เครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลัง

จัดทำโดย

นายธีระภัทร เจริญปรุ รหัสนักศึกษา B5309207

นายวรวุฒิ บุญเป็ง รหัสนักศึกษา B5310333

มหาวิทยาลัยเทคโนโลยีสุรนารี

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 438499 โครงงานวิศวกรรมอิเล็กทรอนิกส์และ 438494 โครงงานศึกษาวิศวกรรมอิเล็กทรอนิกส์ หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคมหลักสูตรวิศวกรรมอิเล็กทรอนิกส์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารีประจำภาคการศึกษาที่ 3 ปีการศึกษา 2556

โครงงาน	เครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลัง
จัดทำโดย	1. นายธีระภัทร เจริญปรุ รหัส B5309207
	2. นายวรวุฒิ บุญเป็ง รหัส B5310333
อาจารย์ที่ปรึกษา	ผศ.ร.อ. ดร.ประโยชน์ คำสวัสดิ์
สาขาวิชา	วิศวกรรมโทรคมนาคม หลักสูตรวิศวกรรมอิเล็กทรอนิกส์
ภาคการศึกษาที่	3/2556

บทคัดย่อ
(Abstract)

โครงงานนี้นำเสนอเครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลังซึ่งมีวัตถุประสงค์ที่จะประยุกต์ใช้ไมโครคอนโทรลเลอร์ตระกูล ARM Cortex-M4F core รุ่น STM32F4Discovery ในการออกแบบระบบซึ่งประกอบด้วยสองส่วนคือ ส่วนของโหนดย่อย (Router) และส่วนของภาคประมวลผลกลาง (Coordinator) ซึ่งโหนดย่อยจะมีหน้าที่ในการวัดค่าจากเซ็นเซอร์วัดความชื้นในดิน เซ็นเซอร์วัดความเข้มแสง และเซ็นเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ในอากาศแล้วทำการส่งข้อมูลแบบไร้สายโดยใช้โมดูล Xbee ไปให้ภาคประมวลผลกลางเพื่อนำข้อมูลไปประมวลผลในการตัดสินใจควบคุมการเปิด-ปิดน้ำที่จ่ายให้กับไร่มันสำปะหลังให้เหมาะสมกับความต้องการน้ำของมันสำปะหลัง และอัพโหลดข้อมูลที่รับมาจากโหนดย่อยขึ้นเซิร์ฟเวอร์โดยใช้โมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ จากผลการทดลองพบว่าโหนดย่อยซึ่งมีหน้าที่วัดค่าต่างๆสามารถวัดค่าจากเซ็นเซอร์วัดความชื้นในดิน เซ็นเซอร์วัดความเข้มแสง และเซ็นเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ในอากาศและส่งต่อให้ภาคประมวลผลกลางได้ และภาคประมวลผลกลางสามารถนำข้อมูลที่ได้รับจากโหนดย่อยไปประมวลผลในการตัดสินใจควบคุมการเปิด-ปิดน้ำที่จ่ายให้กับไร่มันสำปะหลัง และสามารถบันทึกข้อมูลลงหน่วยความจำ SD-Card และส่งข้อมูลขึ้นเซิร์ฟเวอร์ได้ จึงสรุปได้ว่าโครงงานเครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลังนี้เป็นไปตามวัตถุประสงค์ทุกประการ

กิตติกรรมประกาศ

จากการที่คณะจัดทำรายงานได้รับมอบหมายให้ทำโครงงานเรื่อง เครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลัง ส่งผลให้คณะจัดทำรายงานได้รับความรู้และประสบการณ์ต่างๆเกี่ยวกับการเขียนโปรแกรมควบคุมบอร์ดไมโครคอนโทรลเลอร์รุ่น STM32F4 Discovery , การติดต่อแบบไร้สายผ่าน Xbee, การควบคุมอุปกรณ์เปิด-ปิดน้ำด้วยไมโครคอนโทรลเลอร์การส่งข้อมูลขึ้นเซิร์ฟเวอร์ด้วยโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์การออกแบบวงจรอิเล็กทรอนิกส์ และการใช้ไมโครคอนโทรลเลอร์ในการบันทึกข้อมูลลง SD-Cardบัดนี้โครงงานดังกล่าวพร้อมทั้งรายงานได้สำเร็จลงแล้ว ทั้งนี้ด้วยความร่วมมือและสนับสนุนจากบุคคลต่างๆ ดังนี้

1. ผู้ช่วยศาสตราจารย์ ร.อ. ดร.ประโยชน์ คำสวัสดิ์ (อาจารย์ที่ปรึกษาโครงงาน)
2. นางสาวนภาพร พิมปรุ วิศวกร รมศาสตรบัณฑิต
(วิศวกรรม โทรคมนาคม)

ข้าพเจ้าใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่าน โดยเฉพาะคณาจารย์และบุคลากรสาขาวิชาเทคโนโลยีการเกษตร ที่มีส่วนร่วมในการให้ข้อมูลและเป็นที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์ และอาจารย์ที่ปรึกษาโครงงานที่ ให้การดูแลและให้ความเข้าใจเกี่ยวกับพื้นฐานการใช้งานโปรแกรม และอุปกรณ์ต่างๆที่ใช้ในการทำโครงงานซึ่งข้าพเจ้าขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

นา ยธีระภัทร เจริญปรุ
นา ยวรวุฒิ บุญเป็ง

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูป	จ
สารบัญตาราง	ฎ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตในการดำเนินงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่2 ทฤษฎีและหลักการที่เกี่ยวข้อง	4
2.1 บทนำ	4
2.2 วิธีการให้น้ำกับมันสำปะหลัง	4
2.3 Simulink และ Real-time workshop	7
2.4 STM32F4Discovery	10
2.5 aMG F4Connect	12
2.6 aMG Sense – Humidity / Temperature	13
2.7 aMG High precision real time clock	14
2.8 Zigbee (X-Bee)	16
2.9 Et-GSM Sim300CZ	20
2.10 Et-opto relay4	24
2.11 เซ็นเซอร์วัดความเข้มแสง	27

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ (ต่อ)

เรื่อง	หน้า
2.12 เซ็นเซอร์วัดความชื้นในดิน	29
2.13 วาล์วไฟฟ้ารุ่น Hunter PGV 203 ขนาด 2 นิ้ว	29
บทที่ 3 การออกแบบระบบ	31
3.1 บทนำ	**31**
3.2 การออกแบบระบบเบื้องต้น	31
3.3 การออกแบบฮาร์ดแวร์	34
3.4 การออกแบบทางซอฟต์แวร์	39
บทที่ 4 ผลการทดลองและทดสอบการใช้งานจริง	108
4.1 การทดสอบฮาร์ดแวร์และซอฟต์แวร์	108
4.2 การติดตั้งระบบสำหรับการใช้งานจริง	118
4.3 ผลการให้น้ำ	121
บทที่ 5 บทสรุปและข้อเสนอแนะ	124
5.1 บทสรุป	124
5.2 ปัญหาและอุปสรรค	124
5.3 แนวทางในการแก้ไขปัญหา	124
5.4 ข้อเสนอแนะเพิ่มเติม	125
5.5 สิ่งที่ได้รับจากการทำโครงงาน	125
บรรณานุกรม	126
ภาคผนวก	127
ภาคผนวก ก.	128
ภาคผนวก ข.	133
ประวัติผู้เขียน	137

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รูปที่ 3.3 (ก) (ข) ภาคประมวลผลและควบคุมการเปิด-ปิดน้ำ 33

รูปที่ 3.4 ระบบน้ำหยด 33

รูปที่ 3.5 วงจรชาร์ตแบตเตอรี่และแปลงแรงดัน 34

รูปที่ 3.6วงจรแปลงแรงดันกระแสตรง 12 โวลต์เป็น กระแสตรง 24 โวลต์ 35

รูปที่ 3.7 วงจรแปลงแรงดันกระแสตรง 24 โวลต์เป็นกระแสสลับ 24 โวลต์ 36

รูปที่ 3.8 แสดงการทำงานของโปรแกรมโคออร์ดิเนเตอร์ 39

รูปที่ 3.9รูปของบล็อกทั้งหมดแบ่งเป็นสามส่วน 40

รูปที่ 3.10 Target setup 40

รูปที่ 3.11 (ก)Character LCD setup(ข)UART setup(ค) UART setup1 และ (ง) I2C master setup 42

รูปที่ 3.12 การกำหนดค่า Character LCD setup 43

รูปที่ 3.13 การกำหนดค่า UART setup 44

รูปที่ 3.14 การกำหนดค่า UART setup1 45

รูปที่ 3.15 การกำหนดค่า I2C master setup 46

รูปที่ 3.16 รูปแสดงส่วนของการประกาศตัวแปรซึ่งถูกแบ่งเป็น 6 ส่วนย่อย 46

รูปที่ 3.17บล็อกสำหรับเก็บค่าของเวลาทั้งหมด 47

รูปที่ 3.18 บล็อกสำหรับการแสดงผลที่จอ LCD ทั้งหมด 47

รูปที่ 3.19 บล็อกสำหรับเป็นบัฟเฟอร์เก็บข้อมูลส่งขึ้นเซิร์ฟเวอร์ 48

รูปที่ 3.20 บล็อกสำหรับเป็นบัฟเฟอร์เก็บข้อมูลจากโหนดย่อย 50

รูปที่ 3.21 บล็อกสำหรับเป็นบัฟเฟอร์ในการหาค่าเฉลี่ยของเซ็นเซอร์ในดิน 12 ตัว 52

รูปที่ 3.22 รูปแสดงส่วนของการทำงานต่างๆซึ่งถูกแบ่งเป็น 5 ส่วนย่อย 53

รูปที่ 3.23บล็อกสำหรับรับข้อมูลจากโหนดย่อย 54

รูปที่ 3.24(ก) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 1 (ข) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 2 55

รูปที่ 3.25(ก) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 3 (ข) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 4 56

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รูปที่ 3.26(ก) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 5 (ข) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 6 57

รูปที่ 3.27บล็อกสำหรับอ่านค่าเวลาชั่วโมงและนาที 57

รูปที่ 3.28บล็อกสำหรับอ่านค่าวันเดือนปี 58

รูปที่ 3.29 บล็อก Atomic subsystem 58

รูปที่ 3.30 บล็อกสำหรับจัดการกับบัฟเฟอร์แสดงผลที่จอ LCD 59

รูปที่ 3.31 บล็อกสำหรับการแสดงผลที่จอ LCD 60

รูปที่ 3.32 บล็อกสำหรับควบคุมลำดับในการหาค่าเฉลี่ยและการเก็บข้อมูลลงบัฟเฟอร์ 62

รูปที่ 3.33 บล็อกที่จะสั่งให้บล็อก Enabled subsystem หมายเลข 2 ของรูปที่ 3.32 เริ่มทำงาน 62

รูปที่ 3.34 บล็อก Atomic subsystem 63

รูปที่ 3.35 บล็อกสำหรับจัดการค่าของเซ็นเซอร์ที่ผิดเพี้ยน 63

รูปที่ 3.36 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 1 ของรูปที่ 3.35ลงบัฟเฟอร์ 64

รูปที่ 3.37 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 2 ของรูปที่ 3.35ลงบัฟเฟอร์ 64

รูปที่ 3.38 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 3 ของรูปที่ 3.35ลงบัฟเฟอร์ 64

รูปที่ 3.39 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 4 ของรูปที่ 3.35ลงบัฟเฟอร์ 65

รูปที่ 3.40 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 5 ของรูปที่ 3.35ลงบัฟเฟอร์ 65

รูปที่ 3.41 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 6 ของรูปที่ 3.35ลงบัฟเฟอร์ 65

รูปที่ 3.42 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 7 ของรูปที่ 3.35ลงบัฟเฟอร์ 66

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ **หน้า**

รูปที่ 3.43 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 8 ของรูปที่ 3.35ลงบัฟเฟอร์ 66

รูปที่ 3.44 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 9 ของรูปที่ 3.35ลงบัฟเฟอร์ 66

รูปที่ 3.45 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 10 ของรูปที่ 3.35ลงบัฟเฟอร์ 67

รูปที่ 3.46 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 11 ของรูปที่ 3.35ลงบัฟเฟอร์ 67

รูปที่ 3.47 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 12 ของรูปที่ 3.35ลงบัฟเฟอร์ 67

รูปที่ 3.48 บล็อกสำหรับหาผลรวมของค่าความชื้นในดินและจำนวนเซ็นเซอร์ทั้งหมดที่ใช้ได้ 68

รูปที่ 3.49 บล็อกสำหรับหาผลรวมของค่าความชื้นในดินและจำนวนเซ็นเซอร์ทั้งหมดที่ใช้ได้ (ต่อ) 69

รูปที่ 3.50 บล็อกที่จะสั่งให้บล็อก Enabled subsystem หมายเลข 3 ของรูปที่ 3.32 เริ่มทำงาน 69

รูปที่ 3.51 บล็อกสำหรับควบคุมการเปิดปิดของปั้มสูบน้ำ 70

รูปที่ 3.52 บล็อกสำหรับสั่งให้ปั้มสูบน้ำหยุดทำงาน 71

รูปที่ 3.53 บล็อกสำหรับสั่งให้ปั้มสูบน้ำทำงาน 71

รูปที่ 3.54 บล็อกสำหรับเรียงลำดับการเก็บค่าลงบัฟเฟอร์และเขียนลงเอสดีการ์ด 72

รูปที่ 3.55 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 2 ของรูปที่ 3.54 73

รูปที่ 3.56 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 4 ของรูปที่ 3.54 73

รูปที่ 3.57 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 6 ของรูปที่ 3.54 74

รูปที่ 3.58 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 8 ของรูปที่ 3.54 74

รูปที่ 3.59 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 10 ของรูปที่ 3.54 75

รูปที่ 3.60 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 12 ของรูปที่ 3.54 75

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ **หน้า**

รูปที่ 3.61 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 14 ของรูปที่ 3.54 76
รูปที่ 3.62 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 16 ของรูปที่ 3.54 76
รูปที่ 3.63 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 18 ของรูปที่ 3.54 76
รูปที่ 3.64 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 20 ของรูปที่ 3.54 77
รูปที่ 3.65 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 20 ของรูปที่ 3.54 78
รูปที่ 3.66 การกำหนดค่าของบล็อก String buffer processing 1 ของรูปที่ 3.65 78
รูปที่ 3.67 การกำหนดค่าของบล็อก UART Txของรูปที่ 3.65 79
รูปที่ 3.68 การกำหนดค่าของบล็อก String buffer processing ของรูปที่ 3.65 80
รูปที่ 3.69 บล็อกสำหรับรับค่าจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ 81
รูปที่ 3.70 การกำหนดค่าสำหรับบล็อก UART Rx ของรูปที่ 3.69 81
รูปที่ 3.71 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ 82
รูปที่ 3.72 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 82
รูปที่ 3.73 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 82
รูปที่ 3.74 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 83
รูปที่ 3.75 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 83
รูปที่ 3.76 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 83
รูปที่ 3.77 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 84
รูปที่ 3.78 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 84
รูปที่ 3.79 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 85
รูปที่ 3.80 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 85
รูปที่ 3.81 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 85
รูปที่ 3.82 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 86
รูปที่ 3.83 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 86
รูปที่ 3.84 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 86
รูปที่ 3.85 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 87

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รูปที่ 3.86 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ) 87

รูปที่ 3.87 บล็อกสำหรับส่งค่าออกไปติดต่อกับโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ 88

รูปที่ 3.88 บล็อกสำหรับส่งข้อมูลขึ้นเซิร์ฟเวอร์ 90

รูปที่ 3.89 (ก) (ข) และ (ค) String buffer processing 91

รูปที่ 3.90การกำหนดค่า String buffer processing 92

รูปที่ 3.91 การกำหนดค่า String buffer processing 93

รูปที่ 3.92 การกำหนดค่า String buffer processing 94

รูปที่ 3.93 (ก) UART Tx(ข) (ค) และ (ง) String buffer Processing 94

รูปที่ 3.94 การกำหนดค่า UART Tx 95

รูปที่ 3.95 การกำหนดค่า String buffer processing 96

รูปที่ 3.96 การกำหนดค่า String buffer processing1 97

รูปที่ 3.97 การกำหนดค่า String buffer processing2 98

รูปที่ 3.98 แสดงการทำงานของโปรแกรมโหนดย่อย 99

รูปที่ 3.99 Target Setup 99

รูปที่ 3.100 บล็อก UART setup และ I2C master Setup 100

รูปที่ 3.101 การกำหนดค่า UART setup 100

รูปที่ 3.102 การกำหนดค่า I2C master setup 101

รูปที่ 3.103 บล็อกVolatile data storage 101

รูปที่ 3.104 บล็อกสำหรับอ่านค่าอุณหภูมิและความชื้นในอากาศ 102

รูปที่ 3.105 บล็อกสำหรับอ่านค่าความชื้นในดิน 103

รูปที่ 3.106 บล็อกสำหรับอ่านค่าความเข้มแสง 104

รูปที่ 3.107 บล็อกสำหรับอ่านค่าแรงดันแบตเตอรี่ 105

รูปที่ 3.108 บล็อกสำหรับส่งข้อมูลให้โคออร์ดิเนเตอร์ 106

รูปที่ 4.1 สถานที่ใช้ทดสอบหาระยะการส่งข้อมูลของ Xbee 110

รูปที่ 4.2 ข้อมูลที่อยู่ในไฟล์ data.txt ในหน่วยความจำ SD-Card 112

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

รูปที่ 4.3 ข้อมูลบนเซิร์ฟเวอร์ที่ URL: http://203.158.7.36/142001/CASAVA1/open.php 113

รูปที่ 4.4 (ก)เซ็นเซอร์วัดความชื้นในดิน(ข) รีเลย์ที่ใช้ต่อเข้ากับเครื่องสูบน้ำ 116

รูปที่ 4.5 (ก)ปั้มน้ำ (ข) โซลินอยด์วาล์วและวงจรสำหรับขับโซลินอยด์วาล์ว 116

รูปที่ 4.6แผนภาพจุดติดตั้งระบบ 118

รูปที่ 4.7 (ก) สถานที่ติดตั้งจริงบริเวณฟาร์มมหาวิทยาลัยเทคโนโลยีสุรนารี(ข) บ่อน้ำ 119

รูปที่ 4.8 (ก)ชุดวงจรโคออร์ดิเนเตอร์(ข) จอแสดงผล 119

รูปที่ 4.9การติดตั้งโหนดย่อยเพื่อวัดค่าจากเซ็นเซอร์วัดความชื้นในดิน เซ็นเซอร์วัดความเข้มแสง และเซ็นเซอร์วัดอุณหภูมิและความชื้นสมพัทธ์ในอากาศ 120

รูปที่ 4.10 (ก)การติดตั้งเซ็นเซอร์แสง(ข) การติดตั้งเซ็นเซอร์วัดความชื้นในดิน 121

รูปที่ 4.11กราฟเปอร์เซ็นความชื้นในดินของวันที่ 22 เดือน มีนาคม ค.ศ. 2014 ในช่วงเวลาตั้งแต่ 0.01 น. ถึง 23.59 น. 121

รูปที่ 4.12ค่าพิกัดทางไฟฟ้าและทางกลของปั้มน้ำ 122

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 2.1 แสดงความสามารถในการอุ้มน้ำของเนื้อดินแต่ละชนิด 6

ตารางที่ 2.1(ต่อ)แสดงความสามารถในการอุ้มน้ำของเนื้อดินแต่ละชนิด 7

ตารางที่ 2.2 จุดเชื่อมต่อของบอร์ด Et-opto relay4 กับไมโครคอนโทรลเลอร์ 26

ตารางที่ 3.1 ข้อมูลเบื้องต้นของไอซีเบอร์ 7815 34

ตารางที่ 3.2 ข้อมูลเบื้องต้นของไอซีเบอร์ 7805 35

ตารางที่ 3.3 ข้อมูลเบื้องต้นของไอซีเบอร์ NE555N 37

ตารางที่ 3.4 ข้อมูลเบื้องต้นของไอซีเบอร์ LM324N 38

ตารางที่ 3.5ทำการเชื่อมต่อขาสัญญาณของจอ LCD เข้ากับขาสัญญาณของ STM32F4Discovery 41

ตารางที่ 3.6 ทำการเชื่อมต่อขาสัญญาณของโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์เข้ากับขาสัญญาณของ STM32F4Discovery 41

ตารางที่ 3.7 ทำการเชื่อมต่อขาสัญญาณของ Xbeeเข้ากับขาสัญญาณของ STM32F4Discovery 42

ตารางที่ 3.8 ทำการเชื่อมต่อขาสัญญาณของโมดูล aMG high precision real time clock เข้ากับขาสัญญาณของ STM32F4Discovery 42

ตารางที่ 3.9ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์เก็บค่าของเวลา 47

ตารางที่ 3.10ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์การแสดงผลที่จอ LCD\ 48

ตารางที่ 3.11ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์เก็บข้อมูลส่งขึ้นเซิร์ฟเวอร์ 49

ตารางที่ 3.12ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์รับข้อมูลจากโหนดย่อย 50

ตารางที่ 3.12ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์รับข้อมูลจากโหนดย่อย(ต่อ) 51

ตารางที่ 3.11ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์ในการหาค่าเฉลี่ยของเซ็นเซอร์ในดิน 12 ตัว 52

ตารางที่ 3.14 การกำหนดค่าพารามิเตอร์สำหรับบล็อก UART Txที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.71 ถึงรูปที่ 3.8 88

ตารางที่ 3.14 การกำหนดค่าพารามิเตอร์สำหรับบล็อก UART Txที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.71 ถึงรูปที่ 3.83 (ต่อ) 89

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

ตารางที่ 3.15 การกำหนดค่าพารามิเตอร์สำหรับบล็อก String buffer processing (SDIS) ที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.71 ถึงรูปที่ 3.83 89

ตารางที่ 3.16 การกำหนดค่าพารามิเตอร์สำหรับบล็อก String buffer processing ที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.84 90

ตารางที่ 3.17 การกำหนดค่าพารามิเตอร์สำหรับบล็อก UART Txที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.84 (เลือก UART Module เป็น 1 เลือก Transfer เป็น Blocking เลือก Packet mode เป็น String buffer และเลือก End of packet เป็น None) 91

ตารางที่ 3.18กำหนดค่าพารามิเตอร์สำหรับโหนด(ให้เลือก UART Module เป็น 1 เลือก Transfer เป็น Blocking เลือก Packet mode เป็น Binary เลือกจำนวนของเอาต์พุตเป็น Single จำนวน 7 พอร์ต) 106

ตารางที่ 4.1 ผลการทดลองส่งข้อมูลด้วยXbee 109

ตารางที่ 4.2 ความหมายของค่าต่างๆในรูปที่ 4.2 และรูปที่ 4.3 114

ตารางที่ 4.3 การทำงานของปั้มน้ำในสภาวะต่างๆ 117

ตารางที่ 4.4 การทำงานของโซลินอยด์วาล์วสภาวะต่างๆ 117

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

มันสำปะหลังเป็นพืชที่สามารถปลูกได้โดยไม่ต้องให้น้ำเนื่องจากเป็นพืชไร่ที่ทนต่อสภาวะแล้งได้ดีแต่การให้น้ำแก่มันสำปะหลังจะทำให้ได้ผลผลิตที่สูงขึ้นและทำให้ต้นมันสำปะหลังแข็งแรงและทนทานต่อศัตรูพืชมากขึ้น แต่ปริมาณการให้น้ำเป็นเรื่องสำคัญเนื่องจากถ้าให้น้ำแก่มันสำปะหลังในปริมาณที่มากเกินไปจนเกินความต้องการจะทำให้สารอาหารที่จำเป็นต่อการเจริญเติบโตของมันสำปะหลังซึ่งอยู่ในดิน ถูกน้ำชะล้างลงลึกใต้ดินตามแรงโน้มถ่วงของโลก รวมถึงอาจเกิดปัญหารากมันสำปะหลังเน่าเสียได้ อีกทั้ง ในปัจจุบันเทคโนโลยีเข้ามามีบทบาทในชีวิตประจำวันของมนุษย์มากขึ้นในทุกๆด้าน ไม่ว่าจะเป็นเทคโนโลยีด้านความบันเทิง การสื่อติดต่อสื่อสาร ธุรกิจ การแพทย์ อุตสาหกรรม การขนส่ง ไม่เว้นแม้แต่เกษตรกรรม ซึ่งจะช่วยทำให้ได้ผลผลิตในปริมาณมากขึ้น คุณภาพสูงขึ้น ประหยัดต้นทุนแรงงานประหยัดเวลา เป็นต้น หนึ่งในเทคโนโลยีที่เป็นที่นิยมคือ ไมโครคอนโทรลเลอร์ เพราะราคาถูกและสามารถพัฒนาโปรแกรมได้ง่าย ไมโครคอนโทรลเลอร์สามารถใช้ในการควบคุมการทำงานของอุปกรณ์อิเล็กทรอนิกส์ต่างๆ จึงมีแนวคิดสร้างระบบชลประทานสำหรับไร่มันสำปะหลังโดยใช้ไมโครคอนโทรลเลอร์ในการควบคุมการ ให้น้ำในไร่มันสำปะหลังโดยวิเคราะห์ความต้องการน้ำจากความชื้นในดิน และความสามารถในการดูดซึมน้ำในดินของมันสำปะหลังเนื่องจากมันสำปะหลังไม่สามารถดูดน้ำจากดินไปใช้ได้ทั้งหมด การให้น้ำให้พอเหมาะกับความต้องการของมันสำปะหลังจะทำให้ไม่สูญเสียน้ำโดยเปล่าประโยชน์ ทั้งยังบันทึกค่าอุณหภูมิและความชื้นใน ดินและความชื้นใน อากาศลงในหน่วยความจำ SD-Card และอัพโหลดค่าบนเซิ รัฟเวอร์ผ่านโมดูล สื่อสารผ่านเครือข่ายโทรศัพท์ นอกจากนั้นยังใช้แสงอาทิตย์เป็นแหล่งพลังงานซึ่งเป็นพลังงานสะอาด และมีวงจรชาร์ตจากโซล่าเซลล์เข้าแบตเตอร์รี่เพื่อใช้งานในเวลาที่ไม่มีแสงอาทิตย์

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการส่งข้อมูลไร้สายโดยใช้ Xbeeและใช้ Xbeeในการรับ-ส่งข้อมูลได้
2. เพื่อศึกษาการออกแบบวงจรอิเล็กทรอนิกส์และสามารถสร้างเพื่อใช้งานได้
3. เพื่อศึกษาการใช้ไมโครคอนโทรลเลอร์บันทึกข้อมูลลง SD-card และควบคุมการเปิดและปิดของอุปกรณ์เปิด-ปิดน้ำ

4. เพื่อศึกษาการอัพโหลดข้อมูลขึ้นเซิร์ฟเวอร์ด้วยโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์และใช้งานได้

5. เพื่อนำสิ่งประดิษฐ์ไปใช้งานจริงในไร่มันของฟาร์มมหาวิทยาลัยเทคโนโลยีสุนารีได้

1.3 ขอบเขตในการดำเนินงาน

1. ศึกษาการทำงานของอุปกรณ์ เซ็นเซอร์วัดความชื้น เซ็นเซอร์วัดอุณหภูมิ
2. ศึกษาการโปรแกรมไมโครคอนโทรลเลอร์ STM32F4 Discovery
3. ศึกษาการใช้งาน Xbee
4. ศึกษาการใช้งานโมดูลอินเทอร์เน็ต
5. ศึกษาการควบคุมอุปกรณ์เปิดปิดน้ำด้วยไมโครคอนโทรลเลอร์
6. ควบคุมการทำงานของอุปกรณ์เปิดปิดน้ำได้ตามต้องการ

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาค้นคว้าการทำงานของอุปกรณ์ เซ็นเซอร์วัดความชื้นในดิน เซ็นเซอร์วัดอุณหภูมิ และ เซ็นเซอร์แสง
2. ศึกษาค้นคว้าการโปรแกรมไมโครคอนโทรลเลอร์ STM32F4 Discovery
3. ศึกษาค้นคว้าการใช้งาน Xbee
4. เขียนโครงการและ เสนอโครงการกับอาจารย์ที่ปรึกษา
4. เขียนโปรแกรมให้ STM32F4 Discovery อ่านค่าจาก เซ็นเซอร์ทุกชนิด
5. เขียนโปรแกรมส่งข้อมูลไร้สายด้วย Xbee ไปยัง STM32F4 Discovery
6. เขียนโปรแกรมประมวลผลความต้องการน้ำของไร่มันสำปะหลัง
7. ออกแบบวงจรควบคุมการเปิดปิดน้ำ
8. เขียนโปรแกรมควบคุมการเปิดปิดน้ำโดย ไมโครคอนโทรลเลอร์ STM32F4 Discovery
9. เขียนโปรแกรมส่งค่าจากเซ็นเซอร์ขึ้นเซิร์ฟเวอร์ผ่านโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์
10. สร้างอุปกรณ์ต้นแบบทั้งหมดและทดสอบเพื่อให้ได้ตามวัตถุประสงค์

มหาวิทยาลัยเทคโนโลยีสุรนารี

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการทำงานของบอร์ด STM32F4 Discovery

2. เข้าใจหลักการควบคุมการเปิดปิดของอุปกรณ์ไฟฟ้าด้วยไมโครคอนโทรลเลอร์

3. เข้าใจการเขียนโปรแกรมและโหลดโปรแกรมใช้งานได้

4. สามารถใช้งานเครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลังได้จริงในการทำเกษตรกรรม

บทที่2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 บทนำ

เครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลังใช้ไมโครคอนโทรลเลอร์ STM32F4 Discovery ในการประมวลผลและควบคุม ซึ่งจะต้องใช้ Simulink ในการโปรแกรม และต้องใช้การส่งข้อมูลไร้สายด้วย X- bee ส่งข้อมูลขึ้นอินเทอร์เน็ตด้วย โมดูล สื่อสารผ่านเครือข่ายโทรศัพท์ SIM300CZ ในส่วนของการเปิด -ปิดน้ำ ใช้ วาล์วควบคุมด้วยไฟฟ้าหรือเครื่องสูบน้ำ โดยตัดสินใจเปิด-ปิดน้ำ จากความชื้น ในดินนอกจากนั้นยังมีการวัดค่าความเข้มแสง อุณหภูมิ และความชื้นสัมพัทธ์ในอากาศอีกด้วย ในบทที่นี้จะอธิบายหลักการและการใช้งานอุปกรณ์ที่ใช้ในการทำโครงงานดังที่กล่าวมาข้างต้น

2.2 วิธีการให้น้ำกับมันสำปะหลัง

วิธีการให้น้ำพืชมี สี่วิธีด้วยกัน วิธีแรกคือ การให้น้ำทางผิวดิน เป็นการให้น้ำแบบปล่อยให้ท่วมเป็นผืน กับการให้น้ำแบบท่วมร่อง วิธีที่สองคือการให้น้ำแบบสปริงเกอร์ เป็นการให้น้ำด้วยการพ่นน้ำออกจากหัวสปริงเกอร์ขึ้นไปบนอากาศแล้วให้เม็ดน้ำตกลงมาบนแปลงปลูกพืช โดย มีรูปทรงการแผ่กระจายของเม็ดน้ำที่สม่ำเสมอเช่น เดียวกับฝนตก วิธีที่สามคือ การให้น้ำแบบหยด เป็นการให้น้ำแก่พืชที่จุดใดจุดหนึ่งบนผิวดินในเขตบริเวณรากพืช โดยผ่านท่อน้ำหยด และ วิธีสุดท้ายการให้น้ำทางใต้ดิน เป็นการให้น้ำแก่พืช โดยการยกระดับน้ำใต้ดินขึ้นมาให้สูงพอที่น้ำจะไหลซึมขึ้นมาสู่ระดับเขตรากพืชได้แก่การให้น้ำในคูและการให้น้ำไหลเข้าท่อที่ฝังไว้ใต้ดินสำหรับการให้น้ำกับมันสำปะหลัง การให้น้ำแบบแบบน้ำหยด เป็นวิธีการให้น้ำที่เหมาะ โดยสามารถเสริมสร้างความเข็มแข็งให้กับตัวพืช เพื่อให้พืชสามารถพัฒนาตัวเองให้ต้านทานต่อแมลงศัตรูได้ดีขึ้น นอกจากนี้ยังเป็นการสร้างสภาพแวดล้อมที่อยู่อาศัยให้เหมาะสมกับแมลงตัวห้ำและตัวเบียนที่มีอยู่ตามธรรมชาติ แต่ไม่เหมาะสมกับแมลงศัตรูพืชอย่างเพลี้ยแป้ง

การให้น้ำแบบน้ำหยดเป็นการให้น้ำแก่พืชที่จุดใดจุดหนึ่งบนผิวดินในเขตบริเวณรากพืช โดยผ่านท่อน้ำหยด เป็นวิธีที่ไม่สามารถชะล้างและทำลายตัวเพลี้ยแป้งพร้อมถุงไข่ออกจากต้นมันสำปะหลังได้ แต่จะเสริมสร้างความเข็มแข็งให้กับตัวพืช เพื่อให้พืชสามารถพัฒนาตัวเองให้ต้านทานต่อแมลงศัตรูได้ดีขึ้น นอกจากนี้ยังเป็นการสร้างสภาพแวดล้อมที่อยู่อาศัยให้เหมาะสมกับแมลงตัวห้ำและตัวเบียนที่มีอยู่ตามธรรมชาติ การให้น้ำแบบน้ำหยดต้องมีส่วนประกอบ ดังนี้

1. เครื่องสูบน้ำ เพื่อสูบน้ำจากแหล่งน้ำส่งเข้าสู่ระบบน้ำหยด ต้องส่งน้ำให้มีแรงดันอย่างน้อย 0.6 บาร์

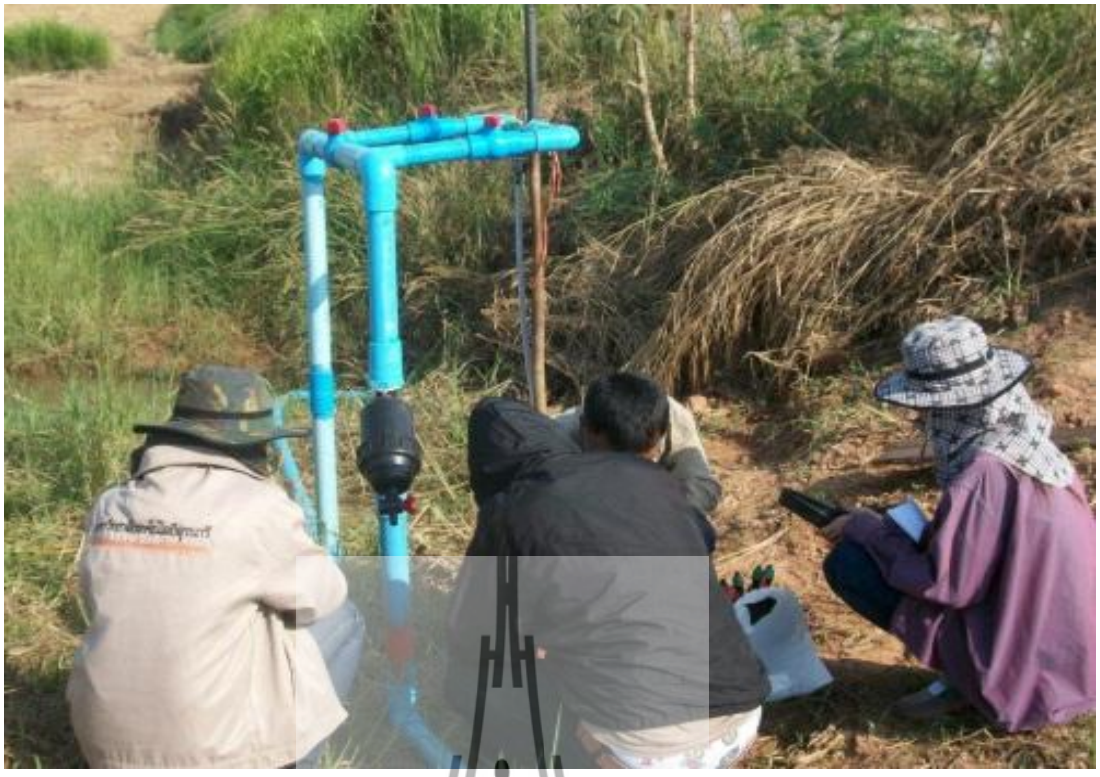
2. ระบบส่งน้ำ ประกอบไปด้วยท่อประธาน เป็นท่อที่ต่อจากแหล่งน้ำ โดยวางไว้บนดินหรือฝังในดินท่อรองประธาน เป็นท่อที่แตกจากท่อประธานอาจใช้ท่อพีวีซีหรือพีอี ขนาด 30-50 มิลลิเมตรท่อน้ำหยด เป็นท่อที่แตกจากท่อรอง วางขนานกับแถวของพืชอาจใช้ท่อพีวีซีหรือพีอี ขนาด 12-20 มิลลิเมตร ยาวไม่เกิน 300 เมตร หัวน้ำหยด เป็นหัวปล่อยน้ำอยู่ติดกับท่อน้ำหยด เป็นตัวควบคุมปริมาณการไหลของน้ำจากท่อน้ำหยดสู่ดิน ขนาดของรู 0.5-1.5 มิลลิเมตร

3. เครื่องกรอง จะทำหน้าที่กรองเอาเศษวัชพืช ใบไม้ ทราย ออกจากน้ำ ถ้าปล่อยให้สิ่งเหล่านี้ผ่านไป จะทำให้หัวน้ำหยดเกิดการอุดตัน เป็นสาเหตุให้ต้นพืชขาดน้ำแล้วชะงักการเจริญเติบโต

4. เครื่องควบคุมการจ่ายน้ำต้นทาง ทำหน้าที่ควบคุมการจ่ายน้ำทั้งระบบจากแหล่งน้ำไปสู่ระบบของการให้น้ำแบบหยด ประกอบด้วยประตูน้ำใหญ่ เครื่องวัดปริมาตรน้ำ เครื่องวัดแรงดันน้ำ เครื่องควบคุมแรงดันประตูป้องกันน้ำไหลกลับ และเครื่องใส่ปุ๋ยหรือสารเคมี

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ 2.1ระบบน้ำหยดในไร่มันสำปะหลัง



รูปที่ 2.2ท่อสูบน้ำจากแหล่งน้ำธรรมชาติ

น้ำเป็นปัจจัยหนึ่งที่สำคัญในการกำหนดผลผลิตของพืช การให้น้ำแก่มันสำปะหลังสามารถเพิ่มผลผลิตให้สูงขึ้นได้ โดนเฉพาะอย่างยิ่งการให้ระบบน้ำหยดซึ่งจากการศึกษาของมหาวิทยาลัยเทคโนโลยีสุรนารีพบว่า การให้น้ำแก่มันสำปะหลังสามารถเพิ่มผลผลิตจาก 4 ต้นต่อไร่เป็น 12 ต้นต่อไร่แต่การให้น้ำนั้นต้องมีการให้ที่ถูกต้องโดยปริมาณน้ำที่ให้และความถี่ของการให้น้ำ คำนวณจากความต้องการน้ำของพืชและความสามารถในการอุ้มน้ำของดิน

ตารางที่ 2.1 แสดงความสามารถในการอุ้มน้ำของเนื้อดินแต่ละชนิด

เนื้อดิน	ความสามารถในการอุ้มน้ำ (มม.น้ำ/ซม. ดิน)		
	รวมทั้งหมด	พืชนำเอาไปใช้ได้	พืชใช้ไม่ได้
1.ดินทราย	1.08	0.6	0.48
2.ดินร่วนปนทราย	1.9	0.95	0.88
3.ดินร่วน	2.85	1.43	1.33
4.ดินร่วนปนตะกอนทราย	3.7	1.85	1.85

ตารางที่ 2.1(ต่อ)แสดงความสามารถในการอุ้มน้ำของเนื้อดินแต่ละชนิด

5.ดินร่วนปนดินเหนียว	3.88	1.65	2.23
6.ดินเหนียว	3.98	1.55	2.43

2.3 Simulink และ Real-time Workshop

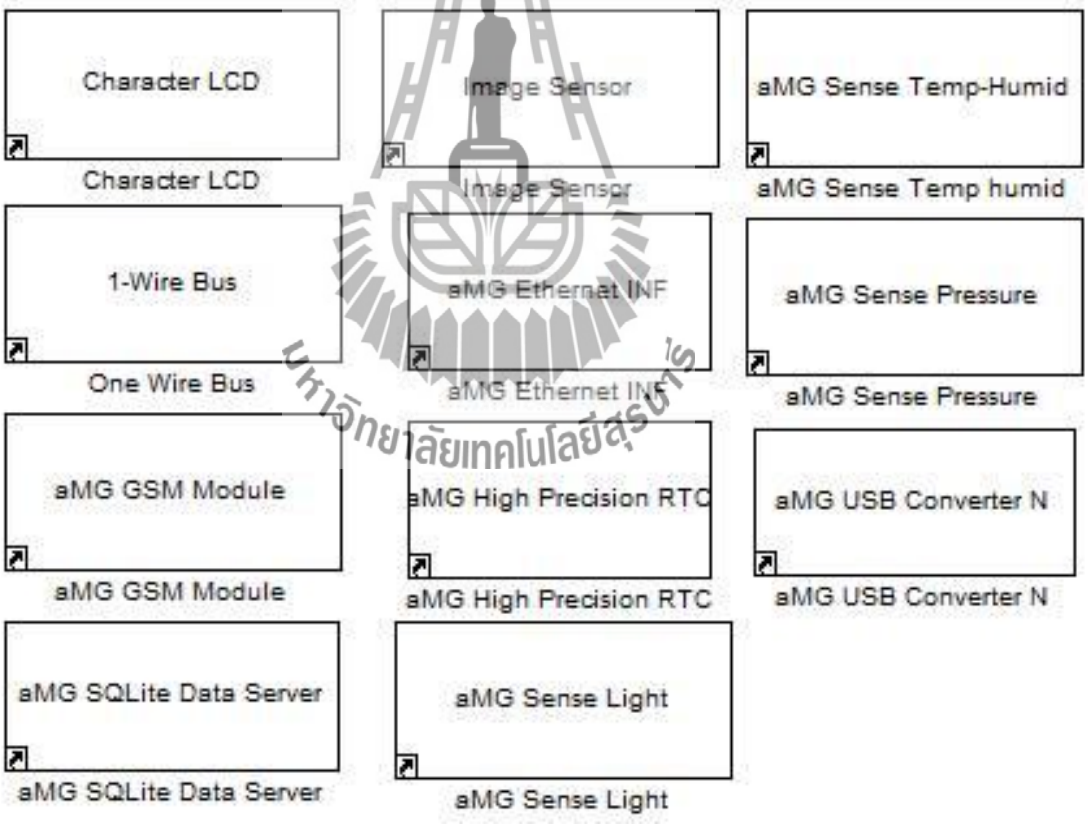
โปรแกรม MATLAB มีเครื่องมือที่ใช้สำหรับการวิเคราะห์และทดสอบระบบโดยการจำลองขึ้นมา ซึ่งก็คือ Simulink เป็นโปรแกรมที่ควบคู่กับ MATLAB ซึ่งเป็นระบบเชิงเส้น (Linear) ระบบไม่เชิงเส้น (Nonlinear) Simulink เป็นโปรแกรม Mouse-Driver ที่ใช้ระบบโมเดลโดยการวาดบล็อกไดอะแกรมบนจอภาพด้วยการใช้เมาส์ทำให้โปรแกรม MATLAB สามารถทำการจำลองระบบได้หลายรูปแบบเช่น แบบที่เป็นเชิงเส้น (Linear) ไม่เชิงเส้น (Nonlinear) เวลาต่อเนื่อง (Continuous- Time) เวลาไม่ต่อเนื่อง (discrete-Time) และระบบหลายอัตรา (Multirate) ซึ่งแต่ละรูปแบบที่นำมาสร้างแบบจำลองในการวิเคราะห์นี้ผู้ใช้จะต้องมีความเข้าใจพื้นฐานการทำงานของแต่ละบล็อกแต่ละบล็อกได้เป็นอย่างดี ตลอดจนเข้าใจระบบโดยรวมของงานที่จะกระทำ

Blocksets เป็นสิ่งที่เพิ่มเติมใน Simulink โดยจะเป็นไลบรารีของบล็อกสำหรับการประยุกต์เฉพาะเช่นการติดต่อสื่อสาร (Communications) การประมวลผลข้อมูล (Signal processing) และระบบไฟฟ้ากำลัง (Power systems)

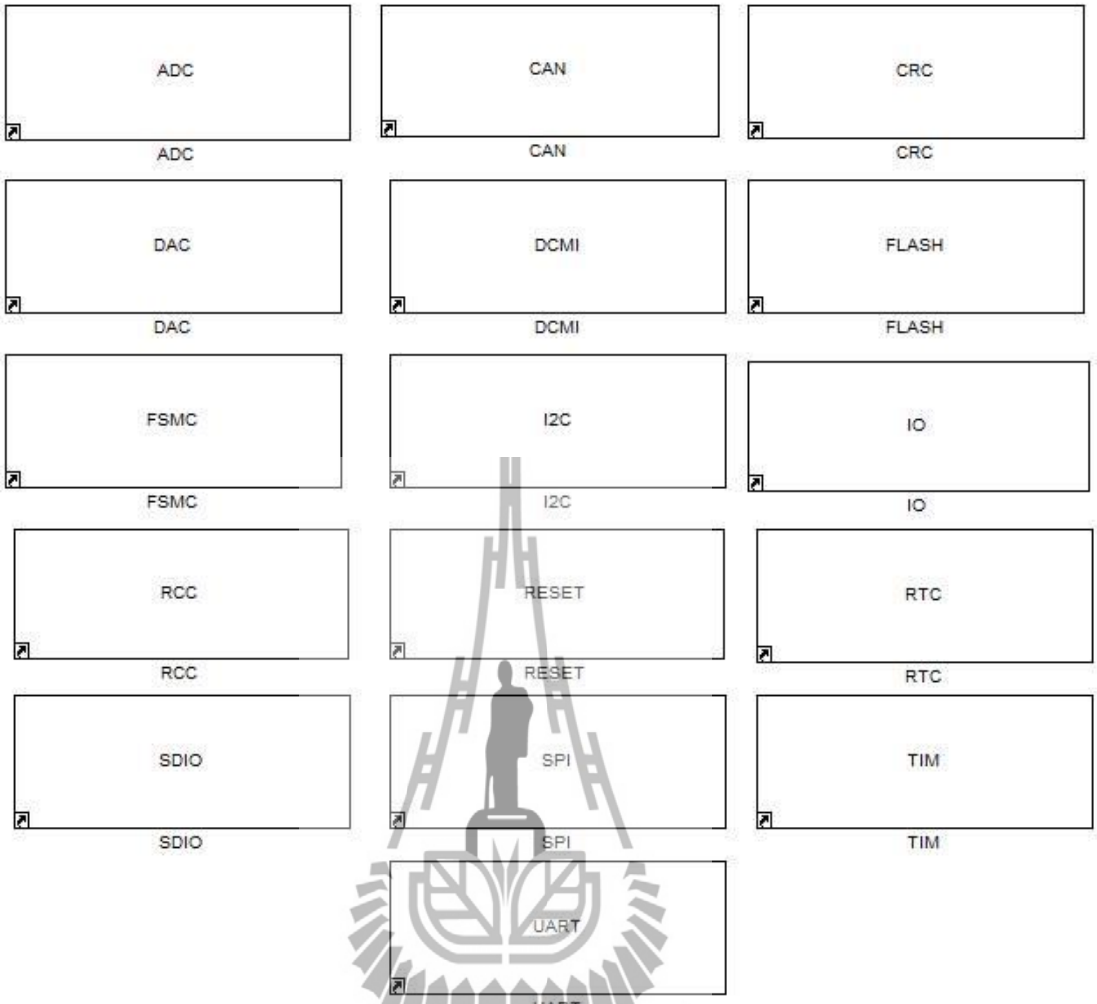
Real-time workshop เป็นโปรแกรมที่สร้าง C Code จากบล็อกไดอะแกรม และสามารถกระทำกับบล็อกไดอะแกรมได้หลากหลายด้วยระบบเวลาจริง (Real-time systems) โปรแกรม MATLAB มีอยู่หลาย Version ซึ่ง Version ดั้งเดิมของโปรแกรม MATLAB จะใช้งานบน DOS ที่มีการคำนวณไม่ยุ่งยากเหมาะสำหรับผู้เริ่มศึกษา คอมพิวเตอร์ที่ใช้ก็ไม่จำเป็นต้องมีพื้นที่หน่วยความจำมาก ใช้ได้กับ CPU ที่มีความเร็วต่ำแต่มีข้อเสียคือฟังก์ชั่น ที่นำมาใช้งานมีน้อยทำให้เขียนโปรแกรมที่มีความซับซ้อนได้ไม่ดี เท่าที่ควรเพราะมีประสิทธิภาพและความเร็วในการประมวลผลต่ำ ต่อมาเมื่อระบบเลือกใช้ได้มากมายจึงทำให้โปรแกรม MATLAB มีประสิทธิภาพและมีความสามารถในการประมวลผลที่เร็วขึ้น Version ใหม่ที่ได้ทำการปรับปรุง ใหม่ให้ดีขึ้นนี้จะใช้งานบน Windows ทำให้ผู้ใช้มีความสะดวกในการใช้งานมากขึ้น ข้อดีของ version ใหม่นี้คือ มีประสิทธิผลที่ดีขึ้น การประมวลผลโปรแกรมที่ซับซ้อนมีความเร็วสูงขึ้น และมีฟังชั่นต่างๆ ใช้เลือกใช้ในสาขาต่างๆ มากมายแต่ก็ต้องใช้กับคอมพิวเตอร์ที่มีหน่วยความจำมาก CPU มีความเร็วสูงและต้องการ Co-Processor ในการช่วยคำนวณแต่เมื่อเปรียบเทียบกับผลดีที่ได้ก็ถือว่าคุ้มใน Simulink จะมี Blocksetsหลายรูปแบบ ซึ่งจะแบ่งได้เป็นดังนี้ คือ Device configuration, Hardware modules, On-chip peripheralsดังแสดงในรูปที่ 2.3 รูปที่ 2.4 และ รูปที่ 2.5

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ 2.3 Device configuration



รูปที่ 2.4 Hardware modules

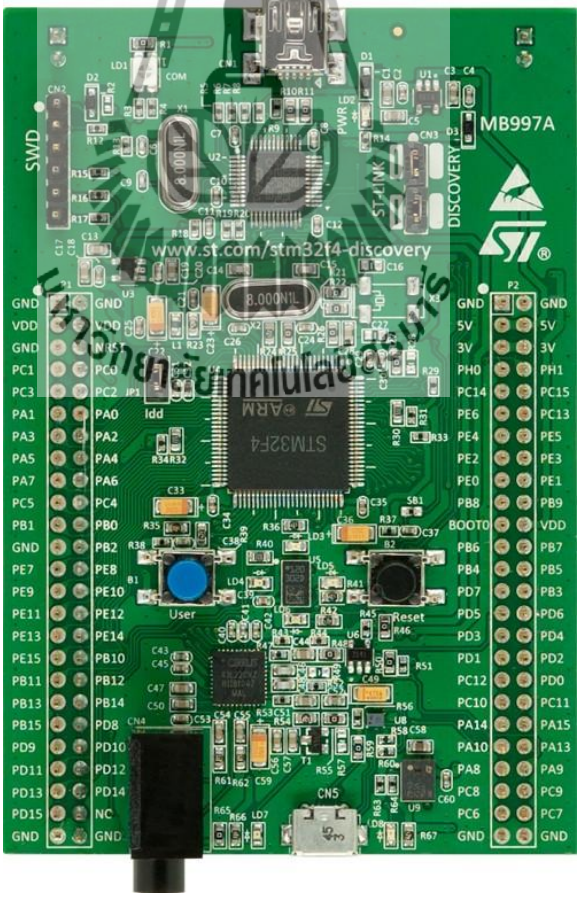


มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ 2.5On-chip peripherals

2.4 STM32F4Discovery

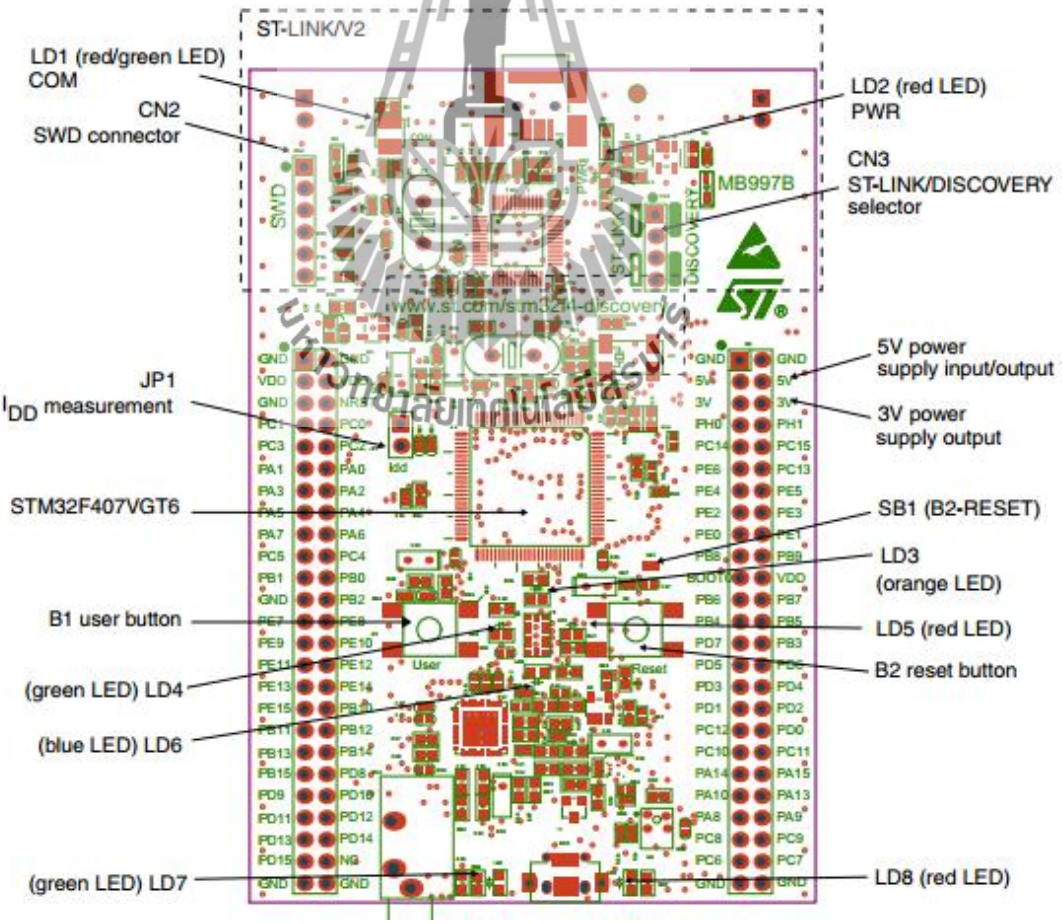
STM32F4 Discovery เป็นชุดทดลองของบริษัท ST โดยใช้ไมโครคอนโทรลเลอร์ตระกูล STM32F407VGT6 32-bit ARM Cortex-M4F โดยในบอร์ดจะประกอบด้วย 2 ส่วนหลักคือ ชุด ST-LINK/V2 ใช้ในการดาวโหลดและดีบักไปยังไมโครคอนโทรลเลอร์ STM32F407VGT6 ที่อยู่ในบอร์ด ผ่านทางพอร์ต USB และส่วนของ บอร์ด STM32F4DISCOVERY เป็นชุดทดลองที่สามารถทางานร่วมกับ MATLAB Simulink ได้ STM32F4DISCOVERY เป็นบอร์ดทดลองอิเล็กทรอนิคส์และเป็นบอร์ดที่ถูกออกแบบมาเพื่อการเรียนการสอนของนักเรียนระดับมหาวิทยาลัยและมัธยมในรายวิชาด้านวิศวกรรมหลายๆด้านเช่นระบบควบคุม , ระบบอัตโนมัติ , หุ่นยนต์, ประมวลผลสัญญาณ Digital(DSP) ฯลฯจุดเด่นของFiOStd Board คือการใช้งานที่ง่ายโดยเฉพาะด้านการเขียนโปรแกรมเนื่องจากเป็นการเขียนโปรแกรมแบบ Graphic Programming ผ่าน Simulink ซึ่งติดตั้งมาพร้อมกับ MATLAB ซึ่งเป็น Module หนึ่งที่อยู่ในโปรแกรม MATLAB ทำให้สามารถทำความเข้าใจการทำงานของโครงงานทั้งในส่วนการทำงานของอุปกรณ์ต่างๆการติดต่อสื่อสารระหว่างอุปกรณ์อัลกอริทึมการเขียนโปรแกรมและอื่นๆ



