Universitatea Politehnica Timișoara

Facultatea de Automatică și Calculatoare

Calculatoare și Tehnologia Informației

**Implementarea unui calculator pentru operaţii aritmetice de bază pe placa**

**Dragon12**

Ghiorma Cristian 3.2

Bratu Cristian 1.2

**Enunt:**

• Se va folosi tastatura integrată pe placa Dragon12 pentru introducerea

operanzilor şi a operatorilor;

• Se vor implementa cel puţin cele 4 operaţii aritmetice de bază (adunare,

scădere, înmulţire, împărţire);

• Operanzii, operatorii şi rezultatul vor fi afişate pe LCD-ul plăcii;

• Se va asigura afişarea rezultatelor care sunt numere întregi, precum şi a

celor reale;

• Se va afisa un mesaj de eroare în cazul împărţirii la 0.

Placa Dragon12-Plus2 folosită:

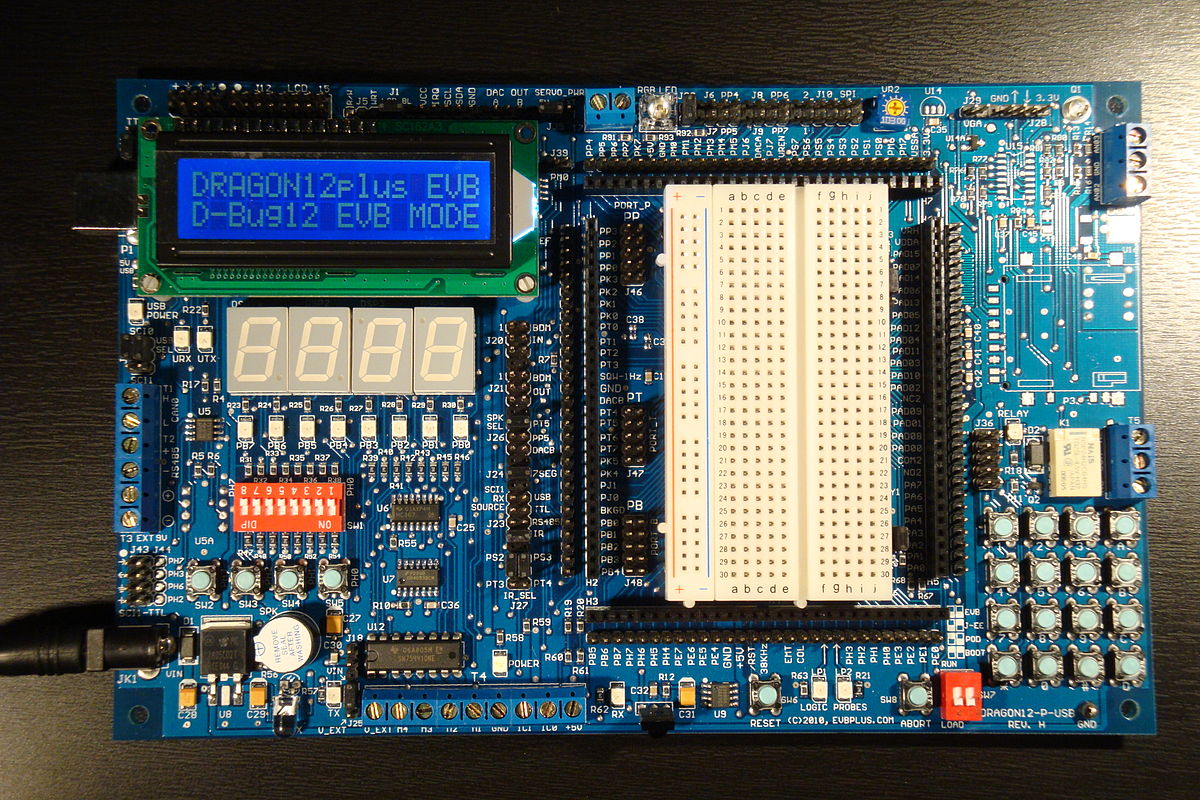
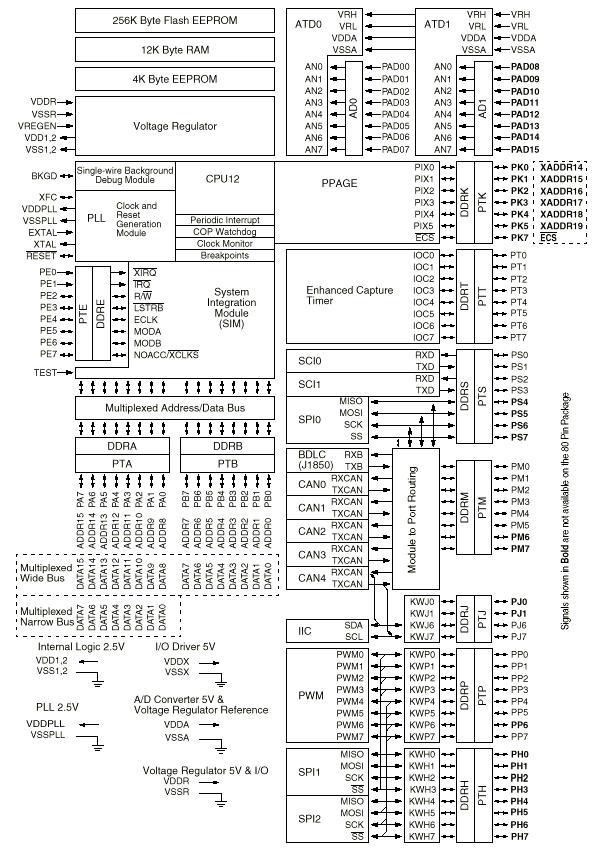
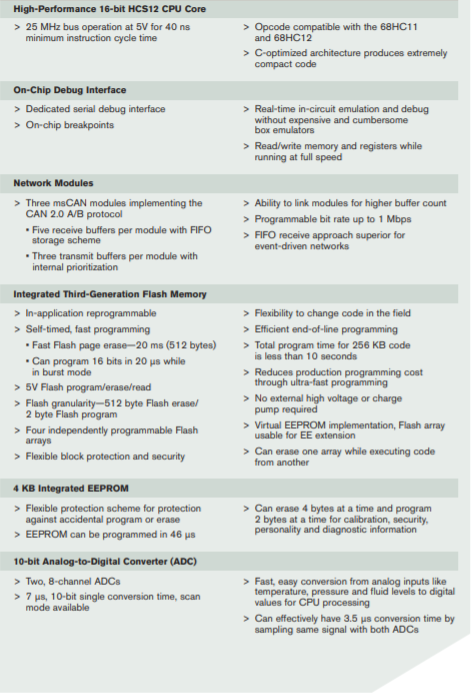


