Experiment No: 9

Aim: Write an application using Raspberry Pi / Begal board tp control the operaion of a hardware simulated lift elevator simulation using Raspberry pi Board.

Aim / objectives:

1. To understand the working principle of lift elevator.
2. To interface the lift elevator Module with Raspberry Pi Model.
3. To program the Raspberry Pi model to control operation of lift elevator module.

~~Software: Raspbian os (IDLE)~~

Hardware Modules: 1) Raspbian os (IDLE)
2) Raspberry Pi Board Module.
3) push Buttons (qty. 8)
4) Seven segment Display (qty-1)
5) Leds (qty. 4)
6) monitor.

Theory:

lift Elevator Module has 2 Parts:

① Moving Part inside the lift 4.
② ~~at~~ stationary Part outside the lift at each floor to call the lift.
③. In this simulation module, we have considered four floors of a building.
④ So the moving part contains four push buttons. out of these four bottons, one button is for each floor having floor no written below it.

(5) The Moving part is also contain a seven segment display to indicate the current floor number when then lift is moving.

(6) by pressing one of these buttons, the user indicates the destination floor.

(7) At each floor, the stationary part contains a button for calling the lift.

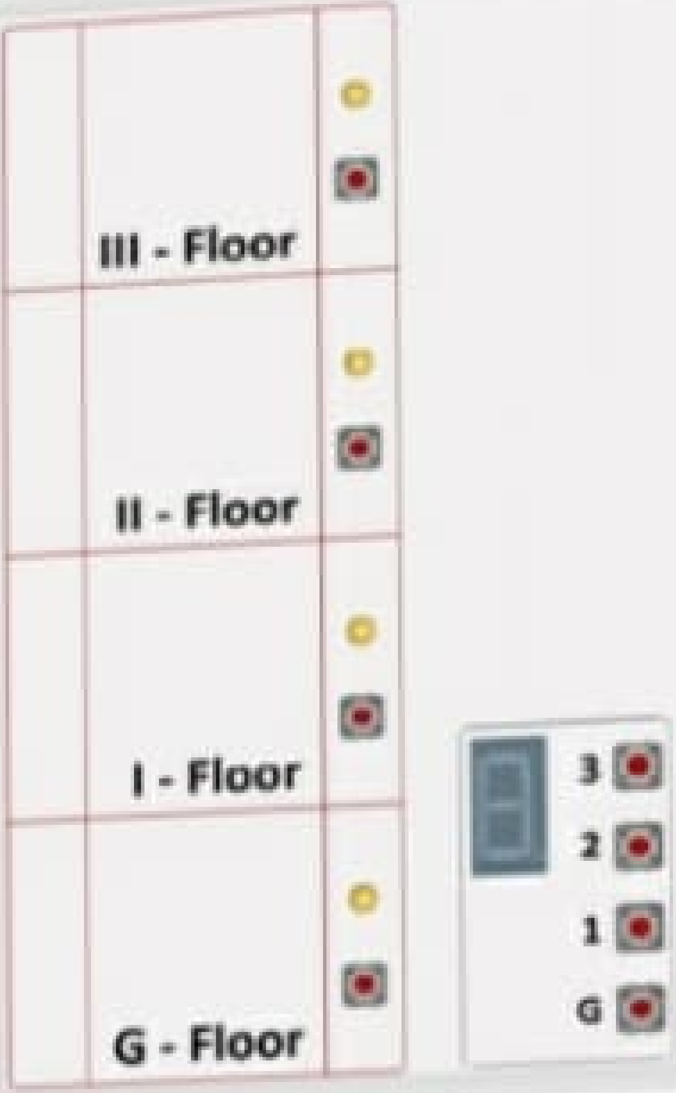
(8) In real life, when the lift is called by any floor, the lifts starts moving towards the particular floor when it reaches the particular floor, the lift door is opened.

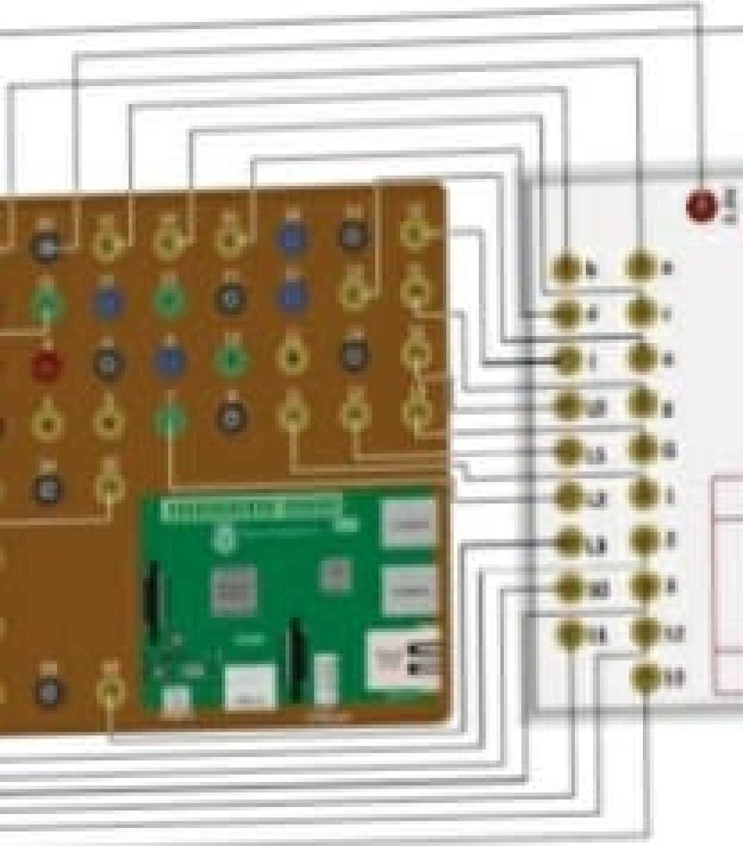
(9) In our module, this situation, is indicated by the "LED-ON" status. so the ON status indicated that the lift has arrived at the particular floor.

