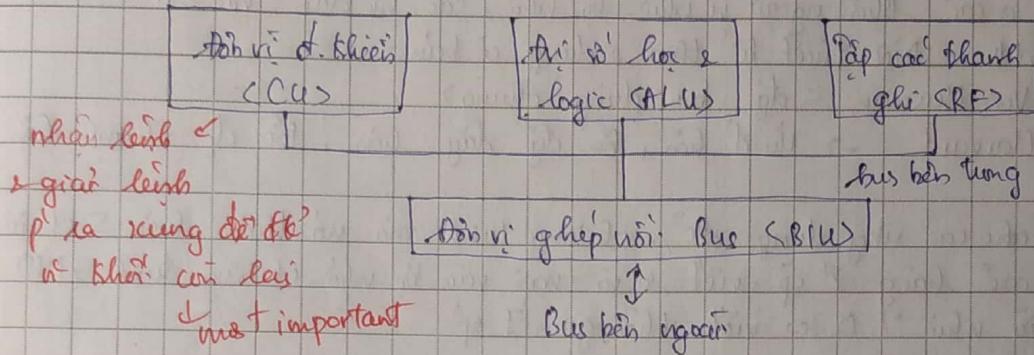


VI XỬ LÝ VÀ CẤU TRÚC MÁY TÍNH

S1. Tổng quan về vi xử lý và cấu trúc máy tính

1) Vi xử lý.

- Chức năng + Điều khiển hoạt động của máy tính
+ Xử lý dữ liệu.
- Nguyên tắc cơ bản: thực hiện theo chương trình nằm trong bộ nhớ chính.
- Cấu trúc cơ bản của Vi xử lý.



- Cấu thành phần cơ bản của Vi xử lý.

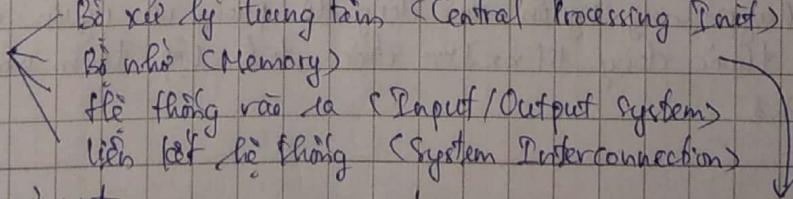
- + Đơn vị điều khiển (Control Unit - CU): điều khiển hoạt động của máy tính theo chương trình để thực hiện
- + Đơn vị logic (Arithmetic and logic Unit - ALU) thực hiện các phép toán và học và các phép toán logic trên các dữ liệu có thể
- + Tập thành phần ghi (Register File - RF): lưu trữ các thông tin tạm thời phục vụ cho hoạt động của CPU.
- + Đơn vị gộp nối: Bus (Bus Interface Unit - BIU): kết nối và trao đổi thông tin giữa bus bên trong (Internal Bus) và Bus bên ngoài (External Bus)

- Tốc độ của Vi xử lý.

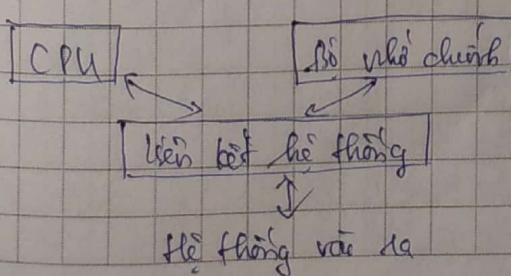
- + Tốc độ của VSL: số lệnh được thực hiện/s (MIPS - Millions of Instructions per second) → MHz: triệu Hz
- + Số lệnh được ghi chép là số
- + Tần số xung nhịp của bộ xử lý
- + Hoạt động theo s xung nhịp (clock) có độ chính xác và bắt kín
- + Độ chính xác VSL sẽ đánh giá qua thời gian tiếp thông qua tần số xung nhịp

2) Cấu trúc máy tính.