รูปที่ 2.6 ชุดทดลอง STM32F4 Discovery

คุณสมบัติของชุดทดลอง STM32F4 Discovery

1. ใช้ไมโครคอนโทรลเลอร์ STM32F407VGT6 32-bit ARM Cortex-M4F core
2. มีหน่วยความจำ flash 1 MB และ RAM 192 KB
3. ใช้แหล่งจ่ายผ่าน USB แรงดัน +5 V หรือแหล่งจ่ายภายนอก +5 V ได้
4. มีแหล่งจ่ายแรงดัน 3 V และ 5 V
5. มีเซ็นเซอร์วัดความเร่ง 3 แกนเบอร์ LIS302DL อยู่บนบอร์ด
6. มีดิจิตอลไมโครโฟนเบอร์ MP45DT02 อยู่บนบอร์ด
7. มีพอร์ต USB OTG FS พร้อมขั้วต่อ MICRO-AB
8. ตัวบอร์ดทำเป็นขั้วต่อแบบ PIN HEADER ใต้ PCB 25x2 จำนวน 2 ชุด
9. มีหลอด LED 8 หลอดแสดงสถานการณ์เชื่อมต่อต่างๆ
10. มี 2 ปุ่มกด ยูเซอร์ และ รีเซ็ต



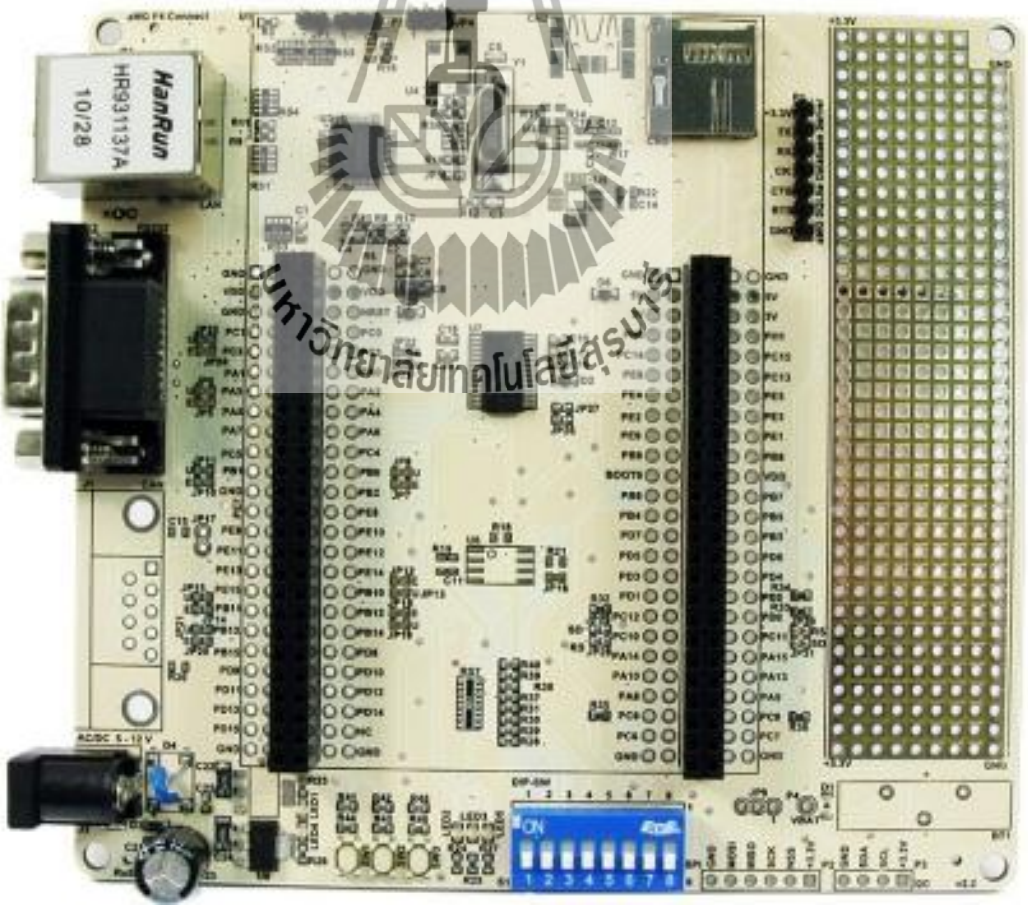
รูปที่ 2.7บอร์ดทดลอง STM32F4 Discovery

ซอร์ฟแวร์ที่ใช้งาน

1. Waijungblocksetหรือเวอร์ชั่นที่ใหม่กว่า
2. Matlab 32-bits 2009a (version 7.8) หรือเวอร์ชั่นที่ใหม่กว่า
3. Simulink 2009 (version 7.3) หรือเวอร์ชั่นที่ใหม่กว่า
4. Real-Time Workshop 2009 (version 7.3) หรือเวอร์ชั่นที่ใหม่กว่า
5. Real-Time Workshop Embedded Coder 2009 (version 5.3) หรือเวอร์ชั่นที่ใหม่กว่า
6. RealView MDK for ARM version 4.0 หรือเวอร์ชั่นที่ใหม่กว่า
7. Microsoft Windows XP SP2 หรือเวอร์ชั่นที่ใหม่กว่า
8. Microsoft .Net Framework version 3.5 หรือเวอร์ชั่นที่ใหม่กว่า

2.5aMG F4Connect

aMG F4Connectเป็นบอร์ดที่ออกแบบมาเพื่อใช้คู่กับบอร์ด STM32F4DISCOVERY



รูปที่ 2.8 บอร์ด aMG F4Connect

มหาวิทยาลัยเทคโนโลยีสุรนารี

โครงสร้างของบอร์ด

1. มีขั้วต่อสำหรับสาย LAN RJ45
2. มีขั้วต่อ RS232 ชนิด DB9 ตัวผู้
3. มีขาสำหรับการติดต่อแบบ SPI และ I2C
4. มีช่องสำหรับเสียบ SDCard
5. มีสวิตซ์สำหรับทดสอบแบบ DIP 8 ตัว
6. มีสวิตซ์สำหรับทดสอบแบบ กดติดปล่อยดับ 3 ตัว
7. มี LED สำหรับทดสอบ 3 ดวง (สีแดง สีเหลือง สีเขียว)
8. มีวงจรรักษาแรงดันขนาด 3.3 โวลต์ 800 mA
9. มีช่องสำหรับเสียบบอร์ด STM32F4DISCOVERY
10. มีช่องขนาด 2.0 mm สำหรับเสียบแจ๊คแหล่งจ่ายกระแสตรงขนาด 5 โวลต์

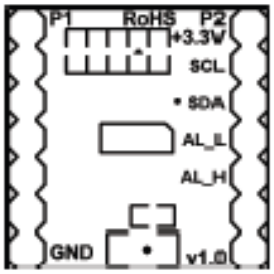
2.6aMG Sense - Humidity / Temperature

aMG Sense - Humidity / Temperature เป็นโมดูลเซนเซอร์วัดอุณหภูมิและความชื้นในอากาศ ซึ่งใช้ IC ของ Honeywell เบอร์ HIH6131

คุณสมบัติของบอร์ด

1. ความแม่นยำในการวัดความชื้น +/- 4% RH
2. สามารถวัดความชื้นได้ดีในช่วง10 ถึง 90 %RH
3. ความแม่นยำในการวัดอุณหภูมิ +/- 1องศาเซียลเซียส
4. สามารถวัดอุณหภูมิได้ในช่วง5 ถึง 50องศาเซียลเซียส
5. ให้เอาต์พุตความละเอียดขนาด 14 บิต
6. ใช้การติดต่อแบบ I2C
7. ทำงานได้ดีในช่วงอุณหภูมิ -25 ถึง 80 องศาเซียลเซียส
8. ใช้แหล่งจ่ายกระแสตรงขนาด 2.5 ถึง 5.5 โวลต์

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 2.9aMG Sense - Humidity / Temperature

โครงสร้างของบอร์ด

+3.3V: เป็นขาสำหรับต่อแหล่งจ่ายกระแสตรงขนาด 2.5 ถึง 5.5 โวลต์

SCL: ขาสัญญาณนาฬิกาสำหรับการรับส่งข้อมูลด้วยระบบบัส I2C

SDA: ขารับส่งข้อมูลด้วยระบบบัส I2C

AL_L: ขาสัญญาณเตือนให้เอาต์พุตเป็น LOW

AL_H: ขาสัญญาณเตือนให้เอาต์พุตเป็น HIGH

GND: ใช้ต่อกราวด์

มหาวิทยาลัยเทคโนโลยีสุรนารี

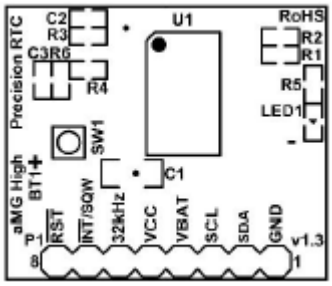
2.7aMG High Precision Real Time Clock

aMG High Precision Real Time Clock เป็นโมดูลระบบฐานเวลา ซึ่งใช้ IC ของ Maxim เบอร์ DS3231SN

คุณสมบัติของบอร์ด

1. ความแม่นยำ±2ppm ในช่วง0°C ถึง +40°C
2. ความแม่นยำ ±3.5ppm ในช่วง -40°C ถึง +85°C
3. มีแบตเตอรี่สำรองเพื่อนให้โมดูลสามารถทำงานต่อได้ในขณะที่ไม่จ่ายไฟ
4. ช่วงของอุณหภูมิที่สามารถทำงานได้ดี -40°C ถึง +85°C
5. ใช้พลังงานต่ำ
6. สามารถนับเวลาเป็น วินาที นาที ชั่วโมง วัน เดือน ปี (นับได้สูงสุดถึงปี 2100)
7. สามารถโปรแกรมตั้งเวลาเตือนได้
8. ใช้การติดต่อแบบ I2C ที่ความถี่ 400kHz
9. สามารถวัดอุณหภูมิและให้เอาต์พุตเป็นดิจิตอล ซึ่งมีความแม่นยำ±3°C

โครงสร้างของบอร์ด



รูปที่ 2.10aMG High Precision Real Time Clock

RST: เป็นขาสำหรับรีเซ็ตโมดูล

INT/SQW: ขาเอาต์พุตสัญญาณ Square Wave และขาสัญญาณอินเตอร์รัพท์(ต้องต่อ R pullup)

32kHz: เป็นขาเอาต์พุตความถี่ 32 kHz (ต้องต่อ R pullup)

VCC: ใช้ต่อกับแหล่งจ่ายกระแสตรงขนาด 2.3 ถึง 5.5 V

GND: ใช้ต่อกราวด์

VBAT: ใช้ต่อกับแบตเตอรี่ 2.3 ถึง 5.5 V เพื่อรักษาการทำงาน ในกรณีที่ไม่มีไฟเลี้ยงจ่าย

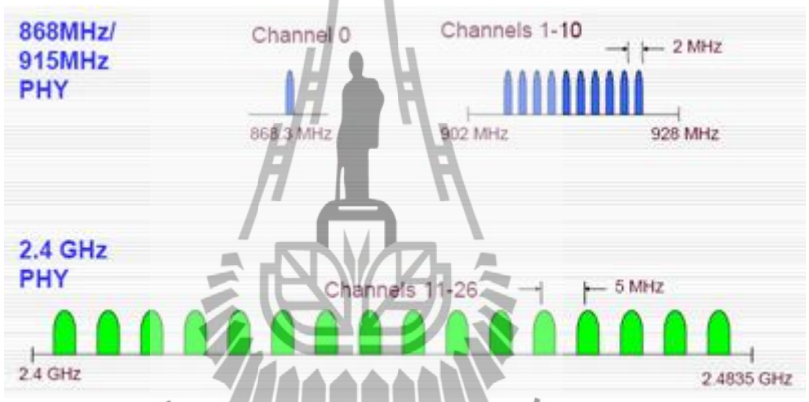
SDA: ขารับส่งข้อมูลด้วยระบบบัส I2C

SCL: ขาสัญญาณนาฬิกาสำหรับการรับส่งข้อมูลด้วยระบบบัส I2C

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.8ZigBee (X-bee)

ZigBeeมาตรฐานสากล กำหนดโดย ZigBee Alliance เป็นการสื่อสารแบบไร้สายที่มีอัตราการรับส่งข้อมูลต่ำ ใช้พลังงานต่ำ ราคาถูก จุดประสงค์ก็เพื่อให้สามารถสร้างระบบที่เรียกว่า Wireless Sensor Network ได้ ซึ่งระบบนี้ จะสามารถทำงาน ในร่ม กลางแจ้ง ทนแดด ทนฝน และอยู่ได้ด้วยแบตเตอรี่ก้อนเล็ก (เช่นถ่าน AA 2 ก้อน) นานเป็นเดือน เป็นปี เหมาะสมใช้งานกับพวก Monitoring ต่าง ๆZigBeeกำหนด ย่านความถี่ใช้งานตามมาตรฐานไว้ 3 ย่านความถี่คือ ย่าน 2.4 Ghz , ย่าน 915 Mhzและย่าน 868 Mhzโดยแต่ละย่านจะมีช่องสัญญาณ 16 ช่อง , 10 ช่อง และ 1 ช่อง ตามลำดับ ส่วน อัตรารับส่งข้อมูล (ทางอากาศ) จะอยู่ที่ 250 Kbps , 40 Kbps , 20 Kbps ตามลำดับเช่นกันโดยในพื้นที่โล่งในระยะสื่อสารประมาณ 200 เมตร สำหรับในอาคารมีระยะสื่อสารประมาณ 30 เมตร สามารถเพิ่มระยะสื่อสารได้โดยการเพิ่ม intermediate node หรือ router



รูปที่ 2.11ย่านความถี่ใช้งานตามมาตรฐาน ZigBee

ZigBeeสามารถสร้างเป็นเครือข่ายได้ ทั้งนี้ ZigBeeได้อ้างอิงมาตรฐานตาม IEEE 802.15.4 โดย IEEE 202.15.4 แบ่งชนิดอุปกรณ์ในเครือข่ายออกเป็น 2 ประเภท คือ FFD (Full Function Device) ซึ่งหมายถึงอุปกรณ์ที่สามารถทำงานได้ทุกอย่างในเครือข่าย และ RFD (Reduce Function Device) ซึ่งหมายถึงอุปกรณ์ที่ถูกลดความสามารถการทำงานในเครือข่าย IEEE 802.15.4 รูปแบบอุปกรณ์ดังนี้

Network Coordinator

- รักษาข้อมูลทั้งหมดของระบบ
- มีความซับซ้อนมากที่สุดในกระบวนการแผนภูมิต้นไม้
- ใช้ความจำและกระบวนการคำนวณมาก

Full Function Device (FFD)

- ใช้โหมดการทำงานของ IEEE 802.15.4 และลักษณะเฉพาะเจาะจงทั้งหมดโดย

มาตรฐาน

- เพิ่มความจำ, กำลังงานคำนวณ ทำให้เป็นอุดมคติสำหรับ Network router function
- สามารถใช้ใน Network edge devices ในขณะที่เครือข่ายติดต่อกับเครือข่ายอื่นๆ หรือ อุปกรณ์ที่ไม่ได้ขึ้นอยู่กับ IEEE 802.15.4 Reduce Function Device (RFD)
- มีการจำกัดรูปแบบการทำงานที่ควบคุมเรื่องความซับซ้อน
- การใช้งานทั่วไปจะใช้ใน Network edge devices

มาตรฐาน IEEE 802.15.4 เป็นมาตรฐานสำหรับการสื่อสารระยะใกล้ที่เน้นการใช้งานพลังงานต่ำและเน้นการสื่อสารที่ทนต่อสภาพสัญญาณรบกวนสูงและเน้นที่มีความง่ายเพื่อให้ชิพการสื่อสารมีราคาถูกส่วนมาตรฐาน ZigBeeเป็นมาตรฐานที่ออกแบบมาเฉพาะสำหรับการติดต่อเครื่องเซ็นเซอร์ชนิดต่างๆเพื่อการพัฒนาแอพพลิเคชัน เช่น เครื่องเซ็นเซอร์สำหรับ Home automation ที่มีการสื่อสารระหว่างเซ็นเซอร์ประตูการสื่อสารแบบ ZigBeeกับกล้องเพื่อบันทึกภาพคนที่เข้ามาในบ้านเป็นแบบ multi-hop routing ที่สามารถส่งข้อมูลไปยังเครื่องเซ็นเซอร์ที่ต้องการโดยผ่านเครื่องเซ็นเซอร์ตัวอื่นๆ ซึ่งคุณสมบัตินี้ไม่ได้รับการสนับสนุนในบลูทูธ การสื่อสารแบบ ZigBeeช่วยให้ขยายรัศมีการส่งของข้อมูลได้ออกไปเรื่อย ZigBeeสำหรับในชั้นnetworkของZigBee จะอยู่บนพื้นฐานของระบบมาตรฐาน IEEE 802.15.4 ซึ่งในชั้นnetwork สามารถรองรับได้ 3 แบบ คือแบบสตาร์(Star) แบบเมช(Mesh) และวงแหวน(Ring Topology) เมื่อมีการเพิ่มจำนวนของเครื่องเซ็นเซอร์ด้วย จากที่ ZigBeeรองรับโทโปโลยีแบบเมช (mesh) ทำให้ต้องมี การค้นหาเส้นทางด้วยตัวเอง (Multiple Routing Algorithms) ซึ่งในตัว ZigBeeนี้จึงจะต้องมีโปรโตคอล (AODV Protocol) ซึ่งการทำงานของ AODV Protocol คือ การส่งข้อความ (Message)ผ่านไปยังโหนดข้างเคียง (Neighbor Node) เพื่อไปยังโหนดที่ต้นทางไม่สามารถติดต่อได้โดยตรง ในระหว่างทางที่ข้อความถูกส่งผ่านไปโปรโตคอล AODV ก็จะทำการค้นหาเส้นทางไปด้วย โดยจะเชื่อว่าจะไม่เกิดการวนลูป (Loop) และพยายามหาเส้นทางที่สั้นที่สุด เพื่อประหยัดเวลาในการส่งข้อมูลและโปรโตคอล AODV ยังสามารถที่จะควบคุมการเปลี่ยนแปลงของเส้นทาง (Route) และสามารถสร้างเส้นทางใหม่ได้หากเกิดข้อผิดพลาด

โครงสร้างของโปรโตคอล ZigBee

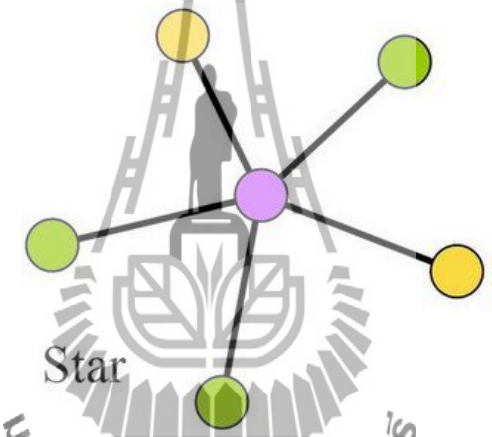
Application Layer เป็นชั้นที่มีส่วนของการทำงานบนเฟรม (Application Framework) ทำหน้าที่ในการจัดการในการเข้าถึงและใช้งานบนเลเยอร์นั้น

Application Support Sub-Layer ทำหน้าที่ในการสร้างเฟรม (Frame) และทำหน้าที่ในการรับส่งข้อมูล

Network Layer ทำหน้าที่ใช้ในการหาเส้นทาง (Routing) ข้อมูลจากต้นทางไปยังปลายทางที่อาจอยู่ภายในเครือข่ายเดียวกันหรือต่างเครือข่ายกัน

เครือข่ายแบบสตาร์ (Star Network)

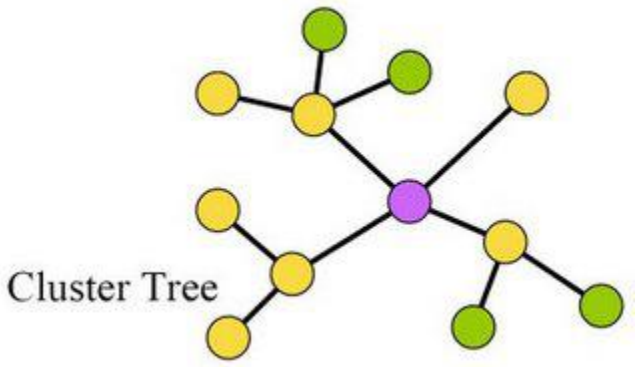
ประกอบด้วยจุดเชื่อมต่อโปรโตคอล ZigBee 1 จุด และอุปกรณ์ปลายทางหลายๆ ในเครือข่ายแบบสตาร์อุปกรณ์ปลายทางทั้งหมดจะสื่อสารกับอุปกรณ์เชื่อมต่อเท่านั้น ถ้าอุปกรณ์ปลายทางหนึ่งต้องการสื่อสารกับอุปกรณ์ปลายทางอื่นๆ ต้องส่งข้อมูลผ่านอุปกรณ์เชื่อมต่อเท่านั้น หรือกล่าวอีกนัยหนึ่งคือ อุปกรณ์เชื่อมต่อทำหน้าที่ส่งข้อมูลไปยังผู้รับ



รูปที่ 2.12 การแสดงเครือข่ายแบบสตาร์

เครือข่ายแบบต้นไม้ (Cluster Tree)

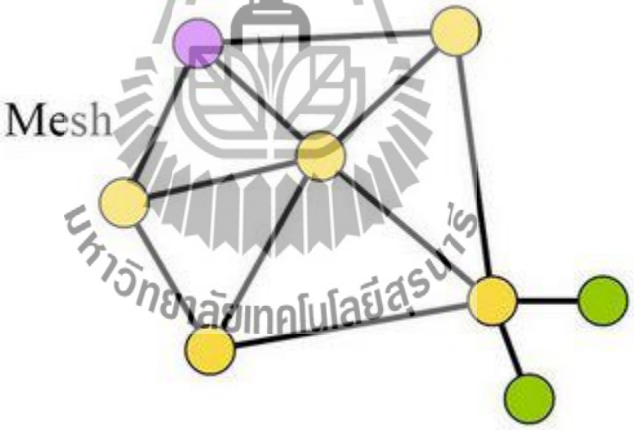
ในเครือข่ายนี้ อุปกรณ์ปลายทางจะสามารถเชื่อมต่อได้กับอุปกรณ์เชื่อมต่อหรือ ZigBee โปรโตคอลเราเตอร์ทำหน้าที่ 2 ประเภท คือ เพิ่มจำนวนโหนดที่สามารถเชื่อมต่ออยู่บนเครือข่ายและขยายขนาดของเครือข่าย เนื่องจากเราเตอร์จะทำหน้าที่ส่งข้อมูลไปยังจุดต่างๆของเครือข่ายได้โดยที่อุปกรณ์ปลายทางไม่จำเป็นต้องอยู่ในระยะการส่งสัญญาณวิทยุ



รูปที่ 2.13การแสดงเครือข่ายต้นไม้

เครือข่ายแบบเมช(Mesh Network)

เครือข่ายแบบเมชใช้กับเครือข่ายแบบต้นไม้ ยกเว้นอุปกรณ์ FFD สามารถส่งข้อมูลไปยัง FFD อื่นได้โดยตรงไม่ต้องผ่านโครงสร้างต้นไม้ ข้อมูลที่ส่งไปยัง RFD จะต้องทำการต่อผ่านอุปกรณ์ RFD ก่อนหน้า ข้อดีของการเชื่อมต่อแบบนี้คือ ช่วยลดอัตราความล่าช้าของการส่งและเพิ่มความน่าเชื่อถือของระบบ



รูปที่ 2.14การแสดงเครือข่ายแบบเมช

เครือข่ายแบบต้นไม้และเครือข่ายแบบเมช มีอีกชื่อว่า เครือข่ายหลายจุด (Multi-Hop) ขณะที่เครือข่ายแบบสตาร์เป็นเครือข่ายจุดเดียว (Single-Hop) เครือข่ายโปรโตคอล ZigBeeเป็นเครือข่ายแบบเชื่อมต่อได้หลายอุปกรณ์พร้อมกัน ซึ่งหมายความว่า จุดเชื่อมต่อในเครือข่ายทุกจุดมีสิทธิในการเข้าถึงตัวกลางที่ใช้ในการสื่อสารเท่าๆกัน มีทุกจุดเชื่อมต่อในเครือข่ายส่งข้อมูลได้ตลอดเวลาที่ช่องสัญญาณว่างอยู่ ในเครือข่ายแบบ Beacon จุดเชื่อมต่อจะสามารถส่งข้อมูลได้ในช่วงเวลาที่ถูกกำหนดล่วงหน้าไว้เท่านั้น

2.9 ET-GSM SIM300CZ

ET-GSM SIM300CZเป็นชุดเรียนรู้และพัฒนาระบบการสื่อสารไร้สายโดยใช้โมดูล GSM/GPRSรุ่น SIM300CZ ของ “SIMCom Ltd.” เป็นอุปกรณ์หลักซึ่ง SIM300CZ เป็นโมดูลสื่อสารระบบ GSM/GPRSขนาดเล็กรองรับระบบสื่อสาร GSM ความถี่ 900/1800/1900MHz โดยสั่งงานผ่านทางพอร์ตสื่อสารอนุกรมRS232 ด้วยชุดคำสั่ง AT Command สามารถประยุกต์ใช้งานได้มากมายหลายรูปแบบไม่ว่าจะเป็นการรับส่งสัญญาณแบบ Voice, SMS, Data, FAX และยังรวมถึงการสื่อสารด้วย Protocol TCP/IP ด้วยซึ่งตามปรกติแล้วถึงแม้ว่าโมดูล SIM300CZ จะมีวงจรและ Firmware บรรจุไว้ภายในตัวเป็นที่เรียบร้อยแล้วก็ยังไม่สามารถนำไปใช้งานได้โดยตรงทันที เนื่องจากในการใช้งานจริงๆนั้นผู้ใช้งานเองจำเป็นต้องออกแบบวงจรรอบนอกที่จำเป็นมาเชื่อมต่อกับขาสัญญาณของตัวโมดูลอีกในบางส่วนไม่ว่าจะเป็นวงจรภาค PowerSupply,วงจรเชื่อมต่อกับ SIM Card รวมไปถึงวงจร Line Driver ของ RS232 เป็นต้นดังนั้นทางทีมงานอีทีทีจึงได้จัดสร้างบอร์ดสำหรับเป็นตัวกลางในการเชื่อมต่อระหว่างโมดูล SIM300CZ กับอุปกรณ์ภายนอกเพื่อให้ผู้ใช้งานสามารถนำโมดูล GSM ของ SIM300CZ ไปทำการทดลองและศึกษาเรียนรู้การสั่งงานต่างๆได้โดยสะดวกก่อนที่จะนำเอาโมดูลตัวนี้ไปออกแบบดัดแปลงและประยุกต์ใช้งานในด้านต่างๆได้ต่อไปในอนาคตซึ่งถึงแม้ว่าวงจรการเชื่อมต่อทั้งหมดที่ทางอีทีทีได้จัดทำขึ้นมานี้จะยังไม่สามารถรองรับการใช้งานทรัพยากรต่างๆที่มีอยู่ภายในโมดูลได้ครบถ้วนทั้งหมดก็ตามทีแต่ในส่วนของการใช้งานโมดูลในส่วนที่เป็นความสามารถหลักๆที่จำเป็นนั้นมีไว้รองรับอย่างครบถ้วนเพียงพอแล้วอย่างไรก็ตามถ้าผู้ใช้งานต้องการพัฒนา Application ที่สูงขึ้นไปก็สามารถประยุกต์ดัดแปลงหรือทำการเชื่อมต่ออุปกรณ์เพิ่มเติมให้กับบอร์ดได้โดยง่ายทั้งนี้ก็เพราะว่าขาสัญญาณต่างๆจากโมดูลในส่วนที่ยังไม่ได้ทำการออกแบบวงจรเตรียมไว้ให้ภายในบอร์ดเช่นขาสัญญาณสำหรับเชื่อมต่อกับ Keyboard,LCD Display และ GPIO ต่างๆนั้นทางอีทีทีเองก็ได้จัดทำเป็นจุดต่อ Connector เตรียมไว้ให้เป็นที่เรียบร้อยแล้วผู้ใช้เพียงแต่ทำการเชื่อมต่อสัญญาณต่างๆจากจุดเชื่อมต่อที่เตรียมไว้ไปยังวงจรส่วนที่ได้ทำการออกแบบไว้ได้โดยสะดวกอยู่แล้ว

มหาวิทยาลัยเทคโนโลยีสุรนารี

คุณสมบัติของบอร์ด

1. มีสวิตช์แบบ Push-Button สำหรับใช้สั่ง เปิด-ปิด การทำงานของโมดูลภายในบอร์ด สามารถวัด
2. มี Socket SIM รองรับ SIM Card พร้อมวงจร ESD ป้องกัน SIM เสียหายสามารถวัด อุณหภูมิได้
3. มีวงจร Regulate แยกอิสระ จำนวน 2 ชุด สามารถใช้กับแหล่งจ่ายภาย Adapter ขนาดตั้งแต่ +5Vขึ้นไป สามารถจ่ายกระแสให้กับโมดูล SIM300CZ และอุปกรณ์เชื่อมต่อต่างๆได้อย่างเพียงพอ

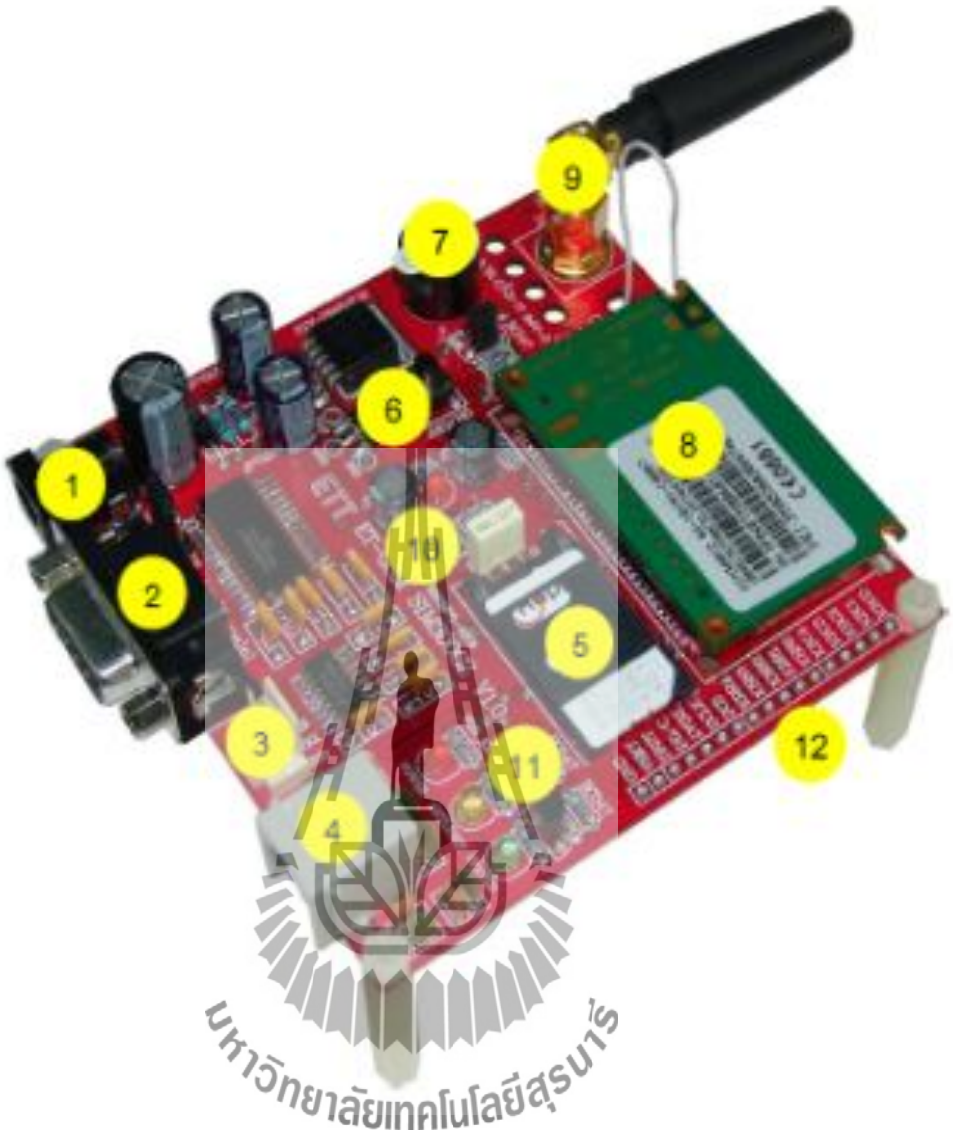
3.1 มีวงจร Regulate ขนาด 4.2V / 3A สำหรับจ่ายให้กับโมดูล SIM300CZ ได้อย่างเพียงพอ สามารถใช้กับ SIM ของระบบ GSM900MHz แบบ 2-Watt ได้อย่างไม่เกิดปัญหา

3.2 มีวงจร Regulate ขนาด 3.3V / 1A สำหรับจ่ายให้กับวงจรเชื่อมต่อภายนอกโดยไม่ต้องไปดึงไฟจากตัวโมดูลมาใช้ ป้องกันปัญหาโมดูลเสียหายจากวงจรภายนอกดึงกระแสเกินพิกัด และสะดวกต่อการออกแบบวงจรเชื่อมต่อเพิ่มเติม โดยไม่ต้องกังวลว่ากระแสจะไม่พอจ่ายให้กับอุปกรณ์

4. มีวงจร Line Driver สำหรับแปลงระดับสัญญาณโลจิกจากโมดูล SIM300CZ ให้เป็น RS232 ระดับมาตรฐานครบทุกเส้นสัญญาณ ทั้งพอร์ตที่ใช้ในการสื่อสารสำหรับสั่งงานโมดูล และ พอร์ตสำหรับใช้ในการพัฒนาโปรแกรม (Debug) สามารถเชื่อมต่อกับพอร์ต RS232 มาตรฐานได้ทันทีทำงานได้
5. มี LED แสดงสถานะพร้อมในบอร์ด สำหรับแสดงสถานะของแหล่งจ่ายไฟ สถานะพร้อมทำงานของโมดูล สถานะในการเชื่อมต่อกับ Network และ สถานะ Power-On/Power-OFF ของโมดูล
6. มีขั้วสำหรับเชื่อมต่อกับ Handset (ชุดปากพูด และหูฟัง ของโทรศัพท์บ้าน) โดยใช้ขั้วต่อแบบ RJ11มาตรฐาน พร้อมวงจร Voice Filter สามารถนำชุด Handset ของโทรศัพท์บ้านต่อเข้ากับบอร์ดทางขั้วต่อแบบ RJ11 สำหรับใช้พูดคุย โทรออก และ รับสายได้โดยสะดวก
7. มี Buzzer พร้อมวงจรขับเพื่อสร้างสัญญาณเสียง ในกรณีมีการโทรเรียกเข้ามายังโมดูล
8. มีจุดยึดเสาอากาศ สำหรับใช้เป็นจุดพักสำหรับเชื่อมต่อกับเสาอากาศแบบต่างๆได้โดยสะดวก
9. มีขั้วต่อสำหรับติดตั้งโมดูล SIM300CZ พร้อมเสารองและสกรูยึดโมดูลกับตัวบอร์ด
10. มีจุดต่อสัญญาณอื่นๆที่เหลือจากโมดูล เช่น Keyboard, Display ,GPIO ,Battery Charger ฯลฯสำหรับให้ผู้ใช้ต่อขยายไปยังวงจรที่ออกแบบเพิ่มเติมได้โดยง่ายและสะดวก

มหาวิทยาลัยเทคโนโลยีสุรนารี

โครงสร้างของบอร์ด



รูปที่ 2.15ET-GSM SIM300CZ

หมายเลข 1: เป็น JACK DC-IN แบบมีขั้ว โดยมีด้านนอกเป็นขั้วบวก และด้านในเป็น GND ใช้สำหรับรับแหล่งจ่ายไฟจากภายนอกโดยออกแบบให้ใช้กับ แหล่งจ่ายไฟขนาด 5V ขึ้นไปที่จ่ายกระแสได้ 1A ถึง 3A

หมายเลข 2: เป็น ขั้วต่อ RS232(DCE) แบบ DB9 ตัวเมีย สำหรับใช้เชื่อมต่อกับสัญญาณ RS232(DTE) แบบ DB9 ตัวผู้ จากคอมพิวเตอร์ PC หรืออุปกรณ์ภายนอกอื่นๆ โดยใช้สาย 9 Pin แบบต่อตรง

มหาวิทยาลัยเทคโนโลยีสุรนารี

หมายเลข 3: เป็น ขั้วต่อ DEBUG ใช้สำหรับพัฒนา และ DEBUG โปรแกรม สำหรับต่อกับ RS232 ในกรณีที่ต้องการพัฒนาโปรแกรมเพิ่มเติมให้กับโมดูล SIM300CZ เอง

หมายเลข 4: เป็น ขั้วต่อ RJ11 สำหรับใช้เชื่อมต่อกับชุด Handset ในกรณีที่ต้องการใช้งานโมดูล SIM300CZ เพื่อโทรออกและรับสาย โดยสามารถเชื่อมต่อกับ Handset มาตรฐานได้ทั่วไป

หมายเลข 5: เป็น Socket สำหรับติดตั้ง SIM Card ให้กับโมดูล

หมายเลข 6: เป็น Switch Push-Button สำหรับใช้ Power-On และ Power-OFF ตัวโมดูล

หมายเลข 7: เป็น Buzzer สำหรับสร้างเสียงเรียกเข้าในกรณีที่มีการโทรเข้ามายังโมดูล SIM300CZ

หมายเลข 8: เป็น จุดรองรับโมดูล SIM300CZ พร้อมเสาและสกรูสำหรับยึดโมดูลกับบอร์ด

หมายเลข 9: เป็น จุดยึด Connector เสาอากาศ GSM/GPRS ย่านความถี่ 900/1800/1900MHz

หมายเลข 10: เป็น LED แสดงแหล่งจ่าย VBAT โดยจะติดสว่างเมื่อมีการจ่ายไฟให้บอร์ดแล้ว

หมายเลข 11: เป็น LED แสดงสถานะของบอร์ด ซึ่งมีด้วยกัน 3 ดวงคือ

11.1 POWER สีแดง จะติดสว่าง เมื่อโมดูลอยู่ในสถานะ Power-ON

11.2 NETLIGHT สีเหลือง จะกระพริบเมื่อโมดูลอยู่ในสถานะ Power-ON

11.3 STATUS สีเขียว จะติดสว่างเมื่อโมดูลอยู่ในสถานะ Power-ON

หมายเลข 12: เป็น จุดต่อสัญญาณเพิ่มเติมในกรณีที่ต้องการประยุกต์ใช้งานโมดูลเพิ่มเติม

คุณสมบัติของโมดูล SIM300CZ

1. รองรับความถี่ GSM/GPRS 900/1800/1900MHz
2. รองรับ GPRS Multi-Slot Class10 และ GPRS Mobile Station Class B
3. รองรับมาตรฐานคำสั่ง AT Command (GSM 07.07 / 07.05 และคำสั่งเพิ่มเติมจาก SIMCOM)
4. องรับ SIM Applications Toolkit
5. ทำงานที่ย่านแรงดัน 3.4V ถึง 4.5V
6. รองรับการเชื่อมต่อภายนอก

6.1 ใช้ได้กับ SIM 3V และ 1.8V

6.2 มีวงจร Analog Audio (MIC & Speaker) จำนวน 2 ชุด

6.3 รองรับ 5x5 Keypad Interface & SPI LCD Interface

6.4 มีระบบ RTC พร้อมวงจร Backup

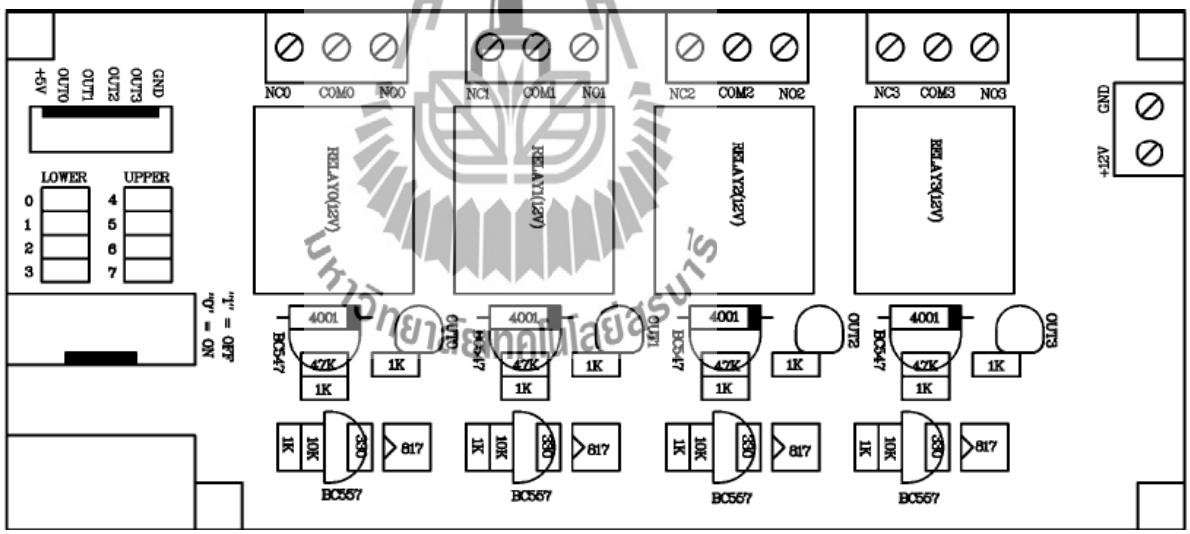
6.5 มีขั้วต่อเสาอากาศภายนอกแบบ Connector และจุดเชื่อมต่อแบบ PAD

6.5 มีระบบ Battery Charge ในตัว

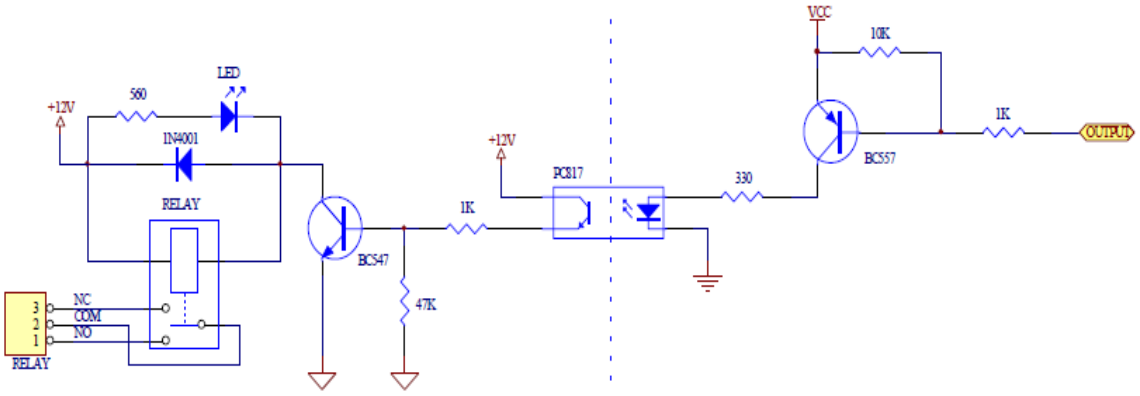
มหาวิทยาลัยเทคโนโลยีสุรนารี

2.10 Et-opto relay4

บอร์ด**Et-opto relay4** เป็นชุดOpto-isolate outputแบบหน้าสัมผัส Relayขนาด 4 ช่องใช้สำหรับควบคุมการทำงานของอุปกรณ์ไฟฟ้าต่างๆในลักษณะของหน้าสัมผัส ON/OFF โดยใช้สัญญาณในการควบคุมการทำงานของ Relayด้วยสัญญาณโลจิก TTL ซึ่งวงจรสำหรับควบคุมการทำงานของ Relayนั้นจะใช้วงจรแบบOpto-isolate ในการควบคุมการทำงานเพื่อป้องกัสัญญาณรบกวนต่างๆที่เกิดจากการทำงานของ Relayและอุปกรณ์ Output ที่ควบคุมการทำงานจากหน้าสัมผัส Relayที่จะย้อนกลับมารบกวนการทำงานของวงจรควบคุมหรือไมโครคอนโทรลเลอร์ต่างๆทำให้ประสิทธิภาพการทำงานของระบบไมโครคอนโทรลเลอร์ต่างๆมีความน่าเชื่อถือมากยิ่งขึ้นโดยการทำงานของวงจรนั้นจะออกแบบให้การทำงานของ Relayทำงานด้วยสภาวะโลจิก “0” และหยุดทำงานด้วยสภาวะโลจิก “1” เพื่อป้องกันไม่ให้ Relayทำงานในขณะที่ระบบไมโครคอนโทรลเลอร์เกิดการ Reset หรือสัญญาณควบคุมของบอร์ด Et-opto relay4 ถูกปล่อยลอยอยู่



รูปที่ 2.16 โครงสร้างของบอร์ด Et-opto relay4



รูปที่ 2.17 วงจรของ ET-OPTO RELAY4 ขนาด 1 ช่อง

การทำงานของบอร์ด

ในการใช้งานบอร์ด Et-opto relay4 เพื่อควบคุมการทำงานของอุปกรณ์ต่างๆในลักษณะของหน้าสัมผัสสวิตช์นั้นจะต้องมีการป้อนแรงดันไฟตรงขนาด +12VDC เพื่อใช้เป็นไฟเลี้ยงขดลวดของ Relayด้วยซึ่งแรงดัน +12VDC นี้ควรเป็นแรงดันไฟตรงที่ได้จากหม้อแปลงซึ่งแยกขดกับชุดที่จะจ่ายให้กับบอร์ดไมโครคอนโทรลเลอร์ด้วยเพื่อป้องกันการรบกวนที่จะเกิดขึ้นกับวงจรไมโครคอนโทรลเลอร์ในขณะที่ Relayหรืออุปกรณ์ Output ที่ถูกควบคุมจาก Relayทำงานซึ่งการทำงานของบอร์ดจะเริ่มจากเมื่อมีการป้อนสถานะโลจิก “0” ให้กับจุด Outputของวงจรซึ่งจะส่งผลให้ขาBase ของทรานซิสเตอร์ BC557 ที่ต่ออยู่กับภาค Input ของOpto isolate (PC817) ได้รับการไบอัสที่ถูกต้องทำให้ทรานซิสเตอร์นำกระแสซึ่งจะทำให้มีแรงดัน +VCC(+5VDC)ไหลออกจากขา Emitter ไปยัง Collector ผ่านตัวต้านทาน 330โอห์มเพื่อจำกัดกระแสให้กับOptoisolate (PC817) ซึ่งส่งผลให้วงจรOptoisolateทำงานเป็นผลให้เกิดมีกระแสไหลจากแหล่งจ่าย +12VDC ผ่านภาค Outputของ Opto isolate (PC817) โดยแรงดัน +12VDC นี้จะถูกจำกัดขนาดของกระแสด้วยตัวต้านทานค่า 1KOHM ก่อนที่จะป้อนให้กับขา Base ของทรานซิสเตอร์ BC547 ที่ต่ออยู่กับวงจรภาค Output ของOpto isolateเป็นผลให้ทรานซิสเตอร์ BC547 นำกระแสซึ่งก็จะทำให้มีกระแสไฟไหลผ่านขดลวดของ Relayทำให้ Relayและ LED ทำงานโดยในการประยุกต์ใช้งานบอร์ดนั้นจะใช้สำหรับควบคุมการทำงานของอุปกรณ์ต่างๆที่ต้องการควบคุมการทำงานด้วยวิธีการ On/Offของหน้าสัมผัสโดยลักษณะของหน้าสัมผัส Relayนั้นจะมีให้เลือกใช้งาน 2 แบบคือNO และ NC โดยในการต่อใช้งานจะต้องเลือกแบบใดแบบหนึ่งโดยจะต่อใช้งานร่วมกับขา COM (Common)

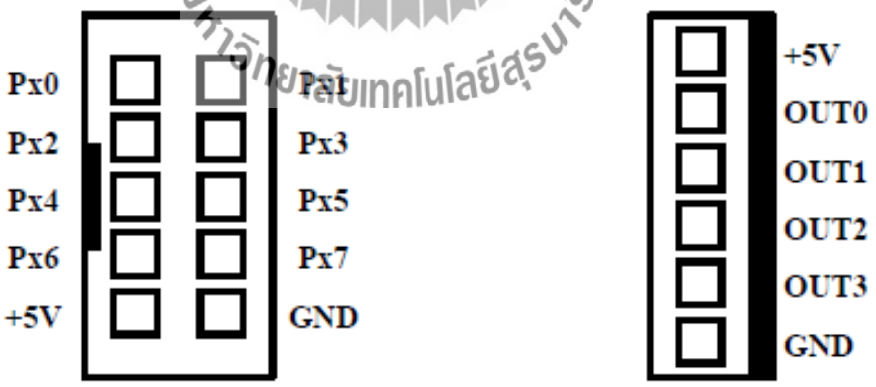
- NO หรือ Normal Open ซึ่งหมายถึงหน้าสัมผัสแบบปรกติเปิดและจะทำงานเมื่อ Relay เริ่มต้นทำงาน

- NC หรือ Normal Close ซึ่งหมายถึงหน้าสัมผัสแบบปรกติปิดโดยจะหยุดทำงานเมื่อRelay เริ่มทำงาน

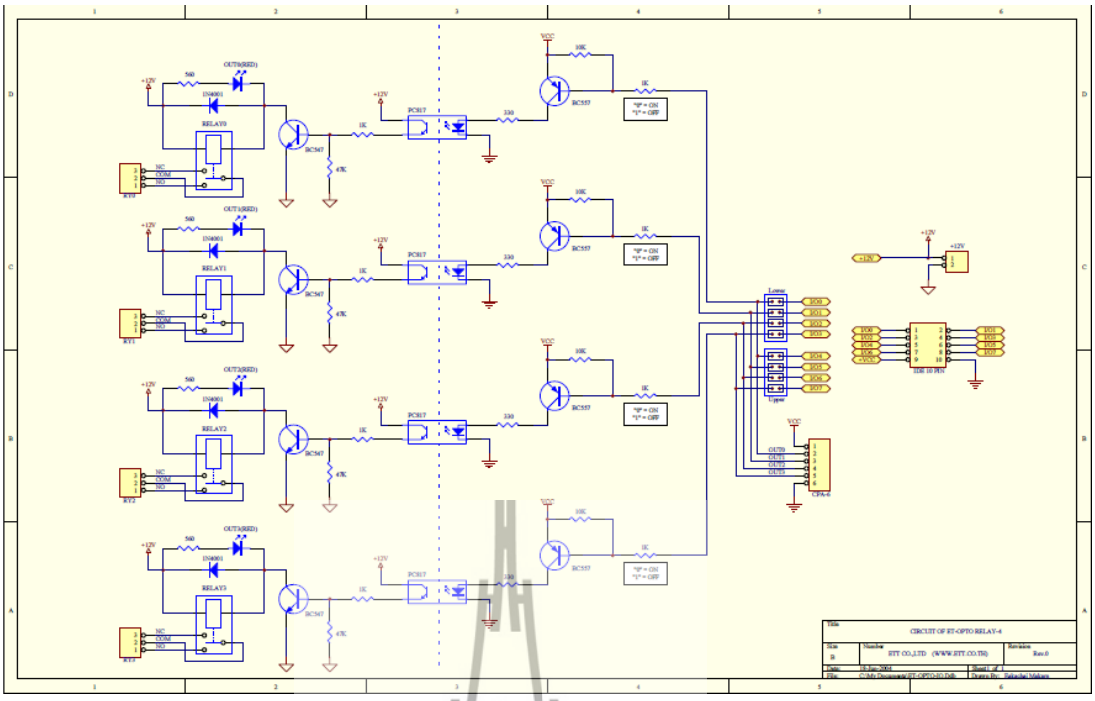
สำหรับการเชื่อมต่อสัญญาณ TTL ไปให้กับวงจรภาค Output ของไมโครคอนโทรลเลอร์นั้นจะสามารถกระทำได้ 2แบบด้วยกันคือต่อผ่านขั้วต่อแบบ IDE ขนาด 10 Pin โดยใช้สายแพร์เป็นสายนำสัญญาณโดยสามารถเลือกกำหนดสัญญาณ TTL ที่ได้จากการทำงานของบอร์ดได้ 2 กลุ่มคือ Lowerและ Upperส่วนอีกวิธีหนึ่งจะสามารถกระทำได้โดยต่อสัญญาณผ่านขั้วต่อแบบ CPA ขนาด 6 Pin ดังตาราง

ตารางที่ 2.2 จุดเชื่อมต่อของบอร์ด Et-opto relay4 กับไมโครคอนโทรลเลอร์

ขั้วต่อ Output (Terminal)	Lower		Upper	
	IDE 10 Pin	CPA 6 Pin	IDE 10 Pin	CPA 6 Pin
Relay0	Px0	Out0	Px4	Out0
Relay1	Px1	Out1	Px5	Out1
Relay2	Px2	Out2	Px6	Out2
Relay3	Px3	Out3	Px7	Out3



รูปที่ 2.18 การจัดเรียงสัญญาณของขั้ว IDE 10 Pin และ CPA-6 Pin

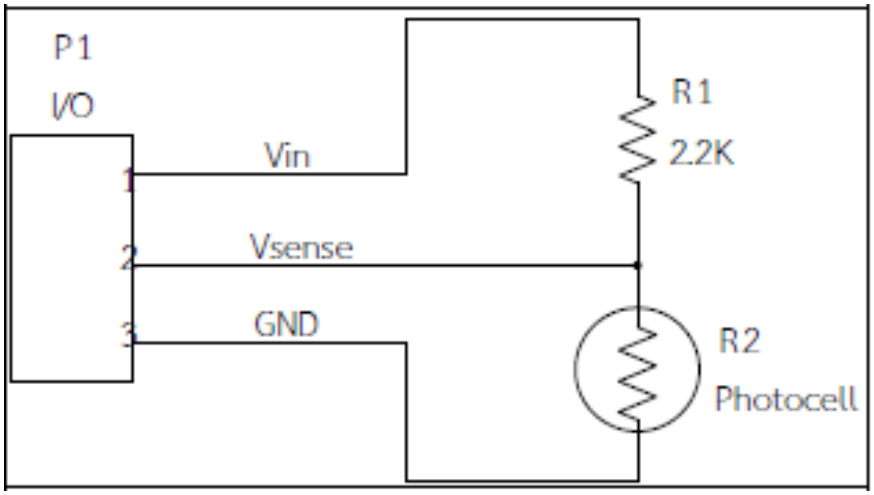


รูปที่ 2.19วงจรทั้งหมดของ Et-opto Relay4

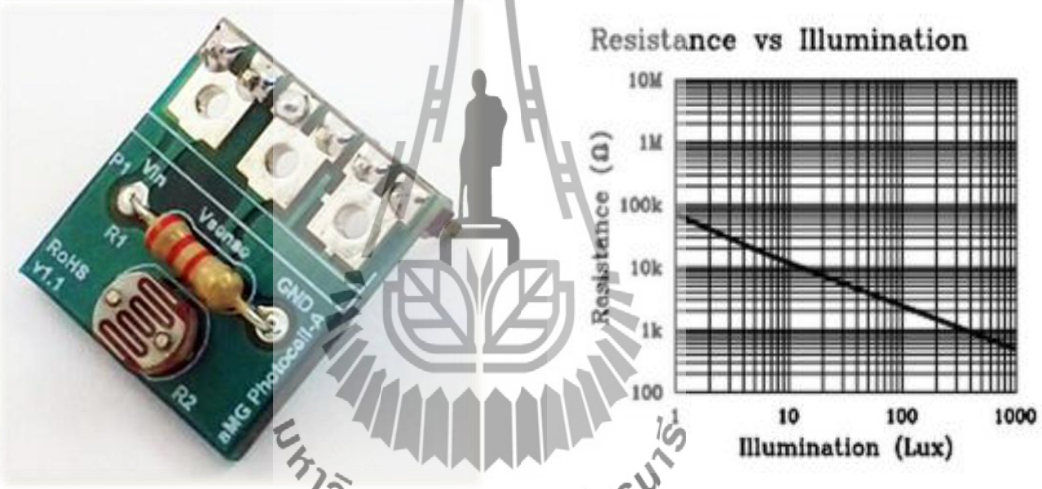
2.11 เซ็นเซอร์วัดความเข้มแสง

เซนเซอร์วัดความเข้มแสงเป็นเซนเซอร์ที่ใช้วัดปริมาณความเข้มของแสงอาทิตย์โดยใช้อุปกรณ์โฟโต้เซลล์ (Photo Cell) ซึ่งทำหน้าที่เปลี่ยนพลังงานแสงเป็นพลังงานไฟฟ้าโดยค่าความต้านทานขึ้นอยู่กับแสงที่ตกกระทบบนตัวโฟโต้เซลล์ถ้ามีแสงมากระทบจะให้ค่าความต้านทานต่ า (หรือได้ค่าแรงดันไฟฟ้าต่ า) แต่ถ้าไม่มีแสงมาตกกระทบจะให้ค่าความต้านทานสูง (ได้ค่าแรงดันไฟฟ้าสูง) สัญญาณที่ได้จากการตรวจวัดเป็นสัญญาณแอนาลอก โดยผ่านกระบวนการแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิตอล ใช้การเข้ารหัสข้อมูลขนาด 12 บิต จะได้ข้อมูลความเข้มแสงที่เป็นค่าของแรงดันไฟฟ้า สามารถหาค่า Vsenseได้จากการแบ่งแรงดันดังสมการ

$$V_{sense} = V_{in} * \frac{R_{photocell}}{R_{photocell} + 2200}$$



รูปที่ 2.20วงจรเซนเซอร์วัดความเข้มแสง



(ก) (ข)

