Qn1.Smart Home Temperature control

initialise setpoint to a value

initialise both cooling_system and heating_system to OFF

Initialise LCD

Read current_temp inside an infinite while loop

Try:

if current_temp> setpoint

Turn ON cooling_system

Turn OFF heating system

Print cooling system activated

Print current_temp on LCD

else if

if current_temp<setpoint

Turn OFF cooling_system

Turn ON heating_system

Print heating system activated.

Print current_temp on LCD

Else

Turn OFF cooling_system

Turn OFF heating_system

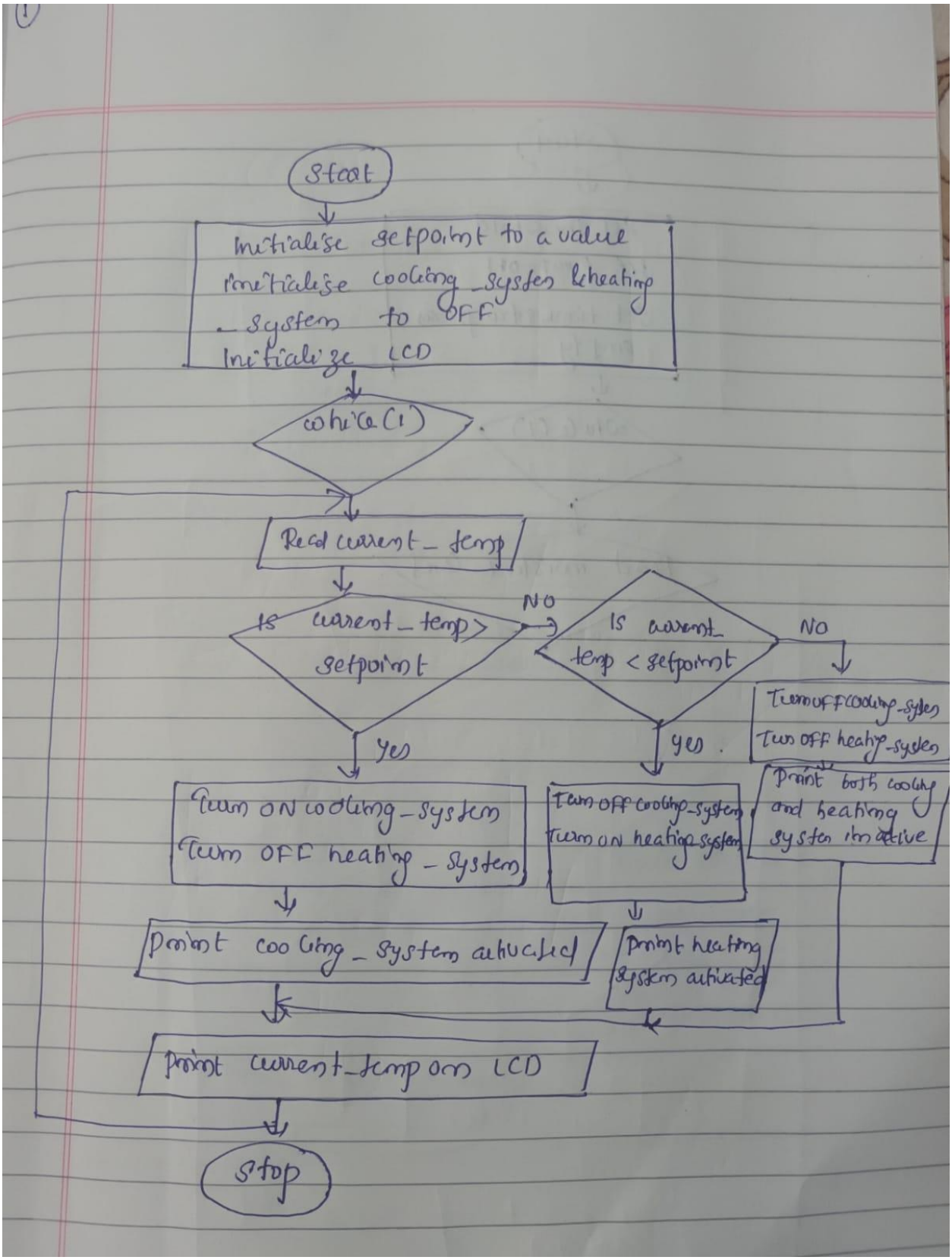
print both cooling and heating system inactive

Print current_temp on LCD

catch

print error

set wait(60) and repeat loop


start
Initialise setpoint to a value
Initialise cooling_system & heating_system to OFF
Initialize LCD
while(1)
Read current_temp
Is current_temp > setpoint
NO
Is current_temp < setpoint
NO
Turn off cooling_system
Turn OFF heating_system
Print both cooling and heating system inactive
yes
yes
Turn ON cooling_system
Turn OFF heating_system
Turn OFF cooling_system
Turn on heating_system
Print cooling_system activated
Print heating system activated
Print current_temp on LCD
stop

