KỸ THUẬT VI XỬ LÝ

BÁO CÁO BÀI TẬP

Nhóm bài tập: 2 – Nhóm môn học: 10

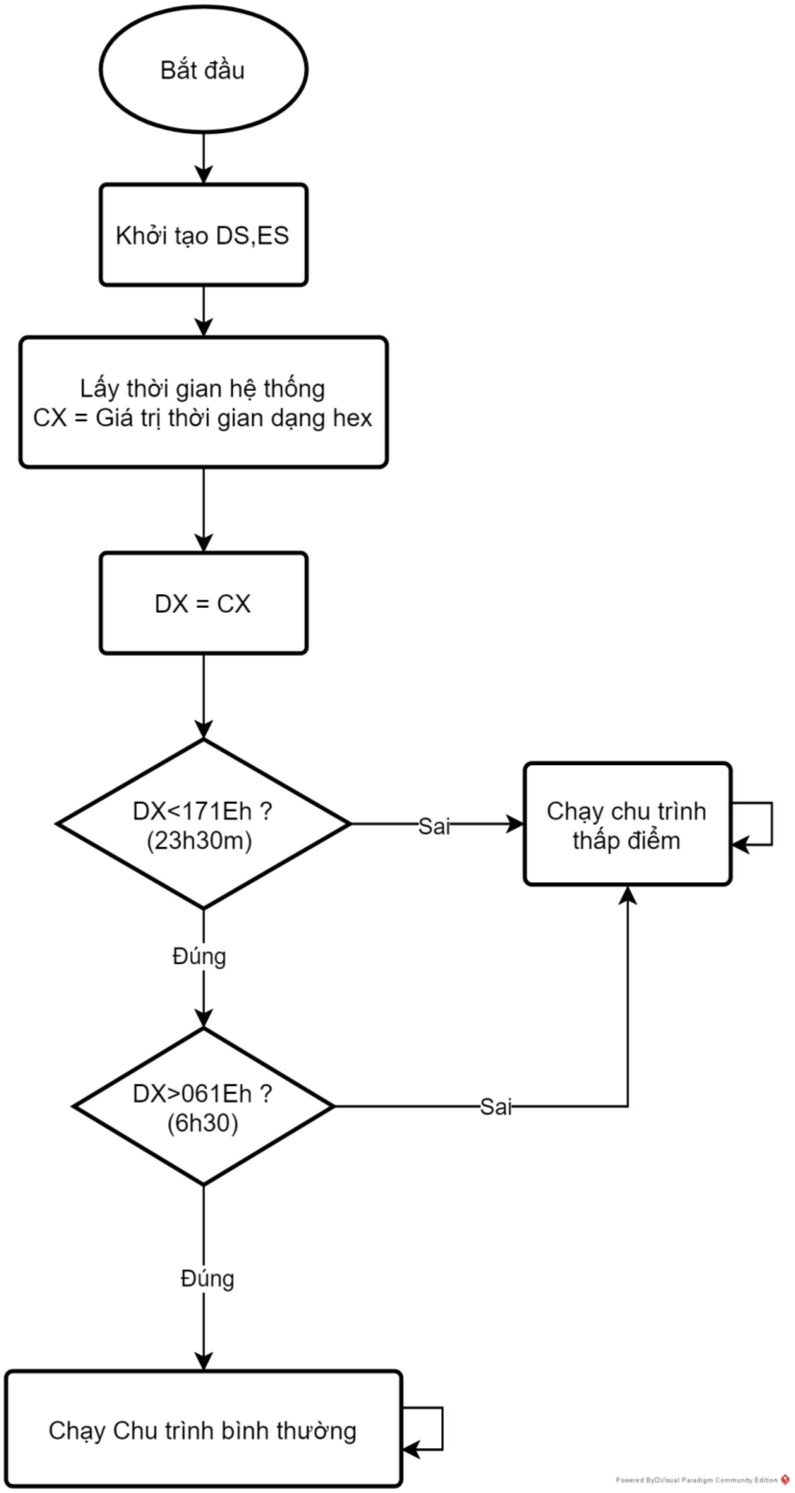
Giảng viên:	TS. Phạm Hoàng Duy	
Thành viên:	Trần Quang Hùng -	B17DCCN291
	Đỗ Đình Chinh -	B18DCCN081
	Ngô Trọng Công -	B18DCCN058
	Phạm Anh Khiêm -	B18DCCN314

