

**ĐẠI HỌC QUỐC GIA
ĐẠI HỌC BÁCH KHOA TP HỒ CHÍ MINH
KHOA ĐIỆN – ĐIỆN TỬ**



**ĐIỆN TỬ HỌC TƯƠNG TỰ VÀ ỨNG
DỤNG (EE3189)
BÀI TẬP LỚN**

LỚP P01-- HK212

NGÀY NỘP: 11/06/2022

Giảng viên hướng dẫn: NGUYỄN HUỖNH HẠC

Sinh viên thực hiện	Mã số sinh viên
Huỳnh Đăng Khoa	1911398

Thành phố Hồ Chí Minh – 2022

ĐỀ BÀI: Xe tự động dò đường

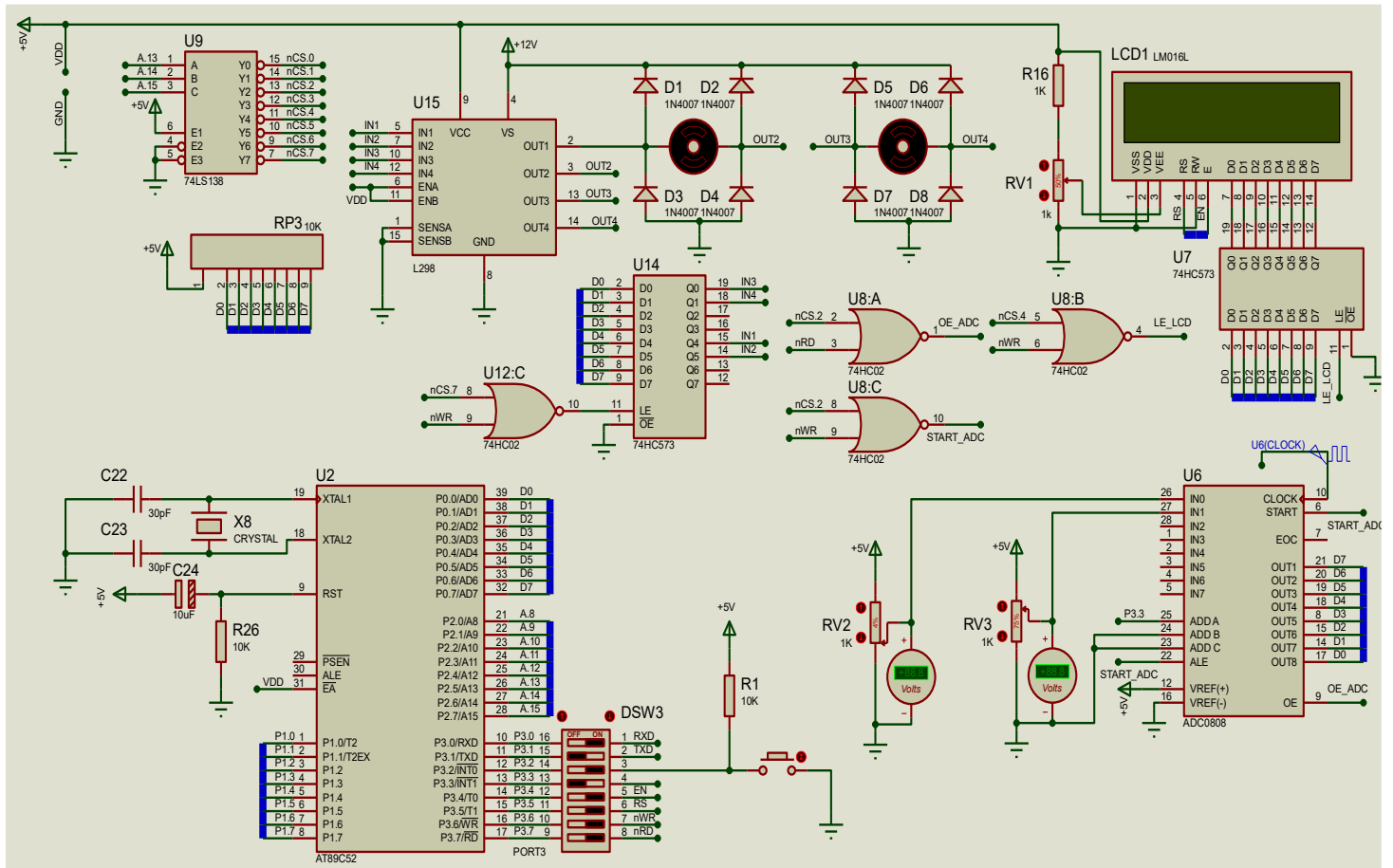
1. Bộ thu phát tia hồng ngoại: nhận tia hồng ngoại phản xạ từ vạch đường.
2. Bộ chuyển đổi ADC chuyển đổi các tín hiệu từ bộ thu hồng ngoại sang giá trị số.
3. Vi điều khiển 8051 thực hiện chức năng đọc các giá trị cảm biến và điều khiển hoạt động của động cơ.
4. Màn hình LCD hiển thị các thông báo, trạng thái khác nhau của xe.
5. Bộ điều khiển động cơ DC 12V
6. Động cơ DC truyền động cho các bánh xe, gồm 2 động cơ và 2 bánh trái phải.

