**Instructiuni programare cu ez-Circuit**

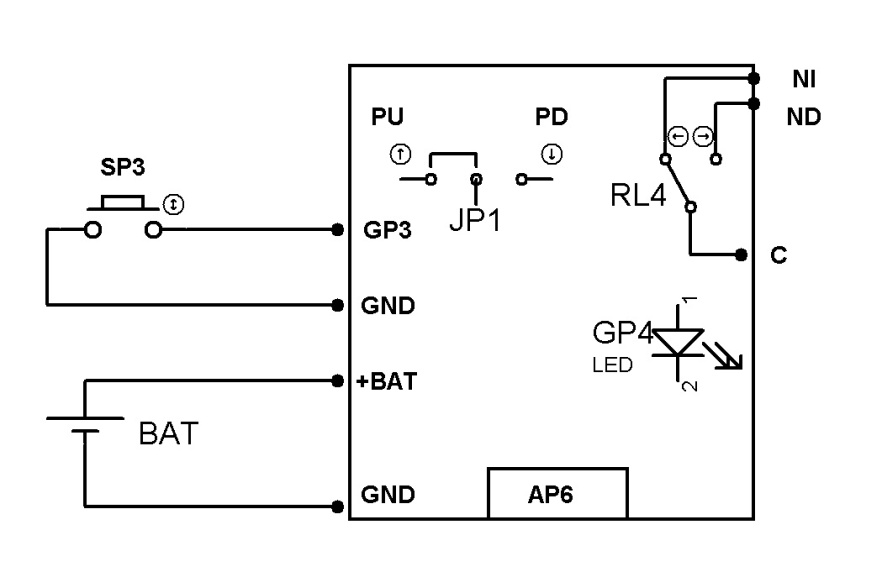
**PROIECTUL**

Ne propunem sa utilizam un automat programabil (model PIC-AP6) pentru a aprinde un LED care va functiona o durata de timp prestabilita - sa zicem 5 secunde. In loc de LED poate fi orice alt element , de exemplu: un releu, un bec, o sonerie etc. conectate la orice iesire disponibila a modulului. Acum este suficient sa observam doar un LED conecat la iesirea GP4 (existent pe modul).

Pornirea se va face de la un buton cu revenire SP3, conectat la intrarea GP3 (exista pe modul).

*Pornind de la acest proiect se poate adauga orice alt element dorit: inca un buton, inca un consumator, temporizari diferite etc.*

Schema de principiu va fi urmatoarea:



Vom utiliza butonul SP3 de pe modul si vom observa LED-ul GP4.

Jumper-ul JP1 corespunzator GP3, va fi conecat in pozitia PU (pull-up) pentru ca butonul este conectat la masa. In acest fel se asigura nivel logic 1 cind butonul nu este apasat si nivel logic 0 cind este apasat.

*Un buton extern poate fi conectat si la tensiunea de alimentare.*

*In acest caz JP1 va fi conectat la PD.*

Desigur, ne intereseaza de aici urmatoarele:

-butonul este conectat la intrarea GP3

-LED-ul este conectat la iesirea GP4

-modulul AP este alimentat la +BAT si GND.

De fapt acestea sint cele mai importante informatii pentru a concepe programul. Aceste informatii se utilizeaza cind concepem un proiect cu un modul AP (ce intrari si ce iesiri utilizam).

Se poate utiliza orice buton conectat extern la una din intrarile GP0,1,2,3.

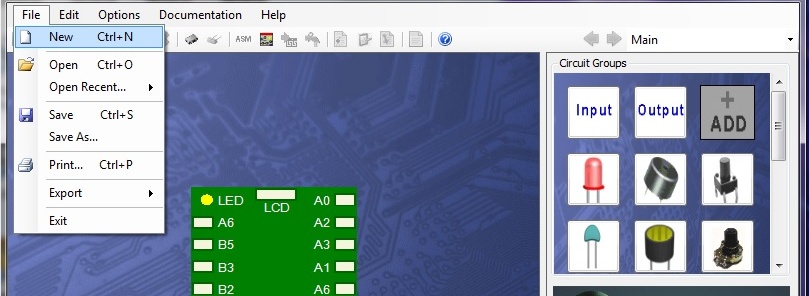
Se poate utiliza orice iesire GP4, GP5.

Pentru detalii vedeti specificatia tehnica a modulului.

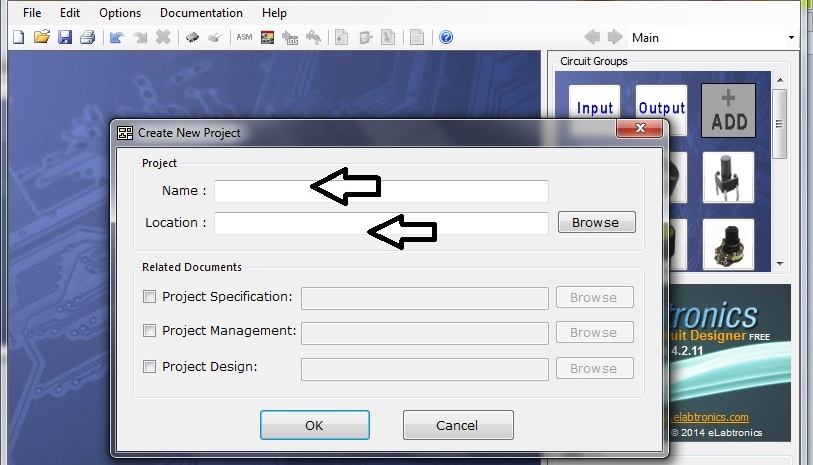
**1.Instalarea programelor**

Descarcati si instalati programul ezCircuit FREE. Acesta va fi utilizat pentru a desena schema utilizata in automatizare. Instalati si programul PICKIT2 de la Microchip. Gasiti instructiuni de descarcare si instalare in sectiunea *Download.*

Lansati porogramul ezCircuit FREE si selectati File/New pentru a incepe un nou proiect.



