รายงานสหกิจศึกษาฉบับสมบูรณ์

ต้นแบบอินเทอร์เน็ตออฟติงเกทเวย์แพลตฟอร์ม
IoT Gateway Platform Prototype

นายพีรพัชร อรุณเรื่อ

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

รายงานสหกิจศึกษาฉบับสมบูรณ์

ต้นแบบอินเทอร์เน็ตออฟติงเกทเวย์แพลตฟอร์ม

IoT Gateway Platform Prototype

นายพีรพัชร อรุณเรื่อ

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงงานสหกิจศึกษา ไอโอทีเกทเวย์โปรโตไทต์

ชื่อ-สกุล นักศึกษา นายพีรพัชร อรุณเรื่อ

คณะ วิศวกรรมศาสตร์ **ภาควิชา** วิศวกรรมอิเล็กทรอนิกส์

ชื่อ-สกุล อาจารย์นิเทศ ผศ.เกรียงไกร สุขสุด

ชื่อ-สกุล อาจารย์นิเทศ อ.ชินภัทร นันทจิวากรชัย

ชื่อ-สกุล ผู้นิเทศงาน นายคำรณ อรุณเรื่อ

สถานประกอบการ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

บทคัดย่อ

อุปกรณ์ IoT Gateway Platform ถูกสร้างขึ้นมาภายใต้กรอบความคิดว่าด้วยต้องการสนับสนุนให้เกิดการใช้และพัฒนาระบบ IoT ภายในประเทศไทยมากขึ้น เช่นการนำระบบ IoT มาควบคุมผลผลิตทางการเกษตร เกิดเป็นระบบ Smart Farm หรือนำระบบ IoT มาใช้ในครัวเรือน เกิดเป็นระบบ Smart Home ซึ่งเป็นการนำเทคโนโลยีมารับใช้ชีวิตมนุษย์ ส่งผลให้ชีวิตความเป็นอยู่ของมนุษย์ดีขึ้น โดยมีคนไทยเป็นผู้เล่นและเป็นเจ้าของเทคโนโลยีที่ตนคิดค้นขึ้น โดยผู้ที่สนใจสามารถนำไปพัฒนาต่อยอดจนทำให้เกิดเกิดสิ่งใหม่ๆ ขึ้นอีกขั้นต่อไปเป็นมรดกทางเทคโนโลยีในที่สุด สุดท้ายแล้วความก้าวหน้าทางเทคโนโลยีจะกลับคืนสู่สังคมจนเกิดการพัฒนาให้ประเทศไทยเป็น 4.0 อย่างเป็นรูปธรรมมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cooperative Title: IoT Gateway Platform Prototype

Student intern name: Peerapat Aroonrua

Faculty: Engineering **Department:** Electronics Engineering

Advisor name: Asst. Prof. Kriangkrai Sooksood

Advisor name: Chinnapat Nantajiwakornchai

Mentor name: Mr.Kamron Aroonrua

Company National Electronics And Computer Technology Center

ABSTRACT

IoT Gateway Platform has made from the idea that I want to encourage people to use and develop IoT system in Thailand. For example, using IoT for farming control and then, turn to be Smart Farm system. Also using IoT in house and then, turn to be Smart Home. The thing is we (Thai people) can create and own this technology, which it will serve our better lives. Later, who else wants to bring this technology to the next steps to make new better things in the future so then it would be a technological heritage at last. And the progression of technology will be beneficial to this country and bring us to Thailand 4.0 in the real way.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ด้วยความอนุเคราะห์จาก นายคำรณ อรุณเรื่อ ซึ่งเป็นที่ปรึกษาปริญญานิพนธ์ ที่คอยแนะนำการดำเนินงานที่เหมาะสม และการแก้ปัญหาอย่างถูกต้อง รวมทั้งคอยให้ความรู้ความเข้าใจที่ชัดเจน จนนำมาซึ่งความสำเร็จของปริญญานิพนธ์ฉบับนี้

ขอขอบคุณเพื่อนต่างสาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ที่คอยรับฟังปัญหา และคอยชี้แนะเกี่ยวกับการเขียนภาษาซี อีกทั้งร่วมช่วยแก้ปัญหาจนสำเร็จลุล่วงไปด้วยดี ตลอดระยะเวลาในการทำปริญญานิพนธ์

ขอขอบคุณครอบครัวอรุณเรื่อ ที่คอยให้กำลังใจตลอดเวลา ขอบคุณนายภัทธิภูมิ ฤทธิ์เพชร ผู้ที่เปรียบเสมือนญาติผู้ใหญ่คนหนึ่ง ที่มีส่วนช่วยให้ข้าพเจ้าได้มีโอกาสเรียนรู้และมีความสนใจเกี่ยวกับอิเล็กทรอนิกส์เป็นบุคคลแรกในชีวิตของข้าพเจ้า ขอบคุณนายวิทยา บัวบุญ และ นายไพรัธ สายเนียม ผู้ที่คอยให้คำปรึกษาเกี่ยวกับเทคนิคเชิงช่างที่มีส่วนอย่างมากในการวิเคราะห์ปัญหาและกระบวนการแก้ปัญหาต่างๆ ที่เกิดขึ้นระหว่างการทำปริญญานิพนธ์

สุดท้ายนี้ขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังแห่งนี้ที่ได้ให้ประสบการณ์ที่ดีต่อข้าพเจ้า ข้าพเจ้าขอขอบคุณมา ณ โอกาสนี้ด้วย นอกจากนี้ข้าพเจ้าขอขอบคุณบุคลากรทางการศึกษา อาจารย์ในสาขาอิเล็กทรอนิกส์และต่างสาขาวิชาที่ถ่ายทอดความรู้กระบวนการความคิดที่มีคุณภาพให้แก่ข้าพเจ้า และผู้ใดที่ให้ความช่วยเหลือไม่มากก็น้อย ทั้งผู้ที่ไม่ได้กล่าวนามทุกท่าน ที่ให้การสนับสนุนช่วยเหลือจนกระทั่งปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ขอกราบขอบพระคุณอย่างสูง

นายพีรพัชร อรุณเรื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญ	1
1.2 ภาพรวมหรือโครงสร้างของโครงงาน	1
1.3 วัตถุประสงค์โครงงาน	3
1.4 ขั้นตอนการดำเนินงาน	3
1.5 อุปกรณ์ที่ใช้	3
1.5.1 ฮาร์ดแวร์	3
1.5.2 ซอฟแวร์	4
1.6 ขั้นตอนการดำเนินงาน	4
1.7 ขอบเขตของการวิจัย	4
บทที่ 2 ทฤษีที่เกี่ยวข้อง	5
2.1 ภาคจ่ายไฟ	5
2.1.1 แรงกระเพื่อมไฟฟ้า	5
2.1.2 ศักย์ไฟฟ้า	6
2.1.3 กระแสไฟฟ้า	6
2.2 รูปแบบการสื่อสาร	7
2.2.1 การสื่อสารผ่านสายเคเบิ้ล	7
2.2.1.1 มาตรฐานการสื่อสารแบบซีเรียล	7
2.2.1.2 มาตรฐานการสื่อสารแบบ I2C	8
2.2.1.3 การรับส่งข้อมูลผ่านมาตรฐานการสื่อสาร RS485	11
2.2.2 การสื่อสารไร้สาย	12
2.2.2.1 Bluetooth	12
2.2.2.2 WIFI	12
2.2.2.3 Narrow Band	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.3 Microcontroller ESP32	13
บทที่ 3 การวิเคราะห์และการออกแบบ	16
3.1 การออกแบบการทำงานของระบบ	17
3.2 การออกแบบภาคจ่ายไฟให้ระบบ	17
3.2.1 หลักการทำงานของภาคจ่ายไฟ	18
3.2.2 การคำนวณค่าความจุของตัวเก็บประจุภายในภาคจ่ายไฟ	18
3.3 การแปลงมาตรฐานการสื่อสารแบบอนุกรมไปสู่มาตรฐานการสื่อสาร RS485	20
3.4 วงจรกำเนิดสัญญาณนาฬิกา	22
3.5 วงจรลดระดับแรงดันของสัญญาณดิจิตอล	22
3.6 สัญญาณเข้าออกภายใน Microcontroller ESP32	24
3.6.1 รีสตาร์ท (Restart)	24
3.6.2 การอัพโหลดซอฟแวร์ (Boots Loader)	25
3.6.3 หลอดไฟแสดงสถานะ	26
3.6.4 การอ่านค่าจากตัวต้านทานที่มีค่าเปลี่ยนแปลงตามแสง	27
3.6.5 การส่งผ่านข้อมูลด้วยมาตรฐานการสื่อสารไอทูซี (I2C)	29
3.6.6 การส่งผ่านข้อมูลด้วยมาตรฐานการสื่อสารซีเรียล (Serial,TTL)	30
3.6.7 การอ่านค่าจากโมดูลสัญญาณนาฬิกา (Real Time Clock : RTC)	31
3.7 อุปกรณ์ต้นแบบ	31
บทที่ 4 ผลการทดลองในการดำเนินโครงการ	35
4.1 วงจร Reset	35
4.2 วงจร Logic Converter	36
4.3 วงจร Serial to Rs485	36
4.4 ระดับแรงดันไฟฟ้าของสัญญาณดิจิตอลในรูปแบบการสื่อสาร I2C	37
4.5 ระดับแรงดันไฟฟ้าของสัญญาณดิจิตอลในรูปแบบการสื่อสาร Serial	38
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	39
5.1 บทสรุปโครงงาน	39
5.2 ข้อเสนอแนะในการทำวิจัยครั้งต่อไป	39
บรรณานุกรม	40
ภาคผนวก	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

ภาพที่	หน้า
1.1 ความสัมพันธ์ระหว่างโมดูลต่างๆภายใน IoT Gateway Platform	2
1.2 Gateway เปรียบเสมือนประตูระหว่าง Cloud Computing และ Sensor Network	2
1.3 ตารางเวลาปฏิบัติงาน	4
2.1 ลักษณะวงจรภาคจ่ายไฟพื้นฐาน	5
2.2 แรงดันกระเพื่อม	5
2.3 แสดงการเปรียบเทียบให้กระแสไฟฟ้าเป็นน้ำ	6
2.4 กฎของโอห์ม(Ohm law)	6
2.5 พอร์ทเชื่อมต่อฮาร์ทแวร์ RS485	7
2.6 ลักษณะสัญญาณ TTL	7
2.7 ลักษณะข้อมูลในการส่งแบบ TTL	8
2.8 การเชื่อมต่อสายสัญญาณแบบไอทูซี (I2C)	8
2.9 แสดงลักษณะการติดต่อสื่อสารแบบไอทูซี (I2C)	8
2.10 ลักษณะสัญญาณเริ่มต้น-หยุด (Start-Stop Condition)	9
2.11 บิตควบคุม (Mode)	9
2.12 ลักษณะช่วงเวลาในกรรับส่งข้อมูล	10
2.13 แสดงวงจรเอาต์พุตของอุปกรณ์ระบบ I2C	11
2.14 ลักษณะการใช้งานมาตรฐานการสื่อสารแบบ RS485	11
2.15 ลักษณะการเชื่อมต่อสายไฟในมาตรฐานการสื่อสารแบบ RS485	12
2.16 โมดูล Narrow Band จากผู้ให้บริการสัญญาณโทรศัพท์ TRUE	13
2.17 Function Block Diagram	14
2.18 ESP32 Pin lay out	15
2.19 ESP32 Power Scheme	15
3.1 จุดมุ่งหมายการออกแบบ	16
3.2 ช่องทางการสื่อสารระหว่างเซ็นเซอร์ผู้ใช้งานและระบบคราวด์	17
3.3 วงจรภาคจ่ายไฟที่ใช้	17
3.4 จุดทำงานของ ESP32 ที่ผู้ใช้แนะนำ	18
3.5 แสดงวงจร Power Supply	18
3.6 สมการในการคำนวณหาค่าตัวเก็บประจุในภาคจ่ายไฟ	19
3.7 รูปวงจร Serial to TTL	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

ภาพที่	หน้า
3.8 ค่าความจุของตัวเก็บประจุ	20
3.9 ค่าความจุของตัวเก็บประจุที่มีผลต่อการออกแบบใช้งานของ MAX232,MAX485	21
3.10 ลักษณะแรงดันของลอจิกของมาตรฐานการสื่อสารแบบ TTL และ RS232	21
3.11 รูปวงจร RTC	22
3.12 ลักษณะระดับสัญญาณเข้าและขาออกเมื่อผ่านวงจร Logic Converter	23
3.13 รูปวงจร Logic converter	23
3.14 ESP 32	24
3.15 วงจรรีสตาร์ท	24
3.16 Enable Pin / Resets Pin	25
3.17 สมการกฎของโอห์ม	25
3.18 Data Sheet แสดงช่อง IO0	26
3.19 รูปวงจร Boots Loader	26
3.20 วงจรแสดงจุดเชื่อมต่อ LED	26
3.21 ลักษณะวงจร LDR	27
3.22 อ่านค่าจาก LDR	27
3.23 แสดงโครงสร้างของการสื่อสารแบบไอทูซี (I2C)	28
3.24 GPIO ที่ใช้เป็น I/O Serial	29
3.25 คำสั่งที่ใช้ในการใช้งาน Serial Port	29
3.26 คำสั่งภาษาซีที่เรียกเวลาจากโมดูล RTC	30
3.27 วงจร RTC	30
3.28 Full schematic IoT Gateway Platform	31
3.29 อุปกรณ์ต้นแบบบนแผ่น PCB อเนกประสงค์	32
3.30 Recommended PCB Land Pattern	32
3.31 Iot Gateway Platform Prototype 3D	33
4.1 Serial Monitor แสดงผล Reset	35
4.2 3.3o 5 V	36
4.3 5v to 3.3 V	36
4.4 ลักษณะสัญญาณเมื่อถูกแปลงรูปแบบการสื่อสาร	37
4.5 แสดง Data ที่ถูกส่งไปในรูปแบบการสื่อสาร I2C	37
4.6 สัญญาณนาฬิกา	38
4.7 แสดง Data ที่ถูกส่งไปในรูปแบบการสื่อสาร Serial	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

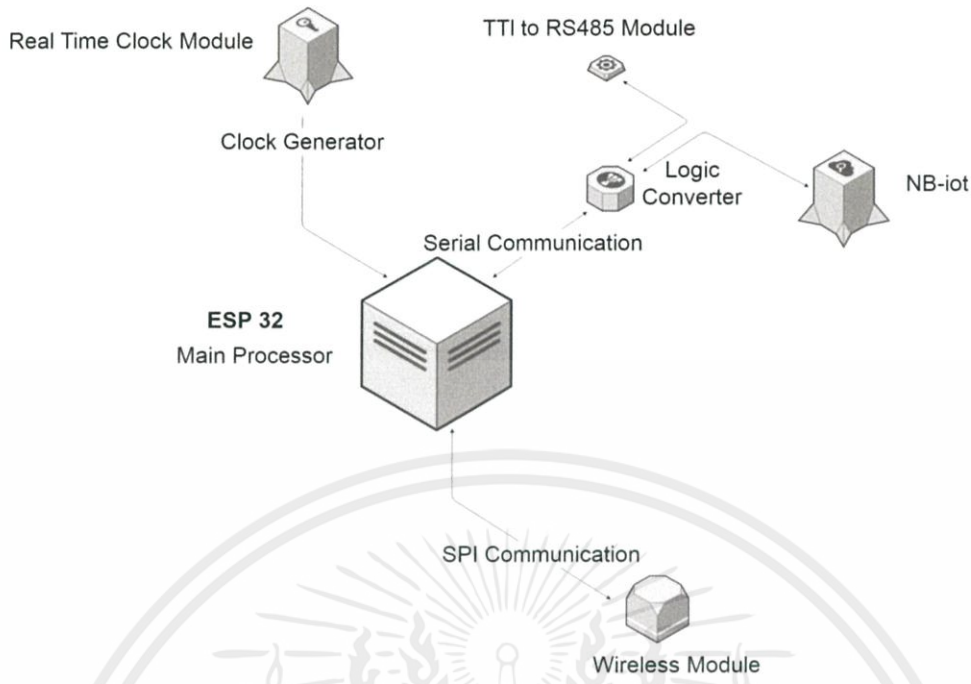
ปริญญานิพนธ์ฉบับนี้เสนอระบบการทำงานของ IoT Gateway Platform รวมไปจนถึงขั้นตอนการผลิตชิ้นงานต้นแบบ เนื่องจากระบบ Internet of Things จะถูกนำมาใช้อย่างกว้างขวางในทุกๆกิจกรรมในการดำเนินชีวิต IoT Gateway Platform จะทำให้เกิดการประยุกต์ใช้งาน IoT อย่างกว้างขวาง ซึ่ง IoT Gateway Plat Form มีจุดประสงค์การออกแบบให้ยืดหยุ่นกับผู้ใช้ในทุกระดับ ซึ่งอาจส่งผลให้เป็นแรงขับเคลื่อนระบบ IoT ให้มีการพัฒนาใช้งานในทุ่งหย่อมหญ้า

1.2 ภาพรวมหรือโครงสร้างรวมของโครงงาน

ระบบจะประกอบไปด้วย Microcontroller เบอร์ ESP32 เป็นหัวใจสำคัญของระบบทำงานร่วมกับโมดูลย่อย 5 โมดูลคือ โมดูลสัญญาณนาฬิกา (RTC) โมดูลแปลงระดับสัญญาณ (Logic Converter) โมดูลแปลงรูปแบบการสื่อสารแบบซีเรียลไปสู่รูปแบบการสื่อสารแบบอาเอสสี่แปดห้า (Serial to RS485) โมดูลสื่อสารของหน่วยงานเอกชน (Narrow Band) โมดูลสื่อสารไร้สาร (Wireless)

