Group No: 48 **Project No: 12**

Assignment on
Microprocessor based Washing Machine

Submitted By
Ram Karnani 2012B3A7750P
Ashish Tilokani 2012B3A7563P
Akash Sharma 2012B2A7771P
Shubham Singh 2012B3A7456P

Submitted to:

In partial fulfillment of the course
Microprocessor Programming and Interfacing (EEE F241)

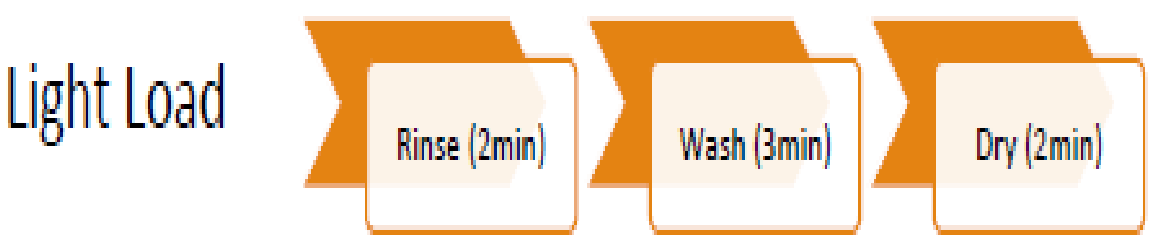
INDEX

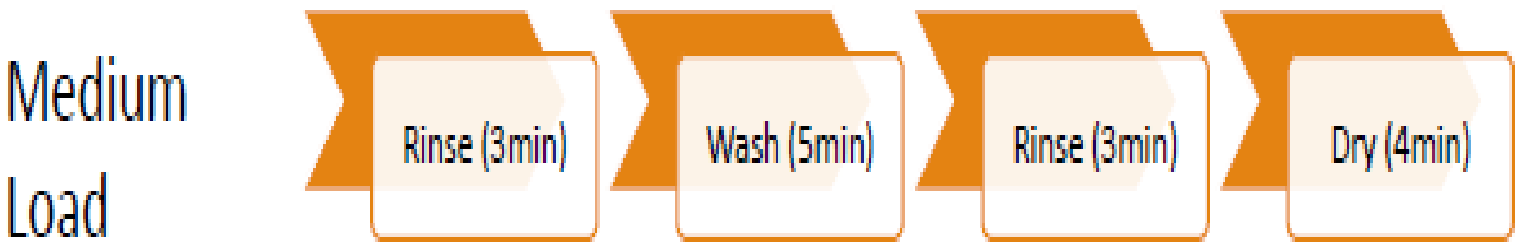
ACKNOWLEDGEMENT

We would like to express our deepest appreciation to Mr. G Sai Sesha Chalapathi, Mr. Ashish Mishra, Dr. S Gurunarayanan, and Mr. Sainath and all the microprocessors and interfacing faculty for their everlasting support without which this project would not have been possible. We would also like to express our heartfelt gratitude to Dr Anupama.K.R, without whom the experience would not have been as enriching and memorable.

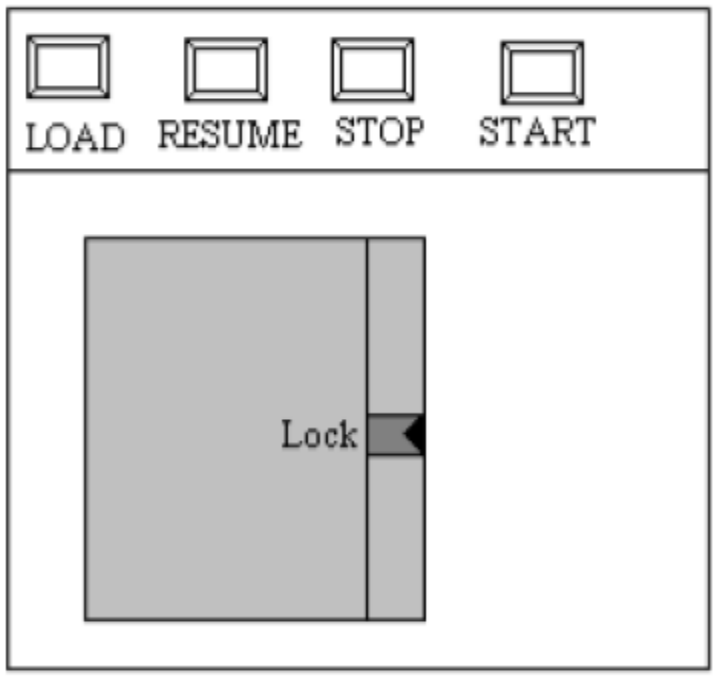
Problem Specification: Automatic Washing Machine

Three different types of load: Light, Medium and Heavy
Three different cycles: Rinse, Wash and Dry
Depending on the load the number of times a cycle is done and the duration of the cycle varies.





User Interface



The number of times the load button is pressed determines load :
1press- light
2 presses – medium
3 presses –heavy

To begin washing process START is pressed

Pressing STOP can stop the process

- The Washing Machine is a single tub machine.
- The Washing machine is made of a Revolving Tub and an Agitator. The Agitator is activated during the Rinse and Wash cycle; revolving tub is active only during the Dry cycle. The door of the washtub should remain closed as long as the agitator is active.
- Before each cycle the water level is sensed. At the beginning of the cycle the water level

should be at the maximum possible level, the water should be completely drained during dry cycle. The cycle should begin only when the water level is correct.

- At the end of each cycle a buzzer is activated. The user should drain the water at the

end of the rinse/wash cycle and refill the water for the next cycle; once this has been completed the user can press the resume button.

- At the beginning of the wash cycle the user should add the detergent.
- At the end of the complete wash process the Buzzer is sounded.
- User can turn off system by pressing STOP Button
- Different sounds are used for different events.

ASSUMPTIONS

1. The user can press the load button either once, twice or thrice before pressing the start button.

2. The maximum height of the Water level in the washing machine is 1 meter.

3. In order to check whether the water level in the machine is full or empty, we have used metal contacts.

4. Three different LEDS are used to demonstrate the working of different cycles- RINSE, WASH AND DRY.

5. An LED is used to demonstrate the open door.

6. LEDS are used in the common cathode configuration.

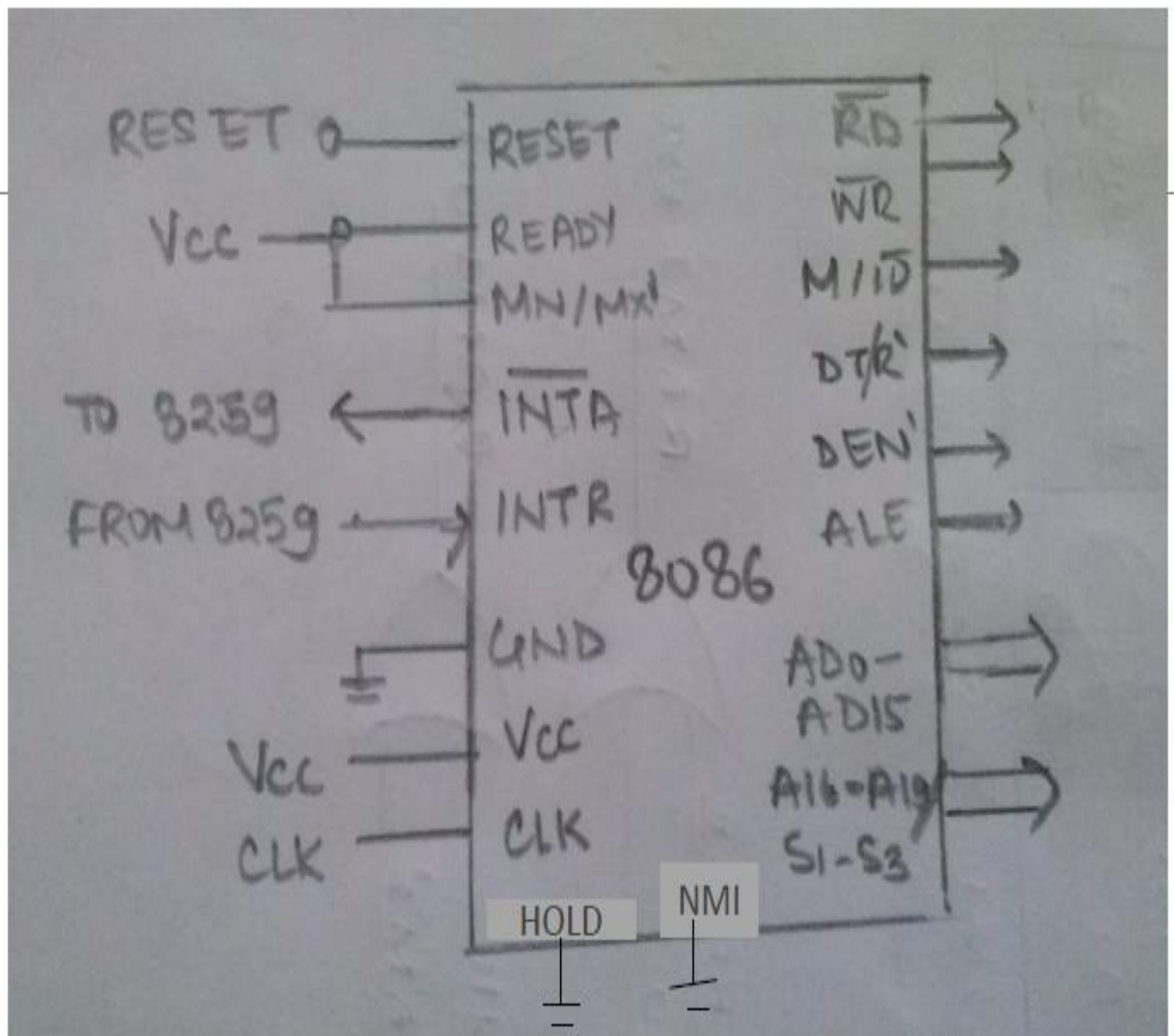
7. We have assumed that a 2 min cycle corresponds to 10 seconds. Hence, one minute of a cycle corresponds to 5 seconds. This has been done to decrease the waiting time for the results.

8. The sensors have been shown in Proteus as LEDs. We have tried to stimulate the idea of sensor, to detect water level, using 2 metal conductors connected to port of 8255.

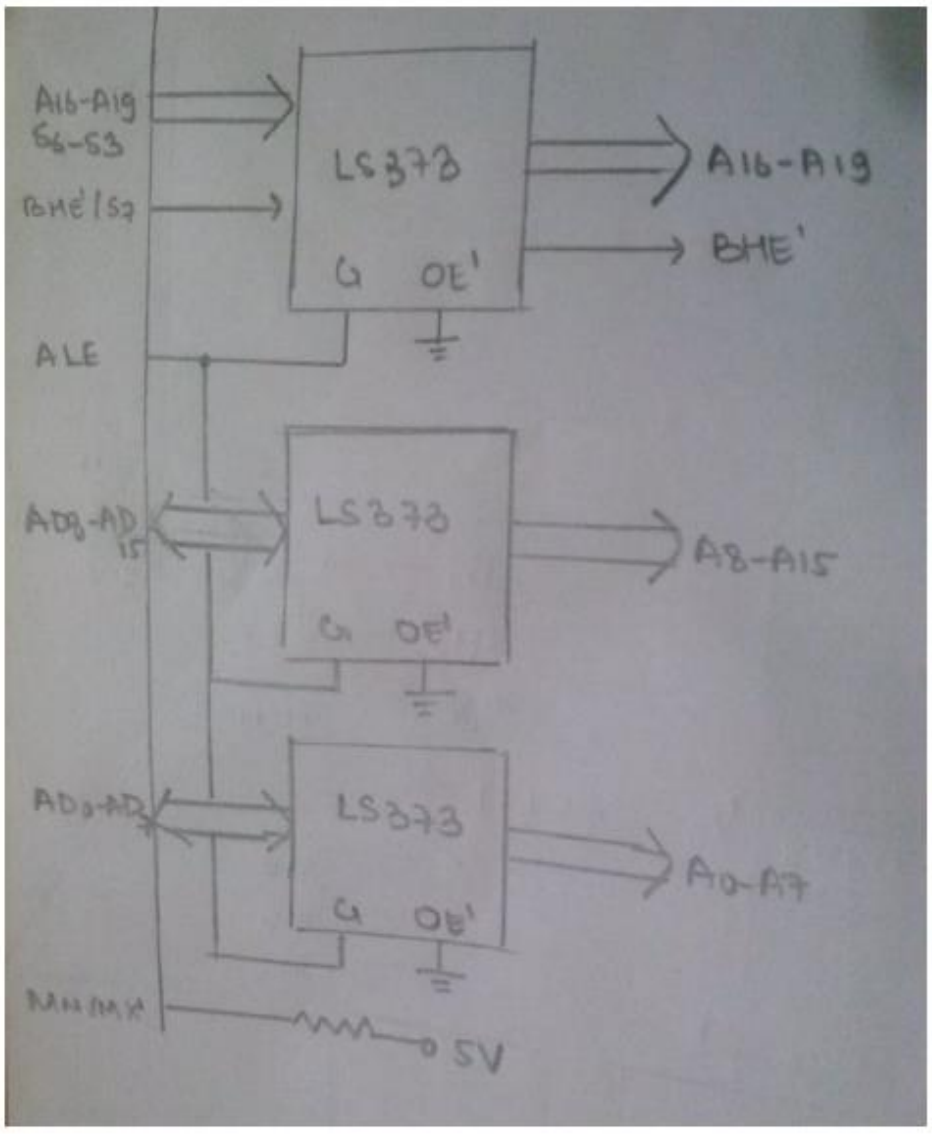
List of the components used

SR. NO	HARDWARE	CHIP NO.	NUMBER
1.	Microprocessor: The programming unit which executes the program and controls the other units of the system.	INTEL 8086	1
2.	Read Only Memory	2732(4k x2)	2
3.	Random Access Memory Data storage	6116(2K x2)	2
4.	Programmable Peripheral Interface: The interfacing device which connects the buffered Micro Processor to the I/O devices	INTEL 8255	2
5.	PIC-Programmable Interrupt Controller Adds 8 vectored priority encoded interrupts to the microprocessor.	8259	1
6.	Octal Bus Transceiver	74LS245	2
7.	Octal Latches	74LS373	3
8.	Input Buzzer		3
9.	Programmable Interval Timer	8253	1

