Microprocessors and Interfacing

Fire Alarm System

G-106

Members:

Alan Tom Jose: 2018A7PS0187G

Aanand S J: 2018A7PS0139G

Prajval M: 2018A3PS0299G

Ajay James Alex: 2018A7PS0633G

Madhav M Anand: 2018A8PS0028G

Contents

Description of System

This system checks for abnormal smoke content in a room and under such conditions throws open all exit doors and windows and opens a valve that releases the gas to put-out the fire. An Alarm is also sounded; this alarm is sounded until the smoke level in the room drops to an acceptable level. The room has two doors and four windows. The smoke detection system is made up of three smoke sensors placed on the ceiling of the room. When at least two of three detectors get turned on, the alarm system is activated. If only one of them is activated a different alarm sound is produced indicating probable malfunction of alarm. The system can be activated or deactivated using a single switch. .

Specifications of the problem:

1. The system can be activated/deactivated by a single switch.
2. Three smoke detecting sensors are placed in the ceiling which detect the smoke content in the room.
3. A fire alarm that is sounded when smoke exceeds normal levels for at least two sensors.
4. Two doors, four windows and valves are controlled by motors.
5. The motors and alarms are activated only when two of three sensors detect an abnormal level in smoke levels.
6. If one of them is activated, a single alarm is sounded.

Assumptions & Justifications

1. Clockwise direction closes all doors, valves and windows and anticlockwise direction opens all doors, valves and windows.

2. We are using MC145010 as the smoke sensor. The IC is used with an infrared photoelectric chamber and detection is accomplished by sensing scattered light from minute smoke particles or other aerosols. The output of the smoke sensor ranges from 0V to 5V. An ADC connected to variable voltage source is used in the design to model the output of the smoke sensor.

3. We have interfaced memory as follows:

   ROM1 – 00000_H – $00FFF_H$

   ROM2 – $FF000_H$ – $FFFFF_H$

   RAM 1– 01000_H – $01FFF_H$

4. When we power on the system, all the doors, windows and valve are closed by passing the appropriate control signals through PPI. After this, interrupts are continuously raised to measure the sensor outputs. If both the sensors exceed a threshold value then only doors, windows and valves are opened. In addition to this, an alarm is also sounded.

5. As long as both the sensors are above threshold value, all the doors, windows and valve will remain open. If any sensor falls below the threshold value then close all the doors, windows and valve and alarm is also turned off.

6. The opening of all doors, windows and valves is controlled by a stepper motor which operates in steps with the use of gears. A stepper motor can be programmed to stop rotating after a certain number of steps, which can be programmed through our code.

Components used:

- 6116 (RAM) – 2 units used
- 2716 (ROM) – 4 units used
- ADC0808 (1 unit):
  - 8-bit ADC
  - 8 channel
  - 1 MHz clock input
  - Resolution 19.5 mV

- 8254 (Programmable interval timer)
  - 24 pin IC

- 8255 (Programmable Peripheral Interface (PPI) chip)
  - Contains three 8-bit ports.
  - 24 input/output pins.
  - Port A has input lines
  - Port B and port C (upper and lower) are output lines

- 8259(Programmable Interrupt Controller)
  - Interrupt generated at interval of 0.1 seconds.

- 8086(Microprocessor)
    - Operating Clock Speed – 5MHz
    - 40 pins
    - 20 de-multiplexed Address Lines and 16 de-multiplexed Data lines

- 74LS373(octal latch)
    - 3 Latches used
- 74LS245 (Octal Bus Transmitter/Receiver) – 2 units used
- 74LS244 (Unidirectional Octal Buffer) – 1 units used
- 74LS04 (Hex inverter) – 2 not gates used.
- 74LS32 (OR Gate IC) – 8 OR Gates used
- L297 (Stepper motor controller): Signals from your microprocessor and translates them into stepping signals to send to the L298
- L298 (Motor driver)
    - Drives the stepper motor
    - 4 output pins connected to one motor
- Smoke Sensor MC145010
    - Used with an infrared photoelectric chamber

Memory Organization:

Total RAM used by system = 4KB (2 Nos)

Total ROM chips used by system = 8KB (4Nos)

RAM chip used = 6116.

ROM chip used =2716

RAM

Starting address of RAM = 01000_H

Ending address of RAM = $01FFF_H$.

Total 4KB

EvenBankofRAM=01000_H,01002_H, 01004_H, 01006_H,. .....,$01FFE_H$.
OddBankofRAM=01001_H,01003_H, 01005_H, 01007_H,. .....,$01FFF_H$.

RAM used to house the data segment and stack segment.

ROM

Starting address of ROM1 = 00000_H

Ending address of ROM1 = $00FFF_H$.

Total 4KB

EvenBankofR0M1=00000_H,00002_H, 00004_H, 00006_H,. .....,$00FFE_H$.
OddBankofR0M1=00001_H,00003_H, 00005_H, 00007_H,. .....,$00FFF_H$.

Starting address of ROM2 = $FF000_H$

Ending address of ROM2 = $FFFFF_H$.