Qn2.Automated plant watering system

Set threshold

Set pump=OFF

Set timestamp as empty

Read moisture_level inside an infinite while loop

if moisture_level<threshold then

    Turn ON the pump

    print pump is active

    LED=ON

    Log timestamp to SD

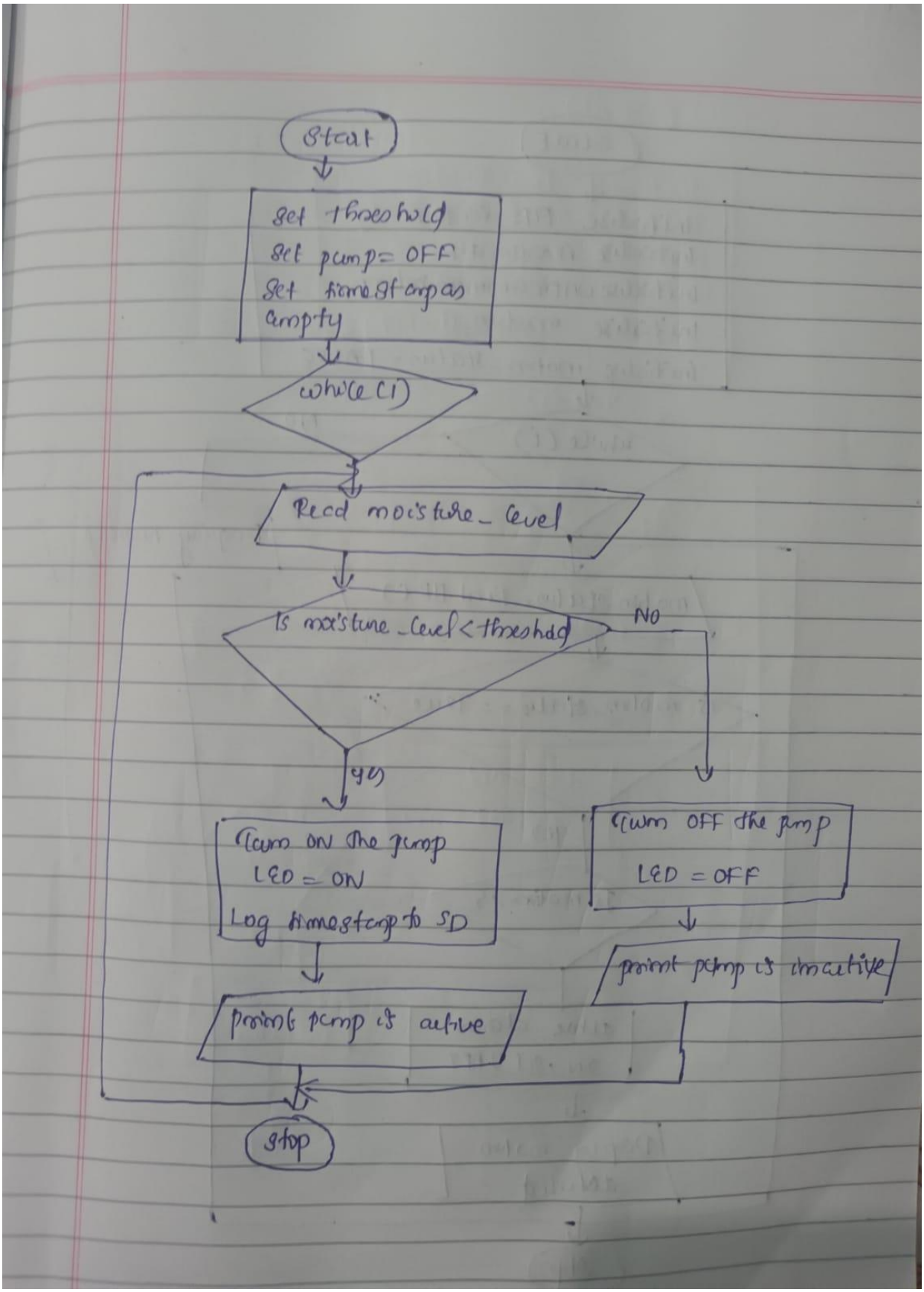
else

    Turn OFF the pump

    print pump is inactive

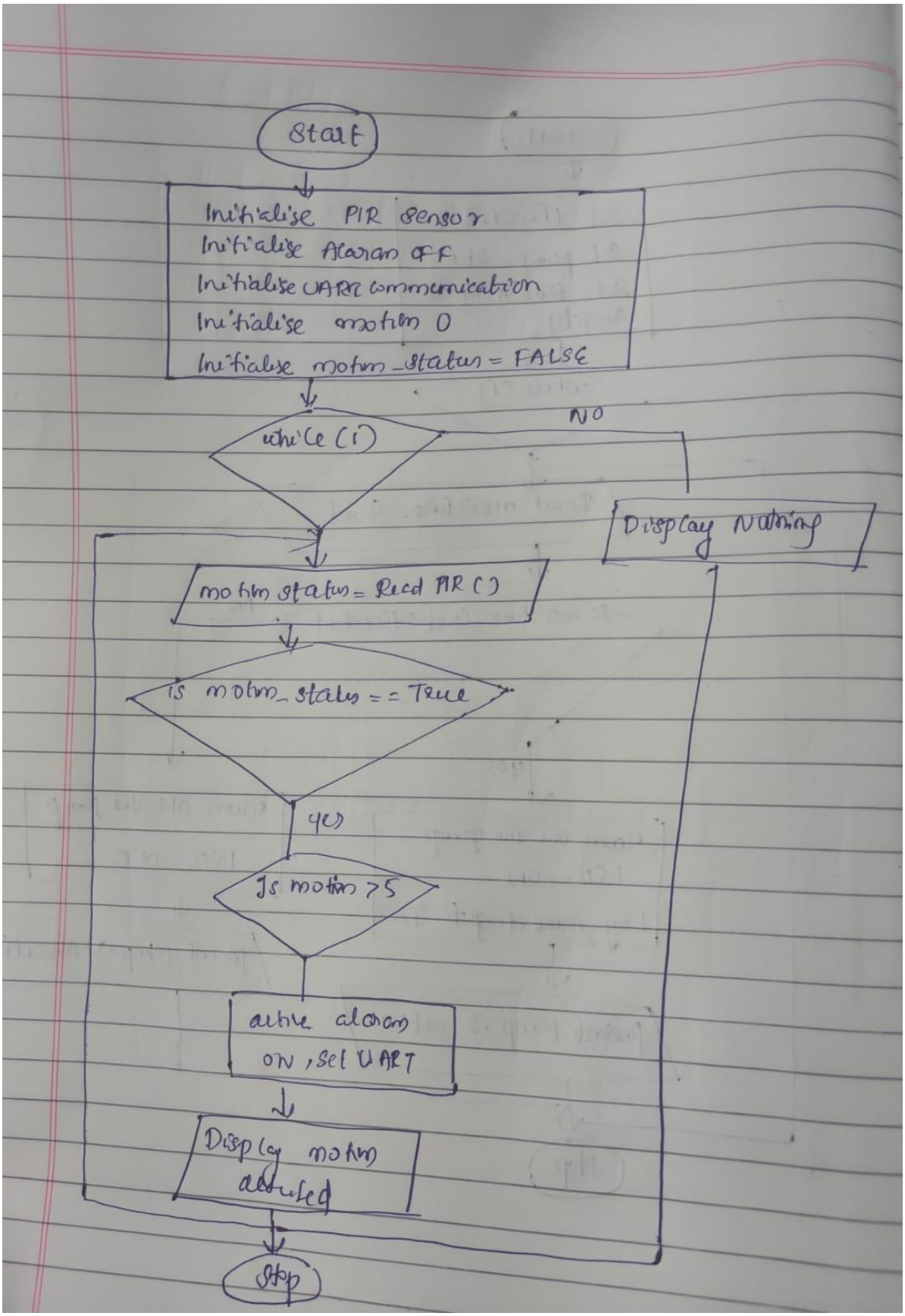
    LED=OFF

set wait(3600) and repeat loop


Start
Set threshold
Set pump = OFF
Set timestamp as empty
while (1)
Read moisture_level
Is moisture_level < threshold
NO
yes
Turn ON the pump
LED = ON
Log timestamp to SD
Turn OFF the pump
LED = OFF
Print pump is active
Print pump is inactive
stop

QN3.Motion Detection Alaram system

```
Initialize PIR sensor
Initialize Alaram OFF
Initialize UART commumication
Initialize motion 0
Initialize motion_status=FALSE
While(1)
motion_status=read PIR()
if(motion_ status == TRUE)
        motion=motion+1
if(motion>5)
        activate alarm ON
        activate UART ON print "motion detected"
        deactivate alarm OFF
        motion =0
        motion status FALSE
else
        print Nothing
wait(1 seconds)
```


Start
Initialise PIR Sensor
Initialise Alarm OFF
Initialise UART communication
Initialise motion 0
Initialise motion_status = FALSE
while (1)
NO
Display Nothing
motion status = Read PIR ()
is motion_status == True
YES
Is motion > 5
active alarm ON, Sel UART
Display motion detected
Stop

qn4.Heart rate monitor

initialize buzzer OFF

initialize LCD OFF

initialize SD

Initialise current_heartrate

Initialize heartrate _sensor

Set count=0

Set heart_rate=0

while(1)

heart current = read heart rate sensor()

heartrate =heartrate + heart current
count =count + 1
if(count==60)
avgheart = heartrate/count

Display avgheart ,heart current

if(averageheart >100)

