รายงานสหกิจศึกษาฉบับสมบูรณ์

ตรวจจับวัตถุในระบบ ASRS (Automated Storage and Retrieval System)
Optical Sensor Interfacing for ASRS

นายสิริเดช พนาธเนศ

สาขาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

รายงานสหกิจศึกษาฉบับสมบูรณ์

ตรวจจับวัตถุในระบบ ASRS (Automated Storage and Retrieval System)

Optical Sensor Interfacing for ASRS

นายสิริเดช พนาธเนศ

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงงานสหกิจศึกษา	ตรวจจับวัตถุในระบบ ASRS (Automated Storage and Retrieval System)
ชื่อ-สกุล นักศึกษา	นายสิริเดช พนาธเนศ
คณะ	วิศวกรรมศาสตร์ **ภาควิชา** วิศวกรรมคอมพิวเตอร์
ชื่อ-สกุล อาจารย์นิเทศ	อ.จิระศักดิ์ สิทธิกร และ อ.บัณฑิต พัสยา
ชื่อ-สกุล ผู้นิเทศงาน	นายพงษ์ศักดิ์ ทองปลอด
สถานประกอบการ	บริษัท ซีเกท เทคโนโลยี ประเทศไทย

บทคัดย่อ

การศึกษาครั้งนี้มีวัตถุประสงค์เพื่อออกแบบวงจร และพัฒนาเฟิร์มแวร์ของไมโครคอนโทรลเลอร์เพื่อใช้งานในระบบจัดเก็บผลิตภัณฑ์ที่อยู่ระหว่างกระบวนการทดสอบ เพื่อทำให้ระบบมีความสามารถในการตรวจสอบว่า ในช่องเก็บของมีผลิตภัณฑ์อยู่หรือไม่ โดยใช้เทคโนโลยี Optical sensor สำหรับการตรวจจับวัตถุในช่องเก็บของ ซึ่งจะทำให้ระบบมีความสามารถในการทำงานแบบอัตโนมัติได้มีประสิทธิภาพสูงขึ้น

ผู้วิจัยเลือกพัฒนาเฟิร์มแวร์ด้วยไมโครคอนโทรลเลอร์ STM32F107VCT6 ซึ่งเป็นบอร์ดที่ใช้สำหรับทดลอง และเรียนรู้การทำงานของตัว Optical sensor และการส่งข้อมูลผ่าน CAN Bus โดยจะใช้ไมโครคอนโทรลเลอร์ STM32F303K8 และ STM32F042K6 มาใช้ทดสอบความเหมาะสมของตัวประมวลผลเพื่อหาว่าไมโครคอนโทรลเลอร์รุ่นใดที่มีความเหมาะสมกับการนำไปใช้งานจริงในระบบมากกว่ากัน ซึ่งผู้วิจัยใช้โปรแกรม STM32CubeMX และ Keil uVersion5 ซึ่งเป็น MDK ที่ได้รับความนิยมสูงสำหรับการเขียนโปรแกรมเพื่อควบคุมชิปที่เป็น ARM โดยใช้ภาษา C ในการเขียนโปรแกรม และใช้ VL6180X ของบริษัท STMicrocontroller ที่เป็น Proximity, Gesture และ Ambient Light Sensor โดยใช้การสื่อสารแบบ I^2C ในการติดต่อกับตัว Sensor และใช้ CAN Bus เพื่อสื่อสารระหว่างไมโครคอนโทรลเลอร์กับเครื่องคอมพิวเตอร์

คำสำคัญ: Optical sensor, Microcontroller, UART, I^2C, CAN Bus, CAN Open

Cooperative Title: Optical Sensor Interfacing for ASRS

Student Intern Name: Mr. Siridej Phanathanate

Faculty: Engineering **Department:** Computer Engineering

Advisor Name: Mr. Jirasak Sittigorn and Mr. Bundit Pasaya

Mentor Name: Mr. Pongsak Tongplod

Company: Seagate Technology (Thailand) Ltd.

ABSTRACT

This study aims to design an electronic circuit and develop firmware of the microcontroller to use in the storage system which is in the process of testing. The microcontroller board will conduct the system to have an ability to detect an object in the storage slot by using optical sensor. It will make the system works automatically with higher efficiency.

The researcher chooses to develop the firmware by using STM32F107VCT6 microcontroller for testing and learning the operation of optical sensor and the data transmission through CAN Bus. The suitability of processor is testified by STM32F303K8 and STM32F042K6 microcontrollers to compare and find an appropriate one to use in the actual system. The researcher uses the programs called "STM32CubeMX" and "Keil uVersion5" which are the well-known MDK for programing in order to control ARM chip using C Programing Language. "VL6180X" sensor from STMicroelectronics, which is used to distinguish an object, is Proximity, Gesture, and Ambient Light Sensor. Therefore, I^2C is used to communicate between sensor and a microcontroller, and CAN Bus is used to communicate between microcontroller and a computer.

Keywords: Optical sensor, Microcontroller, UART, I^2C, CAN Bus, CAN Open

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้เสร็จสมบูรณ์ได้ด้วยความช่วยเหลือจากหลายท่านทั้งทางตรง และทางอ้อม ซึ่งจะสำเร็จไม่ได้หากปราศจากความช่วยเหลือของบุคคลเหล่านี้

ขอขอบคุณ อาจารย์ผู้นิเทศงาน อาจารย์จิระศักดิ์ สิทธิกร ซึ่งเป็นผู้ที่มานิเทศงาน และช่วยให้คำแนะนำในการทำงาน การแก้ไขปัญหา และจุดบกพร่องของโครงงาน ซึ่งทำให้โครงงานสมบูรณ์มากยิ่งขึ้น

ขอขอบคุณ อาจารย์บัณฑิต พัสยา ซึ่งเป็นผู้ที่คอยดูแล และให้คำแนะนำในด้านของงานสหกิจศึกษา ซึ่งทำให้การทำโครงงานสะดวกมากยิ่งขึ้น

ขอขอบคุณ นายอุทัย ศัทโท หัวหน้างานที่ให้คำปรึกษา แนะนำหลักการในการทำงาน และรูปแบบของงานที่เป็นมาตรฐานตามหลักการที่ถูกต้องอย่างสม่ำเสมอ ซึ่งทำให้ตัวโครงงานสำเร็จลุล่วงอย่างมีคุณภาพ

ขอขอบคุณ นายธนุ พันธ์แก้ว พี่เลี้ยงที่คอยให้คำปรึกษา และให้ความรู้ทางด้านอิเล็กทรอนิกส์ที่นำมาประยุกต์ใช้ในโครงงาน ทำให้การทำโครงงานสำเร็จลุล่วงไปได้

ขอขอบคุณ นายพงษ์ศักดิ์ ทองปลอด พี่เลี้ยงที่คอยดูแล และจัดหาอุปกรณ์ ตลอดจนให้ทำแนะนำเกี่ยวกับการทำโครงงาน ทำให้โครงงานสำเร็จได้ด้วยดี

ขอขอบคุณ บริษัท ซีเกท เทคโนโลยี ประเทศไทย ที่เอื้อเฟื้อสถานที่ และทุนในการทำโครงงานทั้งหมด

สุดท้ายนี้ขอขอบคุณ บิดา มารดา ครอบครัว เพื่อนๆ ตลอดจนผู้เกี่ยวข้องที่ไม่ได้กล่าวนามทุกท่าน ที่เป็นกำลังใจ ให้การสนับสนุน และความช่วยเหลือในการทำโครงงานครั้งนี้จนสำเร็จลุล่วงไปได้

สิริเดช พนาธเนศ

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูปภาพ VII

บทที่ 1 บทนำ 1

1.1. ความเป็นมา และความสำคัญ 1

1.2. วัตถุประสงค์ของงาน 1

1.3. ขอบเขตของการทำงาน 1

1.4. วิธีดำเนินการทำงาน 4

1.5. ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง 5

2.1. ทฤษฎีที่เกี่ยวข้อง 5

บทที่ 3 วิธีดำเนินงาน 8

3.1. วิธีดำเนินการทำงาน 8

3.2. แผนการดำเนินงาน 19

บทที่ 4 ผลการทำงาน 20

4.1. ผลการทดสอบการสื่อสารผ่าน CAN ในบอร์ดทดลอง (STM32F107VCT6) 20

4.2. ผลการทดสอบการสื่อสารผ่าน I^2C ในบอร์ดทดลอง (STM32F107VCT6) 20

4.3. ผลการทดสอบการสื่อสารผ่าน UART ในบอร์ดทดลอง (STM32F107VCT6) 20

สารบัญ (ต่อ)

หน้า

4.4. ผลการทดสอบการทำงานภาพรวมของระบบโดยใช้บอร์ดทดลอง (STM32F107VCT6).....21

4.5. ผลการทดสอบการทำงานของตัวเซ็นเซอร์แสง 23

4.6. ผลการทดสอบการทำงานภาพรวมของระบบด้วยบอร์ดจริง (STM32F303K8) 25

บทที่ 5 สรุปผลการทำงาน และข้อเสนอแนะ 27

5.1 บทสรุป 27

5.2 ปัญหาอุปสรรค และแนวทางแก้ไข 27

5.3 แนวทางการพัฒนาต่อ 27

บรรณานุกรม 28

สารบัญตาราง

ตารางที่ หน้า

3.2.1 แผนการดำเนินงาน 19

สารบัญรูปภาพ

ภาพที่ หน้า

1.1 STM32F107VCT6 Development Board 2

1.2 STM32F303K8 Nucleo Board 3

1.3 (ก) VL6180X Top side 3

1.3 (ข) VL6180X Bottom side 3

2.1 การคำนวนระยะทางของ VL6180X 7

3.1 (ก) Flowchart ของ Master board (Main process) 9

3.1 (ข) Flowchart ของ Master board (Interrupt process) 10

3.2 Flowchart ของ Slave board 11

3.3 STM32F303K8 15

3.4 VL6180X 16

3.5 MAX14840EASA+ 16

3.6 SN65HVD230 17

3.7 TPS76928 17

3.8 (ก) PCB Board Top 18

3.8 (ข) PCB Board Bottom 18

4.1 ผล CAN Interrupt ของ Master 21

4.2 ผล Debugging ของ Master 22

4.3 ผล Debugging ของ Slave 23

4.4 ผลการทดสอบอ่านค่าเซ็นเซอร์กับวัตถุสีขาวมัน 24

4.5 ผลการทดสอบอ่านค่าเซ็นเซอร์กับวัตถุสีดำด้าน 25

4.6 ผลการทดสอบ CAN Interrupt ของระบบ 26

บทที่ 1
บทนำ

1.1. ความเป็นมา และความสำคัญ

เนื่องจากในปัจจุบัน ทางบริษัทได้มีกระบวนการผลิตหลายขั้นตอน บริษัทจึงต้องการระบบที่จะสามารถจัดเก็บ และจัดเรียงตัวงานให้เป็นระเบียบเพื่อให้ระบบสามารถนำตัวงานไปทดสอบในขั้นตอนต่างๆได้อย่างมีประสิทธิภาพ โดยตัวระบบต้องมีความสามารถในการทำงานแบบอัตโนมัติ เพื่อลดปริมาณแรงงานมนุษย์ที่ใช้ในการทำงาน ซึ่งทางบริษัทได้มอบหมายงานให้แผนก R&D เป็นฝ่ายที่ดูแลตัวโปรเจตนี้ โดยตัวผู้วิจัยเองได้รับมอบหมายให้ทำงานส่วนหนึ่งในการออกแบบ Firmware และ PCB ของเซ็นเซอร์ตัวหนึ่งของระบบ ASRS เพื่อให้ระบบสามารถตรวจสอบได้ว่า ภายในช่องจัดเก็บตัวงาน มีตัวงานที่ถูกจัดเก็บอยู่หรือไม่ และตรวจสอบว่าตัวงานอยู่ในตำแหน่งใด เพื่อให้ระบบสามารถจัดการกับการจัดเก็บตัวงานได้อย่างมีประสิทธิภาพ และสามารถนำไปประยุกต์ใช้กับส่วนอื่นๆในระบบทั้งหมดได้ดียิ่งขึ้น