HARDWARE:

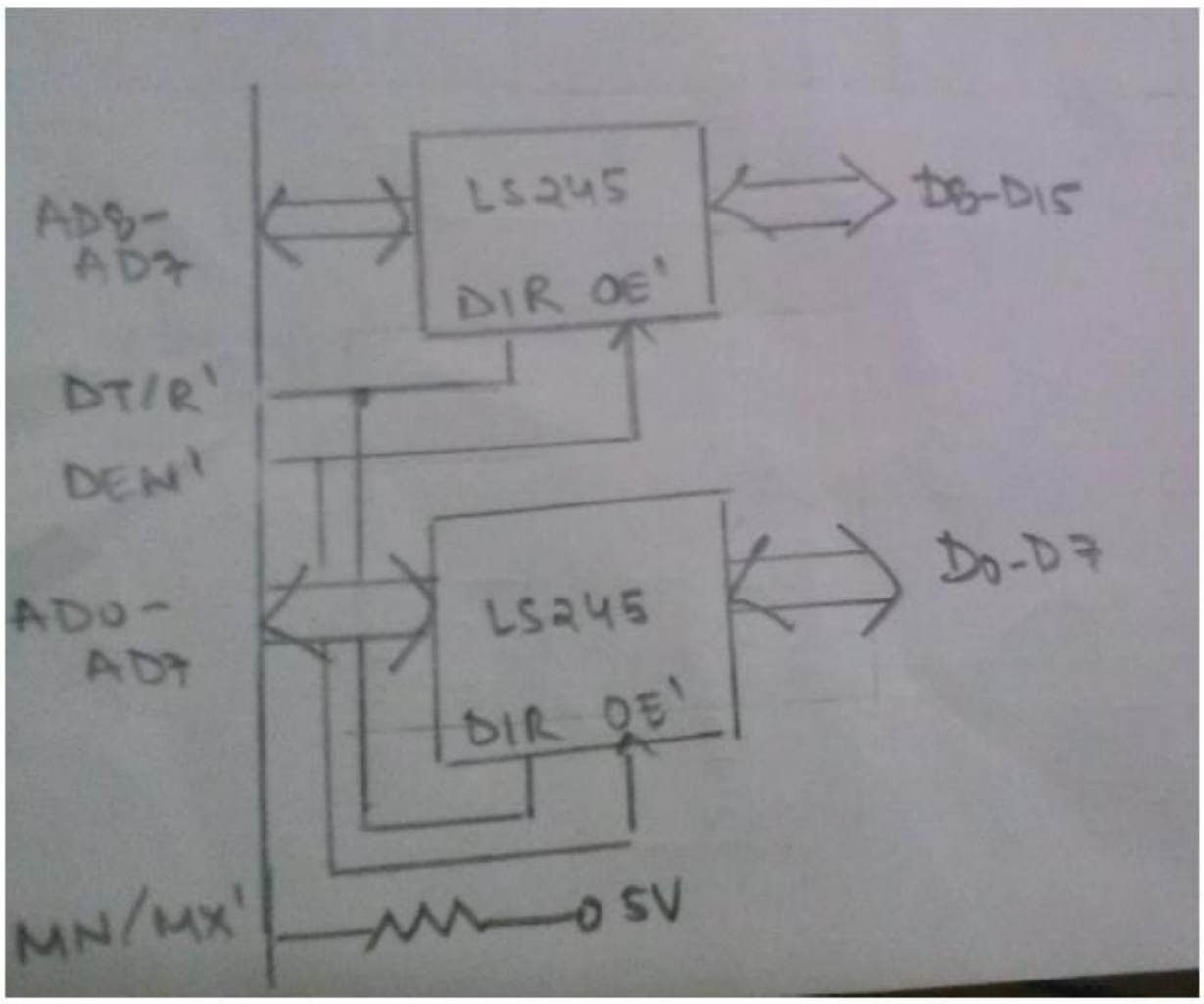

8086
RESET
Vcc
TO 8259
FROM 8259
Vcc
CLK
RESET
READY
MN/MX'
INTA
INTR
GND
Vcc
CLK
HOLD
NMI
8086
RD
WR
M/IO
DT/R'
DEN'
ALE
AD0-AD15
A16-A19
S1-S3

8086



System Bus of 8086 (Address)

8086



System Bus of 8086 (Data)

Interfacing 8255, 8253 and 8259

Address:

8255(1): 00H TO 06H

8255(2): 08H TO 0EH

8253: 10H TO 16H

8259: 18H TO 1AH

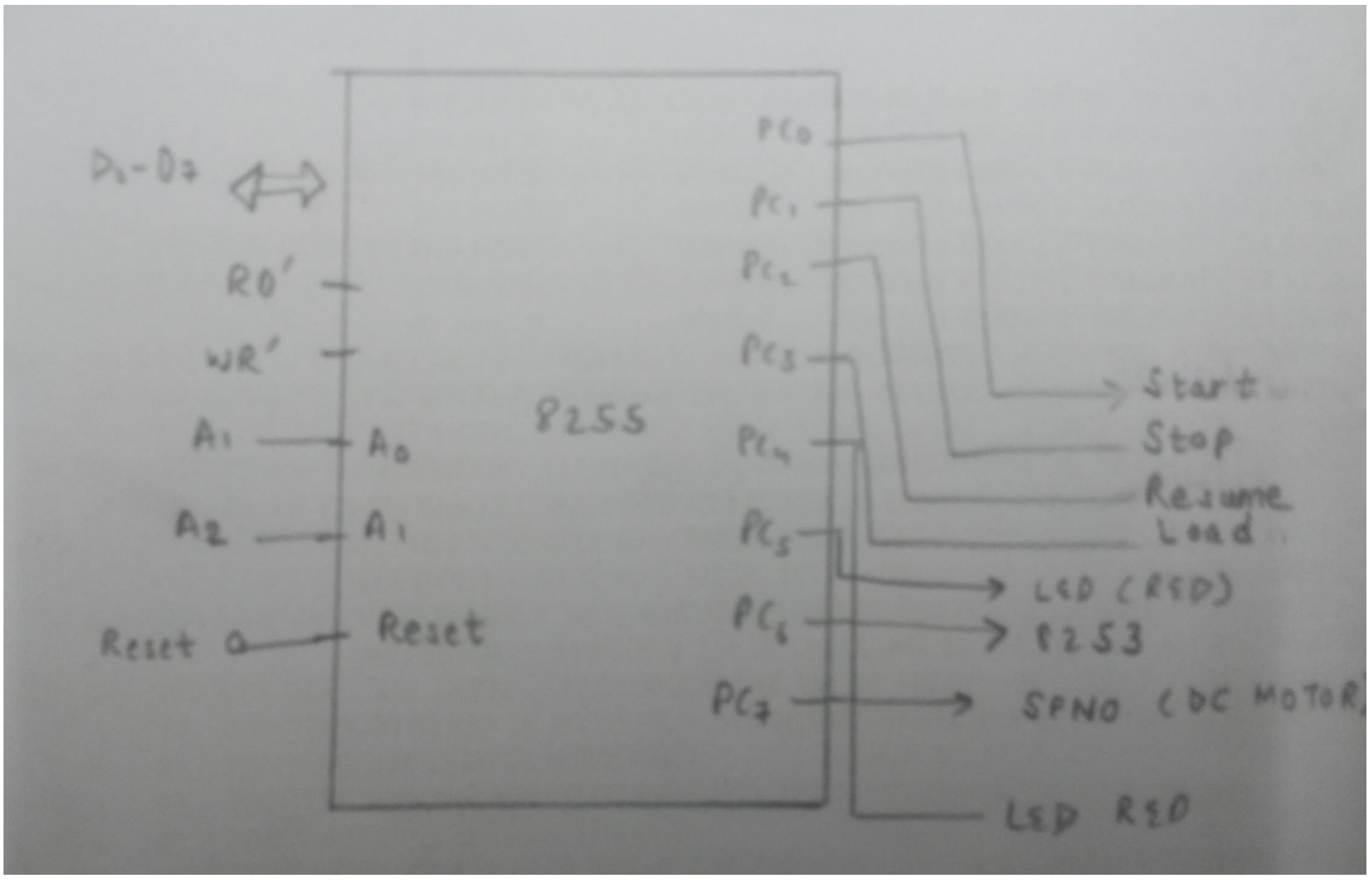
Incremental Addressing

Memory Map:

DEVICE	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
6116																				
SRAM1																				
From 02000h	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
To 02FFEh	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0
2732																				
EPROM1																				
From 00000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
To 01FFEh	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0
6116																				
SRAM2																				
From 02001h	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
To 02FFFh	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
2732																				
EPROM2																				
From 00001h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
To 01FFFh	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

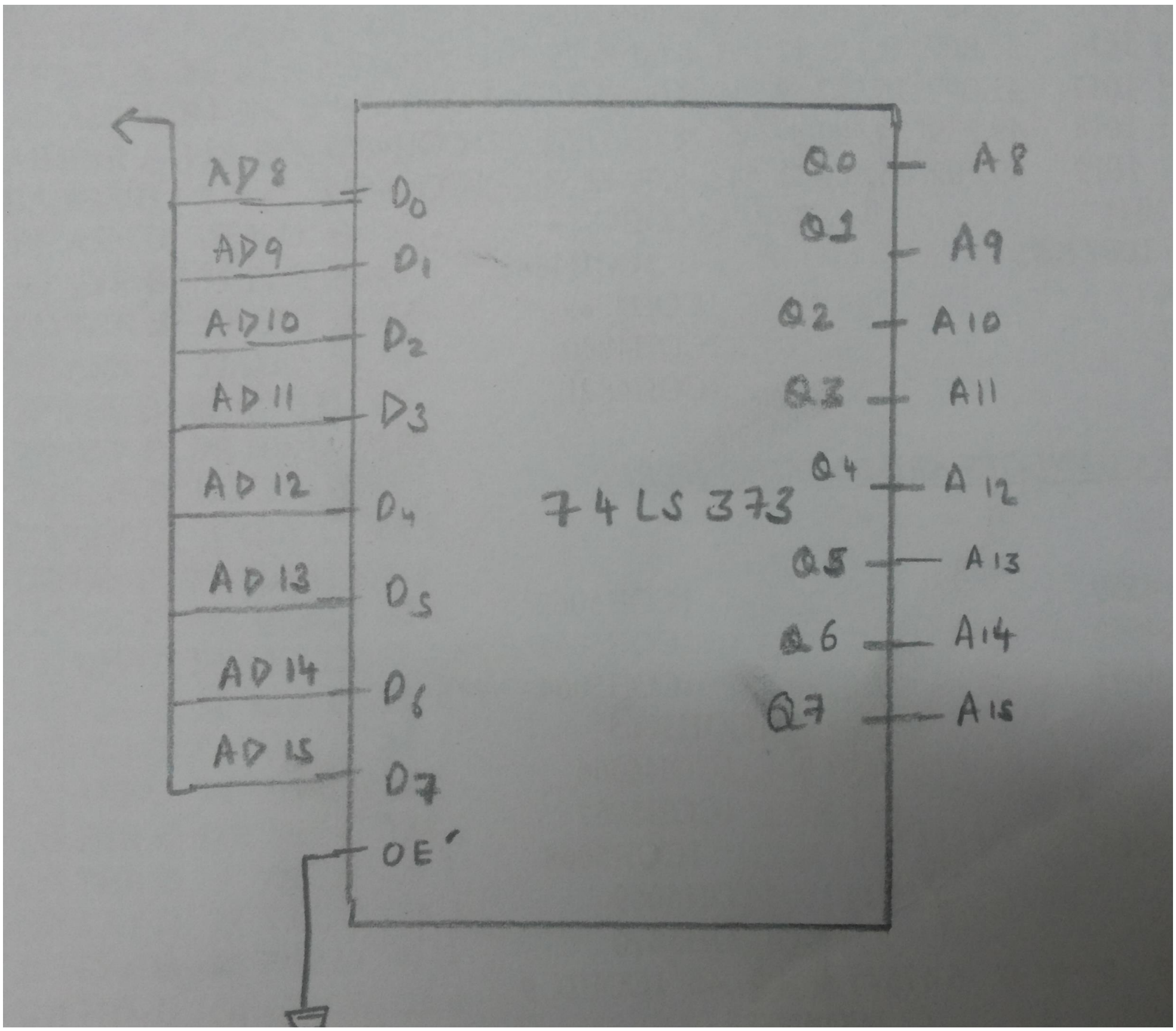
8255(1)

