

ADC

Chuyển đổi analog sang kỹ thuật số là gì?

Đó là một quá trình thu tín hiệu điện tương tự (chẳng hạn như âm thanh được thu bởi micrô) và chuyển đổi nó thành một loạt các giá trị "Kỹ thuật số" số để được lưu trữ / xử lý bởi máy tính kỹ thuật số hoặc DSP. Thiết bị điện tử được sử dụng cho quá trình chuyển đổi này đã được biết đến là A / D hoặc **ADC** (Bộ chuyển đổi tương tự sang kỹ thuật số).

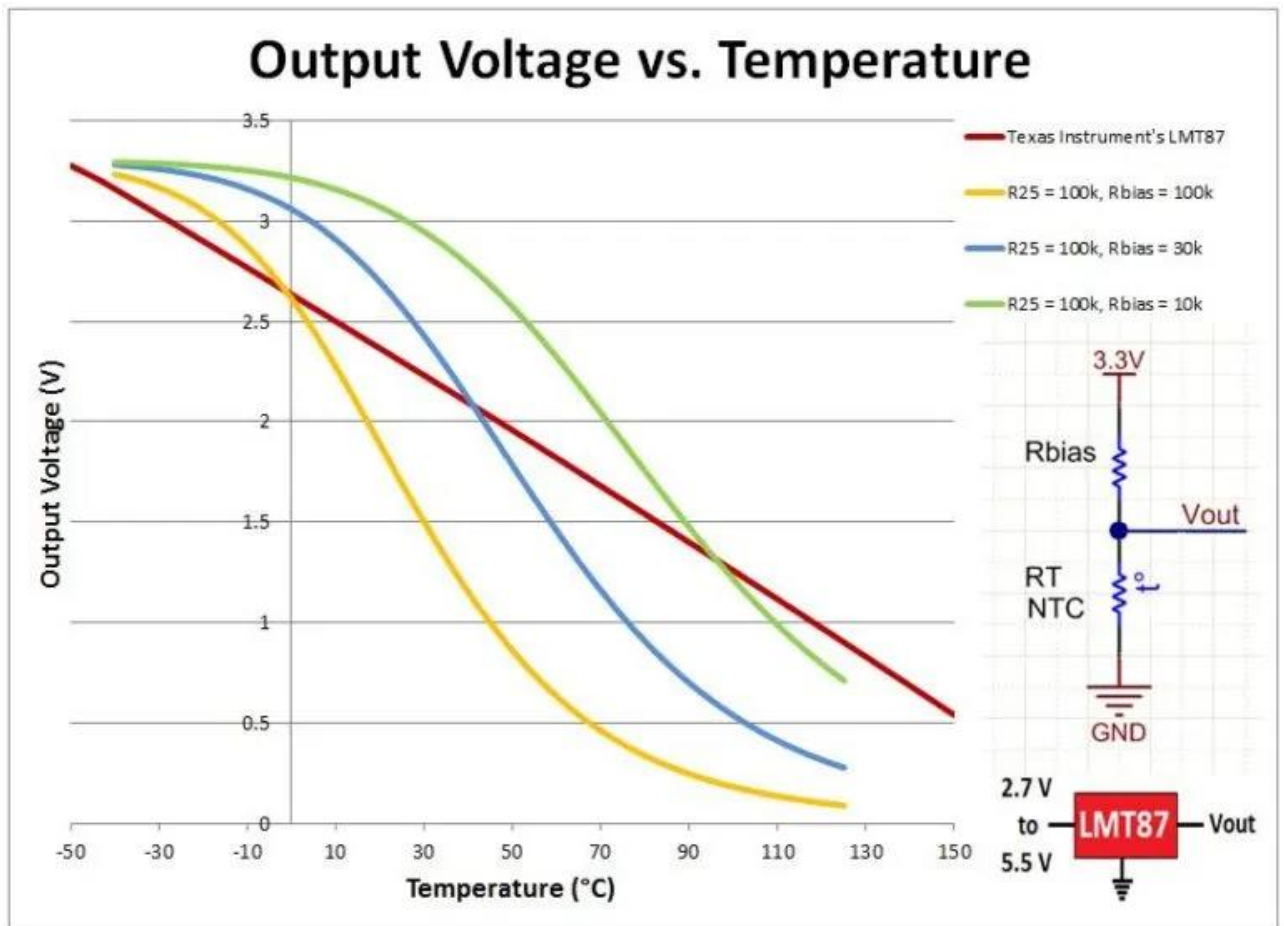
Có rất nhiều loại bộ chuyển đổi A / D khác nhau trên mạng. Và tùy thuộc vào cách nó hoạt động và mức độ hoàn thành công việc, bạn phải trả nhiều hơn hoặc ít hơn tương ứng cho mỗi đơn vị ADC.

Và trước khi điều tra thêm về các loại ADC và chi tiết kỹ thuật của các nguyên tắc làm việc, chúng ta hãy đặt câu hỏi về tính hữu ích của chuyển đổi A / D. Tại sao chúng ta cần sử dụng ADC ngay từ đầu?