Hà Nội, 2020

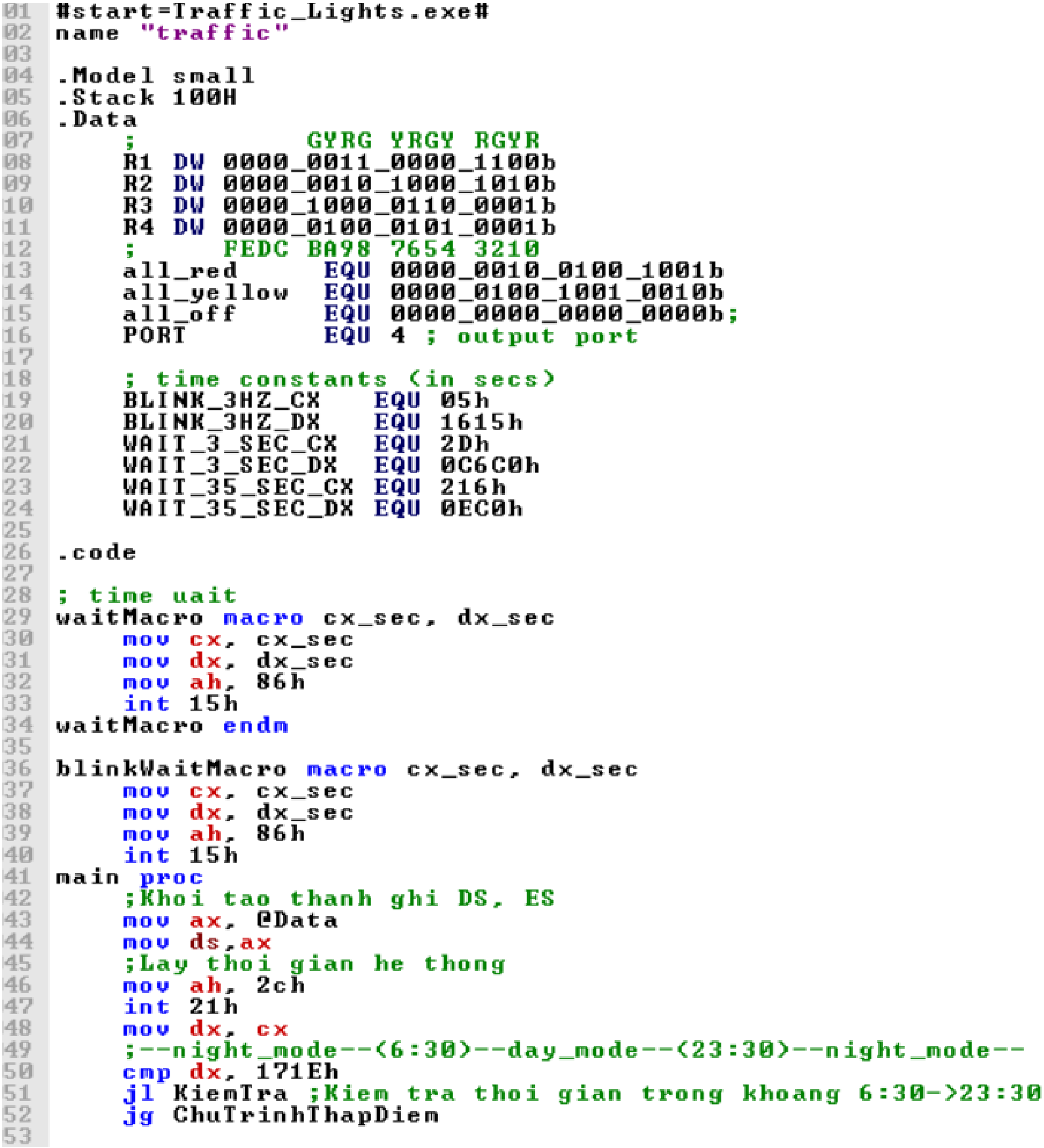
MỤC LỤC

I – ĐIỀU KHIỂN ĐÈN GIAO THÔNG:

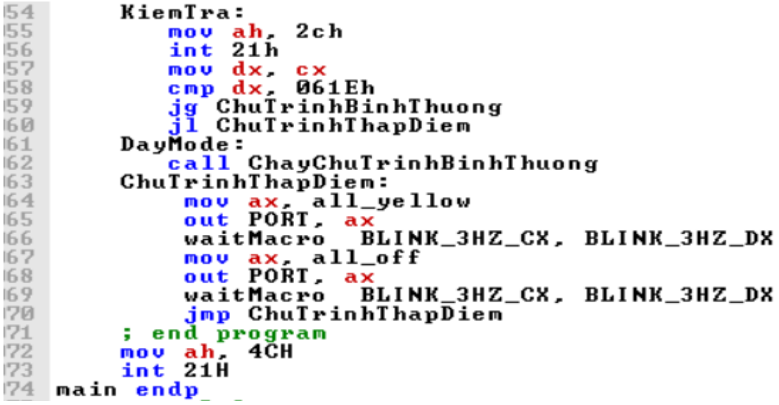
1. Lưu đồ thuật toán:



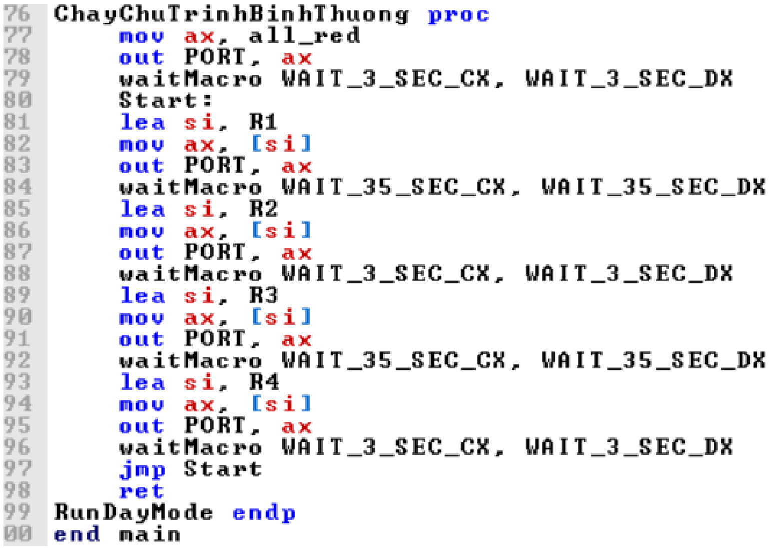
2. Mã nguồn:



```
#start=Traffic_Lights.exe#
name "traffic"

.Model small
.Stack 100H
.Data
    ;          GYRG YRGY RGYR
    R1 DW 0000_0011_0000_1100b
    R2 DW 0000_0010_1000_1010b
    R3 DW 0000_1000_0110_0001b
    R4 DW 0000_0100_0101_0001b
    ;     FEDC BA98 7654 3210
    all_red     EQU 0000_0010_0100_1001b
    all_yellow  EQU 0000_0100_1001_0010b
    all_off     EQU 0000_0000_0000_0000b;
    PORT        EQU 4 ; output port

    ; time constants (in secs)
    BLINK_3HZ_CX   EQU 05h
    BLINK_3HZ_DX   EQU 1615h
    WAIT_3_SEC_CX  EQU 2Dh
    WAIT_3_SEC_DX  EQU 0C6C0h
    WAIT_35_SEC_CX EQU 216h
    WAIT_35_SEC_DX EQU 0EC0h

.code

; time uait
waitMacro macro cx_sec, dx_sec
    mov cx, cx_sec
    mov dx, dx_sec
    mov ah, 86h
    int 15h
waitMacro endm

blinkWaitMacro macro cx_sec, dx_sec
    mov cx, cx_sec
    mov dx, dx_sec
    mov ah, 86h
    int 15h
main proc
    ;Khoi tao thanh ghi DS, ES
    mov ax, @Data
    mov ds,ax
    ;Lay thoi gian he thong
    mov ah, 2ch
    int 21h
    mov dx, cx
    ;--night_mode--(6:30)--day_mode--(23:30)--night_mode--
    cmp dx, 171Eh
    jl KiemTra ;Kiem tra thoi gian trong khoang 6:30->23:30
    jg ChuTrinhThapDiem