- Cấu thành phần của máy tính



- Cấu trúc cơ bản của máy tính



Bộ nhớ Cache	RAM	ROM
↓	Chia sẻ	chia sẻ
lưu trữ thông tin (bộ nhớ)	lập	không
đem	đem	đem
→ mực	đem	đem

- Các thành phần có basic của máy tính.
 - + Bộ xử lý trung tâm: Điều khiển hoạt động của tất cả các bộ phận
 - + Bộ nhớ chính: Chứa các chương trình và dữ liệu đang được sử dụng
 - + Hệ thống vào ra: Tạo điều thông tin giữa nó và ngoài
 - + Kênh kết nối: Kết nối và vận chuyển thông tin giữa các bộ phận với nhau

3) Vi điều khiển và ứng dụng

a) Vi điều khiển

- Là một máy tính nhỏ tích hợp trên 1 chip
- Là một hệ thống bao gồm một vi xử lý có hiệu suất để dùng và già thành thấp kết hợp với các khía cạnh ngoại vi như bộ nhớ, các nút điều khiển
- Kiến trúc của VLSI: Thời gian là theo dạng có bước:
 - + Kiến trúc Von Neumann < dg bus dùng chung
 - + Kiến trúc Harvard → Ưu điểm: Bộ dg dày dặn, tối ưu về thời gian

b) Ứng dụng

- Thời gian (c) cho các ứng dụng dtc, chương trình mà nó thực hiện & chương trình dtc) để xác hàng ↑ số viết và nộp vào bộ nhớ chương trình (ROM)
- Ưu điểm + Tính linh hoạt
 - + Công suất tiêu thụ thấp
 - + Thiết kế gọn nhẹ
 - + Độ bền cao
- Nhược điểm: Tối ưu thấp

4. Các Vật lý thông dụng

5. Giới thiệu PIC

- PIC: Programmable Intelligent Computer: Máy tính nhỏ không có bộ nhớ trung tâm
 - [Peripheral Interface]
 - Peripheral Interface Controller: Bộ dtc giao tiếp ngoài vi
 - là dòng vật liệu 8 bit của Microchip
- Đặc điểm:
 - + Thiết kế theo kiến trúc kiến trúc Harvard
 - + Bộ nhớ bus chỉ 8 bit
 - + Nội bộ: Lệnh được thực hiện & 1 chu kỳ may
 - + Bộ nhớ bus giữa CPU và bộ nhớ chương trình có thể là 12, 14 or 16 bit
- PIC 18 chế tạo từ Microchip nhanh tốc độ thực hiện là 10MIPS (10 tr/second/s)
- Một bộ súng thêm 1 số lệnh nhanh như:
 - + Có thể hoạt động ở chế độ ngủ (sleep mode) và dừng điện tiêu thụ 0.2 mA
 - + Có thể chuyển sang nguồn dự phòng nếu nguồn chính bị mất
 - + Các chip thuộc dòng 18FxxJxx có thể tích hợp thêm module Ethernet chuẩn IEEE 802.3

S2. Kiến trúc phần cứng PIC18F4520

1) Khởi quật và PIC18F4520

- Khởi CPU
- Khởi bộ nhớ
- Khởi tần số dòng
- Khởi Reset
- Khởi GPIO

- Khởi Timer
- Khởi PWM
- Khởi USART, MSSP
- Khởi nguồn

* Khởi VREF: můn hổn chiai cài ghiệp nôi. Või khô.

- Khởi Reset
- Khởi tần số dòng
- Khởi nguồn

2) Moot dòng Reset

1) Reset = lệnh Reset

2) Reset do đầy ngăn klop (stack full reset)

3) Reset do rỗng ngăn klop (stack underflow Reset)

4) Reset từ chân MCLR: bùi tinh hổi binh thig

5) Reset do WatchDog Timer (WDT)

6) Reset do bắt nguồn POR (Brown on Reset) (Power on Reset)

7) Reset do sút điện áp nguồn POR (Brown on Reset)

8) Bộ tao dòng (f = 40MHz)

- Nguồn xung từ bên ngoài - thay đổi tần số công hổing - bên ngoài: LP, VT, HS

RC, RCIO, RC, BCIO, HSPLL

- Nguồn xung từ chân MOSI & MISO

- Nguồn xung từ bộ tao dòng nội: INOSC 8MHz : INT01 & INT02

- Nguồn xung từ bộ tao dòng nội: INTRC 512Hz

- HS

Còn thiết lập giao với thành phần CONFIG1H

→ Thiết lập cấu hình OSC = HS;

pragma config OSC = HS

* Các công GPIO và các thành phần liên quan

- Chuẩn dòng và bộ tao PIC18F4520: có 3 chuẩn

+ PDIP 40 chân → thường dùng nhất

+ TQFP 44 chân

+ QFN 44 chân

- Có 5 cổng (PORT)

PORT A RAO → RA7

PORT B RB0 → RB7

PORT C RC0 → RC7

PORT D RD0 → RD7

PORT E RB0 → RB3

56 chân vòi ra

- 1 chân GPIO.

