Examining Synchronization Mechanisms for Smart Home Application

By: Artiom Suruc, Tudor Ciutacu, Anna Manucharyan

TABLE OF CONTENTS

Smart Home applications

Smart Home applications transform traditional homes into technologically advanced and interconnected environments

A network of devices and systems automate and control various aspects of daily life, such as:

- lighting,
- security,
- temperature,
- entertainment, and more

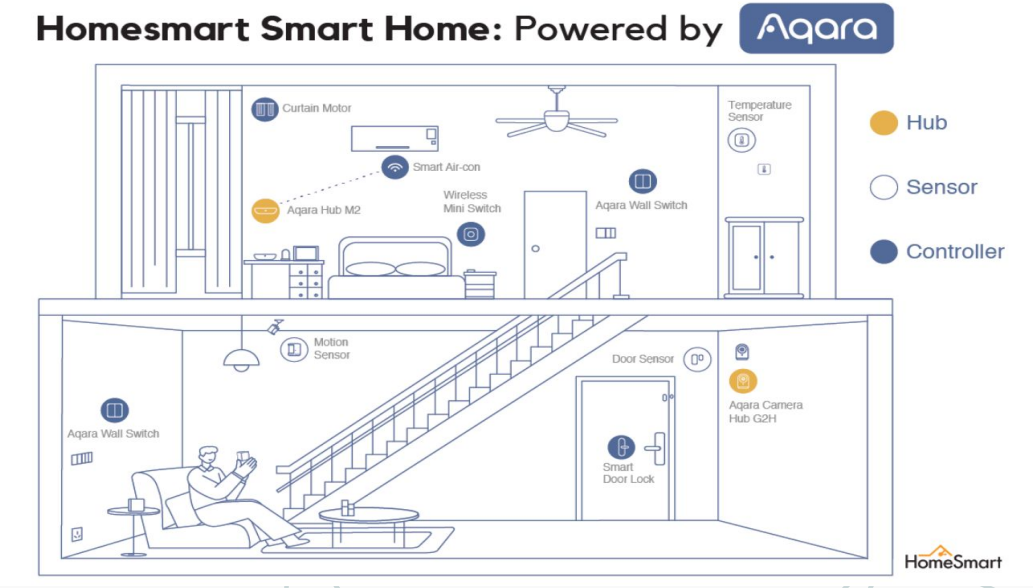
Robust synchronization mechanisms are needed to ensure seamless communication and coordination between different devices within a smart home system

Without proper synchronization, smart home devices may encounter conflicts, delays, or inconsistent behavior, undermining the overall functionality and usability of the system.

Smart Home Systems

Advantages	Challenges
Energy efficient	Costly
Secure	Battery life
Comfort and customisation	Internet dependence
Ease of access and convenience	Privacy concern

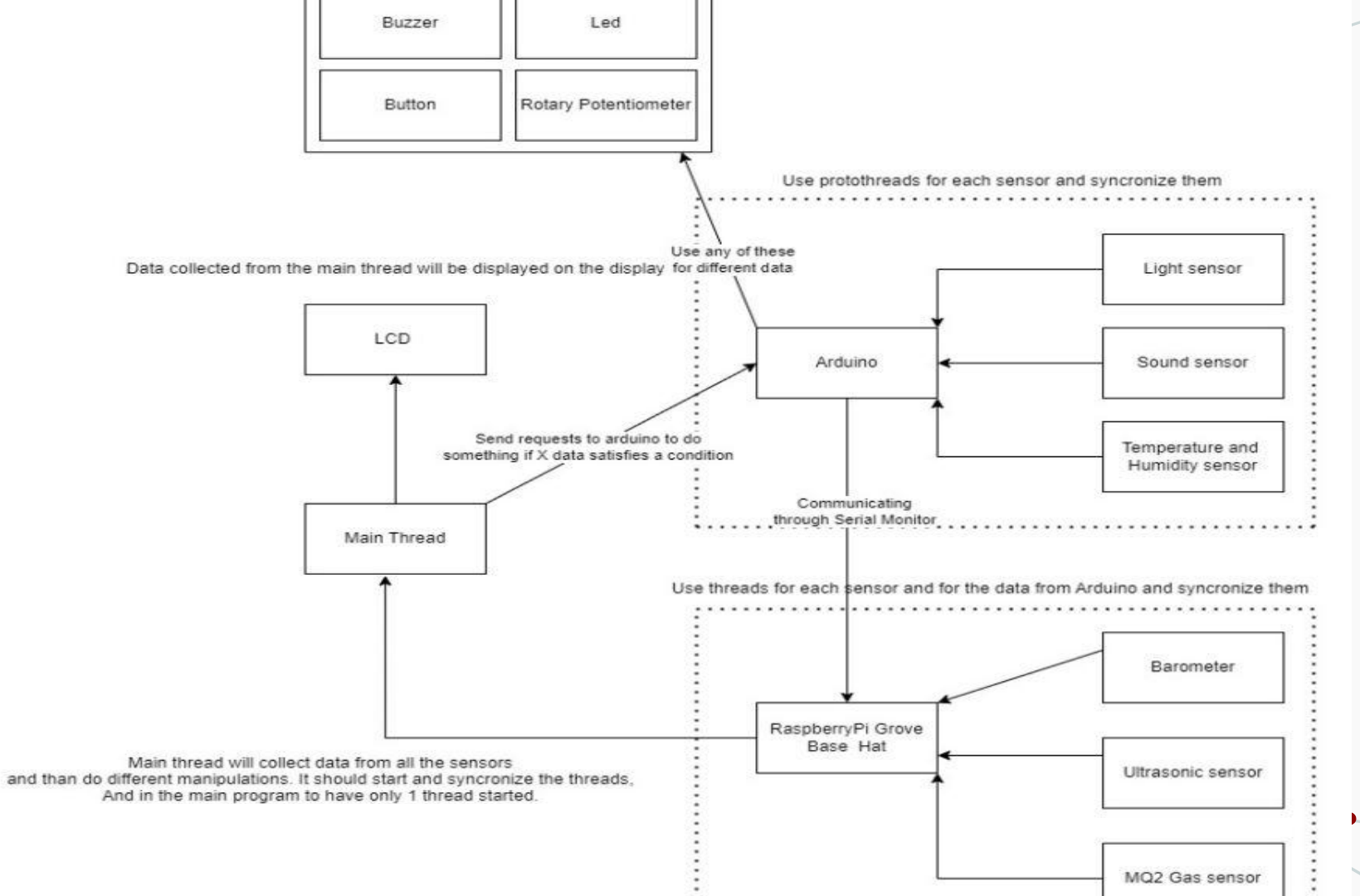
Example of a Smart Home Application: Singapore



