



ระบบตรวจจับพฤติกรรมกรรมการหลัดโดยใช้เซ็นเซอร์วัดความเร่งแบบ 3 แกน

โดย FiO Std Board

นางสาวจันทนิภา กาญจนนพวงศ์ รหัส B5307821
นางสาวกรรณก พุทชะ รหัส B5317424

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 427499 โครงงานวิศวกรรมโทรคมนาคม
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2546
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
ประจำภาคการศึกษาที่ 2 ปีการศึกษา 2556

เครื่องวัดระดับแอลกอฮอล์ด้วยลมหายใจ

คณะกรรมการสอบโครงการ


ผศ.ดร. ชุตินา พรหมมาก

(กรรมการ/อาจารย์ที่ปรึกษาโครงการ)


(ผศ.ดร.มนต์ทิพย์ภา อูซารสกุล)

กรรมการ


(ผศ.ร.อ.ดร.ประโยชน์ คำสวัสดิ์)

กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นำรายงาน โครงการนี้เป็น ส่วนหนึ่งของการศึกษา
ระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม วิชา 427499 โครงการวิศวกรรมโทรคมนาคม
ประจำปีการศึกษา 2556

โครงการ ระบบตรวจจับพฤติกรรมกรรมการหลักมโดยใช้เซ็นเซอร์วัดความเร่งแบบ 3 แกน
โดย FiO Std Board
ผู้จัดทำ นางสาวจันทนิกา กาญจนนพวงศ์ รหัส B5307821
นางสาวกรชนก พุททะ รหัส B5317424
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. วิภาวี หัตถกรรม
สำนักวิชา วิศวกรรมศาสตร์
สาขาวิชา วิศวกรรมโทรคมนาคม
ภาคการศึกษา 2/2556

บทคัดย่อ

โครงการเรื่อง ระบบตรวจจับพฤติกรรมกรรมการหลักมโดยใช้เซ็นเซอร์วัดความเร่งแบบ 3 แกน โดย FiO Std Board มีจุดมุ่งหมายเพื่อนำเสนอวิธีการตรวจสอบการหลักม และพฤติกรรมต่างๆ ในชีวิตประจำวันของมนุษย์ ซึ่งระบบตรวจจับพฤติกรรมนี้จะสามารถนำไปประยุกต์ใช้กับผู้สูงอายุ หรือผู้พิการที่ต้องการการดูแลอย่างใกล้ชิด โดยพฤติกรรมที่สนใจมีทั้งหมด 7 กลุ่มได้แก่ การนั่ง, การยืน, การยืนแล้วนั่ง, การนั่งแล้วยืน, การเดิน, การวิ่ง และการนอน ซึ่งพฤติกรรมเหล่านี้จะถูกเก็บเป็นฐานข้อมูลเพื่อให้ระบบทำการตรวจสอบและประมวลผล ซึ่งระบบประกอบไปด้วย 4 ส่วน ได้แก่ ส่วนของภาครับ ภาคส่ง การประมวลผล และการแสดงผล โดยภาครับและภาคส่งจะใช้ X-bee ประมวลผลค่าที่รับได้จากเซ็นเซอร์ความเร่ง 3 แกน x, y, z โดย FiO Std Board และแสดงผลไปยังเครื่องคอมพิวเตอร์ โดยทำการประมวลผล จากการใช้การวิเคราะห์ จากค่า Threshold ทำการตรวจสอบขนาดของความเร่งของพฤติกรรมก่อนและหลัง เช่น การยืนแล้วลงไปนอนติดต่อกันนานเกิน ระบบจะทำการส่งสัญญาณไปยังส่วนแสดงผลว่ามีการหลักมเกิดขึ้น เป็นต้น จากที่กล่าวมาแล้วข้างต้นจะเห็นถึงความสำคัญของระบบตรวจจับพฤติกรรมกรรมการหลักมโดยใช้เซ็นเซอร์วัดความเร่งแบบ 3 แกน ซึ่งถ้าหากสามารถตรวจจับพฤติกรรมเหล่านี้ได้ ระบบก็จะมีการส่งสัญญาณแจ้งเตือนหากเกิดอุบัติเหตุเกิดขึ้น ซึ่งจะมีประโยชน์อย่างยิ่งในการลดอุบัติเหตุที่จะเกิดขึ้นกับผู้สูงอายุ หรือผู้พิการ

กิตติกรรมประกาศ

โครงการเรื่อง ระบบตรวจจับพฤติกรรมการหล่มโดยเซ็นเซอร์วัดความเร่งแบบ 3 แกน โดย FiO Std Board นี้ สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์และคำแนะนำจากบุคคลดังต่อไปนี้

ขอขอบคุณ ผู้ช่วยศาสตราจารย์ ดร.วิภาวี หัตถกรรม อาจารย์ที่ปรึกษาโครงการ ที่ได้ให้คำชี้แนะ แนวคิด ตลอดจนแก้ไขข้อบกพร่องต่างๆ

ขอขอบคุณ นางสาว นภาพร พิมปรุ ที่ได้ให้คำปรึกษาและแนะนำเกี่ยวกับ FiO Std Board และ การใช้เซ็นเซอร์

ขอขอบคุณ นายธีระภัทร เจริญปรุ และ นายวรวิทย์ บุญเป็ง ที่ได้ให้คำปรึกษาและแนะนำเกี่ยวกับการใช้งานโปรแกรม Simulink MATLAB