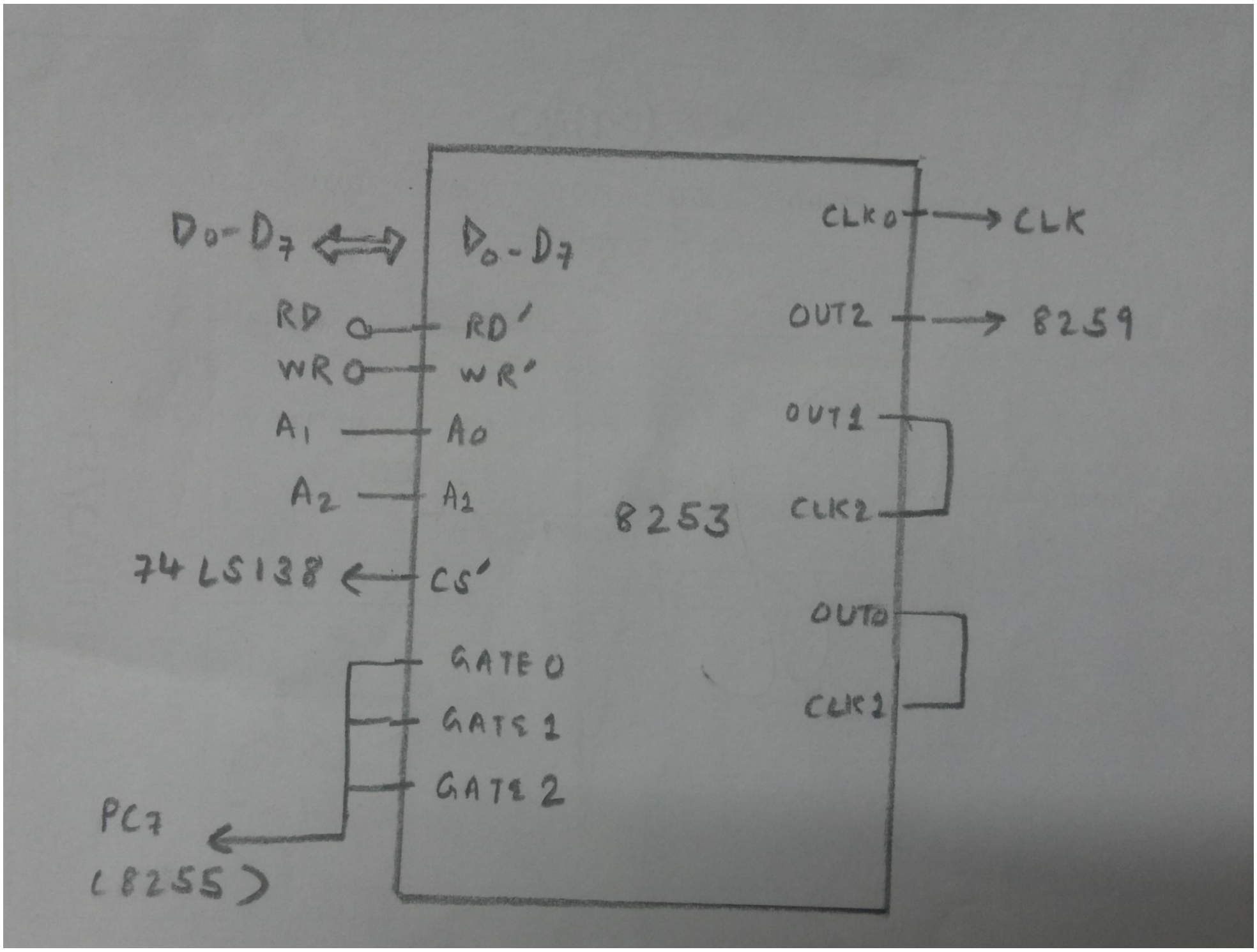
(PPI)


D0-D7
RD'
WR'
A1
A0
A2
A1
Reset
Reset
8255
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
Start
Stop
Resume
Load
LED (RED)
8253
SPNO (DC MOTOR)
LED RED

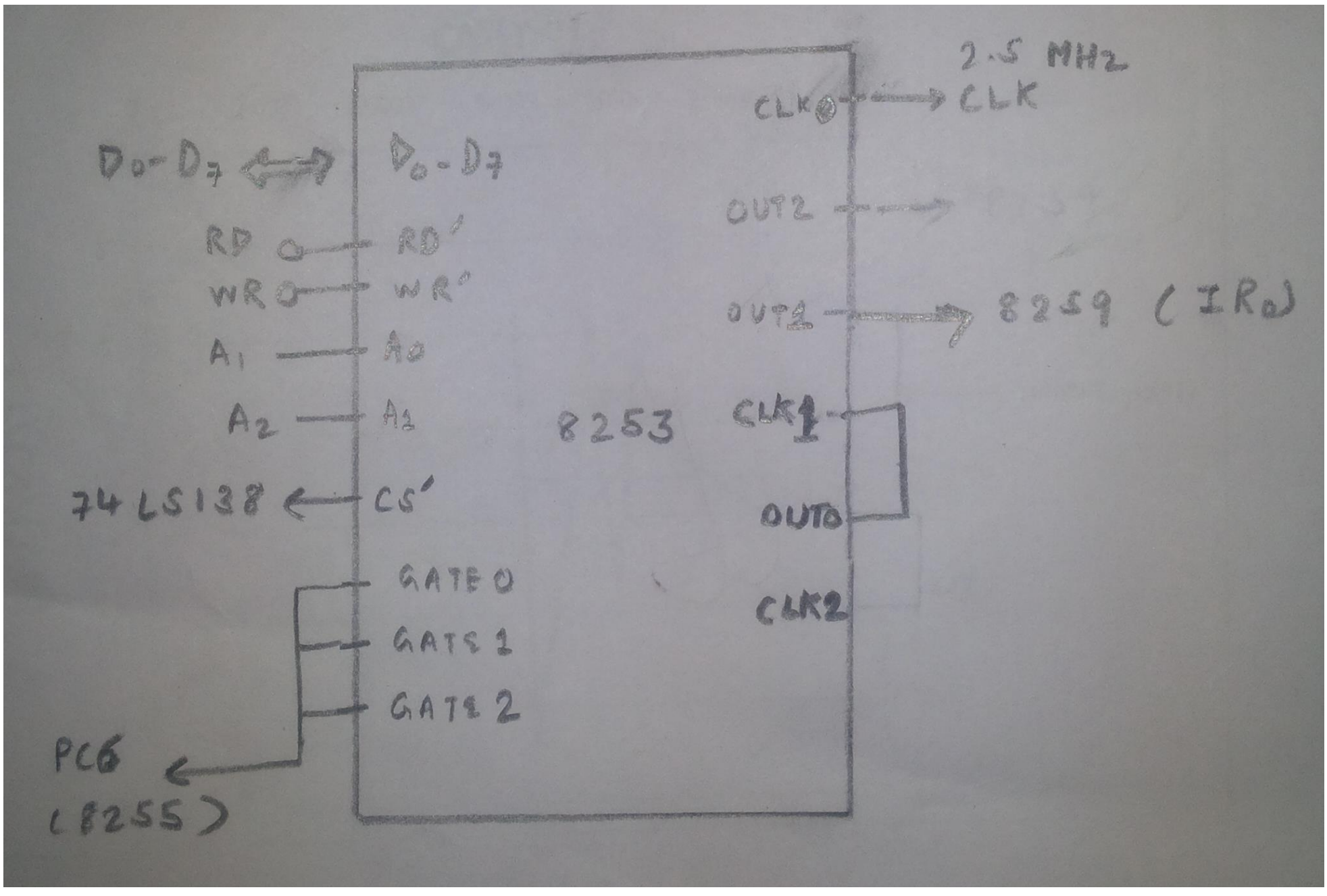
74LS373

(OCTAL LATCH)



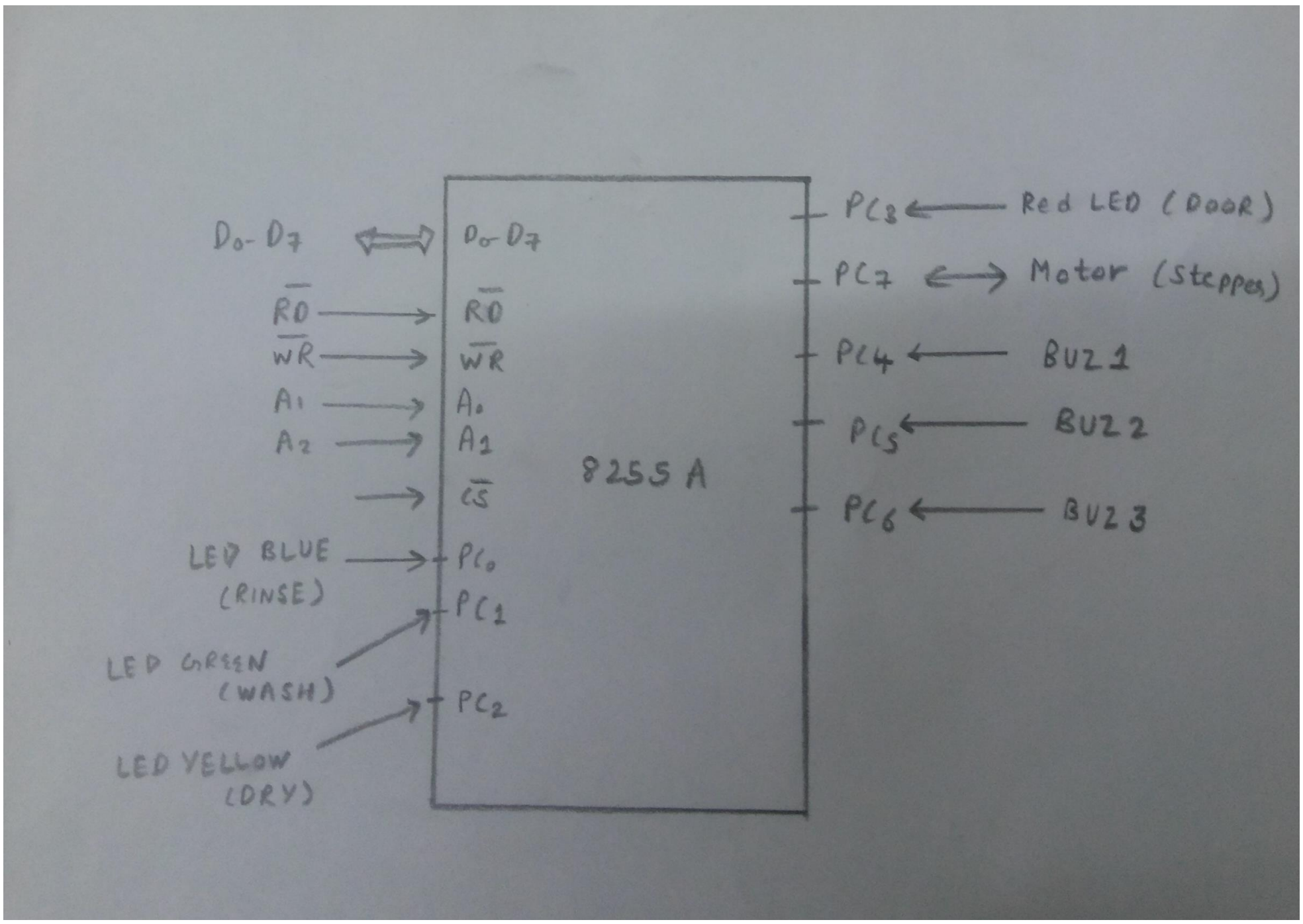

D0-D7
D0-D7
RD
RD'
WR
WR'
A1
A0
A2
A1
8253
74LS138
CS'
GATE 0
GATE 1
GATE 2
PC7
(8255)
CLK0
CLK
OUT2
8259
OUT1
CLK2
OUT0
CLK1

8253(Timer)

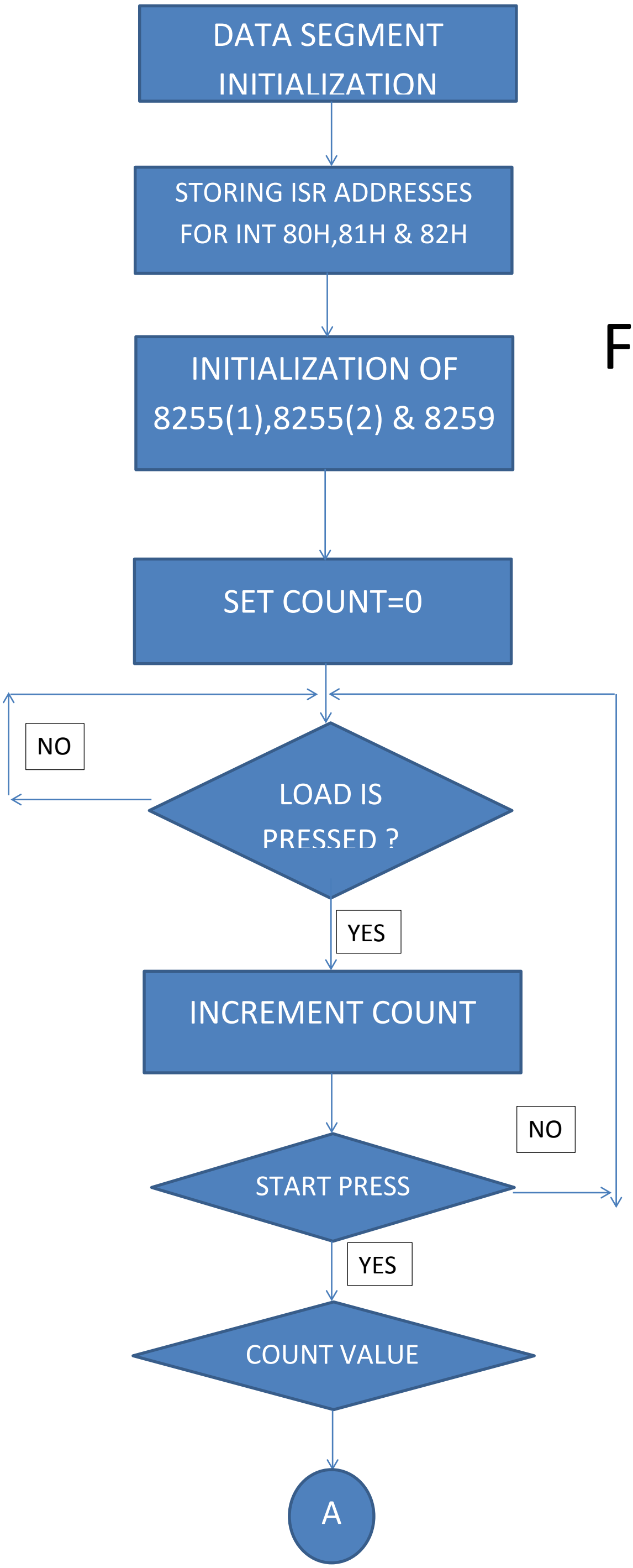

2.5 MHz
CLK
CLK
D0-D7
D0-D7
OUT2
RD
RD'
WR
WR'
OUT2
8259 (IR0)
A1
A0
A2
A1
8253
CLK1
74LS138
CS'
OUT0
GATE 0
CLK2
GATE 1
GATE 2
PC6
(8255)

8255(2)