Diagrama plăcii:

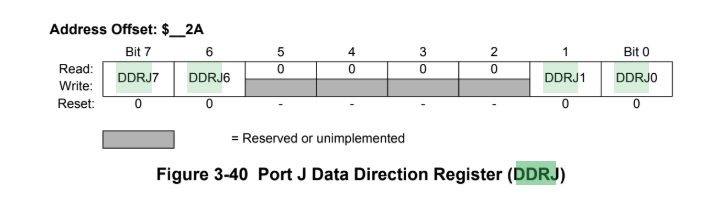


MC9S12DT256:



Registers

1)DDRJ=0x02

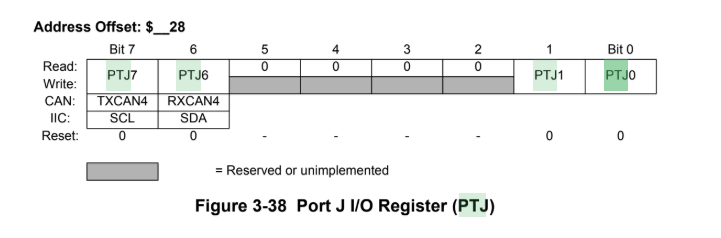


DDRJ[7:6][1:0] — Data Direction Port J

1 = Associated pin is configured as output.