ขอขอบคุณเพื่อนๆ สาขาวิศวกรรมโทรคมนาคมทุกๆ คนที่คอยให้ความช่วยเหลือในด้านต่างๆ

คณะผู้จัดทำโครงการใคร่ขอขอบพระคุณทุกๆ ท่านที่ได้กล่าวไปแล้วไว้ ณ ที่นี้สำหรับส่วนดีของโครงการชิ้นนี้ขออุทิศให้แก่คณาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้แก่คณะผู้จัดทำโครงการหากโครงการชิ้นนี้มีข้อผิดพลาดประการใดทางคณะผู้จัดทำโครงการใคร่ขออนุมัติรับและขออภัยมา ณ ที่นี้

นางสาวจันทนิภา กาญจนนพวงศ์

นางสาวกรชนก พุทธะ

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญตาราง	จ
สารบัญรูปภาพ	ฉ
บทที่ 1 บทนำ	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของการศึกษาโครงการ	2
1.4 ผลที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการดำเนินงาน	3
บทที่ 2 ทฤษฎี และหลักการที่เกี่ยวข้อง	4
2.1 บทนำ	4
2.2 Simulink	4
2.3 ZigBee (X-bee)	6
2.4 ประสิทธิภาพของระบบ	10
บทที่ 3 ขั้นตอนการดำเนินงาน	12
3.1 บทนำ	12
3.2 การออกแบบระบบ	12
3.2.1 FiO Std Board	13
3.2.2 เซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L)	16
3.2.3 X-bee	19
3.2.4 คอมพิวเตอร์โน้ตบุ๊ก Acer Aspire 4740 332G32Mn	22
3.3 การทำงานของระบบ	23
3.4 การเขียนโปรแกรมควบคุมระบบการทำงานของฮาร์ดแวร์	23
3.5 การเลือก Threshold	38

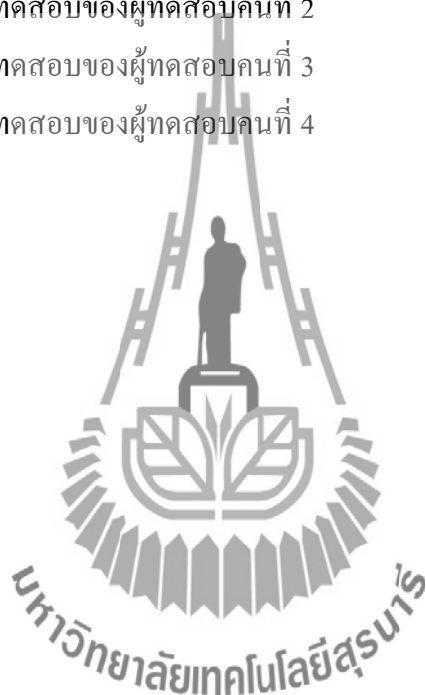
สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 4 ผลการทดลอง	40
4.1 บทนำ	40
4.2 ทดสอบการหักล้ม	40
4.2.1 ขั้นตอนการทดลอง	40
4.2.1 ผลการทดลอง	42
4.3 วิเคราะห์ผลการทดลอง	46
บทที่ 5 สรุปผลการทดลอง	47
5.1 สรุปผลการทดลอง	47
5.2 สิ่งที่ได้รับจากการทำโครงการ	47
5.3 ปัญหาและอุปสรรค	47
5.4 ข้อเสนอแนะ	48
เอกสารอ้างอิง	49
ภาคผนวก	50
ภาคผนวก ก	51
ภาคผนวก ข	92
ประวัติผู้เขียน	95



สารบัญตาราง

เรื่อง	หน้า
ตารางที่ 3.1 การจัดการฯ เซ็นเซอร์วัดความเร่ง 3 แกน และฟังก์ชันในการทำงาน	17
ตารางที่ 3.2 การจัดหาโมดูล X-bee และฟังก์ชันในการทำงาน	21
ตารางที่ 3.3 ตารางค่าความเร่งขณะล้ม	37
ตารางที่ 4.1 แสดงผลการทดสอบของผู้ทดสอบคนที่ 1	42
ตารางที่ 4.1 แสดงผลการทดสอบของผู้ทดสอบคนที่ 2	43
ตารางที่ 4.1 แสดงผลการทดสอบของผู้ทดสอบคนที่ 3	44
ตารางที่ 4.1 แสดงผลการทดสอบของผู้ทดสอบคนที่ 4	45



สารบัญรูปภาพ

รูปที่	หน้า
รูปที่ 2.1 แสดง Add-On Module	5
รูปที่ 2.2 แสดง List of Add-On Modules Block	5
รูปที่ 2.3 แสดง On-Chip Peripherals	6
รูปที่ 2.4 ย่านความถี่ใช้งานตามมาตรฐาน ZigBee	7
รูปที่ 2.5 การแสดงเครือข่ายแบบสตาร์	9
รูปที่ 2.6 การแสดงเครือข่ายต้นไม้	9
รูปที่ 2.7 การแสดงเครือข่ายแบบเมช	10
รูปที่ 3.1 อุปกรณ์ระบบ	12
รูปที่ 3.2 FiO Std Board	13
รูปที่ 3.3 FiO Std PCB LAYOUT	15
รูปที่ 3.4 เซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L)	16
รูปที่ 3.5 การจัดขาของเซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L)	16
รูปที่ 3.6 เอาต์พุตในทิศทางการวางตัวของเซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L)	18
รูปที่ 3.7 X-bee	19
รูปที่ 3.8 X-bee และ การจัดขาของ X-bee	20
รูปที่ 3.9 คอมพิวเตอร์โน้ตบุ๊ก Acer Aspire 4740 332G32Mn	22
รูปที่ 3.10 การติดตั้งโมดูล X-bee ลงบนบอร์ด X-bee USB Adapter	23
รูปที่ 3.11 หน้าต่างโปรแกรม X-CTU	24
รูปที่ 3.12 หน้าต่างแจ้งผลการติดต่อ	24
รูปที่ 3.13 ผลการอ่านค่าจากโมดูล X-bee ก่อนการตั้งค่า	25
รูปที่ 3.14 การตั้งค่า X-bee	26
รูปที่ 3.15 แสดงข้อความแจ้งสถานการณ์ทำงาน	27
รูปที่ 3.16 X-bee มีการรับส่งข้อมูลผ่านโปรแกรม X-CTU	28
รูปที่ 3.17 การตั้งค่า Jumper และ Switches	29
รูปที่ 3.18 การทดสอบการเชื่อมต่อ FiO Sta Board	30
รูปที่ 3.19 แสดง Error เมื่อไม่พบบอร์ด	30