1.2. วัตถุประสงค์ของงาน

1.2.1. เพื่อทำให้ระบบ ASRS สามารถตรวจจับวัตถุภายในช่องเก็บได้เองอย่างมีประสิทธิภาพ

1.3. ขอบเขตของการทำงาน

1.3.1. ความสามารถของ Firmware

1.3.1.1. ใช้ STM32CubeMX และ Keil uVersion5 เป็นเครื่องมือในการพัฒนา Firmware

1.3.1.2. ใช้ RealTerm ในการ debug และทดสอบผลของ Uart

1.3.1.3. ใช้ Kvaser CanKing ในการ debug และทดสอบผลของ CAN Bus

1.3.1.4. ใช้ yEd ในการออกแบบ Flowchart

1.3.1.5. ใช้ PCBWeb ในการออกแบบ PCB

1.3.1.6. ตรวจจับได้ว่าในช่องจัดเก็บมีวัตถุอยู่หรือไม่ภายในระยะ 0 - 100 มิลลิเมตร

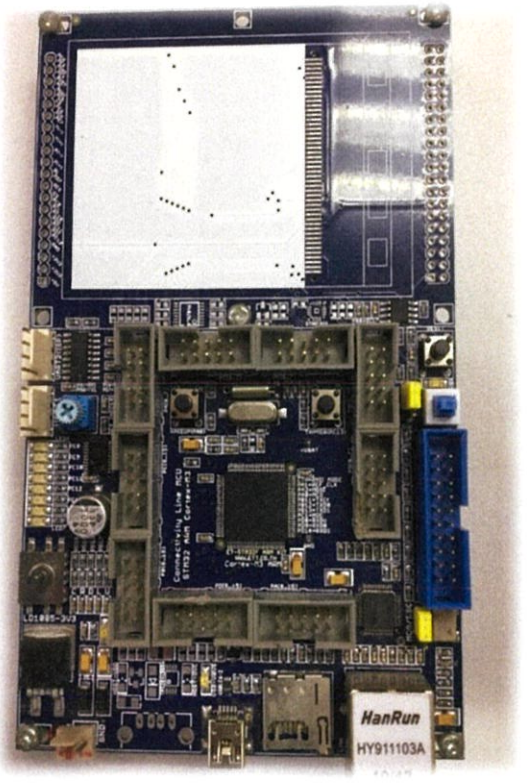
1.3.1.7. Master สั่งการตัว Slave ได้ทุกตัวภายในตู้เก็บของเดียวกัน

1.3.1.8. Slave สามารถส่งข้อมูลสถานะของวัตถุภายในช่องเก็บไปยัง Master ในรูปแบบ 1-to-1

1.3.1.9. Master สามารถส่งข้อมูลสถานะที่บอกว่าขณะนี้มีวัตถุอยู่ในช่องใดบ้าง ตำแหน่งใดบ้าง ไปยังเครื่องคอมพิวเตอร์ที่ใช้ตรวจสอบระบบได้

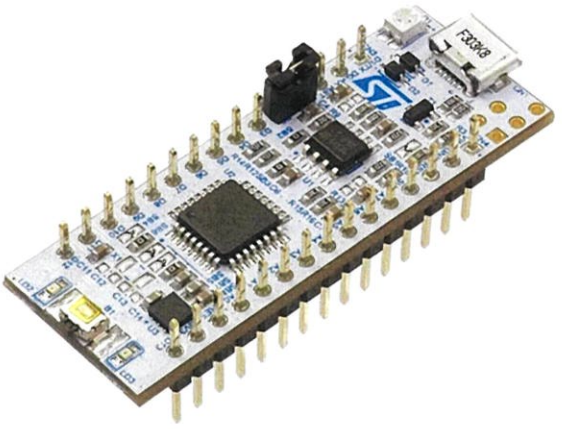
1.3.2. อุปกรณ์

1.3.2.1. ใช้ STM32F107VCT6 Development Board และ STM32F303K8 Nucleo Board ในการพัฒนา



ภาพที่ 1.1 STM32F107VCT6 Development

1.3.2.2. ใช้ Chip STM32F303K8 ในการใช้งานจริง (ใช้ STM32F303K8 Nucleo Board สำหรับการทดลอง)



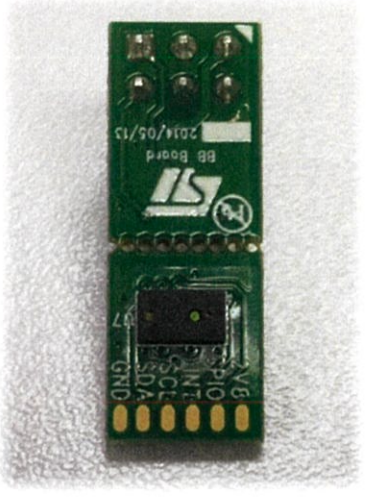
ภาพที่ 1.2 STM32F303K8 Nucleo

1.3.2.3. ใช้ I^2C สำหรับการ Communication ระหว่าง Controller Board กับ Optical Sensor

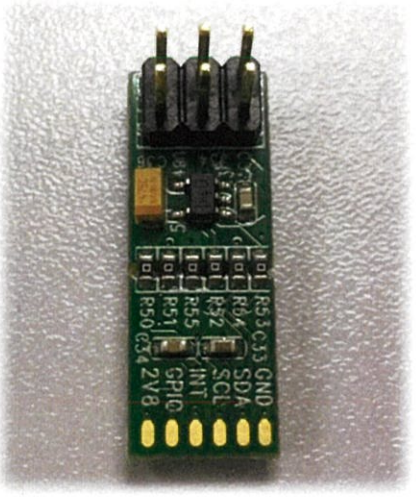
1.3.2.4. ใช้ RS-485 สำหรับการ Communication ระหว่าง Controller Board

1.3.2.5. ใช้ CAN Bus สำหรับการ Communication ระหว่าง Controller Board กับ Computer

1.3.2.6. ใช้ Optical Sensor ในการตรวจจับวัตถุ (VL6180X)



ภาพที่ 1.3 (ก) VL6180X Top



ภาพที่ 1.3 (ข) VL6180X

1.3.2.7. ตัว Optical Sensor สามารถตรวจจับระยะห่างของวัตถุได้ดีในระยะ 0 – 100 มิลลิเมตร

1.3.3. ระยะเวลาวิจัย

ใช้เวลาทั้งหมด 18 สัปดาห์ในการทำงาน

1.4. วิธีดำเนินการทำงาน

1.4.1. รับทราบความต้องการของระบบ

1.4.2. ปรับแต่งความต้องการของระบบ

1.4.3. ออกแบบการทำงานของฟังก์ชันต่างๆของระบบ

1.4.4. ศึกษาการทำงานของตัวอุปกรณ์

1.4.5. ออกแบบ Flowchart ของระบบ

1.4.6. ปรับปรุง Flowchart ของระบบ

1.4.7. สั่งซื้ออุปกรณ์

1.4.8. เขียนโปรแกรม

1.4.9. ทดสอบฟังก์ชันต่างๆ

1.4.10. ปรับแต่งประสิทธิภาพ

1.4.11. ทดสอบระบบทั้งหมด

1.4.12. ออกแบบ PCB

1.5. ประโยชน์ที่คาดว่าจะได้รับ

1.5.1. สามารถนำผลงานไปพัฒนาใช้ได้จริงในสายการผลิต

1.5.2. สามารถทำให้ระบบ ASRS ทำงานได้อย่างมีประสิทธิภาพมากขึ้น

1.5.3. ได้นำความรู้ที่เรียนมาประยุกต์ใช้กับงานจริง

1.5.4. ได้รับความรู้ในการสื่อสารผ่าน CAN Bus และสามารถนำมาใช้งานกับตัว Firmware ได้

1.5.5. ได้รับความรู้ในการสื่อสารกับตัวเซ็นเซอร์ผ่าน I^2C ,การใช้งานตัวเซ็นเซอร์ และสามารถนำมาใช้งานจริงได้

บทที่ 2
แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1. ทฤษฎีที่เกี่ยวข้อง

2.1.1. UART Communication

Universal Asynchronous Receiver-Transmitter คือ Hardware ภายในอุปกรณ์คอมพิวเตอร์ที่ใช้สำหรับการสื่อสารแบบ Asynchronous serial ที่สามารถเปลี่ยนแปลงรูปแบบ และความเร็วของข้อมูลที่ต้องการส่งได้ โดยในการทดลอง ผู้ทดลองได้กำหนดรูปแบบของข้อมูลเป็น Data 7 bits, 1 stop bit โดยใช้ความเร็วเป็น 115,200 bits/sec สำหรับการสื่อสารระหว่างไมโครคอนโทรลเลอร์ กับคอมพิวเตอร์ เพื่อใช้ในการ Debugging ผลลัพธ์ของโปรแกรม และนำไปต่อยอดในการใช้สื่อสารร่วมกับการสื่อสารรูปแบบ RS-485 สำหรับการสื่อสารระหว่างไมโครคอนโทรลเลอร์อีกด้วย

2.1.2. RS-485 Data Transmission

เป็นรูปแบบการสื่อสารที่พัฒนามาจากการสื่อสารรูปแบบ RS-422 ที่เป็น Differential transmission เพื่อทำให้รองรับการสื่อสารรูปแบบที่มีการใช้ตัวอุปกรณ์ภายในระบบที่มากขึ้น ซึ่งการสื่อสารรูปแบบ RS-485 สามารถรองรับอุปกรณ์ได้ 32 อุปกรณ์ โดยสามารถสื่อสารได้ระยะไกลที่สุด 1.2 km. ที่ความเร็ว 100kb/s และความเร็วในการสื่อสารอยู่ที่ 100kb/s – 10Mb/s

2.1.3. CAN Bus Communication

เป็นรูปแบบการสื่อสารที่ทำให้ไมโครคอนโทรลเลอร์สามารถติดต่อสื่อสารกับตัวอุปกรณ์ได้โดยไม่ต้องพึ่งพาเครื่องคอมพิวเตอร์ โดยมีความสามารถคือ

- มี ID เป็น 11 หรือ 29 Bits และส่งข้อมูลในแต่ละครั้งได้ 0 ถึง 8 Bytes
- เป็นการสื่อสารแบบ Peer-to-Peer ทุก Node สามารถมองเห็นข้อมูลภายในเครือข่าย แต่ไม่สามารถอ่านข้อมูลที่ตัวเองส่งออกไปได้
- การเพิ่มตัวอุปกรณ์สามารถทำได้ง่าย
- มีการแบ่งลำดับความสำคัญของข้อมูลโดย Node ที่มีเลข ID น้อยจะมีลำดับความสำคัญสูงกว่า

