

Hardware Synthesis Lab

Final Project : Internet of Things (IoT)

นำเสนอ

อ. ดร. พิษณุ สิทธิอมร

ตอนเรียนที่ 3 กลุ่มที่ 1

สุธิดา สุคนธ์นันท์กุล	5931073121
-----------------------	------------

ณัฐพนธ์ ธรรมอาชีพ	5931022121
-------------------	------------

ปณณวิชญ์ ปงคำ	5930319421
---------------	------------

หัสกฤษฎี เกิดบัณฑิต	5931074821
---------------------	------------

Embedded System Laboratory 2110366 ภาคเรียนที่ 1 ปีการศึกษา 2561

คณะวิศวกรรมศาสตร์ ภาควิชาคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

วัตถุประสงค์ของโครงการ

ในการทำโครงการ Final project ในรายวิชา Embedded System Laboratory 2110366 ครั้งนี้ มีจุดประสงค์เพื่อให้ได้นำความรู้ที่ได้รับจากผู้สอนในห้องเรียน ความคิดสร้างสรรค์ และความสามารถในการจัดการการวางแผน เพื่อที่จะทำการสร้าง IoT device (Internet of Things device) ซึ่งในการทำโครงการในครั้งนี้ กลุ่มของเรได้นำอุปกรณ์ STM32F407 มาใช้ในการอ่านและเขียนค่าจากตัวตรวจวัด (Sensors) โดยกลุ่มของเรจะนำ ESP8266 (NodeMCU) มาใช้ในการติดต่อสื่อสารระหว่างอุปกรณ์ STM32 เข้ากับ Internet โดยผ่านสัญญาณ WiFi

นอกจากอุปกรณ์ STM32 ของเรจะสามารถเชื่อมต่ออินเทอร์เน็ตจาก ESP8266 แล้ว เพื่อที่จะอุปกรณ์ของเรานั้นเป็นอุปกรณ์แบบ Internet of things device (IoT) อย่างสมบูรณ์ เรานำเครื่องมือ NETPIE มาช่วยเพื่อความสะดวกในการจัดการอุปกรณ์ต่างๆ ที่เชื่อมต่อกันผ่าน Internet

กลุ่มของเรเลือกที่จะทำ อุปกรณ์ตรวจจับขโมย สำหรับโครงการในครั้งนี้ โดยสร้างสรรค์ขึ้นภายใต้แนวคิด Smart Home or Aging Society โดนกลุ่มของเรเลือกที่จะนำ Ultrasonic sensor มาใช้ในการสร้างสรรค์อุปกรณ์ดังกล่าว

นอกจากที่จะได้ใช้ความรู้ที่ได้เรียนมา ความคิดสร้างสรรค์ และความสามารถในการจัดการการวางแผนแล้ว กลุ่มของเรายังได้ใช้ความสามารถในการทำงานเป็นทีม (team works skill) จากการทำงานเป็นกลุ่มด้วยกันสี่คน โดยแต่ละคนจะได้รับหน้าที่มอบหมายต่างๆ ดังนี้

1. System architecture
2. Front-end development
3. Embedded System development
4. UI designer and development

ความหมายเพิ่มเติม

Internet of things : IoT เป็นระบบที่มีความเกี่ยวข้องกันของอุปกรณ์คำนวณ (Computing device) ซึ่งต้องมีความสามารถในการสื่อสารกันผ่านเครือข่ายโดยติดต่อใช้มนุษย์หรือไม่ใช้มนุษย์ก็ได้ ซึ่งในปัจจุบันได้นำไปประยุกต์ใช้ในสาขาต่างๆ เช่นใช้ในทางการแพทย์ ใช้ในการเกษตร ภายในบ้าน หรือแม้กระทั่งถูกนำไปประยุกต์ใช้ภายในเมืองใหญ่ๆ

NETPIE ถูกพัฒนาขึ้นโดย NECTEC เมื่อปี 2015 เพื่อใช้เป็น cloud-based IoT platform-as-a-service โดยจะเชื่อมต่อ NETPIE กับ NodeMCU

System Architecture

ในการออกแบบและวางแผนการสร้างอุปกรณ์ตรวจจับข้อมูลสำหรับโครงการนี้ กลุ่มของเรามีแนวคิดเบื้องต้นที่จะหาแนวทางการแจ้งเตือนเจ้าของบ้าน และคนที่อยู่ในบ้านหากมีการบุกรุกเข้ามาของบุคคลภายนอก