activate buzzer ON

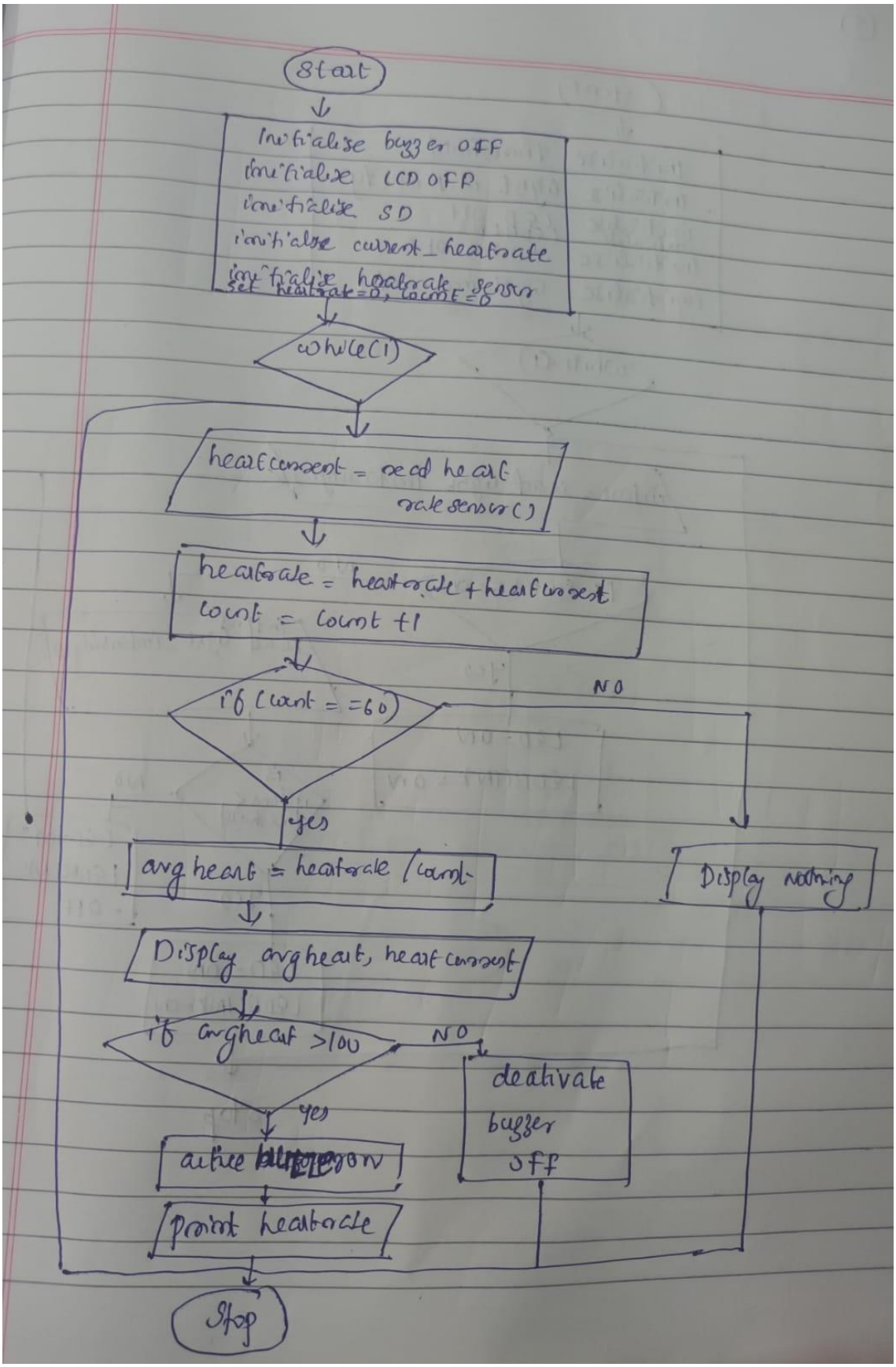
print heartrate

else:
deactivate buzzer OFF

set count = 0

heartrate =0

wait(1 second) and repeat loop


Start
Initialise buzzer OFF
Initialize LCD OFF
initialize SD
initialise current_heartrate
initialize heartrate, sensor
set heartrate=0, count=0
while(1)
heartcurrent = read heart rate sensor()
heartrate = heartrate + heartcurrent
count = count +1
if (count ==60)
NO
yes
avg heart = heartrate / count
Display nothing
Display avgheart, heartcurrent
if avgheart >100
NO
yes
deativate buzzer off
print heartrate
Stop

Qn5.LED control based on light sensor

Initialize threshold

Initilaise light_intensity

Initialize LED OFF

Initialize Switch OFF

Initialize light_intensity

While(1)

Intens=read light_intensity()

If(Switch==ON)

    LED=ON

    LEDBLINK=ON

else

If(intens<threshold)

     LED=ON

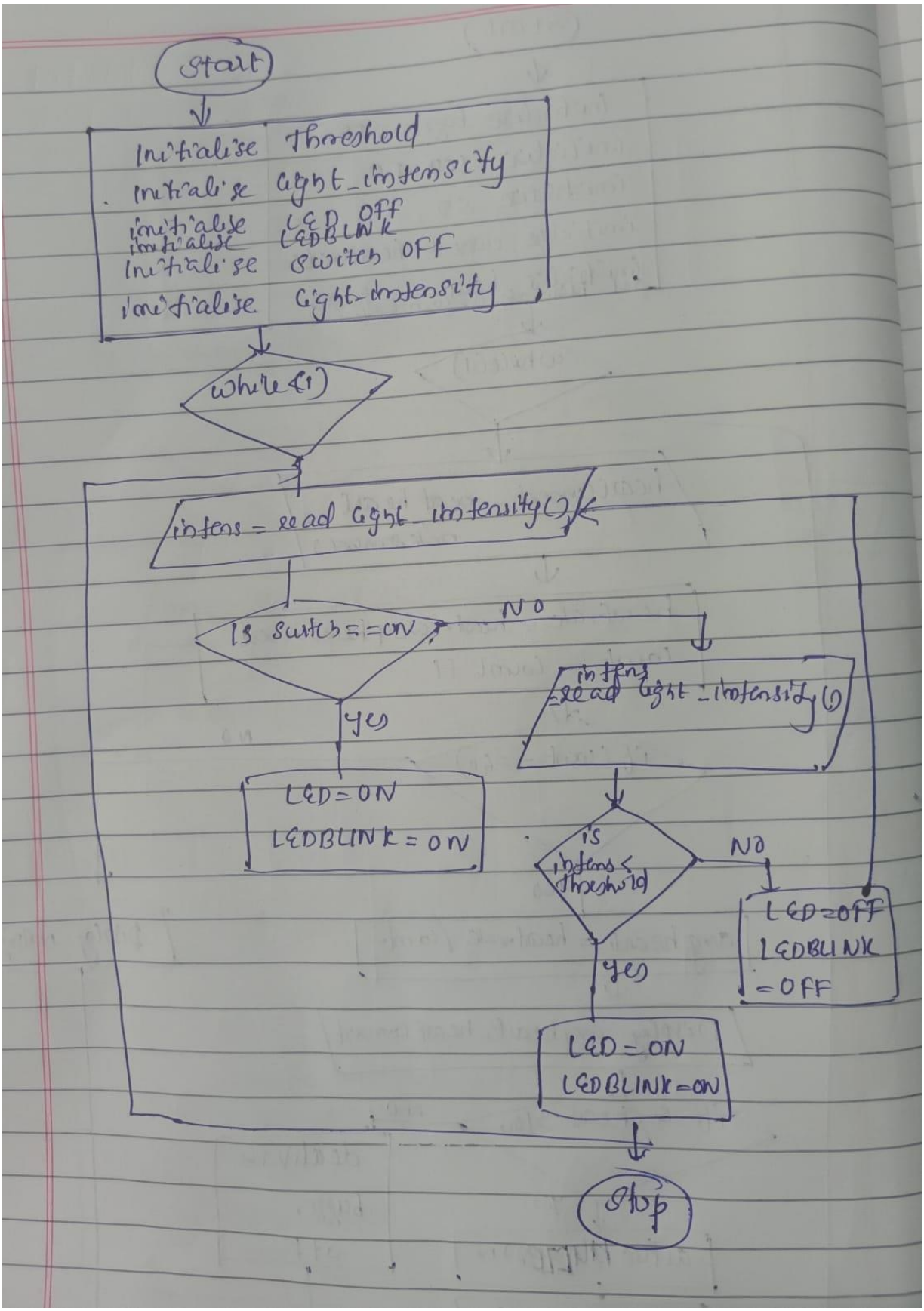
    LEDBLINK=ON

Else

    LED=OFF

    LEDBLINK=OFF

Set wait(60) and repeat the loop.


