Esercitazioni PIC Midrange.

Inserto_b : Il clock

Ecco alcune pagine che riguardano i sistemi di clock dei Midrange.

Nei Baseline abbiamo trovato disponibili diverse opzioni per il clock primario, abbastanza uniformi. Queste sono selezionabili nella configurazione iniziale e, per tutti, c'è la disponibilità di un oscillatore interno a 4MHz, in alcuni modelli commutabile tra 4 e 8MHz.

Per i **Midrange** abbiamo una situazione molto meno uniforme: dobbiamo considerare che essi hanno una ampia varietà di sistemi di clock, che possiamo dividere grosso modo in alcuni gruppi:

- Chip da considerare obsoleti (16F84, 16F876/77, 16F7x, ecc.) che **funzionano solo con clock esterni e non dispongono di oscillatore interno**.
- Chip che **non dispongono di oscillatore interno, ma solo di alcuni modi esterni** (ad es.12F716)
- Chip che **dispongono di oscillatore interno che richiede calibrazione nel registro OSCCAL** (12F629/12F675, 12F630/12F676)
- Chip che **dispongono di oscillatore interno pre tarato e non richiedono calibrazione**, ma hanno la possibilità di aggiustamento con il registro OSCTUNE.

Questi ultimi sono la maggioranza, ma anche tra questi esistono ampie differenze:

- Chip con clock interno programmabile con 8 frequenze (tipicamente 31kHz- 8MHz)
  Ad es. 16F690, 12F683, 12F635, 16F684, 16F88x
- chip con clock interno programmabile con 4 frequenze (tipicamente 31kHz-8MHz), ma con modalità di oscillatore esterno limitate al modo EC. Ad esempio 16F753
- chip con clock interno programmabile con 8 frequenze (tipicamente 62.5kHz- 16MHz) con PLL, ma con modalità di oscillatore esterno limitate al modo EC. Ad esempio 16F/20/721

A queste “variazioni di base” dobbiamo aggiungere la possibile presenza di numerose funzioni ausiliarie per la sicurezza, come il fail safe clock monitor, o per il risparmio energetico, come il clock switching, il dual clock , ecc.

Un altro particolare che pare aggiunto per complicare la vita all'utente consiste nel fatto che non tutti i chip dotati di multi oscillatore hanno per default lo stesso clock.
Ad esempio, 12LF1522 parte con 500kHz, mentre 16F753 con 1MHz e 16F690 con 4MHz.

E' evidente che l'ambiente Midrange è stato teatro di sperimentazioni di varie possibilità che sono state introdotte in vari modelli, oltre al fatto che spesso i chip sono realizzati in base a specifiche richieste di OEM.

Questa situazione assai fluida, ancora una volta, conferma la necessità di consultare il foglio dati del componente prima di deciderne l'uso per una specifica applicazione.

Per i chip che usiamo in queste esercitazioni, possiamo creare una tabella delle caratteristiche principali:

PIC	Modi	Special	Fosc/4	Calibrazione
12F629 **12F675**	• LP Low Power Crystal 32kHz typ. • XT Crystal/Res 0.1- 4MHz • HS High Speed 4-20MHz • RC External R/C 20MHz max • INTOSC Internal Osc 4MHz • EC External Clock In 20MHz max	-	**RC INTOSC**	si **OSCCAL**
16F690	• LP Low Power Crystal 32kHz typ. • XT Crystal/Res 0.1- 4MHz • HS High Speed 4-20MHz • RC External R/C 20MHz max • **INTOSC Internal Osc 31KHz-8MHz** • EC External Clock In 20MHz max	• Doppio oscillatore interno commutabile • 8 frequenze selezionabili Default 4MHz		no **OSCTUNE**
16F684 **16F688**	• LP Low Power Crystal 32kHz typ. • XT Crystal/Res 0.1- 4MHz • HS High Speed 4-20MHz • RC External R/C 20MHz max • **INTOSC Internal Osc 31KHz-8MHz** • EC External Clock In 20MHz max	• 8 frequenze selezionabili default 4MHz		

12F629/675 richiedono la calibrazione del'oscillatore interno.
16F690 e 16F688/684 hanno l'oscillatore pre calibrato.

12F629/675 hanno una sola frequenza per l'oscillatore interno, che è fissa a 4MHz.
16F690 dispone di una selezione di 8 valori di frequenza, commutabili da programma. Quella di default è 4MHz.
16F688/684 dispogono di 4 valori di frequenza commutabili da programma e quella di default è 4MHz.

Queste ultime particolarità rendono evidente che, se pure il pinout la struttura dei chip sia analoga, occorre in ogni caso verificare le reali possibilità del chip che si intende usare.
Così, non possiamo dare per certa la frequenza di default dell' oscillatore interno, anche se questo è il valore più comune: una istruzione con clock a 4MHz impiega 1us, ma con clock di 1MHz il tempo passerà a 4us, il che altererà tutti i risultati delle funzioni in cui il tempo è importante.

Calibrazione dell' oscillatore per 12F629 e 12F675.

Dove utilizziamo l'oscillatore interno è sempre consigliabile procedere alla sua calibrazione: come visto per i Baseline, questa consiste nel trascrivere nel registro di controllo dell'oscillatore interno il valore di calibrazione introdotto in fabbrica.

Con questa azione si porta la precisione dell'oscillatore all'1% o migliore, il che è più che adeguato per gran parte delle applicazioni.

Se per i Baseline si tratta della prima istruzione da eseguire, non è così per i 12F629/675.
Abbiamo tratteggiato nell'introduzione come la struttura dell'oscillatore interno sia diversa tra le due famiglie. In una tabella:

Baseline	12F629/675
• Il vettore reale di reset punta all' ultima locazione di memoria dove si trova scritto il valore di calibrazione di fabbrica dell'oscillatore, sotto forma di una istruzione: `movlw valorecalibrazione`	• Il vettore di reset punta a 00. Il valore di calibrazione è sempre scritto nell'ultima locazione di memoria, ma sotto forma di una istruzione: `retlw valorecalibrazione`

La disposizione è assai più vantaggiosa di quella dei Baseline, perchè:

- nei Baseline è obbligo calibrare l'oscillatore prima che il contenuto di W sia modificato (dato che non è possibile accedere successivamente alla memoria programma come dati).
- **qui è possibile calibrare in qualsiasi punto del programma**, dato che il valore di calibrazione può essere richiamato in qualsiasi momento come subroutine.

Quindi non è obbligo per i Midrange calibrare immediatamente, ma, se non sono richieste altre istruzioni iniziali con maggiore priorità, non c'è nessuna ragione per non effettuare la calibrazione ad inizio programma, giusto per non dimenticarsela.

Abbiamo visto che **retlw** esegue un ritorno da subroutine, ponendo in WREG il valore in oggetto. Quindi la procedura di calibrazione consisterà nel chiamare **come subroutine** l'istruzione posta alla fine della memoria programma (e consistente nel solo **retlw valore calibrazione**).

Questo fa si che il valore di calibrazione stabilito in fabbrica sia posto in WREG e basterà copiarlo nel registro di calibrazione.

Attenzione perchè il registro non è in banco 0!

Quindi è richiesto un cambio di banco, dato che al POR il default azzera gli switch e punta al banco 0.

In istruzioni:

```
RESVEC    ORG 0x00

; calibrazione oscillatore interno
     banksel OSCCAL    ; OSCCAL in banco 1
     call    0x03FF    ; rientro con il valore_calibrazione in W
     movwf   OSCCAL    ; calibrazione
```

Questo vale per tutti i chip Midrange dotati di **OSCCAL**.

Ovviamente, se si utilizza un oscillatore con componenti esterni, la calibrazione di quello interno non è necessaria, come non è necessaria nel caso di applicazioni non time-sensitive (come era la prima esercitazione); però, **dato che occupa pochi microsecondi, è opportuno effettuarla ogni volta che venga utilizzato l'oscillatore interno.**

Se non calibriamo immediatamente, ma dopo varie istruzioni, come in questo caso, vuol dire che dal vettore di reset fino a ora abbiamo lavorato con un clock non calibrato. Però, la cosa non influisce minimamente sul funzionamento del programma finora eseguito, che fino a questo punto non ha alcuna criticità: che il ciclo di istruzione sia stato 1us o 1,05us è indifferente.

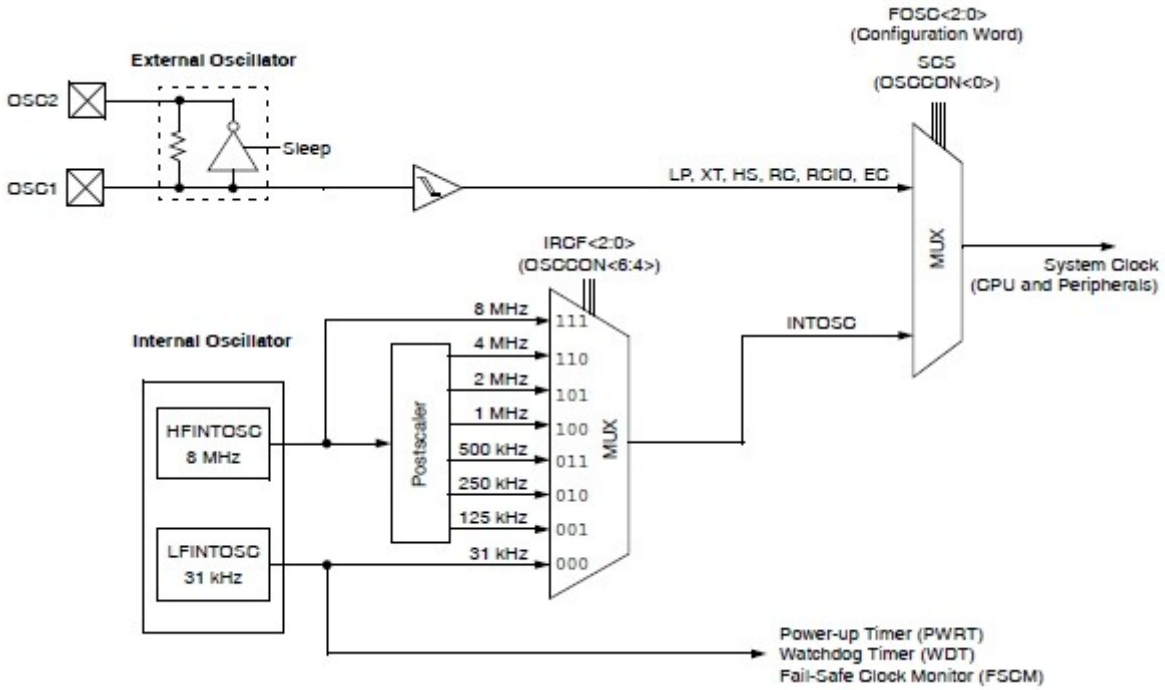
Solamente quando si avvierà la routine di tempo, è opportuno in ogni caso che questo tempo sia quanto possibile esatto. Anche se una applicazione come quella in corso avesse un clock non perfettamente preciso, la cosa non si noterebbe minimamente durante la sequenza di accensione dei LED.
Però non esiste ragione per non cercare comunque la migliore condizione di esecuzione del programma.
Se, poi, i tempi da generare sono critici, come in una trasmissione seriale, avere un clock quanto più preciso possibile diventa un obbligo.

