Secure Smart Home System using STM32

Computer Interfacing (CSE360) Project
BRAC University
Fall 2024

Section: 3
Group 5
Group Members:

1. Saimum Reza Siam
2. Fahim Nahar Shaily
3. Md. Adnan Hossain Sadi
4. Mehrab Muhtasim Sayem

Submission Date: 26 December 2024

Abstract:

The Project aims to solve the security and privacy-based challenges in IoT-based smart home systems. A secure, scalable solution is developed using an STM32F446RE microcontroller. The system uses sensors and modules such as air quality sensor, DHT11 temperature, and humidity sensor, PZEM004T power module, serial camera, SD module, and relays. These sensors and modules help to monitor the environmental and electrical parameters i.g air quality, temperature, voltage, and power consumption. It ensures the control over home appliances in a smart home system. Moreover, the system uses communication protocols such as UART, GPIO, ADC, and FMC to connect the peripherals with the STM32 microcontroller. MQTT and HTTP protocol are used to handle data exchange with cloud and camera transmission. The system allows a user to monitor and control the home system remotely via a web or mobile based application system. It also looks after any unusual or suspicious activity in the smart home system to detect any possible intruder in the system. Thus it protects the security and privacy of the IoT-based smart home system. The project has a scope for future development i.g end to end encryption. Scalability and automation can be implemented in future upgrades of the project. Therefore, it is an IoT-based smart home system implementing the knowledge of interfacing with proper communication protocol and security measures.

Introduction

Problem Statement:

Solving the security threats and privacy vulnerabilities of today's IoT-based smart home system using computer interfacing knowledge while ensuring seamless system control and monitoring.

Objective:

The objective of this project is to develop a secure smart home system that can monitor and control the devices attached to the system in a secure way. The Project successfully interfaces various sensors with an STM32 microcontroller. It also successfully implements the communication protocols for the hardware interfacing required for the project.
It aims to collect data from different sensors and modules of the smart home system such as voltage, current, power, energy, frequency, temperature, humidity, and air quality.
Finally, a user gets to remotely monitor the smart home system using a web or mobile application. It also tracks any suspicious activity in the system to identify a security or privacy threat while ensuring seamless home system control and monitoring.

Significance:

The Project showcases the way to interface multiple sensors and modules with an STM32 microcontroller. It implements the fundamentals of IoT-based systems such as data acquisition, transmission, monitoring, and control. This project has significant real-world applications. There were 16.6 billion IoT devices connected in 2023, which gets 13% growth at the end of 2024 (Sinha, 2024). This project can help to ensure privacy and secure control over IoT-based systems like Smart Home System.

Interfacing Design

A. Interfacing Components:

1. STM32F446RE Microcontroller: Responsible for reading data from all sensors, processing the sensor data, implementing control logic to determine when to turn the home appliances on/off using relays as switches, communicating with the SD card for storing data and connecting to the user interface. (STM32F446RE datasheet, n.d.)

2. Air Quality Sensor: Measures the level of pollutants in the air, such as VOCs (volatile organic compounds), carbon monoxide, particulate matter, and other gases. The sensor provides an analog output signal proportional to the level of pollutants. It uses an ADC or analog to digital converter to send a signal to the microcontroller since it can only receive digital data. (Grove - Air Quality Sensor v1.3, n.d.)

3. DHT11 Temperature & Humidity Sensor: Measures both temperature and relative humidity every certain seconds. It provides digital output signals as the measured values. The single-wire protocol of the DHT11 requires a GPIO pin for manually controlling the timing and state of the pin to send and receive data. The STM32 microcontroller uses a GPIO pin configured as an output to initiate communication and as an input to read the data from the sensor.(DHT11 Technical Data Sheet, 2015)

4. PZEM004T Power Module: Measures various electrical parameters, including voltage(V), current(I), power(P), energy(E), power factor(PF), and frequency(F), of an electrical load. It uses UART for its data transmission and sends the parameters serially. (PZEM-004T Datasheet, 2019)