Nhu cầu chuyển đổi tín hiệu tương tự sang dữ liệu kỹ thuật số của chúng ta bắt nguồn từ thực tế là máy tính của chúng ta là máy tính kỹ thuật số. Họ chỉ không thể xử lý tín hiệu tương tự và do đó, cần có một thiết bị chuyển đổi tín hiệu từ miền tương tự sang miền kỹ thuật số (ADC).

Hầu hết các tín hiệu đều có bản chất tương tự và các cảm biến điện tử mà chúng tôi đang sử dụng để nắm bắt những hiện tượng này cũng tương tự. Ví dụ: cảm biến nhiệt độ chuyển đổi nhiệt độ ở ° C thành điện áp tương tự tỷ lệ thuận với giá trị của nhiệt độ. Micrô, cảm biến áp suất, cảm biến ánh sáng cũng vậy, v.v.

Do đó, chúng ta cần một cách để đọc điện áp tương tự và chuyển đổi nó thành các giá trị kỹ thuật số mà chúng ta có thể lập trình máy tính của mình để thao tác với nó bằng toán học để đạt được sự giám sát và / hoặc kiểm soát thời gian thực hoặc bất cứ điều gì.

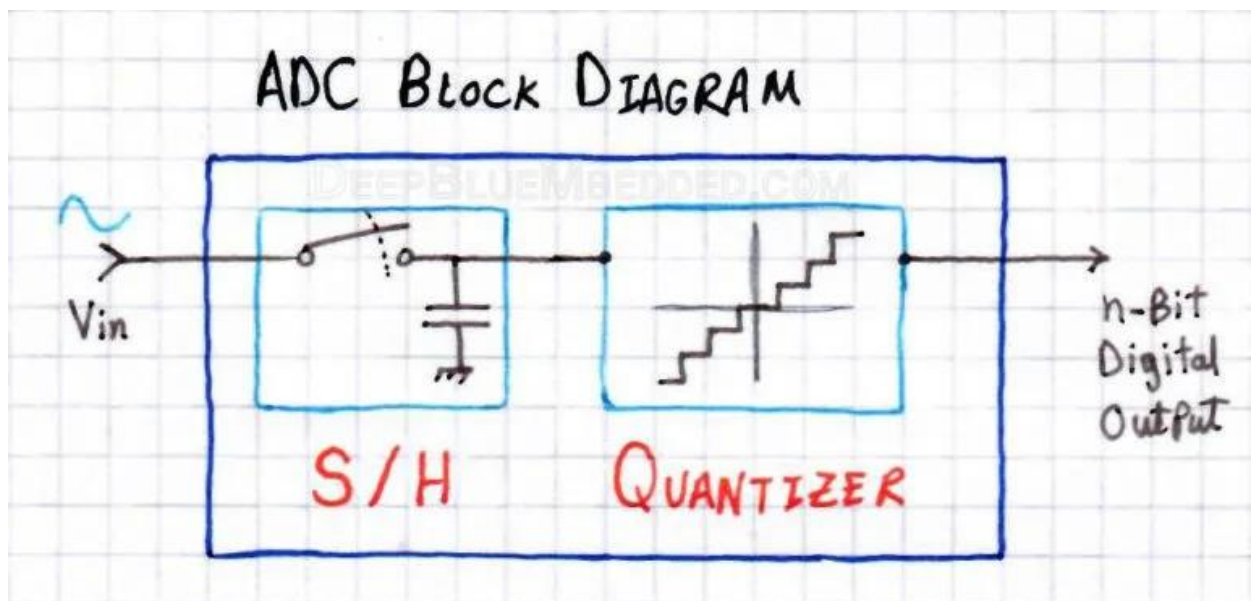


ADC hoạt động như thế nào?

