحساسات الثلاثة

لديها نمطين للعمل :

1.2.1. continuous-conversion mode :

DS1631A سيبدأ التحويل بعد عند وصول التغذية مباشرة ، بينما DS1631 and DS1731 ستبدأ التحويل بعد تلقي أمر البدء T command ، وجميع الحساسات ستستمر بعملية بتحديث درجة الحرارة ولن تتوقف الا عند أمرها بالتوقف عن طريق الأمر Stop Convert T command عندها وفقط ستعود الى حالة الخمول ويمكن اعادة تشغيل الحساس في أي وقت باستخدام الأمر Convert T command

1.2.2. one-shot mode :

تقوم بعملية تحويل واحدة بعد تلقيها أمر التحويل

DS1631A :يقدم نتيجة تحويل واحدة عند تغذيته ، أما DS1631 and DS1731 يقدم نتيجة تحويل واحدة بعد تلقيه T command ومن أجل جميع الحساسات في هذا النمط عندما تنتهي من عملية التحويل ستود الى حالة الخمول وستبقى في هذه الحالة الى أن تأتيها أمر تحويل جديد T command

ويتم اختيار نمط العمل عن طريق البت 1SHOT bit من المسجل CONFIGURATION REGISTER

نتيجة التحويل ستكون بعرض 9, 10, 11, or 12 bits وذلك حسب دقة المحول 0.5°C, 0.25°C, 0.125°C, and 0.0625°C على الترتيب وعليك ملاحظة أن زمن التحويل سيتضاعف مع كل بت اضافي

بعد كل عملية تحويل فإن النتيجة ستكون رقم مكون من 16–bit سيخزن في مسجل مكون من بايتين يدعى temperature register ، بت الإشارة (S) sign bit يحدد اذا كانت الحرارة موجبة أو سالبة ، من أجل درجة الحرارة الموجبة S = 0 ومن أجل درجة الحرارة السالبة سيكون S = 1 . الأمر Read Temperature command يسمح لنا بقراءة هذا المسجل ومعرفة درجة الحرارة .

12-BIT RESOLUTION TEMPERATURE/DATA RELATIONSHIP

TEMPERATURE(C°)	DIGITAL OUTPUT(BINARY)	DIGITAL OUTPUT(HEX)
+125	**0111 1101 0000 0000**	7D00h
+25.0625	0001 1001 0001 0000	1910h
+10.125	0000 1010 0010 0000	0A20h
+0.5	**0000 0000 1000 0000**	0080h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1000 0000	FF80h
-10.125	1111 0101 1110 0000	F5E0h
-25.0625	1110 0110 1111 0000	E6F0h
-55	1100 1001 0000 0000	C900h

الجدول 14

TEMPERATURE REGISTER FORMAT

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
MS Byte	S	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LS Byte	2^-1	2^-2	2^-3	2^-4	0	0	0	0

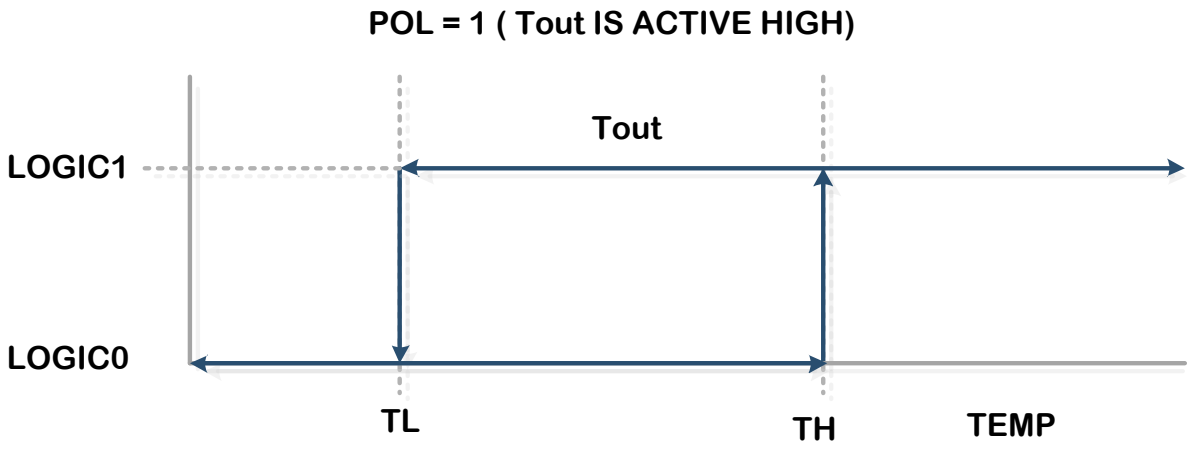
الشكل 64

البتات من 0 الى 3 تُقرئ صفر دائما . وعند تهيئة نتيجة القياس بعرض 12-bit فإن البتات هي أول 12 بت أكثر أهمية أي البتات (من 15 الى 4) هي التي تحوي على نتيجة القياس .

وعند تهيئة نتيجة القياس بعرض 11-bit فإن البتات التي تعنينا هي أول 11 بت أكثر أهمية أي(البتات من 15الى 5) حيث هي التي تحوي على نتيجة القياس .

OPERATION—THERMOSTAT FUNCTION

الرجل TOUT : يتم تحديثها بعد كل عملية تحويل ، وتعتمد قيمتها على نتيجة مقارنة درجة الحرارة الرقمية مع القيم الحدية العليا والدنيا حيث تبقى على قيمتها الحالية حتى اكتمال عملية التحويل التالية ، وعندما تتساوى درجة الحرارة الرقمية أو تتجاوز القيمة المخزنة في مسجل القيمة العليا (TH) فإن TOUT ستصبح فعالة وستبقى فعالة حتى تنخفض درجة الحرارة الى أقل من القيمة المخزنة في (TL) انظر الشكل التالي، ويتم تفعيل TOUT برمجيا عن طريق البت (POL) في المسجل configuration register



الشكل 65

المستخدم يقوم بتخزين القيمة العليا والدنيا في المسجل TH و المسجل TL كالصيغة التالية متضمن اشارة (S) MSb (bit 15) ويتم تحديدها عن طريق R0 و R1 (من المسجل configuration register) بشكل يراعي عرض النتيجة الذي تم اختياره .

RESOLUTION CONFIGURATION

R1	R0	RESOLUTION (BIT)	CONVERSION TIME (MAX)
0	0	9	93.75ms
0	1	10	187.5ms
1	0	11	375ms
1	1	12	750ms

الجدول 15

مثلا : اذا كان عرض النتيجة 10-bit فإن بتات TL،TH من 0 الى 5 تقرأ كأصفار ودرجة الحرارة الرقمية تقارن من أول عشر بتات أكثر أهمية 10 MSbs من المسجلين TL , TH

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
MS Byte	S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LS Byte	2^-1	2^-2	2^-3	2^-4	0	0	0	0

الشكل 66

المسجلين TL،TH من نوع EEPROM; لذلك فإن هذين المسجلين لا يفقدان محتوياتهما بفصل التغذية ، ويتم تخزين القيم في هذين المسجلين عن طريق أمر Access TH و أمر Access TL وقبل أن نقوم بتغيير محتوياتهما يجب ايقاف عملية التحويل عن طريق الأمر Stop Convert T command وذلك في حال كان الحساس في وضع التحويل المستمر . وفي حال أننا لم نستخدم ميزة المقارنة مع وتم ايقافها من البت (POL) فإننا نستطيع استخدام المسجلين للأغراض العامة

MSB	bit6	bit5	bit4	bit3	bit2	bit1	LSB
DONE	THF	TLF	NVB	R1	R0	POL*	1SHOT*

* NV (EEPROM)

الشكل 67

هناك ميزة أخرى وهما العلمين TLF،THF حيث يقدمان سجل عن درجة حرارة الطقس . يكون هذين العلمين عند التشغيل أصفار ، وفي حال تجاوزت درجة الحرارة القيمة المخزنة في المسجل TH فإن العلم THF يفعل ، كذلك في حال انخفاض درجة الحرارة عن القيمة المخزنة في TL فإن TLF سيفعل . وسيبقى هذا العلم مفعل حتى يقوم المستخدم بتصفيره ، أو عند اعادة وصل التغذية .

