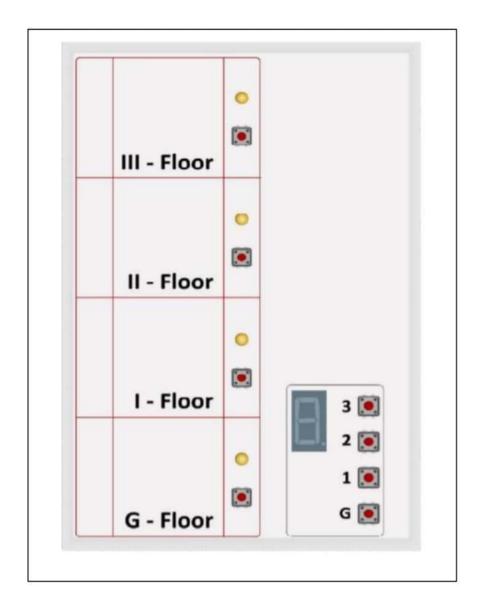
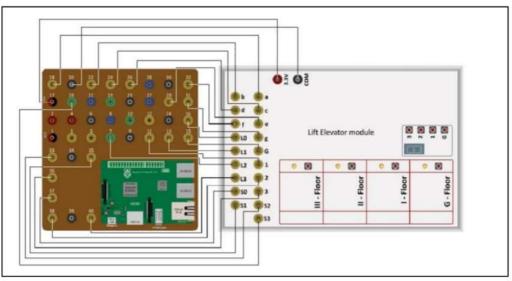
## Experiment no :-9 Aim 3- write an application wing Raspberry -pil begalboard to control the operation of hardware simulated lift elevator software-hardware modules :-- Raspbian as - Rospheery Pi board module - push bythons - seven segment display 3- Leds - monitor Algorithm :-- Import GPIO 4 time libraries - set GPTo mode aspec board - declare 4 push byttons pins of stationary part - declare 4 LED pins at each floor for detection of door close + open. - declare 4 push bytton pins of moving part - declare 7 pins of 7 segment display - set the push button pins as I/P. - set 7 segment display pins 4 LED pins as opp. - store value of each digit of 7 segment display in variables. - In while loop, it 'Author- one' is pressed then lift at floor 1 4 LED at hoor I get on for see then gets off. - person enters in lift + press push byttons of any one from in moving lift.

	CALL STREET, VICTORIAN OF THE STREET,
	- 7 segment display displays from no of destination.
	THE PARTY OF THE P
	observation :-
	# interfacing lift elevator module with Raspberry-19-3
	import RPI. GPTO as GPTO
	import time
	Floor bytton0 = 37
	Floor By Hon 1 = 35
	Floor Buffun 2 = 33
	Flourpyttons = 19
	THE THE PERSON OF THE PERSON O
	Liftorttono = 15
	Liftortton) =11
	LiftButton2 = 38
	Liftbytton3 = 36
	# GPZo setup for LEDS
	F1041ed0 = 16
	Floated 1 = 13
	Floorled 2 = 7
10.00	Floorled3 = 40
	the sale of the parties and the sale was become be to the be-
	# GPTO setup for 7 seg display
	SegAPin = 18
	segapin = 22
	Sey Can = 29
	Seg Dpin = 26
	SegFpn = 29
	seg fpin = 32 seg fpin = 31
	seg 9 pin = 31
3500	
	Scanned with CamScanner





```
GPTO-SEtrode (GPTO. BOARD)
 Gp to se twomings (False)
 GP to setup ( Floor buttono, GP to. in)
 GPTO-setup (Flour buffon) GPTO. in)
 apro-setyp (Froorby Hon 2, apro- 11)
 GPTO- setyp (From by Hon3, GP To. in)
GPIO. setyp (Liftbyttono, GPIO. in)
 GPIO. setup (Liftby Hon 1, GPIO, in)
 GP to setup (lifts 4 Hon 2, GPZo, in)
GPZo. setup (lift by Hon3, GPZo. in)
 GPTO Setup (segAPID, GPTO.OUT)
GPIO. setyp (seg BPin, GPIO. OVT)
GPTO-setup (segopin, GPTO.OUT)
GPIO. setup (seg DPin, GPIO. OUT)
GPTO. setup (SO EPID, GPTO.OVT)
GPTO. Setyp (Seg FPIN, GPTO. OUT)
GPTO Setyp (SegGPin, GPZO.04T)
digit(1x = [0,0,0,0,0,0,0]
digito = [1,1,1,1,1,0]
digit = [0,1,1,0,0,0,0]
digit3 = [1,1,1,1,0,0,1]
gpin = [18,22,24,26,29,32,315
 def digdisp (digit):
for x in range (0,7):
GPTO. OUTPUT (Spin (X), digitalia)
```

Scanned with CamScanner

```
for my in range (0,7):
GP To Output (gpain [4], digit [x])
while True :
if Capto inpyf ( floody Hono) = True);
GPIO. output (Moorled, 0,1)
pnn + 'o'
digdisp (digito)
 time sleep (1)
GPTO sytoyt (from led 0,1)
 time steep (3)
While True:
if CEPTO. input (liftbutton) == True):
priof 'floor' ONE:
 diadisp (diait o)
 bine sleep (1)
diadisp (digit1)
 time-slepp(2)
 bregk
elif (GPTO input (lifebuffonz) == True):
print foor Two'
digdisp (digito)
time sleep (U)
digdisp (digit 1)
time sleep (1)
digdisp (digitz)
time-sleep (2)
 break
```

```
elif (GP To input (Lift by Hon3) == True):
  priot foor Three!
  digdisp (digito)
  time sleep (1)
  digdisp (digit 1)
  bne-sleep (1)
  digdisp (digita)
  time-sleep (1)
 digdise (digit 3)
  time-sleep (2)
  bHgk
 Clif (GPTO-inoy t (frootbutton 1) == True):
 apro-output (housed,1)
 11. toud
 digdisp (digito)
 time sleep (1)
 dig disp (digit1)
 time sleep (1)
time-sleep (4)
apto output (foor led 1,0)
While Trye
if (apto input (lift buttons) = = True):
print from ZERO'
digdisp (digitz)
time sleep (1)
digdise (digiti)
time-sleep(1)
disdisp (digita)
```

brak elif (GP TO input (liftbuffon) = True): print foot one, digdisp (digita) time-sleep (1) disdisp (disit1) time-sleep (2) bleck elif Capto input (liftbutton 2) = True): print 'frog Two' digdisp (digita) time sleep (2) heg K PISP: time. sleep (3) digdisp (digito) apto output ( fronted 1,0) GP To. 04 tout (floor led 2,0) apro-output (from led 3,0) GPI outout (floor bed o, o) Conclusion :-Hence, successfully implemented control operation of lift elevator module by using Raspberry Pi.