ทุกโมดูลบน IoT Gateway Platform จะสามารถทำงานได้สัมพันธ์กันได้ด้วยสัญญาณนาฬิกา โดยจะเป็นหน้าที่ของโมดูลสัญญาณนาฬิกา (RTC) เป็นผู้ผลิตสัญญาณนาฬิกาให้กับทุกโมดูล ทำให้การส่งข้อมูลจากเซ็นเซอร์ภายนอกที่ถูกนำมาเชื่อมต่อกับ IoT Gateway Platform ผ่านช่องการเชื่อมต่ออาเอสสี่แปปดห้า (RS485)และระบบการสื่อสารไร้สาย (Wireless) และเข้าสู่ Microcontroller ผ่านการจัดเลียงลำดับข้อมูลให้มีรูปแบบตามที่เซิฟเวอร์ (Server) และ Gateway ตกลงร่วมกันหลังจากนั้นข้อมูลจะถูกส่งผ่านระบบอินเตอร์เน็ตผ่านโมดูลสื่อสารของหน่วยงานเอกชน (Narrow Band) และความสามารถพิเศษ Microcontroller เบอร์ ESP32 ขึ้นสู่ระบบคลาวด์ (Clound) เป็นไปได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

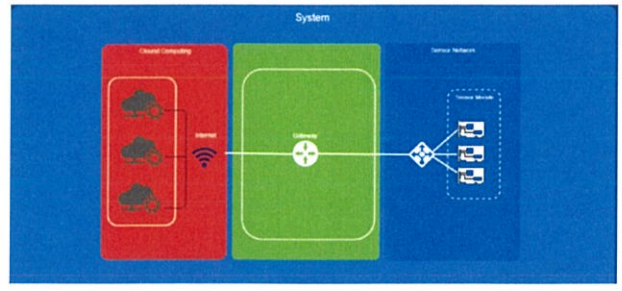


รูปที่ 1.1 ความสัมพันธ์ระหว่างโมดูลต่างๆภายใน IoT Gateway Platform

จากรูปจะเห็นได้ว่าการส่งข้อมูลเข้าออกกับไมโครโปรเซสเซอร์(Microprocessor)เบอร์ ESP32 ด้วยมาตรฐานการสื่อสารแบบซีเรียล (Serial) จำเป็นต้องทำการแปลงระดับแรงดันของลอจิกด้วยโมดูลแปลงระดับแรงดัน (Logic Converter) เพื่อป้องกันความเสียหายแก่อุปกรณ์ซึ่ง Microcontroller เบอร์ ESP32 สามารถสื่อสารด้วยมาตรฐานการสื่อสารแบบซีเรียล(Serial) ที่มีระดับแรงดันรอจิกต่ำเท่านั้น (LVTTL) หรือมีระดับแรงดันของริจิกเพียง 3.3V เท่านั้น

IoT Gateway Platform เปรียบเสมือนประตูที่สามารถเปิดไปมาระหว่าง Sensor Network และระบบ Cloud Computing หน้าที่ของ IoT Gateway คือรับข้อมูลจาก Sensor Network และส่งข้อมูลขึ้นสู่ระบบ Cloud Computing ด้วยรูปแบบการสื่อสารที่ตกลงร่วมกัน มีเวลาในการติดต่อสื่อสารเป็นตัวกำหนดจังหวะในการรับ-ส่งข้อมูล

Microprocessor บน IoT Gateway จะไม่มีการประมวลผลข้อมูลใดๆทั้งสิ้นก่อนส่งขึ้นสู่ระบบ Cloud Computing



รูปที่ 1.2 Gateway เปรียบเสมือนประตูระหว่าง Cloud Computing และ Sensor Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 วัตถุประสงค์ของโครงงาน

เพื่อผลิตอุปกรณ์ที่มีส่วนช่วยกระตุ้นและพัฒนาระบบ IoT ภายในประเทศให้มีการใช้งานอย่างกว้างขวางมากยิ่งขึ้น ผู้ใช้งานทุกกลุ่มสามารถใช้งานได้ ตั้งแต่ระดับนักศึกษา นักพัฒนา ผู้ประกอบการขนาดเล็ก (Start Up) ระดับอุตสาหกรรม อุปกรณ์ถูกพัฒนาขึ้นภายใต้แนวทางสี่ข้อคือ 1.ยืดหยุ่นต่อผู้ใช้งาน (Flexible) 2.ใช้พลังงานต่ำ(Low - Power) 3.ต้นทุนต่ำ (Low Cost) 4.ขนาดเล็ก(Size)

1.4 ขั้นตอนการดำเนินงาน

-ทำความเข้าใจระบบ IOT

-ศึกษาความต้องการของผู้ใช้ในระบบ IOT Gate way

-ศึกษา Microcontroller เบอร์ ESP32

-เริ่มทำชิ้นงานต้นแบบชิ้นที่หนึ่ง

-ศึกษาความต้องการของผู้ใช้เพื่อพัฒนาอุปกรณ์ให้สามารถใช้งานกับผู้ใช้งานได้กว้างขวางมากขึ้น

-เริ่มทำชิ้นงานต้นแบบชิ้นที่สอง

-ทดลองใช้งานอุปกรณ์กับผู้ใช้สามกลุ่ม

1.นักศึกษา : ทดลองใช้ในงานเพาะเลี้ยงเนื้อเยื่อของนักศึกษาคณะวิทยาศาสตร์ชั้นปีที่ 4

2.ผู้ประกอบการขนาดเล็ก (Start Up) : นำไปใช้ในระบบสมาร์ทฟาร์ม

3.ระดับอุตสาหกรรม : ใช้เพื่อควบคุมหม้อต้มแรงดันสูง(Boiler)

1.5 อุปกรณ์ที่ใช้

1.5.1 ฮาร์ดแวร์

- คอมพิวเตอร์ (Computer) จำนวน 1 เครื่อง
- ออสซิลโลสโคป (Oscilloscope) จำนวน 1 เครื่อง
- พาวเวอร์ซัพพลาย (Power Supply) จำนวน 1 เครื่อง
- โมดูลสื่อสารของหน่วยงานเอกชน (Narrow Band) จำนวน 1 บอร์ด
- โมดูลสื่อสารไร้สาย (Wireless) จำนวน 1 บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5.2 ซอฟแวร์

-โปรแกรม VS Code

-โปรแกรม Arduino

-โปรแกรม LT Spice

1.6 ขั้นตอนการดำเนินงาน

หัวข้อ	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. กำหนดหัวข้อโครงงาน	←	→														
2. ศึกษาหลักการทำงานและเสนอฟังก์ชันการทำงาน		←	—	→												
3. เลือกอุปกรณ์ที่จะทำมาใช้งานในวงจรพร้อมการคำนวณค่า					←	—	—	→								
4. ออกแบบลายวงจรบนโปรแกรม Altium Design						←	—	—	—	→						
5. เริ่มประกอบและทดสอบอุปกรณ์ต้นแบบบนแผ่น PCB อเนกประสงค์											←	→				
6.สร้างโมเดล 3D บนโปรแกรม Altium Design														←	—	→
7.จัดทำการนำเสนองาน													←	—	—	→

รูปที่ 1.3 ตารางเวลาปฏิบัติงาน

1.7 ขอบเขตของการวิจัย

ศึกษาและพัฒนาอุปกรณ์ Iot Gateway ที่ถูกสร้างขึ้นบน Platform Arduino โดยใช้ Microprocessor ESP32 หรืออาจเรียกชื่อโครงงานว่า Iot Gateway ESP32 Platform โดยมุ่งเน้นไปในส่วนของการสร้างอุปกรณ์ฮาร์ดแวร์ มีความรู้พื้นฐานเกี่ยวกับการเขียนภาษาซี สามารถเขียนคำสั่งและสามารถทดสอบสั่งการอุปกรณ์ฮาร์ดแวร์ที่สร้างขึ้นมาได้ สร้างชิ้นงานต้นแบบที่อยู่ในรูปของแผ่น PCB อเนกประสงค์ และสร้างโมเดลสามมิติของชิ้นงานจริงก่อนเข้าสู่กระบวนการผลิต

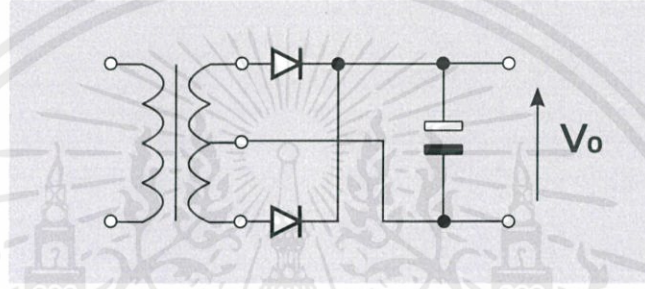
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2

ทฤษฎีที่เกี่ยวข้อง

2.1 ภาคจ่ายไฟ

วงจรภาคจ่ายไฟถือว่าเป็นหัวใจสำคัญของวงจรอิเล็กทรอนิกส์ทุกวงจรในโลกใบนี้ เพราะหากขาดวงจรนี้ไปวงจรย่อยภายในระบบคงไม่สามารถทำงานได้นักอิเล็กทรอนิกส์ทุกคนควรออกแบบวงจรนี้ให้มีความเสถียรและมีอัตราการสูญเสียที่น้อย (Power Loss)

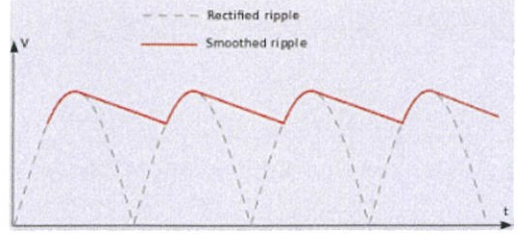


รูปที่ 2.1 ลักษณะวงจรภาคจ่ายไฟพื้นฐาน

ส่วนประกอบพื้นฐานของวงจรภาคจ่ายไฟจะประกอบไปด้วย 3 ส่วนหลักคือหม้อแปลง (Step Down transformer) 2.ไดโอด (Diode) 3.ตัวเก็บประจุ (Capacitor) ความเสถียรของวงจรภาคจ่ายไฟสามารถแสดงได้โดยคุณสมบัติสามข้อคือ 1.แรงกระเพื่อมไฟฟ้า (Ripple) 2.ศักย์ไฟฟ้า (Voltage) 3.การแสไฟฟ้า(Current)

2.1.1 แรงกระเพื่อมไฟฟ้า (Ripple)

แรงกระเพื่อมไฟฟ้า (Ripple) หรือแรงดันกระเพื่อมเป็นตัวเลขที่ส่งผลให้ไฟฟ้ากระแสตรงไม่สามารถเป็นไฟฟ้ากระแสตรงในอุดมคติได้ เกิดขึ้นในกระบวนการเรียงกระแส (Rectifier) ซึ่งเป็นผลมาจากอุปกรณ์สองชนิดหลักคือไดโอด (Diode) และตัวเก็บประจุ (Capacitor) แรงกระเพื่อมไฟฟ้าอาจไม่มีผลกับวงจรอิเล็กทรอนิกส์ขนาดใหญ่แต่ในระบบอิเล็กทรอนิกส์ขนาดเล็กเช่นในส่วนวงจรควบคุมมักเกิดปัญหาขึ้นได้เช่นทำงานผิดพลาด ออสซิเรท (oscillate) หรือถึงขั้นทำให้เกิดความเสียหายขึ้นแก่อุปกรณ์



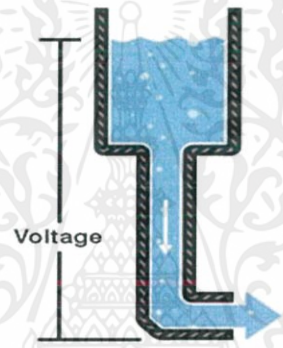
รูปที่ 2.2 แรงดันกระเพื่อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเส้นปะแสดงลักษณะแรงดันเมื่อผ่าน Diode Rectifier มีลักษณะเป็นไฟฟ้ากระแสตรงเพียงชั่วขณะ แอมพลิจูดสูงสุดจะเปลี่ยนแปลงไปตามเฟสของไฟฟ้ากระแสสลับ เส้นสีแดงหมายถึงแรงดันไฟฟ้าเมื่อออกจากวงจรภาคจ่ายไฟเกิดซึ่งเกิดจากการชาร์ตและคายประจุของตัวเก็บประจุ แต่ยังคงไม่เป็นไฟฟ้ากระแสตรงในอุดมคติ

2.1.2 ศักย์ไฟฟ้า(Voltage)

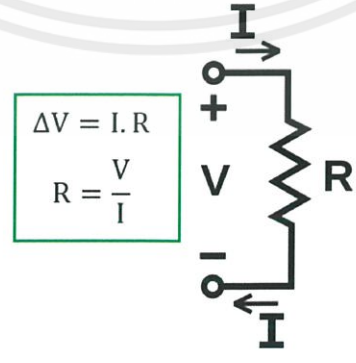
ศักย์ไฟฟ้าหรือแรงเคลื่อนไฟฟ้าเป็นปริมาณทางไฟฟ้าที่ทำให้เกิดการเคลื่อนที่ของประจุ(อิเล็กตรอน) เกิดขึ้นบริเวณจุดสองจุด โดยศักดิ์ไฟฟ้าที่มากย่อมทำให้เกิดการเคลื่อนที่ของประจุมากตามไปด้วย หากเปรียบเทียบกระแสไฟฟ้าเสมือนน้ำศักดิ์ไฟฟ้าคือความสูงที่แตกต่างกันระหว่างสองจุดน้ำไหลจากที่สูงไปที่ต่ำฉันใด กระแสไฟฟ้าย่อมไหลจากศักดิ์สูงไปศักดิ์ต่ำฉันนั้น



รูปที่ 2.3 แสดงการเปรียบเทียบให้กระแสไฟฟ้าเป็นน้ำ

2.1.3 กระแสไฟฟ้า

กระแสไฟฟ้าที่เคลื่อนผ่านบริเวณหนึ่งๆ เกิดขึ้นจากความต่างศักดิ์ระหว่างสองจุด โดยปริมาณกระแสไฟฟ้าที่เกิดขึ้นเป็นไปตามกดของโอห์ม



รูปที่ 2.4 กฎของโอห์ม(Ohm law)

เมื่อเกิดการเคลื่อนที่ของประจุ ย่อมทำให้เกิดกระแสไฟฟ้าและเป็นผลทำให้เกิดกำลังงานขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 รูปแบบการสื่อสาร

อุปกรณ์ Microcontroller จำเป็นต้องมีการติดต่อสื่อสารกับเซ็นเซอร์ ซึ่งภายใน IoT Gateway platform เต็มไปด้วยการติดต่อสื่อสารระหว่างโมดูลมากมายจำแนกออกเป็นสองลักษณะ ย่อยคือการสื่อสารแบบไร้สายและผ่านสายเคเบิ้ล

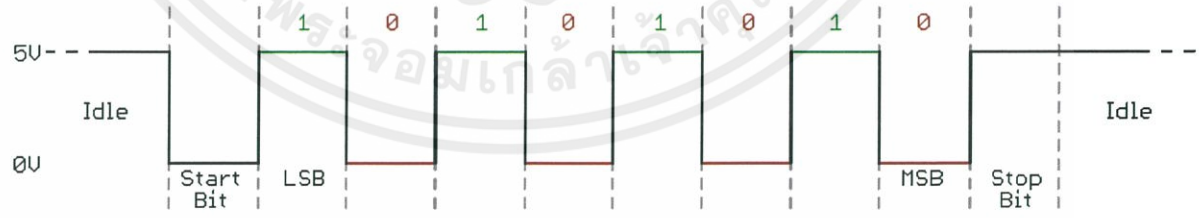
2.2.1 การติดต่อสื่อสารผ่านสายเคเบิ้ล

ถูกนำมาใช้เพื่อแลกเปลี่ยนข้อมูล(Data) ภายใน IOT gateway platform และยังสามารถติดต่อสื่อสารกับ Sensor Module ภายนอกได้อีกด้วย โดยสามารถเชื่อมต่อด้วยมาตรฐานการสื่อสารแบบ ซีเรียล(TTL),ไอทูซี(I2C),อาเอสสี่แปดห้า(RS485)

รูปที่ 2.5 พอร์ทเชื่อมต่อฮาร์ทแวร์ RS485

2.2.1.1 มาตรฐานการสื่อสารแบบซีเรียล (Transistor Transistor Logic : TTL)

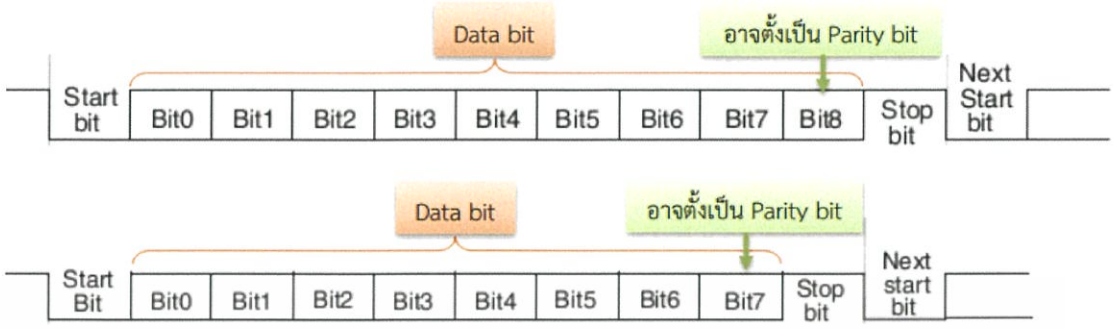
มีหน้าที่หลักในการติดต่อสื่อสารระหว่างทรานซิสเตอร์มีลักษณะเป็นสัญญาณเพ้าขนาด 5V