CONFIGURATION REGISTER

نستطيع عن طريق هذا المسجل التحكم بجميع خيارات الحساس والوصول الى جميع ميزاته مثل :

التحكم بعرض نتيجة التحويل ، TOUT ، نمط العمل ، كذلك يقدم معلومات للمستخدم عن تقدم عملية التحويل ، نستطيع قراءة محتويات هذا المسجل عن طريق أمر الوصول Access Config command

وعند الكتابة الى هذا المسجل فإن عمليات التحويل يجب أن تتوقف عن طريق Stop Convert T command في حال كان الحساس يعمل في نمط التحويل المستمر ، لاحظ أن البتين POL and 1SHOT من نوع EEPROM وبالتالي يمكن برمجتها مسبقا حسب الرغبة ، أما البتات الباقية فهي من نوع SRAM

DONE—Temperature Conversion Done (Read Only)	الحالة الافتراضية عند بدء التشغيل = 1 DONE = 0 : عملية التحويل قيد التشغيل DONE = 1 :عملية التحويل اكتملت
THF—Temperature High Flag (Read/Write)	الحالة الافتراضية عند بدء التشغيل = 0 THF = 0 : درجة الحرارة المقاسة لم تتجاوز القيمة المخزنة في المسجل TH منذ بدء عمل الحساس أي منذ وصول التغذية اليه TLF = 1 : في نقطة ما منذ بدء تشغيل الحساس فإن درجة الحرارة أصبحت أقل من القيمة المخزنة في TL قيمة TLF ستبقى 1 حتى يتم تصفيرها من قبل المستخدم أو عند اعادة التشغيل ،
NVB—NV Memory Busy (Read Only)	الحالة الافتراضية عند بدء التشغيل = 0 NVB = 1 : الكتابة الى الذاكرة EEPROM memory قيد الكتابة NVB = 0 : الذاكرة EEPROM غير مشغولة
R1—Resolution Bit 1 (Read/Write)	الحالة الافتراضية عند بدء التشغيل = 1
R0—Resolution Bit 0 (Read/Write)	الحالة الافتراضية عند بدء التشغيل = 1

POL*—TOUT Polarity (Read/Write)	الحالة المصنعية 0 لكن هذا البت من نوع EEPROM لذلك فهو يحتفظ بآخر قيمة اسندت اليه POL = 1 : TOUT تفعل عند المنطق العالي POL = 0 : TOUT تفعل عند المنطق المنخفض
1SHOT*—Conversion Mode (Read/Write)	الحالة عند بدء التشغيل : يأخذ آخر قيمة اسندت اليه الحالة المصنعية 0 لكن هذا البت من نوع EEPROM لذلك فهو يحتفظ بآخر قيمة اسندت اليه 1SHOT = 1 : النمط One-Shot ، يقوم بعملية تحويل واحدة ثم يعود الى وضع الخمول SHOT = 0 : النمط المستمر ، ستبدأ التحويل بعد تلقي أمر البدء T command وستستمر بتحديث درجة الحرارة حتى تتلقى أمر الإيقاف.

الجدول 16

الفصل الثامن

1-Wire protocol

يتميز هذا البرتوكول أنه يحتاج الى سلك واحد فقط بالإضافة الى الأرضي ،أي بالنسبة للمتحكم فإنه يلزمنا فقط رجل واحدة I/O لتحقيق اتصال مع تجهيزة Dallas 1-Wire .

يمكن أن يتحقق هذا البروتوكول عن طريق :

- Software فقط
- طريق موديول U(S)ART

يعتبر هذا البروتوكول ذو سرعة منخفضة ويعمل بسرعتين :

- Standard 15 kbps
- Overdrive : 125kbps

وهو يعتمد على الاتصال التسلسلي غير المتزامن ، يمكن وصل تجهيزة أو أكثر على نفس الخط لكن سيد واحد فقط master يمكن وصله الى الخط . يكو الخط في حالة اللاعمل idle على المنطق العالي وذلك بسبب مقاومة الشد وأي تجهيزة موصولة مع الخط يجب أن تكون قادرة على سحب الخط الى المنطق المنخفض . الإشارات على الخط one wire مقسمة الى slots زمن كل slot يساوي 60us . ويتم ارسال بت معطيات واحد في الـ slot ، السيد يقوم بتهيئة جميع الاتصالات على الخط ، هناك خمسة أوامر أساسية لتحقيق اتصال البرتوكول one Wire :

1.1. “Write 1” signal

يقوم السيد بسحب الخط الى المنطق المنخفض لمدة من 1 الى 15µ ثم يقوم بعدها بتحرير الخط لبقية الـ Slot

الشكل 68

.

1.2. “Write 0” signal

يقوم السيد بسحب الخط لفترة 60µ على الأقل لكن لا تتجاوز 120µ

الشكل 69

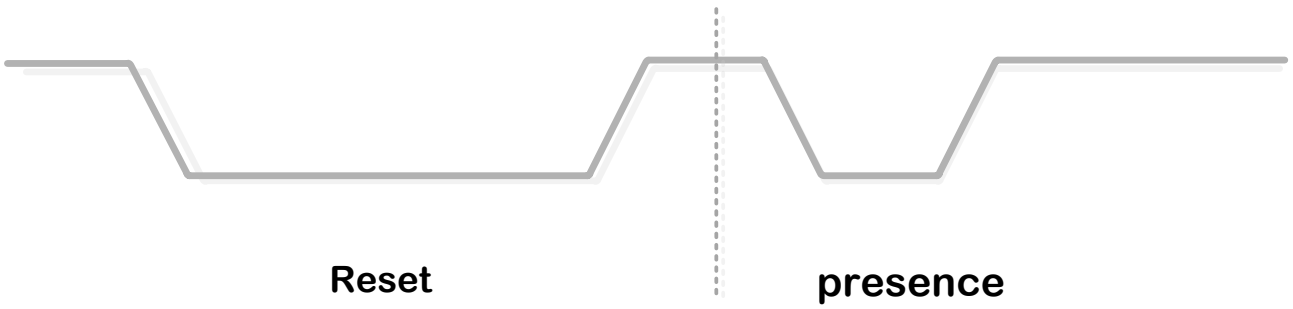
1.3. “Read” signal

يقوم السيد بسحب الخط الى المنطق المنخفض لمدة من 1 الى 15µ. التجهيزة أو التابع تقوم بمسك الخط على المنطق المنخفض اذا أرادت أن ترسل " 0 " أما اذا ارادت أن ترسل "1 " تقوم ببساطة بتحرير الخط

الشكل 70

1.4. “Reset/Presence” signal

السيد يقوم بسحب الخط مدة 480µs أو 8slots الى المنطق المنخفض بعد ذلك يقوم بتحريره . هذه الفترة الطويلة تدعى “Reset” فإذا كان هناك أي تجهيزة على الخط فإنها تقوم بالإستجابة وسحب الخط الى المنطق المنخفض لمدة 60µ طبعا بعد أن يقوم السيد بتحرير الخط



الشكل 71

الفصل التاسع

Recommended Standard-232

1.1. بروتوكول الاتصال التسلسلي (Recommended Standard-232) :

وجد هذا المعيار الضوء سنة 1962 من قبل الشركة الإلكترونية الأمريكية (EIA) حيث تم في هذا المعيار توصيف مستويات الجهود والأزمنة والبروتوكولات والوظائف أي توصيف كامل فيزيائيا وكهربائيا .

الشكل 72

وهو معيار يستخدم لنقل المعطيات بشكل تسلسلي ويعمل بنمطين :

- متزامن :

_وهو مستخدم مع الموصل 25-pin

_يحتاج الى خط مستقل لتأمين نبضات الساعة بين المرسل والمستقبل

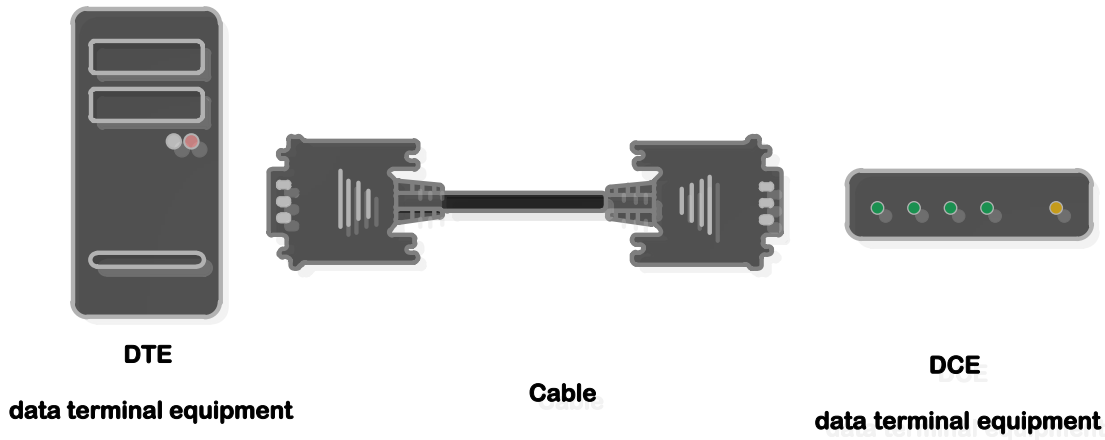
- غير متزامن :

_هذا النمط هو الأكثر استخداما وهو شائع جدا ويستخدم مع الموصل 9-pin (COM PORT)

