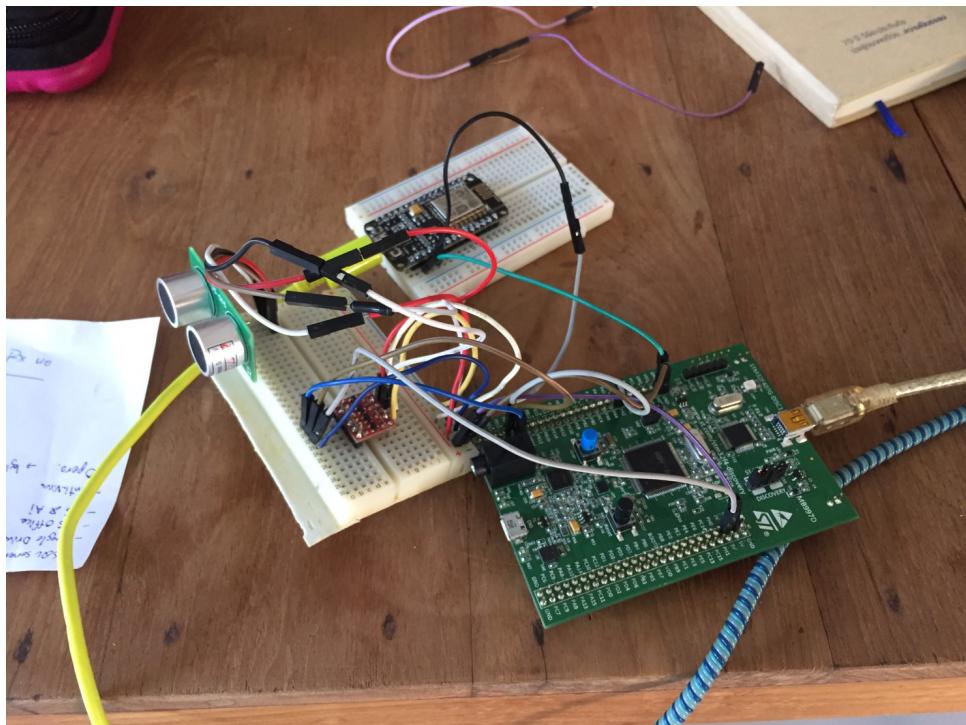


Trespassing Detector



สมาชิกกลุ่ม

ศตสุข	วิภาวดีลปน	5931062221
จักรกฤษณ์	ทรงอำนาจเจริญ	5931008421
พันณ์นิชา	จิตรวิโรจน์	5931041021
ชัยพล	ແຍ້ມບານ	5931015821

เสนอ

อ. ดร. พิชญา สิทธิอมร

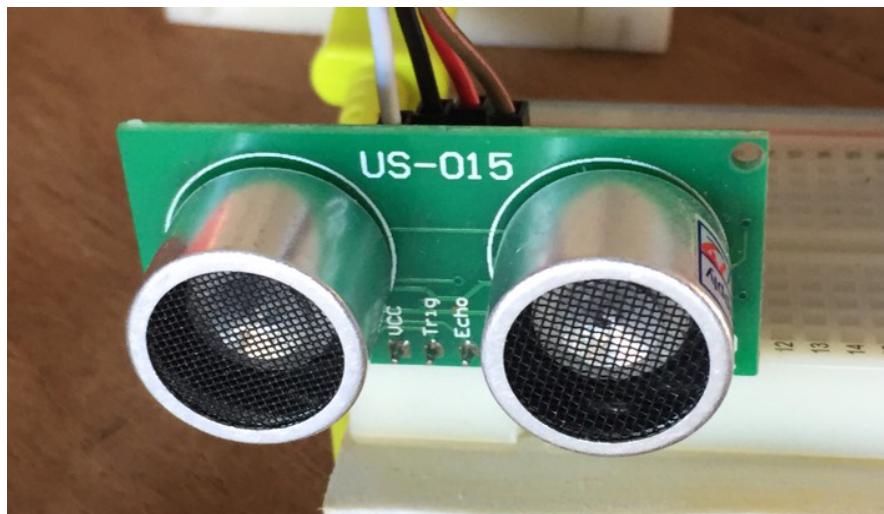
รายงานนี้เป็นส่วนหนึ่งของวิชา 2110366 Embedded System Laboratory