รูปที่ 2.6 ลักษณะสัญญาณ TTL

โดยลักษณะการทำงานจะเริ่มทำงานด้วยลอจิกเริ่ม(Start Bit) เป็นลอจิก 0 ตามด้วยข้อมูลขนาด 8-9บิตขึ้นอยู่กับการใช้งาน และลงท้ายด้วยลอจิกหยุด(Stop Bit) มีขนาดตั้งแต่ 0.5 , 1 ,1.5 ไปจนถึง 2 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

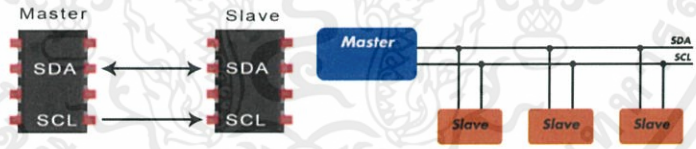


รูปที่ 2.7 ลักษณะข้อมูลในการส่งแบบ TTL

แต่ในอุปกรณ์อิเล็กทรอนิกส์สมัยใหม่จะใช้ระดับแรงดันของสัญญาณที่ 3.3V มีชื่อเรียกว่า (Low level Transistor Transistor Logic : LVTTL)

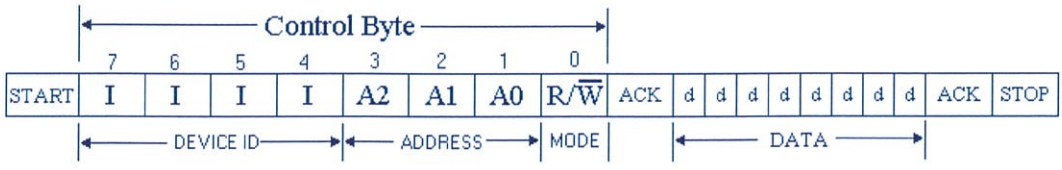
2.2.1.2 มาตรฐานการสื่อสารแบบไอทูซี (I2C)

I^2C = I2C Bus ย่อมาจาก Inter Integrate Circuit Bus (IIC) นิยมเรียกสั้นๆว่า I^2C BUS (ไอทูซีบัส) เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสาร ระหว่าง ไมโครคอนโทรลเลอร์ (MCU)การสื่อสารแบบไอทูซี(I2C) จะประกอบไปด้วยสายไฟ 2 เส้นคือ SDA และ SCK โดยจะมีการส่งข้อมูลตามจังหวะสัญญาณนาฬิกา โดยการสื่อสารจะถูกแยกเป็นสองส่วนคือผู้รับและผู้ส่ง (Host – Slave)



รูปที่ 2.8 การเชื่อมต่อสายสัญญาณแบบไอทูซี (I2C)

การสื่อสารแบบไอทูซี(I2C)มีส่วนช่วยให้ประหยัด Pin บน Microcontroller โดยมีการเชื่อมต่อ SDA และ SCK เข้ากับ Slave ทุกModuleจะสามารถแลกเปลี่ยนข้อมูลกันได้ทันที



รูปที่ 2.9 แสดงลักษณะการติดต่อสื่อสารแบบไอทูซี (I2C)

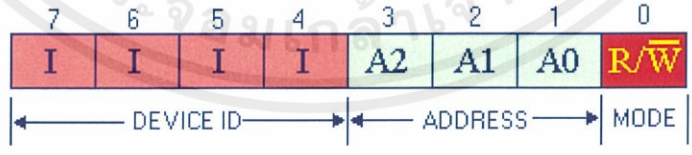
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับส่งข้อมูลแบบไอทูซี(I2C)เริ่มต้นด้วยการส่งสถานะเริ่มต้น(Start Condition)เพื่อแสดงการขอบัส แล้วตามด้วยรหัสควบคุม (Control Byte) ซึ่งเป็นรหัสประจำตัวของเซ็นเซอร์แต่ละตัวที่-ถูกเชื่อมต่อกับ มาสเตอร์ (Master) ในระบบ (ในที่นี้คือ Microcontroller ESP32) เปรียบเสมือนบ้านเลขที่ หลังจากมีการติดต่อจากมาสเตอร์(Master)ว่าต้องการข้อมูล Slave จะมีการส่งสถานะรับรู้(Acknowledge) เพื่อบอกให้มาสเตอร์ (Master) รับรู้ว่าข้อมูลที่ถูกส่งมีความถูกต้อง และสิ้นสุดกระบวนการสื่อสารด้วยสถานะสิ้นสุด (Stop Condition) เพื่อสิ้นสุดการใช้บัส

รูปที่ 2.10 ลักษณะสัญญาณเริ่มต้น-หยุด (Start-Stop Condition)

เมื่อมาสเตอร์(Master)ไม่มีการใช้งานบัสหรือไม่มีการร้องขอข้อมูลจาก Slave SDA และ SCL จะมีสถานะเป็น 1 ตลอดเวลา

เมื่อมาสเตอร์(Master)มีการใช้บัสหรือมีการร้องขอข้อมูลจาก Slave SDA จะมีการเปลี่ยนแปลง สถานะจาก 1 เป็น 0 ในขณะที่ SCL ยังคงสถานะเป็น 1 เมื่อสิ้นสุดกระบวนการส่งข้อมูล Master จำเป็นต้องมีการส่งสถานะสิ้นสุดคือ ให้ SDA เปลี่ยนจาก 0 เป็น 1 ในขณะที่ SCL มีค่าเป็น 1

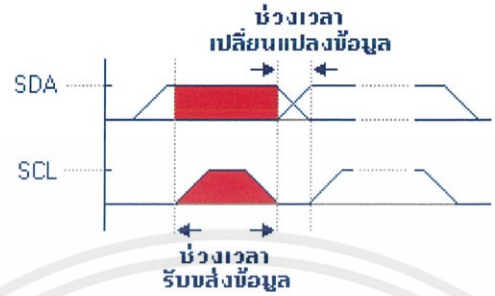


รูปที่ 2.11 บิตควบคุม (Mode)

รหัสควบคุมประกอบไปด้วยแอดเดรสประจำตัว (Address) ประกอบไปด้วย บิต 1-7 รหัสประจำตัวของอุปกรณ์ประกอบด้วยรหัสประจำตัวจากผู้ผลิต (Product ID) 4บิต (บิตที่ 4-7) ที่เปลี่ยนแปลงแก้ไขไม่ได้ และ Device Address 3 บิต (บิตที่ 1-3) ซึ่งผู้ใช้สามารถกำหนดเองได้ รวมแล้วเป็นรหัส 7 บิต ใช้ระบุชื่อของออุปกรณ์เพื่อให้มาสเตอร์(Master)เลือกข้อขอมูลได้ถูกต้องจากเซ็นเซอร์ (Sensor) แต่ละตัวโดยอุปกรณ์ที่ต่อพ่วงอยู่บนบัส(Bus) ไม่สามารถมี Address ซ้ำกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิตควบคุมการเขียน (Mode) บิต 0 เมื่อMaster ต้องการเขียนข้อมูลไปยังอุปกรณ์ก็กำหนดให้บิตนี้เป็น 0 และเมื่อต้องการอ่านข้อมูลจากอุปกรณ์จะกำหนดให้บิตนี้เป็น 1



รูปที่ 2.12 ลักษณะช่วงเวลาในกรรับส่งข้อมูล

Master และ Slave จะทำการรับส่งข้อมูลในที่มี SCL เป็น 1 และเปลี่ยนแปลงข้อมูลจะทำในสภาวะที่ SCL เป็น 0

ข้อเด่นของรูปแบบการสื่อสาร i2C คือสามารถเชื่อมต่ออุปกรณ์ชิพไอซีที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ โดยอุปกรณ์บนระบบบัสตัวหนึ่งอาจใช้แรงดันไฟฟ้า 5V และอุปกรณ์บนระบบบัสอีกตัวหนึ่งใช้แรงดันไฟฟ้า 12V การต่อร่วมกันบนระบบ I2C สามารถต่อพ่วงใช้งานอุปกรณ์ได้ในลักษณะเดียวกันกับ กรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงหรือแรงดันไฟฟ้าเท่ากัน นั้นคือให้ต่อสายข้อมูล SDA และสายสัญญาณนาฬิกา SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกันและต้องต่อตัวต้านทานพูลอัพ เข้ากับแรงดันไฟฟ้า 5V

การเขียนโปรแกรมเพื่อควบคุมอุปกรณ์ระบบ I2C

ในการเขียนโปรแกรมภาษาซีเพื่อให้ไมโครคอนโทรลเลอร์สามารถติดต่อควบคุมสั่งอุปกรณ์ระบบ I2C ได้นั้นต้องเริ่มต้นด้วยการเรียกใช้งานฟังก์ชั่น I2C (I2C.h) ซึ่งภายในจะบรรจุสภาวะของระบบ I2C เอาไว้ สภาวะของระบบ I2C ต่างๆมีดังนี้

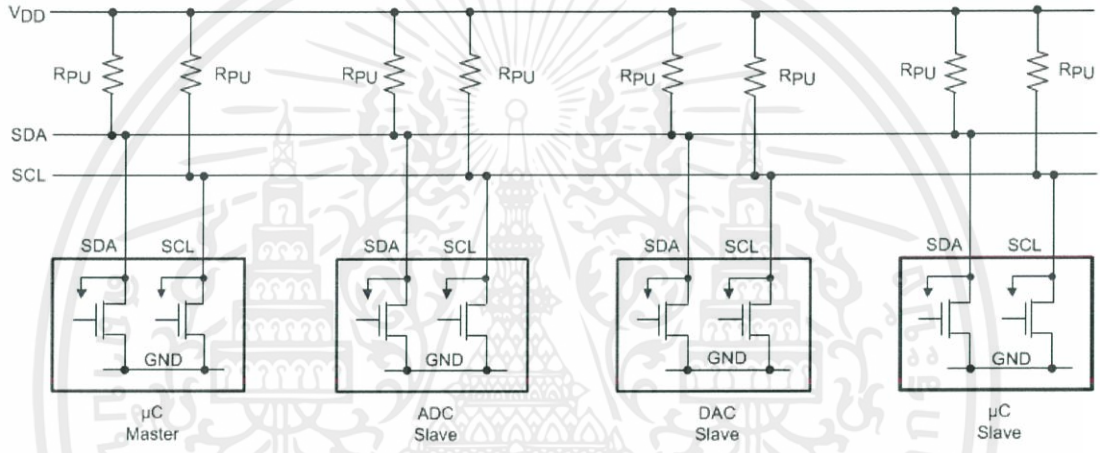
1. I2C_START เป็นส่วนฟังก์ชันโปรแกรมสำหรับสร้างสภาวะเริ่มต้นการติดต่อสื่อสารของระบบ I2C
2. I2C_STOP เป็นส่วนฟังก์ชันโปรแกรมสำหรับสร้างสภาวะหยุดติดต่อสื่อสารของระบบ I2C
3. I2C_WRDATA เป็นส่วนฟังก์ชันโปรแกรมสำหรับสร้างสภาวะส่งข้อมูลขนาด 1 ไบท์ (8บิต) โดยส่งข้อมูลทีละ 1 บิตลงในอุปกรณ์ระบบ I2C และมีการตรวจสอบสอบสัญญาณการตอบรับ (ACK) จากอุปกรณ์ Slave ด้วยการคืนค่าข้อมูล คือ “0” แสดง**มีการตอบรับ**จากอุปกรณ์Slave, “1” แสดง**ไม่มีการตอบรับ**จากอุปกรณ์ Slave

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. I2C_RDDATA เป็นส่วนฟังก์ชันโปรแกรมสำหรับสร้างสภาวะรับข้อมูลขนาด 1 ไบท์ (8บิต) โดยรับข้อมูลทีละ 1 บิตจากอุปกรณ์ I2C

5. I2C_CLK เป็นส่วนฟังก์ชันโปรแกรมสำหรับสร้างสภาวะสัญญาณนาฬิกา 1 ลูกคลื่น

สายข้อมูล SDA และสายสัญญาณนาฬิกา SCL เป็นสายสัญญาณแบบ 2 ทิศทาง (Bi-Directional Line) ให้อัตราการสื่อสารสูงถึง 100 กิโลบิตต่อวินาที โดยต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน +5V ไว้เพื่อให้สายมีสภาวะลอจิกสู.ในขณะที่ไม่มีการติดต่อใช้งาน ทั้งนี้เพื่อช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสองวงจรเอาท์พุตที่ต่ออยู่บนระบบ I2C ควรมีลักษณะเป็นวงจรเดรนเปิดหรือคอลเลคเตอร์เปิดดังแสดงในรูป แสดงวงจรเอาต์พุตของอุปกรณ์ระบบ I2C



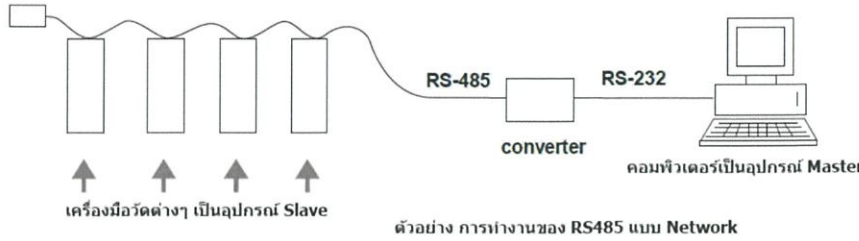
รูปที่ 2.13 แสดงวงจรเอาต์พุตของอุปกรณ์ระบบ I2C

2.2.1.3 การรับส่งข้อมูลผ่านมาตรฐานการสื่อสารอาเอสสี่แปดห้า (RS485 Communication)

มาตรฐานการสื่อสารแบบ RS485 คือมาตรฐานการสื่อสารข้อมูลดิจิตอลแบบอนุกรมมีลักษณะการส่งและการส่งข้อมูลแบบเลือกรับหรือส่งได้อย่างใดอย่างหนึ่งเท่านั้นในช่วงเวลาเดียวกัน(Half Duplex)ข้อดีของมาตรฐานการสื่อสารแบบ RS485 คือสามารถเชื่อมต่อกับอุปกรณ์ต่อพ่วง(Slave)ได้สูงสุดถึง 256 ตัวแต่ แต่จำเป็นต้องเชื่อมต่อกับอุปกรณ์ต่อพ่วงที่มีความต้านทานภายในที่มีค่าความต้านทานมากกว่า

1 MOhm เท่านั้น โดยการเรียกข้อมูลจากอุปกรณ์ต่อพ่วง (Slave) จะใช้ระบบเรียกชื่อแอดเดรส (Address) ประจำตัวของอุปกรณ์ต่อพ่วง (Slave) แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ลักษณะการใช้งานมาตรฐานการสื่อสารแบบ RS485

การส่งข้อมูลดิจิตอลผ่านมาตรฐานการเชื่อมต่อแบบ RS4850 จะถูกส่งผ่านเคเบิลเพียงสองเส้นเท่านั้น โดย A และ B เป็นตัวบอกระดับแรงดันลอจิกดิจิตอล โดยใช้ความแตกต่างของแรงดันไฟฟ้าระหว่างขั้ว A และ B เป็นตัวบ่งบอกลอจิกของสัญญาณดิจิตอล

รูปที่ 2.15 ลักษณะการเชื่อมต่อสายไฟในมาตรฐานการสื่อสารแบบ RS485

เมื่อระดับแรงดันที่จุด A มีค่าน้อยกว่าที่จุด B กำหนดให้สัญญาณดิจิตอลเป็นลอจิก 1 เมื่อระดับแรงดันที่จุด A มีค่ามากกว่าที่จุด B กำหนดให้สัญญาณดิจิตอลเป็นลอจิก 0

2.2.2 การติดต่อสื่อสารแบบไร้สาย

ถูกนำเข้ามาใช้เพื่อเชื่อมต่อกับอุปกรณ์ต่อพ่วงระหว่าง IOT Gateway Platform และ IOT Gateway Platform กับระบบเซิฟเวอร์ (Server) หรือระบบคลาวด์ (Cloud) เป็นหลัก สามารถติดต่อสื่อสารผ่านทาง Bluetooth,Wifi,Wireless

2.2.2.1 Bluetooth

ถูกนำมาใช้เพื่ออัพเดตซอฟแวร์สำหรับผู้ผลิตเท่านั้น โดยฟังก์ชั่นดังกล่าวเป็นความสารมารถของอุปกรณ์ไมโครโปรเซสเซอร์(Microprocessor) ESP32Nโดยการสื่อสารแบบบลูทูธ(Bluetooth)จะสื่อสารผ่านคลื่นวิทยุความถี่ 2.4Ghz

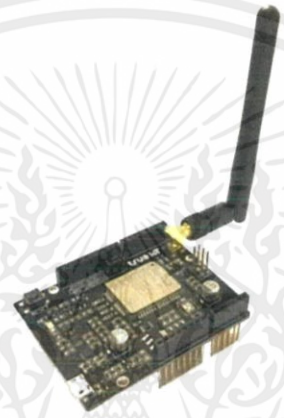
2.2.2.2 Wifi

ถูกนำมาใช้เพื่อติดต่อระหว่าง IOT Gateway Platform และเซิฟเวอร์ (Server) หรือระบบคลาวด์ (Cloud) เท่านั้น โดยการเชื่อมต่อ Wifi เป็นความสามารถของไมโครโปรเซสเซอร์ (Microprocessor) ESP32 โดยจำเป็นต้องมีผู้ให้บริการอินเตอร์เน็ตในรัศมี 5เมตร หรืออยู่ในวงแลนด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.3 Narrow Band