+ 2 chân nòng } Xuất dữ liệu
Nhập dữ liệu

+ Chứa 16 bit liệu } Thiết lập WR TRIS = 0 ; WR PORT = 0
Xuất dữ liệu LAT, đọc PORT

+ Chứa vào dữ liệu { Thiết lập RD TRIS = 1 ; RD PORT = 1;
 Nhập dữ liệu PORT

* Phát động vào các thành ghi liên quan

- Thành ghi TRIS sẽ thiết lập hướng dữ liệu

 TRIS Bit = 0 // Chân GPIO có cùa

 TRIS Bit = 1 // Chân GPIO có cùa vào

- Thành ghi PORT sẽ xuất or nhập dữ liệu

- Thành ghi LAT sẽ thiết lập (chốt dữ liệu ra) → cho vào, gửi (hoặc) 1
 trạng thái

Thực tế: dtc công P10 của mikro PIC18F4520 cần thiết lập

+ Thành ghi TRIS x chọn hướng dữ liệu

+ Thành ghi PORT x xuất nhập dữ liệu

VD: Muốn dtc PORT cần thiết lập TRISA, PORTA

* Lập trình: mô phỏng ghi/đọc theo byte, bit

(Huong chuong trinh):

- Khai báo thư viện định danh p1c18f4520.h // include <p18f4520.h>

- Cài luentk VDD

+ Sd bộ clau dong chay do fcs

+ Sd chien RF3 lam chien Reset

+ % dùng Watchdog Timer

+ % sd chay do nguon cap tu mạch lap ICSP // pragma config LVP = OFF

- Khai báo cac bien, cau truc, hieu do, chuong trinh con

int x; char m[50];

void high_isr(void)

{

 // Cac cau lenh

void main()

{ // Cac cau lenh }

Note: Lập trình P10 dữ liệu chs cac PORT A1B1B2 co chinh cau chien ANX (lê
dau vao Nguoi ti) can thiết lập cau lenh ADCON1 = 0x0f
 de cac chien ANX la dauлас
 tinh nguyen chien VAD = cau lenh \$ define.

§3. Ngắt ngoại

Đoàn all

1) khai quát về ngắt.

- khái niệm: Là tạm thời dừng công việc hiện tại & chuyển sang thực hiện 1 nhiệm vụ khác (thí: cấp thiết, quan trọng hơn), sau đó lại quay lại thực hiện tiếp công việc cũ (dang thực hiện đó)
- Ví dụ:

khi đang thực hiện cái lệnh nào đó trong chương trình chính \rightarrow tin hiệu ngoại \rightarrow dừng chương trình chính \rightarrow lui lại cái thành ghi vào ngăn xếp \rightarrow thực hiện cái lệnh c) chương trình con phục vụ ngoại \rightarrow về chương trình chính, lại lại ghi lại cái thành gì \rightarrow thực hiện tiếp cái lệnh của chương trình chính

2) Ngắt ngoại trên pic 18f4520

- Vector ngoại: có 2 vector ngoại + nhau (CON08: ngoại vị trí cao, CON08: vị trí thấp)
- Có 15 nguồn ngoại + nhau

+ Ngắt trong (Internal Interrupt): Nguyên nhân gây ra ngoại do tạo từ bên trong Või Või Tiên Tumors, bị biến đổi không tự -> (ADC) biến đổi xong.

+ Ngắt ngoại: (External Interrupt): Nghĩa là ngoại do tạo bởi ra từ mạch điện bên ngoài Või < C6 PIC18 có 3 nguồn ngoại: INT0, INT1 & INT2) \rightarrow gây ra ngoại, mạch điện cần có mạch điện bên ngoài tạo ra sự chuyển mức logic từ 0 \rightarrow 1 or chỉ mức logic từ 1 \rightarrow 0 chia đều thành RB0/INT0; RB1/INT1; RB2/INT2

3) Quá trình

a) Các bit điều khiển ngoại ngoại

- O chì để mở đóng, 1 nguồn ngoại sẽ đóng mức vị trí cao. Ta cần khởi tạo + GIE = 1: cho phép ngoại toàn cục (Global interrupt Enable)
- + INTxIE = 1 ($x=0, 1$ or 2) cho phép nguồn ngoại x.
- + INTxEN = 0 or INTxEDGx = 1.