$$(X/V_{ref}) * 1023 =$$

$$(x/5) * 1023$$

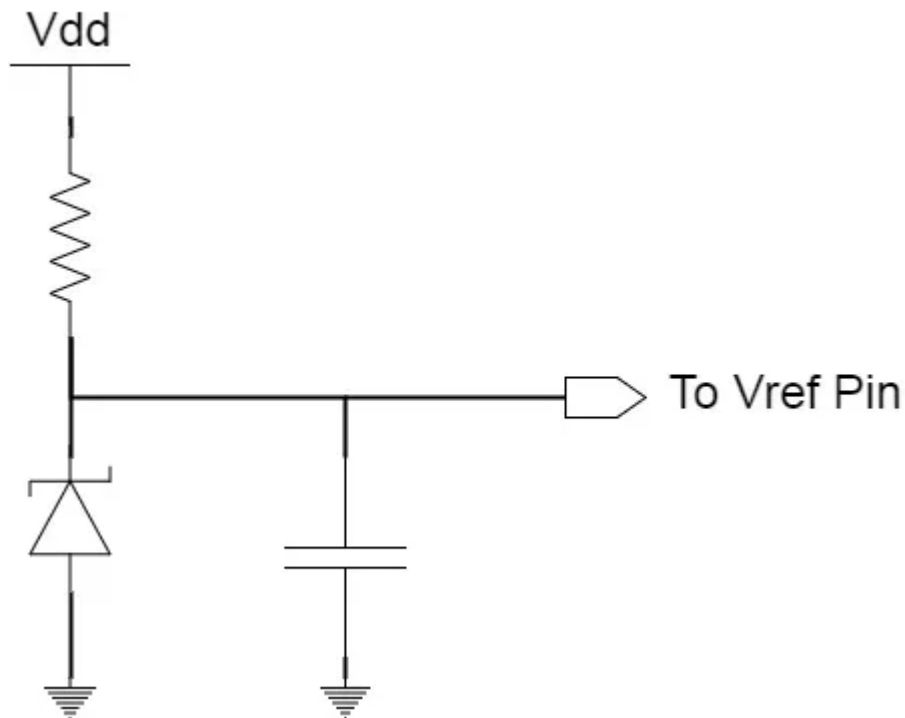
Sơ đồ khối chung cho ADC



Analog Reference For ADC

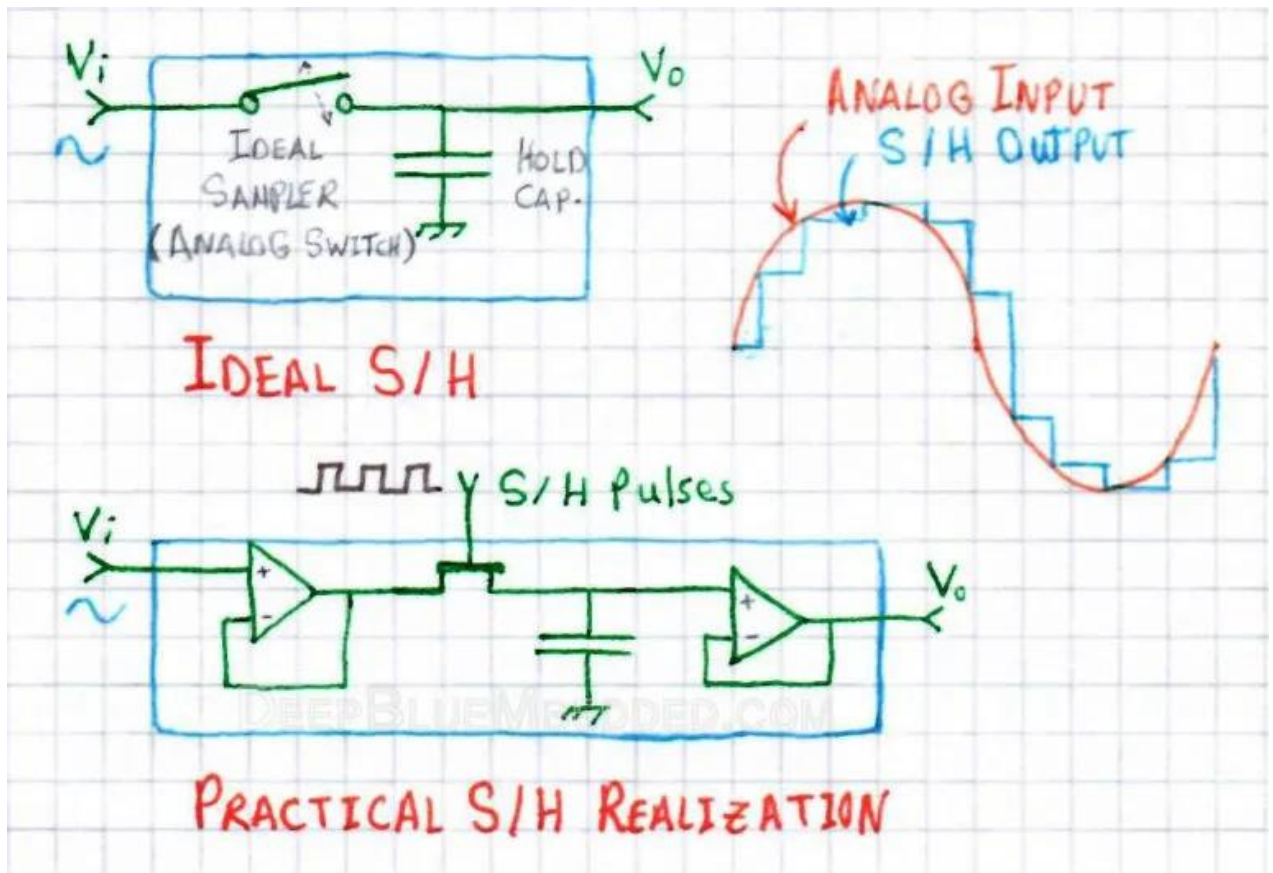
Quá trình chuyển đổi analog sang kỹ thuật số cần một tham chiếu điện áp cực kỳ ổn định V_{Ref+} và V_{Ref-} đại diện cho sự thay đổi điện áp tối đa cho phép cho đầu vào để được chuyển đổi chính xác thành giá trị kỹ thuật số đối với các giới hạn được đặt bởi tham chiếu tương tự V_{Ref} .