โมดูล Narrow Band เป็นโมดูลที่ถูกสร้างและพัฒนาจากหน่วยงานเอกชน “TRUE “ ซึ่งมีจุดประสงค์เพื่อตอบสนองงานวิจัยและอุปกรณ์เกี่ยวกับ IOT โดยมีการเชื่อมต่อแลกเปลี่ยนสื่อสารข้อมูลกับไมโครโปรเซสเซอร์ (Microprocessor) ESP32 ได้โดยตรงผ่านมาตรฐานการสื่อสารแบบ TTL ก่อนจะส่งข้อมูลต่างๆที่ได้รับจากไมโครโปรเซสเซอร์ (Microprocessor) ESP32 ไปสู่เซิฟเวอร์ (Server)หรือระบบคลาวด์ (Cloud) ด้วยคลื่นโทรศัพท์ 3G



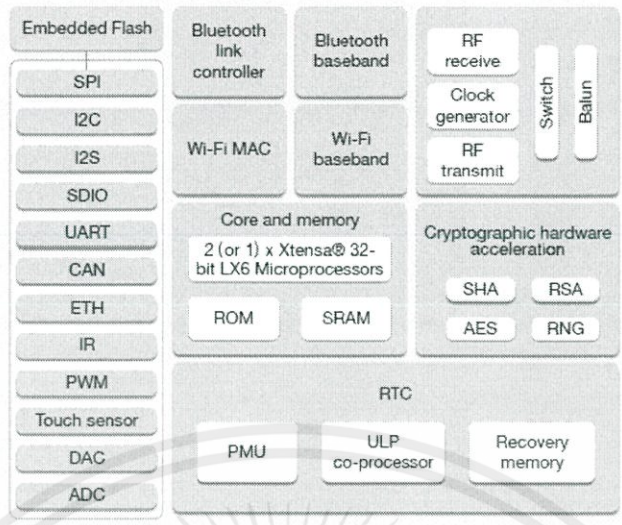
รูปที่ 2.16 โมดูล Narrow Band จากผู้ให้บริการสัญญาณโทรศัพท์ TRUE

2.3 ไมโครโปรเซสเซอร์ ESP32

ESP32 เป็นไอซีไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ WIFIโดยเชื่อมต่อกับเราเตอร์ไวไฟได้โดยตรง และ Bluetooth 4.2 BLE (Bluetooth Low Energy) ผลิตโดยบริษัท Espressif ด้วยฟังก์ชั่นที่ถูกบรรจุลงมาใน ESP32 จึงสามารถนำไปประยุกต์ใช้กับหลากหลายแอปพริเคชั่น เช่น เน็ตเวิร์คเซ็นเซอร์ที่มีการทำงานหลายภาระงานในเวลาเดียวกันแต่ยังคงมีการกินพลังงานต่ำ (Low-Power Sensor Networks) การประยุกต์ใช้งานกับงานทางด้านเสียง เป็นอุปกรณ์ในการแปลงไฟล์เพลงที่ถูกจัดเก็บให้อยู่ในไฟล์ดิจิตอลเป็นสัญญานอนาลอกแต่ยังคงได้คุณภาพเสียงที่ดีอยู่

ซีพียูใช้สถาปัตยกรรม Tensilica LX6 2 แกนสมอง สถาปัตยกรรมนี้ใช้งาน ULP(Ultra Low Power Processor) สามารถ ใช้งาน Deep Sleep 2 Core Processor แต่ยังคงเชื่อมต่อ Wifi ได้และปลุกตื่นได้ ใช้การแสเพียง 150 micro amp และสามารถปรับความเร็วในการทำงานได้ตั้งแต่ 80Mhz จนถึง 240Mhz ทำให้ประหยัดพลังงานมาก เนื่องจากซีพียูมีสองแกนสมอง (Dual Core) จึงมีฟังก์ชั่นพิเศษให้เลือกใช้ **คือ Free RTOS** ซึ่งเป็นการทำงานหลายภาระงานพร้อมกันในเวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

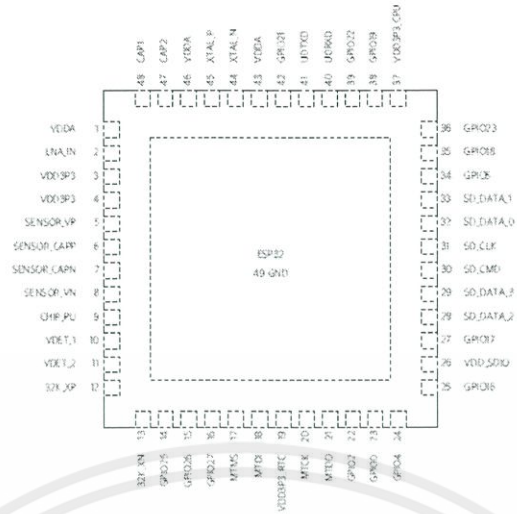


รูปที่2.17 Function Block Diagram

ฟังก์ชันการใช้งานของ ESP32 รองรับการเชื่อมต่อบัสต่างๆ ดังนี้

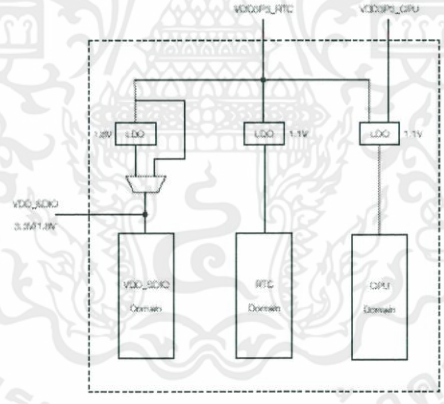
- มี GPIO จำนวน 32 ช่อง
- รองรับ UART จำนวน 3 ช่อง
- รองรับ SPI จำนวน 3 ช่อง
- รองรับ I2C จำนวน 2 ช่อง
- รองรับ ADC จำนวน 12 ช่อง
- รองรับ DAC จำนวน 2 ช่อง
- รองรับ I2C จำนวน 2 ช่อง
- รองรับ PWM / Timer ทุกช่อง
- รองรับการเชื่อมต่อกับ SD-Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 ESP32 Pin lay out

การเชื่อมต่อแหล่งจ่ายไฟให้กับ ESP32 จำเป็นต่อเชื่อมต่อแหล่งจ่ายไฟจำนวนสามขาคือ VDD3P3_RTC เพื่อจ่ายไฟให้แก่ CPU และ RTC ,VDD3p3_CPU เพื่อจ่ายไฟให้แก่ CPU ,VDD_SDIO จ่ายไฟให้แก่ **LDO**



รูปที่ 2.19 ESP32 Power Scheme

การใช้งานพอร์ตในลักษณะอินพุต/เอาต์พุต จำเป็นต้องเลือกใช้ขาของอุปกรณ์ให้เหมาะสมกับฟังก์ชันที่ต้องการใช้งาน (จำเป็นต้องศึกษาจาก Data Sheet) ซึ่งในบางขาของอุปกรณ์อาจจะมีไม่มีฟังก์ชันที่ต้องการใช้งานเช่น ADC,DAC ,PWM เป็นต้น โดยการเลือกใช้ขาของอุปกรณ์ให้เป็นอินพุต/เอาต์พุตสามารถกำหนดค่าจากซอฟแวร์ การอ่านค่าจาก GPIO สามารถอ่านจาก**รีจิสเตอร์**ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การวิเคราะห์และการออกแบบ

ในบทนี้จะอธิบายเกี่ยวกับหลักการออกแบบ IoT Gateway platform โดยหลักการการออกแบบจะคำนึงถึงผู้ใช้เป็นหลัก ให้มีลักษณะยืดหยุ่นในการใช้งาน สามารถนำไปประยุกต์ใช้ได้ตั้งแต่นักศึกษานักพัฒนาไปจนถึงระดับอุตสาหกรรม การออกแบบจะขึ้นอยู่กับความท้าทายสี่ข้อคือ 1. ต้นทุนต่ำ 2.การกินพลังงานต่ำ 3.ยืดหยุ่น 4.ขนาดเล็ก

รูปที่ 3.1 จุดมุ่งหมายการออกแบบ

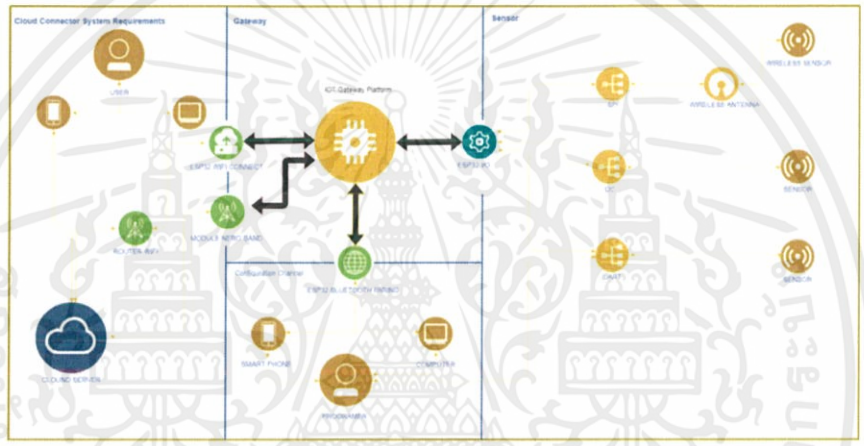
Computer Simulation ถูกนำมาใช้ในการออกแบบและประมวลผลก่อนทำชิ้นงานต้นแบบเพื่อลดต้นทุนในงานวิจัย การวางแผนงานที่มีประสิทธิภาพตลอดจนมีที่ปรึกษาที่เชี่ยวชาญย่อมลดระยะเวลาในการวิจัยและศึกษาค้นคว้าข้อมูลได้ถูกต้องตรงประเด็น

หนึ่งในความท้าทายในการออกแบบวงจรอิเล็กทรอนิกส์ของนักอิเล็กทรอนิกส์ทั่วโลกคือความต้องการลดการกินพลังงานให้มากที่สุด หนึ่งในวิธีแก้ปัญหาคือใช้เทคโนโลยีเซอเฟสเมาส์ (surface mount technology) ซึ่งมีขนาดเล็กและมีพลังงานสูญเสียในตัวอุปกรณ์ค่อนข้างต่ำ IoT Gateway Platform Prototype ยังสามารถลดการกินพลังงานด้วยความสามารถของ ESP32 Platform ด้วยโหมดประหยัดพลังงาน สามารถใช้โหมดประหยัดพลังงานด้วยวิธีทาง Programming

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การออกแบบการทำงานของระบบ

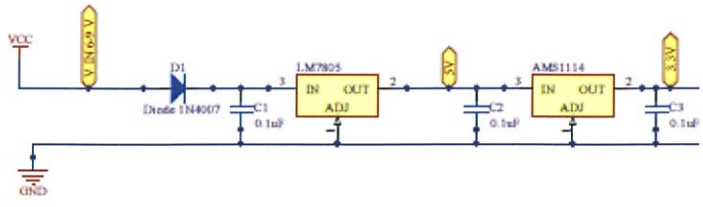
IOT Gateway platform ประกอบไปด้วยไมโครโปรเซสเซอร์(Microprocessor) ESP32 ซึ่งเป็นหน่วยประมวลผลหลักของโมดูล โดยไมโครโปรเซสเซอร์ ESP32 จะติดต่อสื่อสารกับโมดูลเชื่อมต่อภายนอก IOT Gateway platform ผ่านมาตรฐานการสื่อสารแบบอนุกรมในรูปแบบอะซิงโครนัสจะใช้การสื่อสารผ่านไอทูซี (I2C) ในรูปแบบซิงโครนัสใช้การสื่อสารผ่านซีเรียล(Serial Port) ผ่านการสื่อสารแบบไร้สายโดยการใช้โมดูลสื่อสารแบบไวเรส(Wireless) IOT Gateway platform จะติดต่อสื่อสารกับเซิฟเวอร์ด้วยระบบ Wifi และผ่านทางผู้ให้บริการอินเตอร์เน็ตของบริษัทเอกชน ผ่าน Nero Band



รูปที่ 3.2 ช่องทางการสื่อสารระหว่างเซ็นเซอร์ผู้ใช้งานและระบบคลาวด์

3.2 การออกแบบภาคจ่ายไฟให้กับระบบ

ภาคจ่ายไฟถือเป็นวงจรที่เป็นหัวใจสำคัญในวงจรอิเล็กทรอนิกส์ทุกวงจร ในการออกแบบภาคจ่ายไฟจะคำนึงถึงคุณภาพของกระแสไฟฟ้า ขนาดของอุปกรณ์อิเล็กทรอนิกส์ไม่ให้มีขนาดใหญ่จนเกินไป และปลอดภัยต่อผู้ใช้งานเป็นหลัก



รูปที่ 3.3 วงจรภาคจ่ายไฟที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 หลักการทำงานของภาคจ่ายไฟ

ไดโอด D1 มีหน้าที่ป้องกันการจ่ายไฟสลับขั้วของผู้ใช้(สำหรับผู้ใช้เริ่มต้น) ซึ่งป้อกันความเสียหายที่อาจเกิดขึ้นจากความผิดพลาดได้ โดยผู้ออกแบบไม่มีจุดประสงค์ให้ D1 ทำหน้าที่เรียงกระแส(Rectifier) เพราะฉะนั้นไม่ควรใช้ไฟฟ้ากระแสสลับจ่ายให้กับวงจร

ตัวเก็บประจุ C1 มีหน้าที่ลดแรงดันกระเพื่อม(Ripple) ให้มีขนาดที่รับได้ ซึ่ง Ripple มักจะมีผลกระทบกับวงจรอิเล็กทรอนิกส์ที่มีขนาดเล็กและระบบประมวลผล(Microprocessor)

3.2.2 การคำนวณค่าความจุของตัวเก็บประจุในภาคจ่ายไฟ

การออกแบบถูกอ้างอิงจาคู่มือการใช้งานของผู้ผลิต(Datasheet) Microcontroller ESP32

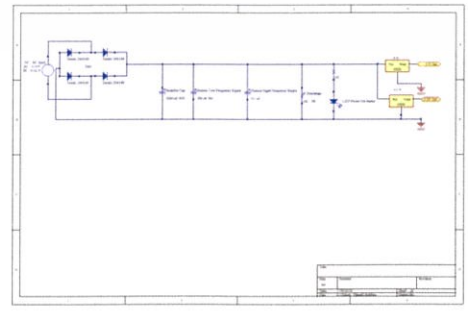
5.2 Recommended Operating Conditions

Table 8: Recommended Operating Conditions

Symbol	Parameter	Min	Typical	Max	Unit
VDDA, VDD3P3, VDD3P3_RTC [1], VDD_SDIO (3.3V mode)[2]	-	2.3	3.3	3.6	V
VDD3P3_CPU	-	1.8	3.3	3.6	V
I_{VDD}	Current delivered by external power supply	0.5	-	-	A
T [3]	Operating temperature	–40	-	125	°C

รูปที่ 3.4 จุดทำงานของ ESP32 ที่ผู้ใช้แนะนำ

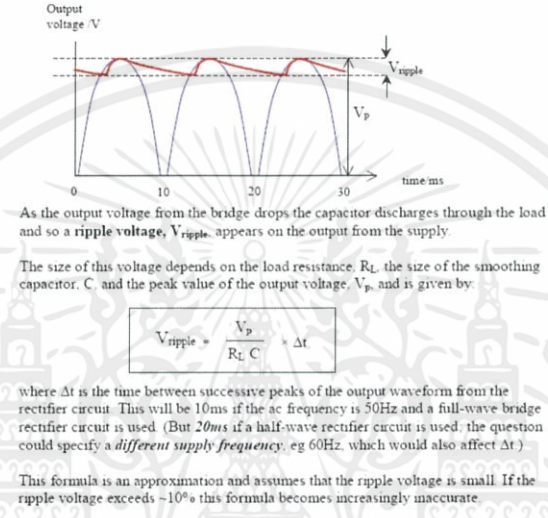
ตัวเก็บประจุที่ถูกใส่เข้าไปในวงจรมีจุดประสงค์เพื่อต้องการลดแรงดันกระเพื่อมที่มากับไฟฟ้ากระแสตรง และทำให้ได้ไฟฟ้ากระแสตรงที่เรียบที่สุดมีช่วงสวิงอยู่ในช่วง Operating Voltage เท่านั้น โดยค่าของตัวเก็บประจุอาจถูกคำนวณจากสมการด้านล่างนี้



รูปที่ 3.5 แสดงวงจร Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่แสดงอยู่ในสกิมเมตริกซ์ เป็นค่าอุปกรณ์จริงที่ใช้ในการทดสอบ บริดไดโอด มีหน้าที่แปลงกระแสสลับเป็นกระแสตรง ซึ่งมีผลดีคือสามารถเป็นทางเลือกให้สำหรับผู้ใช้ได้ว่าต้องการเลือกแหล่งจ่ายแบบใด (ใช้กระแสสลับก็ได้ใช้กระแสตรงก็ได้) ซึ่งแน่นอนว่า C rectifier จำเป็นต้องมีขนาดที่ใช้เพื่อสามารถทำหน้าที่ Rectifier (เรียงกระแส) ได้สมบูรณ์ ซึ่งขัดต่อเป้าหมายเรื่องขนาดของงานวิจัย