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
รูปที่ 3.20 การเชื่อมต่อเซ็นเซอร์วัดความเร่ง 3 แกน เข้ากับ FiO Std Board	31
รูปที่ 3.21 การเชื่อมต่อX-bee ภาคส่ง เข้ากับ FiO Std Board	32
รูปที่ 3.22 การเชื่อมต่อX-bee ภาครับ เข้ากับ FiO Std Board	32
รูปที่ 3.23 บล็อก Simulink ของฮาร์ดแวร์ภาคส่ง	33
รูปที่ 3.24 สมการในบล็อก Embedded MATLAB Function	34
รูปที่ 3.25 บล็อกภายใน Enable Subsystem ของภาคส่ง	34
รูปที่ 3.26 บล็อกภายใน Enable Subsystem1 ของภาคส่ง	35
รูปที่ 3.27 บล็อก Simulink ของ ฮาร์ดแวร์ ภาครับ	35
รูปที่ 3.28 บล็อกภายใน Enable Subsystem ของภาครับ	36
รูปที่ 3.29 ฮาร์ดแวร์ไร้สาย	36
รูปที่ 3.30 ใส่เงื่อนไขค่า Threshold ใน บล็อก Embedded MATLAB Function	38
รูปที่ 3.31 ภาครับส่งข้อมูลผ่าน X-bee มายัง คอมพิวเตอร์โน้ตบุ๊ก	39
รูปที่ 4.1 ติดตั้งอุปกรณ์ Hardware ที่ตำแหน่งเอวของผู้ทดสอบ	40
รูปที่ 4.2 ติดตั้ง X-bee ภาครับเข้ากับคอมพิวเตอร์โน้ตบุ๊ก	41
รูปที่ 4.3 เปิดโปรแกรมของภาครับ และเปิด Port ที่เราใช้สำหรับ X-bee ภาครับ	41

บทที่ 1

บทนำ

จากปัญหากลุ่มผู้สูงอายุไม่สามารถดูแลตัวเองได้ จำเป็นต้องได้รับการดูแลอย่างใกล้ชิดจากลูกหลานหรือบุคลากรทางการแพทย์ [1]-[3] แต่จากผลกระทบของปัญหาการขาดแคลนบุคลากรทางการแพทย์ที่มีอยู่อย่างจำกัดก่อนอยู่แล้ว ทำให้บุคลากรทางการแพทย์ไม่เพียงพอต่อการรองรับความต้องการในการดูแลอย่างใกล้ชิดในส่วนนี้ได้ ปัญหาการหกล้มของผู้สูงอายุก็เป็นปัญหาหนึ่ง ที่มักจะเป็นสาเหตุทำให้เกิดการสูญเสียเกิดขึ้นได้ ซึ่งปัญหาการหกล้มอาจเกิดจากสาเหตุการป่วยแล้วเกิดอุบัติเหตุ การเป็นลมแล้วล้ม หรืออาจเกิดจากอุบัติเหตุลื่นล้ม หรือสะดุดล้ม ปัญหานี้จึงเป็นปัญหาหนึ่งที่เกี่ยวข้องโดยตรงกับเรื่องสุขภาพของกลุ่มผู้สูงอายุ โดยกลุ่มคนดังกล่าวมีความเสี่ยงที่จะเกิดอันตรายจากการสูญเสียหรือเป็นอัมพาตหลังจากที่ล้มแล้วได้สูงกว่าวัยอื่นๆ อุบัติเหตุเกี่ยวกับการล้มมักเกิดในบริเวณที่ผู้สูงอายุอยู่เพียงคนเดียว ซึ่งแม้ว่าจะมีผู้ที่คอยดูแลอย่างใกล้ชิดแต่ก็ไม่สามารถดูแลได้ตลอดเวลา และหากเกิดการล้มขึ้นแล้วปัญหาใหญ่ที่ตามมาคือการไม่ได้รับการช่วยเหลือได้อย่างทันท่วงที

ปัจจุบันนวัตกรรมมากมายที่มีการคิดค้นเพื่อสร้างความปลอดภัยในระดับหนึ่งจากเดิมซึ่งอาจใช้การทำงานของกล้องวงจรปิด ที่จะทำให้เพียงแต่การบันทึกภาพเหตุการณ์ต่างๆเอาไว้เท่านั้น แต่ตัวกล้องจะไม่สามารถแจ้งสัญญาณเตือนได้ เปลี่ยนมาเป็นการใช้เทคโนโลยีการตรวจจับเหตุการณ์หกล้มด้วยการประมวลผลบนคอมพิวเตอร์ หรือการติดตั้งอุปกรณ์เซ็นเซอร์ที่ตัวผู้ถูกเฝ้าติดตามเพื่อให้อุปกรณ์ที่ทำการติดตั้งสามารถวิเคราะห์ข้อมูลและรายงานผลได้ด้วยตัวเอง