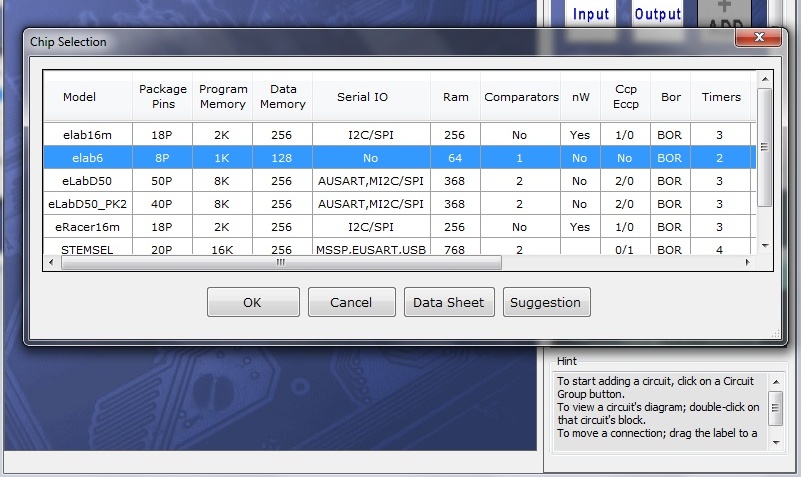
In fereastra nou aparuta completati numele proiectului (Name) si folderul (Location) unde doriti sa salvati proiectul. Puteti utiliza Browse pentru a alege calea dorita. Apasati OK.

Proiectul nostru va avea numele timer6.

Selectati Options/Chip Selection:

Va apare fereastra urmatoare in care va trebui sa slectati tipul automatului utilizat.

Selectati elab6 pentru AP6. Apasati OK.



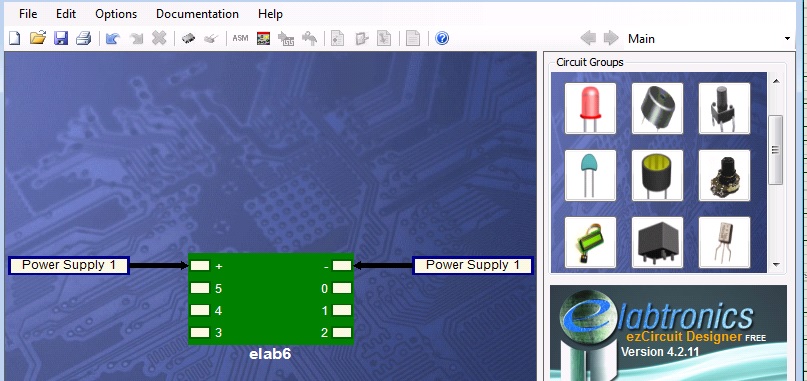
Observati imaginea automatului programabil. Acesta are disponibile urmatoarele resurse:

-terminalele de alimentare + si – care corespund in realitate la +BAT si, respectiv, GND

-terminale programabile intrari/iesiri GP0,GP1,GP2,GP3

-iesiri GP4,GP5

-iesiri cu LED-uri la GP4 si GP5



Daca suprapuneti cursorul peste terminale veti observa o scurta descriere pentru fiecare dintre acestea. Unele terminale pot avea mai multe functii programabile. In principiu, functiile de baza sint:

-intrari digitale (la ele se conecteaza butoane sau senzori de tip contact)

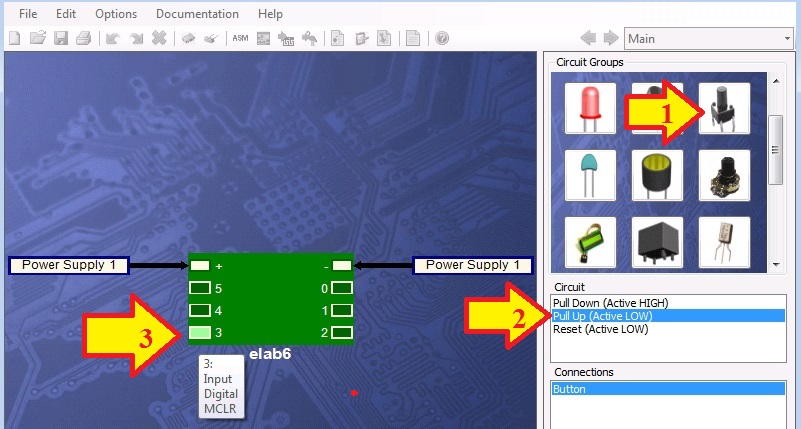
-intrari analogice (la ele se conecteaza semnale analogice provenite de la potentiometre, senzori de lumina, de presiune, de temperatura etc)

-iesiri digitale (la ele se conecteza LED-uri, motoare, relee etc)

Pentru o descriere completa a posibilitatilor de conectare trebuie consultata documentatia automatului programabil utilizat.

Acum putem incepe “desenarea” schemei.

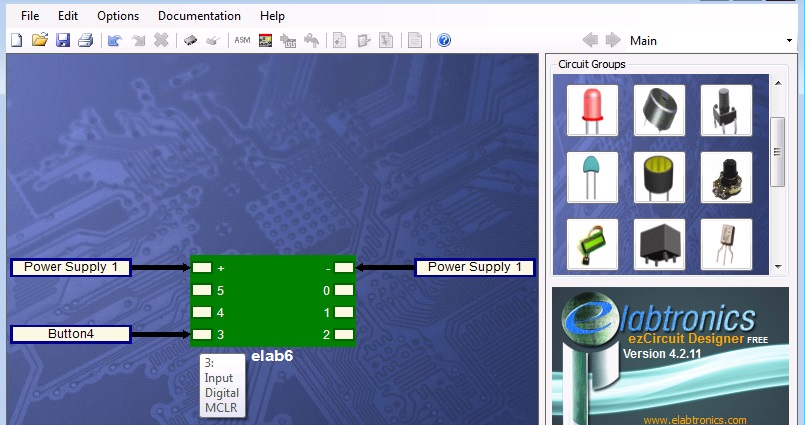
Selectati pe panoul cu componente din dreapta, imaginea butonului (1) - clic pe butonul stinga, pe imagine.