start
Initialise Threshold
Initialise Light_intensity
Initialise LED OFF
Initialise LEDBLINK
Initialise Switch OFF
Initialise Light intensity
while(1)
intens = read light intensity()
Is switch == ON
NO
yes
intens = read light intensity()
LED = ON
LEDBLINK = ON
Is intens < threshold
NO
yes
LED = OFF
LEDBLINK = OFF
LED = ON
LEDBLINK = ON
Stop

QN6.Digital Stopwatch

Initialize start,stop and reset button

Initialize LCD

Initialise SD card

Check start button

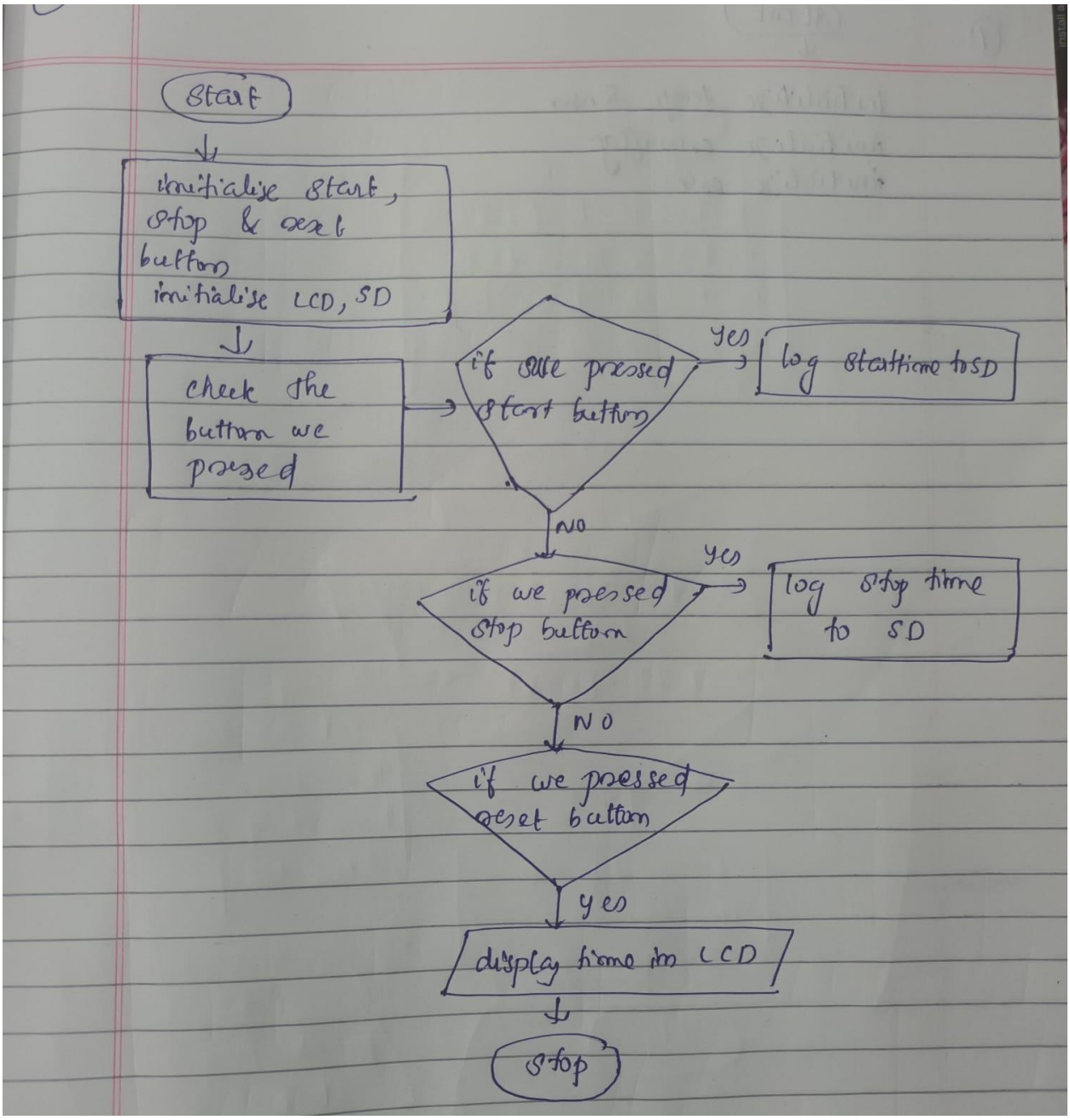
If we press start button then

    Log start_time to SD

Check reset button

If we press reset button then

    Print time in LCD



QN7.Temperature logging system

Initialize temp_sensor

initialise errorflag

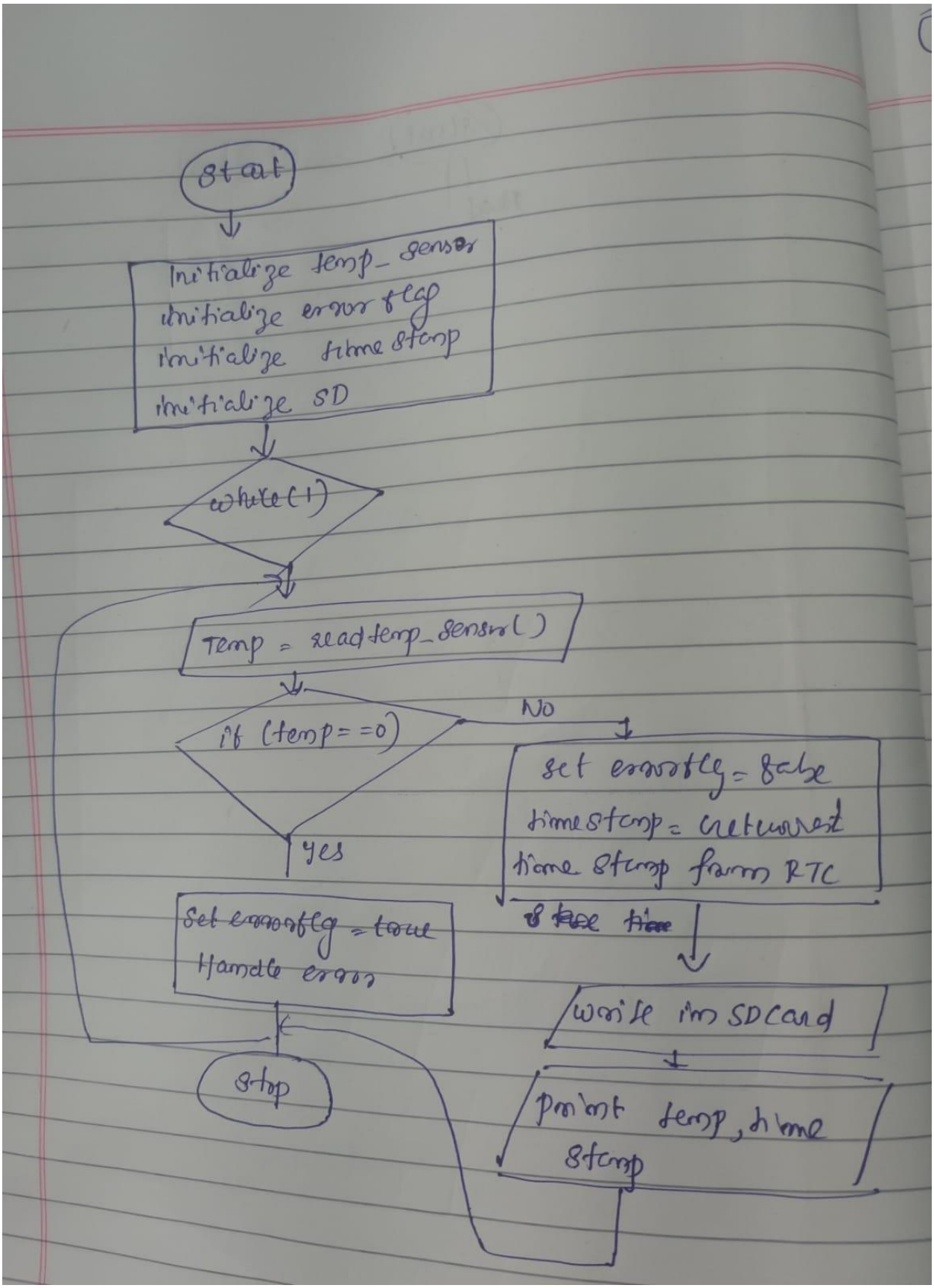
Initialize timestamp

Initialize SD

While(1)

Temp=read temp_sensor()

```
If(temp==0)
        Set errorflag=true
        Handle error
Else
        Set errorflag=false
Set timestamp=Get current time from RTC
Store timestamp and Temp to SD
If user requests data then
        Retrieve data from storage display
Wait(10 minutes) and repeat the loop
```


start
Initialize temp_sensor
initialize error flag
initialize time stamp
initialize SD
while(1)
Temp = read temp_sensor()
if (temp==0)
No
yes
set errorflg = false
timestamp = get current
time stamp from RTC
write in SD card
print temp, time
stamp
set errorflg = true
Handle error
stop

QN8.Bluetooth controlled robot

initialize Bluetooth

initialize LED

read data from Bluetooth
if command!=Null then

process commands

if command is "forward"

set device to move forward

provide feedback through LED

else if command is "backward"

set device to move backward

provide feedback through LED

else if command is "right"

set device to move right

provide feedback through LED

else if command is "left"

set device to move left

provide feedback through LED

else if command is "speed"