1.1 หลักการ และเหตุผล

ปัจจุบันเครือข่ายเซ็นเซอร์ไร้สาย (Wireless Sensor Network) เป็นเทคโนโลยีในการติดต่อสื่อสารที่นิยมมากขึ้น การใช้อุปกรณ์เซ็นเซอร์ขนาดเล็กๆ จำนวนมากเพื่อตรวจวัดคุณสมบัติต่างๆ อาทิเช่น อุณหภูมิ ความร้อน ความชื้นสัมพัทธ์ ของสิ่งแวดล้อมที่เราสนใจและนำผลข้อมูลเหล่านั้นมาสร้างความรู้ใหม่ เพื่อที่จะตอบสนองกับการเปลี่ยนแปลงของสภาพแวดล้อมได้อย่างอัตโนมัติ เทคโนโลยีไร้สายนี้สามารถนำมาประยุกต์ใช้ให้เกิดประโยชน์ได้หลากหลายสาขา ไม่ว่าจะเป็นด้านเกษตรกรรม ด้านอุตสาหกรรม ด้านการรักษาความปลอดภัย และด้านการแพทย์ โดยในด้านการแพทย์นั้นมีการประยุกต์ใช้เครือข่ายเซ็นเซอร์ไร้สายในหลายงานเช่นการตรวจวัดคนไข้ผ่านทางระบบเครือข่ายไร้สาย โดยระบบนี้จะใช้เซ็นเซอร์ที่สามารถวัดค่าที่จำเป็นสำหรับการรักษา เช่น ค่าอัตราการเต้นของหัวใจเข้ามาช่วยในการตรวจรักษาคคนไข้ และอีกด้านที่มีการประยุกต์ใช้งาน

เซ็นเซอร์ไร้สายคือการใช้งานกับผู้สูงอายุ โดยประยุกต์เซ็นเซอร์ไร้สายเพื่อวิเคราะห์พฤติกรรมและลักษณะการเคลื่อนที่ของผู้สูงอายุ ว่าในขณะนั้นผู้สูงอายุกำลังมีพฤติกรรมอย่างไร เช่น ยืน วิ่ง เดิน นอน นั่งหรือหกล้ม ซึ่งหากเราสามารถตรวจจับพฤติกรรมเหล่านี้ได้ จะมีประโยชน์อย่างยิ่งในการลดอุบัติเหตุที่อาจเกิดขึ้นกับผู้สูงอายุหรือมีการแจ้งเตือนหากเกิดอุบัติเหตุขึ้นในกรณีที่ในขณะนั้นไม่มีผู้ดูแลผู้สูงอายุ

การพัฒนาฮาร์ดแวร์ให้เป็นอุปกรณ์ตรวจจับพฤติกรรมการหกล้มโดยใช้เซ็นเซอร์ 3 แกน โดยอุปกรณ์วัดความเร่งในแนวแกน x , y , และ z พารามิเตอร์ดังกล่าวสามารถนำมาวิเคราะห์ เพื่อแบ่งแยกลักษณะพฤติกรรมเคลื่อนที่ของมนุษย์ โดยค่าความเร่งในแนวแกน x , y , และ z จะผ่านการวิเคราะห์ ผ่านบล็อก Simulink โดยการใช้ค่า Threshold ที่ผ่านการทดสอบแล้วว่าสามารถตรวจจับพฤติกรรมการหกล้มได้จริง ซึ่งเป็นหลักการที่ เหมาะสมที่จะนำมาวิเคราะห์ข้อมูลของลักษณะพฤติกรรมของมนุษย์ที่ต้องการรายงานผลอย่างรวดเร็วนั่นเอง

1.2 วัตถุประสงค์

1. เพื่อพัฒนาฮาร์ดแวร์อุปกรณ์ตรวจวัดความเร่งโดยใช้เซ็นเซอร์ความเร่งสามแกน ประมวลผลโดย FiO Std Board และรับส่งข้อมูลโดย X-bee
2. เพื่อสร้างอุปกรณ์ต้นแบบสำหรับตรวจจับลักษณะพฤติกรรมการหกล้มของมนุษย์
3. เพื่อศึกษาโปรแกรม MATLAB
4. เพื่อนำความรู้ที่ได้ไปใช้ในการประกอบวิชาชีพ
5. เพื่อเข้าใจการทำงานร่วมกันของ FiO Std Board กับ โปรแกรม MATLAB Simulink

1.3 ขอบเขตของการศึกษาโครงการ

1. ศึกษาความรู้พื้นฐานในการใช้โปรแกรม MATLAB
2. ศึกษาการใช้อุปกรณ์ FiO Std Board โดยติดต่อกับ MATLAB
3. พัฒนาโปรแกรมเพื่อตรวจดูว่าผู้ที่สวมใส่กำลังยืน เดิน วิ่ง นั่ง ยืน หรือนอน

1.4 ผลที่คาดว่าจะได้รับ

1. สามารถประยุกต์ใช้งานอุปกรณ์ตรวจวัดความเร่ง จาก FiO Std Board ในระบบเครือข่าย เซ็นเซอร์ไร้สาย
2. สามารถสร้างอุปกรณ์ต้นแบบสำหรับตรวจจับพฤติกรรมการยืน เดิน วิ่ง นั่ง หรือนอน ของมนุษย์

3. มีความรู้ความเข้าใจในการใช้งานโปรแกรม MATLAB Simulink
4. มีความรู้นำไปประยุกต์ใช้ในการประกอบวิชาชีพ

