# เครื่องควบคุมและสั่งการเล่นมัลติมีเดียระยะไกลผ่านอินเตอร์เน็ต ด้วยไมโครคอนโทรลเลอร์

# MULTIMEDIA REMOTE CONTROLLER WITH MICROCONTROLLER

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

# เครื่องควบคุมและสั่งการเล่นมัลติมีเดียระยะไกลผ่านอินเตอร์เน็ต ด้วยไมโครคอนโทรลเลอร์

# MULTIMEDIA REMOTE CONTROLLER WITH MICROCONTROLLER

วรวุฒิ บุญเกต
อัครชัย แต้มชัชวาลย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# MULTIMEDIA REMOTE CONTROLLER WITH MICROCONTROLLER

AKARACHAI TAMCHASCHAVAL

WORAWUT BOONKATE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERRING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUE OF TECHNOLOGY LARDKRABANG

ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2559

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

**หัวข้อปริญญานิพนธ์** เครื่องควบคุมและสั่งการเล่นมัลติมีเดียระยะไกลผ่านอินเตอร์เน็ต
ด้วยไมโครคอนโทรลเลอร์

MULTIMEDIA REMOTE CONTROLLER WITH MICROCONTROLLER

**นักศึกษาผู้จัดทำ** นายวรวุฒิ บุญเกต **รหัสนักศึกษา** 56011069

นายอัครชัย แต้มชัชวาลย์ **รหัสนักศึกษา** 56011438

**ปริญญา** วิศวกรรมศาสตรบัณฑิต

**สาขาวิชา** วิศวกรรมการวัดคุม

**ปีการศึกษา** 2559

| อาจารย์ผู้ควบคุมวิทยานิพนธ์ | ลายมือชื่อ |
|---|---|
| รศ.ทรงชัย วีระทวีมาศ | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **หัวข้อปริญญานิพนธ์** | เครื่องควบคุมและสั่งการเล่นมัลติมีเดียระยะไกลผ่านอินเตอร์เน็ต ด้วยไมโครคอนโทรลเลอร์<br>MULTIMEDIA REMOTE CONTROLLER WITH MICROCONTROLLER |
| **นักศึกษาผู้จัดทำ** | นายวรวุฒิ บุญเกต **รหัสนักศึกษา** 56011069<br>นายอัครชัย แต้มชัชวาลย์ **รหัสนักศึกษา** 56011438 |
| **อาจารย์ที่ปรึกษา** | รองศาสตรจารย์ทรงชัย วีระทวีมาศ |
| **ปีการศึกษา** | 2559 |


## บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อศึกษาวิจัยการทำงานแบบไร้สายของไมโครคอนโทรลเลอร์ในการควบคุมเครื่องเล่นมัลติมีเดียจากระยะไกล ทำให้ไม่ต้องใช้คนควบคุมนั่งเฝ้าเครื่องเพื่อเปิดหรือปิด และเพื่อนำมาประยุกต์ใช้ในระบบอุตสาหกรรม โดยในปัจจุบันนั้นการควบคุมระยะไกลเป็นที่นิยมแต่ยังไม่เป็นวงกว้างนักเนื่องจากต้องใช้อุปกรณ์ในการควบคุมหลายชุดหรือหากใช้ซอฟต์แวร์ก็มีราคาแพง โครงการนี้จึงจัดทำขึ้นเพื่อนำเทคโนโลยีไมโครคอนโทรลเลอร์อย่างง่าย หรือ บอร์ด ESPino เข้ามาประยุกต์ใช้งานร่วมกับระบบอินเตอร์เน็ต และ ระบบอินเตอร์เน็ตไร้สายที่เพื่อให้สามารถควบคุมเครื่องเล่นมัลติมีเดียซึ่งในขอบเขตของเรานี้ระบุให้เป็นการเล่นไฟล์ MP3 และจะควบคุมได้ผ่านโทรศัพท์มือถือหรือคอมพิวเตอร์ที่มีเว็บบราวเซอร์โดยสามารถควบคุมได้จากทุกที่ที่สามารถเข้าถึง Internet ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **Thesis Title** | MULTIMEDIA REMOTE CONTROLLER WITH MICROCONTROLLER |
| **Authors** | Mr.Akarachai Tamchaschaval |
| | Mr.Worawut Boonkate |
| **Thesis Advisor** | Assoc.Prof. Songchai Weerathaweemas |
| **Year** | 2016 |


## ABSTRACT

Nowadays, the remote control through the Internet requires a control device or software that very expensive. Therefore, the purpose of this project is to study the use of microcontrollers for remote control and program media playback through wireless Internet that can be control a media player that store MP3 files in the memory card and control playback through a smart phone or computer. This can be controlled from anywhere with Internet access.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีด้วยความกรุณาของอาจารย์ที่ปรึกษาปริญญานิพนธ์ ได้แก่ รองศาสตรจารย์ทรงชัย วีระทวีมาศ และ อาจารย์ในหลักสูตรวิศวกรรมการวัดคุม ที่ได้ให้คำปรึกษา การแนะนำ ชี้แนะในการศึกษาค้นคว้า แนะนำขั้นตอนและวิธีจัดทำโครงงานจนสำเร็จลุล่วงด้วยดี และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้การสนับสนุนเงินในการดำเนินงาน คณะผู้จัดทำจึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอกราบขอบพระคุณ บิดา มารดา ที่ให้กำลังใจในการศึกษาเล่าเรียนและให้การสนับสนุนในทุกๆด้านและสมาชิกในกลุ่มที่ให้ความร่วมมือเป็นอย่างดีในการทำโครงงานครั้งนี้จนกระทั่งประสบความสำเร็จด้วยดี

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า


บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....VII
สารบัญรูป.....VIII

บทที่ 1 บทนำ.....10
1.1 ความสำคัญของปริญญานิพนธ์.....10
1.2 วัตถุประสงค์ของวิทยานิพนธ์.....10
1.3 ขอบเขตวิทยานิพนธ์.....10
1.4 ขั้นตอนการศึกษา.....10
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....11

บทที่ 2 เอกสารที่เกี่ยวข้อง.....12
2.1 ความสำคัญของเทคโนโลยีสารสนเทศและอินเตอร์เน็ต.....12
2.1.1 ความสำคัญของเทคโนโลยีสารสนเทศ.....12
2.1.2 ความสำคัญของอินเทอร์เน็ต.....13
2.2 แนวคิดเรื่อง Internet of Things .....13
2.2.1 ความหมายของ Internet of Everything (Internet of Things).....13
2.2.2 ประวัติความเป็นมาและพัฒนาการของแนวคิด Internet of Things.....14
2.3 ข้อมูลเกี่ยวกับไมโครคอนโทรลเลอร์และภาษาที่ใช้ในการเขียนโปรแกรมควบคุม.....16
2.3.1 ไมโครคอนโทรลเลอร์.....16
2.3.1.1 บอร์ด Arduino.....17
2.3.1.1.1 ความหมายของ Arduino.....17
2.3.1.1.2 รูปแบบการเขียนโปรแกรมบน Arduino.....18
2.3.1.1.3 บอร์ด Arduino ในรูปแบบต่างๆ.....18
2.3.1.2 บอร์ด ESPino.....19
2.3.1.2.1 ความหมายของบอร์ด ESPino.....19



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

หน้า


2.3.1.2.2 คุณสมบัติของบอร์ด ESPino.....20
2.3.1.2.3 คุณสมบัติของตัวโมดูล WROOM-02.....21
2.3.1.2.4 คุณสมบัติของบอร์ด ESPino by ThaiEasyElec.com.....22
2.3.1.2.5 ส่วนประกอบของบอร์ด ESPino by ThaiEasyElec.com....23
2.3.1.2.6 เตรียมการเขียนโปรแกรม ESPino ด้วย Arduino IDE.....24
2.3.1.2.7 ผังวงจรบอร์ด ESPino.....24
2.3.2 ภาษาที่ใช้เขียน โปรแกรมควบคุมไมโครคอนโทลเลอร์.....25
2.4 Cloud และ Protocol ที่ใช้ในการสื่อสารระหว่าง Controller กับ User.....25
2.4.1 ความหมายของ Cloud.....25
2.4.2 Protocol ที่ใช้ในการสื่อสารระหว่าง Controller กับ User.....26
2.4.2.1 ความหมายของโปรโตคอล (Protocol ).....26
2.4.2.2 ความหมายของ MQTT Protocol.....27
2.5 ภาษาที่ใช้ในการเขียน Web Site รับคำสั่งควบคุม.....29
2.5.1 ความหมายของ JavaScript.....29
2.5.2 ความเป็นมาของ JavaScript.....29
2.5.3 สิ่งที่ JavaScript สามารถทำได้.....29
2.5.4 ข้อดีและข้อเสียของ JavaScript.....30

**บทที่ 3 วิธีการดำเนินการ.....31**
3.1 องค์ประกอบและขั้นตอนการทำงานโดยรวมของระบบ.....31
3.2 รายละเอียดการออกแบบส่วนต่างๆของระบบ.....32
3.2.1 การออกแบบส่วนฮาร์ดแวร์.....32
3.2.2 การออกแบบส่วนซอฟต์แวร์.....34
3.2.2.1 การเชื่อมต่อ Wi-Fi และการรับค่าจากจาก MQTT Broker.....34
3.2.2.2 การนำเอาสัญญาณหลัง Payload มาใช้สั่งการเล่นไฟล์ MP3.....35
3.2.2.3 เตรียมการเขียนเว็บเพื่อใช้ Publish ข้อมูลคำสั่ง.....36
3.3 รายละเอียดการทำงานส่วนต่างๆของระบบ.....37



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

หน้า


**บทที่ 4 ผลการทดลอง.....39**

4.1 ผลการทดสอบการเชื่อมต่อ MQTT ผ่านอินเตอร์เน็ตในวงเปิด.....39

4.2 ผลการทดสอบการสื่อสารของ MQTT กับ MQTT Broker.....40

4.3 ผลการทดสอบการเล่นไฟล์เสียงบน Arduino.....41

4.4 ผลการทดสอบการควบคุมการเล่นไฟล์เสียงบน Arduino ผ่าน MQTT Broker.....42

4.5 ผลการทดสอบการควบคุมการเล่นไฟล์เสียงผ่านเว็บบนคอมพิวเตอร์.....44

4.6 ผลการทดสอบการควบคุมการเล่นไฟล์เสียงผ่านเว็บบนสมาร์ทโฟน.....46

4.7 ผลการทดสอบการควบคุมผ่านเว็บโดยเป็นเว็บที่อยู่บนเซิร์ฟเวอร์.....45

4.8 ผลการทดสอบการควบคุมผ่านเว็บโดยเป็นเว็บที่อยู่บนเซิร์ฟเวอร์.....47
และตั้งเวลาการทำงาน

**บทที่ 5 สรุปผลและข้อเสนอแนะ.....48**

5.1 สรุปผล.....48

5.2 ข้อเสนอแนะ.....48

**บรรณานุกรม.....49**

**ภาคผนวก.....50**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่ หน้า

2.1 ขาของ Espino.....23



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป


| รูปที่ | หน้า |
|---|---|
| 2.1 Internet Connects Us | 15 |
| 2.2 Smart Home | 15 |
| 2.3 Smart City | 15 |
| 2.4 Arduino Uno R3 | 18 |
| 2.5 Arduino Ethernet with PoE Module | 19 |
| 2.6 บอร์ด ESPino | 19 |
| 2.7 ส่วนประกอบของบอร์ด ESPino | 23 |
| 2.8 ผังวงจรบอร์ด ESPino | 24 |
| 2.9 Cloud Computing | 26 |
| 2.10 แสดงการ Publisher MQTT | 28 |
| 3.1 บล็อกไดอะแกรมแสดงขั้นตอนการทำงานโดยรวมของระบบ | 31 |
| 3.2 แสดงการต่อวงจรรวมของระบบในส่วนของฮาร์ดแวร์ | 32 |
| 3.3 บอร์ด ESPino | 33 |
| 3.4 DF Player Mini | 34 |
| 3.5 ชิ้นงานเบื้องต้นที่แสดงการเชื่อมต่อและหลักการทำงาน | 38 |
| 4.1 แสดง Serial Monitor | 39 |
| 4.2 แสดง Serial Monitor | 39 |
| 4.3 แสดง MQTT Web socket | 40 |
| 4.4 แสดง MQTT Web socket | 40 |
| 4.5 แสดง ผลการทดสอบส่งข้อมูล โดยที่ ESPino รับ ข้อมูลจาก MQTT ได้ | 41 |
| 4.6 แสดงตัวอย่างชิ้นงาน | 41 |
| 4.7 แสดงหน้า Serial Monito | 42 |
| 4.8 แสดงการเชื่อมต่อ ESPino เข้ากับ Wifi Router และ ESPino เชื่อมต่อกับ MQTT | 43 |
| 4.9 แสดงการทดสอบส่งคำสั่งด้วยคำสั่งพื้นฐาน “Play” | 43 |
| 4.10 ตัวอย่างชิ้นงานที่เล่นเพลงจากคำสั่ง Play | 44 |
| 4.11 แสดงหน้า Web Site ที่สามารถเชื่อมต่อกับ MQTT ได้สำเร็จ | 44 |
| 4.12 แสดงหน้า Web Site ที่ไม่สามารถเชื่อมต่อได้ | 45 |
| 4.13 แสดงหน้า Serial Monitor ที่รับคำสั่ง Play | 45 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป (ต่อ)

**รูปที่** **หน้า**

4.14 แสดงหน้า Web Site ในแนวนอน บนโทรศัพท์.....46

4.15 แสดงหน้า Web Site ที่เพิ่มช่องสำหรับการกำหนดเวลา.....46



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันการควบคุมแบบไร้สายสร้างความสะดวกสบายให้กับผู้คนมากในทุกด้าน โครงการนี้จัดทำขึ้นเพื่อศึกษาวิจัยการทำงานแบบไร้สาย ร่วมกับ ไมโครคอนโทรลเลอร์ในการควบคุมมัลติมีเดียจากระยะไกล เนื่องจากปัจจุบันนั้นการควบคุมระยะไกลเริ่มเป็นที่นิยม แต่ยังไม่เป็นวงกว้างมากนัก เนื่องจากต้องใช้อุปกรณ์ในการควบคุมหรือซอฟต์แวร์ที่มีราคาแพง ข้าพเจ้าจึงจัดทำโครงการชิ้นนี้ขึ้น เพื่อนำอุปกรณ์ไมโครคอนโทรลเลอร์อย่างง่าย คือ ESPino มาประยุกต์ใช้งานร่วมกับระบบอินเตอร์เน็ต และระบบแลนไร้สายเพื่อให้เราสามารถควบคุมเครื่องเล่นมัลติมีเดียได้ ซึ่งในขอบเขตของเรานี้ระบุให้เป็นการควบคุมเล่นไฟล์ MP3 และจะใช้ควบคุมได้ผ่านโทรศัพท์มือถือหรือคอมพิวเตอร์ที่มีเว็บบราวเซอร์ โดยสามารถควบคุมได้จากทุกที่ที่สามารถเข้าถึง Internet ได้อีกด้วย

## 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1) ศึกษาหลักการทำงานและเขียนโปรแกรมควบคุมของ ESPino
2) ศึกษาการทำงานของอุปกรณ์เครือข่ายไร้สาย
3) ศึกษาการสื่อสารของข้อมูลผ่าน Protocol ที่มีขนาดเล็กๆ
4) ศึกษาการเขียนโปรแกรมควบคุมการเล่นไฟล์ MP3
5) ศึกษาการเขียนเว็บเพื่อใช้สั่งงานระบบ

## 1.3 ขอบเขตปริญญานิพนธ์

1) สามารถควบคุมการเล่นไฟล์ MP3 ผ่านระบบอินเตอร์เน็ตได้โดยที่ไม่จำเป็นต้องอยู่ในวงเครือข่ายเดียวกัน

## 1.4 ขั้นตอนการศึกษา

1) ศึกษาหลักการทำงานและช่องการเชื่อมต่อต่างๆ ของ ESPino
2) ศึกษาการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ชนิด ESPino
3) ศึกษาการควบคุมระยะไกลผ่านทางอินเตอร์เน็ต และ ระบบแลนไร้สาย
4) ศึกษาการเขียนเว็บเพื่อควบคุมจากอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ความรู้ในระบบการทำงานของไมโครคอนโทรลเลอร์ ESPino