Oscillatori interni multi frequenza.

Piccoli Midrange come 12F629/675 si posso considerare dei Baseline “evoluti”: hanno le opzioni base dei modi di oscillatore, compreso quello interno ai classici 4MHz, e richiedono una calibrazione.

La maggior parte dei Midrange, invece, dispongono di un sistema di clock primario abbastanza complesso, oltre al fatto di non richiedere calibrazione.

Questo è il diagramma funzionale del sistema di clock per 16F690 e per buona parte dei Midrange:



Abbiamo sostanzialmente un doppio oscillatore RC interno di precisione, con un errore nominale minore dell'1%.

 Per questi chip NON è prevista alcuna manovra di calibrazione, in quanto l'oscillatore è aggiustato al meglio in fase di produzione. Il processo produttivo si è evoluto ed ha permesso di evitare una calibrazione da programma. Questo risparmi dimenticanze e istruzioni.

Peraltro, come vediamo più avanti, è disponibile un registro particolare per operare, dove richiesto, uno spostamento di qualche punto percentuale della frequenza nominale.

Nel diagramma notiamo due sezioni, una denominata **`LFINTOSC`** (Low Frequency INTernal OSCillator - oscillatore interno a bassa frequenza) con una frequenza tipica di 31kHz e una chiamata **`HFINTOSC`** (High Frequency) che genera una frequenza di 8MHz.

L'energia richiesta dall'oscillatore **`LFINTOSC`** è molto bassa ed è ideale per realizzazioni a basso consumo.

Le uscite di entrambi gli oscillatori sono convogliate ad un multiplex dotato di prescaler, che consente di utilizzare come clock principale 8 diversi valori di frequenza.

La selezione del multiplex è effettuata con il registro **OSCCON**.

OSCCON – OSCILLATOR CONTROL REGISTER (ADDRESS: 8Fh)

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS(1)	HTS	LTS	SCS
bit 7							bit 0

Osserviamo che è situato in banco 1.

Possiamo dire, semplificando, che i chip dotati di **OSCCAL** richiedono calibrazione. Quelli dove è presente **OSCCON** non richiedo calibrazione.

I bit **IRCF2:0** di **OSCCON** sono i selettori del multiplex, secondo questa tabella:

IRFC2:0	Clock
000	31kHz
001	125kHz
010	250kHz
011	500kHz
100	1MHz
101	2MHz
110	4MHz default
111	8MHz

Questi bit possono essere variati da programma. **Il default al POR è 4MHz.**

Gli altri bit del registro sono utilizzati per la verifica della stabilizzazione dell'oscillatore (HTS, LTS), per la selezione della sorgente interna (SCS) e per lo startup (OSTS).
Queste funzioni particolari, assieme al two-speed start up, clock internal-external switching e al fail safe clock non saranno considerate in queste esercitazioni.

Da osservare che, nonostante l'oscillatore interno sia pre tarato, c'è a disposizione un ulteriore registro di controllo che ha lo scopo di aggiustare la frequenza generata. Si tratta dell'**OSCTUNE**.

OSCTUNE – OSCILLATOR TUNING RESISTOR (ADDRESS: 90h)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7							bit 0

Anche questo si trova in banco 1.

Lo scopo dei bit **TUN4:0** è quello di variare la frequenza nominale attorno al valore centrale di circa un +/-12%, secondo questa regola:

TUN4:0	Frequenza
0111	massima
0110	
.	
0001	
0000	nominale
1111	
.	
1001	
1000	minima

Questa opzione permette di distanziarsi dalla frequenza nominale per applicazioni particolari. Da osservare che ha effetto solo sulle frequenze generate da **HFINTOSC.**

Il multiplex può selezionare sia gli oscillatori interni che i modi esterni (EC, RC, LP, XT, HS). Da osservare che questa scelta è possibile sia dalla ***Configuration Word***, sia con il bit **SCS** di **OSCCON**.

D'altra parte, i 31kHz generati dall'**LFINTOSC** sono utilizzati anche dai moduli WDT, dal PWRT (Power on Timer) e dal sistema di sicurezza del fail safe clock. Questo oscillatore è una sorgente di clock sempre attiva perchè a basso consumo energetico.

Con qualche variazione o funzione addizionale, questo è il sistema di clock di gran parte dei Midrange, ma occorre considerare che alcuni modelli presentano delle differenze abbastanza sensibili.

Quanto detto rende sempre più valida la necessità, in caso di dubbio, di consultare il foglio dati del componente che si desidera impiegare.

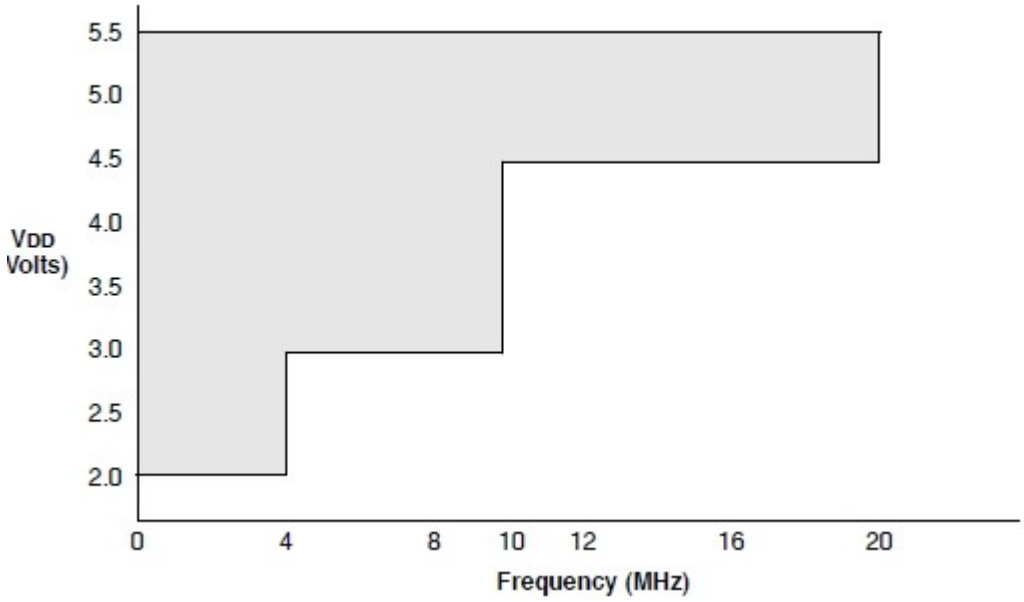
Frequenza e tensione Vdd.

E' evidente che la possibilità di avere un chip ad una frequenza fissa è limitante rispetto alla disponibilità di più valori, dato che una maggiore frequenza consente di avere tempi di esecuzione più veloci per potersi adeguare alle necessità del processo controllato (comunicazioni seriali, misure, conversioni AD, ecc.).

Peraltro, va ben compreso che non è un obbligo mandare il chip alla massima velocità possibile, in quanto, per l'esecuzione del programma, può essere sufficiente un clock più basso. La scelta è legata al fatto che il consumo di energia aumenta in modo quadratico con l' aumentare della frequenza; ne deriva che una applicazione a basso consumo dovrà avere un clock più basso possibile e, in ogni caso, abbassando la frequenza dell'oscillatore avremo un risparmio di energia.

Inoltre, alte frequenze di lavoro sono possibili solo in proporzione diretta alla Vdd: i PIC, in generale, possono lavorare con tensione tra 2 e 5.5V, ma la tensione è legata la massima frequenza possibile.

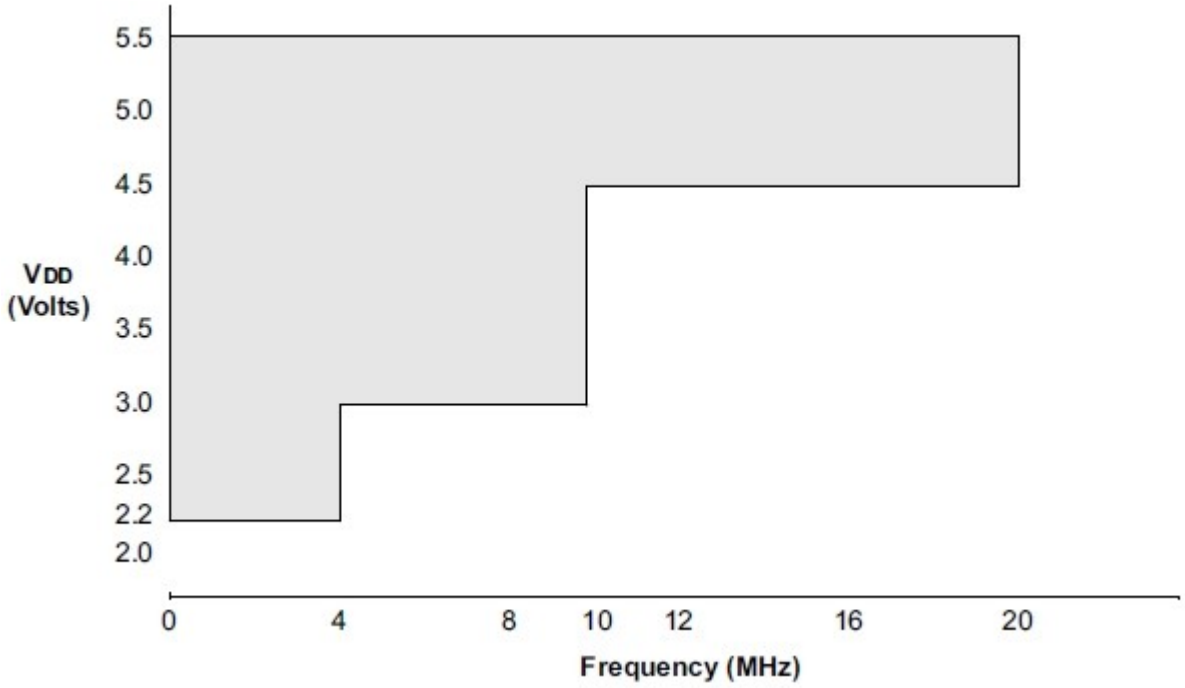
Un esempio è il diagramma seguente, relativo al 16F690, orientativamente, valido per tutti i chip:



La tabella indica che fino a 4MHz di clock non creano problemi a qualsiasi tensione di alimentazione. Frequenze fino a 10MHz richiedono almeno una Vdd=3V e sarà possibile superare con sicurezza i 10MHz solo con Vdd di almeno 4.5V.