รูปที่ 3.6 สมการในการคำนวณหาค่าตัวเก็บประจุในภาคจ่ายไฟ

จะเห็นว่าภายในสมการขนาดของ C มีผลทำให้ V ripple ลดน้อยลง (ยิ่งมาก Ripple ยิ่งน้อย)

ภายในชิ้นงานจริงจะมีการปรับแก้คือ ไม่ใช้วงจรบริดไดโอดอีกต่อไป ใส่เพียง Shockey Diode เพื่อป้องกันการใส่สายไฟสลับขั้วเท่านั้น (ควรใช้เฉพาะไฟกระแสตรง) C rectifier จึงไม่จำเป็นต้องมีขนาดใหญ่อีกต่อไป อาจจะใช้เพียง 220uF/50V SMD ก็เพียงพอ

แนะนำให้ใช้แรงดันไฟเลี้ยงในช่วย 9-12 V เพื่อจะทำให้ Regulator 5V และ 3V สามารถ Regulated แรงดันออกมาได้ถูกต้อง และไม่เกิด Power Loss มากจนเกินไป

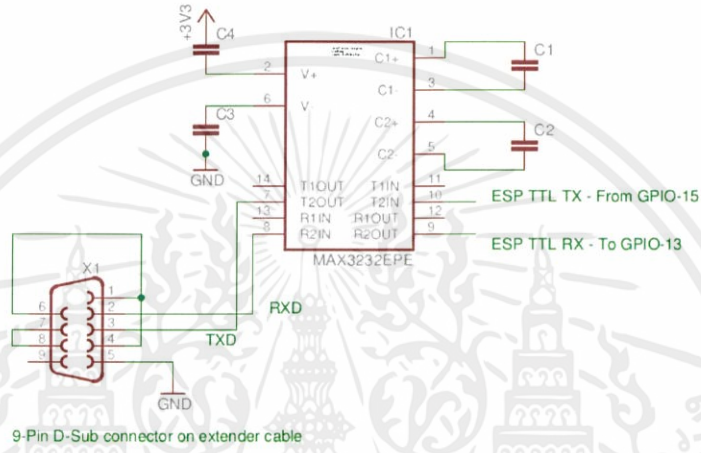
3V สำหรับเป็นแหล่งจ่ายพลังงานหลักให้ ESP32 และแหล่งจ่ายสำหรับภายนอกบอร์ด 5V สำหรับบางโมดูลภายใน Gateway และเป็นแหล่งจ่ายไฟสำหรับส่วนต่อขยายนอกบอร์ด

โดยกระแสของแหล่งจ่ายที่แนะนำ ถูกคำนวณใหม่ภายหลัง เมื่อมีการเดินระบบแบบเต็มกำลังและวัดกระแสไฟฟ้าที่มากที่สุดขณะเครื่องทำงาน ควรมี Safety Factor 25 เปอร์เซ็นต์ (ประมาณ 1 แอมป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การแปลงมาตรฐานการสื่อสารแบบอนุกรมเป็นมาตรฐานการสื่อสารแบบ R485 (TTL To RS232)

โดยโมดูลนี้ถูกนำมาใช้เพื่อต้องการส่งข้อมูลได้ไกลขึ้นและสามารถใช้กับเครื่องจักรในอุตสาหกรรมได้ โดยสามารถใช้วงจรรวม(IC : integrated circuit) เบอร์ Max485ซึ่งเป็นอุปกรณ์อิเล็กทรอนิกส์ที่มีหน้าที่เพื่อแปลงข้อมูลที่มีการสื่อสารแบบ TTl เป็นแบบ RS485 ได้



รูปที่ 3.7 รูปวงจร Serial to TTL

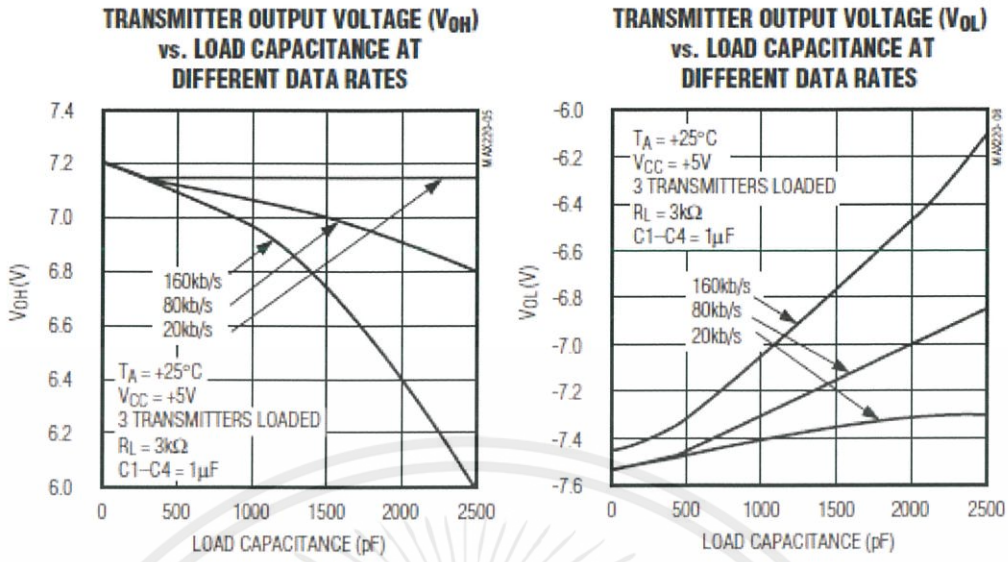
อุปกรณ์ตัวเก็บประจุ C1-C4 เลือกใช้ค่าความจุตามที่ผู้ผลิตแนะนำ

Electrical Characteristics—MAX223/MAX230–MAX241 (continued)

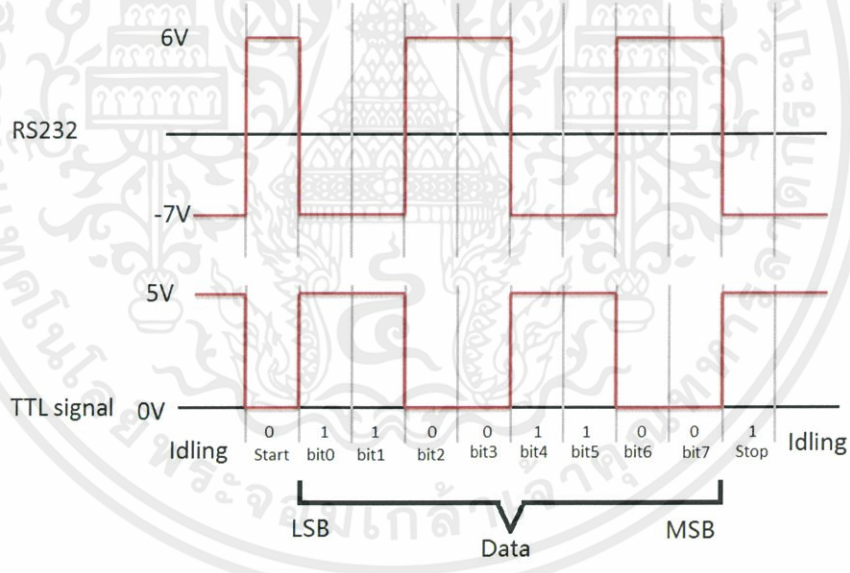
(MAX223/230/232/234/236/237/238/240/241: V_{CC} = +5V ±10%; MAX233/MAX235: V_{CC} = +5V ±5%; C1–C4 = 1.0µF; MAX231/MAX239: V_{CC} = +5V ±10%; V+ = +7.5V to +13.2V; T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 5)

รูปที่ 3.8 ค่าความจุของตัวเก็บประจุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ค่าความจุของตัวเก็บประจุที่มีผลต่อการออกแบบใช้งานของ MAX232



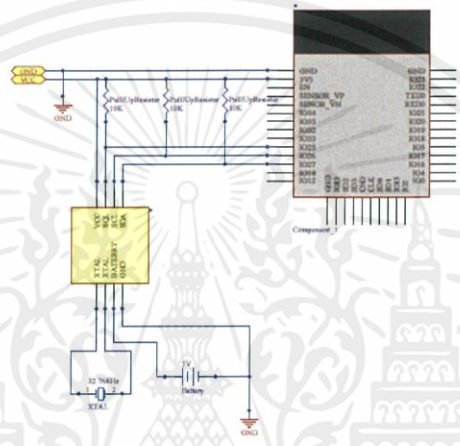
รูปที่ 3.10 ลักษณะแรงดันของลอจิกของมาตรฐานการสื่อสารแบบ TTL และ RS232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 วงจรกำเนิดสัญญาณนาฬิกา (RTC:Real Time Clock)

วงจรกำเนิดสัญญาณนาฬิกาถูกนำมาใช้ใน IOT Gateway Platform เพื่อสร้างเวลาปัจจุบันให้แก่วงจรแม้ว่า IOT Gateway Platform จะถูกตัดกระแสไฟฟ้า โดยสัญญาณนาฬิกาที่ถูกสร้างขึ้นอาจนำมาใช้เพื่อกำหนดจังหวะหรือเวลาในการส่งข้อมูลระหว่างโมดูล

ซึ่งสัญญาณนาฬิกาจะถูกสร้างขึ้นมาจากวงจรรวม (IC: integrated circuit) เบอร์ DS1307และมีการส่งเวลาที่ถูกสร้างขึ้นให้กับหน่วยประมวลผล(Microprocessor)ESP32 ผ่านมาตรฐานการสื่อสารไอทูซี(I2c)



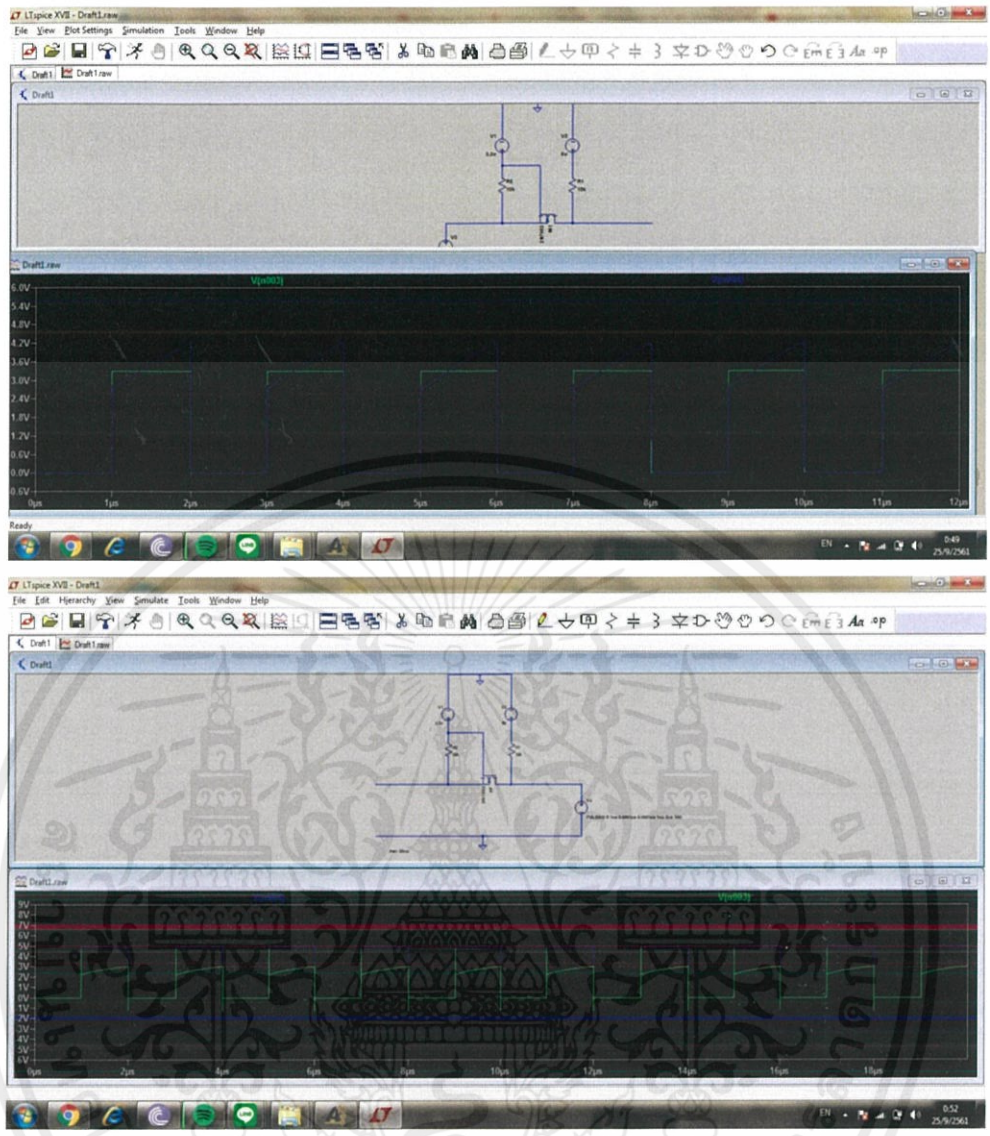
รูปที่ 3.11 รูปวงจร RTC

3.5 วงจรลดระดับแรงดันของสัญญาณดิจิตอล (Logic converter)

วงจรนี้มีส่วนป้องกันความเสียหายที่อาจเกิดแก่ส่วนควบคุม โดยสัญญาณดิจิตอลที่เข้าออก ESP32 จะเป็นลักษณะสัญญาณเพาที่มีระดับแรงดัน 3.3 V หากมีสัญญาณดิจิตอลที่มีระดับแรงดันเกิน 3.3v อาจส่งผลให้ไมโครโปรเซสเซอร์(Microprocessor)ESP32 ได้รับความเสียหายได้ เหตุการณ์ดังกล่าวอาจเกิดขึ้นได้ในขณะมีการเชื่อมต่อกับโมดูลภายนอกที่มีระดับสัญญาณที่ 5V

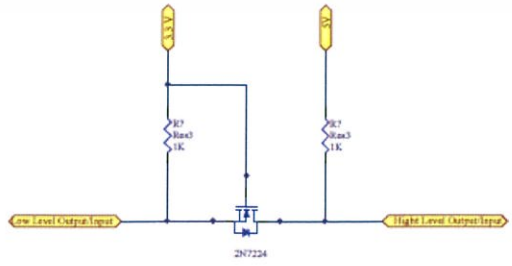
วงจรลดระดับแรงดันของสัญญาณดิจิตอลมีหน้าที่แปลงระดับสัญญาณขาออก (Transmission:TX) ที่มีระดับแรงดัน 3.3V เป็น 5V และลดระดับแรงดันขาเข้า (Recive:RX)จาก 5V เป็น 3.3V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ลักษณะระดับสัญญาณเข้าและขาออกเมื่อผ่านวงจร Logic Converter

วงจรถูกสร้างโดยใช้คุณสมบัติของ N-channel mosfet (N-mos Characteristic) มาสร้างเป็นสวิตช์ ทำให้มีเกิดสัญญาณเพาศักดิ์สูง (Hight Level Logic) เกิดขึ้นเฉพาะเมื่อมีสัญญาณเพาศักดิ์ต่ำ(Low Level Logic) เข้ามาเท่านั้น



รูปที่ 3.13 รูปวงจร Logic converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

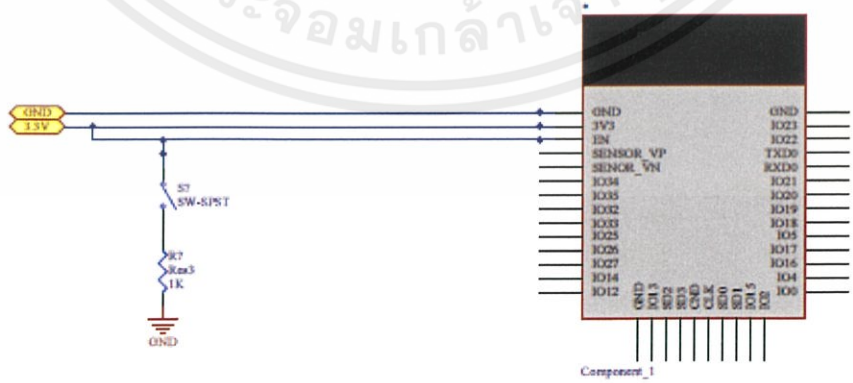
3.6 สัญญาณเข้าออกภายในไมโครโปรเซสเซอร์ (Microprocessor) ESP32

รูปที่ 3.14 ESP 32

เนื่องจาก ESP32 เป็นหน่วยประมวลผลหลักเราจำเป็นต้องกำหนดและบริหารการใช้ขาของตัวอุปกรณ์(General Purpose Input Output : GPIO)เป็นอย่างดี ซึ่งการเลือกใช้ขาของอุปกรณ์สามารถ พิจารณาได้จากคำแนะนำของผู้ผลิต(Datasheet)

3.6.1 รีสตาร์ท (Restart)

กระบวนการรีสตาร์ท(Restart)ของ ESP32 จะเกิดขึ้นเมื่อมีสัญญาณลอจิก 1 เข้ามาที่ EN (Enablepin) ขา RST(RESET) เป็นขาที่ทำหน้าที่ในการกำหนดให้ไมโครคอนโทรลเลอร์เข้าสู่สภาวะเริ่มต้นการทำงานใหม่โดยปกติที่ขารีเซตจะมีสภาวะทางลอจิกเป็น 0 และถ้าป้อนสภาวะทางลอจิก 1 เป็นเวลาไม่น้อยกว่า 2 **แมชชีนไซเกิล** ก็จะทำให้ไมโครคอนโทรลเลอร์เกิดสภาวะรีเซตขึ้น



รูปที่3.15 วงจรรีสตาร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสวิตซ์ W1 มีหน้าที่ในการตัด – ต่อวงจร ทำให้ที่ขา EN (Enable pin) เป็นลอจิก 1 (Logic High) เมื่อวงจรเปิด และเป็นลอจิก 0 (Logic Low) เมื่อวงจรปิด

2. PIN DEFINITIONS

Name	No.	Type	Function
CHIP_PU	9	I	High: On; enables the chip Low: Off; resets the chip Note: Do not leave the CHIP_PU pin floating.
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4

รูปที่ 3.16 Enable Pin / Resets Pin

ซึ่งผู้ผลิตได้กำหนดไว้ชัดเจนภายในคู่มือการใช้งาน ควรกำหนดลอจิกอะไรเพื่อให้ MCU ทำงานได้ตามความต้องการ ตัวต้านทาน R1 มีหน้าที่ป้องกันไม่ให้กระแสไหลลงกราวด์ (GND)อย่างฉับพลัน ค่าความต้านทานที่ควรเลือกใช้ควรมีค่าเยอะประมาณหนึ่ง

$$v = IR$$

รูปที่ 3.17 สมการกฎของโอห์ม

จากสมการข้างต้น กำหนดให้ V คือความต่างศักดิ์ I คือกระแสไฟฟ้า R คือค่าความต้านทาน หากแทนค่าความต้านให้มีค่าน้อยมาก (เป็น 0) จะทำให้มีกระแสที่ไหลลงกราวด์มีค่าเป็นอนันต์ อาจทำให้เกิดความเสียหายแก่วงจรภาคจ่ายไฟได้

3.6.2 การอัพโหลดซอฟแวร์ (Boots Loader)

ไมโครโปรเซสเซอร์(Microprocessor)ESP32 จะยอมให้มีการอัพโหลดซอฟแวร์เข้าสู่เมมโมรี่(Memory)ภายในได้ต่อเมื่อขาของอุปกรณ์ช่อง IO0 เป็นลอจิก 0 (Logic Low)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. PIN DEFINITIONS

Each strapping pin is connected to its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impedance, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or use the host MCU's GPIOs to control the voltage level of these pins when powering on ESP32.