- มีการส่งข้อมูลใหม่เองหากเกิดข้อผิดพลาด และหากเกิดข้อผิดพลาดเยอะเกินไป Node นั้นจะตัดตัวเองออกจากเครือข่าย
- ความเร็วในการส่งข้อมูลเริ่มตั้งแต่ 10Kbps ถึง 1Mbps
- มีการรับมือกับการเกิด Noise โดยการพันสายเป็นเกรียว
- เครือข่ายสามารถทำงานได้เพียงแค่เชื่อมต่อ Ground ของตัวอุปกรณ์เข้าด้วยกัน แม้ว่าระดับ Voltage DC ที่ใช้จะแตกต่างกัน

โดยผู้วิจัยได้ใช้ความเร็วในการรับส่งข้อมูล 1Mbps เพื่อให้การสื่อสารในระบบเป็นไปได้อย่างรวดเร็วที่สุด และให้ ID ของแต่ละ Node มีความยาว 11 Bits ซึ่งจากความสามารถของตัว CAN Bus เองทำให้ระบบการสื่อสารมีความเสถียรมากกว่าการสื่อสารรูปแบบอื่นๆ และทำให้ประหยัดงบประมานทางด้านสายสัญญานที่ใช้ในการสื่อสาร

2.1.4. CAN Open Protocol

เป็น Protocol ที่ใช้สำหรับ Embedded system โดยจะทำงานเกี่ยวกับ OSI model ตั้งแต่ Network layer ขึ้นไป CAN Open ที่ผู้วิจัยใช้เป็น Protocol ที่ประกอบด้วย 2 รูปแบบการทำงาน คือ รูปแบบแรกเป็นรูปแบบคำสั่งสำหรับการ Write Access ซึ่งประกอบด้วย 2 ส่วนย่อยดังต่อไปนี้

- Host send: CMD จำนวน 1 byte ที่บอกว่าจะส่งข้อมูลจำนวนกี่ Byte, Index 2 bytes โดยเรียงจาก LSB to MSB, Sub-Index 1 byte และตามด้วย Data 1-4 bytes ขึ้น โดยเรียงจาก LSB to MSB
- Slave answers: RES จำนวน 1 byte ที่จะบอกสถานะการรับว่าสำเร็จหรือไม่, Index 2 bytes โดยเรียงจาก LSB to MSB, Sub-Index 1 byte และตามด้วย Error code 1-4 bytes ในกรณีที่เกิด Error ขึ้น

รูปแบบที่สองเป็นรูปแบบสำหรับการ Read Access ซึ่งประกอบด้วย 2 ส่วนย่อยดังต่อไปนี้

- Host send: CMD จำนวน 1 byte ที่บอกว่าจะทำการอ่านข้อมูล, Index 2 bytes โดยเรียงจาก LSB to MSB, Sub-Index 1 byte และตามด้วย Reserved 4 bytes
- Slave answers: RES จำนวน 1 byte ที่จะบอกจำนวนข้อมูลที่จะส่ง หรือบอกว่าเกิดข้อผิดพลาด, Index 2 bytes โดยเรียงจาก LSB to MSB, Sub-Index 1 byte และตาม

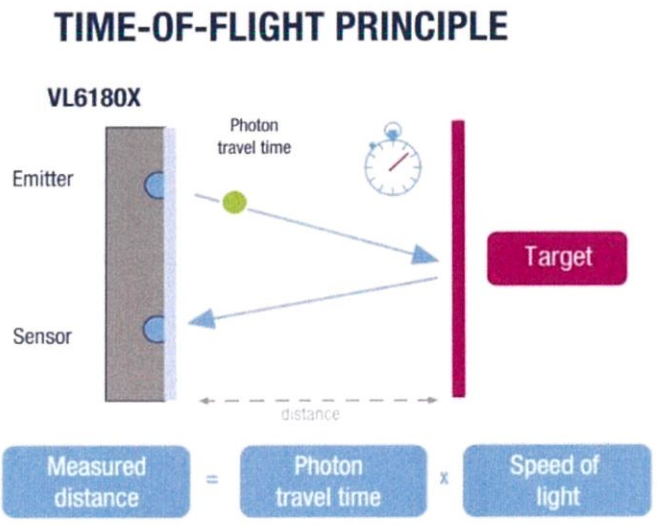
ด้วย Data 1-4 bytes หรือ Error code 1-4 bytes ในกรณีที่เกิด Error ขึ้น โดยเรียงจาก LSB to MSB

2.1.5. I^2C Communication

เป็นรูปแบบการสื่อสารที่ออกแบบมาเพื่อการสื่อสารที่ไม่ต้องการความเร็วในการส่งข้อมูล สูงมาก และระยะทางไม่ไกลมากนัก โดยอาจจะมีอุปกรณ์ที่ทำหน้าที่เป็น Master ได้หลายตัวพร้อมกันโดยใช้สายสัญญาณเพียงสองเส้นคือ SDA (Serial Data) และ SCL (Serial clock) ซึ่งความเร็วในการส่งข้อมูลจะแบ่งตาม Mode ของการสื่อสารได้ดังนี้ Low speed: 10kbit/s, Standard mode: 100kbit/s, Full speed: 400kbit/s, Fast mode: 1Mbit/s และ High speed: 3.2Mbit/s โดยจะมีการแบ่ง Address ของอุปกรณ์เป็นสองรูปแบบคือ แบบ 7 bits และแบบ 10 bits ซึ่งจะขึ้นอยู่กับตัวอุปกรณ์ว่ากำหนดมาเป็นแบบใด โดยผู้วิจัยได้ใช้อุปกรณ์ที่มี Address จำนวน 7 bits และมีค่า I^2C Clock speed สูงสุด 400kHz แต่เนื่องจากไมโครคอนโทรลเลอร์มี Clock speed ต่ำ ผู้วิจัยจึงปรับค่า I^2C clock speed ลดลงเหลือเพียง 300kHz ซึ่งเป็นค่าที่ยังคงทำให้ไมโครคอนโทรลเลอร์สามารถสื่อสารกับตัวเซ็นเซอร์ได้

2.1.6. Optical sensing

เป็นเทคโนโลยีที่ใช้ค่าแสงที่วัดได้ หรือการเปลี่ยนแปลงของแสงแปลงเป็นสัญญาณอิเล็กทรอนิกส์ ซึ่งตัวเซ็นเซอร์ VL6180X นั้น ได้นำหลักการนี้มาใช้ร่วมกับเทคโนโลยี Time of Flight โดยมีหลักการในการคำนวนหาระยะทางดังนี้



ภาพที่ 2.1 การคำนวนระยะทางของ

บทที่ 3
วิธีดำเนินงาน

3.1. วิธีดำเนินการทำงาน

3.1.1. รับทราบความต้องการของระบบ

การที่จะได้มาซึ่งความต้องการของระบบนั้น ทางบริษัทได้มีการจัดการประชุมหารือกันระหว่างแต่ละฝ่ายว่าระบบใหม่ที่จะจัดทำขึ้นมานั้น จำเป็นจะต้องมีความสามารถอะไรบ้าง โดยตัวผู้วิจัยเองได้รับทราบความต้องการของระบบมาจากทางพี่เลี้ยง ซึ่งเป็นผู้ที่ได้เข้าร่วมการประชุมเพื่อรับทราบความต้องการต่างๆของระบบโดยตรง ซึ่งพี่เลี้ยงจะเป็นผู้ที่คอยดูแล และประเมินผลงานที่ผู้วิจัยได้ทำขึ้นตลอดช่วงเวลาที่ผู้วิจัยได้ทำสหกิจศึกษา

3.1.2. ปรับแต่งความต้องการของระบบ

ในการทำโปรเจคแต่ละโปรเจคจะต้องใช้ระยะเวลาในการดำเนินงานนาน ทางแผนกจะมีการประชุมงานกันอยู่เรื่อยๆ เพื่อรับทราบ อัพเดทความต้องการ และรับทราบปัญหาต่างๆที่เกิดขึ้นระหว่างการทำโปรเจคเพื่อหาข้อสรุปที่ดีที่สุดสำหรับตัวโปรเจคนั้นๆ ซึ่งทางพี่เลี้ยงของผู้วิจัยได้มีส่วนร่วมในการประชุมแต่ละครั้ง เพื่อรับทราบปัญหาต่างๆที่เกิดขึ้นระหว่างช่วงการทำโปรเจค เมื่อประชุมในแต่ละครั้งเสร็จ ทางพี่เลี้ยงจะมาแจ้งปัญหา และอัพเดทความต้องการต่างๆของระบบให้กับผู้วิจัย เพื่อให้ตัวโปรเจคที่ออกมามีคุณภาพตามความต้องการของบริษัท

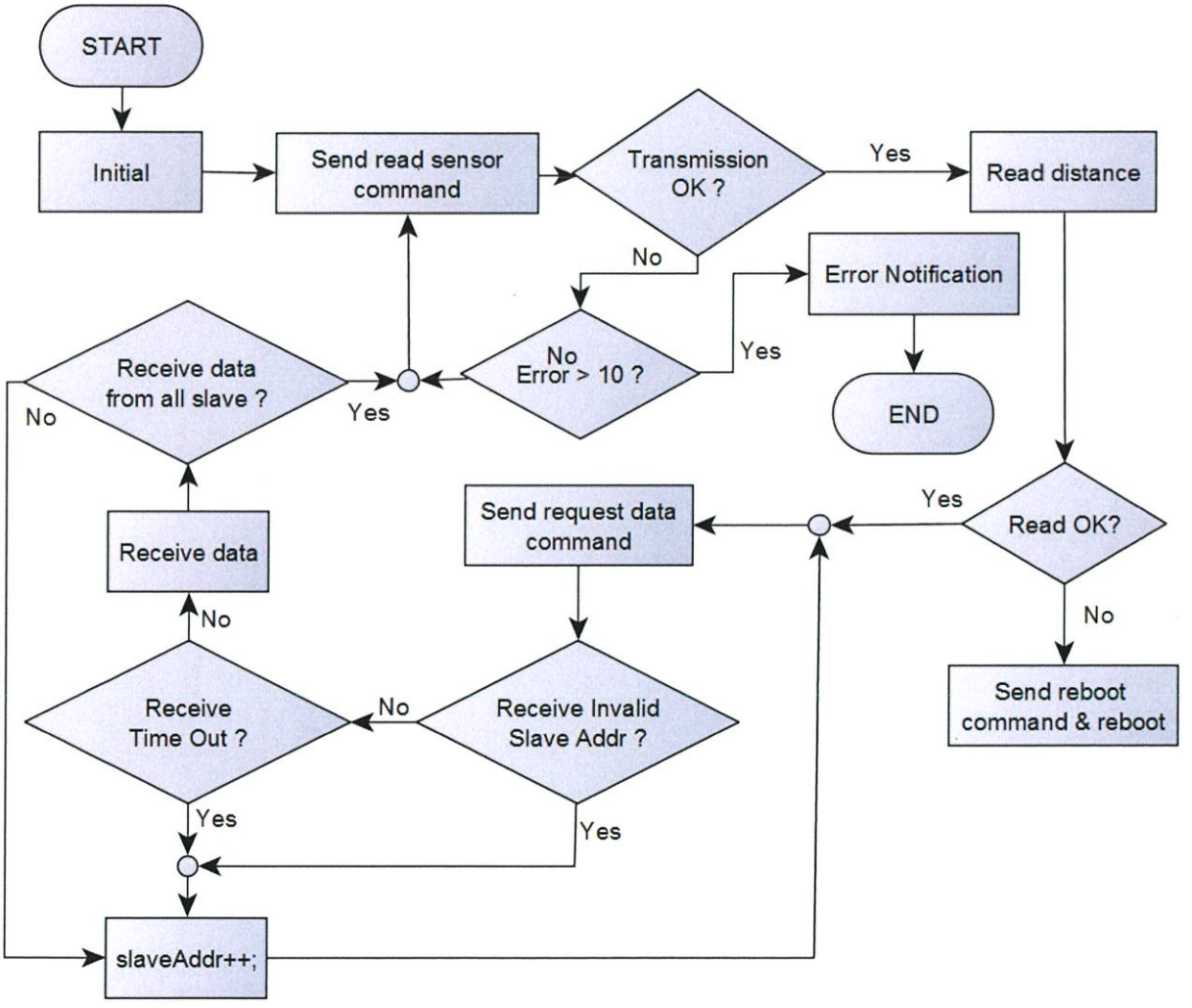
3.1.3. ศึกษาการทำงานของฟังก์ชันต่างๆของระบบ