Currently, the smart household penetration in Singapore is approximately 33.4% and Statista expects the number to hit 68% by 2026

It is estimated that Smart Home application reduces its energy costs by 5 to 10 percent

WHAT WE ARE WORKING ON


Buzzer
Led
Button
Rotary Potentiometer
Use protothreads for each sensor and syncronize them
Use any of these
for different data
Data collected from the main thread will be displayed on the display
LCD
Arduino
Light sensor
Sound sensor
Temperature and
Humidity sensor
Send requests to arduino to do
something if X data satisfies a condition
Communicating
through Serial Monitor
Main Thread
Use threads for each sensor and for the data from Arduino and syncronize them
Barometer
RaspberryPi Grove
Base Hat
Ultrasonic sensor
MQ2 Gas sensor
Main thread will collect data from all the sensors
and than do different manipulations. It should start and syncronize the threads,
And in the main program to have only 1 thread started.

Sensors

Arduino

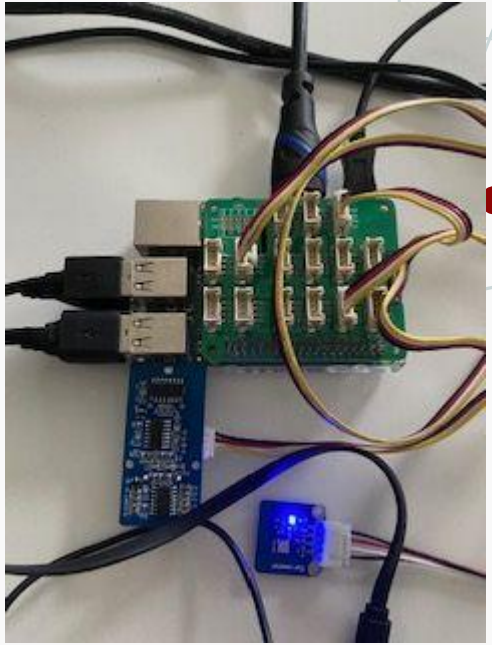
- Temperature and Humidity
- Light
- Sound

Raspberry Pi

- Barometer
- Gas (Air Quality)
- Ultrasound

Arduino Grove Beginner Kit

Raspberry Pi



Threads

Arduino

01

Arduino thread collects data from the Arduino sensors (temperature and humidity, light, and sound)

Barometer
Ultrasound
Gas sensor

02

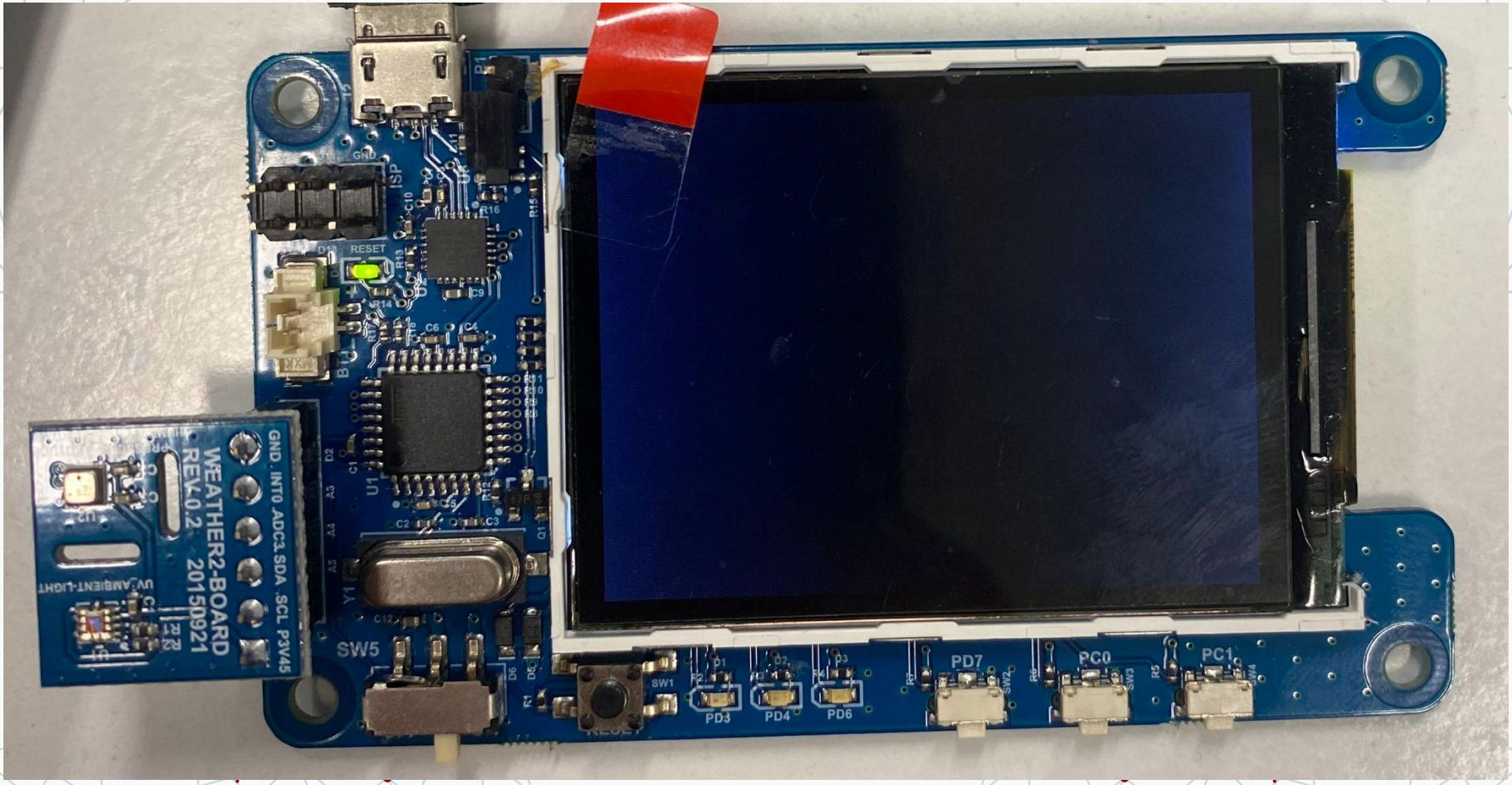
We have three separate threads for the above mentioned sensors