```



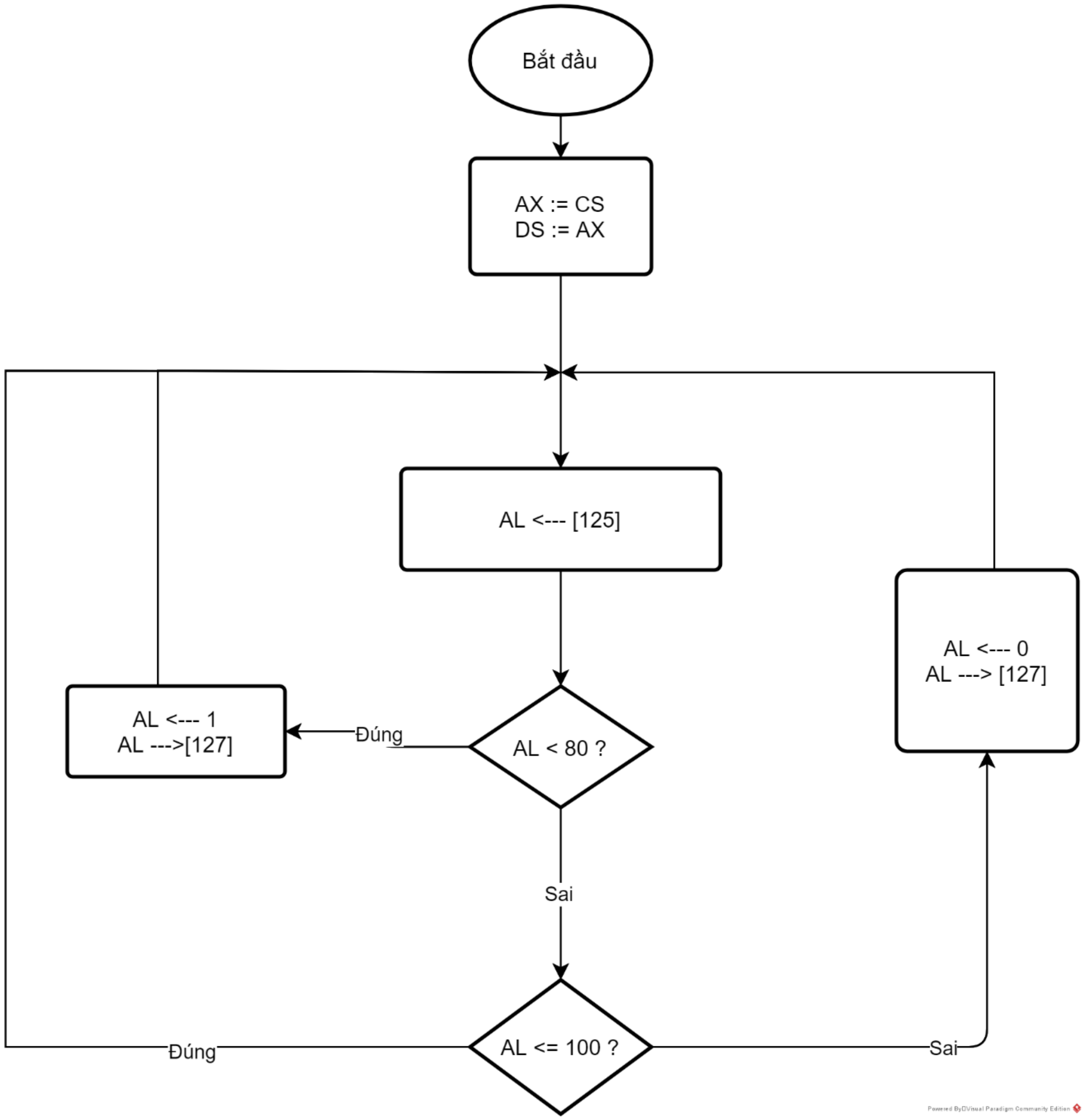
```
    KiemTra:
       mov ah, 2ch
       int 21h
       mov dx, cx
       cmp dx, 061Eh
       jg ChuTrinhBinhThuong
       jl ChuTrinhThapDiem
    DayMode:
       call ChayChuTrinhBinhThuong
    ChuTrinhThapDiem:
        mov ax, all_yellow
        out PORT, ax
        waitMacro  BLINK_3HZ_CX, BLINK_3HZ_DX
        mov ax, all_off
        out PORT, ax
        waitMacro  BLINK_3HZ_CX, BLINK_3HZ_DX
        jmp ChuTrinhThapDiem
    ; end program
    mov ah, 4CH
    int 21H
main endp
```