Selectati din lista derulanta linia Pull Up (2) -*Active Low*, deoarece butonul nostru este conectat la masa si va activa cind este apasat, adica intrarea va fi in starea Low =0V.

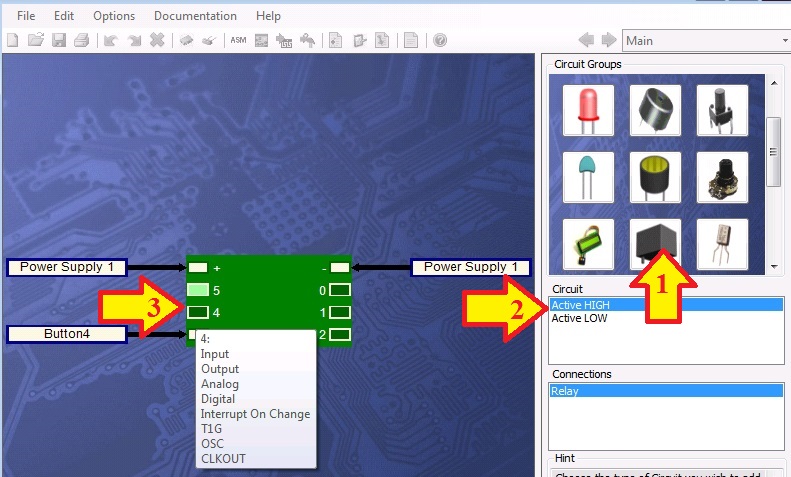
Daca suprapuneti cursorul peste terminalul 3 (GP3) va apare descrierea functiilor posibile . Noi vom folosi terminalul pe post de intrare digitala – fiindca semnalul va fi doar +5V sau 0V datorita butonului si conexiunii de Pull-up cu rezistorul de pe modul . Asignarea in mod digital se face automat deoarece se conecteaza un element cu doua stari – butonul.

Dati clic pe dreptunghiul din dreptul lui 3. Imaginea se va schimba in urmatoarea :

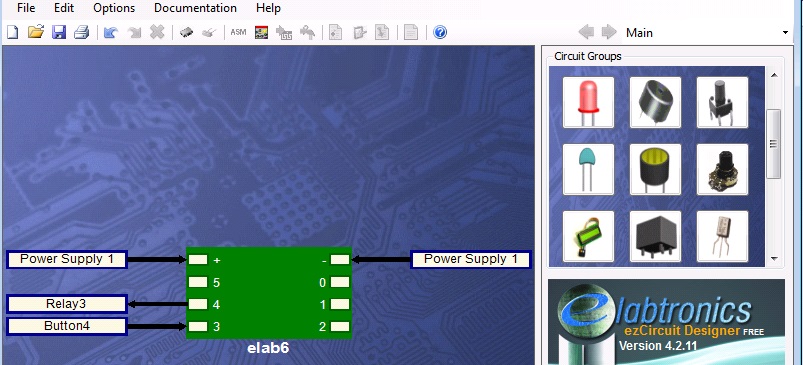


Se observa aparitia conexiunii unui buton denumit automat *Button4*. Denumirea se poate schimba dar acum nu este cazul (pentru modificare se da clic dreapta pe *Button4*.

Acum vom conecta o iesire. Vom utiliza un releu. Selectam imaginea releului (1) si apoi din lista selectam *Active HIGH* (2) deoarece dorim ca atunci cind iesirea este *HIGH* (adica 1 logic) sa se anclanseze releul. Asemantor, putem selecta un releu , un buzzer sau un tranzistor.

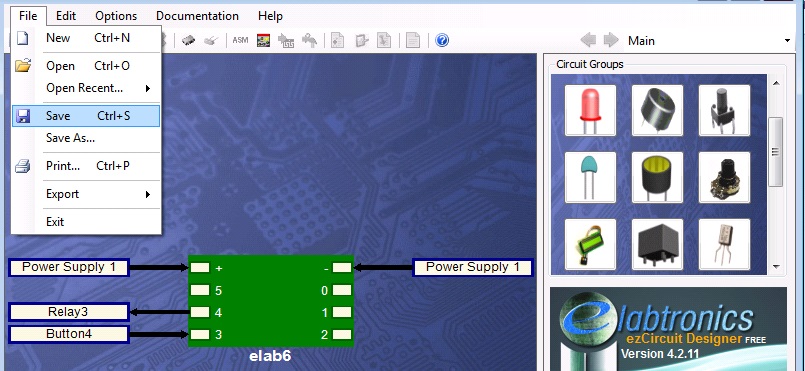


Dam clic pe pinul 4 si vom avea conectat releul . Acesta se denumeste automat *Relay3*. Denumirea se poate schimba, dar acum nu ajuta.



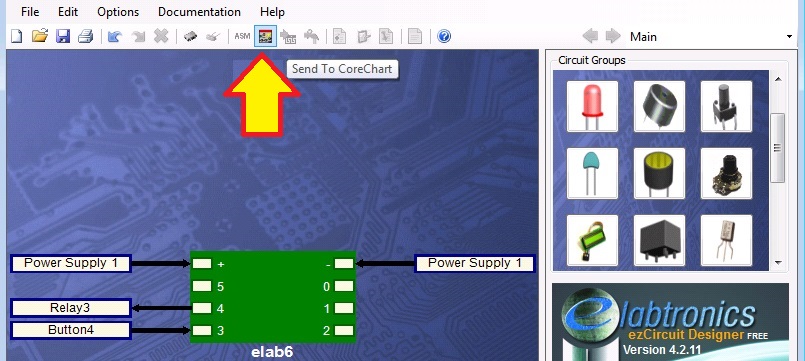
Desenarea schemei a luat sfirsit. Similar se pot adauga si alte elemente utilizind panoul din dreapta.

Elementele gri nu se pot selecta in versiunea free dar nici nu sint extraordinar de utile.