Khi xảy ra ngoại, bit có nguồn ngoại đóng bằng (INTxIF) & được đặt (set) ≥ 1

Locbit	Nguyên ngoại	INT0	INT1	INT2
Cho phép ngoại toàn cục		GIE - GIEH		
Cho phép nguồn ngoại		INT0IE	INT1IE	INT2IE
	Thanh ghi	Thanh ghi	Thanh ghi	
	INTCON	INTCON8	INTCON8	
Ngắt =единим (C=0)	INTEDG0	INTEDG1	INTEDG2	
or единим (C=1)	Thanh ghi	Thanh ghi	Thanh ghi	
	INTCON2	INTCON2	INTCON2	
Có ngoại	INT0IF	INT1IF	INT2IF	
	Thanh ghi	Thanh ghi	Thanh ghi	
	INTCON	INTCON3	INTCON3	

5) Các bước lập trình sử dụng ngoại

- Viết khung chương trình có sử dụng
- Viết các lệnh của chương trình
- + Khởi tạo các PORT (ADCON1, TRIS)

- + Set bit ngắt toàn cục INTCON bits. GIE - GIEH = 1;
- + Set bit cho phép nguồn ngắt Working wing INTCON bits. INT0IE or INTCON3 bits. INT1IE, or INTCON3 bits. INT2IE;
- + Chọn nguồn cung cấp (dòng) = xod / set bit Working wing
INTCON bits. INT0GO = 0 ;
or INTCON2 bits. INT1GO = 0 ;
or INTCON2 bits. INT2GO = 0 ;
- + Cài lệnh để theo dõi bài
- > Viết cài lệnh của chính can phuc vi ngat
+ Khi có ngắt Working wing INTCON bits. INT0IF = 0 ;
or INTCON3 bits. INT1IF = 0 ;
or INTCON3 bits. INT2IF = 0 ;
- + Cài lệnh theo dõi bài

§4 Cài bộ đếm thời (đếm)

- * Cài tạo & het của các bộ timer
 - Cài tạo các bộ Timer
 - + Một bộ Timer n-bit chia tạo thành bởi n flip-flop mà mỗi flip-flop
 - + Mỗi bước xung chia cho flip-flop đầu tiên
 - + Sau đó mỗi n flip-flop là một bước xung giao tiếp thời cài n bit bộ đếm
 - Hoạt động của Timer
 - Bộ Timer sẽ đếm tiến ou lùi (18F4520 chỉ có bộ đếm tiến) khi nhận xung, mức tiến này có nguồn xung clock đầu vào (nút thường or invers)
- * TIMER 0
 - Timer 0 PIC18F4520
 - + Vì điều khiển PIC18F4520 có 4 bộ Timer,het & chế độ đếm + định thời
 - + Timer 0 có thể het & chế độ 8 bit or 16 bit, định thời or chế độ đếm pha thời vẫn và cài hỉnh = phần mềm
- * Các bước lập trình đc' P18F4520
 - Viết khung chương trình theo định dạng
 - Viết cài lệnh của chương trình chính
 - + Khởi tạo các PORT (ADCON1, TRIS)
 - + Khởi tạo thành ghi TOCON
 - + Nạp giá trị ban đầu cho các thành ghi TMROH & TMROL chip wing y/c đc bài
 - + Set bit TMROON (en & đc' TIMER, bắt đầu đếm, bắt đầu y/c áo đc bài)

$$F_{\text{timer clock}} = F_{\text{osc}} / 4 | \text{Prescale Value}$$

$$\text{timer clock} = \frac{1}{F_{\text{osc}} \cdot \text{Prescale Value}}$$

đi qua 16 lần
= 2¹⁶
9 qua 8 lần
= 1

$$\text{Delay} = \text{Timer Count} * \frac{\text{Prescale Value}}{\text{Fosc}/4}$$

$$\text{Timer Count} = \frac{\text{Delay} * \text{Fosc}/4}{\text{Prescale Value}}$$

$$\text{Timer Reg Init} = \text{Timer Max Value} - \text{Timer Count} + 1$$

65535 | 16bit

256 | 8bit

giao diện nạp =

Fosc

TMR0H =

(65536 - Timer counter) / 256

256.