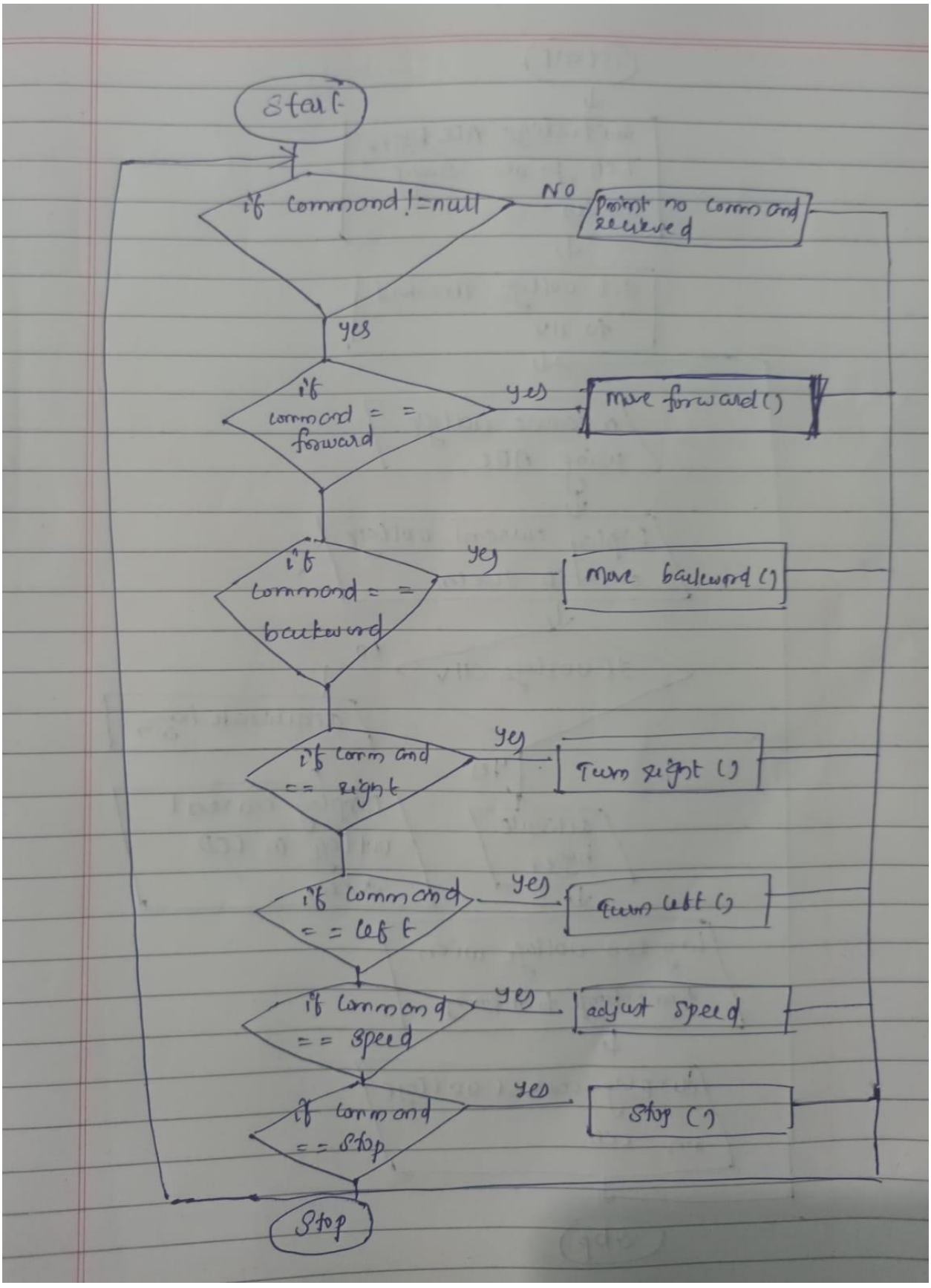
adjust device speed

else if command is "stop"

else

print no command recieved

stop device


Start
if command != null
NO
Print no command received
yes
if command == forward
yes
move forward()
if command == backward
yes
move backward()
if command == right
yes
Turn right()
if command == left
yes
Turn left()
if command == speed
yes
adjust speed
if command == stop
yes
stop()
Stop

qn9.Battery monitoring system

START

Initialize ADC

Initialize buzzer

Initialize LCD

Initialize power-saving modes

Set voltage threshold to 11V

For every minute

Measure voltage using ADC

Display current voltage on LCD screen

IF voltage < 11V THEN

Activate buzzer

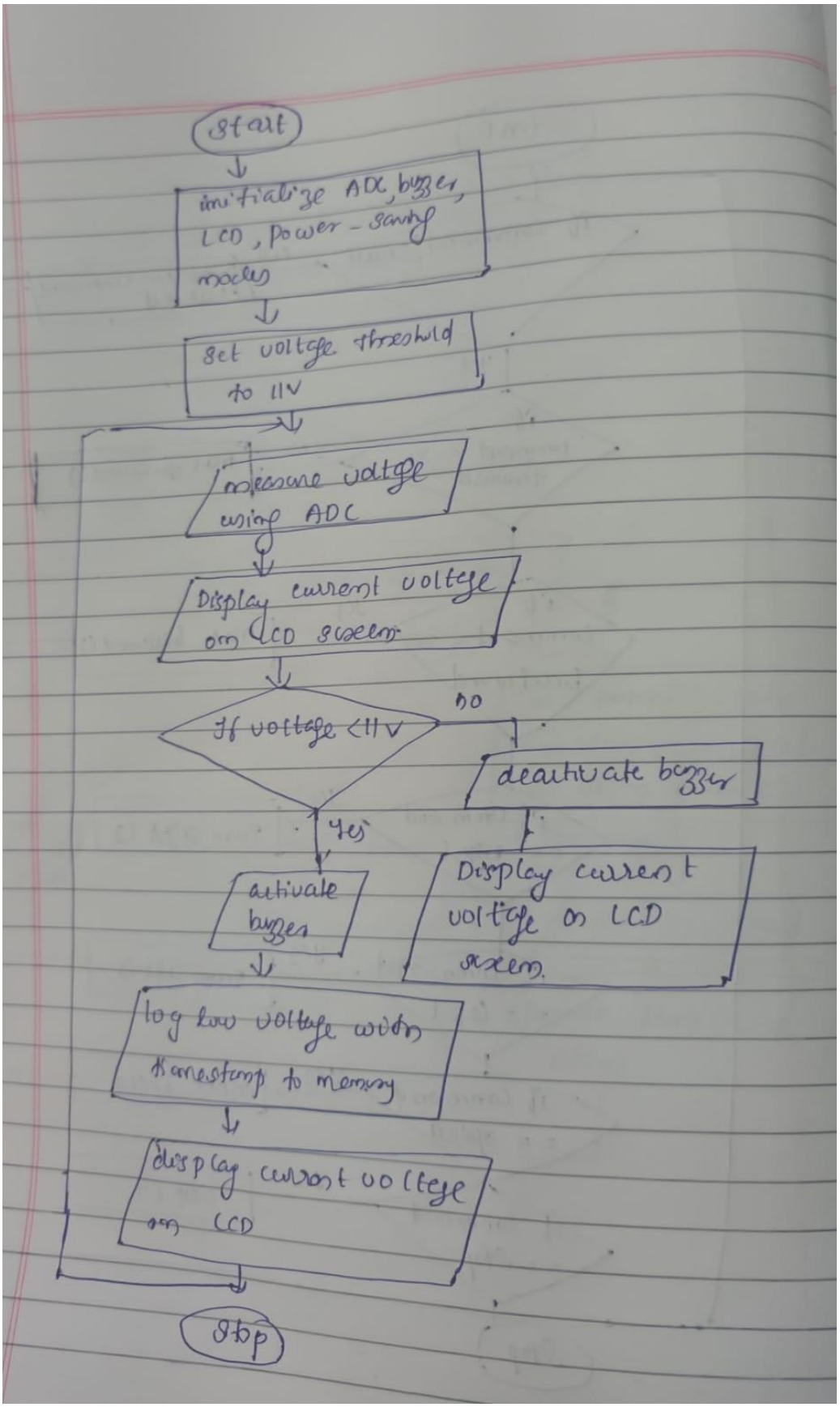
Log "Low voltage alert" with timestamp to memory