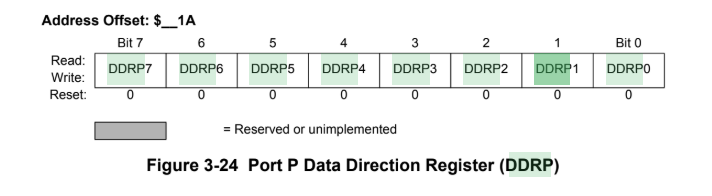
0 = Associated pin is configured as input.

2)PTJ&=~0x02;



If the data direction bits of the associated I/O pins are set to 1, a read returns the value of the port register,otherwise the value at the pins is read.

3)DDRP=0x0F



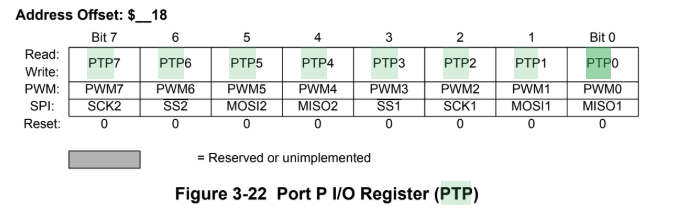
This register configures each port P pin as either input or output.

DDRP[7:0] — Data Direction Port P

1 = Associated pin is configured as output.

0 = Associated pin is configured as input.

4)PTP =0x0F



If the data direction bits of the associated I/O pins are set to 1, a read returns the value of the port register, otherwise the value at the pins is read.

5)DDRA=0x0F

DDRA[0:3] make columns as input

DDRA[4:7] make rows as output

6)DDRB=0xFF

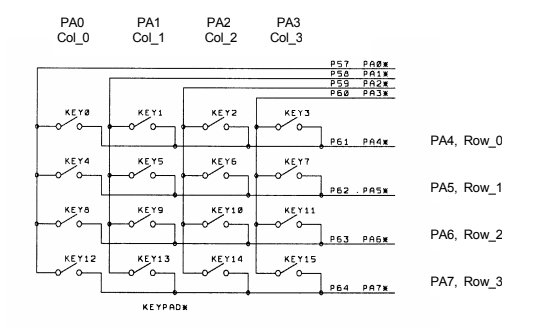
DDRB[0:7] make port B output

Keypad

Port A is an 8-bit bi-directional port. Its primary usage is for a 4X4 keypad. If the port is not

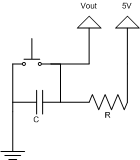
used for the keypad, it can be used as a general-purpose I/O.

The schematic for the keypad connections is shown below:



**Keypad Debouncing**

Whenever a key is pressed, there are small mechanical vibrations that cause noise on the input, which can cause the microcontroller to detect several keypresses instead of just one. You can solve this problem using software or hardware. Here is an example of a hardware keypad debouncing solution:

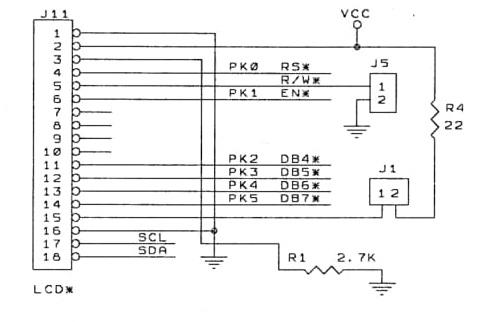


The above circuit is a simple low pass filter. The capacitor takes time to charge and discharge which smooths the voltage and filters out high frequency noise. The amount of time it takes to charge and discharge is dependent on the values of R and C used in the circuit.

The software solution is to introduce a small delay after a key is pressed and check to make sure the key is still pressed after the delay.

LCD Display

* The Dragon12 board has a 16 character x 2 line display
* Each character is a 5 x 7 bit matrix •
* A controller chip (Hitachi HD44780) converts ASCII characters to 5x7 bit image
* The controller chip is connected to Port K of the MC9S12:
  + Bit 0 of Port K (PK0) selects command (1) or data (0)
  + Bit 1 of Port K (PK0) enables the data transfer
  + Bits 5 through 2 Port K (PK5-2) contain the data
  + Bit 7 of Port K (PK7) can be used to select read or write.
* The LCD on the Dragon12 board is set up for write only. So we need to cut a trace to be able to read from the LCD.



Port K is an 8-bit bi-directional port. It’s used for the LCD display module. If the port is not used for the LCD display, it can be used as a general-purpose I/O port.