Le frequenze superiori a 8MHz dovranno essere generate con componenti esterni (modi HS, RC e EC).

Possiamo confrontarlo con quello del 12F629/675



Notiamo che per questi ultimi, la minima tensione di alimentazione è 2.2V.
Le frequenze superiori a 4MHz dovranno essere generate con componenti esterni (modi HS, RC e EC).

Quindi, occorre prestare attenzione al rapporto tra clock e Vdd, onde evitare aree di funzionamento che non garantiscono sufficiente sicurezza.

FOSC/4

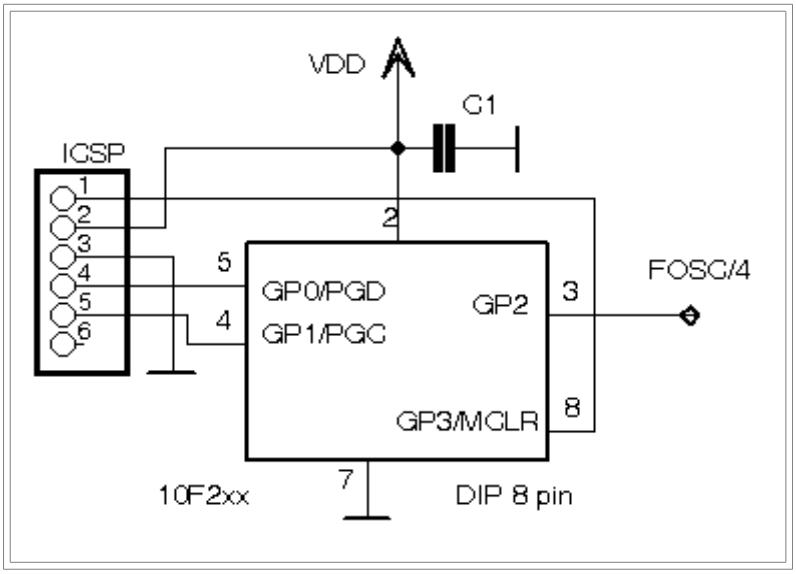
Sostanzialmente, tutti i chip dispongono della possibilità di avere su un pin la frequenza del clok nella forma **Fosc/4** , ovvero la frequenza del ciclo di istruzione.

Ha lo scopo di fornire il clock ad altri dispositivi, così come CLKIN permette di usare per il chip un clock proveniente dall'esterno.

L'analisi di questo segnale è riservata a chi dispone di un oscilloscopio, un frequenzimetro o di almeno di un probe per individuare i livelli logici. Il segnale potrà essere utilizzato per verificare strumentalmente la reale frequenza di lavoro dell'oscillatore.
Però, è opportuno che, anche non disponendo della strumentazione adeguata, si segua quando esemplificato, dato che si tratta di informazione comunque importanti.

Siccome viene impiegato un pin di I/O per ottenere questa uscita, la sua selezione è effettuata nella Configuration Word iniziale e non può essere modificata durante l'esecuzione del programma.

Possiamo fare una verifica utilizzando i 12F629/675 in questa semplice configurazione:



Che il microcontroller non abbia alcunchè collegato, se non ICSP e l'alimentazione dipende dal fatto che quello che ci importa verificare è solo l'uscita della frequenza di clock.

Come prima azione nel sorgente occorrerà impostare il config iniziale in modo da abilitare l'uscita **CLKOUT** sul **GP4**:

```
; Oscillatore interno con CLKOUT, no WDT, no CP, no BOR, PWRT on, no MCLR
  __CONFIG  _CP_OFF & _CPD_OFF & _BODEN_OFF & _MCLRE_OFF & _PWRTE_ON
& _WDT_OFF & _INTRC_OSC_CLKOUT
```

Possiamo effettuare alcune prove variando il contenuto di **OSSCAL**

OSCCAL — OSCILLATOR CALIBRATION REGISTER (ADDRESS: 90h)

R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	—	—
bit 7							bit 0

I bit **CAL5:0** variano la frequenza dell'oscillatore in questo modo:

CAL5:0	Frequenza
11111	massima
10000	media
00000	minima

Se usiamo la procedura di calibrazione, sarà caricato il valore individuato in produzione come il più corretto. Modificando il contenuto del registro tra 00h e FCh (i bit 0 e 1 non sono usati) noteremo il corrispondente variare della frequenza, che, da un valore nominale di 1MHz (4MHz/4) salirà o scenderà di circa il 12%.

 **Attenzione: il valore 00h, indicato come medio, NON è il valore di calibrazione, che potrà essere molto diverso.**

Il sorgente per una simile applicazione sarà riducibile a

```
; scelta del processore
 #ifdef   __12F629
        #include <p12F629.inc>
 #endif
 #ifdef   __12F675
         #include <p12F675.inc>
 #endif
         radix  dec
;*****************************************************************
; Oscillatore interno, no WDT, no CP, no BOR, PWRT on, no MCLR
;
  __CONFIG  _CP_OFF & _CPD_OFF & _BODEN_OFF & _MCLRE_OFF & _PWRTE_ON &
_WDT_OFF & _INTRC_OSC_CLKOUT

;*****************************************************************
        ORG     0x00

        banksel OSCCAL
        call    03FFh
        movwf   OSCCAL
        goto    $
  END
```

Sul pin **GP4/CLKOUT** dovremo leggere una frequenza di 1MHz +/- 1%.

Se carichiamo valori diversi nell'**OSCCAL** dovremo rilevare variazioni della frequenza nei limiti indicati.

```
; scelta del processore
 #ifdef   __12F629
        #include <p12F629.inc>
 #endif
 #ifdef   __12F675
         #include <p12F675.inc>
 #endif
         radix  dec
;*****************************************************************
; Oscillatore interno, no WDT, no CP, no BOR, PWRT on, no MCLR
;
  __CONFIG  _CP_OFF & _CPD_OFF & _BODEN_OFF & _MCLRE_OFF & _PWRTE_ON &
_WDT_OFF & _INTRC_OSC_CLKOUT

;*****************************************************************
        ORG     0x00

        banksel OSCCAL
        movlv   0xFC       ; valore tra 0 e FCh
        movwf   OSCCAL
        goto    $
  END
```

Analogamente si potrà procedere per altri tipi di chip, individuando la corretta label per attivare la funzione di **CLKOUT**

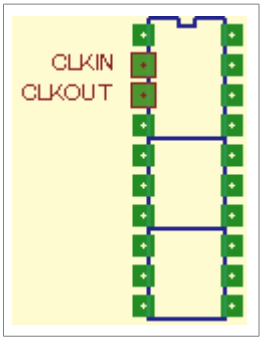
Per quanto riguarda chip con più di 8 pin, la funzione **CLKOUT** è solitamente condivisa con **RA4**.

Ricordiamo che **CLKOUT** è disponibile anche nella modalità EC, oltre che in quella con oscillatore interno **INTOSC**.

Oscillatori esterni

Oltre alle modalità interne, la maggior parte dei chip offre la possibilità di realizzare oscillatori esterni come visto per i Baseline.
Solitamente la frequenza massima è 20MHz.

Sono richiesti uno o due pin che non possono essere usati per altre applicazioni e sono determinati nella configurazione iniziale.
Questi pin sono identificati come **OSC1/OSC2** o **CLKIN/CLKOUT**; essendo **sovrapponibili pin to pin**, la disposizione è questa:



Package	CLKIN	CLKOUT
8 pin	pin 2	pin 3
14 pin		
20 pin		

La disposizione dei pin nei chip a 18 è differente e non sovrapponibile.

Rivediamo ora alcune cose già trattate parlando dei Baseline, ma di notevole importanza.

Le opzioni del clock e il config

Le possibili opzioni di funzionamento del clock sono descritte nei fogli dati dei singoli componenti, ma è possibile averne rapidamente una lista consultando non l' intero data sheet, bensì i soli files ***nomeprocessore.inc*** relativi ai vari chip.

Abbiamo visto come questo file sia l' elenco delle label pre definite delle funzioni principali del chip e che questo file sia leggibile con un qualsiasi editor.

Prendiamo, ad esempio il file Nella sezione **_CONFIG** troviamo come prime voci le modalità dell' oscillatore. Ad esempio, per p12f629.inc e lo apriamo con un editor. Troviamo questa situazione:

```
;----- CONFIG Options -----------------------------------------------
_FOSC_LP        EQU  H'3FF8'  ; LP osc.: Low power crystal on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN
_LP_OSC         EQU  H'3FF8'  ; LP osc.: Low power crystal on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN
_FOSC_XT        EQU  H'3FF9'  ; XT osc.: Crystal/resonator on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN
_XT_OSC         EQU  H'3FF9'  ; XT osc.: Crystal/resonator on GP4/OSC2/CLKOUT and GP5/OSC1/CLKIN
_FOSC_HS        EQU  H'3FFA'  ; HS osc.: High speed crystal/resonator on GP4/OSC2/CLKOUT and
GP5/OSC1/CLKIN
_HS_OSC         EQU  H'3FFA'  ; HS osc.: High speed crystal/resonator on GP4/OSC2/CLKOUT and
GP5/OSC1/CLKIN
_FOSC_EC        EQU  H'3FFB'  ; EC: I/O function on GP4/OSC2/CLKOUT pin, CLKIN on GP5/OSC1/CLKIN
_EC_OSC         EQU  H'3FFB'  ; EC: I/O function on GP4/OSC2/CLKOUT pin, CLKIN on GP5/OSC1/CLKIN
_FOSC_INTRCIO   EQU  H'3FFC'  ; INTOSC osc.: I/O function on GP4/OSC2/CLKOUT pin, I/O function on
GP5/OSC1/CLKIN
_INTRC_OSC_NOCLKOUT  EQU  H'3FFC'  ; INTOSC osc.: I/O function on GP4/OSC2/CLKOUT pin, I/O
function on GP5/OSC1/CLKIN
_FOSC_INTRCCLK       EQU  H'3FFD'  ; INTOSC osc.: CLKOUT function on GP4/OSC2/CLKOUT pin, I/O
function on GP5/OSC1/CLKIN
_INTRC_OSC_CLKOUT    EQU  H'3FFD'  ; INTOSC osc.: CLKOUT function on GP4/OSC2/CLKOUT pin, I/O
function on GP5/OSC1/CLKIN
_FOSC_EXTRCIO        EQU  H'3FFE'  ; RC osc.: I/O function on GP4/OSC2/CLKOUT pin, RC on
GP5/OSC1/CLKIN
_EXTRC_OSC_NOCLKOUT  EQU  H'3FFE'  ; RC osc.: I/O function on GP4/OSC2/CLKOUT pin, RC on
GP5/OSC1/CLKIN
_FOSC_EXTRCCLK       EQU  H'3FFF'  ; RC osc.: CLKOUT function on GP4/OSC2/CLKOUT pin, RC on
GP5/OSC1/CLKIN
_EXTRC_OSC_CLKOUT    EQU  H'3FFF'  ; RC osc.: CLKOUT function on GP4/OSC2/CLKOUT pin, RC on
GP5/OSC1/CLKIN
```