หน้าที่ของ system architecture คือการออกแบบระบบเพื่อตอบสนองการแก้ปัญหาข้างต้นคือ เราจะตรวจจับได้อย่างไรว่ามีการบุกรุกของบุคคลภายนอก และเราจะแก้ปัญหานั้นได้อย่างไร

ในระยะเริ่มต้นของการทำโครงการนี้ ทางกลุ่มได้จัดเลือกและมอบหมายงานให้กระผมเป็นผู้นำในด้าน system architecture แต่ความคิดของคนคนเดียวอาจจะไม่ดีพอ เลยมีการระดมความคิด (Brain storm) เกี่ยวกับ to-be system ของระบบว่าระบบควรมีระบบออกมาเป็นอย่างไร มีความน่าใช้งานมากแค่ไหน สมเหตุสมผลกับการนำไปใช้หรือไม่ และได้สรุปผลลัพธ์เป็นแนวทางอย่างคร่าวคร่าว และนำแนวคิดดังกล่าวไปพิจารณาปรับปรุงภายหลังและนำมาเสนอสมาชิกทุกคนในกลุ่ม หากคนในกลุ่มยอมรับและเห็นด้วยกับระบบแล้ว ต้องทำให้แน่ใจว่าสมาชิกทุกคนในกลุ่มมีความเข้าใจในระบบตรงกัน เพื่อป้องกันปัญหาที่อาจเกิดขึ้นในการพัฒนาโครงการในขั้นต่อไป

หลังจากที่ได้ระดมความคิดของสมาชิกบุคคลในกลุ่มแล้ว ผู้ออกแบบระบบของเราได้มีแนวคิดที่จะจับการบุกรุกของบุคคลภายนอกโดยมีแนวคิดมาจากการที่เราทุกคนต้องเปิดประตูก่อนเข้าบ้าน ในทำนองเดียวกัน หากมีผู้บุกรุกเข้ามา เราจึงสันนิษฐานเบื้องต้นว่าผู้บุกรุกนั้นก็ต้อง

เข้ามาทางประตู เราจึงพิจารณาหาวิธีการตรวจจับว่า ในการที่จะเข้ามาทางประตูนั้น ต้องเกิดอะไรขึ้นบ้าง

โดยขั้นต้น กลุ่มของเราได้พิจารณาชนิดของประตูที่ได้รับความนิยมในปัจจุบัน โดยเลือกมาสองแบบ ซึ่งจากการค้นคว้าเราจะได้ประตูชนิดต่างกันสองชนิดเป็นหลัก คือ

1. ประตูบานเปิด จะเป็นประตูที่สามารถเปิดเข้าหรือเปิดออกทางฝั่งใดฝั่งหนึ่ง ซึ่งประตูบานเปิดนั้น จะสามารถเปิดเป็นบานเดี่ยวหรือบานคู่ก็ได้ สำหรับประตูบานคู่มิลักษณะการเปิดกว้างได้มากถึง 180 องศา และสำหรับประตูบานเดี่ยวจะเปิดได้ 90 องศา โดยทั่วไปจะเห็นประตูบานเปิดเดี่ยวมากกว่าเพื่อประหยัดพื้นที่
2. ประตูบานเลื่อน จะเป็นประตูที่ใช้การเลื่อนเป็นการเปิดประตู โดยจะเลื่อนไปด้านข้าง ประตูลักษณะนี้จะเหมาะเป็นประตูระเบียงหรือกันห้อง

เมื่อเริ่มพิจารณาลักษณะร่วมกันของประตู กลุ่มของเราเลือกที่จะนำ Ultrasonic มาใช้เพื่อที่จะวัดระยะห่างระหว่างประตูและผนังที่ติดประตู ในกรณีที่ด้านที่เปิดประตูอยู่ห่างจากผนังมาก เราจะได้เลือกสร้างผนังจำลอง เพื่อให้ค่าที่ได้จากการวัดไม่คลาดเคลื่อนและมีความแม่นยำสำหรับการประยุกต์ใช้ในประตูในรูปแบบต่าง ๆ ซึ่งการทำงานของอุปกรณ์ของเราจะทำการส่งเสียงร้องหากมีการเปิดประตูเกิดขึ้น (ค่าที่ได้จาก Ultrasonic มีค่าเพิ่มมากขึ้นเกิดค่าที่กำหนด) และระบบเปิดอยู่ ตัว STM32 จะส่งเสียงร้องออกมาเพื่อทำการแจ้งเตือนบุคคลในบ้านและรอบๆ บ้านว่าเกิดการเปิดประตูที่ไม่ได้ปิดระบบก่อนขึ้น ซึ่งแนวคิดนี้เปิดแนวคิดที่นำไปประยุกต์ใช้และพัฒนาต่อได้อย่างดี เพราะเป็นแนวคิดที่ไม่ซับซ้อนมาก และสามารถนำไปใช้ได้จริง