Total 4KB

EvenBankofR0M2=$FF000_H$,$FF002_H$, $FF004_H$, $FF006_H$,. .....,$FFFFE_H$.
OddBankofR0M2=$FF001_H$,$FF003_H$, $FF005_H$, $FF007_H$,. .....,$FFFFF_H$.

ROM used to house the code segment.It depends on the size of the program.

(C)IO BASED MEMORY MAPPING:

- 8255 Programmable Peripheral Interface:

One 8255(Programmable Peripheral Interface) are used to communicate with other input and output devices. It is organized in the following manner.

PORT A: 00H

PORT B: 02H

PORT A: 04H

CONTROL REGISTER: 06H

- 8253 Programmable Interval Timer

We need to check values from sensors every 0.1 second. For this we need a timer and hence we are using 8253. It is organized in the following way:

TIMER A: 08H

TIMER B: 0AH

TIMER A: 0CH

CONTROL REGISTER: 0EH

- 8259 Interrupt Controller

Used to set interrupts

A0: 10H

A1:12H

Other Devices

Stepper motor controller

Stepper motors are DC motors that move in discrete steps. They have multiple coils that are organized in groups called phases.By energizing each phase in sequence the motor will rotate, one step at a time. It converts digital pulses into mechanical shaft rotation. They are low cost, highly reliable and provide high torque at low speeds.

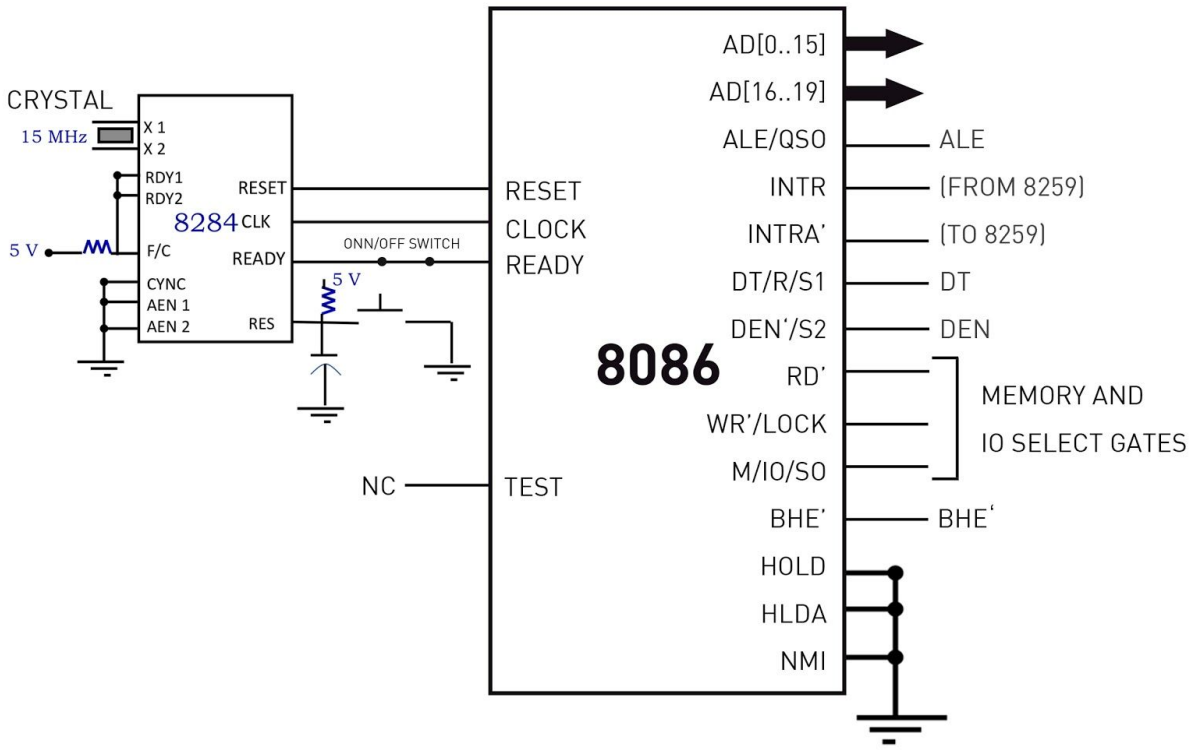
This Stepper motor controller uses the L297 andL298N driver combination; it can be used as stand alone or controlled by a microcontroller. It is designed to accept step pulses at up to 25,000 per second. An on-board step pulse generator can be used if desired (40-650 pps range). Single supply operation is standard.Here we use a voltage of 5 as VCC and 12 at VS in LS298 and a frequency of 10Hz to drive the motor.

