My name: Đoàn Lê Gia Bảo (CSE2023)

ID: 10423008

1, Introduction to IoT

Investigating an IoT Project: Remote Patient Monitoring (RPM) in Healthcare

The Remote Patient Monitoring (RPM) system is a well-known IoT-based healthcare project that allows doctors to monitor patients outside of hospitals using connected devices. These devices collect real-time health data like heart rate, blood pressure, oxygen levels, and glucose levels and send it to healthcare providers for continuous monitoring.

Upsides of the Project

1. Improved Patient Care – RPM helps doctors detect health issues early and provide timely intervention, reducing emergency hospital visits.

2. Convenience for Patients – Patients, especially those with chronic illnesses, can stay at home while still being monitored by their healthcare provider.

3. Cost Reduction – Fewer hospital visits mean lower medical costs for both patients and hospitals.

4. Efficient Data Collection – IoT devices provide real-time data that helps doctors make informed decisions quickly.

5. Pandemic Benefits – During COVID-19, RPM allowed doctors to monitor patients remotely, reducing the risk of virus spread.

Downsides of the Project

1. Privacy and Security Risks – Patient data is transmitted online, making it vulnerable to hacking and data breaches.

2. Technology Dependency – RPM requires a stable internet connection and power, which can be unreliable in some areas.

3. High Initial Costs – The installation and maintenance of IoT devices can be expensive.

4. Training Challenges – Both patients and healthcare workers need technical knowledge to use RPM effectively.

Privacy Considerations

To address privacy concerns, RPM systems use end-to-end encryption, authentication protocols, and HIPAA-compliant cloud storage to ensure patient data remains secure.

Conclusion

Remote Patient Monitoring is a revolutionary IoT project in healthcare, offering numerous benefits but also posing challenges. With improved security measures and technological advancements, RPM is expected to transform the future of healthcare globally.

2, Deeper dive

Comparing Microcontrollers and Single-Board Computers

Comparison Table

Features	**Microcontroller**	**Single-Board Computer**
Processing Power	Low, designed for simple tasks	High, capable of running full OS

Operating system	No OS, runs a single program at a time	Runs Linux, Windows IoT, or similar
Power Consumption	Very Low power usage	Higher power consumption
Memory & Storage	Limited RAM and no onboard storage	More RAM and onboard storage (SD card or built-in)
Connectivity	Limited (only UART, I2C, SPI)	Has Wifi, Bluetooth, Ethernet, and USB ports
Task Complexity	Handles simple, real-time tasks	Can handle multitasking and complex applications
Price	Usually cheaper ($5 - $30)	More expensive ($30 - $100+)
Usage	Best for embedded systems, robotics, automation	Best for full applications, servers, and IoT gateways

Reasons for Using a Microcontroller Over an SBC

1. Real-Time Processing – Microcontrollers handle real-time tasks efficiently, such as reading sensors and controlling motors with minimal delay.