Main

03

Main thread integrates all the threads and starts all the processes

LCD Odroid Show-2



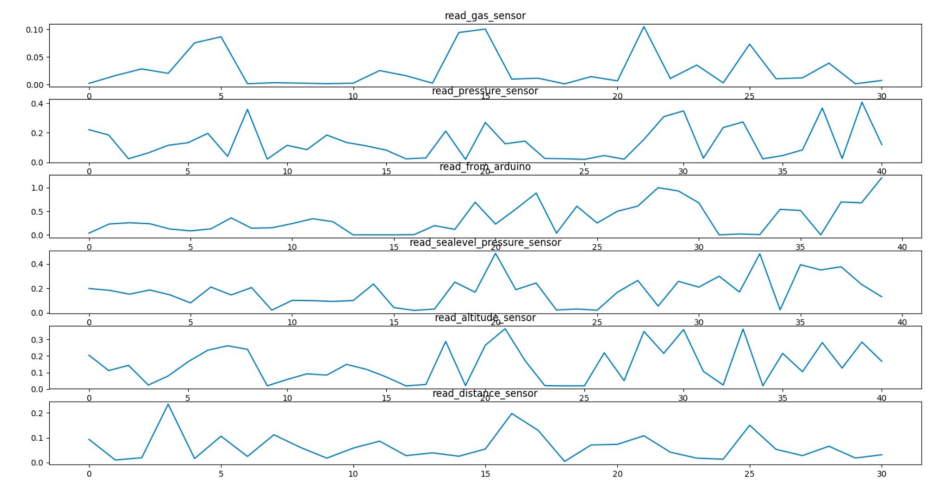

Distance: 6.14 cm
WEATHER2-BOARD
REV 0.2
20150921
GND INT0 ADC3 SDA SCL P3V45
PD3
PD4
PD6
PD7
PC0
PC1
SW1
SW5

To solve this issue we used a shared library in C vt_100command.c and used Python to C Interface to integrate it into our program


Temperature: 26.30C
Humidity: 68
Light level: 2841ux
Sound level: 295db
Gas: 58
Distance: 4.24cmmm
Pressure: 100107.00Pa
Altitude: 101.90m
Sea-lvl Pressure: 100114Pa
WEATHER2-BOARD
REV 0.2 20150921
SW5
OFF
ON
RESET
PD3
PD4
PD6
PD7
PC0
PC1