เมื่อผู้วิจัยได้รับทราบความต้องการของตัวโปรเจคแล้ว ผู้วิจัยได้ทำการหารือกับทางพี่เลี้ยงว่าตัวโปรเจค ควรจะมีฟังก์ชันการทำงานเป็นอย่างไรบ้าง เมื่อทราบความต้องการของระบบแล้ว ผู้วิจัยได้ทำการศึกษาการทำงานของแต่ละฟังก์ชันภายในระบบว่า จะต้องประกอบด้วยอะไรบ้าง ใช้ความรู้อะไรบ้างในการดำเนินการ เพื่อที่ผู้วิจัยจะสามารถเขียนโปรแกรมเพื่อควบคุมการทำงานของแต่ละฟังก์ชันได้อย่างมีประสิทธิภาพ

3.1.4. ออกแบบ และปรับปรุง Flowchart ของระบบ

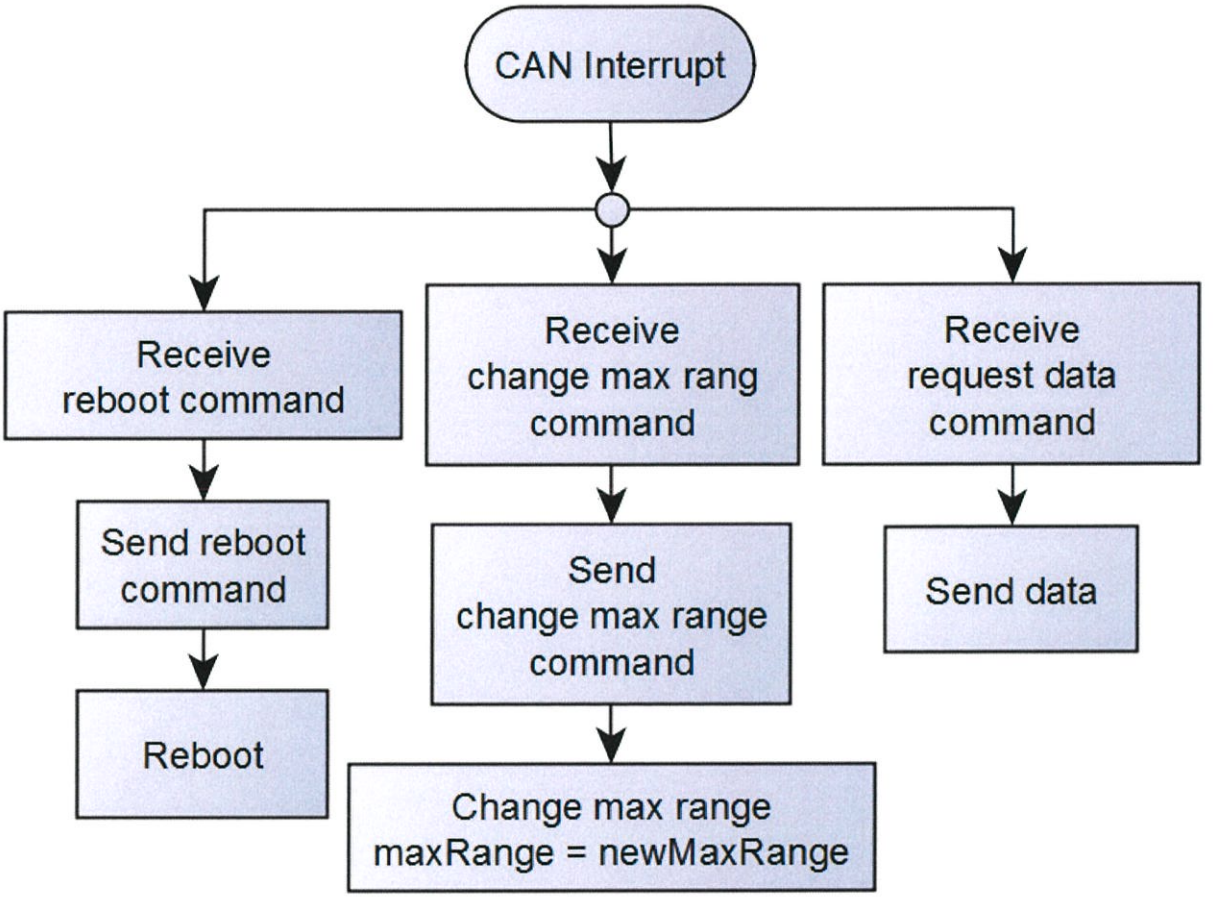
เมื่อรับทราบฟังก์ชันการทำงานในส่วนย่อยๆของตัวระบบแล้ว ผู้วิจัยได้ทำการออกแบบขั้นตอนของการทำงานภายในระบบว่าต้องมีขั้นตอนในการทำงานอย่างไรบ้าง โดยระหว่างการออกแบบ ผู้วิจัยได้มีการปรึกษากับทางพี่เลี้ยงอยู่เรื่อยๆ และทำการปรับปรุงเพื่อให้ผลงานที่ออกมาเป็นไปอย่างถูกต้องตามความต้องการของระบบ โดยได้รูปแบบของการทำงานดังนี้

- Flowchart ของ Master board (Main process)



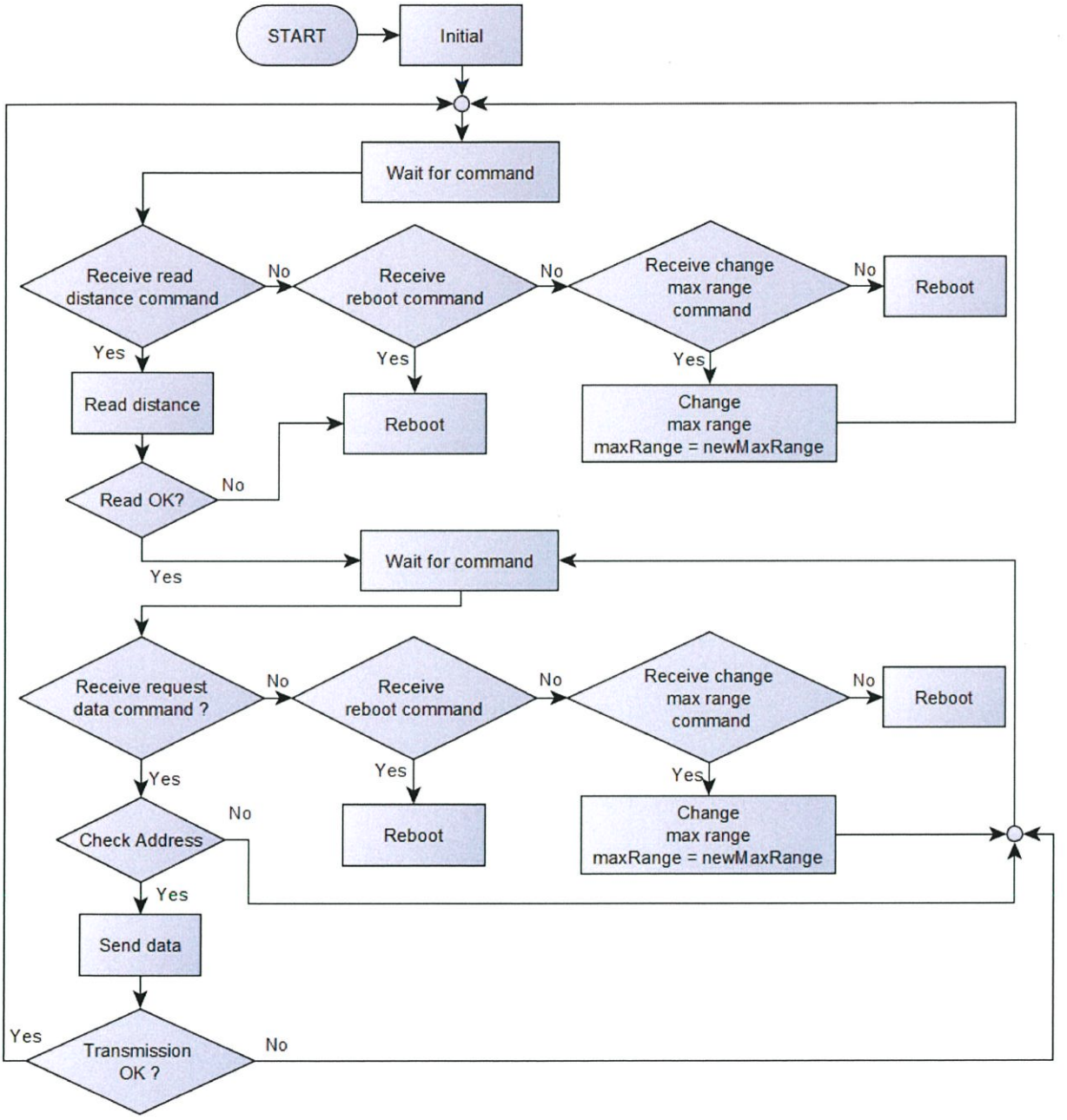
ภาพที่ 3.1 (ก) Flowchart ของ Master board (Main process)

- Flowchart ของ Master board (Interrupt process)



ภาพที่ 3.1 (ข) Flowchart ของ Master board (Interrupt process)

- Flowchart ของ Slave board



ภาพที่ 3.2 Flowchart ของ Slave board

3.1.5. ศึกษาการทำงานของตัวอุปกรณ์

เมื่อทำการออกแบบการทำงานของตัวระบบเสร็จแล้ว ผู้วิจัยได้ทำการศึกษาตัวอุปกรณ์ที่จะนำมาใช้ภายในระบบ ซึ่งอุปกรณ์แต่ละชิ้นที่ผู้วิจัยศึกษา จะต้องนำข้อมูลของแต่ละอุปกรณ์มาปรึกษากับทางพี่เลี้ยงก่อนที่จะทำการสั่งซื้อมาทำการทดลองใช้งาน และทดสอบประสิทธิภาพเพื่อประหยัดงบประมานในการสั่งซื้ออุปกรณ์ และคัดกรองอุปกรณ์เพื่อหาอุปกรณ์ที่เหมาะสมสำหรับการทำงานภายในระบบ

3.1.6. สั่งซื้ออุปกรณ์

การสั่งซื้ออุปกรณ์นั้นมีขั้นตอนในการสั่งซื้อหลายขั้นตอนซึ่งเป็นไปตามระบบของทางบริษัท และต้องใช้เวลานานในการดำเนินการ ซึ่งทางพี่เลี้ยงเป็นฝ่ายคอยดูแลเรื่องการสั่งซื้ออุปกรณ์สำหรับการทำโปรเจคของผู้วิจัย

