1.Programul CCS C Compiler. Instalare.Creare proiect.

Descarcati ultima versiune a programului de la adresa : <u>http://www.ccsinfo.com/compdemo.php</u> Instalati programul si lansati-l in executie (vom utiliza meniurile si cuvintele cheie in limba engleza). NOTA: Oricare versiune mai veche functioneaza asemanator.

Vom utiliza modulul PIC-AP20 cu schema simplificata desenata mai jos.



Aici am desenat doar intrerupatorul SA4 (conectat la A4), LED-ul LB7 (conectat la B7) si buzzer-ul BUZ-B6 (conectat la B6). Pe masura ce avansam vom desena si alte elemente.

Observati organizarea pe porturi a microcontroler-ului (A,B,C,D,E). Acestea au in general cite 8 terminale notate de la 0 la 7 (doar E are terminale de la 0 la 3).

Ca orice sistem ce contine un procesor si acesta are nevoie de un oscilator. Avantajul acestui microcontroler este ca are un oscilator intern care poate fi programat sa functioneze la anumite frecvente. De obicei utilizam frecventa cea mai mare (8MHz) daca nu avem restrictii la consumul de curent.

Deoarece utilizam oscilatorul intern vom avea la dispoziite terminalele RA6 si RA7 pentru a le folosi ca intrari/iesiri. Daca dorim putem utiliza un cristal de cuart conectat la aceste doua terminale. Conform schemei mai sint necesare doua condensatoare de cca 22pF.

Pe baza acestei scheme vom incepe sa configuram parametri in CCS C Compiler (CCSC).

NOTA: evident pentru o intelegere superioara, este indicat sa studiati si fisa tehnica a PIC16F887.

Deschideti CCSC si selectati File/New/Project Wizard , ca in imaginea de mai jos:



Se va deschide o feresatra unde trebuie sa salvati proiectul.



Dupa salvare se va deschide o noua fereastra unde trebuie sa faceti citeva setari.

In partea din stinga se observa parametri care trebuie *setati*. Puteti naviga pentru a vedea despre ce este vorba . Nu vom folosi decit citiva parametri pentru inceput.

In partea dreapta vom alege tipul microcontroler-ului, oscilatorul intern si frecventa la care va functiona oscilatorul. Trebuie sa retineti ca frecventa reala de tact este de 4 ori mai mica decit cea specificata acolo.

Project Wizard - D:\LUC	Roboprog\CURS_C\Programe curs CCS\minicurs_1.ccspjt	<										
File Help												
Device Selection	Options Code General											
	Device Family: Debug Code Device:											
Peripherals	Clock											
Analog	Clock Speed: 8,000,000 Hz 2 MIPS USB Low Speed											
Communications	CPU Clock Speed: 0 Hz V Clock Out											
SPI	WDT Boost WDTCON											
Drivers	Enabled WDI Reset WDI CON											
Header Files	following to restart 36 ms Div by 8 WDT during calls to: 72 ms Div by 4											
High/Low Voltage	getc() and fgetc() 144 ms Div by 2 288 ms Times 1											
Interrupts	[2c_read()] 576 ms Times 2 Delay Functions 1152 ms Times 4											
I/O Pins	Reset: 1.1 ms 2304 ms Times 8 Times 16 Times 16											
Timers 0-2	Times 32											
Timers 3-up	Times 128											
LCD (Internal)												
	Chip: PIC16F887 Frequency: 8,000,000 Create Project Cancel											
1.4.		_										

Derulati lista din stinga pina la ultimul parametru.

Desi acum nu ne este necesar este o idee buna sa invalidati terminalul de reset. In acest fel dispunem de inca o intrare (acest terminal poate fi doar intrare nu si iesire). Bifati optiunea Include Fuses.

Observati ca puteti alege sa blocati citirea neautorizata a programului si a memoriei EEPROM.

File Help											
Timers 0-2	^	Options Code									
Timers 3-up		Fuses									
LCD (Internal)											
LCD (External)		Power Up Timer Master Clear pin enabled									
CapacitiveTouch		Code protected from reads Data EEPROM Code Protected									
RTCC		No brownout reset \checkmark									
CCP/Vref		✓ Internal External Switch Over mode enabled ✓ Fail-safe clock monitor enabled									
Comparator		Low Voltage Programming on B3(PIC16) or B5(PIC18)									
USB		Brownout reset at 4.0V ~ Program memory not write protected ~									

Pentru inceput setarile sint suficiente.

Este interesant ca la fiecaree parametru puteti vedea ce cod introduce in program. Selectati **Code** in partea dreapta sus pentru a vedea codul scris de compilator.

Dupa aceste setari apasati butonul Create Project (drepata jos).

4	CCS C Co	m	piler						
F	ile	Edi	it Search	Options	Compile	View	Tools	Debug	
De	evice Edito	or	D <u>e</u> vice Selecto	or <u>File</u> Comp	Dare <u>B</u> ase C	onverter	<u>S</u> erial Mo	nitor C Tool	In acest moment se va afisa o pagina cu citeva "instructiuni". Cel mai important este ca ni se indica unde putem scrie programul nostru: //TODO: User Code
r	🕤 We	elco	ome 🔤 min	icurs_1.c					,,,
Files	1		<pre>#include <</pre>	minicurs_	1.h>				
: The Pro	2 3 4 5	F	void main()					
jects	6	ļţ	while(T	RUE)					
s 🚺 Identifiers	7 8 9 10 11 12 13	-	{ //TO .}	DO: User	Code				

Apasati tasta F9 a PC-ului pentru a compila programul (sau selectati meniul Compile si apasati Compile).

In stinga, pe verticala este scris : "Files". Daca apasati acel buton veti putea selecta fisierul de tip .h care contine setarile initiale realizate in *wizard* (in general nu este foarte interesant). Se pot face editari aici, de exemplu se poate schimba frecventa la 4MHz.



Din acest moment sintem pregatiti sa scriem primul program.