Xe tự động dò đường đi theo 1 đường dẫn màu đen được vẽ trên mặt đường màu trắng. Trong khi di chuyển, xe luôn đọc giá trị của các cảm biến hồng ngoại. Nếu xe đang đi đúng hướng thì nó sẽ tiếp tục di chuyển theo hướng về phía trước. Khi các cảm biến thông báo cho vi điều khiển biết rằng xe không đi theo vạch dẫn đường thì vi điều khiển sẽ quay xe theo hướng trái hoặc phải.

Yêu cầu:

- Trình bày ý tưởng thực hiện
- Vẽ sơ đồ khối phân cứng
- Lưu đồ giải thuật + chương trình

I. SƠ ĐỒ KẾT NỐI PHẦN CỨNG



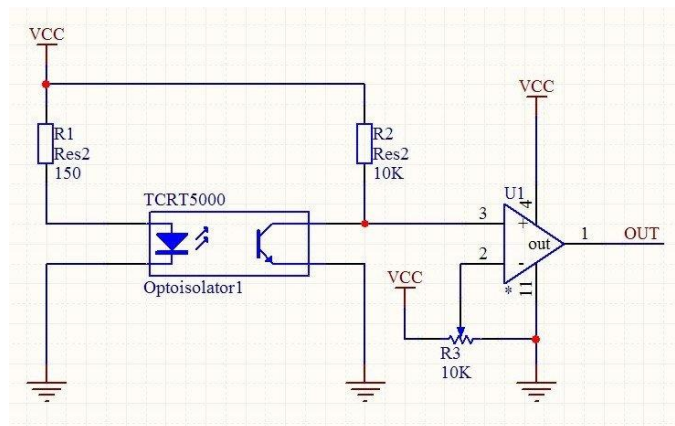
1. Bộ thu/phát hồng ngoại

Cảm biến thu/phát hồng ngoại TCRT5000 gồm 1 mắt thu (trắng) và 1 mắt phát (xanh)

- Trong line cho ra mức 1
- Ngoài line cho ra mức 0

Bên trong là một bộ so sánh điện áp, sử dụng IC LM393 (bao gồm 2 opamp). Cho ra 2 chân A0 (analog) và D0 (digital).

Ta sử dụng chân A0 để kết nối với ADC, chuyển đổi sang giá trị số để MCU có thể đọc được.