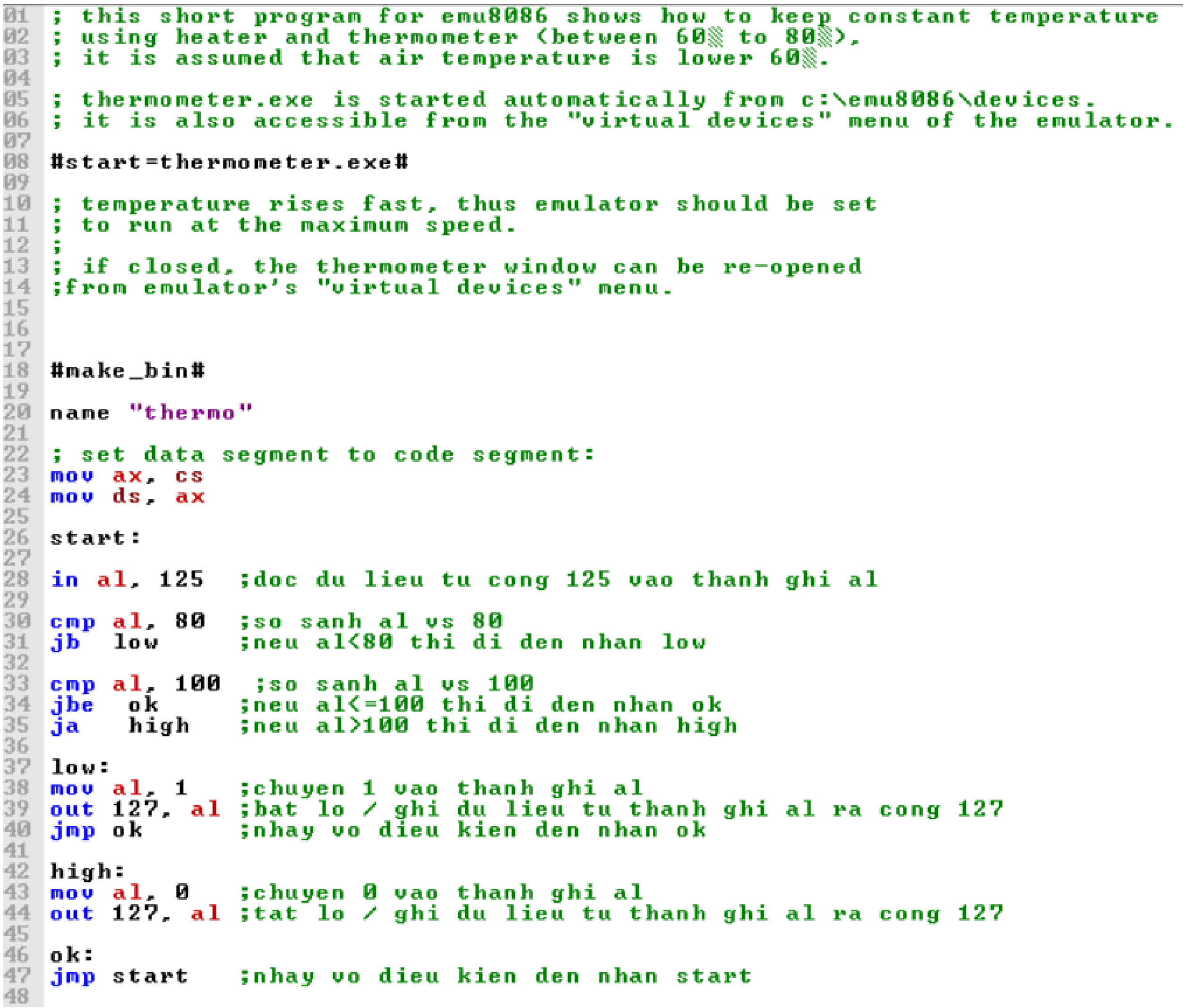
```
ChayChuTrinhBinhThuong proc
     mov ax, all_red
     out PORT, ax
     waitMacro WAIT_3_SEC_CX, WAIT_3_SEC_DX
     Start:
     lea si, R1
     mov ax, [si]
     out PORT, ax
     waitMacro WAIT_35_SEC_CX, WAIT_35_SEC_DX
     lea si, R2
     mov ax, [si]
     out PORT, ax
     waitMacro WAIT_3_SEC_CX, WAIT_3_SEC_DX
     lea si, R3
     mov ax, [si]
     out PORT, ax
     waitMacro WAIT_35_SEC_CX, WAIT_35_SEC_DX
     lea si, R4
     mov ax, [si]
     out PORT, ax
     waitMacro WAIT_3_SEC_CX, WAIT_3_SEC_DX
     jmp Start
     ret
RunDayMode endp
end main
```