ภาคการศึกษา ต้น ปีการศึกษา 2561

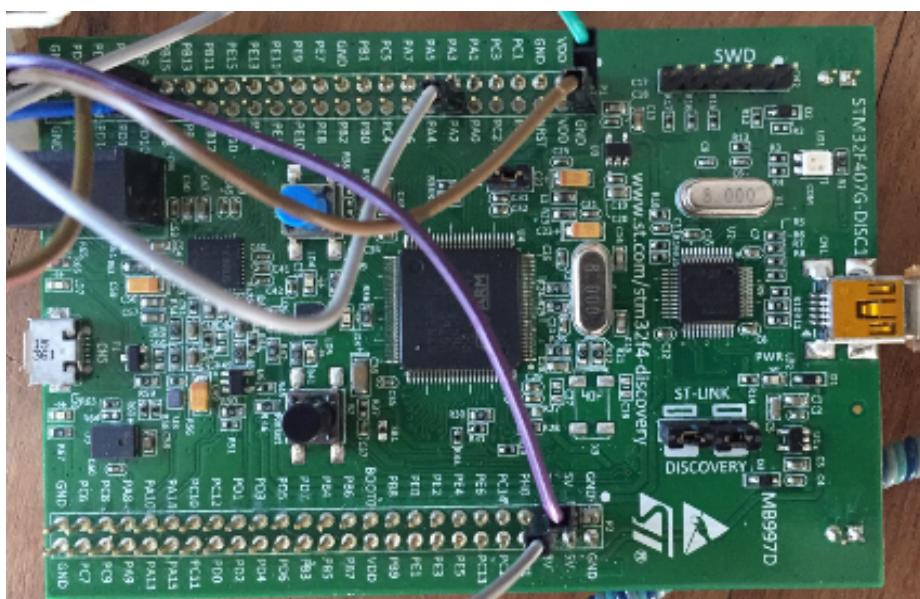
Description

Trespassing Detector เป็นอุปกรณ์ตรวจจับบุคคลที่ผ่านเข้ามาในที่ที่เราติดตั้งอุปกรณ์นี้ไว้ โดยมีการแจ้งเวลาที่คนผ่านล่าสุดเป็นเวลาเท่าไหร่ และผู้ใช้สามารถกำหนดเวลาที่ต้องการดูข้อมูลได้โดยจะแสดงตารางนับจำนวนคนที่ผ่าน และแสดงตารางเวลาคนที่มีเข้ามา