Smoke Sensor

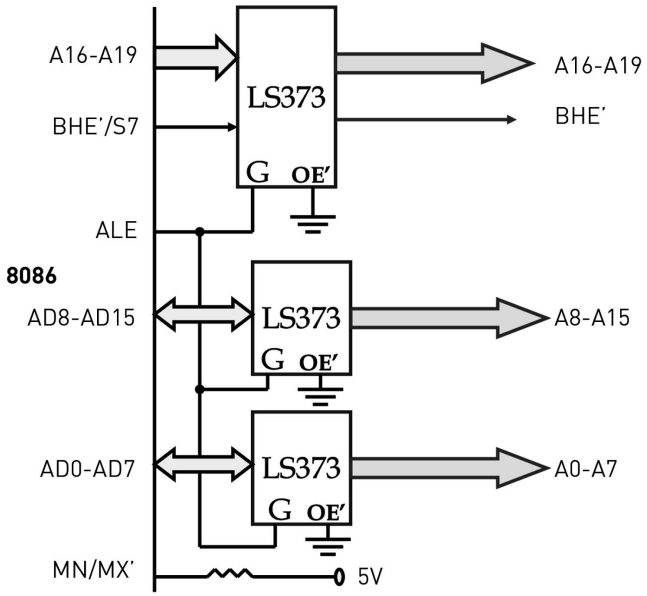
We are using MC145010 as the smoke sensor. The IC is used with an infrared photoelectric chamber and detection is accomplished by sensing scattered light from minute smoke particles or other aerosols. The output of the smoke sensor ranges from 0V to 5V. An ADC connected to a variable voltage source is used in the design to model the output of the smoke sensor.

DESIGN

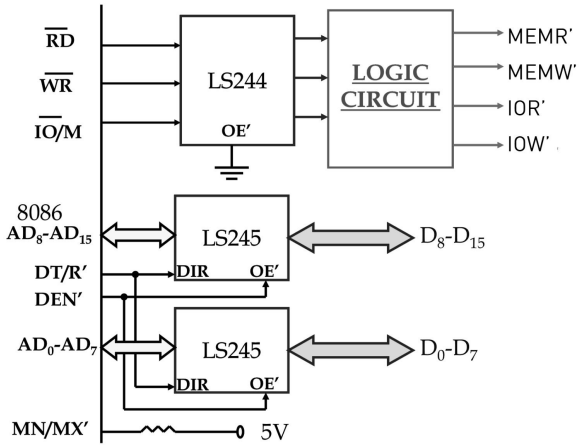
- 8086


CRYSTAL
15 MHz
X 1
X 2
RDY1
RDY2
8284
CLK
F/C
READY
CYNC
AEN 1
AEN 2
RES
RESET
5 V
5 V
ONN/OFF SWITCH
RESET
CLOCK
READY
NC
TEST
8086
AD[0..15]
AD[16..19]
ALE/QSO
INTR
INTRA'
DT/R/S1
DEN'/S2
RD'
WR'/LOCK
M/IO/SO
BHE'
HOLD
HLDA
NMI
ALE
(FROM 8259)
(TO 8259)
DT
DEN
MEMORY AND
IO SELECT GATES
BHE'

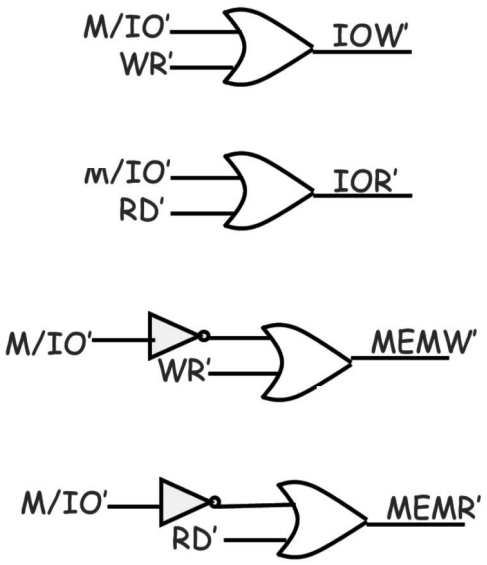
- System Bus (Address)



- System Bus(Data+Control)