2) ความรู้ในด้านโปรแกรมคำสั่งต่างๆ การเขียน Code ควบคุม

3) การใช้งานอุปกรณ์ต่อเชื่อมต่างๆ

4) เครื่องควบคุมและสั่งการเล่นมัลติมีเดียระยะไกลผ่านอินเตอร์เน็ตด้วยไมโครคอนโทรลเลอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ทฤษฎีและหลักการที่เกี่ยวข้อง

ในการจัดทำโครงงานเครื่องควบคุมและสั่งการเล่นมัลติมีเดียระยะไกลผ่านอินเตอร์เน็ตด้วยไมโครคอนโทรลเลอร์นี้ ผู้จัดทำโครงงานได้ศึกษาเอกสารและจากเว็ปไซต์ต่างๆที่เกี่ยวข้องดังต่อไปนี้

2.1 ความสำคัญของเทคโนโลยีสารสนเทศและอินเตอร์เน็ต

2.2 แนวคิดเรื่อง Internet of Things

2.3 ข้อมูลเกี่ยวกับไมโครคอนโทรลเลอร์และภาษาที่ใช้ในการเขียนโปรแกรมควบคุม

2.4 Cloud และ Protocol ที่ใช้ในการสื่อสารระหว่าง Controller กับ User

2.5 ภาษาที่ใช้ในการเขียน Web site รับคำสั่งควบคุม

## 2.1 ความสำคัญของเทคโนโลยีสารสนเทศและอินเตอร์เน็ต

### 2.1.1 ความสำคัญของเทคโนโลยีสารสนเทศ

โอภาส เอี่ยมสิริวงศ์ (2551 : 12) ได้สรุปว่า เทคโนโลยีสารสนเทศมีความสำคัญต่อการพัฒนาประเทศในด้านต่างๆ เป็นอย่างมาก ตัวอย่างเช่น

1) การศึกษาเทคโนโลยีสารสนเทศช่วยในด้านการค้นคว้าศึกษาแหล่งข้อมูล ทำให้การศึกษาง่ายขึ้นและไร้ขีดจำกัด ผู้เรียนมีความสะดวกในการค้นคว้าวิจัย

2) การดำรงชีวิตประจำวัน ทำให้มีความสะดวกคล่องตัวและรวดเร็วในการทำกิจกรรมต่างๆที่เกิดขึ้นในชีวิตประจำวัน สามารถทำงานได้หลากหลายอย่างในเวลาเดียวกันได้ หรือ ทำงานใช้เวลาน้อยลง

3) การดำเนินธุรกิจ ทำให้มีการข่งขันระหว่างธุรกิจมากขึ้น ทำให้มีการพัฒนาองค์กรเพื่อให้ทันกับข้อมูล ข่าวสารอยู่ตลอดเวลาอันส่งผลต่อการพัฒนาประเทศอย่างต่อเนื่อง

4) อัตราการขยายตัวทุกๆด้านที่เพิ่มขึ้นอย่างรวดเร็ว เพราะมีการติดต่อสื่อสารที่เจริญก้าวหน้าทันสมัย รวดเร็ว ถูกต้องและทำให้เป็นโลกที่ไร้พรมแดน

5) ระบบการทำงานมีคอมพิวเตอร์มาใช้ซึ่งสามารถทำงานได้มากขึ้น งานบางอย่างมนุษย์ทำไม่ได้ก็ใช้คอมพิวเตอร์ช่วยทำงานแทนซึ่งได้ผลถูกต้องรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2 ความสำคัญของอินเทอร์เน็ต

สุพรรษา ยวงทอง (2548 : 9) กล่าวว่า ในปัจจุบันอินเทอร์เน็ตมีบทบาทและมีความสำคัญต่อชีวิตประจำวันของคนเราเป็นอย่างมาก เพราะทำให้วิถีชีวิตเราทันสมัยและทันเหตุการณ์อยู่เสมอ เนื่องจากอินเทอร์เน็ตจะมีการเสนอข้อมูลข่าวปัจจุบัน และสิ่งต่างๆที่เกิดขึ้นให้ผู้ใช้ทราบเปลี่ยนแปลงไปทุกวัน สารสนเทศที่เสนอในอินเทอร์เน็ตจะมีมากมายหลายรูปแบบเพื่อสนองความสนใจและความต้องการของผู้ใช้ทุกกลุ่ม อินเทอร์เน็ตจึงเป็นแหล่งสารสนเทศสำคัญทุกคนเพราะค้นคว้าสิ่งที่ตนสนใจได้ทันทีโดยไม่ต้องเสียเวลา เดินทางไปที่ห้องสมุด หรือแม้แต่การรับรู้ข่าวสารทั่วโลกก็สามารถค้นหาและอ่านได้ในอินเทอร์เน็ต จากเว็บไซต์ต่างๆของหนังสือพิมพ์ ดังนั้นอินเทอร์เน็ตจึงมีความสำคัญกับวิถีชีวิตของคนเราในปัจจุบันเป็นอย่างมากในทุกด้าน ไม่ว่าจะเป็นบุคคลที่อยู่ในวงการธุรกิจ บุคคลที่อยู่ในด้านการศึกษา ทุกคนต่างก็ได้รับประโยชน์จากอินเทอร์เน็ตด้วยกันทั้งนั้น

อภิชัย เรืองศิริปิยะกุล (2549 : 26) กล่าวว่า ด้านการศึกษา อินเทอร์เน็ตมีความสำคัญ ดังนี้

1) สามารถใช้เป็นแหล่งค้นคว้าข้อมูล ไม่ว่าจะเป็นข้อมูลทางวิชาการ ข้อมูลด้านความบันเทิง ด้านการแพทย์ และอื่นๆที่น่าสนใจ

2) ระบบเครือข่ายอินเทอร์เน็ต จะทำหน้าที่เปรียบเสมือนเป็นห้องสมุดขนาดใหญ่

3) นักเรียน นักศึกษา สามารถใช้อินเทอร์เน็ตติดต่อมหาวิทยาลัยหรือโรงเรียนอื่นๆ เพื่อค้นหาข้อมูลที่กำลังศึกษาอยู่ได้ ทั้งที่ข้อมูลที่เป็นข้อความเสียง ภาพเคลื่อนไหวต่างๆ

## 2.2 แนวคิดเรื่อง Internet of Things

### 2.2.1 ความหมายของ Internet of Everything (Internet of Things)

อุไรพร ชลสิริรุ่งสกุล (2554 : 18) กล่าวว่า เทคโนโลยี Internet of Things (IoT) หมายถึงการที่สิ่งต่างๆ ถูกเชื่อมโยงทุกสิ่งทุกอย่างเข้าสู่โลกอินเทอร์เน็ต ทำให้มนุษย์สามารถสั่งการ ควบคุมใช้งานอุปกรณ์ต่างๆ ผ่านทางเครือข่ายอินเตอร์เน็ต เช่น การสั่งเปิด-ปิด อุปกรณ์เครื่องใช้ไฟฟ้า รถยนต์ โทรศัพท์มือถือ เครื่องมือสื่อสาร เครื่องใช้สำนักงาน เครื่องมือทางการเกษตร เครื่องจักรในโรงงานอุตสาหกรรม อาคาร บ้านเรือน เครื่องใช้ในชีวิตประจำวันต่างๆ ผ่านเครือข่ายอินเทอร์เน็ต เป็นต้น โดยเทคโนโลยีนี้จะมีทั้งประโยชน์และความเสี่ยงไปพร้อมๆกัน เพราะหากระบบรักษาความปลอดภัยไม่ดีพอ จะทำให้ผู้ไม่ประสงค์ดีเข้ามากระทำการที่ไม่พึงประสงค์ต่ออุปกรณ์ ข้อมูลสารสนเทศ หรือความเป็นส่วนตัวของบุคคลได้ ดังนั้น การพัฒนาไปสู่ Internet of Things จึงมีความจำเป็นต้องพัฒนามาตรการและเทคนิคในการรักษาความปลอดภัยไอทีควบคู่กันไปด้วย หรือบางแห่งเรียกว่า M2M ย่อมาจาก Machine to Machine คือ เทคโนโลยีอินเอร์เน็ตที่เชื่อมอุปกรณ์กับเครื่องมือต่างๆ เช่น โทรศัพท์มือถือ รถยนต์ ตู้เย็น โทรทัศน์ และ อื่นๆ เข้าไว้ด้วยกัน โดยการเชื่อมโยงช่วยให้การสื่อสารกันได้ผ่านระบบอินเทอร์เน็ต จากการคาดการณ์ในปี ค.ศ. 2020 สิ่งต่างๆ กว่าแสนล้านชิ้นจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเชื่อมต่อกันได้ด้วยระบบ IoT ซึ่งจะส่งผลให้ผู้บริโภคทั่วไปจะเริ่มคุ้นเคยกับเทคโนโลยีที่ทำให้พวกเขา สามารถควบคุมสิ่งของต่างๆ ทั้งจากในบ้านและสำนักงาน หรือ จากที่ไหนก็ได้ทั้งนั้น

### 2.2.2 ประวัติความเป็นมาและพัฒนาการของแนวคิด Internet of Things

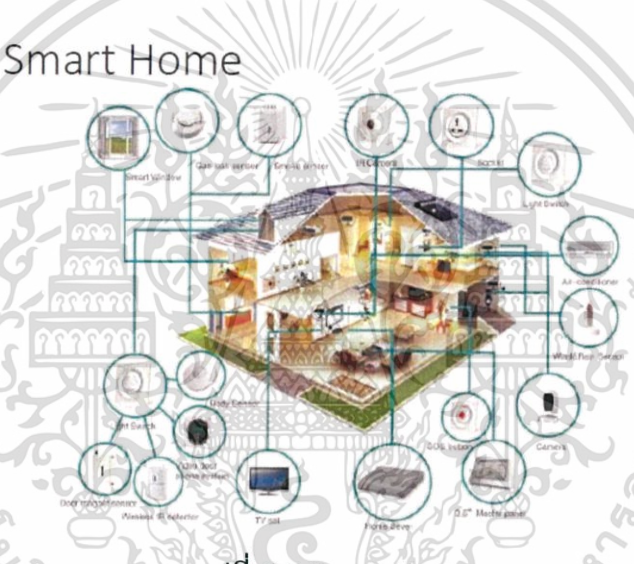
ดร.มหศักดิ์ เกตุฉ่ำ (2556 : 15) ได้สรุปว่า ถูกคิดค้นโดย Kevin Ashton ในปี 1999 ภายใต้โครงการชื่อ "Auto-ID Center" ทีมมหาวิทยาลัย Massachusetts Institute of Technology จากเทคโนโลยี RFID ที่ทำให้เป็นมาตรฐานระดับโลกสำหรับ RFID Sensors ต่างๆที่จะเชื่อมต่อกันได้ ต่อมาในยุคหลังปี 2000 โลกมีอุปกรณ์อิเล็กทรอนิกส์ออกมาเป็นจำนวนมากและมีการใช้คำว่า Smart ซึ่งในที่นี้คือ Smart Device, Smart Grid, Smart Home, Smart Network, Smart Intelligent Transportation

ซินแคล (Bruce Sinclair, 2016 : 24) กล่าวว่า แนวคิดการนำเอาข้อมูลของทุกสิ่งทุกอย่างขึ้นไปไว้บน Internet นั้น ในเวลานี้เทคโนโลยีที่มีอยู่สามารถรองรับแนวคิดนี้เป็นที่เรียบร้อยแล้ว เหลือเพียงแค่การทยอยเปลี่ยนอุปกรณ์แต่ละชนิดในชีวิตประจำวันให้รองรับการใช้งานกับแนวคิด IoT อุปกรณ์ IoT ในตอนนี้มีหลายประเภทที่มีออกมาวางจำหน่ายแล้ว เช่น Smart Watch หลายยี่ห้อที่เก็บเอาข้อมูล GPS และ Heart Rate รวมถึงข้อมูลทางสุขภาพของผู้ใช้แล้วส่งข้อมูลเข้าไปยัง Server ส่วนกลางเพื่อวิเคราะห์สถิติการออกกำลังกายรวม ทั้งวางแผนการออกกำลังกายให้ล่วงหน้า, เครื่องปรับอากาศที่สามารถเชื่อมต่อกับ Wi-Fi ที่บ้าน แม้ว่าเราจะอยู่นอกบ้านสามารถใช้ Application บน Smartphone สั่งงานให้เปิดแอร์ที่บ้านได้

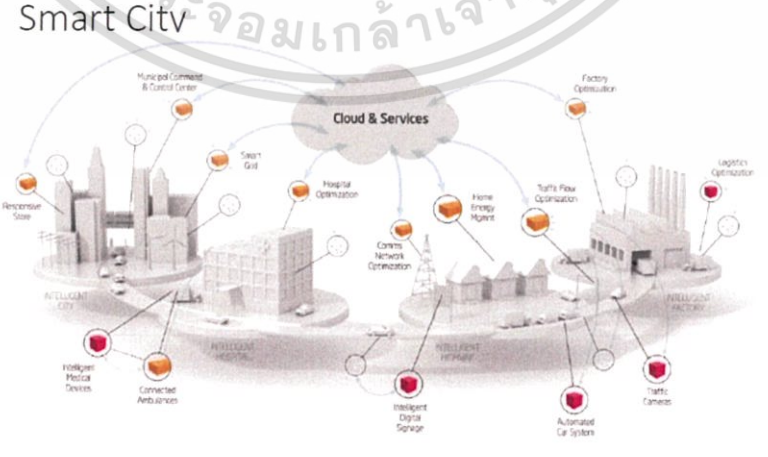
บาร์ก้า (Arshdeep Bahga, 2016 : 32) กล่าวว่า หลายบริษัทในเวลานี้ให้ความสำคัญกับ IoT เป็นอย่างมาก ในแง่การบริหารจัดการทรัพยากรขององค์กรแล้ว หากมีความพร้อมและเตรียมการที่ดีแล้วการนำเอา Machine เข้ามาทำงานแทนที่มนุษย์ถือว่าเป็นประโยชน์อย่างมาก นั่นเพราะมนุษย์เป็นทรัพยากรที่ควบคุมได้ยากที่สุด มีเหนื่อยล้า เบื่อ หิว อารมณ์ไม่ดี ซึ่งจะส่งผลต่องานที่ออกมา ผิดกับ Machine ที่ต้องการพลังงานไฟฟ้าและซ่อมบำรุงรักษาเท่านั้น ไม่ต้องดูแลสวัสดิการดูแล คอยเอาอกเอาใจเหมือนกับมนุษย์ นอกจากนี้ Machine ยังมีศักยภาพในการทำงานมากกว่ามนุษย์ในหลายๆ ด้านอีกด้วย ที่เราเห็นในตอนนี้ก็คือ "AlphaGo ที่เป็น AI (ปัญญาประดิษฐ์) ที่สามารถเอาชนะมนุษย์ได้อย่างไม่มีข้อโต้เถียง"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.1** Internet Connects Us




**รูปที่ 2.2** Smart Home




**รูปที่ 2.3** Smart City

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 ข้อมูลเกี่ยวกับไมโครคอนโทรลเลอร์และภาษาที่ใช้ในการเขียนโปรแกรมควบคุม

### 2.3.1 ไมโครคอนโทรลเลอร์

ความหมายของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ คืออะไร

ดร.เดชฤทธิ์ มณีธรรม ( 2553 : 22) ได้กล่าวว่า ไมโครคอนโทรลเลอร์หรือเอ็มซียู คือ คอมพิวเตอร์ขนาดเล็กที่อยู่บนแผงวงจรรวบรวมที่ซึ่งจะบรรจุไปด้วยแกนกลางโพรเซสเซอร์ หน่วยความจำและส่วนประกอบอินพุตเอาท์พุต

โดยไมโครคอนโทรเลอร์นั้นถูกออกแบบมาเพื่อใช้งานบนคอมพิวเตอร์ส่วนบุคคลและการใช้งานในหลายรูปแบบขึ้นอยู่กับโครงสร้าง และการออกแบบของระบบ โดยที่เราสามารถเขียนโปรแกรมเพื่อควบคุมการทำงานตามที่ต้องการได้

สำเริง เต็มราม ( 2552 : 28) ได้กล่าว ไมโครคอนโทรลเลอร์นั้นถูกใช้ในการของอุปกรณ์อัตโนมัติต่างๆเพื่อใช้ในการควบคุมการผลิต เช่น กล่องควบคุมการทำงานของเครื่องยนต์ อวัยวะเทียม รีโมทคอนโทรลเครื่องจักรในสำนักงานหรือแม้กระทั่งของเล่น โดยยิ่งมีขนาดเล็กก็ยิ่งมีราคาแพง โดยในปัจจุบันได้มีขนาดลดลงเป็นอย่างมาก