Cách dễ nhất để đảm bảo chữ V ổn định V_{Ref} là sử dụng điện trở và tụ điện để chống lại bất kỳ sự sụt giảm điện áp đột ngột nào và bảo vệ hệ thống của bạn. Bạn cũng có thể sử dụng diode Zener để đảm bảo tham chiếu điện áp ổn định giúp miễn nhiễm với hệ thống của bạn trước bất kỳ thay đổi đột ngột nào trong việc tăng / giảm nguồn điện. Khái niệm này được chỉ ra trong sơ đồ dưới đây.



Lấy mẫu ADC

Tín hiệu đầu vào Analog là một hàm liên tục trong thời gian phải được giữ không đổi trong một khoảng thời gian ngắn trong đó ADC có thể chuyển đổi giá trị tức thời thành biểu diễn kỹ thuật số của nó. Quá trình này được thực hiện bởi mạch Mẫu / Giữ về cơ bản là một **Công tắc Tương tự + một Tụ điện Giữ**. Nó có thể trở nên phức tạp hơn trong các bộ chuyển đổi nâng cao nhưng chúng tôi đang giả định mạch S / H rất cơ bản.



Tỷ lệ lấy mẫu

Tốc độ ADC chuyển đổi tín hiệu tương tự liên tục thành dữ liệu kỹ thuật số được gọi là "Tốc độ lấy mẫu". Và nếu T_s mất thời gian để chuyển đổi một mẫu duy nhất, thì tốc độ lấy mẫu của ADC này là $F_s = 1 / T_s$. Sau đó, tín hiệu tương tự ban đầu có thể được sao chép từ các giá trị kỹ thuật số thời gian rời rạc bằng phép nội suy toán học. Độ chính xác trong quy trình này được quyết định bởi hiệu ứng kết hợp của tốc độ lấy mẫu và lỗi lượng tử hóa.

Định lý lấy mẫu

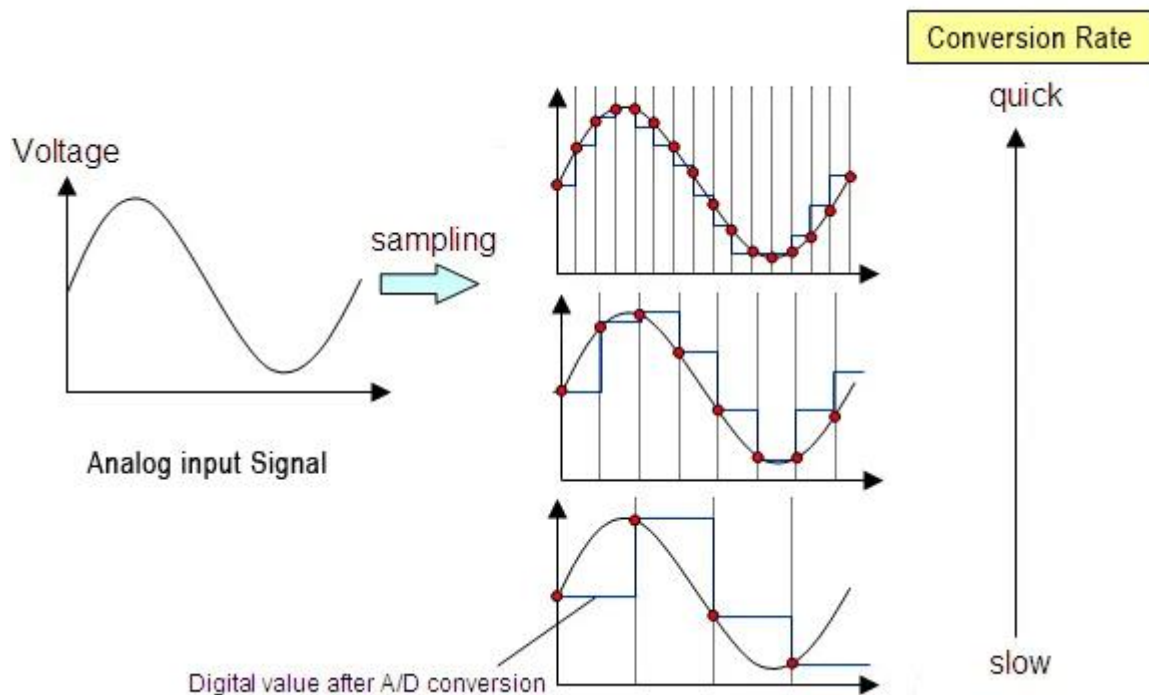
Về mặt lý thuyết, và để có được thông tin tối thiểu về tín hiệu tương tự ban đầu, ADC phải lấy mẫu và chuyển đổi tín hiệu tương tự với tần số $F > = 2F_{\text{Max}}$. Điều này thỏa mãn định lý lấy mẫu Shannon-Nyquist.

F_s -> Tần số lấy mẫu của ADC

F_{Max} -> tần số tối đa trong tín hiệu tương tự đang được chuyển đổi

Vì phạm quy tắc này sẽ dẫn đến các vấn đề tiếp theo và mất thông tin quan trọng về tín hiệu. Gần như không thể có được ý tưởng về tín hiệu ban đầu trông như thế nào trước khi chuyển đổi. Đó là lý do tại sao chúng tôi thực tế chọn

một ADC có tốc độ lấy mẫu (F_s) khoảng $10 \times F_{Max}$ Để tránh hoạt động gần với những hạn chế cơ bản của việc lấy mẫu.