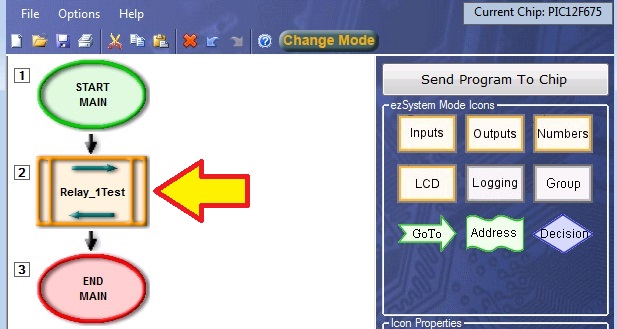


Acum va trebui sa salvam schema . Selectati File/Save.

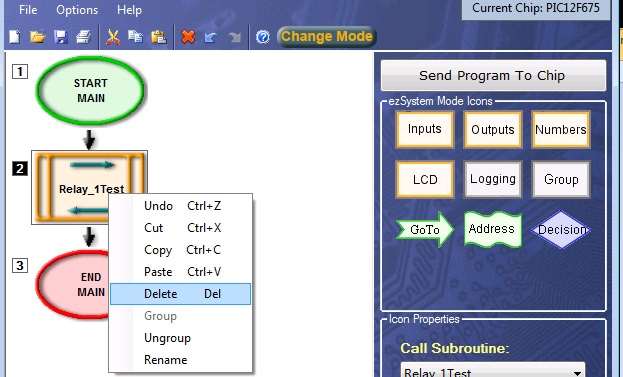
Ultima operatiune aici: trimitem schema la programul CoreChart. Acesta va fi utilizat pentru programare. Selectati imaginea marcata cu sageata si se va deschide CoreChart .



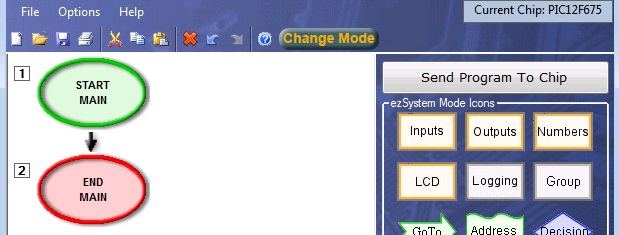
Imaginea initiala va fi urmatoarea. Aici se observa un program de test deja scris dar care nu este util.



Vom sterge acest program. Clic dreapta pe blocul LED\_1Test. Stergeti selectind Delete.



Vor ramine doar inceputul (START MAIN) si sfirsitul programului (END MAIN).



Citeva cuvinte despre acest program care va fi implementat si despre programare.

Dupa alimentarea montajului electronic cu modulul AP sistemul va functiona astfel :

-va astepta apasarea butonului (care va fi un buton existent pe placa sau conectat in exterior)

-dupa apasarea butonului va activa iesirea la care este conectat releul si va aprinde LED-ul (GP4)

-se va porni o temporizare (timer) cu durata stabilita (5 secunde)

-dupa expirarea timpului programat (a timer-ului) se va da comanda de dezactivare a iesirii

-programul va reincepe executia de la inceput (START)

Este clar ca in program vom folosi elemente de programare (imagini) care :

-citesc intrarea unde este conectat butonul

-pornesc un timer

-activeaza / dezactiveaza o iesire

-realizeaza intoarcerea (salt) la inceputul programului

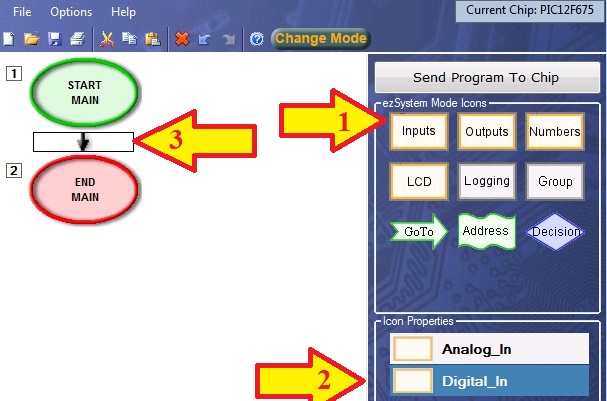
**Implementarea programului**

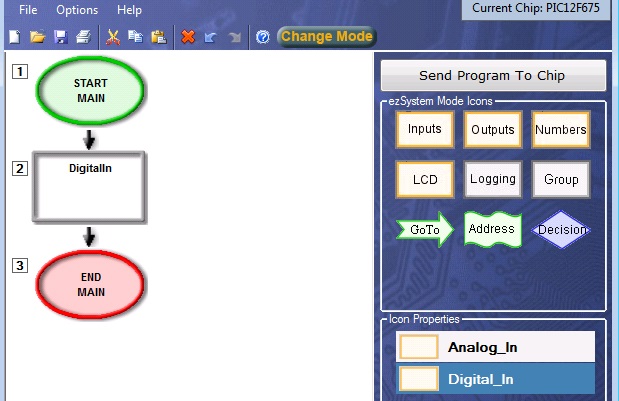
Vom incepe sa “scriem”programul.

-Selectati (1) pe panoul din dreapta butonul *Inputs* (intrari) pentru ca dorim sa citim butonul care este conectat la o intrare (GP3)

-Selectati (2) *Digital\_In* fiindca intrarea este digitala (are doar doua valori 1 sau 0 logic adica 5V sau 0V)

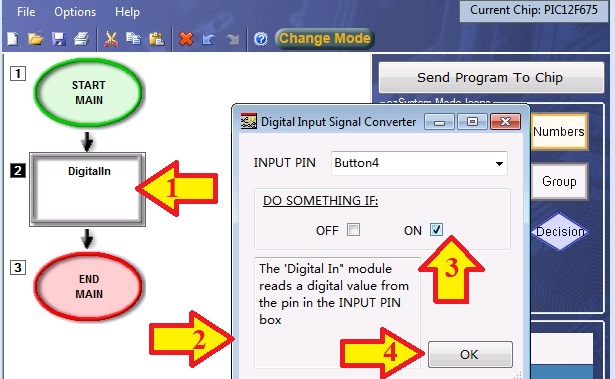
-Selectati (3) sageata neagra indicata mai jos .