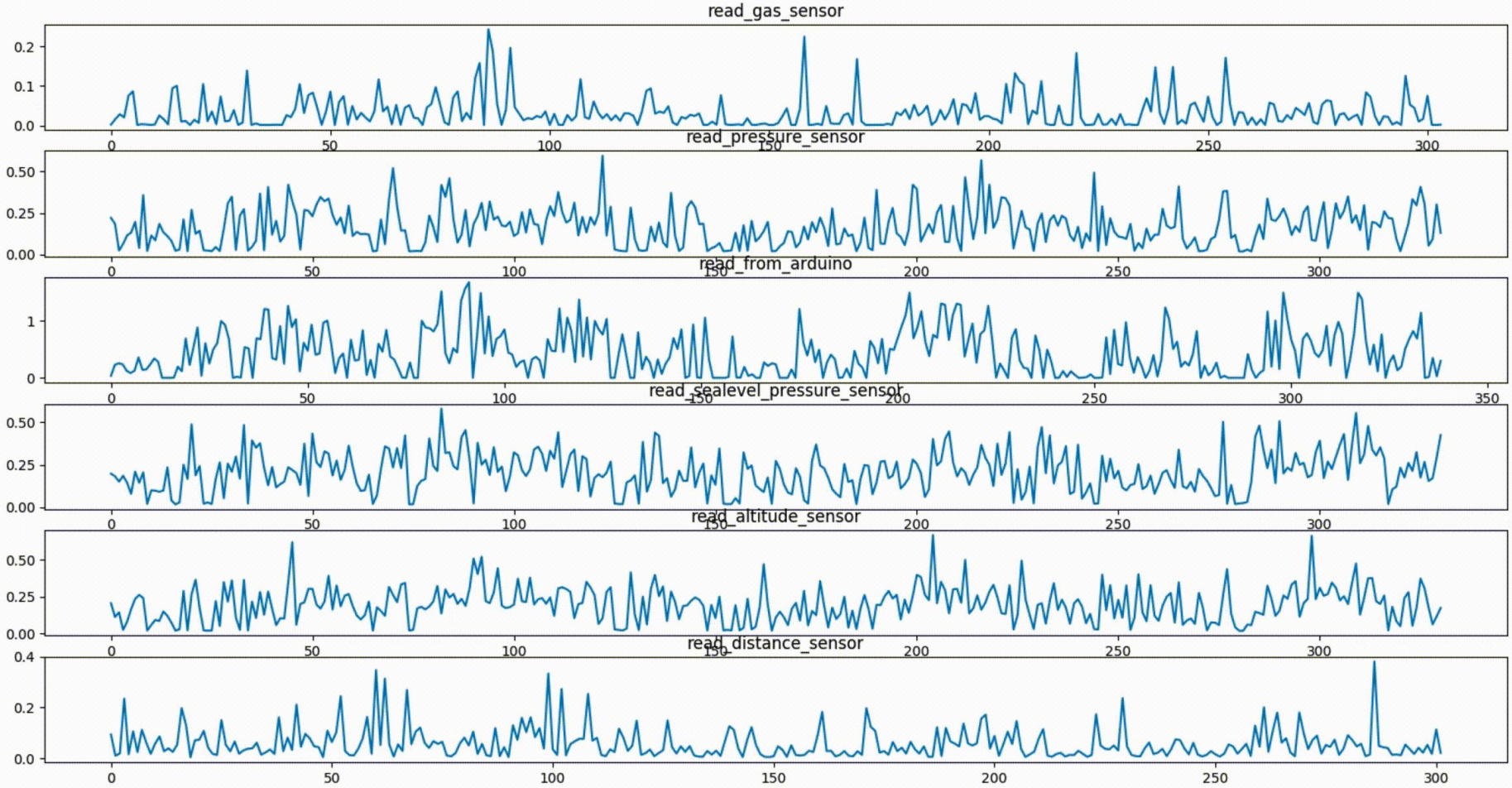
Results

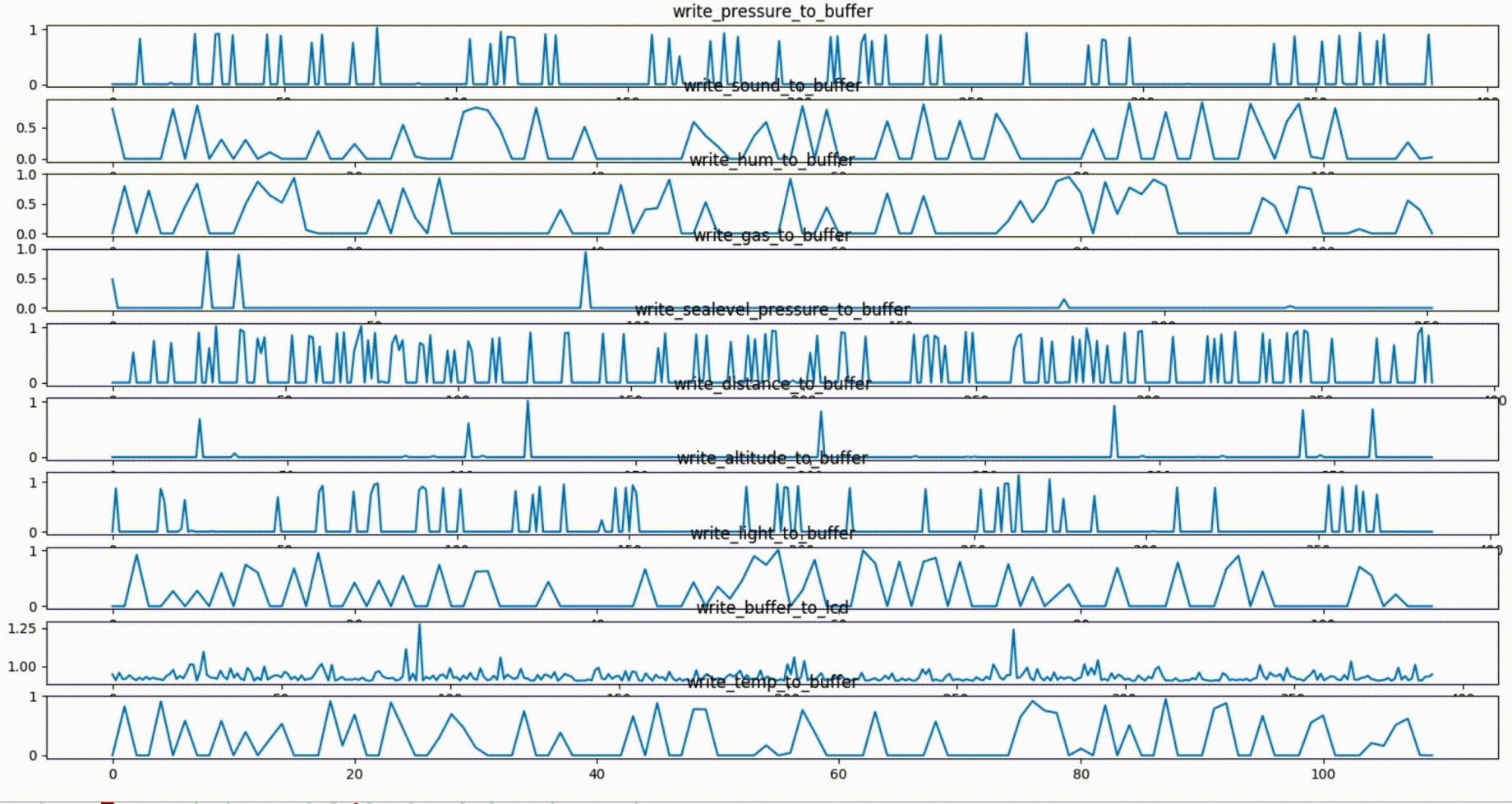
Time



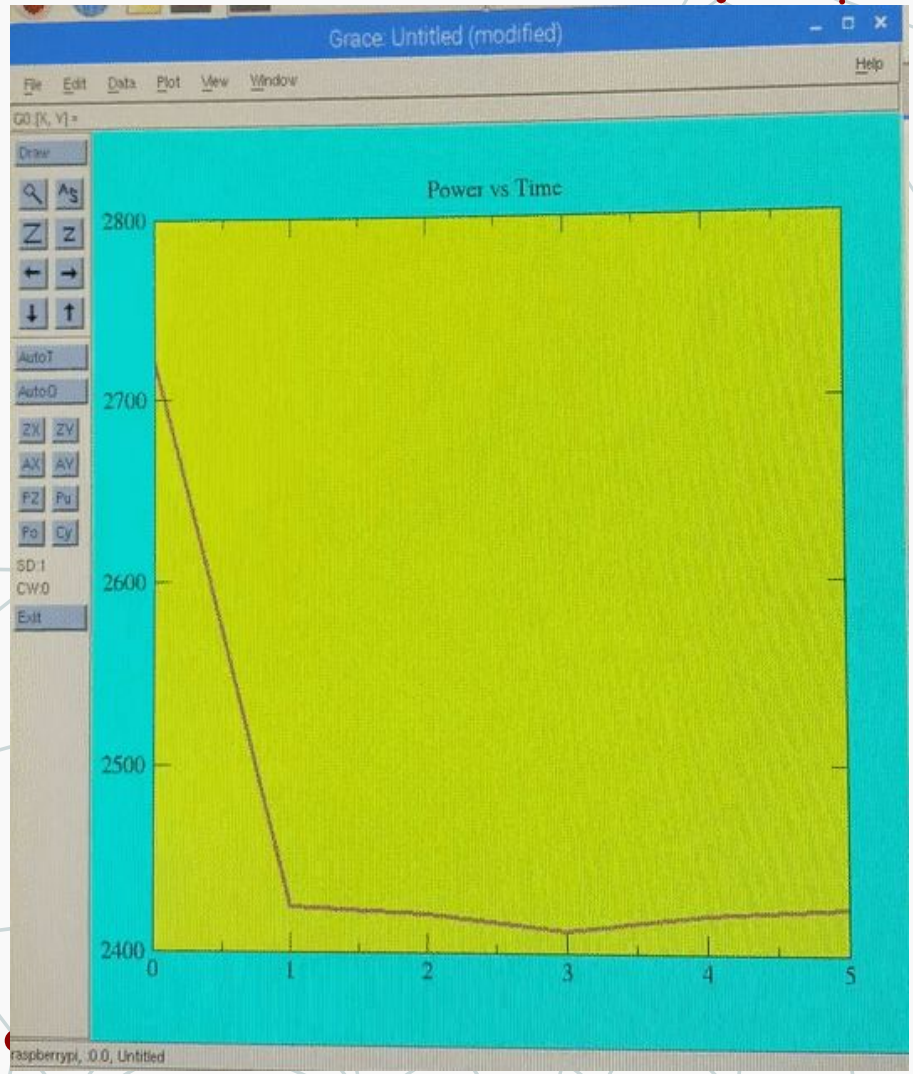
Energy

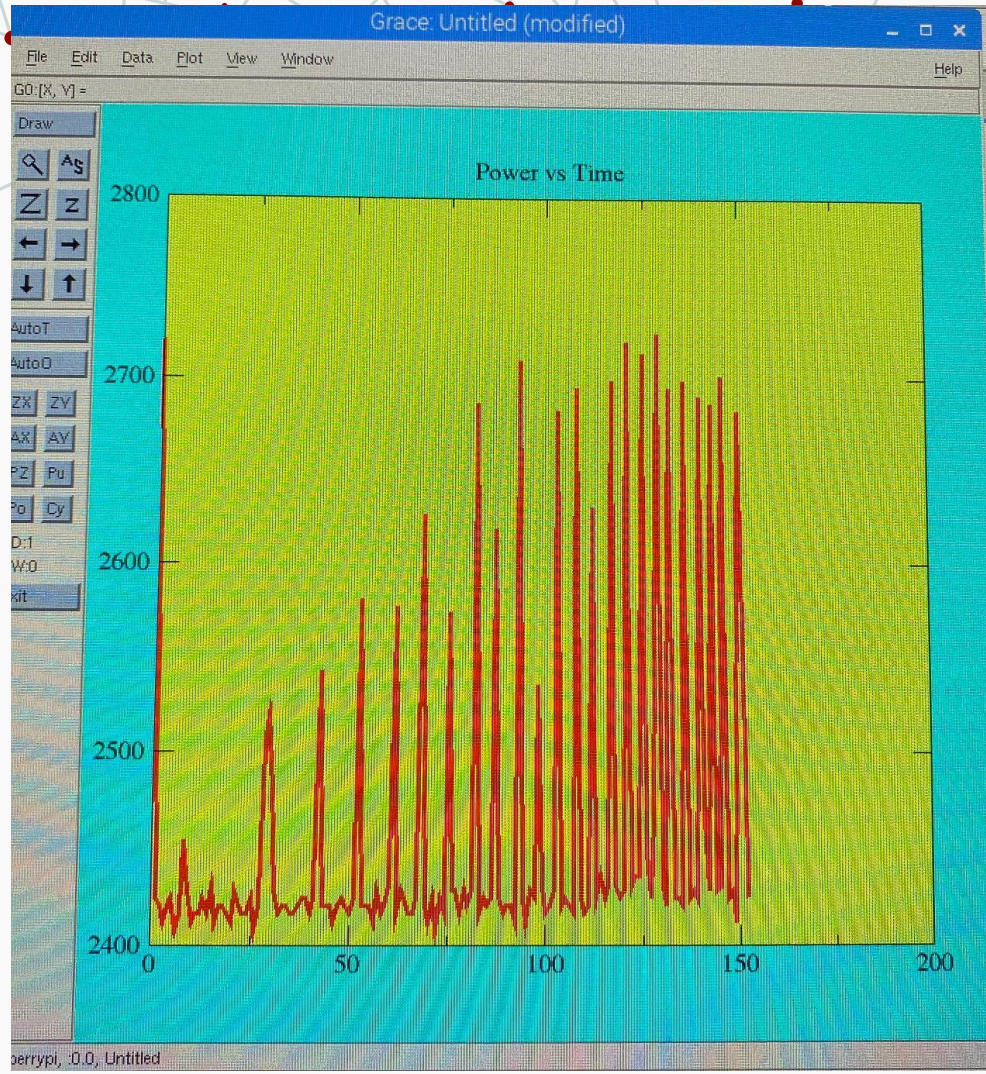
For measuring the time we put a timer before reading and timer after reading, then we subtract the two values and send it in the buffer. We do the same for measuring the writing time