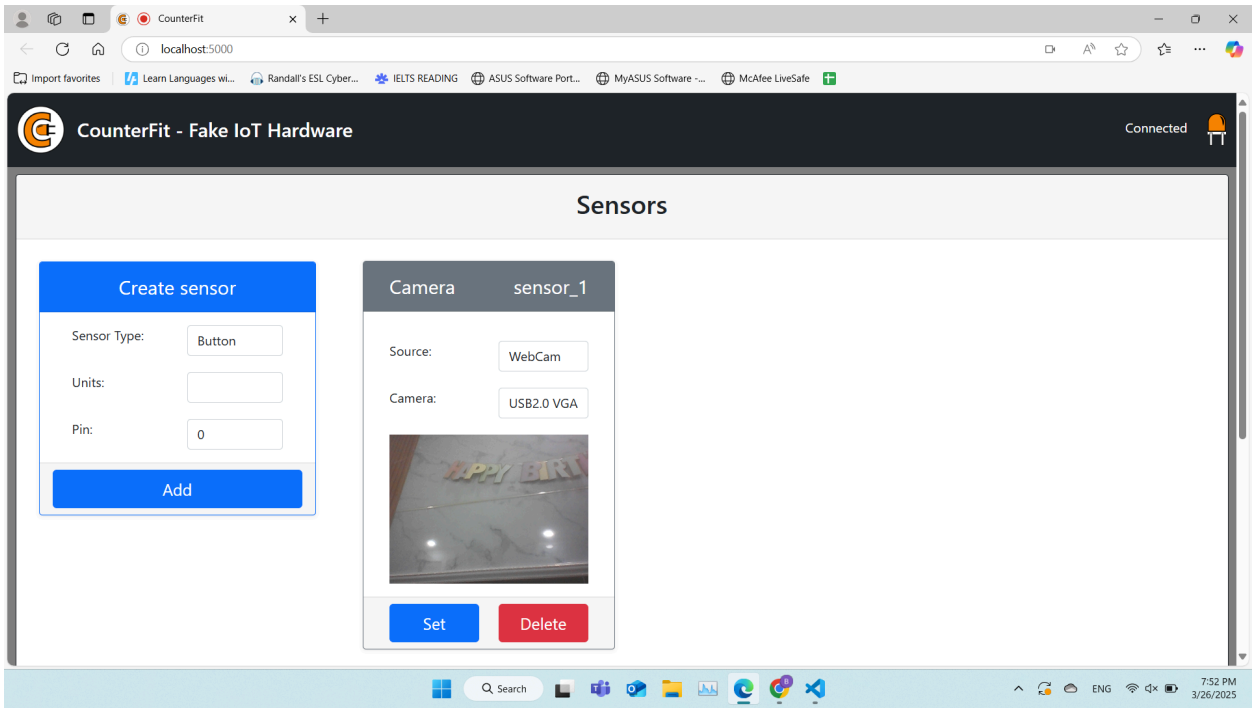
2. Lower Power Consumption – Ideal for battery-powered applications like wearable devices, smart sensors, and embedded systems.

Reasons for Using an SBC Over a Microcontroller

1. Full OS and Multitasking – An SBC can run Linux or Windows, supporting software like Python, databases, and web servers for more complex projects.

2. More Connectivity Options – SBCs have Wi-Fi, Bluetooth, and USB ports, making them ideal for IoT projects, networking applications, and media centers.

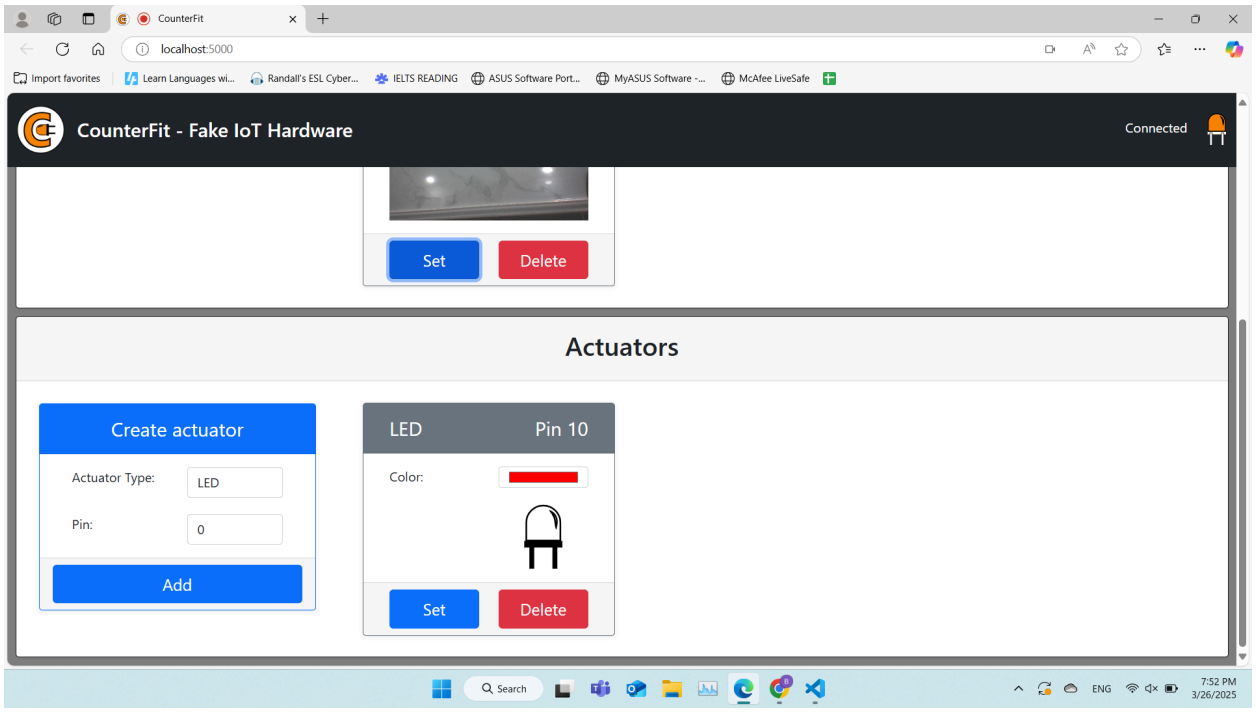
3, Sensors and Actuators (Using CounterFit - Fake IoT Hardware)