3.1.7. เขียนโปรแกรม

ผู้วิจัยได้ทำการเริ่มเขียนโปรแกรมจำลองการทำงานของตัวระบบโดยใช้อุปกรณ์ที่ทางบริษัทมีอยู่เพื่อทดสอบการทำงานของโปรแกรมในส่วนต่างๆของระบบ เนื่องการจากสั่งซื้อตัวอุปกรณ์ที่จะนำมาใช้จริง ต้องใช้ระยะเวลาในการสั่งซื้ออุปกรณ์นาน

ตัวโปรแกรมจำลองที่ผู้วิจัยได้ทำการพัฒนาขึ้นได้อ้างอิงจากความสามารถที่ตัวระบบต้องการ โดยหลังจากที่ผู้วิจัยได้ทำการเขียนโปรแกรมจำลองเสร็จสิ้น ผู้วิจัยได้ทำการทดสอบการทำงานของโปรแกรมจำลองนั้น และทำการเขียนโปรแกรมใหม่สำหรับใช้งานจริง โดยเปลี่ยนรุ่นของไมโครคอนโทรลเลอร์เป็นรุ่นที่ใช้งานจริงในระบบ และเปลี่ยนฟังก์ชันการทำงานของการอ่านค่าเซ็นเซอร์ และ Protocol ในการสื่อสารระหว่างไมโครคอนโทรลเลอร์กับเครื่องคอมพิวเตอร์ที่ใช้ควบคุมระบบให้เป็นมาตรฐานตามรูปแบบที่บริษัทใช้

หลังจากที่ได้รับอุปกรณ์ที่จะใช้ในงานจริงมาแล้ว ผู้วิจัยได้ทำการปรับเปลี่ยนการตั้งค่า Pin ของ Microcontroller ใหม่ ให้ตรงตามที่จะใช้งานจริง และได้ทำการปรับแก้โปรแกรมบางส่วนเพื่อให้สามารถใช้งานตัวเซ็นเซอร์จริงได้

3.1.8. ทดสอบฟังก์ชันต่างๆ

ผู้วิจัยได้ทำการแบ่งการทดสอบฟังก์ชันการทำงานของโปรเจคเป็น 2 ส่วนใหญ่ๆ คือ

ส่วนแรก เป็นระบบที่ผู้วิจัยได้ทำการทดสอบการทำงานแบบจำลองของตัวงาน คือการทดสอบฟังก์ชันที่สร้างขึ้นมาเพื่อทดลองใช้งานการตั้งค่า และคำสั่งต่างๆที่ต้องนำไปใช้จริงในตัวงาน โดยแบ่งเป็นฟังก์ชันย่อยๆตามการทำงานในส่วนต่างๆดังต่อไปนี้

- การสื่อสารด้วย UART เพื่อนำไปพัฒนาใช้กับการสื่อสารระหว่าง Master กับ Slave ภายในระบบ

  โดยผู้วิจัยได้ทำการทดลองรับ-ส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ ซึ่งผลของการทดลองใช้งานคือ ฟังก์ชันการสื่อสารด้วย UART สามารถทำได้ตามปกติ
- การสื่อสารด้วย CAN Bus และการใช้งาน CAN Interrupt เพื่อนำไปพัฒนาใช้กับการสื่อสารระหว่างเครื่องคอมพิวเตอร์ที่ควบคุมระบบกับตัว Master

  โดยผู้วิจัยได้ทำการศึกษาการตั้งค่าที่จำเป็นต่อการใช้งานการสื่อสารผ่าน CAN Bus ของตัวไมโครคอนโทรลเลอร์จากอินเตอร์เน็ต และขอคำแนะนำจากทางพี่เลี้ยง ซึ่งภายหลังจากการค้นหาข้อมูลของการตั้งค่าที่จำเป็นก่อนการใช้งานการสื่อสารผ่าน CAN Bus และได้ทำการทดลองใช้งานโดยการให้คอมพิวเตอร์ส่งคำสั่งไปยังตัวไมโครคอนโทรลเลอร์เพื่อให้เปลี่ยนการตั้งค่าของระบบ สั่งให้ไมโครคอนโทรลเลอร์ Reboot และการสั่งให้ไมโครคอนโทรลเลอร์ส่งค่าที่อ่านได้จากเซ็นเซอร์มายังเครื่องคอมพิวเตอร์ ซึ่งผลการทดสอบคือ ฟังก์ชันการสื่อสารผ่าน CAN Bus สามารถทำงานได้ตามปกติ
- การสั่งการเซ็นเซอร์ผ่าน I^2C เพื่ออ่านค่าจากตัวเซ็นเซอร์

  โดยเซ็นเซอร์ที่ผู้วิจัยในทำการทดลองใช้คือเซ็นเซอร์ที่สามารถวัดค่า Helium และอุณหภูมิได้ ซึ่งการตั้งค่าการสื่อสารต่างๆที่จำเป็นมีอยู่ใน Datasheet ของตัวเซ็นเซอร์ ภายหลังจากการทดลองสั่งการให้ตัวเซ็นเซอร์ส่งค่าที่อ่านได้มายังตัวไมโครคอนโทรลเลอร์พบว่า ฟังก์ชันสำหรับสั่งการตัวเซ็นเซอร์สามารถใช้งานได้ตามปกติ

ส่วนที่สอง เป็นระบบที่จะนำไปใช้งานจริง โดยมีการทำงานตามความต้องการของบริษัท ซึ่งมีการทดสอบ และผลของการทดสอบในส่วนต่างๆดังต่อไปนี้

- การสื่อสารด้วย UART ที่เลือกใช้รูปแบบ RS-485

  โดยผู้วิจัยได้ทำการทดลองด้วยการใช้ Master ส่งข้อมูลไปค่าหนึ่ง แล้วให้ Slave ตอบกลับมาว่าค่าที่ได้รับคือค่าอะไร ซึ่งผลการทดลองคือ การสื่อสารด้วย UART ที่เลือกใช้รูปแบบ RS-485 สามารถใช้งานได้อย่างถูกต้อง
- การสื่อสารผ่าน CAN Bus และใช้งาน CAN Interrupt โดยใช้ Protocol CAN Open

  ซึ่งผู้วิจัยได้ทำการทดลองโดยใช้เครื่องคอมพิวเตอร์ส่งคำสั่ง Read data เพื่อทดลองให้ Master ส่งค่าๆหนึ่งกลับมายังเครื่องคอมพิวเตอร์ ซึ่งผลการทดลองคือ การสื่อสารผ่าน CAN Bus และใช้งาน CAN Interrupt โดยใช้ Protocol CAN Open สามารถทำงานได้อย่างถูกต้อง
- การสั่งการเซ็นเซอร์ผ่าน I^2C เพื่ออ่านค่าระยะหางของวัตถุ

  โดยผู้วิจัยได้ทำการทดลองใช้ตัวเซ็นเซอร์ VL6180X ซึ่งเป็นเซ็นเซอร์แสงที่จะนำไปใช้งานภายในระบบ โดยให้บอร์ดไมโครคอนโทรลเลอร์ทดลองอ่านค่าระยะหางระหว่างวัตถุกับตัวบอร์ด ซึ่งค่าที่อ่านได้จะขึ้นอยู่กับความสามารถในการสะท้องแสงของวัตถุ หากวัตถุมีความสามารถในการสะท้อนแสงมาก ค่าที่ตัวเซ็นเซอร์อ่านได้จะมีค่าตรงกับค่าจริงมากกว่าวัตุที่มีความสามารถในการสะท้อนแสงได้น้อย ผู้วิจัยได้ทำการทดลองเปรียบเทียบค่าที่ตัวเซ็นเซอร์อ่านได้กับค่าจริง โดยใช้วัตถุที่มีผิวต่างกันสองชนิด คือ วัตถุขาวมัน และวัตถุดำด้าน ซึ่งผลการทดลองจะกล่าวไว้ในบทถัดไป

3.1.9. ปรับแต่งประสิทธิภาพของ Firmware

หลังจากที่ทำการทดสอบฟังก์ชันการทำงานต่างๆของระบบแล้ว ผู้วิจัยได้ทำการปรับแต่งประสิทธิภาพการทำงานของฟังก์ชันย่อยๆภายในระบบเพื่อให้ระบบสามารถทำงานได้อย่างมีประสิทธิภาพที่สุด

3.1.10. ทดสอบระบบทั้งหมด

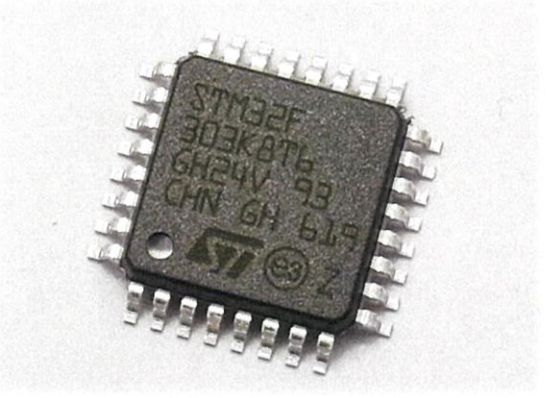
หลังจากที่ผู้วิจัยได้ทำการปรับแต่งฟังก์ชันการทำงานในส่วนย่อยๆแล้ว ผู้วิจัยได้ทำการทดสอบการทำงานโดยรวมของระบบอีกครั้งเพื่อให้แน่ใจว่าระบบสามารถทำงานได้อย่างถูกต้องตรงตามความต้องการของบริษัท โดยผลที่ได้คือ ตัว Firmware สามารถทำงานได้อย่างถูกต้อง

3.1.11. ออกแบบ PCB

หลังจากการทดสอบระบบทั้งหมดผู้วิจัยได้รับมอบหมายให้ทำการออกแบบ PCB ของบอร์ดที่จะนำไปใช้งานภายในระบบ ซึ่งผู้วิจัยได้ทำการศึกษา และค้นหาโปรแกรมสำหรับการออกแบบ PCB โดยผู้วิจัยเลือกใช้โปรแกรม PCBWeb ซึ่งเป็น Freeware ที่สามารถออกแบบ PCB และสร้าง Template สำหรับตัวอุปกรณ์ที่จำเป็นต้องใช้ได้เอง

ในการออกแบบ PCB ผู้วิจัยได้ทำการนำอุปกรณ์ต่างๆที่จำเป็นต่อการทำงานของตัวระบบ แล้วทำการสร้าง Layout และ Footprint ของตัวอุปกรณ์โดยอ้างอิงจาก Datasheet ของตัวอุปกรณ์นั้นๆ ซึ่งอุปกรณ์สำคัญหลักๆมีดังนี้

- STM32F303K8 Chip (Controller)

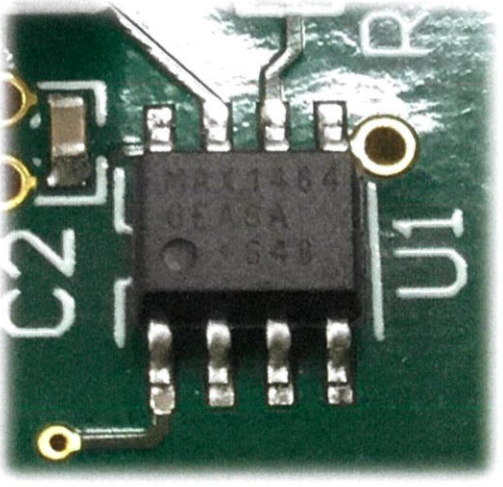


ภาพที่ 3.3 STM32F303K8

- VL6180X (Optical sensor)