The LCD has the following pins:

* Pin 1 GND
* Pin 2 VCC (5V)
* Pin 3 Via a 220 Ohm resistor to GND
* Pin 4 PK0 RS pin for LCD module
* Pin 5 PK7 R/W pin for LCD module
* Pin 6 PK1 EN pin for LCD module
* Pin 7 Not used
* Pin 8 Not used
* Pin 9 Not used
* Pin 10 Mot used
* Pin 11 PK2 DB4 pin for LCD module
* Pin 12 PK3 DB5 pin for LCD module
* Pin 13 PK4 DB6 pin for LCD module
* Pin 14 PK5 DB7 pin for LCD module
* Pin 15 Via a 22 Ohm resistor to VCC LED backlight for LCD module
* Pin 16 GND

Codul Programului:

//On Dragon12+ board, the rows and columns of 4x4 keypad are connected to PORTA.

//See page 26 of Dragon12+ User's Manual

//As you press any key the ASCII value for the key is placed on the LEDs of PORTB

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include <hidef.h> /\* common defines and macros \*/

#include "mc9s12dg256.h" /\* derivative-specific definitions \*/

#define LCD\_DATA PORTK

#define LCD\_CTRL PORTK

#define RS 0x01

#define EN 0x02

void COMWRT4(unsigned char);

void DATWRT4(unsigned char);

void mSDelay(unsigned int);

const unsigned char keypad[4][4] = //maparea tastaturii

{

'1','2','3','A', //A='+'

'4','5','6','B', //B='-'

'7','8','9','C', //C='/'

'\*','0','#','D' //D='=', '#'='Clear'

};

unsigned char column,row;

void main(void){

int number1=0;

int number2=0;

int rezultat=0;

int buffer[10];

int i=0,j=0,k;

int aux=0;

int iesire=0; //conditie pentru cazuri speciale

int negativ=0; //indicator pentru rezultat negativ

int adunare=0;

int scadere=0;

int inmultire=0;

int impartire=0;

DDRB = 0xFF; //MAKE PORTB OUTPUT

DDRJ |= 0x02;

PTJ &= ~0x02; //ACTIVATE LED ARRAY ON PORT B

DDRP |= 0x0F; //

PTP |= 0x0F; //TURN OFF 7SEG LED

DDRA = 0x0F; //MAKE ROWS INPUT AND COLUMNS OUTPUT

DDRK = 0xFF;

COMWRT4(0x33); //reset sequence provided by data sheet

mSDelay(1);

COMWRT4(0x32); //reset sequence provided by data sheet

mSDelay(1);

COMWRT4(0x28); //Function set to four bit data length

//2 line, 5 x 7 dot format

mSDelay(1);

COMWRT4(0x06); //entry mode set, increment, no shift

mSDelay(1);

COMWRT4(0x0E); //Display set, disp on, cursor on, blink off

mSDelay(1);

COMWRT4(0x01); //Clear display

mSDelay(1);

COMWRT4(0x80); //set start posistion, home position

mSDelay(1);

while(1){ //OPEN WHILE(1)

do{ //OPEN do1

PORTA = PORTA | 0x0F; //COLUMNS SET HIGH

row = PORTA & 0xF0; //READ ROWS

}while(row == 0x00); //WAIT UNTIL KEY PRESSED //CLOSE do1

do{ //OPEN do2

do{ //OPEN do3

mSDelay(1); //WAIT

row = PORTA & 0xF0; //READ ROWS

}while(row == 0x00); //CHECK FOR KEY PRESS //CLOSE do3

mSDelay(15); //WAIT FOR DEBOUNCE

row = PORTA & 0xF0;

}while(row == 0x00); //FALSE KEY PRESS //CLOSE do2

while(1){ //OPEN while(1)

PORTA &= 0xF0; //CLEAR COLUMN

PORTA |= 0x01; //COLUMN 0 SET HIGH

row = PORTA & 0xF0; //READ ROWS

if(row != 0x00){ //KEY IS IN COLUMN 0

column = 0;

break; //BREAK OUT OF while(1)