2. PrescalerValue. Fcài tạo

TMR0L = (65536 - Timer counter) % 256

§5. Ngắt Timer

1) Khái quát về ngắt Timer

- Khiêm: Ngắt Timer là sự kiện qua toàn chuỗi trình chính để chuyển sang + hoặc hiện chương trình con phục vụ ngắt mới. Khi có báo ngắt (TMRxIF) của timer sẽ set bit "1".

- Lý do: Để thực hiện 1 tác vụ nào đó trước đó, kíp lấp kín sau các khoảng thời gian xác định. Ngắt thường giúp đỡ thực hiện tác vụ này vì nó có thể thực hiện nhiều tác vụ ở chuỗi trình chính

2) Các bit điều khiển ngắt Timer

Mỗi timer PIC18F4520 có 3 bit để ngắt gồm:

- bit cho phép ngắt
- bit vào tiền ngắt
- bit có ngắt (có báo tràn)

Timer

0

Bit cho phép ngắt

TMR0IE

^{0: cài}
_{1: ngắt}

INTCON<5>

Bit vào tiền ngắt

TMR0IP

^{0: kíp}
_{1: chờ}

INTCON<2>

Có ngắt

TMR0IF

^{0: không ngắt}
_{1: có ngắt}

INTCON<2>

1

TMR1IE

PIE1<0>

TMR1IP

PIR1<0>

TMR1IF

PIR1<0>

2

TMR2IE

PIE1<1>

TMR2IP

PIR1<1>

TMR2IF

PIR1<1>

3

TMR3IE

PIE2<0>

TMR3IP

PIR2<0>

TMR3IF

PIR2<0>

3. Các bước để ngắt Timer

- Khởi tạo các PORT (ARDOND, TRIS)

- Khởi tạo thành ghi đc' Timer: chế độ hoạt động, nguồn xung clock, biến số chia tần

- Set bit cho phép ngắt tràn của

- Lên chương trình ngắt

- Set + bit cho phép nguồn ngắt thường

- Thiết lập giao diện cho thành ghi chưa só' đếm của Timer

- Các lệnh để bắt

→ Để bắt Timer, ta chương trình con phục vụ ngắt ta cần chú ý các điều sau:

- Dùng Timer: TMRxON = 0; (kết thúc dừng or% theo vòng lặp ai kíp)

- Nạp lại giao diện cho thành ghi chưa só' đếm

- Xóa có ngắt: TMRxIF = 0

- Bắt Timer: TMRxON = 1; (chạy lệnh này nếu có dùng lệnh dừng Timer)

- Các lệnh theo dõi

8.6. Truyền thông nối tiếp qua USART

1. Tổng quan về truyền thông nt

- Khiêm: là p̄ truyền tin trong đó các bit mang thông tin để truyền kẽ' tiếp nhau trên một đường dây vật lý. Tại một thời điểm trên bến truyền cũng như bến nhận chỉ có thể truyền / nhận các 1 bit
- So sánh truyền tin song² thu:
 - + Về ct': Truyền các 2 khoảng cách xa
 - Nếu kinh chờing truyền dài
 - Có viền khung
- ít Nhì ct': Bởi độ truyền chậm hơn, phải tách nhau do các thiết bị pha phải có cơ chế có khả năng chuyển đổi nt sang song², song² sang nt khi số pha thay đổi trước đó. Thông tin
- Có 3 pha thiết truyền tin nt:
 - + Pha thứ nhất khung: Các byte chứa thông tin sẽ truyền liên tiếp từ vị trí đầu tiên của khung và chia thành các khung = bit đồng bộ (Sync)
 - + Pha thứ 2 khung: Các byte chứa các frame từ khung với một khung. Một khung có bit đầu = 1 bit Start, tiếp theo là các bit thông tin. kẽ' tiếp là bit ktra chẵn lẻ & kết thúc là bit Stop.
 - + Pha thứ 3: Tiếp theo pha thứ 2 là khung của xphg thiết trên

