

CONVERTIDOR ADC

Objectius.

- Conèixer les principals característiques del convertidor analògic digital.
- Adquirir els coneixements necessaris per programar correctament els registres que controlen el convertidor.
- Crear una funció que ens permeti obtenir la conversió del canal que especifiquem.

Característiques.

La majoria de sensors ens proporcionen un voltatge que és proporcional a la mesura que es vol prendre. Així, un sensor de temperatura ens donarà un voltatge que serà proporcional a la temperatura que està llegint. El mateix passa si volem mesurar una pressió, una velocitat, una direcció, un cabdal, etc.

El μ Controlador únicament és capaç de processar informació digital. És per això que cal la transformació d'un voltage (senyal analògic) en una combinació de 1's i 0's (senyal digital). El circuit que permet fer aquesta transformació rep el nom de conversor analògic digital o simplement ADC.

El μ Controlador 80C552, disposa d'un conversor ADC amb aquestes característiques:

- Conversor A/D d'aproximacions successives de 10 bits.
- 8 canals d'entrada multiplexats
- Temps de conversió 50 cicles màquina. ($50 * 12/11.0592\text{Mcps} = 54 \mu\text{seg}$)
- Dues tensions de referència: Vref+ connectada a 5v i Vref- connectada a un divisor de tensió variable amb una resistència ajustable.
- Pin STADC per iniciar la conversió per hard.
- Dos registres per controlar-lo: ADCON i ADCH.

Resolució.

El conversor és de 10 bits. Això vol dir que podem tenir $2^{10}=1024$ combinacions possibles. La resolució del conversor és la tensió necessària per canviar un bit. És evident que variarà segons el marge de tensió que volem convertir. El marge a convertir el fixen les tensions de referència. Posem un exemple:

Vref+=5v i Vref-=0v.

El marge és de 5v. Això vol dir que la resolució serà de $5/1024=4.88\text{mV}$.

voltatge	Conversió binari										Decimal
5v	1	1	1	1	1	1	1	1	1	1	1023
1v	0	0	1	1	0	0	1	1	0	1	205
9.76mV	0	0	0	0	0	0	0	0	1	0	2
4.88mV	0	0	0	0	0	0	0	0	0	1	1
0v	0	0	0	0	0	0	0	0	0	0	0

Si volem conèixer el resultat de la conversió per un voltage qualsevol farem:

$$\text{Resultat}=(\text{voltage}-V_{\text{ref-}})/\text{resolució}$$

Per 1V tindrem:

$$\text{Resultat}=(1000-0)/4,88=205$$

Anem a veure un altra exemple en el qual hem variat la $V_{\text{ref-}}$ a 3v.

$V_{\text{ref+}}=5\text{v}$ i $V_{\text{ref-}}=3\text{v}$

El marge ara és de 2v. Això vol dir que la resolució serà de $2/1024=1.95\text{mV}$.

voltage	Conversió binari										Decimal
5v	1	1	1	1	1	1	1	1	1	1	1023
4v	1	0	0	0	0	0	0	0	0	0	512
3.00390mV	0	0	0	0	0	0	0	0	1	0	2
3.00195mV	0	0	0	0	0	0	0	0	0	1	1
3v	0	0	0	0	0	0	0	0	0	0	0

Si volem conèixer el resultat de la conversió per 4v farem:

$$\text{Resultat}=(4000-3000)/1.95=512$$

El registre de control ADCON.

És un registre de 8 bits que no és direccionable bit a bit. Permet iniciar la conversió del canal que seleccionem i ens indica quan ha finalitzat per poder llegir el resultat.

7	6	5	4	3	2	1	0
ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0

AADR0-2	Són tres bits que permeten escollir a quin dels 8 canals farem la conversió
ADCS	Quan es posa a 1 inicia la conversió del canal seleccionat. Quan acaba la conversió, es posa a zero automàticament.
ADCI	És un Flag que es posa a 1 per indicar que la conversió ha finalitzat. Cal posar-lo a zero per programa abans d'iniciar una nova conversió.
ADEX	Quan està a 1 permet iniciar la conversió externament per el pin STADC. En aquest cas el bit ADCS no tindrà cap efecte. Si està a 0 aleshores l'inici es fa posant a 1 el bit 3 ADCS. Ara el pin STADC no té cap efecte.
ADC.0-1	Són els dos bits més baixos del resultat de la conversió.

El registre ADCH.

És un registre de 8 bits que conté els 8 bits més alts del resultat de la conversió.