}

PORTA &= 0xF0; //CLEAR COLUMN

PORTA |= 0x02; //COLUMN 1 SET HIGH

row = PORTA & 0xF0; //READ ROWS

if(row != 0x00){ //KEY IS IN COLUMN 1

column = 1;

break; //BREAK OUT OF while(1)

}

PORTA &= 0xF0; //CLEAR COLUMN

PORTA |= 0x04; //COLUMN 2 SET HIGH

row = PORTA & 0xF0; //READ ROWS

if(row != 0x00){ //KEY IS IN COLUMN 2

column = 2;

break; //BREAK OUT OF while(1)

}

PORTA &= 0xF0; //CLEAR COLUMN

PORTA |= 0x08; //COLUMN 3 SET HIGH

row = PORTA & 0xF0; //READ ROWS

if(row != 0x00){ //KEY IS IN COLUMN 3

column = 3;

break; //BREAK OUT OF while(1)

}

row = 0; //KEY NOT FOUND

break; //step out of while(1) loop to not get stuck

} //end while(1)

if(row == 0x10){

if (keypad[0][column]=='A') {

adunare=1;

number1=aux;

aux=0; // se verifica primul rand de taste + cazul apasarii tastei A pentru adunare, caz in care incepe sa se formeze termenul 2

DATWRT4('+');

mSDelay(1);

}else {

aux=aux\*10+(keypad[0][column]-'0'); // crearea termenilor

DATWRT4(keypad[0][column]); // afisare tasta apasata

mSDelay(1);

}

}

else if(row == 0x20)

{

if (keypad[1][column]=='B')

{

scadere=1;

number1=aux;

aux=0; // se verifica al doilea rand de taste + cazul apasarii tastei B pentru scadere, caz in care incepe sa se formeze termenul 2

DATWRT4('-');

mSDelay(1);

}

else

{

aux=aux\*10+(keypad[1][column]-'0');

DATWRT4(keypad[1][column]);

mSDelay(1);

}

}

else if(row == 0x40){

if (keypad[2][column]=='C') {

impartire=1;

number1=aux;

aux=0;// se verifica al treilea rand de taste + cazul apasarii tastei C pentru impartire, caz in care incepe sa se formeze termenul 2

DATWRT4('/');

mSDelay(1);

}else

{

aux=aux\*10+(keypad[2][column]-'0');

DATWRT4(keypad[2][column]);

mSDelay(1);

}

}

else if(row == 0x80){

if (keypad[3][column]=='D')

{

number2=aux;

DATWRT4('='); // se verifica cazul apasarii tastei D pentru efectuarea operatiei, caz in care se formeaza rezultatul

mSDelay(1);

if(adunare==1) // daca se indeplineste conditia adunare=1, operatia efectuata este cea de adunare

rezultat=number1+number2;

else if (scadere==1) // daca se indeplineste conditia scadere=1, operatia efectuata este cea de scadere

{

rezultat=number1-number2;

if (rezultat<0) {

rezultat=rezultat\*-1; // daca rezultatul este negativ, se seteaza indicatorul pe 1 pentru a se afisa semnul '-' in fata rezultatului

negativ=1;

}

}

else if (inmultire==1) // daca se indeplineste conditia inmultire=1, operatia efectuata este cea de inmultire

{

if (number1==0 || number2==0) // se trateaza cazul in care unul dintre termeni este 0, se afiseaza 0

{

iesire=1;

COMWRT4(0xC0);

mSDelay(1);

DATWRT4('0');

mSDelay(1);

}

else

rezultat=number1\*number2;

}

else if (impartire==1) // daca se indeplineste conditia impartire=1, operatia efectuata este cea de impartire

{

if (number2==0) // daca al doilea termen este 0, atunci se afiseaza mesajul "EROARE"

{

iesire=1;

COMWRT4(0xC0);

mSDelay(1);

DATWRT4('E');

mSDelay(1);

DATWRT4('R');

mSDelay(1);

DATWRT4('O');

mSDelay(1);

DATWRT4('A');

mSDelay(1);

DATWRT4('R');

mSDelay(1);

DATWRT4('E');

mSDelay(1);

}

else

rezultat=number1/number2;

}

if (iesire==0)

{

COMWRT4(0xC0); //se seteaza afisarea pe a doua linie a display-ului LCD

mSDelay(1);

if (negativ==1)