(10). In real life, as soon as the entering users get finished the lift door is closed & the lifts starts moving toward the des^n.

(11) In our Module, this situation is indicated by "LED OFF" Status.

fig


III - Floor
II - Floor
I - Floor
G - Floor
3
2
1
G



Safety Precautions:
① first, make all the Connections as per steps given below.
② Power supply.

Steps for assembling Circuit:
① Connect all the pins of lift elevator Modul to pin of Raspberry Pi Module as

Procedure: ① write the program as per the alg^n given.
② Save the program
③ Run Code as using Run Module

① Import GPIO & time libraries.
② Set GPIO mode as per board.
③ Declare four push button pins of the Stationary Part.
④ Declare four LED pin at each floor for detection of door close & open.
⑤ Declare four Push button Pins of the Moving part.
⑥ Declare seven Pins of seven segment display.
⑦ Set the push button Pin as input.
⑧ Set the Seven segment display Pin & LED Pins as O/P.
⑨ Store the Value of each digital of 7 Segment display in Variables.
⑩ In the while loop of ~~each~~ digit of seven segment display in variables.
if "Button-one" is pressed then lift at floor 1 & LED at floor 1 get on for 5 second then gets off.

(11) Person enters in the lift & presses the push button of any floor in the moving lift.

(12) The seven segment display the floor number of the destinations.

Observation:

observe the o/p on LED's & seven segment display.

```
# interfacing Lift Elevator Module
      with Raspberry Pi-3.

import RPi.GPIO as GPIO
import time

floorButton0=37
floorButton1=35
floorButton2=33
floorButton3=19

LiftButton0=15
LiftButton1=11
LiftButton2=38
LiftButton3=36

# GPIO setup for the LEDs

floorLed0=16
floorLed1=13
floorLed2=7
floorLed3=40
```

```
# GPIO setup for the 7 segment display
segAPin = 18
segBPin = 22
segCPin = 24
segDPin = 26
segEPin = 29
segFPin = 32
segGPin = 31

GPIO.sedmode (GPIO.BOARD)
GPIO.setwarnings (False)

GPIO.setup(FloorButton0 , GPIO.IN)
GPIO.setup (FloorButton1 , GPIO.IN)
GPIO.setup (FloorButton2, GPIO.IN)
GPIO.setup (FloorButton3, GPIO.IN)

GPIO.setup (liftButton0, GPIO.IN)
GPIO.setup (liftButton1, GPIO.IN)
GPIO.setup (liftButton2, GPIO.IN)
GPIO.setup (liftButton3, GPIO.IN)

GPIO.setup(FloorLEd0, GPIO.out) #floor 1
GPIO.setup (FloorLEd1, GPIO.out) -11- 2
GPIO.setup (FloorLEd2, GPIO.out -11- 3
GPIO.setup (FloorLEd3, GPIO.Out -11- 4

GPIO.setup (segAPin, GPIO.out).
GPIO.Setup (segBPin, GPIO.out)
GPIO.setup (segCPin, GPIO.out)
GPIO.setup (segDPin, GPIO.out)
GPIO.setup (segEPin, GPIO.out)
GPIO.setup (segFPin, GPIO.out)
GPIO.setup (segGPin, GPIO.out)
```

```
digitclr =[0,0,0,0,0,0,0]
digit0 =[1,1,1,1,1,1,0]
digit1=[0,1,1,0,0,0,0]
digit 2=[1,1,0,1,1,0,1]
digit 3=[1,1,1,1,0,0,1]

gpin=[18 22, 24 26 ,29,32,31]
 def digdisp(digit):
 for x in range (0,7):
 GPIO.output (gpin[x] ,digit [x])

for x in range (0,7]:
GPIO.output [gpin [x], digit[x]]

while true:

if (GPIO. input (FloorButton 0)==True):
GPIO.output (FloorLed 0, 1)

Print" 0"

digdisp (digit0)
timesleep(1)
GPIO.output (floorLed 0, 0)
time.sleep(3)

while True:
 If (GPIO.input (liftButton 1)==True):
 Print'floor ONE'
 ~~disp~~ digdisp(digit0)
 time.sleep(1)
 digdisp (digit1)
 time.sleep(2)
 break

 elif (GPIO. input (liftButton2)==True):
```

```
Print" floor two"
disgdisp(digit0)
time.sleep(1)
digdisp(digit1)
time.sleep(1)
digdisp (digit2)
time.sleep(2)
break

elif (GPIO.input(liftButton3)==true):
    print" floor three"
    digdisp (digit0)
    time sleep (1)
    dispdisp (digit1)
    time.sleep(1)
    digdispl (digit2)
    time.sleep(1)
    digdisp (digit3)
    time.sleep(2)
   brak

elif (GPIO.input (FloorButton1)== True):

GPIO.output(FloorLed, 1)
Print" 1"

digdisp(digit0)
time.sleep(1)
digdisp(digit1)
time.sleep(1)
time.sleep(4)

GPIO.output (Floorled 1, 0)
```

```
else:
    #### time.sleep(3)
    digdisp(digit0)
    GPIO.output(floorLed0,0)
    GPIO.output(floorLed1,0)
    GPIO.output(floorLed2,0)
    GPIO.output(floorLed3,0)

else:
#### time.sleep(3)
digdisp(digit0)
GPIO.output(floorLed1,0)
GPIO.output(floorled2,0)
GPIO.output(floorled3,0)
GPIO.output(floorled0,0).
```