

Convertorul analog digital

Obiectiv

Familiarizarea cu placa de dezvoltare ce va fi folosită pe parcursul ședințelor de laborator și cu instrumentele software folosite (mediul de dezvoltare MPLAB și compilatorul C30).

Introducere

Un convertor permite transformarea mai multor linii analogice în semnale digitale, dar nu simultan. Numărul maxim de linii analogice care pot fi transformate de un convertor A/D reprezintă numărul **canalelor** acestuia, 9 canale în cazul DSP icului.

Ca orice alt modul, și convertorul A/D are atașat un bit pentru activarea sau dezactivarea lui. Acest bit este ADON din registrul ADCON0. Dacă ADON este 1, modulul A/D este activat. În caz contrar, modulul este inactiv.

Configurarea convertorului si realizarea unei conversii

- Configurarea pinilor analogici (AD1PCFGH<15:0> or AD1PCFGL<15:0> din registrul ADCON1);
- Configurarea corespunzătoare a registrilor pentru tensiuni de referință pinii să fie de ieșire, iar pentru canalul ales pinul corespunzător să fie de intrare(AD1CON2<15:13>);

- Selectarea ceasului analogic de conversie pentru a se potrivi ratei de date dorite, cu ceasul procesorului (AD1CON3 <7:0>);
- Determinarea numărului de canale S/H folosite (AD1CON2 <9:8> și AD1PCFGH <15:0> și AD1PCFGL <15:0>);
- Selectarea frecvenței adecvate necesară conversiei (AD1CON1 <07:5> și AD1CON3 <12:8>)
- Selectarea modului în care rezultatele sunt prezentate în buffer (AD1CON1 <9:8>); Selectarea formatului de aliniere a rezultatului;
- Activarea modului de conversie A/D (AD1CON1<15>);

Opțional, dacă se dorește utilizarea întreruperilor, se pot adăuga următoarele setări:

- Ștergerea bitului AD1IF;
- Selectarea priorității întreruperii ADC.

Caracteristici Cheie

Convertorul AD pe 10 biți:

- Apropierea succesive de conversie (SAR);
- Viteza de conversie până la 1Msps;
- Până la 9 pini de intrare analogică;
- Eșantionarea simultană până la 4 pini de intrare;
- Mod de scanarea automată al canalelor;
- Sursa de trigger selectabilă poate fi declanșată;
- Patru opțiuni de aliniere a rezultatului (cu semn, fără semn / fracțional, integer);

- Operații în modurile "Sleep" și "Idle"

Convertorul AD pe 12 biți:

- Viteza de conversie până la 500 Ksps;
- Există doar un singur amplificator, prin urmare eșantionarea simultană a mai multor canale nu este posibilă

Convertorul AD poate avea până la 32 de pini de intrare analogice: AN0-AN31. Aceste intrări analogice sunt conectate prin multiplexoare la patru amplificatoare (S&H): CH0-CH3. Multiplexoare analogice de intrare au două seturi de biți de control, desemnate ca MUXA (CHySA / CHyNA) și MUXB (CHySB / CHyNB). Acești biți de control selectează o anumită intrare analogică pentru conversie. MUXA și MUXB poate selecta alternativ de intrarea analogică pentru conversie.

Orice subgrup de intrări analogice (AN0 / AN31 în funcție de disponibilitate) pot fi selectate la cererea userului. Intrările selectate sunt convertite în ordine crescătoare folosind CH0.

Convertorul AD suportă eșantionarea simultană folosind mai multe canale de S & H pentru eșantionarea intrărilor în același timp și efectuează apoi conversia pentru fiecare canal secvențial. În mod implicit, mai multe canale sunt eșantionate și convertite secvențial.

Pentru dispozitive cu DMA, convertorul AD este conectat la un singur rezultat-tampon. Cu toate acestea, rezultatul conversiei multiple poate fi stocat în

bufferul DMA RAM fără a suprasolicita procesorul când DMA ul este folosit cu comparatorul AD. Fiecare rezultat de conversie este transformată în una dintre cele patru formate de ieșire pe 16 biți atunci când este citită din buffer.

Observație:

1. "Y" este utilizat cu MUXA și MUXB pentru a specifica numerele de canal S sau H ($y = 0$ sau 123).
2. Există doi pini de intrare analogică pentru tensiunea de referință externă (VREF +, VREF-). Aceste intrări de referință de tensiune pot fi împărtășite cu alți pini analogici de intrare.