โดยภายในไมโครคอนโทรลเลอร์จะประกอบไปด้วย หน่วยประมวลผล หน่วยความจำชั่วคราว หน่วยความจำถาวร พอร์ตอินพุต พอร์ตเอาท์พุต และส่วนพิเศษ ขึ้นอยู่กับกระบวนการผลิตที่แต่ละบริษัทจะผลิตขึ้นมาใส่คุณสมบัติพิเศษลงไป เช่น

- ADC (Analog to Digital) ส่วนภาครับสัญญาณอนาล็อกแปลงไปเป็นสัญญาณดิจิตอล
- DAC (Digital to Analog) ส่วนภาคส่งสัญญาณดิจิตอลแปลงไปเป็นสัญญาณอนาล็อก
- $I^2C$ (Inter Integrate Circuit Bus) เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอกซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ Serial Data (SDA) และสาย Serial Clock (SCL) ซึ่งสามารถเชื่อมต่ออุปกรณ์จำนวนหลายๆตัวเข้าด้วยกันได้ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น

- SPI (Serial Peripheral Interface) เป็นการเชื่อมต่อกับอุปกรณ์เพื่อรับส่งข้อมูลแบบซิงโครนัส(Synchronize) มีสัญญาณนาฬิกาเข้ามาเกี่ยวข้องระหว่างไมโครคอนโทรลเลอร์(Microcontroller) หรือจะเป็นอุปกรณ์ภายนอกที่มีการรับส่งข้อมูลแบบ SPI อุปกรณ์ที่ทำหน้าที่ เป็น Master โดยปกติแล้วจะเป็นไมโครคอนโทรลเลอร์ หรืออาจกล่าวได้ว่าอุปกรณ์ Master จะต้องควบคุมอุปกรณ์ Slave ได้ โดยปกติตัว Slave มักจะเป็นไอซี (IC) หน้าที่พิเศษต่างๆ เช่น ไอซีอุณหภูมิ ไอซีฐานเวลานาฬิกาจริง (Real-Time Clock) หรืออาจเป็นไมโครคอนโทรลเลอร์ที่ทำหน้าที่ในโหมด Slave ก็ได้เช่นกัน

- PWM (Pulse Width Modulation) การสร้างสัญญาณพัลส์แบบสแควร์เวฟที่สามารถปรับเปลี่ยนความถี่และ Duty Cycle ได้เพื่อนำไปควบอุปกรณ์ต่างๆเช่น มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- UART (Universal Asynchronous Receiver Transmitter) ทำหน้าที่ในการรับส่งข้อมูลแบบอะซิงโครนัสสำหรับมาตรฐานการรับส่งข้อมูลแบบ RS-232

ดอนสัน ปงผาบ และ ทิพวัลย์ คำน้ำนอง (2549 :38) ได้แบ่งประเภทไมโครคอนโทรลเลอร์ แบ่งตามสถาปัตยกรรม (การผลิตและกระบวนการทำงานระบบการประมวลผล) ที่มีใช้ในปัจจุบันยกตัวอย่างดังนี้

- ไมโครคอนโทรลเลอร์ตระกูล PIC (บริษัทผู้ผลิต Microchip ไมโครชิป)
- ไมโครคอนโทรลเลอร์ตระกูล MCS51 (บริษัทผู้ผลิต Atmel, Phillips)
- ไมโครคอนโทรลเลอร์ตระกูล AVR (บริษัทผู้ผลิต Atmel)
- ไมโครคอนโทรลเลอร์ตระกูล ARM7, ARM9 (บริษัทผู้ผลิต Atmel, Phillips, Samsung)
- ไมโครคอนโทรลเลอร์ตระกูล Basic Stamp (บริษัทผู้ผลิต Parallax)
- ไมโครคอนโทรลเลอร์ตระกูล PSOC (บริษัทผู้ผลิต CYPRESS)
- ไมโครคอนโทรลเลอร์ตระกูล MSP (บริษัทผู้ผลิต Texas Intruments)
- ไมโครคอนโทรลเลอร์ตระกูล 68HC (บริษัทผู้ผลิต MOTOROLA)
- ไมโครคอนโทรลเลอร์ตระกูล H8 (บริษัทผู้ผลิต Renesas)
- ไมโครคอนโทรลเลอร์ตระกูล RABBIT (บริษัทผู้ผลิต RABBIT SEMICONDUCTOR)
- ไมโครคอนโทรลเลอร์ตระกูล Z80 (บริษัทผู้ผลิต Zilog)
- ไมโครคอนโทรลเลอร์ตระกูล Arduino ซึ่งใช้ชิพ AVR (บริษัทผู้ผลิต Atmel )

### 2.3.1.1 บอร์ด Arduino

#### 2.3.1.1.1 ความหมายของ Arduino

Arduino คืออะไร

ประจิน พลังสันติกุล (2548 : 36) ได้กล่าวว่า Arduino อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดุยโน่) เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษาทั้งนี้ผู้ใช้งานยังสามารถดัดแปลงเพิ่มเติมพัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรม ต่อได้อีกด้วย

แคนเทโฟร่า (Arduino Cantafora, 2546 : 43) กล่าวว่า ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด (ดูตัวอย่างรูปที่ 2.4) หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ (ดูตัวอย่างรูปที่ 2.5) เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประจิน พลังสันติกุล (2548 : 36) กล่าวว่า จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

1. ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
2 มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
3. Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
4. ราคาไม่แพง
5. Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

**2.3.1.1.2 รูปแบบการเขียนโปรแกรมบน Arduino**

1. เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ArduinoIDE ซึ่งสามารถดาวน์โหลดได้จาก Arduino.cc/en/main/software

2. หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้และหมายเลข Com Port

3. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออับโหลดเรียบร้อยแล้ว จะแสดงข้อ ความแถบข้างล่าง “Done Uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที

**2.3.1.1.3 บอร์ด Arduino ในรูปแบบต่างๆ**

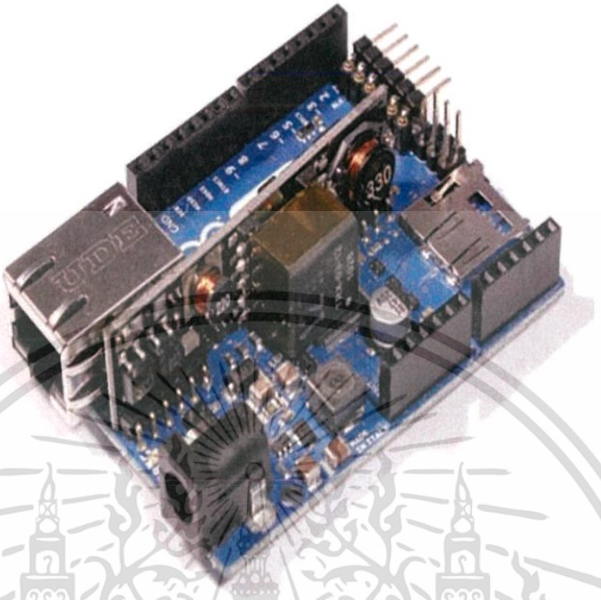
1) Arduino Uno R3 เป็นบอร์ด Arduino ที่ได้รับความนิยมมากที่สุด เนื่องจากราคาไม่แพง ส่วนใหญ่โปรเจคและ Library ต่างๆ ที่พัฒนาขึ้นมา Support จะอ้างอิงกับบอร์ดนี้เป็นหลัก และข้อดีอีกอย่างคือกรณีที่ MCU เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองได้ง่าย

**รูปที่ 2.4** Arduino Uno R3

2) Arduino Ethernet with PoE module เป็นบอร์ด Arduino ที่ใช้ MCU เบอร์เดียวกับ Arduino Uno SMD ในบอร์ดมีชิป Ethernet และช่องสำหรับเสียบ SD Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวมทั้งโมดูล POE ทำให้บอร์ดนี้ สามารถใช้แหล่งจ่ายไฟจากสาย LAN ได้โดยตรง โดยไม่ต้องต่อ Adapter เพิ่ม แต่บอร์ด Arduino Ethernet with PoE Module นี้จะไม่มีพอร์ต USB ทำให้เวลาโปรแกรมต้องต่อบอร์ด USB to Serial Converter เพิ่มเติม



**รูปที่ 2.5** Arduino Ethernet with PoE module

### 2.3.1.2 บอร์ด ESPino

#### 2.3.1.2.1 ความหมายของ บอร์ด ESPino

ThaiEasyElec (2560 :46) กล่าวว่า บอร์ด ESPino เป็นบอร์ดไมโครคอนโทรลเลอร์พร้อมโมดูลสื่อสารเครือข่ายคอมพิวเตอร์ไร้สาย (Wireless LAN) บอร์ด ESPino ใช้โมดูล WROOM-02 (EFDV455) ชิพ ESP8266 Wi-Fi SoC จากทาง Espressif Systems เป็นบอร์ดที่ข้าพเจ้าเลือกใช้เนื่องจากสามารถใช้โปรแกรม Arduino IDE ในการเขียนได้และมีฟังชั่นการทำงานครอบคลุมความต้องการของโครงการนี้

**รูปที่ 2.6** บอร์ด ESPino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.3.1.2.2 คุณสมบัติของบอร์ด ESPino

ThaiEasyElec (2560 :58) ได้กล่าวว่า บอร์ด ESPino เป็นบอร์ดไมโครคอนโทรลเลอร์พร้อมโมดูลสื่อสารเครือข่ายคอมพิวเตอร์ไร้สาย (Wireless LAN) บอร์ด ESPino ใช้โมดูล WROOM-02 (EFDV455) ชิพ ESP8266 Wi-Fi SoC จากทาง Espressif Systems ไมโครคอนโทรลเลอร์ขนาด 32-bit แบบประหยัดพลังงาน ความเร็ว 80MHz หน่วยความจำแบบแฟลชเมมโมรี่ 4MB รองรับการเชื่อมต่อเครือข่ายมาตรฐาน IEEE 802.11 b/g/n ความถี่ 2.4GHz พร้อม TCP/IP Stack มีอินเตอร์เฟส ได้แก่ GPIO SDIO PWM ADC HSPI UART I2C I2S สามารถพัฒนาโปรแกรมบนแพลตฟอร์ม Arduino ได้ โดยติดตั้ง Board Support Package ของ ESP8266/Arduino เพิ่มเติมลงในArduino IDE บอร์ด ESPino มาพร้อมกับวงจร USB-to-Serial ชิพ CP2104 สามารถเสียบเข้ากับคอมพิวเตอร์และติดตั้งไดร์ฟเวอร์ แล้วสามารถโปรแกรม ESP8266 ได้ผ่านพอร์ต USB ได้โดยไม่ต้องเชื่อมต่อเครื่องโปรแกรมเพิ่มเติม พร้อมส่วนของวงจรอัพโหลดอัตโนมัติ ไม่ต้องกดปุ่ม Program และ Reset บนบอร์ดในตอนอัพโหลด

ThaiEasyElec (2560 :60) กล่าวว่า บอร์ด ESPino เหมาะสำหรับการนำไปพัฒนางานด้าน Internet of Things (IoT) เป็นอุปกรณ์ปลายทางเพื่ออ่านค่าจากเซ็นเซอร์หรือส่งค่าควบคุมไปยังอุปกรณ์ผ่าน Wi-Fi สามารถนำไปใช้ได้ตั้งแต่การเรียนรู้และการใช้งานจริง เหมาะกับการนำไปประยุกต์ใช้ในการพัฒนาระบบต่างๆ ยกตัวอย่างเช่น

- Home Automation and Home Security
- Smart Lights and Plugs
- Home Appliances
- Sensors and Detectors
- Security Alarms
- Remote Controls and Toys
- Monitors and Scales
- Mesh Network
- Industrial Wireless Control
- Wearable Electronics
- Wi-Fi Location-Aware Devices
- Security ID Tags
- Wi-Fi Position System Beacons

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.2.3 คุณสมบัติของตัวโมดูล WROOM-02 (ส่วน WirelessModule )

EspressifSystems (2560 : 27) กล่าวถึงคุณสมบัติของตัวโมดูล WROOM-02

- รองรับมาตรฐาน IEEE 802.11 b/g/n ความถี่ 2.4 GHz

- มีไมโครคอนโทรลเลอร์ 32 บิต แบบประหยัดพลังงาน ความเร็ว 80/160 MHz พร้อมหน่วย ความจำแฟลชเมมโมรี่ขนาด 4MB อยู่ภายใน

- มีวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล (ADC) ความละเอียด 10 บิตอยู่ภายใน แรงดันสูงสุด 1 โวลต์

- มีส่วน RF Switch, Balun, LNA, PA, Matching Network, PLL, Regulators และ PMU อยู่ภายใน

- มีอินเตอร์เฟสหลากหลาย ได้แก่

o GPIO ขา 0 1 2 3 4 5 12 13 14 15 16

o UART ขา 1(Tx) 3(Rx) 15(Tx2) 13(Rx2)

o SPI (Software) ขา 15(SS) 14(SCK) 13(MOSI) 12(MISO)

o I2C (Software) ขา 4(SDA) 5(SCL)

o ADC/TOUT (10-bit, 0 – 1 Vdc)

- รองรับการทำ Antenna Diversity

- ใช้พลังงานในโหมด Deep Sleep น้อยกว่า 10 uA

- กินกระแสเมื่อ Power Down น้อยกว่า 5 uA

- ใช้พลังงาน Standby น้อยกว่า 1.0 mW (DTIM3)

- Wake Up แล้วเริ่มรับ-ส่งข้อมูลใช้เวลาต่ำกว่า 2ms

- กำลังส่ง +20 dBm ในโหมด 802.11 b

- อุณหภูมิทำงานในช่วง -40 – 125 องศาเซลเซียส

- โมดูลได้รับมาตรฐาน FCC CE TELEC

- รองรับการทำงานโหมด Station SoftAP และ Station + SoftAP

- รองรับระบบรักษาความปลอดภัย WPA และ WPA2

- รองรับโปรโตคอล TCP และ UDP

- รองรับการทำ Wi-Fi Protected Setup (WPS)

- รองรับการทำ Mesh Networking

- รองรับการทำ Smart Link ร่วมกับ Android และ iOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.3.1.2.4 คุณสมบัติของบอร์ด ESPino**

ThaiEasyElec (2560 : 52) ได้กล่าวถึงคุณสมบัติของบอร์ด ESPino

- ใช้โมดูล WROOM-02 ใช้ชิพ ESP8266 Wi-Fi SoC จาก Espressif Systems

- มีวงจร USB-to-UART ใช้ชิพ CP2104 จาก Silicon Labs สำหรับโปรแกรมและสื่อสารผ่านพอร์ตอนุกรมของบอร์ด

- ใช้ไฟเลี้ยงผ่านพอร์ต Micro USB พร้อมวงจร Regulator เพื่อเป็นแหล่งจ่ายบนบอร์ด

- สามารถเขียนโปรแกรมและอัพโหลดผ่าน Arduino IDE โดยใช้ Board Support Package ของ ESP8266/Arduino

- มีสวิตช์ PROG สำหรับโปรแกรมตัวบอร์ด

- มีสวิตช์RESET สำหรับรีเซ็ตบอร์ด

- มีวงจร Auto Program สามารถอัพโหลดโปรแกรมผ่าน Arduino IDE ได้โดยไม่ต้องกดสวิตช์

- มีหลอด LED ต่อกับ GPIO สำหรับผู้ใช้สั่งแสดงสถานะตามต้องการต่อกับขา GPIO16

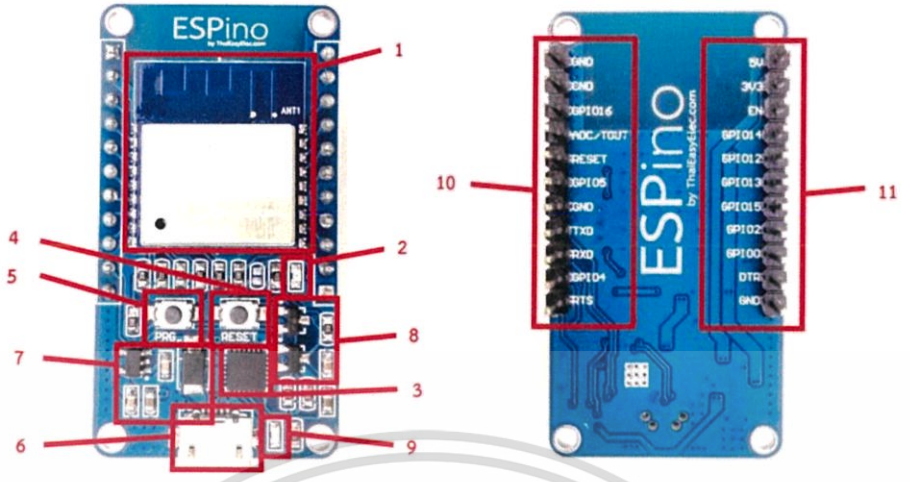
- มีหลอด LED แสดงสถานะขณะรับ-ส่งข้อมูลผ่านพอร์ตอนุกรมหรือโปรแกรมตัวบอร์ด