After reset, the strapping pins work as normal-function pins.

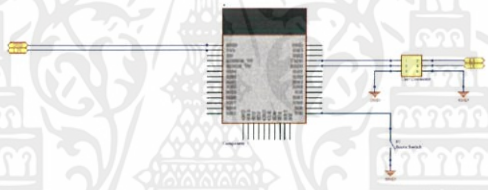
Refer to Table 2 for a detailed boot-mode configuration by strapping pins.

Table 2: Strapping Pins

Voltage of Internal LDO (VDD_SDIO)			
Pin	Default	3.3V	1.8V
MTDI	Pull-down	0	1
Booting Mode			
Pin	Default	SPI Boot	Download Boot
GPIO0	Pull-up	1	0
GPIO2	Pull-down	Don't-care	0

รูปที่ 3.18 Data Sheet แสดงช่อง IO0

จากคู่มือการใช้งานของผู้ผลิต (Datasheet) กล่าวไว้ชัดเจนว่าอุปกรณ์จะตอบสนองเมื่อมีลอจิกเป็น 0 (Logic Low) (Active Low)

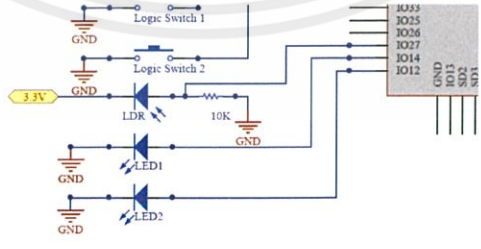


รูปที่ 3.19 รูปวงจร Boots Loader

สวิตช์ S2 และตัวต้านทาน R2 ที่อยู่ภายในวงจรทำหน้าที่เช่นเดียวกับสวิตช์S1และ R1 ใน วงจรรีเซต

3.6.3 หลอดไฟแสดงสถานะ

IOT Gateway Platform จำเป็นต้องมี LED (led light emitting diode) แสดงสถานะเพื่อใช้บ่งบอกสถานะ การส่งข้อมูลตลอดจนสถานะของฟังก์ชันบลูทูธ (Bluetooth)



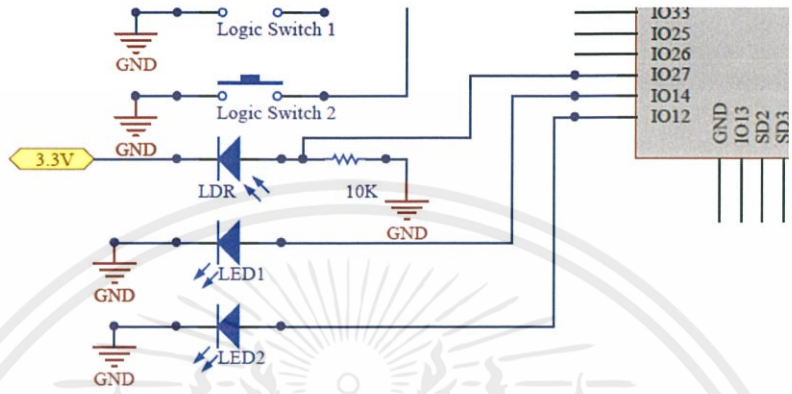
รูปที่ 3.20วงจรแสดงจุดเชื่อมต่อ LED

ออกแบบให้ LED ถูกควบคุมด้วย GPIO 14 , GPIO 12 ของ Microcontroller ESP32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.4 การอ่านค่าจากตัวต้านทานที่มีค่าเปลี่ยนแปลงตามแสง

ตัวต้านทานเปลี่ยนค่าตามแสดงหรือ LDR (Light Dependent Resistor) จะมีค่าความต้านทานเปลี่ยนไปตามแสงที่ตกกระทบ



รูปที่ 3.21ลักษณะวงจร LDR

การอ่านค่าของ LDR ด้วยไมโครโปรเซสเซอร์(Microprocessor)ESP32 จะใช้หลักการอ่านค่าแรงดันที่เปลี่ยนไปที่โหนด (Node1) ซึ่งอาศัยหลักการแบ่งแรงดัน (Voltage Devider) วงจรถูกออกแบบให้สามารถอ่านค่าได้จาก GPIO27

```
int LDR_Pin = A0; //analog pin 0

void setup(){
Serial.begin(9600);
}

void loop(){
int LDRReading = analogRead(LDR_Pin);

Serial.println(LDRReading);
delay(250);
}
```

รูปที่ 3.22 อ่านค่าจาก LDR

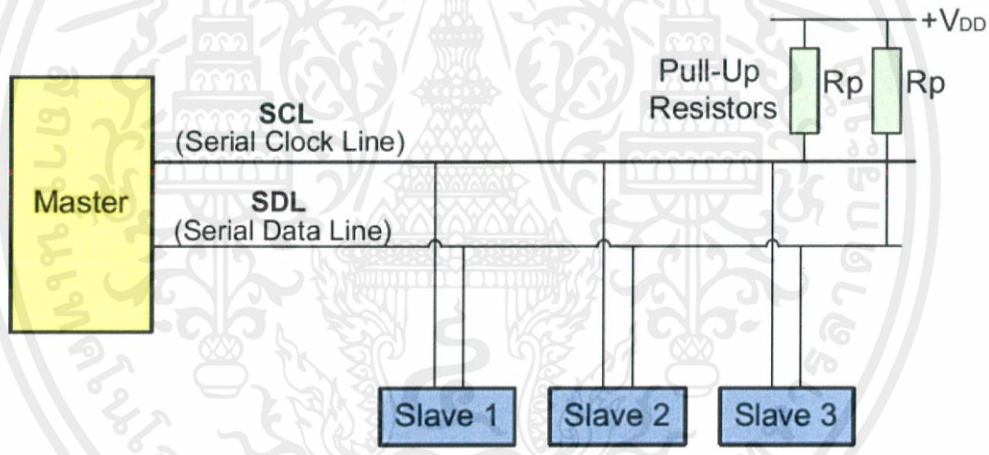
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าใช้คำสั่ง Analog read เพื่ออ่านค่าแรงดันไฟฟ้าที่มีการเพิ่มลดเป็นลักษณะสัญญาณอนาล็อก (Analog)

วงจรอ่านค่าความต้านทานที่เปลี่ยนแปลงตามแสงถูกใส่มาใน IOT Gateway platform เพื่อให้ยืดหยุ่นต่อผู้ใช้งาน เช่นสามารถทดสอบการอ่านข้อมูลจากตัวต้านทานเปลี่ยนค่าตามแสงแล้วส่งข้อมูลสู่เซิฟเวอร์เพื่อทดสอบการส่งข้อมูล หรือ ใช้เพื่อเป็นเซ็นเซอร์วัดสภาพแวดล้อมขึ้นอยู่กับผู้ใช้จะประยุกต์

3.6.5 การส่งผ่านข้อมูลด้วยมาตรฐานการสื่อสารไอทูซี (I2C)

การสื่อสาร I2C ใช้สายส่งดิจิตอลแบบ Bidirectional Open-drain line ซึ่งสามารถรับส่งข้อมูลได้ในเส้นเดียวกับ จำนวน 2 ชุด คือ Serial Data Line (SDA) ใช้ทำหน้าที่ส่งข้อมูล และ Serial Clock Line (SCL) ทำหน้าที่ส่งสัญญาณนาฬิกา ความเร็วของการรับส่งข้อมูลนั้นขึ้นอยู่กับความถี่ของสัญญาณนาฬิกาในสาย



รูปที่ 3.23 แสดงโครงสร้างของการสื่อสารแบบไอทูซี(I2C)

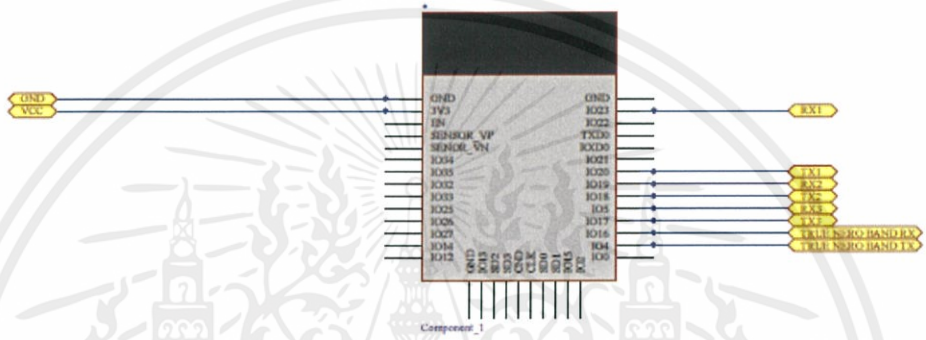
จากภาพแสดงโครงสร้างการสื่อสาร ESP32 จะถูกมองให้เป็น Master ส่วนโมดูลย่อยอื่นที่ต้องการติดต่อสื่อสารกับ ESP32 จะถูกมองให้เป็น Slave ซึ่งโมดูลย่อยอื่นจะมีแอดเดรสประจำตัว(Address) ซึ่งเราจะสามารถอ่านข้อมูลจากโมดูลย่อยด้วยวิธีการทางซอฟแวร์ โดยการโปรแกรมมิ่งให้ ESP32 อ่านค่าจากแอดเดรส(Address)นั้นๆ

ไมโครโปรเซสเซอร์(Microprocessor)ESP32 จะเป็นตัวกำหนด Serial Data Line (SDA)และ Serial Clock Line (SCL) ให้กับโมดูลย่อยตัวอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.6 การส่งผ่านข้อมูลด้วยรูปแบบการสื่อสารซีเรียล (Serial,TTL)

การส่งผ่านข้อมูลเพื่อติดต่อสื่อสารกับโมดูลย่อยด้วยมาตรฐานการสื่อสารแบบซีเรียล (Serialport) หรือTTL จะประกอบไปด้วยช่องการส่งข้อมูล (Transmission:TX) ช่องการรับข้อมูล (Recive:RX) ด้วยการทั้งหมดสามชุด ใช้เพื่อติดต่อสื่อสารกับโมดูลภายนอก 1 ชุด ใช้เพื่อติดต่อสื่อสารกับโมดูลสื่อสารผ่านผู้ให้บริการอินเตอร์เน็ต (Neroband) 1 ชุด สำหรับแปลงเป็นมาตรฐานการสื่อสารแบบ RS232 (TTL to RS232) 1 ชุด ออกแบบให้ใช้ขาของตัวอุปกรณ์ (General Purpose Input Output:GPIO)



รูปที่ 3.24 GPIO ที่ใช้เป็น I/O Serial

การกำหนดทางเข้าทางออกของสัญญาให้เป็น RX หรือ TX สามารถทำได้ด้วยวิธีการทางโปรแกรมมิ่ง

```
#include <SoftwareSerial.h>

#define rxPin 2
#define txPin 3
SoftwareSerial mySerial(rxPin, txPin);
void setup() {
  pinMode(txPin, OUTPUT);
  pinMode(rxPin, INPUT);
  mySerial.begin(9600);
  Serial.begin(9600);
```

รูปที่ 3.25 คำสั่งที่ใช้ในการใช้งาน Serial Port

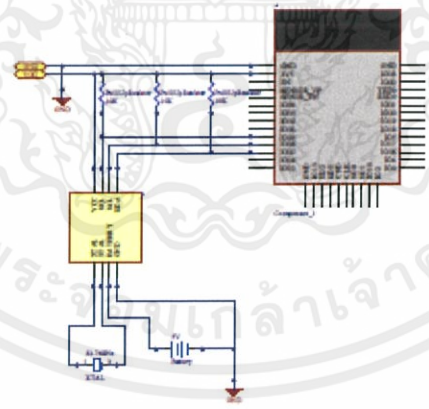
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.7 การอ่านค่าจากโมดูลสร้างสัญญาณนาฬิกา (Real Time Clock : RTC)

โมดูล RTC ใช้รูปแบบการสื่อสาร I2C ในการส่งหรือรับข้อมูลเวลาที่ถูกสร้างขึ้นจาก IC DS1307 โดย Microcontroller ESP32 สามารถเรียกขอเวลาจาก DS1307 จากแอดเดรส 0x68

```
#include "Wire.h"
#define DS1307_I2C_ADDRESS 0x68 // the I2C address of Tiny RTC
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
return ( (val/10*16) + (val%10) );
}
// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
return ( (val/16*10) + (val%16) );
}
// Function to set the currnt time, change the second&minute&hour to the right time
void setDateDs1307()
```

รูปที่ 3.26 คำสั่งภาษาซีที่เรียกเวลาจากโมดูล RTC

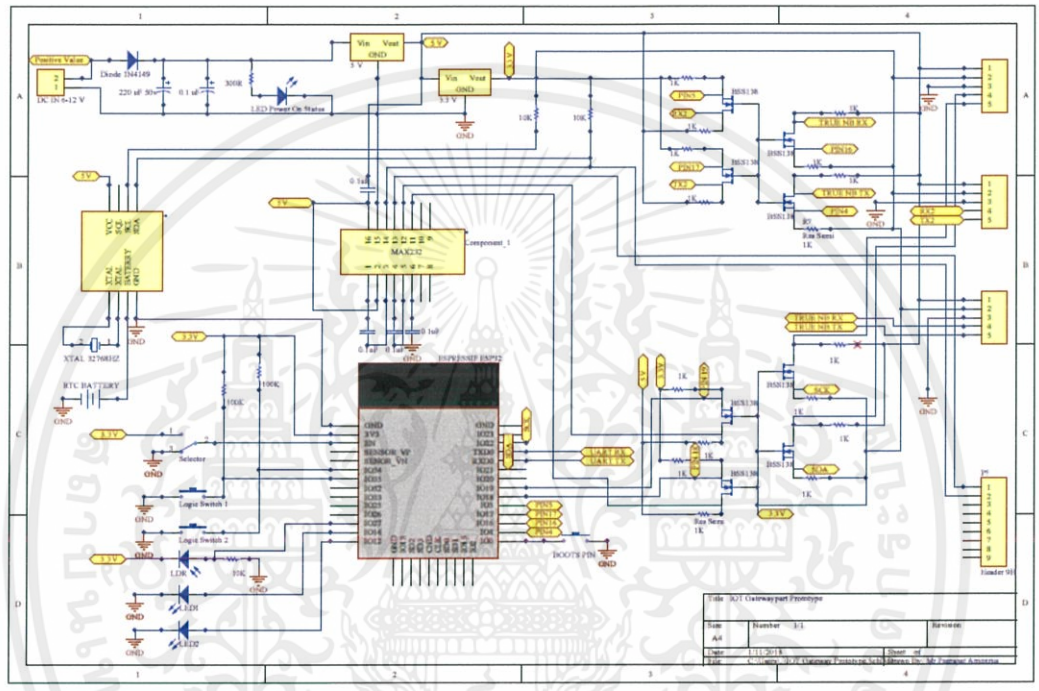


รูปที่ 3.27 วงจร RTC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 อุปกรณ์ต้นแบบ