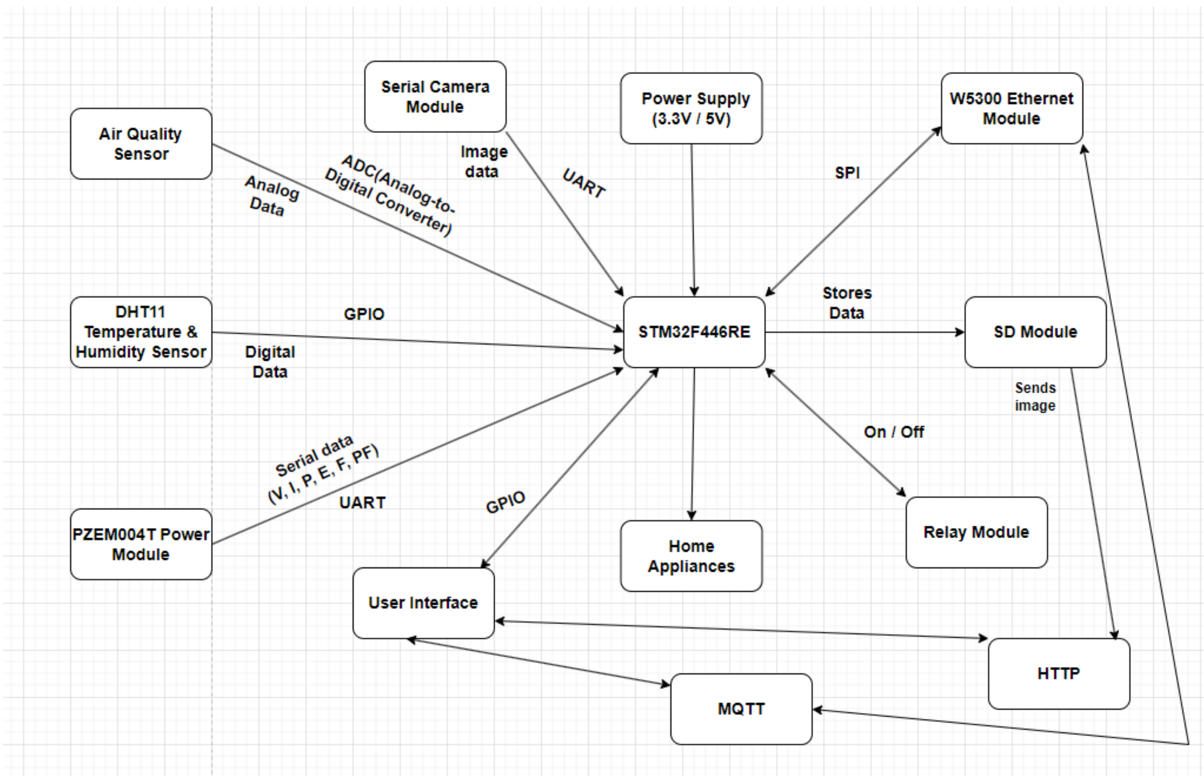
5. Serial Camera Module: Captures images and transmits them to the microcontroller by a serial communication interface (UART). (uCAM-II Datasheet, n.d.)

6. SD Module: Used as a storage for data logging. The microcontroller can store sensor data, images captured by the camera, and other relevant information on the SD card.

7. Relay Module: Act as electrically controlled switches. The microcontroller drives the relays to turn home appliances on/off depending on the control logic and sensor readings. It can be used to control various appliances and equipment with large currents. It is equipped with high-current relays that work under AC250V10A or DC30V 10A. (TS0011 Datasheet, n.d.)

8. W5300: The W5300 is an Ethernet controller chip designed for network connectivity in embedded systems and it acts as a bridge between the STM32F446RE and the Ethernet network. It uses the FMC protocol. (Wiznet, n.d)

B. System Block Diagram:



C. Interfacing Challenges:

1. **Signal Integrity:** Long wire lengths, improper grounding, and high-frequency signals can introduce noise and distortion, leading to data errors.
   Solutions: shielded cables, grounding, short signal traces.
2. **Noise:** Electrical noise from the power supply, electronic devices, or environment can interfere with sensor readings, microcontroller operation, and communication.
   Solutions: Use of a well-regulated power supply with adequate filtering to minimize noise on the power lines, Grounding, using shielding.
3. **Timing Issues:** Mismatched clock speeds or delays in data transmission can lead to communication errors, especially with timing-critical interfaces like UART and SPI.
   Solutions: Use of appropriate clock sources and ensuring proper synchronization between the microcontroller and peripherals.
4. **Sensor Compatibility:** Ensuring compatibility between sensor output signals and microcontroller input requirements e.g., voltage levels and signal types.
   Solutions: Use level shifters or op-amps for signal conditioning if necessary to match sensor output levels to the microcontroller's input range.
5. **Relay Switching Noise:** Switching relays can generate electrical noise that can interfere with other components and introduce errors in sensor readings or microcontroller operation.
   Solutions: Using diodes across the relay contacts to suppress the voltage spikes generated during switching and Isolating relay circuits from sensitive electronics.
6. **Data Integrity:** Data corruption can occur during data transmission or storage.
   Solutions: Implement error detection and correction mechanisms (e.g., parity checks, checksums) for data communication.

Communication Protocols:

Protocol Selection:

The protocols used in this project are UART, GPIO, ADC, and FMC.

UART is an asynchronous, full-duplex serial communication interface that is suitable for point-to-point connections where no clock signals are necessary.