- คอนเนคเตอร์ตัวผู้แถวเดี่ยว 11 ขา จำนวน 2 แถว สามารถเสียบลงบนบอร์ดทดลองได้เป็นขาเชื่อมต่อต่างๆ ได้แก่ 5V 3.3V GND EN RESET GPIO0 GPIO2 GPIO4 GPIO5 GPIO12 GPIO13 GPIO14 GPIO15 GPIO16 ADC/TOUT TXD RXD RTS DTR

- ขนาดบอร์ด กว้าง 25.55 มม., ยาว 47.70 มม., สูง 14 มม. รวมเฮดเดอร์ (4 มม. ไม่รวมเฮด เดอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.2.5 ส่วนประกอบของบอร์ด ESPino by ThaiEasyElec.com




**รูปที่ 2.7** ส่วนประกอบของบอร์ด ESPino

หมายเลข 1 โมดูล WROOM-02 (Wi-Fi SoC)

หมายเลข 2 หลอด LED ขา GPIO16

หมายเลข 3 ชิพ CP2104 (USB-to-Serial)

หมายเลข 4 สวิตช์ RESET

หมายเลข 5 สวิตช์ PROGRAM

หมายเลข 6 พอร์ต Micro USB

หมายเลข 7 วงจร Voltage Regulator

หมายเลข 8 วงจร Auto Program

หมายเลข 9 หลอด LED สถานะ UART

หมายเลข 10 คอนเนคเตอร์ P1

หมายเลข 11 คอนเนคเตอร์ P2

**ตารางที่ 2.1** ขาของ ESPino

| P1 (ซ้าย) | ขาสัญญาณ | P2 (ขวา) | ขาสัญญาณ |
|---|---|---|---|
| GND | Ground | 5V | 5 Volts |
| GND | Ground | 3V3 | 3.3 Volts |
| GPIO16 | GPIO16 | EN | Enable |
| ADC/TOUT | ADC | GPIO14 | GPIO14 / SCK |
| RESET | Reset | GPIO12 | GPIO12 / MISO |
| GPIO5 | GPIO5 / SCL | GPIO13 | GPIO13 / MOSI / Rx2 |
| GND | Ground | GPIO15 | GPIO15 / SS / Tx2 |
| TXD | Tx / GPIO1 | GPIO2 | GPIO2 |
| RXD | Rx / GPIO3 | GPIO0 | GPIO0 |
| GPIO4 | GPIO4 / SDA | DTR | DTR |
| RTS | RTS | GND | Ground |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.3.1.2.6 เตรียมการเขียนโปรแกรม ESPino ด้วย Arduino IDE

1) เสียบสายฝั่ง Micro USB เข้ากับ ESPino แล้วเสียบสาย USB เข้ากับเครื่องคอมพิวเตอร์

2) หากเครื่องที่ใช้ยังไม่มีไดรฟเวอร์ของ CP2104 สามารถดาวน์โหลดได้จากเว็บไซต์ของที่ https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx เลือกตามระบบปฏิบัติการที่ใช้

3) เมื่อติดตั้งเรียบร้อยจะแสดงอุปกรณ์ใน Computer Management > Device Manager โดยจะเป็น COM Port ต่างๆ ยกตัวอย่างเป็น COM104

4) ดาวน์โหลด Arduino IDE จากเว็บไซต์ Arduino.cc ที่ https://www.arduino.cc/en/Main/Software และเลือกติดตั้งตาม ระบบปฏิบัติการที่ใช้ Windows : (เลือกแบบไฟล์ ZIP แตกไฟล์แล้วรันใช้งานได้ทันที)

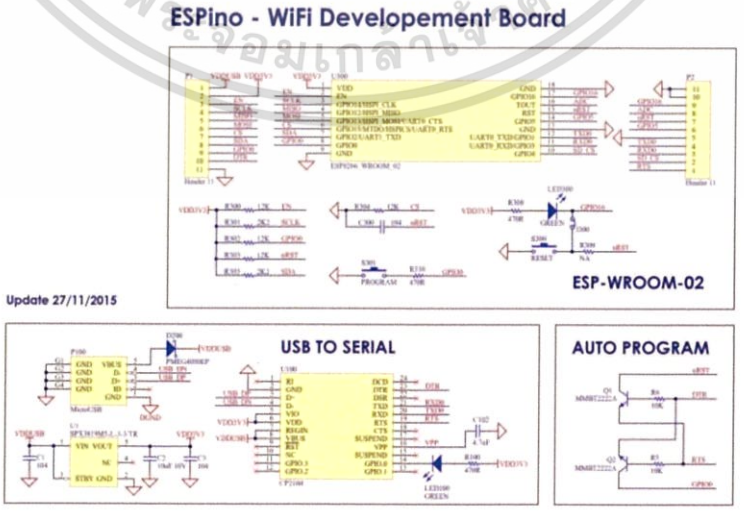
5) เปิดโปรแกรม Arduino IDE แล้วเลือกเมนู File > Preference

6) ในหน้าต่าง Preferences ในช่อง Additional Boards Manager URLs ให้เพิ่ม Package สำหรับ ESP8266 ที่ http://arduino.esp8266.com/stable/package_esp8266com_index.json แล้วกดปุ่ม OK ออกจากหน้าต่าง Preference

7) โปรแกรมจะดาวน์โหลดลิสต์ เสร็จแล้วเลื่อนลงมาที่ esp8266 by ESP8266 Community เลือกเวอร์ชั่น 2.0.0 จากเมนูแล้วกดปุ่ม Install เพื่อดาวน์โหลด Board Support Package ของ ESP8266

8) เมื่อติดตั้งเรียบร้อยแล้วจะแสดงข้อความ Version 2.0.0 และ INSTALLED ดังภาพ กดปุ่ม Close ปิดหน้าต่าง Boards Manage

#### 2.3.1.2.7 ผังวงจรบอร์ด ESPino




**รูปที่ 2.8** ผังวงจรบอร์ด ESPino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2 ภาษาที่ใช้เขียน โปรแกรมควบคุมไมโครคอนโทลเลอร์

- ภาษา Assembly
- ภาษา Basic
- ภาษา C
- ภาษา Pascal

ภาษาดังกล่าวที่กล่าวในเบื้องต้น ไมโครคอนโทรลเลอร์บางตระกูลจะใช้ได้ครบทุกภาษา และสำหรับบางตระกูลจะใช้ได้บางภาษา ขึ้นอยู่กับบริษัทผู้ผลิต Software ที่ใช้เขียนภาษาไมโครคอน-โทรลเลอร์นั้นจะผลิตออกมาให้ Support หรือไม่

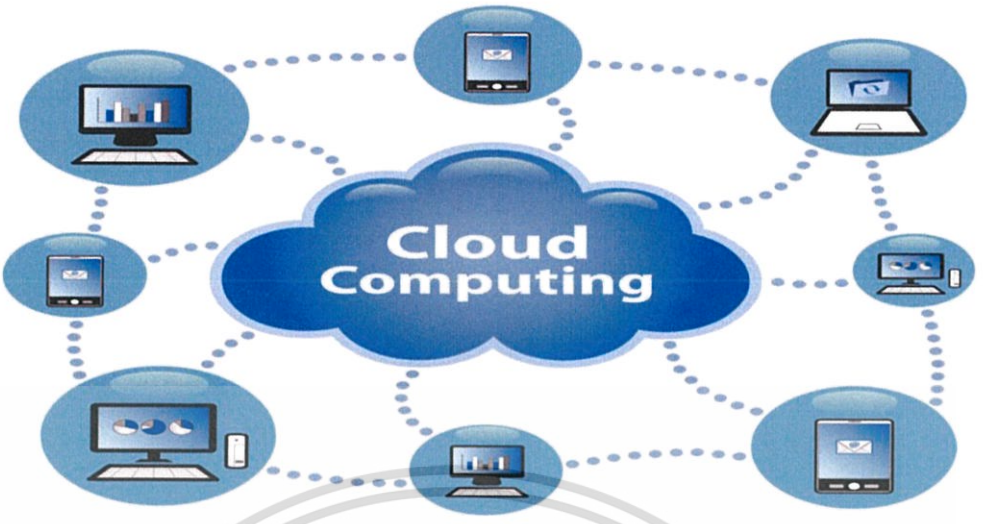
## 2.4 Cloud และ Protocol ที่ใช้ในการสื่อสารระหว่าง Controller กับ User

### 2.4.1 ความหมายของ Cloud

มาดิเซ็ตติ (Vijay Madisetti, 2559 : 41) กล่าวว่า Cloud คือ ระบบหนึ่งระบบ ที่รวมเอาเครื่องคอมพิวเตอร์จำนวนมากเข้าด้วยกัน แต่สามารถจัดสรรทรัพยากรเพื่อให้ เพียงพอต่อการใช้งานได้ ไม่ว่าจะเป็น CPU, Ram และ Storage โดยเราสามารถสร้าง Instance ขึ้นมาแล้วเลือกความต้องการของทรัพยากรระบบที่เราต้องการใช้งานเพื่อเป็น Server ได้ เหตุผลที่เรียก Cloud นั่นเพราะว่าให้มองระบบทั้งหมด เป็นเสมือนกลุ่มเมฆกลุ่มหนึ่ง ที่เราไม่สามารถทราบได้ว่าทรัพยากรภายในกลุ่มเมฆนั้นมีอะไรอยู่บ้าง Cloud นับว่าเป็น ทางเลือกใหม่สำหรับการใช้งานบริการ Server ที่ไม่ต้องลงทุนอุปกรณ์เอง สามารถดูแลรักษาได้ง่าย มีการรับรองว่าระบบว่าไม่ล่มง่ายๆ อีกทั้งยังสามารถเพิ่มทรัพยากรที่เราต้องการได้ในทันที เช่น ผู้ใช้Amazon AWS เลือก Package ที่เป็น CPU Dual Core, Ram 2 GB, HDD 20 GB ไว้ อยู่มาวันหนึ่งผู้ใช้จำเป็นต้องใช้ทรัพยากรมากกว่านี้เพื่อรองรับผู้ใช้งานที่มากขึ้น ผู้ใช้ สามารถเปลี่ยนความต้องการของทรัพยากรได้ในทันที ข้อมูลเก่าที่ใช้งานมาก่อนหน้านี้ก็สามารถใช้งานต่อไปได้โดยไม่ต้อง ติดตั้งระบบใหม่

บาร์ก้า (Arshdeep Bahga, 2558 : 180) กล่าวว่า Cloud คือ ระบบที่สามารถเก็บข้อมูลไว้บน Server (ผู้ให้บริการ) ข้อดีคือไม่ต้องกังวลกับข้อมูลสูญหายและสามารถเข้าถึงข้อมูลได้จากทุกที่เพียงเชื่อมต่อ Internet ทั้งนี้ ระบบCloud มีทั้งแบบใช้งานฟรีและเสียค่าบริการตามแพ็กเกจปริมาณข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.9 cloud computing

### 2.4.2 Protocol ที่ใช้ในการสื่อสารระหว่าง Controller กับ User

#### 2.4.2.1 ความหมายของโปรโตคอล (Protocol )

คูโรเซ่ (Jim Kurose, 2541 : 59) กล่าวว่า โปรโตคอล (Protocol) คือ ข้อกำหนดหรือข้อตกลงในการสื่อสารระหว่างคอม- พิวเตอร์ หรือภาษาสื่อสารที่ใช้เป็น ภาษากลางในการสื่อสารระหว่างคอมพิวเตอร์ด้วยกัน การที่เครื่องคอมพิวเตอร์ที่ถูกเชื่อมโยงกันไว้ในระบบจะสามารถติดต่อสื่อสารกันได้นั้น จำเป็นจะต้องมีการสื่อสารที่เรียกว่า โปรโตคอล(Protocol) เช่นเดียวกับคนเราที่ต้องมีภาษาพูดเพื่อให้สื่อสารเข้าใจกันได้

รอสส์ (Keith Ross, 2549 : 71) กล่าวว่า โปรโตคอลช่วยให้ระบบคอมพิวเตอร์สองระบบ ที่แตกต่างกันสามารถสื่อสารกันอย่างเข้าใจได้ คือข้อตกลงที่กำหนดเกี่ยว กับการสื่อสารระหว่างเครื่องคอมพิวเตอร์ต่างๆ ทั้งวิธีการส่งและรับข้อมูล วิธีการตรวจสอบข้อผิดพลาดของการส่งและรับข้อมูล การแสดงผลข้อมูลเมื่อส่งและรับกันระหว่างเครื่องสองเครื่อง ดังนั้นจะเห็นได้ว่าโปรโตคอลมีความสำคัญมากในการสื่อสารบนเครือ-ข่าย หากไม่มีโปรโตคอลแล้ว การสื่อสารบนเครือข่ายจะไม่สามารถเกิดขึ้นได้

รอสส์ (Keith Ross, 2549 : 80) กล่าวว่า การที่เครื่องคอมพิวเตอร์ที่ถูกเชื่อมโยงกันไว้ในระบบ จะสามารถติดต่อสื่อสารกันได้นั้น จำเป็นจะต้องมีภาษาสื่อสารที่เรียกว่า โปรโตคอล (Protocol ) ซึ่งในระบบ Internet จะใช้ภาษาสื่อสารมาตรฐานที่ชื่อว่า TCP/IP เป็นภาษาหลัก ดังนั้นหากเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นเครื่องระดับไมโครคอมพิวเตอร์ มินิคอมพิวเตอร์ หรือเมนเฟรมคอมพิวเตอร์ ก็สามารถเชื่อมโยงเข้าสู่อินเทอร์เน็ตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP ย่อมาจากคำว่า Transmission Control Protocol

IP ย่อมาจากคำว่า Internet Protocol

สุวัฒน์ ปุณณชัยยะ (2560 :21) กล่าวว่า TCP/IP คือ ชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ตโดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ

สุพจน์ ปุณณชัยยะ (2560 : 29) ได้สรุปว่า TCP และ IP มีหน้าที่ต่างกัน คือ

1) TCP จะทำหน้าที่ในการแยกข้อมูลเป็นส่วน ๆ หรือที่เรียกว่า Package ส่งออกไป ส่วน TCP ปลายทาง ก็จะทำการรวบรวมข้อมูลแต่ละส่วนเข้าด้วยกัน เพื่อนำไปประมวลผลต่อไป โดยระหว่างการรับส่งข้อมูลนั้นก็จะมีการตรวจสอบความถูกต้องของ ข้อมูลด้วย ถ้าเกิดผิดพลาด TCP ปลายทางก็จะขอไปยัง TCP ต้นทางให้ส่งข้อมูลมาใหม่

2) IP จะทำหน้าที่ในการจัดส่งข้อมูลจากเครื่องต้นทางไปยังเครื่องปลายทาง โดยอาศัย IP Address

ตัน ตัณฑ์สุทธิวงศ์ (2560 : 82) ได้ตัวอย่างของโปรโตคอล

1) โปรโตคอล HTTP หรือ Hypertext Transfer Protocol จะใช้เมื่อเรียกโปรแกรมบราวเซอร์ (Browser)

2) โปรโตคอล TCP/IP หรือ Transfer Control Protocol/Internet Protocolคือเครือข่ายโปรโตคอลที่สำคัญมากที่สุด เนื่องจากเป็นโปรโตคอลที่ใช้ในระบบเครือข่าย Internet รวมทั้ง Intranet ซึ่งประกอบด้วย 2 โปรโตคอลคือ TCP และ IP

3) โปรโตคอล SMTP หรือ Simple Mail Transfer Protocol คือ โปรโตคอลที่ใช้ในการรับส่งจดหมายอิเล็กทรอนิกส์บนเครือข่ายอินเทอร์เน็ต