Front-end development

Front-end development จะทำหน้าที่เปลี่ยนข้อมูลที่ได้รับให้เป็น Graphic interface เพื่อให้ผู้ใช้งานได้เห็นและโต้ตอบกับข้อมูล โดยผ่าน HTML CSS หรือ React และ JavaScript ซึ่งจะได้ผลลัพธ์เป็น Web application เพื่อให้ผู้ใช้ได้ใช้งานโดยตรง

ในการทำ Front-end web development ของกลุ่มเราจะใช้ React เป็นหลัก โดยจะทำการปฏิบัติการบน Web Platform โดยที่ Front-end development จะมุ่งเน้นไปยังการพัฒนาเว็บไซต์และติดต่อรับส่งข้อมูลกับ embedded system เป็นหลัก นั่นคือผ่าน library microgear

สำหรับการส่งข้อมูล จะมีการส่งข้อมูล ON, OFF, CLOSE สำหรับการ activate system, deactivate system และ ปิดเสียงลำโพงที่ต่อกับ stm32 ตามลำดับ

สำหรับการรับข้อมูล จะมีการรับข้อมูลอยู่ 3 รูปแบบ คือ ON, OFF, OPEN นั่นคือ activate system, deactivate system และ ประตูถูกเปิด ตามลำดับ

การรับส่งข้อมูลกับ embeded system จะมีผลต่อการ set state ของหน้าเว็บไซต์ ให้มีการแสดงผลต่อผู้ใช้ได้อย่างถูกต้อง

ซึ่งบทบาทของ front end จะไม่เหมือนกัน UI development and design แต่ทั้งสองคนต้องทำงานร่วมกัน เพื่อให้ระบบสอดคล้องกันทั้งหมด และต้องมี system architecture ในการพิจารณาตลอดการพัฒนาเพื่อให้ระบบนั้น สอดคล้องไปตามที่วางแผนไว้ และมีความเข้าใจตรงกันทุกคนสำหรับสมาชิกในกลุ่มทุกคน



ตัวอย่างหน้าเว็บไซต์

```
microgear.on("message", (topic, data) => {  
  if (data == "ON") {  
    this.setState({ state: 1 });  
  } else if (data == "OFF") {  
    this.setState({ state: 3 });  
  } else if (data == "OPEN") {  
    this.setState({ state: 2 });  
  }  
});
```

```
switchPress(logic) {  
  if (logic == 1) {  
    microgear.chat(thing1, "ON");  
  } else if (logic == 0) {  
    microgear.chat(thing1, "OFF");  
  }  
  if (logic == 2) {  
    microgear.chat(thing1, "CLOSE");  
  }  
}
```

ตัวอย่างการติดต่อกับ embeded system ผ่าน library microgear

Embedded System development

สำหรับ Embedded system development เราจะ ออกแบบ Embedded system เพื่อตอบสนองปัญหาการตรวจจับคนเข้าออกที่ไม่พึงประสงค์ โดยอุปกรณ์ที่เราใช้ในการพัฒนาโครงการนี้ขึ้นคือ STM32F407 จำนวน 1 เครื่อง ESP8266 (NodeMCU) จำนวน 1 เครื่อง US-015 Ultrasonic module จำนวน 1 เครื่อง และเครื่องส่งสัญญาณเสียง 1 เครื่อง

ในการติดต่อกันระหว่าง STM32F407 และ US-015 Ultrasonic module จะติดต่อกันโดยผ่าน GPIO port ซึ่งในที่นี้ กลุ่มของเราใช้ GPIOE11 และ GPIOE12 ในการติดต่อกันระหว่างสองอุปกรณ์ และต่อ US-015 เข้ากับ Ground และตัวจ่ายไฟด้วย