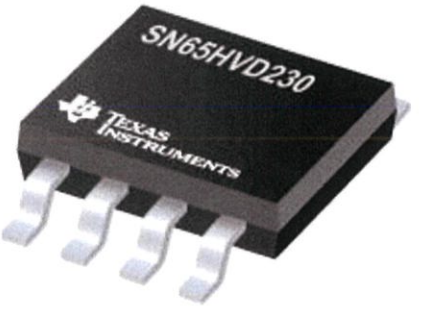
ภาพที่ 3.4 VL6180X

- MAX14840E (RS-485: Transceiver)



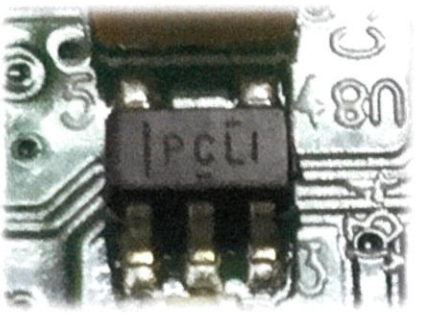
ภาพที่ 3.5 MAX14840EASA+

- SN65HVD230 (CAN: Transceiver)



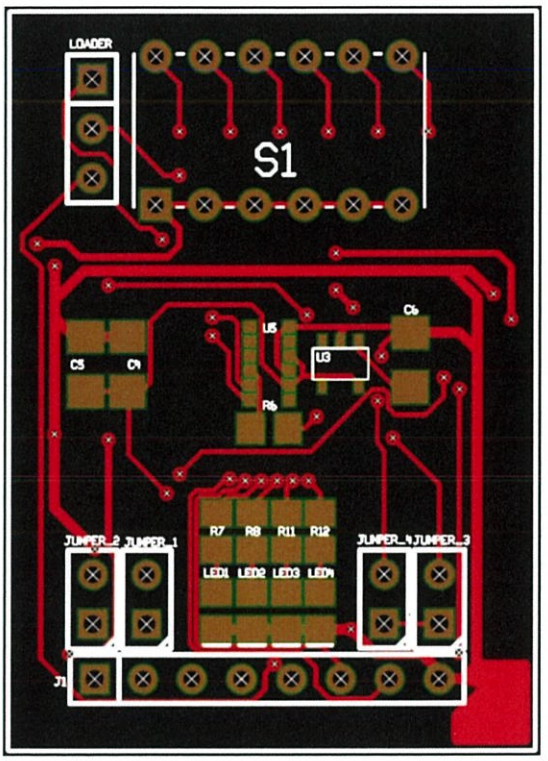
ภาพที่ 3.6 SN65HVD230

- TPS76928 (Regulator)

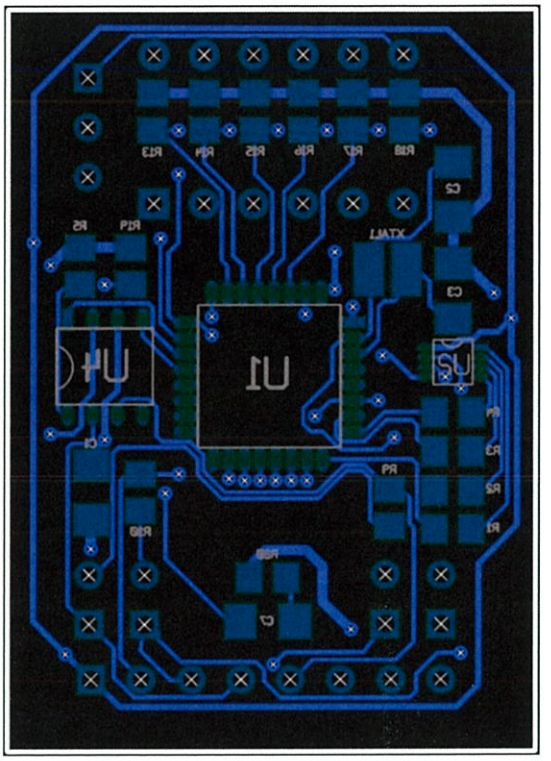


ภาพที่ 3.7 TPS76928

ภายหลังจากที่ผู้วิจัยได้ออกแบบ PC เสร็จ ผู้วิจัยได้ส่งให้ทางพี่เลี้ยงตรวจสอบข้อบกพร่อง เพื่อปรับแก้ให้เหมาะสมกับความต้องการมากที่สุด โดยได้ผลงานดังนี้



ภาพที่ 3.8 (ก) PCB Board Top



ภาพที่ 3.8 (ข) PCB Board Bottom

3.2. แผนการดำเนินงาน

ตารางที่ 3.2.1 แผนการดำเนินงาน

ลำดับ	หัวข้องาน	เดือนที่ 1				เดือนที่ 2				เดือนที่ 3				เดือนที่ 4				เดือนที่ 5			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	รับทราบความต้องการของระบบ																				
2	ปรับแต่งความต้องการของระบบ																				
3	ออกแบบการทำงานของฟังก์ชันต่างๆของระบบ																				
4	ศึกษาการทำงานของตัวอุปกรณ์																				
5	ออกแบบ Flowchart ของระบบ																				
6	ปรับปรุง Flowchart ของระบบ																				
7	สั่งซื้ออุปกรณ์																				
8	เขียนโปรแกรม																				
9	ทดสอบฟังก์ชันต่างๆ																				
10	ปรับแต่งประสิทธิภาพ																				
11	ทดสอบระบบทั้งหมด																				
12	ออกแบบ PCB																				
13	จัดทำรูปเล่มรายงาน																				

บทที่ 4
ผลการทำงาน

4.1. ผลการทดสอบการสื่อสารผ่าน CAN ในบอร์ดทดลอง (STM32F107VCT6)

จากการศึกษาการทำงานของ CAN Bus และการใช้งาน CAN Bus บนบอร์ดทดลองโดยใช้ Data rate 1Mbit/s ซึ่งในแต่ละ Node จะมี CAN ID เป็นตัวบ่งบอกถึงตำแหน่งของ Node นั้น โดย CAN Bus เป็นการสื่อสารแบบ Half-Duplex ซึ่งจะทำให้การสื่อสารในแต่ละครั้ง ทำได้เพียงรับ หรือส่งเท่านั้น จึงทำให้ลดโอกาสที่จะทำให้เกิดปัญหาข้อมูลชนกันภายในสายการสื่อสาร

โดยจากการที่ผู้วิจัยได้ทดสอบการใช้งานโดยการโปรแกรมการรับ-ส่งข้อมูลพื้นฐานลงบนตัวไมโครคอนโทรลเลอร์พบว่า การทำงานสามารถเป็นไปตามที่ผู้วิจัยคาดการณ์ไว้

4.2. ผลการทดสอบการสื่อสารผ่าน I^2C ในบอร์ดทดลอง (STM32F107VCT6)

จากการศึกษาการทำงานของ I^2C และการใช้งาน I^2C บนบอร์ดทดลองโดยการสื่อสารกับตัว Helium sensor ที่มีการสั่งการโดยใช้การสื่อสารแบบ I^2C โดยใช้ความถี่ 50kHz ซึ่งในการที่จะสั่งงานให้ตัวเซ็นเซอร์ทำงานนั้น จำเป็นจะต้องส่งคำสั่งตามที่ตัวเซ็นเซอร์ได้กำหนดไว้ใน Datasheet ผ่านทาง I^2C

โดยจากการทดลองโปรแกรมเพื่อติดต่อสื่อสารกับตัว Helium sensor ผ่านทาง I^2C ผู้วิจัยพบว่าการสื่อสารกับตัวเซ็นเซอร์สามารถทำงานได้

4.3. ผลการทดสอบการสื่อสารผ่าน UART ในบอร์ดทดลอง (STM32F107VCT6)

จากการศึกษาการทำงานของ UART และการใช้งาน UART บนบอร์ดทดลองโดยการสื่อสารระหว่างคอมพิวเตอร์กับบอร์ดทดลอง โดยการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ และบอร์ดทดลอง โดยมี Bit rate 115200 bit/s ซึ่งเป็นการทดลองทั้งรับ-ส่งข้อมูล พบว่าค่าที่รับ-ส่งระหว่างคอมพิวเตอร์กับบอร์ดทดลอง ถูกต้องตามที่ผู้วิจัยคาดการณ์ไว้

4.4. ผลการทดสอบการทำงานภาพรวมของระบบโดยใช้บอร์ดทดลอง (STM32F107VCT6)

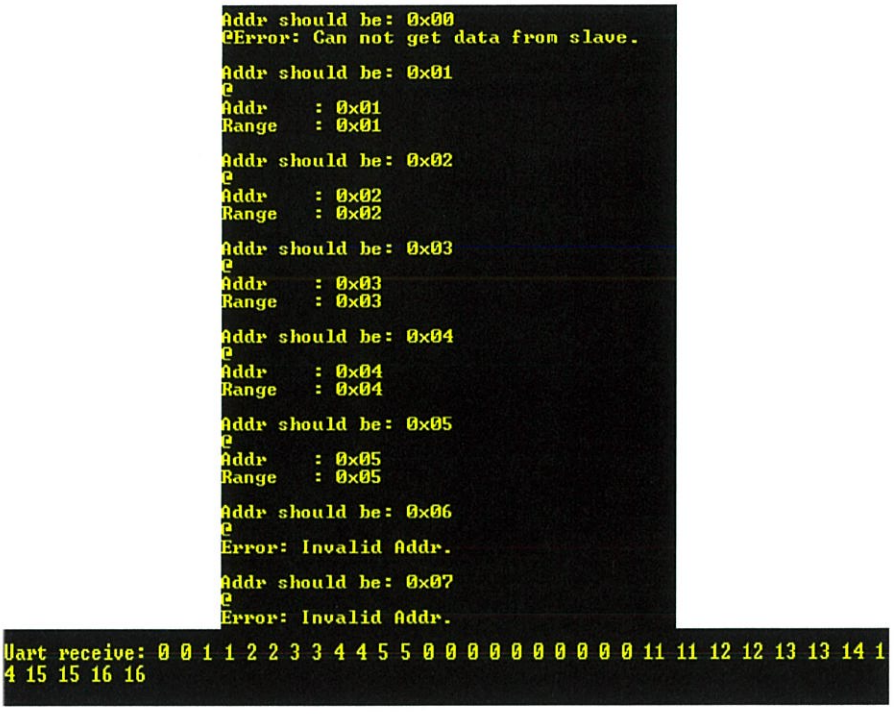
การทดสอบระบบครั้งนี้ ผู้วิจัยทดสอบเพื่อให้แน่ใจว่าการทำงานภาพรวมของฟังก์ชันในการทำงานของการสื่อสารแบบต่างๆ ถูกต้องตามที่ผู้วิจัยคาดการณ์ไว้ และสามารถจำลองการทำงานของระบบจริงได้ในระดับหนึ่ง ซึ่งการทำงานในระบบจริงต้องมีบอร์ดที่เป็นตัว Master ที่จะคอยสั่งงานบอร์ดที่เป็นตัว Slave ภายในเครือข่ายย่อย และบอร์ด Master ต้องรับคำสั่งจาก CAN Interrupt จากเครื่อง Monitor ภายในระบบใหญ่ทั้งหมดได้ ผู้วิจัยจึงแบ่งการทดสอบเป็น 2 รูปแบบ ดังนี้