 Va apare un dreptunghi notat 2 ce contine textul *DigitalIn:*

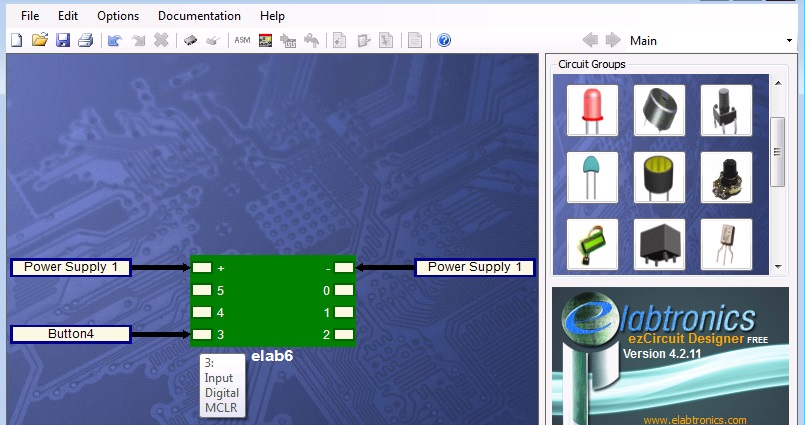
**

Dati dublu clic pe dreptunghiul 2 (1). Se va deschide o fereastra (2) unde se poate parametriza citirea intrarii la care este conectat butonul. Intr-adevar, avem nevoie sa programam in asa fel incit citirea intrarii sa realizeze ceva (*DO SOMETHING IF*).

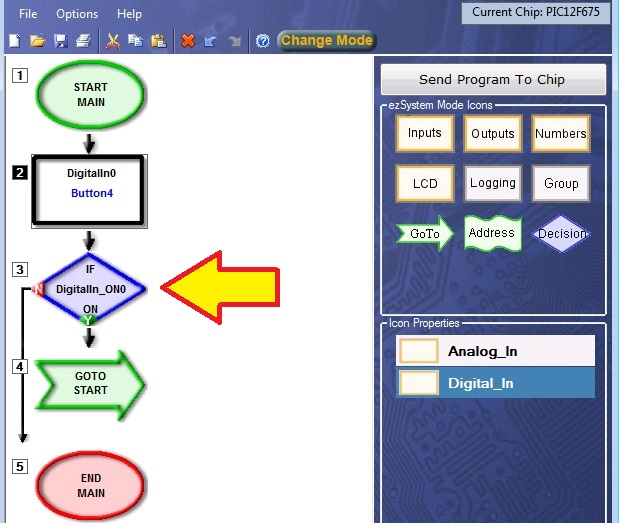
Vom selecta ON (3) si vom apasa OK (4).



Observati legatura cu programul de desenare a schemei (*ezCircuit Designer)*: butonul este conectat la pinul GP3 al modulului (si in realitate la pinul GP3 al circuitului integrat PIC12F675). Acesta a aparut si in descrierea din fereastra de parametrizare (cu numele Button4).



Acum in programul nostru (grafic) a aparut un nou element: blocul numerotat 3.

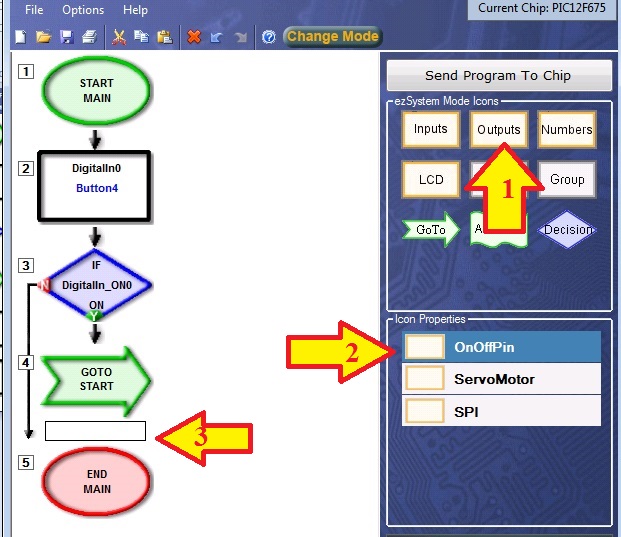
Acesta se numeste element de decizie.

Pe baza starii intrarii GP3 ( a butonului SP3) se poate lua o decizie: fie se revine la START (inceput program) fie se merge la END MAIN (sfirsit program).

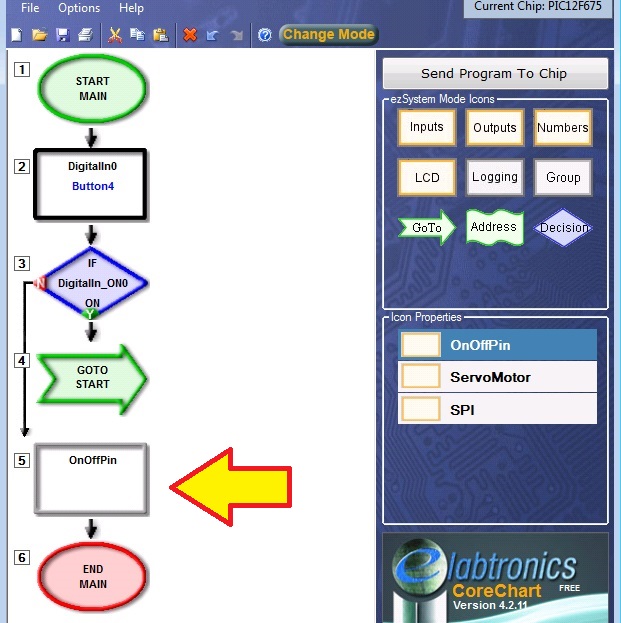
Programul nu este, insa, gata.