read_gas_sensor
read_pressure_sensor
read_from_arduino
read_sealevel_pressure_sensor
read_altitude_sensor
read_distance_sensor


write_pressure_to_buffer
write_sound_to_buffer
write_hum_to_buffer
write_gas_to_buffer
write_sealevel_pressure_to_buffer
write_distance_to_buffer
write_altitude_to_buffer
write_light_to_buffer
write_buffer_to_lcd
write_temp_to_buffer
0
20
40
60
80
100

We measured the energy consumed by the system with Telnet, we read the values one by one and sent them to Xmgrace for graphing it. We update the graph every time a new value is read from telnet


Grace: Untitled (modified)
File Edit Data Plot View Window Help
G0:[X, Y] =
Draw
AutoT
AutoO
ZX ZV
AX AV
FZ Pu
Po Cy
SD:1
CW:0
Exit
Power vs Time
2800
2700
2600
2500
2400
0
1
2
3
4
5
raspberrypi, :0.0, Untitled


Grace: Untitled (modified)
File Edit Data Plot View Window Help
G0:[X, Y] =
Draw
AutoT
AutoO
ZX ZV
AX AV
PZ Pu
Po Cy
D:1
W:0
xit
Power vs Time
2800
2700
2600
2500
2400
0
50
100
150
200
berrypi, :0.0, Untitled

- For measuring the energy consumed by individual components we measured the initial total energy consumed by the program, and then disconnected the components one by one, measured the energy and subtracted from the total