ครั้งที่หนึ่ง ผู้วิจัยได้จำลองสถานการณ์ให้เครื่องคอมพิวเตอร์เป็นเสมือนตัว Master ในระบบ และตัวบอร์ดทดลองเป็นเสมือน Slave ในระบบ เพื่อจะทดสอบความสามารถของตัว Slave โดยมีรูปแบบการทำงานคือ Master จะส่งคำสั่งไปยังตัว Slave เพื่อให้ตัว Slave ไปอ่านค่าจากเซ็นเซอร์ซึ่งผู้วิจัยได้ใช้ตัว Helium sensor ในการทดลอง เมื่อ Slave ได้รับคำสั่ง จะมีการตรวจสอบว่าค่าที่ได้รับจากตัว Master ถูกต้องตามที่กำหนดไว้หรือไม่ หากค่าที่ได้รับมาถูกต้อง จะอ่านค่าจากตัวเซ็นเซอร์แล้วเก็บค่าไว้เพื่อรอคำสั่งต่อไป แต่หากค่าที่รับมาไม่ตรงไปตามที่กำหนด ตัว Slave จะทำการ reboot ตัวเองทันทีเพื่อป้องกันระบบค้าง ในกรณีที่ค่าคำสั่งที่ตัว Slave ได้รับถูกต้อง ตัว Slave จะรอคำสั่งขอค่าที่ได้จากการอ่านเซ็นเซอร์จากตัว Master เมื่อตัว Master ส่งคำสั่งขอค่าที่อ่านได้จากเซ็นเซอร์มายังตัว Slave ตัว Slave จะนำค่าที่ได้รับไปตรวจสอบว่าเป็นคำสั่งที่ถูกต้องหรือไม่ ถ้าถูกต้องจะทำการส่งค่าที่อ่านได้จากเซ็นเซอร์ไปยังตัว Master แล้วรอคำสั่งให้อ่านค่าจากเซ็นเซอร์ต่อไป แต่ถ้าค่าที่รับมาไม่ตรงไปตามที่กำหนด ตัว Slave จะ reboot ทันที การทำงานของตัว Slave จะเป็นแบบนี้วนไปเรื่อยๆ

Output Window

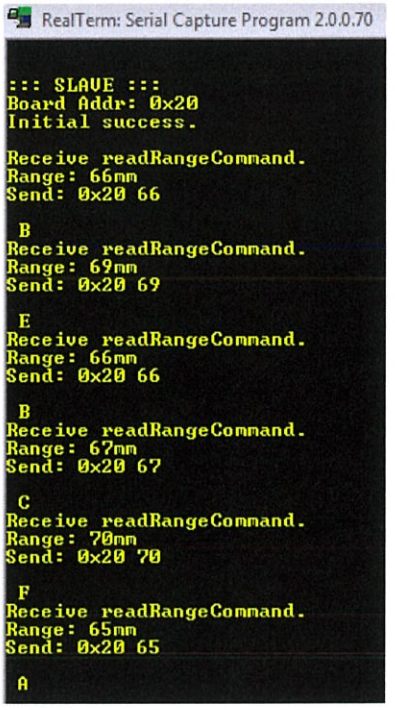
Chn	Identifier	Flg	DLC	D0	1	2	3	4	5	6	D7	Time	Dir
0	32		8	64	0	0	4	0	0	0	0	2143.304930	T
0	32		8	75	0	0	4	0	0	0	0	2143.305130	R
0	32		8	75	0	0	4	1	1	0	0	2143.305280	R
0	32		8	75	0	0	4	2	2	0	0	2143.305430	R
0	32		8	75	0	0	4	3	3	0	0	2143.305580	R
0	32		8	75	0	0	4	4	4	0	0	2143.305730	R
0	32		8	75	0	0	4	5	5	0	0	2143.305880	R
0	32		8	75	0	0	4	0	0	0	0	2143.306030	R
0	32		8	75	0	0	4	0	0	0	0	2143.306190	R
0	32		8	75	0	0	4	0	0	0	0	2143.306340	R
0	32		8	75	0	0	4	0	0	0	0	2143.306490	R
0	32		8	75	0	0	4	0	0	0	0	2143.306640	R
0	32		8	75	0	0	4	11	11	0	0	2143.306790	R
0	32		8	75	0	0	4	12	12	0	0	2143.306940	R
0	32		8	75	0	0	4	13	13	0	0	2143.307090	R
0	32		8	75	0	0	4	14	14	0	0	2143.307240	R
0	32		8	75	0	0	4	15	15	0	0	2143.307390	R
0	32		8	75	0	0	4	16	16	0	0	2143.307540	R
0	32		8	75	0	0	4	18	67	0	0	2143.307690	R

ภาพที่ 4.1 ผล CAN Interrupt ของ Master



ภาพที่ 4.2 ผล Debugging ของ Master

ครั้งที่สอง ผู้วิจัยได้จำลองสถานการณ์ให้เครื่องคอมพิวเตอร์เป็นเสมือนตัว Slave ในระบบ และตัวบอร์ดทดลองเป็นเสมือนตัว Master ในระบบ เพื่อจะทดสอบความสามารถของตัว Master โดยมีรูปแบบการทำงานคือ ตัว Master จะทำการ broadcast คำสั่งไปยังตัว Slave ทุกตัว เพื่อให้ตัว Slave ไปอ่านค่าจากเซ็นเซอร์ หากทำการ broadcast ไม่สำเร็จ Master จะทำการ broadcast ใหม่ไปเรื่องๆ จนครบ 10 ครั้ง หากยังไม่สามารถ broadcast ได้ master จะทำการ แสดงไฟ error status และหยุดระบบเพื่อรอรับการแก้ไข แต่ถ้าหากทำการ broadcast คำสั่งสำเร็จ ตัว Master จะอ่านค่าจากเซ็นเซอร์ เมื่อ Master อ่านค่าจากเซ็นเซอร์เสร็จแล้ว จะทำการส่งคำสั่งขอค่าที่อ่านได้จากเซ็นเซอร์ไปยัง Slave ทุกตัว โดยแต่ละครั้งที่ส่งคำสั่งไป ตัว Master จะรอจนกว่า Slave ตัวที่ได้รับคำสั่งนั้นตอบกลับ และเมื่อได้ข้อมูลมาแล้ว Master จะทำการเก็บข้อมูลนั้นไว้ เพื่อรอการส่งต่อไปยังเครื่อง Monitor ภายในระบบ โดยมีการเช็ค Timeout ถ้าหาก Slave ไม่ได้ส่งข้อมูลมาภายในระยะเวลาที่กำหนด Master จะข้ามข้อมูลของ Slave ตัวนั้นไป เมื่อ Master ได้รับข้อมูลจาก Slave ทุกตัวภายในเครือข่ายของตัวเอง ตัว Master จะรอ Timer เพื่อส่งคำสั่งไปให้ตัว Slave อ่านค่าจากเซ็นเซอร์ อีกครั้ง การทำงานของตัว Master จะเป็นแบบนี้วนไปเรื่อยๆ โดยในระหว่างการทำงานของตัว Master นั้น หากมีการส่ง CAN Interrupt มาจากเครื่อง Monitor เพื่อขอข้อมูล ตัว Master จะทำการส่งข้อมูลครั้งสุดท้ายที่ตัว Master มีการทำงานไปให้เครื่อง Monitor โดยผ่าน CAN Bus



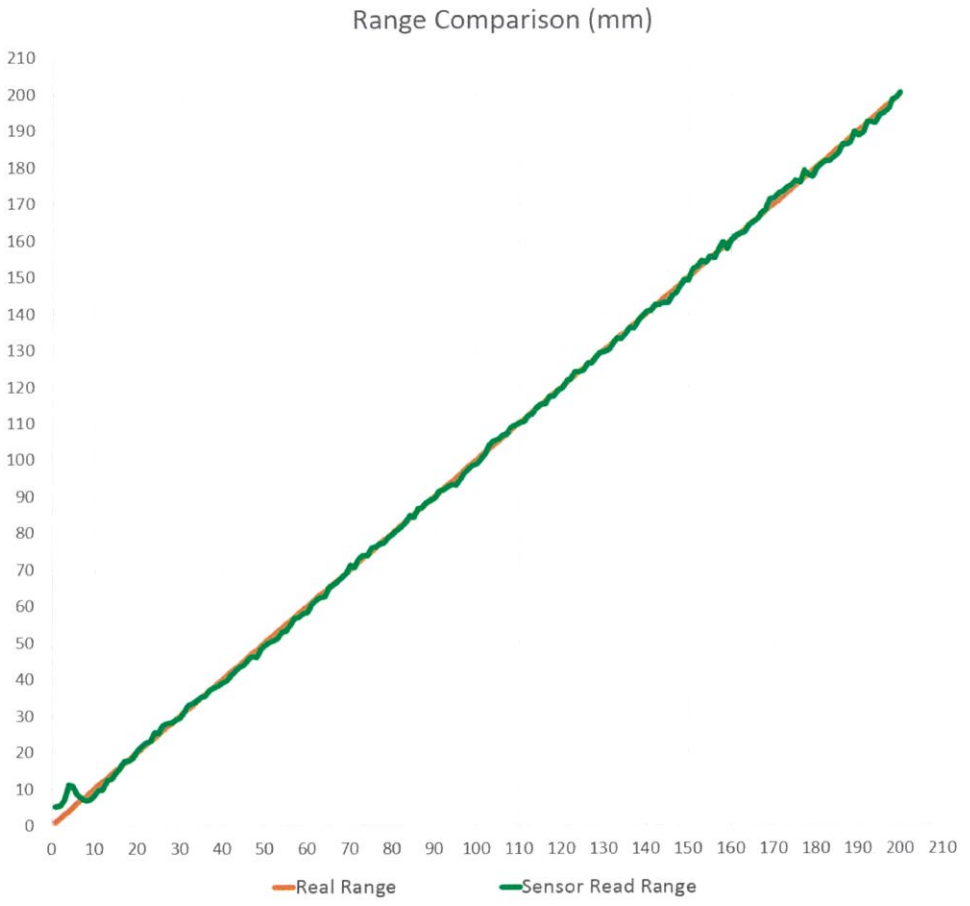
ภาพที่ 4.3 ผล Debugging ของ Slave

โดยจากทดสอบภาพรวมของระบบจำลองนี้ พบว่า การทำงานของระบบจำลอง เป็นไปอย่างถูกต้องตามที่ผู้วิจัยกำหนดไว้

4.5. ผลการทดสอบการทำงานของตัวเซ็นเซอร์แสง

การทดลองครั้งนี้ทำเพื่อตรวจสอบความสามารถในการสะท้อนของวัตถุที่มีสีของพื้นผิวที่แตกต่างกันสองชนิดคือ ชนิดแรกจะเป็นวัตถุที่มาสีขาวมัน ซึ่งผู้วิจัยได้ทำการทดลองวัดในระยะ 0-200 mm โดยทุกๆ 1 mm จะทำการอ่านค่าเซ็นเซอร์ 10 รอบแล้วนำค่าที่ได้มาเฉลี่ยเพื่อนำไปวาดกราฟเปรียบเทียบกับค่าจริง และชนิดที่สองคือวัตถุที่จะใช้ในงานจริงซึ่งมีพื้นผิดที่เป็นสีดำด้าน ซึ่งผู้วิจัยได้ทำการทดลองวัดในระยะ 0-200 mm โดยทุกๆ 1 mm จะทำการอ่านค่าเซ็นเซอร์ 10 รอบแล้วนำค่าที่ได้มาเฉลี่ยเพื่อนำไปวาดกราฟเปรียบเทียบกับค่าจริง