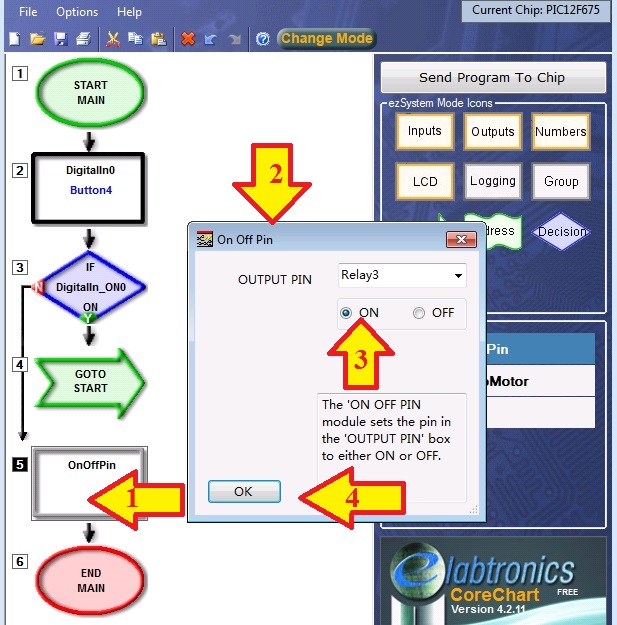
In continuare vom introduce si iesirea la care se conecteaza releul.

Selectati *Outputs* apoi *OnOffPin* si dati clic unde indica sageata notata 3.



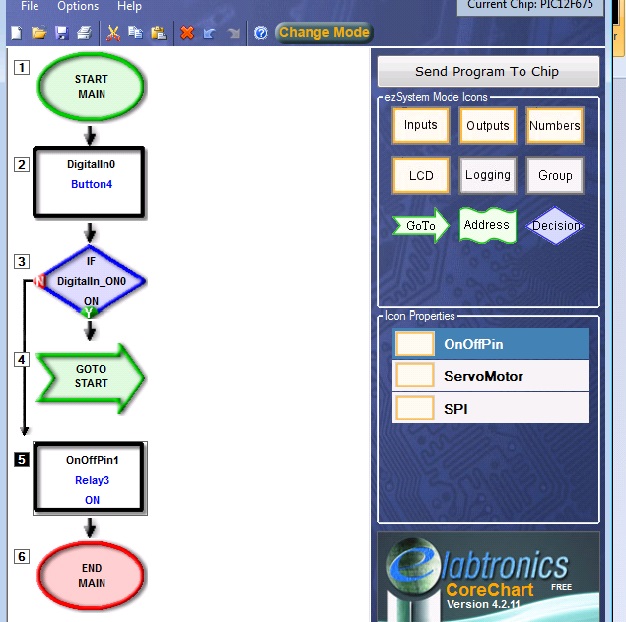
Va apare un nou element: iesirea notata *5.*



Acum va trebui sa parametrizam iesirea adica sa ii alocam o valoare (poate fi actionata adica valoare 1 logic sau neactionata adica valoare logica 0).

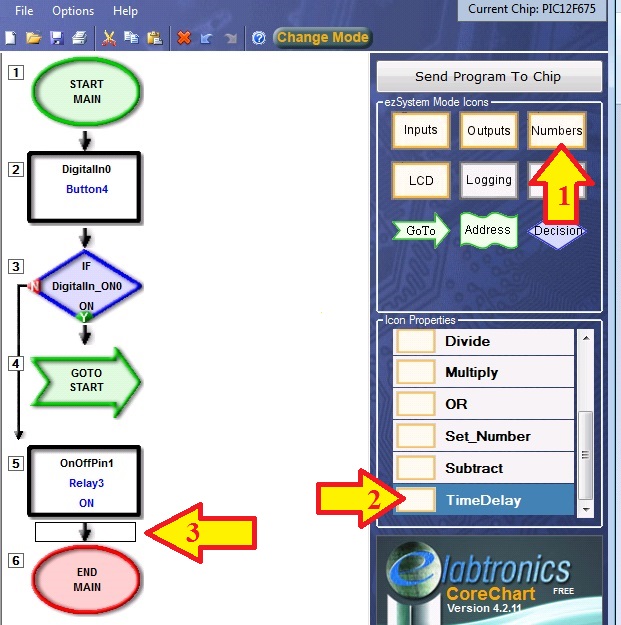
Daca se da dublu clic pe blocul de iesire (1) se deschide o noua fereastra (2).

Selectati ON (3) fiindca acum dorim sa activam iesirea si apoi OK (4).



Iesirea a fost alocata – blocul 5.

Acum vom introduce temporizarea. Selectati *Numbers(1) ,* derulati lista de jos si selectati *TimeDelay(2).*

**

Aceasta este denumirea utilizata pentru circuitul de temporizare.

Selectati apoi zona indicata de sageata 3.

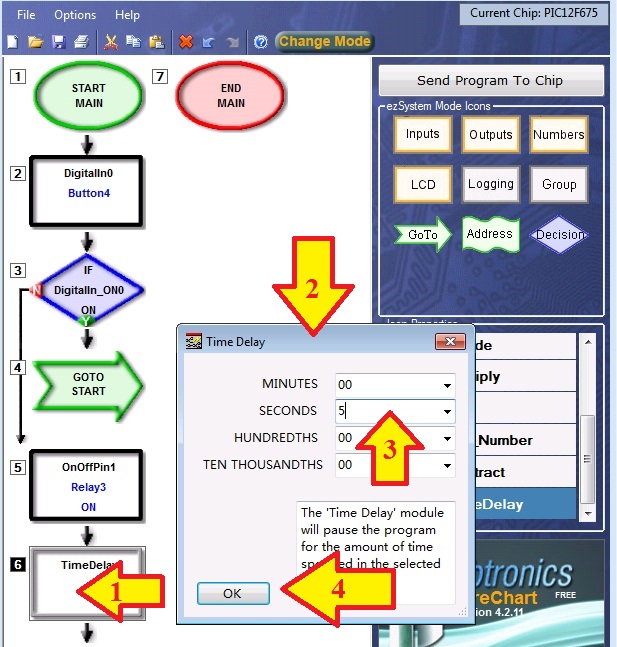
Elementul de temporizare va fi introdus.