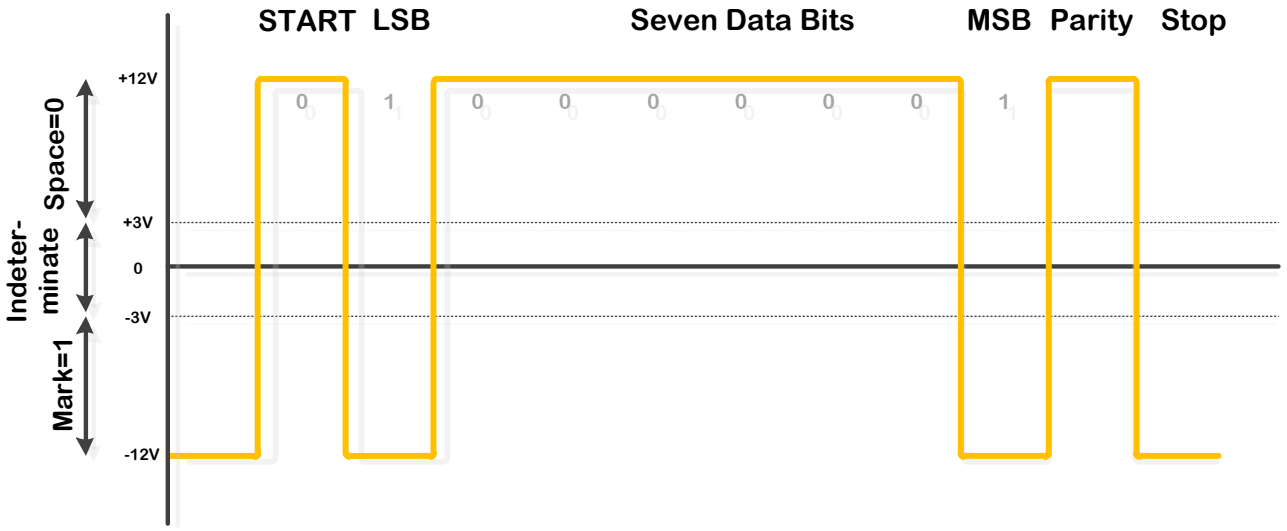
_لا يحتاج الى خط مستقل لتأمين نبضات الساعة بين المرسل والمستقبل .

البروتوكول RS232 الذي يقوم بإرسال المعطيات بت_بت وبطريقة معينة .و هذا البروتوكول يستخدم فقط سلكين (طبعا بالإضافة للأرضي)من أجل نقل المعطيات في كلا الاتجاهين (الإرسال و الاستقبال).الجهاز المستقبل تكون الرجل TX في حالة IDL عند 1 منطقي وتدعى MARK. المعطيات تبدأ بإرسال عندما تصبح هذه الرجل 0 منطقي وتدعى Space. البت الأول يرسل من خلاله بت البداية Start وهو صفر منطقي. بعد ذلك يتم ارسال بتات المعطيات وعددها 7 أو 8 بعدها يرسل بت وهو اختياري أي يمكننا عدم ارساله optional parity، البت الأخير يتم ارساله عند المنطق 1 وهو بت التوقف .

عادة يكون عدد البتات في الإطار الواحد هو 10 بتات حيث يحتوي الإطار على بت البداية + 8 بتات للمعطيات + بت التوقف ،(بدون البت parity)



الشكل 73



الشكل 74

خانة Parity Bit : وتستخدم فقط من أجل اكتشاف الخطأ لكن لا تستطيع تصحيحه تاخذ احدى الحالتين :

Even (زوجي) :ويتم وضع قيمتها بحيث تبقي أو تجعل عدد الواحدات في الإطار المرسل عددا زوجيا

Odd (فردي) : ويتم وضع قيمتها بحيث تبقي أو تجعل عدد الواحدات في الإطار المرسل عددا فرديا

والجدول التالي يعطي العديد من الأمثلة :

Example for Parity bit			
Parity bit	**Even**/ Odd	**BIN**	**ASCII**
0	Even	01000001	**A**
1	Odd	01000001	**A**
0	Even	01010011	**S**
1	Odd	01010011	**S**
1	Even	01000011	**C**

الجدول 17

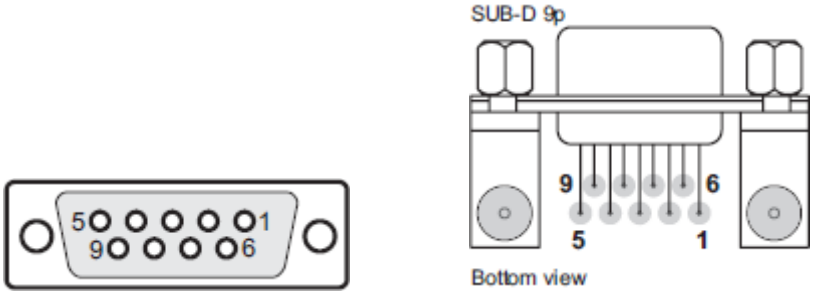
أشكال بوابة RS-232 :

DB9

RJ-45 (غير مستخدم)

DB25 (غير مستخدم)

DB9 وهو يعرف باسم منفذ الـ COM في الحاسب الشخصي والشكل التالية يوضح توزع أرجل هذه البوابة وهو يصلح للاستخدام مع الإرسال غير المتزامن فقط ولا يمكن استخدامه مع المتزامن لغياب الإشارات اللازمة ، ويمكن أن يستخدم لمسافة تصل الى 15 متر.



الشكل 75

والجدول التالي يوضح تسمية وعمل كل رجل من أرجل هذه البوابة :

RS232 DB-9 Connector Pin Out			
Description	**Signal Description**	**Signal Name**	**Pin #**
	Carrier Detect	CD	**1**
قطب الاستقبال	Receive Data	RXD	**2**
قطب الإرسال	Transmit Data	TXD	**3**
	Data Terminal Ready	DTR	**4**
الأرضي	Signal Ground / Common	GND	**5**
	Data Set Ready	DSR	**6**
	Request To Send	RTS	**7**
	Clear To Send	CTS	**8**
	Ring Indicator	RI	**9**

الجدول **18**

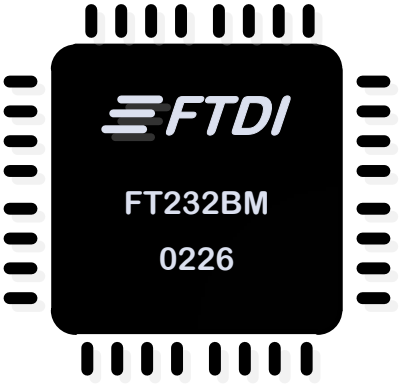
من الجدول الساب نحتاج فقط ثلاثة أقطاب لتقيق الاتصال غير المتزامن :

9-pin Connector		
Signal Description	**Signal Name**	**Pin #**
Receive Data	RXD	**2**
Transmit Data	TXD	**3**
Signal Ground / Common	GND	**5**

الجدول **19**

بالفترة قررت بعض شركات المصنعة للوحة الأم الاستغناء عنه واستخدام منافذ USB عوضا عنه لكن مازال بالإمكان استخدام وصلات جاهزة تقوم بتحويل بروتوكول الـUSB الى المعيار RS232 ، أو عن طريق بعض الدارات الخاصة ولعل شهرها على الإطلاق هي FT232BM وهي مستخدمة ضمن الوصلات الجاهزة .

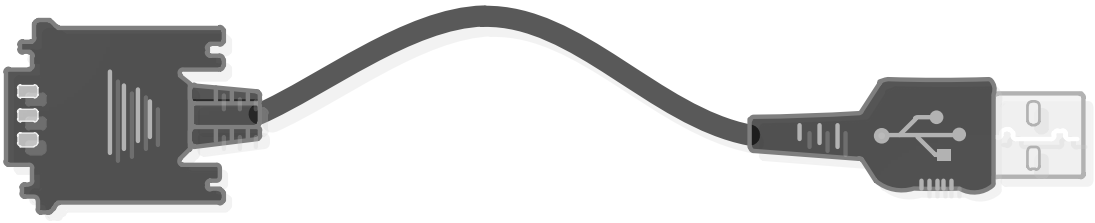
الشكل : شريحة FT232BM من شركة FTDI تقوم بالتحويل من معيار USB الى RS232 وبالعكس .!!



الشكل 76

الشكل 77

الشكل : وصلة جاهزة تقوم بالتحويل من معيار USB الى RS232 وبالعكس .!!



الشكل 78

1.2. نافذة (USART) :

وهو اختصار The universal synchronous/asynchronous receive transmit

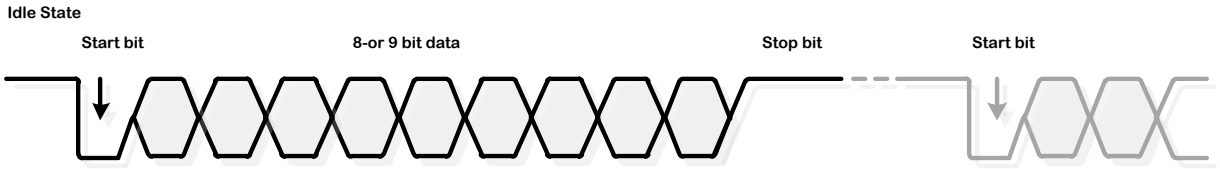
الاتصالات التسلسلية تعتبر طريقة بسيطة وموثوقة لإرسال المعطيات عبر مسافات طويلة ، وبالإضافة الى أنه يتم توفير عدد كبير من أقطاب المتحكم وكذلك الكلفة المنخفضة للاتصالات التسلسلية. الطريقة الأكثر انتشارا تعتمد على البروتوكول RS232 المعياري .