รูปที่ 2.21 (ก) เซนเซอร์วัดความเข้มแสง (ข) กราฟความสัมพันธ์ระหว่างค่าความต้านทานกับความเข้มแสง

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.12 เซ็นเซอร์วัดความชื้นในดิน (EC-5)

คุณสมบัติเบื้องต้น

1. ใช้แหล่งจ่ายกระแสตรงขนาด 2.5 ถึง 3.6 โวลต์

2. สามารถทำงานได้ในช่วงอุณหภูมิตั้งแต่ -40 องศาเซลเซียสถึง 60 องศาเซลเซียสช่วงของการวัดคือ ตั้งแต่ 0 โวลต์จนถึงค่าแรงดันสูงสุด (แรงดันไฟเลี้ยงที่จ่ายให้เซ็นเซอร์) และ

3. ใช้สมการในการคำนวณหาเปอร์เซ็นต์ความชื้นในดินดังนี้

$$\text{เปอร์เซ็นต์ความชื้นในดิน} = 18(\text{แรงดันที่อ่านได้จากเซ็นเซอร์}) + 6$$

รูปที่ 2.22 เซ็นเซอร์วัดความชื้นในดิน

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.13 วาล์วไฟฟ้ารุ่น Hunter PGV 203 ขนาด 2 นิ้ว

คุณสมบัติเบื้องต้น

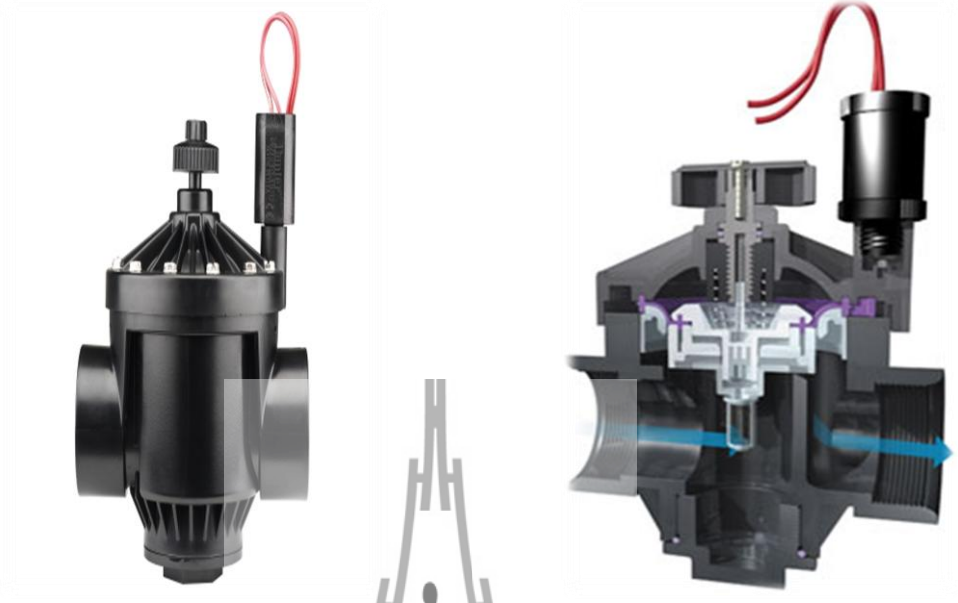
1. อัตราการไหล 2 ถึง 120 แกลลอนต่อนาที

2. ทนแรงดัน 1.4 ถึง 10.3 บาร์

3. ใช้งานที่อุณหภูมิไม่เกิน 66 องศาเซลเซียส

4. ใช้แหล่งจ่ายแรงดันกระแสสลับ 24 โวลต์

5. กินกระแส 370 มิลลิแอมป์



(ก) (ข)

รูปที่ 2.23 (ก) วาล์วไฟฟ้า hunter PGV203 (ข) ภาพตัดของวาล์วไฟฟ้า hunter PGV203

มหาวิทยาลัยเทคโนโลยีสุรนารี

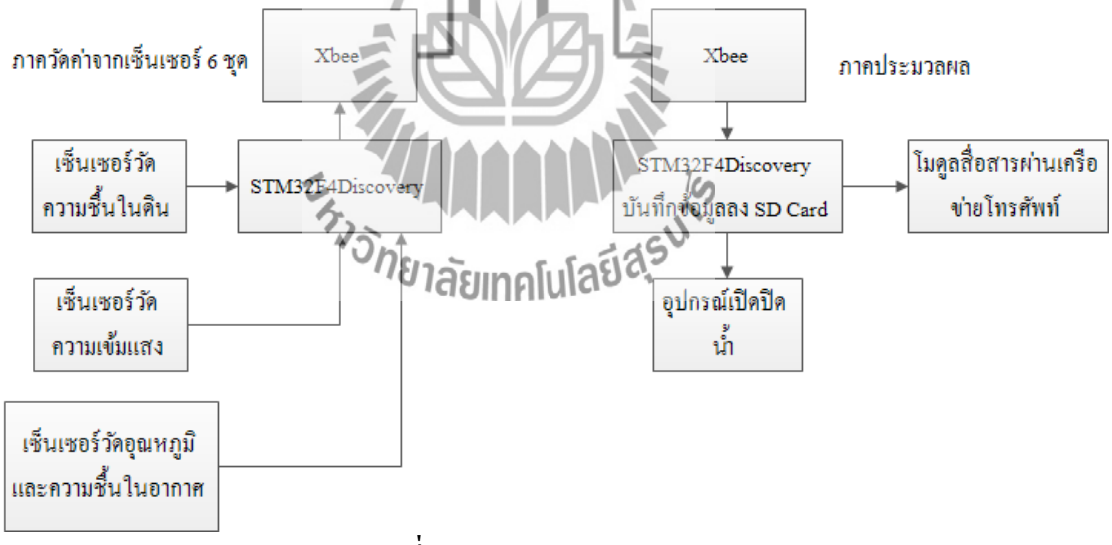
บทที่3
การออกแบบระบบ

3.1 บทนำ

บทนี้กล่าวถึงการออกแบบระบบของเครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลังโดยมีการวัดค่าจากอุปกรณ์วัดค่าต่างๆ การส่งสัญญาณแบบไร้สาย อุปกรณ์ประมวลผล อุปกรณ์เปิดปิดน้ำ รวมไปถึงการออกแบบฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการทำโครงงาน

3.2 การออกแบบระบบเบื้องต้น

การออกแบบระบบของเครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลังแบ่งออกเป็นสองส่วนคือส่วนของภาควัดค่าด้วยอุปกรณ์วัดค่าต่างๆ หรือโหนดย่อย (Router) และส่วนของภาคประมวลผล หรือโคออร์ดิเนเตอร์ (Coordinator) ดังแผนภาพรวมของระบบที่แสดงในรูปที่ 3.1



รูปที่ 3.1 แผนภาพรวมของระบบ

3.2.1 ภาควัดค่าด้วยอุปกรณ์วัดค่าต่างๆ หรือโหนดย่อย (Router)

ใช้บอร์ด STM32F4Discovery รับค่าจากอุปกรณ์วัดค่าแสง อุปกรณ์วัดค่าความชื้นในดิน อุปกรณ์วัดอุณหภูมิและความชื้น โดยการติดตั้งอุปกรณ์วัดความชื้นในดินจะติดตั้งโหนดย่อย (Router) ละ 2 จุดโดยมีความลึกในการติดตั้งต่างกัน คือ 15 เซนติเมตร และ 25 เซนติเมตร แปลงค่า

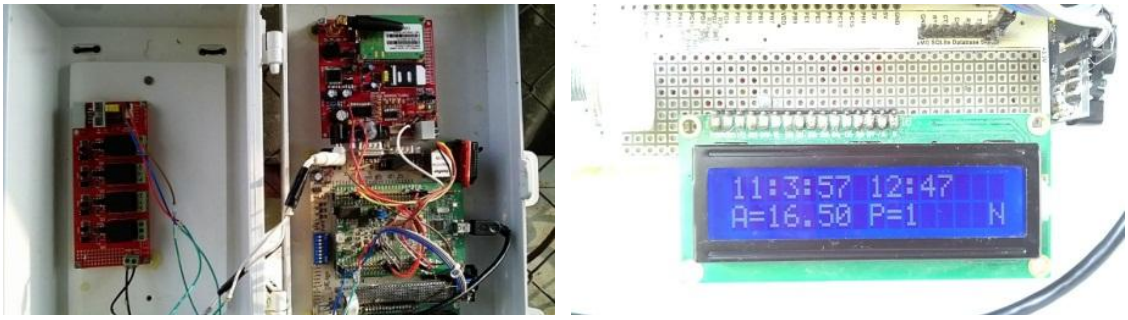
จากแอนะล็อกเป็นดิจิตอล และทำการส่งค่าทั้งหมดแบบไร้สายผ่านการติดต่อแบบอนุกรมโดยใช้โมดูลXbee ในการส่งข้อมูลไปยังภาคประมวลผล และใช้พลังงานจากแสงอาทิตย์โดยมีโซล่าเซลล์ที่มีกำลัง 10 วัตต์ ต่อกับวงจรชาร์ตแบตเตอรี่จากนั้นแปลงแรงดันจาก 12โวลต์จากแบตเตอรี่ให้เหลือ 5 โวลต์เพื่อเป็นแหล่งจ่ายสำหรับบอร์ด STM32F4Discovery จึงทำให้สามารถใช้งานในเวลากลางคืนซึ่งไม่มีแสงอาทิตย์ได้ นอกจากนั้นยังมีวงจรสำหรับวัดค่าแรงดันจากแบตเตอรี่อีกด้วย ทำให้สามารถตรวจสอบแรงดันแบตเตอรี่ได้ตลอดเวลาจากข้อมูลที่ถูกส่งขึ้นไปเก็บไว้ที่เซิร์ฟเวอร์ ลักษณะการติดตั้งโหนดย่อยจะเป็นดังรูปที่ 3.2 (ก) และรูปที่ 3.1 (ข)

(ก) (ข)

รูปที่ 3.2 (ก) โหนดย่อยที่ติดตั้งในไร่(ข) วงจรที่อยู่ในโหนดย่อย

3.2.2 ภาคประมวลผลและควบคุมการเปิดปิดน้ำ (Coordinator)

ใช้บอร์ด STM32F4Discovery ทำการโปรแกรมให้เป็นภาคประมวลผลโดยจะทำการรับค่าจากโหนดย่อยทั้ง 6 จุดผ่านโมดูล Xbee แล้วนำค่าทั้งหมดบันทึกลงในหน่วยความจำ SD-Card และนำค่าทั้งหมดส่งขึ้นเซิร์ฟเวอร์โดยใช้โมดูลสื่อสารผ่านเครือข่ายโทรศัพท์จากนั้นนำค่าความชื้นในดินทั้งหมดมาหาค่าเฉลี่ยเพื่อทำการเปรียบเทียบกับค่าความชื้นในดินอ้างอิงต่ำสุดสำหรับการสั่งเปิดปั้มน้ำ และเปรียบเทียบกับค่าความชื้นในดินอ้างอิงสูงสุดที่สำหรับการสั้งปิดปั้มน้ำ โดยใช้ข้อมูลค่าความชื้นในดินจากการทดลองให้น้ำแบบธรรมดา (ให้คนควบคุมการเปิด-ปิดปั้มน้ำ) เป็นเวลา 1 สัปดาห์ ทั้งนี้ภาคประมวลผลและควบคุมการเปิด-ปิดน้ำได้ทำการติดตั้งในบริเวณห้องควบคุมเครื่องสูบน้ำซึ่งอยู่บริเวณใกล้กับบ่อน้ำและใช้แหล่งจ่ายไฟกระแสสลับ 220 โวลต์แปลงให้เป็นกระแสตรงขนาด 5 โวลต์เพื่อจ่ายให้กับ STM32F4Discovery และโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ดังรูปที่ 3.3 (ก) และรูปที่ 3.3 (ข)



(ก) (ข)

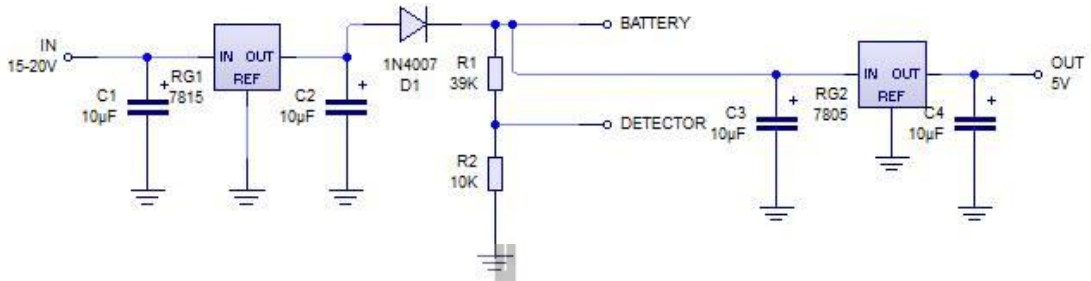
รูปที่ 3.3 (ก) (ข) ภาคประมวลผลและควบคุมการเปิด-ปิดน้ำ

ในส่วนของการควบคุมการจ่ายน้ำสำหรับไร่มันสำปะหลัง ได้ออกแบบการใช้งานใน 2 ลักษณะการใช้งาน คือการควบคุมการเปิด-ปิดเครื่องสูบน้ำเพื่อสูบน้ำจากบ่อและจ่ายน้ำด้วยระบบน้ำหยด หรือในอีกกรณีคือที่มีถังเก็บน้ำอยู่บนที่สูงที่สามารถปล่อยน้ำลงมาได้ จะควบคุมการเปิดปิดน้ำโดยใช้โซลินอยด์วาล์วและจ่ายน้ำด้วยระบบน้ำหยดดังรูปที่ 3.4

รูปที่ 3.4 ระบบน้ำหยด

3.3 การออกแบบฮาร์ดแวร์

การออกแบบวงจรที่ใช้ในโครงงานเครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลังวงจรแรกคือวงจรชาร์ตแบตเตอรี่จากโซล่าเซลล์และแปลงแรงดันจาก 12 โวลต์เป็น 5 โวลต์สำหรับไมโครคอนโทรลเลอร์ STM32F4Discovery



รูปที่ 3.5 วงจรชาร์ตแบตเตอรี่และแปลงแรงดัน

จากรูปที่ 3.5 วงจรรับอินพุตกระแสตรงแรงดัน 15 ถึง 20 โวลต์ จากโซล่าเซลล์ผ่านตัวเก็บประจุ C1 ทำหน้าที่กรองแรงดันให้เรียบ จากนั้นเข้าไปยังไอซี RG1 เบอร์ 7815 (สามารถดูข้อมูลเบื้องต้นของไอซีเบอร์ 7815 ได้จากตารางที่ 3.1) ทำหน้าที่รักษาระดับแรงดันให้คงที่ 15 โวลต์ และตัวเก็บประจุ C2 ทำหน้าที่กรองแรงดันให้เรียบเพื่อชาร์ตแบตเตอรี่โดยมีไดโอด D1 เบอร์ 1N4007 ทำหน้าที่ป้องกันกระแสไหลย้อนกลับจากแบตเตอรี่สู่โซล่าเซลล์ในกรณีที่แรงดันอินพุตต่ำกว่าแรงดันแบตเตอรี่ ตัวต้านทาน R1 และ R2 เป็นวงจรแบ่งแรงดันเพื่อให้แรงดันที่ออกไปสู่ขาตรวจสอบแบตเตอรี่ของไมโครคอนโทรลเลอร์ STM32F4Discovery มีขนาดแรงดันสูงสุดไม่เกิน 3 โวลต์ที่แบตเตอรี่เต็ม จากนั้นแรงดัน 12 โวลต์จากแบตเตอรี่จะผ่านไปยัง C3 ซึ่งทำหน้าที่กรองแรงกันให้เรียบเพื่อส่งต่อไปยังไอซี RG2 เบอร์ 7805 (สามารถดูข้อมูลเบื้องต้นของไอซีเบอร์ 7805 ได้จากตารางที่ 3.2) ซึ่งทำหน้าที่รักษาระดับแรงดันให้คงที่ 5 โวลต์ และผ่าน C4 เพื่อกรองแรงดันให้เรียบก่อนออกเอาต์พุตซึ่งเป็นแหล่งจ่ายกระแสตรงขนาด 5 โวลต์ให้กับไมโครคอนโทรลเลอร์ STM32F4Discovery

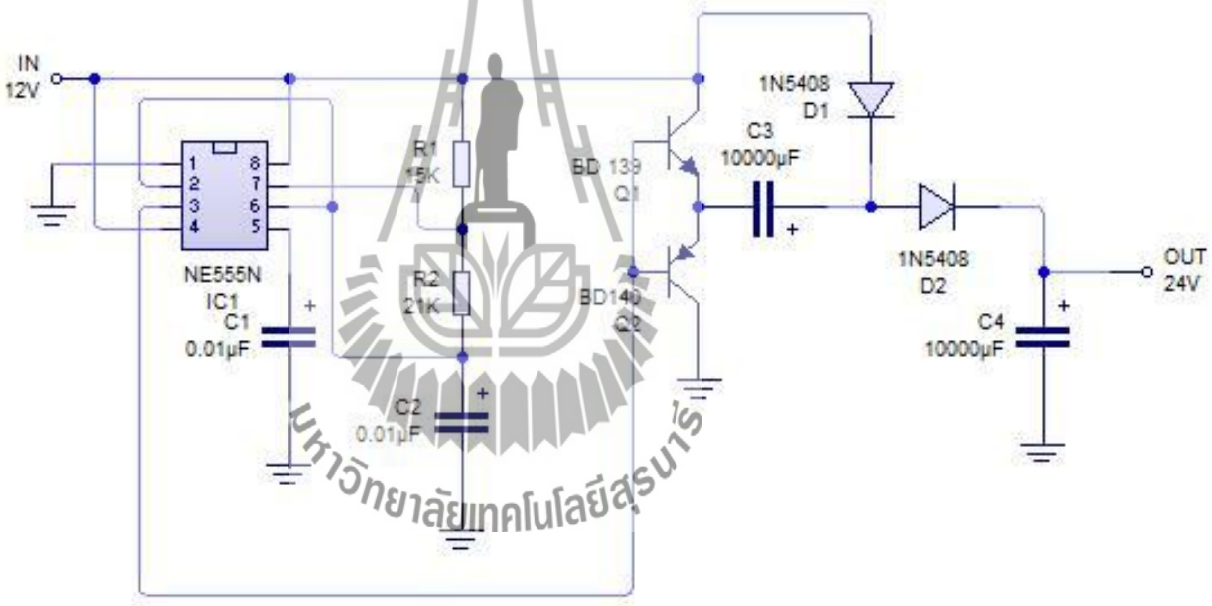
ตารางที่ 3.1 ข้อมูลเบื้องต้นของไอซีเบอร์ 7815

ขาสัญญาณ	หน้าที่
IN	อินพุต
REF	กราวด์
OUT	เอาต์พุต 15 โวลต์

ตารางที่ 3.2 ข้อมูลเบื้องต้นของไอซีเบอร์ 7805

ขาสัญญาณ	หน้าที่
IN	อินพุต
REF	กราวด์
OUT	เอาต์พุต 5 โวลต์

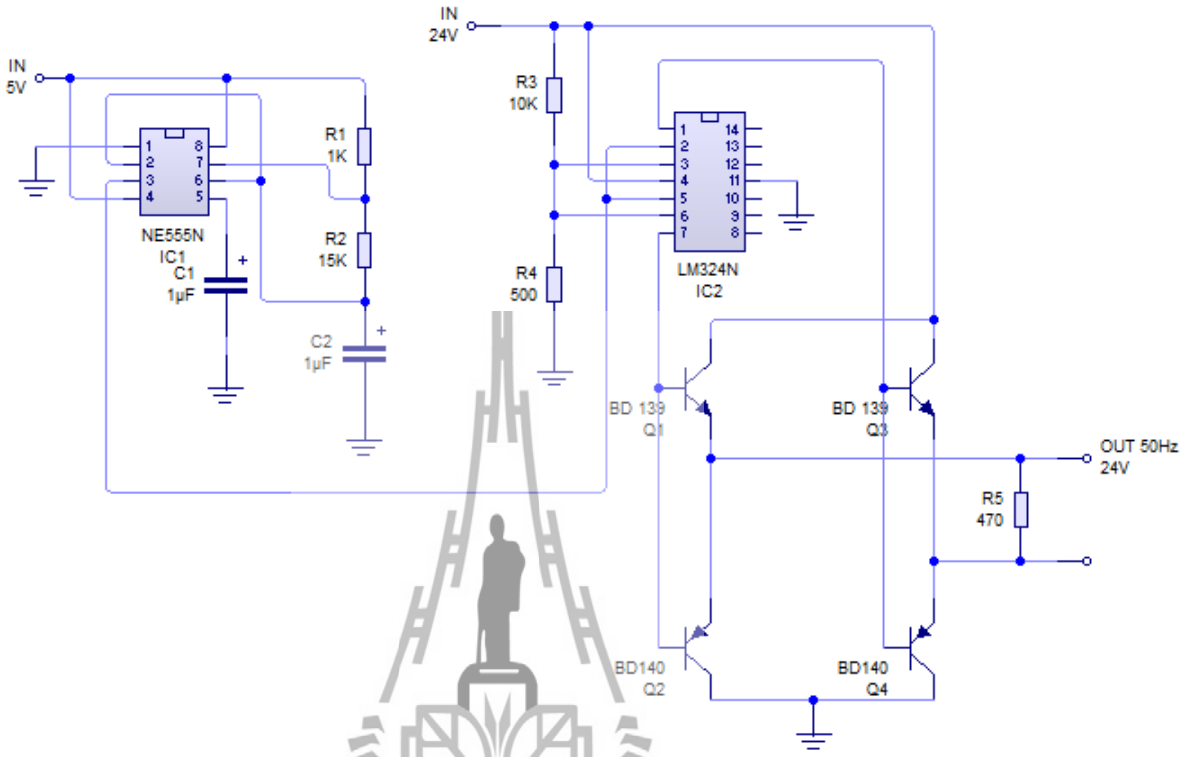
วงจรที่สองคือวงจรแปลงแรงดันกระแสตรงจากแบตเตอรี่ 12 โวลต์เป็นแรงดันกระแสสลับ 24 โวลต์เป็นแหล่งจ่ายสำหรับวาล์วควบคุมด้วยไฟฟ้า โดยแบ่งออกเป็น 2 วงจรย่อย คือ วงจรแปลงแรงดันกระแสตรง 12 โวลต์เป็นกระแสตรง 24 โวลต์ และวงจรแปลงกระแสตรง 24 โวลต์เป็นกระแสสลับ 24 โวลต์ 50 เฮิรตซ์



รูปที่ 3.6วงจรแปลงแรงดันกระแสตรง 12 โวลต์เป็น กระแสตรง 24 โวลต์

จากรูปที่ 3.6 วงจรจะมีอินพุตกระแสตรง 12 โวลต์ จ่ายให้กับชุดกำเนิดความถี่ซึ่งประกอบไปด้วย IC1 เบอร์ NE555N, C1, C2, R1, และ R2 1 (สามารถดูข้อมูลเบื้องต้นของไอซีเบอร์ NE555N ได้จากตารางที่ 3.3) และมีเอาต์พุตความถี่ 2.5 kเฮิรตซ์ (โดยสามารถคำนวณความถี่ได้จากสมการที่ 3.1 ถึง สมการที่ 3.4) ออกจากขาที่ 3 ของ IC1 เพื่อควบคุมการทำงานของทรานซิสเตอร์ Q1 และ Q2 ให้เปิดปิดสลับกัน เมื่อ Q1 ปิดและ Q2 เปิดจะมีกระแสไหลผ่านไดโอด D1 เบอร์ 1N5408 ทำให้ตัวเก็บประจุ C3 ชาร์ตจนเต็มมีขนาดแรงดันประมาณ 12 โวลต์ จากนั้นเมื่อ Q1 เปิด และ Q2 ปิด จะมีกระแสไหลผ่านไดโอด D2 เบอร์ 1N5408 ทำให้ตัวเก็บประจุ C4 ชาร์ตจนเต็มมี

ขนาดแรงดันประมาณ 24 โวลต์เนื่องจากแรงดันอินพุต 12 โวลต์รวมกับแรงดันจากตัวเก็บประจุ C3 อีก 12โวลต์ จะได้เอาต์พุตกระแสตรงขนาด 24 โวลต์เพื่อนำไปแปลงให้เป็นแรงดันกระแสสลับ 24 โวลต์ต่อไป



รูปที่ 3.7 วงจรแปลงแรงดันกระแสตรง 24 โวลต์เป็นกระแสสลับ 24 โวลต์

จากรูปที่ 3.7 วงจรจะมีอินพุตกระแสตรง 24 โวลต์ R3 และ R4 ทำหน้าที่แบ่งแรงดันเพื่อเป็นแรงดันอ้างอิงให้กับขา 3 และขา 4 ของ IC2 เบอร์ LM324N (สามารถดูข้อมูลเบื้องต้นของไอซีเบอร์ LM324N ได้จากตารางที่ 3.4) ซึ่งจะทำหน้าที่เปรียบเทียบแรงดันกับขา 2 และขา 5 ของ IC2 ตามลำดับ ซึ่งขา 2 และขา 5 ของ IC2 ได้รับแรงดันจากวงจรภาคกำเนิดความถี่ขาที่ 3 ของ IC1 เบอร์ NE555N ซึ่งมีความถี่ขนาด 47เฮิรตซ์และให้แรงดันขนาด 0 และ 5 โวลต์จะส่งผลให้แรงดันเอาต์พุตขาที่ 1 และ 7 ของ IC2 เปิดปิดสลับกัน จะได้ขนาดของแรงดันขณะปิดเป็น 0 โวลต์และขนาดของแรงดันขณะเปิดเป็น 24 โวลต์ เพื่อไปควบคุมการเปิดปิดของทรานซิสเตอร์ Q1-Q4 ต่อไปกรณีที่1 ที่ IC2 เมื่อขาที่ 1 ปิดและขาที่ 7 เปิด จะส่งผลทำให้มีกระแสไหลผ่านทรานซีสเตอร์ Q1 R5 และ Q4 ตามลำดับ จะได้แรงดันด้านบวกขนาด 24 โวลต์กรณีที่2 ที่ IC2 เมื่อขาที่ 1 เปิดและขาที่ 7 ปิด จะส่งผลทำให้มีกระแสไหลผ่านทรานซีสเตอร์ Q3 R5 และ Q2 ตามลำดับ จะได้แรงดันด้านลบขนาด 24 โวลต์จากที่กล่าวมาข้างต้นจะทำให้ได้แรงดันกระแสสลับขนาด 24 โวลต์ 47 เฮิรตซ์หรือประมาณ 50 เฮิรตซ์นั่นเอง

ตารางที่ 3.3 ข้อมูลเบื้องต้นของไอซีเบอร์ NE555N

ขาสัญญาณ	หน้าที่
1	GND
2	Trigger
3	Output
4	Reset
5	Control voltage
6	Threshold
7	Discharge
8	Vcc

สมการสำหรับคำนวณหาความถี่ของวงจรกำเนิดสัญญาณด้วยไอซีเบอร์ NE555N

$$T1 = 0.7 \times (R1+R2) \times C \quad (3.1)$$

$$T2 = 0.7 \times R1 \times C \quad (3.2)$$

$$T = 0.7 \times (R1 + (2 \times R2)) \times C \quad (3.3)$$

$$f = 1.45 / (R1 + (2 \times R2)) \times C \quad (3.4)$$

โดยที่

T1 คือ ช่วงเวลาที่ออน

T2 คือ ช่วงเวลาที่ออฟ

T คือ คาบ

f คือ ความถี่

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 3.4 ข้อมูลเบื้องต้นของไอซีเบอร์ LM324N

ขาสัญญาณ	หน้าที่
1	Out 1
2	Inputs 1
3	Inputs 1
4	Vcc
5	Inputs 2
6	Inputs 2
7	Out 2
8	Out 3
9	Inputs 3
10	Inputs 3
11	Vee
12	Inputs 4
13	Inputs 4
14	Out 4

มหาวิทยาลัยเทคโนโลยีสุรนารี

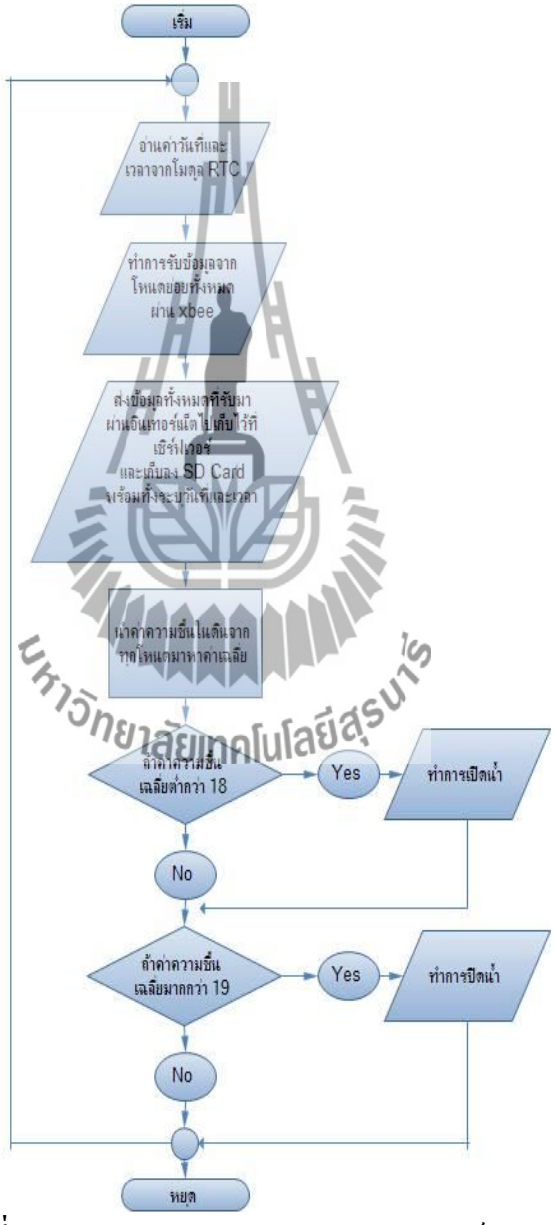
3.4 การออกแบบทางซอฟต์แวร์

การออกแบบทางซอฟต์แวร์นั้นจะแบ่งเป็นสามส่วนคือ ส่วนของโคออร์ดิเนเตอร์ ส่วนของโหนดย่อย (Router) และส่วนของไฟล์บนเซิร์ฟเวอร์

3.4.1 การออกแบบทางซอฟต์แวร์ในส่วนของโคออร์ดิเนเตอร์

3.4.1.1 แผนภาพการทำงานของโปรแกรม

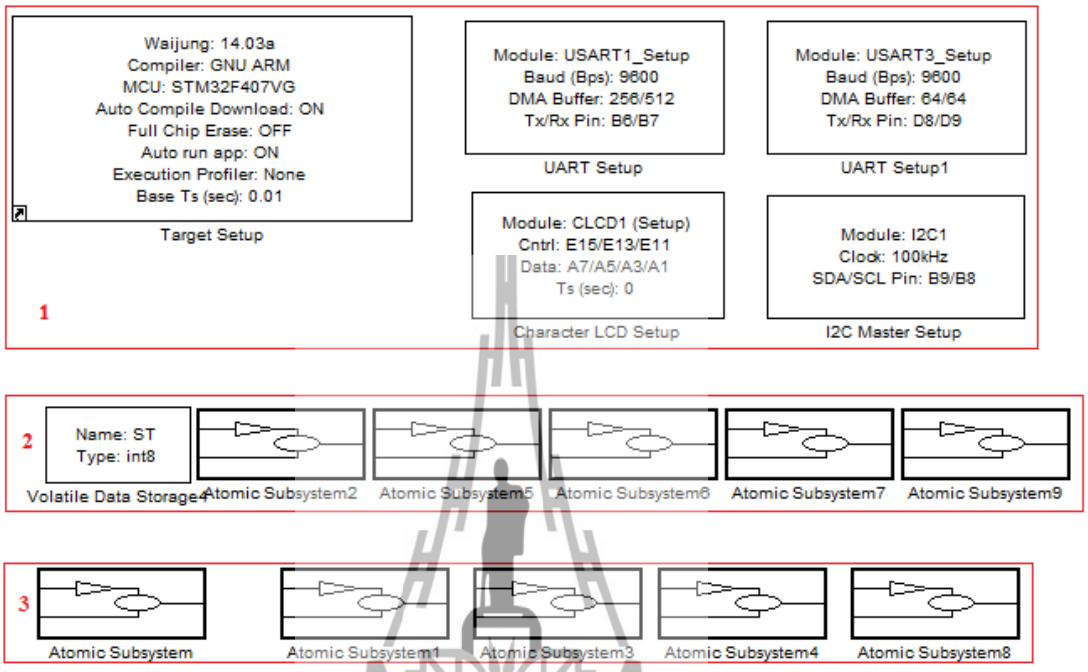
การทำงานของโคออร์ดิเนเตอร์โดยรวมสามารถอธิบายได้ดังรูปที่ 3.9



รูปที่ 3.8 แสดงการทำงานของโปรแกรมโคออร์ดิเนเตอร์

3.4.1.2 การออกแบบโปรแกรมสำหรับโคออร์ดิเนเตอร์

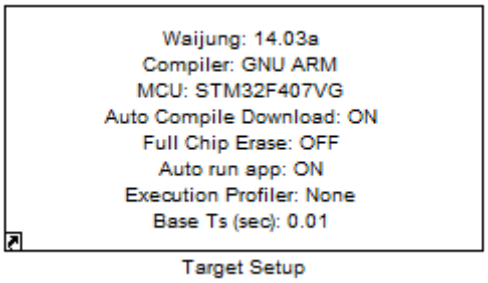
สามารถแบ่งอธิบายโปรแกรมเป็นสามส่วนใหญ่ๆ(เขียนกำกับส่วนด้วยตัวเลขสีแดงและแบ่งส่วนด้วยกรอบสี่เหลี่ยมสีแดง)คือ ส่วนที่ 1 ส่วนของการประกาศบล็อกเริ่มต้นส่วนที่ 2 ส่วนของการประกาศตัวแปร ส่วนที่ 3 ส่วนของการทำงานต่างๆ ดังรูปที่ 3.10



รูปที่ 3.9 รูปของบล็อกทั้งหมดแบ่งเป็นสามส่วน

ส่วนที่ 1 (หมายเลข 1 ของรูปที่ 3.10) ส่วนของการประกาศบล็อกเริ่มต้น

1. ทำการลากบล็อก Target setup ดังรูปที่ 3.10 เพื่อกำหนด Compiler ในโครงงานนี้เลือกใช้ GNU ARM



รูปที่ 3.10 Target setup

2. ทำการลากบล็อก Character LCD setupดังรูปที่ 3.12 (ก) แล้วทำการกำหนดค่าต่างๆ เช่น ขาควบคุม RS RW EN และขาข้อมูล D4 ถึง D7 ดังรูปที่ 3.13 (ทำการเชื่อมต่อขาสัญญาณของจอ LCD เข้ากับขาสัญญาณของ STM32F4Discovery ดังตารางที่ 3.5)

3. ทำการลากบล็อก UART setup ดังรูปที่ 3.12 (ข) แล้วทำการกำหนดค่าต่างๆ เช่น อัตตราการส่งข้อมูลเป็น 9600 บิตต่อวินาที และอื่นๆดังรูปที่ 3.14 (ทำการเชื่อมต่อขาสัญญาณของโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์เข้ากับขาสัญญาณของ STM32F4Discovery ดังตารางที่ 3.6)

4. ทำการลากบล็อก UART setup1 ดังรูปที่ 3.12 (ค) แล้วทำการกำหนดค่าต่างๆดังรูปที่ 3.15 (ทำการเชื่อมต่อขาสัญญาณของ Xbee เข้ากับขาสัญญาณของ STM32F4Discovery ดังตารางที่ 3.7)

5. ทำการลากบล็อก I2C master setup ดังรูปที่ 3.12 (ง) แล้วทำการกำหนดค่าต่างๆดังรูปที่ 3.16 (ทำการเชื่อมต่อขาสัญญาณของโมดูล aMG high precision real time clock เข้ากับขาสัญญาณของ STM32F4Discovery ดังตารางที่ 3.8)

ตารางที่ 3.5ทำการเชื่อมต่อขาสัญญาณของจอ LCD เข้ากับขาสัญญาณของ STM32F4Discovery

ขาสัญญาณของจอ LCD	**ขาสัญญาณของ STM32F4Discovery**
RS	PE15
RW	PE13
EN	PE11
D4	PA7
D5	PA5
D6	PA3
D7	PA1

ตารางที่ 3.6 ทำการเชื่อมต่อขาสัญญาณของโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์เข้ากับขาสัญญาณของ STM32F4Discovery

ขาสัญญาณของโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์	**ขาสัญญาณของ STM32F4Discovery**
Tx	Rx
Rx	Tx
RTS	3V
GND	GND

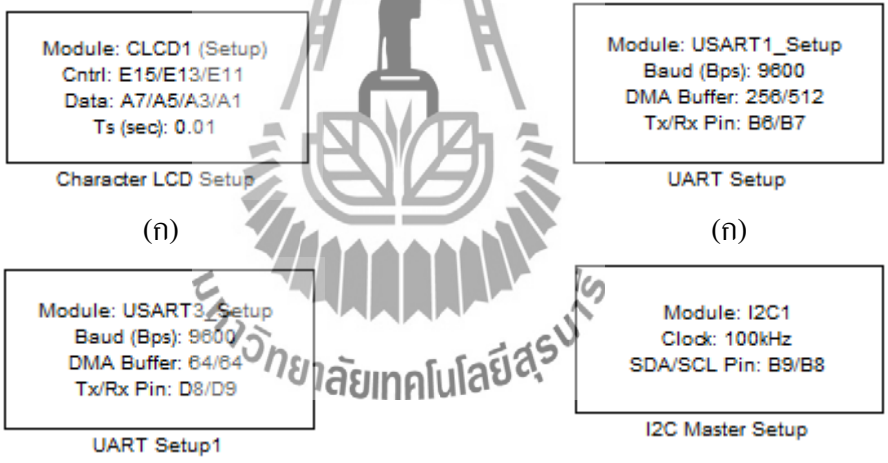
มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 3.7 ทำการเชื่อมต่อขาสัญญาณของ Xbee เข้ากับขาสัญญาณของ STM32F4Discovery

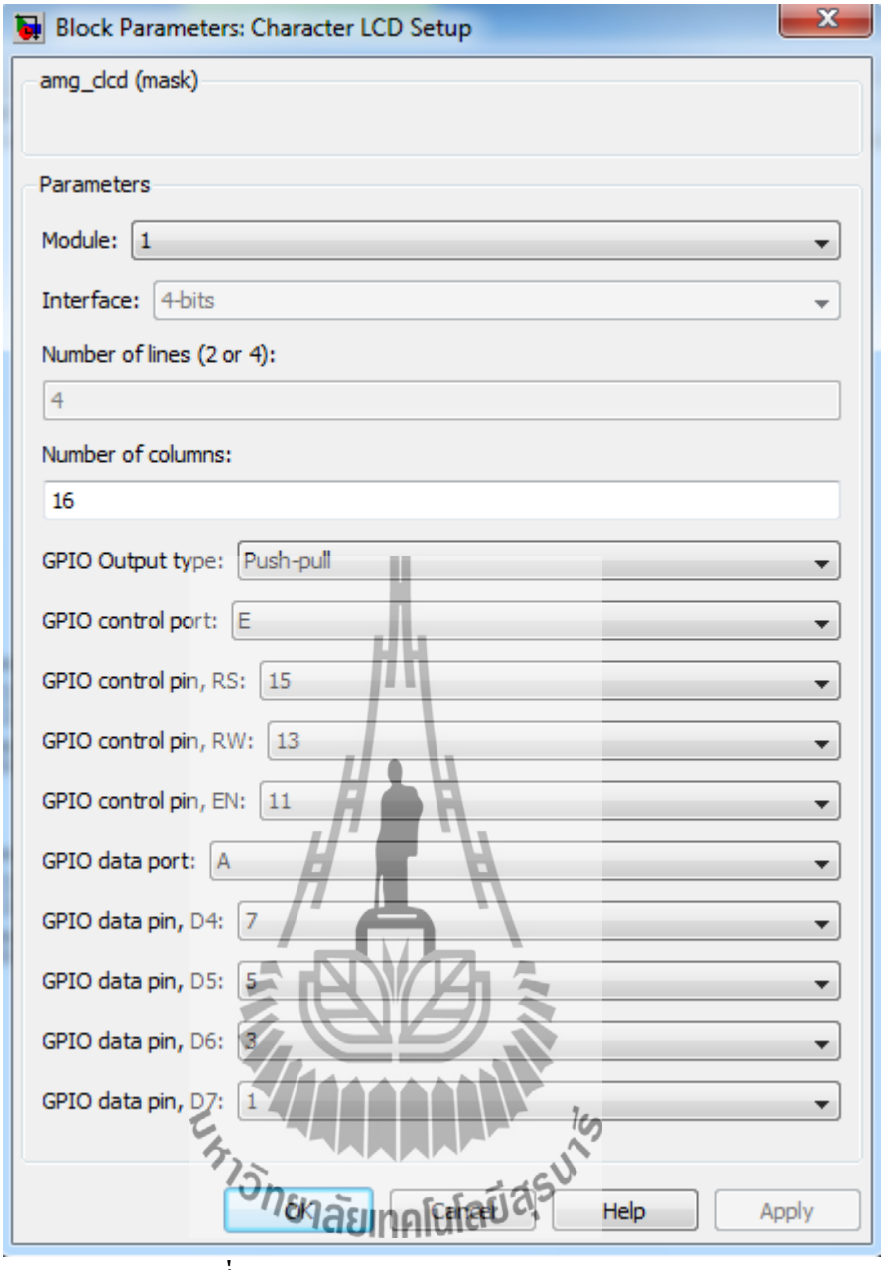
ขาสัญญาณของ Xbee	ขาสัญญาณของ STM32F4Discovery
DOUT	Rx
DIN	Tx
GND	GND

ตารางที่ 3.8 ทำการเชื่อมต่อขาสัญญาณของโมดูล aMG high precision real time clock เข้ากับขาสัญญาณของ STM32F4Discovery

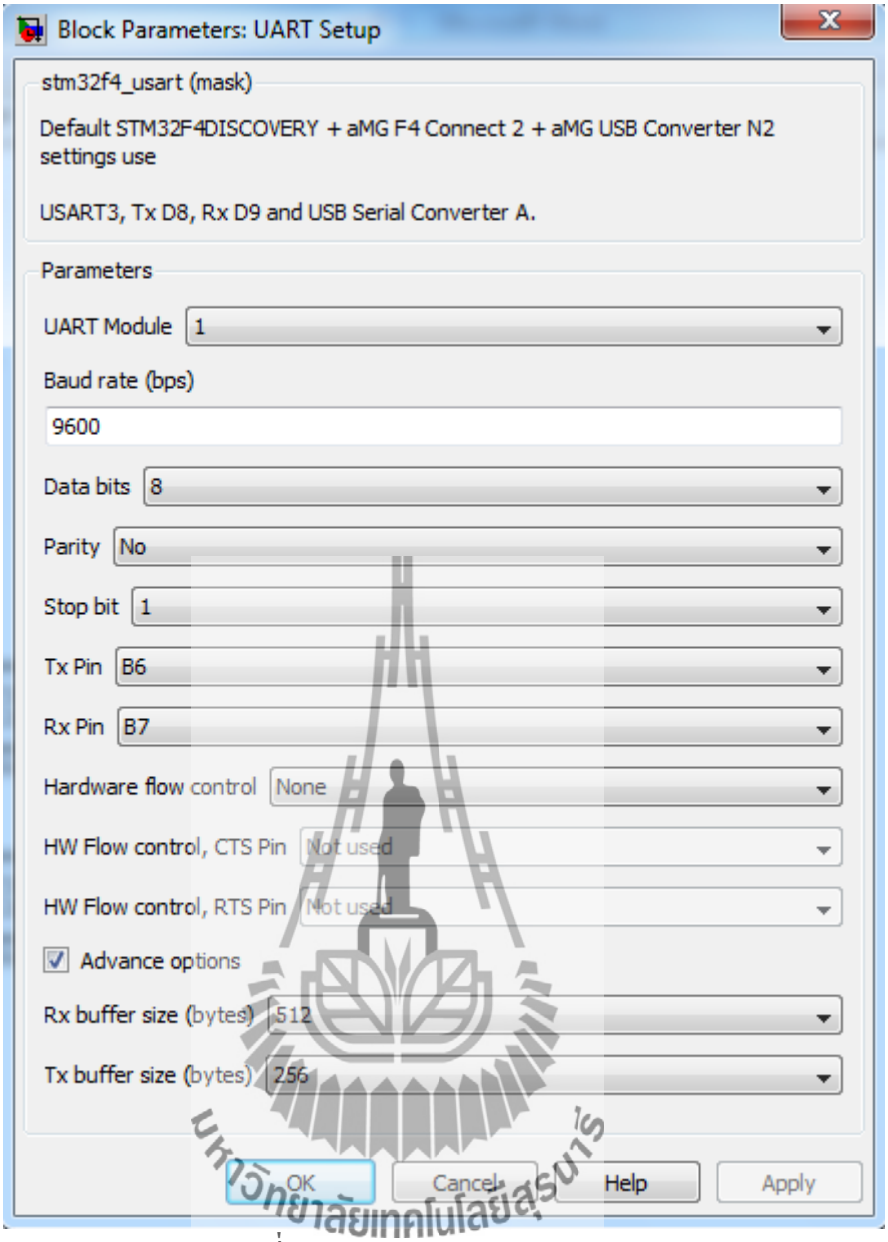
ขาสัญญาณของโมดูลaMG high precision real time clock	ขาสัญญาณของ STM32F4Discovery
SCL	PB8
SDA	PB9
GND	GND



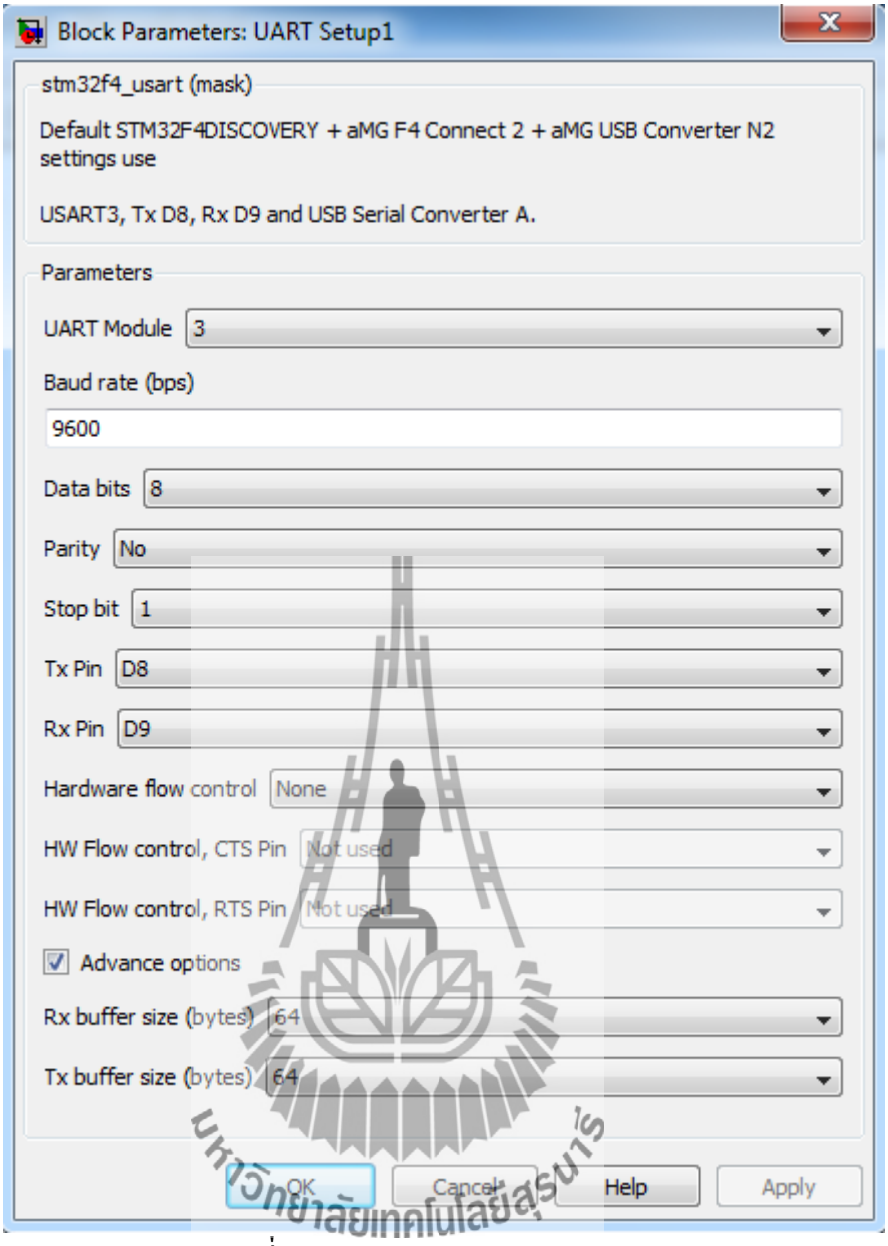
รูปที่ 3.11 (ก)Character LCD setup(ข)UART setup(ค) UART setup1 และ (ง) I2C master setup



รูปที่ 3.12 การกำหนดค่า Character LCD setup

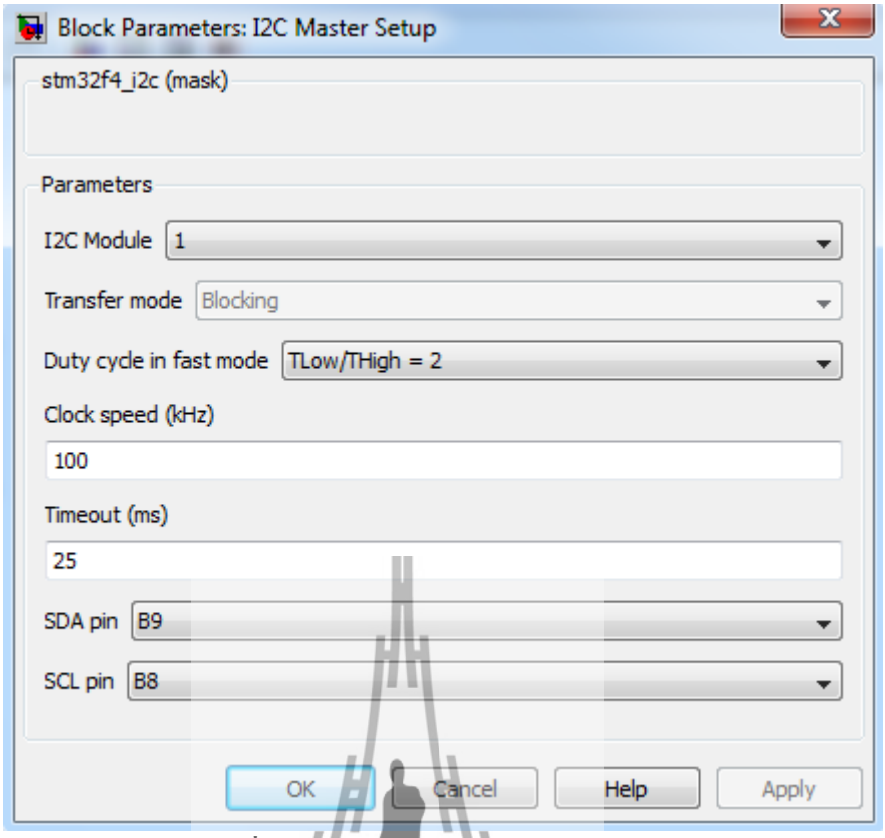


รูปที่ 3.13 การกำหนดค่า UART setup



รูปที่ 3.14 การกำหนดค่า UART setup1

มหาวิทยาลัยเทคโนโลยีสุรนารี



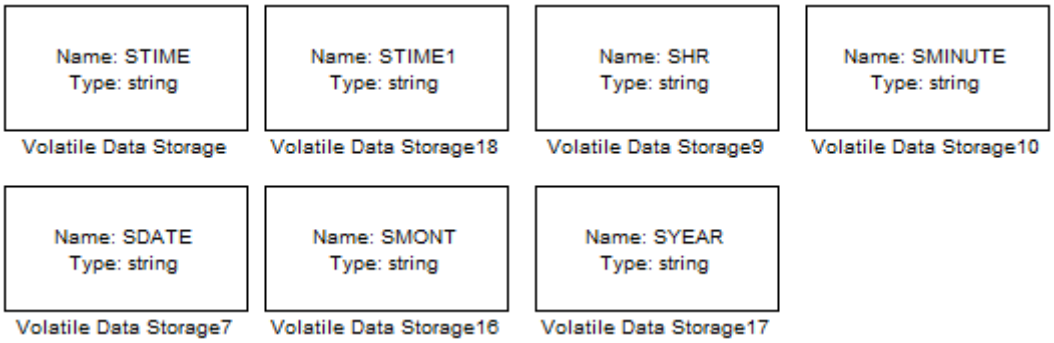
รูปที่ 3.15 การกำหนดค่า I2C master setup

ส่วนที่ 2 (หมายเลข 2 ของรูปที่ 3.10) ส่วนของการประกาศตัวแปร

ในส่วนนี้จะแบ่งอธิบายเป็น 6 ส่วนย่อยซึ่งเขียนกำกับด้วยตัวเลขสีแดงดังรูปที่ 3.17

รูปที่ 3.16 รูปแสดงส่วนของการประกาศตัวแปรซึ่งถูกแบ่งเป็น 6 ส่วนย่อย

1. ทำการลากบล็อก Volatile data storageดังรูปที่ 3.17 หมายเลข 1 พร้อมทั้งตั้งชื่อตัวแปรเป็น ST กำหนดประเภทของข้อมูลเป็น int8 และกำหนดค่าเริ่มต้นเป็น 0
2. ทำการลากบล็อก Atomic subsystem ดังรูปที่ 3.17หมายเลข 2 จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อกสำหรับเป็นบัฟเฟอร์เก็บค่าของเวลาดังรูปที่ 3.18 (ค่าพารามิเตอร์ต่างๆของแต่ละบล็อกสามารถดูได้จากตารางที่ 3.9)