ADC Resolution & Quantization (Độ phân giải ADC & quantization)

Resolution

Độ phân giải của ADC cho biết số lượng giá trị rời rạc mà nó có thể tạo ra trong phạm vi các giá trị tương tự. Độ phân giải cũng xác định độ lớn của lỗi lượng tử hóa và do đó xác định tỷ lệ tín hiệu trên nhiễu trung bình tối đa có thể (SNR) cho một ADC điển hình.

Độ phân giải Q phụ thuộc vào cả số lượng bit n được tạo ra bởi bộ định lượng và cả FSR (phạm vi toàn diện) cho đường dây điện áp tham chiếu tương tự. Dưới đây là công thức tính độ phân giải của lượng tử hóa Q về cơ bản là một phép chia cho điện áp FSR cho số lượng cấp 2^n

$$Resolution(Q) = \frac{FSR}{2^n}$$

Có nghĩa là, một ADC 8-Bit với một tham chiếu tương tự V_{Ref} của 5v có độ phân giải là

$$Resolution(Q) = \frac{5V}{2^8} = 0.195V$$

Đó là lý do tại sao bạn có khả năng tăng độ phân giải chuyển đổi ADC bằng cách giảm tham chiếu điện áp tương tự miễn là bạn đang đo các tín hiệu nhỏ. Ngoài ra, bạn có thể khuếch đại các tín hiệu nhỏ như âm thanh được thu bởi micrô thường nằm trong phạm vi milli-volt.

Mô-đun ADC trong Vi điều khiển PIC

Đối với phần còn lại của hướng dẫn này, chúng ta sẽ thảo luận về Mô-đun ADC trong vi điều khiển PIC16F. Làm thế nào nó hoạt động và làm thế nào để cấu hình và sử dụng nó để đọc các kênh analog cho các dự án của chúng tôi.

Mô tả phần cứng

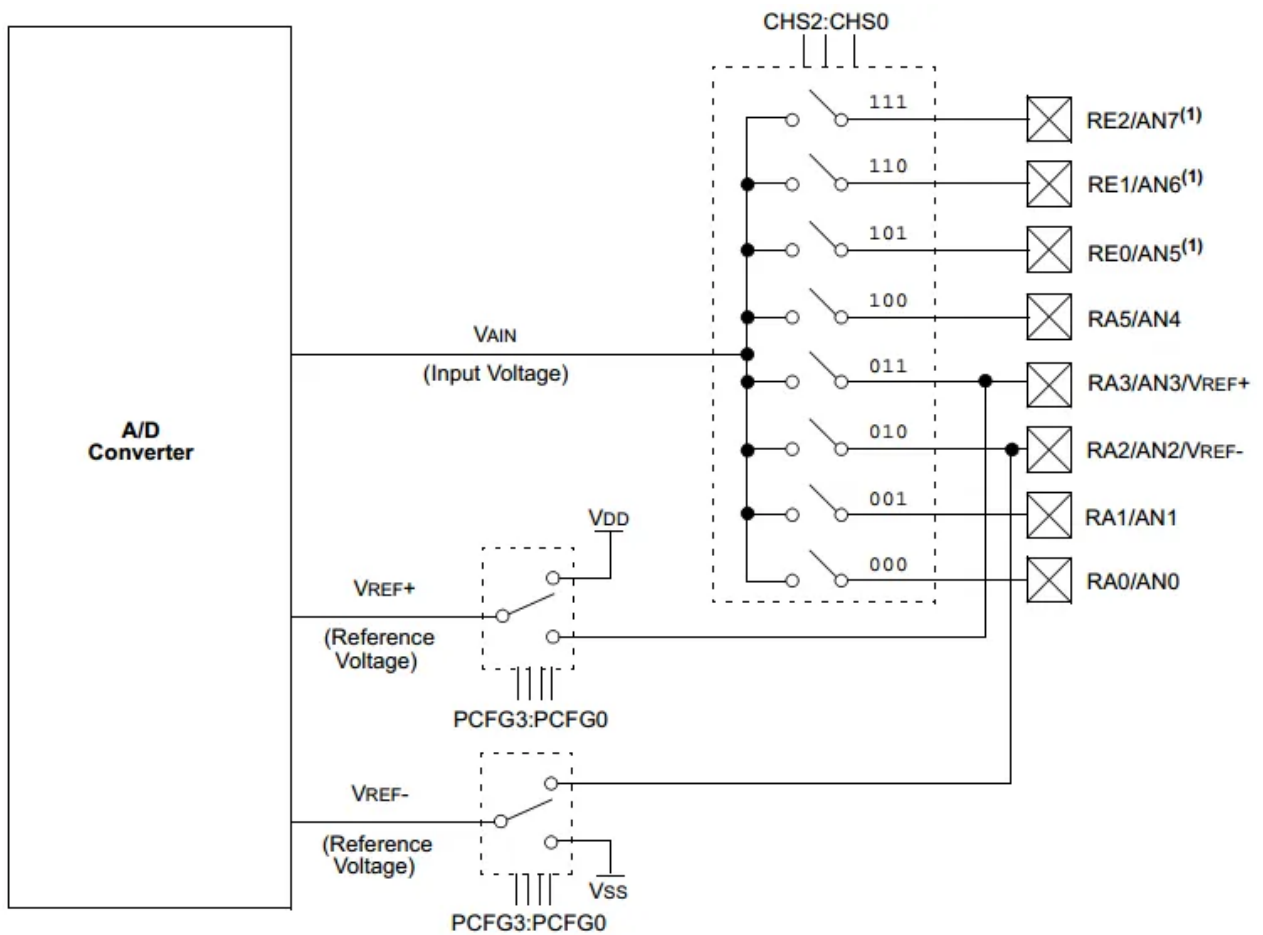
Mô-đun Chuyển đổi Analog-to-Digital (A / D) có 5 kênh cho các thiết bị 28 chân và **8 kênh** cho các thiết bị 40/44 chân. Việc chuyển đổi tín hiệu đầu vào tương tự dẫn đến một số kỹ thuật số **10-Bit** tương ứng.