II – ĐIỀU KIỂN BẾP THÔNG MINH:

1. Lưu đồ:



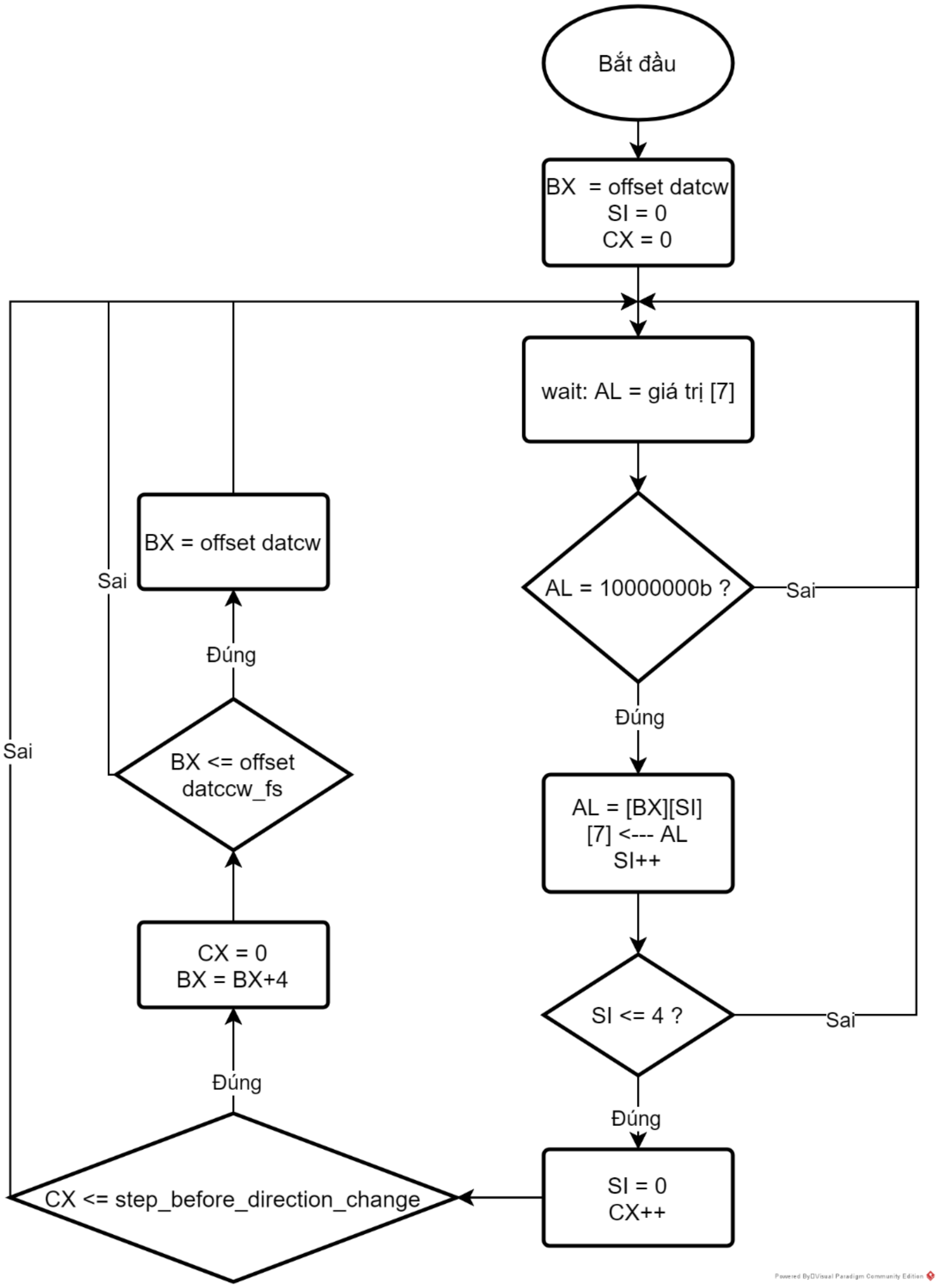
2. Mã nguồn:



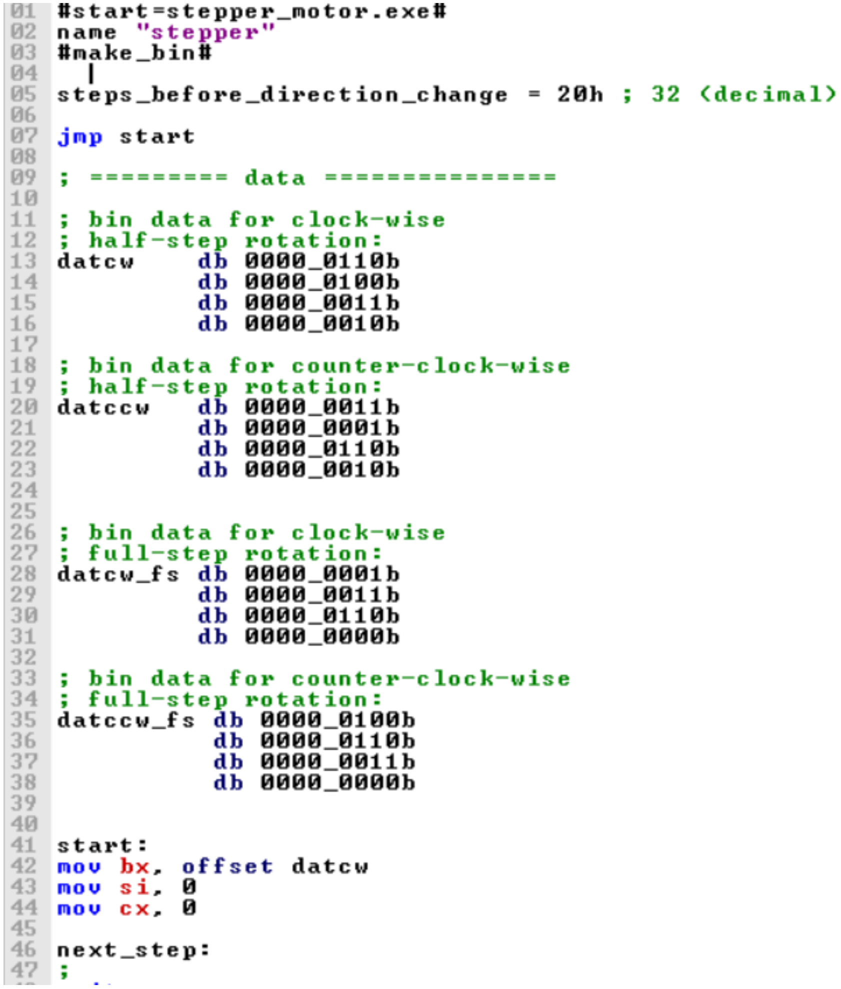
```
; this short program for emu8086 shows how to keep constant temperature
; using heater and thermometer (between 60▒ to 80▒),
; it is assumed that air temperature is lower 60▒.

; thermometer.exe is started automatically from c:\emu8086\devices.
; it is also accessible from the "virtual devices" menu of the emulator.

#start=thermometer.exe#

; temperature rises fast, thus emulator should be set
; to run at the maximum speed.
;
; if closed, the thermometer window can be re-opened
;from emulator's "virtual devices" menu.



#make_bin#

name "thermo"

; set data segment to code segment:
mov ax, cs
mov ds, ax

start:

in al, 125   ;doc du lieu tu cong 125 vao thanh ghi al

cmp al, 80   ;so sanh al vs 80
jb  low      ;neu al<80 thi di den nhan low

cmp al, 100   ;so sanh al vs 100
jbe  ok      ;neu al<=100 thi di den nhan ok
ja   high    ;neu al>100 thi di den nhan high

low:
mov al, 1    ;chuyen 1 vao thanh ghi al
out 127, al ;bat lo / ghi du lieu tu thanh ghi al ra cong 127
jmp ok       ;nhay vo dieu kien den nhan ok

high:
mov al, 0    ;chuyen 0 vao thanh ghi al
out 127, al ;tat lo / ghi du lieu tu thanh ghi al ra cong 127

ok:
jmp start    ;nhay vo dieu kien den nhan start

