


Embedded Systems Programming on STM32 MCU

การโปรแกรมระบบสมองกลฝังตัวบน

ไมโครคอนโทรลเลอร์ STM32


Internet of Things and Protocols

 การสื่อสารระหว่างอุปกรณ์ IoT กับคอมพิวเตอร์ส่วนกลาง ด้วย Application Layer protocols ตัวอย่างเช่น

 HTTP/HTTPS

 WebSocket & STOMP (Simple Text Oriented Messaging Protocol) over WebSocket

 REST (Representational State Transfer) (RESTful HTTP)

 CoAP (Constrained Application Protocol) & SMCP (Simple Media Control Protocol)

 XMPP (Extensible Message and Presence Protocol)

 MQTT (Message Queue Telemetry Transport)/SMQTT/MQTT-SN (MQTT for Sensor Networks)

 DDS (Data Distribution Service for Real-Time Systems)

 AMQP (Advanced Message Queuing Protocol)



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

OSI Layer 7 Protocols in IoT

HTTP/HTTPS

🌿 RFC9112 (HTTP/1.1) RFC9113 (HTTP/2) และอื่นๆ

🌊 ปัจจุบันใช้ HTTP/1.1 (1997), HTTP/2 (2015) , HTTP/3 (2022)

🌿 วางตัวอยู่บน TCP

🌿 ทำงานในลักษณะของ Request-Response (client-server model)

🌊 เว็บเซิร์ฟเวอร์เปิดช่องทางการเชื่อมต่อรอไว้ (TCP port 80 เป็นต้น)

🌊 การเชื่อมต่อกระทำเป็นช่วงสั้นๆ สำหรับการร้องขอข้อมูลและการส่งกลับข้อมูล
หนึ่งครั้ง

🌊 เว็บเบราว์เซอร์(หรืออื่นใด)ทำหน้าที่เป็น client ร้องขอข้อมูลจาก web server

🌊 Client ส่ง HTTP request ไปยังเซิร์ฟเวอร์

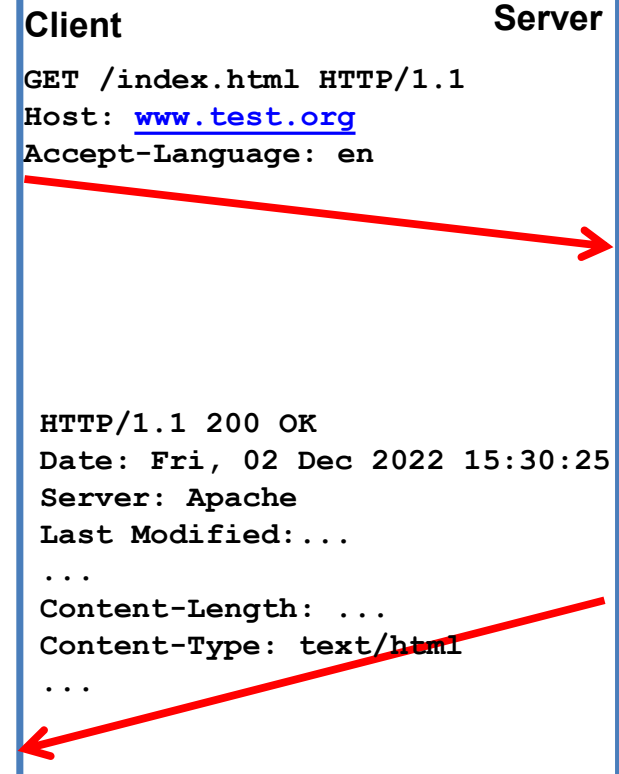
🌊 เว็บเซิร์ฟเวอร์รับการเชื่อมต่อ (established) ตามโปรโตคอล TCP และรับข้อมูลที่ส่งมาจาก
ทางฝั่งไคลเอนต์