Mô-đun A / D có đầu vào tham chiếu điện áp cao và thấp là phần mềm có thể lựa chọn cho một số kết hợp của VDD, VSS, RA2 hoặc RA3.

Bộ chuyển đổi A / D có một tính năng độc đáo là có thể hoạt động khi thiết bị ở chế độ Ngủ. Để hoạt động ở chế độ Ngủ, đồng hồ A / D phải được lấy từ bộ dao động RC bên trong của A / D.

Sơ đồ khối

Dưới đây là sơ đồ khối cho bộ chuyển đổi A / D trong bộ vi điều khiển PIC16F877



ADC Special Function Registers

Mô-đun A / D có bốn thanh ghi. Các thanh ghi này là

- A / D Kết quả đăng ký cao (ADRESH)
- A / D Kết quả Đăng ký thấp (ADRESL)
- Đăng ký kiểm soát A / D 0 (ADCON0)
- Thanh ghi kiểm soát A / D 1 (ADCON1)

Kết quả chuyển đổi A/D 10-Bit được lưu trữ trong thanh ghi (ADRESH+ADRESL).

Hoạt động của mô-đun ADC được kiểm soát hoàn toàn bởi các thanh ghi (ADCON0 + ADCON1).

ADCON0

ADCON0 REGISTER (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0**: A/D Conversion Clock Select bits (ADCON0 bits in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-3 **CHS2:CHS0**: Analog Channel Select bits

000 = Channel 0 (AN0)
 001 = Channel 1 (AN1)
 010 = Channel 2 (AN2)
 011 = Channel 3 (AN3)
 100 = Channel 4 (AN4)
 101 = Channel 5 (AN5)
 110 = Channel 6 (AN6)
 111 = Channel 7 (AN7)

Note: The PIC16F873A/876A devices only implement A/D channels 0 through 4; the unimplemented selections are reserved. Do not select any unimplemented channels with these devices.

bit 2 **GO/DONE**: A/D Conversion Status bit

When ADON = 1:

1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)

0 = A/D conversion not in progress

bit 1 **Unimplemented**: Read as '0'

bit 0 **ADON**: A/D On bit

1 = A/D converter module is powered up

0 = A/D converter module is shut-off and consumes no operating current

Bit 7-6: Lựa chọn tần số chuyển đổi của bộ ADC.

bit 7-6 **ADCS1:ADCS0**: A/D Conversion Clock Select bits (ADCON0 bits in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

- Bit 5-3 : Lựa chọn kênh chuyển đổi:

000 = Channel 0 (AN0)
001 = Channel 1 (AN1)
010 = Channel 2 (AN2)
011 = Channel 3 (AN3)
100 = Channel 4 (AN4)
101 = Channel 5 (AN5)
110 = Channel 6 (AN6)
111 = Channel 7 (AN7)

- Bit 2: GO/DONE.

Khi bộ ADC bắt đầu chuyển đổi, bit này ở mức 1, khi quá trình chuyển đổi hoàn thành bit này sẽ về mức 0 báo quá trình chuyển đổi hoàn tất

- Bit 1: ADON : Bit cho phép bộ ADC hoạt động

ADON = 1: Bộ ADC được làm việc

ADON = 0: Tắt bộ ADC

ADCON1

11-2: ADCON1 REGISTER (ADDRESS 9Fh)

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM**: A/D Result Format Select bit

1 = Right justified. Six (6) Most Significant bits of ADRESH are read as '0'.