ในการติดต่อกันระหว่าง STM32F407 และ ESP8266 (NodeMCU) จะติดต่อกันโดยผ่าน GPIO port ซึ่งในที่นี้ กลุ่มของเราใช้ GPIOA3 และ GPIOA2 ในการติดต่อกันระหว่างสองอุปกรณ์ และต่อ ESP8266 เข้ากับ Ground และตัวจ่ายไฟต่อเข้ากับเครื่องคอมพิวเตอร์โดยตรงสำหรับการติดต่อกันระหว่าง STM32F407 และ NodeMCU จะติดต่อกันผ่าน UART ตาม port ที่กล่าวมาข้างต้น

สำหรับการติดต่อกันระหว่าง ESP8266 (NodeMCU) และ NETPIE เราจะทำการ `#include <MicroGear.h>` เพื่อที่จะใช้ library ต่างๆในการคุยกับ NETPIE โดยใช้ APPID KEY และ SECRET ในการยืนยันการใช้งานกับ NETPIE ซึ่ง Application ID , Application Key (Device Key), Application Secret นำมาจาก NETPIE หลังจากสร้าง device key เสร็จโดยที่ ESP8266 (NodeMCU) จะส่งข้อมูลเป็น String คือ “ON” และ “OFF” เพื่อที่จะแสดงผลผ่าน web browser ว่าตัวอุปกรณ์ตรวจจับขโมยของเรานั้นมีสถานะเป็นอย่างไร ในกรณีที่ ESP8266 (NodeMCU) ส่ง “ON” มา Web จะแสดงผลว่า อุปกรณ์ของเรานั้นกำลังเปิดทำงานอยู่ในกรณีที่ มีการเปิดประตูระหว่างที่อุปกรณ์ทำงาน ESP8266 (NodeMCU) จะส่ง “OPEN” มา จะแสดงผลว่าอุปกรณ์ของเรานั้นกำลังเปิดทำงานอยู่และประตูกำลังเปิดอยู่ ในกรณีที่ ESP8266 (NodeMCU) ส่ง “OFF” มา Web จะแสดงผลว่าอุปกรณ์ของเรานั้นไม่ได้กำลังทำงานอยู่ และเมื่อเปิดประตูในขณะที่อุปกรณ์ไม่ได้ทำงานอยู่ ก็จะไม่มีการแจ้งเตือนหรือแสดงผลที่ต่างกันออกไป ซึ่งตัวอย่างคำสั่งที่ใช้ติดต่อกันเช่น `microgear.on` `microgear.chat` เป็นต้น

สำหรับ US-015 Ultrasonic module จะมีระยะตรวจจับ 2 เซนติเมตรถึง 4 เมตร ในการประยุกต์ใช้ US-015 Ultrasonic module จะพิจารณาค่าที่อ่านได้จากเซ็นเซอร์อัลตราโซนิก

แล้วส่งค่าจาก STM32F407 ไปยัง ESP8266 (NodeMCU) แล้วนำค่าดังกล่าวมาพิจารณาใน NodeMCU ว่าเข้าเงื่อนไขใด (อุปกรณ์ปิดอยู่, อุปกรณ์เปิดอยู่, อุปกรณ์เปิดอยู่และประตูเปิดขึ้น) แล้วจึงค่อยส่งค่าที่พิจารณาแล้วมาเพื่อแสดงผล ออกทาง web browser โดยค่าเริ่มต้นที่อ่านได้จาก US-015 Ultrasonic module เราจะกำหนดจากระยะห่างระหว่าง US-015 Ultrasonic module และผนัง (รวมกรณีที่ใช้ผนังจำลอง) ระบบจะแจ้งว่าเป็นกรณีที่อุปกรณ์กำลังทำงานและประตูเปิดเมื่อค่าที่อ่านได้จาก US-015 Ultrasonic module มีค่ามากกว่า 10 เซนติเมตรโดยประมาณ

```
/* If a new message arrives, do this */
void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen) {
    Serial.print("Incoming message --> ");
    msg[msglen] = '\0';
    Serial.println((char *)msg);
    if(msg[0]=='O' && msg[1]=='N'){
        Serial.println("Turn on sensor");
        mySerial.write("1");
        microgear.chat("my_server", "ON");
    }
    if(msg[0]=='O' && msg[1]=='F' && msg[2]=='F'){
        Serial.println("Turn off sensor");
        mySerial.write("0");
        microgear.chat("my_server", "OFF");
    }
    if(msg[0]=='C' && msg[1]=='L' && msg[2]=='O' && msg[3]=='S' && msg[4]=='E'){
        Serial.println("Close...");
        mySerial.write("2");
        microgear.chat("my_server", "CLOSE");
    }
    //Serial.println((char *)msg);
}
```