تتم تحقيق هذه النافذة إما عن طريق :

Hardware : أغلب المتحكمات تأتي متضمنة على نافذة USART ضمن بنيتها الداخلية

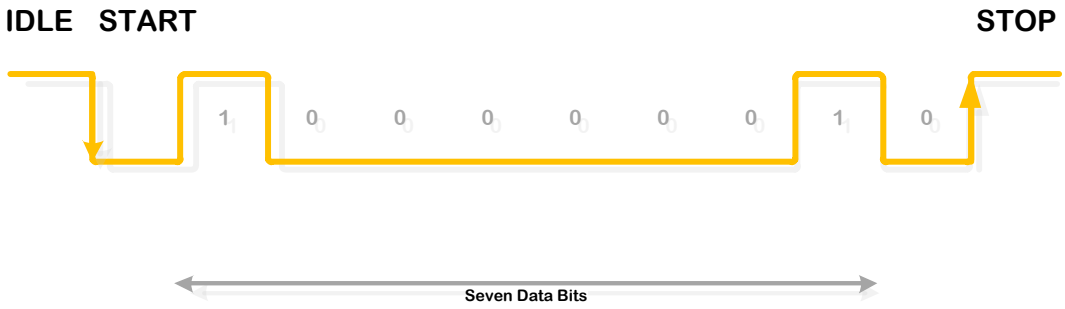
Software : بالنسبة للمتحكمات التي لا تحوي على نافذة USART (مثل PIC18F84) ضمن بنيتها الصلبة فإنه لا زال بالإمكان تحقيق هذه النافذة عن طريق الـ Software .

هذه النافذة تتيح تحقيق اتصال تسلسلي مع طيف واسع جدا من التجهيزات مثل شاشات LCD ، الذواكر ، وكذلك الحاسب الشخصي PC .



الشكل 79

مثال ارسال المحرف " A " :الشكل التالي يظهر كيف يتم ارسال المحرف "A" باستخدام البروتوكول ، شيفرة الآسكي لهذا المحرف هي "0100000010 وكما هو واضح من الشكل : 0100 0001



الشكل 80

أولا يتم ارسال بت البداية (Start bit) ، بعده يتم ارسال 8 أو (7) بتات للمعطيات ، وأخيرا يتم ارسال بت التوقف (Stop bit) .في الاتصال غير المتزامن يتم استخدام بت البداية والتوقف من أجل تحقيق المزامنة بين المرسل والمستقبل .عندما يكون الخط في حالة IDLE فسيكون المرسل على الجهد العالي و عندما يصبح المرسل جاهزا للإرسال فإنه يسحب الخط الى الجهد المنخفض معلما المستقبل بأن عليه الاستقبال بعد ذلك يتم ارسال بتات المعطيات يعقبها بت التوقف الذي يعلن نهاية الإطار، هذه العملية تتكرر مع كل بايت مرسل هذه الصيغة من الإرسال تدعى ASY وذلك لأن المستقل يزامن بنفسه المعطيات عن طريق بت البداية .

بعبارة أخرى فإنه يتم كبسلة بايت المعطيات بـ بت البداية و بـ بت التوقف في حين أن تحقيق الاتصال المتزامن sync لا يتطلب هذه الكبسلة وإنما يتطلب نبضات ساعة مستقلة عن المعطيات .يجب أن يتم تهيئة المرسل والمستقبل بنفس الإعدادات والجدول التالي يبين هذه الإعدادات

Setting			
9600 bps (default)	110 bps 300 bps 600 bps 1200 bps 2400 bps 4800 bps 9600 bps 14400 bps 19200 bps 38400 bps 56000 bps 57600 bps 115200 bps 128000 bps 256000 bps	معدل البود	**Baud Rate**
Eight (default)	Five Six Seven Eight	عدد بتات المعطيات	**Data Bits**
One Stop bit (default)	One Stop bit One and Half Stop bit Tow stop bit	عدد بتات التوقف	**Stop Bits**
None (default)	None Even Odd	خانة الإيجابية	**Parity**

الجدول 20

وهو نفس بروتوكول RS232 باستثناء أن مستوى الجهود مختلف ومعكوس والجدول التالي يوضح مستويات الجهود

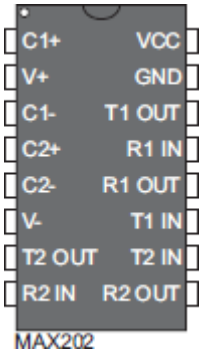
Volt Levels			
المنطق	**مستوى الجهد في RS232**	**مستوى الجهد لنافذة USART في المتحكم**	**ملاحظات**
1	ضمن المجال من 3- الى 25- والقيمة القياسية هي 12-	+5V (منطق الـ TTL)	
0	ضمن المجال من 3+ الى 25+ والقيمة القياسية هي 12+	0V (منطق الـ TTL)	

الجدول 21

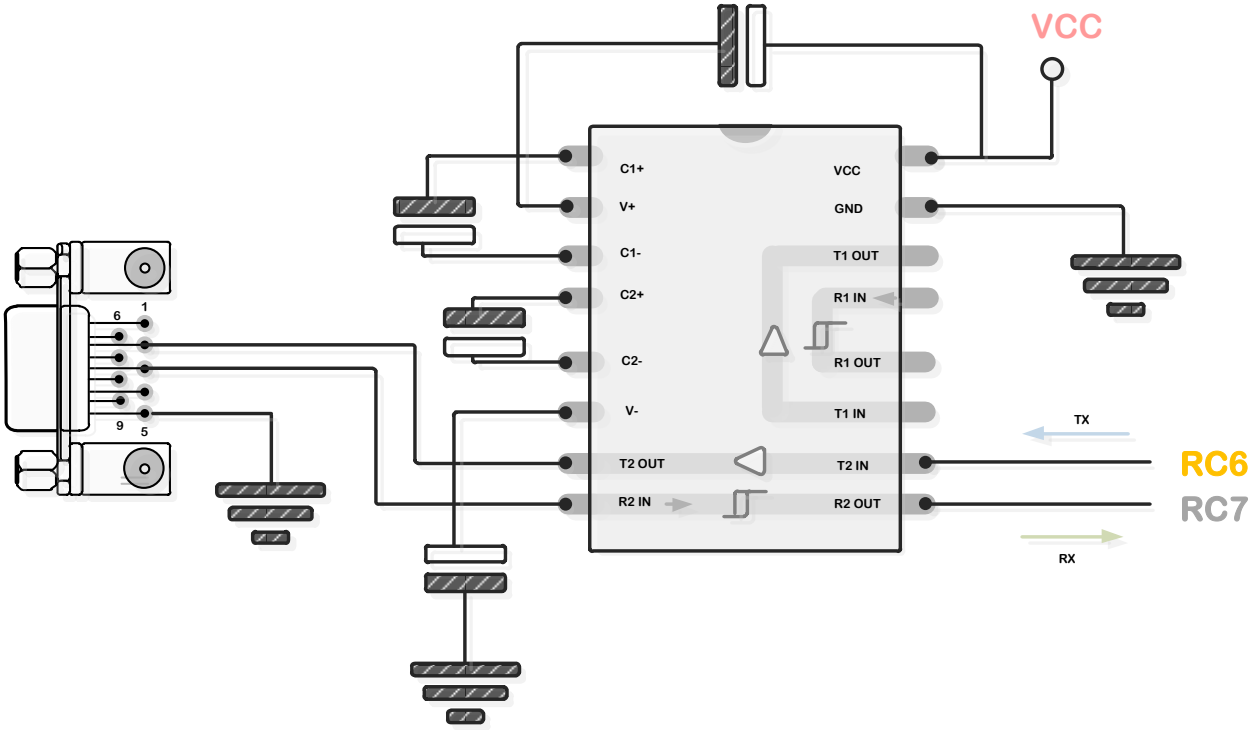
من الجدول السابق كل ما نحتاجه لتحقيق اتصال بين نافذة USART في المتحكم و منفذ الCOM الذي يعتمد على البروتوكول RS232 المعياري هو دارة خارجية يقوم بإزاحة اشارات الـ TTL الخاصة بالمتحكم الى مستوى الـ RS232 والعكس بالعكس .

هذا يمكن تحقيقه ببساطة باستخدام شريحة تدعى MAX232 المصنعة من قبل شركة MAXIM ، هذه الشريحة تحتاج عددا قليلا من المكثفات الخارجية ، تستخدم الشريحة المكثفات من أجل الشحن كي تستطيع توليد 12V− و 12V+ الذي يتطلبه المعيار RS232 .

وأسهل طريقة لأرسال المعطيات من خلال المنفذ التسلسلي للـ PC هو استخدام تطبيق يدعى Hyper Terminal الذي يأتي مع نظام التشغيل Windows ، كذلك سنستخدم برنامج تم كتابته بلغة الـ C# ، وبرنامج آخر عن طريق الماتلاب .