Energy Consumption

LCD

359mW

Arduino

156mW

Gas Sensor

66mW

Barometer

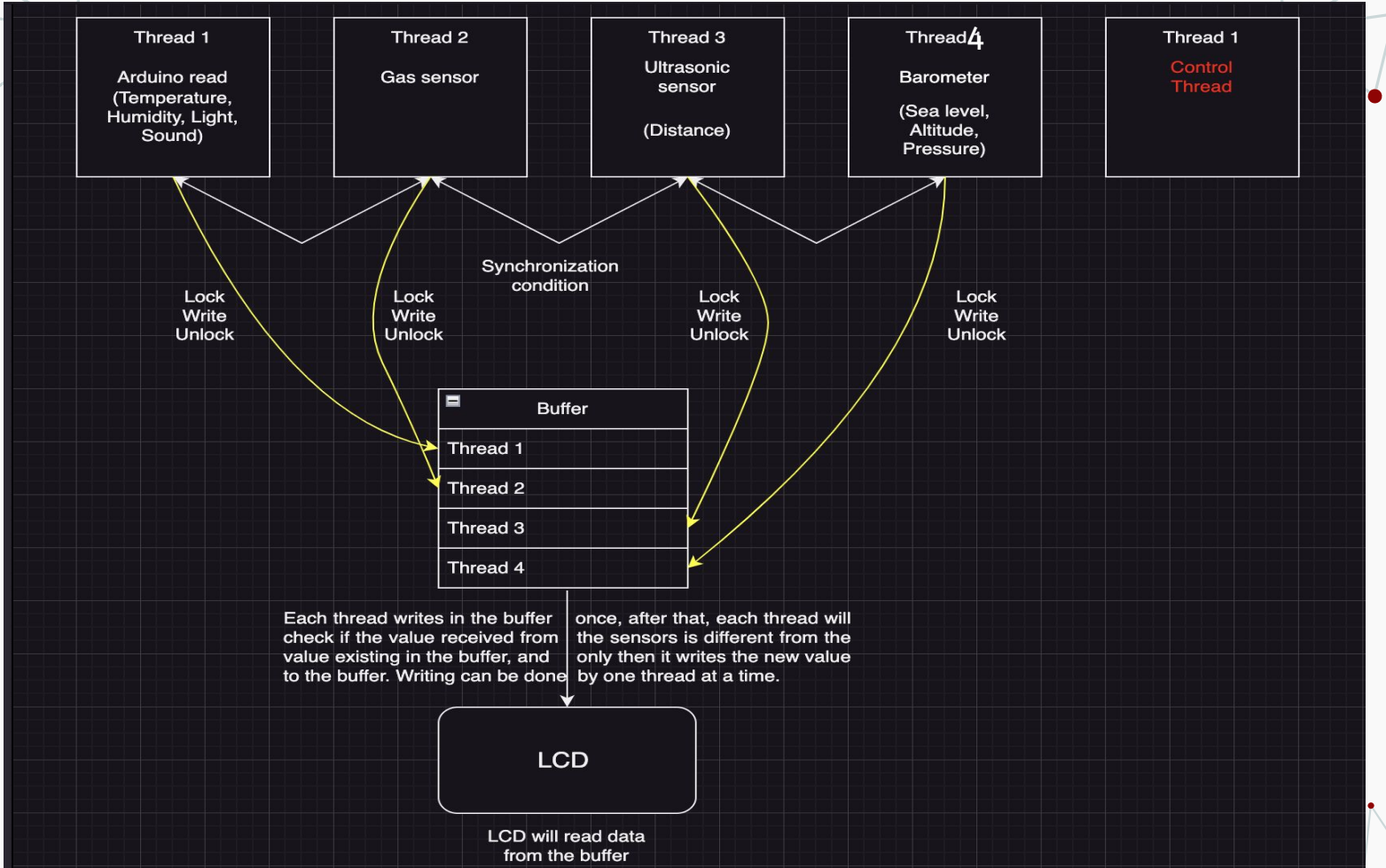
7mW

Ultrasound

34mW

Total (when both the python program and c program are executed)

878mW


Thread 1
Arduino read (Temperature, Humidity, Light, Sound)
Thread 2
Gas sensor
Thread 3
Ultrasonic sensor
(Distance)
Thread 4
Barometer
(Sea level, Altitude, Pressure)
Thread 1
Control Thread
Lock
Write
Unlock
Lock
Write
Unlock
Synchronization condition
Lock
Write
Unlock
Lock
Write
Unlock
Buffer
Thread 1
Thread 2
Thread 3
Thread 4
Each thread writes in the buffer once, after that, each thread will check if the value received from the sensors is different from the value existing in the buffer, and only then it writes the new value to the buffer. Writing can be done by one thread at a time.
LCD
LCD will read data from the buffer

Future Development of the Project

-Energy Efficiency Monitoring

-Air Quality Monitoring and Control

-Integration with Personal Digital Assistant

-Security System

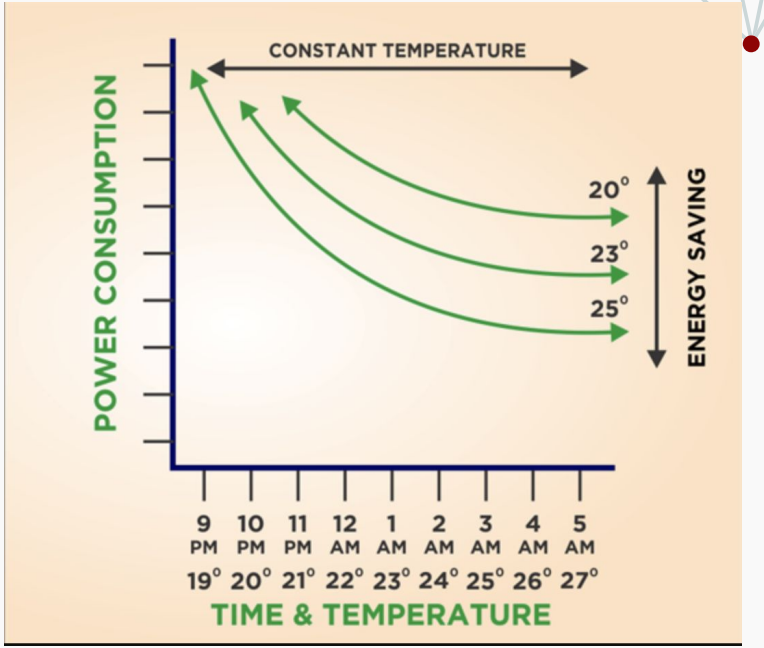
-Machine Learning and Predictive Analytics

-Expand Sensor Network

Energy Efficiency Monitoring

Optimizing Home Energy Consumption Using IoT

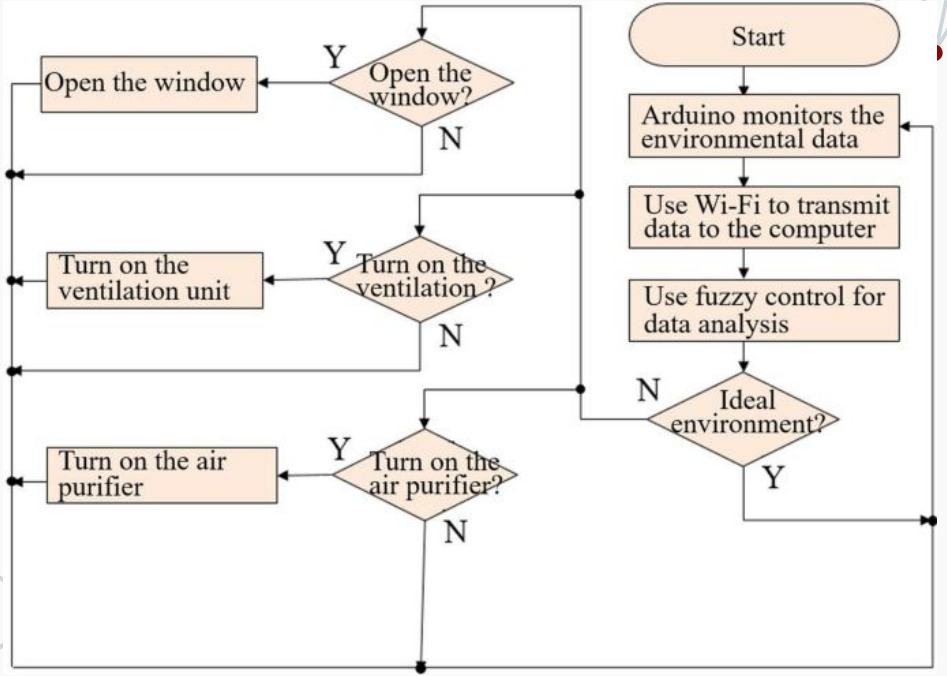
-Utilize light and temperature sensors to adjust home's heating and lighting system.
-Monitor energy usage patterns for identifying inefficiencies and eliminating waste.
-Automated adjustments for optimal energy consumption, based on sensor data.