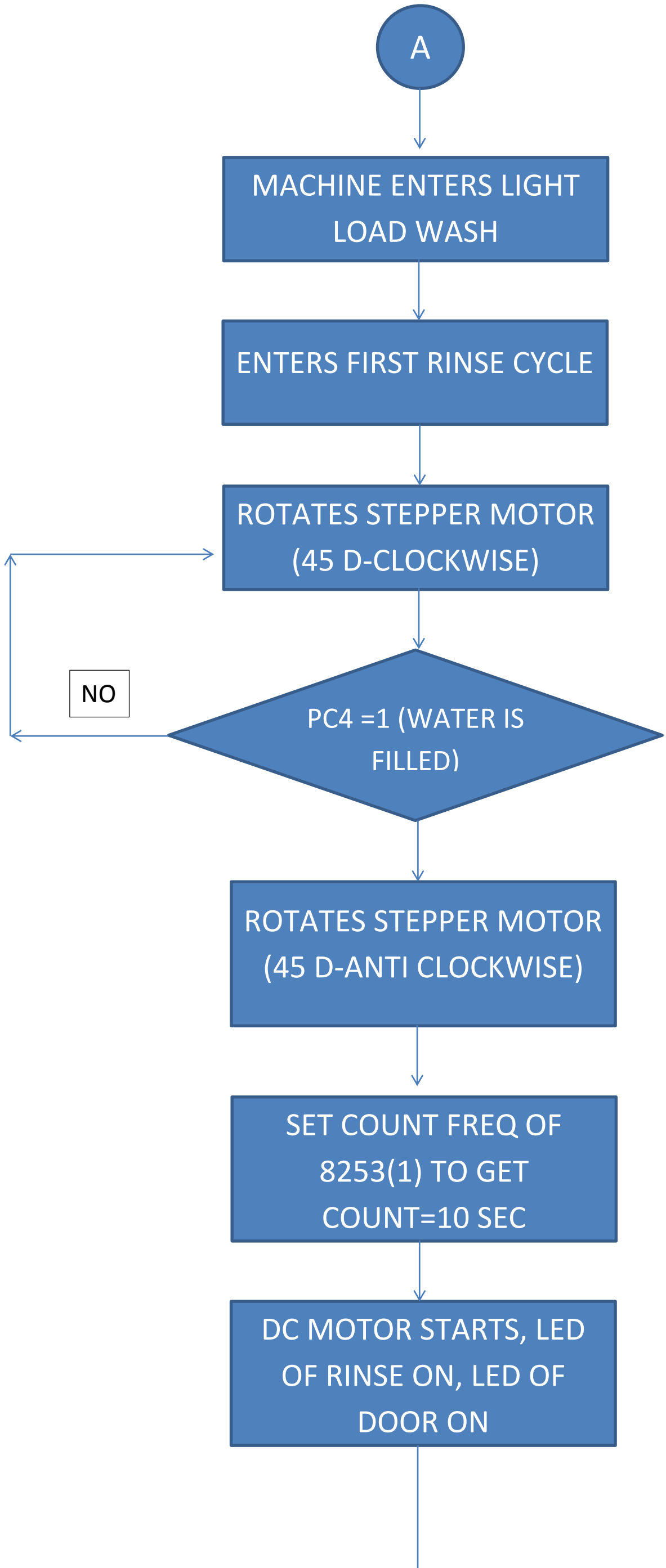
(PPI)

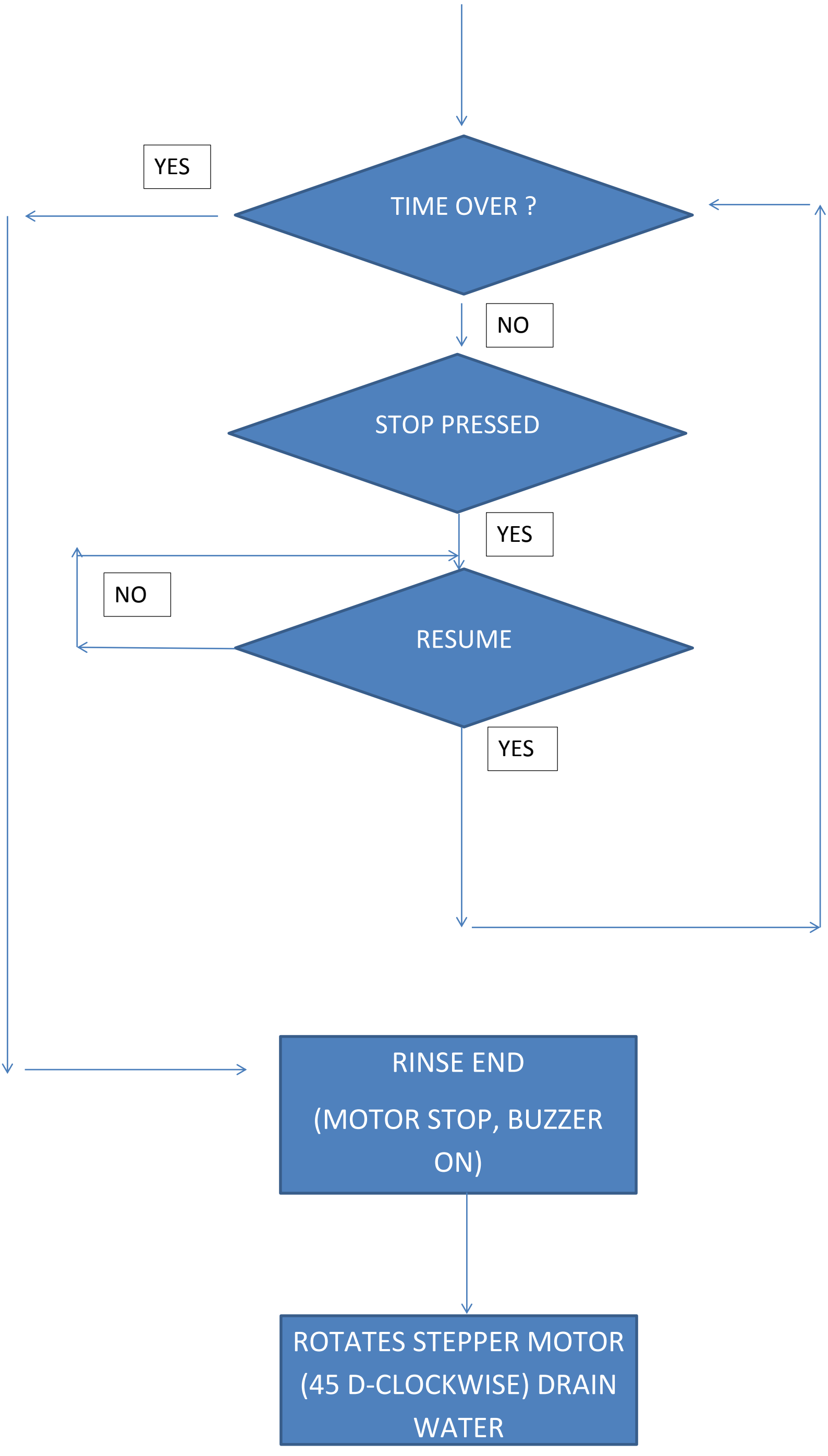

D0-D7
D0-D7
RD
WR
A1
A2
A0
A1
CS
8255 A
LED BLUE (RINSE)
PC0
PC1
LED GREEN (WASH)
PC2
LED YELLOW (DRY)
PC3 ← Red LED (DOOR)
PC7 ↔ Motor (Steppes)
PC4 ← BUZ 1
PC5 ← BUZ 2
PC6 ← BUZ 3

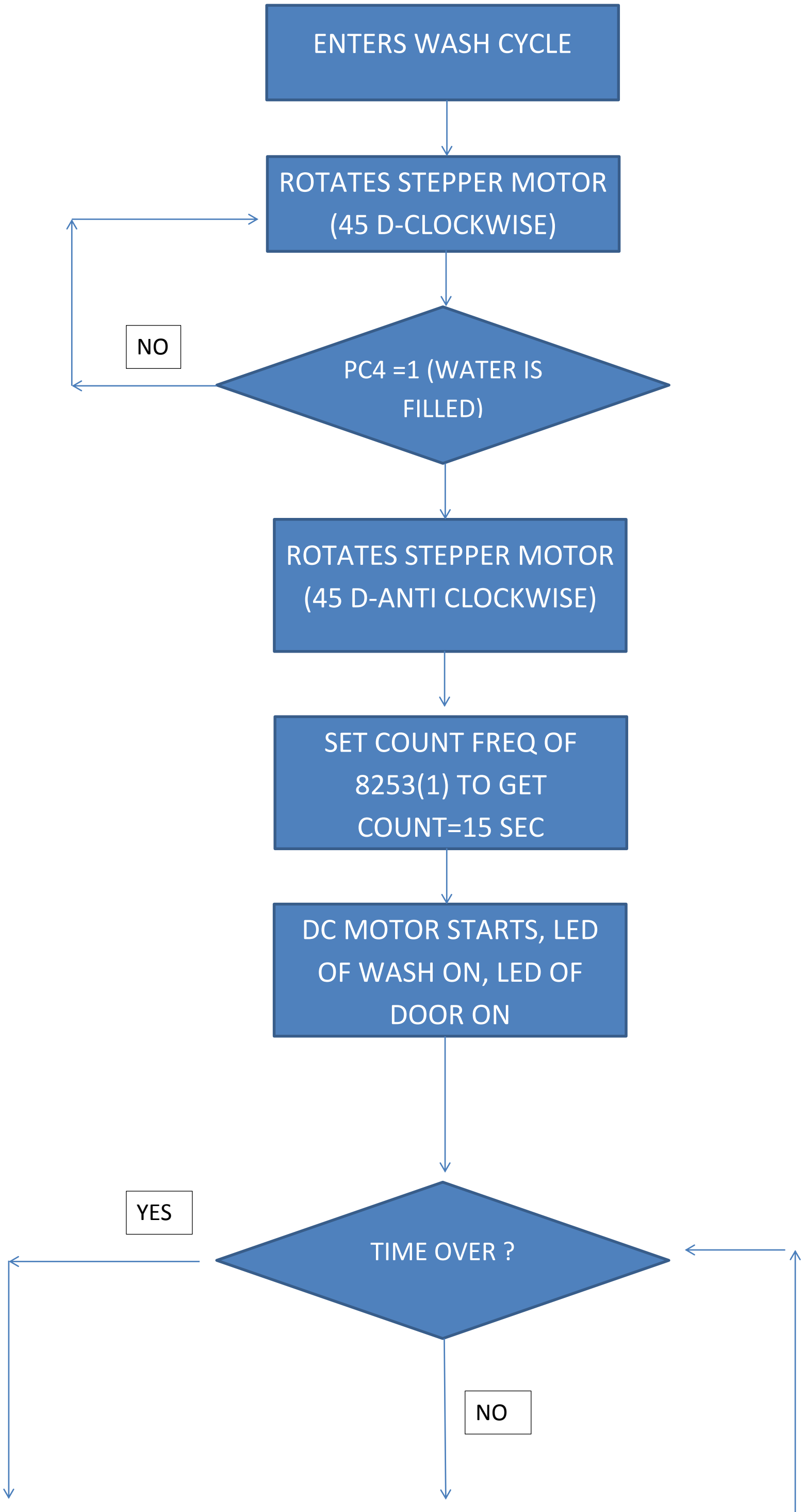
FLOWCHART

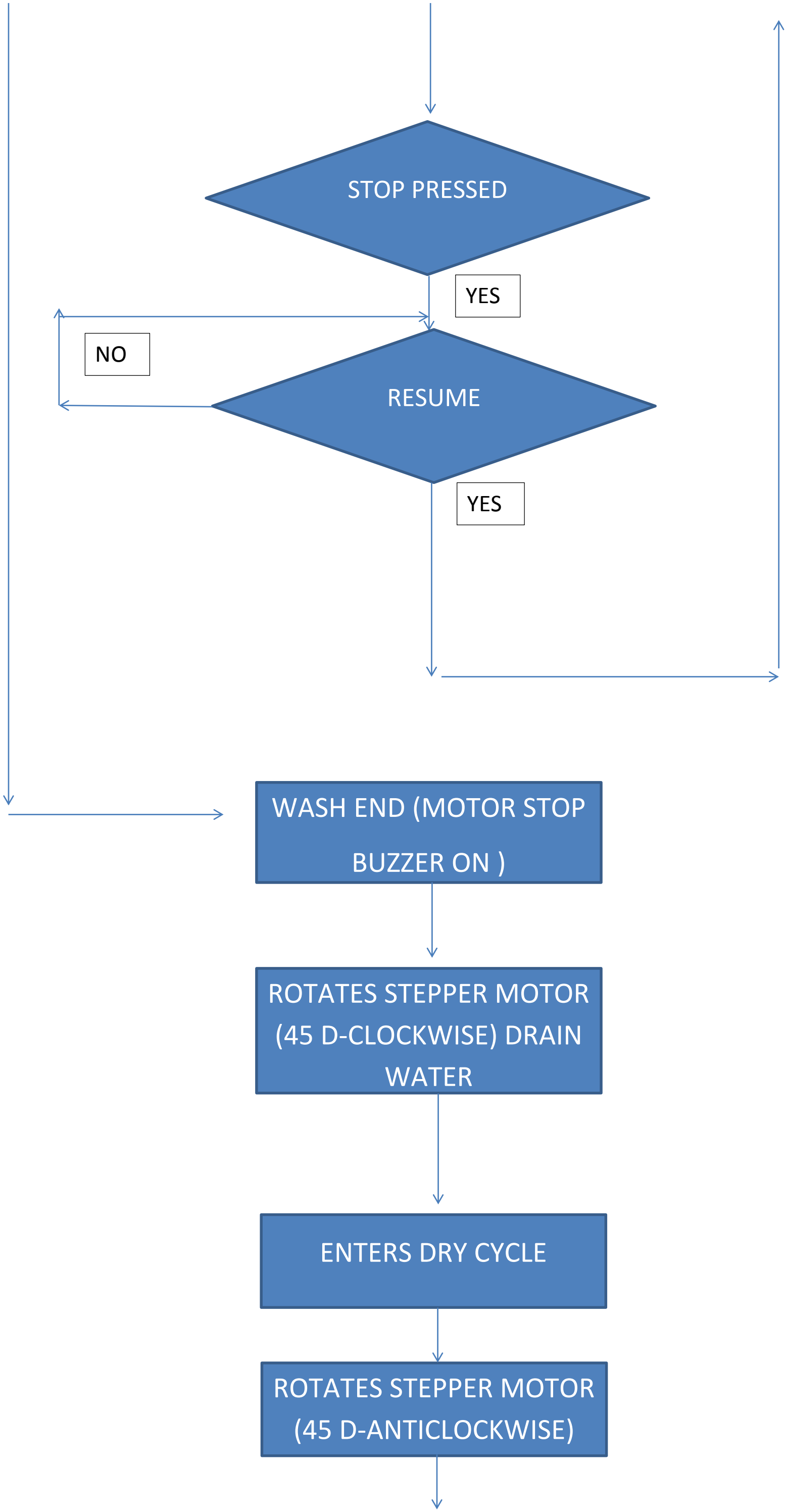

DATA SEGMENT INITIALIZATION
STORING ISR ADDRESSES FOR INT 80H,81H & 82H
INITIALIZATION OF 8255(1),8255(2) & 8259
SET COUNT=0
NO
LOAD IS PRESSED ?
YES
INCREMENT COUNT
NO
START PRESS
YES
COUNT VALUE
A