```

III – ĐIỀU KHIỂN MÔ TƠ:

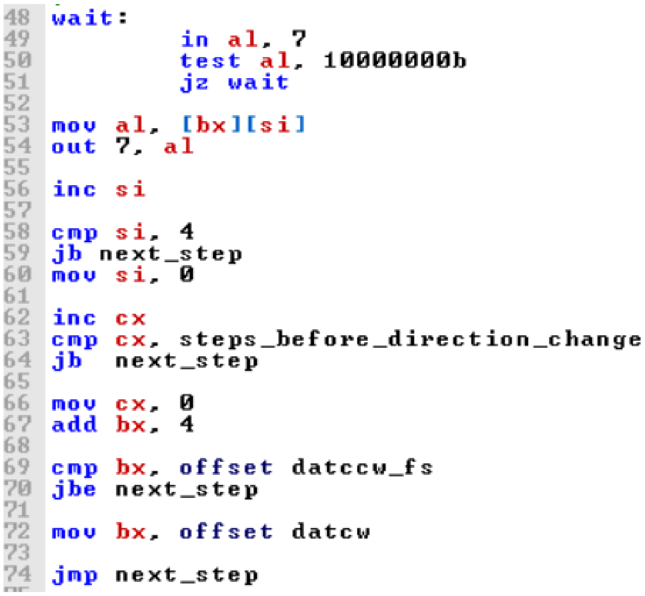
1. Lưu đồ:


Bắt đầu
BX = offset datcw
SI = 0
CX = 0
wait: AL = giá trị [7]
AL = 10000000b ?
Sai
Đúng
AL = [BX][SI]
[7] <--- AL
SI++
SI <= 4 ?
Sai
Đúng
SI = 0
CX++
CX <= step_before_direction_change
Sai
Đúng
CX = 0
BX = BX+4
BX <= offset datccw_fs
Sai
Đúng
BX = offset datcw
Powered By Visual Paradigm Community Edition

2. Mã nguồn:



```
#start=stepper_motor.exe#
name "stepper"
#make_bin#

steps_before_direction_change = 20h ; 32 (decimal)

jmp start

; ========= data ===============

; bin data for clock-wise
; half-step rotation:
datcw     db 0000_0110b
          db 0000_0100b
          db 0000_0011b
          db 0000_0010b

; bin data for counter-clock-wise
; half-step rotation:
datccw    db 0000_0011b
          db 0000_0001b
          db 0000_0110b
          db 0000_0010b


; bin data for clock-wise
; full-step rotation:
datcw_fs  db 0000_0001b
          db 0000_0011b
          db 0000_0110b
          db 0000_0000b

; bin data for counter-clock-wise
; full-step rotation:
datccw_fs  db 0000_0100b
           db 0000_0110b
           db 0000_0011b
           db 0000_0000b


start:
mov bx, offset datcw
mov si, 0
mov cx, 0

next_step:
;
```



```
wait:
        in al, 7
        test al, 10000000b
        jz wait

mov al, [bx][si]
out 7, al

inc si

cmp si, 4
jb next_step
mov si, 0

inc cx
cmp cx, steps_before_direction_change
jb   next_step

mov cx, 0
add bx, 4

cmp bx, offset datccw_fs
jbe next_step

mov bx, offset datcw

jmp next_step
```