الشكل 81



الشكل 82

Advanced PIC Microcontroller Projects in C : Dogan Ibrahim

Programming 8-bit Microcontroller in C : Martin P. Bates

http://embedded-lab.com/

http://www.ftdichip.com/

http://www.wikipedia.org/

http://www.mikroe.com/

الفصل العاشر

USB

يعد منفذ USB (الناقل التسلسلي العام) واحدا من أكثر المنافذ استخداما وشيوعا في أيامنا هذه ، فالحاسب ، والكاميرا ، و GPS ، ومشغلات mp3 ، الطابعات والماسحات والمودمات كلها مزودة بمنفذ يؤمن تحقيق الاتصال من خلال المنفذ USB ، سأشرح في هذا القسم المبادئ الأساسية لهذا البروتوكول لنتمكن فهم التطبيقات المبنية على هكذا نوع من البروتوكولات ، يحسب هذا البروتوكول على البروتوكولات المعقدة ، الغوص في تفاصيله يخرجنا عن الهدف من هذا المشروع، بالمقابل فإن توفر مكتبات جاهزة قامت بها شركات وأفراد تعنى بتسهيل التعامل مع هذا المنفذ سيكون كفيلا بتغطية تفاصيل التعامل مع هذا المنفذ . هذا البروتوكول يعتبر واجهة تسلسلية عالية السرعة هذه الواجهة بإمكانها تزويد الطرفية بالتغذية . التيار الأعظمي الذي يمكن استجراره من هذا المنفذ هو 100mA عند 5V .

تعريف المصطلحات :

المضيف (Host) : هو جهاز مسؤول عن المسرى وفي حالتنا هذه هو عبارة عن جهاز الحاسب .

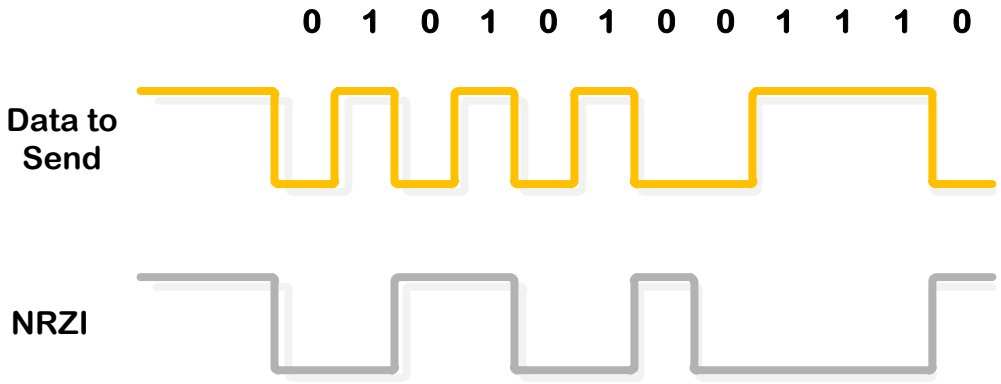
التجهيزة (device) : وهي في حالتنا عبارة عن المتحكم PIC18F452 .

المجمع (hub) : وهو جهاز يتيح وصل أكثر من تجهيزة وكذلك يقوم بإدارة الطاقة ، أيضا يحقق التحويل في السرعة بين المضيف و التجهيزة اذا اضطر الأمر .

البوابة (port):المعنى الفيزيائي لها هو عبارة عن الموصل (connector) ، أما المعنى البرمجي فهي موقع قابل للعنونة ، أي أن البرمجيات تحمل على عاتقها مراقبة البوابة و التحكم بها من خلال عنوان البوابة ، الميزة الأساسية لبوابات المسرى usb و التي تميزها عن البوابات الأخرى مثل rs-232 أن بوابات المسرى usb جميعها موصول بممر واحد الى المضيف ، أما rs-232 فكل بوابة مستقلة عن الأخرى .

1.1. تقنية ارسال المعطيات :

الإشارات المرسلة من المضيف تستخدم تقنية تشفير تدعى NRZI (non-return to zero inverted) ، عند كل تغيير الى المنطق المنخفض فإن مستوى الإشارة يعكس ، بينما عند التغيير الى المنطق العالي فإن الإشارة تبقى على حالها . يتم حشر صفر منطقي (0-bit) بعد كل ست واحدات متتالية يؤدي الى زيادة طول قطار المعطيات والشكل التالي يبين تقنية NRZI .



الشكل 83

تشفير NRZ الواحد يمثل بالمستوى العالي أما الصفر فيمثل بالمستوى المنخفض (حقيقة لا يوجد تشفير) ، احدى مساوئ هذا التمثل أن المستقبل قد يفقد التزامن في حال تعاقب عدد كبير من البتات عند نفس المستوى المنطقي . لذلك في بروتوكول rs232 سيتم اضافة بت بداية وبت أو أكثر للتوقف يتم استخدامها من أجل ضبط التزامن بين المرسل والمستقبل

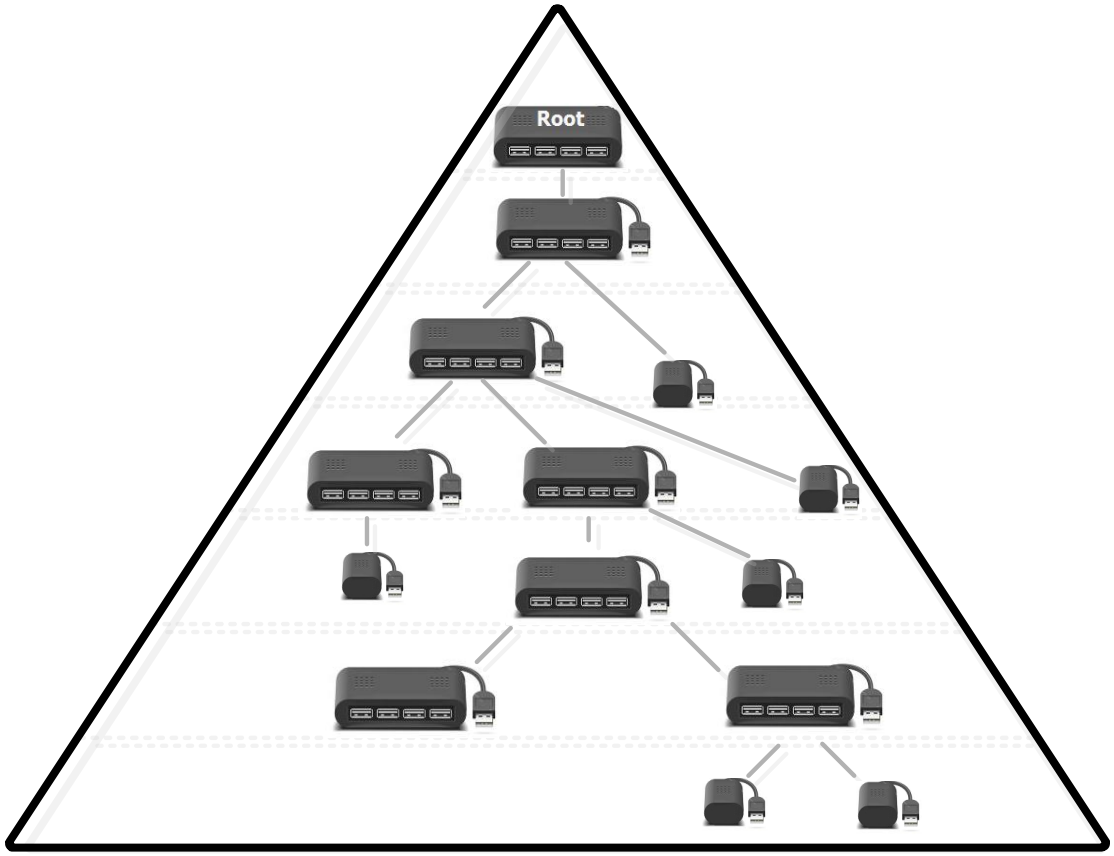
حزمة المعطيات المرسلة من قبل المضيف ترسل الى كل تجهيزة موصولة مع المسرى حيث تتدفق من خلال الممر المشترك ، كل التجهيزات تستقبل المعطيات ، لكن واحدة فقط ذات العنوان المقصود تقبل استلام المعطيات . بالمقابل فإن تجهيزة واحدة فقط يمكنها ارسال المعطيات الى المضيف بنفس الزمن .

1.2. شعارات المسرى (Logos) :

السرعة المنخفضة أو الكاملة	السرعة العالية	السرعة المنخفضة أو الكاملة	السرعة العالية	السرعة الفائقة
Basic-Speed Version	**Hi**-Speed **Version**	Basic-Speed Version On-The-Go non-PC hosts (Embedded Hosts)	Hi-Speed Version On-The-Go non-PC hosts (Embedded Hosts)	Hi-Speed Version

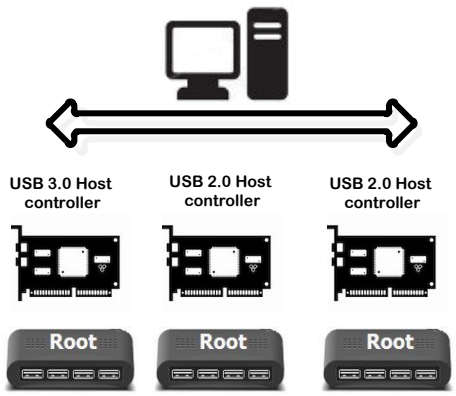
1.3. التركيب البنيوي للمسرى (Bus Topology):

الناقل USB يدعم حتى 127 طرفية (7–bit) للعنونة ، أما العنوان 0 فيستخدم لأغراض خاصة ويتكون الناقل من أربع أسلاك بطول 3 أو حتى 5m) ، يمكن وصلها لكن بطريقة خاصة عن طريق المجمعات Hubs حيث يمكن أن يحوي المجمع الواحد على 4 أو 8 أو حتى 16 بوابة port حيث يمكن وصل الطرفية الى المجمع hub ثم وصل المجمع HUB الى مجمع HUB آخر ببنية هرمية وهكذا ..حتى 6 طبقات ، وبالتالي فإن أعظم مسافة عند استخدام هذه الطريقة هي 30m تتحقق باستخدام خمسة مجمعات Hubs .