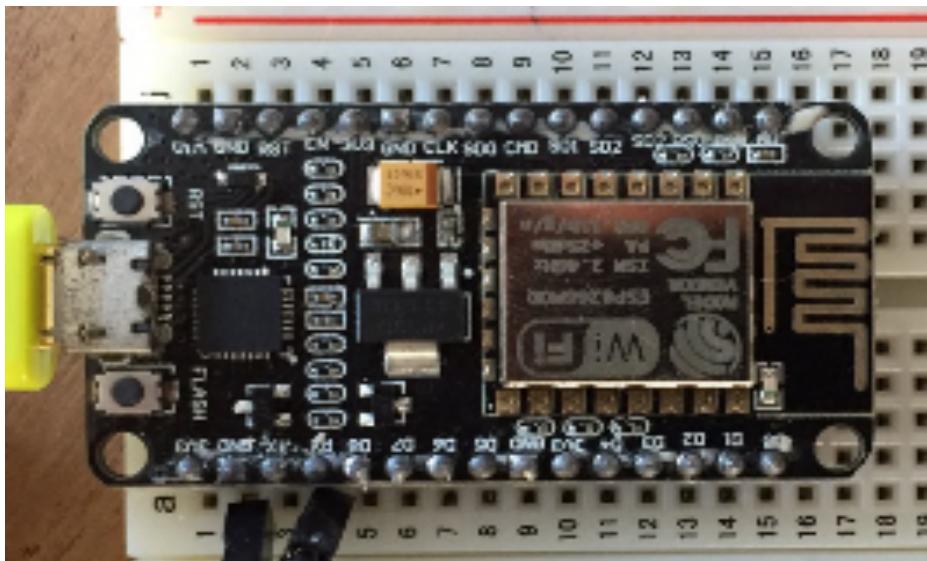
Requirements



An ultrasonic sensor



STM32F407 Board

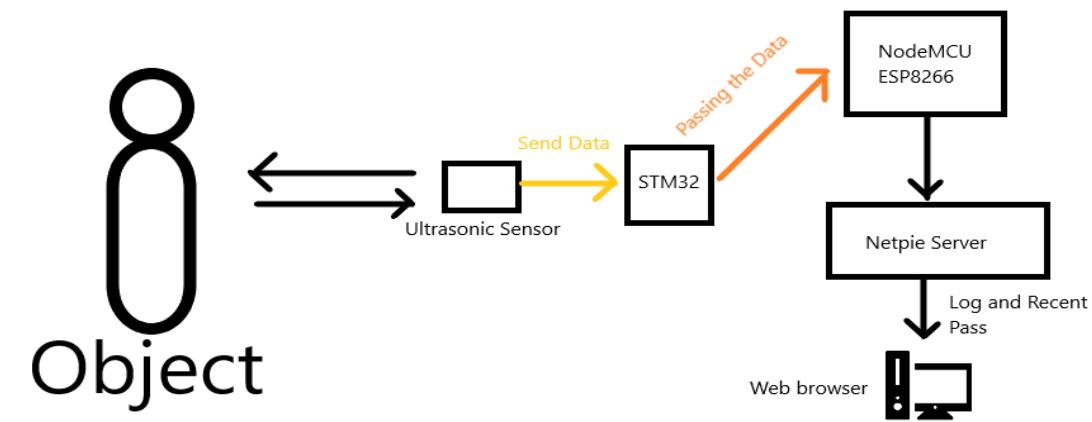


NodeMCU (esp8266)

How it works

เมื่อวัตถุผ่าน Sensor ตรวจจับ Sensor จะข้อมูลส่งไปบอก STM32 ในรูปของสัญญาณ High และ STM32 จะคำนวนระยะเวลาที่วัตถุใช้ผ่าน รวมถึงประมาณผลข้อมูล และส่งไปยัง NodeMCU เพื่อเชื่อมต่ออินเตอร์เน็ต และส่งข้อมูลไปที่เซิร์ฟเวอร์ของ NetPie ด้วย หลังจากที่ข้อมูลไปอยู่บน NetPie และเราจะใช้หน้าเว็บเพื่อดึงข้อมูลจากNetPie มาแสดง

ในส่วนของการติดต่อระหว่าง NodeMCU และ STM32 เราจะใช้ uart เป็น protocol ในการสื่อสาร ส่วน Ultrasonic Sensor จะต่อสายไฟ 5V GND Trigger(สำหรับบอกว่าจะเริ่มวัด โดยส่งสัญญาณ high ไปให้ Trigger) Echo(บอกเวลาที่ใช้ไปกลับนานเท่าไหร) เนื่องจาก US-015 ใช้ไฟ 5V เราจึงต้องใช้ ตัวแปลงไฟ เพื่อไม่ให้ GPIO_PIN เสีย เนื่องจากไฟเกิน



How to use

1. ให้วาง sensor ไว้ตรงทางผ่านของประตูที่มีระยะห่าง 0.3-2 เมตร
2. กดปุ่มสีดำที่ STM32 เพื่อตั้งค่าระบบใหม่
3. เปิด browser เพื่อเชื่อมต่อเข้ากับระบบ ที่หน้าเว็บ จะมี Recent pass บอกสถานะล่าสุดที่มีคนผ่าน sensor จะมีการอัพเดตเองอัตโนมัติเมื่อมีสิ่งไดผ่าน
4. ผู้ใช้สามารถกรอกข้อมูลเพื่อเลือกช่วงเวลาที่ต้องการดูข้อมูล
 - Start date : วันและเวลาเริ่มต้นของช่วงเวลา
 - End date : วันและเวลาสิ้นสุดของช่วงเวลา
 - Duration : ช่วงเวลาที่ใช้ในการแบ่งกลุ่มข้อมูลเพื่อนับจำนวนคนที่ผ่าน

TrespassingChecker

Connected to TrespassingChecker

Recent Pass: Fri Dec 21 2018 13:26:08 GMT+0700 (เวลาอินโดเจน)

Select time Passing

Start date -- : --

End date -- : --

Duration

5. เมื่อกดปุ่ม submit จะแสดงข้อมูลตามระยะเวลาที่เรากำหนดไว้ โดยจะแสดงตาราง Passing Summary จะเก็บจำนวนคนผ่านเข็นเซอร์ในทุกๆ duration ที่เลือกไว้ และตาราง Passing Log จะเก็บ Log ทุกๆครั้งที่มีคนผ่าน โดยด้านหลังคือเวลาที่ใช้ผ่านเข็นเซอร์หน่วยเป็นวินาที

TrespassingChecker

Connected to TrespassingChecker

Recent Pass: Fri Dec 21 2018 13:26:08 GMT+0700 (ເວລາອິນໄຕຈິນ)

Select time Passing

Start date

End date

Duration

Passing Summary

Time	Total Passes
17-12-2018 08:30:00 - 17-12-2018 09:00:00	20
17-12-2018 09:00:00 - 17-12-2018 09:30:00	35
17-12-2018 09:30:00 - 17-12-2018 10:00:00	3
17-12-2018 10:00:00 - 17-12-2018 10:30:00	2