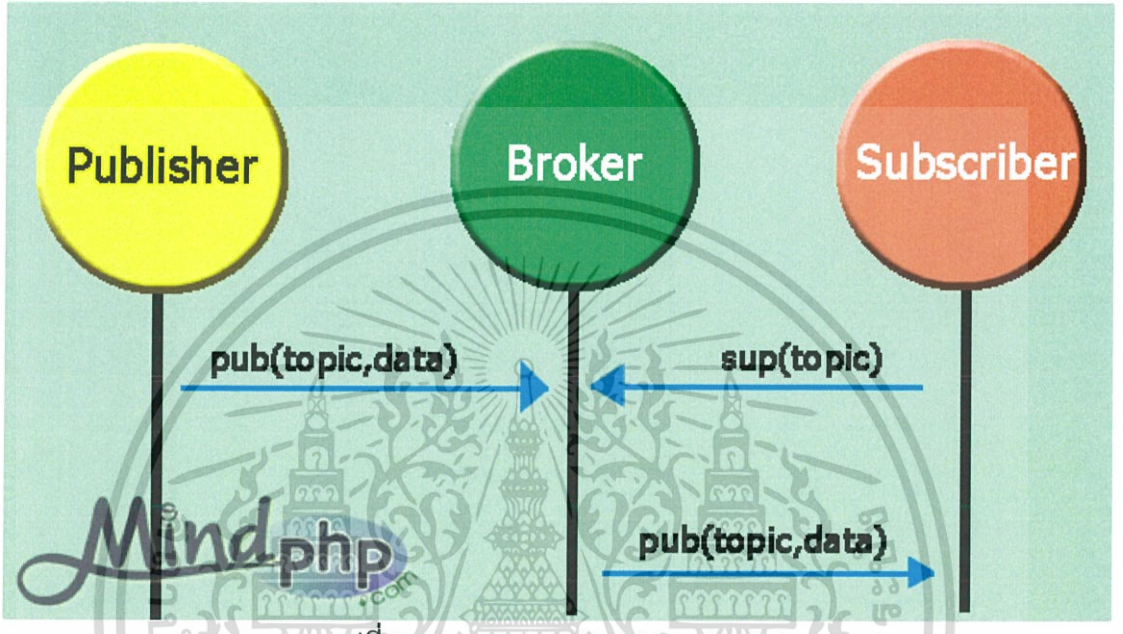
นอกจากโปรโตคอลที่กล่าวมาข้างต้นแล้ว ยังมีโปรโตคอลต่างๆอีกมากมาย เช่น การโอนย้ายแฟ้มระหว่างกัน ใช้โปรโตคอลชื่อ FTP หรือ File Transfer Protocol การโอนย้ายข่าวสารระหว่างกันก็ใช้โปรโตคอลชื่อ NNP หรือ Network News Transfer Protocol และยังมีโปรโตคอลที่สำคัญสำหรับการสอบถามข้อมูลข่าวสารระหว่างกัน ซึ่งเป็นโปรโตคอลที่มีประโยชน์มาก โปรโตคอลนี้มีชอว่า ICMP หรือ Internet Control Message Protocol เป็นต้น

### 2.4.2.2 ความหมายของ MQTT Protocol

ปวเรศ บวรภัทรวดี (2560 : 76) ได้กล่าวว่า MQTT คือ Message Queuing Telemetry Transport (MQTT) เป็น Protocol ที่ออกแบบมาเพื่อการเชื่อมต่อแบบ M2M (Machine-to-Machine) คืออุปกรณ์กับอุปกรณ์ สนับสนุนเทคโนโลยี iot (Internet of Things) คือเทคโนโลยีที่อินเทอร์เน็ตเชื่อมต่อกับอุปกรณ์ต่าง ๆ เช่น โทรศัพท์มือถือ รถยนต์ โทรทัศน์ ตู้เย็น เข้ากับอินเทอร์เน็ตทำให้สามารถเชื่อมโยงสื่อสารกับอุปกรณ์ต่างๆได้ โดยผ่านเครือข่ายอินเทอร์เน็ต ซึ่งจะทำให้มนุษย์สามารถควบคุมอุปกรณ์ต่างๆ จากที่อื่นได้ เช่นการสั่งปิดเปิดไฟในบ้านจากที่อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากโปรโตคอลตัวนี้มีน้ำหนักเบาออกแบบมาเพื่อใช้งานกับอุปกรณ์อิเล็ก-ทรอนิกส์ขนาดเล็ก การรับส่งข้อมูลในเครือข่ายที่มีขนาดเล็ก แบนร์วิธต่ำ ใช้หลักการแบบ Publisher / Subscriber คล้ายกับหลักการที่ใช้ใน Web Service ที่ต้องใช้ Web Server เป็นตัวกลางระหว่างคอมพิวเตอร์ของผู้ใช้ แต่ MQTT จะใช้ตัวกลางที่เรียกว่า Broker เพื่อทำหน้าที่จัดการคิว รับ-ส่ง ข้อมูลระหว่างอุปกรณ์และทั้งในส่วนที่เป็น Publisher และ Subscriber ดังภาพ




รูปที่ 2.10 แสดงการ Publisher MQTT

จากรูปที่ 2.10 จะเห็นได้ว่า Topic จะเป็นตัวอ้างอิงหลักข้อมูลที่จะ Publisher ออกไปยัง Broker จะต้องมี Topic กำกับไว้เสมอทางฝ่าย Subscriber ก็จะอ้างถึง Topic เพื่อเรียกข้อมูลที่ต้องการเหมือนกับการสมัครเป็นสมาชิกของหนังสือพิมพ์ฉบับหนึ่งชื่อของหนังสือก็เปรียบเหมือน Topic และผู้ผลิตก็คือ Publisher เมื่อถึงเวลาที่หนังสือเสร็จ ผู้ส่ง Broker ก็จะนำหนังสือพิมพ์มาส่งให้เรา ตัวอย่าง แอพพลิเคชั่นที่ใช้งานคือ Facebook Messenger

สนธยา นงนุช (2560 : 162) ได้สรุปองค์ประกอบของ MQTT Protocol ว่าจะประกอบไปด้วย Broker , Publisher และ Subscriber แต่ล่ะอย่างก็จะทำหน้าที่แต่กต่างกันออกไปโดย Broker ทำหน้าที่เป็นตัวกลางคอยจัดการกับ ข้อความโดย อ้างอิงจาก Topic Publisher จะทำหน้าที่คอยส่งข้อมูลไปยังหัวข้อนั้นๆ Subscriber จำทำหน้าที่คอยดูการเปลี่ยนแปลงของ Message ที่อ้างอิงด้วย Topic เช่นถ้ามีหัวข้อหน้าสนใจและมีการเปลี่ยนแปลงก็จะทำการดึงข้อมูลนั้นๆ มาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 ภาษาที่ใช้ในการเขียน Web site รับคำสั่งควบคุม

### 2.5.1 ความหมายของ JavaScript

จตุรพัชร์ พัฒนทรงศิวิไล (2548 : 12) กล่าวว่า JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (Script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช่ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (Interpret) หรือเรียกว่า อ็อบเจ็กโอเรียลเต็ด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และทางฝั่งเซิร์ฟเวอร์ (Server)

### 2.5.2 ความเป็นมาของ JavaScript

สมบูรณ์ พัฒน์ธีรพงศ์ (2546 : 24) ได้อธิบายว่า JavaScript ถูกพัฒนาขึ้นโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปจึงได้ร่วมมือกับบริษัทซันไมโครซิส-เต็มส์ ทำการปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี 2538 แล้วตั้งชื่อใหม่ว่า JavaScript JavaScript สามารถทำให้การสร้างเว็บเพจมีลูกเล่นต่าง ๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือ การกรอกข้อความในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ (เรียกว่าเป็น Client-side Script) ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ดี สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชั่นใหม่ๆออกมาด้วย (ปัจจุบันคือรุ่น 1.5) ดังนั้น ถ้านำโค้ดของเวอร์ชั่นใหม่ ไปรันบนบราวเซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิด Error ได้

### 2.5.3 สิ่งที่ JavaScript สามารถทำได้

1) JavaScript ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายๆได้ โดยไม่ต้องพึ่งภาษาอื่น

2) JavaScript มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่นเมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox ก็สามารถสั่งให้เปิดหน้าใหม่ได้ ทำให้เว็บไซต์ของเรามีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) JavaScript สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้ นั่นคือสามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้ หรือหน้าแสดงเนื้อหาสามารถซ่อนหรือ แสดงเนื้อหาได้แบบง่ายๆนั่นเอง

4) JavaScript สามารถใช้ตรวจสอบข้อมูลได้ สังเกตว่าเมื่อเรากรอกข้อมูลบางเว็บไซต์เช่น Email เมื่อเรากรอกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่าเรากรอกผิด หรือลืมกรอกอะไรบางอย่าง เป็นต้น

5) JavaScript สามารถใช้ในการตรวจสอบผู้ใช้ได้ เช่น ตรวจสอบว่าผู้ใช้ ใช้ Web Browser อะไร

6) JavaScript สร้าง Cookies (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

### 2.5.4 ข้อดีและข้อเสียของ JavaScript

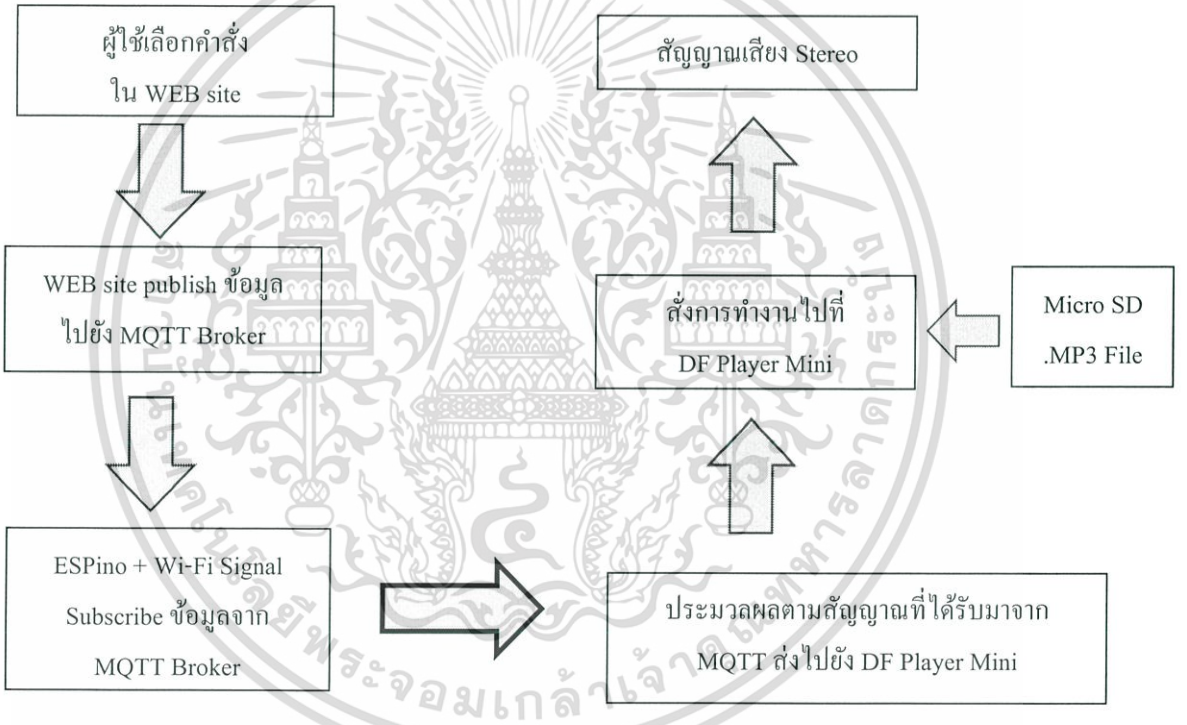
จตุรพัชร์ พัฒนทรงศิวิไล (2548 : 124) ได้สรุปว่า การทำงานของ JavaScript เกิดขึ้นบนบราวเซอร์ (เรียกว่าเป็น Client-side Script) ดังนั้นไม่ว่าคุณจะใช้เซิร์ฟเวอร์อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (เรียกว่า Server-side Script) ดังนั้นจึงต้องใช้บนเซิร์ฟเวอร์ ที่สนับสนุนภาษาเหล่านี้เท่านั้น อย่างไรก็ดี จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา Server-side Script อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# วิธีการการดำเนินการ

การออกแบบและพัฒนาเครื่องควบคุมและสั่งการเล่นมัลติมีเดียระยะไกลผ่านอินเตอร์เน็ตด้วยไมโครคอนโทรลเลอร์ จะนำทฤษฎีจากบทที่ 2 มาใช้สำหรับออกแบบและพัฒนาระบบ โดยเนื้อหาในบทนี้จะกล่าวถึงภาพรวมของระบบและอธิบายรายละเอียดต่างๆในการออกแบบและพัฒนาระบบ

## 3.1 องค์ประกอบและขั้นตอนการทำงานโดยรวมของระบบ




**รูปที่ 3.1** บล็อกไดอะแกรมแสดงขั้นตอนการทำงานโดยรวมของระบบ

จากรูปที่ 3.1 เครื่องควบคุมและสั่งการเล่นมัลติมีเดียระยะไกลผ่านอินเตอร์เน็ตด้วยไมโครคอนโทรลเลอร์ ประกอบไปด้วยส่วนสำคัญ 2 ส่วน ได้แก่ 1. ส่วนเล่นไฟล์ MP3 และส่งสัญญาณเสียง 2. ส่วนเชื่อมต่อระบบอินเตอร์เน็ตเพื่อควบคุมและส่วนประมวลผล ซึ่งในบทนี้จะแสดงให้เห็นภาพรวมของการทำงานขององค์ประกอบทั้งสองโดยเริ่มจากการทำงานของส่วนเชื่อมต่ออินเตอร์เน็ต ที่จะรอรับสัญญาณที่จะนำมาควบคุมการเล่นไฟล์ MP3 ในส่วนเล่นไฟล์

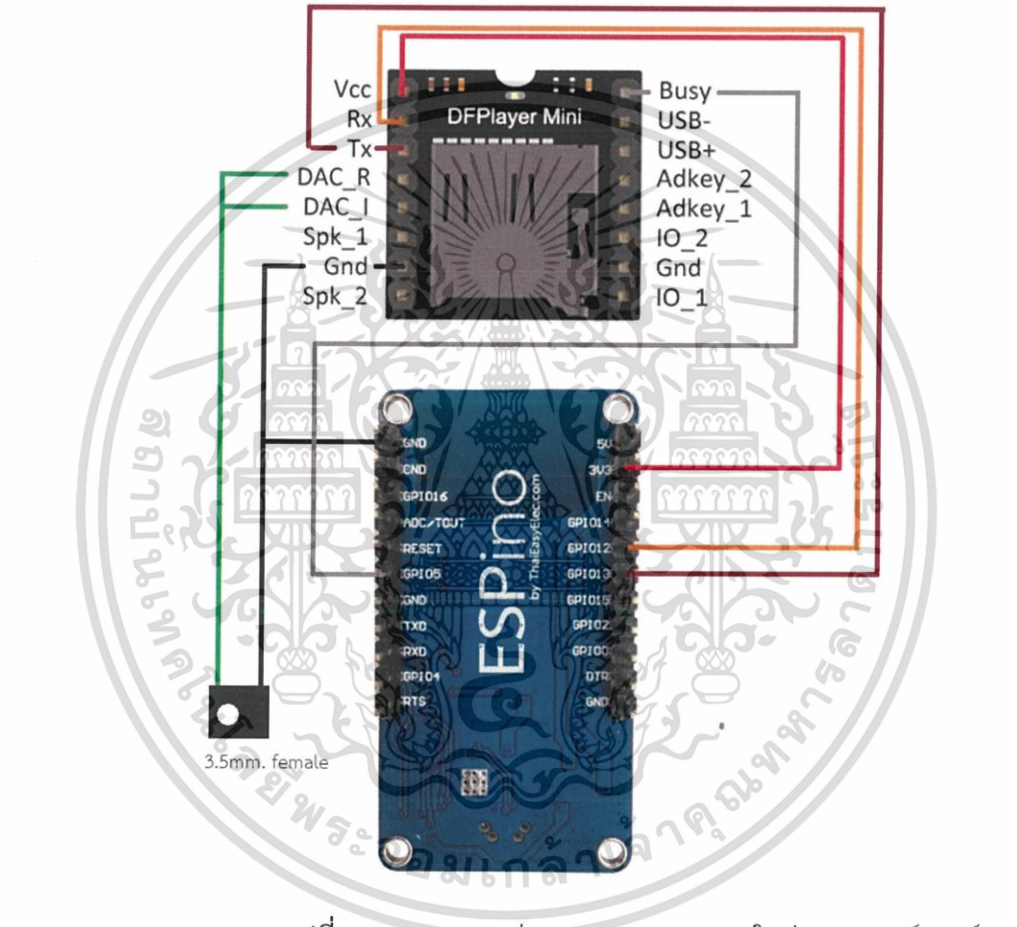
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2 รายละเอียดเครื่องมือที่ใช้ในการออกแบบส่วนต่างๆของระบบ