Va trebui sa parametrizam acest

element.

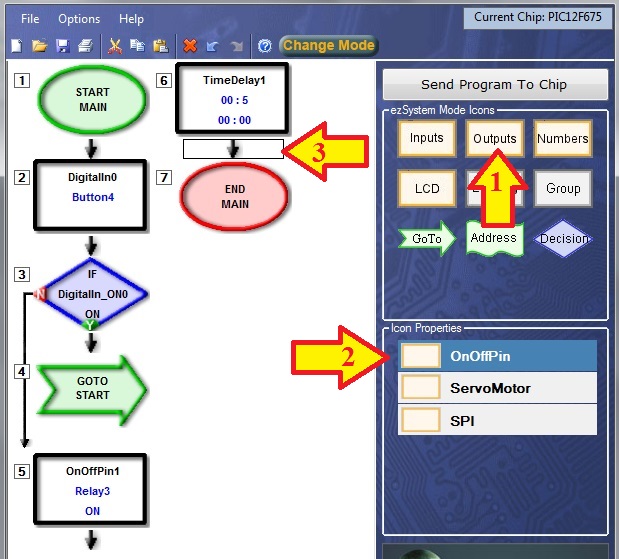
Dati dublu clic pe TimeDelay (1) si se va deschide o noua fereastra (2).

Scrieti 5 la SECONDS (3) si apasati OK (4).



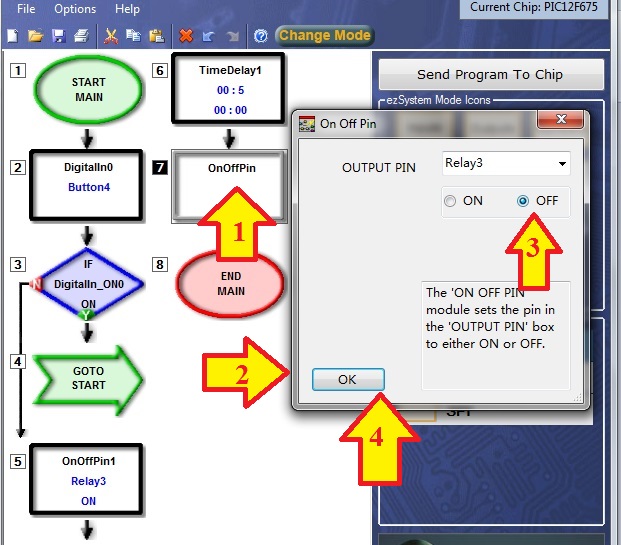
In continuare introducem o noua stare a iesirii deoarece dupa expirarea timpului iesirea trebuie sa se opreasca (sa treaca in 0 logic, OFF).

Secventa este cea deja utilizata: selectati *Outputs* (1), *OnOffPin*(2) si dati clic sub casuta *TimeDelay* (3).



Acum parametrizam iesirea - adica ii alocam o valoare. In acest caz ii alocam valoarea 0 fiindca timpul a expirat.

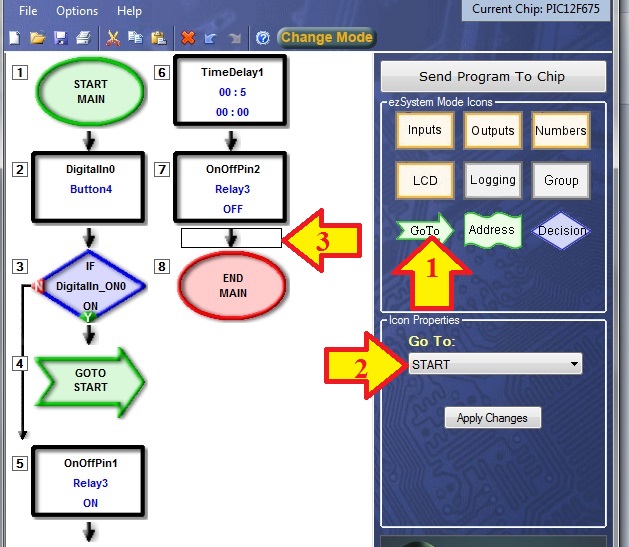
Procedam ca mai inainte: dublu clic pe iesire (1), se deschide o noua fereastra (2), se selecteaza OFF (3) apoi se apasa OK (4).



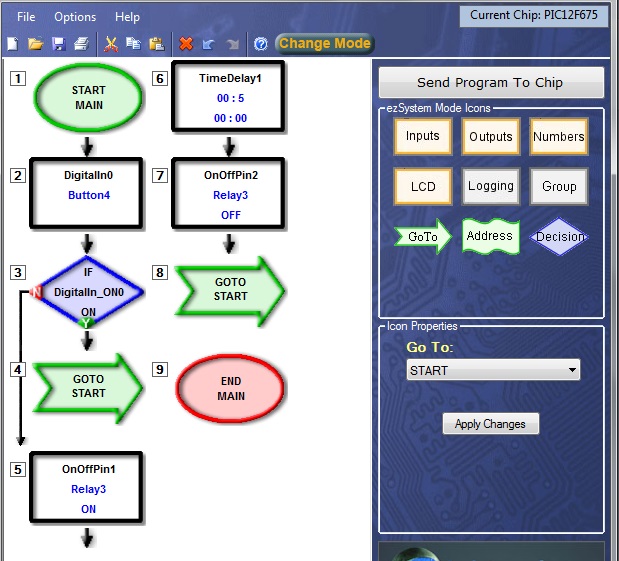
In principiu programul este gata dar va trebui sa revina la inceput (START) pentru a reincepe un nou ciclu la apasarea butonului.

Vom introduce o instructiune de salt (GoTo) care va dirija executia spre inceputul programului.

Selectati GoTo (1) apoi START (2) si dati clic unde indica sageata 3.