الشكل 84

وهو مبني على الطريقة النجمية متعددة الطبقات ، وتعتبر هذه البنية ممتازة حيث أن حدوث خلل في مجمع ما لا يعني بالضرورة الخلل في عمل المسرى ككل . طريقة الوصل السابقة تعبر عن الحالة الفيزيائية أما منطقيا فكافة التجهيزات موصولة مع المسرى العام عن طريق نفس خط المعطيات ، ومن أجل زيادة عرض الحزمة فيمكن استخدام أكثر من ممر معطيات وبالتالي أكثر من مجمع جذري وأكثر من host controller .



الشكل 85

حيث أن الـ host controller والـ root hub هما المسؤولان عن تمكين نظام التشغيل من التواصل مع الطرفيات يقومان معا بعدة وظائف مثل اكتشاف وصل وفصل الأجهزة .

1.4. اصدارات البروتوكول :

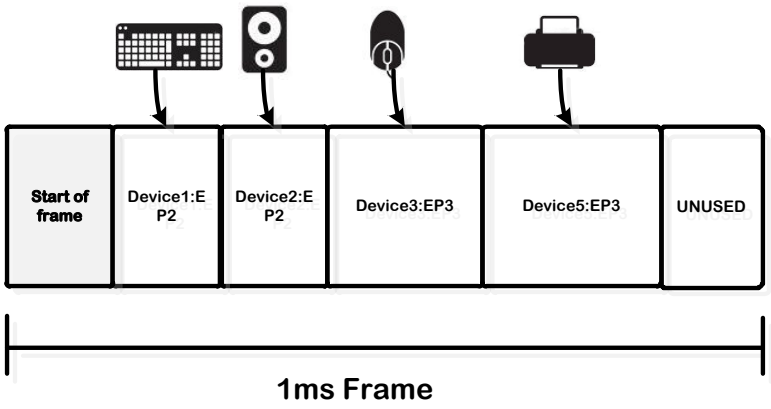
هناك اصدارين من هذا البروتوكول وحديثا صدرت إصدار ثالث :

- USB1.1 وهو يدعم سرعتين (السرعة المنخفضة 1.5M ، والسرعة الكاملة 12Mb/s)
- بينما الإصدار USB2.0 يدعم (السرعة المنخفضة 1.5M ، والسرعة الكاملة 12Mb/s والسرعة العالية,480Mbps)
- أما الإصدار الثالث USB3.0 (بسرعة :4.8 Gbits/s أي أسرع بعشر مرات من اسلافه عالية السرعة SuperSpeed)، هذا الإصدار يتطلب وجود تجهيزات Hardware على اللوحة الأم للحاسب فهو يحتوي على 9 أسلاك وهو غير مدعوم حاليا من قبل المتحكمات ! والمستفيد الأكبر من هذه السرعة الفائقة هي تطبيقات الفيديو حيث يستخدم قطبين للإرسال اضافيين وقطبين استقبال اضافيين و أرضي اضافي عن السرعة العالية حيث أن الأقطاب الإضافية تنشئ ناقل اضافي يعمل على التفرع مع السرعة العالية USB 2.0 port . لكن ماذا عن التوافقية ؟ طرفيات USB 1.1 and USB 2.0 متوافقة تماما مع USB 3.0 ports لكن لن تستفيد من السرعة الفائقة . كم من الوقت حتى يختفي USB 2.0 من تطبيقاتنا ؟ هناك أكثر من عشرة بليون طرفية تستخدم USB 2.0 لذلك فلن يختفي بسرعة بالإضافة الى أن كلفة USB 3.0 تفوق كلفة اسلافه ، حاليا ستدعم تطبيقات الفيديو والأقراص التخزينية USB 3.0 بينما التطبيقات مثل لوحات المفاتيح والماوس ستبقى مستخدمة USB 2.0 .

1.5. عرض اطار المعطيات

بالنسبة للسرعة المنخفضة والسرعة الكاملة يكون عرض الإطار 1ms

أما بالنسبة للسرعة العالية فيكون الإطار مقسما الى ثمانية اطارات صغرية (microframes) عرض كل إطار صغري 125µs .

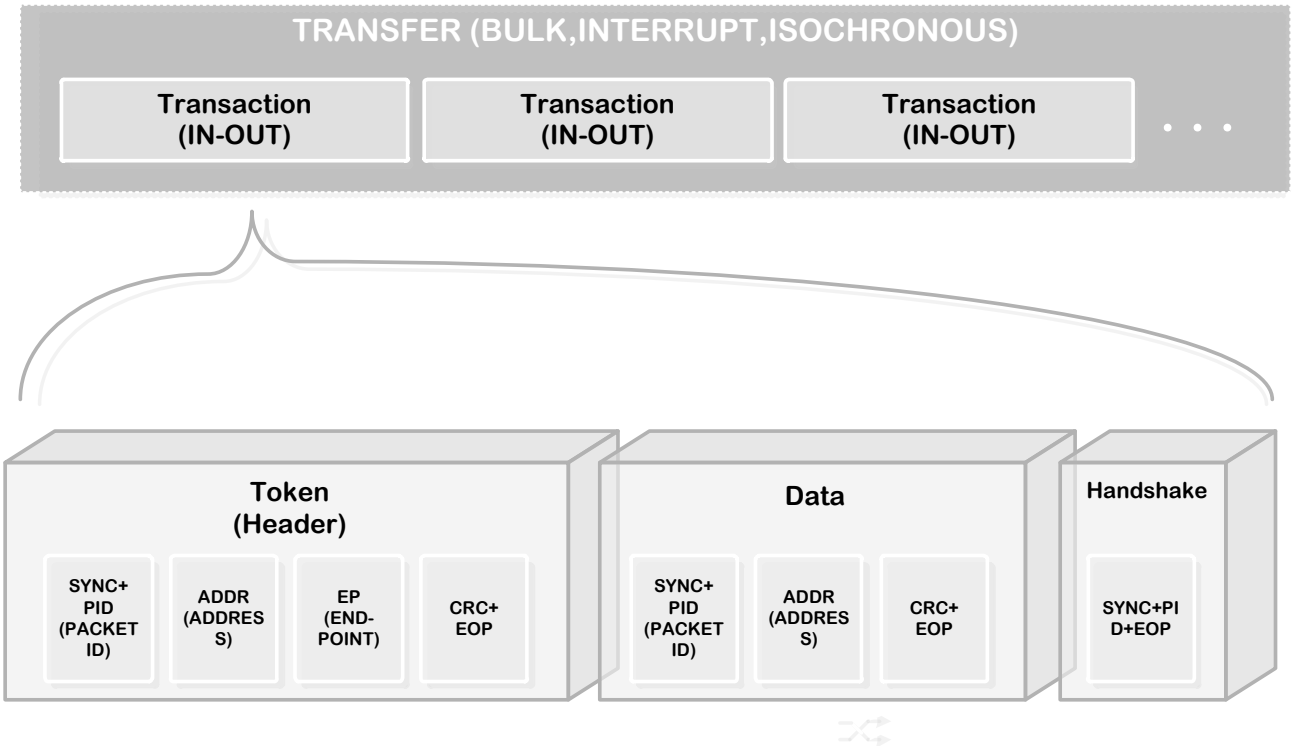


الشكل 86

كل اطار يتألف من عدد من المناقلات transaction .