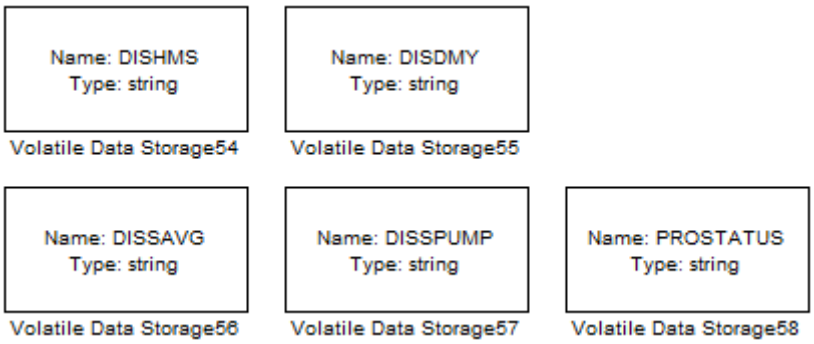


รูปที่ 3.17บล็อกสำหรับเก็บค่าของเวลาทั้งหมด

ตารางที่ 3.9ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์เก็บค่าของเวลา

Storage Name	Storage Type	Initial value
STIME	string	'Initial string'
STIME1	string	'Initial string'
SHR	string	'0'
SMINUTE	string	'0'
SDATE	string	'0'
SMONT	string	'0'
SYEAR	string	'0'

3. ทำการ ลากบล็อก Atomic subsystem ดังรูปที่ 3.17หมายเลข 3 จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อกสำหรับเป็นบัฟเฟอร์การแสดงผลที่จอ LCD ดังรูปที่ 3.19 (ค่าพารามิเตอร์ต่างๆของแต่ละบล็อกสามารถดูได้จากตารางที่ 3.10)

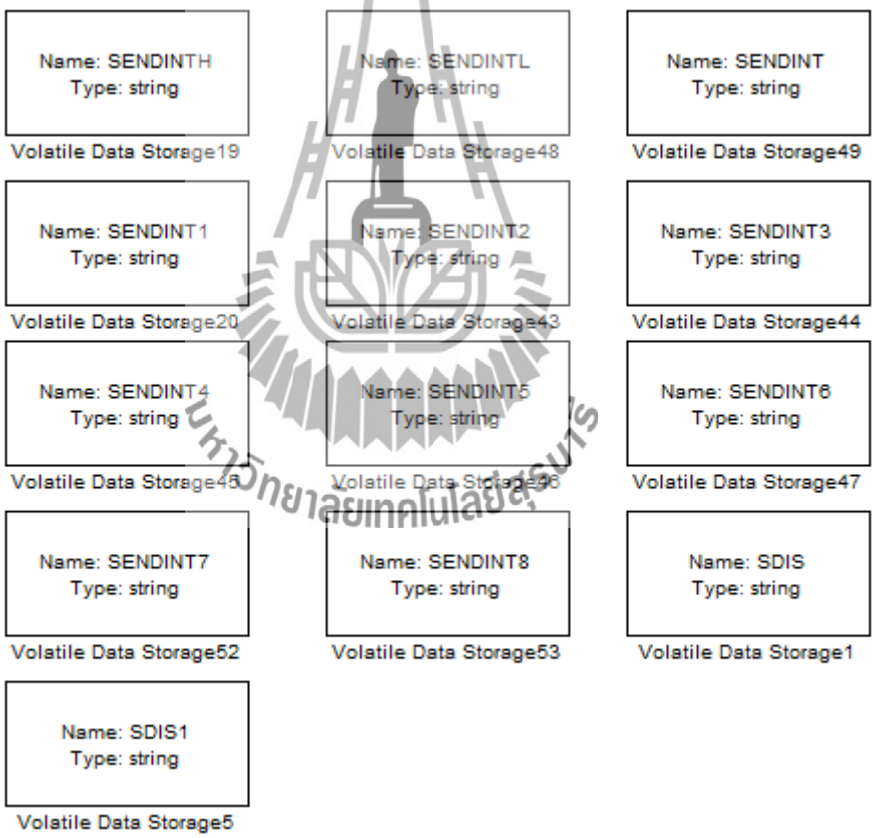


รูปที่ 3.18 บล็อกสำหรับการแสดงผลที่จอ LCD ทั้งหมด

ตารางที่ 3.10ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์การแสดงผลที่จอ LCD

Storage Name	Storage Type	Initial value
DISHMS	string	'Initial string'
DISDMY	string	'Initial string'
DISSAVG	string	'Initial string'
DISSPUMP	string	'Initial string'
PROSTATUS	string	'N'

4. ทำการ ลากบล็อก Atomic subsystem ดังรูปที่ 3.17หมายเลข 4 จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อกสำหรับเป็นบัฟเฟอร์เก็บข้อมูลส่งขึ้นเซิร์ฟเวอร์ดังรูปที่ 3.20 (ค่าพารามิเตอร์ต่างๆ ของแต่ละบล็อกสามารถดูได้จากตารางที่ 3.11)



รูปที่ 3.19 บล็อกสำหรับเป็นบัฟเฟอร์เก็บข้อมูลส่งขึ้นเซิร์ฟเวอร์

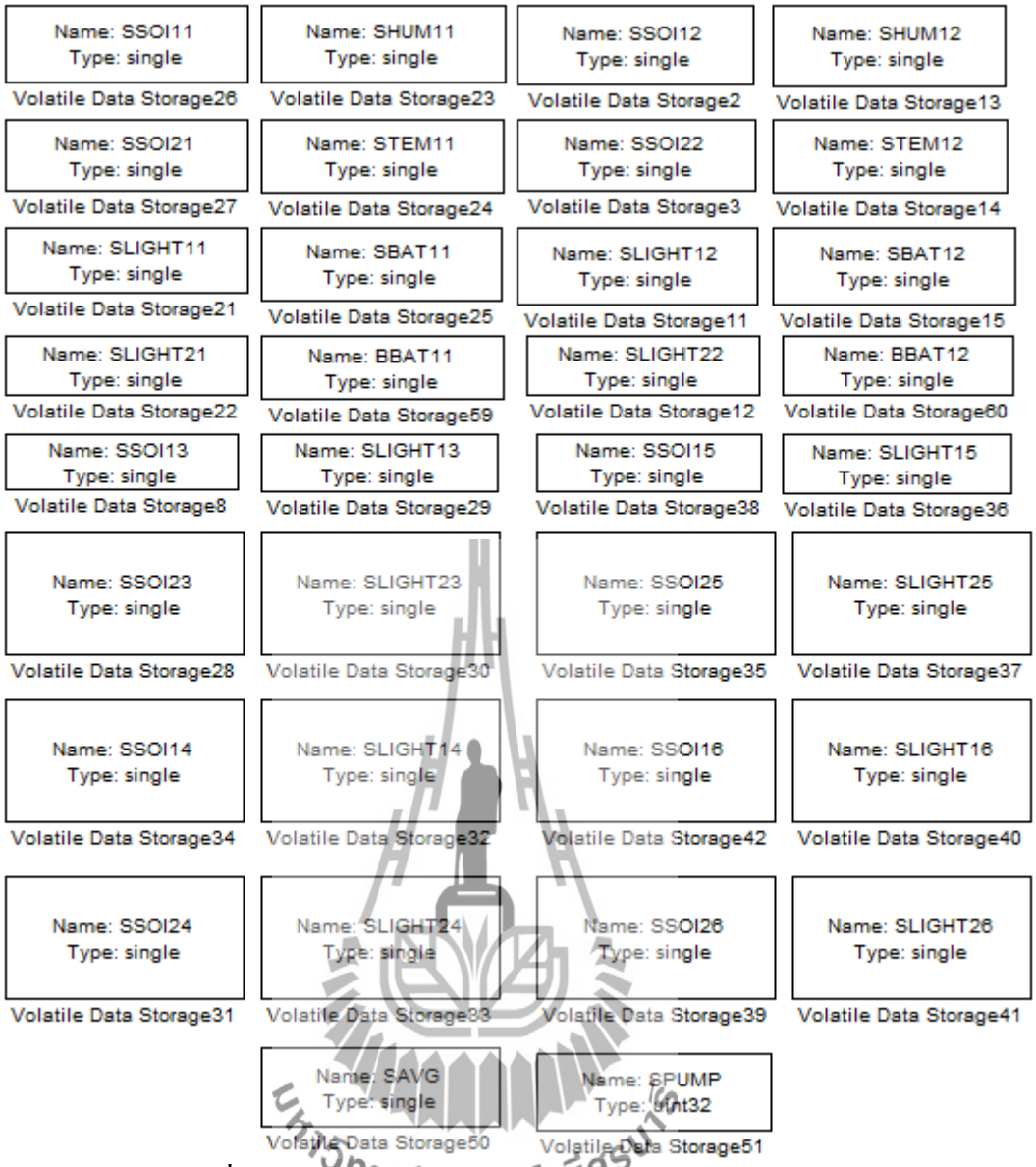
มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 3.11ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์เก็บข้อมูลส่งขึ้นเซิร์ฟเวอร์

Storage Name	Storage Type	Initial value
SENDINTH	string	'GET /142001/CASAVA1/index.php?'
SENDINTL	string	' HTTP/1.1\r\nhost: 203.158.7.36\r\n\r\n\x1A'
SENDINT	string	'INTERNET'
SENDINT1	string	'INTERNET'
SENDINT2	string	'INTERNET'
SENDINT3	string	'INTERNET'
SENDINT4	string	'INTERNET'
SENDINT5	string	'INTERNET'
SENDINT6	string	'INTERNET'
SENDINT7	string	'INTERNET'
SENDINT8	string	'INTERNET'
SDIS	string	'COMMAND START'
SDIS1	string	'GSM BACK'

5. ทำการ ลากบล็อก Atomic subsystem ดังรูปที่ 3.17หมายเลข 5 จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อกสำหรับเป็นบัฟเฟอร์รับข้อมูลจาก โหนดย่อยดังรูปที่ 3.21 (ค่าพารามิเตอร์ต่างๆของแต่ละบล็อกสามารถดูได้จากตารางที่ 3.12)

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.20 บล็อกสำหรับเป็นบัฟเฟอร์เก็บข้อมูลจากโหนดย่อย

ตารางที่ 3.12ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์รับข้อมูลจากโหนดย่อย

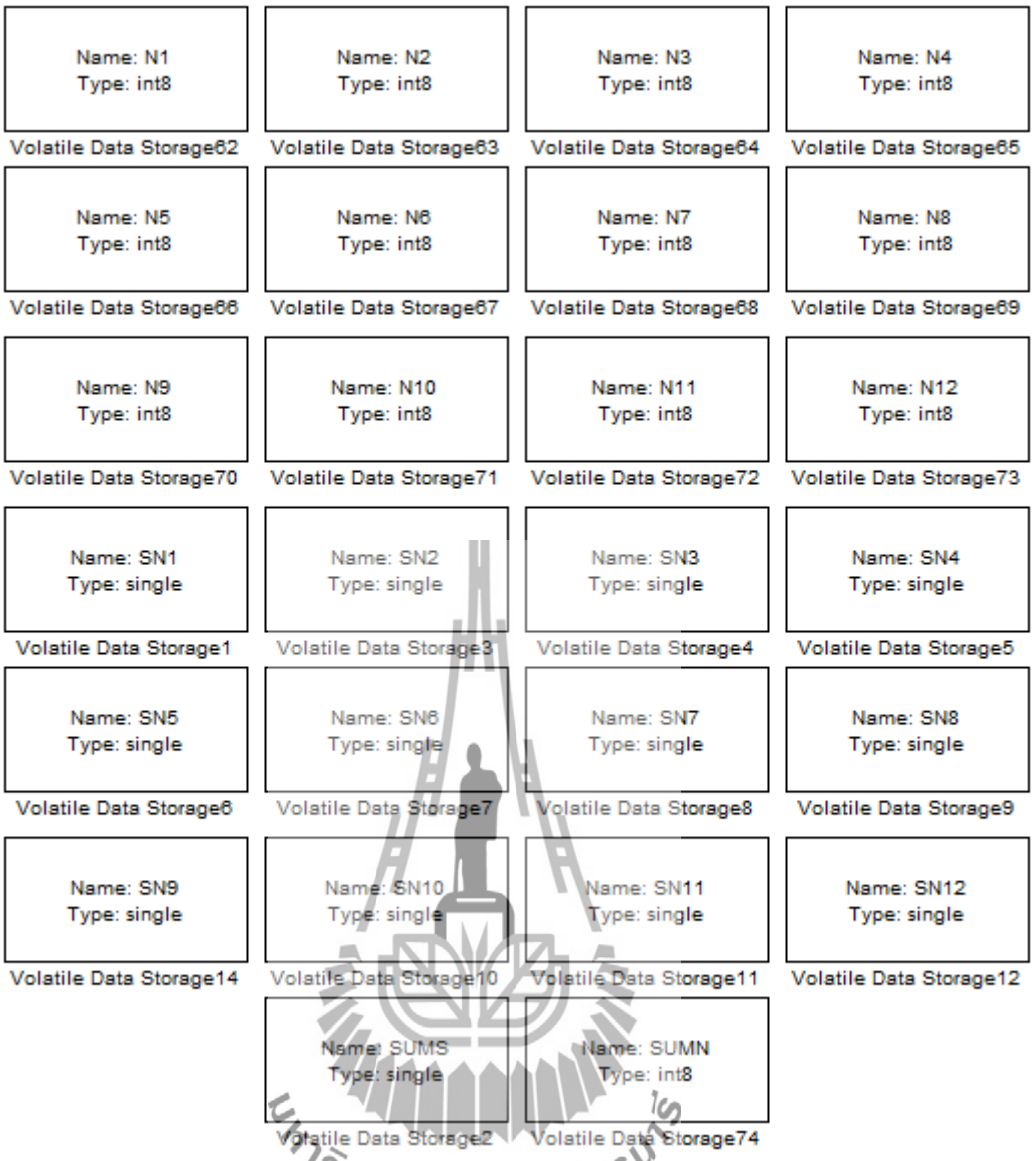
Storage Name	Storage Type	Initial value
SSOI11	single	20
SSOI21	single	20
SLIGHT11	single	20
SLIGHT21	single	20
SHUM11	single	20
STEM11	single	20
SBAT11	single	20

ตารางที่ 3.12ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์รับข้อมูลจากโหนดย่อย(ต่อ)

Storage Name	Storage Type	Initial value
SSOI12	single	20
SSOI22	single	20
SLIGHT12	single	20
SLIGHT22	single	20
SHUM12	single	20
STEM12	single	20
SBAT12	single	20
SSOI13	single	20
SSOI23	single	20
SLIGHT13	single	20
SLIGHT23	single	20
SSOI14	single	20
SSOI24	single	20
SLIGHT14	single	20
SLIGHT24	single	20
SSOI15	single	20
SSOI25	single	20
SLIGHT15	single	20
SLIGHT25	single	20
SSOI16	single	20
SSOI26	single	20
SLIGHT16	single	20
SLIGHT26	single	20
SAVG	single	20
SPUMP	uint32	0

มหาวิทยาลัยเทคโนโลยีสุรนารี

6. ทำการ ลากบล็อก Atomic subsystem ดังรูปที่ 3.17หมายเลข 6 จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อกสำหรับเป็นบัฟเฟอร์ในการหาค่าเฉลี่ยของเซ็นเซอร์ในดิน 12 ตัวดังรูปที่ 3.22 (ค่าพารามิเตอร์ต่างๆของแต่ละบล็อกสามารถดูได้จากตารางที่ 3.13)



รูปที่ 3.21 บล็อกสำหรับเป็นบัฟเฟอร์ในการหาค่าเฉลี่ยของเซ็นเซอร์ในดิน 12 ตัว

ตารางที่ 3.11ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์ในการหาค่าเฉลี่ยของเซ็นเซอร์ในดิน 12 ตัว

Storage Name	Storage Type	Initial value
N1	int8	1
N2	int8	1
N3	int8	1
N4	int8	1
N5	int8	1
N6	int8	1

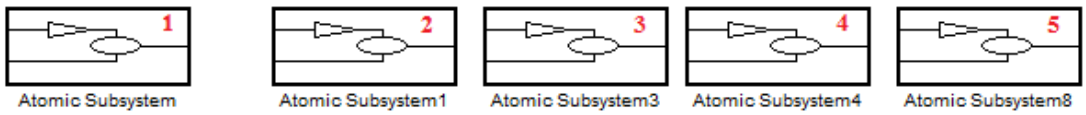
ตารางที่ 3.11ค่าพารามิเตอร์ของบล็อกสำหรับเป็นบัฟเฟอร์ในการหาค่าเฉลี่ยของเซ็นเซอร์ในดิน 12 ตัว(ต่อ)

Storage Name	Storage Type	Initial value
N7	int8	1
N8	int8	1
N9	int8	1
N10	int8	1
N11	int8	1
N12	int8	1
SN1	single	20
SN2	single	20
SN3	single	20
SN4	single	20
SN5	single	20
SN6	single	20
SN7	single	20
SN8	single	20
SN9	single	20
SN10	single	20
SN11	single	20
SN12	single	20
SUMN	int8	1
SUMS	single	240

มหาวิทยาลัยเทคโนโลยีสุรนารี

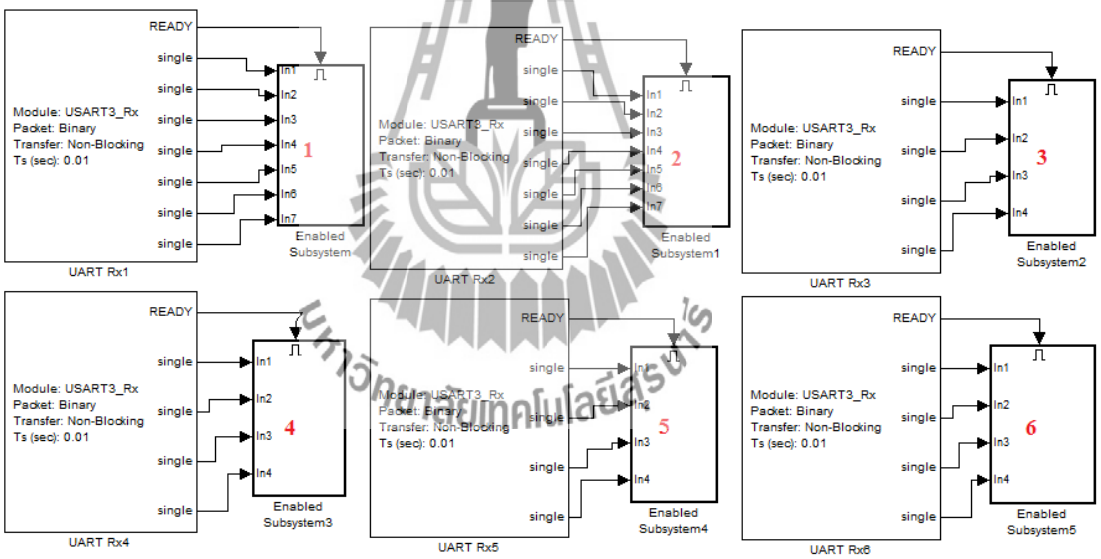
ส่วนที่ 3(หมายเลข 3 ของรูปที่ 3.10) ส่วนของการทำงานต่างๆ

ในส่วนนี้จะแบ่งอธิบายเป็น 5 ส่วนย่อยซึ่งเขียนกำกับด้วยตัวเลขสีแดงดังรูปที่ 3.22

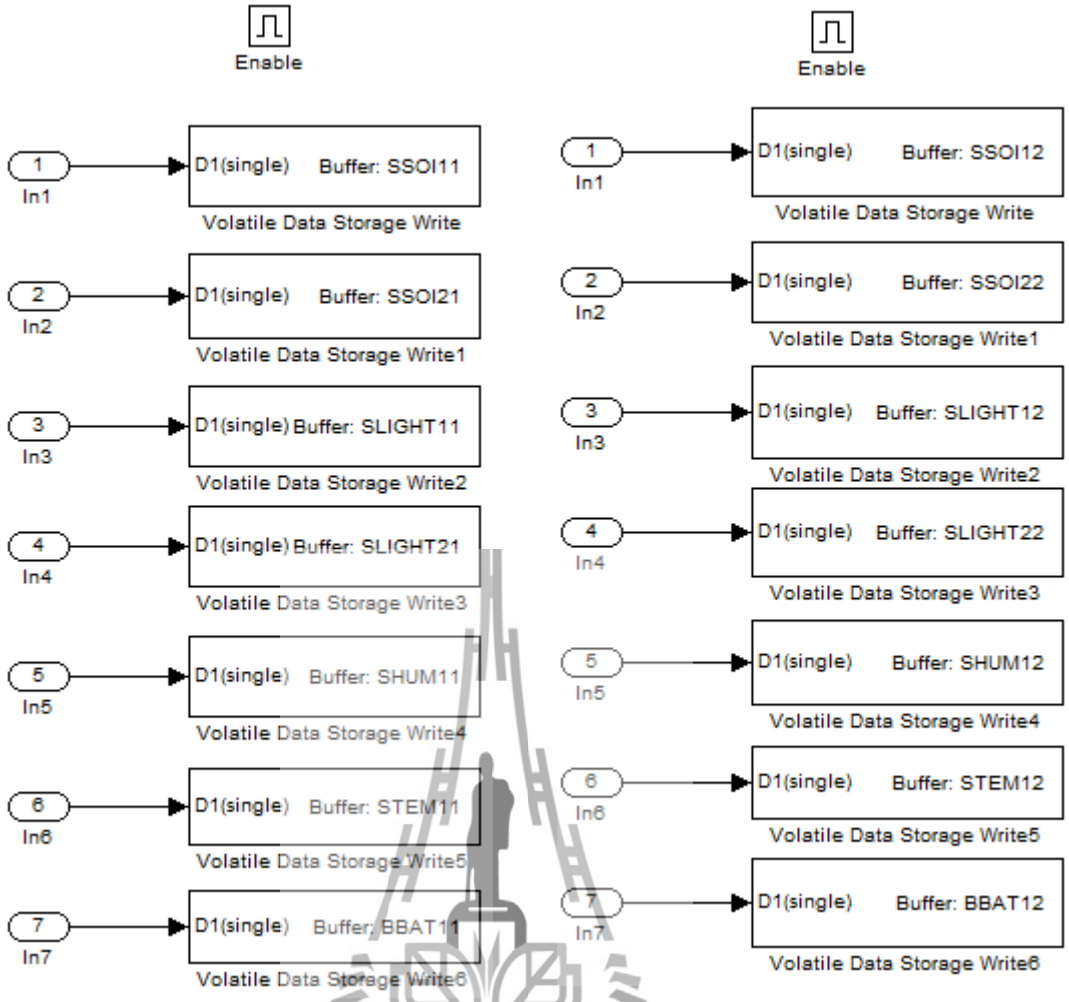


รูปที่ 3.22 รูปแสดงส่วนของการทำงานต่างๆซึ่งถูกแบ่งเป็น 5 ส่วนย่อย

1.ในส่วนของหมายเลข 1 ของรูปที่ 3.22 ทำการ ลากบล็อก Atomic subsystem จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อกสำหรับรับข้อมูลจากโหนดย่อย ดังรูปที่ 3.23 จากนั้นดับเบิ้ลคลิกเข้าไปในบล็อก Atomic subsystem หมายเลข 1 ของรูปที่ 3.23 แล้วทำการลากบล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 1 ลงตัวแปร ดังรูปที่ 3.24(ก)จากนั้นดับเบิ้ลคลิกเข้าไปในบล็อก Atomic subsystem หมายเลข 2 ของรูปที่ 3.23 แล้วทำการลากบล็อก สำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 2 ลงตัวแปร ดังรูปที่ 3.24(ข) จากนั้นดับเบิ้ลคลิกเข้าไปในบล็อก Atomic subsystem หมายเลข 3 ของรูปที่ 3.23 แล้วทำการลากบล็อก สำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 3 ลงตัวแปรดังรูปที่ 3.25(ก) จากนั้นดับเบิ้ลคลิกเข้าไปในบล็อก Atomic subsystem หมายเลข 4 ของรูปที่ 3.23 แล้วทำการลากบล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 4 ลงตัวแปร ดังรูปที่ 3.25(ข) จากนั้นดับเบิ้ลคลิกเข้าไปในบล็อก Atomic subsystem หมายเลข 5 ของรูปที่ 3.23 แล้วทำการลากบล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 5 ลงตัวแปร ดังรูปที่ 3.26(ก) จากนั้นดับเบิ้ลคลิกเข้าไปในบล็อก Atomic subsystem หมายเลข 6 ของรูปที่ 3.23 แล้วทำการลากบล็อก สำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 6 ลงตัวแปรดังรูปที่ 3.26(ข)

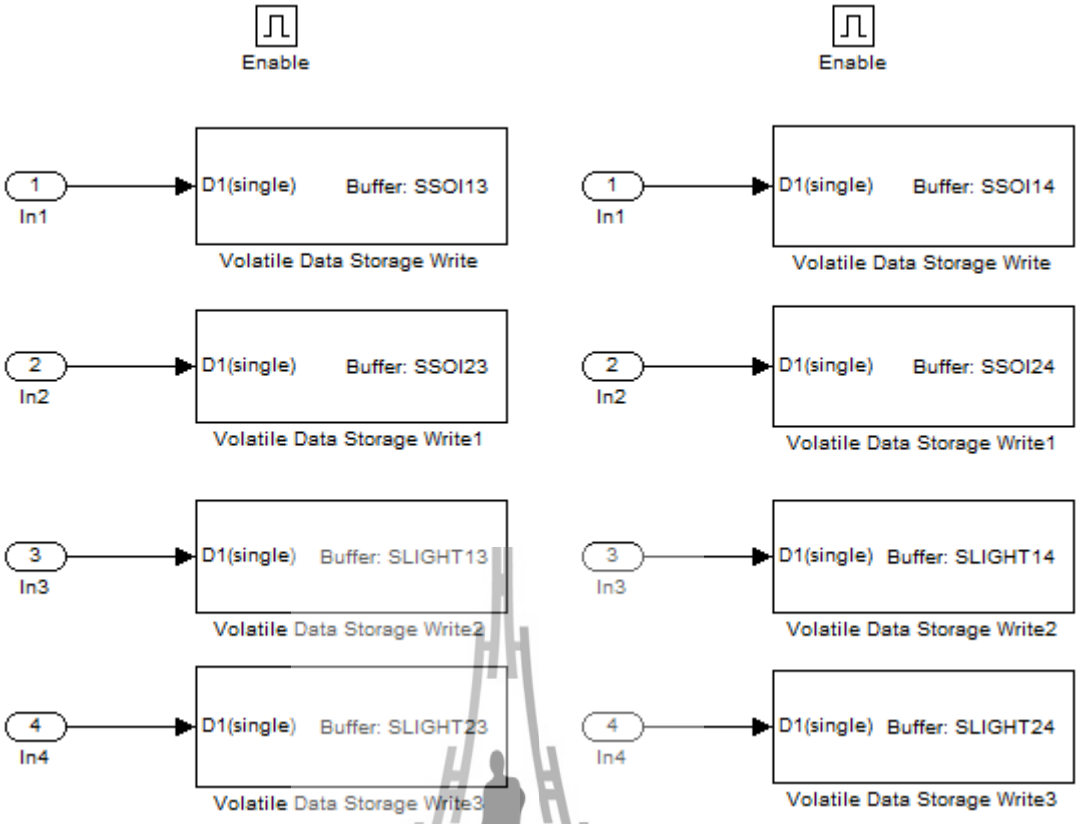


รูปที่ 3.23บล็อกสำหรับรับข้อมูลจากโหนดย่อย



รูปที่ 3.24(ก) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 1 (ข) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 2

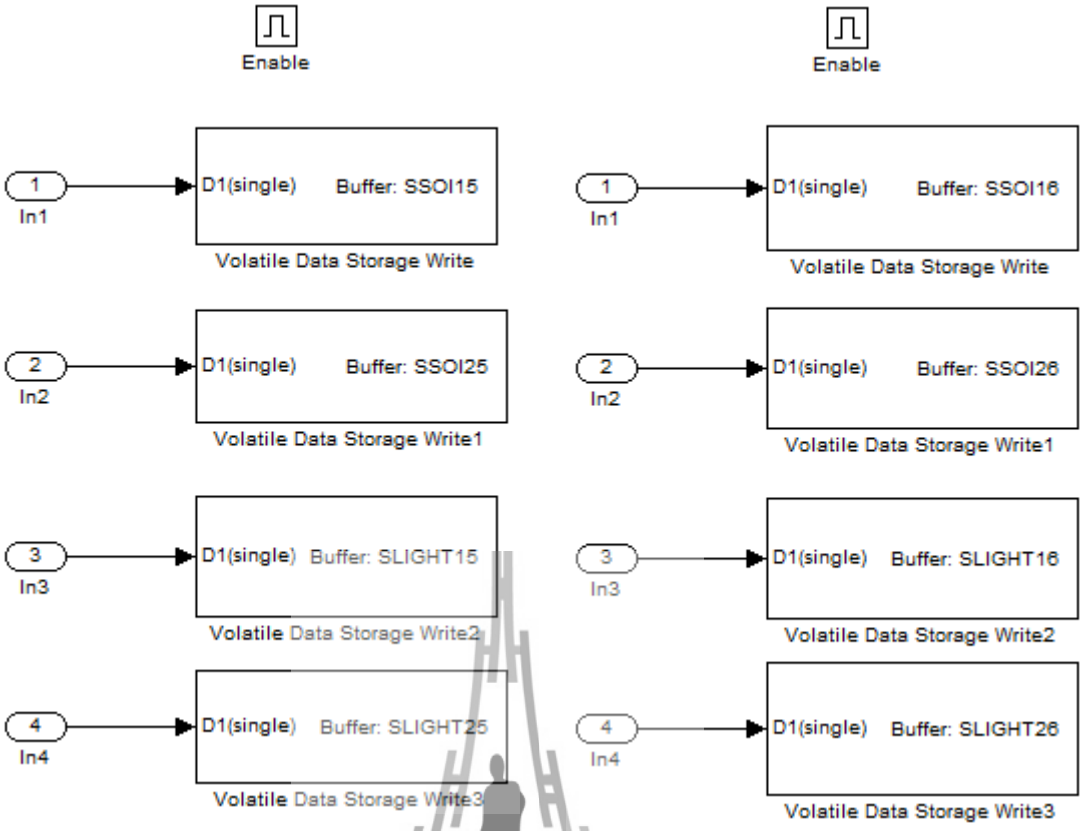
มหาวิทยาลัยเทคโนโลยีสุรนารี



(ก) (ข)

รูปที่ 3.25(ก) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 3 (ข) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 4

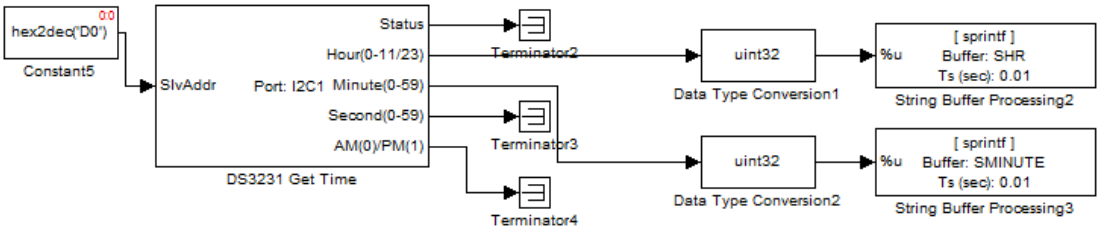
มหาวิทยาลัยเทคโนโลยีสุรนารี



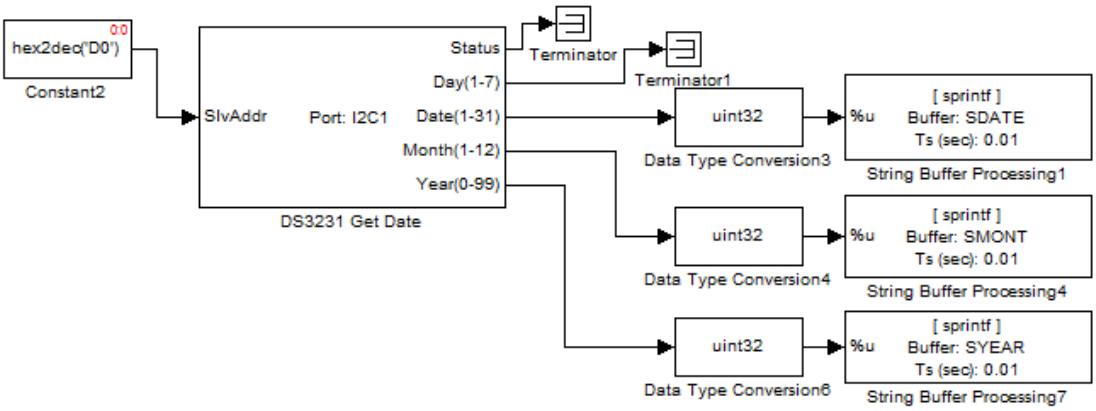
(ก) (ข)

รูปที่ 3.26(ก) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 5 (ข) บล็อกสำหรับเก็บค่าที่รับมาจากโหนดย่อยโหนดที่ 6

2. ในส่วนของหมายเลข 2 ของรูปที่ 3.22 ทำการ ลากบล็อก Atomic subsystem จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อกสำหรับอ่านค่าเวลาชั่วโมงและนาที พร้อมทั้งเก็บลงตัวแปรดังรูปที่ 3.27 จากนั้นทำการลากบล็อกสำหรับอ่านค่าวันเดือนปี พร้อมทั้งเก็บลงตัวแปรดังรูปที่ 3.28



รูปที่ 3.27บล็อกสำหรับอ่านค่าเวลาชั่วโมงและนาที

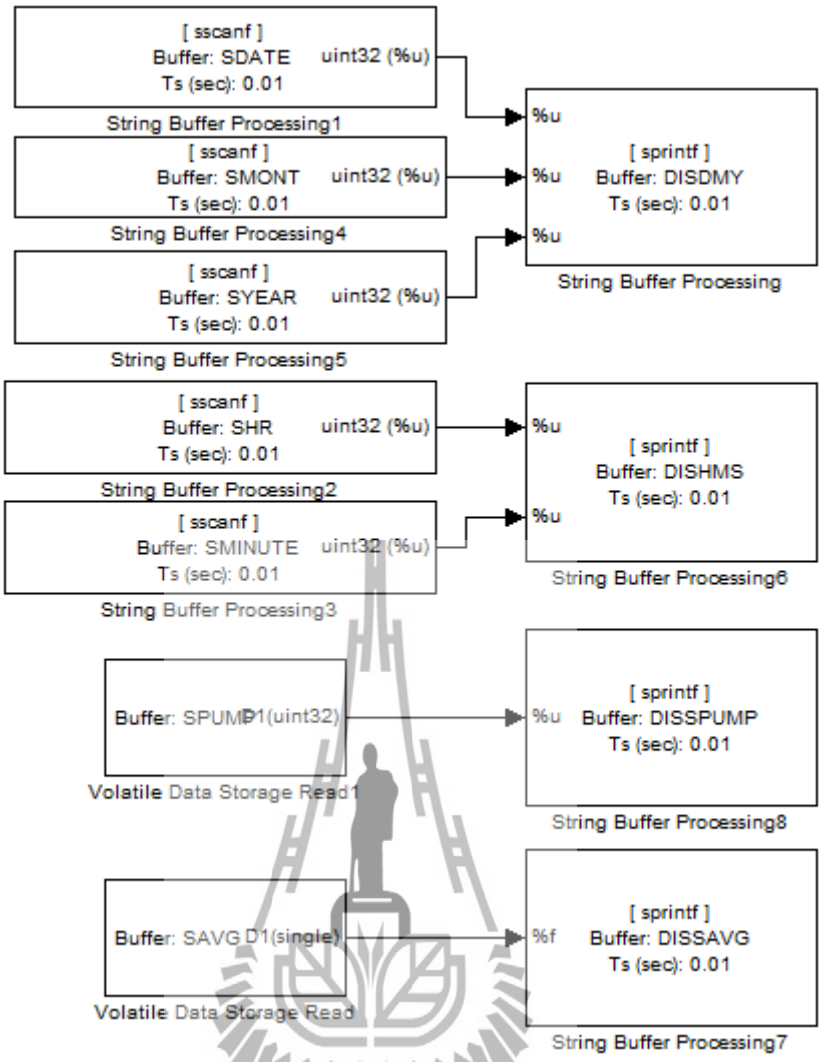


รูปที่ 3.28บล็อกสำหรับอ่านค่าวันเดือนปี

3. ในส่วนของหมายเลข 3 ของรูปที่ 3.22 ทำการ ลากบล็อก Atomic subsystem จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อก Atomic subsystem อีก 2 บล็อกดังรูปที่ 3.29 จากนั้นดับเบิ้ลคลิกเข้าไปใน Atomic subsystem หมายเลข 1 ของรูปที่ 3.29 แล้วทำการลากบล็อกสำหรับ จัดการกับบัฟเฟอร์แสดงผลที่จอ LCDดังรูปที่ 3.30จากนั้นดับเบิ้ลคลิกเข้าไปใน Atomic subsystem หมายเลข 2 ของรูปที่ 3.29 แล้วทำการลากบล็อกสำหรับการแสดงผลที่จอ LCDดังรูปที่ 3.31

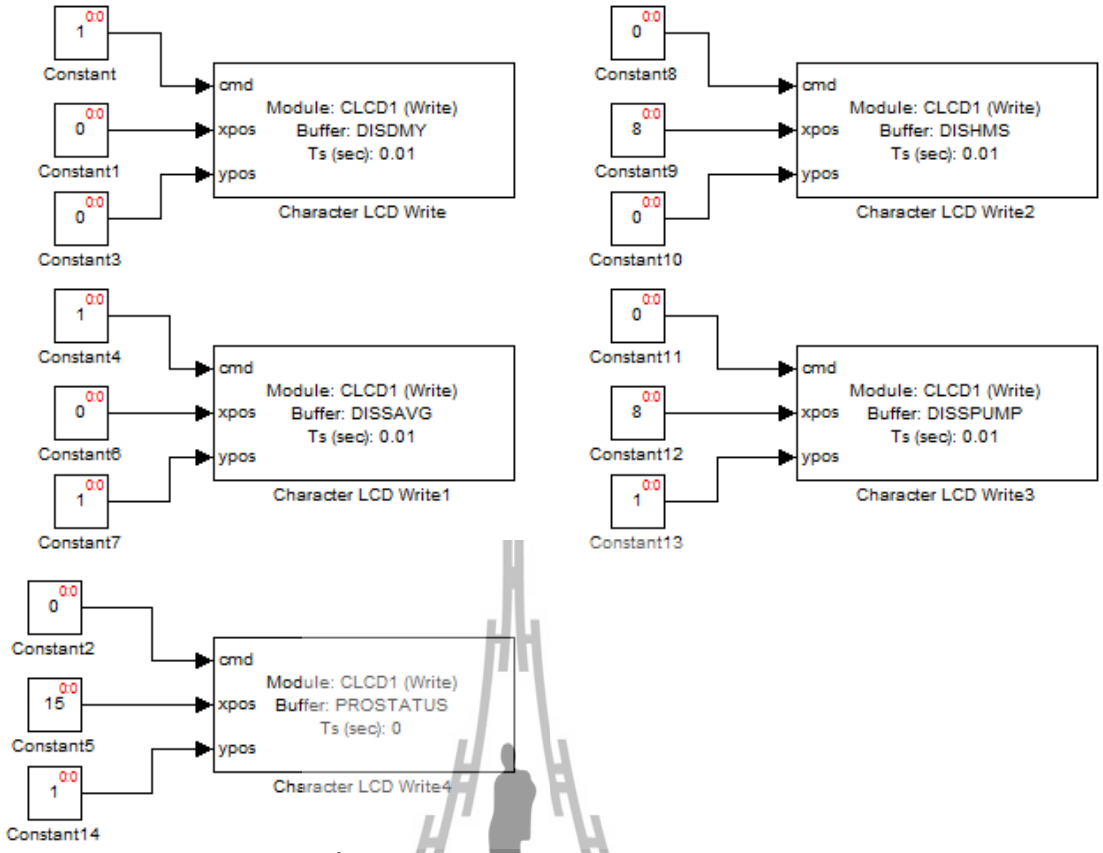
รูปที่ 3.29 บล็อก Atomic subsystem

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.30 บล็อกสำหรับจัดการกับบัฟเฟอร์แสดงผลที่จอ LCD

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.31 บล็อกสำหรับการแสดงผลที่จอ LCD

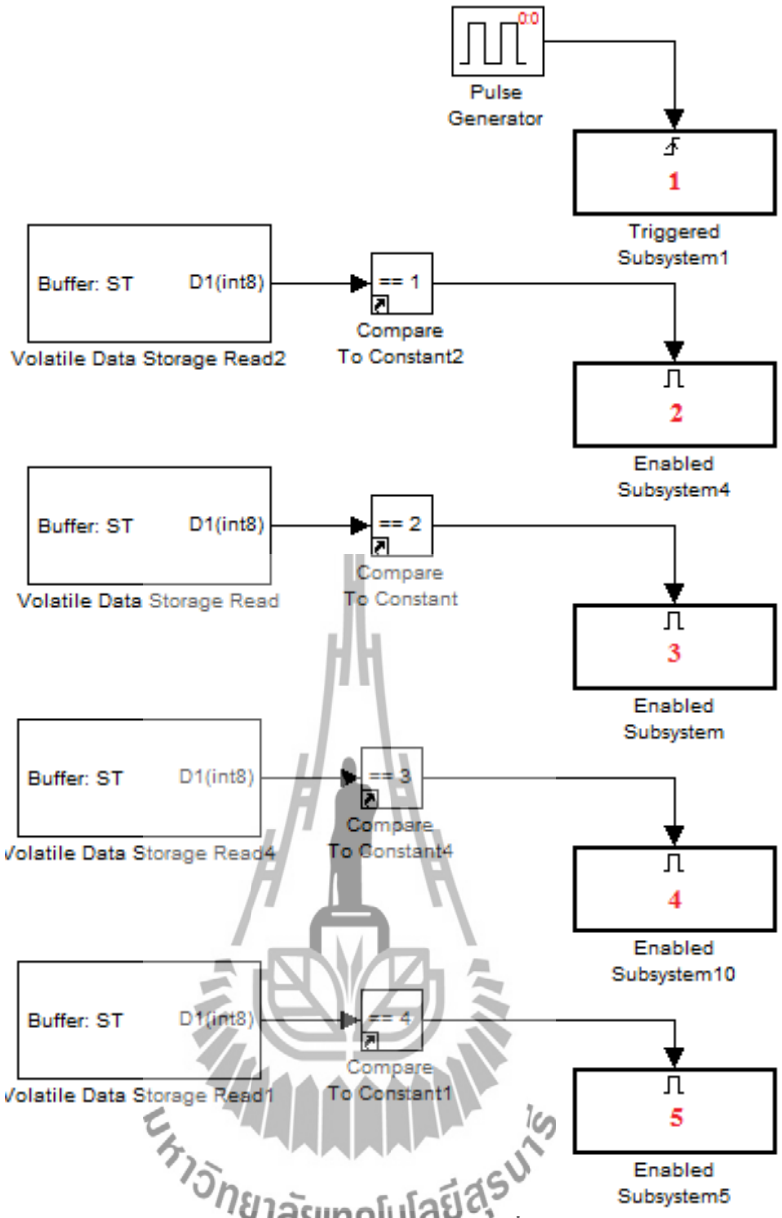
4.ในส่วนของหมายเลข 4 ของรูปที่ 3.22 ทำการ ลากบล็อก Atomic subsystem จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อกสำหรับควบคุมลำดับในการหาค่าเฉลี่ยและการเก็บข้อมูลลงบัฟเฟอร์ดังรูปที่ 3.32 จากนั้นดับเบิ้ลคลิกเข้าไปใน Triggered subsystemหมายเลข 1 ของรูปที่ 3.32 แล้วทำการลากบล็อกที่จะสั่งให้บล็อก Enabled subsystem หมายเลข 2 ของรูปที่ 3.32 เริ่มทำงาน ดังรูปที่ 3.33 จากนั้นดับเบิ้ลคลิกเข้าไปใน Enabled subsystemหมายเลข 2 ของรูปที่ 3.32 แล้วทำการลากบล็อก Atomic subsystemอีก 3 บล็อกพร้อมทั้งกำหนดค่า Priority ของทั้ง 3 บล็อกให้ตรงกับตัวเลขสีแดงที่เขียนกำกับไว้ ดังรูปที่ 3.34จากนั้นดับเบิ้ลคลิกเข้าไปใน Atomic subsystemหมายเลข 1 ของรูปที่ 3.34 แล้วทำการลากบล็อกสำหรับจัดการค่าของเซ็นเซอร์ที่ผิดเพี้ยนดังรูปที่ 3.35จากนั้นดับเบิ้ลคลิกเข้าไปใน Atomic subsystemหมายเลข 1 ถึง 12 ของรูปที่ 3.35 แล้วทำการลากบล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนลงบัฟเฟอร์ดังรูปที่ 3.36 ถึงรูปที่ 3.36 ตามลำดับจากนั้นดับเบิ้ลคลิกเข้าไปใน Atomic subsystemหมายเลข 2 ของรูปที่ 3.34 แล้วทำการลากบล็อกสำหรับหาผลรวมของค่าความชื้นในดินและจำนวนเซ็นเซอร์ทั้งหมดที่ใช้ได้ดังรูปที่ 3.48และ รูปที่ 3.49 จากนั้นดับเบิ้ลคลิกเข้าไปใน Atomic subsystemหมายเลข 3 ของรูปที่ 3.34 แล้วทำการลากบล็อกที่จะสั่งให้บล็อก Enabled subsystem หมายเลข 3 ของรูปที่ 3.32 เริ่มทำงานดังรูปที่ 3.50

5. ดับเบิ้ลคลิกเข้าไปใน Enabled subsystemหมายเลข 3 ของรูปที่ 3.32 แล้วทำการลากบล็อกสำหรับควบคุมการเปิดปิดของปั้มสูบน้ำดังรูปที่ 3.51 จากนั้นดับเบิ้ลคลิกเข้าไปใน Enabled subsystem หมายเลข 1 ของรูปที่ 3.51 แล้วทำการลากบล็อกสำหรับสั่งให้ปั้มสูบน้ำหยุดทำงานดังรูปที่ 3.52 จากนั้นดับเบิ้ลคลิกเข้าไปใน Enabled subsystem หมายเลข 2 ของรูปที่ 3.51 แล้วทำการลากบล็อกสำหรับสั่งให้ปั้มสูบน้ำทำงานดังรูปที่ 3.53

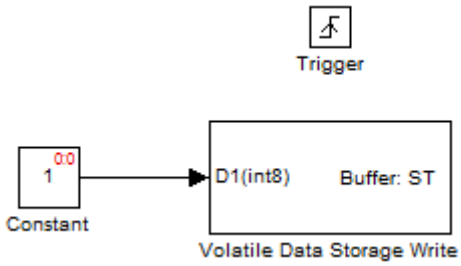
6. ดับเบิ้ลคลิกเข้าไปใน Enabled subsystemหมายเลข 4 ของรูปที่ 3.32 แล้วทำการลากบล็อกสำหรับเรียงลำดับการเก็บค่าลงบัฟเฟอร์และเขียน ลงเอสดีการ์ด พร้อมทั้งกำหนดค่า Priority ของทั้ง 20 บล็อกให้ตรงกับตัวเลขสีแดงที่เขียนกำกับไว้ ดังรูปที่ 3.54 จากนั้นดับเบิ้ลคลิกเข้าไปในบล็อก Atomic subsystemหมายเลข 2,4,6,8,10,12,14,16,18 และ 20 ของรูปที่ 3.54 แล้วทำการลากบล็อกดังรูปที่ 3.55 ถึงรูปที่ 3.64 ตามลำดับ (ค่าพารามิเตอร์ต่างๆของบล็อก Data logger และบล็อก String buffer processingของรูปที่ 3.55 ถึงรูปที่ 3.61 สามารถดูได้จากตารางที่ 3.12 และ 3.13)

7. ดับเบิ้ลคลิกเข้าไปใน Enabled subsystemหมายเลข 5 ของรูปที่ 3.32 แล้วทำการลากบล็อกสำหรับเริ่มการติดต่อกับโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ดังรูปที่ 3.65

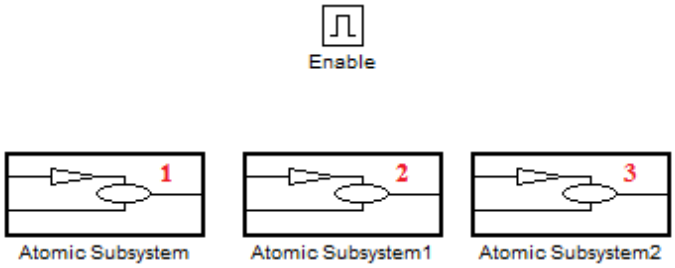
มหาวิทยาลัยเทคโนโลยีสุรนารี



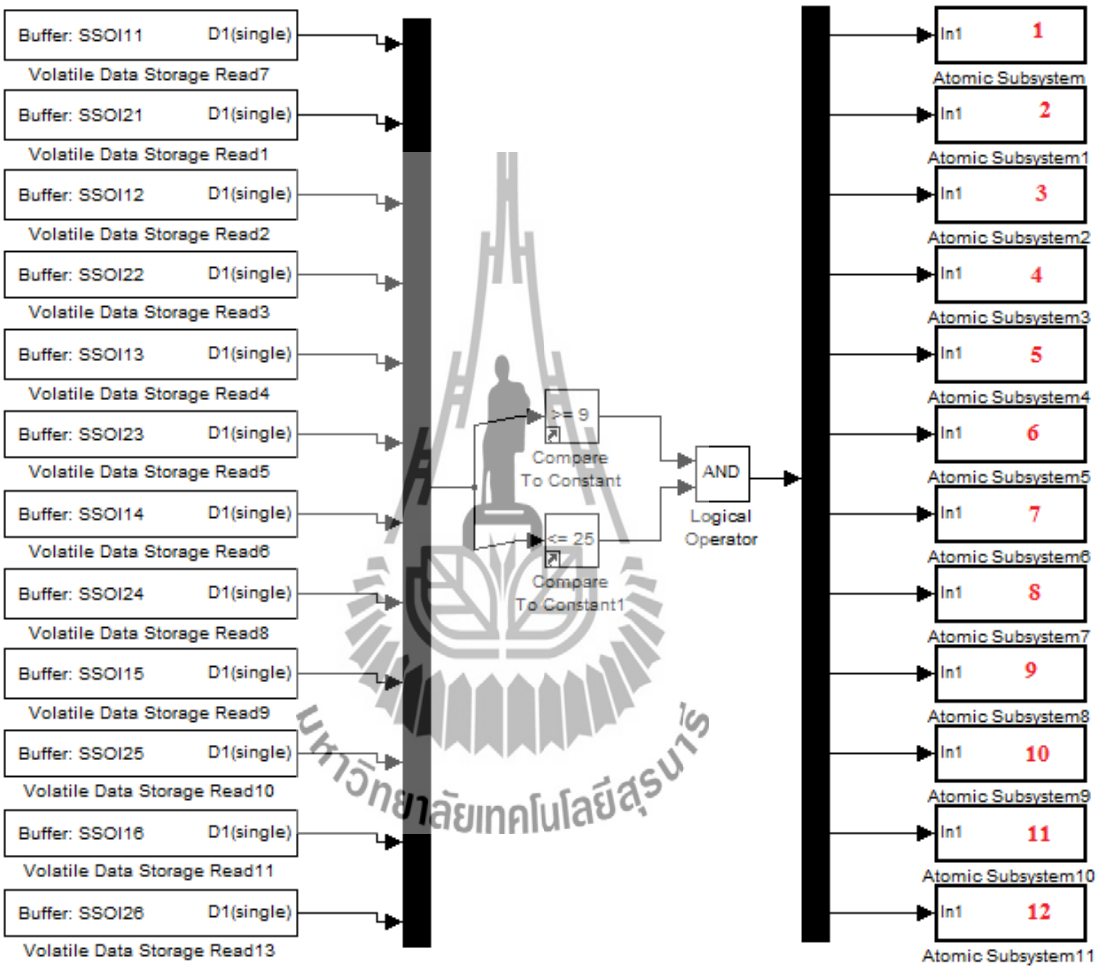
รูปที่ 3.32 บล็อกสำหรับควบคุมลำดับในการหาค่าเฉลี่ยและการเก็บข้อมูลลงบัฟเฟอร์



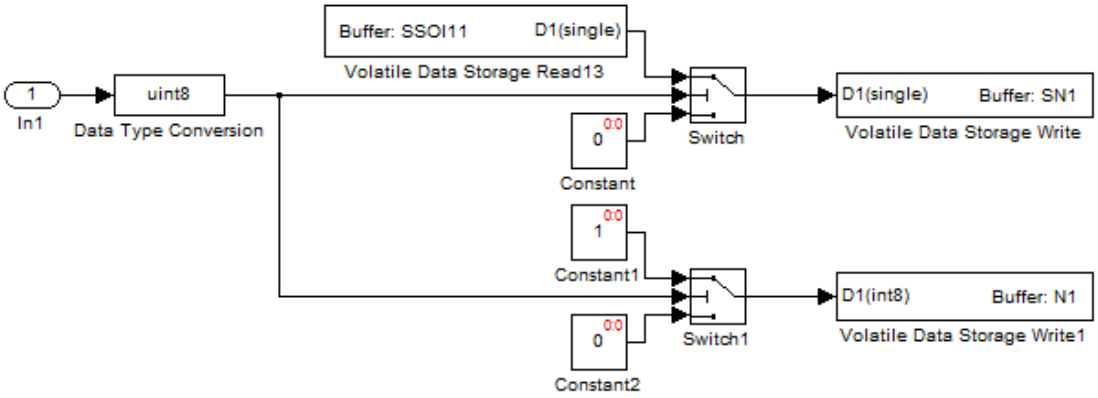
รูปที่ 3.33 บล็อกที่จะสั่งให้บล็อก Enabled subsystem หมายเลข 2 ของรูปที่ 3.32 เริ่มทำงาน



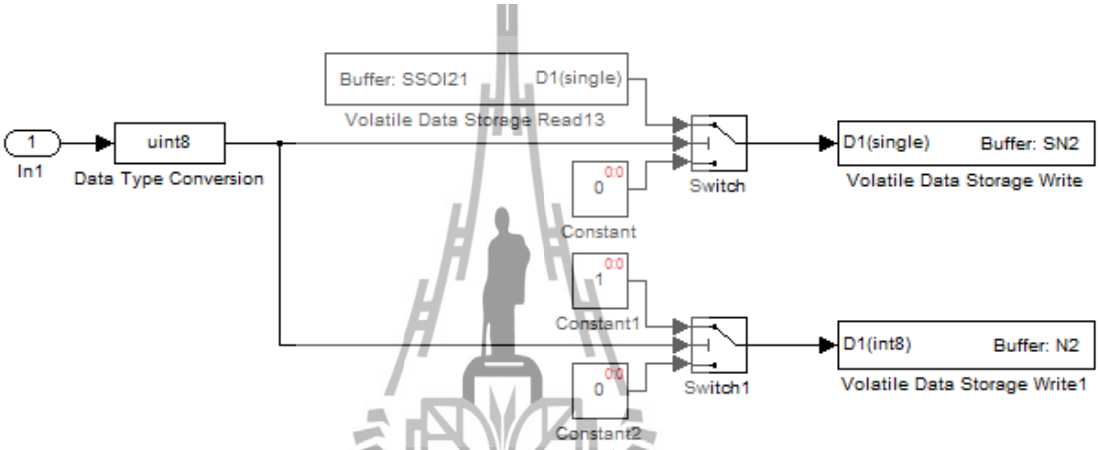
รูปที่ 3.34 บล็อก Atomic subsystem



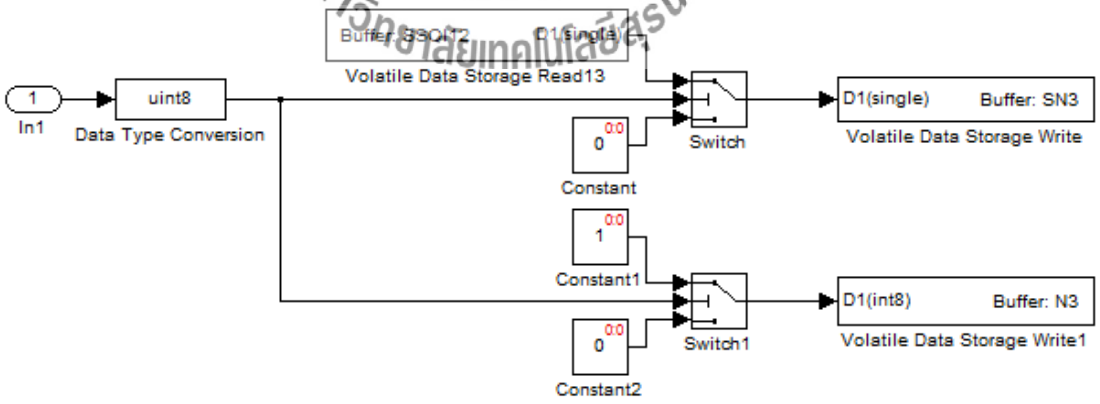
รูปที่ 3.35 บล็อกสำหรับจัดการค่าของเซ็นเซอร์ที่ผิดเพี้ยน