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาอาจารย์ที่ปรึกษาเกี่ยวกับขอบเขตของโครงการที่จะทำ
2. จัดหาอุปกรณ์ที่เกี่ยวข้อง
3. ศึกษาข้อมูลเกี่ยวกับ อุปกรณ์ X-bee เบื้องต้น
4. ศึกษาข้อมูลเกี่ยวกับ อุปกรณ์ FiO Std Board เบื้องต้น
5. ศึกษาวิธีการใช้โปรแกรม MATLAB Simulink
6. เขียนคำสั่งในโปรแกรม MATLAB เพื่อประยุกต์ใช้งานกับ FiO Std Board
7. ออกแบบแพ็คเกจจิ้งและประกอบชิ้นงาน ให้กะทัดรัดเพื่อสะดวกแก่การใช้งานจริง
8. ทดสอบการทำงานและปรับปรุงแก้ไขโปรแกรม
9. จัดทำรูปเล่มรายงานของโครงการเพื่อเสนออาจารย์ประจำสาขาวิชา



บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 บทนำ

ในบทนี้จะกล่าวถึงทฤษฎีและหลักการของ Simulink ที่นำมาใช้สร้างบล็อกเพื่อทำการโปรแกรมลงฮาร์ดแวร์ ทฤษฎีของ ZigBee (X-bee) และการหาค่าประสิทธิภาพเพื่อนำมาวิเคราะห์ผลการทดลองที่ได้

2.2 Simulink

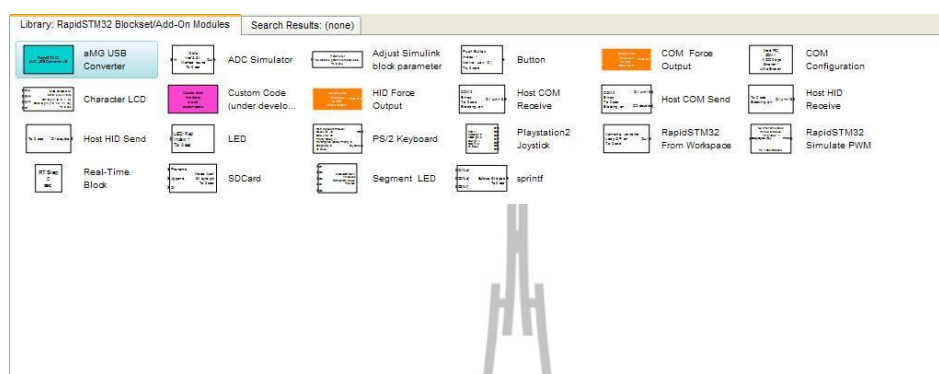
โปรแกรม MATLAB มีเครื่องมือที่ใช้สำหรับการวิเคราะห์และทดสอบระบบโดยการจำลองขึ้นมา ซึ่งก็คือ Simulink เป็นโปรแกรมที่ควบคู่กับ MATLAB ซึ่งเป็นระบบเชิงเส้น (linear) ระบบไม่เชิงเส้น (Nonlinear) Simulink เป็นโปรแกรม Mouse-Driver ที่ใช้ระบบโมเดลโดยการวาดบล็อกไดอะแกรมบนจอภาพด้วยการใช้เมาส์ทำให้โปรแกรม MATLAB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

Block sets 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 ที่มีความเร็วต่ำแต่มีข้อเสียคือฟังก์ชัน ที่นำมาใช้งานมีน้อยทำให้เขียนโปรแกรมที่มีความซับซ้อนได้ไม่ดีเท่าที่ควร เพราะมีประสิทธิภาพและความเร็วในการประมวลผลต่ำ ต่อมาเมื่อระบบเลือกใช้ได้มากมายจึงทำให้โปรแกรม MATLAB มีประสิทธิภาพและมีความสามารถในการประมวลผลที่เร็วขึ้น Version ใหม่ที่ได้ทำการปรับปรุง ใหม่ให้ดีขึ้นนี้จะใช้งานบน Windows ทำให้ผู้ใช้มีความสะดวกในการใช้งานมากขึ้น ข้อดีของ version ใหม่ก็คือ มีประสิทธิภาพที่ดีขึ้น การประมวลผลโปรแกรมที่ซับซ้อนมีความเร็วสูงขึ้น และมีฟังก์ชันต่างๆ ใช้

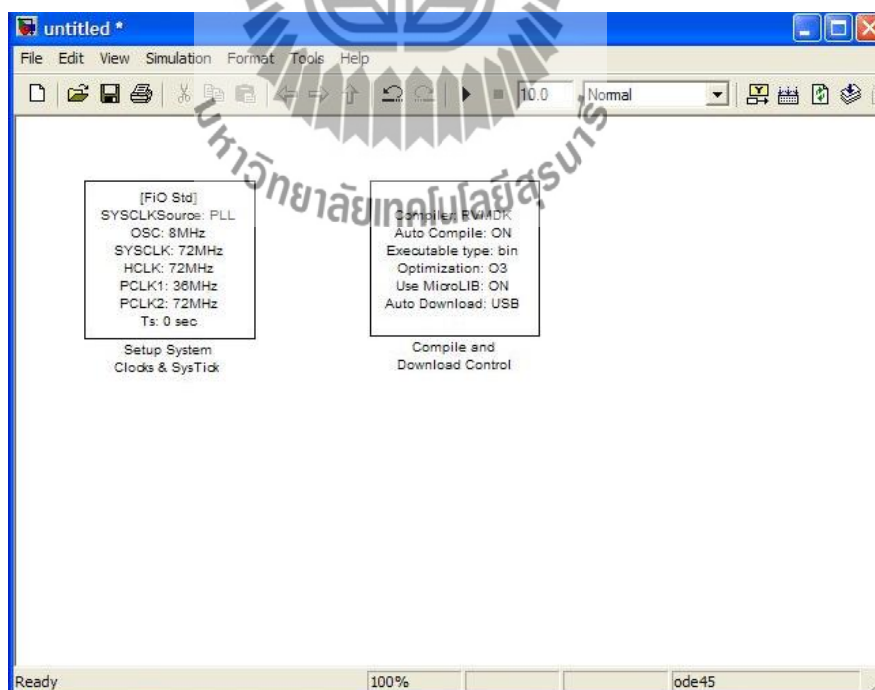
เลือกใช้ในสาขาต่างๆ มากมายแต่ก็ต้องใช้กับคอมพิวเตอร์ที่มีหน่วยความจำมาก CPU มีความเร็วสูง และต้องการ Co-Processor ในการช่วยคำนวณแต่เมื่อเปรียบเทียบกับผลที่ได้ก็ถือว่าคุ้ม ค่า ใน Simulink จะมี Block sets หลายรูปแบบ ซึ่งจะแบ่งได้เป็นดังนี้