2. Module BUSESART trên pic18f4520

- Module USART : RCG1/R67 < Truyền (nhận)
- Số USART cần cài: Lệnh các bit sau: bit SPBRN < RCSTA < 7 > = 1
bit TRISC < 7 > = 1
bit TRISC < 6 > = 0

- Phản ghi trạng thái truyền & đk' (RXSTA)
- Phản ghi đk' trạng thái nhận (RCSTA)
- Phản ghi đk' tốc độ baud (BAUDCON)

Tối đa BAUD

- + bit BRG 16 (BAUDCON < 1 >) lựa chọn tối đa baud là chỉ có 8bit hay 16bit
- + Cúp thanh ghi SPBRGH: SPBRG : để đk' khung riêng tối đa baud
- Bảng lựa chọn chỉ có 16 là công thức tính tối đa baud

Cúp hànhan cat bit Chế độ BRG / EUSART

SYN	BRG16	BRGH
0	0	0
0	0	1
0	1	0
0	1	1
1	0	X
1	1	X

8 bit / đồng hồ
16

Công thức tính tối đa baud

$$\left\{ \begin{array}{l} FOSC = [64(n+1)] \\ FOSC = [16(n+1)] \\ \cancel{FOSC = 16(n+1)} \\ FOSC = [4(n+1)] \end{array} \right.$$

X: giá trị bắt buộc

n: cúp ghi của cúp thanh ghi SPBRGH: SPBRG

- VD. Fosc = 16 MHz, tốc độ baud = 9600
chỉ đk' 8 bit BRG :

- Tốc độ mong muốn = $FOSC / [64 (còn + 1)]$

$$\Rightarrow n = \frac{FOSC}{\text{tốc độ mong muốn} \times 64} - 1$$

$$= \frac{16 \cdot 10^6}{9600 \cdot 64} - 1 = 25,042 \approx 25$$

Delay 10 KTCY x (n+1)

$$\Rightarrow \text{tốc độ} \approx \frac{x \cdot 10,000 \times 0,5ms}{10,000 \cdot 0,5ms} = x$$

- Tốc độ tinh toán thực tế: $= 16 \cdot 10^6 / [64 (còn + 1)] = 26,15$

$$- Suy su: \frac{9615 - 9600}{9600} = 0,16\%$$

3. Một số hàm C) thư viện USART.h

Busy USART: Hỗn hoảng của port truyền nt

Close USART:關 USART

Open USART: Cài định PORT nt

puts USART: Truyền một maing ra PORT USART

Read USART: Đọc 1 byte từ PORT USART

Write USART: Truyền 1 byte từ PORT USART

§7. Biến đổi tần số - số (ADC)

1) Giải thích: biến đổi tần số - số (ADC)

- Kh: ADC (Analog to Digital Converter), là một thiết bị vào tần số và pha (t) sẽ sinh ra mã đầu ra dạng số

+ Đầu vào tần số V_A

+ Số bit đầu ra số n

- Hạt động: + Đầu vào tần số V_A

+ lệnh Start: ra lệnh bắt đầu bắt

+ BCO: tinh hieu bat chuyen doi; reong

+ V_A > V_{ref}: dùng chuyen doi

+ Xung Clock: reong nhip chuyen doi ADC

- Các thông số của bộ ADC

+ Độ phân giải: độ bắc chí = số bit của tín hiệu số đầu ra, độ phân giải càng nhiều sai số càng nhỏ, độ chính xác càng cao. Thông thường số bit đầu ra là 8, 10, 12, 16, 32 bit...

+ Điện áp tham chiếu: là giao diện áp số để so sánh với tín hiệu đầu vào, bao gồm điện áp tham chiếu tiến (+) V_{REF+}, và điện áp tham chiếu lùi (-) V_{REF-}

+ Tốc độ chuyển đổi: tốc độ chỉ định thời gian để hoàn thành 1 lần chuyển đổi A/D (nén 1).