Passing log

Time	Pass Duration (Seconds)
17-12-2018 08:32:58	1
17-12-2018 08:33:10	2
17-12-2018 08:33:16	1
17-12-2018 08:33:21	1
17-12-2018 08:33:55	2
17-12-2018 08:34:12	1
17-12-2018 08:34:17	1
17-12-2018 08:34:25	1
17-12-2018 08:35:00	1
17-12-2018 08:42:00	1
17-12-2018 08:42:11	1
17-12-2018 08:55:53	1
17-12-2018 08:56:03	1
17-12-2018 08:56:11	1
17-12-2018 08:56:26	1
17-12-2018 08:58:35	1
17-12-2018 08:58:57	1
17-12-2018 08:59:00	1
17-12-2018 08:59:18	8
17-12-2018 09:00:00	1

Coding

STM32

```
uint32_t Read_Time(void)
{
    __IO uint8_t flag=0;
    __IO uint32_t disTime=0;
    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_9,GPIO_PIN_SET);
    HAL_Delay(10);
    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_9,GPIO_PIN_RESET);

    while(flag == 0)
    {
        while(HAL_GPIO_ReadPin(GPIOB,GPIO_PIN_8) == 1)
        {
            disTime++; //+1000 Because of micro
            flag = 1;
            //HAL_Delay(1);
        }
    }
    return disTime;
}
```

Code ส่วนของการอ่านข้อมูลจาก Ultrasonic Sensor โดย จะมีการ Trigger ด้วยไฟ High เป็นเวลา 10 ms เพื่อบอก sensor ให้วัดระยะทางได้แล้ว

ตัว sensor ก็จะวัดระยะทางที่เสียงใช้เดินทางไปกลับ แล้ว return ออกมา

```
void Set_Threshold(void){
    uint32_t summation = 0;
    HAL_GPIO_TogglePin(GPIOB,GPIO_PIN_15);
    HAL_Delay(5000); //Make sensor stable

    HAL_GPIO_TogglePin(GPIOB,GPIO_PIN_14);
    for(int i = 0;i<20;i++){
        summation += Read_Time();
        HAL_Delay(250);
    }
    uint32_t avg = summation/20;
    avg = avg*(120); //20% threshold
    threshold = avg/100;
    lowerBound = summation/20;
    lowerBound = lowerBound*75;
    lowerBound = lowerBound/100;
    HAL_GPIO_TogglePin(GPIOB,GPIO_PIN_14);
    HAL_GPIO_TogglePin(GPIOB,GPIO_PIN_15);

}
```

Code ส่วนของการตั้งค่าเริ่มต้น แบบอัตโนมัติ ว่าค่าที่ sensor อ่านได้โดยทั่วไปเป็นเท่าไหร่ แล้วมากแค่ไหนหรือน้อยแค่ไหน แปลว่ามีคนผ่าน

```
i = Read_Time();

if(i>threshold || i<lowerBound){
    HAL_GPIO_TogglePin(GPIOB,GPIO_PIN_12);
    HAL_Delay(250);
    HAL_GPIO_TogglePin(GPIOB,GPIO_PIN_12);
    timeDuration++;
}
else{
    HAL_Delay(250);
    if(timeDuration>0){
        sprintf(earth,"%d\r\n",1+timeDuration/4);
        HAL_UART_Transmit(&huart2,earth,sizeof(earth),50); //send to nodeMCU
    }
    timeDuration=0;
}
```

Code ใน while loop จะคอยอ่านค่าของเซ็นเซอร์ทุกๆ 250 ms และหาก sensor มีการเปลี่ยนแปลงต่างไปจากปกติระบบจะ เพิ่มค่า duration และเมื่อการผ่านจบลง ระบบจะส่งข้อมูลช่วงเวลาไปให้ NodeMCU

NodeMCU

```
if(Serial.available()) {
    String s = Serial.readString(); //read from uart
    String jsonData = "pass:" + s;
    jsonData += "";
    Serial.println("Publish...");
    Serial.println("String is " + s);
    microgear.chat("HTML_web", s);
    microgear.writeFeed("PassLog", jsonData);
    Serial.println(jsonData);
}
```

เมื่อ STM32 ได้ส่งข้อมูลมา NodeMCU จะรับข้อมูลนั้น แล้วแปลงรูปแบบเป็น JSON แล้วจึงส่งไปที่ NetPie Feed รวมถึง ส่งข้อมูลไปให้ browser ด้วย