-Add-On Modules



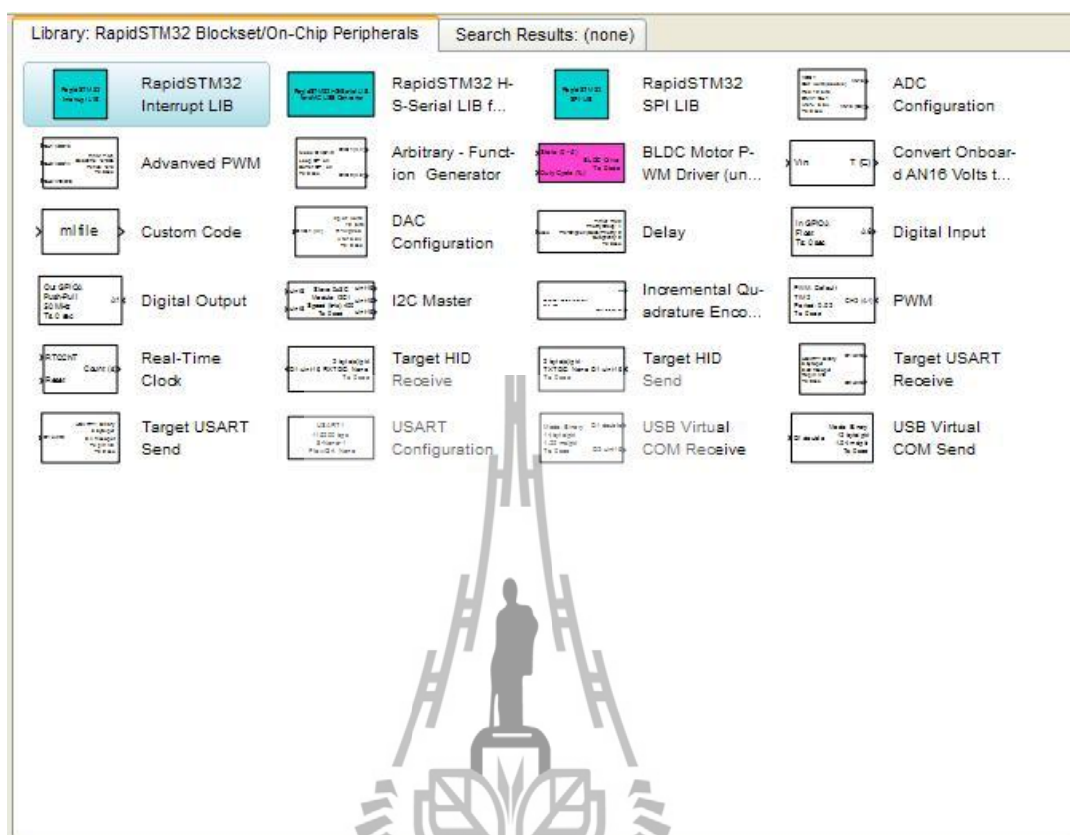
รูปที่ 2.1 แสดง Add-On Module

- List of Add-On Modules Blocks



รูปที่ 2.2 แสดง List of Add-On Modules Block

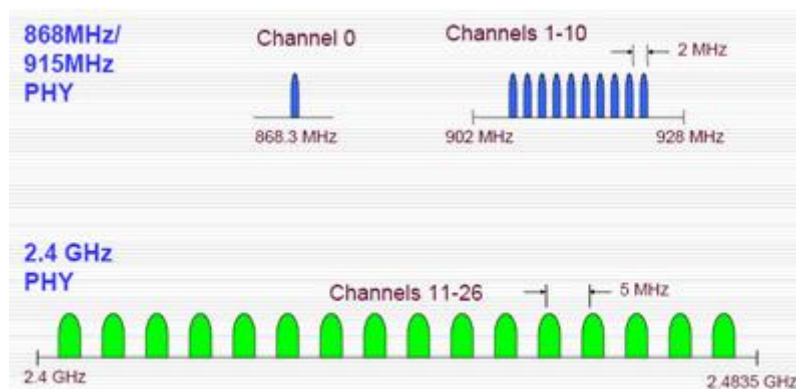
- On-Chip Peripherals



รูปที่ 2.3 แสดง On-Chip Peripherals

2.3 ZigBee (X-bee)

ZigBee มาตรฐานสากล 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 ๆ ZigBee กำหนด ย่านความถี่ใช้งานตามมาตรฐานไว้ 3 ย่านความถี่คือ ย่าน 2.4 GHz , ย่าน 915 MHz และย่าน 868 MHz โดยแต่ละย่านจะมีช่องสัญญาณ 16 ช่อง , 10 ช่อง และ 1 ช่อง ตามลำดับ ส่วน อัตรารับส่งข้อมูล (ทางอากาศ) จะอยู่ที่ 250 Kbps , 40 Kbps , 20 Kbps ตามลำดับเช่นกัน โดยในพื้นที่โล่งในระยะสื่อสารประมาณ 200 เมตร สำหรับในอาคารมีระยะสื่อสารประมาณ 30 เมตร สามารถเพิ่มระยะสื่อสารได้โดยการเพิ่ม intermediate node หรือ router