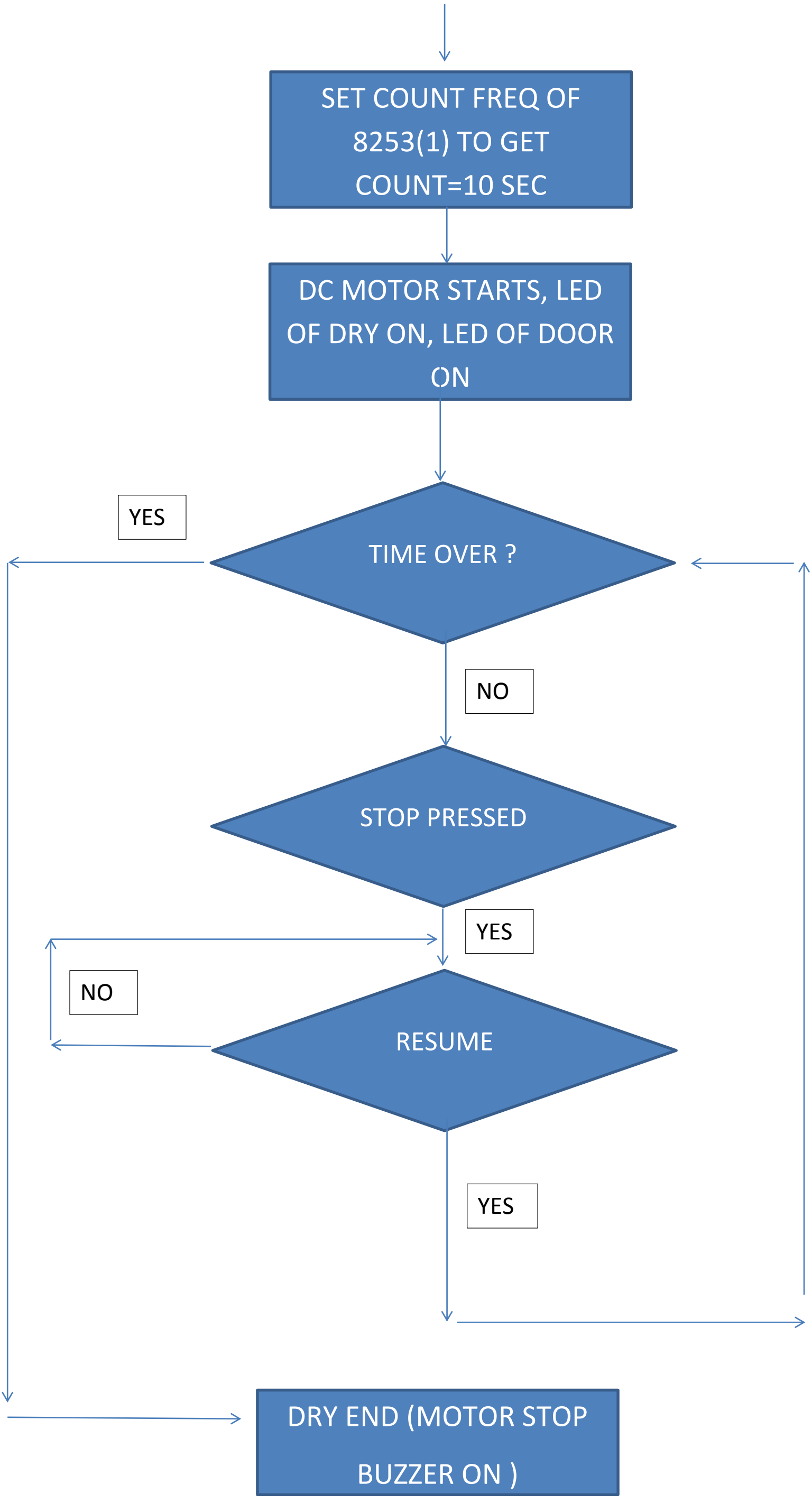
LIGHT LOAD (COUNT= 1)



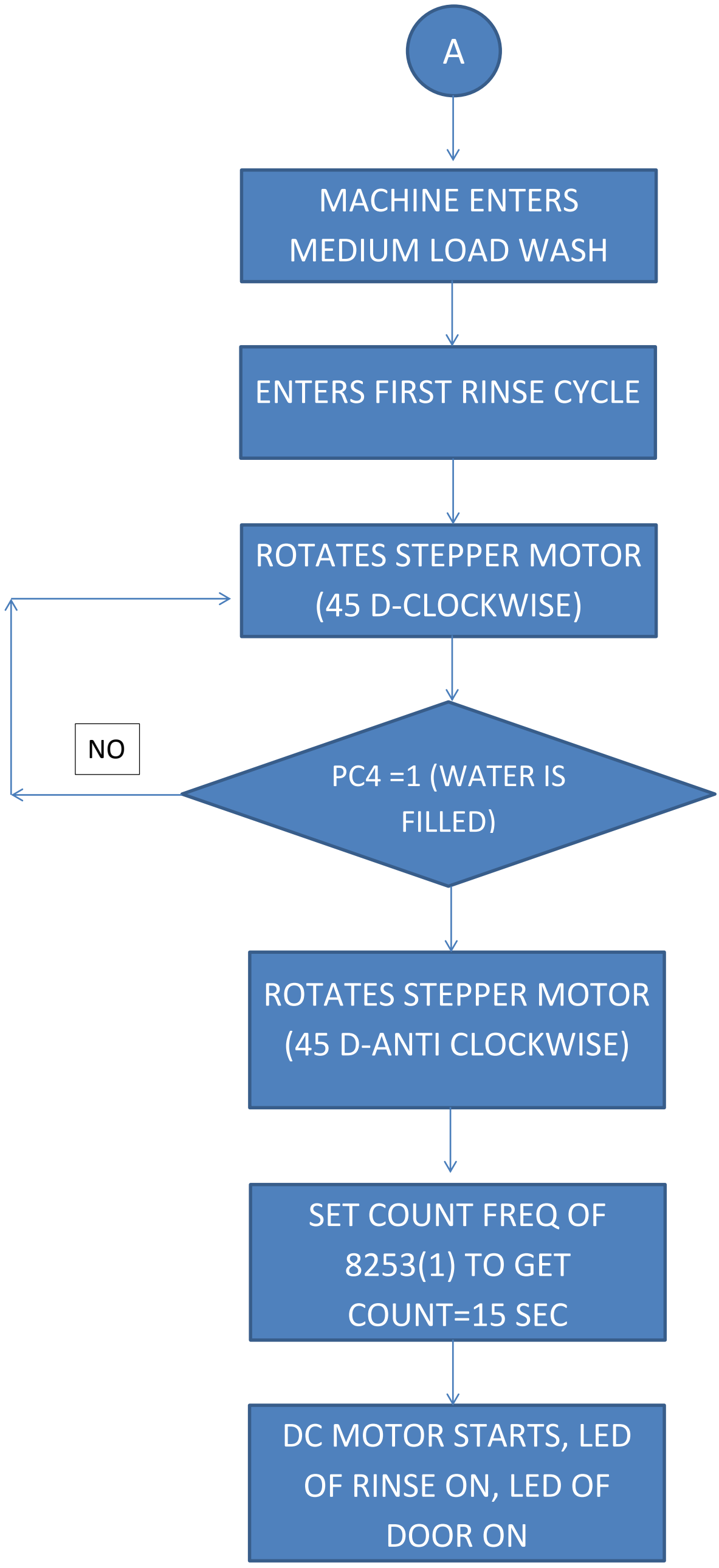

TIME OVER ?
YES
NO
STOP PRESSED
YES
RESUME
NO
YES
RINSE END
(MOTOR STOP, BUZZER ON)
ROTATES STEPPER MOTOR (45 D-CLOCKWISE) DRAIN WATER

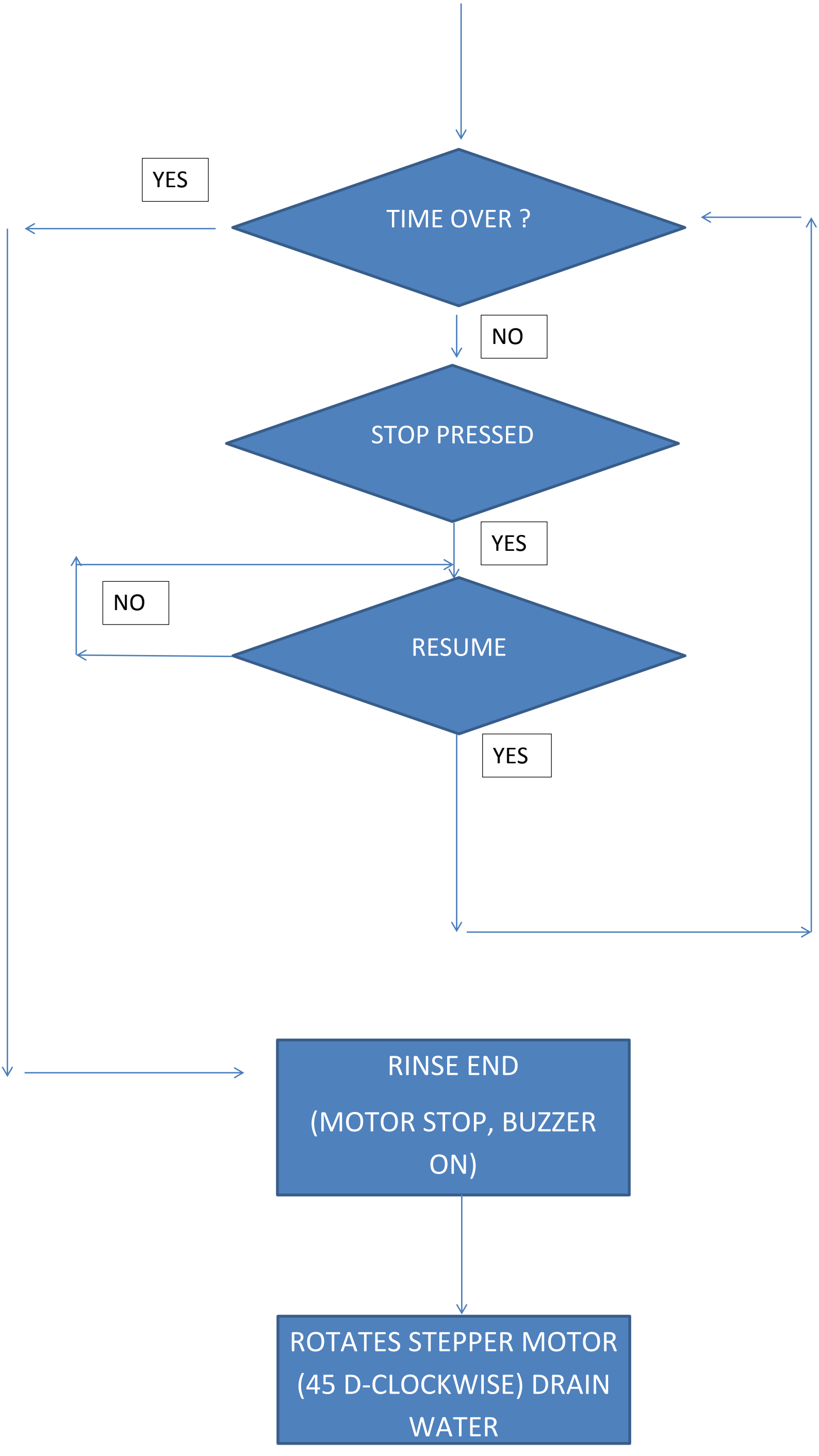

ENTERS WASH CYCLE
ROTATES STEPPER MOTOR (45 D-CLOCKWISE)
NO
PC4 =1 (WATER IS FILLED)
ROTATES STEPPER MOTOR (45 D-ANTI CLOCKWISE)
SET COUNT FREQ OF 8253(1) TO GET COUNT=15 SEC
DC MOTOR STARTS, LED OF WASH ON, LED OF DOOR ON
YES
TIME OVER ?
NO