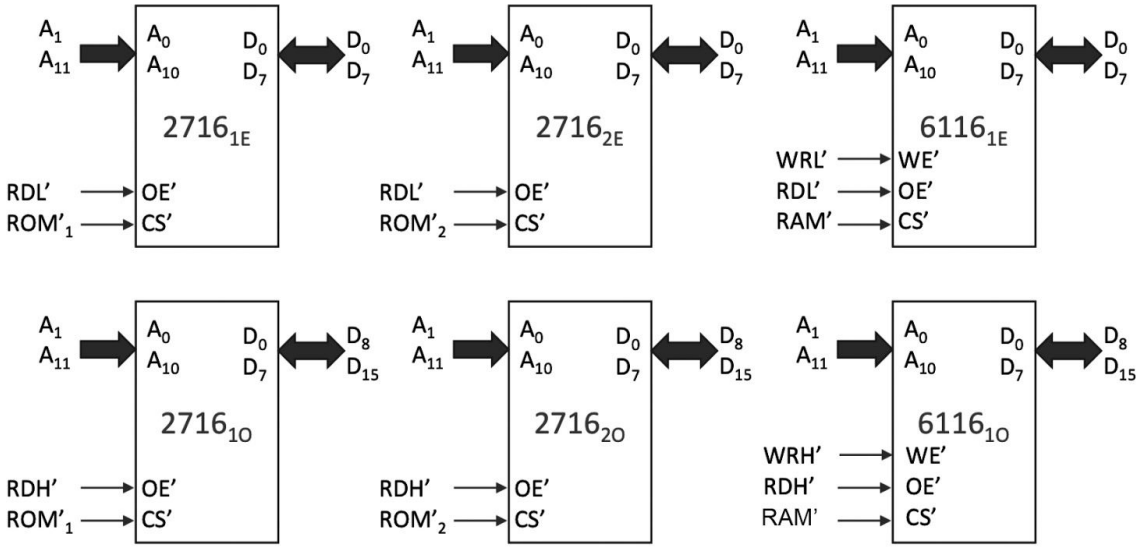


I/O and Memory Read Logic

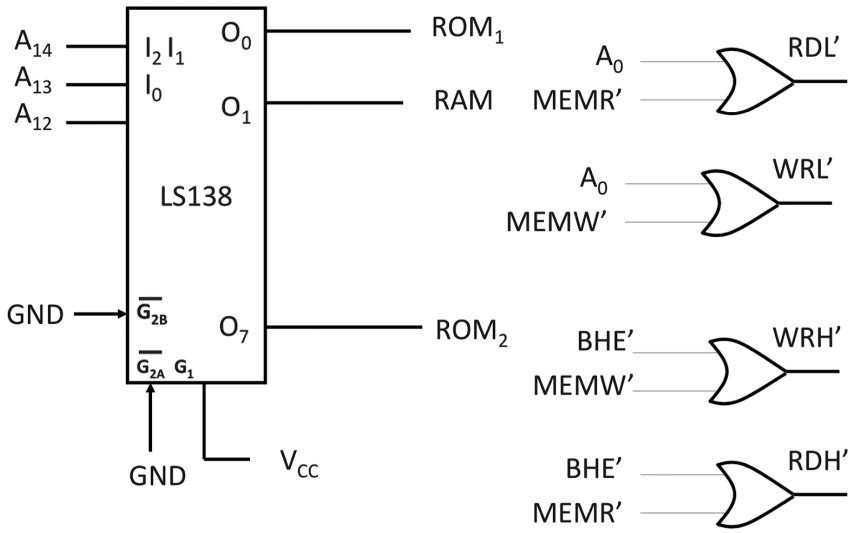


M/IO’	RD’	WR’	Bus Cycle
1	0	1	MEMR’
1	1	0	MEMW’
0	0	1	IOR’
0	1	0	IOW’