IN LINE: Xe tiếp tục chạy thẳng

OUT LINE: Xe rẽ hướng

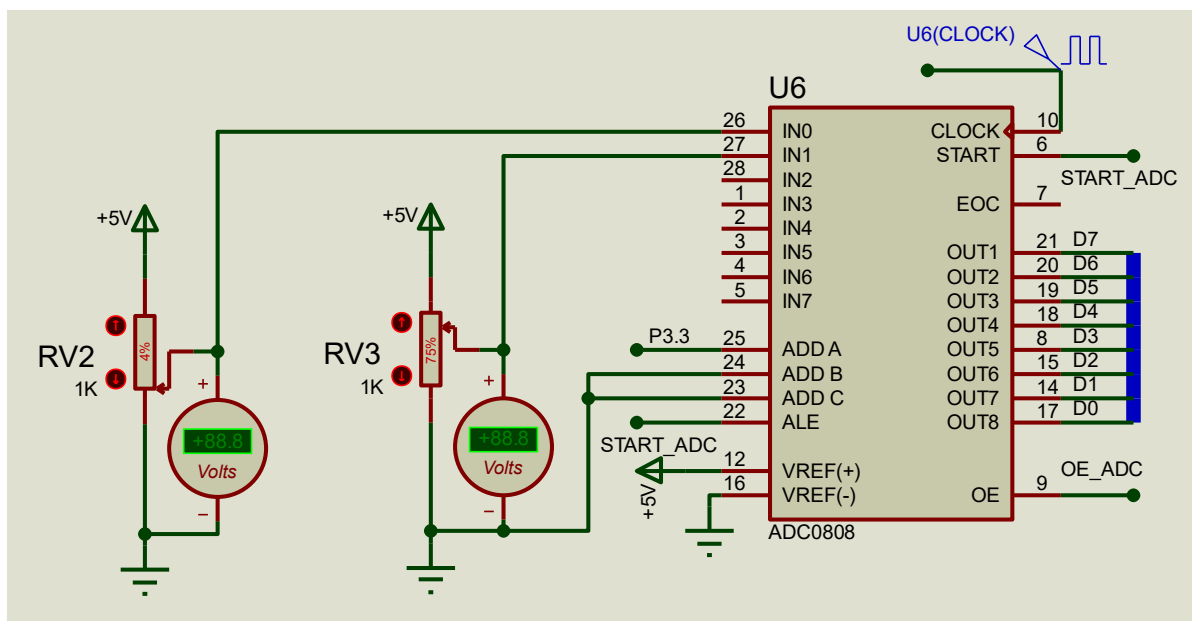
OUT LINE: Xe tiếp tục chạy thẳng

Cảm biến thu/phát hồng ngoại TCRT5000 gồm 1 mắt thu (trắng) và 1 mắt phát (xanh)

- Trong line cho ra mức 1
- Ngoài line cho ra mức 0

2. Bộ chuyển đổi ADC

Bộ chuyển đổi ADC chuyển đổi các tín hiệu từ bộ thu hồng ngoại sang giá trị số.



Ta sử dụng 2 biến trở RV2 và RV3 để tạm mô tả 2 cảm biến hồng ngoại, ngõ vào đưa vào kênh IN0 và IN1.

Để chốt kênh cần chuyển đổi, ta sử dụng P3.3 nối với ADD A của ADC, 2 đường địa chỉ còn lại ta nối xuống mass. Tín hiệu chốt sẽ được nối chung với tín hiệu bắt đầu chuyển đổi của ADC.

Mức điện áp của cảm biến hồng ngoại mức cao khoảng từ 4 – 4.8V, mức thấp khoảng 0 – 0.3V. Khi đó ta dùng bộ ADC sau khi chuyển đổi IN0 và IN2, so sánh 2 giá trị này với 153 (tương đương cỡ 3V) để nhận biết in/out line.

