

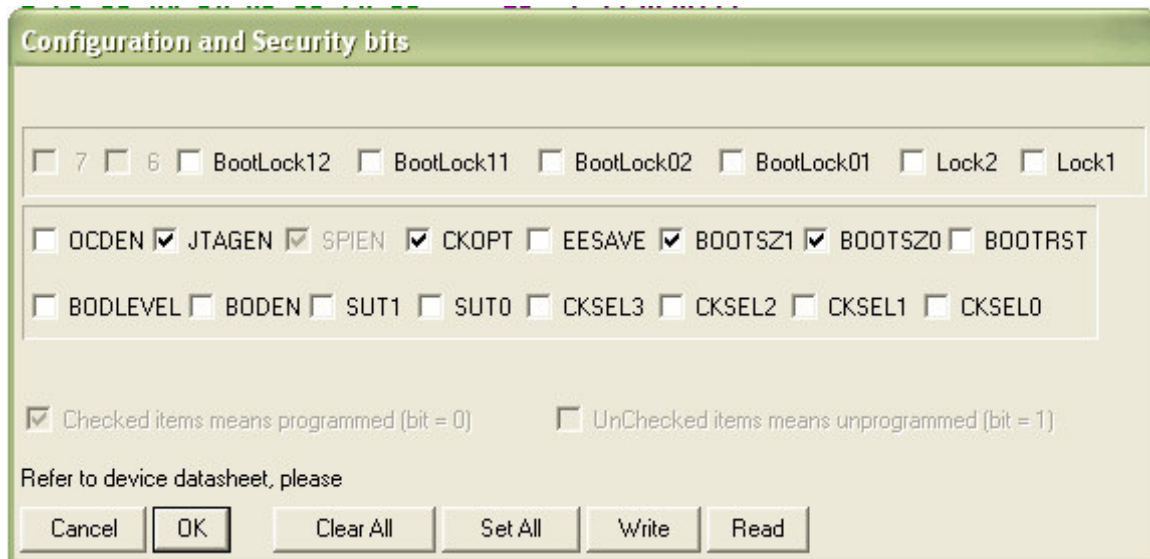
Tutorial pentru configurarea ceasului ATmega16

Nu iti merge interfata seriala? Ai incercat sa setati ceasul microcontrollerului si acum nu-l mai poti programa? Nu reusesti sa te sincronizezi cu alte dispozitive? Probabil ca nu ai configurat corect ceasul microcontrollerului...

Foarte multe dintre problemele ce ne-au fost semnalate in legatura cu montajul de proiect s-au datorat configurarii incorecte a ceasului. Chiar si eu m-am incurcat in PonyProg atunci cand a trebuit sa configurez ceasul microcontroller-ului.

Povestea scurta este urmatoarea: microcontrollerul este echipat cu un oscilator intern. In setarea implicita, microcontrollerul foloseste acest oscilator la frecventa de 1 MHz. Pentru a putea folosi cristalul ca si sursa pentru semnalul de ceas sunt necesari urmatoorii pasi:

1. Conecteaza cablul serial pe portul de programare
2. Deschide Ponyprog
3. Incearca sa programezi microcontrollerul cu un program oarecare. Daca programarea reuseste, mergi la pasul 4. Altfel, corecteaza problema si reia de la pasul 1.
4. Alege optiunea "Security and Configuration Bits..." din meniul Command
5. Realizeaza urmatoarea configuratie:



6. Uita-te de doua ori sa te asiguri ca setarea este intocmai cu cea din figura de mai sus!!!
7. Da comanda Write

Asta e tot. Acum microcontrollerul foloseste ca sursa pentru semnalul de ceas oscilatorul de pe placa.

Nu merge?

In cazul in care nu mai poti accesa microcontrollerul in urma acestei operatii, probabil ca ai probleme cu circuitul (de ceas, cel mai probabil). Incearca sa montezi microcontrollerul pe placuta altui coleg si sa testezi daca merge.

Daca merge, atunci ai in mod sigur o problema cu circuitul. Daca nu merge, atunci aproape sigur nu ai respectat pasul 6. Adreseaza-te asistentului de proiect pentru ajutor sau citeste mai departe.