Memory Interfacing

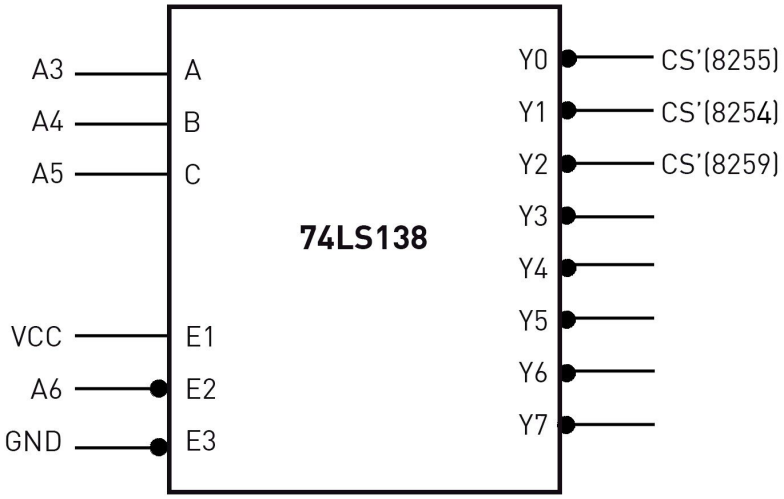


Memory Decoder

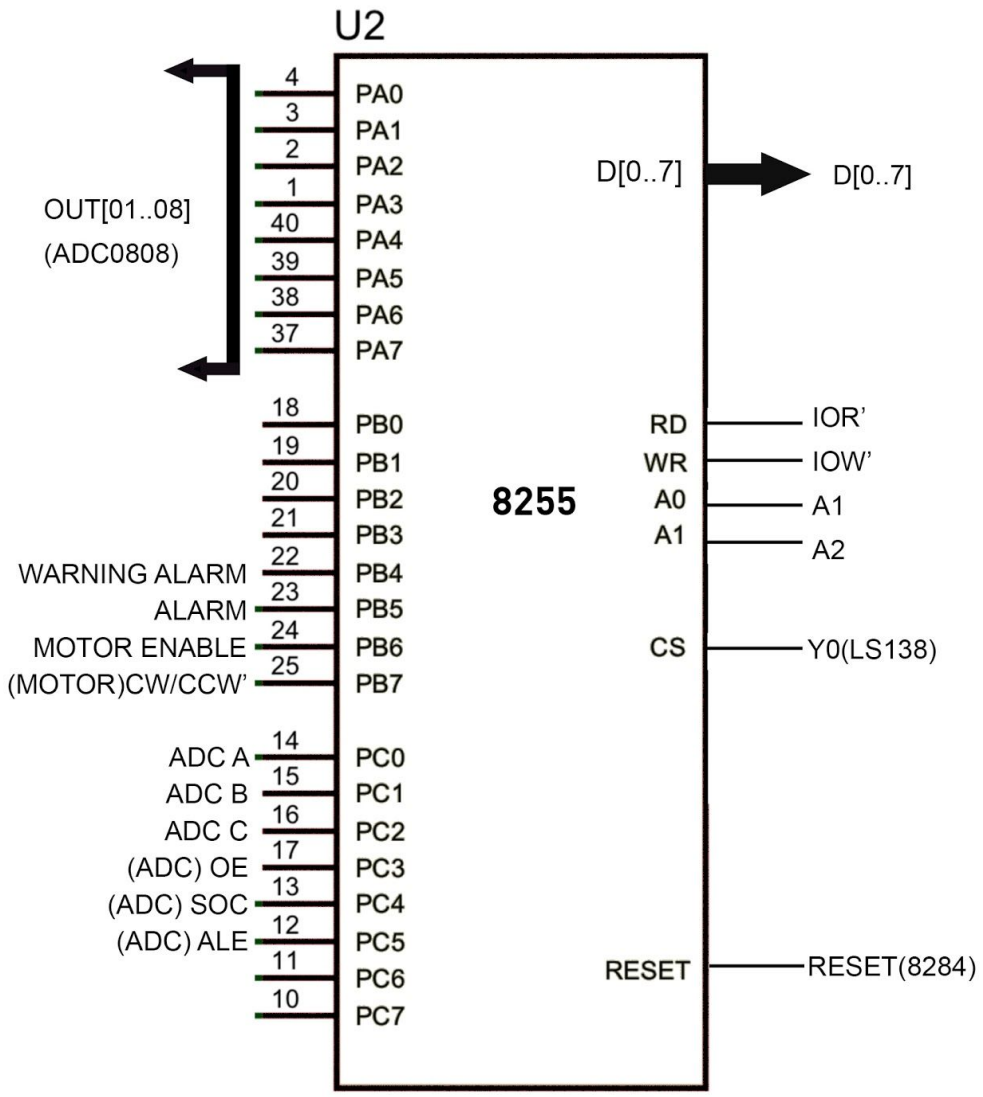


Memory Decoder

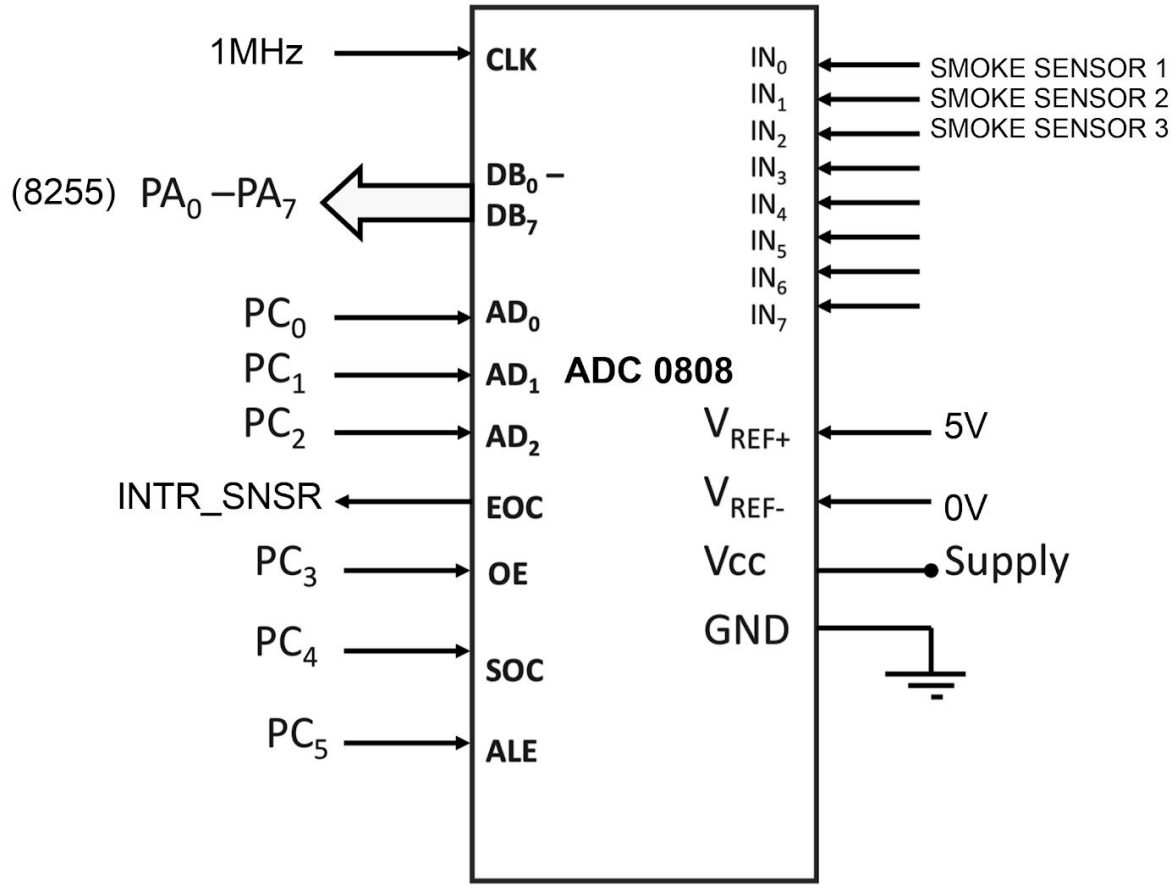
I/O Decoder



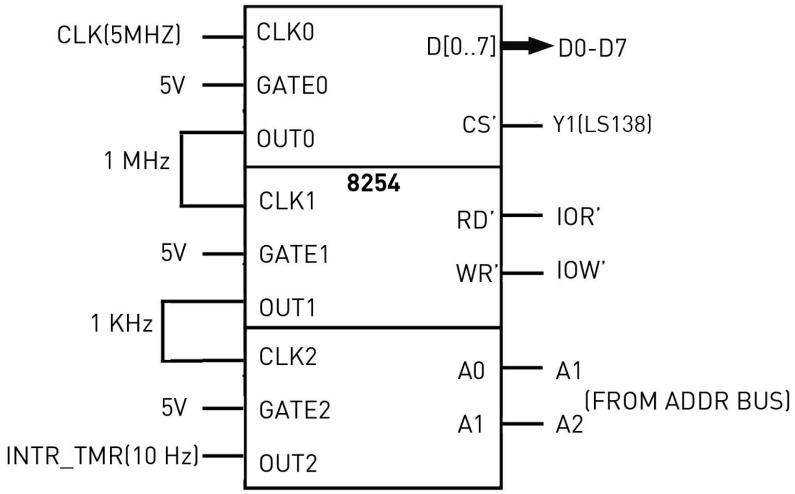
- 8255


U2
8255
OUT[01..08]
(ADC0808)
4
3
2
1
40
39
38
37
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
D[0..7]
D[0..7]
18
19
20
21
22
23
24
25
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
WARNING ALARM
ALARM
MOTOR ENABLE
(MOTOR)CW/CCW'
RD
WR
A0
A1
IOR'
IOW'
A1
A2
CS
Y0(LS138)
14
15
16
17
13
12
11
10
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
ADC A
ADC B
ADC C
(ADC) OE
(ADC) SOC
(ADC) ALE
RESET
RESET(8284)

- ADC


1MHz
CLK
(8255) PA0 –PA7
DB0 –
DB7
PC0
AD0
PC1
AD1
ADC 0808
PC2
AD2
INTR_SNSR
EOC
PC3
OE
PC4
SOC
PC5
ALE
IN0
IN1
IN2
IN3
IN4
IN5
IN6
IN7
SMOKE SENSOR 1
SMOKE SENSOR 2
SMOKE SENSOR 3
VREF+
5V
VREF-
0V
Vcc
Supply
GND