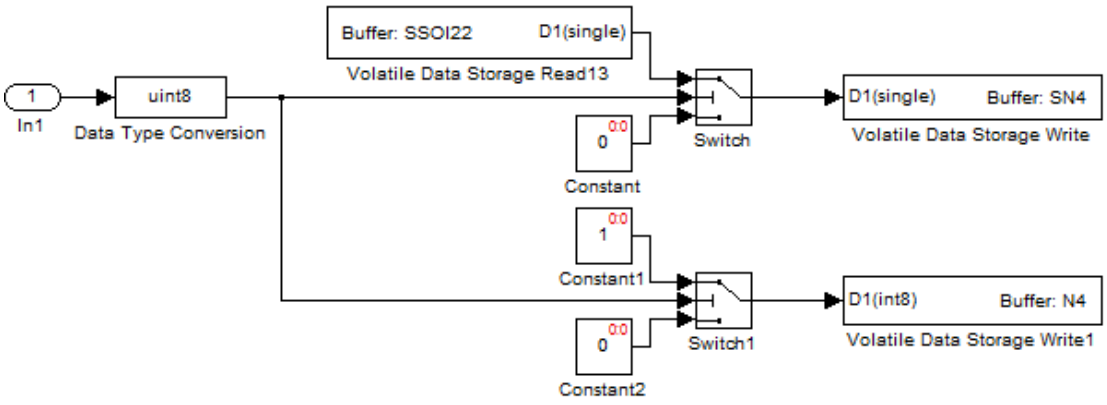
รูปที่ 3.36 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 1 ของรูปที่ 3.35ลงบัฟเฟอร์



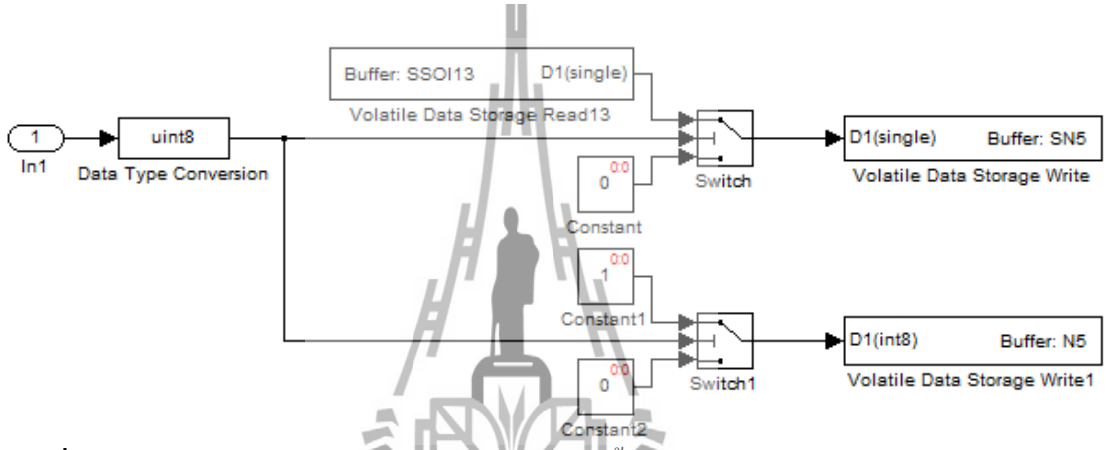
รูปที่ 3.37 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 2 ของรูปที่ 3.35ลงบัฟเฟอร์



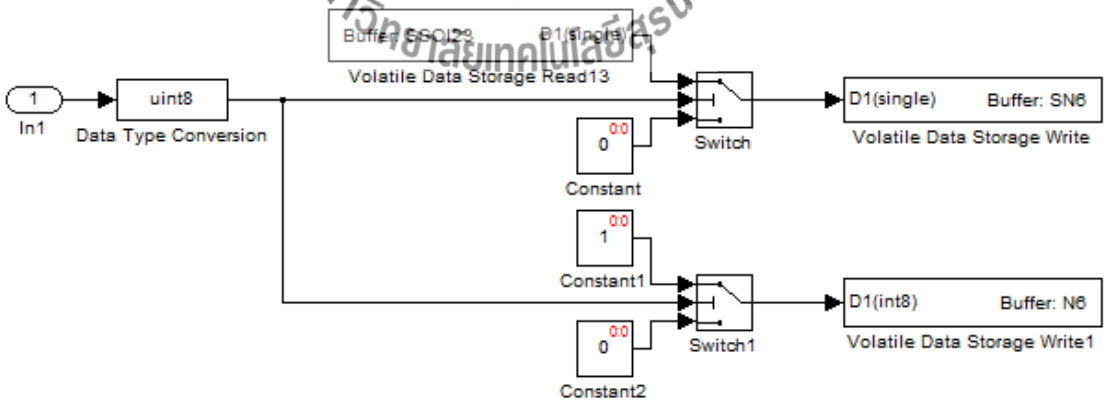
รูปที่ 3.38 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 3 ของรูปที่ 3.35ลงบัฟเฟอร์



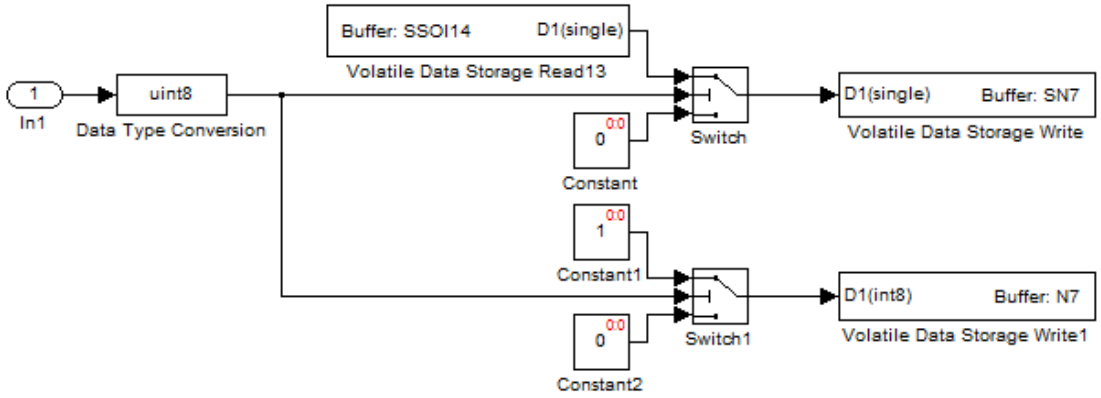
รูปที่ 3.39 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 4 ของรูปที่ 3.35ลงบัฟเฟอร์



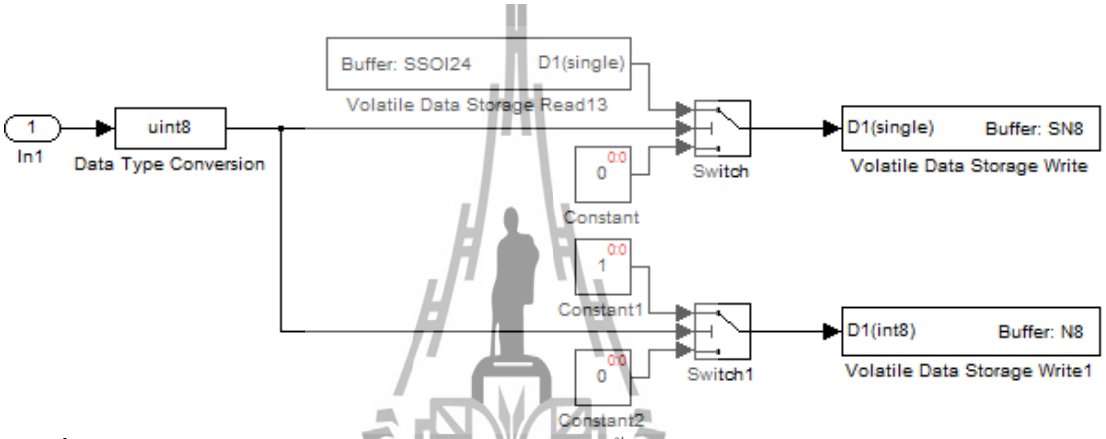
รูปที่ 3.40 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 5 ของรูปที่ 3.35ลงบัฟเฟอร์



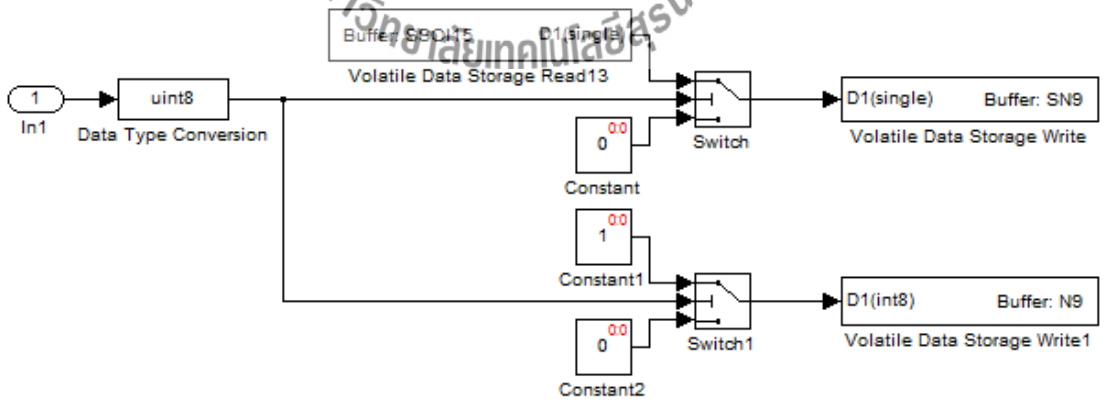
รูปที่ 3.41 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 6 ของรูปที่ 3.35ลงบัฟเฟอร์



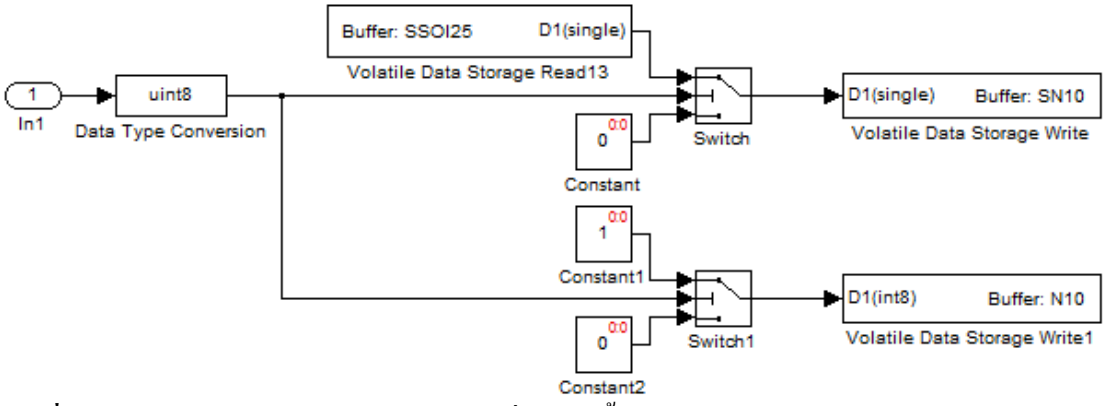
รูปที่ 3.42 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 7 ของรูปที่ 3.35ลงบัฟเฟอร์



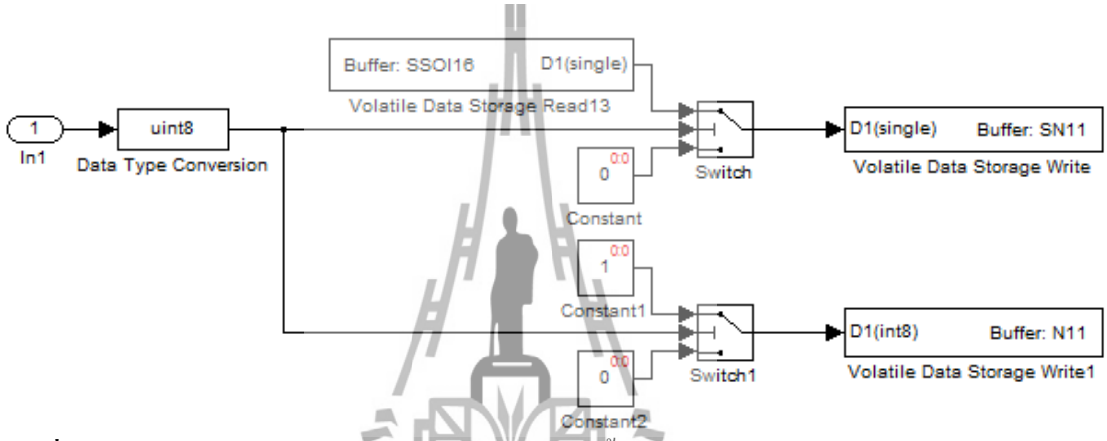
รูปที่ 3.43 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 8 ของรูปที่ 3.35ลงบัฟเฟอร์



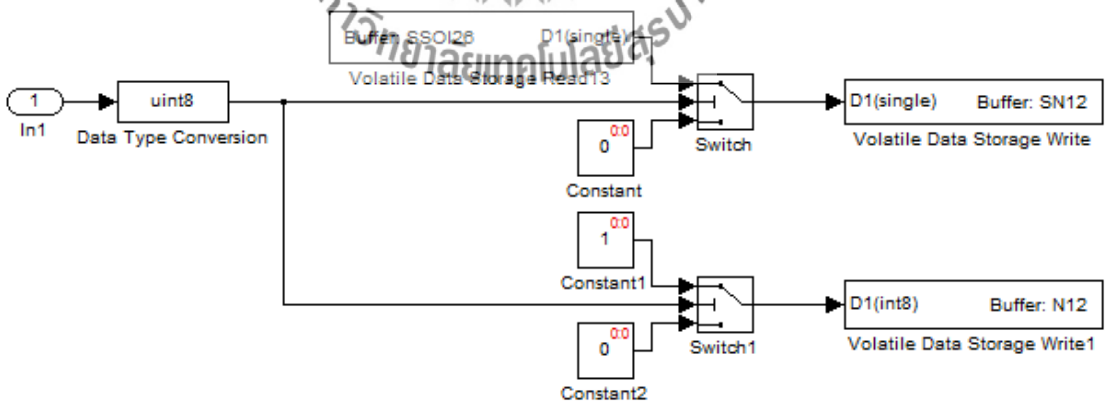
รูปที่ 3.44 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 9 ของรูปที่ 3.35ลงบัฟเฟอร์



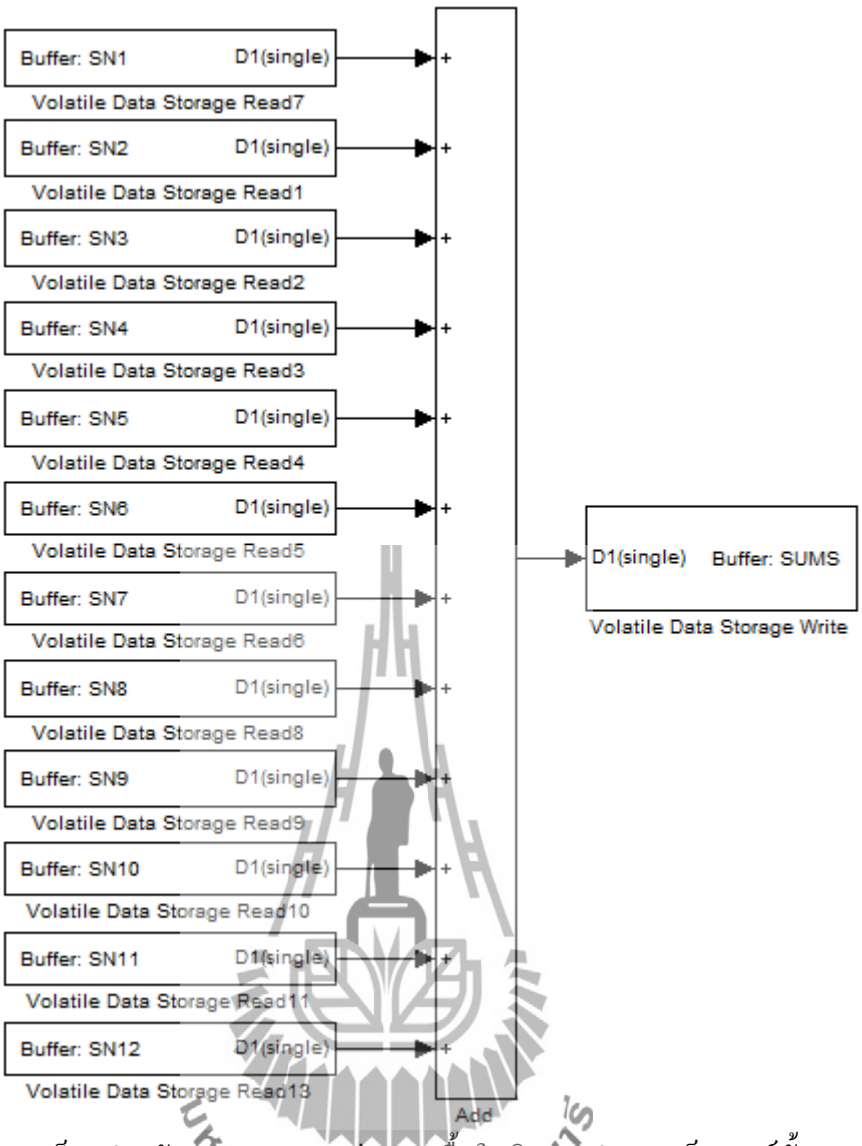
รูปที่ 3.45 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 10 ของรูปที่ 3.35ลงบัฟเฟอร์



รูปที่ 3.46 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 11 ของรูปที่ 3.35ลงบัฟเฟอร์

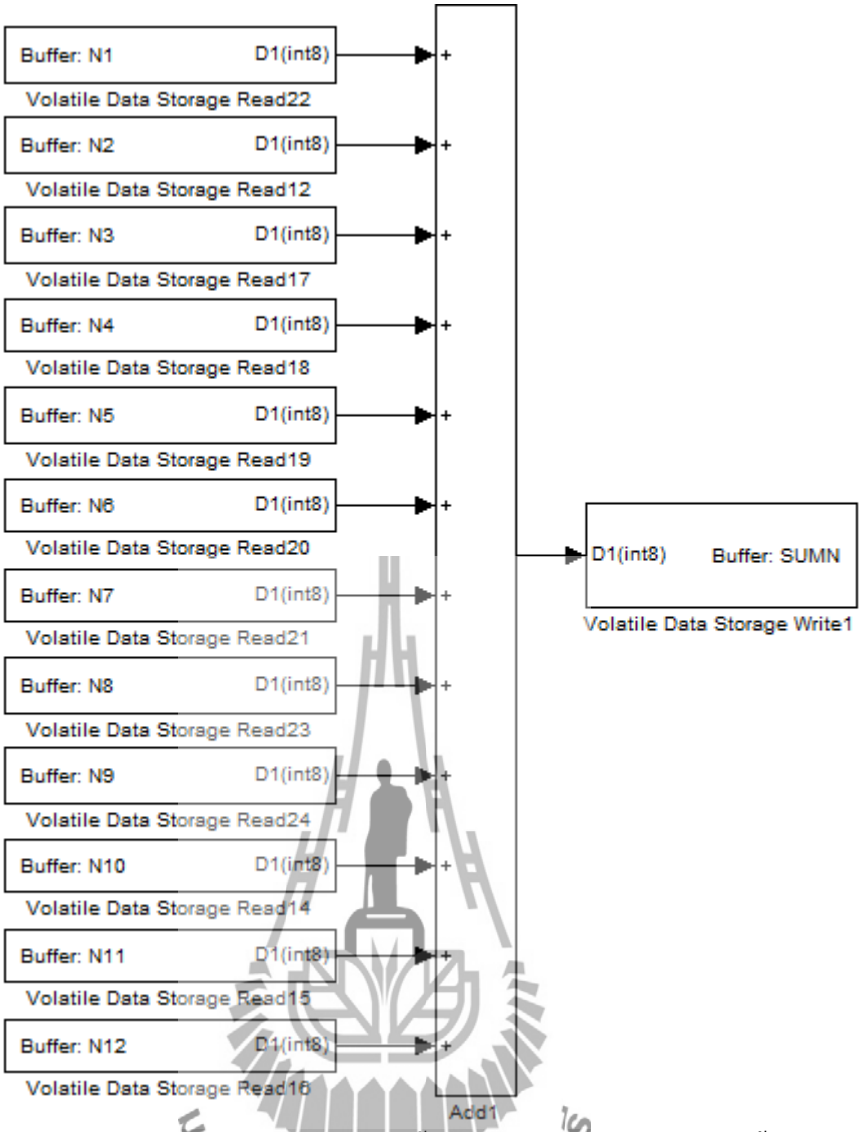


รูปที่ 3.47 บล็อกสำหรับเก็บค่าของเซ็นเซอร์ที่ไม่ผิดเพี้ยนจาก Atomic subsystemหมายเลข 12 ของรูปที่ 3.35ลงบัฟเฟอร์



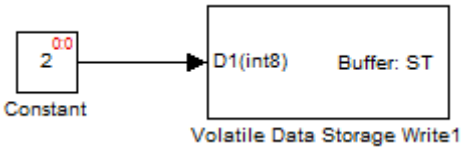
รูปที่ 3.48 บล็อกสำหรับหาผลรวมของค่าความชื้นในดินและจำนวนเซ็นเซอร์ทั้งหมดที่ใช้ได้

มหาวิทยาลัยเทคโนโลยีสุรนารี

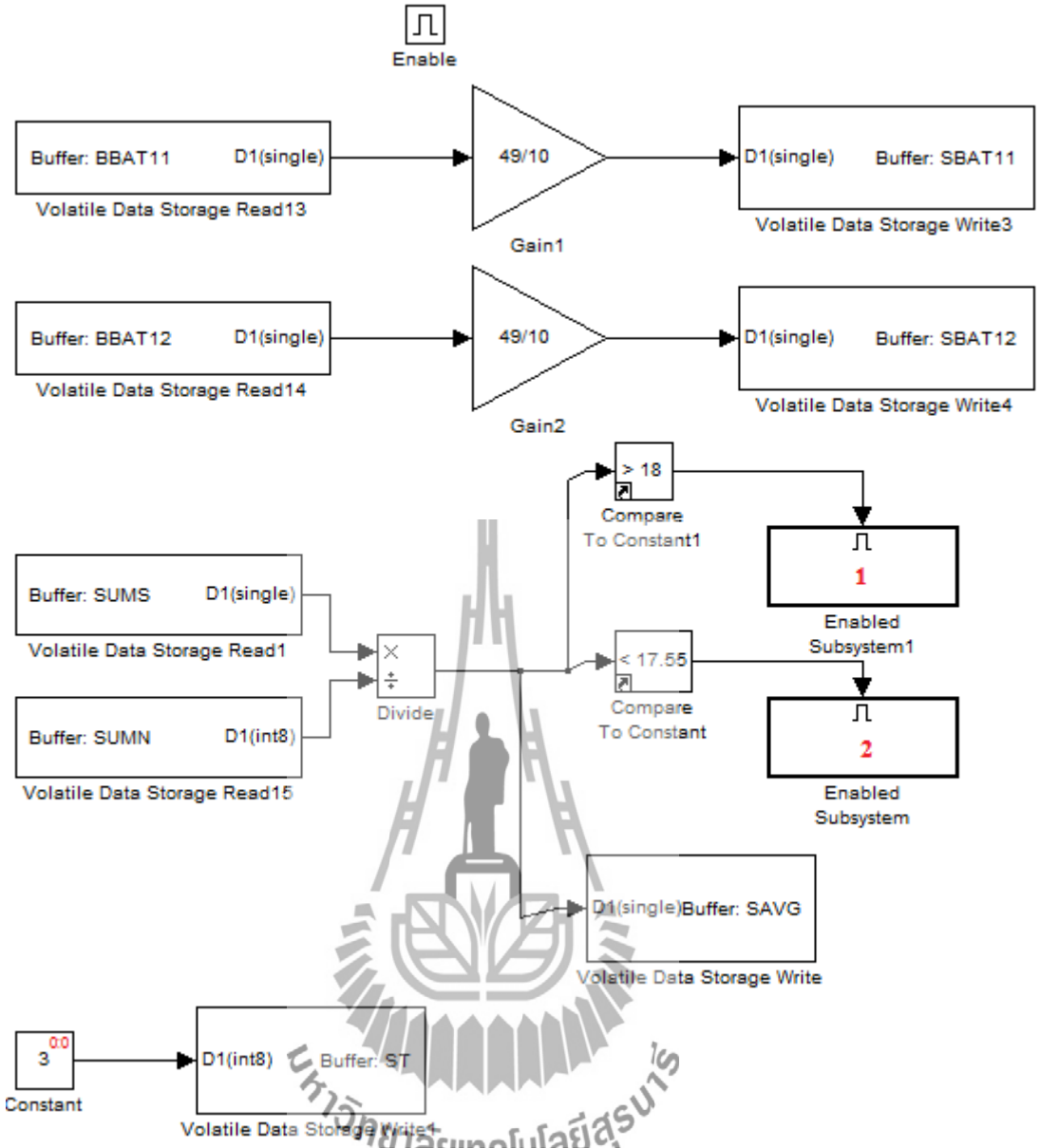


รูปที่ 3.49 บล็อกสำหรับหาผลรวมของค่าความชื้นในดินและจำนวนเซ็นเซอร์ทั้งหมดที่ใช้ได้ (ต่อ)

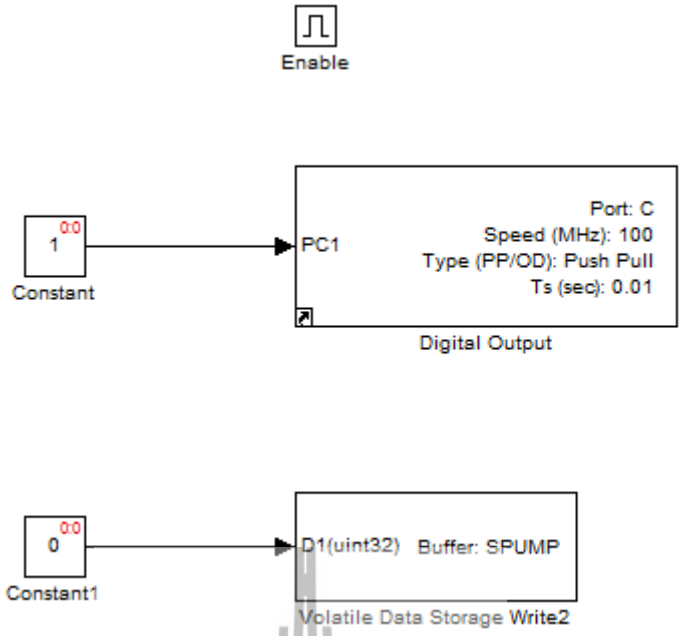
มหาวิทยาลัยเทคโนโลยีสุรนารี



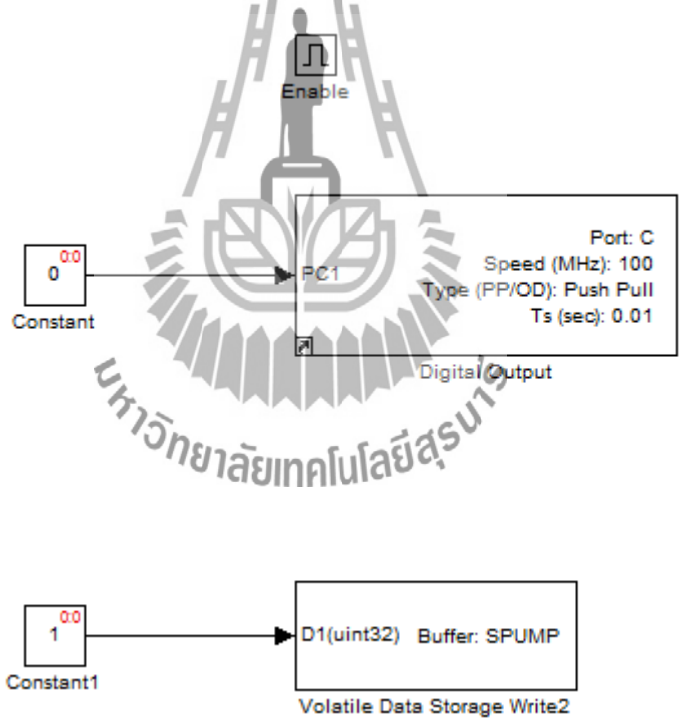
รูปที่ 3.50 บล็อกที่จะสั่งให้บล็อก Enabled subsystem หมายเลข 3 ของรูปที่ 3.32 เริ่มทำงาน



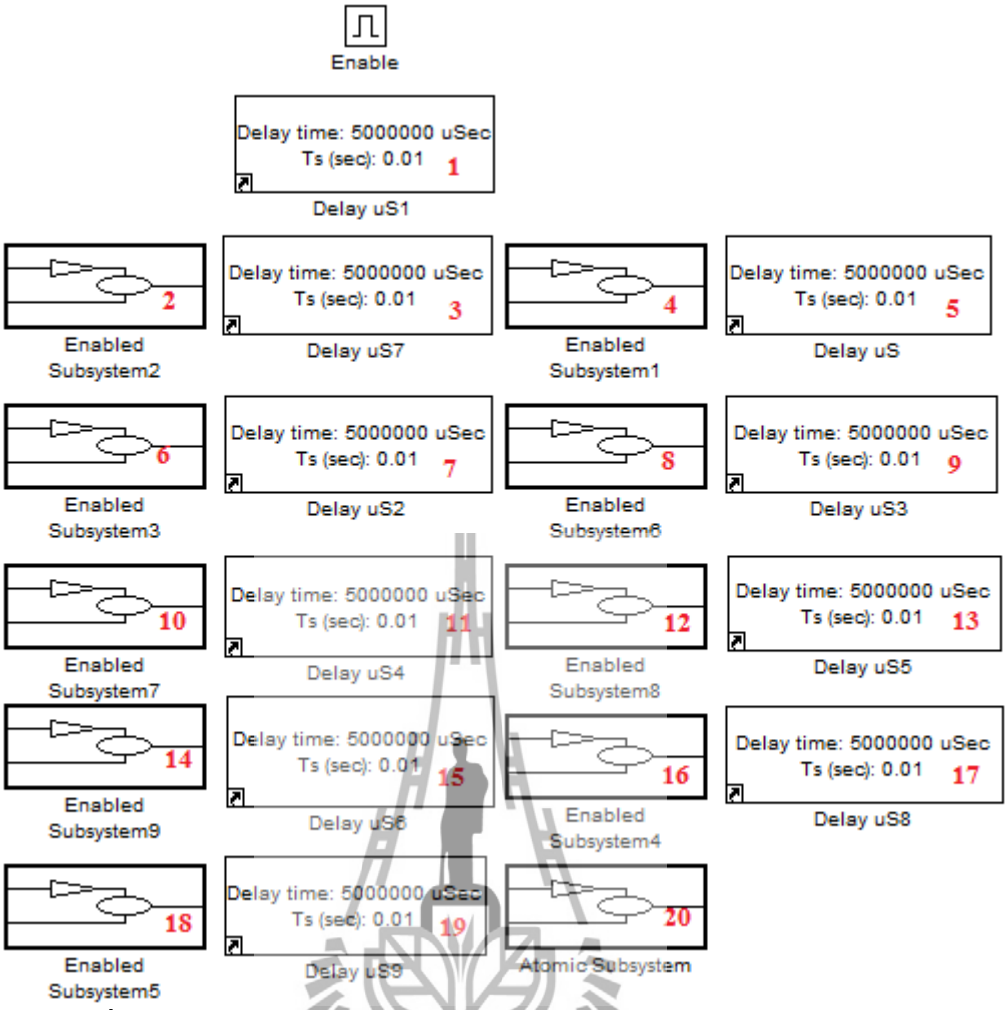
รูปที่ 3.51 บล็อกสำหรับควบคุมการเปิดปิดของปั๊มสูบน้ำ



รูปที่ 3.52 บล็อกสำหรับสั่งให้ปั๊มสูบน้ำหยุดทำงาน

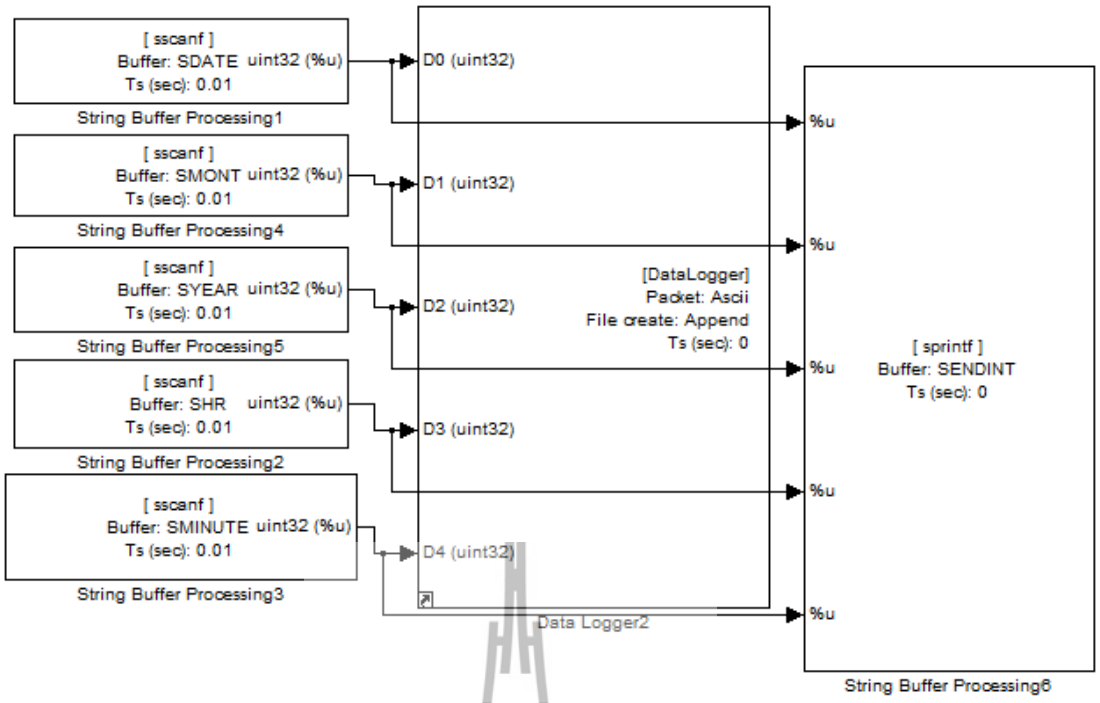


รูปที่ 3.53 บล็อกสำหรับสั่งให้ปั๊มสูบน้ำทำงาน

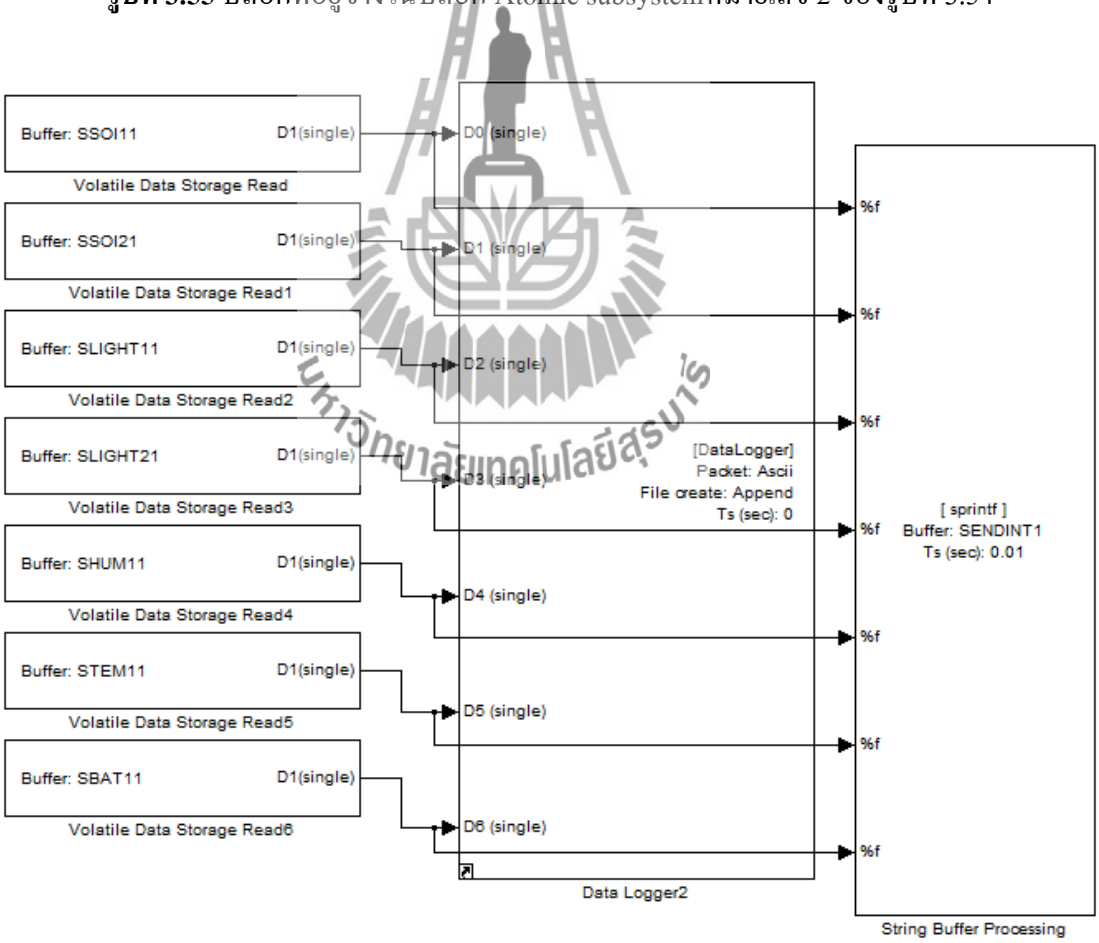


รูปที่ 3.54 บล็อกสำหรับเรียงลำดับการเก็บค่าลงบัฟเฟอร์และเขียนลงเอสดีการ์ด

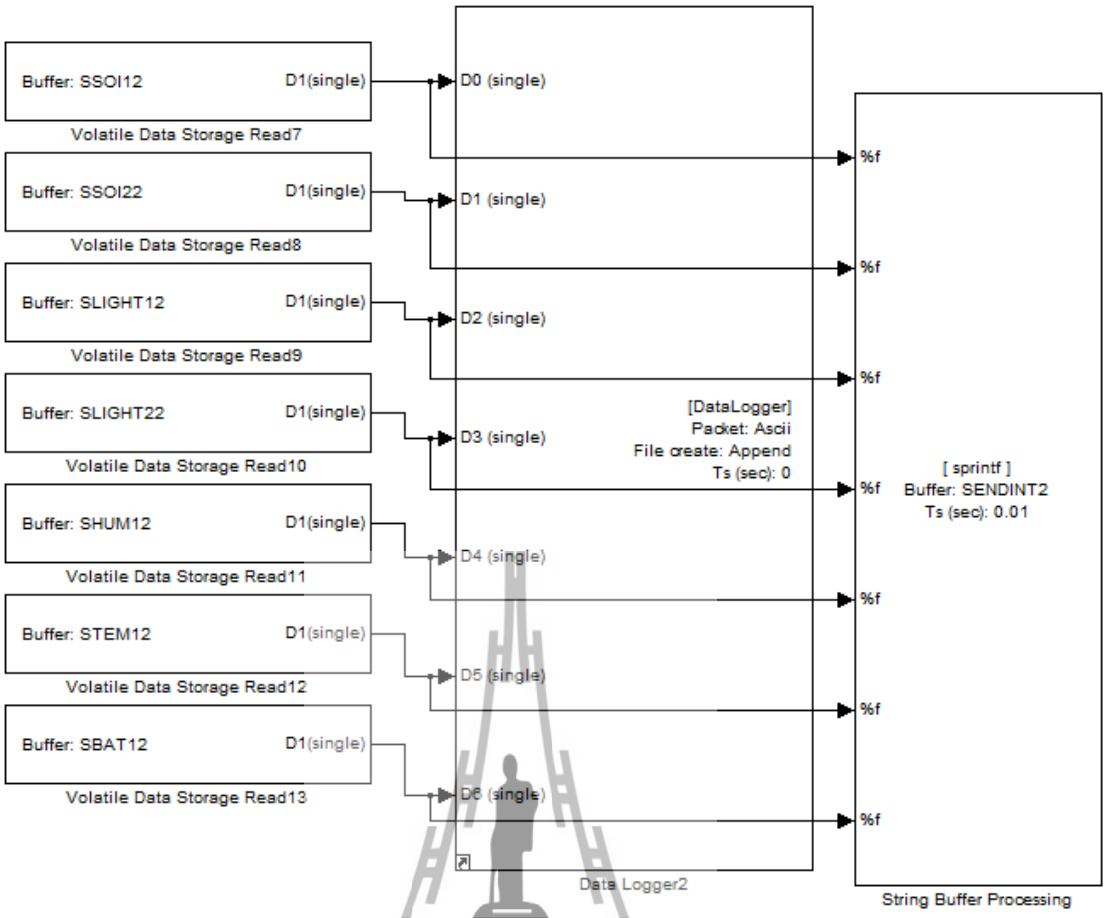
มหาวิทยาลัยเทคโนโลยีสุรนารี



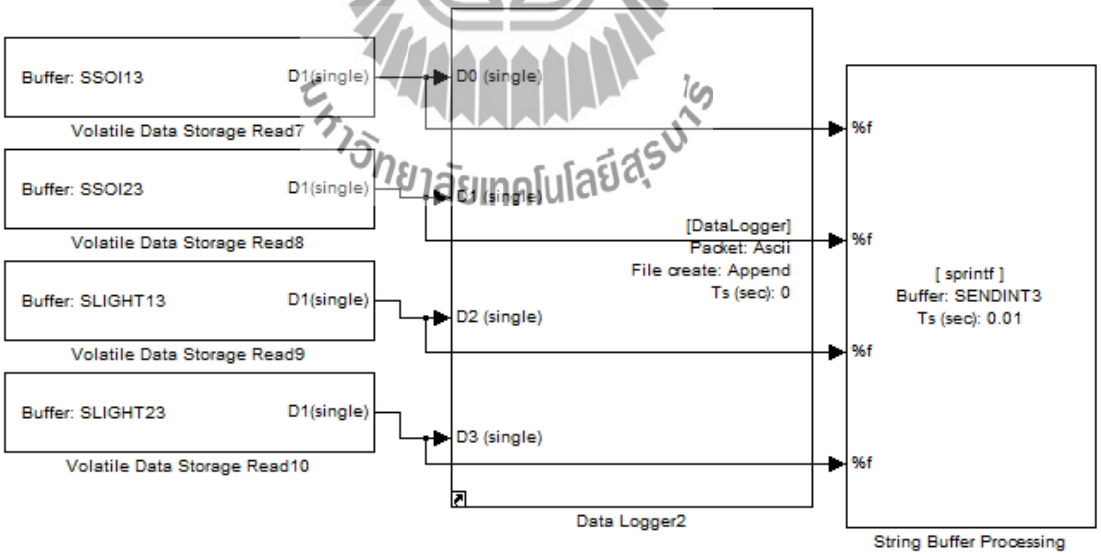
รูปที่ 3.55 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 2 ของรูปที่ 3.54



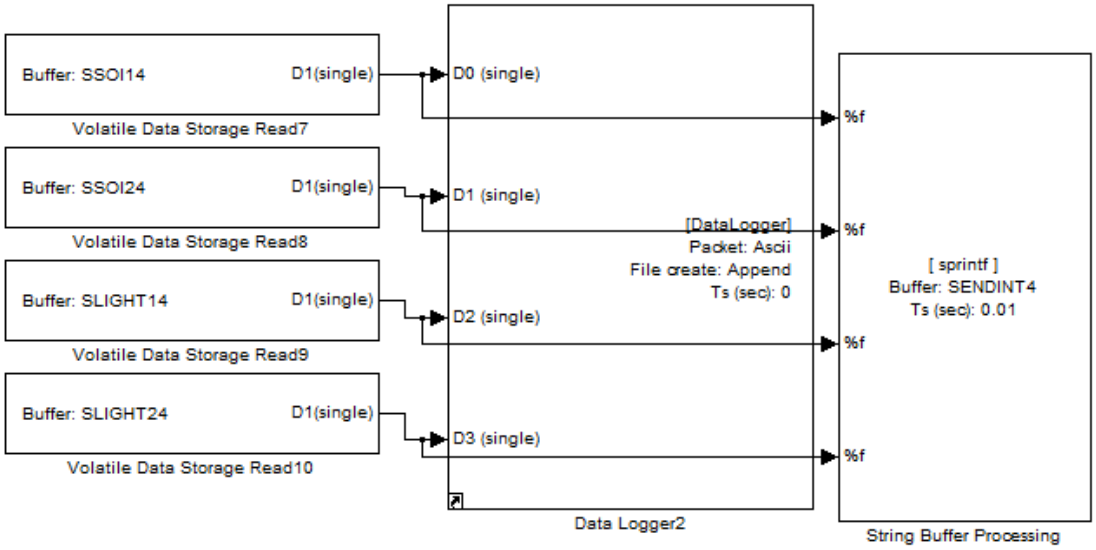
รูปที่ 3.56 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 4 ของรูปที่ 3.54



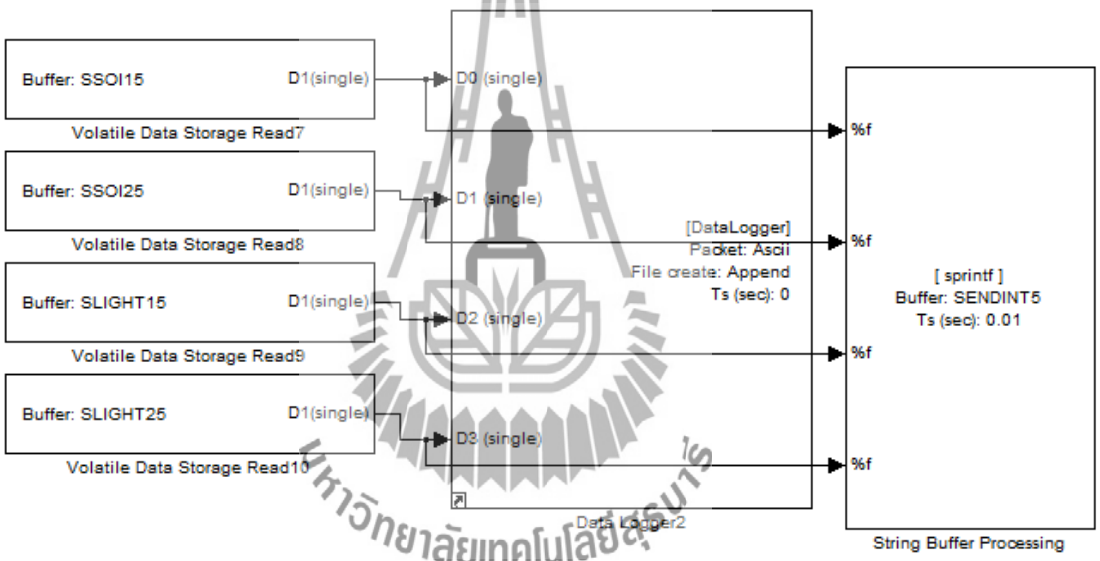
รูปที่ 3.57 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 6 ของรูปที่ 3.54



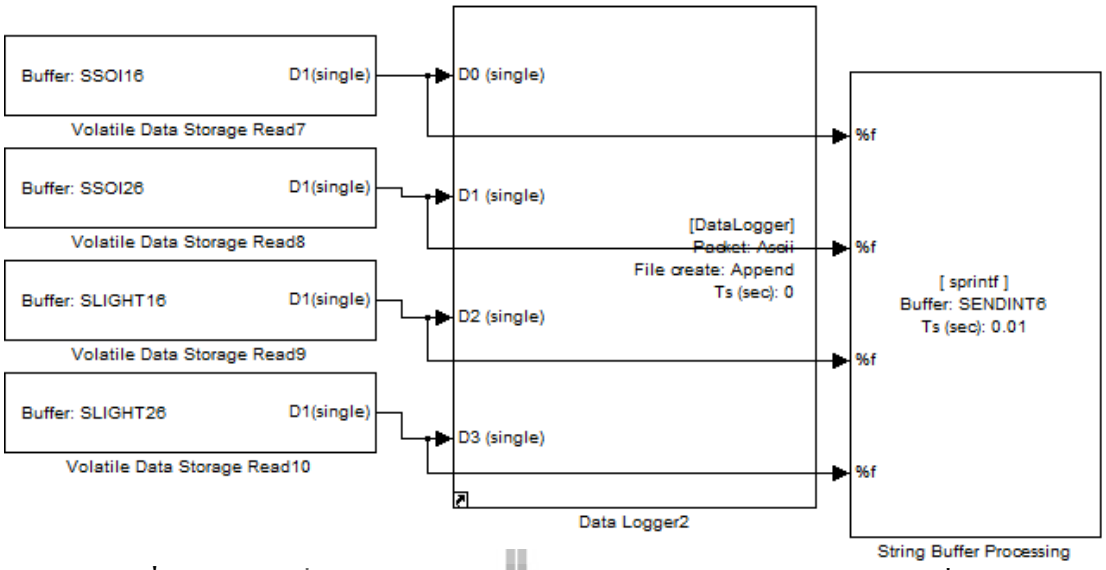
รูปที่ 3.58 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 8 ของรูปที่ 3.54



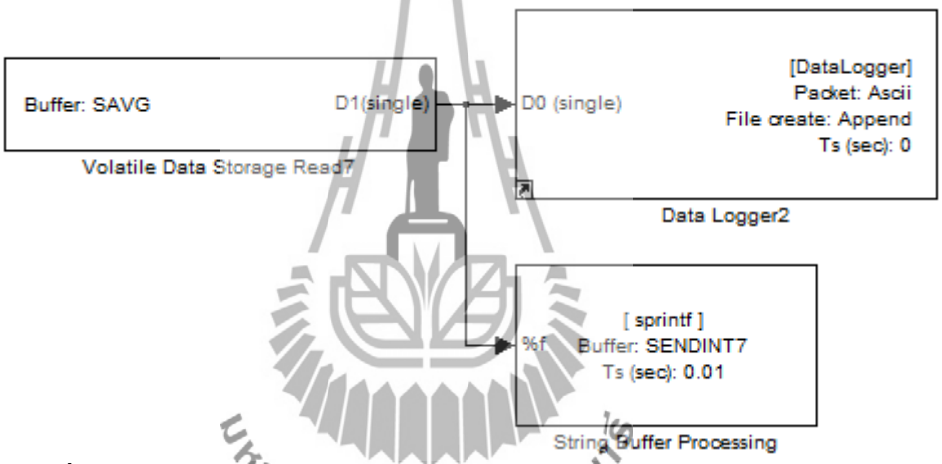
รูปที่ 3.59 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 10 ของรูปที่ 3.54



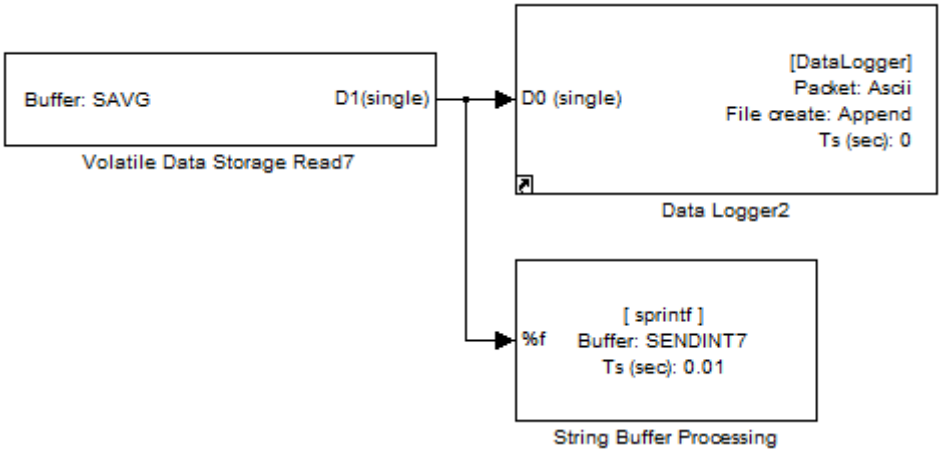
รูปที่ 3.60 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 12 ของรูปที่ 3.54



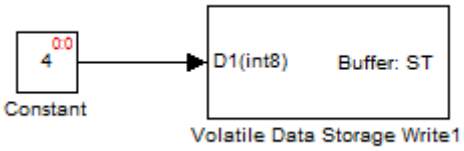
รูปที่ 3.61 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 14 ของรูปที่ 3.54



รูปที่ 3.62 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 16 ของรูปที่ 3.54



รูปที่ 3.63 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 18 ของรูปที่ 3.54



รูปที่ 3.64 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 20 ของรูปที่ 3.54

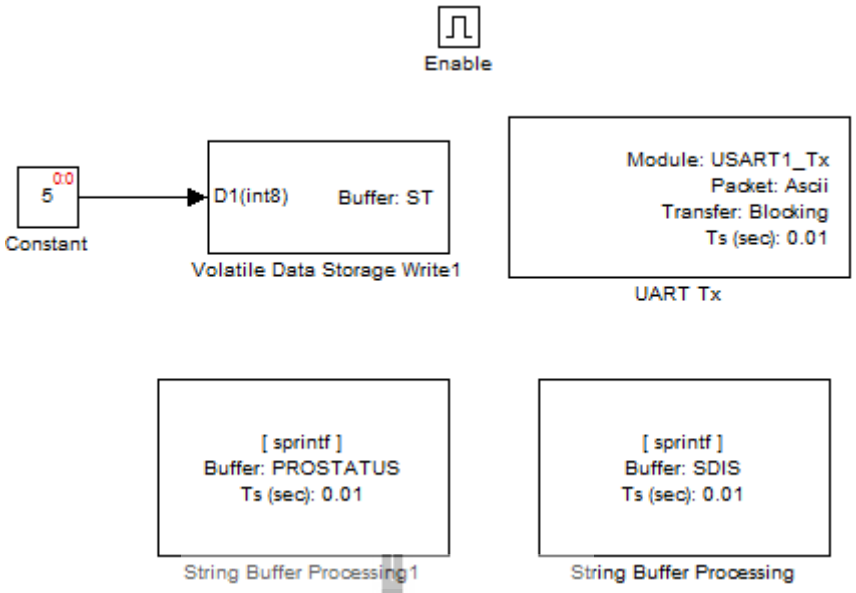
ตารางที่ 3.12 ค่าพารามิเตอร์ของบล็อก Data logger

รูปที่	Filename	Header	Ascii format
3.55	'data.txt'	''	'%u/%u/%u,%u:%u\r\n'
3.56	'data.txt'	''	'NODE1:S1=%.2f,S2=%.2f,L1=%.2f,L2=%.2f,H1=%.2f,T1=%.2f,B1=%.2f\r\n'
3.57	'data.txt'	''	'NODE2:S3=%.2f,S4=%.2f,L3=%.2f,L4=%.2f,H2=%.2f,T2=%.2f,B2=%.2f\r\n'
3.58	'data.txt'	''	'NODE3:S5=%.2f,S6=%.2f,L5=%.2f,L6=%.2f\r\n'
3.59	'data.txt'	''	'NODE4:S7=%.2f,S8=%.2f,L7=%.2f,L8=%.2f\r\n'
3.60	'data.txt'	''	'NODE5:S9=%.2f,S10=%.2f,L9=%.2f,L10=%.2f\r\n'
3.61	'data.txt'	''	'NODE6:S11=%.2f,S12=%.2f,L11=%.2f,L12=%.2f\r\n'