Am citit tutorialul asta prea tarziu...

Iti spuneam ca si eu am gresit atunci cand a trebuit sa configurez ceasul. Mi-am dat seama abia apoi ca facusem cea mai comuna greseala. Sa speram ca fix pe asta ai facut-o si tu.

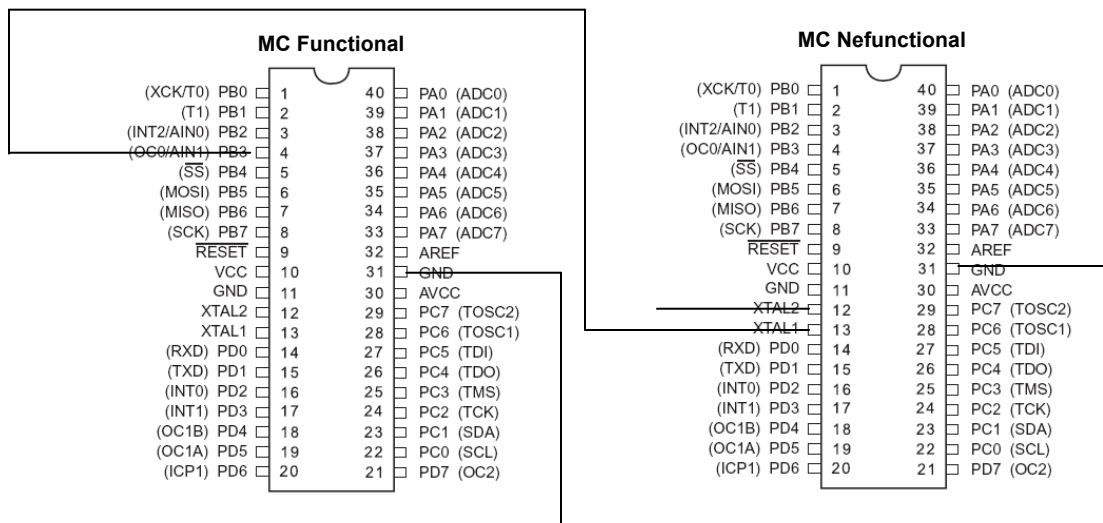
In fereastra Ponyprog scrie asa: "Checked items mean programmed (bit=0)". Deci bifat inseamna 0. Daca in datasheet ai vazut ca pentru oscilator extern bitii CKSEL trebuie sa fie 1 si i-ai bifat fara sa citesti nota din fereastra Ponyprog, atunci ai selectat ca si sursa pentru ceas "External Clock". Asta inseamna ca ai nevoie de semnal de ceas extern pentru a putea repara microcontrollerul.

Cel mai simplu, roaga un coleg sa te ajute cu placa lui (functionala). Ce aveti de facut este sa programati placa buna cu urmatorul program :

```
#include <avr/io.h>

void main(void){
    DDRB = 0xFF;
    TCCR0= 0x99;
    OCR0 = 1;
}
```

Asta face microcontrollerul sa outputeze semnal de ceas pe pinul OC0 (aka pinul 4 sau PB3). Realizati urmatoarele conexiuni:



Cu alte cuvinte, ce ai de facut este:

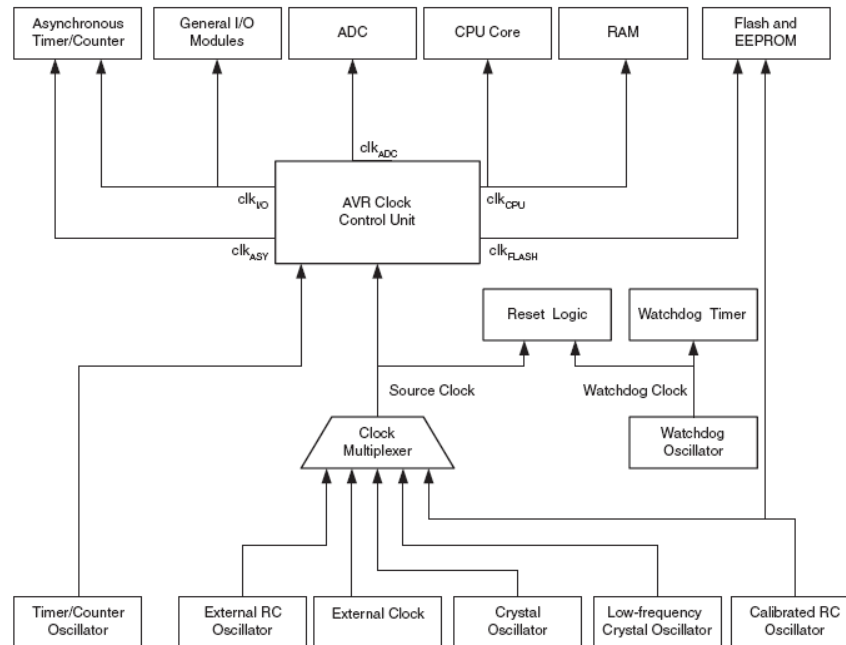
1. Sa deconectezi cristallul de la microcontrollerul bolnav
2. Sa conectezi iesirea OCO de la microcontrollerul bun la intrarea de ceas XTAL1 a celui nefunctional, lasand XTAL2 in aer
3. Sa conectezi apoi punctele de masa ale celor doua montaje
4. Sa alimentezi cele doua placute

In acest moment, daca incerci sa programezi din nou microcontrollerul ar trebui sa mearga. Seteaza de aceasta data corect ceasul si reconecteaza cristallul. Voila!

Ce e cu configuratia asta, in definitiv?

Daca ai ajuns sa citesti pana aici, ma simt dator sa iti explic de ce microcontrollerul trebuie configurat asa.

In datasheet-ul microcontrollerului, in sectiunea System Clock apare urmatoarea diagrama:



Se observa ca semnalul de ceas poate proveni din mai multe surse: Oscilator extern RC, Ceas extern, Cristal, etc. Aceste surse sunt multiplexate la intrarea unitatii de control pentru ceas. Multiplexorul selecteaza una dintre intrari pe baza unei configuratii stocate in memoria Flash, asa numiti "Flash Fuse Bits". La pagina 25 din datasheet apare urmatorul tabel:

Device Clocking Option	CKSEL3..0
External Crystal/Ceramic Resonator	1111 - 1010
External Low-frequency Crystal	1001
External RC Oscillator	1000 - 0101
Calibrated Internal RC Oscillator	0100 - 0001
External Clock	0000

Note: 1. For all fuses "1" means unprogrammed while "0" means programmed.

In functie de modul in care sunt setati "fuse bitii" CKSEL, multiplexorul selecteaza una dintre sursele de ceas. Cand cumparami microcontrollerul acesti biti sunt 0001, adica "Calibrated Internal RC Oscillator". Ce inseamna acest lucru? Microcontrollerul include un oscilator RC ce este setat ca si sursa implicita pentru ceas. Daca ne uitam mai departe in documentatie, gasim urmatorul tabel:

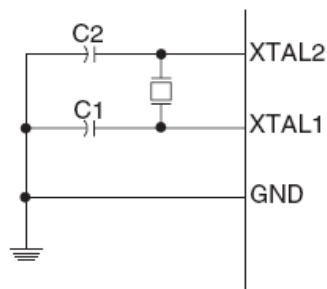
Table 9. Internal Calibrated RC Oscillator Operating Modes

CKSEL3..0	Nominal Frequency (MHz)
0001 ⁽¹⁾	1.0
0010	2.0
0011	4.0
0100	8.0

Note: 1. The device is shipped with this option selected.