{

DATWRT4('-');

mSDelay(1);

}

if (rezultat==0)

{

DATWRT4('0');

mSDelay(1);

}

while(rezultat!=0) //se creaza vectorul pentru afisare

{

buffer[i++]=rezultat%10;

rezultat=rezultat/10;

}

for (k=i-1; k>=0; k--) //se afiseaza rezultatul

{

DATWRT4(buffer[k]+'0');

mSDelay(1);

}

}

}

else if (keypad[3][column]=='\*')

{

inmultire=1;

number1=aux;

aux=0; // se verifica ultimul rand de taste + cazul apasarii tastei '\*' pentru inmultire, caz in care incepe sa se formeze termenul 2

DATWRT4(keypad[3][column]);

mSDelay(1);

}

else if (keypad[3][column]=='#')

{

COMWRT4(0x01); //Clear display

mSDelay(1000);

COMWRT4(0x80); //set start posistion, home position

number1=0; // termenul 1

number2=0; // termenul 2

rezultat=0; //rezultatul

memset(&buffer[10], 0, sizeof(buffer));

i=0,j=0,k;

aux=0;

iesire=0; //conditie pentru cazuri speciale

negativ=0; //indicator pentru rezultat negativ

adunare=0;

scadere=0; //indicator pentru operatia efectuata

inmultire=0;

impartire=0;

}

else

{

aux=aux\*10+(keypad[3][column]-'0');

DATWRT4(keypad[3][column]);

mSDelay(1);

}

}

do{

mSDelay(15);

PORTA = PORTA | 0x0F; //columns set high

row = PORTA & 0xF0; //read rows

}while(row != 0x00); //check debounce

}

}

void COMWRT4(unsigned char command)

{

unsigned char x;

x = (command & 0xF0) >> 2; //shift high nibble to center of byte for Pk5-Pk2

LCD\_DATA =LCD\_DATA & ~0x3C; //clear bits Pk5-Pk2

LCD\_DATA = LCD\_DATA | x; //sends high nibble to PORTK

mSDelay(1);

LCD\_CTRL = LCD\_CTRL & ~RS; //set RS to command (RS=0)

mSDelay(1);

LCD\_CTRL = LCD\_CTRL | EN; //rais enable

mSDelay(5);

LCD\_CTRL = LCD\_CTRL & ~EN; //Drop enable to capture command

mSDelay(15); //wait

x = (command & 0x0F)<< 2; // shift low nibble to center of byte for Pk5-Pk2

LCD\_DATA =LCD\_DATA & ~0x3C; //clear bits Pk5-Pk2

LCD\_DATA =LCD\_DATA | x; //send low nibble to PORTK

LCD\_CTRL = LCD\_CTRL | EN; //rais enable

mSDelay(5);

LCD\_CTRL = LCD\_CTRL & ~EN; //drop enable to capture command

mSDelay(15);

}

void DATWRT4(unsigned char data)

{

unsigned char x;

x = (data & 0xF0) >> 2;

LCD\_DATA =LCD\_DATA & ~0x3C;

LCD\_DATA = LCD\_DATA | x;

mSDelay(1);

LCD\_CTRL = LCD\_CTRL | RS;

mSDelay(1);

LCD\_CTRL = LCD\_CTRL | EN;

mSDelay(1);

LCD\_CTRL = LCD\_CTRL & ~EN;

mSDelay(5);

x = (data & 0x0F)<< 2;

LCD\_DATA =LCD\_DATA & ~0x3C;

LCD\_DATA = LCD\_DATA | x;

LCD\_CTRL = LCD\_CTRL | EN;

mSDelay(1);\

LCD\_CTRL = LCD\_CTRL & ~EN;

mSDelay(15);

}

void mSDelay(unsigned int itime){

unsigned int i; unsigned int j;

for(i=0;i<itime;i++)

for(j=0;j<4000;j++);

}

Bibliografie

* Es\_Lab1.pdf
* S12DP256PIMV2-portguide.pdf
* dragon12-plus2\_hcs12\_manual.pdf
* <http://www.cae.tntech.edu/~scanfield/me4370/labs/calculator/index.html>
* Lecture\_24.pdf