Regiștri de control:

1. ADxCON1: ADCx Control Register 1(1)
2. ADxCON2: ADCx Control Register 2(1)
3. ADxCON3: ADCx Control Register 3(1)
4. ADxCON4: ADCx Control Register 4(1,2)
5. ADxCHS123: ADCx Input Channel 1, 2, 3 Select Register(1)
6. ADxCHS0: ADCx Input Channel 0 Select Register(1)
7. AD1CSSH: ADC1 Input Scan Select Register High(1)
8. ADxCSSL: ADCx Input Scan Select Register Low(1)
9. AD1PCFGH: ADC1 Port Configuration Register High(1,3)
10. ADxPCFGL: ADCx Port Configuration Register Low(1)

Regiștri ADxCON1, ADxCON2 și ADxCON3 controlează operațiile convertorului AD.

Registrul ADxCON4 stabilește numărul de conversii rezultate stocate într-un tampon de DMA pentru fiecare intrare analogică.

Regiștrii ADxCHS123 și ADxCHS0 selectați pinii de intrare care să fie conectați la S & H(amplificatori).

Regiștrii ADCSSH/L selctează ca intrările să fie secvențial scanate.

Regiștrii ADxPCFGH / L configurează pinii analogici de intrare, ca intrări analogice sau digitale I / O.

Registru de control AD1CON1:

R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
ADON	—	ADSIDL	ADDMABM	—	AD12B	FORM<1:0>	
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0 HC,HS	R/C-0 HC, HS
SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE
bit 7						bit 0	

Legend:	HC = Cleared by hardware	HS = Set by hardware
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

bit 15 **ADON:** ADC Operating Mode bit

1 = ADC module is operating

0 = ADC is off

bit 14 **Unimplemented:** Read as '0'

bit 13 **ADSIDL:** Stop in Idle Mode bit

1 = Discontinue module operation when device enters Idle mode

0 = Continue module operation in Idle mode

bit 12 **ADDMABM:** DMA Buffer Build Mode bit

1 = DMA buffers are written in the order of conversion. The module provides an address to the DMA channel that is the same as the address used for the non-DMA stand-alone buffer.

0 = DMA buffers are written in Scatter/Gather mode. The module provides a scatter/gather address to the DMA channel, based on the index of the analog input and the size of the DMA buffer.

bit 11 **Unimplemented:** Read as '0'

bit 10 **AD12B:** 10-bit or 12-bit Operation Mode bit

1 = 12-bit, 1-channel ADC operation

0 = 10-bit, 4-channel ADC operation

Registrul de control AD1CON2:

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
VCFG<2:0>			—	—	CSCNA	CHPS<1:0>	
bit 15			bit 8				

R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	—	SMPI<3:0>				BUFM	ALTS
bit 7							
							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 15-13 **VCFG<2:0>**: Converter Voltage Reference Configuration bits

	ADREF+	ADREF-
000	AVDD	AVSS
001	External VREF+	AVSS
010	AVDD	External VREF-
011	External VREF+	External VREF-
1xx	AVDD	AVSS

bit 12-11 **Unimplemented**: Read as '0'

bit 10 **CSCNA**: Scan Input Selections for CH0+ during Sample A bit

1 = Scan inputs

0 = Do not scan inputs

bit 9-8 **CHPS<1:0>**: Selects Channels Utilized bits

When AD12B = 1, CHPS<1:0> is: U-0, Unimplemented, Read as '0'

1x = Converts CH0, CH1, CH2 and CH3

01 = Converts CH0 and CH1

00 = Converts CH0

bit 7 **BUFS**: Buffer Fill Status bit (only valid when BUFM = 1)

1 = ADC is currently filling buffer 0x8-0xF, user should access data in 0x0-0x7

0 = ADC is currently filling buffer 0x0-0x7, user should access data in 0x8-0xF

Registrul de control AD1CON3:

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	—	SAMC<4:0> ⁽¹⁾				
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS<7:0> ⁽²⁾							
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15 **ADRC:** ADC Conversion Clock Source bit
1 = ADC internal RC clock
0 = Clock derived from system clock

bit 14-13 **Unimplemented:** Read as '0'

bit 12-8 **SAMC<4:0>:** Auto Sample Time bits⁽¹⁾
11111 = 31 TAD
•
•
•
00001 = 1 TAD
00000 = 0 TAD

bit 7-0 **ADCS<7:0>:** ADC Conversion Clock Select bits⁽²⁾
11111111 = Reserved

Registrul de control AD1CON4:

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DMABL<2:0>		
bit 7							bit 0