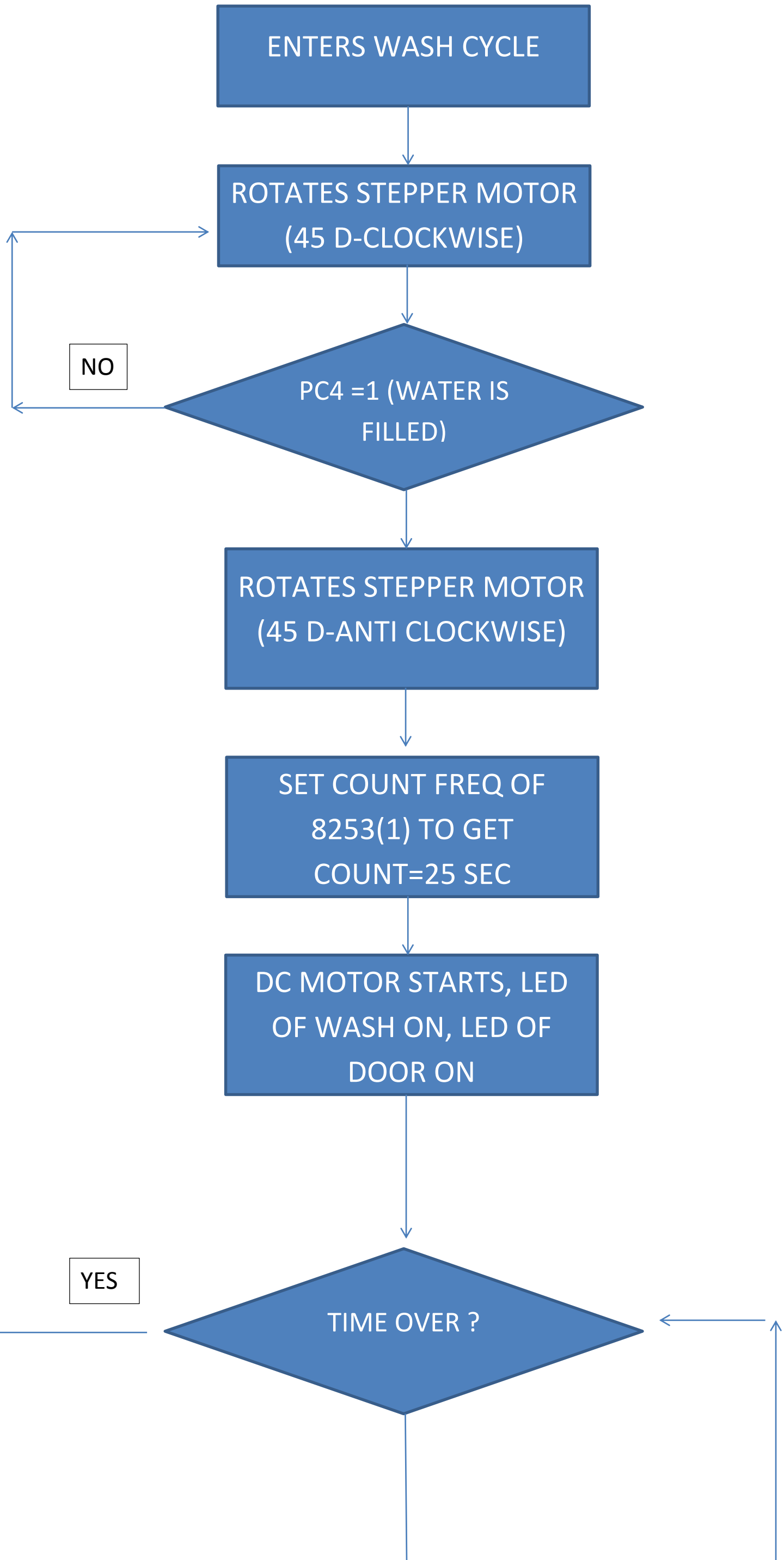

STOP PRESSED
YES
NO
RESUME
YES
WASH END (MOTOR STOP
BUZZER ON)
ROTATES STEPPER MOTOR
(45 D-CLOCKWISE) DRAIN
WATER
ENTERS DRY CYCLE
ROTATES STEPPER MOTOR
(45 D-ANTICLOCKWISE)

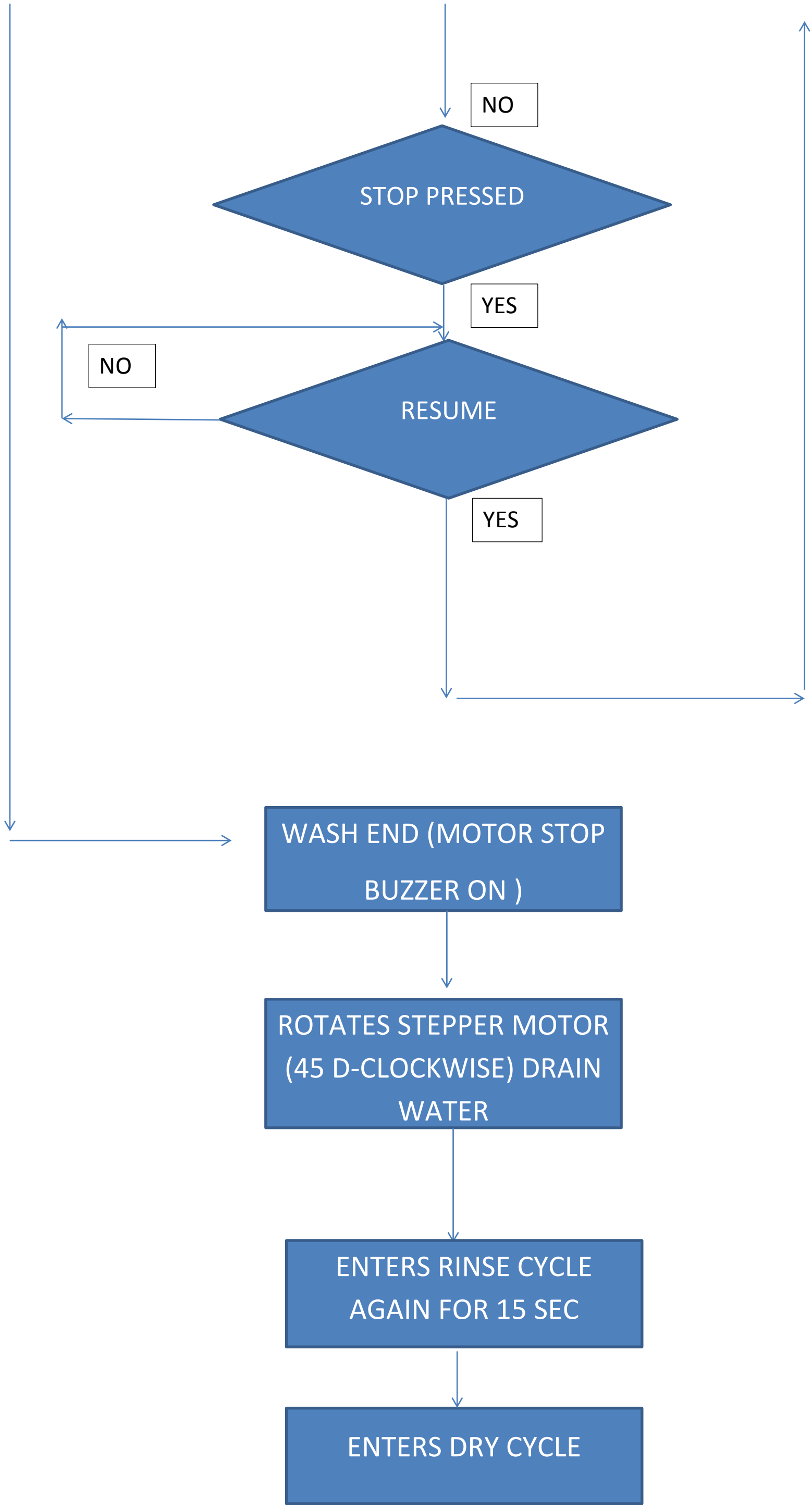

SET COUNT FREQ OF 8253(1) TO GET COUNT=10 SEC
DC MOTOR STARTS, LED OF DRY ON, LED OF DOOR ON
TIME OVER ?
YES
NO
STOP PRESSED
YES
RESUME
NO
YES
DRY END (MOTOR STOP BUZZER ON)

MEDIUM LOAD (COUNT= 2)




YES
TIME OVER ?
NO
STOP PRESSED
YES
NO
RESUME
YES
RINSE END
(MOTOR STOP, BUZZER ON)
ROTATES STEPPER MOTOR (45 D-CLOCKWISE) DRAIN WATER


ENTERS WASH CYCLE
ROTATES STEPPER MOTOR (45 D-CLOCKWISE)
NO
PC4 =1 (WATER IS FILLED)
ROTATES STEPPER MOTOR (45 D-ANTI CLOCKWISE)
SET COUNT FREQ OF 8253(1) TO GET COUNT=25 SEC
DC MOTOR STARTS, LED OF WASH ON, LED OF DOOR ON
YES
TIME OVER ?


NO
STOP PRESSED
YES
NO
RESUME
YES
WASH END (MOTOR STOP BUZZER ON)
ROTATES STEPPER MOTOR (45 D-CLOCKWISE) DRAIN WATER
ENTERS RINSE CYCLE AGAIN FOR 15 SEC
ENTERS DRY CYCLE

ROTATES STEPPER MOTOR (45 D-ANTICLOCKWISE)

SET COUNT FREQ OF 8253(1) TO GET COUNT=20 SEC

DC MOTOR STARTS, LED OF DRY ON, LED OF DOOR ON

TIME OVER ?