Riassumendo, sono disponibili i seguenti modi:

- **XT, HS, LP** - Oscillatore esterno a cristallo o con risonatore ceramico
- **EC** - Sorgente di clock esterna, con o senza CLKOUT
- **INTRC** - Oscillatore interno, con o senza CLKOUT
- **EXTRC** - Oscillatore RC esterno, con o senza CLKOUT

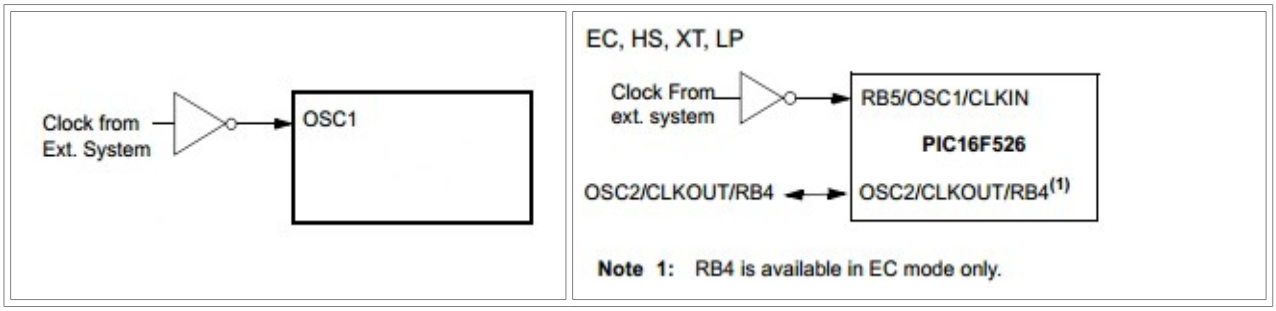
Per ognuno dei modi possono essere presenti più alias, individuabili dalla stessa descrizione. Questa situazione consente di compilare sorgenti in cui sia usata uno o l'altra delle label.

Altri chip avranno diverse opzioni.

La scelta di uno dei modi non **INTRC** dovrà essere accompagnata dalla corispondente disponibilità dell'hardware necessario

Clock esterno – EC

L'opzione **EC** (***External Clock***) corrisponde al fatto che il pin **`OSC1/CLKIN`** diventa l' ingresso per la frequenza esterna e non può assumere altra funzione.In questo caso, il pin viene predisposto dal config per ricevere il segnale esterno che dovrà avere caratteristiche compatibili con il microcontroller (livelli TTL, tempi di salita e discesa indicati nel foglio dati).



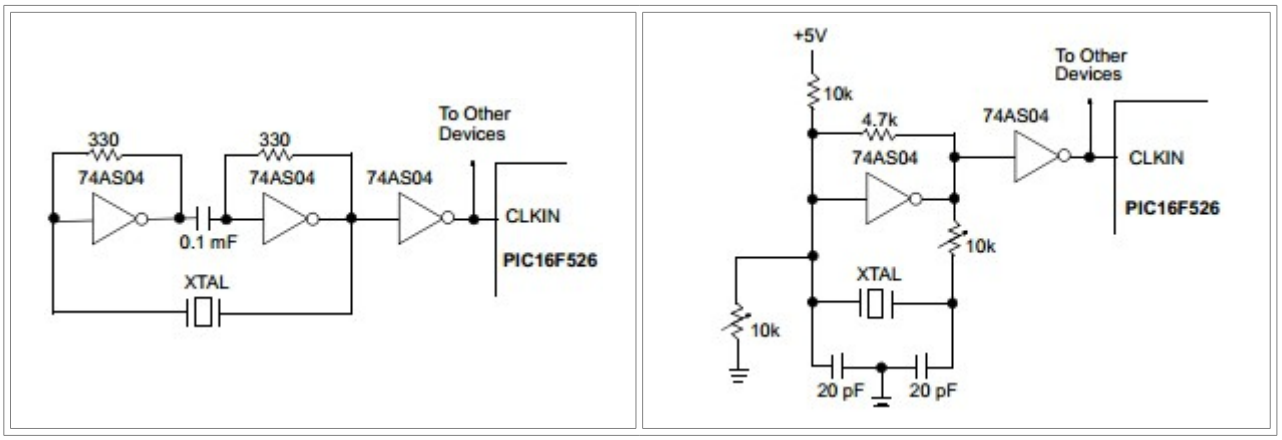
Alcuni chip dispongono solamente di questa opzione in alternativa all'oscillatore interno.
Dove è ammessa, l'opzione EC può essere accompagnata dalla funzione CLKOUT vista sopra.

Questo modo non è di uso comune, ma è applicabile dove esiste già nel sistema un generatore di clock, ad esempio quando il PIC fa parte di una scheda complessa assieme ad altri processori.

Oppure quando è necessaria una specifica frequenza che viene prodotta da un oscillatore apposito, di cui ne esistono moltissimi modelli con varie caratteristiche di precisione e stabilità e con la temperatura e di dimensioni diverse.

L'uso di questi componenti è solitamente dettato dalla necessità di elevate precisioni e stabilità.

Nel caso in cui non si impieghi un oscillatore integrato, ma si realizzi con gate, occorre utilizzare TTL (LS, AS, ecc) o CMOS HCT per avere il giusto andamento del segnale.
I fogli dati forniscono alcuni esempi di possibili oscillatori esterni.

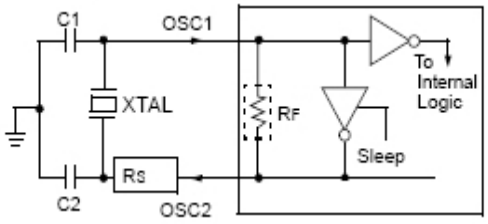


Rispetto al clock integrato, una fonte esterna può servire più processori o altri circuiti.
Rispetto ai 4/8MHz del clock interno, il clock esterno può salire a 16-20MHz, con il corrispondente aumento di prestazioni: se a 4MHz una istruzione è eseguita in 1 us, a 20MHz richiede solo 200ns.

Il foglio dati del componente specifica la frequenza massima applicabile in funzione della tensione di alimentazione, che, come abbiamo visto, è proporzionale.

Clock esterno - LP, XT, HS

Possiamo realizzare un oscillatore controllato a cristallo (quarzo o oscillatore ceramico), da pochi kHz a un paio di decine di MHz, aggiungendo alcuni componenti esterni:



La struttura interna dei gate corrispondenti ai pin **`OSC1/OSC2`** viene modificata al config in modo da realizzare un oscillatore che usa come elemento di stabilità e frequenza un cristallo di quarzo oppure un elemento ceramico. Occorre aggiungere la coppia di condensatori ***C1/C2***: solitamente si tratta di ceramici NP0 tipicamente da 18-22pF.
La resistenza ***Rs*** va aggiunta utilizzando cristalli low power per limitare la corrente.

La resistenza ***Rf*** il cui scopo è portare il gate in zona di funzionamento lineare, è integrata ed ha un valore di circa 10Mohm. Solitamente non ne occorre una esterna .
Da notare che questa opzione impegna entrambi i pin **`OSC1/OSC2`** che, pertanto, non potranno avere altra destinazione. Esistono tre modi con si configura il gate dell' oscillatore:

Modo	Frequenza	C1/C2
LP	32kHz	15-18pF
XT	200kHz-4MHz	15-68pF
HS	4-20MHz	15-47pF

L' opzione **LP** (***Low Power***) viene scelta tipicamente per quarzi a **32768Hz** usati per generare tempi precisi, orologi, data logger e simili. Questo genere di cristalli funzionano con una potenza di 1-2 microwatt, mentre la maggior parte dei quarzi HF possono dissipare da 100 a 500 microwatt.
Per questo motivo, quando è necessario limitare la dissipazione di potenza: un eccesso di corrente può non impedire l' oscillazione, ma questa avviene su armoniche elevate ed è instabile e casuale, oltre al fatto che una driving con potenza eccessiva accorcia la vita del cristallo.
Da qui potrebbe rendersi necessaria la ***Rs***, principalmente per elementi ultra low power del genere usato in orologeria.

Nella pratica, la scelta dei componenti per l' oscillatore LP può essere laboriosa e può non essere immediato trovare un cristallo adeguato al funzionamento su OSC1/OSC2 in LP, sopratutto tra i recuperi da smontaggio di schede vari; inoltre, la frequenza generata è legata al valore dei condensatori in modo molto più deciso che per i modi HS e XT, che non sono troppo critici.
Per LP si va da 6-12pF a un centinaio e può essere necessario avere valori diversi per C1 e C2 per ottenere la giusta frequenza di oscillazione.

Anche se dai condensatori dipende molto la frequenza generata, usando valori inadeguati, se il quarzo è adatto, il funzionamento difficilmente può mancare; con valori dell'ordine di 33-100pF, l' oscillazione può essere anzi più stabile, ma il tempo di start-up (avvio) dell'oscillatore può diventare sensibilmente elevato (anche secondi !).

L'opzione **XT** (***XT****al*) va usata con cristalli o risonatori ceramici da 200kHz a 4MHz. Rispetto al modo precedente, cambia l'energia disponibile per l' oscillatore, che viene aumentata. I condensatori, tra 18 e 47pF non sono particolarmente critici, ma occorrono elementi di buona qualità.

L' opzione **HS** (***H****igh* ***S****peed*) **è** da adottare per cristalli da 4MHz in su, dove l' energia richiesta dall' oscillatore è massima (la già citata necessità di maggiore energia con l' aumentare della frequenza).