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

bit 15-3 **Unimplemented:** Read as '0'

bit 2-0 **DMABL<2:0>:** Selects Number of DMA Buffer Locations per Analog Input bits
111 = Allocates 128 words of buffer to each analog input
110 = Allocates 64 words of buffer to each analog input
101 = Allocates 32 words of buffer to each analog input
100 = Allocates 16 words of buffer to each analog input
011 = Allocates 8 words of buffer to each analog input
010 = Allocates 4 words of buffer to each analog input
001 = Allocates 2 words of buffer to each analog input
000 = Allocates 1 word of buffer to each analog input

Registru de selecție AD1CHS123:

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CH123NB<1:0>		CH123SB
bit 15						bit 8	

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CH123NA<1:0>		CH123SA
bit 7							bit 0

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-9 **CH123NB<1:0>:** Channel 1, 2, 3 Negative Input Select for Sample B bits

dsPIC33FJ32MC302, dsPIC33FJ64MC202/802, and dsPIC33FJ128MC202/802 devices only:

If AD12B = 1:

11 = Reserved
10 = Reserved
01 = Reserved
00 = Reserved

If AD12B = 0:

11 = Reserved
10 = Reserved
01 = CH1, CH2, CH3 negative input is VREF-
00 = CH1, CH2, CH3 negative input is VREF-

dsPIC33FJ32MC304, dsPIC33FJ64MC204/804, and dsPIC33FJ128MC204/804 devices only:

If AD12B = 1:

11 = Reserved
10 = Reserved
01 = Reserved
00 = Reserved

Registru de selecție AD1CHS0:

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	CH0SB<4:0>				
bit 15							bit 8

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	CH0SA<4:0>				
bit 7							bit 0

Legend:

R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

bit 15 **CH0NB:** Channel 0 Negative Input Select for Sample B bit

1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is VREF-

bit 14-13 **Unimplemented:** Read as '0'

bit 12-8 **CH0SB<4:0>:** Channel 0 Positive Input Select for Sample B bits

dsPIC33FJ32MC304, dsPIC33FJ64MC204/804, and dsPIC33FJ128MC204/804 devices only:

01000 = Channel 0 positive input is AN8

Registrul de selecție AD1CSSL:

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	CSS8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 15-9 **Unimplemented:** Read as '0'

bit 8-0 **CSS<8:0>:** ADC Input Scan Selection bits

1 = Select ANx for input scan

0 = Skip ANx for input scan

Note 1: On devices without nine analog inputs, all AD1CSSL bits can be selected by user application. However, inputs selected for scan without a corresponding input on device converts VREFL.

2: CSSx = ANx, where x = 0 through 8.

Registrul de selecție AD1PCFGL:

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PCFG8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 15-9 **Unimplemented:** Read as '0'

bit 8-0 **PCFG<8:0>:** ADC Port Configuration Control bits

1 = Port pin in Digital mode, port read input enabled, ADC input multiplexor connected to AVss