Sensor: Camera USB 2.0 VGA UVC Webcam (Webcam or Private File Upload)

1. **What it does**
    - A camera sensor captures visual information from the environment and converts it into a digital format. In the Counterfeit virtual environment, it can be used for object detection, facial recognition, or recording video input from a webcam or uploaded file.
2. **The electronics/hardware used inside**
    - The camera module consists of an image sensor that detects light intensity and color. It includes a lens to focus light, an image processor to refine the signal, and an interface to communicate with the system.
3. **Is it analog or digital?**
    - The camera is the digital sensor.
4. **Units and range of inputs/measurements**

- The main measurement unit is pixels.
- The camera captures light intensity and color in RGB values.
- Frame rate is measured in FPS.
- Light sensitivity is measured in ISO values, and the field of view is measured in degrees.



Actuator: LED

1. **What it does**
   - An LED emits light when electrical current flows through it. In the Counterfeit virtual environment, it can be used to simulate real-world feedback, such as indicating system status, providing visual notifications, or mimicking real hardware actions.
2. **The electronics/hardware used inside:** An LED consists of a semiconductor diode that emits light when current passes through it. The hardware includes:
   - A power source
   - A resistor to limit current and prevent damage
   - A microcontroller or driver circuit to control brightness and on/off states

3. **Is it analog or digital?**

   A LED can function as digital. A simple on/off LED is digital (binary states with 0: OFF and 1: ON).

4. **Units and range of inputs/measurements**
   - Voltage range: Red LEDs operate at 1.8V-2.2V.
   - Current: Typically 10-20mA.
   - Brightness: Measured in lumens.
   - Wavelength: Determines color, measured in nanometers.

4, Connect Internet

Comparison of AMQP and MQTT

1. **Power Usage:**
   MQTT is designed for low-power devices, making it highly efficient for IoT applications with constrained resources. Conversely, AMQP is more heavyweight and requires higher power consumption due to its complex protocol structure and additional features like message queuing.
2. **Security:**
   Both AMQP and MQTT support TLS encryption for secure communication. However, AMQP has a more robust built-in security model, including fine-grained access control and authentication mechanisms. MQTT relies primarily on username-password authentication and TLS but lacks built-in access control features.
3. **Message Persistence:**
   AMQP is designed with built-in message queuing, ensuring message durability even if the recipient is offline. MQTT supports Quality of Service (QoS) levels, allowing messages to be retained until delivery is confirmed, but it does not have a native queuing mechanism like AMQP.

Comparison of HTTP/HTTPS and MQTT

1. **Power Usage:**
   HTTP/HTTPS consumes significantly more power than MQTT because it operates over TCP with a request-response model. MQTT, with its lightweight publish-subscribe mechanism, is optimized for minimal power consumption, making it more suitable for IoT applications.

2. **Security:**
   HTTPS offers strong security with TLS encryption, ensuring end-to-end data protection. MQTT also supports TLS, but security implementations are often dependent on the application layer, requiring additional configurations for authentication and authorization.
3. **Message Persistence:**
   HTTP is stateless, meaning it does not inherently support message persistence. If a client disconnects, the request is lost unless manually handled with retries. MQTT, with its QoS levels and retained messages, ensures that messages can persist and be delivered once the client reconnects.

Overall, MQTT is better suited for IoT applications requiring low power consumption, efficient message delivery, and persistent connections, while AMQP and HTTPS provide more robust security and queuing mechanisms for enterprise and web applications.