Comunemente, **XT** e **HS** sono quelli più impiegati per elementi esterni: la scelta di uno o dell' altro dipende dal tipo di cristallo usato e va definita, assieme ai valori dei condensatori, per ottenere una oscillazione stabile in tutta la gamma di tensioni di alimentazione e temperature a cui il microcontroller sarà sottoposto. E' possibile che uno stesso cristallo possa funzionare in entrambi i modi, ma probabilmente uno dei due darà una forma d'onda ed una stabilità migliore.

Alcune note:

- una volta impostato un modo dell' oscillatore con componenti esterni, questi dovranno essere cablati come previsto, pena il mancato funzionamento del microcontroller.

- scegliendo nel config un modo inadeguato ai componenti usati è probabile che l'oscillatore non si avvii o che funzioni in modo casuale e su frequenze diverse da quella nominale

- elevata stabilità si ottiene solo utilizzando componenti di qualità. I condensatori dovranno essere ceramici a coefficiente di temperatura neutro (NP0), le resistenze a strato metallico e il cristallo del tipo adatto per questo uso, evitando elementi di recupero malamente dissaldati (in quanto un eccesso di temperatura sui pin fa decadere le prestazioni del cristallo). In ogni caso, la frequenza generata non sarà più precisa o stabile di quanto lo siano i componenti utilizzati.

- le voci del **config** relative ai modi dell' oscillatore sfortunatamente non sono univoche per tutti i PIC, tanto che Microchip ha aggiunto nel file ***.inc*** vari alias. Questo non evita che essi possano essere insufficienti, rendendo necessario leggere il file per reperire le giuste label.

Come vediamo dalle esercitazioni, la gran parte delle applicazioni semplici e anche meno semplici lavora correttamente con l' oscillatore interno. I fogli dati riportano una precisione dell' ordine dell'1%, più che adeguata anche per applicazioni abbastanza critiche, come una trasmissione seriale asincrona, mentre la stabilità con la temperatura è adeguata nell' ambito delle variazioni comuni.

Inoltre, la calibrazione dell' oscillatore interno permette di variare entro un certo ambito la

frequenza generata e consente sia di ottenere valori leggermente diversi dalla fondamentale, sia di aggiustarne il valore in conseguenza di forti variazioni della temperatura.

NOTA IMPORTANTE

Che molti esempi sul web riportino il cristallo e i condensatori come elemento tipico di un circuito con PIC dipende essenzialmente dal fatto che **il chip non dispone di clock interno**, **come i PIC obsoleti** (es. 1**6F84, 16F876/877**).
Però, spesso, dipende solamente dalla mancata conoscenza delle possibili configurazioni alternative dell'oscillatore e, soprattutto, del fatto che i PIC meno obsoleti hanno tutti l'opzione di uno più frequenze disponibili attraverso l'oscillatore interno.

Anche a riguardo della precisione, va detto che un cristallo serve solamente nelle applicazioni time-sensitive, ovvero che richiedono una elevata precisione nelle temporizzazioni.

Dal punto di vista più generale, a meno che sia necessaria una frequenza estremamente precisa/stabile o il chip non abbia oscillatore interno, usare un cristallo esterno a 4MHz o, peggio, un RC esterno **quando esiste l'oscillatore interno è del tutto inutile e sacrificano pin altrimenti utilizzabili per l' I/O**.

Gli oscillatori esterni vanno usati quando:

- il chip non dispone di oscillatore interno (chip obsoleti)
- è necessaria l' elevata stabilità del quarzo, che è maggiore di quella dell' oscillatore interno, soprattutto al variare della temperatura.
- sono necessarie frequenze precise di valori diversi da quelli ottenibili dall' oscillatore interno, ad esempio 32KHz per RTC o 2.476MHz per le trasmissioni seriali, ecc.

Una nota ulteriore per chi volesse fare misure con l' oscilloscopio sull' oscillatore a cristallo: il pin da testare è **`OSC2/CLKOUT`**, che è quello a impedenza minore e dove la sonda dell' oscilloscopio o frequenzimetro crea il minore disturbo. Applicando la sonda su **`OSC1`** si caricherà l' oscillatore con l' impedenza e la capacità della stessa.

Microchip ha emesso alcuni documenti sugli oscillatori:

- **AN826** - Crystal Oscillator Basics and Crystal Selection for rfPICTM and PICmicro® Device
- **AN949** - Making Your Oscillator Work
- **DD31002a** - Oscillator

e qui **trovate altre pagine sull' argomento**.

In ogni caso, utilizzando componenti adeguati ed un cablaggio decente, non abbiamo mai verificato problemi nell'implementazione di oscillatori a cristallo.
Se avete problemi in questo ambito, prima di accusare il chip o altro:

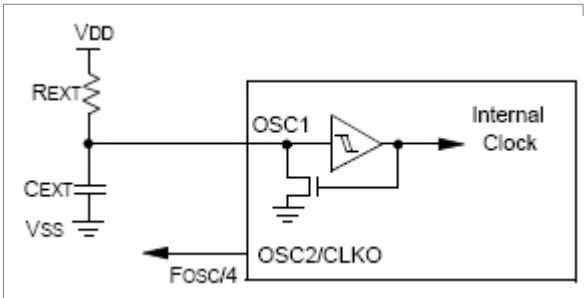
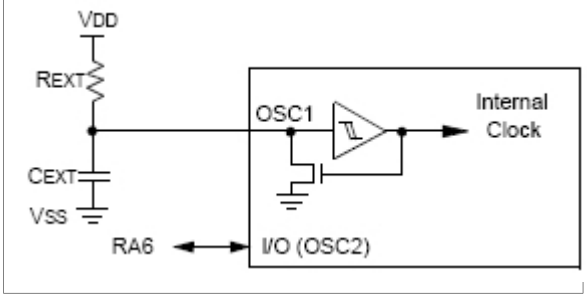
- **verificate i componenti e la correttezza dei collegamenti, sopratutto se state usando breadbord che possono avere contatti di inserzione danneggiati**
- **verificate la corretta impostazione dell'oscillatore scelto nel config**.

Se qui è tutto a posto, la causa è certo da cercarsi nel cristallo che è di tipo non adeguato (ad esempio, elementi che sono previsti per lavorare overtone, come i quarzi per trasmissione, o cristalli per low power inadeguati all' uso con microcontroller) oppure è danneggiato, come capita con elementi di recupero.
Raramente i condensatori pongono problemi; usando componenti nuovi, ceramici o ceramici mutistrato, NP0 o a bassa deriva, non ci sono mai problemi nell' avvio dell' oscillazione, anche se i valori non perfettamente centrati posso portare ad un qualche piccolo spostamento nella frequenza.

In relazione alla possibilità di aggiustamento della frequenza, si può considerare la possibilità di usare un trimmer capacitivo assieme a ***C1*** per aggiustarne la capacità in funzione delle caratteristiche del quarzo.

Clock esterno - RC

Un oscillatore esterno può essere realizzato anche usando una **rete RC**. Abbiamo due possibilità:

	La prima consente di avere la rete RC esterna, collegata su **OSC1**, e su **OSC2** ottenere in uscita la **Fosc/4**.
	La seconda consente di avere la rete RC esterna, collegata su **OSC1**, e su **OSC2** avere disponibile la funzione di I/O digitale del pin.

Questo modo va utilizzato in un unico caso:

- quando **occorre una frequenza non disponibile internamente**,
- ma **di cui non importa nè precisione, nè stabilità** (*timing insensitive application* - applicazioni non sensibili alle temporizzazioni)
- e la richiesta progettuale è il minimo costo

Infatti, una certa precisione e stabilità sono ottenibili solamente utilizzando per R e C componenti di elevata qualità: se resistenze di precisione e stabili col variare della temperatura sono facilmente reperibili, lo stesso non si può dire dei condensatori. E se la ricerca nel progetto è relativa al basso costo, questo non si ottiene certo usando resistenze a strato metallico allo 0.1% e condensatori in mica.

Oltre a questo, ottenere un valore preciso di frequenza è spesso aleatorio, data la disponibilità di valori per R e C solamente nelle serie E e della loro tolleranza intrinseca.

Anche se l' RC esterno è teoricamente impiegabile fino alla frequenza massima di lavoro del chip, questo modo è consigliabile e necessario quando si vogliano **clock con frequenze basse**, non possibili con i cristalli, ad esempio 1 o 2kHz.
In questo senso va fatta una considerazione fondamentale:


in effetti, i PIC sono dispositivi "**statici**", nel senso che **non hanno la necessità di un clock minimo per poter funzionare.**
Il clock parte da 0, ovvero, può essere interrotto, lasciando in sospeso l' esecuzione del programma. Questa è la chiave del funzionamento del modo **sleep**, dove lo stop al processore avviene proprio arrestando il clock.

E' ovvio che un basso clock provoca un tempo di ciclo istruzione proporzionalmente lungo: ad esempio, 4MHz di clock danno un tempo di esecuzione di 1us, ma 4kHz portano il tempo di esecuzione a 1ms per istruzione.

Per contro, **il consumo energetico si riduce drasticamente**: per PIC16F506, ad esempio, la corrente di alimentazione è valutata 1.4mA @ 5V con un clock di 20MHz e passa a soli 11uA @ 32KHz, con la possibilità di abbassare la tensione Vdd a 2V (foglio dati, param. D010).

Dal punto di vista realizzativo, le specifiche indicano che la ***Rext*** deve essere compresa tra 3 e 100Kohm, mentre Cext può essere anche omesso utilizzando le capacità parassite del circuiti, ma questa scelta è del tutto sconsigliata per evidenti ragioni. Quindi ***Cext*** potrà partire da 20pF in su.
La sezione "***Electrical Characteristic***" del foglio dati indica le specifiche dell' applicazione.

L' oscillatore RC è basato sulla carica del condensatore Cext attraverso la resistenza Rext. Raggiunta circa la tensione 0.75xVdd, un transistor interno scarica rapidamente il condensatore e la tensione cade a circa 0.25Vdd, ottenendo la tipica onda triangolare, che sarà poi squadrata dai circuiti successivi (vedi schemi della tabella precedente). E' possibile calcolare il periodo dell' oscillazione ottenibile da una combinazione RC con la formula:

$$T = (R*C)\ ln\ [\ Vdd / (Vdd-Vih)]$$

dove è la tensione di commutazione dello Schmitt trigger posto all' ingresso del pin OSC1 e del valore di circa 0.9Vdd (param. D043).
Ad esempio, per ***Vdd=5V, R=10k e C=22pF*** si ha:

$$T = (10k*22pF)\ ln\ [\ 5\ /\ (5-4.5)] = 506ns$$

Per cui la frequenza sarà:

$$F = 1/\ T = 1\ /\ 506ns = 1.976MHz$$

e: $Fosc/4 = 494kHz$

Se consideriamo la sola alimentazione a 5V, possiamo usare una formulazione abbreviata:

$$T = R\ *C\ *\ 2.3$$

Ovviamente il valore reale ottenuto dipende dalla tolleranza dei componenti R e C usati e dalle caratteristiche e tolleranze costruttive del chip. In più, va considerata la loro variazione con la temperatura.