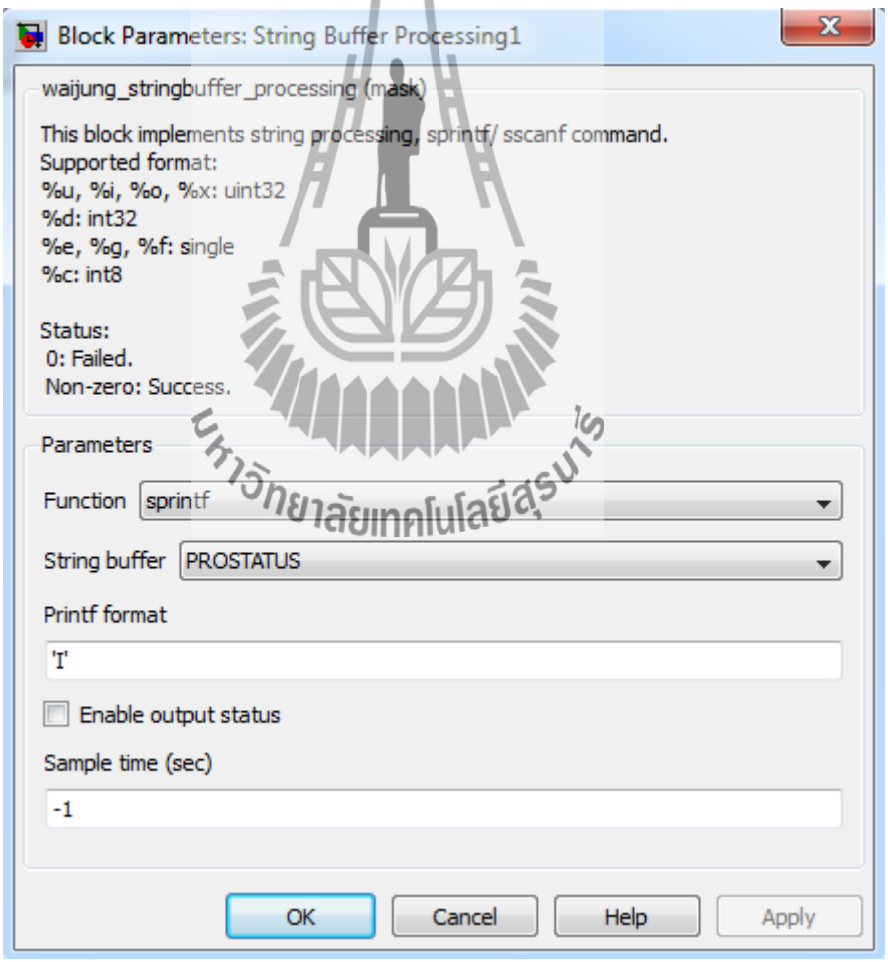
ตารางที่ 3.13 ค่าพารามิเตอร์ของบล็อก String buffer processing

รูปที่	Function	String buffer	Printf format
3.55	sprintf	SENDINT	'L1=%u/%u/%u,%u:%u&'
3.56	sprintf	SENDINT1	'L2=NODE1:S1=%.2f,S2=%.2f,L1=%.2f,L2=%.2f,H1=%.2f,T1=%.2f,B1=%.2f&'
3.57	sprintf	SENDINT2	'L3=NODE2:S3=%.2f,S4=%.2f,L3=%.2f,L4=%.2f,H2=%.2f,T2=%.2f,B2=%.2f&'
3.58	sprintf	SENDINT3	'L4=NODE3:S5=%.2f,S6=%.2f,L5=%.2f,L6=%.2f&'
3.59	sprintf	SENDINT4	'L5=NODE4:S7=%.2f,S8=%.2f,L7=%.2f,L8=%.2f&'
3.60	sprintf	SENDINT5	'L6=NODE5:S9=%.2f,S10=%.2f,L9=%.2f,L10=%.2f&'
3.61	sprintf	SENDINT6	'L7=NODE6:S11=%.2f,S12=%.2f,L11=%.2f,L12=%.2f&'

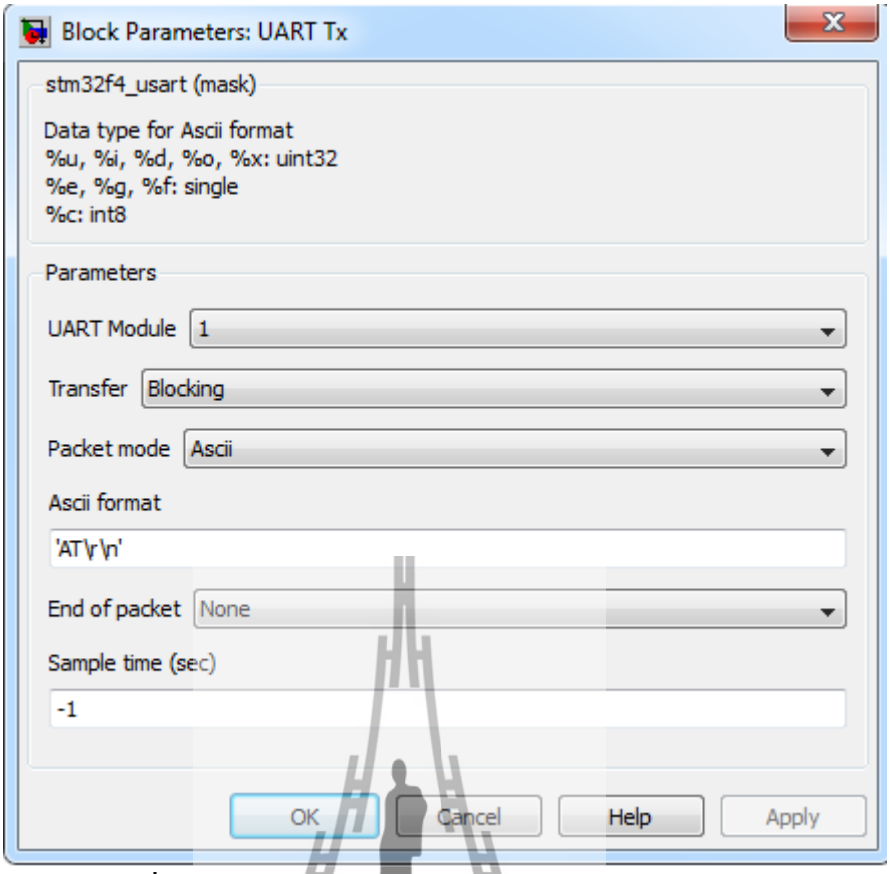
มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.65 บล็อกที่อยู่ข้างในบล็อก Atomic subsystemหมายเลข 20 ของรูปที่ 3.54

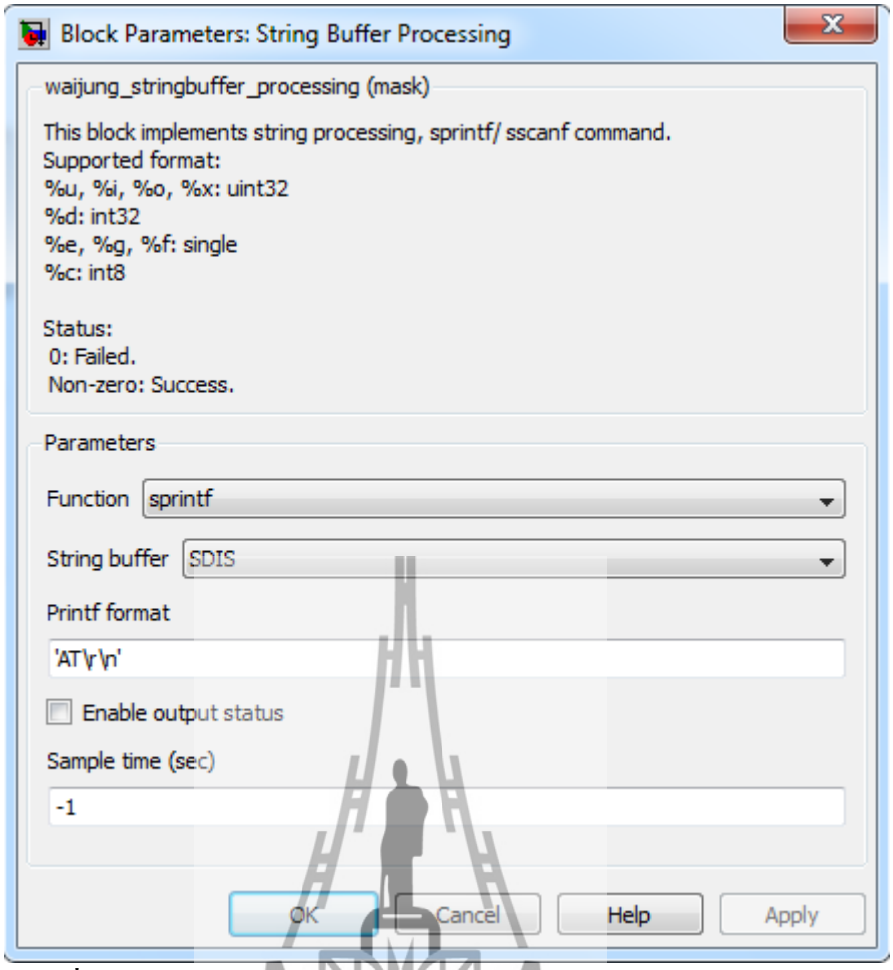


รูปที่ 3.66 การกำหนดค่าของบล็อก String buffer processing 1 ของรูปที่ 3.65



รูปที่ 3.67 การกำหนดค่าของบล็อก UART Tx ของรูปที่ 3.65

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.68 การกำหนดค่าของบล็อก String buffer processing ของรูปที่ 3.65

8. ในส่วนของหมายเลข 5 ของรูปที่ 3.22 ทำการ ลากบล็อก Atomic subsystem จากนั้นดับเบิ้ลคลิกเข้าไปแล้วทำการลากบล็อกสำหรับรับค่าจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ ดังรูปที่ 3.69 และทำการกำหนดค่าให้กับบล็อก UART Rx ดังรูปที่ 3.70

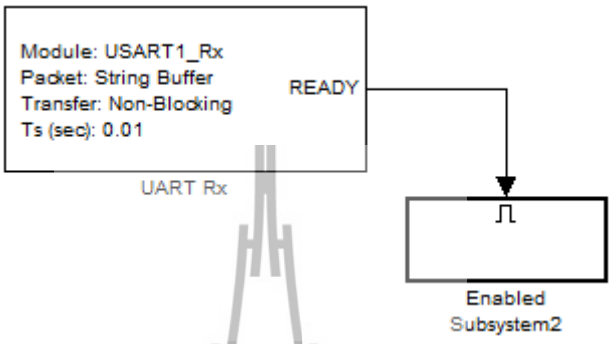
9. ดับเบิ้ลคลิกเข้าไปในบล็อก Enabled subsystem ของรูปที่ 3.69 แล้วทำการลากบล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ดังรูปที่ 3.71 ถึงรูปที่ 3.86ตามลำดับ

10. ดับเบิ้ลคลิกเข้าไปในบล็อก Triggered subsystem ของรูปที่ 3.71 ถึงรูปที่ 3.83 แล้วทำการลากบล็อกสำหรับส่งค่าออกไปติดต่อกับโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ดังรูปที่ 3.87 ตามลำดับจนครบ จากนั้นทำการกำหนดค่าพารามิเตอร์ดังตารางที่ 3.14 และตารางที่ 3.15

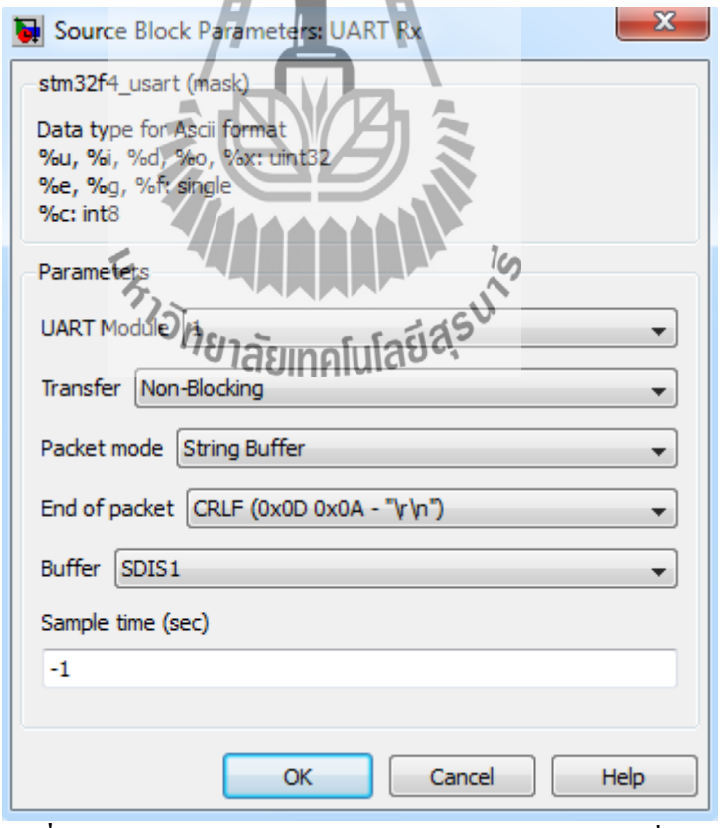
11. ดับเบิ้ลคลิกเข้าไปในบล็อก Triggered subsystem ของรูปที่ 3.84 แล้วทำการลากบล็อกสำหรับส่งข้อมูลขึ้นเซิร์ฟเวอร์ จากนั้นทำการกำหนดค่า Priority ของแต่ละบล็อกให้ตรงกับหมายเลขที่กำกับดังรูปที่ 3.88 จากนั้นทำการกำหนดค่าพารามิเตอร์ของแต่ละบล็อกดังตารางที่ 3.16 และตารางที่ 3.17

12. ดับเบิ้ลคลิกเข้าไปในบล็อก Triggered subsystem ของรูปที่ 3.85 แล้วทำการลากบล็อก String buffer processingดังรูปที่ 3.89 (ก) (ข) และ (ค) พร้อมทั้งทำการกำหนดค่าดังรูปที่ 3.90 3.91 และรูปที่ 3.92 ตามลำดับ

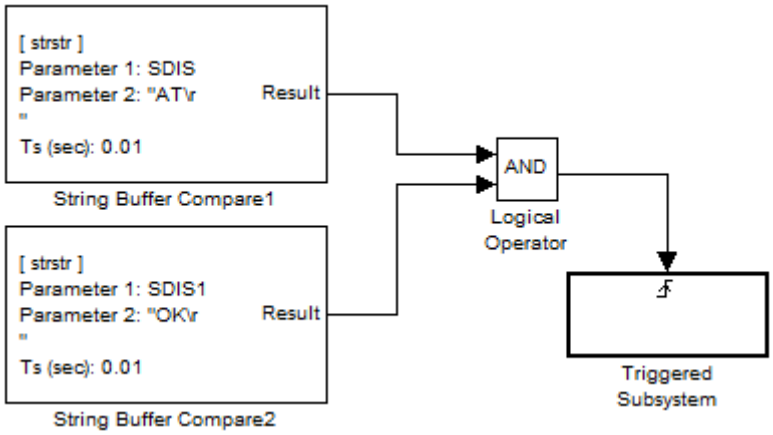
13. ดับเบิ้ลคลิกเข้าไปในบล็อก Triggered subsystem ของรูปที่ 3.86 แล้วทำการลากบล็อก UART Tx และ String buffer processingดังรูปที่ 3.93 (ก) (ข) (ค)และ (ง) พร้อมทั้งทำการกำหนดค่าดังรูปที่ 3.94 3.95 3.96 และรูปที่ 3.97 ตามลำดับ



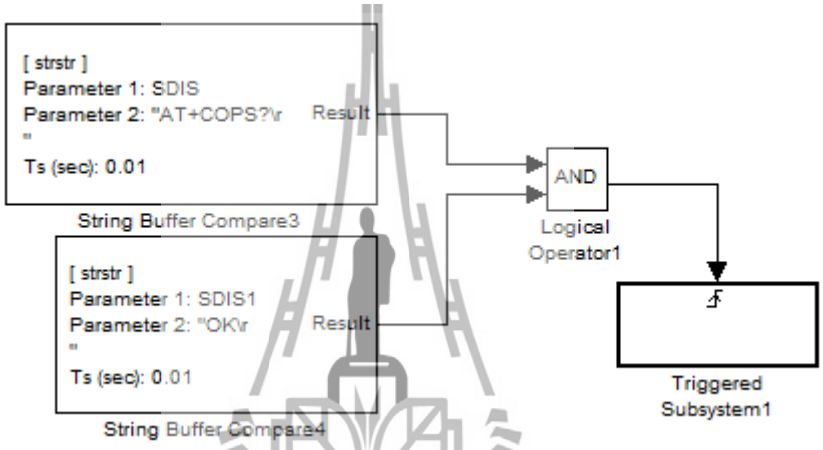
รูปที่ 3.69 บล็อกสำหรับรับค่าจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์



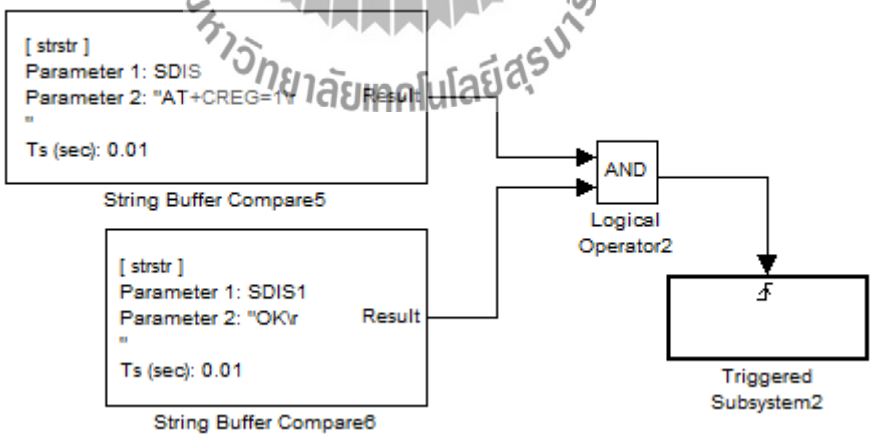
รูปที่ 3.70 การกำหนดค่าสำหรับบล็อก UART Rx ของรูปที่ 3.69



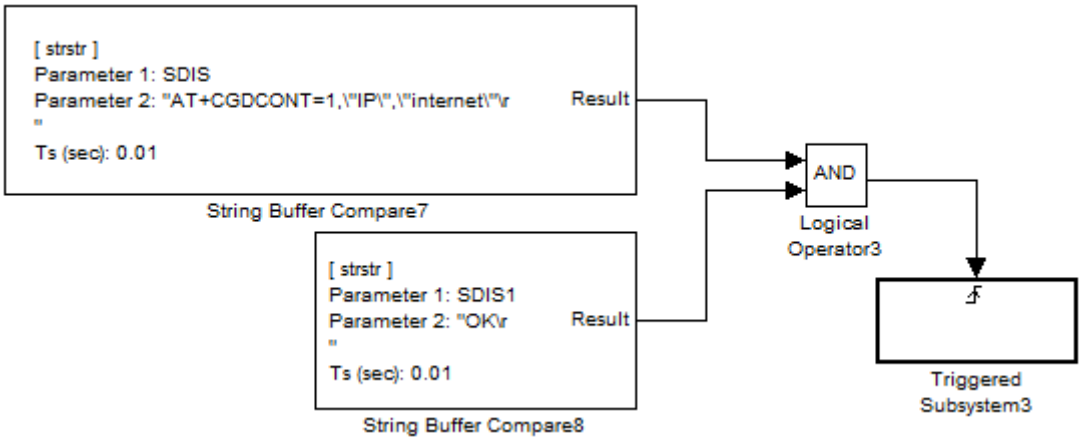
รูปที่ 3.71 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์



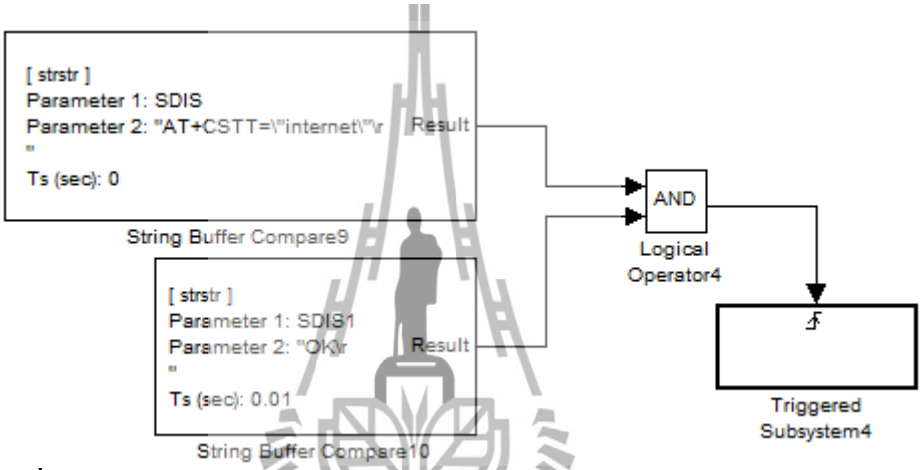
รูปที่ 3.72 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



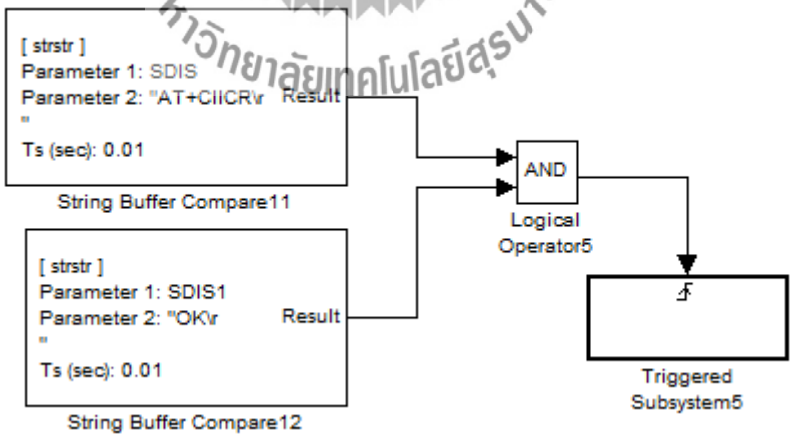
รูปที่ 3.73 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



รูปที่ 3.74 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)

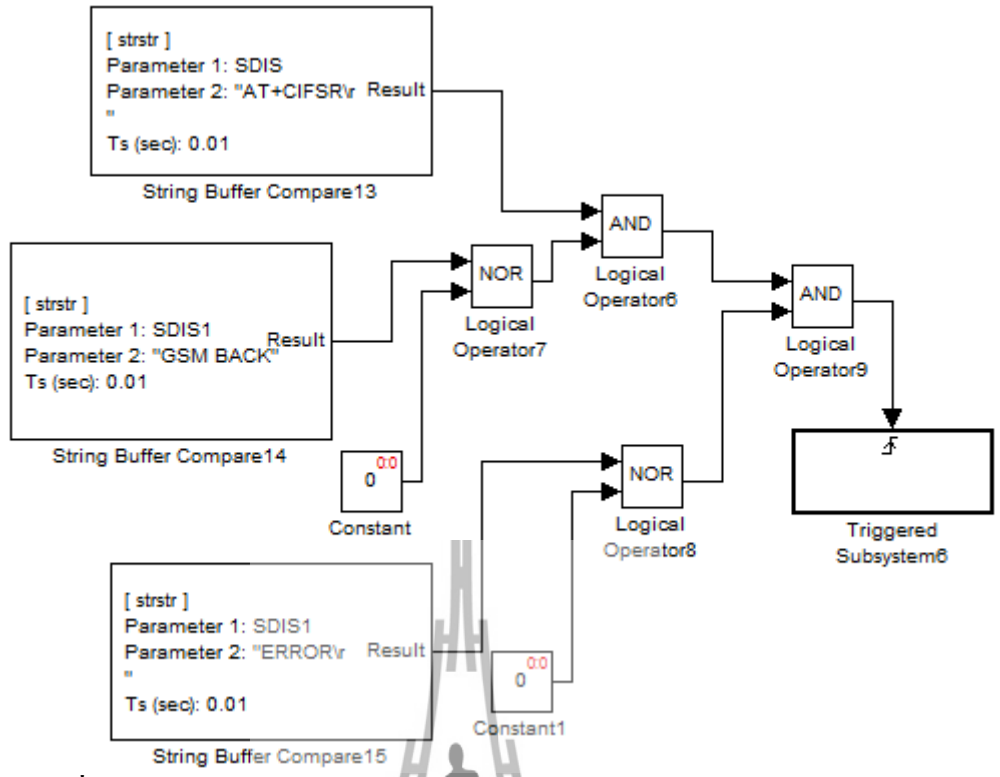


รูปที่ 3.75 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)

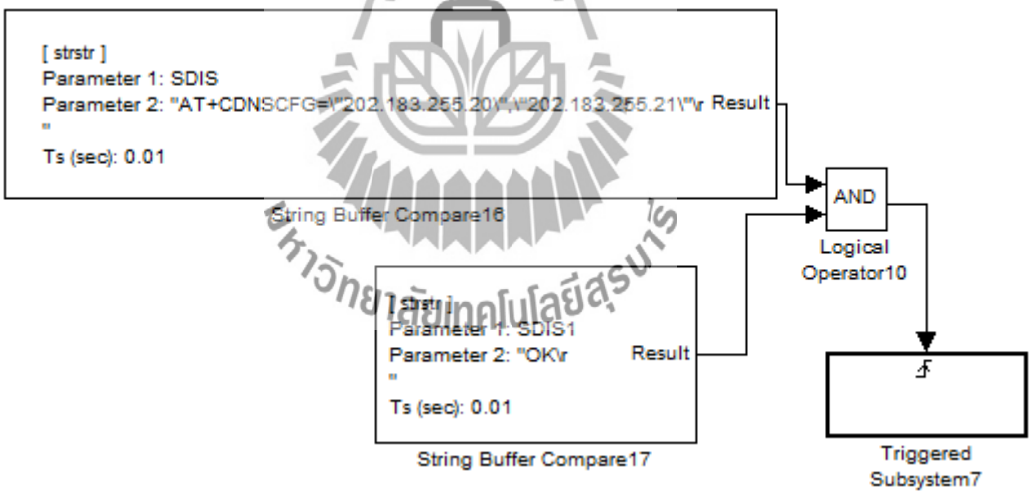


รูปที่ 3.76 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)

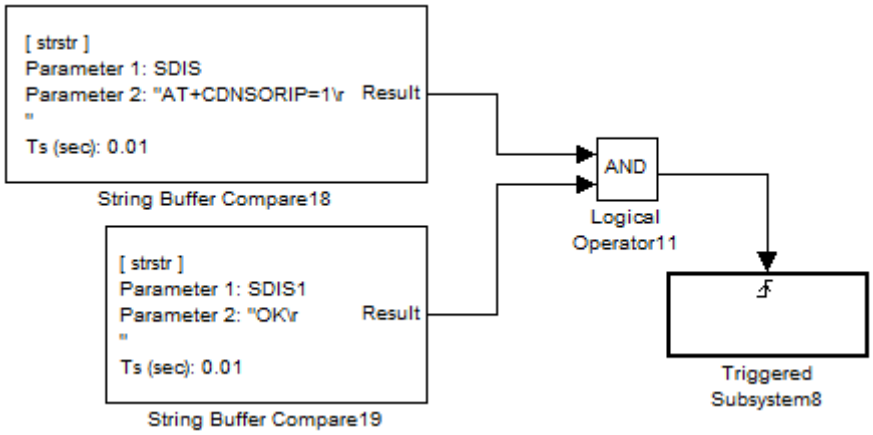
มหาวิทยาลัยเทคโนโลยีสุรนารี



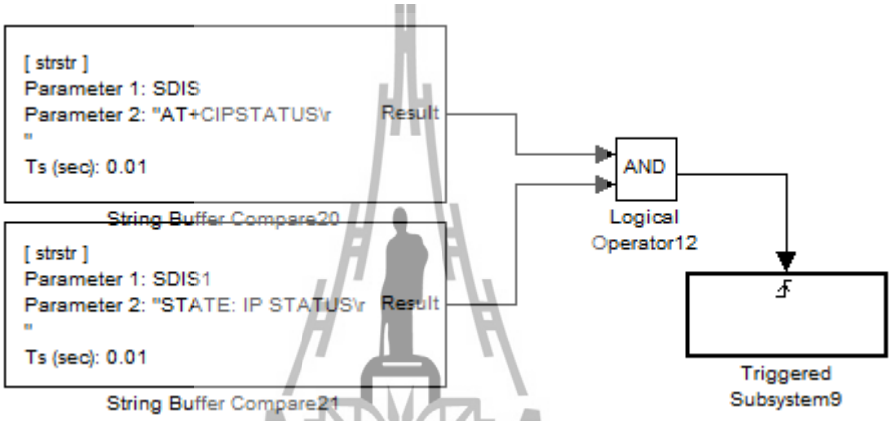
รูปที่ 3.77 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



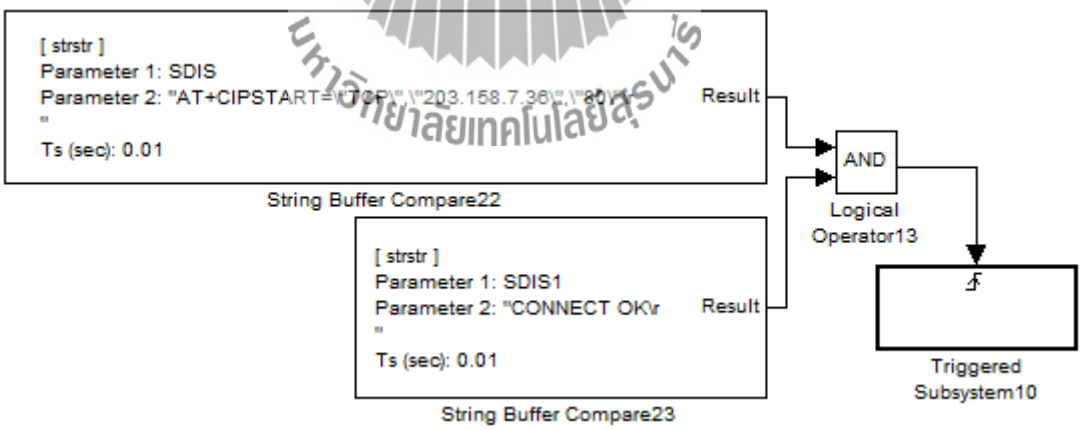
รูปที่ 3.78 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



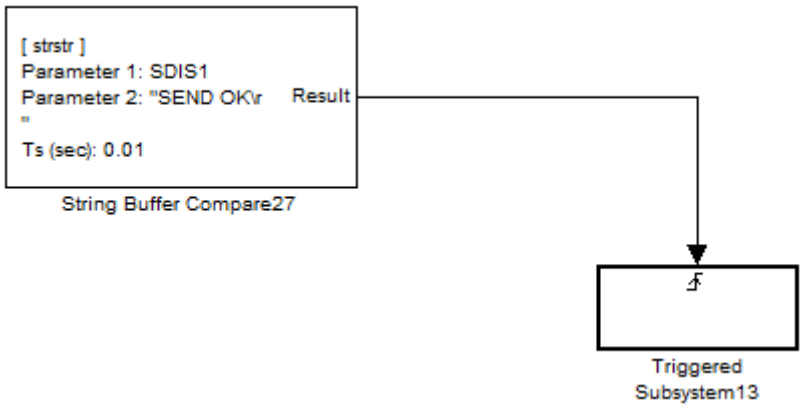
รูปที่ 3.79 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



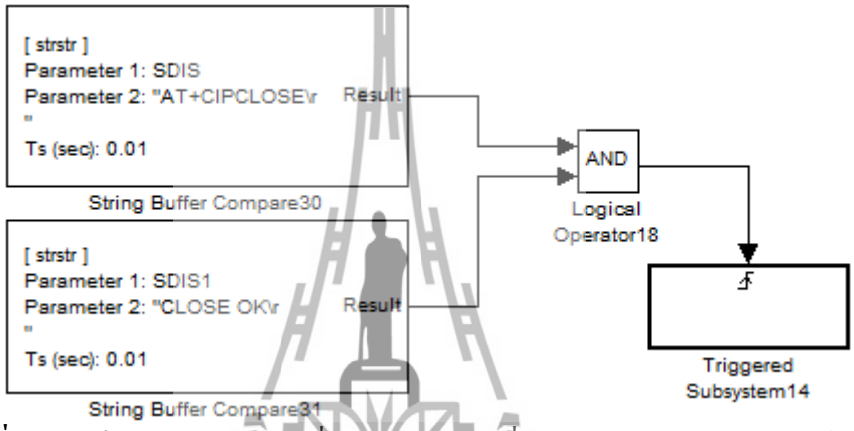
รูปที่ 3.80 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



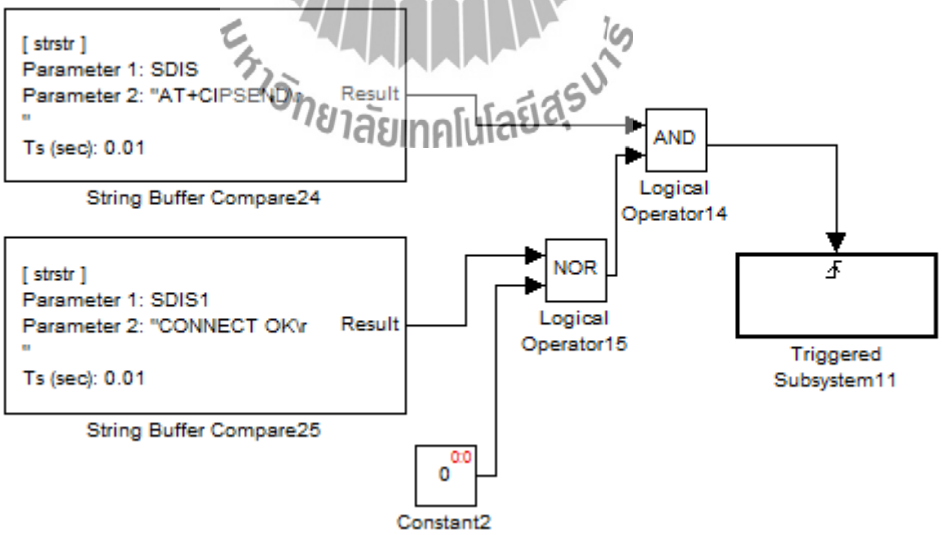
รูปที่ 3.81 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



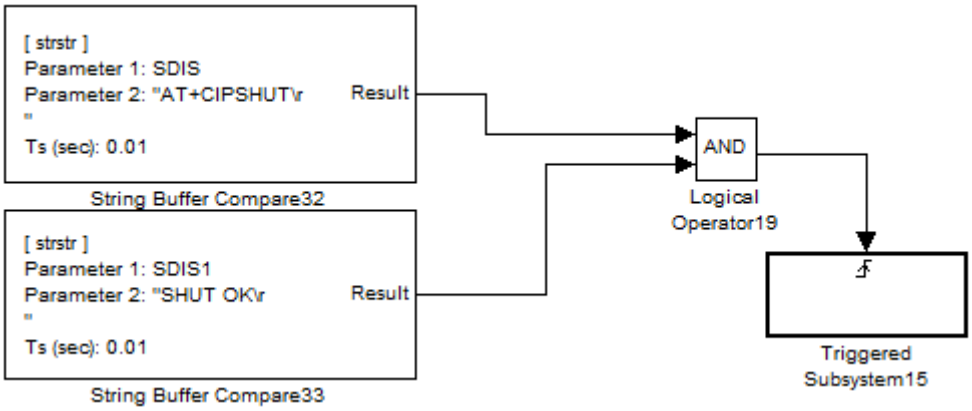
รูปที่ 3.82 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



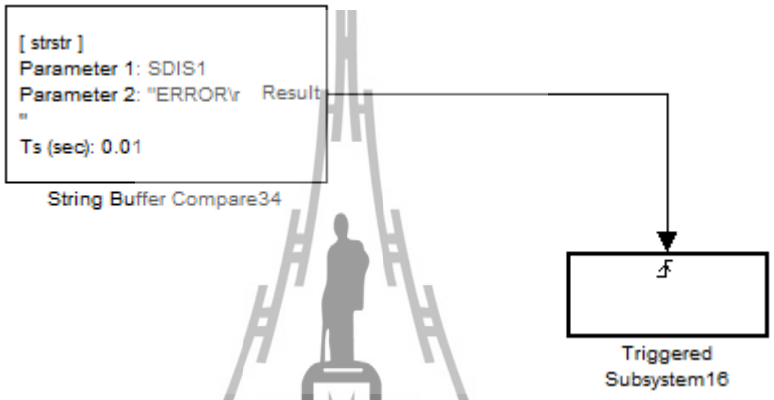
รูปที่ 3.83 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



รูปที่ 3.84 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)

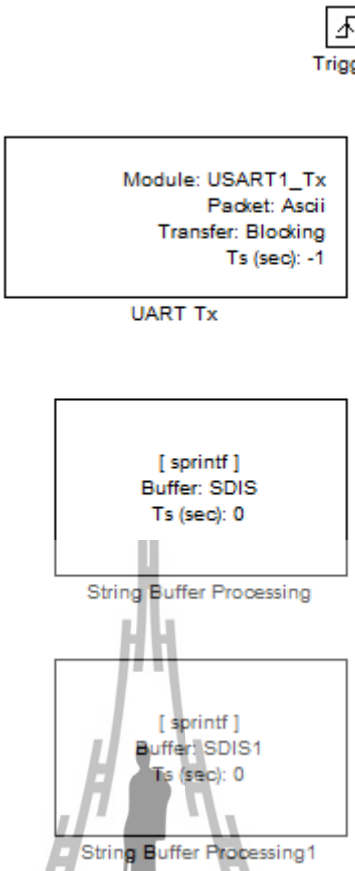


รูปที่ 3.85 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)



รูปที่ 3.86 บล็อกเปรียบเทียบค่าที่รับมาจากโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ (ต่อ)

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.87 บล็อกสำหรับส่งค่าออกไปติดต่อกับโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์

ตารางที่ 3.14 การกำหนดค่าพารามิเตอร์สำหรับบล็อก UART Tx ที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.71 ถึงรูปที่ 3.83

รูปที่	**UART Module**	**Transfer**	**Packet mode**	**Ascii format**	**End of packet**
3.71	1	Blocking	Ascii	'AT+COPS?\r\n'	None
3.72	1	Blocking	Ascii	'AT+CREG=1\r\n'	None
3.73	1	Blocking	Ascii	'AT+CGDCONT=1,\"IP\",\"internet\"\r\n'	None
3.74	1	Blocking	Ascii	'AT+CSTT=\"internet\"\r\n'	None
3.75	1	Blocking	Ascii	'AT+CIICR\r\n'	None
3.76	1	Blocking	Ascii	'AT+CIFSR\r\n'	None
3.77	1	Blocking	Ascii	'AT+CDNSCFG=\"202.183.255.20\",\"202.183.255.21\"\r\n'	None

มหาวิทยาลัยเทคโนโลยีสุรนารี

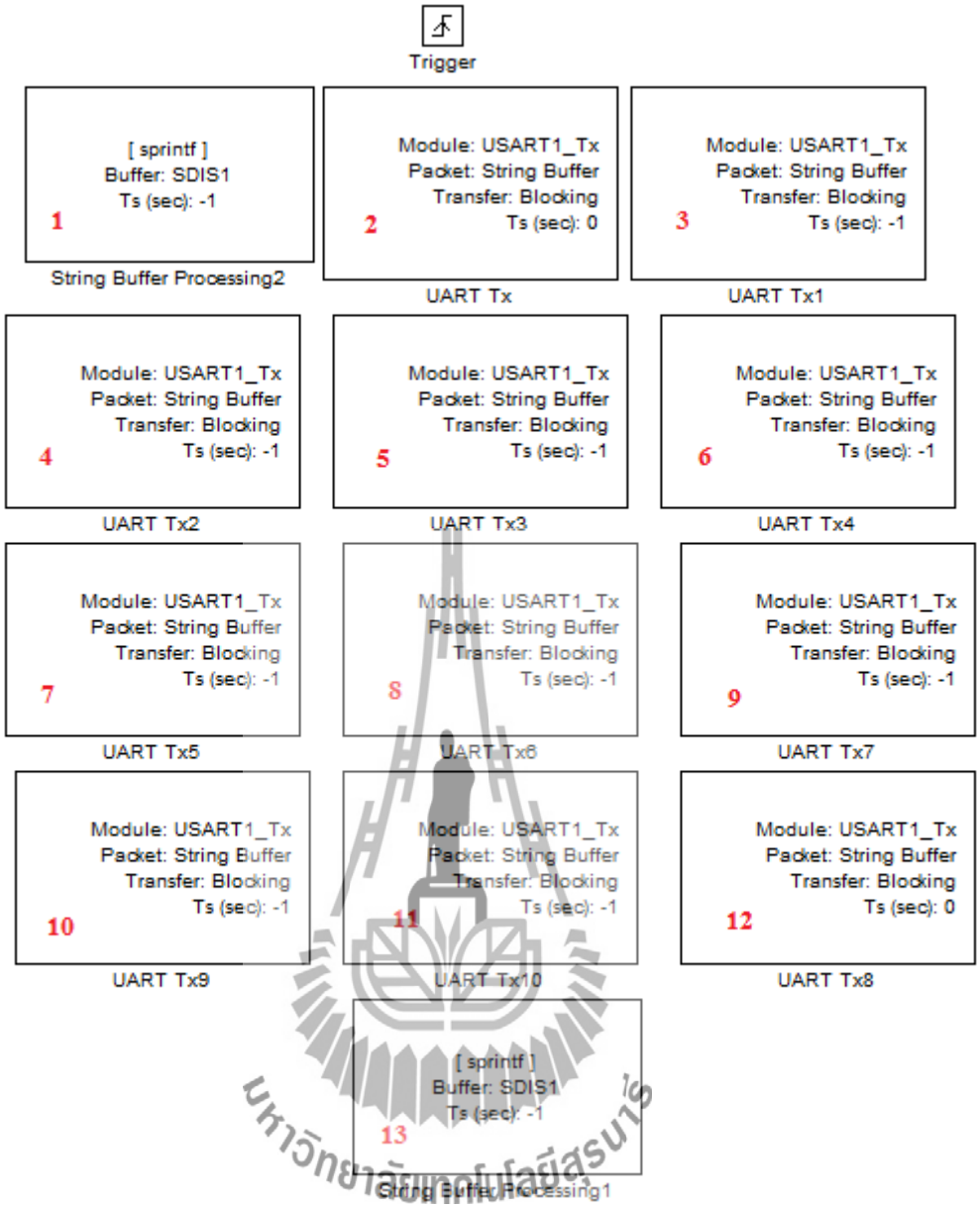
ตารางที่ 3.14 การกำหนดค่าพารามิเตอร์สำหรับบล็อก UART Tx ที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.71 ถึงรูปที่ 3.83 (ต่อ)

รูปที่	UART Module	Transfer	Packet mode	Ascii format	End of packet
3.78	1	Blocking	Ascii	'AT+CDNSORIP=0\r\n'	None
3.79	1	Blocking	Ascii	'AT+CIPSTATUS\r\n'	None
3.80	1	Blocking	Ascii	'AT+CIPSTART=\"TCP\",\"203.158.7.36\",\"80\"\r\n'	None
3.81	1	Blocking	Ascii	'AT+CIPSEND\r\n'	None
3.82	1	Blocking	Ascii	'AT+CIPCLOSE\r\n'	None
3.83	1	Blocking	Ascii	'AT+CIPSHUT\r\n'	None

ตารางที่ 3.15 การกำหนดค่าพารามิเตอร์สำหรับบล็อก String buffer processing (SDIS) ที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.71 ถึงรูปที่ 3.83

รูปที่	Function	String buffer	Printf format
3.71	sprintf	SDIS	'AT+COPS?\r\n'
3.72	sprintf	SDIS	'AT+CREG=1\r\n'
3.73	sprintf	SDIS	'AT+CGDCONT=1,\"IP\",\"internet\"\r\n'
3.74	sprintf	SDIS	'AT+CSTT=\"internet\"\r\n'
3.75	sprintf	SDIS	'AT+CIICR\r\n'
3.76	sprintf	SDIS	'AT+CIFSR\r\n'
3.77	sprintf	SDIS	'AT+CDNSCFG=\"202.183.255.20\",\"202.183.255.21\"\r\n'
3.78	sprintf	SDIS	'AT+CDNSORIP=0\r\n'
3.79	sprintf	SDIS	'AT+CIPSTATUS\r\n'
3.80	sprintf	SDIS	'AT+CIPSTART=\"TCP\",\"203.158.7.36\",\"80\"\r\n'
3.81	sprintf	SDIS	'AT+CIPSEND\r\n'
3.82	sprintf	SDIS	'AT+CIPCLOSE\r\n'
3.83	sprintf	SDIS	'AT+CIPSHUT\r\n'

มหาวิทยาลัยเทคโนโลยีสุรนารี



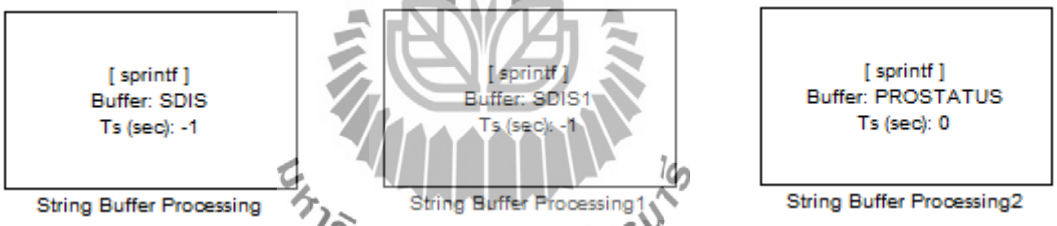
รูปที่ 3.88 บล็อกสำหรับส่งข้อมูลขึ้นเซิร์ฟเวอร์

ตารางที่ 3.16 การกำหนดค่าพารามิเตอร์สำหรับบล็อก String buffer processing ที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.84

บล็อกหมายเลข	Function	String buffer	Printf format
1	sprintf	SDIS1	'BEGINSEND'
13	sprintf	SDIS1	'GSM BACK'

ตารางที่ 3.17 การกำหนดค่าพารามิเตอร์สำหรับบล็อก UART Tx ที่อยู่ในบล็อก Triggered subsystem ของรูปที่ 3.84 (เลือก UART Module เป็น 1 เลือก Transfer เป็น Blocking เลือก Packet mode เป็น String buffer และเลือก End of packet เป็น None)

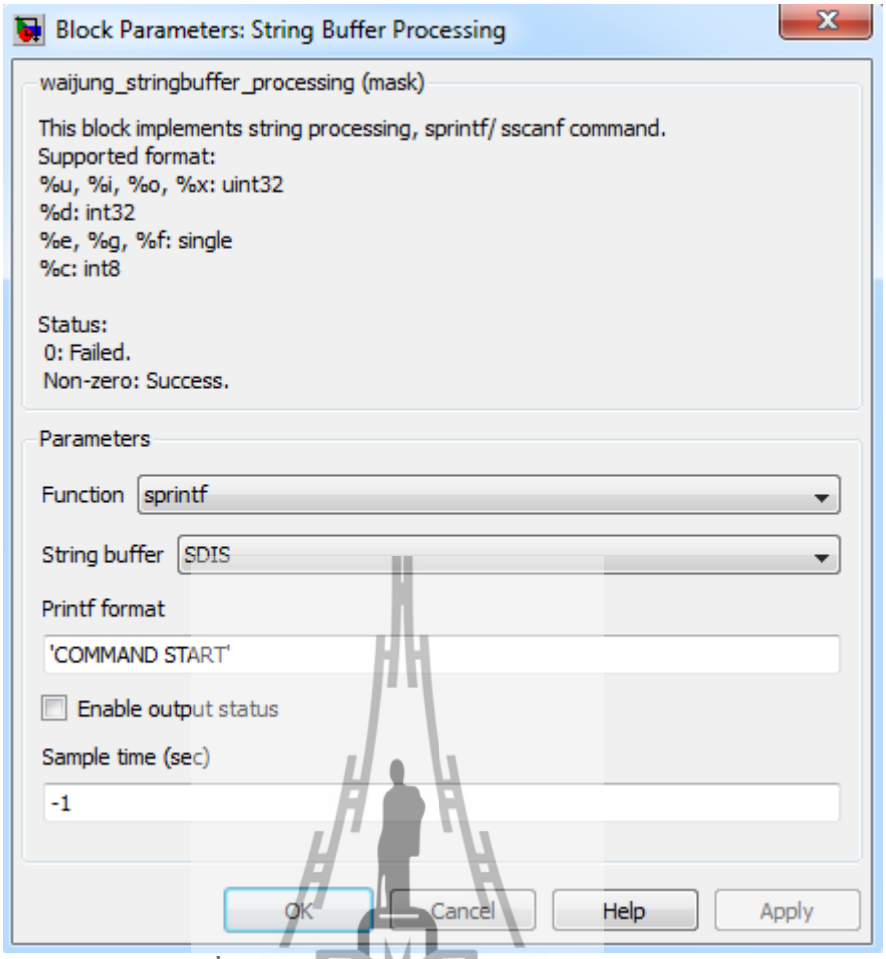
บล็อกหมายเลข	Buffer
2	SENDINTH
3	SENDINT
4	SENDINT1
5	SENDINT2
6	SENDINT3
7	SENDINT4
8	SENDINT5
9	SENDINT6
10	SENDINT7
11	SENDINT8
12	SENDINTL



(ก) (ข) (ค)

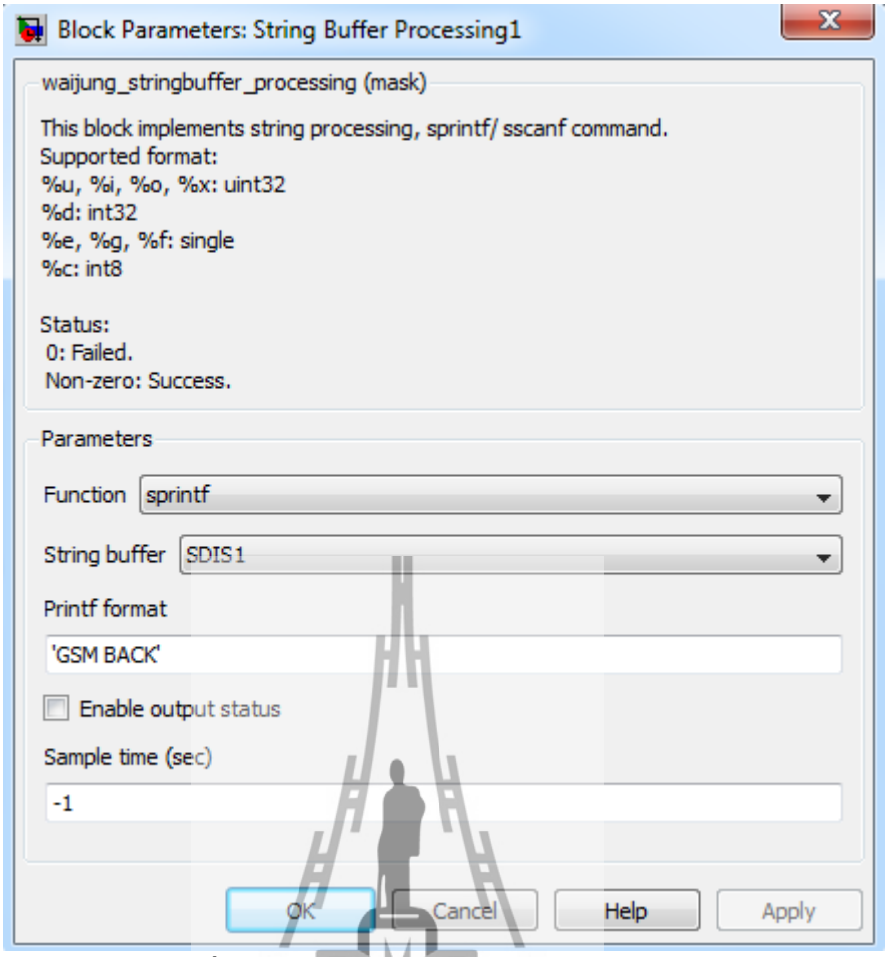
รูปที่ 3.89 (ก) (ข) และ (ค) String buffer processing

มหาวิทยาลัยเทคโนโลยีสุรนารี



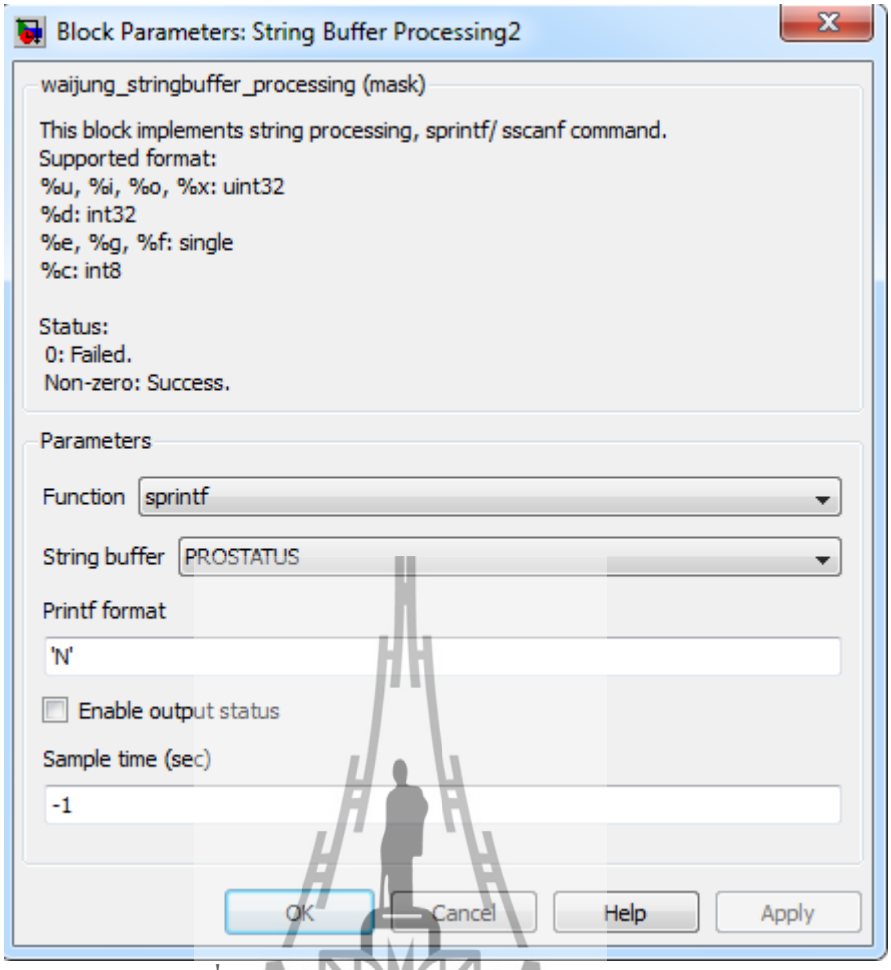
รูปที่ 3.90 การกำหนดค่า String buffer processing