YES

NO

STOP PRESSED

YES

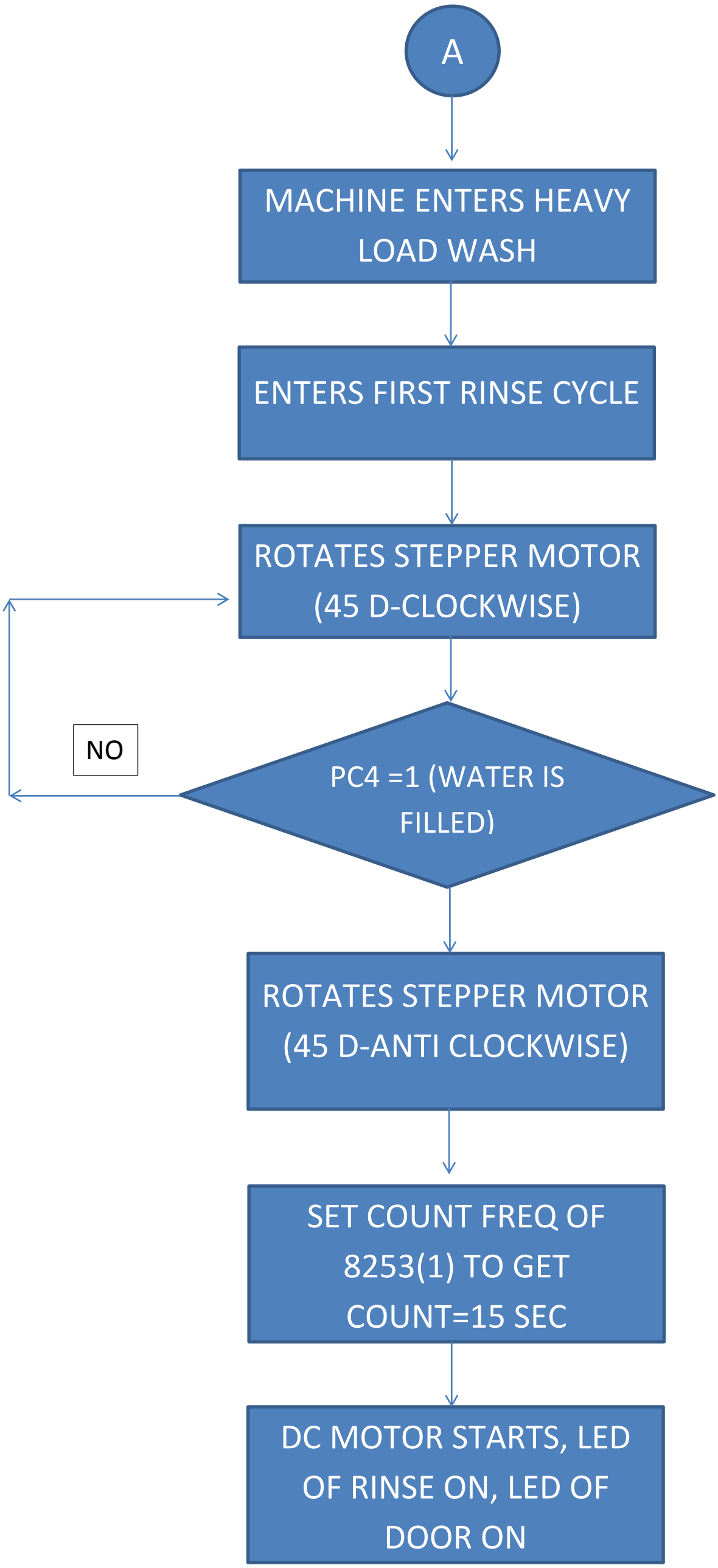
NO

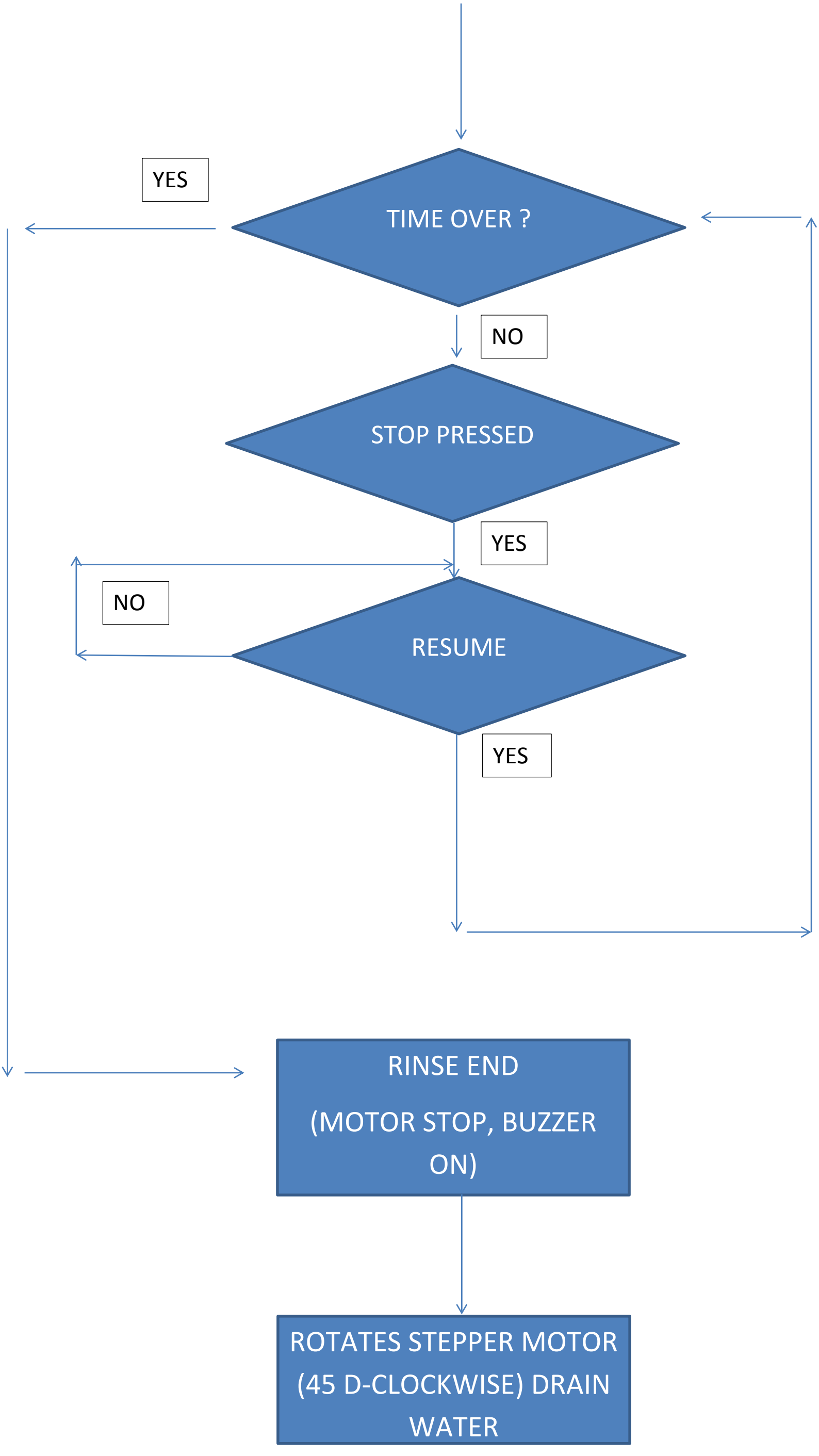
RESUME

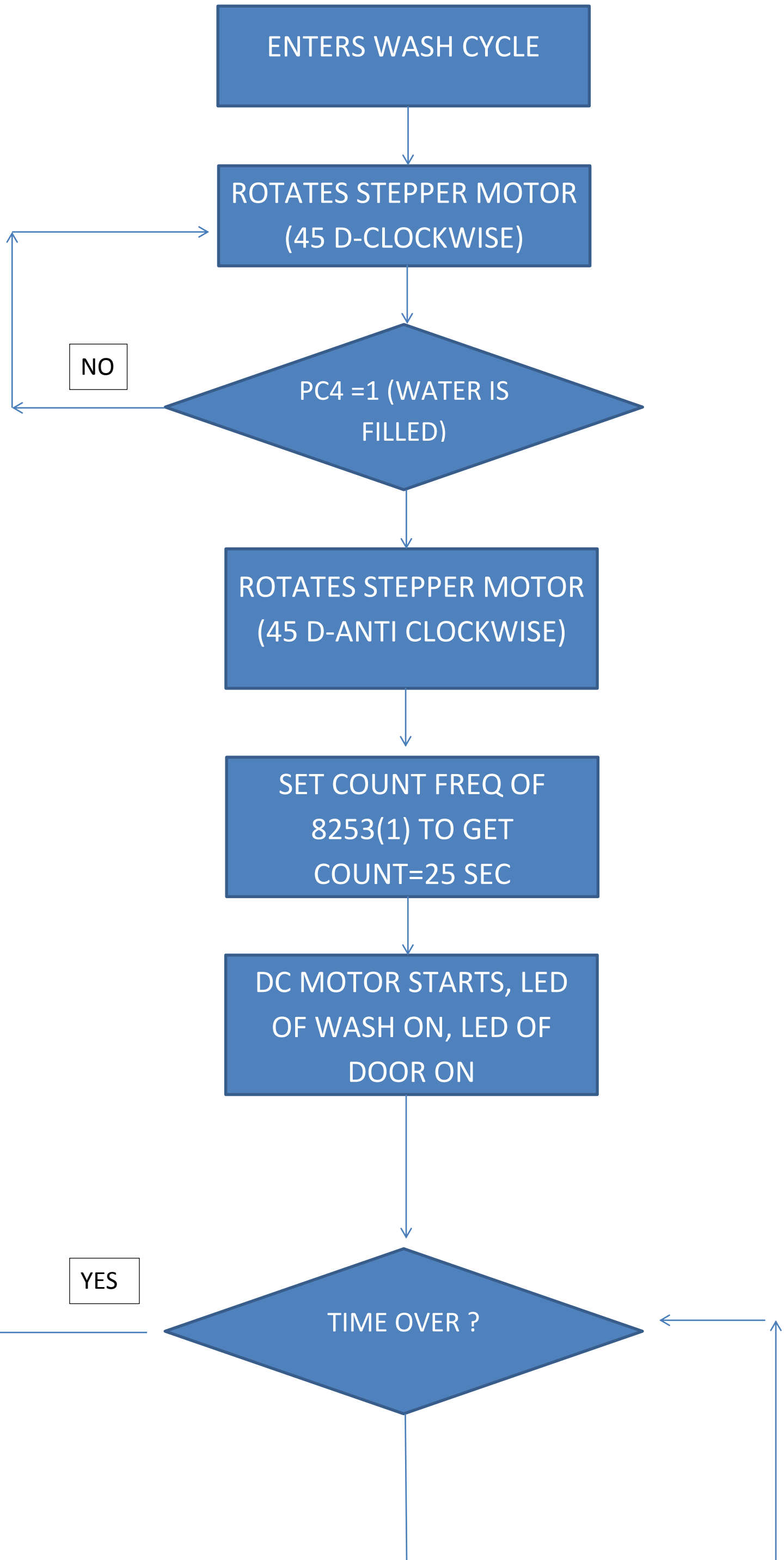
YES

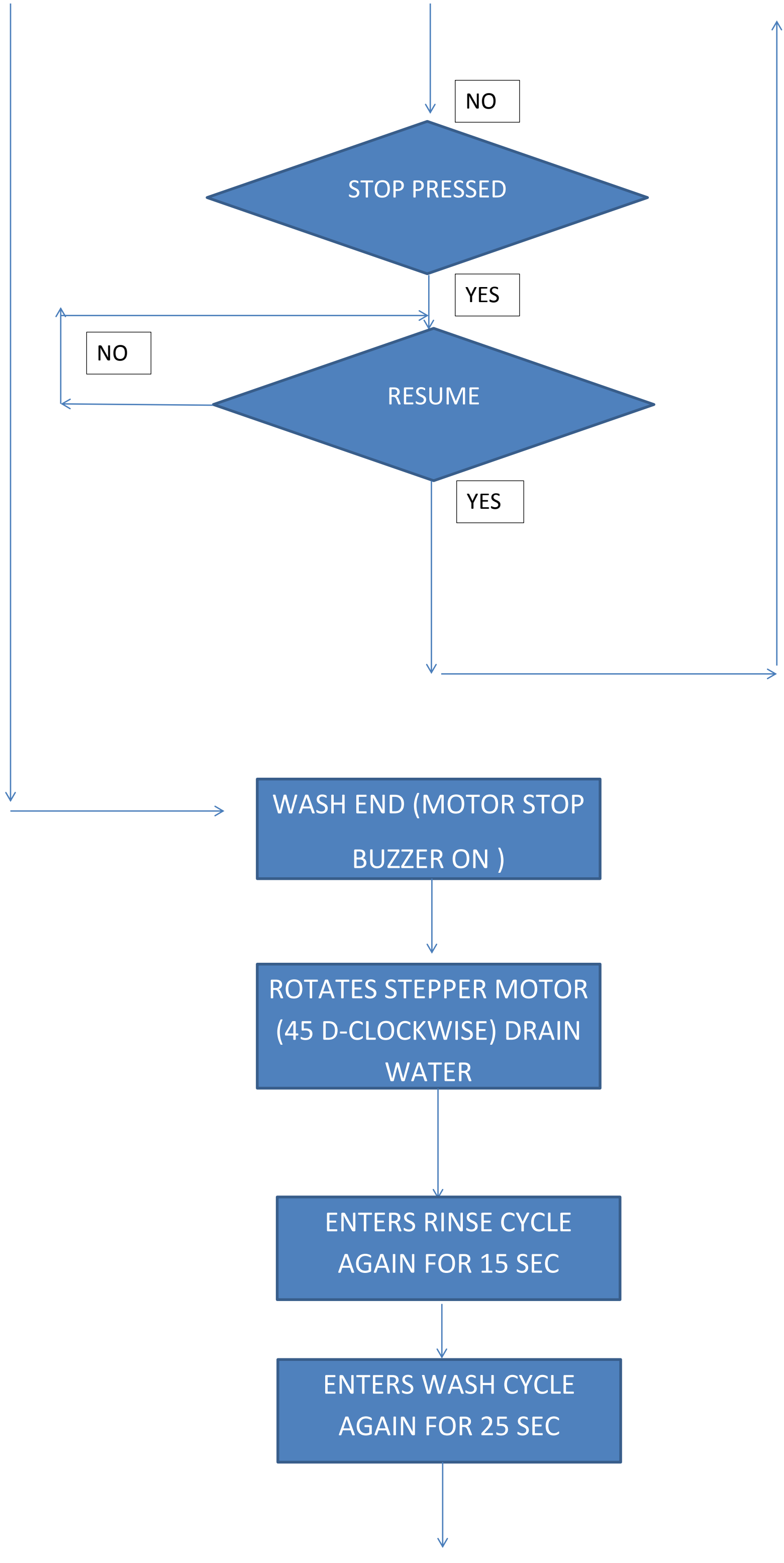
DRY END (MOTOR STOP BUZZER ON)

HEAVY LOAD (COUNT= 3)




YES
TIME OVER ?
NO
STOP PRESSED
YES
NO
RESUME
YES
RINSE END
(MOTOR STOP, BUZZER ON)
ROTATES STEPPER MOTOR (45 D-CLOCKWISE) DRAIN WATER


ENTERS WASH CYCLE
ROTATES STEPPER MOTOR (45 D-CLOCKWISE)
NO
PC4 =1 (WATER IS FILLED)
ROTATES STEPPER MOTOR (45 D-ANTI CLOCKWISE)
SET COUNT FREQ OF 8253(1) TO GET COUNT=25 SEC
DC MOTOR STARTS, LED OF WASH ON, LED OF DOOR ON
YES
TIME OVER ?


NO
STOP PRESSED
YES
NO
RESUME
YES
WASH END (MOTOR STOP BUZZER ON)
ROTATES STEPPER MOTOR (45 D-CLOCKWISE) DRAIN WATER
ENTERS RINSE CYCLE AGAIN FOR 15 SEC
ENTERS WASH CYCLE AGAIN FOR 25 SEC

ENTERS RINSE CYCLE AGAIN FOR 15 SEC

ENTERS DRY CYCLE

ROTATES STEPPER MOTOR (45 D-ANTICLOCKWISE)

SET COUNT FREQ OF 8253(1) TO GET COUNT=20 SEC

DC MOTOR STARTS, LED OF DRY ON, LED OF DOOR ON

TIME OVER ?