الاتصال :

عملية النقل transfer تتكون من واحد أو أكثر من المناقلات transaction والتي بدورها تتألف من ثلاث رزم (token ، data ، handshake)

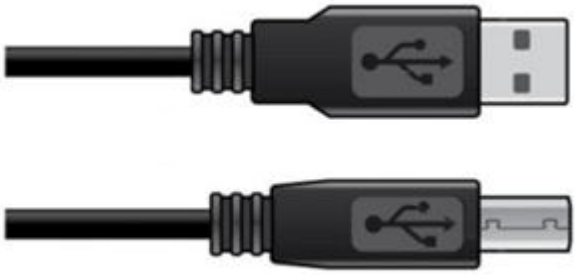


الشكل 87

1.6. أشكال الموصلات (Connectors) :

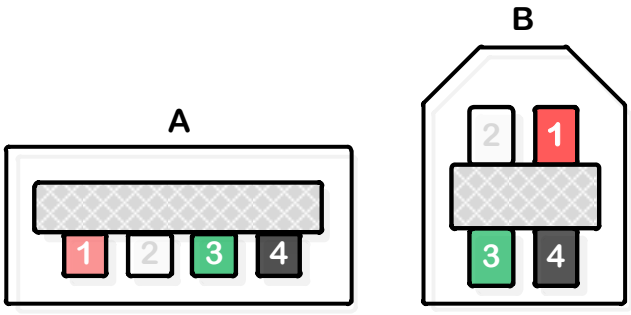
هناك نوعين قياسيين من موصلات USB ، النوع A والنوع B كما يظهر الشكل التالي :

هناك أيضا نوع يدعى min–B يستخدم مع الأجهزة التي تراعي الحجم كثيرا كالكاميرات ، وما شابه وهو يمتلك خمسة أرجل حيث الرجل الخامسة تدعى ID وهي غير مستخدمة في حالتنا (وانما تستخدمها بعض الأجهزة لتعرف النمط الافتراضي للجهاز)



الشكل 88

أما ترتيب الأرجل لهذين الموصلين فيبينها الشكل التالي



Pin-Out of USB connectors

الشكل 89

أما ألوان الأسلاك الاربعة للنوعين A و B فتعطى بالجدول التالي :

Pin no.	Name	Color
1	+5.0V	Read
2	-Data	White
3	+Data	Green
4	Not used	-
5	Ground	Black

الجدول 22

Mini USB pin assignments

من الجدول السابق نرى أن هناك رجلين من أجل نقل المعطيات هما +Data، Data– أما السلكين فيكونان مجدولين (لتقليل الضجيج عند السرعات الكبيرة) يؤمنان نقل الإشارة التفاضلية ، الكلام السابق لا يشمل USB 3.0 فإن له بنية مختلفة .

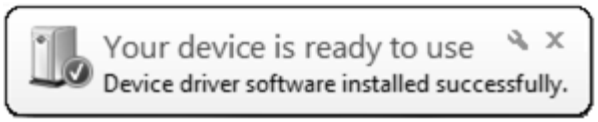
1.7. اساسيات النقل (Transfer Basics):

- اتصالات الإعداد Configuration Communications

في هذا النوع من الاتصالات يقوم المضيف بجمع المعلومات عن الطرفية وذلك عن طريق ارسال سلسلة من الطلبات القياسية (يمكن اعتبارها أسئلة من أجل التعارف) فيقوم البرنامج المخزن بالطرفية device's firmware بالإجابة عن هذه الأسئلة ويقوم بإرسال المعلومات المطلوبة.

هناك 11 طلب قياسي ، بعض الطرفيات لا تدعم جميع هذه الطلبات فتقوم بإرسال اشارة تدل على أنها لا تدعم هذا الطلب .

عملية التعداد بالنسبة لنظام Windows تتم بالخفاء وبشكل آلي باستثناء اظهار رسالة تدل على اكتشاف جهاز جديد ونجاح العملية أم لا كما في الشكل التالي .



خطوات التعداد :

1	**The user plugs a device into a USB port**	يقوم المستخدم بوصل الطرفية الى بوابة المسرى
2	**The hub detects the device**	يكتشف المجمع وجود الطرفية عن طريق مراقبة الجهود على كل بوابة من بواباته حيث أن للمجمع مقاومة قيمتها 15k موصولة للأرضي من أجل كلا الخطين D+ ، D- أما بالنسبة للطرفية فيتم وصل مقاومة 1.5k بين خط D+ و الجهد 3.3V من أجل السرعة الكاملة أما بالنسبة للطرفية فيتم وصل مقاومة 1.5k بين خط D- و الجهد 3.3V من أجل السرعة المنخفضة
3	**The host learns of the new device**	يقوم المجمع بإخبار المضيف بوجود طرفية جديدة
4	**The hub detects whether a device is low or full speed**	يقوم المجمع باكتشاف سرعة الطرفية هل هي ذات سرعة منخفضة أم كاملة ، من خلال الخط ذي الجهد المرتفع .
5	**The hub resets the device**	يقوم host controller بإرسال طلب للـ Hub يسأله بأن يقوم بترسيت البوابة حيث يقوم المجمع بجعل كلا الخطين D+ and D- على الصفر منطقي لمدة 10ms
6	**The host learns if a fall-speed device supports high speed**	يكتشف المضيف هل هذه الطرفية التي تدعم السرعة الكاملة تدعم السرعة العالية أم لا
7	**The hub establishes a signal path between the device and the bus**	يؤسس المجمع مسار بين الطرفية والمسرى من خلال نقطة الاتصال 0 عبر قناة الاتصال الافتراضية Default pipe
8	**The host sends a Get_Descriptor request to learn the maximum packet size**	يرسل المضيف طلب " الحصول على الواصف " للطرفية وذلك لمعرفة الحجم الأعظمي للرزمة

	of the default pipe	
9	**The host assigns an address**	يقوم المضيف بإرسال عنوان للطرفية حيث تقوم الطرفية بالاستجابة وتخزين هذا العنوان وارسال اشارة acknowledge ، ومن الآن وصاعدا سيتم التعامل مع الطرفية عن طريق هذا العنوان
10	**The host learns about the device's abilities**	يرسل المضيف طلب " الحصول على الواصف " لكن هذه المرة يقوم بقراءة كل ما ترسله الطرفية وسيتم شرح الواصف لكن حاليا يكفي أن نعلم أنه بنية معطيات توصف الطرفية أي تضم معلومات مفصلة عن الطرفية
11	**The host assigns and loads a device driver**	بعد أن يتلقى المضيف الواصف من الطرفية فسيتمكن من تحديد أفضل برنامج قيادة لهذ الطرفية
12	**The host's device driver selects a configuration**	يقوم برنامج القيادة بإرسال الإعدادات الى الطرفية

- اتصالات التطبيقات Application Communications

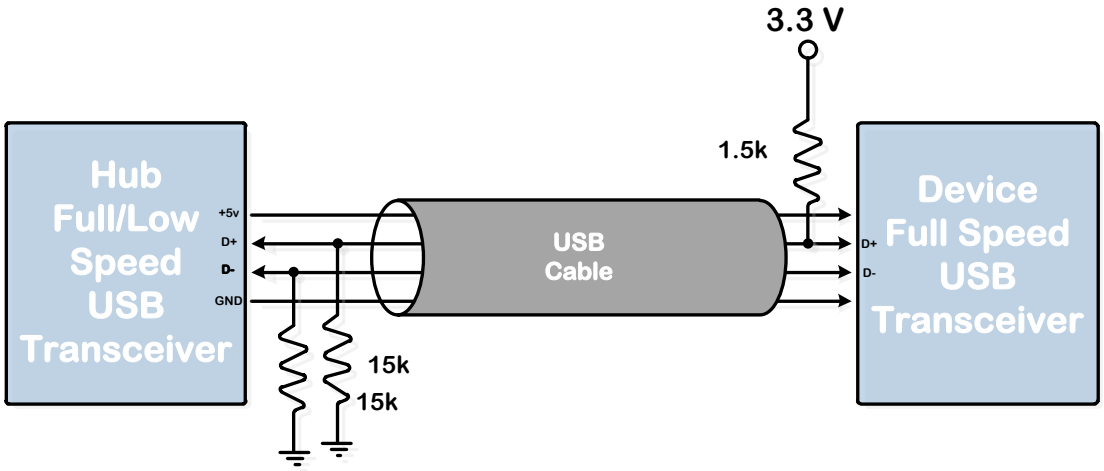
بعد الانتهاء من تعداد الطرفية وشحن برنامج القيادة يمكن للتطبيقات المكتوبة بلغة عالية المستوى من التخاطب مع الطرفية في هذا المشروع سيتم استخدام C# للتعامل مع مسرى الUSB.

1.8. تحديد السرعة :

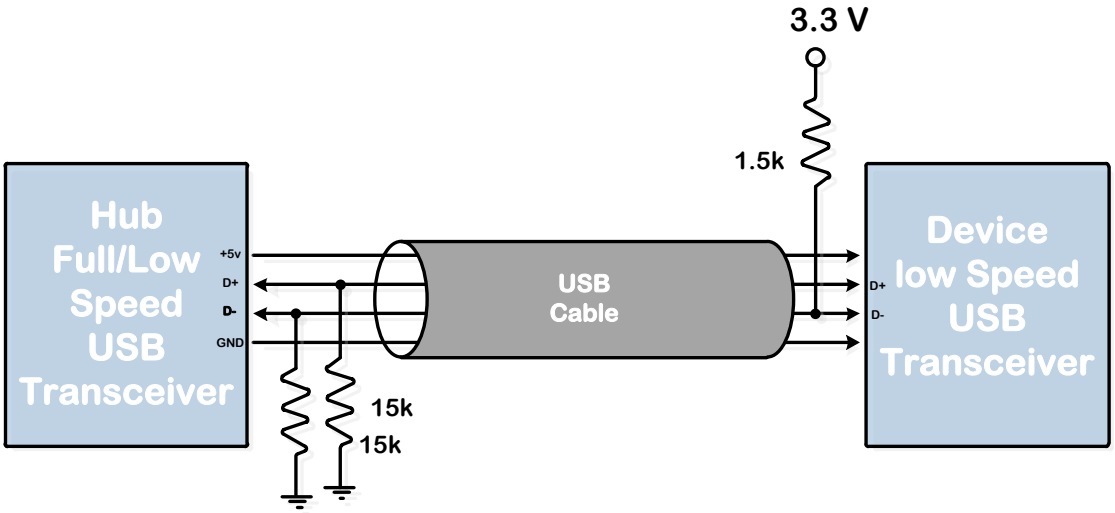
عند وطل الطرفية الى بوابة ما يقوم المجمع بتحديد هل هذه التجهيزة تدعم السرعة المنخفضة أم السرعة الكاملة عن طريق قياس أي من الخطوط D+ أو D− هو الأكثر ايجابية

بعد ذلك وفي حال كانت الطرفية تدعم السرعة الكاملة يقو المجمع باختبار هل هذه الطرفية تدعم السرعة العالية أم لا عن طريق ارسال اشارة مصافحة وفي حال نجاح المصافحة تقوم الطرفية بإزالة الوصل مع التغذية 3.3 وتبدأ بالاتصالات بالسرعة العالية .