- 8254

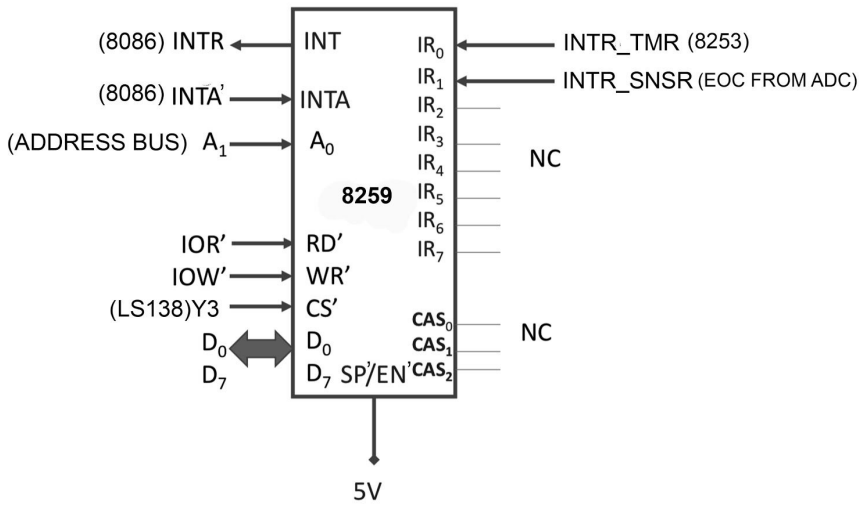


Timer 0 count 5

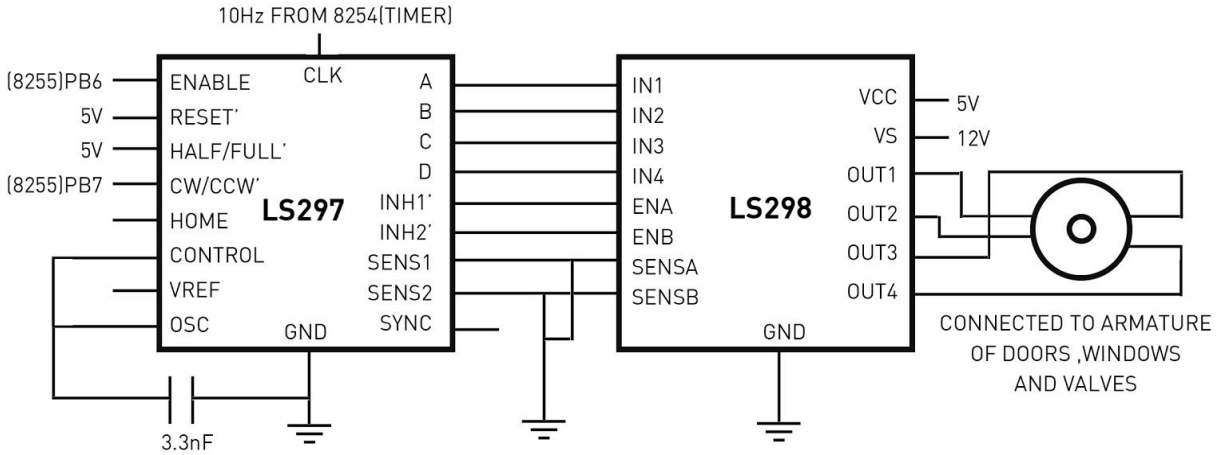
Timer 1 count 1000

Timer 2 count 100

- 8259

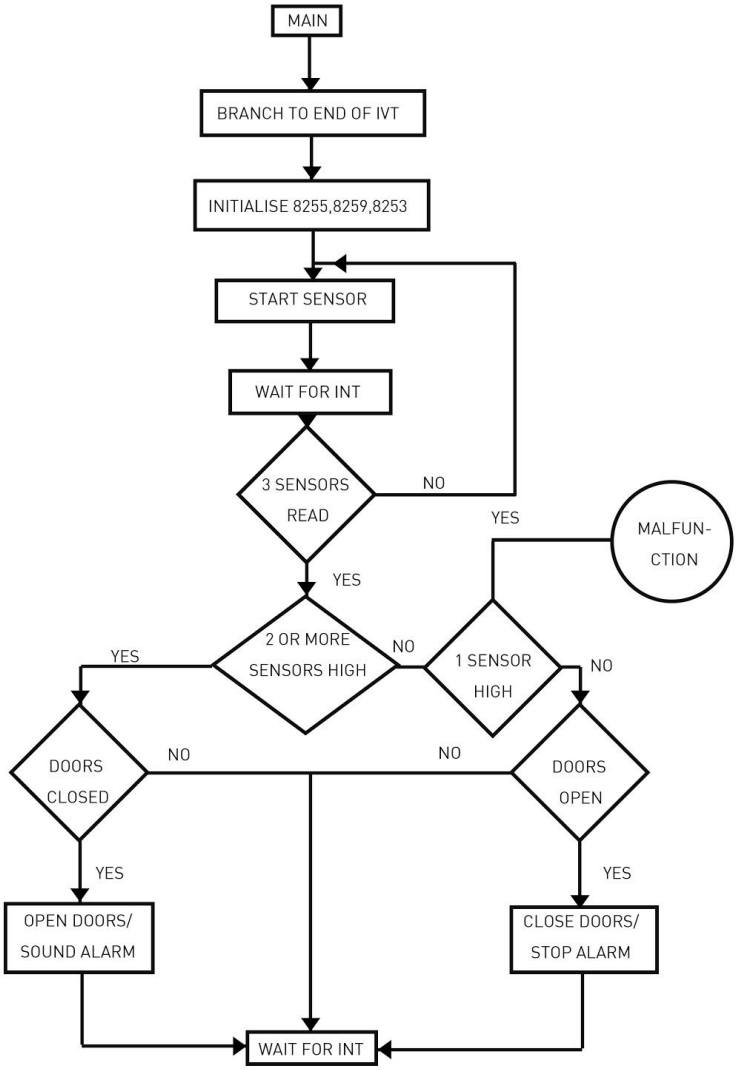


- Stepper Motor (L297,L298)