🌊 ประมวลผลข้อมูลและส่งกลับ ขึ้นอยู่กับ Content-Type

🌊 อาจเป็นข้อความ

🌊 อาจเป็นข้อมูลไบนารี

🌊 ปิดการเชื่อมต่อ (close connection) แล้วรอไคลเอนต์อื่นร้องขอต่อไป



Asst.Prof. Thanwa SRIPRAMONG

PRESENTER

TODAY TOPIC IS

OSI Layer 7 Protocols in IoT

WebSocket

- 🌿 RFC6455
- 🌿 วางตัวอยู่บน TCP และอาศัย HTTP เพื่อเริ่มการเชื่อมต่อ
- 🌿 เมื่อเชื่อมต่อสำเร็จ จะคง established ไปจนกว่าฝ่ายใดฝ่ายหนึ่งปิด
 - 💡 แพ็กเก็ตข้อมูลถูกส่งระหว่างกันได้อย่างอิสระ
- 🌿 ขั้นตอนการทำงาน
 - 💡 ไคลเอนต์ร้องขอเปิด WebSocket อาศัย HTTP
 - 💡 เว็บเบราว์เซอร์ตอบรับการอัปเดตเป็น WebSocket Protocol
 - 🌐 เซอร์เวอร์เปิดพอร์ต TCP หมายเลขตามที่กำหนดไว้ล่วงหน้า
 - 🌐 ไคลเอนต์เชื่อมต่อ TCP
 - 💡 จากนั้นรับส่งข้อมูลกันโดยใช้โครงสร้างข้อมูลตามที่ระบุไว้ในมาตรฐานของ WebSocket

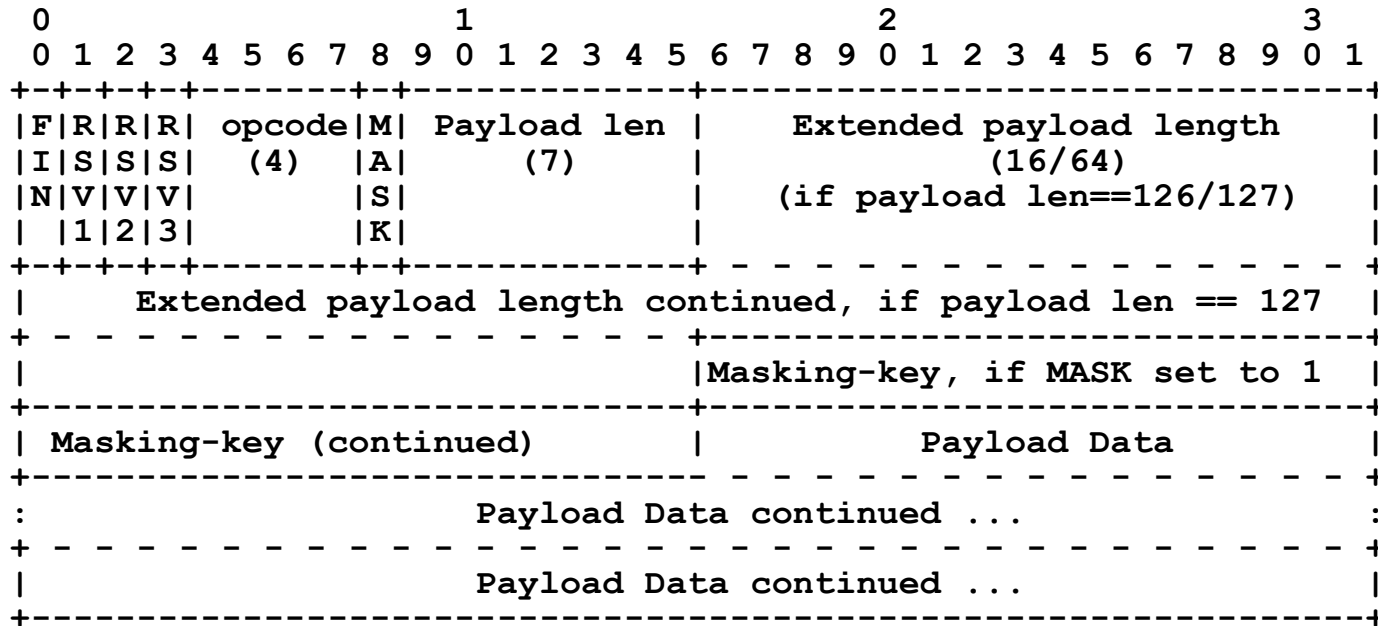


Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

OSI Layer 7 Protocols in IoT

WebSocket



Client

Server

GET /chat HTTP/1.1

Host: www.test.org

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: ข้อมูล 16 ไบต์

เข้ารหัสแบบ base64

Sec-WebSocket-Version: ...

HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: ข้อมูล MD5 Hash
ขนาด 20 ไบต์ เข้ารหัสแบบ base64

รับส่งข้อมูลระหว่างกันตาม WebSocket

Protocol (บน TCP พอร์ตตามที่สองฝ่าย
กำหนดไว้ล่วงหน้า)



Asst.Prof. Thanwa SRIPRAMONG

PRESENTER

TODAY TOPIC IS

OSI Layer 7 Protocols in IoT

WebSocket: JavaScript Example

```
// Creates new WebSocket object with a wss URI as the parameter
const socket = new WebSocket('wss://game.example.com/ws/updates');

// Fired when a connection with a WebSocket is opened
socket.onopen = function () {
    setInterval(function() {
        if (socket.bufferedAmount == 0)
            socket.send(getUpdateData());
    }, 50);
};

// Fired when data is received through a WebSocket
socket.onmessage = function(event) {
    handleUpdateData(event.data);
};

// Fired when a connection with a WebSocket is closed
socket.onclose = function(event) {
    onSocketClose(event);
};

// Fired when a connection with a WebSocket has been closed because
// of an error
socket.onerror = function(event) {
    onSocketError(event);
};
```

Client	Server
GET /chat HTTP/1.1 Host: www.test.org Upgrade: websocket Connection: Upgrade Sec-WebSocKet-Key: ข้อมูล 16 ไบต์ เข้ารหัสแบบ base64 Sec-WebSocKet-Version: ...	
	HTTP/1.1 101 Switching Protocols Upgrade: websocket Connection: Upgrade Sec-WebSocket-Key: ข้อมูล MD5 Hash ขนาด 20 ไบต์ เข้ารหัสแบบ base64
	รับส่งข้อมูลระหว่างกันตาม WebSocket Protocol (บน TCP พอร์ตตามที่สองฝ่าย กำหนดไว้ล่วงหน้า)



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

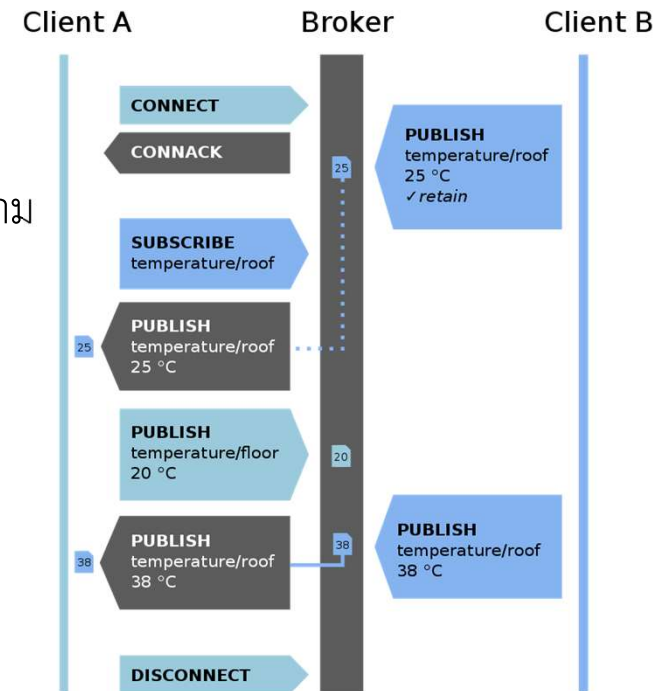
TODAY TOPIC IS

OSI Layer 7 Protocols in IoT

MQTT



- ISO/IEC 20922
- วางตัวอยู่บน Layer ที่ต้องการความเสถียรของข้อมูลเช่น TCP
- แต่เดิมชื่อ MQ Telemetry Transport (มาจากหุ่นคอมพิวเตอร์ IBM MQ)
- อาจใช้ TLS encryption ซึ่งมี certificate, username, password เพื่อยกระดับความปลอดภัย
- ทำงานในลักษณะของ publish-and-subscribe messaging
 - เครื่องตัวกลาง message broker ทำหน้าที่รับ/ส่งทอดข้อความระหว่างเครื่องไคลเอนต์ภายในระบบ
 - จัดเก็บข้อความเป็นกลุ่มที่เรียกว่า topic
 - ที่อาจจะมี subtopic ได้ เช่น test/a1 test/a2 help เป็นต้น
 - ไคลเอนต์ทำหน้าที่เป็นผู้ส่งข่าวสาร และ/หรือ ผู้รับข่าวสาร
 - ไคลเอนต์สามารถขอเชื่อมต่อหรือปลดการเชื่อมต่อได้ตลอดเวลา
 - ผู้ส่งข่าวสาร publish ข่าวสารตาม topic ที่กำหนดไปยัง broker
 - ผู้รับข่าวสาร subscribe ข่าวสารตาม topic ที่ต้องการ เมื่อ broker ได้รับข่าวสารใน topic ที่ตรงกับที่ subscribe ไว้จะส่งต่อไปยังผู้รับข่าวสาร



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

OSI Layer 7 Protocols in IoT