- Full-Speed Device Detection



- Low-Speed Device Detection



1.9. صفوف التجهيزات (Device Classes):

كل مجموعة من التجهيزات (الطرفيات) التي تتشابه بالوظائف فإنه من الأنسب تعريف صف خاص بها وبالتالي برنامج قيادة شبه موحد،(حيث أن أنظمة التشغيل مثل windows تتضمن برامج قيادة من أجل الصفوف الشهيرة مثل الأجهزة الصوتية ، الماسحات الضوئية والكمرات ، الطابعات ، HID)

والجدول التالي يوضح هذه الأصناف :

Device class	Description	Example device
0x00	Reserved	-
0x01	USB audio device	Sound card
0x02	USB communications device	Modem, fax
0x03	USB human interface device	Keyboard, mouse
0x07	USB printer device	Printer
0x08	USB mass storage device	Memory card, flash drive
0x09	USB hub device	Hubs
0x0B	USB smart card reader device	Card reader
0x0E	USB video device	Webcam, scanner
0xE0	USB wireless device	Bluetooth

الجدول 23

صف أجهزة التخاطب مع الإنسان HID (Human Interface Device) : تندرج تحت هذا الصف كل الطرفيات التي تنقل كتلا من المعلومات من المضيف أو اليه بسرعات متوسطة مثل لوحات المفاتيح ، الماوس ، والمتحكمات في حالتنا هذه .

في هذا الصف فإن الطرفيات على الأغلب ينبغي أن تتفاعل مع المستخدم بشكل مباشر مثل لوحة المفاتيح أو الماوس ، حيث تقوم الطرفية بإرسال المعلومات الى المضيف بزمن لا يمكن توقعه.

يستخدم الصف HID :

إما وصلة التحكم الافتراضية Default Control Pipe

أو وصلة المقاطعة an interrupt Pipe

الشكل النموذجي لهذا البروتوكول يتكون من مضيف واحد يتصل مع تجهيزة أو أكثر موصولة عن طريق مسرى الـUSB ، المضيف PC يكون لديه مجمع جذري Embedded Root Hub والذي يحتوي عادة على بوابتي USB .

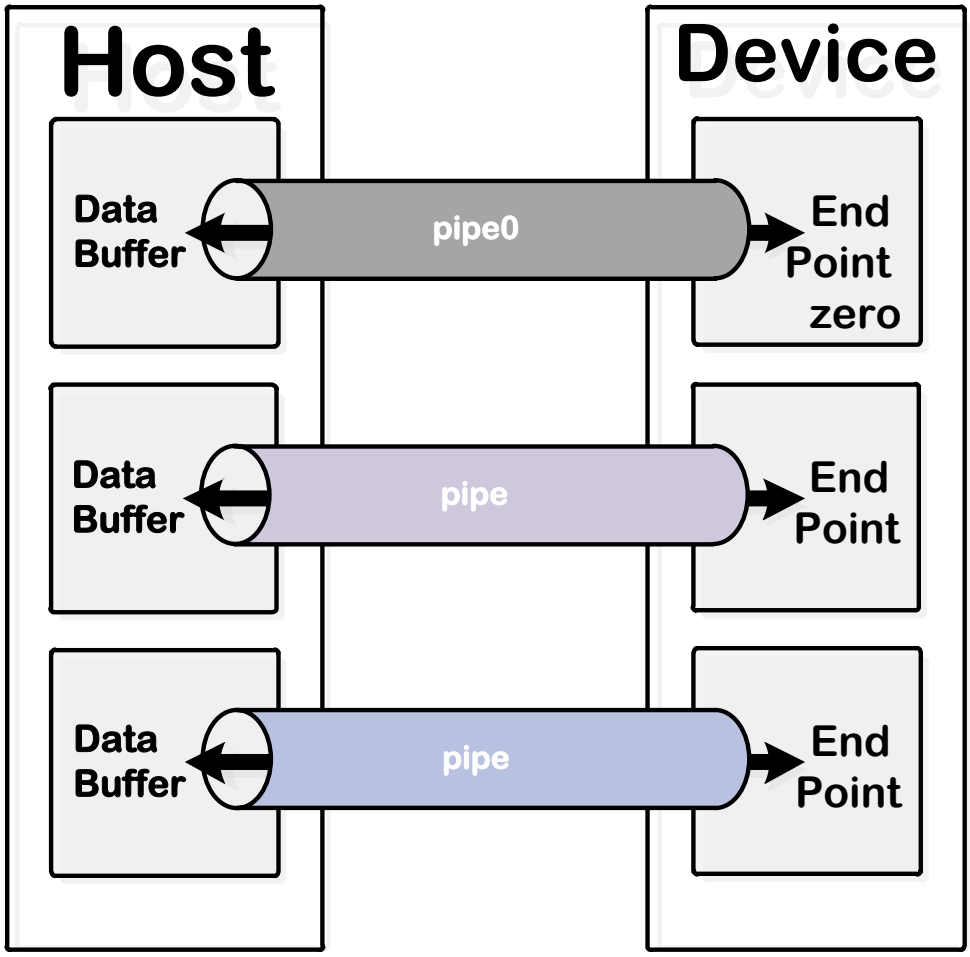
1.10. مصطلحات عامة :

نقطة الاتصال Endpoint : وهي عبارة عن ذاكرة مؤقتة (عدة مسجلات) يتم عنونتها بشكل فريد وهي موجودة ضمن التجهيزة (المتحكم) .

المعطيات المخزنة فيها هي عبارة عن معطيات مستقبلة أو معطيات تنتظر الإرسال ، وتشكل منبع للمعطيات أو مستقر لها .

نقطة الاتصال تحدد بـ:

- رقم النقطة : وهو ضمن المجال من 0 الى 15 حيث تخصص نقطة الإتصال EP0 كنقطة تحكم
- اتجاهها : يعرف الاتجاه من وجهة نظر المضيف أي أن EP IN هي منبع معطيات بالنسبة لل HOST أي أن المعطيات تنتقل من الطرفية الى المضيف .
- لكل نقطة اتصال واصف خاص بها (descriptor) : يتضمن عنوانها ونمط الاتصال و حجم المعطيات الأعظمي .....



الشكل 90

قناة الاتصال Pipe : وهي عبارة عن وصلة منطقية تربط المضيف مع نقطة اتصال في التجهيزة أو الطرفية .

يقوم المضيف بإنشاء قنوات الاتصال عند وصل الطرفية مع المسرى العام ويقوم بازالتها عند فصلها ، و ربما ينشئ أو يحذف وصلات أثناء العمل حسب اللزوم.

لكن دائما يوجد قناة اتصال من اجل التحكم وهي موصولة مع EP0 .

وأهم ميزة لقناة الاتصال هو عرض الحزمة حيث يقوم المضيف بفحص عرض الحزمة المتاح قبل تأسيس الوصلة فإن كانت قناة الإتصال تتطلب عرض حزمة أكبر من الحزمة المتاحة فلن يقوم بإنشاء الوصلة ويجبر التجهيزة على الإنتظار أو ربما تقوم التجهيزة بطلب عرض حزمة أقل

المراجع :

http://www.usb.org/

USB Complete Jan Axelson , June 2001

USB Design by Example - A Practical Guide to Building IO Devices : John Hyde

: Don Anderson) USB System Architecture (USB 2.0

Advanced_PIC_Microcontroller_Projects_in_C

Programming 16-Bit PIC Microcontrollers in C

PIC Microcontrollers - 50 Projects for Beginners and Experts

Programming 8-bit PIC Microcontrollers in C with Interactive Hardware Simulation

Robotics experiment withPIC microcontroller

لوددت أن أقوم بتوثيق جميع المراجع لكن بسبب تسليم المشروع قبل أوانه اضطرت الى توثيق بعضها ، ايضا الكثير من الأكواد والدارات العملية غير موجودة لنفس السبب ، ان شاء الله في نسخة قادمة سنتلافى ذلك

أخر تعديل في 20/11/2012

وختاما أسأل الله الفرج القريب

https://www.facebook.com/groups/495894171/31691