ในการออกแบบระบบโดยรวมรายละเอียดของการออกแบบทั้งหมดแบ่งออกเป็นออก 2 ส่วนได้แก่ 1. ส่วนของฮาร์ดแวร์ที่ใช้รับสัญญาณอินเตอร์เน็ตและประมวลผล 2. ส่วนของโปรแกรมซอฟต์แวร์

### 3.2.1 การออกแบบส่วนฮาร์ดแวร์

การออกแบบ เครื่องมือที่ใช้และการต่อวงจรของระบบในส่วนของฮาร์ดแวร์




**รูปที่ 3.2** แสดงการต่อวงจรรวมของระบบในส่วนของฮาร์ดแวร์

จากรูปที่ 3.2 ใช้ ESPino เป็นอุปกรณ์ควบคุมและรับสัญญาณเพื่อความเหมาะสมในการออกแบบระบบและ DF Player Mini เป็น Decoder Module ใช้กับไฟล์ .mp3 โดยจะมีการทำงานเป็น 3 ส่วนดังนี้

1. ส่วนรับสัญญาณคำสั่งเพื่อนำมาประมวลผล (ESPino + Wi-Fi Signal)
2. ส่วนที่ควบคุมการเล่นเพลงของ (ESPino + DF Player Mini)
3. ส่วนแสดงผลเป็นภาคจ่ายสัญญาณเสียง Stereo ด้วย Jack 3.5 mm. (DF Player Mini + Jack)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยใช้ ESPino เป็น Microcontroller เนื่องจากแพลตฟอร์ม ESPino มีข้อได้เปรียบดังนี้

1. ขนาดเล็กกว่า Arduino Uno
2. ทำงานที่ 32 bit ซึ่ง Arduino Uno ทำงานที่ 8 bit
3. Memory มากกว่า Arduino Uno โดยมีถึง 4MB จึงสามารถรองรับโปรแกรมได้มากกว่า
4. ติดตั้ง Wi-Fi Module มาแล้ว
5. สามารถใช้ Arduino IDE ในการเขียนได้ และรองรับ Library ใกล้เคียงกับ Arduino

และ ใช้ DF Player Mini เป็น MP3 Decoder เนื่องจากมีข้อได้เปรียบดังนี้

1. สามารถให้สัญญาณเสียงได้ทั้งแบบ Mono และ Stereo
2. สามารถใช้ชุดคำสั่งสั่งการทำงานได้โดยตรงจึงสามารถควบคุมได้ผ่าน ESPino
3. ติดตั้ง Micro SD Card แบบ Built-in ไม่ต้องใช้ Module Micro SD Card

**3.2.1.1 การทำงานในส่วนรับสัญญาณคำสั่งเพื่อนำมาประมวลผล**

การรับสัญญาณนั้นได้เลือกใช้สัญญาณ Wi-Fi ในการรับเนื่องจากไม่ต้องใช้สายในการเชื่อมต่ออีกทั้งยังไม่ต้องเพิ่ม Module เนื่องจาก ESPino built-in esp8266 มาแล้ว ไม่ต้องดัดแปลงทำปุ่ม Reset และ Program เอง

**รูปที่ 3.3** บอร์ด ESPino

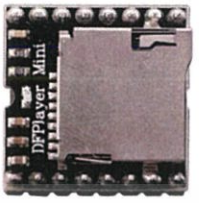
ส่วนที่ควบคุมการเล่นเพลง

ใช้ ESPino เชื่อมต่อกับพอร์ตต่างๆของ DF Player Mini ที่มี Micro SD Card ซึ่งมีไฟล์ MP3 อยู่แล้ว ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนแสดงผลเป็นภาคจ่ายสัญญาณเสียง Stereo ด้วย Jack 3.5mm.

เนื่องจาก DF Player Mini มีภาคการส่งสัญญาณเสียงแบบ Stereo มาอยู่แล้วจึงมีการทำ Jack Female 3.5 mm. ให้สามารถนำ Output ไปเชื่อมต่อกับลำโพงได้โดยตรง




**รูปที่ 3.4** DF Player Mini

### 3.2.2 การออกแบบส่วนซอฟต์แวร์

การออกแบบโปรแกรมของการทำงานของระบบแบ่งออกเป็น 4 ส่วนหลัก ซึ่งเป็นผลมากจากการทำงานของฮาร์ดแวร์ดังนี้

#### 3.2.2.1 การเชื่อมต่อ Wi-Fi และการรับค่าจากจาก MQTT Broker เป็นคำสั่งดังนี้

```
#include <ESP8266WiFi.h>

// การเชื่อมต่อ Wi-Fi
const char* ssid = "ssid";
const char* password = "11111111";

// ใช้ในการสื่อสารระหว่าง Controller กับ MQTT Server เพื่อที่จะ
Subscribe คำสั่งที่ใช้กับ DF Player Mini
#define mqtt_server "m13.cloudmqtt.com"
#define mqtt_port 10874
#define mqtt_user "Dream"
#define mqtt_password "999999"
WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
        Serial.println();
        Serial.print("Connecting to ");
        Serial.println(ssid);
        WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
      }
Serial.println("");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
 }
void loop() {
  if (!client.connected()) {
 Serial.print("Attempting MQTT connection...");
 if (client.connect("ESP8266Client", mqtt_user, mqtt_password)) {
  Serial.println("connected");
 client.subscribe("DV");
 } else {
  Serial.print("faiDV, rc=");
  Serial.print(client.state());
 Serial.println(" try again in 5 seconds");
 delay(5000);
 return;
   }
  }
  client.loop();

 // ส่วนการรับสัญญาณที่ใช้ควบคุม DF Player Mini
 }
 void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  String msg = "";
  int i=0;
  while (i<length) msg += (char)payload[i++];
  if (msg == "play") {
```

3.2.2.2 การนำเอาสัญญาณหลัง Payload มาใช้สั่งการเล่นไฟล์ MP3 กับ DF Player Mini + Micro SD Card

```
// เป็นการรับเอาคำสั่งสำหรับควบคุม DF Player Mini มาใช้โดยคำสั่งเหล่านี้จะเป็นคำสั่งพื้นฐานมากับตัว DF Player Mini
while (i<length) msg += (char)payload[i++];
 if (msg == "play") {
   mp3_play ();
   Serial.println("playing");
 } else if (msg == "stop") {
   mp3_stop ();
   Serial.println("stop");
 } else if (msg == "next") {
   mp3_next ();
   Serial.println("next");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  } else if (msg == "prev") {
    mp3_prev ();
    Serial.println("previous");
  } else if (msg == "pause") {
    mp3_pause ();
    Serial.println("paused");
  } else if (msg == "p1") {
    mp3_play (1);
    Serial.println("song1");
  } else if (msg == "p2") {
    mp3_play (2);
    Serial.println("song2");
  } else if (msg == "p3") {
    mp3_play (3);
    Serial.println("song3");
  } else if (msg == "p4") {
    mp3_play (4);
    Serial.println("song4");
  }
}
```

3.2.2.3 ส่วนของการเขียนเว็บเพื่อใช้ Publish ข้อมูลคำสั่ง

```
// การกำหนดให้ Controller กับ Website มีพอร์ตตรงกัน และให้
Username กับ Password เข้าถึงข้อมูล
var config = {
        mqtt_server: "m15.cloudmqtt.com",
        mqtt_websockets_port: 3XXXX,
        mqtt_user: "XXXX",
        mqtt_password: "XXXXXXXX"
};
// การสร้างปุ่มเพื่อสั่ง Publish คำสั่งเช่น ปุ่ม Play ปุ่ม Stop
$(document).ready(function(e){
        client = new Paho.MQTT.Client(config.mqtt_server,
config.mqtt_websockets_port, "web_" + parseInt(Math.random() * 100,
10));
        Paho.MQTT.Client("m15.cloudmqtt.com", 32903, "web_" +
parseInt(Math.random() * 100, 10));
        client.connect({
                useSSL: true,
                userName: config.mqtt_user,
                password: config.mqtt_password,
                onSuccess: function()
        $("#status").text("Connected").removeClass().addClass("connect
ed");
                        client.subscribe("/DV");
                        mqttSend("/DV", "play");
                },
                onFailure: function(e){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                $("#status").text("Error : " + e).removeClass().addClass("error");
                // console.log(e);
            }
        });
            client.onConnectionLost = function(responseObject) {
            if (responseObject.errorCode !== 0) {
                $("#status").text("onConnectionLost:" + responseObject.errorMessage).removeClass().addClass("connect");
                setTimeout(function() { client.connect() }, 1000);
            }
        }
            $("#dv-play").click(function(e) {
            mqttSend("/DV", "play");
        });
            $("#dv-stop").click(function(e) {
            mqttSend("/DV", "stop");
            });
            $("#dv-next").click(function(e) {
            mqttSend("/DV", "next");
            });
            $("#dv-pause").click(function(e) {
            mqttSend("/DV", "pause");
            });
        $("#dv-prev").click(function(e) {
            mqttSend("/DV", "prev");
        });
        $("#dv-p1").click(function(e) {
            mqttSend("/DV", "p1");
        });
        $("#dv-p2").click(function(e) {
            mqttSend("DV", "p2");
        });
        $("#dv-p3").click(function(e) {
            mqttSend("DV", "p3");
        });
        $("#dv-p4").click(function(e) {
            mqttSend("DV", "p4");
        });
```

## 3.3 รายละเอียดขั้นตอนการทำงานส่วนต่างๆของระบบ

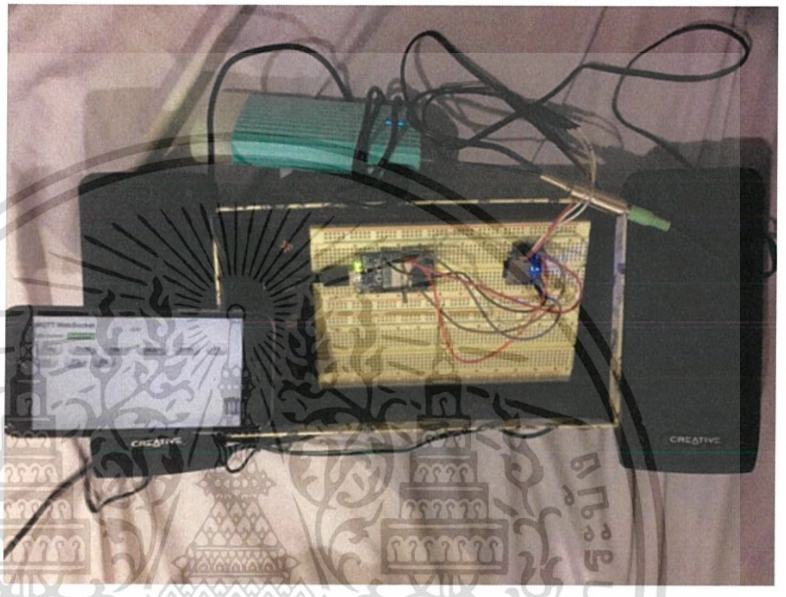
3.3.1 เริ่มต้นการทำงานเมื่อเปิดการทำงาน ESPino จะเริ่มการเชื่อมต่อสัญญาณ Wi-Fi เป็นอันดับแรก

3.3.2 เมื่อเชื่อมต่อสัญญาณ Wi-Fi ได้แล้ว ESPino จะเชื่อมต่อกับ MQTT Broker เพื่อรอ Subscribe ข้อมูลจาก MQTT Broker

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 เมื่อเจอ MQTT Broker แล้ว ESPino จะได้ Subscribe ข้อความแรกจาก MQTT Broker โดยจะเป็นคำสั่ง Play แรก

3.3.4 เมื่อมีการเลือกคำสั่งที่หน้าเว็บจะมีการ Publish ข้อมูลไปยัง MQTT เพื่อให้มีข้อความใหม่ไปแทนที่ อาจเป็นคำสั่ง Stop เพื่อใช้หยุดเพลง คำสั่ง Pause เพื่อหยุดเพลงชั่วคราว หรือคำสั่ง Next เพื่อเล่นเพลงถัดไป หรือคำสั่งอื่นๆ เมื่อได้รับมาแล้ว ESPino ที่ Subscribe ได้ก็จะสั่งการทำงานของ DF Player Mini เป็นข้อความคำสั่งใหม่ที่ได้รับมาแทน



**รูปที่ 3.5** ชิ้นงานเบื้องต้นที่แสดงการเชื่อมต่อและหลักการทำงาน

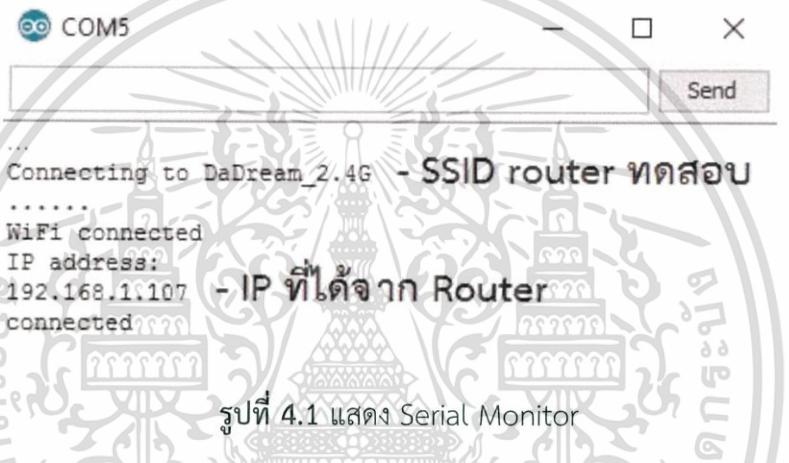
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4
# ผลการทดลอง

## 4.1 ผลการทดสอบการเชื่อมต่อ MQTT ผ่านอินเตอร์เน็ตในวงเปิด

ในการเก็บผลการทดลองจะแสดงผลการทดลองให้เห็นในรูปแบบของภาพถ่ายหน้าจอทั้งในฝั่งอุปกรณ์ และ Web Socket โดยฝั่งอุปกรณ์จะใช้ Serial Monitor

การทดลองเชื่อมต่อ Wifi กับ ESPino นี่คือผลการทดลอง กรณีรหัสถูกต้องและสามารถเชื่อมต่อได้




**รูปที่ 4.1** แสดง Serial Monitor

การทดลองเชื่อมต่อ Wifi กับ Espino นี่คือผลการทดลอง กรณีที่ไม่สามารถเชื่อมต่อได้ จะแสดง “.....” จนกว่าจะเจอ SSID และ Password ที่ตรงกัน

**รูปที่ 4.2** แสดง Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 ผลการทดสอบการสื่อสารของ MQTT กับ MQTT Broker

ในการทดสอบนี้จะใช้ MQTT Web socket สื่อสารกับอุปกรณ์ ESPino โดยจะทำการทดสอบโดยจะตั้งค่าให้ Port และ Username ของ MQTT และ Password เข้าใช้งานถูกต้อง

ฝั่งอุปกรณ์จะตั้งค่าแบบนี้

```
#define mqtt_server "m13.cloudmqtt.com"
#define mqtt_port 10874
#define mqtt_user "Dream"
#define mqtt_password "999999"
```