0 = Port pin in Analog mode, port read input disabled, ADC samples pin voltage

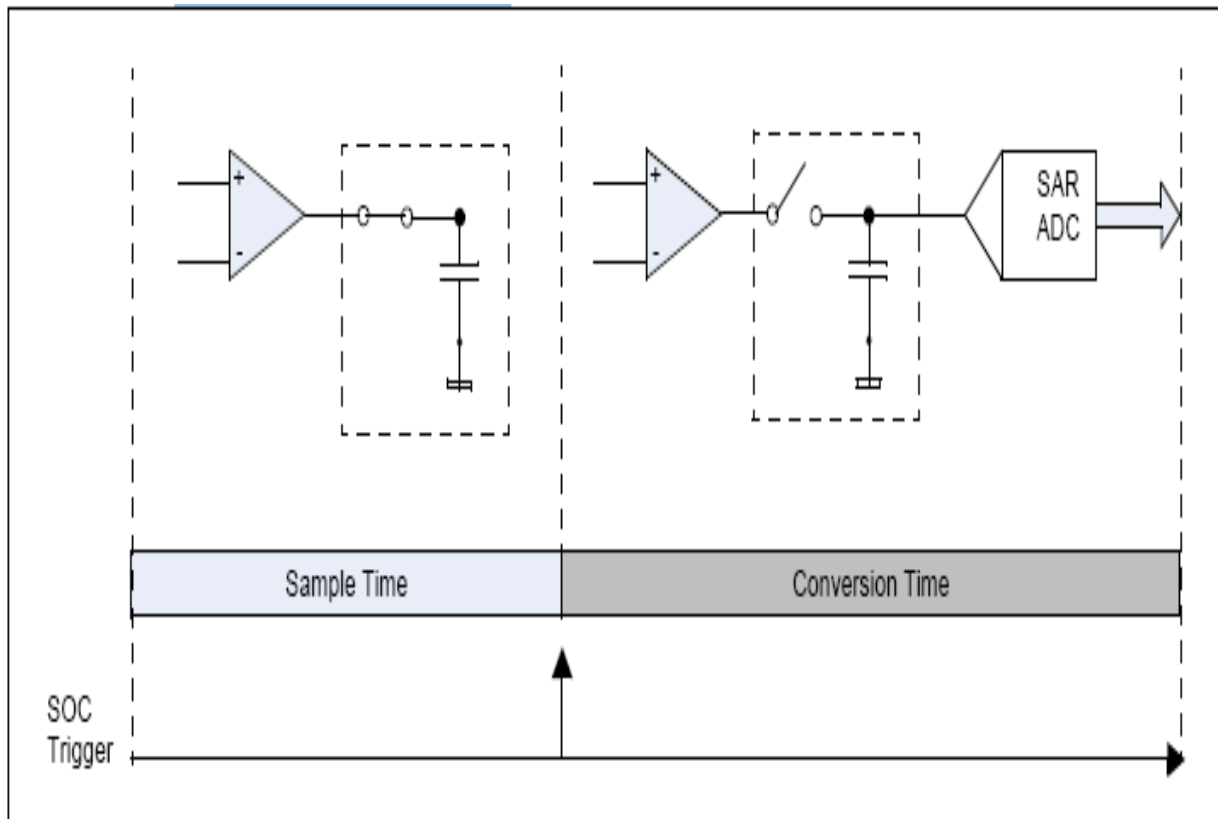
Prezentarea Sample Time/Conversion Time: (eșantionării și conversiei a unei secvențe)

Conversia A/D este un proces în trei etape:

1. Semnalul tensiunii de intrare este conectată la condensator.
2. Condensatorul este deconectat de la intrare.
3. Tensiunea tocată este convertită în numărul de biți echivalent.

Cele două etape distincte, Sample Time și Conversion Time, sunt controlate independent:

Sample Conversion Sequence



Sample Time

Intrarea analog selectată este conectat la condensator eşantion. Există un timp minim de probe pentru a se asigura că amplificatorul S & H oferă precizia dorită pentru conversia A/D.

Faza de eşantionare poate fi setată să pornească automat la conversie sau manual, prin setarea bitului de eşantionare(SAMP) în registrul de control ADC 1 (ADxCON1 <1>). Faza de eşantionare este controlată de bitul auto-esantionare(ASAM) în registrul de control ADC 1(ADxCON1<2>).

Dacă auto-esantionarea este activată, timpul de esantionare(TSMP) al modului ADC este egal cu numărul de ciclu TAD definit de SAMC<4:0> biti(ADxCON3<12:8>)

$$TSMP = SAMC<4:0> \cdot TAD$$

\

Timpul de Conversie

Trigerul de la începutul conversiei (SOC) încheie perioada de esantionare și porneste o conversie A/D. În timpul perioadei de conversie, condensatorul de esantionare este deconectat de la multiplexor, și voltajul stocat este convertit în biți digitali echivalenți. Timpul de conversie pt 10-biți și 12-biți sunt descriși de ecuațiile de mai jos :

10-Bit ADC

$$TCONV = 12 \cdot TAD$$

unde:

$TCONV$ = Timpul de conversie

TAD = ADC Clock Period

12-Bit ADC

$TCONV = 14 \cdot TAD$

$TCONV$ = Timpul de conversie

TAD = ADC Clock Period

Exemplu lucru cu conversie AD

main.c

```
#if defined(__dsPIC33F__)
#include "p33fxxxx.h"
#endif

#include "adcDrv1.h"

// Select Internal FRC at POR
_FOSCSEL(FNOSC_FRC);
// Enable Clock Switching and Configure
_FOSC(FCKSM_CSECMD & OSCIOFNC_OFF);
_FWDT(FWDTEN_OFF);          // Watchdog Timer Enabled/disabled by
user software

void initPLL(void)
{
// Configure PLL prescaler, PLL postscaler, PLL divisor
PLLFBFBD = 41;              // M = 43 FRC
//PLLFBFBD = 30;            // M = 32 XT
CLKDIVbits.PLLPOST=0;       // N1 = 2
CLKDIVbits.PLLPRE=0;        // N2 = 2

// Initiate Clock Switch to Internal FRC with PLL (NOSC = 0b001)
__builtin_write_OSCCONH(0x01); // FRC
//__builtin_write_OSCCONH(0x03); // XT
__builtin_write_OSCCONL(0x01);

// Wait for Clock switch to occur
while (OSCCONbits.COSC != 0b001); // FRC
//while (OSCCONbits.COSC != 0b011); // XT

// Wait for PLL to lock
while(OSCCONbits.LOCK!=1) {};
}
```

```

void main (void)
{
// Disable Watch Dog Timer
    RCONbits.SWDTEN=0;

// Peripheral Initialisation
    initPLL();
    initAdc1();           // Initialize ADC
    initTmr3();           // Initialise TIMER 3

// Background Loop
    while (1)
    {
        }
    }
}

```

adcDrv.c

```

#if defined(__dsPIC33F__)
#include "p33fxxxx.h"
#endif

#include "adcDrv1.h"

#define SAMP_BUFF_SIZE      10           // Dimensiunea
buffer-ului in care se salveaza rezultatele conversiei

//INITIALIZARE ADC PENTRU SCANARE CANAL AN4(RB2)
void initAdc1(void)
{
    AD1CON1bits.AD12B = 1; // conversie AD pe 12 biti
    AD1CON1bits.FORM = 0; // rezultat conversie integer
    AD1CON1bits.SSRC = 2; // timerul 3 starteaza conversia
    AD1CON1bits.ASAM = 1;  // incepe esantionarea noii valori
imediat dupa terminarea
                           // unei conversii

    AD1CON2bits.CSCNA = 1; // scaneaza intrarile pe CH0+ in timpul
achizitiei A
    AD1CON2bits.CHPS = 0; // converteste doar CH0
    AD1CON2bits.SMPI = 0; // incrementeaza adresa DMA dupa
terminarea fiecarei conversii
}

```

```

    AD1CON3bits.ADRC = 0; // foloseste ceasul magistralei
    AD1CON3bits.ADCS = 63; // Timpul necesar unei conversii este
    de 22.4 us
    // Ceasul pentru conversia AD are formula  $T_{ad}=T_{cy}*(adcs+1)$ 
    //  $T_{ad}=T_{cy}*(adcs+1)=(1/40)*64=1.6us$ 

    // Se seteaza intrarile analogice
    AD1CSSLbits.CSS4 = 1; // Selectam intrarea analogica
    AN4(RB2) pentru a fi scanata

    // Scriem registrul de configurare al portului
    // Se va folosi doar registrul low al portului de configurare deoarece
    dsPIC33FJ128MC802
    // nu are implementati mai mult de 6 pini pentru ADC
    AD1PCFGL=0xFFFF; // Setam toti pinii portului ADC1 pe
    modul digital,
    // si activeaza citirea la intrarea portului
    AD1PCFGLbits.PCFG4 = 0; // Setam pinul AN4(RB2) pe intrare
    analogica,
    // ADC verifica voltajele pe acel pin (achizitie AD)
    IFS0bits.AD1IF = 0; // Reseteaza flag-ul intreruperii
    convertorului AD
    IPC3bits.AD1IP = 6; // Seteaza prioritatea intreruperii
    convertorului AD
    IEC0bits.AD1IE = 1; // Permite intreruperea
    convertorului AD

    AD1CON1bits.ADON = 1;
}

// Timer-ul 3 este setat sa starteze conversia AD la fiecare 125
microsecunde (8Khz Rate).
void initTmr3()
{
    TMR3 = 0;
    PR3 = 4999;
    T3CONbits.TON = 1; // Start Timer 3
}

int ain4Buff[SAMP_BUFF_SIZE];
int sampleCounter=0;

// rutina de tratare a intreruperii convertorului AD
void __attribute__((interrupt, no_auto_psv)) _ADC1Interrupt(void)

```

```
{
    ain4Buff[sampleCounter++] = ADC1BUF0;
    if(sampleCounter == SAMP_BUFF_SIZE)
        sampleCounter = 0;
    IFS0bits.AD1IF = 0;      // Achita intreruperea convertorului AD
}
```

adcDrv.h

```
#ifndef __ADCDRV_H__
#define __ADCDRV_H__
```

// External Functions

```
extern void initAdc1(void);
extern void initTmr3();
extern void __attribute__((__interrupt__)) _ADC1Interrupt(void);
```

```
#endif
```