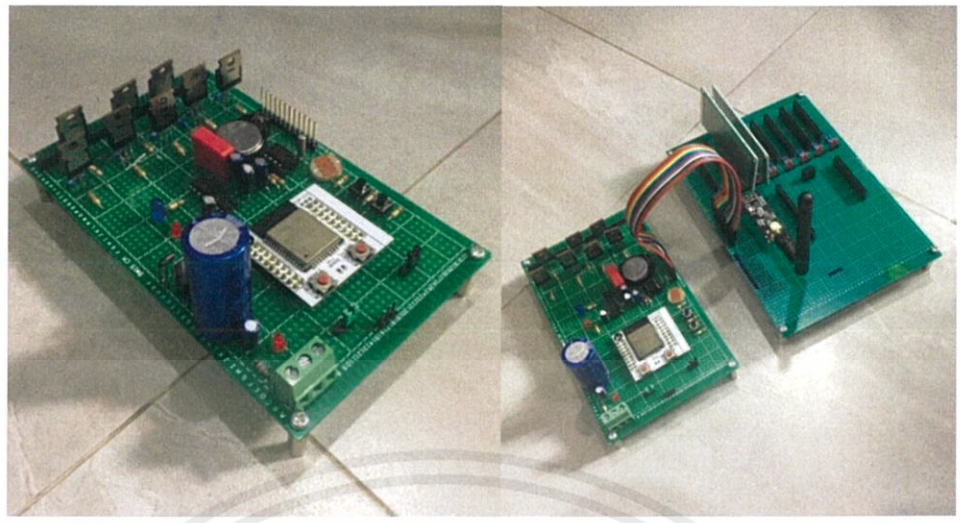
อุปกรณ์ต้นแบบถูกจัดสร้างขึ้นหลังจากทำการทดสอบแต่ละโมดูลย่อยแต่ละโมดูล เสร็จสิ้นลงและได้ผลลัพธ์ว่า แต่ละโมดูลสามารถทำหน้าที่ของตนเองได้อย่างสมบูรณ์แบบในขณะที่อยู่แยกอิสระต่อกัน อุปกรณ์ต้นแบบชิ้นที่หนึ่งถูกสร้างขึ้นโดยการนำเอาโมดูลย่อยแต่ละโมดูลมาประกอบให้อยู่บนแผ่น PCB อเนกประสงค์เพียงแผ่นเดียว



รูปที่ 3.28 Full schematic IoT Gateway Platform

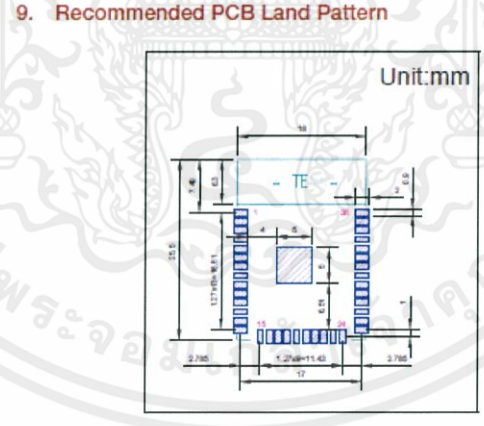
อุปกรณ์ต้นแบบถูกสร้างขึ้นบนแผ่น PCB อเนกประสงค์ เพื่อง่ายต่อการประกอบและทดลองอุปกรณ์ที่ใช้ยังเป็นอุปกรณ์อิเล็กทรอนิกส์ที่มีขนาดใหญ่ และจะถูกปรับใช้เป็นอุปกรณ์ SMD ในชิ้นงานจริง แต่อุปกรณ์ที่ใช้ยังคงเป็นอุปกรณ์ที่มีค่าเดียวกันกับอุปกรณ์บนแผ่น PCB อเนกประสงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



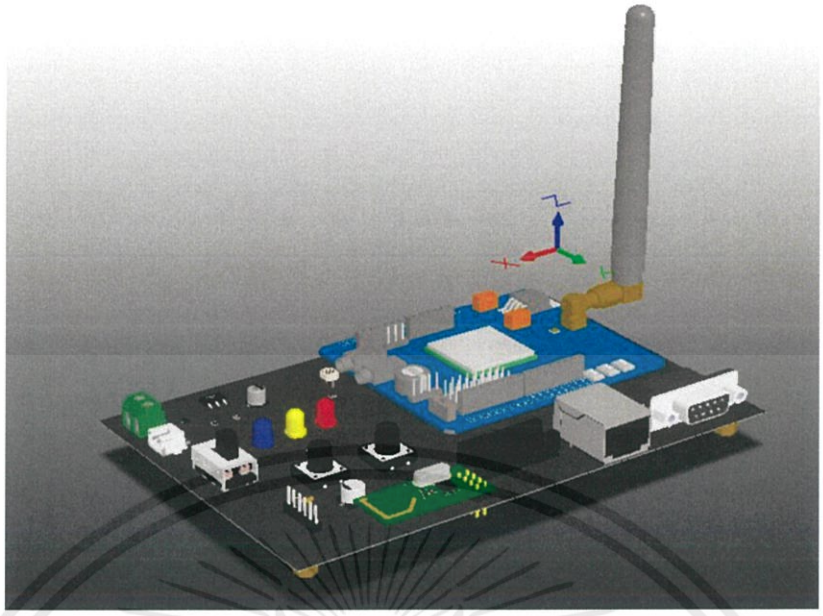
รูปที่ 3.29 อุปกรณ์ต้นแบบบนแผ่น PCB อเนกประสงค์

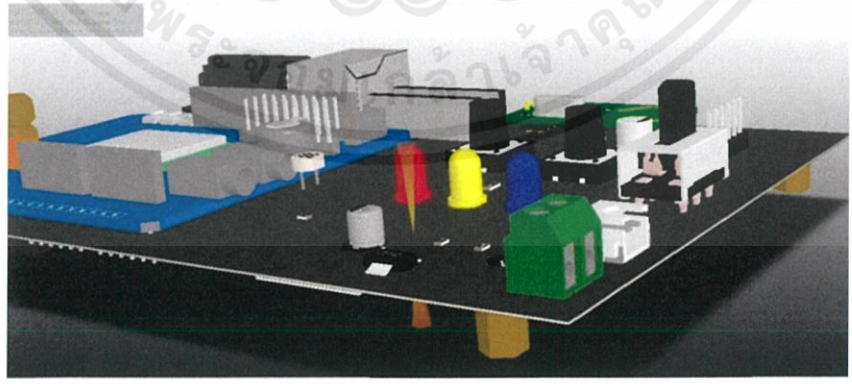
หลังจากมีการทดสอบอุปกรณ์ต้นแบบที่อยู่ในลักษณะ PCB อเนกประสงค์เรียบร้อยแล้วจึงเริ่มทำการออกแบบ PCB 3D เพื่อแสดงตัวอย่างขนาด เบอร์ จำนวน ของอุปกรณ์ที่เลือกใช้ รวมไปถึงจะทราบขนาดและ layout ของแผ่น PCB เพื่อประเมินต้นทุนการผลิต โดยทำการออกแบบบนโปรแกรม Altium Design ซึ่งการจัดวางอุปกรณ์ทำตามที่คู่มือผู้ผลิต Microcontroller แนะนำทุกขั้นตอน



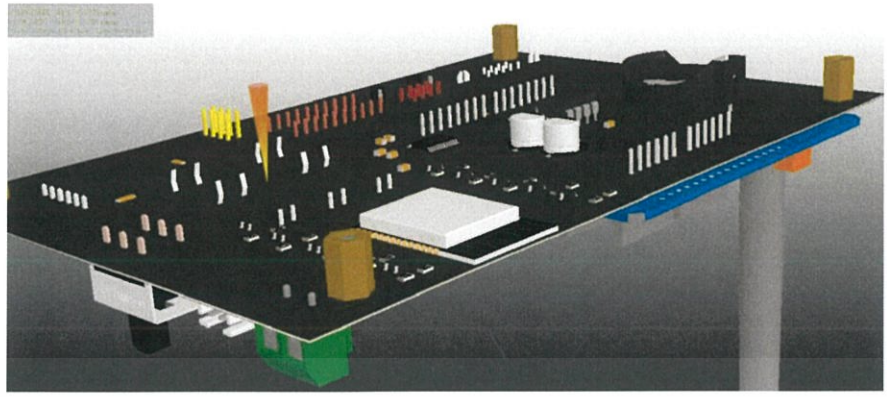
รูปที่ 3.30 Recommended PCB Land Pattern

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.31 Iot Gateway Platform Prototype 3D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

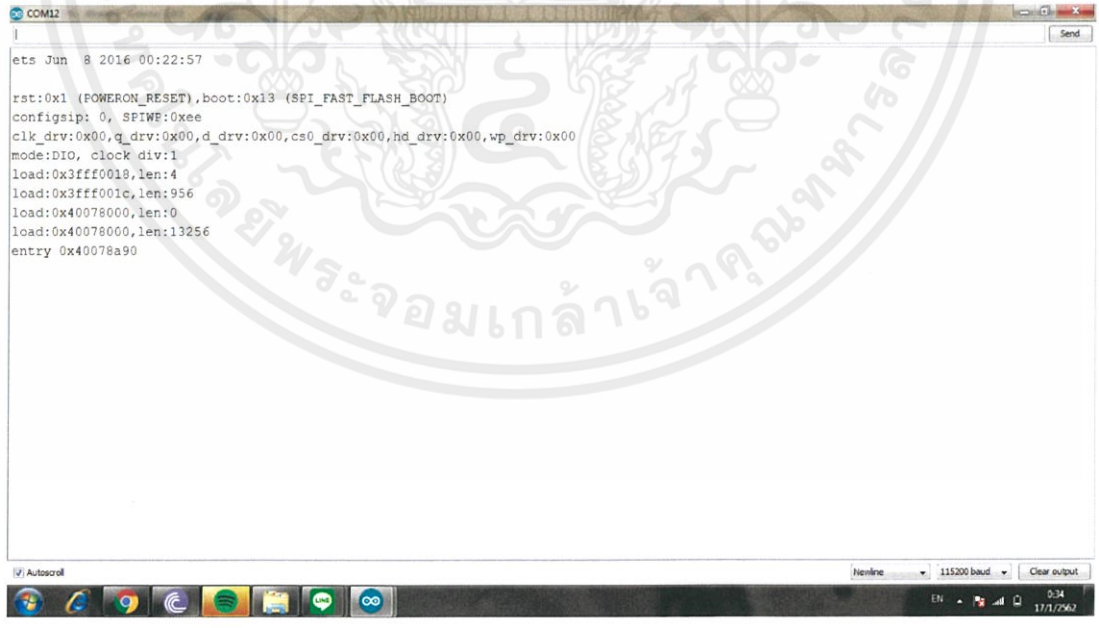
บทที่ 4

ผลการทดลองในการดำเนินโครงการ

บทนี้จะกล่าวเกี่ยวกับผลการทดลองในการดำเนินโครงการ ซึ่งจะแสดงส่วนของโปรแกรมมิ่งที่ถูกอัพโหลดเข้าไปสู่ไมโครโปรเซสเซอร์นำไปสู่กระบวนการควบคุมอุปกรณ์ฮาร์ดแวร์ และทำให้อุปกรณ์ฮาร์ดแวร์ทำงานได้ตรงเป้าหมายและเกิดผลลัพธ์ที่ถูกต้อง โดยผลการทดลองอาจถูกแยกออกเป็นสองส่วนเพื่อง่ายต่อการอธิบาย โดยจะแยกเป็นผลการทดลองเพื่อทดสอบการทำงานของโมดูลย่อยต่างๆภายใน IoT Gateway Platform และส่วนที่เป็นการวัดระดับสัญญาณทางไฟฟ้าของการรับส่งข้อมูลเพื่อนำมาเปรียบเทียบกับระดับสัญญาณไฟฟ้าในทางทฤษฎีของมาตรฐานการสื่อสารแบบ I2C และ TTL

4.1 วงจร Reset

กระบวนการ Reset จะเกิดขึ้นทันทีหลังจากให้ลอจิก Low เข้าสู่ Enable Pin และสามารถแสดงผลการรีเซตของ Microcontroller ผ่าน Serial Monitor

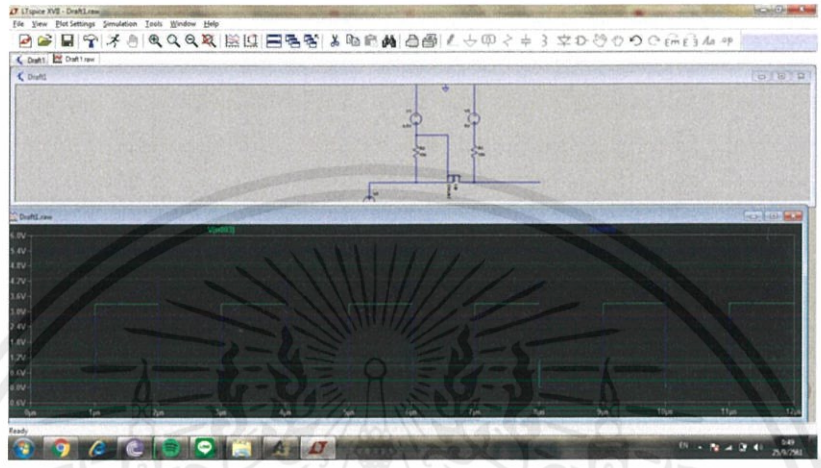


รูปที่ 4.1 Serial Monitor แสดงผล Reset

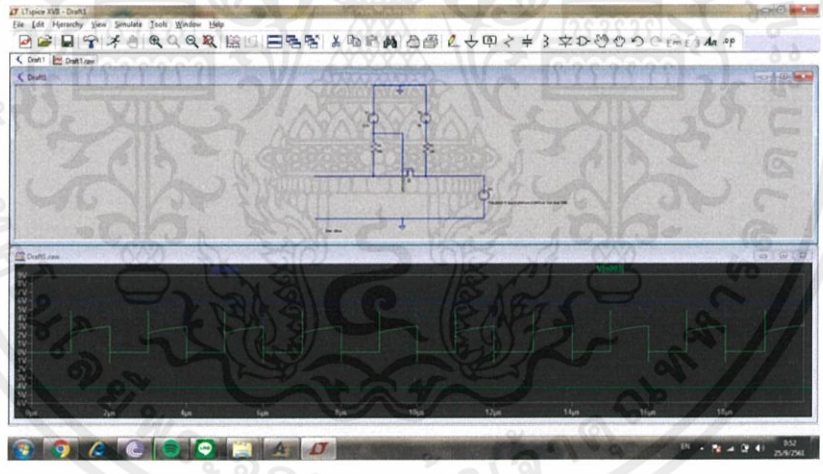
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 โมดูล Logic converter

การแสดงผลของวงจร Logic Converter สามารถศึกษาได้จากการ Simulation บนโปรแกรม Lt Spice และอาจสังเกตได้จากการวัดระดับ Logic จาก Oscilloscope ในรูปแบบการสื่อสารต่างๆ



รูปที่ 4.2 to 5 V



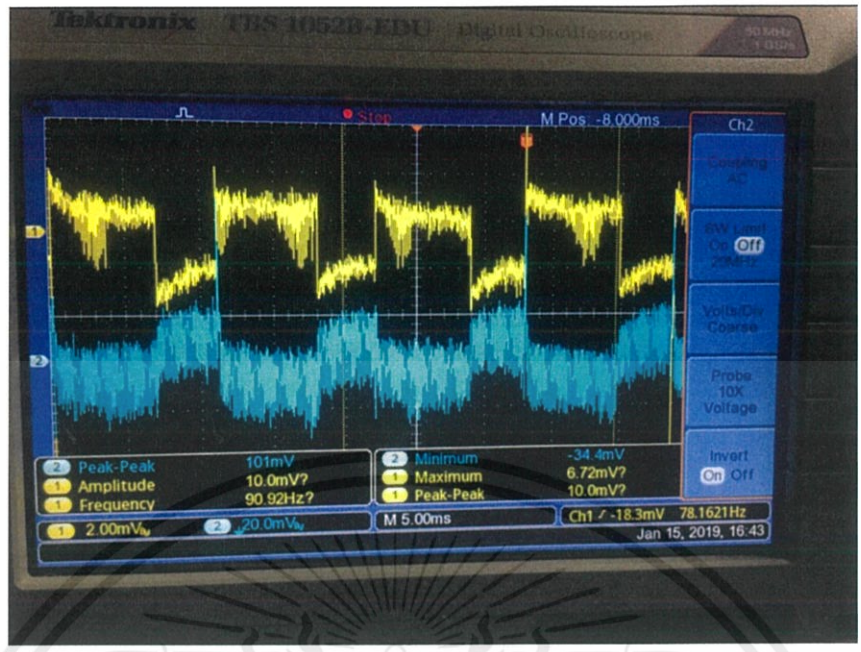
รูปที่ 4.3 5V to 3.3V

4.3 โมดูล Serial to RS485

ลักษณะของ Logic ที่ผ่านวงจรแปลงรูปแบบการสื่อสาร ที่ถูกวัดด้วย Oscilloscope ผลที่ได้คือลักษณะแรงดันของ Logic มีความคล้ายคลึงกับระดับ Logic ในทางทฤษฎี

ระดับแรงดันของ Logic อาจจะไม่สามารถสังเกตได้อย่างแม่นยำเป็นผลมาจากระดับของ Spice ที่เกิดขึ้น แต่ลักษณะทางกายภาพของการแปลง Logic มีการเกิดขึ้นอย่างชัดเจน