เมื่อรับข้อความจาก NETPIE, ESP8266 จะทำการตอบกลับและส่งข้อมูลต่อให้ STM32F407


```

while (1)
{
    /* USER CODE END WHILE */
    MX_USB_HOST_Process();

    /* USER CODE BEGIN 3 */
    time=0;
    HAL_GPIO_WritePin(GPIOE,GPIO_PIN_12,0);
    mDelay(2);
    HAL_GPIO_WritePin(GPIOE,GPIO_PIN_12,1);
    mDelay(6);
    HAL_GPIO_WritePin(GPIOE,GPIO_PIN_12,0);
    while(!HAL_GPIO_ReadPin(GPIOE,GPIO_PIN_11));
    while(HAL_GPIO_ReadPin(GPIOE,GPIO_PIN_11)){
        time++;
    }
    sprintf(buf,"%d\n\r",time);
    if(time > 5000 && b==1){
        sprintf(signal,"%s" "1");
        HAL_UART_Transmit(&huart2,signal,sizeof(signal),1000);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_11,1);
    }

    HAL_Delay(1);
}

```

STM32F407 รับค่าจาก US-015 และส่งสัญญาณให้ ESP8266 เมื่อเปิดประตู

```

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart){
    if(huart->Instance == USART2){
        HAL_UART_Receive_IT(&huart2,com,1);
        if(strcmp(com,"1")==0){
            b=1;
        }
        if(strcmp(com,"0")==0){
            b=0;
        }
        if(strcmp(com,"2")==0){
            HAL_GPIO_WritePin(GPIOB,GPIO_PIN_11,0);
            b=1;
        }
    }
}
/* USER CODE END 0 */

```

STM32F407 เปิดปิดสัญญาณแจ้งเตือนจาก US-015
เมื่อได้รับคำสั่งจาก ESP8266 ผ่าน UART Protocol

UI designer and development

UI designer and development จะทำหน้าที่ออกแบบ User interface สำหรับการติดต่อกันระหว่าง user และระบบ โดยที่ UI designer and development จะทำงานควบคู่กับ Front-end development เพื่อพัฒนาความรู้สึกโดยรวมของการใช้งานสำหรับผู้ใช้งาน (Overall user experience) โดยมีจุดประสงค์หลักคือ ทำงานเกี่ยวกับ Graphic design element, Research on the design and its effectiveness, Analysis of customers, Design prototyping, Design implementation with developers

สำหรับหน้าเว็บไซต์นี้จะมีการออกแบบให้มีสีหลักๆ อยู่ 2 สี นั่นคือ สีเขียว และสีแดง เป็นสัญลักษณ์บ่งบอกการ activate และ deactivate ตามลำดับ และมีการใช้รูปภาพเป็นสัญลักษณ์แสดงผลให้ผู้ใช้งานเข้าใจง่าย นั่นคือ สัญลักษณ์รูปโล่บ่งบอกความ secure นั่นคือความปลอดภัย สัญลักษณ์ warning บ่งบอกถึงการเตือนภัยกรณีประตูถูกเปิด และสัญลักษณ์ deactivate บ่งบอกว่าขณะนี้ระบบไม่ได้กำลังทำงานอยู่

UI designer and development จะมีหน้าที่ซ้ำซ้อนกับ Front-end แต่ไม่เหมือนกันทีเดียว ต้องช่วยกันทำโดยที่ UI designer and development จะมีหน้าที่เกี่ยวกับความสวยงามและความน่าใช้เป็นหลัก ต้องเน้นไปที่ความสะดวกในการใช้งาน ความไหลลื่นของ application ทดสอบการทำงานของระบบ และ application



สีเขียวและแดง represent การ activate และ deactivate system



Symbol represent secure, warning, deactivate ตามลำดับ