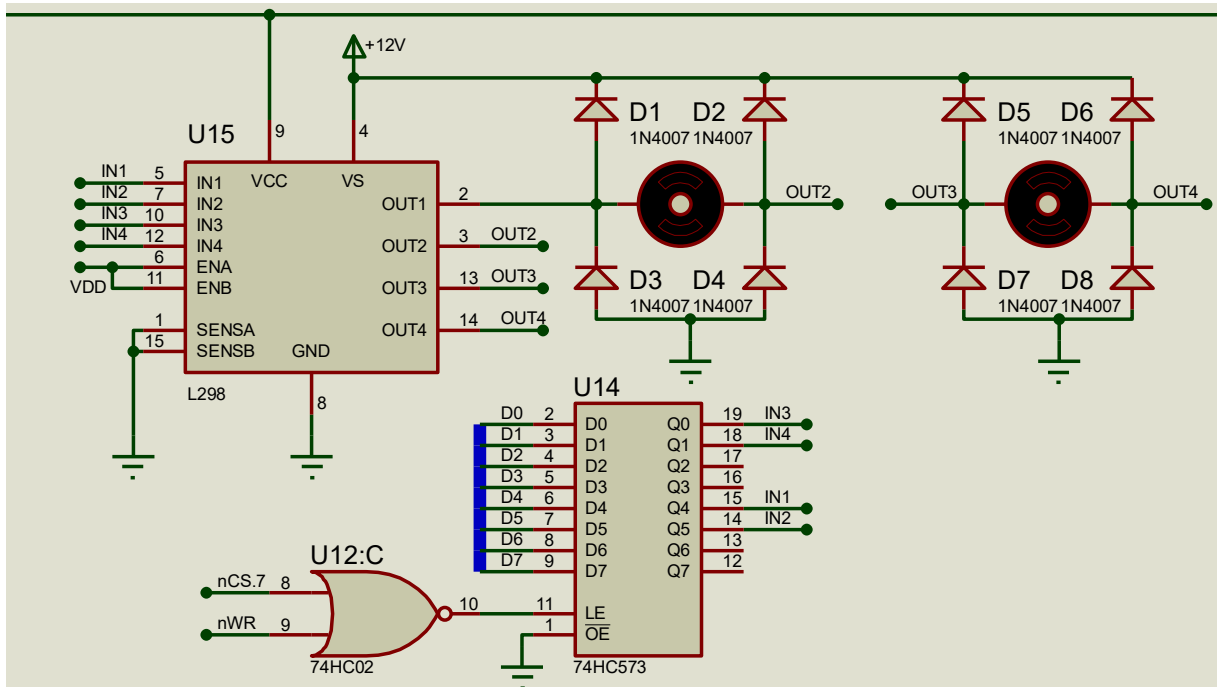
3. Vi điều khiển 8051

Mặc khác, ta sử dụng 1 nút nhấn kết nối với chân /INT0 dùng để tăng tốc độ của động cơ bằng cách tăng Duty Circle lên 10% mỗi lần nhấn.

Màn hình LCD hiển thị các thông báo, trạng thái khác nhau của xe.

- [illegible]

Động cơ DC truyền động cho các bánh xe, gồm 2 động cơ và 2 bánh trái phải.



II. Ý tưởng thực hiện, giải thuật và chương trình

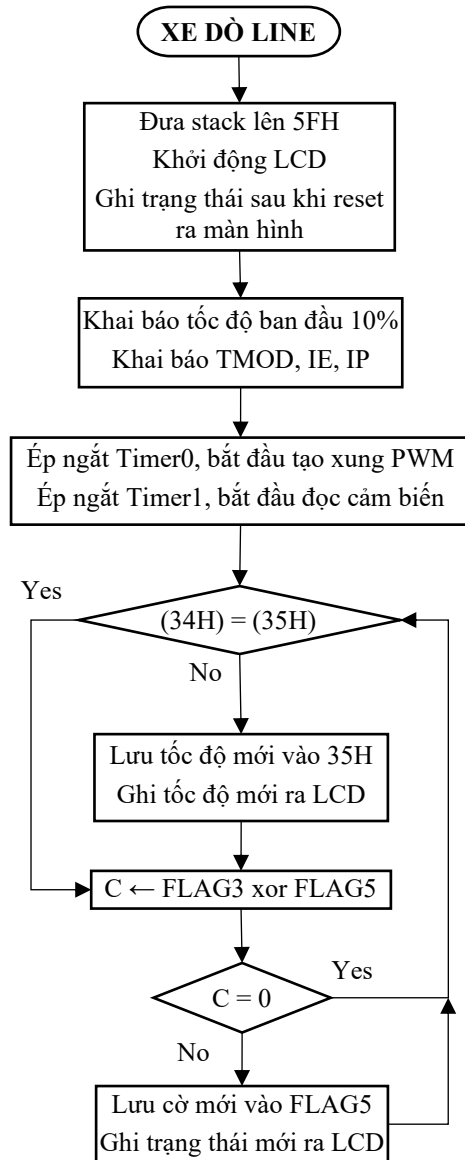
Đầu tiên, dựa trên sơ đồ kết nối phần cứng, ta sẽ định nghĩa cho trình biên dịch một số chỉ thị để phục vụ cho việc giao tiếp với LCD, với cảm biến hồng ngoại thông qua ADC, điều khiển động cơ, bao gồm các cờ phục vụ quá trình xét điều kiện, nhận biết hướng rẽ, hiển thị trạng thái, tốc độ động cơ,...