In pratica.

Si potranno utilizzare quarzi fino a 20MHz. Vanno bene cristalli in contenitore HC49 o comunque adatti per l' impiego con oscillatori del genere. I condensatori potranno essere ceramici di qualità da 15-27pF.

Potremo usare anche risonatori ceramici a due pin o quelli a tre pin; in quest'ultimo caso non occorrono i condensatori esterni perchè sono integrati nel componente.

Ecco come si presentano esternamente alcuni di questi componenti:

Cristallo: 1-20MHz modo **XT** o **HS**	Cristallo: 32-200kHz modo **LP**	Risonatore ceramico 2 pin modo **XT** o **HS**	Risonatore ceramico 3 pin modo **XT** o **HS**
			

Se usate componenti di recupero, assicuratevi che si tratti di oggetti adeguati all'uso con microcontroller.
In particolare, non sono adatti cristalli per oscillatori serie, cristalli per oscillatori in armonica, filtri

ceramici di MF, cristalli di grosse dimensioni recuperati da apparati valvolari (che richiedono maggiore energia di quella disponibile), ecc.

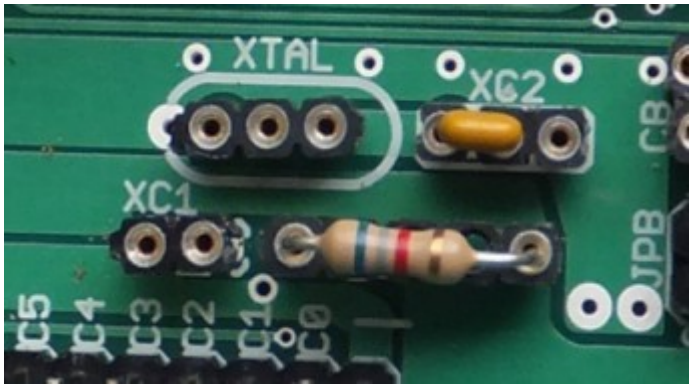
La scelta del **modo LP** è obbligatoria per i **quarzi low power da 32kHz** usati in orologeria; per modelli di dimensioni molto miniaturizzate può essere necessario aggiungere la ***Rs*** come da schema precedente, il cui valore potrà variare da qualche kohm a 200-400kohm, a seconda del cristallo usato.

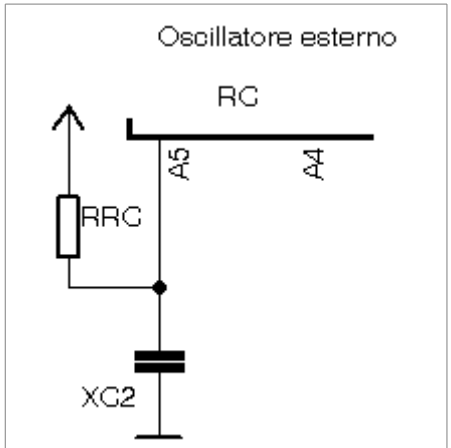
Nei modi con cristallo esterno normalmente non occorre una resistenza esterna tra i pin dell'oscillatore (Rf nello schema), anche se è possibile aggiungerne una di valore elevato (1-10Mohm) con funzione di garantire una oscillazione sicura.

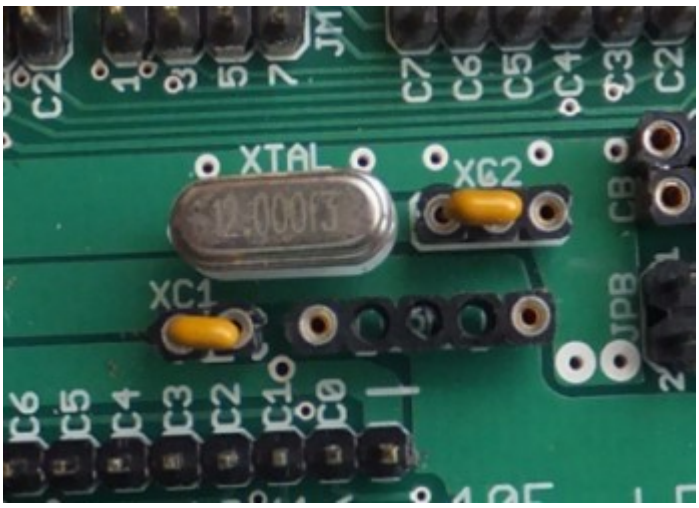
Per altri quarzi, solitamente per frequenze al di sotto dei 4MHz, va bene il modo **XT**, ma se il cristallo è "duro" va impostato **HS**, anche per frequenze minori. Questo può rendersi necessario soprattutto con oscillatori ceramici.
Nominalmente, **HS** va scelto per cristalli al di sopra dei 4MHz.

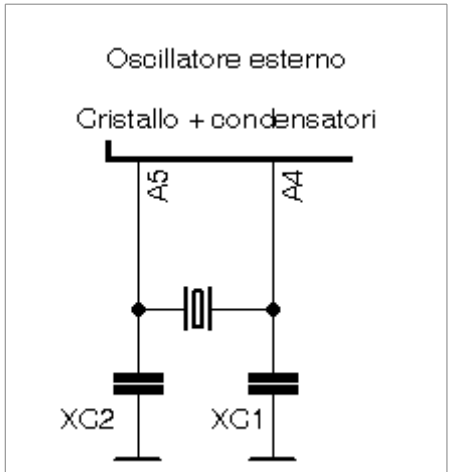
La scheda di sviluppo dispone della possibilità di sperimentare tutte le possibili variazioni dell' oscillatore esterno in quanto esiste un' area dedicata ai componenti.

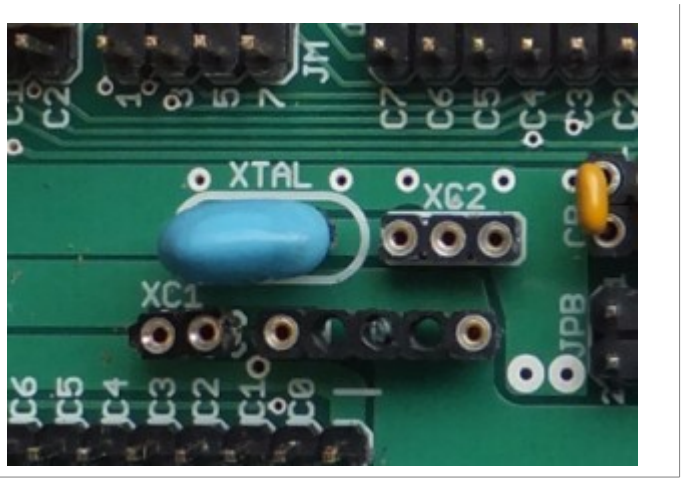
Nel reale, le foto seguenti presentano un ingrandimento dell' area dei componenti dell' oscillatore:

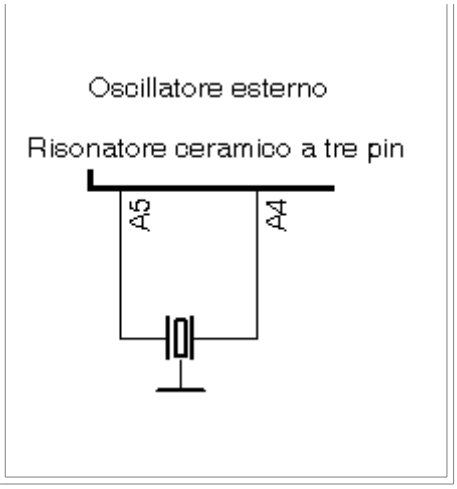












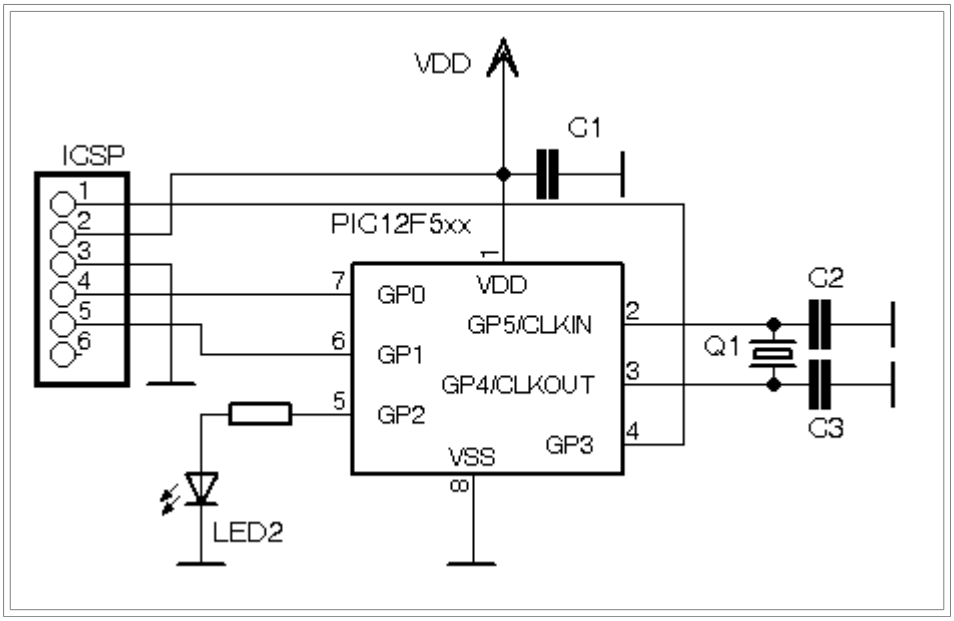
Osserviamo che il risonatore ceramico a tre pin integra i condensatori, per cui non occorre aggiungerli.
la scelta dell' oscillatore ceramico al posto del quarzo è dettata dal minore costo di questo componente e dalla sua funzionalità, seppure a fronte di minore precisione/stabilità.