0 = Left justified. Six (6) Least Significant bits of ADRESL are read as '0'.

bit 6 **ADCS2**: A/D Conversion Clock Select bit (ADCON1 bits in shaded area and in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-4 **Unimplemented**: Read as '0'

bit 3-0 **PCFG3:PCFG0**: A/D Port Configuration Control bits

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O

C/R = # of analog input channels/# of A/D voltage references

- Bit 7 : ADFM:

Do có 2 thanh ghi chứa giá trị ADC sau khi chuyển đổi, nhưng bộ ADC là ADC10bit nên sẽ có 6bit chứa giá trị '0', việc xác định 6bit đó ở bên phía thanh ghi ADRESH hay ADRESL do bit ADFM này quyết định

ADFM = 1 : 6bit này là 6bit cuối của thanh ghi ADRESH

ADFM = 0: 6bit này là 6bit đầu tiên của thanh ghi ADRESL

- Bit 6 : loại chọn tuần số tác động

bit 6 **ADCS2**: A/D Conversion Clock Select bit (ADCON1 bits in shaded area and in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

- Bit 5-4: không sử dụng
- Bit 3-0: 4 bit này có chức năng cấu hình chức năng của từng chân trong bộ ADC (Mình dịch theo mình hiểu, cũng k được sát với datasheet).

Mô-đun A / D có đầu vào tham chiếu điện áp cao và thấp là phần mềm có thể lựa chọn cho một số kết hợp của VDD, VSS, RA2 hoặc RA3. Bạn có thể chọn các tham chiếu điện áp theo cách tương tự như lựa chọn kênh tương tự bằng cách sử dụng bảng trên.

Hãy giả sử một $V_{\text{Ref}+}$ của $V_{\text{Dd}} = 5\text{v}$ và $V_{\text{Ref}-}$ của $V_{\text{Ss}} = 0\text{v}$ cho hướng dẫn này. Điều này dẫn đến phạm vi toàn diện (FSR) là 5v. Do đó, độ phân giải chuyển đổi A / D sẽ là

Bắt đầu chuyển đổi A / D

Đặt bit **GO /DONE** sẽ bắt đầu chuyển đổi mới cho kênh đã chọn. Một chuyển đổi mới sẽ không bắt đầu trước Thời gian điều tra **T** đã trôi qua.

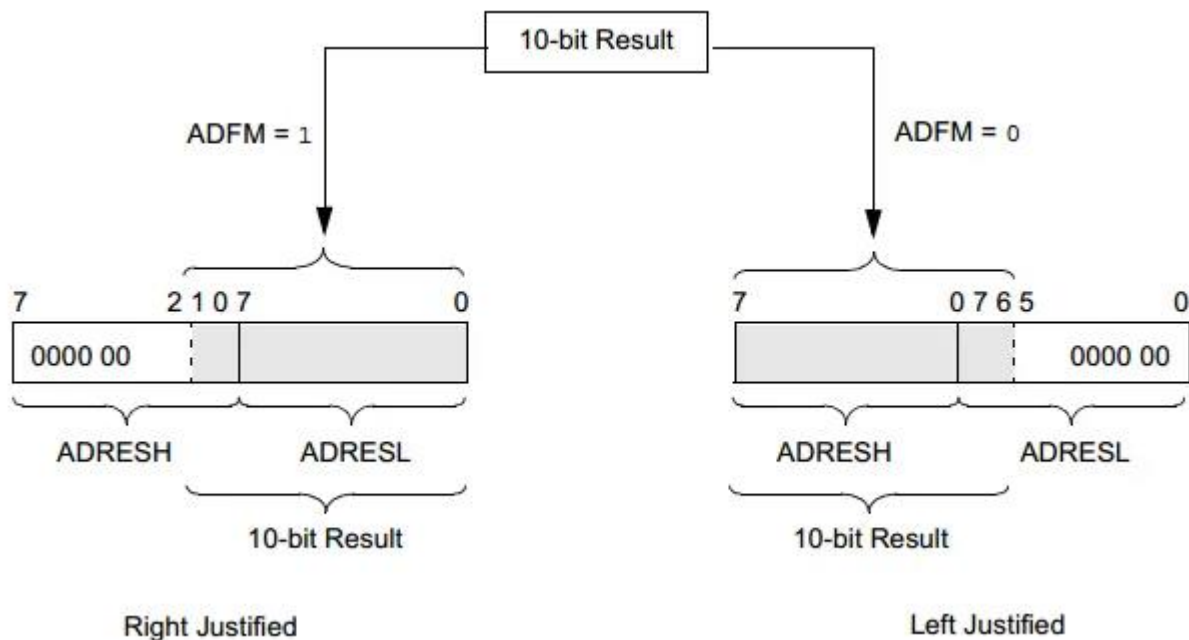
Xóa bit GO /DONE trong quá trình chuyển đổi sẽ hủy bỏ chuyển đổi hiện tại. Cặp thanh ghi kết quả A/D sẽ KHÔNG được cập nhật với mẫu chuyển đổi A/D đã hoàn thành một phần. Nghĩa là, các thanh ghi (ADRESH: ADRESL) sẽ tiếp tục chứa giá trị của chuyển đổi đã hoàn thành cuối cùng (hoặc giá trị cuối cùng được ghi vào các thanh ghi ADRESH: ADRESL). Sau khi hủy chuyển đổi A/D, quy trình chuyển đổi tiếp theo trên kênh đã chọn sẽ tự động được bắt đầu. Bit GO / DONE sau đó có thể được đặt để bắt đầu chuyển đổi.