Bảng 1

Tên	Chỉ thị	Địa chỉ	Ghi chú
RS	BIT	P2.0	Giao tiếp LCD 16x2
E	BIT	P2.1	
FLAG0	BIT	00H	Cờ báo trạng thái xung PWM, =0 khi xung đang ở mức 0, = 1 khi xung đang ở mức 1
FLAG1	BIT	01H	Cờ motor trái
FLAG2	BIT	02H	Cờ motor phải
FLAG3	BIT	03H	Cờ in/out line
FLAG4	BIT	04H	
FLAG5	BIT	05H	Cờ nhớ trạng thái in/out line

STT	Bộ nhớ và Ngoại vi	Địa chỉ truy xuất	Ghi chú
1		0000h – 1FFFh	nCS0
2	Bộ nhớ RAM ngoài	2000h – 3FFFh	nCS1
3	Ra lệnh START ADC và chốt kênh cần chuyển đổi	4000h – 5FFFh	Chỉ ghi nCS2
4	Đọc 8bits của ADC	4000h – 5FFFh	Chỉ đọc nCS2
5			nCS3
6	Điều khiển chốt 573 chốt 8 bits dữ liệu của LCD	8000h – 9FFFh	Chỉ ghi nCS4
7		A000h – BFFFh	nCS5
8		C000h – DFFFh	nCS6
9	Điều khiển chốt 573 chốt 4 bits điều khiển của 2 động cơ trái phải	E000h – FFFFh	Chỉ ghi nCS7

SƠ ĐỒ KHỐI TỔNG QUÁT CHO TOÀN BỘ CHƯƠNG TRÌNH



Ý tưởng:

Sử dụng ngắt Timer 0 để điều chế xung điều khiển động cơ, xung PWM tần số 1.25KHz ($T = 0.8\text{ms}$).

- Giá trị nạp cho mức 1 nằm ở 30H, 31H
- Giá trị nạp cho mức 0 nằm ở 32H, 33H
- Ô nhớ 34H ghi tốc độ động cơ (1–10)
- Trình ngắt sẽ đọc các cờ điều khiển tương ứng để cấp, hoặc không cấp xung cho động cơ.

Sử dụng ngắt Timer1 để điều khiển ADC đọc giá trị cảm biến sau mỗi 1ms. Trình ngắt sau đó sẽ nhận biết trạng thái rồi điều khiển các cờ tương ứng.

Sử dụng ngắt ngoài 0 cho phép người dùng tăng tốc độ động cơ. Khi tăng hết mức (100%) sẽ trở về mức 10%.

Chương trình

; Xe đo line su dung cam bien hong ngoai

;

;Written by: HUYNH DANG KHOA 11:11PM

08/06/2022

;

;LCD

RS BIT P3.5

EN BIT P3.4

;

FLAG0 BIT 00H ;PWM, =1 khi xung PWM ở mức 1 và ngược lại

FLAG1 BIT 01H ;MOTOR trái

FLAG2 BIT 02H ;MOTOR phải

FLAG3 BIT 03H ;cờ IN/OUT LINE

FLAG4 BIT 04H ;

FLAG5 BIT 05H ;cờ nhớ để so với FLAG3

FLAG6 BIT 06H ;

FLAG7 BIT 07H ;

;

IN_ADC BIT P3.3 ;Chốt địa chỉ ADC

;

ORG 0000H

LJMP MAIN

LJMP INT0_ISR

ORG 000BH

LJMP T0_ISR

ORG 001BH

LJMP T1_ISR

;