HTML+JS

```
microgear.on('message', function (topic, msg) {
    var date = Date();
    document.getElementById("recent").innerHTML =date;
    console.log(msg); // for debug
});
```

จะมีฟังก์ชัน รอรับข้อความจาก NodeMCU เพื่อรอการ update ข้อมูลของคนผ่านล่าสุด โดยจะเอาเวลาที่มีอยู่ไว้ผ่านนั้นมาแสดง

```
// Begin accessing JSON data here
var data = JSON.parse(this.response);
console.log(request.status);
if (request.status >= 200 && request.status < 400) {
    myjson = data;
    console.log(data)
    var summary = document.getElementById("summaryTable");
    summary.innerHTML = genSummaryTable();
    var log = document.getElementById("logTable");
    log.innerHTML = json2table();
} else {
    var error = document.getElementById("error");
    error.innerHTML = 'ERROR';
}
}
```

ในส่วนนี้จะเป็นการส่ง request ไปให้ NetPie เพื่อขอข้อมูลมาจาก Feed ที่เป็น JSON และแปลงเป็นตารางออกมาระบบแสดง

```
function getLogTable(dateStart, dateEnd) //generating html code of json data
{
    var headerRow = '';
    var headers = '';
    var valueRows = '';
    var valueRow = '';
    var values = '';
    var table = '';
    headers += '<th>Time</th><th>Pass Duration (Seconds)</th>';
    headerRow = '<tr>' + headers + '</tr>';
    for(i=0; i < myjson.data[0].values.length; i++)
    {
        values = '';
        //console.log(myjson.data[0].values[i][0], dateStart, dateEnd);
        if(checkTime(myjson.data[0].values[i][0], dateStart, dateEnd))
        {
            values+= '<td><center>' + tstoTime(myjson.data[0].values[i][0]) + '</center></td>'; //timestamp
            for(j=0; j < myjson.data.length; j++)
            {
                values+= '<td><center>' + myjson.data[j].values[i][1] + '</center></td>'; //values entry
            }
            valueRow = '<tr>' + values + '</tr>'; //one value row
            valueRows = valueRows + valueRow; //all value rows
        }
    }
    table = '<table>' + headerRow + valueRows + '</table>';
    return table;
}
```

ในส่วนนี้จะเป็นการนำข้อมูล log ที่เก็บไว้มาแสดงลงในเว็บ

```

function genSummaryTable(dateStart, dateEnd, blockDuration)
{
    var headerRow = '';
    var headers = '';
    var valueRows = '';
    var valueRow = '';
    var values = '';
    var table = '';
    headers += '<th>Time</th><th>Total Passes</th>';
    headerRow = '<tr>' + headers + '</tr>';
    var blockStart = new Date(dateStart);
    var blockEnd = new Date(dateStart);
    blockEnd.setMinutes(blockEnd.getMinutes() + parseInt(blockDuration));
    //console.log(blockStart, blockEnd);
    var count, temp_date;
    var i = 0;
    while(1)
    {
        //onsole.log(dtCompare(blockStart, dateEnd));
        //console.log(blockStart, blockEnd);
        if(dtCompare(blockStart, dateEnd) >= 0) break;
        if(dtCompare(blockEnd, dateEnd) == 1) blockEnd = Date(dateEnd);
        //if(i == myjson.data[0].values.length) break;
        //console.log(blockStart, blockEnd);
        count = 0;
        while(1)
        {
            if(i == myjson.data[0].values.length) break;
            temp_date = new Date(myjson.data[0].values[i][0]);
            console.log(temp_date, blockEnd);
            if(dtCompare(temp_date, blockEnd) == 1) break; //TBM
            if(dtCompare(temp_date, blockStart) >= 0) count++;
            i++;
        }
        //console.log(count);
        if(count != 0)
        {
            values = '';
            values+= '<td><center>' + tstoTime(blockStart) + " - " + tstoTime(blockEnd) + '</center></td>';
            values+= '<td><center>' + count + '</center></td>';
            valueRow = '<tr>' + values + '</tr>'; //one value row
            valueRows = valueRows + valueRow; //all value rows
        }
        blockStart.setMinutes(blockStart.getMinutes() + parseInt(blockDuration));
        //console.log(blockEnd.getMinutes() + parseInt(blockDuration));
        blockEnd.setMinutes(blockEnd.getMinutes() + parseInt(blockDuration));
        //console.log(blockStart, blockEnd);
    }
    table = '<table>' + headerRow + valueRows + '</table>';
    return table;
}

```

ในส่วนนี้จะเป็นการสร้างตารางเพื่อมาแบ่งข้อมูลตาม duration ที่ตั้งไว้