Display current voltage on LCD screen

ELSE

Deactivate buzzer

Display current voltage on LCD screen


start
initialize ADC, buzzer, LCD, Power-saving modes
Set voltage threshold to 11V
measure voltage using ADC
Display current voltage on LCD screen
If voltage <11V
No
deactivate buzzer
Yes
activate buzzer
Display current voltage on LCD screen
log low voltage with timestamp to memory
display current voltage on LCD
stop

Q10.RFID based access control system

Initialize RFID reader

Initialize buzzer

Initialize SD

WAIT for RFID tag scan

IF RFID tag scanned THEN

GET scanned RFID tag

IF scanned RFID tag is in authorized list THEN

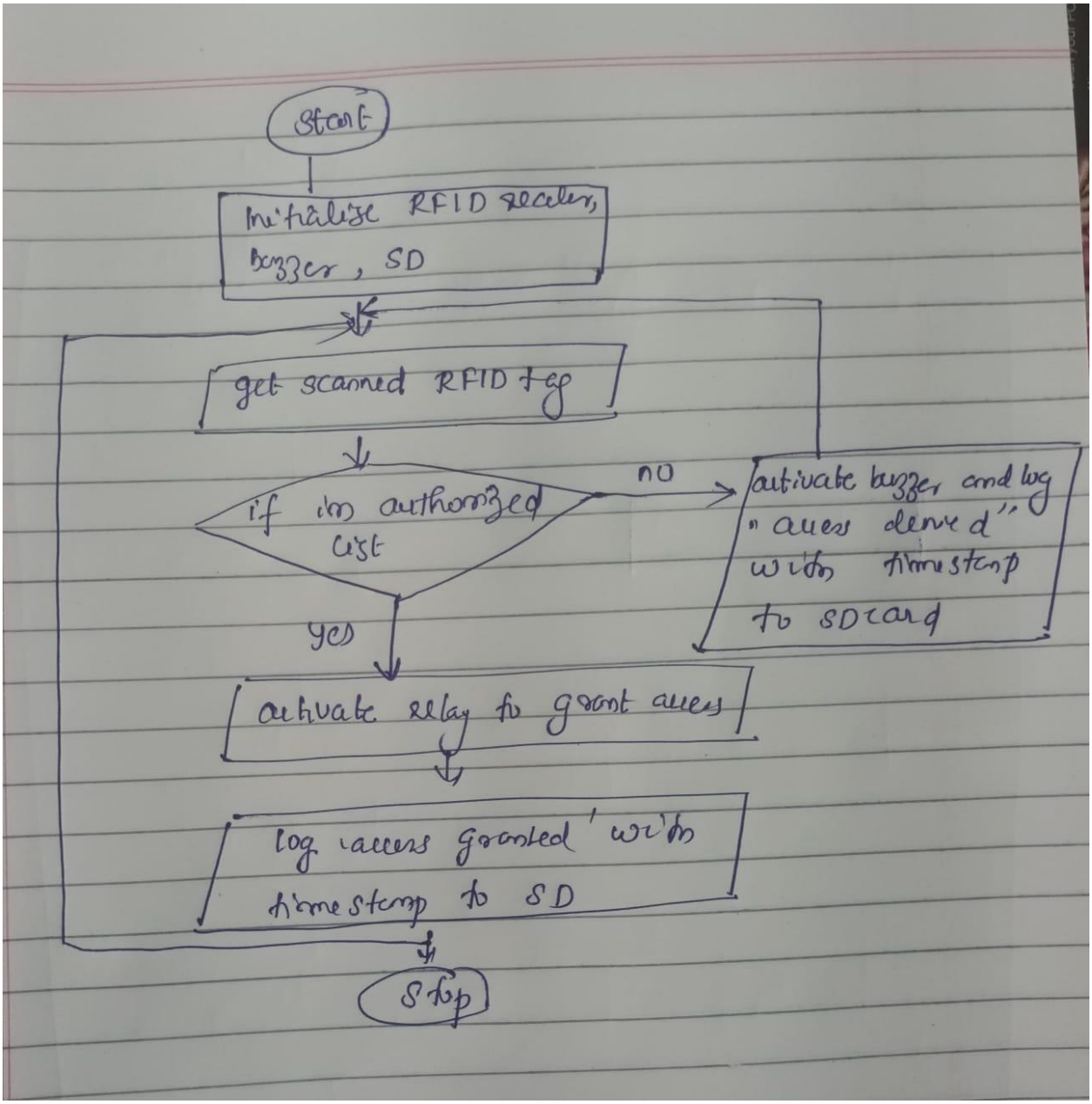
Activate relay to grant access

Log "user access is granted" with timestamp to SD

ELSE

Activate buzzer to deny access

Log "user access is denied" with timestamp to SD


Start
Initialize RFID reader, buzzer, SD
get scanned RFID tag
if in authorized list
no
activate buzzer and log "access denied" with timestamp to SD card
yes
activate relay to grant access
log 'access granted' with timestamp to SD
Stop