**รูปที่ 4.3** แสดง MQTT Web socket

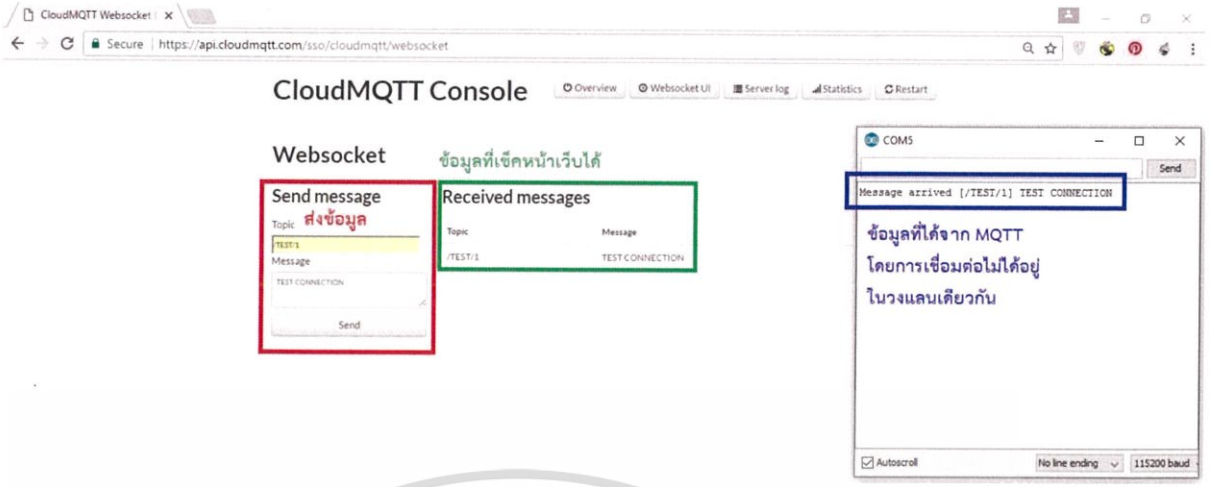
โดยตั้งให้ตรงกับที่ CloudMQTT แจกให้

```
void loop() {
 if (!client.connected()) {
   ////////////Serial.print("Attempting MQTT connection...");
   if (client.connect("ESP8266Client", mqtt_user, mqtt_password)) {
    Serial.println("connected");
   ////////////  client.subscribe("/ESP/LED");
    client.subscribe("/TEST/1");   - topic ที่ใช้ทดสอบ
```

รับส่งข้อมูลผ่าน topic “/TEST/1”

**รูปที่ 4.4** แสดง MQTT Web socket

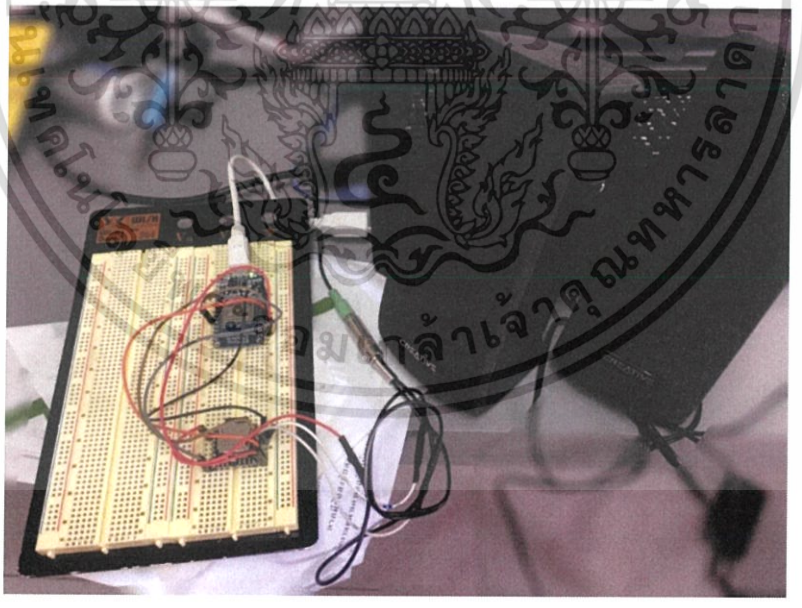
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.5** แสดงผลการทดสอบส่งข้อมูล โดยที่ ESPino รับข้อมูลจาก MQTT ได้

## 4.3 ผลการทดสอบการเล่นไฟล์เสียงบน Arduino

ในการทดสอบนี้จะเชื่อมต่อ ESPino กับ DF Player Mini และนำเพลงใส่ใน Micro SD เพื่อทดสอบการเล่นไฟล์เสียงกับคำสั่งพื้นฐาน ซึ่ง ESPino ก็สามารถทำงาร่วมกันกับ DF Player Mini ได้เป็นอย่างดี และได้เสียงออกมาเป็นที่น่าพอใจ โดยคุณภาพเสียงจะขึ้นอยู่กับลำโพงที่ใช้ด้วย



**รูปที่ 4.6** แสดงตัวอย่างชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าการแสดง Serial Monitor

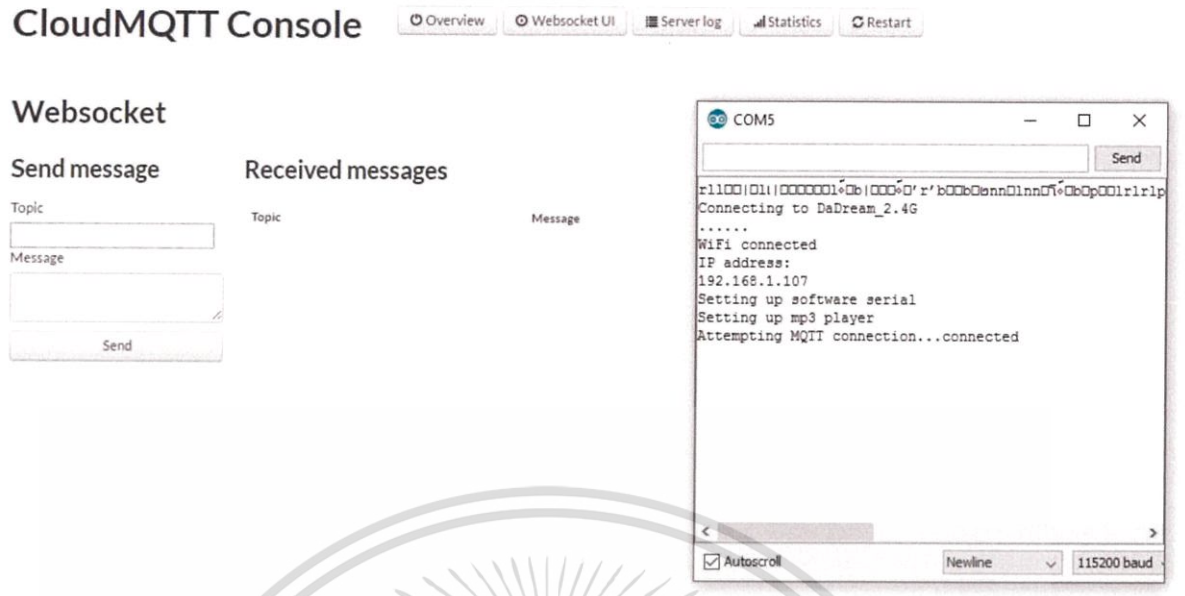
**รูปที่ 4.7** แสดงหน้า Serial Monitor

**4.4 ผลการทดสอบการควบคุมการเล่นไฟล์เสียงบน Arduino ผ่าน MQTT Broker**

ในการทดสอบนี้จะนำเอาชุดคำสั่งของ DF Player Mini มาทำงานร่วมกันกับ MQTT โดยให้ MQTT เป็นตัว Publish ข้อมูลและให้ ESPino ทำหน้าที่ Subscribe ข้อมูล ข้อมูลที่ได้จะกระตุ้นการทำงานของ ESPino และสั่ง DF Player Mini เล่นเพลงใน Micro SD card

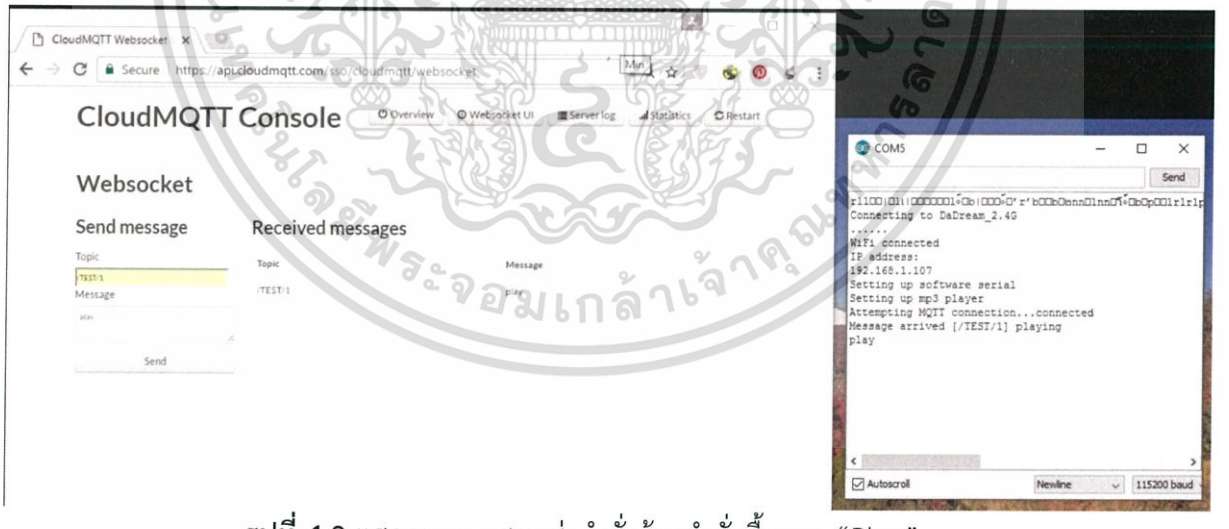
ทำการเชื่อมต่อ ESPino เข้ากับ Wifi Router และ ESPino เชื่อมต่อกับ MQTT อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.8** แสดงการเชื่อมต่อ ESPino เข้ากับ Wifi Router และ ESPino เชื่อมต่อกับ MQTT

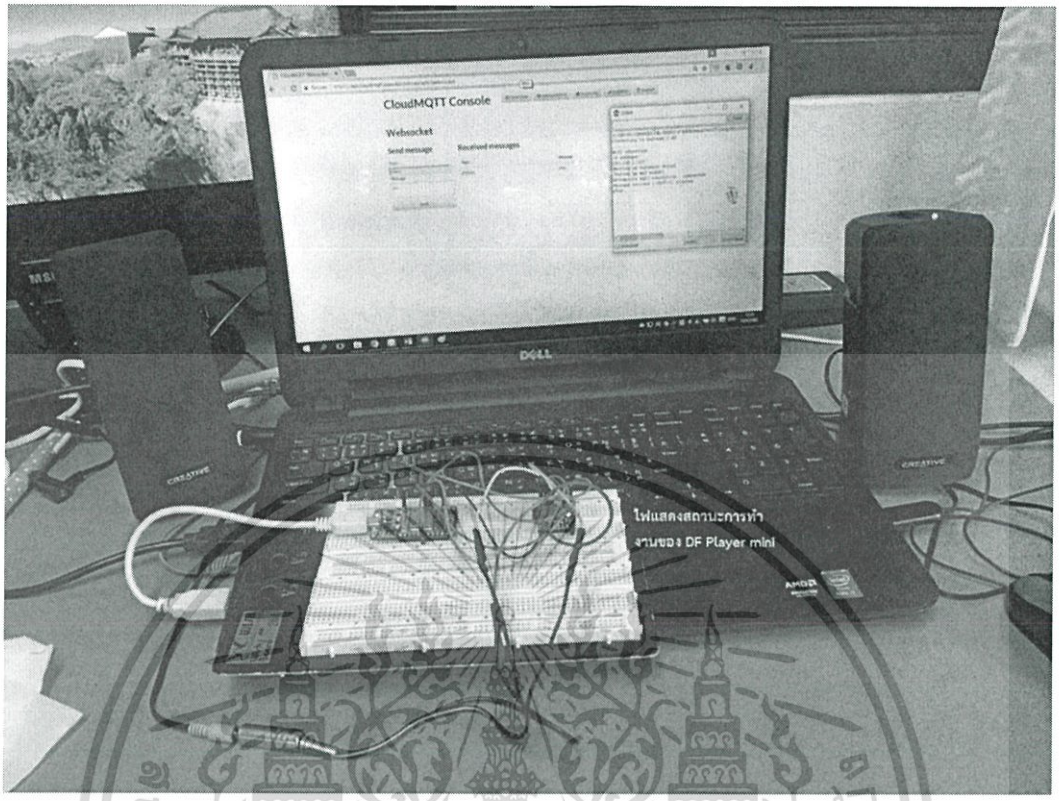
ทดสอบการส่งคำสั่งด้วยคำสั่งพื้นฐาน “Play” ซึ่ง ESPino จะแปลงคำสั่งเป็น mp3_play ที่เป็นคำสั่งเริ่มเล่นเพลงของ DF Player Mini




**รูปที่ 4.9** แสดงการทดสอบส่งคำสั่งด้วยคำสั่งพื้นฐาน “Play”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะสังเกตได้ว่า ESPino ได้สั่งให้การเล่นเพลงแล้ว โดยจะมีไฟสีฟ้าที่ DF Player Mini ติดขึ้น




รูปที่ 4.10 ตัวอย่างชิ้นงานที่เล่นเพลงจากคำสั่ง Play

## 4.5 ผลการทดสอบการควบคุมการเล่นไฟล์เสียงผ่านเว็บบนคอมพิวเตอร์

การทดสอบนี้จะทดสอบการทำงานของ ESPino ที่เชื่อมต่อกับ Wifi และ MQTT โดยจะไม่ใช้การสั่งงานผ่านทาง MQTT Web socket แต่จะใช้การสั่งงานจากเว็ปไซต์ที่เขียนขึ้นเพื่อรองรับการสั่งการเล่นไฟล์ MP3 ใน Micro SD card

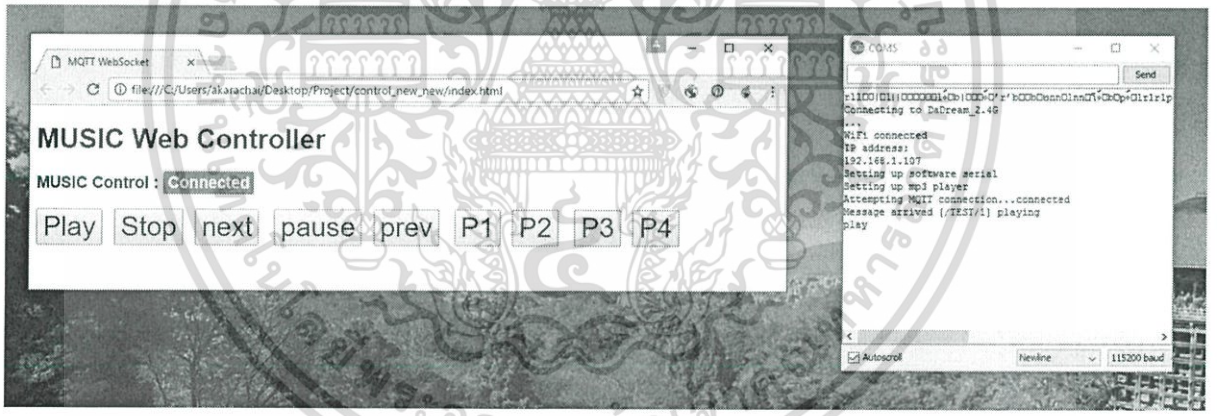
รูปที่ 4.11 แสดงหน้า Web Site ที่สามารถเชื่อมต่อกับ MQTT ได้สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเว็บสามารถเชื่อมต่อกับ MQTT ได้จะแสดง “Connected” พร้อมพื้นหลังสีเขียว หากเชื่อมต่อไม่ได้จะแสดง “Connect...” พร้อมพื้นหลังสีส้ม

รูปที่ 4.12 แสดงหน้า Web Site ที่ไม่สามารถเชื่อมต่อได้

จะพบว่าเมื่อคลิกที่ปุ่ม Play บนหน้าเว็บ ESPino จะได้รับคำสั่ง “Play” ซึ่งจะแปลงเป็นคำสั่ง “mp3_play” ให้กับ DF Player Mini เพื่อสั่งการทำงานเล่นเพลง จะพบว่าไฟสีฟ้าติด



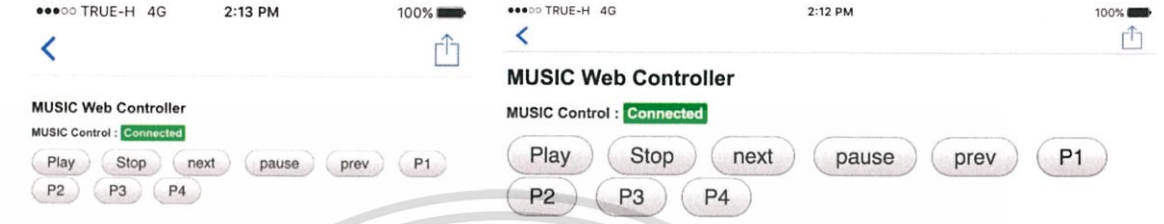

รูปที่ 4.13 แสดงหน้า Serial Monitor ที่รับคำสั่ง Play

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.6 ผลการทดสอบการควบคุมการเล่นไฟล์เสียงผ่านเว็บบนสมาร์ทโฟน