YES

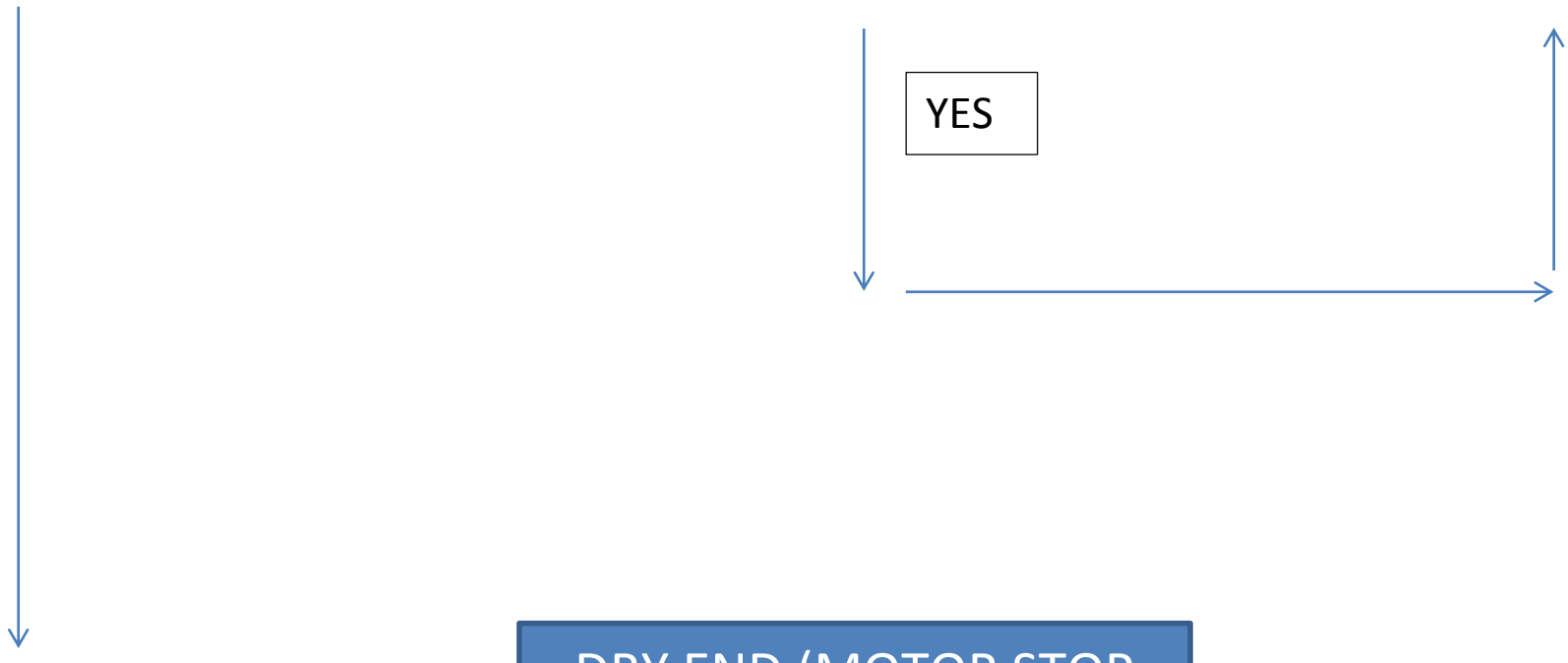
NO

STOP PRESSED

YES

NO

RESUME



CODE:

```
.model small

.8086

.stack 1024

.data


      INT0       EQU       0*4   ; four bytes taken by CS:IP


      TABLE    DB   00000001b


;8255-1
      porta equ 00h
      portb equ 02h
      portc equ 04h
      creg equ 06h
```

```
;8255-2
        port2a equ 08h
        port2b equ 0ah
        port2c equ 0ch
        creg2 equ 0eh

;8253 - 10h
        clk0 equ 10h
        clk1 equ 12h
        clk2 equ 14h
        creg3 equ 16h

;8259 - 18h

        p1 equ 18h
        p2 equ 1ah

        DAT1 db 00h
        DAT2 db 00h

.code
.startup
                        cli                                                     ;-----------disable
the interrupt

            ; set data segment
```

```
mov ax, @data
mov ds, ax

; set the ISRs address table
mov ax, 0
mov es, ax                          ; SMALL mode used put CS:IP in IVT starting at 00000
mov bx, cs
mov ax, ac_isr                          ; find the offset of the ISR procedure
            mov es:[INT0], ax               ; enter the ip value of ISR in IVT
            mov es:[INT0+2], bx                 ; enter the cs value of iSR

;----------------------------------------------Initializing 8255(1) and 8255(2)----------------;

            ;c lower is for buttons (8255-1)
            ;c3-load
            ;c2-resume
            ;c1-stop
            ;c0-start

            ;c lower is for buzzers
            ;c4-rinse
            ;c5-wash
            ;c6-dry
```

```
START:

        mov al,10001011b
        out creg,al

        mov al,10000010b
        out creg2,al

mov al,00001000b        ;setting PC4=0, upper sensor
out creg,al

mov al,00001010b   ;setting PC5=1, lower sensor
out creg,al

mov al,00001100b   ;setting PC6=0, gate of 8253
out creg,al

mov al,00001110b   ;setting PC7=0, for disabling DC motor to rotate tub and
agitator
        out creg,al



;----------------------------------------------Initialize 8259--------------;

        ; ICW1 (edge triggered, single 8259a, 80x86 cpu)
        mov al, 00010111b
```

```
out p1, al

; ICW2 (base interrupt number 0x00)
mov al, 00000000b
out p2, al

; ICW4 (not special fully nested, non-buffered, auto EOI, 80x86 cpu)
mov al, 00000001b
out p2, al
```

```
        ; OCW1 (unmask all interrupt bits)
        mov al, 00h
        out p2, al


;---------------------------------------------- input from user-----------;
;----------------------------------------------check if start is pressed-----;

        mov bx,0
```

```
;--------------------------------Assumption: the user presses load button only between one and three times------------------;


    LOAD:       in al,portc
                and al,00001000b
                cmp al,00001000b
                jnz LOAD
```

```
; De bounce key press
CALL DEBOUNCE

in al,portc
and al,00001000b
cmp al,00001000b
jnz LOAD

        inc bx     ;still pressed  ;----------to store the count of presses----------;

        in al,portc
        and al,00000001b
        cmp al,00000001b
        jnz LOAD            ;----------start not pressed , check for load presses again----;

        CALL DEBOUNCE

        in al,portc
        and al,00000001b
        cmp al,00000001b
        jnz LOAD

        cmp bx,1            ;----------check which label to go, whether light,medium or heavy--------;
        jz LIGHT
```

```
cmp bx,2
jz MEDIUM

cmp bx,3
jz HEAVY

cmp bx,0
jz LOAD
```

```
;----------------------------------DAT1 and DAT2 store the count for the second counter of 8253 depending on the cycle--------;
;----------------first counter generates a pulse of 1khz by taking a count of 2500d and this is the cascaded to second counter-----MODE -3 is usedhing----------;

        LIGHT:    MOV DAT1,10h
                  MOV DAT2,27h

                  mov al,00000001b ;LED=ON for rinse
                  out creg2,al

                  CALL RINSE

                  MOV DAT1,98h
                  MOV DAT2,3Ah
                  CALL WASH
```

```
MOV DAT1,10h
MOV DAT2,27h
CALL RINSE

MOV DAT1,10h
MOV DAT2,27h
CALL DRY

jmp START                    ;----------for taking new load----------;

MEDIUM:     MOV DAT1,98h
        MOV DAT2,3Ah
        CALL RINSE

        MOV DAT1,0A8h
        MOV DAT2,61h
        CALL WASH

        MOV DAT1,98h
        MOV DAT2,3Ah
        CALL RINSE

        MOV DAT1,20h
        MOV DAT2,4Eh
        CALL DRY

        jmp START
```

```
HEAVY:  MOV DAT1,98h
        MOV DAT2,3Ah
        CALL RINSE

        MOV DAT1,0A8h
        MOV DAT2,61h
        CALL WASH

        MOV DAT1,98h
        MOV DAT2,3Ah
        CALL RINSE

        MOV DAT1,20h
        MOV DAT2,4Eh
        CALL DRY

        jmp START
```

```
;------------------ISR--------------------------------;

ac_isr proc far  ; ISR procedure for INT-0/use 0 to 7 in proteus;
                 ;  OCW3 (no action, no polling, read ISR next read)
                 mov al, 00001011b  ; control word to read ISR
                 out 10h, al

                 ; Read ISR register to check for pending interrupts
                 in al, 10h

                 ; Find the index of the pending interrupt
                 mov si, 0
                 mov ah, 1
                 mov cx, 8

search:          test al, ah  ; bitwise AND
                 jnz done ; if ISR is set go to done
                 inc si
                 shl ah, 1
                 loop search

done:            ; OCW2 (non-specific EOI command) for resetting ISR
                 mov al, 00100000b
                 out 10h, al
```

```
; code for switching off all the LED's to indicate end of a particular cycle
mov al,00000000b
out creg2,al

mov al,00000010b
out creg2,al

mov al,00000100b
out creg2,al

        mov al,00000110b
        out creg2,al

        mov al,00001100b   ;------------------setting PC6=0, gate of 8253, to disable 8253 so that it doesn't generate further interrupts
        out creg,al

        iret
ac_isr endp

.exit

;----------------------------------------RINSE PROCEDURE-----------------------;

RINSE PROC NEAR

                call run_motor2   ;step angle of 45 degree to open valve
```

```
FILL:       in al,portc      ;water level sensor
            and al,10H
            cmp al,10H
            jnz FILL

;step angle of 45 degree to open valve close valve

            call stop_motor2

            mov al,00110110b       ;mode 3 for counter 0,1
            out creg3,al

            mov al,01110110b
            out creg3,al

            mov al,0c4h  ;counter 1 initialize for load-10s
            out clk0,al

            mov al,09h
            out clk0,al      ;1ms pulse

            mov al,DAT1 ;counter 2 initialize for load-10s
            out clk1,al

            mov al,DAT2
            out clk1,al      ;10s pulse
```

```
mov al,00001101b      ; gating signal ON
out creg,al

mov al,00000001b ;LED=ON for rinse
out creg2,al

mov al,00000111b ;LED=ON for door closed
out creg2,al

        call run_motor1   ;DC motor On for rinse

        sti

LOC:       in al,portc
        and al,0
        cmp al,0
        jz RINSE_END

        in al,portc             ;check for stop
        and al,00000010b
        cmp al,00000000b
        jnz LOC

; De bounce key press
CALL DEBOUNCE

in al,portc
```

```
        and al,00000010b
        cmp al,00000000b
        jz STOPR

        jmp LOC

STOPR:  mov al,00001100b
        out creg,al             ;gating signal OFF

        in al,portc
        and al,00000100b
        cmp al,00000100b        ;resume
        jnz STOPR

        ; De bounce key press
        CALL DEBOUNCE

        in al,portc
        and al,00000100b
        cmp al,00000000b        ;resume
        jnz STOPR

        mov al,00001101b        ; gating signal ON, count resumes
        out creg,al

        jmp LOC
```

```
RINSE_END:
        call stop_motor1 ;  DC motor off

        call rinseBuzzer

        call run_motor2   ;  valve open to drain water

DRAIN:   in al,portc              ;water level sensor
        and al,20H
                cmp al,20H
                jnz DRAIN

                call stop_motor2

                RET
                RINSE ENDP

;-----------------------------------------WASH PROCEDURE----------------------;

WASH PROC NEAR

                call run_motor2   ;step angle of 45 degree to open valve

        FILLW:    in al,portc      ;water level sensor
                and al,10H
                cmp al,10H
                jnz FILLW
```

```
;step angle of 90 degree to open valve close valve

        call stop_motor2

        mov al,00110110b       ;mode 3 for counter 0,1
        out creg3,al

        mov al,01110110b
        out creg3,al

        mov al,0c4h  ;counter 1 initialize for load-10s
        out clk0,al

        mov al,09h
        out clk0,al     ;1ms pulse

        mov al,DAT1 ;counter 2 initialize for load-10s
        out clk1,al

        mov al,DAT2
        out clk1,al     ;10s pulse

        mov al,00001101b       ; gating signal ON
        out creg,al

        mov al,00000011b ;LED=ON for wash
```

```
        out creg2,al

        mov al,00000111b ;LED=ON for door closed
        out creg2,al

        call run_motor1   ;DC motor On for wash

LOCW:   in al,portc
        and al,0
        cmp al,0
        jz WASH_END

        in al,portc             ;check for stop
        and al,00000010b
        cmp al,00000000b
        jnz LOCW

        ; De bounce key press
        CALL DEBOUNCE

        in al,portc
        and al,00000010b
        cmp al,00000000b
        jz STOPRW

        jmp LOCW
```

```
STOPRW: mov al,00001100b
        out creg,al             ;gating signal OFF

        in al,portc
        and al,00000100b
        cmp al,00000000b        ;resume
        jnz STOPRW

        ; De bounce key press
        CALL DEBOUNCE

        in al,portc
        and al,00000100b
        cmp al,00000000b        ;resume
        jnz STOPRW

        mov al,00001101b        ; gating signal ON, count resumes
        out creg,al

        jmp LOCW

WASH_END:
        call stop_motor1 ;  DC motor off

        call washBuzzer

        call run_motor2   ;  valve open to drain water
```

```
DRAINW:      in al,portc            ;water level sensor
             and al,20H
             cmp al,20H
             jnz DRAINW

             call stop_motor2

             RET
             WASH ENDP

;-----------------------------------------DRY PROCEDURE---------------------------;

DRY PROC NEAR

             mov al,00110110b      ;mode 3 for counter 0,1
             out creg3,al

             mov al,01110110b
             out creg3,al

             mov al,0c4h  ;counter 1 initialize for load-10s
             out clk0,al

             mov al,09h
             out clk0,al     ;1ms pulse
```

```
mov al,DAT1 ;counter 2 initialize for load-10s
out clk1,al

mov al,DAT2
out clk1,al     ;10s pulse

mov al,00001101b      ; gating signal ON
out creg,al

        mov al,00000101b ;LED=ON for dry
        out creg2,al

        mov al,00000111b ;LED=ON for door closed
        out creg2,al

        call run_motor1   ;DC motor On for dry

LOCD:     in al,portc
        and al,0
        cmp al,0
        jz DRY_END

        in al,portc             ;check for stop
        and al,00000010b
        cmp al,00000010b
        jnz LOCD
```

```
; De bounce key press
CALL DEBOUNCE

in al,portc
and al,00000010b
cmp al,00000000b
jz STOPRD
        jmp LOCD

STOPRD: mov al,00001100b
        out creg,al             ;gating signal OFF

        in al,portc
        and al,00000100b
        cmp al,00000000b        ;resume
        jnz STOPRD
; De bounce key press
CALL DEBOUNCE

in al,portc
and al,00000100b
cmp al,00000000b        ;resume
jnz STOPRD
```

```
        mov al,00001101b      ; gating signal ON, count resumes
        out creg,al

        jmp LOCD


DRY_END:
        call stop_motor1 ;  DC motor off

                call dryBuzzer


        RET
        DRY ENDP


;---------------------------------MOTOR1-----------------------------------------
run_motor1 proc near

        mov al,00001111b
        out creg,al

ret
run_motor1 endp
```

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

stop_motor1 proc near

        mov al,00001110b
        out creg,al

ret
stop_motor1 endp

;-------------------------------MOTOR 2-----------------------------------------

run_motor2 proc near

        mov al,00001111b
        out creg2,al

ret
run_motor2 endp

;----------------------------;

stop_motor2 proc near

        mov al,00001110b
        out creg2,al
```

```
ret

stop_motor2 endp

;-------------------------------------------------------------------------------

debounce proc near

MOV CX, 4E20H    ; delay of 20ms
DELAY:    LOOP DELAY

ret

debounce endp

;----------------------------------RinseBuzzer---------------------------------------

rinseBuzzer proc near

        mov al,00001001b              ;buzzer-rinse ON to signify RINSE end
        out creg2,al



        mov bx,200

        ;De bounce key press
x1R:    MOV CX, 4E20H    ; delay of 20ms
        DELAYR: LOOP DELAYR
```

```
        dec bx
        cmp bx,00h
        jnz x1R

        mov al,00001000b ;buzzer off
        out creg2,al

ret
rinseBuzzer endp
;-------------------------------------------------------------------------------

;----------------------------------WashBuzzer-------------------------------------

washBuzzer proc near

        mov al,00001011b            ;buzzer-rinse ON to signify WASH end
        out creg2,al

        mov bx,200

        ;De bounce key press
x1w:    MOV CX, 4E20H   ; delay of 20ms
        DELAYW:     LOOP DELAYW

        dec bx
        cmp bx,00h
```

```
        jnz x1w


        mov al,00001010b ;buzzer off
        out creg2,al


ret
washBuzzer endp
;-------------------------------------------------------------------------------
;----------------------------------DryBuzzer--------------------------------------

dryBuzzer proc near

        mov al,00001101b            ;buzzer-rinse ON to signify dry end
        out creg2,al

        mov bx,200

        ;De bounce key press
x1d:    MOV CX, 4E20H    ; delay of 20ms
        DELAYD: LOOP DELAYD


        dec bx
        cmp bx,00h
        jnz x1d


        mov al,00001100b ;buzzer off
```

```
        out creg2,al

ret
dryBuzzer endp
;-------------------------------------------------------------------------------



end
```