Programul final va arata ca in imaginea urmatoare:



Legenda:

1=START program

2=citire intrare

3=decizie in functie de starea intrarii GP3

4= -daca nu s-a apasat butonul intrarea este 1 logic (ON) si se merge inapoi la START

-daca s-a apasat butonul intrarea este 0 logic (OFF) si se merge la (5)

5=iesirea GP4 se activeaza (ON)

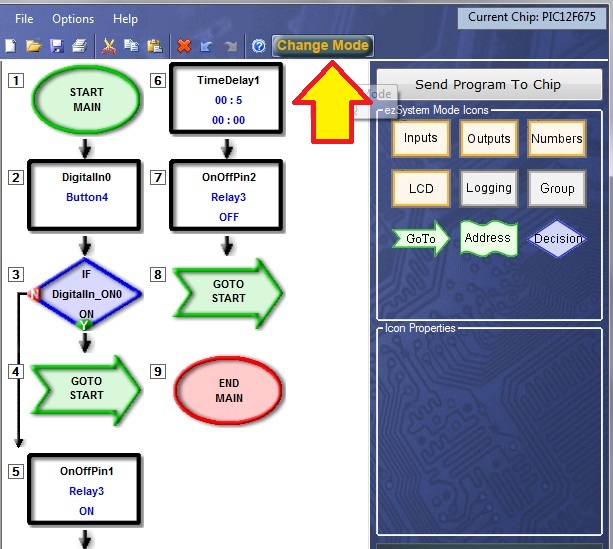
6=se porneste TimeDelay si se asteapta 5 secunde (iesirea este activa in acest timp)

7=timpul a expirat si iesirea B2 se dezactiveaza (OFF)

8=se sare inapoi la START pentru a se relua programul

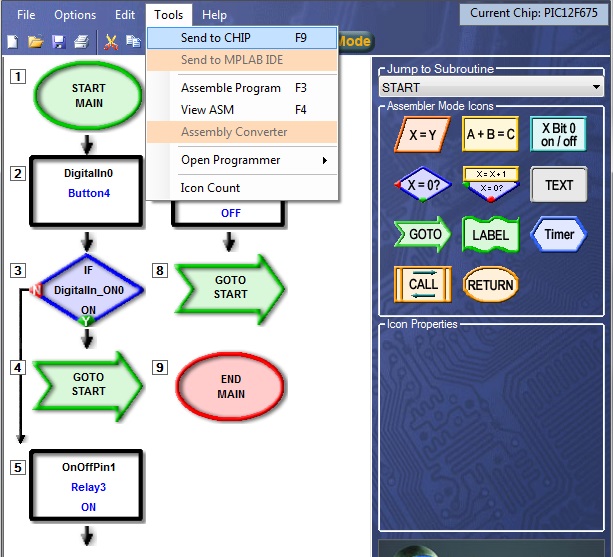
9=sfirsit program END MAIN (fara semnificatie in acest caz); daca lipsea elementul 8 atunci programul se oprea trecind la 9 (acesta ar fi fost pasul 8, de fapt) repornirea se putea face dupa oprirea si repornirea alimentarii modulului (reset).

Programul este complet dar este intr-un format pe care microcontrolerul nu il va intelege. De aceea urmatoarea etapa consta in transformarea programului acesta cu elemente grafice intr-un fisier de tip hex (contine codurile instructiunilor).

Selectati Change Mode.

Selectati *Send to CHIP* si se va crea fisierul timer6.hex in folderul de lucru.

Programul va fi asamblat si se va crea un fisier tutor6 .hex in folderul de lucru.



Pentru programara microcontroler-ului procedati astfel:

-opriti alimentarea modulului

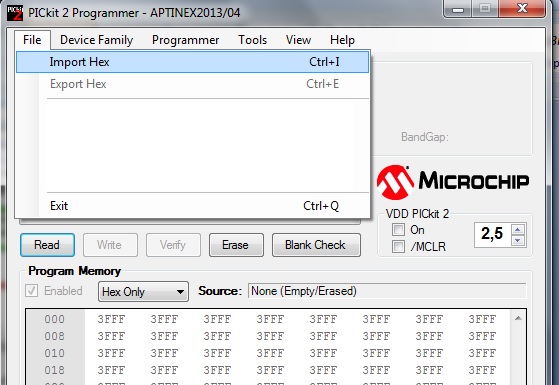
-treceti jumperul RUN/PROG pe pozitia PROG

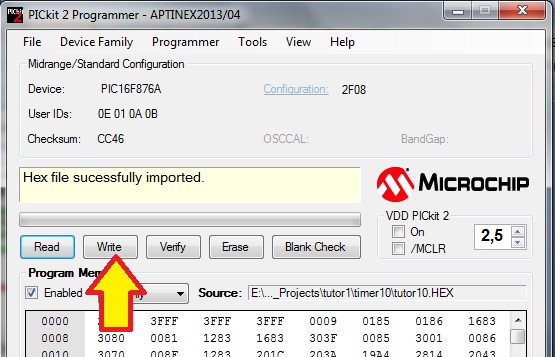
-conectati programatorul la PC

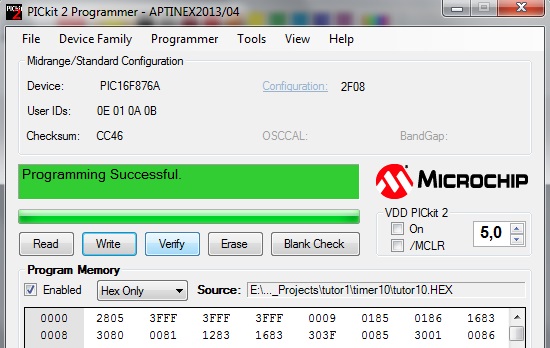
-lansati programul PICKIT2

-conectati programatorul la portul ICSP al modulului (atentie la pini)

-incarcati fisierul **tutor6.hex** (se gaseste in folderul proiectului)



-selectati WRITE



-dupa scriera corecta

-decuplati programatorul de la modul

-puneti jumperul RUN/PROG pe RUN

-alimentati modulul

-testati functionarea

Alternativ puteti scoate circuitul integrat din soclu (dupa oprirea alimentarii) si il puteti introduce in adaptorul livrat, in soclul destinat pentru IC cu 8 pini.