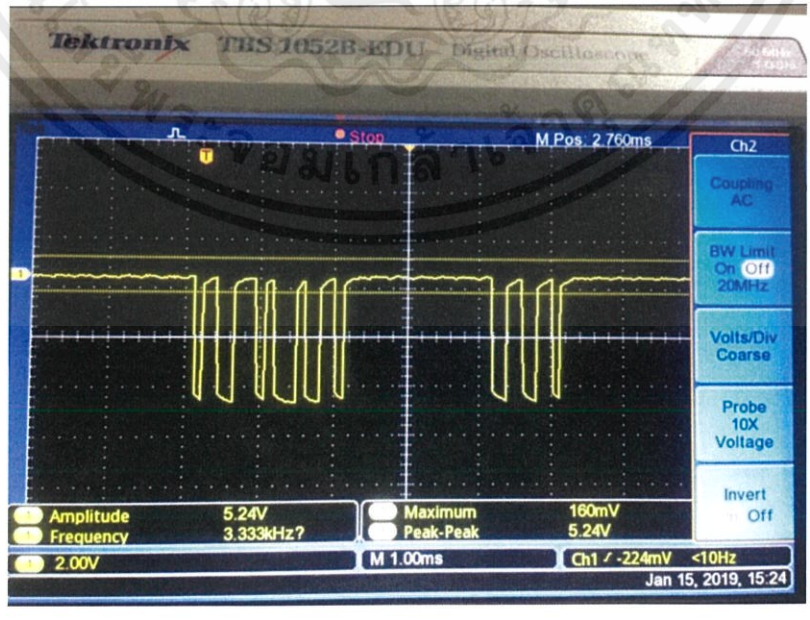
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ลักษณะสัญญาณเมื่อถูกแปลงรูปแบบการสื่อสาร

4.4 ระดับแรงดันไฟฟ้าของ Logic ในรูปแบบการสื่อสาร I2C

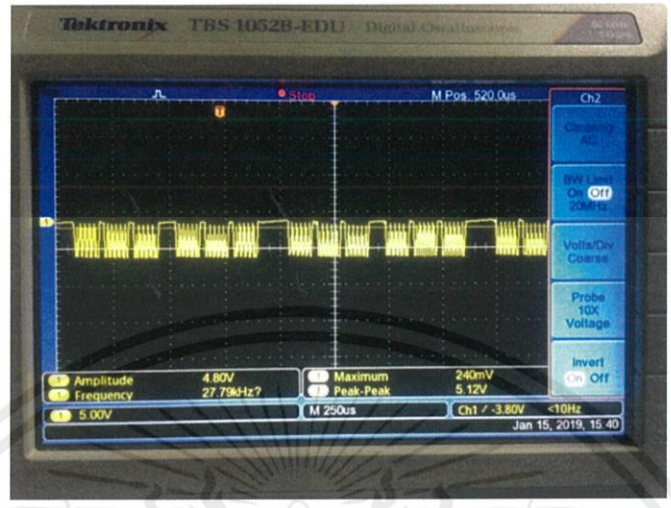
ลักษณะ Logic ของข้อมูลที่ถูกส่งไปในรูปแบบการสื่อสาร I2C สามารถสังเกตผ่านหน้าจอ **Serial Monitor** และการวัดระดับแรงดันไฟฟ้าในสายไฟที่ถูกเชื่อมต่อในรูปแบบการสื่อสาร I2C โดยจะสามารถสังเกตเห็นเป็นสัญญาณสี่เหลี่ยม เกิดขึ้นทันทีที่พร้อม **Serial Monitor** เมื่อมีการรับหรือส่งข้อมูลผ่านรูปแบบการสื่อสาร I2C



รูปที่ 4.5 แสดง Data ที่ถูกส่งไปในรูปแบบการสื่อสาร I2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการวัดระดับแรงดันไฟฟ้าในสายไฟที่ทำหน้าที่ส่งสัญญาณนาฬิกา (Clk) แสดงให้เห็นถึงระดับแรงดันไฟฟ้าของ Logic ที่ถูกส่งออกมาจาก Microprosessor ซึ่งเกิดขึ้นตลอดเวลา

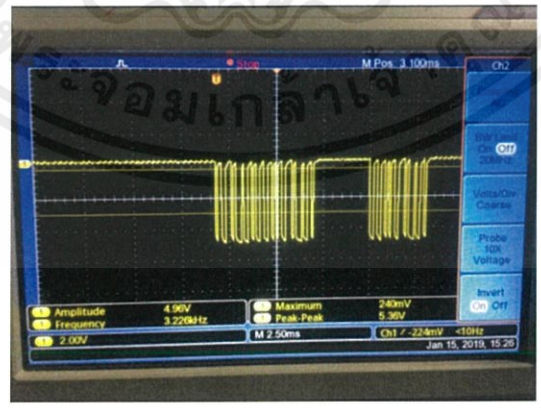


รูปที่ 4.6 สัญญาณนาฬิกา

การวัดระดับสัญญาณทางไฟฟ้า ถูกวัดหลังจากลอจิกผ่านวงจร Logic Converter เรียบร้อยแล้วจึงจะเห็นได้ว่าสามารถวัดระดับสัญญาณทางไฟฟ้าได้ประมาณ 5V

4.5 ระดับแรงดันไฟฟ้าของ Logic ในรูปแบบการสื่อสาร Serial

ลักษณะ Logic ของข้อมูลที่ถูกส่งไปในรูปแบบการสื่อสาร Serial จะมีความคล้ายกับแบบ I2C โดยสามารถสังเกตผ่านหน้าจอ **Serial Monitor** และการวัดระดับแรงดันไฟฟ้าของ Logic สามารถวัดที่สาย Rx,Tx เทียบกับกราวด์ โดยจะสามารถสังเกตเห็นเป็นสัญญาณสี่เหลี่ยม เกิดขึ้นทันทีทีพร้อม **Serial Monitor** เมื่อมีการรับหรือส่งข้อมูลผ่านรูปแบบการสื่อสารแบบ Serial



รูปที่ 4.7 แสดง Data ที่ถูกส่งไปในรูปแบบการสื่อสาร Serial

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 บทสรุปโครงงาน

โครงงานนี้นำเสนอแนวทางการสร้าง iot Gateway Platform โดยเลือกใช้ ESP32 Platform iot Gateway Platform จะสามารถทำงานได้บรรลุเป้าหมายได้ครบถ้วนจะต้องประกอบไปด้วยสองส่วนคือ ส่วนที่เป็นซอฟแวร์และส่วนที่เป็นฮาร์ดแวร์ โดยโครงงานชิ้นนี้ถูกจัดทำขึ้นในระหว่างที่ถูกมอบหมายภาระงานให้รับผิดชอบดูแลและออกแบบอุปกรณ์ฮาร์ดแวร์เพื่อเตรียมการโปรแกรมมิ่ง โดยความอัจฉริยะของ iot Gateway Platform จะขึ้นกับความสามารถและความคิดสร้างสรรค์ของตัวโปรแกรมเมอร์ ในระหว่างดำเนินงาน ชิ้นงานต้นแบบที่ถูกสร้างขึ้น ได้ถูกทดสอบหลายครั้งด้วยสถานการณ์ที่หลากหลาย จนมีความมั่นใจในระดับหนึ่งว่าสมควรแก่เวลาที่จะถูกสร้างเป็นแผ่นวงจรรวมและถูกผลิตออกมาเพื่อขายสู่ท้องตลาด

5.2 ข้อเสนอแนะในการทำวิจัยต่อไป

ในการตั้งค่าคำสั่งหรือเงื่อนไขสำหรับ Iot Gateway Platform ที่ถูกสร้างด้วย ESP32 Platform ผู้ใช้งานจำเป็นต้องมีความเชี่ยวชาญในการเขียนภาษา C ในระดับหนึ่ง ซึ่งภาษา C เป็นภาษาระดับล่าง (ใกล้ชิดกับอุปกรณ์ฮาร์ดแวร์) ซึ่งในการโปรแกรมมิ่งให้ iot Gateway Platform ทำงานได้อย่างชาญฉลาดหรือมีเงื่อนไนการเลือกทำงานมากๆ การเขียนคำสั่งยิ่งทำได้ยาก จึงไม่เหมาะกับงานที่มีความซับซ้อนสูง iot Gatewat Platform ที่เหมาะสมกับงานที่ต้องการความซับซ้อนสูงอาจถูกสร้างจากไมโครโปรเซสเซอร์ตัวอื่นเช่น ARM Platform , raspberry pi เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ธีรวัฒน์ ประกอบผล .// การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ .// สำนักพิมพ์ ส.ส.ท .// (ธันวาคม 2544)

[2] สุทธินันท์ ทรงขำ .// การออกแบบฮาร์ดแวร์ไมโครคอมพิวเตอร์ขั้นสูง .// บริษัท ซีเอ็ดยูเคชั่นจำกัด(มหาชน) .// (2538)

[3] พรจิต ประทุมสุวรรณ .// ทฤษฎีและการใช้งาน(PC/PLC)PROGRAMMABLE CONTROLLER .// โรงพิมพ์เรือนแก้วการพิมพ์ .//(กุมภาพันธ์ 2536)

[4] เดชฤทธิ์ มณีธรรม,สำเริง เต็มราม .// คัมภีร์ไมโครคอนโทรลเลอร์ MICROCONTROLLER PIC .// หจก.ไทยเจริญการพิมพ์ .// (กุมภาพันธ์ 2549)

[5] William J.Beyda .// Basic Data Communication .// University Of California , Berkeley International Business Machines Corporation

[6] Frederick F. Driscoll .// Data Communication .// Wentworth Institute Of technology

[7] Robert L. Ellis .// Designing Data Network .// Prentice-Hall,Inc.,Englewood Cliffs,NJ

[8] Thai Easy Elec .// UART/TTL/RS232/MAX3232คืออะไร .// สืบค้นวันที่ 14/8/60 ./ จาก https://sites.google.com/site/phasathaionline/hnwy-kar-reiyn-ru9

[9] Mindtek Online articles .// การใช้งาน RTC (Real Time Clock) ด้วย DS1307 .// สืบค้นวันที่ 17/8/60 ./ จาก http://www.mind-tek.net/ds1307.php

[10] Iot Sharing .// Demo 41 : ESP32 Connects with nRF24L01 2.4 Ghz Wireless chip .//สืบค้นวันที่ 2/9/60./จากhttp://www.iotsharing.com/2018/03/esp-and-raspberry-connect-with-nrf24l01.html

[11] Aimagin Blog .// การใช้งานพอร์ตสื่อสาร SPI .// สืบค้นวันที่ 6/9/60 ./ จาก http://aimagin.com/blog/spi/?lang=th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

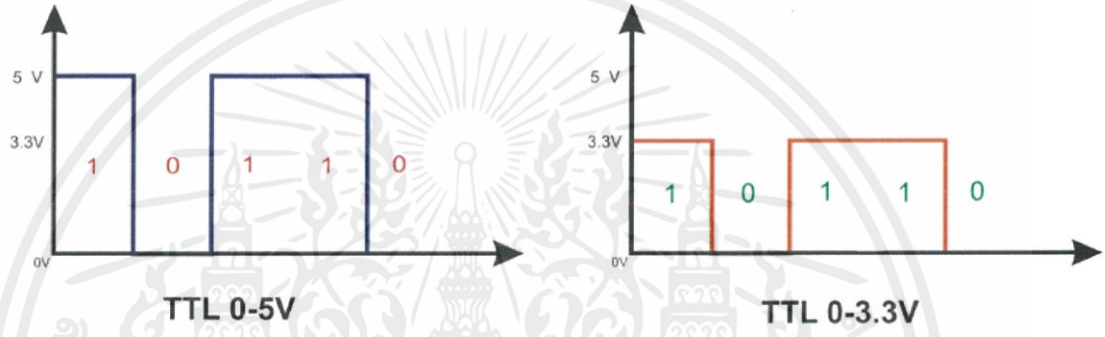
[12] Hackday.IO .// Make Your Own Arduino Nano.(DIY Arduino Nano) //
สืบค้นวันที่ 7/9/60 ./จาก https://hackaday.io/project/21291-make-your-own-arduino-nano-diy-arduino-nano

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

LVTTL

TTL เป็นระดับแรงดันที่ถูกกำหนดขึ้นในยุคแรกๆเพื่อใช้ระหว่าง Transistor กับ Transistor ภายในวงจรรวม(IC) ดังนั้น TTL จะใช้ระดับแรงดัน อยู่ที่ 0 – 5 V แต่ในปัจจุบันมีอุปกรณ์หลายเบอร์ที่ทำงานในช่วง 0 – 3.3 V (เรียกแรงดันระดับนี้ว่า LVTTL) ซึ่งผู้ใช้ควรตรวจสอบจาก Datasheet ของอุปกรณ์ที่ใช้เสียก่อนว่าเป็นระดับแรงดันแบบใด เพราะหากใช้ผิดประเภทจะทำให้อุปกรณ์เสียหาย



Register

เป็นส่วนประกอบหนึ่งใน Microcontroller ทำหน้าที่ในการเก็บข้อมูลชั่วคราว ซึ่งเก็บผลจากการคำนวณโดยแยกพื้นที่ส่วนหนึ่งของหน่วยความจำภายใน Microcontroller มาใช้ นั้นคือ Register เป็นหน่วยความจำส่วนหนึ่งใน CPU

Serial Monitor

การใช้งาน Serial Monitor จำเป็นต้องเขียนภาษาซีในการสั่งงาน อาจแบ่งออกเป็นสามส่วนหลักคือ การเรียกใช้งานไลบรารี่ HardwareSerial.h ตั้งค่าเริ่มต้นใช้งาน Serial Port ในส่วนของ Void Setup ความเร็วในการรับส่งข้อมูล (Serial baud rate)

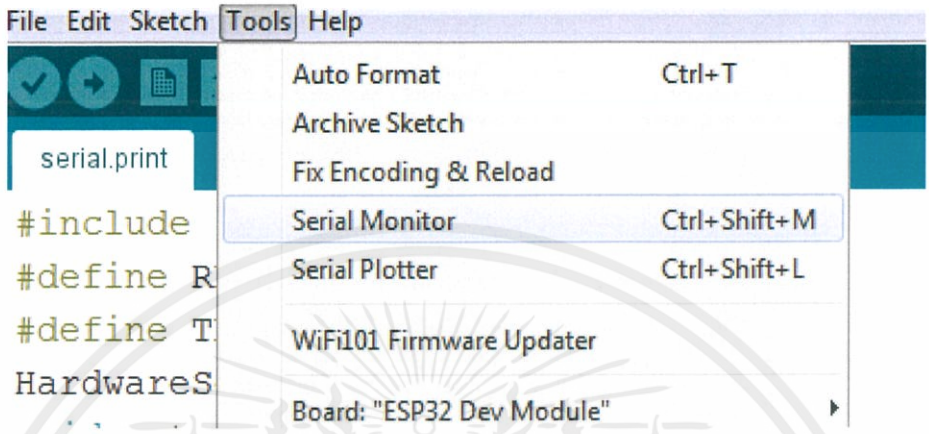
```
#include "HardwareSerial.h"
#define RXPIN1 27
#define TXPIN1 26
HardwareSerial myserial_1(1);
void setup() {
  myserial_1.begin(9600, SERIAL_8N1, RXPIN1, TXPIN1);

}

void loop() {
 Serial.print("ABC");
 delay(2000);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปิด Serial Monitor จาก Tool > Serial Monitor



เมื่อเปิด Serial Monitor ขึ้นมาแล้วต้องเลือก Baud Rate ให้ตรงกับที่ตั้งค่าไว้ในหน้าเขียนโค้ด

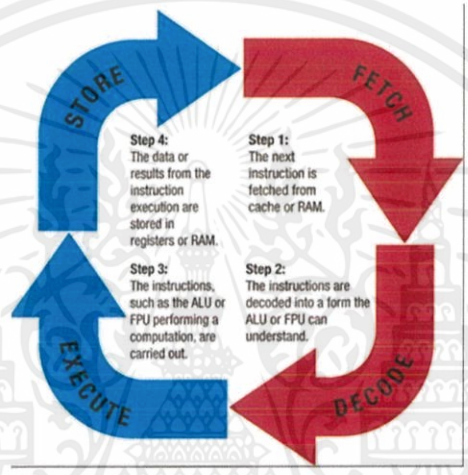
แมชชีนไซเคิล

หมายถึงการทำงานของคอมพิวเตอร์จะต้องทำตามโปรแกรมที่กำหนดไว้ในหน่วยความจำ โดยโปรแกรมเกิดจากการนำคำสั่งมาต่อเรียงกัน เมื่อคอมพิวเตอร์ทำงาน หน่วยควบคุมการอ่านคำสั่งต่างๆ เข้ามาประมวลผลในซีพียู โดยวงรอบของการทำคำสั่งของซีพียูประกอบด้วยขั้นตอนการทำงานพื้นฐาน 4 ขั้นตอนดังนี้

1. ขั้นตอนการรับเข้าข้อมูล (fetch) เริ่มแรกหน่วยควบคุมรหัสคำสั่งและข้อมูลที่จะประมวลผลจากหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ขั้นตอนการถอดรหัส (decode) เมื่อรหัสคำสั่งเข้ามาอยู่ในซีพียูแล้ว หน่วยควบคุมจะถอดรหัสคำสั่งแล้วส่งคำสั่งและข้อมูลไปยังหน่วยคำนวณและตรรกะ
3. ขั้นตอนการทำงาน (execute) หน่วยคำนวณและตรรกะทำการคำนวณโดยใช้ข้อมูลที่ได้รับมาถอดรหัสคำสั่ง และทราบแล้วว่าต้องการทำอะไร ซีพียูก็จะทำตามคำสั่งนั้น
4. ขั้นตอนการเก็บ (store) หลังจากทำคำสั่งก็จะเก็บผลลัพธ์ที่ได้ไว้ในหน่วยความจำ



เวลาของแมชชีนไซเคิลขึ้นอยู่กับ Crystal Oscillator ที่มีหน้าที่สร้างความถี่ให้กับ Microcontroller ตัวอย่างการคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้