ORG 0033H

MAIN:

MOV SP,#4FH

;

CALL INITIAL ;khởi động LCD

MOV DPTR,#TAB_LINE1

CALL PRINT_LINE ;In dòng 1

MOV A,#0C0H

CALL DELAY_2MS

CLR RS

CALL WRITE_OUT

MOV DPTR,#TAB_LINE2

CALL PRINT_LINE ;In dòng 2

;

SETB EA ;Khai báo ngắt

MOV IP,#02H ;Khai báo ưu tiên ngắt

;

MOV TMOD,#11H ;Khai báo TMOD

```

;-----
MOV 30H,#HIGH(-80) ;Khởi tạo tốc độ động cơ ban đầu
MOV 31H,#LOW(-80)
MOV 32H,#HIGH(-720)
MOV 33H,#LOW(-720)
MOV 34H,#1 ;Mức 1/10
;-----
SETB ET0
SETB TF0 ;ép ngắt TIMER 0
;-----
SETB IT0 ;ngắt ngoài là ngắt cạnh xuống
SETB EX0
;-----
SETB ET1
SETB TR1 ;ép ngắt TIMER 1
;-----

LOOP_00:
MOV A,34H
CJNE A,35H,SPEED_CHANGE ;so sánh tốc độ trước đó
SJMP LOOP_01

SPEED_CHANGE:
MOV 35H,34H ;lưu tốc độ mới
CALL WRITE_SPEED ;ghi tốc độ mới
;----- FLAG3 xor FLAG5 để xe có khác không?

LOOP_01:
CLR EA
MOV C,FLAG3
JNC LOOP_02
MOV C,FLAG5
CPL C
JMP LOOP_03

LOOP_02:
MOV C,FLAG5
;-----

LOOP_03:
SETB EA
JNC LOOP_00 ;không khác thì quay trở về xét lại từ đầu
CLR EA
MOV C,FLAG3 ;lưu cờ mới vào FLAG5
MOV FLAG5,C
SETB EA
CALL WRITE_STATE ;ghi trạng thái mới

SJMP LOOP_00 ; quay trở về xét lại từ đầu

TAB_LINE1: DB 'STATE : LINE'
TAB_LINE2: DB 'SPEED : %'
;=====
PRINT_LINE: MOV R0,#16 ;chương trình con in 1 hàng ra LCD
MOV A,#0
PRINT_LINE_1: PUSH ACC
CALL PRINT_CHAR
POP ACC

```

```

        INC    A
        DJNZ   R0,PRINT_LINE_1
        RET

;=====
PRINT_CHAR:                                ;chương trình con ghi 1 ký tự ra LCD, có xét mã NULL
        MOV    A,@A+DPTR
        CJNE   A,#00H,PRINT_CHAR1
        SETB   C
        RET
PRINT_CHAR1:  CALL  DELAY_2MS
        SETB   RS
        PUSH   DPH
        PUSH   DPL
        CALL   WRITE_OUT
        POP    DPL
        POP    DPH
        CLR    C
PRINT_CHAR2:  RET
;-----
WRITE_STATE:                                ;chương trình con ghi trạng thái mới
                                                ;dòi con trở về 08H
        MOV    A,#88H
        CALL   DELAY_2MS
        CLR    RS
        CALL   WRITE_OUT
        JB     FLAG5,IN_LINE
        MOV    39H,#'O'                    ;"OUT"
        MOV    3AH,#'U'
        MOV    3BH,#'T'
        JMP    EXIT_WRITE_STATE
IN_LINE:
        MOV    39H,#' '                    ;" IN"
        MOV    3AH,#'I'
        MOV    3BH,#'N'
EXIT_WRITE_STATE:
        MOV    R1,#39H
        CALL   WRITE_RAM                    ;Ghi 39H, 3AH, 3BH ra LCD
        RET
;-----
WRITE_SPEED:
        MOV    A,#0CCH                    ;dòi con trở về 4CH
        CALL   DELAY_2MS
        CLR    RS
        CALL   WRITE_OUT
        MOV    A,35H
        CJNE   A,#10,WRITE_SPEED_00
        MOV    36H,#'1'
        MOV    37H,#'0'
        MOV    38H,#'0'
        JMP    WRITE_SPEED_01