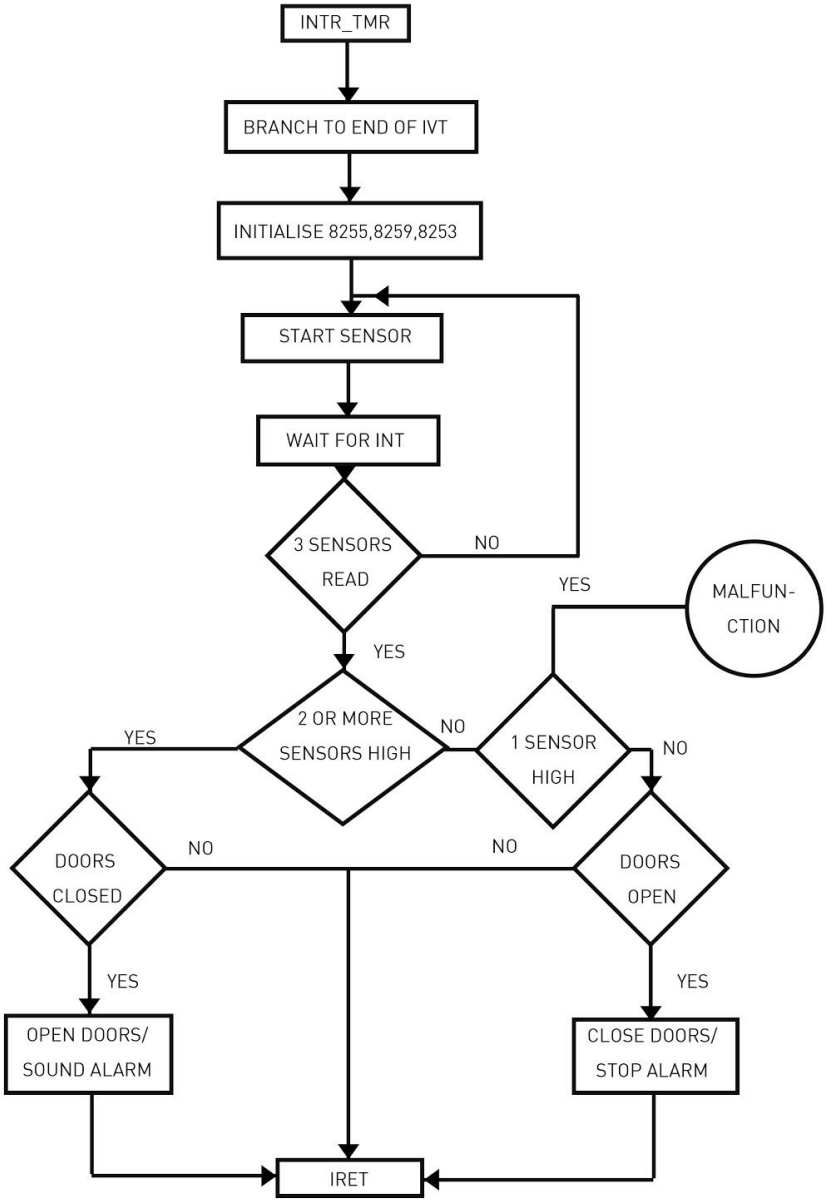


FLOWCHART

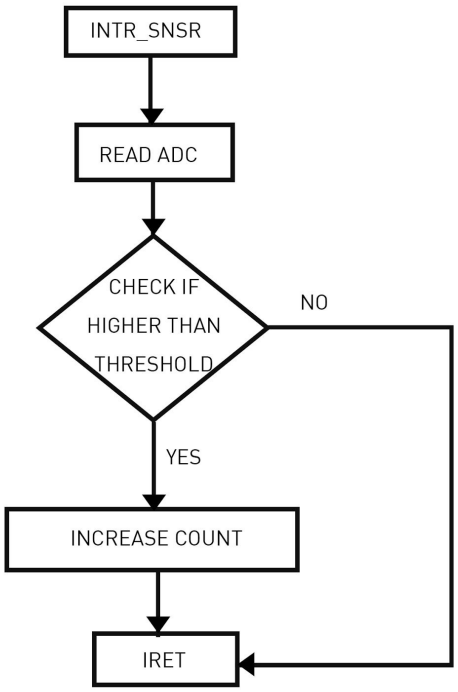
- Main



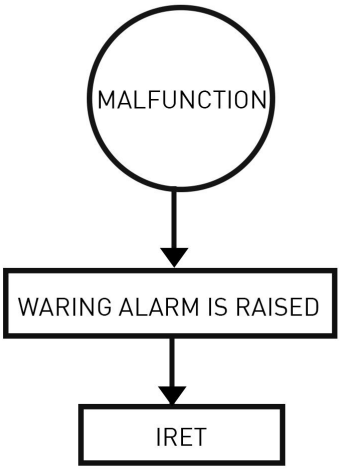
- Interrupt timer ISR



- ADC ISR (ISR 2)



- MALFUNCTION



Variations in Proteus Implementation with Justification

1. Using 0.1 second delay as 8259 does not work in proteus – EOC is used as NMI and the Timer Int replaced by software delay as 0.1 seconds.
2. The 7 segment display is temporarily kept in design for reference of door stage: 1- for door closed, 0 -for door open.
3. Design also has RED LED to indicate the alarm system is activated and doors are opened. BLUE LED to indicate malfunction when only 1 sensor is activated is also used in design.
4. ROM in only 00000 – as proteus allows to change reset address.
5. Using 8253 – as 8254 not available in Proteus.
6. Clock is at 2 MHz as the clock generated for 8086 requires a long rise and fall time of clock. So ADC clock will be only 500KHz; that is within the 1 MHz max clock that can be provided.
7. 2732 is used as 2716 – not available in Proteus.
8. Using a gate-based circuit for memory – does the same as LS 138 here .
9. Smoke Sensor is replaced by DC voltage (0V-5V) as smoke sensors not available in Proteus.

Firmware

Implemented using emu8086 attached.

LIST OF ATTACHMENTS

1. Manuals
    a. ADC 0808
    b. MC145010
    c. L297
    d. L298
    e. Stepper Motor
    f. Alarm
2. Proteus File – alarm.dsn

3. EMU8086 ASM File – firealarm.asm

4. Binary File after assembly – firealarm.bin