รูปที่ 2.4 ย่านความถี่ใช้งานตามมาตรฐาน ZigBee

ZigBee สามารถสร้างเป็นเครือข่ายได้ ทั้งนี้ ZigBee ได้อ้างอิงมาตรฐานตาม IEEE 802.15.4 โดย IEEE 202.15.4 แบ่งชนิดอุปกรณ์ในเครือข่ายออกเป็น 2 ประเภทคือ FFD (Full Function Device) ซึ่งหมายถึงอุปกรณ์ที่สามารถทำงานได้ทุกอย่างในเครือข่าย และ RFD (Reduce Function Device) ซึ่งหมายถึงอุปกรณ์ที่ถูกลดความสามารถในการทำงานในเครือข่าย IEEE 802.15.4 รูปแบบอุปกรณ์ดังนี้

Network Coordinator

- รักษาข้อมูลทั้งหมดของระบบ
- มีความซับซ้อนมากที่สุดในระบบการแผนภูมิต้นไม้
- ใช้ความจำและกระบวนการคำนวณมาก

Full Function Device (FFD)

- ใช้โหม้ดการทำงานของ IEEE 802.15.4 และลักษณะเฉพาะเจาะจงทั้งหมดโดย

มาตรฐาน

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

มาตรฐาน IEEE 802.15.4 เป็นมาตรฐานสำหรับการสื่อสารระยะใกล้ที่เน้นการใช้งานพลังงานต่ำและเน้นการสื่อสารที่ทนต่อสภาพสัญญาณรบกวนสูงและเน้นที่ความง่ายเพื่อให้ชีพการสื่อสารมีราคาถูกลงส่วนมาตรฐาน ZigBee เป็นมาตรฐานที่ออกแบบมาเฉพาะสำหรับการติดต่อ

เครื่องเซ็นเซอร์ชนิดต่างๆเพื่อการพัฒนาแอปพลิเคชัน เช่น เครื่องเซ็นเซอร์สำหรับ Home Automation ที่มีการสื่อสารระหว่างเซ็นเซอร์ประเภทการสื่อสารแบบ ZigBee กับกล้องเพื่อบันทึกภาพคนที่เข้ามาในบ้านเป็นแบบ multi-hop routing ที่สามารถส่งข้อมูลไปยังเครื่องเซ็นเซอร์ที่ต้องการโดยผ่านเครื่องเซ็นเซอร์ตัวอื่นๆ ซึ่งคุณสมบัตินี้ไม่ได้รับการสนับสนุนในบลูทูธ การสื่อสารแบบ ZigBee ช่วยให้ขบวนการส่งของข้อมูลได้ออกไปเรื่อย ZigBee สำหรับในชั้น network ของ ZigBee จะอยู่บนพื้นฐานของระบบมาตรฐาน IEEE 802.15.4 ซึ่งในชั้น network สามารถรองรับได้ 3 แบบ คือแบบสตาร์(Star) แบบเมช(Mesh) และวงแหวน(Ring Topology) เมื่อมีการเพิ่มจำนวนของเครื่องเซ็นเซอร์ด้วย จากที่ ZigBee รองรับโทโปโลยีแบบเมช(mesh) ทำให้ต้องมีการค้นหาเส้นทางด้วยตัวเอง (Multiple Routing Algorithms) ซึ่งในตัว ZigBee นี้จึงจะต้องมีโปรโตคอล (AODV Protocol) ซึ่งการทำงานของ AODV Protocol คือการส่งข้อความ (Message) ผ่านไปยังโหนดข้างเคียง (Neighbor Node) เพื่อไปยังโหนดที่ต้นทางไม่สามารถติดต่อได้โดยตรง ในระหว่างทางที่ข้อความถูกส่งผ่านไปโปรโตคอล AODV ก็จะทำการค้นหาเส้นทางไปด้วย โดยจะเชื่อว่าจะไม่เกิดการวนลูป (Loop) และพยายามหาเส้นทางที่สั้นที่สุดเพื่อประหยัดเวลาในการส่งข้อมูลและโปรโตคอล AODV ยังสามารถที่จะควบคุมการเปลี่ยนแปลงของเส้นทาง (Route) และสามารถสร้างเส้นทางใหม่ได้หากเกิดข้อผิดพลาด

โครงสร้างของโปรโตคอล ZigBee

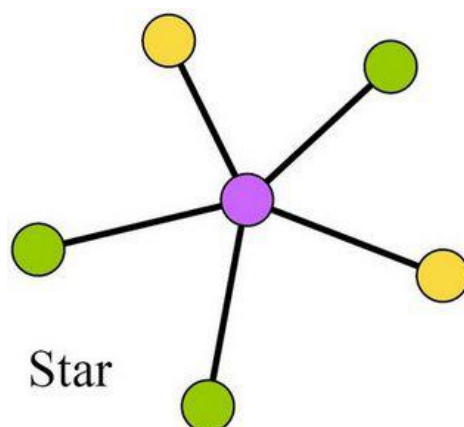
Application Layer เป็นชั้นที่มีส่วนของการทำงานบนเฟรม (Application Framework) ทำหน้าที่ในการจัดการในการเข้าถึงและใช้งานบนเลเยอร์นั้น