Hình dưới đây cho biết các chu kỳ chuyển đổi A / D cần thiết để chuyển đổi tín hiệu tương tự kể từ khi đặt bit GO / DONE cho đến khi kết quả được ghi vào các thanh ghi ADRES. Cũng lưu ý rằng sau khi bit GO được đặt, lần đầu tiên phân khúc có tối thiểu T_{CY} và tối đa là T

A/D Conversion Result Registers

Cặp thanh ghi (ADRESH: ADRESL) là vị trí mà kết quả A / D 10-bit được tải khi hoàn thành chuyển đổi A / D. Cặp thanh ghi này rộng 16 bit. Mô-đun A / D cho phép linh hoạt sang trái hoặc phải biện minh cho kết quả 10 bit trong thanh ghi kết quả 16 bit.

Bit chọn định dạng A / D (ADFM) kiểm soát lý do này. Hình dưới đây cho thấy hoạt động của biện minh kết quả A / D. Trong đó các bit thừa được tải bằng '0'



Cấu hình ADC với PIC MCU

Các bước để định cấu hình mô-đun ADC

Để thực hiện chuyển đổi A / D, hãy làm theo các bước sau:

1. Định cấu hình mô-đun A / D

- Định cấu hình các chân analog / tham chiếu điện áp và I / O kỹ thuật số (ADCON1)

- Chọn kênh đầu vào A / D (ADCON0)
- Chọn đồng hồ chuyển đổi A / D (ADCON0)
- Bật mô-đun A / D (ADCON0)

2. Cấu hình ngắt A / D (nếu muốn):

- Xóa bit
- Đặt bit ADIE
- Đặt bit PEIE
- Đặt bit GIE

3. Chờ thời gian cần thiết.

4. Bắt đầu chuyển đổi

- Đặt bit GO / DONE (ADCON0)

5. Chờ quá trình chuyển đổi A / D hoàn tất bằng một trong hai cách sau:

- Ha bit cho bit GO / DONE sẽ bị xóa (ngắt bị vô hiệu hóa);

HOẶC Chờ ngắt A / D

6. Đọc cặp đăng ký kết quả A / D (ADRESH: ADRESL), xóa bit ADIF nếu được yêu cầu.

7. Đối với lần chuyển đổi tiếp theo, hãy chuyển sang bước 1 hoặc bước 2 theo yêu cầu. Thời gian chuyển đổi A / D trên mỗi bit được định nghĩa là T

Code ví dụ

```
#include <stdio.h>
#include <stdlib.h>
#define _XTAL_FREQ 20000000
#define RS RD2
#define EN RD3
#define D4 RD4
#define D5 RD5
#define D6 RD6
#define D7 RD7
#include <xc.h>
#include "lcd.h"
// BEGIN CONFIG
#pragma config FOSC = HS // Oscillator Selection bits (HS
oscillator)
#pragma config WDTE = OFF // Watchdog Timer Enable bit
(WDT enabled)
#pragma config PWRT = OFF // Power-up Timer Enable bit
(PWRT disabled)
#pragma config BOREN = ON // Brown-out Reset Enable bit
(BOR enabled)
#pragma config LVP = OFF // Low-Voltage (Single-Supply) In-
Circuit Serial Programming Enable bit (RB3 is digital I/O, HV on
MCLR must be used for programming)
#pragma config CPD = OFF // Data EEPROM Memory Code
Protection bit (Data EEPROM code protection off)
```

```
#pragma config WRT = OFF // Flash Program Memory Write  
Enable bits (Write protection off; all program memory may be  
written to by EECON control)
```

```
#pragma config CP = OFF // Flash Program Memory Code  
Protection bit (Code protection off)
```

```
//END CONFIG
```

```
int hour,top;
```

```
void ADC_Init()
```

```
{  
    ADCON0 = 0x41; // 01000001  
    ADCON1 = 0xC0; // 11000000  
}
```

```
unsigned int ADC_Read(unsigned char channel)
```

```
{  
    if(channel > 7)  
        return 0;  
  
    //ADCON0 &= 0xC5; // 1100 0101  
    ADCON0 |= channel<<3;  
    __delay_ms(2);  
    GO_nDONE = 1;  
    while(GO_nDONE);  
    return ((ADRESH<<8)+ADRESL);  
}
```

```
void main(void)
```

```
{
```

```

ADC_Init();
    char s[1];

    TRISD = 0X00;
    TRISC = 0X00;
    Lcd_Init();
    Lcd_Clear();
    int i,a;
    TRISCbits.TRISC0 = 0;
    while(1)
    {

        hour = (ADC_Read(0))/10; //x = 5*read_ADC()/1023;
        if(hour >99)
            hour = 99;

        RC0 = 1;
        if(hour < 40)
            RC0 = 1;
        if(hour > 40)
            RC0 = 0;

        if(hour < 100)
            s;
        Lcd_Set_Cursor(1,1);
    }

```

```

    Lcd_Write_String("Muc nuoc");
    sprintf(s,"%d",(hour));
    Lcd_Set_Cursor(1,10);
    Lcd_Write_String(s);
    Lcd_Set_Cursor(1,13);
    Lcd_Write_Char(37);

//

}

}

```