- Công thức, tính cách áp

$$V_A = \frac{Dout \cdot (V_{ref+} - V_{ref-})}{2^n - 1} + V_{ref-}$$

V_A: điện áp vào tần số

Dout: đầu ra số

n: số bit của ADC

V_{ref+}: điện áp tham chiếu tiến (+)

V_{ref-}: điện áp tham chiếu lùi (-)

2. Bộ ADC tuân vi đặc

- Giải thích

+ Một bộ bộ ADC có 10 bit

- + 12 kênh đầu vào AN0 → AN11, chuyển kênh = phần mềm
- + Tiến áp tham chiếu có thể lựa chọn từ nguồn cấp (VSS, VDD) or bên ngoài (CANL, CANH)
- = phần mềm
- + Nguồn xung cho lấy từ Fosc
- Công thức chuyển đổi tương tự - rõ trên PIC16F1520

$$V_A = Pout + (Vref+ - Vref-) - V_{ref-}$$

(102)

V_A : tiến áp vào (voltage)
Pout: đầu ra số (10 bit)

- Lựa chọn kênh chia tách cho ADC
 - + Chỉ lựa chọn kênh rõ chia tách làm or = biến chia tra cứu & bằng
 - + Giá trị Fosc = 8MHz → số chia tách lựa chọn ≥ 8
- Lựa chọn cách ghi kết quả trên & thành ghi 8bit ADRESH & ADRESL
- Các thành phần ghi đt' của ADC
 - + ADRESH: thành phần ghi chép k'g byte cao
 - + ADRESL: _____ thấp
 - + ADCONO: thành phần ghi đt' AID0
 - + ADCONA: _____ 1
 - + ADCON2: _____ 2

3 Chuyển đổi, lưu hiệu tương tự sang số

- Các bước dt' ADC o như sau: biến, biến, đọc, ghi

Busy ADC	Nhận báo báo
Close ADC	Cài đặt chế độ đóng ADC
Convert ADC	Bắt đầu quá trình ch' đổi A/D
Open ADC	Cài đặt chế độ mở chuyển đổi A/D
Read ADC	Lấy giá trị số của ch' đổi A/D
Set Chan ADC	Chọn kênh đầu vào cho bộ A/D

- Các bước chuyển đổi ADC:

B1. Khởi tạo rò rỉ sa TRISx, ADCON1...

B2. Cài đặt ADC Open ADC();

B3. Chuyển đổi ADC

- + Chọn kênh ADC (nếu cần chuyển kênh) - SetChanADC(..);
- + Bắt đầu quá trình ch' đổi ConvertADC();
- + Chờ chuyển đổi xong While(BusyADC());
- + Phân nhận kết quả số = ReadADC();

§8. Điều chế độ rộng xung (PWM)

→ Giải thích xung PWM & ứng dụng

Khái niệm:

- + Điều chế độ rộng xung (PWM) chỉ viết tắt bởi cụm là Pulse-Width Modulation. Đây là phương pháp điều chế độ rộng xung thông thường có chu kỳ cố định.
- + Periode - là chu kỳ của xung PWM & là ký hiệu là T
- + Duty cycle - là tỷ độ rộng xung chiếm của PWM, ký hiệu DT (điều chỉnh từ 0-100%)

Ứng dụng:

- + Điều khiển tốc độ quay của động cơ DC Servo
- Trong thực tế, PWM là một dạng điều chế tín hiệu, có thể là tín hiệu xung (Timing Wave) và tín hiệu analog (analog signal) mà nói ở một điều và tín hiệu mô đun kim
- + Điều khiển độ sáng của đèn (đèn nền màn hình LCD, đèn sưởi nhà tắm, đèn quang cáo...)
- + Điều khiển tốc độ motor DC (robot, xe điện,...)
- + Điều chỉnh điện áp.

② Bộ tạo xung PWM trên pic 18f4520

- Giải thích:

- + Vì điều khiển pic 18f4520 có module CCP (Capture / Compare / PWM) là CCP1 & CCP2
- CCP có thể hoạt động ở chế độ chụp (so sánh số 1 với xung PWM)
- + Xung PWM1 để tạo tần số CCL1 / RC1, xung PWM2 để tạo tần số CCL2 / RC2 hoặc CCP1 / PR2
- + Số nguyên xung từ Timer2 + tạo chu kỳ = Thanh ghi PR2
- Số bit tần số CCP1 / RC1: CCP1CON <5:4> để tạo độ rộng (x=1 or x=2)

- Thiết lập chu kỳ xung PWM

- + Chu kỳ là khoảng cách từ S → S, do thiết lập = Cycles.