มหาวิทยาลัยเทคโนโลยีสุรนารี

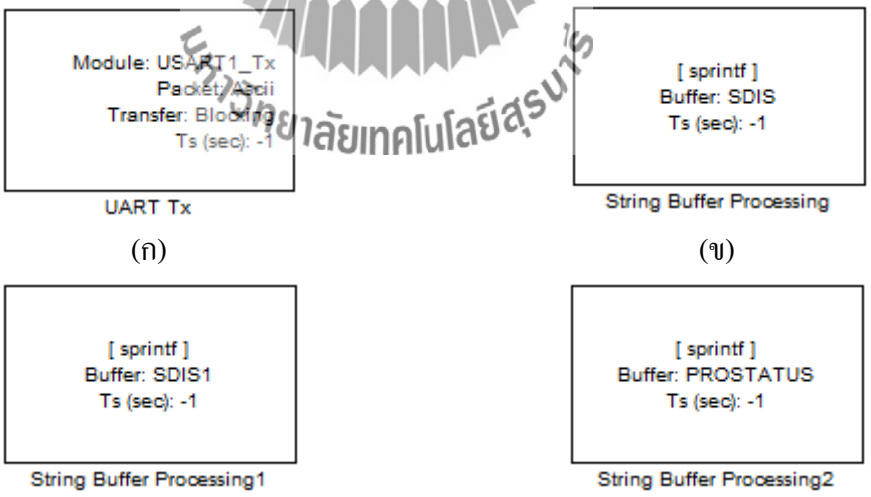


รูปที่ 3.91 การกำหนดค่า String buffer processing

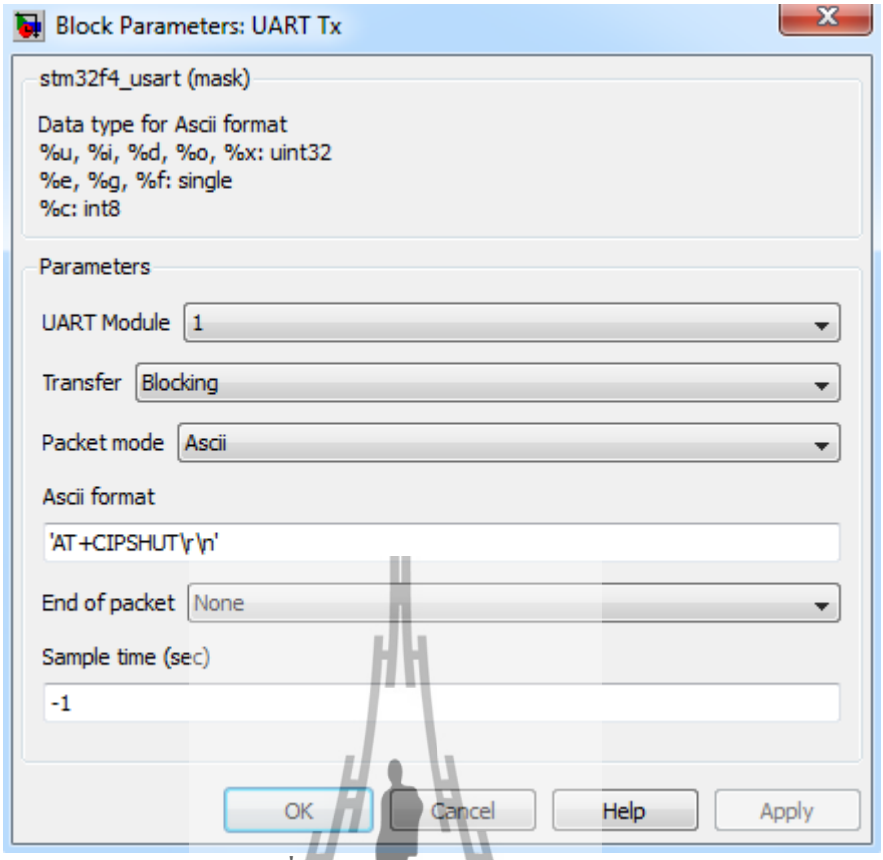
มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.92 การกำหนดค่า String buffer processing

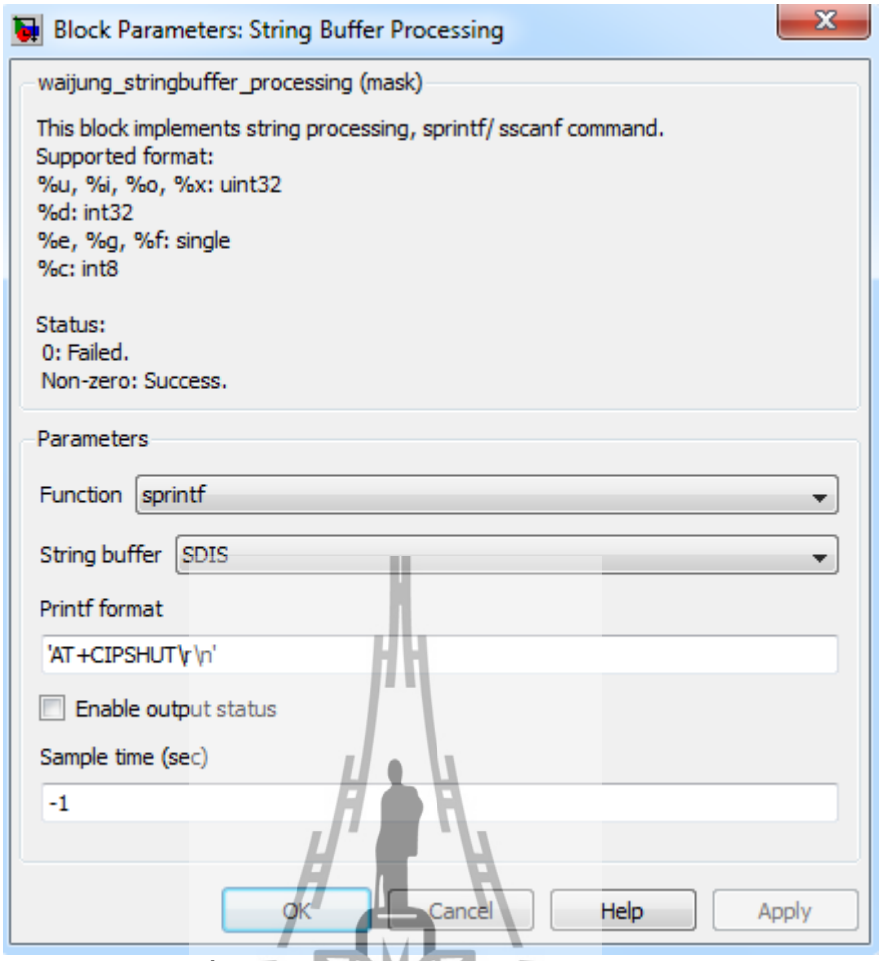


รูปที่ 3.93 (ก) UART Tx (ข) (ค) และ (ง) String buffer Processing



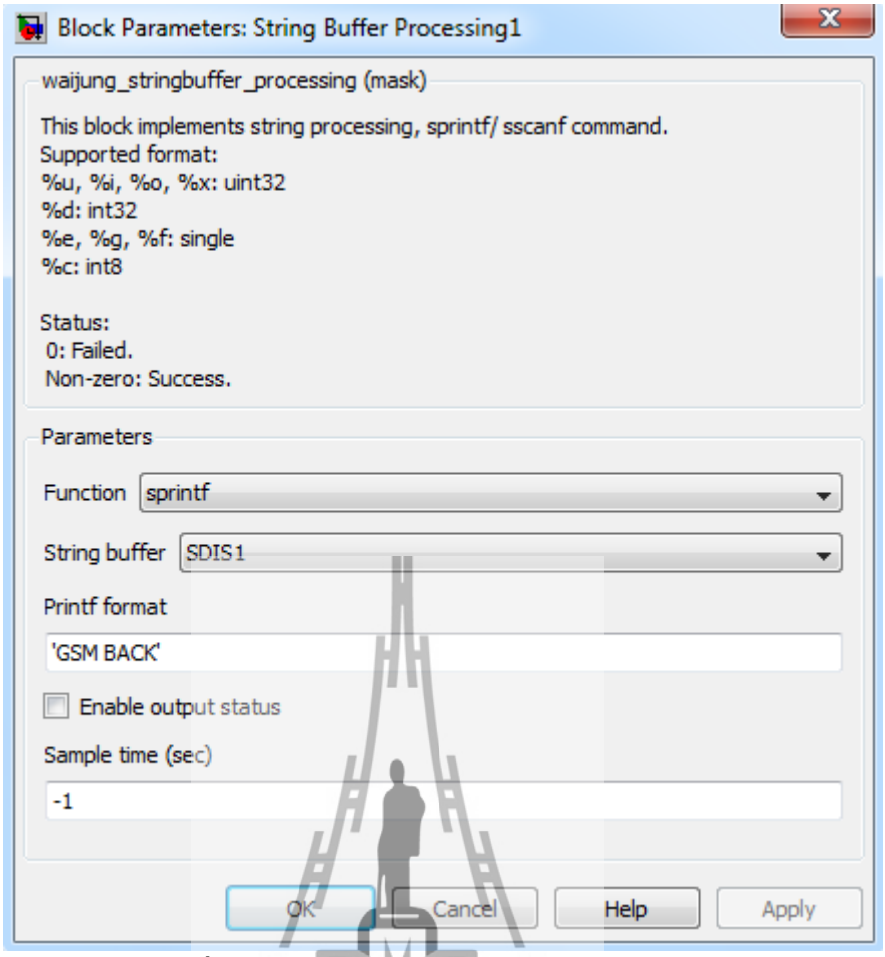
รูปที่ 3.94 การกำหนดค่า UART **Tx**

มหาวิทยาลัยเทคโนโลยีสุรนารี



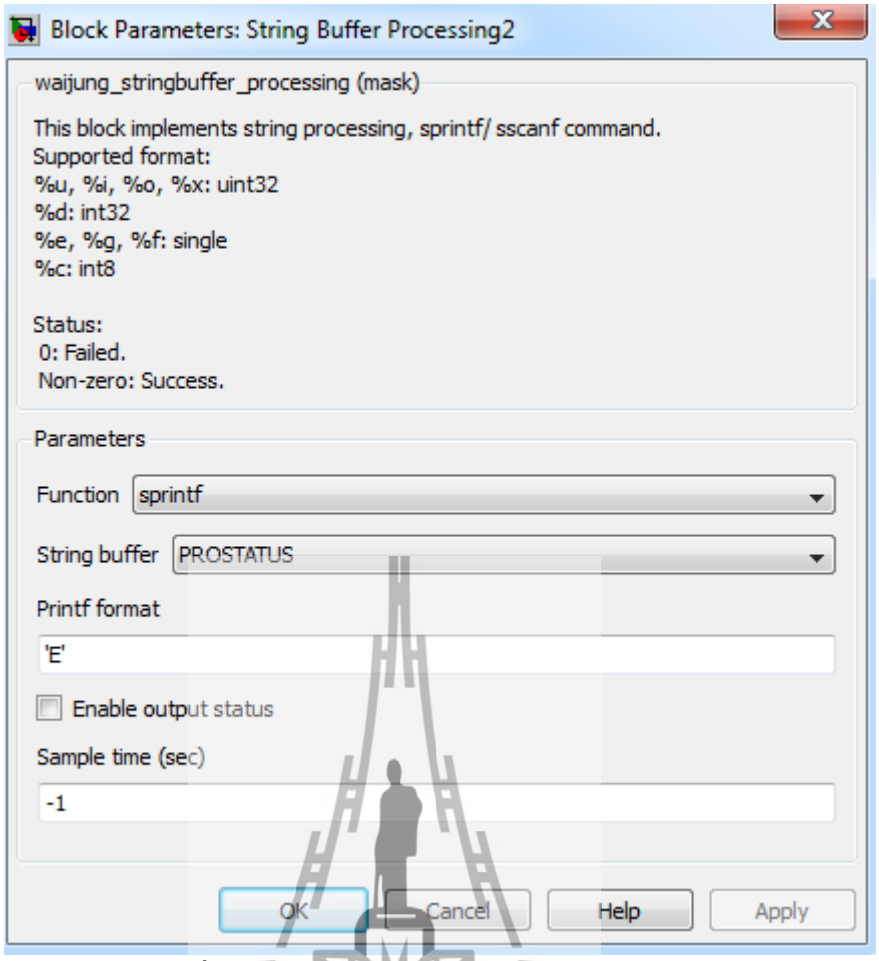
รูปที่ 3.95 การกำหนดค่า String buffer processing

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.96 การกำหนดค่า String buffer processing1

มหาวิทยาลัยเทคโนโลยีสุรนารี



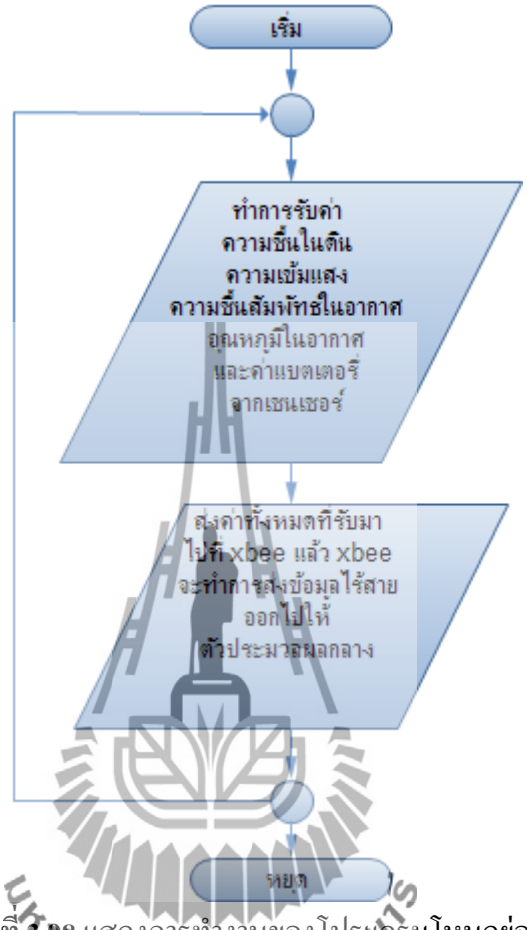
รูปที่ 3.97 การกำหนดค่า String buffer processing2

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.4.2 การออกแบบทางซอฟต์แวร์ในส่วนของโหนดย่อย (Router)

3.4.2.1 แผนภาพการทำงานของโปรแกรม

การทำงานของโหนดย่อยโดยรวมสามารถอธิบายได้ดังรูปที่ 3.98

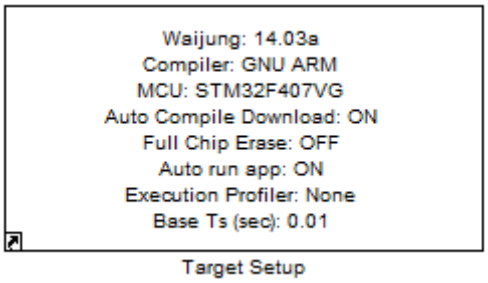


รูปที่ 3.98 แสดงการทำงานของโปรแกรมโหนดย่อย

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.4.2.2 การออกแบบโปรแกรม

1. ทำการลากบล็อก Target setup ดังรูปที่ 3.98เพื่อกำหนด Compiler ในโครงงานนี้เลือกใช้ GNU ARM

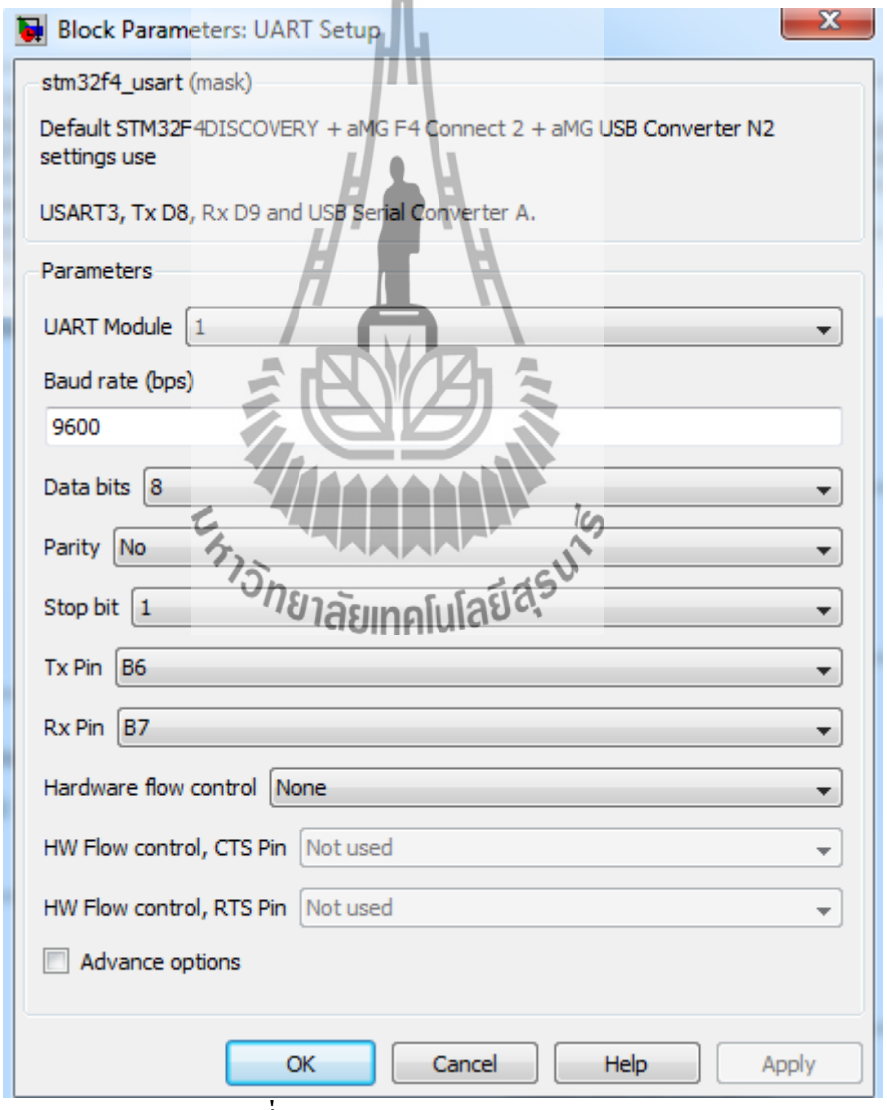


รูปที่ 3.99 Target Setup

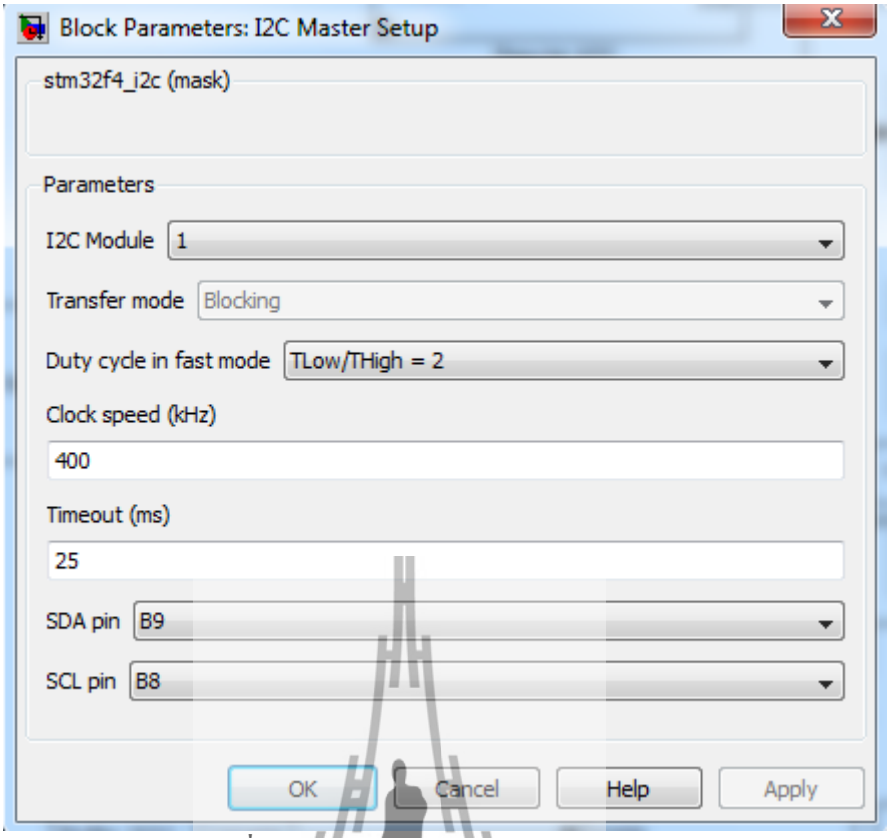
2. ทำการลากบล็อก UART setup และ I2C master setup ดังรูปที่ 3.99 (ก) และ (ข) พร้อมทั้งกำหนดค่าพารามิเตอร์ดังรูปที่ 3.100 และรูปที่ 3.101 ตามลำดับ

(ก) (ข)

รูปที่ 3.100 บล็อก UART setup และ I2C master Setup

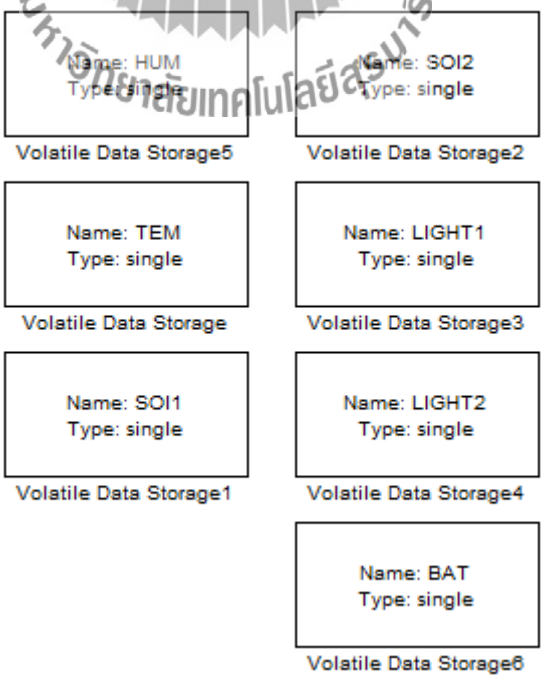


รูปที่ 3.101 การกำหนดค่า UART setup



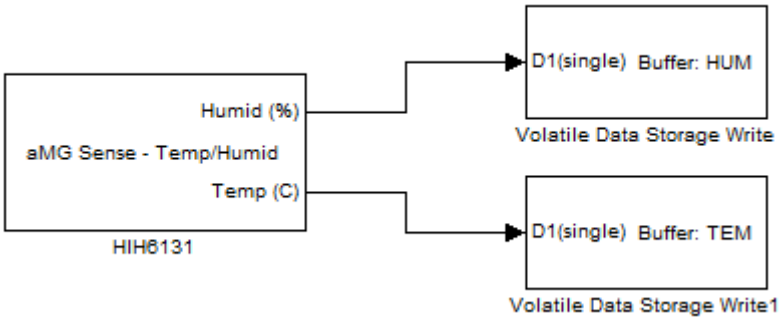
รูปที่ 3.102 การกำหนดค่า I2C master setup

3. ทำการลากบล็อก Volatile data storage พร้อมทั้งกำหนดประเภทและชื่อของตัวแปรดังรูปที่ 3.102



รูปที่ 3.103 บล็อกVolatile data storage

4. ทำการลากบล็อกสำหรับอ่านค่าอุณหภูมิและความชื้นในอากาศพร้อมทั้งเก็บลงตัวแปรดังรูปที่ 3.103



รูปที่ 3.104 บล็อกสำหรับอ่านค่าอุณหภูมิและความชื้นในอากาศ

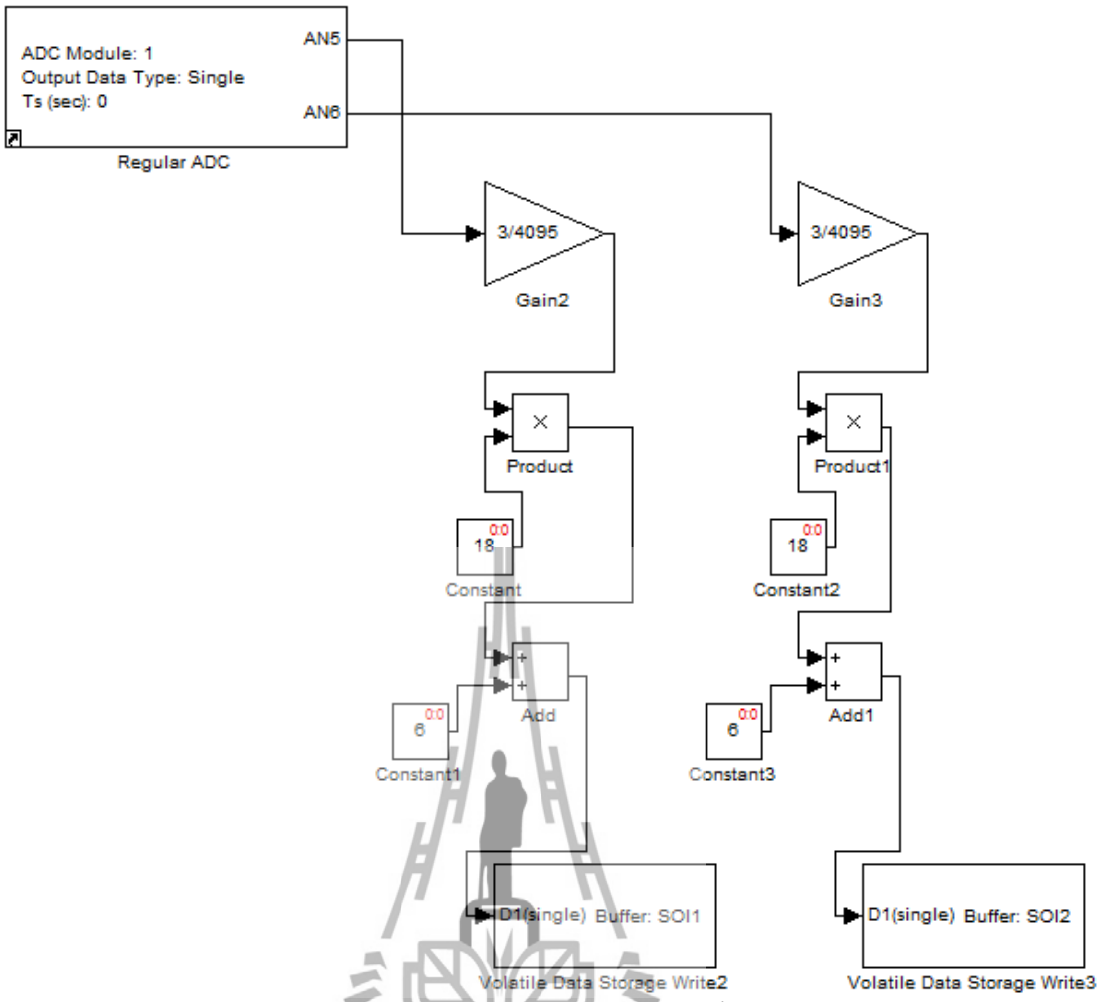
5. ทำการลากบล็อกตามสมการที่ (3.5)จะได้บล็อกสำหรับอ่านค่าความชื้นในดินดังรูปที่ 3.104

$$\text{ความชื้นในดิน (\%)} = (18 \times (A \times (3 / 4095))) + 6 \quad (3.5)$$

โดยที่

A คือค่าที่อ่านได้จากบล็อกแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.105 บล็อกสำหรับอ่านค่าความชื้นในดิน

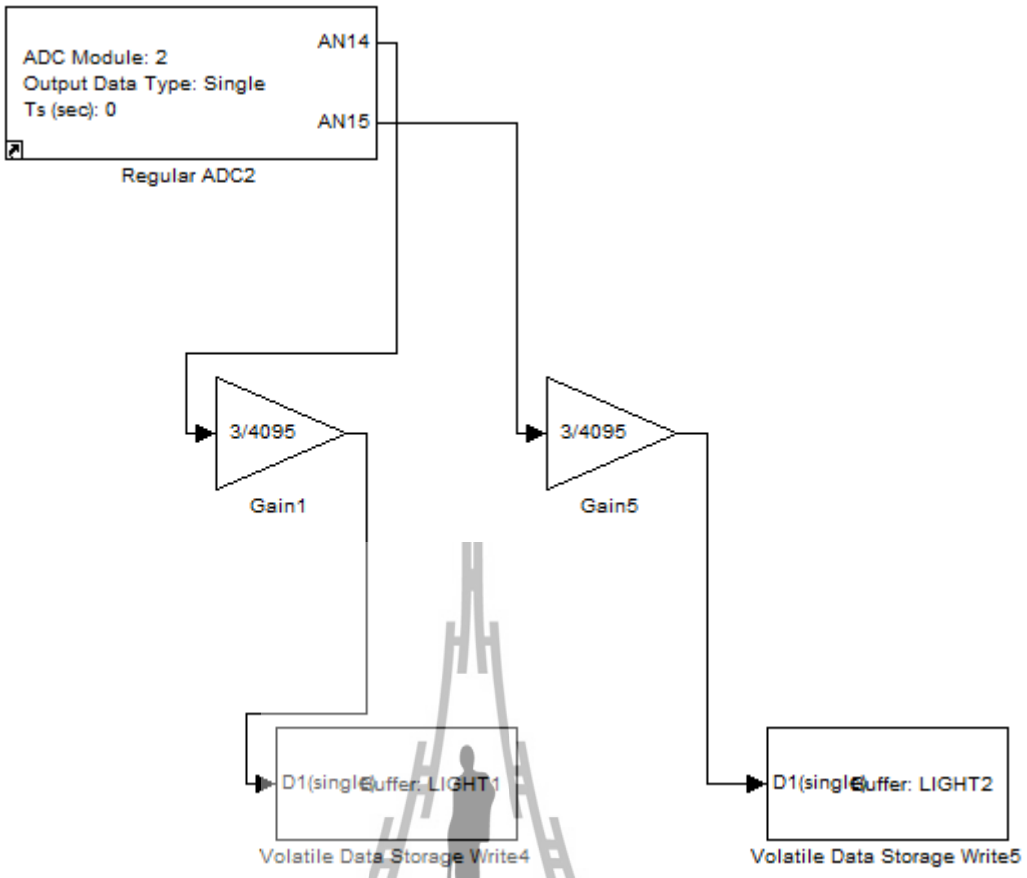
6. ทำการลากบล็อกตามสมการที่ (3.6)จะได้บล็อกสำหรับอ่านค่าความเข้มแสงดังรูปที่ 3.105

$$\text{ความเข้มแสง (โวลต์)} = A \times (3 / 4095) \tag{3.6}$$

โดยที่

A คือค่าที่อ่านได้จากบล็อกแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.106 บล็อกสำหรับอ่านค่าความเข้มแสง

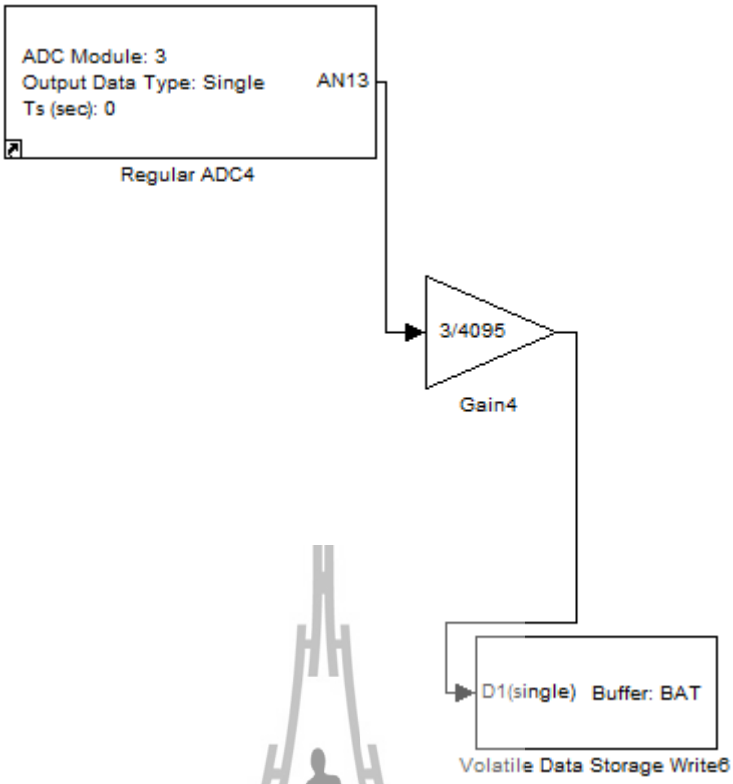
7. ทำการลากบล็อกตามสมการที่ (3.7)จะได้บล็อกสำหรับอ่านค่าแรงดันแบตเตอรี่ดังรูปที่ 3.106

แรงดันแบตเตอรี่ (โวลต์) =A × (3 / 4095) (3.7)

โดยที่

A คือค่าที่อ่านได้จากบล็อกแปลงสัญญาณแอนะล็อกเป็นดิจิตอล

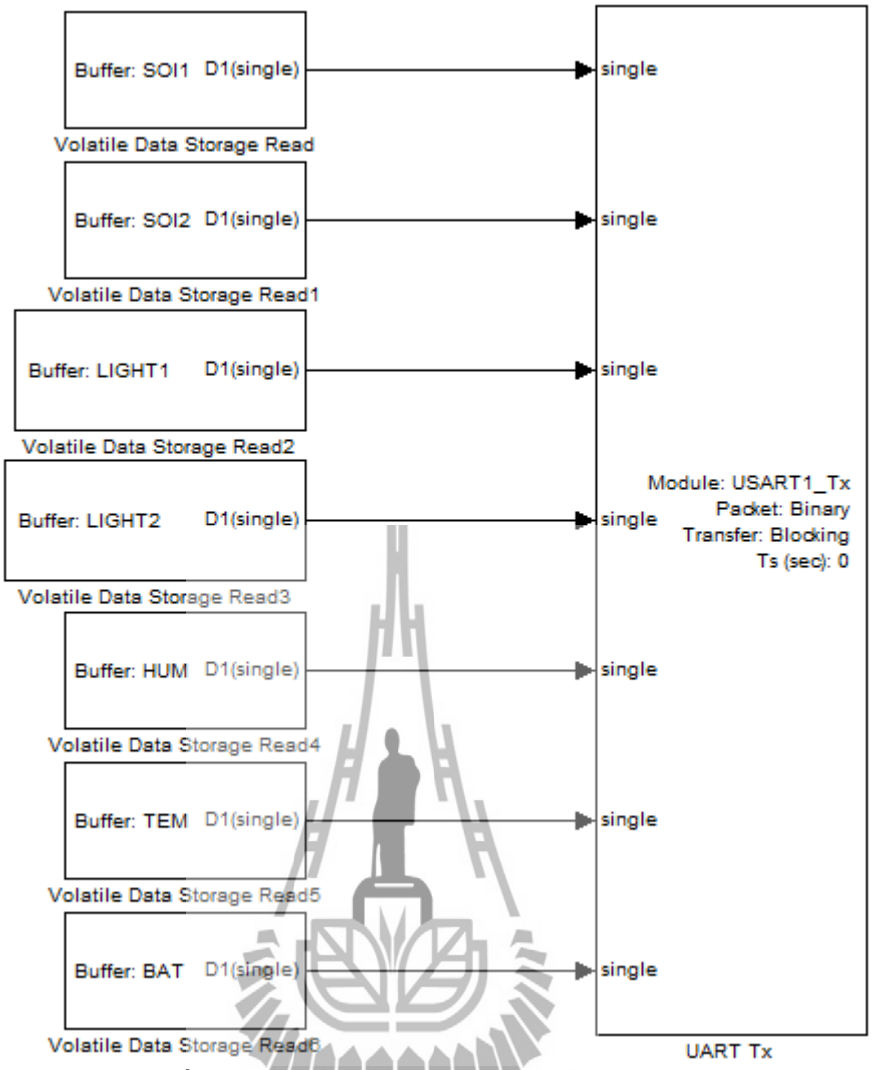
มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.107 บล็อกสำหรับอ่านค่าแรงดันแบตเตอรี่

8. ทำการลากบล็อกสำหรับส่งข้อมูลให้โคออร์ดิเนเตอร์ดังรูปที่ 3.107พร้อมทั้งทำการกำหนดค่าพารามิเตอร์โดยเลือกค่าจากตารางที่ 3.18 เช่น ถ้าโหนดนี้เป็นโหนดที่ 1 ก็ให้เลือกค่าพารามิเตอร์ของโหนดที่หนึ่งจากตารางที่ 3.18

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.108 บล็อกสำหรับส่งข้อมูลให้โคออร์ดิเนเตอร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตารางที่ 3.18 กำหนดค่าพารามิเตอร์สำหรับโหนด(ให้เลือก UART Module เป็น 1 เลือก Transfer เป็น Blocking เลือก Packet mode เป็น Binary เลือกจำนวนของเอาต์พุตเป็น Single จำนวน 7 พอร์ต)

โหนดที่	Binary header	Binary terminator
1	'2A 34'	'0D 09'
2	'29 34'	'0D 08'
3	'28 34'	'0D 07'
4	'27 34'	'0D 06'
5	'26 34'	'0D 05'
6	'25 34'	'0D 04'

3.4.3 การออกแบบทางซอฟต์แวร์ในส่วนของไฟล์ที่อยู่บนเซิร์ฟเวอร์

ในส่วนนี้จะใช้ภาษา PHP ในการเขียนโปรแกรมสำหรับรับค่าจากโคออร์ดิเนเตอร์แล้วเก็บลงฐานข้อมูล โดยโปรแกรมมีลักษณดังนี้

```
<?php

    $a = fopen("test.txt","a+");                    //เปิดไฟล์ชื่อ test.txt สำหรับเก็บข้อมูล
    fputs($a,$L1);                                  //เขียนค่าที่อยู่ในตัวแปร L1 ลงในไฟล์
    fputs($a,"\r\n");                               //เขียนค่าเพื่อทำการขึ้นบรรทัดใหม่ในไฟล์
    fputs($a,$L2);                                  //เขียนค่าที่อยู่ในตัวแปร L2 ลงในไฟล์
    fputs($a,"\r\n");                               //เขียนค่าเพื่อทำการขึ้นบรรทัดใหม่ในไฟล์
    fputs($a,$L3);                                  //เขียนค่าที่อยู่ในตัวแปร L3 ลงในไฟล์
    fputs($a,"\r\n");                               //เขียนค่าเพื่อทำการขึ้นบรรทัดใหม่ในไฟล์
    fputs($a,$L4);                                  //เขียนค่าที่อยู่ในตัวแปร L4 ลงในไฟล์
    fputs($a,"\r\n");                               //เขียนค่าเพื่อทำการขึ้นบรรทัดใหม่ในไฟล์
    fputs($a,$L5);                                  //เขียนค่าที่อยู่ในตัวแปร L5 ลงในไฟล์
    fputs($a,"\r\n");                               //เขียนค่าเพื่อทำการขึ้นบรรทัดใหม่ในไฟล์
    fputs($a,$L6);                                  //เขียนค่าที่อยู่ในตัวแปร L6 ลงในไฟล์
    fputs($a,"\r\n");                               //เขียนค่าเพื่อทำการขึ้นบรรทัดใหม่ในไฟล์
    fputs($a,$L7);                                  //เขียนค่าที่อยู่ในตัวแปร L7 ลงในไฟล์
    fputs($a,"\r\n");                               //เขียนค่าเพื่อทำการขึ้นบรรทัดใหม่ในไฟล์
    fputs($a,$L8);                                  //เขียนค่าที่อยู่ในตัวแปร L8 ลงในไฟล์
    fputs($a,"\r\n");                               //เขียนค่าเพื่อทำการขึ้นบรรทัดใหม่ในไฟล์
    fputs($a,$L9);                                  //เขียนค่าที่อยู่ในตัวแปร L9 ลงในไฟล์
    fputs($a,"\r\n");                               //เขียนค่าเพื่อทำการขึ้นบรรทัดใหม่ในไฟล์
    fclose($a);                                     //ทำการปิดไฟล์
    echo("successfully");          //ข้อความตอบกลับไปที่โคออร์ดิเนเตอร์ “successfully”

    ?>
```

หมายเหตุ ค่าตัวแปร L1 ถึง L9 จะถูกกำหนดเมื่อโคออร์ดิเนเตอร์ได้ทำการร้องขอหน้าเว็บนี้

บทที่ 4

ผลการทดลอง และ ทดสอบการใช้งานจริง

4.1 การทดสอบฮาร์ดแวร์และซอฟต์แวร์

<u>การทดลองที่ 1</u>การทดสอบการวัดค่าจากเซ็นเซอร์และส่งข้อมูลแบบไร้สาย

วัตถุประสงค์

1. เพื่อทดสอบการวัดค่าจากเซ็นเซอร์
2. เพื่อทดสอบการส่งข้อมูลไร้สายผ่านพอร์ตอนุกรมไปยังภาคประมวลผลสำหรับควบคุมการเปิดปิดน้ำ
3. เพื่อวัดระยะทางที่ไกลที่สุดที่ Xbeeสามารถส่งข้อมูลได้

ขั้นตอนการทดลอง

1. โปรแกรมให้ STM32F4Discovery ภาควัดค่าด้วยอุปกรณ์ต่างๆรับค่าจากเซ็นเซอร์โดยการแปลงค่าแอนะล็อกเป็นดิจิตอล
2. โปรแกรมให้ STM32F4Discovery ภาควัดค่าด้วยอุปกรณ์ต่างๆส่งค่าที่ได้ไปยังภาคประมวลผลและควบคุมการเปิดปิดน้ำโดยใช้ Xbee
3. เคลื่อนย้ายภาควัดค่าด้วยอุปกรณ์ต่างๆออกห่างจากภาคประมวลผลและควบคุมการเปิดปิดน้ำอย่างช้าๆจนกว่าจะหยุดส่งสัญญาณ
4. บันทึกผลการทดลอง

มหาวิทยาลัยเทคโนโลยีสุรนารี

ผลการทดลอง

ตารางที่ 4.1 ผลการทดลองส่งข้อมูลด้วยXbee

ระยะทาง (เมตร)	สถานะการส่งข้อมูล
20	✓
40	✓
60	✓
80	✓
100	✓
120	✓
140	✓
145	✗
150	✗

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ 4.1 สถานที่ใช้ทดสอบหาระยะการส่งข้อมูลของ Xbee

มหาวิทยาลัยเทคโนโลยีสุรนารี

สรุปผลการทดลอง

จากผลการทดลองสรุปได้ว่า ภาควัดค่าด้วยอุปกรณ์เซ็นเซอร์แบบต่างๆ ว่าสามารถอ่านค่าจากเซ็นเซอร์ได้และสามารถส่งข้อมูลไร้สายผ่านพอร์ตอนุกรมไปยังภาคประมวลผลสำหรับควบคุมการเปิดปิดน้ำได้ จากการทดลองพบว่า Xbeeที่ใช้งานมีความสามารถในการส่งได้ไกลที่สุดประมาณ 140 เมตร(ตามสเปคกำหนดว่าส่งได้ไกล 100 เมตร)

<u>การทดลองที่ 2</u>การทดลองบันทึกข้อมูลลง SD-Card และส่งข้อมูลขึ้นเซิร์ฟเวอร์

วัตถุประสงค์

1. เพื่อทดสอบการบันทึกข้อมูลที่ได้จากการตรวจวัดด้วยเซ็นเซอร์ลงหน่วยความจำ SD-Card

2. เพื่อทดสอบการส่งข้อมูลขึ้นเซิร์ฟเวอร์ด้วยโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์

ขั้นตอนการทดลอง

1. โปรแกรมให้ STM32F4Discovery ภาคประมวลผลและควบคุมการเปิดปิดน้ำ รับค่าจาก STM32F4Discovery ภาควัดค่าด้วยอุปกรณ์วัดค่าต่างๆด้วย Xbee

2. โปรแกรมให้ STM32F4Discovery ภาคประมวลผลและควบคุมการเปิดปิดน้ำนำข้อมูลที่ได้รับจากภาควัดค่าด้วยอุปกรณ์วัดค่าต่างๆบันทึกลงบนหน่วยความจำ SD-Card ด้วย Data logger

3. โปรแกรมให้ STM32F4Discovery ภาคประมวลผลและควบคุมการเปิดปิดน้ำ ข้อมูลที่ได้รับจากภาควัดค่าด้วยอุปกรณ์วัดค่าต่างๆส่งขึ้นเซิร์ฟเวอร์ผ่านพอร์ตอนุกรมด้วย UART ผ่านโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์โดยระบุปลายทางไปที่เซิร์ฟเวอร์ของมหาวิทยาลัยเทคโนโลยีสุรนารีมี URL เป็น http://203.158.7.36/142001/CASAVA1/index.php

4. เปิดไฟล์ data.txt ในหน่วยความจำ SD-Card เพื่อดูข้อมูลที่บันทึกไว้

5. ทำการตรวจสอบข้อมูลที่ URL: http://203.158.7.36/142001/CASAVA1/open.php เพื่อดูข้อมูลที่ทำการส่งขึ้นเซิร์ฟเวอร์

6. เปรียบเทียบข้อมูลจาก SD-Card และจาก http://203.158.7.36/142001/CASAVA1/open.php

มหาวิทยาลัยเทคโนโลยีสุรนารี

ผลการทดลอง

ในรูปที่ 4.2 และรูปที่ 4.3 แสดงข้อมูลที่อยู่ในไฟล์ data.txt ในหน่วยความจำ SD-Card และข้อมูลบนเซิร์ฟเวอร์ที่URL: http://203.158.7.36/142001/CASAVA1/open.php และในตารางที่ 4.2 อธิบายความหมายของค่าต่างๆในรูปที่ 4.2 และรูปที่ 4.3

NODE3:S5=14.92,S6=14.05,L5=3.30,L6=3.30

NODE4:S7=11.24,S8=12.67,L7=3.29,L8=3.30

NODE5:S9=20.68,S10=21.60,L9=3.30,L10=1.44

NODE6:S11=16.04,S12=21.66,L11=3.30,L12=1.30

SAVG=17.71

SPUMP=1

26/3/14,0:27

NODE1:S1=20.74,S2=19.86,L1=0.05,L2=2.68,H1=57.74,T1=26.55,B1=12.03

NODE2:S3=20.14,S4=18.73,L3=3.00,L4=2.95,H2=54.15,T2=26.78,B2=11.72

NODE3:S5=14.93,S6=14.05,L5=3.30,L6=3.30

NODE4:S7=11.29,S8=12.69,L7=3.29,L8=3.30

NODE5:S9=20.56,S10=21.49,L9=3.29,L10=1.43

NODE6:S11=16.08,S12=21.21,L11=3.30,L12=1.30

SAVG=17.65

SPUMP=1

26/3/14,0:37

NODE1:S1=20.78,S2=20.11,L1=0.10,L2=2.68,H1=58.09,T1=26.66,B1=12.03

NODE2:S3=19.98,S4=18.76,L3=3.00,L4=2.91,H2=54.49,T2=26.85,B2=11.71

NODE3:S5=14.93,S6=14.05,L5=3.30,L6=3.30

NODE4:S7=11.28,S8=12.70,L7=3.29,L8=3.30

NODE5:S9=20.62,S10=21.72,L9=3.30,L10=1.44

NODE6:S11=16.05,S12=21.37,L11=3.30,L12=1.30

SAVG=17.70

SPUMP=1

รูปที่ 4.2 ข้อมูลที่อยู่ในไฟล์ data.txt ในหน่วยความจำ SD-Card

SPUMP=1
26/3/14,0:17
NODE1:S1=20.84,S2=19.93,L1=0.04,L2=2.68,H1=57.34,T1=26.40,B1=12.05
NODE2:S3=20.04,S4=18.83,L3=2.95,L4=2.96,H2=53.84,T2=26.71,B2=11.71
NODE3:S5=14.92,S6=14.05,L5=3.30,L6=3.30
NODE4:S7=11.24,S8=12.67,L7=3.29,L8=3.30
NODE5:S9=20.68,S10=21.60,L9=3.30,L10=1.44
NODE6:S11=16.04,S12=21.66,L11=3.30,L12=1.30
SAVG=17.71
SPUMP=1
26/3/14,0:27
NODE1:S1=20.74,S2=19.86,L1=0.05,L2=2.68,H1=57.74,T1=26.55,B1=12.03
NODE2:S3=20.14,S4=18.73,L3=3.00,L4=2.95,H2=54.15,T2=26.78,B2=11.72
NODE3:S5=14.93,S6=14.05,L5=3.30,L6=3.30
NODE4:S7=11.29,S8=12.69,L7=3.29,L8=3.30
NODE5:S9=20.56,S10=21.49,L9=3.29,L10=1.43
NODE6:S11=16.08,S12=21.21,L11=3.30,L12=1.30
SAVG=17.65
SPUMP=1
26/3/14,0:37
NODE1:S1=20.78,S2=20.11,L1=0.10,L2=2.68,H1=58.09,T1=26.66,B1=12.03
NODE2:S3=19.98,S4=18.76,L3=3.00,L4=2.91,H2=54.49,T2=26.85,B2=11.71
NODE3:S5=14.93,S6=14.05,L5=3.30,L6=3.30
NODE4:S7=11.28,S8=12.70,L7=3.29,L8=3.30
NODE5:S9=20.62,S10=21.72,L9=3.30,L10=1.44
NODE6:S11=16.05,S12=21.37,L11=3.30,L12=1.30
SAVG=17.70
SPUMP=1
26/3/14,0:47
NODE1:S1=20.70,S2=19.87,L1=0.10,L2=2.67,H1=58.91,T1=26.20,B1=12.04
NODE2:S3=20.03,S4=18.75,L3=2.98,L4=2.94,H2=55.21,T2=26.55,B2=11.69
NODE3:S5=14.95,S6=14.03,L5=3.30,L6=3.30
NODE4:S7=11.28,S8=12.73,L7=3.30,L8=3.30
NODE5:S9=21.02,S10=21.72,L9=3.30,L10=1.43
NODE6:S11=16.38,S12=21.55,L11=3.29,L12=1.30
SAVG=17.75
SPUMP=1

มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปที่ 4.3 ข้อมูลบนเซิร์ฟเวอร์ที่ URL: http://203.158.7.36/142001/CASAVA1/open.php

ตารางที่ 4.2 ความหมายของค่าต่างๆในรูปที่ 4.2 และรูปที่ 4.3

สัญลักษณ์	ความหมาย
26/3/14,0:27	วันที่ 26 เดือน มีนาคม ค.ศ. 2014 เวลา 0:27 น.
S	ค่าความชื้นในดินจากเซ็นเซอร์วัดความชื้นในดิน
L	ค่าความเข้มแสงจากเซ็นเซอร์วัดความเข้มแสง
H	ค่าความชื้นสัมพัทธ์ในอากาศ
T	ค่าอุณหภูมิในอากาศ
B	ค่าแบตเตอรี่
SAVG	ค่าเฉลี่ยของความชื้นในดินทั้งหมด
SPUMP	ค่าสถานะปั๊ม(SPUMP = 1 คือเปิด,SPUMP = 0 คือปิด)

สรุปผลการทดลอง

จากผลการทดลองจะเห็นได้ว่าข้อมูลที่อยู่ในไฟล์ data.txt ในหน่วยความจำ SD-Cardและข้อมูลบนเซิร์ฟเวอร์ที่ URL: http://203.158.7.36/142001/ CASAVA1/open.php ตรงกันทุกประการ จึงสรุปผลการทดลองได้ว่าอุปกรณ์ที่ออกแบบขึ้นสามารถบันทึกข้อมูลลง SD-Card และส่งข้อมูลขึ้นเซิร์ฟเวอร์ด้วยโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ได้อย่างถูกต้อง

มหาวิทยาลัยเทคโนโลยีสุรนารี

<u>การทดลองที่ 3</u> การทดสอบการเปิด-ปิดน้ำ

วัตถุประสงค์

1 . เพื่อทดสอบการควบคุมการทำงานของรีเลย์ (Relay)

2 . เพื่อทดสอบการใช้รีเลย์ควบคุมการทำงานของเครื่องสูบน้ำ

3. เพื่อทดสอบการใช้งานวงจรเรียงกระแส ทวีแรงดัน และขับกระแสสำหรับโซลินอยด์วาล์ว

ขั้นตอนการทดลอง

1 . โปรแกรม STM32F4Discovery ให้รับค่าจากเซ็นเซอร์วัดความชื้นในดิน 1 จุด

2 โปรแกรม STM32F4Discovery ให้ทำการเปรียบเทียบค่าความชื้นจากเซ็นเซอร์วัดความชื้นในดิน (EC-5) ถ้าต่ำกว่าค่าอ้างอิงให้รีเลย์ทำงาน(ค่าอ้างอิงถูกกำหนดไว้ที่ 18%)

3. โปรแกรม STM32F4Discovery ให้ทำการเปรียบเทียบค่าความชื้นในดินถ้าสูงกว่าค่าอ้างอิงให้รีเลย์หยุดทำงาน

4. ต่อรีเลย์เข้ากับเครื่องสูบน้ำขนาด 750 วัตต์ 220 โวลต์ 50 เฮิรตซ์

5 . นำเซ็นเซอร์วัดความชื้นในดินไปทำการจุ่มน้ำ (จำลองสถานะดินเปียก) และเช็ดให้แห้ง (จำลองสถานะดินแห้ง) สลับกัน

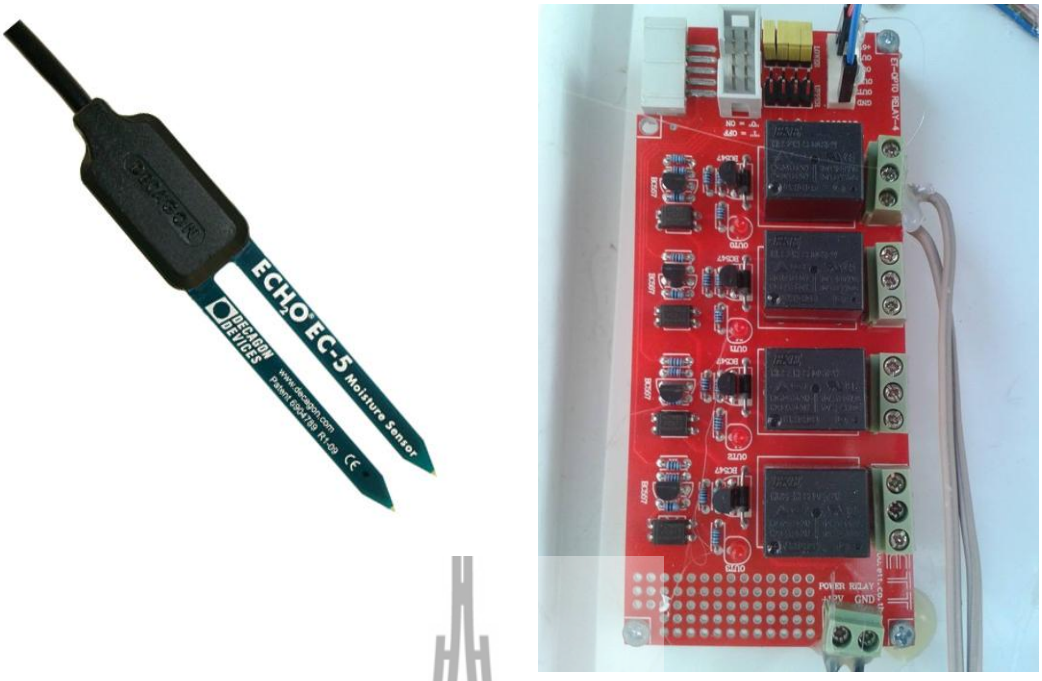
6 . สังเกตและบันทึกผล

7 . เปลี่ยนจากเครื่องสูบน้ำเป็นโซลินอยด์วาล์ว โดยต่อโซลินอยด์วาล์วเข้ากับวงจรขับโซลินอยด์วาล์วกระแสสลับขนาด 24 โวลต์ 50 เฮิรตซ์

8. ทำซ้ำตามขั้นตอนที่ 5 และขั้นตอนที่ 6

รูปที่ 4.4 และรูปที่ 4.5 แสดงอุปกรณ์ที่ใช้ในการทดลองที่ 3

มหาวิทยาลัยเทคโนโลยีสุรนารี



(ก) (ข)

(ที่มา : http://www.ictinternational.com)

รูปที่ 4.4 (ก)เซ็นเซอร์วัดความชื้นในดิน(ข) รีเลย์ที่ใช้ต่อเข้ากับเครื่องสูบน้ำ

(ก) (ข)

รูปที่ 4.5 (ก)ปั้มน้ำ (ข) โซลินอยด์วาล์วและวงจรสำหรับขับโซลินอยด์วาล์ว

มหาวิทยาลัยเทคโนโลยีสุรนารี

ผลการทดลอง

ตารางที่ 4.3 การทำงานของปั้มน้ำในสภาวะต่างๆ

การทดลอง 5 ครั้ง ในสถานะต่างๆ	**สถานะเครื่องสูบน้ำ**	
	ปิด	**เปิด**
ทดลองครั้งที่ 1 (เปียก)	✓	-
ทดลองครั้งที่ 2 (แห้ง)	-	✓
ทดลองครั้งที่ 3 (เปียก)	✓	-
ทดลองครั้งที่ 4 (แห้ง)	-	✓
ทดลองครั้งที่ 5 (เปียก)	✓	-