In this project,

- UART is used to communicate with the PZEM-004T Power Module and the Grove Serial Camera.

FMC is a parallel, high-speed communication protocol that is used to access external peripherals such as memory or boards.

In this project,

- The W5300 TOE shield is connected to the neucleo board using the FMC communication module.

GPIO: Peripherals that don't fall under any specific communication protocol, follow GPIO or general purpose I/O protocols, which allows devices like sensors to communicate with the MCU through a digital 1/0 to control or read its state.

In this project,

- It is used in the Relay Module which controls appliances by toggling PC0, PC1, PC2, and PC4 pins connected via the J10 header of the W5300 TOE Shield.

- Also the DHT-11 Temperature and Humidity Sensor that Connects to the D3 pin on the Grove Base Shield to transmit digital data.

ADC is a microcontroller feature that allows communication with analog devices like sensors that produce different voltage levels as it converts analog voltage levels from sensors into digital values for processing.

- Here it is used for the Air Quality Sensor, analog output is read through the A0 pin of the Grove Base Shield, which is connected to the STM32's ADC input.

For communicating with the cloud, we are using the **MQTT** which is a machine-to-machine standards-based messaging protocol.

HTTP is used for transferring data from the STM32 microcontroller to the mobile app.

Protocol Justification:

UART: PZEM-004T Power Module and Grove Serial Camera both need full duplex ,faster data transfer for real-time data processing,UART offers low latency. It's natively supported by STM32

FMC: For real-time Ethernet data transmission. FMC guarantees dependable and low-latency connections. It is essential for managing fast data transfer with the W5300 TOE Shield.

GPIO: uses pins cycling between 1 and 0 to directly control devices. Suitable for low-frequency digital signals. Easy to configure. Directly integrates with DHT-11 and relays without any additional hardware.

ADC: Required for capturing continuous analog signals from the Air Quality Sensor. Doesn't need any external ADC hardware as it's a built-in feature of STM32. Necessary for precise measurements.

MQTT: MQTT is ideal for IoT applications where there are frequent, small data exchanges. It's suitable for low-bandwidth, high-latency environments. It has scalability for multiple peripherals and provides real-time bidirectional communication. It is necessary in this project for its publish(uploading sensor data to the cloud) and subscribe(controlling relays from app) system.

HTTP: HTTP is ideal for transferring images as part of a client-server model. The images captured by the Grove Serial Camera are sent to HTTP in order to let the app monitor its data. Http's versatility is making it the best option for this scenario.

Protocol Interoperability: The STM32 allows seamless communication between devices with different protocols as it was ensured that we only use the protocols that are directly supported by the STM32 such as GPIO, ADC, UART, FMC, and that no extra hardware components are needed. Error detection methods, like parity checks for UART and noise filtering for ADC, makes sure that the data isn't corrupted. GPIO provides simple digital control for devices like relays, while the buffers synchronize data from protocols with varying speeds.

Implementation Plan:

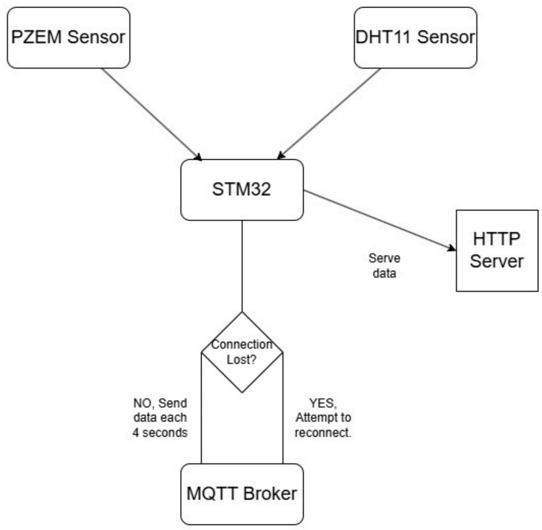
Methodology:

```
Enable UART2 clock
Configure GPIO pins for UART2 TX and RX
Configure UART2 baud rate, word length, stop bits, parity, and other parameters
Enable UART2
Wait for TX buffer to be empty
Send character 'ch' to UART2 transmit register
Wait for RX buffer to be full
Read character from UART2 receive register and return it
While not end of string:
  Send current character from buffer using UART_SendChar()
  Increment buffer index
While number of characters received is less than 'len':
  Receive character using UART_ReceiveChar()
  Store received character in buffer
  If received character is '\r' or '\n', exit the loop
```

```
Enable clock for GPIOA and GPIOC
Configure GPIOA Pin 1 as output (for LED)
Configure GPIOC Pin 13 as input (for button)
Initialize 'value' to 1 (button not pressed)
If button is pressed (GPIOC Pin 13 is low):
  Set 'value' to 0
If 'value' is 1:
  Turn on LED (set GPIOA Pin 1 high)
Else if 'value' is 0:
  Turn off LED (set GPIOA Pin 1 low)
```

```
Main file Pseudocode:
BEGIN main
  INITIALIZE GPIO, ADC, FMC, UART, network, and sensors
  TRY MQTT connection 3 times, SET mqtt_connect_flag TO 1 if successful
  WHILE true
    READ PZEM sensor data
    READ DHT11 data (temperature, humidity)
    START ADC, STORE VOC if conversion complete
    IF MQTT disconnected, TRY reconnect
    SEND data to MQTT
    DELAY 5 seconds
END main
FUNCTION send_to_cloud(sensor_data)
  FORMAT and PUBLISH data to MQTT broker
FUNCTION http_server_demo
  SETUP network and HTTP server
  PROCESS HTTP requests in loop
FUNCTION write_image_sd
  MOUNT SD, WRITE image, UNMOUNT SD
FUNCTION test_sd_card
  MOUNT SD, WRITE and READ test file, UNMOUNT SD
FUNCTION Error_Handler
  DISABLE interrupts, ENTER infinite loop
```