ผลการทดสอบของวัตถุชนิดแรก คือตัวเซ็นเซอร์มีความสามารถในการตรวจจับวัตถุได้ดี และเมื่อนำค่าที่เซ็นเซอร์วัดได้มาเปรียบเทียบกับค่าจริงได้ผลดังนี้



ภาพที่ 4.4 ผลการทดสอบอ่านค่าเซ็นเซอร์กับวัตถุสีขาวมัน

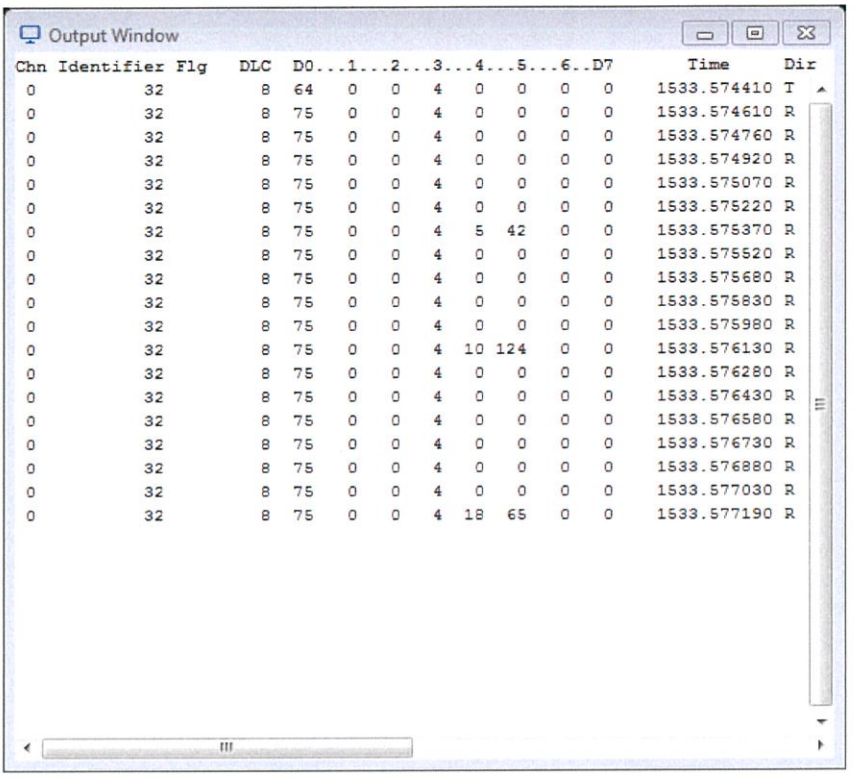
ผลการทดสอบของวัตถุชนิดที่สอง คือตัวเซ็นเซอร์มีความสามารถในการอ่านค่าได้ไม่ค่อยดีนัก และเมื่อนำค่าที่เซ็นเซอร์วัดได้มาเปรียบเทียบกับค่าจริงได้ผลดังนี้

ภาพที่ 4.5 ผลการทดสอบอ่านค่าเซ็นเซอร์กับวัตถุสีดำด้าน

4.6. ผลการทดสอบการทำงานภาพรวมของระบบด้วยบอร์ดจริง (STM32F303K8)

การทดลองในครั้งนี้ ผู้วิจัยได้ทำการเปลี่ยนบอร์ดไมโครคอนโทรลเลอร์จาก STM32F107VCT6 เป็นบอร์ดไมโครคอนโทรลเลอร์ STM32F303K8 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่เหมาะสมกับงานที่จะนำไปใช้มากกว่า STM32F042K6 โดยรูปแบบการทำงานจะมีขั้นตอนเหมือนกับ "การทดลองที่ 4.4" ซึ่งจะแบ่งการทำงานออกเป็นสองส่วนคือ ส่วนของ Master และส่วนของ Slave แต่เปลี่ยนจากการอ่านค่าเซ็นเซอร์จากที่อ่านค่า Helium เป็นอ่านค่าระยะห่างจากตัวเซ็นเซอร์ไปยังวัตถุที่ใช้งานจริงภายในระบบ ซึ่งต้องใช้ความถี่ของ I^2C เป็น 250kHz เนื่องจากความถี่สูงสุดที่ 400kHz ที่ได้บอกไว้ใน Datasheet นั้น

ไม่สามารถทำงานร่วมกับ Clock speed ของตัวไมโครคอนโทรลเลอร์ที่จะนำไปใช้ในงานจริงได้ และหลังจากการได้ทดลองให้ระบบทำงาน 100 รอบ พบว่าใช้เวลาในการทำงานประมาน 20 วินาที ดังนั้นระบบสามารถทำงานได้ในความเร็วประมาณ 5 รอบต่อ 1 วินาที



Chn	Identifier	Flg	DLC	D0	1	2	3	4	5	6	D7	Time	Dir
0	32		8	64	0	0	4	0	0	0	0	1533.574410	T
0	32		8	75	0	0	4	0	0	0	0	1533.574610	R
0	32		8	75	0	0	4	0	0	0	0	1533.574760	R
0	32		8	75	0	0	4	0	0	0	0	1533.574920	R
0	32		8	75	0	0	4	0	0	0	0	1533.575070	R
0	32		8	75	0	0	4	0	0	0	0	1533.575220	R
0	32		8	75	0	0	4	5	42	0	0	1533.575370	R
0	32		8	75	0	0	4	0	0	0	0	1533.575520	R
0	32		8	75	0	0	4	0	0	0	0	1533.575680	R
0	32		8	75	0	0	4	0	0	0	0	1533.575830	R
0	32		8	75	0	0	4	0	0	0	0	1533.575980	R
0	32		8	75	0	0	4	10	124	0	0	1533.576130	R
0	32		8	75	0	0	4	0	0	0	0	1533.576280	R
0	32		8	75	0	0	4	0	0	0	0	1533.576430	R
0	32		8	75	0	0	4	0	0	0	0	1533.576580	R
0	32		8	75	0	0	4	0	0	0	0	1533.576730	R
0	32		8	75	0	0	4	0	0	0	0	1533.576880	R
0	32		8	75	0	0	4	0	0	0	0	1533.577030	R
0	32		8	75	0	0	4	18	65	0	0	1533.577190	R

ภาพที่ 4.6 ผลการทดสอบ CAN Interrupt ของระบบ

ในการทดสอบ ผู้วิจัยได้ทำการเปิดระบบทิ้งไว้เป็นเวลา 3 วัน เพื่อทดสอบความสามารถในการใช้งานของตัว Firmware ผลทดสอบภาพรวมของระบบนี้ พบว่า การทำงานของระบบ เป็นไปอย่างปกติโดยไม่เกิดข้อผิดพลาดใดๆ

บทที่ 5
สรุปผลการทำงาน และข้อเสนอแนะ

5.1 บทสรุป

ภายหลังจากที่ผู้วิจัยได้ทำการศึกษาหาข้อมูลที่จำเป็นต่อการทำโครงงานครั้งนี้ แล้วทำการเขียน Firmware และทดสอบการทำงานของตัว Firmware โดยการเปิดให้ระบบทำงานติดต่อกันเป็นเวลา 3 วัน เพื่อทดสอบความเสถียรในการทำงานพบว่า การทำงานของตัว Firmware เป็นไปอย่างถูกต้องตามความต้องการของระบบ และภายหลังจากที่ผู้วิจัยได้ออกแบบ PCB และส่งให้ทางพี่เลี้ยงตรวจสอบ ได้ผลว่า PCB ที่ออกแบบนั้น เป็นที่พึงพอใจต่อทางพี่เลี้ยง

5.2 ปัญหาอุปสรรค และแนวทางแก้ไข

5.2.1 การสั่งซื้ออุปกรณ์ใช้เวลานานเนื่องจากต้องเป็นไปตามขั้นตอนของทางบริษัท แนวทางการแก้ปัญหาคือ ทำการหารายชื่ออุปกรณ์ที่ต้องใช้ให้เร็วขึ้น

5.2.2 อุปกรณ์ที่สั่งซื้อมาไม่ตรงตามที่ต้องการเนื่องจาก Manufacturer part number ของอุปกรณ์ไม่ตรงกับตัวอุปกรณ์ที่ต้องการ แนวทางการแก้ปัญหาคือ ตรวจสอบ Manufacturer part number ให้ละเอียดก่อนทำการสั่งซื้อ

5.3 แนวทางการพัฒนาต่อ

5.3.1 เพิ่มความถี่ในการตรวจสอบวัตถุภายในช่องเก็บของ

5.3.2 พัฒนาระบบให้ใช้ร่วมกับการวัดค่าความสว่าง และการตรวจจับรูปแบบการเคลื่อนที่ของวัตถุเพื่อเพิ่มความแม่นยำในการตรวจจับวัตถุให้มีประสิทธิภาพสูงขึ้น

5.3.3 พัฒนารูปแบบการแจ้งเตือนเมื่อเกิดข้อผิดพลาดระหว่างการทำงาน เพื่อการแก้ปัญหาที่ดีขึ้น

บรรณานุกรม

Erich Stygrer. *STMicroelectronics VL6180X Time-of-Flight LIDAR Sensor.* Retrieved October 11, 2017. https://mcuoneclipse.com/2016/12/03/tutorial-stmicroelectronics-vl6180x-time-of-flight-lidar-sensor/

Le Tan Phuc. (2017). *STM32F0 Tutorial 7: I2C example Part 1.* Retrieved October 22, 2017. https://www.youtube.com/watch?v=v1_f5u5pChs

Mic De Duiwel. (2016). *STM32F429 HAL library CAN_Transmit timeout.* Retrieved October 9, 2017. https://electronics.stackexchange.com/questions/261430/stm32f429-hal-library-can-transmit-timeout

QUICK REFERENCE FOR RS485, RS422, RS232 AND RS423. Retrieved November 9, 2017. http://www.rs485.com/rs485spec.html

STM32F303 User manual. Retrieved November 22, 2017. http://www.st.com/content/ccc/resource/technical/document/user_manual/e3/0e/88/05/e8/74/43/a0/DM00231744.pdf/files/DM00231744.pdf/jcr:content/translations/en.DM00231744.pdf

The Basics of CANopen. Retrieved November 14, 2017. http://www.ni.com/white-paper/14162/en/

The I2C Clock Speed. Retrieved November 14, 2017. https://www.i2c-bus.org/speed/

VL6180X basic ranging application note. Retrieved October 26, 2017. https://cdn.sparkfun.com/datasheets/Sensors/Proximity/VL6180_ApplicationNote.pdf

VL6180X range and ambient light sensor quick setup guide. Retrieved October 26, 2017. http://www.st.com/content/ccc/resource/technical/document/design_tip/4c/30/46/13/89/4e/41/d5/DM00123019.pdf/files/DM00123019.pdf/jcr:content/translations/en.DM00123019.pdf

VL6180X-SATEL Data brief. Retrieved November 27, 2017. http://www.st.com/content/ccc/resource/technical/document/data_brief/dc/9a/25/6c/bd/12/4a/08/DM00150183.pdf/files/DM00150183.pdf/jcr:content/translations/en.DM00150183.pdf

VL6180X Satellite Symbol. Retrieved November 8, 2017. https://easyeda.com/riggwelter/component/VL6180X_Satellite-N71jBuM4Y