Volendo sperimentare questi oscillatori, sarà necessario:

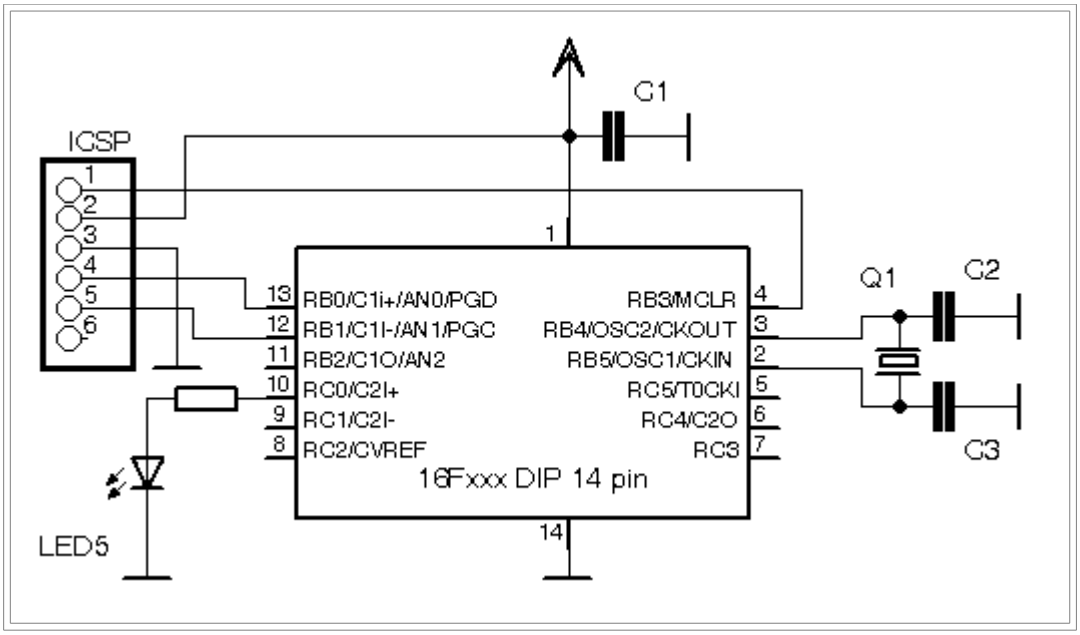
- inserire i componenti richiesti da ogni schema
- considerare che uno o due pin non possono essere utilizzati per altri scopi
- e che nel sorgente va modificato il config in funzione del modo scelto

Se operiamo su supporti diversi dalla **LPCuB** basterà realizzare i circuiti indicati, avendo cura di verificare che le connessioni siano sicure, soprattutto su breadboard.

Si possono sperimentare oscillatori esterni a cristallo con i seguenti schemi:

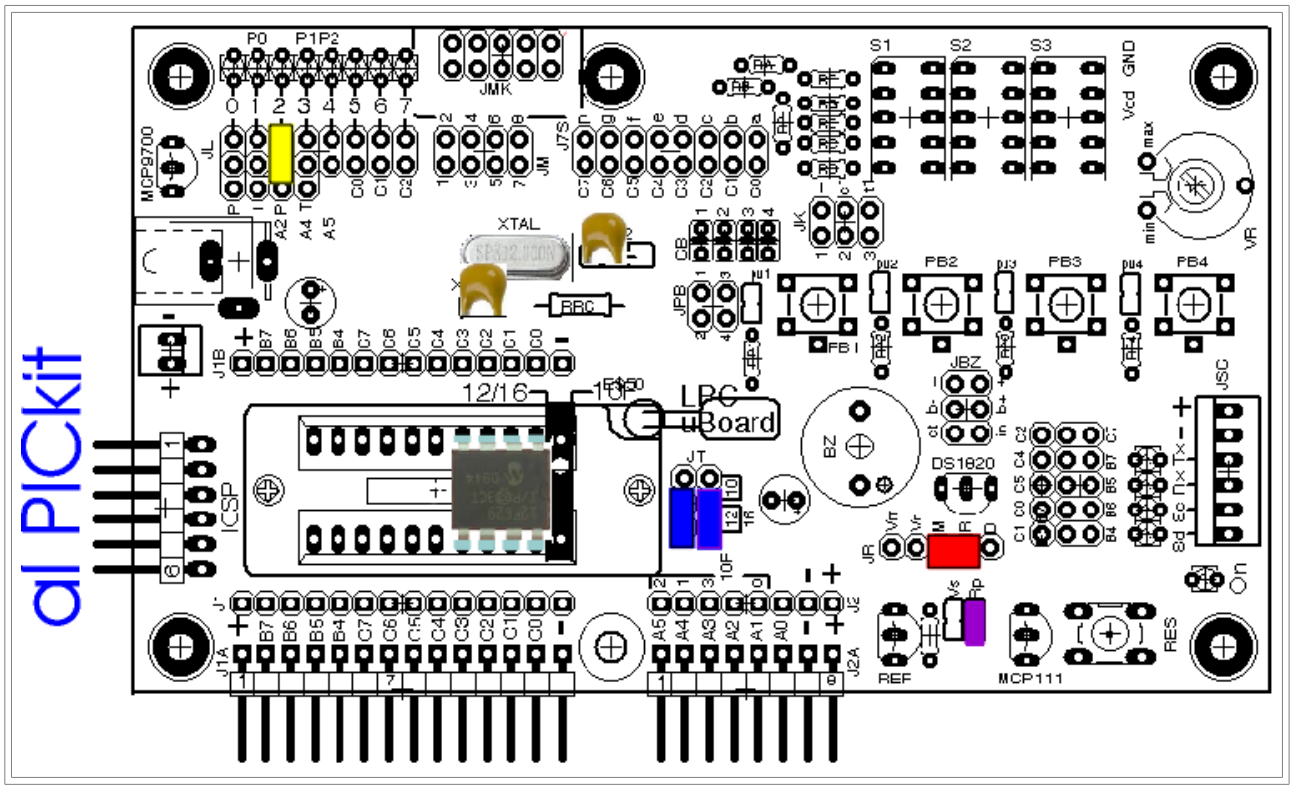


Si potranno utilizzare quarzi fino a 16-20MHz, anche se il chip funziona senza problemi con frequenze maggiori, ma si tratterebbe di "overclock", condizione non certificate dal costruttore.
Per i 16Fxxx:

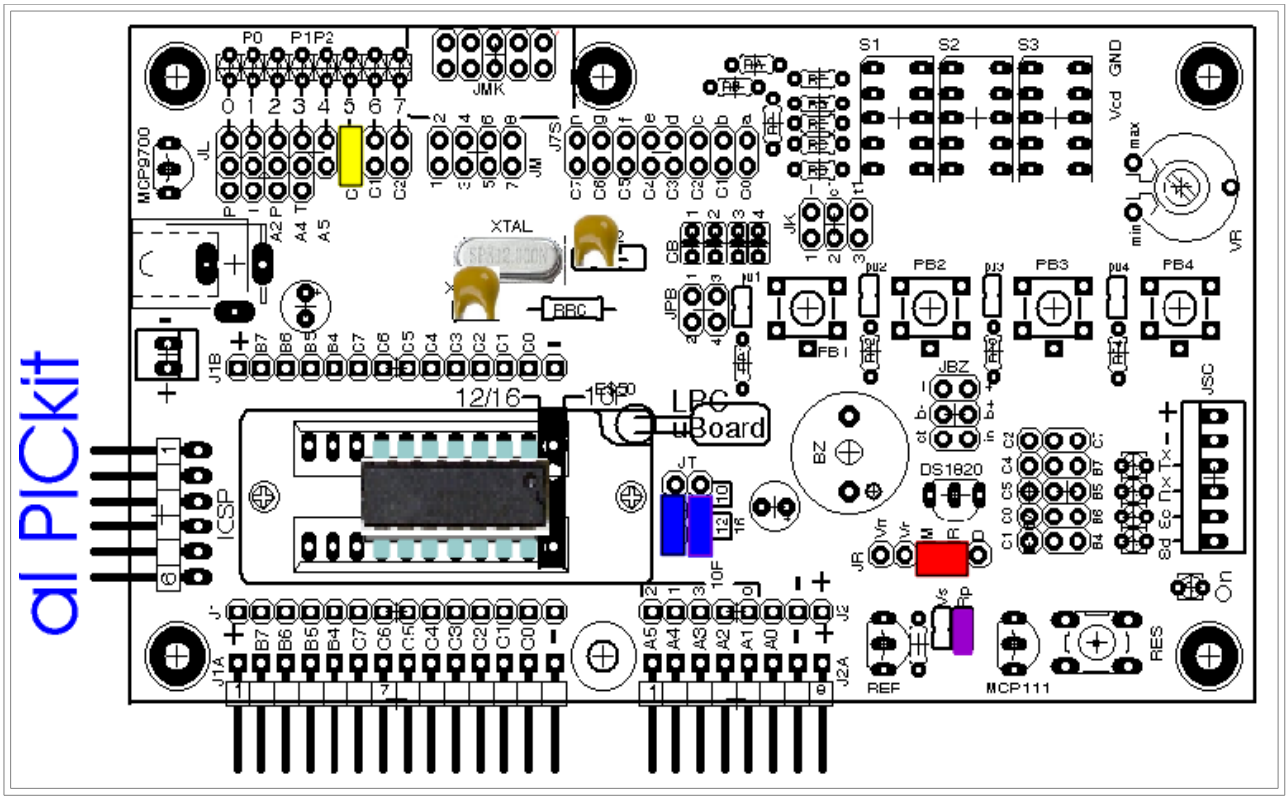


E così pure per i 20 pin, sempre impegnando i soliti OSC1/OSC2.