Flowchart:



Expected Outcomes: The Smart Home System will continuously collect data from sensors such as PZEM and DHT11. The data will be sent to an MQTT broker. The system will also send data to an HTTP server and perform SD card operations. The user will be able to monitor and control the interfaces using the system.

Future Work and Potential Application:

The smart home solution we provided through this proposed device focuses on security at the cost of scalability. However, this approach is not feasible as we will have to develop separate applications/ websites for each hub we design. To remedy this issue, we plan to implement some version of end-to-end encrypted data communication between the hub and the app through the Internet. This obviously will not be as secure as a separate app solution we implemented in this project but we will be able to implement this device in a much larger number of households. As the data transmission will be end-to-end encrypted a server will not be required and thus the possibility of data leakage will decrease significantly. In this way we can find a middle ground between security and scalability. Additionally, we also plan to increase the devices that are connected to the smart home hub which will make the hub more versatile and useful. For example, we can implement a circuit which will turn on the Air Conditioner/ Fans automatically when the room temperature reaches a certain threshold and turn that off when the temperature is below another threshold. Such functionality will provide the users convenience along with saving energy costs which is beneficial for the user but more importantly less power consumption will result in less amount of greenhouse gas emission. Thus the hub can help us to protect our environment and fight against global warming. Another future feature we are really looking forward to adding to the hub is a smart lock system for the whole house. This will enable users to activate and deactivate specific locks of the house from anywhere when needed. Furthermore, we want to add a voice command feature to the hub which will recognize specific clauses and act on the command.

The current version of the smart home hub can be implemented in every single household. Some use cases of the device are:

1. **Security System:** We connected a camera to the smart home hub which can be accessed from anywhere. This can be used to observe what is happening in the house when the user is somewhere else. So, the user can know instantly if any kind of burglary takes place in his/ her house and can notify the police on time.

2. **Child Monitoring:** Parents can know what their children are doing from the comfort of their rooms through the installed cameras.

3. **Automated Fan/ Light control:** Users can set schedules to turn on/ off lights and fans through the smart home hub.

4. **Environment Readings:** The hub has sensors to measure the temperature and air quality of the room. So the users can know how healthy of an environment their house is and act on improving the environment.

There are many other potential use cases for the smart home hub and we believe in this modern age every single household should have access to such a device to make their life a little more convenient and enjoyable

References:

Sinha, S. (2024, December 5). *State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally*. IoT Analytics. https://iot-analytics.com/number-connected-iot-devices/

STM32F446RE datasheet. (n.d.). Retrieved from https://www.st.com/resource/en/datasheet/stm32f446re.pdf

Grove - Air Quality Sensor v1.3 (n.d.). Retrieved from https://cdn-reichelt.de/documents/datenblatt/A300/101020078_01.pdf

DHT11 Technical Data Sheet. (2015). Retrieved from https://www.scribd.com/document/390814013/DHT11-Technical-Data-Sheet-Translated-Version-1143054

PZEM-004T Datasheet-User Manual. (2019). Retrieved from https://innovatorsguru.com/wp-content/uploads/2019/06/PZEM-004T-V3.0-Datasheet-User-Manual.pdf

uCAM-II Datasheet. (n.d). Retrieved from https://www.mouser.com/pdfdocs/uCAMII_datasheet_R_1_0.PDF?srsltid=AfmBOopxe-1JXYfz0WM6j0Q4rEmL3_lAwBomS3NfAO9a1YcfeO3diZjn

TS0011 Datasheet. (n.d). Retrieved from https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/5773/TS0011%20DATASHEET.pdf