$$T_{\text{PWM}} = PR_2 \cdot (PR_2 + 1)^{-1} \cdot F_{\text{osc}} \cdot (PV_{\text{PR}_2})$$

T_{PWM} - chu kỳ của xung PWM

F_{osc} - Chu kỳ xung hệ thống

PV_{PR_2} - Hỗn số chia tần số của bộ Timer2

- Thiết lập độ rộng xung cho PWM

Độ rộng xung là khoảng cách từ S → R, do thiết lập = Cycles

$$D9 = (CCP1CON <5:4>) \cdot T_{\text{osc}} \cdot (PV_{\text{PR}_2})$$

D9: độ rộng xung của PWM

Công thức này để phân giải tối đa

③ Điều khiển PWM sử dụng PWM.h

- Các hàm trong thư viện PWM.h

- + Close PWMx : Cắt kết của PWM kênh x
- + Open PWMx : Mở kết của PWM kênh x
- + Set DCPWMx : Thiết lập độ rộng xung cho PWM kênh x

PWM₁ (CCP1) $x=1$

PWM₂ (CCP2) $x=2$

- Cài bước tạo xung PWM

B1. Thiết lập vào I/O

Thiết lập TRISB, TRISC, ADCON2 để CCP1 or CCP2 có ở I/O số

còn lại là CCP2MX - PORTB or CCP2M~~UX~~ = PORTC

B2. Thiết lập chu kỳ

Tính chia PR2 / PV_{TMR2} theo % chu kỳ + thiết lập :

- Hamin Open PWM1 or Open PWM2

- Hết % chia tách PWM2: or Open T2CON (T2CKPS1 · T2CKPS0)

B3. Thiết lập độ rộng xung PWM

Tính chia dutycycle theo công thức thiết lập độ rộng, sau đó $\text{cái} = \text{hamin}$

Set DC PWM1(C) or Set DC PWM2(C)

VD: Fosc = 8MHz, fao ra xung PWM1 (CCP1) có tần số 1kHz, độ rộng xung = 70% chu kỳ

- Thiết lập chu kỳ = cách chia PR2 + % chia tách PV_{TMR2}

Fosc = 8MHz \rightarrow Tosc = 1/8 (us)

F_{PWM} = 1kHz \rightarrow T_{PWM} = 1/1000 (s) = 1000 (us)

PR2 \leftarrow 255 (CCP2 là thành phần 8bit)

PV_{TMR2} có thể nhận giá trị (1 .. 4 .. 16)

$$T_{PWM} = (PR2 + 1)^* 4^* T_{osc} * (PV_{TMR2})$$

$$\text{Chia } PV_{TMR2} = 16 \rightarrow PR2 = 124 \quad (255 \rightarrow \text{phê duyệt})$$

- Tính chia dutycycle

% rộng = 70% chu kỳ : DT = 700 (us)

Tosc = 1/8 (us)

PV_{TMR2} = 16

$$DT = ((CCPxL : CC PxCON (5:4)) * T_{osc} * (PV_{TMR2}))$$

$$\Rightarrow \underline{\text{dutycycle}} = CCPxL : CC PxCON (5:4) = 350$$

15K29

8) Thiết kế ứng dụng sử dụng PIC18F4520

* Quay tròn thiết kế một ứng dụng sử dụng số PIC18F4520

B1. Phân tích bài toán

B2. Thiết kế sơ đồ nguyên



B3. Lập trình & mô phỏng

B4. Thiết kế, làm mạch in & lắp ráp linh kiện trên mạch in

B5. Nạp chương trình vào vi dtb



B6. Chạy thử nghiệm hàn thử

* Nguyên tắc luân thi: $LED + doan = 0^{\circ}$ quét

- Lý do: + Mô hình led + thanh cờ \Rightarrow 8 led, $\angle LED = 16$ chấn
 \rightarrow Lòng phai tài nguyên

- Nhận:

+ Dựa trên nguyên lý luân thi trên vòng mac. Tay I thi, đ' chỉ có 1 LED + doan sáng, các LED + doan khác = 0 $^{\circ}$. Vì cách \neq , các LED + doan luân phiên nhau sáng vs cách lỗ theo ý kinh

+ Điều: grain sáng mới LED c 1/24 s nên qáit = mặt ta nhận ra các LED sáng luân
hiết mà có cảm giác như các LED sáng đồng thời.