ในการทดสอบนี้จะนำเอาหน้าเว็บที่เขียนขึ้นมารันบนสามารถโฟนที่เป็น iOS เพื่อทดสอบความสามารถเข้าถึงสัญญาณอินเตอร์เน็ต

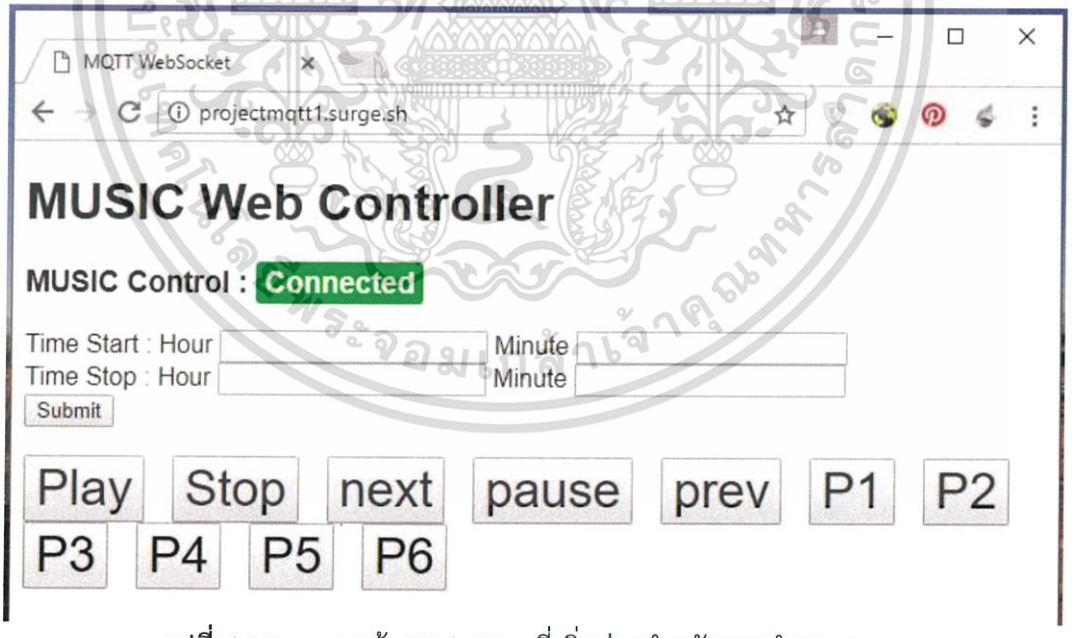
จากการทดลองใช้งานเว็บที่เขียนขึ้นพบว่าสามารถใช้สั่งการทำงานได้เหมือนกับการทดสอบบนคอมพิวเตอร์ทุกประการและสามารถใช้กับหน้าจอทั้งแนวตั้งและแนวนอน




**รูปที่ 4.14** แสดงหน้า Web Site ในแนวนอนบนโทรศัพท์

### 4.7 ผลการทดสอบการควบคุมผ่านเว็บโดยเป็นเว็บที่อยู่บนเซิร์ฟเวอร์

ในการทดสอบนี้จะนำเอาหน้าเว็บที่เขียนขึ้นไปฝากบนเซิร์ฟเวอร์ Surge และลองรัน ผลการทดลองปรากฏว่าสามารถใช้งานได้ดีทุกฟังชั่น




**รูปที่ 4.15** แสดงหน้า Web Site ที่เพิ่มช่องสำหรับการกำหนดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.8 ผลการทดสอบการควบคุมผ่านเว็บโดยเป็นเว็บที่อยู่บนเซิร์ฟเวอร์และตั้งเวลาการทำงาน**

ในการทดสอบนี้จะนำเอาหน้าเว็บที่เขียนขึ้นไปฝากบนเซิร์ฟเวอร์ Surge และลองรัน และใช้แถบตั้งเวลาเปิดปิด ผลการทดลองปรากฏว่าสามารถใช้งานได้ดีทุกฟังชั่น สามารถตั้งเวลาการเปิดปิดเพลงที่ต้องการได้ เพียงใส่เวลาที่หน้าเว็บ




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่5
# สรุปผลและข้อเสนอแนะ

## 5.1 สรุปผล

ปริญญานิพนธ์ฉบับนี้นำเสนอเครื่องควบคุมและสั่งการเล่นมัลติมีเดียระยะไกลผ่านอินเตอร์เน็ต ซึ่งจะนำมาใช้ในการสั่งการทำงานเครื่องกระจายเสียงและสื่อมัลติมีเดียโดยใช้ไมโครคอนโทรลเลอร์ ESPino ในการรับคำสั่งการทำงานและประมวลผลคำสั่งที่ได้รับจากเว็บไซด์ที่ออกแบบมาเพื่อการควบคุมระบบมัลติมีเดีย เพื่อให้สามารถควบคุมการเล่นไฟล์ MP3 ที่อยู่ใน Micro SD Card ได้ตามเงื่อนไขคือ สามารถสั่งการคำสั่งพื้นฐานของเครื่องเล่นเพลงได้แก่ การเล่น (Play) การหยุดชั่วคราว (Pause) การเล่นเพลงถัดไป (Next) การย้อนกลับ (Previous) และการเลือกไฟล์เพลงที่ระบุจำเพาะเจาะจงได้ (P1,...) ผู้ใช้งานสามารถสั่งการทำงานได้ โดยที่ผู้ใช้งานกับเครื่องฯ ไม่จำเป็นต้องอยู่ที่เดียวกันหรือประเทศเดียวกัน เพียงแค่ผู้ใช้งานและเครื่องฯ สามารถเข้าถึงระบบอินเตอร์เน็ตได้ ก็สามารถเชื่อมต่อและสั่งการได้จากทั่วทุกมุมโลกผ่านซอกเก็ต MQTT โดยใช้แบนวิธต่ำมากๆ จึงไม่จำเป็นต้องใช้อินเตอร์เน็ตความเร็วสูงในการเชื่อมต่อ นอกจากนี้ยังสามารถสั่งการทำงานแบบตั้งเวลาได้โดยไม่ต้องใช้อุปกรณ์เพิ่มเติม

## 5.2 ข้อเสนอแนะ

5.2.1 ควรนำไปประยุกต์พัฒนาร่วมกับอุปกรณ์อิเลคทรอนิกส์อื่นๆ เช่นการนำไปรวมกับจอโฆษณาหรือป้ายไฟวิ่งเพื่อให้สามารถใช้ประโยชน์ในการควบคุมได้สูงขึ้น

5.2.2 ควรนำไปใช้รวมกับการเปิดปิดเครื่องใช้ไฟฟ้าภายในบ้านร่วมกับรีเลย์เพื่อใช้ในการสั่งเปิดปิดประตูหรือเปิดปิดไฟป้องกันขโมยเมื่อไม่อยู่บ้าน เนื่องจากระบบมีราคาถูกและสามารถใช้งานได้อย่างแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


[1] Arduino Admin. Arduino Howto. [ออนไลน์]. เข้าถึงได้จาก : https://www.arduino.cc. (วันที่ค้นข้อมูล 20 กันยายน 2559).

[2] Mqtt admin. What is MQTT?. [ออนไลน์]. เข้าถึงได้จาก : http://mqtt.org (วันที่ค้นข้อมูล 30 ธันวาคม 2559).

[3] Thaieasyelec admin. บทความ Arduino คืออะไร? ตอนที่2 มาทำความรู้จักกับ Arduino รุ่นต่างๆกัน. [ออนไลน์]. เข้าถึงได้จาก : http://www.thaieasyelec.com. (วันที่ค้นข้อมูล 1 กันยายน 2559).

[4] Thaieasyelec admin. ESPino Wifi Development Board. [ออนไลน์]. เข้าถึงได้จาก : http://www.thaieasyelec.com. (วันที่ค้นข้อมูล 20 กุมภาพันธ์ 2560).

[5] Thaieasyelec admin. คู่มือการใช้งานบอร์ด. [ออนไลน์]. เข้าถึงได้จาก : http://www.thaieasyelec.com. (วันที่ค้นข้อมูล 20 กุมภาพันธ์ 2560).

[6] @surge_sh. Getting started with Surge. [ออนไลน์]. เข้าถึงได้จาก : http://surge.sh. (วันที่ค้นข้อมูล 1 เมษายน 2560).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อ Wi-Fi และการรับค่าจากจาก MQTT Broker เป็นคำสั่งดังนี้

```
#include <ESP8266WiFi.h>

// Update these with values suitable for your network.
const char* ssid = "ssid";
const char* password = "11111111";

// Config MQTT Server
#define mqtt_server "m13.cloudmqtt.com"
#define mqtt_port 10874
#define mqtt_user "Dream"
#define mqtt_password "999999"
WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

                        while (WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
                                }
                         Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
}
void loop() {
  if (!client.connected()) {
                          Serial.print("Attempting MQTT connection...");
                          if (client.connect("ESP8266Client", mqtt_user,
mqtt_password)) {
                           Serial.println("connected");
                           client.subscribe("DV");
                           } else {
                            Serial.print("faiDV, rc=");
                           Serial.print(client.state());
                          Serial.println(" try again in 5 seconds");
                          delay(5000);
                          return;
    }
  }
  client.loop();
}
void callback(char* topic, byte* payload, unsigned int length) {
 Serial.print("Message arrived [");
 Serial.print(topic);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.print("] ");
  String msg = "";
  int i=0;
  while (i<length) msg += (char)payload[i++];
  if (msg == "play") {
```

การนำเอาสัญญาณหลัง payload มาใช้สั่งการเล่นไฟล์ MP3 กับ

DF player mini + Micro SD card

```
 while (i<length) msg += (char)payload[i++];
  if (msg == "play") {
// client.publish("DV", (digitalRead(DV_PIN) ? "DVON" : "DVOFF"));
    mp3_play ();
    Serial.println("playing");
// return;
  } else if (msg == "stop") {
    mp3_stop ();
    Serial.println("stop");
  } else if (msg == "next") {
    mp3_next ();
    Serial.println("next");
  } else if (msg == "prev") {
    mp3_prev ();
    Serial.println("previous");
  } else if (msg == "pause") {
    mp3_pause ();
    Serial.println("paused");
  } else if (msg == "p1") {
    mp3_play (1);
    Serial.println("song1");
  } else if (msg == "p2") {
    mp3_play (2);
    Serial.println("song2");
  } else if (msg == "p3") {
    mp3_play (3);
    Serial.println("song3");
  } else if (msg == "p4") {
    mp3_play (4);
    Serial.println("song4");
  }
// digitalWrite(DV_PIN, (msg == "DVON" ? HIGH : LOW));
// Serial.println(msg);
}
```

ส่วนของการเขียนเว็บเพื่อใช้ Publish ข้อมูลคำสั่ง

```
var config = {
        mqtt_server: "m15.cloudmqtt.com",
        mqtt_websockets_port: 3XXXX,
        mqtt_user: "XXXX",
        mqtt_password: "XXXXXXXX"
};
$(document).ready(function(e) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        client = new Paho.MQTT.Client(config.mqtt_server,
config.mqtt_websockets_port, "web_" + parseInt(Math.random() * 100, 10));
        Paho.MQTT.Client("m15.cloudmqtt.com", 32903, "web_" +
parseInt(Math.random() * 100, 10));
        client.connect({
                useSSL: true,
                userName: config.mqtt_user,
                password: config.mqtt_password,
                onSuccess: function()
        $("#status").text("Connected").removeClass().addClass("connected");
                        client.subscribe("/DV");
                        mqttSend("/DV", "play");
                },
                onFailure: function(e) {
                        $("#status").text("Error : " + e).removeClass().addClass("error");
                        // console.log(e);
                }
        });
                client.onConnectionLost = function(responseObject) {
                if (responseObject.errorCode !== 0) {
                        $("#status").text("onConnectionLost:" +
responseObject.errorMessage).removeClass().addClass("connect");
                        setTimeout(function() { client.connect() }, 1000);
                }
        }
$("#dv-play").click(function(e) {
    mqttSend("/DV", "play");
 });
        $("#dv-stop").click(function(e) {
    mqttSend("/DV", "stop");
        });
        $("#dv-next").click(function(e) {
    mqttSend("/DV", "next");
        });
        $("#dv-pause").click(function(e) {
    mqttSend("/DV", "pause");
});
$("#dv-prev").click(function(e) {
    mqttSend("/DV", "prev");
  });
$("#dv-p1").click(function(e) {
    mqttSend("/DV", "p1");
  });
$("#dv-p2").click(function(e) {
    mqttSend("DV", "p2");
  });
$("#dv-p3").click(function(e) {
    mqttSend("DV", "p3");
  });
$("#dv-p4").click(function(e) {
    mqttSend("DV", "p4");
});
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Code ส่วนของ web site

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>MQTT WebSocket</title>
<script src="jquery-1.11.3.min.js"></script>
<script src="mqttws31.js"></script>
<script>
    function displaytime()
    {
        var d = new Date();
        var Hname = $("#txtHour").val();
        var Mname = $("#txtMin").val();
        var HnameS = $("#txtHourS").val();
        var MnameS = $("#txtMinS").val();

        $('#idtime').html(d.getHours() + ":" + d.getMinutes());
        $('#idtime2').html(Hname + ":" + Mname);
        $('#idtime3').html(HnameS + ":" + MnameS);

        if (d.getHours() == Hname)
        {
            if(d.getMinutes() == Mname)
            {
                mqttSend("/TEST/1", "play");
            }
        }

        if (d.getHours() == HnameS)
        {
            if(d.getMinutes() == MnameS)
            {
                mqttSend("/TEST/1", "stop");
            }
        }

        setTimeout("displaytime()",1000);
    }
</script>
<style>
body {
    font-family: Arial, Helvetica, sans-serif;
}

#status {
    background: #333;
    color: #FFF;
    border-radius: 3px;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    font-weight: bold;
    padding: 3px 6px;
}

#status.connect {
    background: #E18C1A;
    color: #FFF;
}

#status.connected {
    background: #00AE04;
    color: #FFF;
}

#status.error {
    background: #F00;
    color: #FFF;
}

button {
    font-size: 32px;
}
</style>
<script>
var config = {
    mqtt_server: "m13.cloudmqtt.com",
    mqtt_websockets_port: 30874,
    mqtt_user: "Dream",
    mqtt_password: "999999"
};

$(document).ready(function(e) {

    client = new Paho.MQTT.Client(config.mqtt_server, config.mqtt_websockets_port, "web_" + parseInt(Math.random() * 100, 10));

    client.connect({
        useSSL: true,
        userName: config.mqtt_user,
        password: config.mqtt_password,
        onSuccess: function() {
            // Once a connection has been made, make a subscription and send a message.
            console.log("onConnect");
            $("#status").text("Connected").removeClass().addClass("connected");
            client.subscribe("/TEST/1");
            mqttSend("/TEST/1", "stop");
        },
        onFailure: function(e) {
            $("#status").text("Error : " + e).removeClass().addClass("error");

            mqttSend("/TEST/1", "stop");
        },
        onFailure: function(e) {
            $("#status").text("Error : " + e).removeClass().addClass("error");
        }
    });

    client.onConnectionLost = function(responseObject) {
        if (responseObject.errorCode !== 0) {
            $("#status").text("onConnectionLost:" + responseObject.errorMessage).removeClass().addClass("connect");
            setTimeout(function() { client.connect() }, 1000);
        }
    }

    $("#dv-play").click(function(e) {
        mqttSend("/TEST/1", "play");
    });

    $("#dv-stop").click(function(e) {
        mqttSend("/TEST/1", "stop");
    });
});

var mqttSend = function(topic, msg) {
    var message = new Paho.MQTT.Message(msg);
    message.destinationName = topic;
    client.send(message);
}
</script>
</head>

<body>
<h1>MUSIC Web Controller</h1>

<h3>MUSIC Control : <span id="status" class="connect">Connect...</span></h3>

Time Start : Hour <input type="text" id="txtHour" /> Minute <input type="" id="txtMin"/>

<br />Time Stop : Hour <input type="text" id="txtHourS" /> Minute <input type="" id="txtMinS"/>

<br /><input type="button" value="Submit" onclick="displaytime();" /> <br /><br />

<button id="dv-play" >Play</button>   
<button id="dv-stop" >Stop</button>   
<button id="dv-next" >next</button>   
<button id="dv-pause" >pause</button>   
<button id="dv-prev" >prev</button>   
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้