Per quanto riguarda la disposizione sulla board di sviluppo **LPCuB**, per i DIP-8 pin:

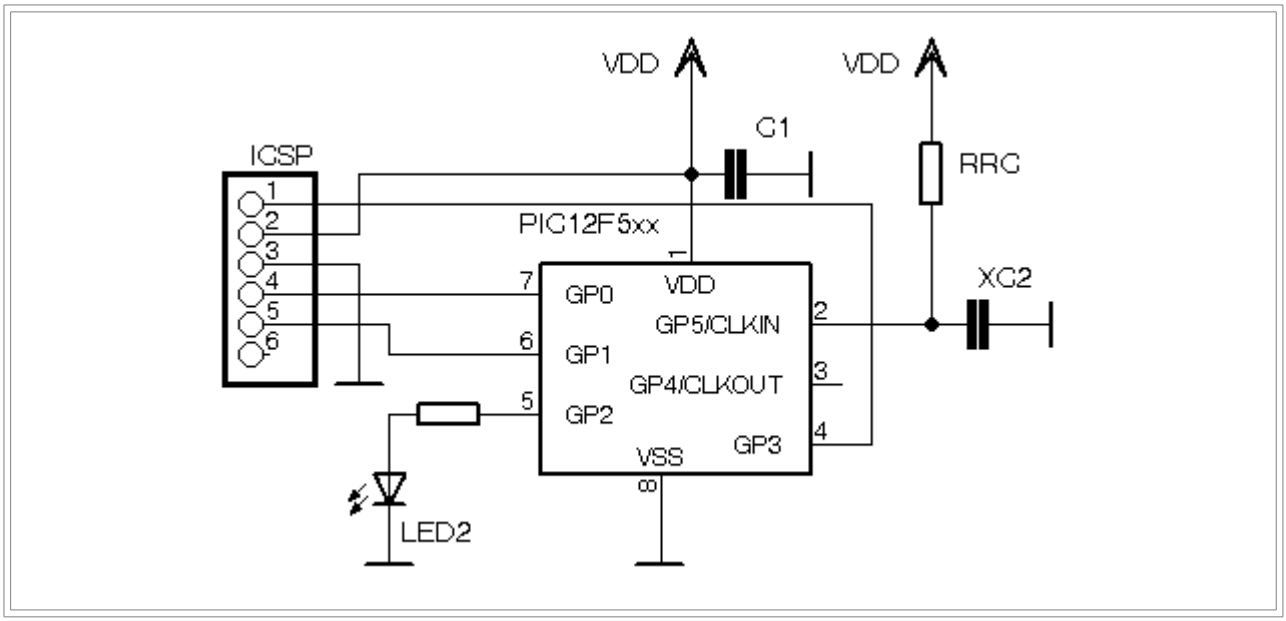


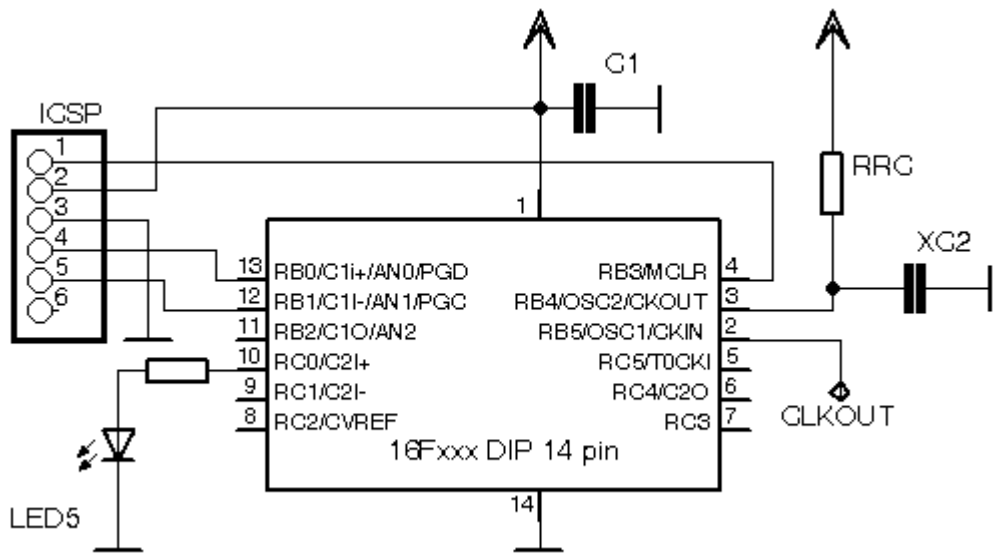
e per i DIP-14 pin:



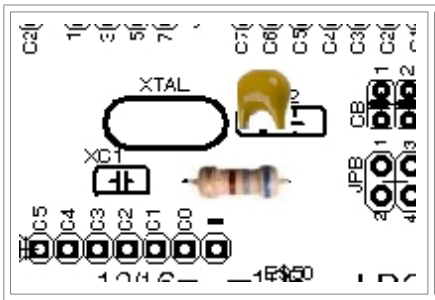
Modalità ExtRC

Possiamo sperimentare anche questo oscillatore.





Nel caso dei PIC a 14pin possiamo abilitare la modalità ExtRC-Clkout e verificare sul pin **RB5/OSC2** l' uscita della frequenza generata/4.



La disposizione dei componenti è identica alla precedente, con la differenza che dovranno essere inseriti la ***Rext*** (**RRC**) e il ***Cext*** (**XC2**)

Lo zoccolo di ***XC2*** consente di installare condensatori passo 2.54" o 5.08". Osservate nella disegno il posizionamento del condensatore col passo più stretto.

Usando una resistenza da 33k, possibilmente di buona qualità (strato metallico, 1 o 2%) ed un condensatore da 33pF (multistrato ceramico o film plastico) dovremmo ottenere una frequenza di oscillazione di:

T = 2.3 x 33 x 33 = 2504 ns -> 399.5Hz

che corrisponde ad un ciclo di istruzione di poco inferiore a 10ms.

Ovviamente potete utilizzare qualsiasi altra combinazione, adeguando il software alla scelta fatta.

Conclusioni

Riassumiamo alcune definizioni:

- **Fosc** : ***Frequency of the oscillator*** - la frequenza dell'oscillatore, di qualunque tipo sia
- **Fosc/4** : la frequenza interna relativa ad un ciclo-istruzione, pari a 1/4 della frequenza dell' oscillatore
- **tcyc** : il tempo di esecuzione (ciclo) di una istruzione

Come possiamo facilmente capire, i diversi tipi di clock non influenzano le istruzioni se non per il diverso tempo di esecuzione. Ricordiamo che questo tempo è dato da:

$$t = 1/Fosc/4$$

Per avere una idea della variazione:

Fosc	Fosc/4 [MHz]	tcyc [us]
32768 Hz	0.008192	122.07
1 MHz	0.25	4
4 MHz	1	1
8 MHz	2	0.5
16 MHz	4	0.25
20 MHz	5	0.2

- Una variazione nella modalità del clock comporta l' obbligo (oltre alla necessità eventuale di aggiungere componenti esterni) della corretta configurazione iniziale. Ricordiamo che la configurazione non fa parte del programma vero e proprio, ma è indispensabile nel sorgente per fissare le caratteristiche dell'oscillatore.
- Aumentando la ***Fosc***, diminuisce il tempo di esecuzione. Contemporaneamente occorre adeguare le routines di tempo, sia che si basino su istruzioni, sia che impieghino un timer; una routine che impiega 10ms a 4MHz, impiegherà solo 5ms a 8MHz e 40ms a 1MHz.

- Ovviamente, l'aumento della frequenza dell'oscillatore, riducendo il tempo di esecuzione delle istruzione, consente maggiori performances del processore.

- il consumo di energia del microcontroller è grosso modo proporzionale al quadrato della frequenza. Applicazioni a basso consumo dovranno utilizzare basse frequenze di lavoro.

- la frequenza di lavoro del microcontroller è proporzionale alla tensione di alimentazione: al diminuire della tensione si riduce anche la possibilità di operare con frequenze elevate.

Per quanto riguarda i vari modi di oscillatore, dove possibili, la scelta va fatta secondo queste considerazioni:

- **oscillatore interno** : praticamente adatto a tutte le applicazioni dove il tempo di esecuzione ammette una tolleranza attorno all'1%. Presenta il doppio vantaggio di non necessitare di componenti esterni e di lasciare due pin liberi come I/O.

- **oscillatore esterno a cristallo** : necessario solamente per i processori che non dispongono di oscillatore interno, in quelle applicazioni dove tempo di esecuzione richiede maggiore precisione e stabilità nel tempo/temperatura oppure è necessaria una frequenza non disponibile attraverso l' oscillatore interno. Occorrono componenti esterni e si occupano i due pin dell' oscillatore.

- **oscillatore esterno** : da utilizzare se nel circuito è già presente una sorgente di clock o se necessita una stabilità e precisione fornibile da oscillatori integrati (TCXO e simili) oppure è richiesta una frequenza non disponibile attraverso l' oscillatore interno. Occupa un pin per l' ingresso del clock esterno.

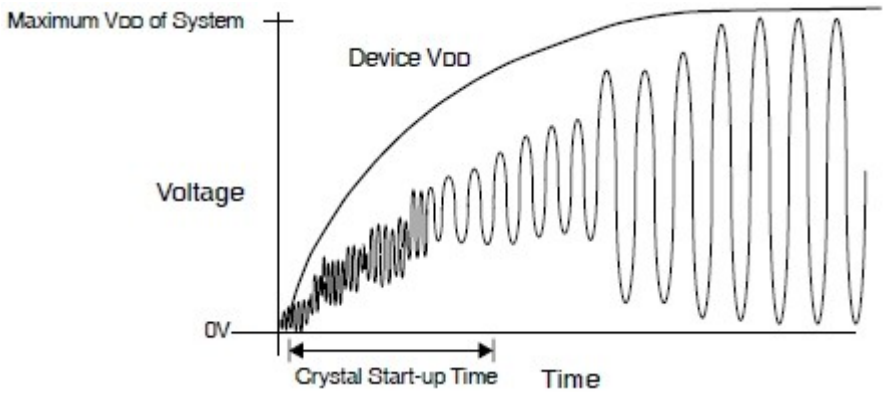
- **oscillatore RC** : può essere utilizzato quando precisione e stabilità del clock non sono elementi determinanti, ma serve una frequenza che non è disponibile attraverso l' oscillatore interno. Occupa un pin per l' ingresso del clock esterno.

In ogni caso, la precisione delle temporizzazioni nel programma dipende dalla precisione della Fosc.

Altre informazioni.

Possono essere utili ulteriori informazioni. Queste fanno riferimento ad aspetti dell'hardware normalmente non presi in considerazione e che solitamente sono riservati ad applicazioni di un certo livello, ad esempio, il tempo di startup dell'oscillatore.

A partire dall'istante dell'applicazione della tensione di alimentazione, come essa aumenta da Vss verso Vdd, l'oscillatore si avvia.



Il tempo che impiega l'oscillazione a iniziare dipende da molti fattori. Questi includono:

• Il tipo di oscillatore implementato
• Frequenza del cristallo o del risonatore
• I valori dei condensatori usati
• Il tempo di salita della tensione di alimentazione
• La temperatura del sistema

• Il rumore del sistema

Vediamo dal grafico che, in ogni caso, all'apparire della tensione la frequenza nominale e stabile dell'oscillatore è raggiunta dopo un certo tempo. Questo può essere importante quando la sua grandezza è superiore a quella del ciclo di istruzione, nel senso che possono essere eseguite più istruzioni mentre il clock si assesta. In applicazioni critiche può essere indispensabile avviare parti del programma solamente quando l'oscillatore si sia stabilizzato.

A questo scopo, il registro **OSCCON** contiene, come abbiamo visto prima, dei flag (**HTS**, LTS, OSTS) che indicano, andando a livello1, la stabilizzazione dell'oscillatore.
L'interrogazione di questi flag permetterà di individuare il momento in cui le successive istruzioni potranno essere eseguite nella certezza di un clock adeguato.

Vedremo applicate le nozioni qui acquisite durante le varie esercitazioni.