Se observa din nou ca 0001 este configuratia implicita si ca frecventa nominala a semnalului in acest caz este de 1 MHz. De asemenea, se mai poate observa faptul ca frecventa maxima a oscilatorului intern este 8 MHz. Ce e in neregula cu acest ceas? Pe langa frecventa scazuta, in cazul oscilatoarelor RC frecventa se modifica foarte mult cu tensiunea de alimentare si cu factorii de mediu. Ca si consecinta, spre exemplu in cazul in care alimentati circuitul de la baterii, cand bateriile incep sa se consume frecventa ceasului se modifica. Acest lucru este de nedorit deoarece genereaza probleme de sincronizare.

Pe placuta noastra exista un cristal cu frecventa de 16 MHz. Revenind la tabelul anterior, se observa ca pentru cristal, bitii trebuie sa fie intre 1111 si 1010. Cautand in documentatie, dam de schema de conectare a oscilatorului extern, pe care probabil o recunoasteti:



De asemenea, mai gasim si tabelul pentru configuratia exacta pentru fuse-bitii CKSEL:

Table 4. Crystal Oscillator Operating Modes

CKOPT	CKSEL3..1	Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101 ⁽¹⁾	0.4 - 0.9	-
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 ≤	12 - 22

Note: 1. This option should not be used with crystals, only with ceramic resonators.

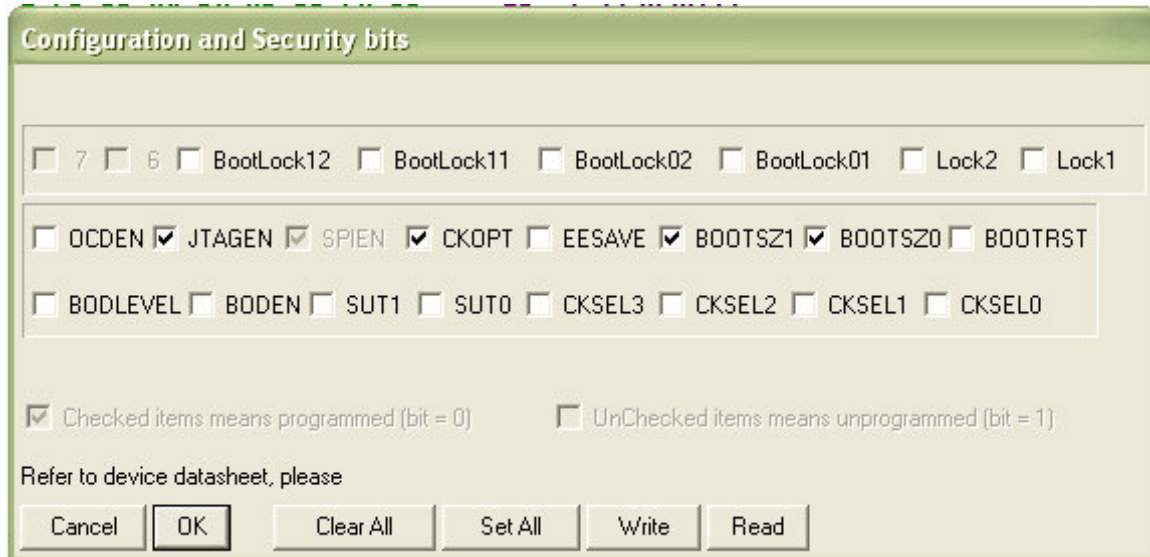
Se pare, insa ca frecventa maxima pentru oscilator ar fi 8 MHz !? Nu tocmai. Microcontroller-ul permite doua moduri de lucru pentru amplificatorul oscilatorului:

un mod in care oscilatorul functioneaza la frecventele din tabel si un mod full rail-to-rail in care frecventele se dubleaza. Pentru modul full rail-to-rail, fuse-bit-ul CKOPT trebuie sa fie programat.

In concluzie:

1. Pentru oscilator extern CKSEL trebuie sa fie intre 1111 si 1010
2. Mai exact, pentru 16 MHz CKSEL 3..1 trebuie sa fie 111
3. Pentru modul full rail-to-rail CKOPT trebuie sa fie 0

Hai sa ne mai aruncam o privire la fereastra de configurare PonyProg :



Se observa ca toti bitii CKSEL sunt nebitati, adica unprogrammed (1), iar bitul CKOPT este programat (0).

Succes!