```

```

WRITE_SPEED_00:
    MOV 36H,#' '
    MOV A,#30H
    ADD A,35H
    MOV 37H,A
    MOV 38H,#'0'
WRITE_SPEED_01:
    MOV R1,#36H
    CALL WRITE_RAM
    RET
;-----

WRITE_RAM:
    ;chương trình con ghi nội dung 3 ô nhớ liên tiếp trữ bởi R1 ra LCD
    MOV R2,#3
WRITE_RAM_00:
    MOV A,@R1
    CALL DELAY_2MS
    SETB RS
    CALL WRITE_OUT
    INC R1
    DJNZ R2,WRITE_RAM_00
    RET
;-----

INITIAL:
    ;chương trình con khởi tạo LCD
    ;8 bits 5x8 dots
    MOV A,#38H
    CALL DELAY_2MS
    CLR RS
    CALL WRITE_OUT
    ;-----
    MOV A,#01H
    ;clscr
    CALL DELAY_2MS
    CLR RS
    CALL WRITE_OUT
    ;-----
    MOV A,#0FH
    ;Hien man hinh, chop ky tu
    CALL DELAY_2MS
    CLR RS
    CALL WRITE_OUT
    ;-----
    MOV A,#06H
    ;Dich con tro sang phai (khi ghi/doc data)
    CALL DELAY_2MS
    CLR RS
    CALL WRITE_OUT
    ;-----
    RET
;-----

DELAY_2MS:
    MOV R7,#4
DELAY_2MS_0:
    MOV R6,#250
    DJNZ R6,$
    DJNZ R7,DELAY_2MS_0
    RET
;-----

```

DELAY_120us:

```
MOV R5,#60
DJNZ R5,$
RET
```

;-----

```
WRITE_OUT:    MOV DPTR,#9000H      ;8000H - 9FFFH ghi LCD
               MOVX @DPTR,A        ;Chương trình con ghi 1 ký tự ra LCD
               SETB EN
               CLR EN
               RET
```

;=====

INT0_ISR: ;trình ngắt ngoài 0, thay đổi 30H,31H,32H,33H,34H để thay đổi tốc độ động cơ

```
PUSH ACC
PUSH PSW
PUSH 82H
PUSH 83H
CLR ET0
```

```
MOV R3,#50      ;Xét 50 lần chống rung phím
SETB P3.2       ;DUA CHOT PORT LEN 1 ROI MOI DOC
CHECK_SW: MOV C,P3.2
             JC EXIT_INT0_ISR
             DJNZ R3,CHECK_SW
```

```
INC 34H
MOV A,34H
CJNE A,#11,INT0_ISR_0
MOV 34H,#1
MOV 30H,#HIGH(-80) ;FFB0H = -80
MOV 31H,#LOW(-80)
MOV 32H,#HIGH(-720) ;FD30H = -720
MOV 33H,#LOW(-720)
SJMP EXIT_INT0_ISR
```