Air Quality Monitoring and Control

Ensuring a Healthier
Home Environment

-Deploy gas sensor, barometer, and humidity sensor to monitor air quality.
-Alert system for when air quality drops (high CO2 levels, low humidity).
-Automated control of IoT devices like air purifiers or humidifiers for air quality improvement.



Integration with Personal Digital Assistant

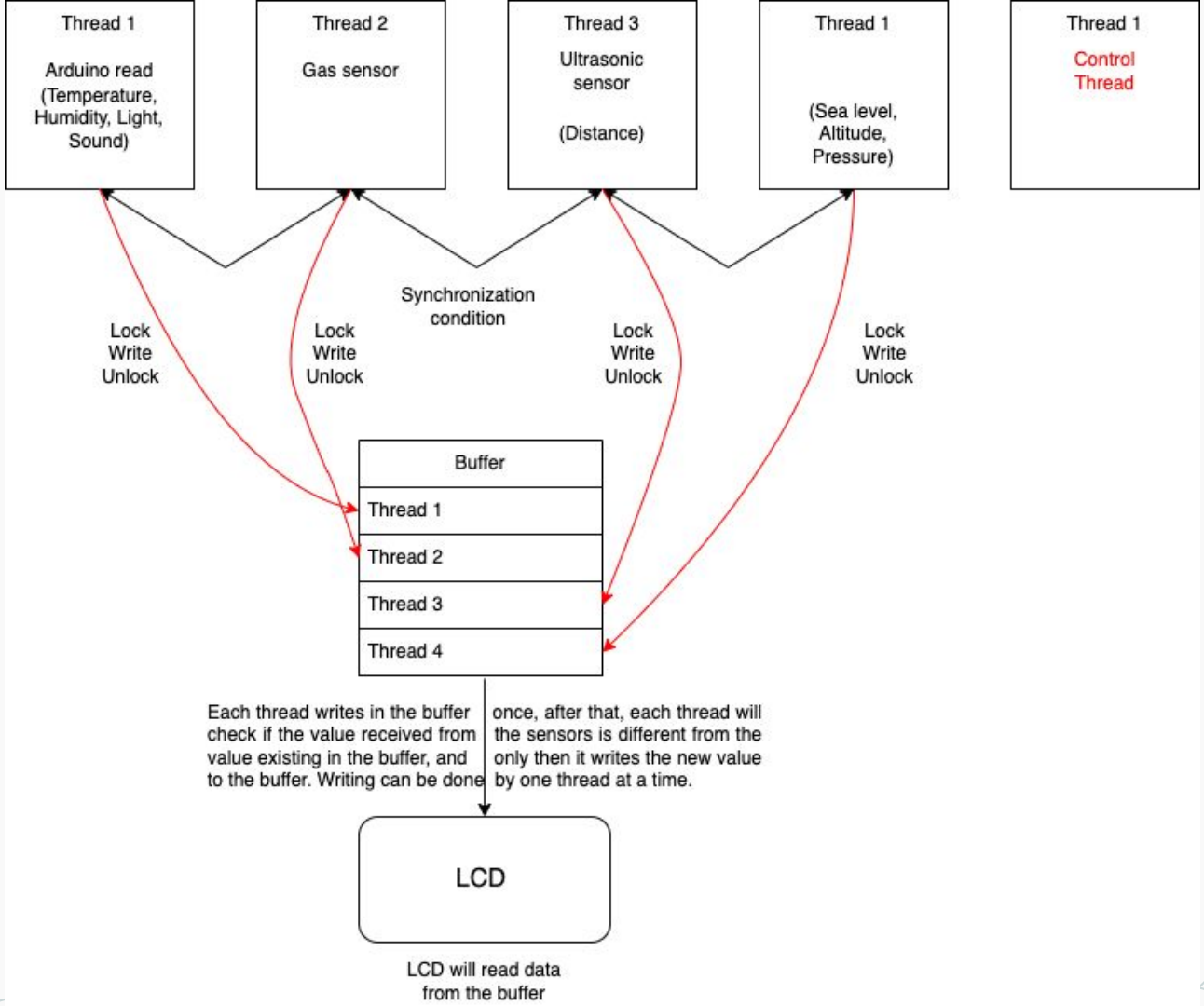
Voice-Controlled Smart Home

-Enhance user interaction through compatibility with Siri, Google Assistant, or Amazon Alexa.
-Voice commands for checking home status and controlling home environment ("Increase the temperature", "Dim the lights").
-Enable users to access and control their smart home system in a hands-free manner.

Developing a cross-platform application

THANK YOU!

REFERENCES


Thread 1
Arduino read (Temperature, Humidity, Light, Sound)
Thread 2
Gas sensor
Thread 3
Ultrasonic sensor
(Distance)
Thread 1
(Sea level, Altitude, Pressure)
Thread 1
Control Thread
Synchronization condition
Lock
Write
Unlock
Lock
Write
Unlock
Lock
Write
Unlock
Lock
Write
Unlock
Buffer
Thread 1
Thread 2
Thread 3
Thread 4
Each thread writes in the buffer once, after that, each thread will check if the value received from the sensors is different from the value existing in the buffer, and only then it writes the new value to the buffer. Writing can be done by one thread at a time.
LCD
LCD will read data from the buffer