7	6	5	4	3	2	1	0
ADC.9	ADC.8	ADC.7	ADC.6	ADC.5	ADC.4	ADC.3	ADC.2

Per obtenir el resultat total de 10 bits, haurem d'agafar els dos bits més baixos que es troben al registre ADCON i sumar-los amb els 8 més alts d'ADCH.

Passos per fer la conversió

Volem fer la conversió del canal 3 i l'inici de la conversió la volem fer internament per programa

1. Escollir quina entrada analògica del port P5 volem fer la conversió. Ho indicarem als tres bits més baixos d'ADCON. Pel canal 3 posarem: AADR2=0, AADR1=1 i AADR0=1.
2. Posarem la bandera de fi de conversió a zero: ADCI=0.
3. Seleccionar que l'inici serà intern: ADEX=0.
4. Iniciar la conversió posant ADCS=1.
5. Esperar que ADCI=1. Això voldrà dir que ha finalitzat la conversió i que el resultat és vàlid.
6. Llegir el resultat.

El programa per fer la conversió del canal 0 seria així:

```
/* **** */
****      Nom:          adc.c          ****
****      Programador: Josep Reixach - reixach@eia.udg.es      ****
****      Data:         14-03-01       ****
****      Descripció:   Fer la conversió del canal 0          ****
/* **** */
#include <io552.h>

main()
{
  int alta, resultat;
  ADCON=0x80; /* canal 0, ADEX=0, ADCI=0, ADCS=1 - Posem en marxa - */
  while ((ADCON & 0x10)==0); /* Esperem el final de la conversió */
  alta=ADCH;
  resultat=(alta<<2 | ADCON>>6); /* resultat de la conversió 0-1023 */
}
```

Funció convadc()

Per treballar millor, s'ha creat una funció a la que s'indica el canal que volem convertir i ens retorna una variable entera amb el resultat de la conversió.

```

/*****
/****      Nom:          fun_adc.h          ****/
/****      Aplicació:   Placa DS552       ****/
/****      Programador: Josep Reixach     ****/
/****      Data:        Desembre 98      ****/
/****      Descripció:  És una funció que retorna la conversió ****/
/****                        ADC del canal especificat          ****/
/*****
/*
Per fer la conversió cal seguir aquests passos:

1.- Indicar al registre ADCON.2,ADCON.1,ADCON.0 el canal que seleccionem per fer la conversió.
    El Canal 7 no es pot fer servir. S'utilitza pel RUN/LOAD.
2.- Posar ADCI=0 (ADCON.4). És l'interrupt flag que cal posar a zero per programa abans de fer una nova conversió.
3.- Posar ADCS=1 (ADCON.3). És el bit d'start. Per programa l'hem de tornar a zero.
    Per iniciar la conversió també, ho podem fer per l'entrada externa STADC. Això ho podem escollir amb ADEX=1 (ADCON.5).
    Si ADEX=0 aleshores la conversió no es pot iniciar externament.
4.- La conversió finalitza quan ADCI=1 (ADCON.4)
5.- Els bits més baixos els podem recollir a ADCON.7,ADCON.6
    Els bits més alts són al registre ADCH.

El registre ADCON no és accessible bit a bit.
Amb 10 bits tenim 1024 combinacions diferents. La placa porta una resistència ajustable multivolta que permet variar la Vref-. per tant la resolució serà:
    resolució = (5v-Vref-)/1024
En el cas que Vref-=0, aleshores la resolució es de 4.88mV.
*/
/*****
/*
int conv_adc(char canal)
/*****
/*
És una funció que retorna la conversió de 10 bits del canal especificat com a paràmetre.
exemple:
int conv;
conv=conv_adc(0); la conversió del canal zero serà assignada a conv.
*/
/*-----*/
int conv_adc(char canal)
{
char n;
int alta,conversio;
#define start_on    ADCON=ADCON | 0x08;    /* adcs=1 */
#define start_off   ADCON=ADCON & 0xf7;   /* adcs=0 */
#define busy_off    ADCON=ADCON & 0xef;   /* adci=0 */
#define extern_off  ADCON=ADCON & 0xdf;   /* adex=0 */

ADCON=(ADCON & 0xf8) | canal);
busy_off;
extern_off;
start_on;
while((ADCON & 0x10)==0);
alta=ADCH; /* Passem de byte a int */
conversio = (alta<<2 | ADCON>>6);
return conversio;
}

```