INT0_ISR_0:

```
MOV A,31H
CLR C
SUBB A,#80
JNC INT0_ISR_1
DEC 30H
```

INT0_ISR_1:

```
MOV 31H,A
MOV A,33H
ADD A,#80
JNC INT0_ISR_2
INC 32H
```

INT0_ISR_2:

```
MOV 33H,A
```

EXIT_INT0_ISR:

```
SETB ET0
POP 83H
POP 82H
```

```

        POP    PSW
        POP    ACC
        RETI
;=====
T0_ISR:                ;trình ngắt Timer0 điều chế xung điều khiển động cơ

        PUSH  ACC
        PUSH  PSW
        PUSH  82H
        PUSH  83H

        CLR   TR0
        MOV   A,34H
        CJNE  A,#10,T0_ISR_01
        MOV   TH0,HIGH(-800)
        MOV   TL0,LOW(-800)
        SETB  TR0
        SETB  FLAG0
        JMP   RUN_MOTOR
        ;-----

T0_ISR_01:             JNB   FLAG0,NON_DUTY
                     CLR   FLAG0
                     MOV   TH0,30H                ;THOI GIAN MUC 0
                     MOV   TL0,31H
                     SETB  TR0
                     JMP   RUN_MOTOR
                     ;-----

NON_DUTY:              SETB  FLAG0
                     MOV   TH0,32H                ;THOI GIAN MUC 1
                     MOV   TL0,33H
                     SETB  TR0
                     JMP   STOP_MOTOR
                     ;-----

RUN_MOTOR:            MOV   DPTR,#0F000H
                     MOV   A,#0
                     JNB   FLAG1,RUN_MOTOR_1      ;xét cờ MOTOR 1
                     ORL   A,#10H                ;chạy bánh trái, 1 quay thuận
                     MOVX  @DPTR,A

RUN_MOTOR_1:          JNB   FLAG2,EXIT_T0_ISR      ;xét cờ MOTOR 2
                     ORL   A,#01H                ;chạy bánh phải
                     MOVX  @DPTR,A
                     SJMP  EXIT_T0_ISR
                     ;-----

STOP_MOTOR:           MOV   DPTR,#0F000H
                     MOV   A,#0
                     MOVX  @DPTR,A

```

EXIT_T0_ISR:

```
POP    83H
POP    82H
POP    PSW
POP    ACC
RETI
```

=====

T1_ISR: ;trình ngắt Timer1, đọc cảm biến rồi điều khiển các cờ báo

```
PUSH    ACC
PUSH    PSW
PUSH    82H
PUSH    83H
```

```
CLR     EA
CLR     TR1
```

```
MOV     TH1,HIGH(-1000) ;1ms kiểm tra 1 lần
```

```
MOV     TL1,LOW(-1000)
```

```
SETB    TR1
```

```
MOV     DPTR,#5000H
```

```
MOV     A,#0
```

```
CLR     IN_ADC
```

;đọc IN0 , cảm biến bên trái

```
MOVX    @DPTR,A
```

```
SETB    EA
```

```
CALL    DELAY_120us
```

```
CLR     EA
```

```
MOVX    A,@DPTR
```

```
MOV     40H,A
```

```
MOV     DPTR,#5000H
```

```
MOV     A,#0
```

```
SETB    IN_ADC
```

```
MOVX    @DPTR,A
```

;đọc IN1 , cảm biến bên phải

```
SETB    EA
```

```
CALL    DELAY_120us
```

```
CLR     EA
```

```
MOVX    A,@DPTR
```

```
MOV     41H,A
```

```
;-----
```

;xử lý dữ liệu đã đọc

```
;CLR     EA
```

```
;-----
```

```
MOV     A,40H
```

```
CJNE    A,#153,$+3
```

;so với 153 (khoảng 3V)

```
CPL     C
```

```
MOV     FLAG1,C
```

```
MOV     A,41H
```

```
CJNE    A,#153,$+3
```

```
CPL     C
```

```

MOV    FLAG2,C
;-----
MOV    C,FLAG1
JNC    T1_ISR_01
MOV    C,FLAG2
CPL    C
JMP    T1_ISR_02
T1_ISR_01: MOV    C,FLAG2
;-----
T1_ISR_02:
JC      T1_ISR_03
SETB    FLAG1      ;giống nhau, xe tiếp tục chạy thẳng
SETB    FLAG2

JNB     FLAG1,CLR_FLAG3
SETB    FLAG3      ;Flag1=1, xe đang trong line
JMP     EXIT_T1_ISR

CLR_FLAG3: CLR    FLAG3      ;;Flag1=0, xe đang ngoài line
JMP     EXIT_T1_ISR

T1_ISR_03:
JNB     FLAG1,TURN_RIGHT
CLR     FLAG1      ;rẽ trái
SETB    FLAG2
CLR     FLAG3
JMP     EXIT_T1_ISR

TURN_RIGHT:
SETB    FLAG1      ;rẽ phải
CLR     FLAG2
CLR     FLAG3

EXIT_T1_ISR:
POP     83H
POP     82H
POP     PSW
POP     ACC
SETB    EA
RETI

END

```

— HẾT —