ตารางที่ 4.4 การทำงานของโซลินอยด์วาล์วสภาวะต่างๆ

การทดลอง 5 ครั้ง ในสถานะต่างๆ	**โซลินอยด์วาล์ว**	
	ปิด	**เปิด**
ทดลองครั้งที่ 1 (เปียก)	✓	-
ทดลองครั้งที่ 2 (แห้ง)	-	✓
ทดลองครั้งที่ 3 (เปียก)	✓	-
ทดลองครั้งที่ 4 (แห้ง)	-	✓
ทดลองครั้งที่ 5 (เปียก)	✓	-

หมายเหตุ ในตารางที่ 4.3 และ 4.4 คำว่าเปียก หมายถึง นำเซ็นเซอร์จุ่มน้ำ (จำลองสถานะดินเปียก) และคำว่าแห้ง หมายถึงเช็ดเซ็นเซอร์ให้แห้ง(จำลองสถานะดินแห้ง)

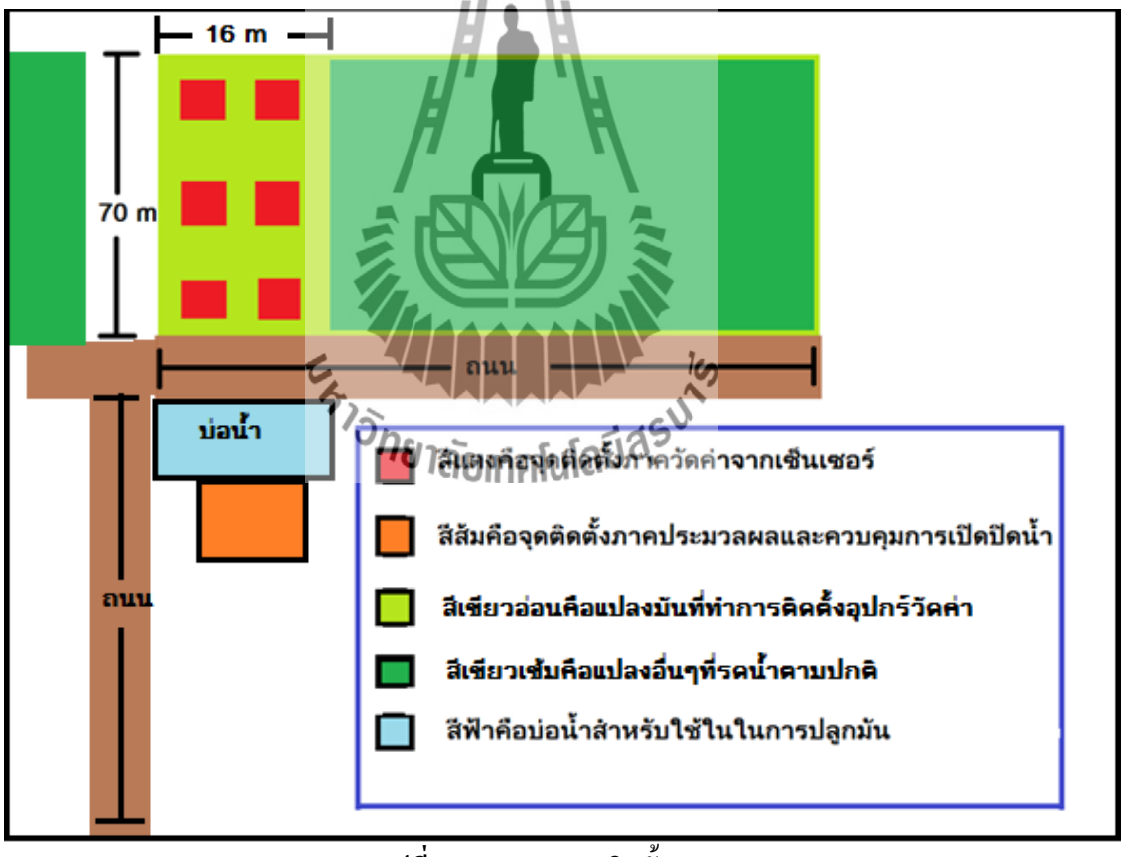
มหาวิทยาลัยเทคโนโลยีสุรนารี

สรุปผลการทดลอง

จากการทดสอบจะเห็นว่า สามารถเปิดปิดน้ำได้ตามต้องการได้ในทั้งสองกรณีดังนั้น สรุปผลการทดลองว่ารีเลย์ วงจรขับโซลินอยด์วาล์วกระแสสลับขนาด 24 โวลต์ 50 เฮิรตซ์ สามารถใช้งานได้ตามวัตถุประสงค์ทุกประการ

4.2 การติดตั้งระบบสำหรับการใช้งานจริง

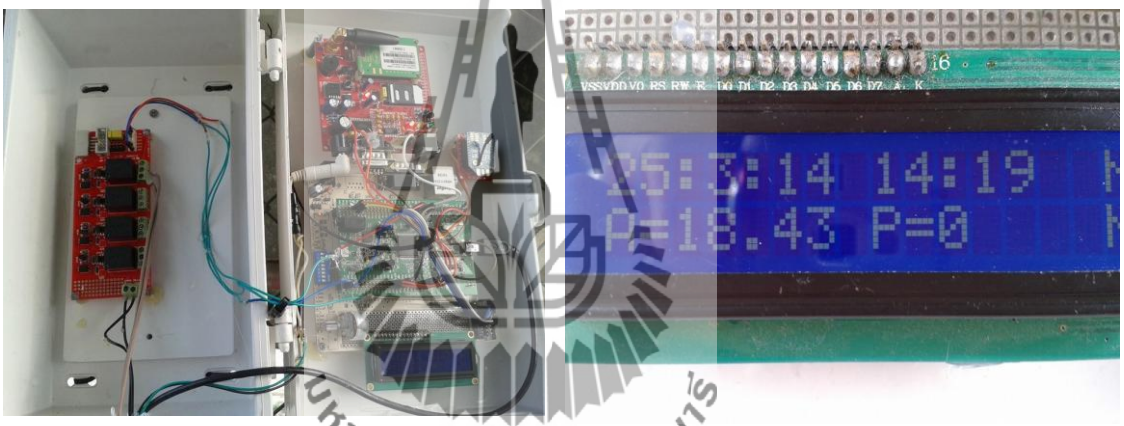
โครงงานนี้ได้ถูกติดตั้งในไร่มันสำปะหลังที่มีความยาว 70 เมตรและความกว้าง 16 เมตรในบริเวณฟาร์มมหาวิทยาลัยเทคโนโลยีสุรนารีโดยได้ทำการติดตั้งอุปกรณ์วัดค่าต่างๆทั้งหมดจำนวน 6 จุด เพื่อให้ได้ค่าความชื้นในดินเฉลี่ยครอบคลุมทั้งพื้นที่ และในส่วนของภาคประมวลผลและควบคุมการเปิดปิดน้ำติดตั้งอยู่บริเวณใกล้บ่อน้ำเพื่อให้สามารถควบคุมเครื่องสูบน้ำได้แสดงดังรูปที่ 4.6 รูปที่ 4.7 และรูปที่ 4.8



รูปที่ 4.6แผนภาพจุดติดตั้งระบบ

(ก) (ข)

รูปที่ 4.7 (ก) สถานที่ติดตั้งจริงบริเวณฟาร์มมหาวิทยาลัยเทคโนโลยีสุรนารี(ข) บ่อน้ำ



(ก) (ข)

รูปที่ 4.8 (ก)ชุดวงจรไดออร์ดิเนเตอร์(ข) จอแสดงผล

จากรูปที่ 4.9 เป็นการติดตั้งโหนดย่อยเพื่อทำการอ่านค่าจากเซ็นเซอร์วัดความชื้นในดิน เซ็นเซอร์วัดความเข้มแสง และเซ็นเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ในอากาศโดยจะทำการติดตั้งเซ็นเซอร์วัดความชื้นในดิน 2 ตัวต่อ 1โหนดย่อยโดยระดับความลึกในการฝังเซ็นเซอร์วัดความชื้นในดินตัวที่ 1 จะฝังลึกลงไป 15 เซนติเมตร และตัวที่ 2 จะฝังลึกลงไป 25 เซนติเมตร ดังรูปที่ 4.10 (ข) และทำการติดตั้งเซ็นเซอร์วัดความเข้มแสงดังรูปที่ 4.10 (ก) จากนั้นทำการติดตั้งโซล่าเซลล์ขนาด 20 วัตต์ และแบตเตอรี่ขนาด 12 โวลต์สำหรับเป็นแหล่งพลังงานให้กับโหนดย่อย

รูปที่ 4.9การติดตั้งโหนดย่อยเพื่อวัดค่าจากเซ็นเซอร์วัดความชื้นในดิน เซ็นเซอร์วัดความเข้มแสง และเซ็นเซอร์วัดอุณหภูมิและความชื้นสมพัทธ์ในอากาศ

มหาวิทยาลัยเทคโนโลยีสุรนารี

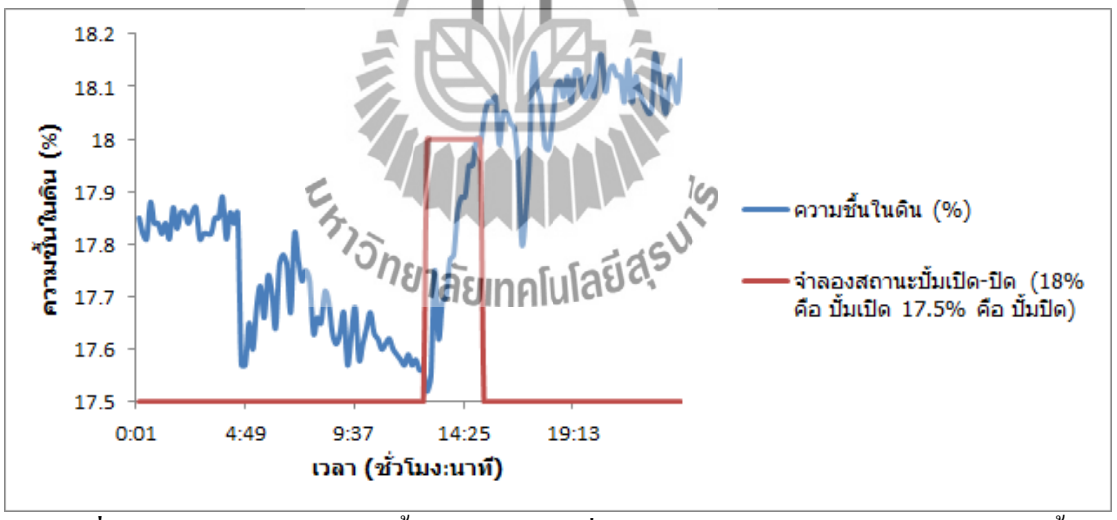
(ก) (ข)

รูปที่ 4.10 (ก)การติดตั้งเซ็นเซอร์แสง(ข) การติดตั้งเซ็นเซอร์วัดความชื้นในดิน

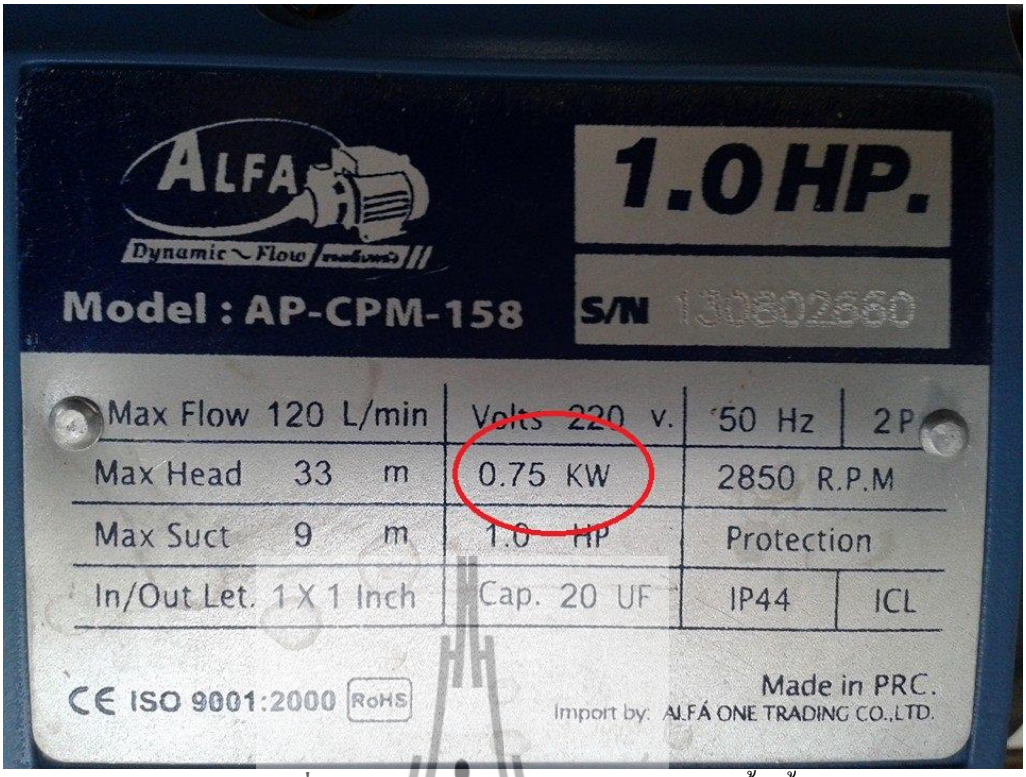
4.3ผลการให้น้ำ

การคำนวณค่าไฟฟ้าและปริมาณน้ำที่ใช้

จากข้อมูลของวันที่ 22 เดือน มีนาคม ค.ศ. 2014 ซึ่งเป็นช่วงเวลาในการทำงานของปั้มน้ำ (ปั้มเปิด) ตั้งแต่เวลา 12.49 ถึง 15.19 น. และแสดงค่าของความชื้นในดินเฉลี่ยได้ดังกราฟในรูปที่ 4.11 และในการใช้งานจริงวงรอบของการให้น้ำคือสัปดาห์ละ 1 ครั้ง



รูปที่ 4.11กราฟเปอร์เซ็นความชื้นในดินของวันที่ 22 เดือน มีนาคม ค.ศ. 2014 ในช่วงเวลาตั้งแต่ 0.01 น. ถึง 23.59 น.



รูปที่ 4.12ค่าพิกัดทางไฟฟ้าและทางกลของปั๊มน้ำ

จากกราฟในรูปที่ 4. 11 สามารถทำการอ่านค่าจำนวณชั่วโมงที่ใช้ในหนึ่งวัน คือช่วงเวลาตั้งแต่ 12.49 น. ถึง 15.19 น. คิดเป็นเวลา 2.5 ชั่วโมง

จากรูปที่ 4.12 ในวงกลมสีแดงสามารถหากำลังไฟฟ้า (วัตต์) ได้ 750 วัตต์

สามารถคำนวณค่าไฟฟ้าที่ใช้ได้ดังนี้

1. คำนวณหาจำนวณหน่วย (ยูนิต) ทั้งหมดที่ใช้ในหนึ่งวัน ดังนี้

จำนวณหน่วย (ยูนิต) = ((750 × 1) / 1000) × 2.5

จะได้ว่าในหนึ่งวันใช้ไฟฟ้าไป **1.875** หน่วย

2. คำนวณหาจำนวณหน่วย (ยูนิต) ทั้งหมดที่ใช้ในหนึ่งเดือน ดังนี้

มีการรดน้ำสัปดาห์ละ 1 ครั้ง

หนึ่งเดือนจะใช้ไฟฟ้าไป 1.875 × 4 = 7.5 หน่วย

จะได้ว่าในหนึ่งเดือนใช้ไฟฟ้าไป **7.5**หน่วย

จะเห็นได้ว่าจัดอยู่ในประเภทใช้ไฟฟ้าไม่เกิน 150 หน่วย

3. คำนวณค่าไฟฟ้าของ 5 หน่วยแรก ดังนี้

5 หน่วยแรก 4.96 บาท

จะได้ค่าไฟฟ้าของ 5 หน่วยแรกเป็น **4.96** บาท

4. คำนวณค่าไฟฟ้าของ 10 หน่วยถัดมา ดังนี้

10 หน่วยถัดมาหน่วยละ 0.7124 บาท

รวมทั้งหมด 2.5 หน่วย

จะได้ 0.7124 × 2.5 = 1.781

จะได้ค่าไฟฟ้าของ 10 หน่วยถัดมาเป็น **1.78**บาท

5. คำนวณค่าไฟฟ้าจากค่า Ft ดังนี้

เนื่องจากค่า Ft ของเดือน ม.ค.-เม.ย. พ.ศ. 2557 คือ 3.82 บาทต่อหน่วย

รวมทั้งหมด 7.5 หน่วย

จะได้ 3.82 * 7.5 = 28.65

จะได้ค่าไฟฟ้าจากค่า Ft เป็น **28.65** บาท

6. ทำการรวมค่าไฟฟ้าทั้งหมดที่ได้จากการคำนวณตั้งแต่ข้อที่ 3 ถึงข้อที่ 5 ดังนี้

จะได้ 4.96 + 1.78 + 28.65 = 35.39

รวมเป็นเงินทั้งสิ้น **35.39**บาทต่อเดือน

สามารถคำนวณปริมาณน้ำที่ใช้ได้ดังนี้

อัตราการไหลของน้ำหยดต่อหัวคือ 2.5 ลิตรต่อชั่วโมงต่อหัวน้ำหยด ในแปลงมีหัวน้ำหยดทั้งหมด 1866 หัว เนื่องจากมีการให้น้ำ 2.5 ชั่วโมงจะใช้น้ำไป 2.5 × 1866 × 2.5 = 11,662.5 ลิตร ในหนึ่งเดือนเปิดปั้ม 4 ครั้งจะใช้น้ำไป 11,662.5 × 4 = **46,650**ลิตร

มหาวิทยาลัยเทคโนโลยีสุรนารี

สรุป

ในการติดตั้งระบบสำหรับการใช้งานจริงนั้นได้ทำการติดตั้ง โหนดย่อยเพื่อทำการอ่านค่าจากเซ็นเซอร์วัดความชื้นในดิน เซ็นเซอร์วัดความเข้มแสง และเซ็นเซอร์วัดอุณหภูมิและความชื้นสัมพัทธ์ในอากาศเพื่อทำการส่งข้อมูลให้กับภาคประมวลผลซึ่งติดตั้งอยู่บริเวณใกล้บ่อน้ำเพื่อควบคุมการเปิด-ปิดน้ำ และเก็บข้อมูลลง SD-Card พร้อมทั้งส่งข้อมูลขึ้นเซิร์ฟเวอร์ จาก ที่ได้ทำการการใช้งานจึงสรุปได้ว่าการติดตั้งระบบสำหรับการใช้งานจริงนี้สามารถทำงานได้ตามต้องการ

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

เครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลังจะทำงานโดยการให้น้ำแก่มันสำปะหลังในปริมาณที่เหมาะสมผ่านระบบน้ำหยด โดยคำนวณจากความชื้นในดินเพื่อควบคุมการเปิดปิดเครื่องสูบน้ำ นอกจากนั้นยังมีการวัดค่า ความเข้มแสง ความชื้นในอากาศ และอุณหภูมิในแต่ละช่วงเวลา โดยได้ทำการบันทึกข้อมูลลงหน่วยความจำ SD-Card และส่งขึ้นเซิร์ฟเวอร์ผ่านโมดูลสื่อสารผ่านเครือข่ายโทรศัพท์ จากการทดลองทั้งหมดสามารถสรุปผลได้ดังนี้

1. การวัดค่าด้วยอุปกรณ์วัดค่าต่างๆสามารถอ่านค่าจากอุปกรณ์วัดความชื้นในดินและส่งข้อมูลให้ภาคประมวลผลและควบคุมการเปิด-ปิดน้ำผ่าน Xbee ได้ โดยมีระยะการรับ-ส่งข้อมูลสูงสุดประมาณ 140 เมตร

2. การประมวลผลและควบคุมการเปิด-ปิดน้ำสามารถรับข้อมูลต่างๆและบันทึกลง SD-Card และส่งขึ้นเซิร์ฟเวอร์ได้

3. การประมวลผลและควบคุมการเปิด-ปิดน้ำสามารถควบคุมการเปิดปิดน้ำได้ทั้งแบบเครื่องสูบน้ำและวาล์วไฟฟ้า

5.2 ปัญหาและอุปสรรค

1. เมื่อเครื่องสูบน้ำไม่ได้ทำงานเป็นเวลานานจะทำให้สูบน้ำไม่ขึ้นเนื่องจากอากาศเข้าไปแทนที่น้ำ

2. ในกรณีที่มีแสงน้อยหลายวันอาจทำให้แบตเตอรี่ไม่เพียงพอ ต่อการทำงานของระบบในเวลากลางคืน

3. ความแม่นยำในการวัดเนื่องจากสภาพแวดล้อมจริงและเซ็นเซอร์วัดความชื้นในดินยังทำการปรับเทียบค่ามาตรฐานยังไม่เรียบร้อย

5.3 แนวทางในการแก้ไขปัญหา

1. ตั้งเวลาเปิดเครื่องสูบน้ำวันละ 5-10 นาที ในวันที่ไม่มีการเปิดน้ำตามปกติ

2. เพิ่มความจุของแบตเตอรี่และกำลังของโซล่าเซลล์

3. เพิ่มจำนวนอุปกรณ์วัดความชื้นในดินเพื่อเพิ่มความแม่นยำ

มหาวิทยาลัยเทคโนโลยีสุรนารี

5.4 ข้อเสนอแนะเพิ่มเติม

ในอนาคตจะทำการใช้ Fuzzy controller เพื่อให้ระบบมีประสิทธิภาพสูงขึ้นและสามารถนำค่าสภาวะแวดล้อมต่างๆ เช่น อุณหภูมิ ความชื้นสัมพัทธ์ในอากาศมาประกอบการตัดสินใจในการควบคุมการเปิด-ปิดน้ำ

5.5 สิ่งที่ได้รับจากการทำโครงงาน

1. เข้าใจหลักการทำงานของบอร์ด STM32F4 Discovery
2. เข้าใจหลักการควบคุมการเปิดปิดของอุปกรณ์ไฟฟ้าด้วยไมโครคอนโทรลเลอร์
3. เข้าใจการเขียนโปรแกรมและโหลดโปรแกรมให้สามารถใช้งานได้
4. สามารถใช้งานเครื่องควบคุมการให้น้ำสำหรับไร่มันสำปะหลังได้จริงใน ไร่มันสำปะหลังของฟาร์มมหาวิทยาลัยเทคโนโลยีสุรนารี

มหาวิทยาลัยเทคโนโลยีสุรนารี

บรรณานุกรม

[1] เอกสารประกอบหน่วยปฏิบัติการที่1ระบบสมองกลฝังตัววิชาปฏิบัติการวิศวกรรมอิเล็กทรอนิกส์ขั้นสูง

[2] http://manuals.decagon.com/Manuals/Discontinued/EC-20-EC-10-EC-5-Soil-Moisture-Sensor-Operators-Manual-(discontinued).pdf

[3] http://www.sumtech.co.th/pc-tools/sim300czwww.pdf

[4] https://www.aimagin.com/amg-sense-humidity-temperature.html

[5] https://www.aimagin.com/amg-high-precision-real-time-clock.html

[6] http://www.thaitapiocastarch.org/article21_th.asp

[7] เอกสารประกอบการเรียนวิชาการวิเคราะห์ดินและพืชสำนักวิชาเทคโนโลยีการเกษตรมหาวิทยาลัยเทคโนโลยีสุรนารี

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ก.

การตั้งค่าอุปกรณ์Xbeeโดยใช้โปรแกรม X-CTU

1.ทำการดาวโหลดไดร์เวอร์D2XX จาก URL:http://www.ftdichip.com/Drivers/D2XX.htm

2. ทำการติดตั้งไดร์เวอร์D2XX ที่ได้ทำการโหลดมา

3. ทำการดาวโหลดโปรแกรมX-CTUจาก

URL:http://www.digi.com/support/productdetail?pid=3352&osvid=57&type=utilities

4.ทำการติดตั้งโปรแกรม X-CTU ที่ได้ทำการโหลดมา

5. หลังจากทำการติดตั้งโปรแกรม X-CTU เสร็จเรียบร้อยแล้วให้ทำการเชื่อมต่อ Xbeeเข้ากับคอมพิวเตอร์

6. ทำการเปิดโปรแกรม X-CTU จะได้ดังรูปที่ ก.1

7. ภายในกรอบ Select com port ให้ทำการเลือก com port ที่ Xbeeกำลังเชื่อมต่ออยู่จากนั้นทำการคลิกตรงปุ่ม Test / Query จากนั้นถ้าไม่มีความผิดพลาดเกิดขึ้นโปรแกรมจะให้ข้อมูลเบื้องต้นของ Xbeeตัวที่กำลังเชื่อมต่ออยู่ดังรูปที่ ก.2

8.ทำการคลิกตรงแท็บModem configuration จะได้ดังรูปที่ ก.3

9. ทำการคลิกตรงปุ่ม Read โปรแกรม X-CTU จะทำการอ่านข้อมูลจาก Xbeeจะได้ดังรูปที่ ก.4

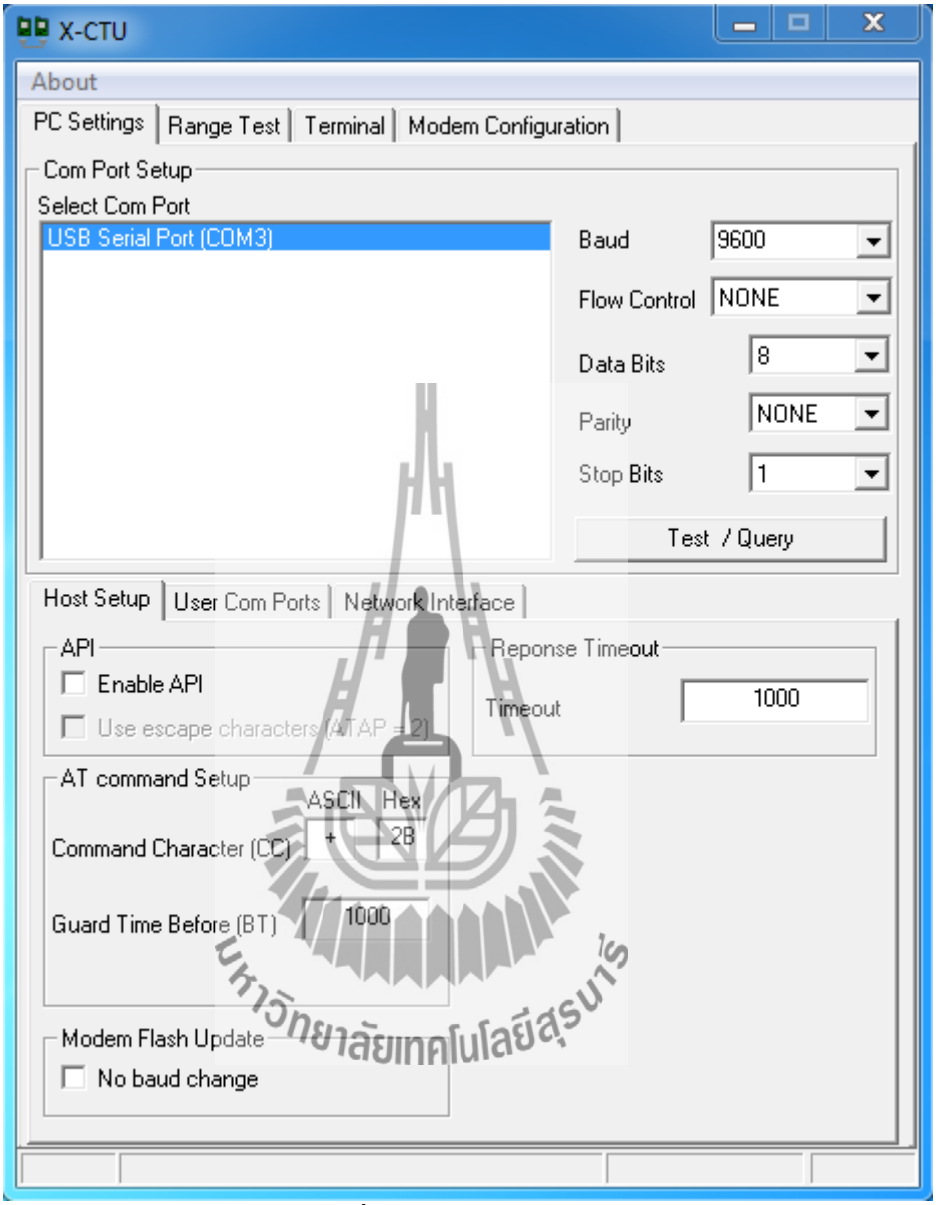
10. ทำการคลิกตรงปุ่ม Download new versions จะได้ดังรูปที่ ก.5

11. ทำการคลิกตรงปุ่ม Web เพื่อให้โปรแกรม X-CTU ทำการอัพเดทเฟิร์มแวร์เวอร์ชั่นใหม่จะได้ดังรูปที่ ก.6จากนั้นให้รอจนการอัพเดทเสร็จสิ้น

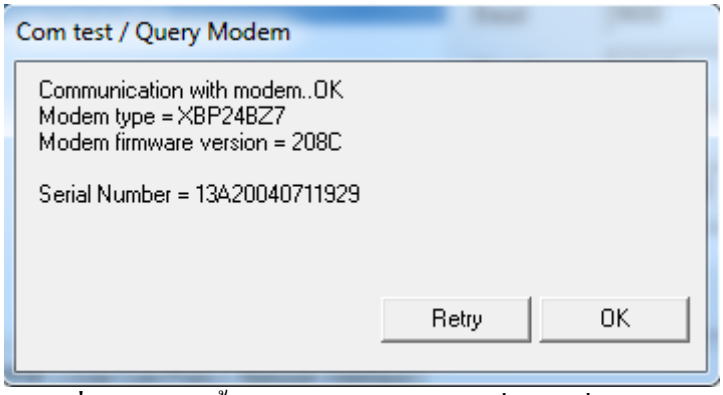
12. ทำการตั้งค่า Xbeeให้ทำหน้าที่เป็น Coordinator ได้โดยการคลิกตรงปุ่ม Read รอจนทำการอ่านข้อมูลเสร็จ จากนั้นทำการเลือก Modem เป็น XBP24BZ7 จากนั้นทำการเลือก Function set เป็น ZIGBEE COORDINATOR AT จากนั้นทำการคลิกที่ปุ่ม Write รอจนการเขียนข้อมูลลงอุปกรณ์ Xbeeเสร็จ จากนั้นทำการคลิกตรงปุ่ม Read อีกครั้งแล้วทำการปรับค่า PAN ID ให้ตามต้องการ (ค่า PAN ID ของ Coordinator ต้องตรงกับของ Router ตัวอื่นๆ จึงจะสามารถติดต่อสื่อสารกันได้) จากนั้นทำการคลิกตรงปุ่ม Write อีกครั้งเพื่อทำการเขียนค่า PAN ID ที่ได้ทำการเปลี่ยนแปลงลงอุปกรณ์ Xbee

13. ทำการตั้งค่า Xbeeให้ทำหน้าที่เป็น Router ได้โดยการคลิกตรงปุ่ม Read รอจนทำการอ่านข้อมูลเสร็จ จากนั้นทำการเลือก Modem เป็น XBP24BZ7 จากนั้นทำการเลือก Function set เป็น ZIGBEE ROUTER AT จากนั้นทำการคลิกที่ปุ่ม Write รอจนการเขียนข้อมูลลงอุปกรณ์ Xbeeเสร็จ จากนั้นทำการคลิกตรงปุ่ม Read อีกครั้งแล้วทำการปรับค่า PAN ID ให้ตามต้องการ (ค่า PAN ID ของ

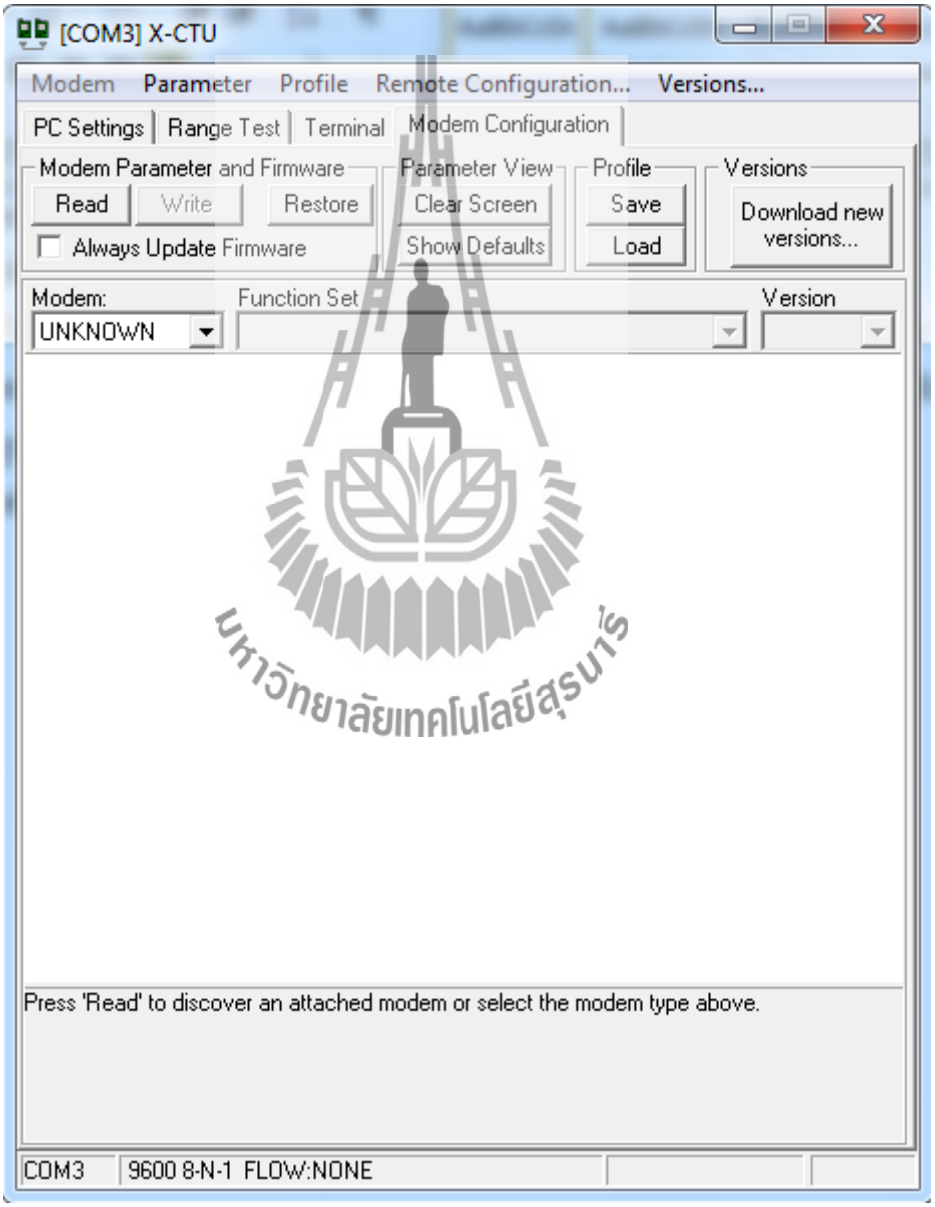
Rouerต้องตรงกับของ Coordinator จึงจะสามารถติดต่อสื่อสารกันได้) จากนั้นทำการคลิกตรงปุ่ม Write อีกครั้งเพื่อทำการเขียนค่า PAN ID ที่ได้ทำการเปลี่ยนแปลงลงอุปกรณ์ Xbee



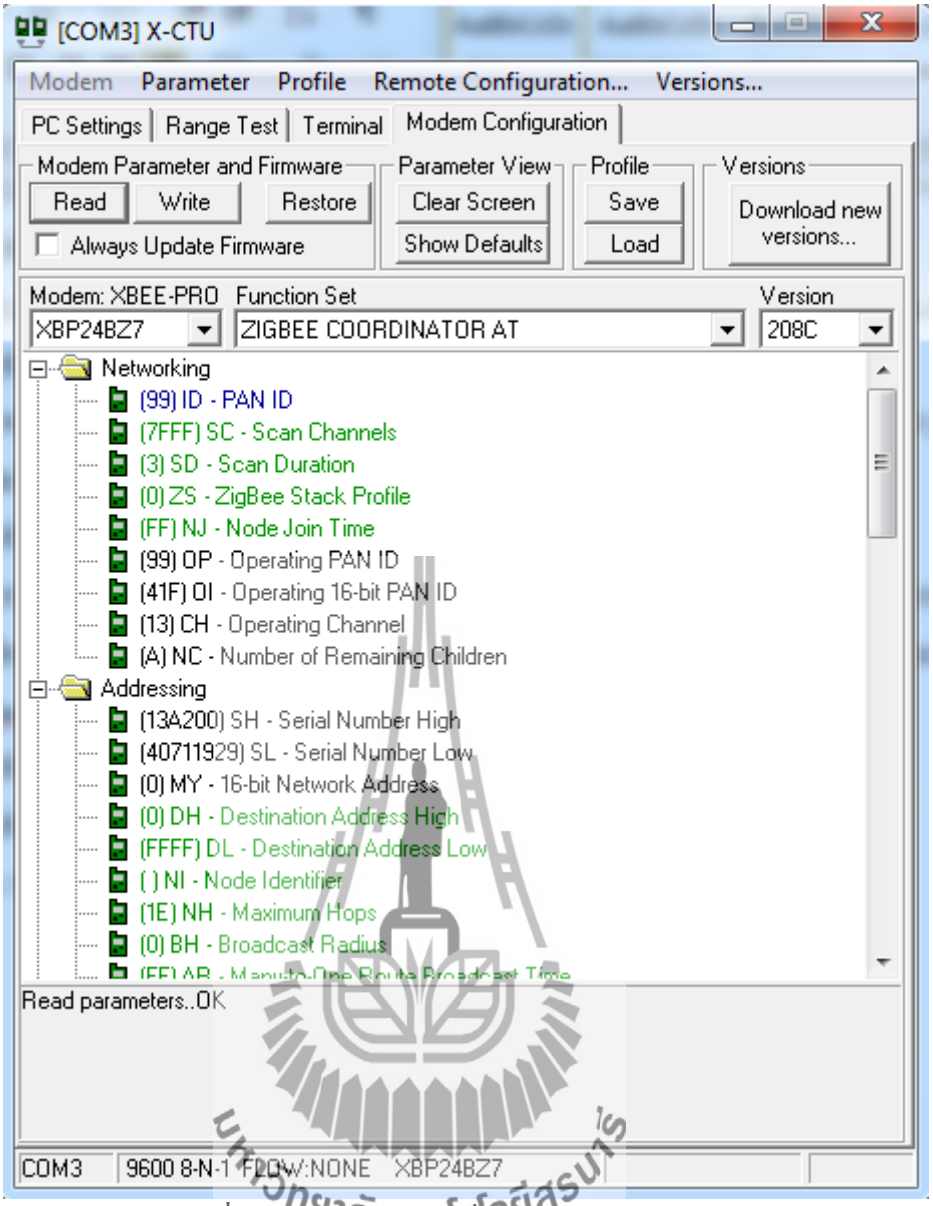
รูปที่ ก.1 โปรแกรม X-CTU



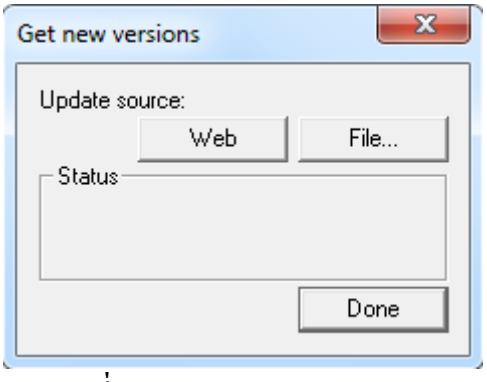
รูปที่ ก.2ข้อมูลเบื้องต้นของอุปกรณ์Xbeeที่กำลังเชื่อมต่ออยู่



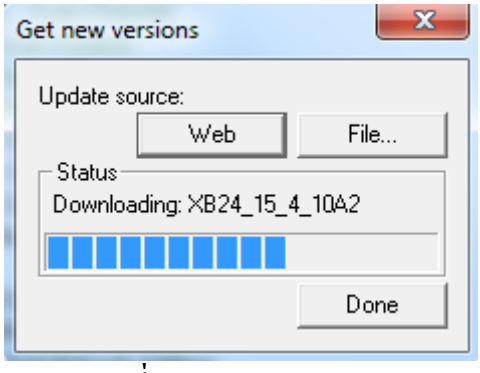
รูปที่ ก.3 Modem configuration



รูปที่ ก.4ข้อมูลของ Xbee ที่ได้จากการคลิกปุ่ม Read



รูปที่ ก.5 Download new versions



รูปที่ ก.6 Update firmware

ภาคผนวก ข.

การติดตั้งโปรแกรมสำหรับการออกแบบโปรแกรมแบบแผนภาพโดยใช้ไวจังบล็อกเซต

1.ทำการติดตั้ง Microsoft .net framework3.5 หรือใหม่กว่า

2. ทำการติดตั้งโปรแกรม STLink Utility

3. ทำการติดตั้งโปรแกรม MATLAB R2009a หรือใหม่กว่า

4. ทำการดาวโหลดไวจังบล็อกเซต สามารถดาวโหลดได้จาก

URL: https://www.aimagin.com/download/

5. หลังจากที่ได้ทำการดาวโหลดไวจังบล็อกเซตเสร็จแล้วจากนั้นให้ทำการแตกไฟล์

6. ทำการเปิดโปรแกรม MATLAB R2009a

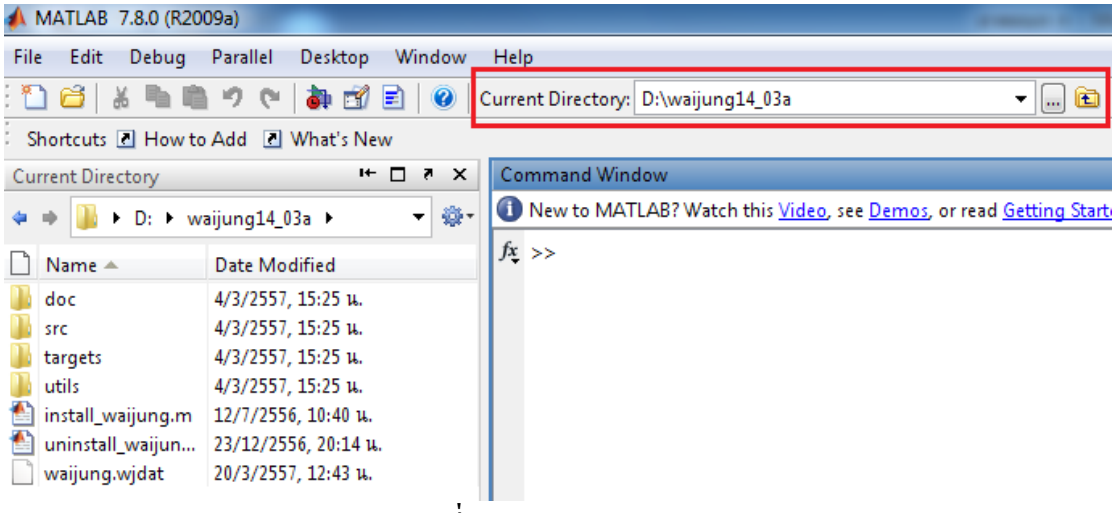
7. ทำการเปลี่ยน Current directory ให้ตรงกับ directory ที่ได้ทำการแตกไฟล์ไวจังบล็อกเซตไว้ดังรูปที่ ข.1 ในบริเวณกรอบสีแดง

8. ทำการคลิกขวาที่ไฟล์ install_waijung.mแล้วคลิก Run file เพื่อทำการติดตั้งไวจังบล็อกเซตดังรูปที่ ข.2 ในบริเวณกรอบสีแดง จากนั้นรอจนกว่าจะทำการติดตั้งเสร็จจะมีข้อความขึ้นว่า Finish waijung installation ในหน้าต่าง Command windowดังรูปที่ ข.3 ในบริเวณกรอบสีแดง

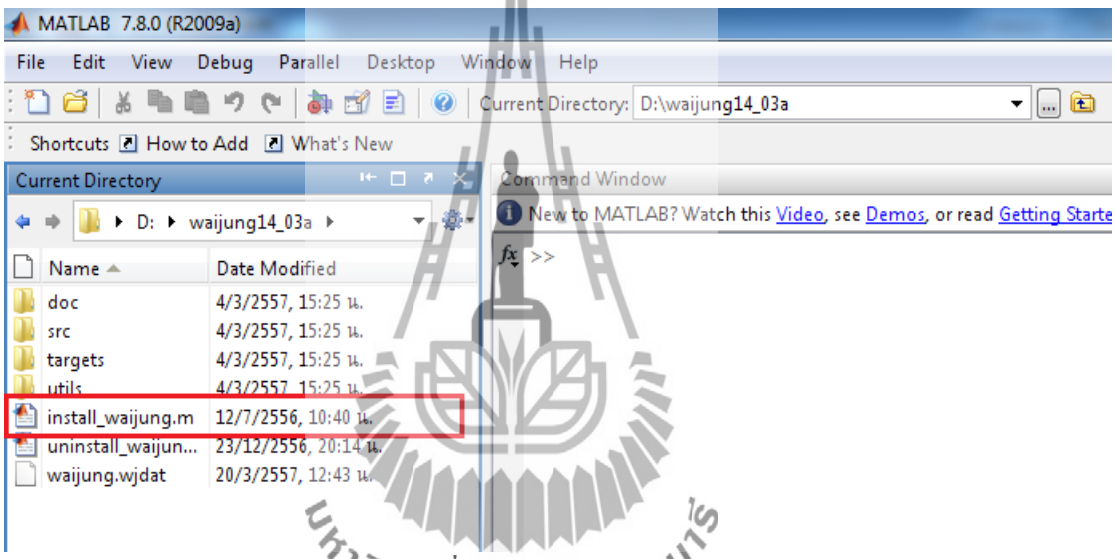
9. ภายในหน้าต่าง Command window ให้พิมพ์คำสั่ง Simulink แล้วกด Enter จะมีหน้าต่าง Simulink library browserขึ้นมาดังรูปที่ ข.4

10. ภายในหน้าต่าง Libraries เลื่อนลงมาข้างล่างแล้วคลิกที่ไวจังบล็อกเซตหนึ่งครั้งจะได้ดังรูปที่ ข.5 จากนั้นรอจนการโหลดข้อมูลไวจังบล็อกเซตเสร็จสิ้น

11. ทำการเชื่อมต่อบอร์ด STM32F4Discovery เข้ากับคอมพิวเตอร์ จากนั้นที่เมนูทำการคลิกที่ File > New > Model จะเป็นการสร้างไฟล์ใหม่ จากนั้นทำการลากบล็อกตามต้องการ แล้วทำการกดปุ่มอัพเดท 2 ครั้ง และทำการกด Ctrl + B เพื่อทำการโหลดโปรแกรมลงบอร์ด STM32F4Discovery รอจนการโหลดโปรแกรมเสร็จสิ้น

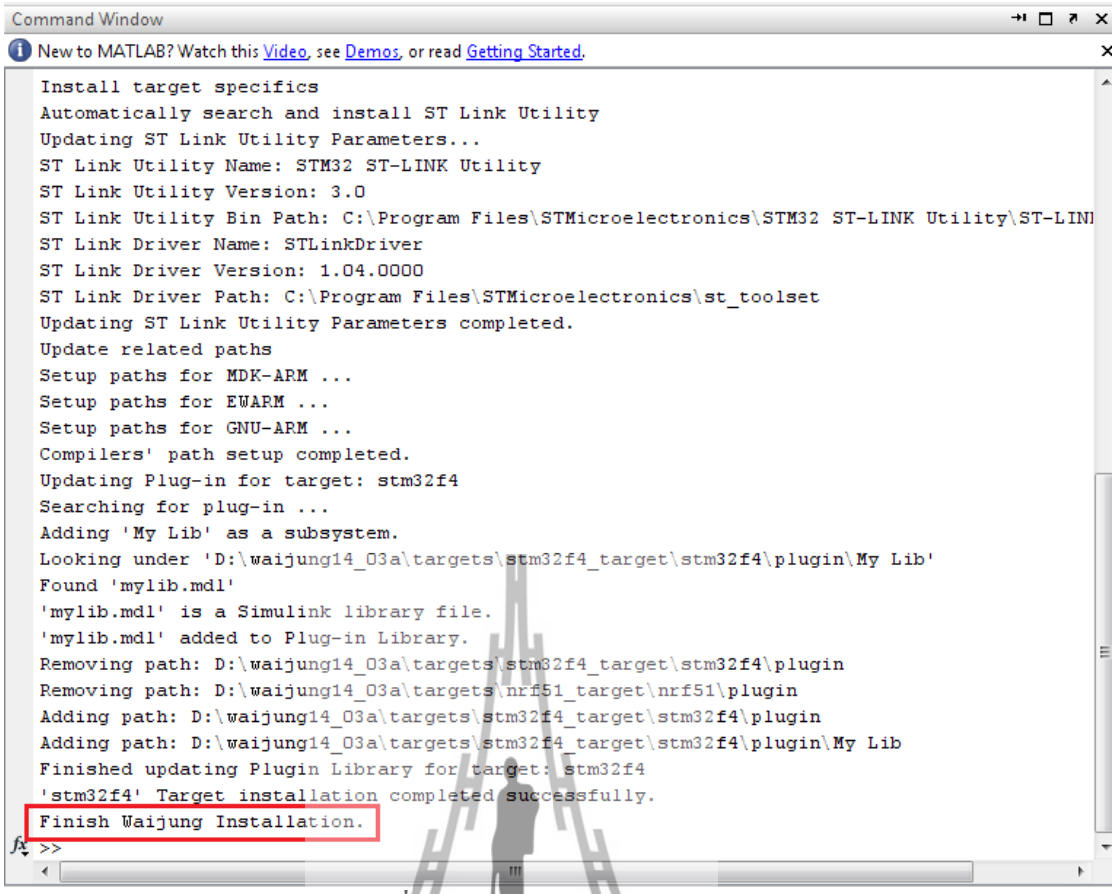


รูปที่ ข.1 Current directory



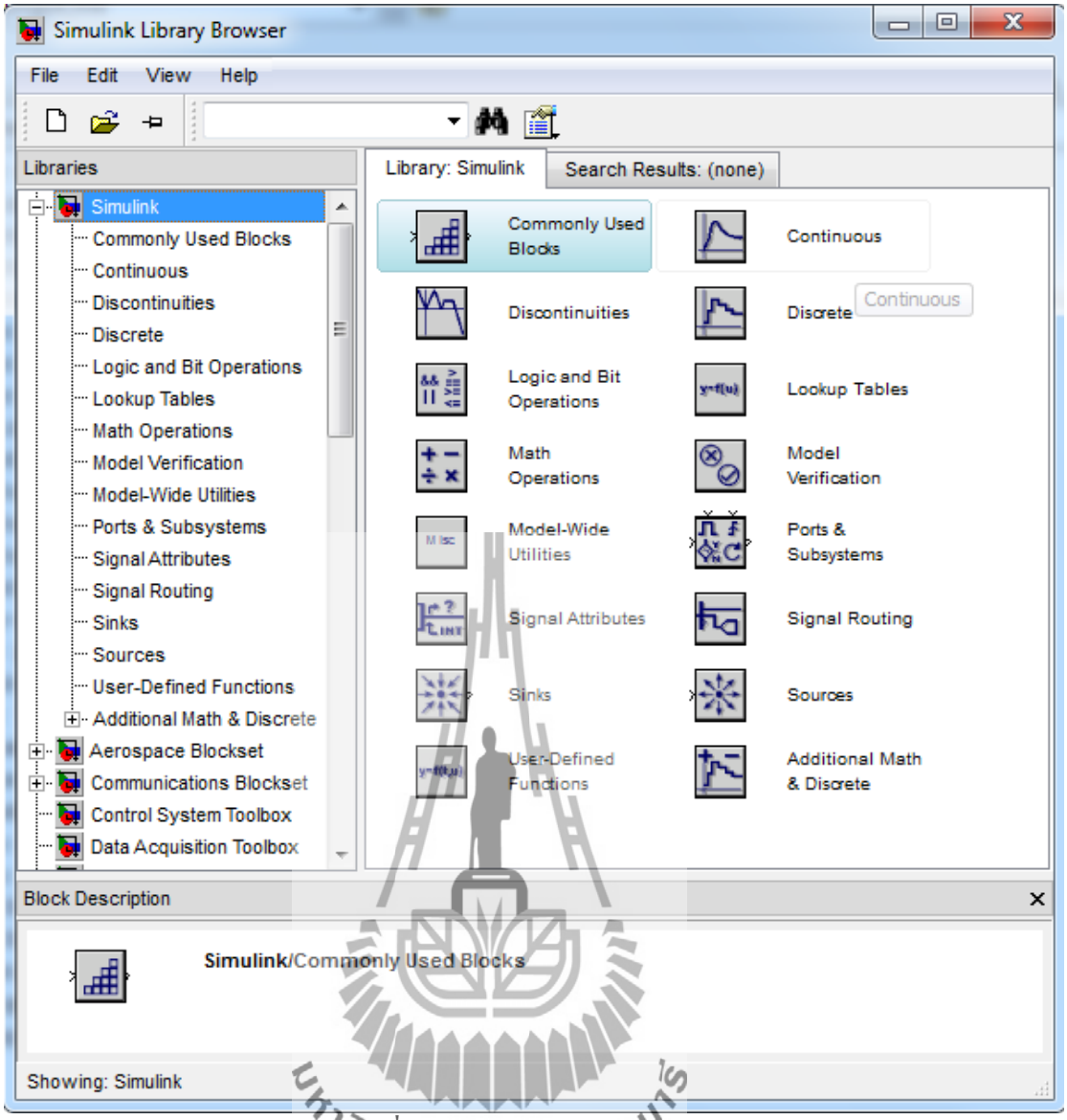
รูปที่ ข.2install_waijung.m

มหาวิทยาลัยเทคโนโลยีสุรนารี

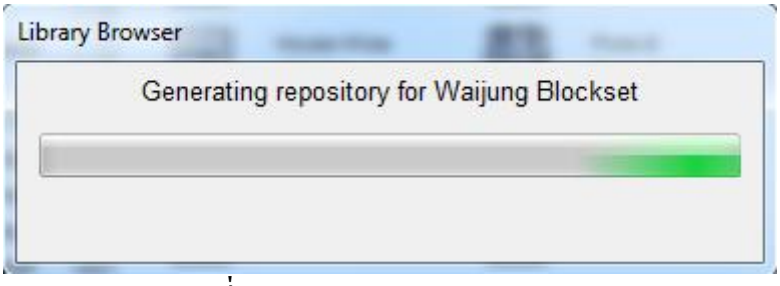


รูปที่ ข.3Finish waijung installation

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ ข.4 Simulink library browser



รูปที่ ข.5 Generating waijungblockset

ประวัติผู้เขียน

นายธีระภัทรเจริญ ปรุเกิดเมื่อวันที่ 6สิงหาคม พ.ศ.2534ภูมิลำเนาอยู่ที่ ตำบล ในเมือง อำเภอ เมือง จังหวัดนครราชสีมาสำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนราชสีมาวิทยาลัยอำเภอเมือง จังหวัด นครราชสีมา เมื่อปีพ .ศ. 2553 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคมหลักสูตรอิเล็กทรอนิกส์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นายวรวุฒิ บุญเป็ง เกิดเมื่อวันที่ 8 พฤษภาคม พ.ศ.2535 ภูมิลำเนาอยู่ที่ ตำบลบ้านมาง อำเภอเชียงม่วน จังหวัดพะเยา สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนดอกคำใต้วิทยาคม อำเภอคำใต้ จังหวัดพะเยา เมื่อปี พ.ศ.2553 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคมหลักสูตรอิเล็กทรอนิกส์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

มหาวิทยาลัยเทคโนโลยีสุรนารี