Application Support Sub-Layer ทำหน้าที่ในการสร้างเฟรม (Frame) และทำหน้าที่ในการรับส่งข้อมูล

Network Layer ทำหน้าที่ใช้ในการหาเส้นทาง (Routing) ข้อมูลจากต้นทางไปยังปลายทางที่อาจอยู่ภายในเครือข่ายเดียวกันหรือต่างเครือข่ายกัน

เครือข่ายแบบสตาร์ (Star Network)

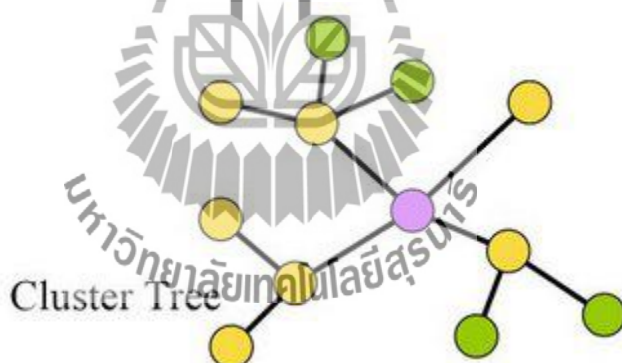
ประกอบด้วยจุดเชื่อมต่อโปรโตคอล ZigBee 1 จุดและอุปกรณ์ปลายทางหลายๆ ในเครือข่ายแบบสตาร์อุปกรณ์ปลายทางทั้งหมดจะสื่อสารกับอุปกรณ์เชื่อมต่อเท่านั้น ถ้าอุปกรณ์ปลายทางหนึ่งต้องการสื่อสารกับอุปกรณ์ปลายทางอื่นๆ ต้องส่งข้อมูลผ่านอุปกรณ์เชื่อมต่อเท่านั้น หรือกล่าวอีกนัยหนึ่งคือ อุปกรณ์เชื่อมต่อทำหน้าที่ส่งข้อมูลไปยังผู้รับดังรูปที่ 2.5



รูปที่ 2.5 การแสดงเครือข่ายแบบสตาร์

เครือข่ายแบบต้นไม้ (Cluster Tree)

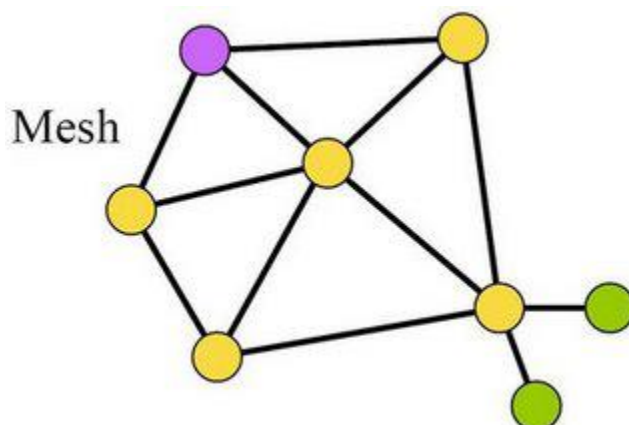
ในเครือข่ายนี้ อุปกรณ์ปลายทางจะสามารถเชื่อมต่อได้กับอุปกรณ์เชื่อมต่อหรือ ZigBee โปรโตคอลเราเตอร์ทำหน้าที่ 2 ประการคือ เพิ่มจำนวนโหนดที่สามารถเชื่อมต่ออยู่บนเครือข่าย และขยายขนาดของเครือข่าย เนื่องจากเราเตอร์จะทำหน้าที่ส่งข้อมูลไปยังจุดต่างๆของเครือข่ายได้ โดยที่อุปกรณ์ปลายทางไม่จำเป็นต้องอยู่ในระยะการส่งสัญญาณวิทยุ ดังรูปที่ 2.6



รูปที่ 2.6 การแสดงเครือข่ายต้นไม้

เครือข่ายแบบเมช (Mesh Network)

เครือข่ายแบบเมชใช้กับเครือข่ายแบบต้นไม้ ยกเว้นอุปกรณ์ FFD สามารถส่งข้อมูลไปยัง FFD อื่นได้โดยตรงไม่ต้องผ่านโครงสร้างต้นไม้ ข้อมูลที่ส่งไปยัง RFD จะต้องทำการต่อผ่านอุปกรณ์ RFD ก่อนหน้า ข้อดีของการเชื่อมต่อแบบนี้คือ ช่วยลดอัตราความล่าช้าของการส่งและเพิ่มความน่าเชื่อถือของระบบ ดังรูปที่ 2.7



รูปที่ 2.7 การแสดงเครือข่ายแบบเมช

เครือข่ายแบบต้นไม้และเครือข่ายแบบเมช มีอีกชื่อว่า เครือข่ายหลายจุด (Multi-Hop) ขณะที่เครือข่ายแบบสตาร์เป็นเครือข่ายจุดเดียว (Single-Hop) เครือข่ายโปรโตคอล ZigBee เป็นเครือข่ายแบบเชื่อมต่อได้หลายอุปกรณ์พร้อมกัน ซึ่งหมายความว่า จุดเชื่อมต่อในเครือข่ายทุกจุดมีสิทธิในการเข้าถึงตัวกลางที่ใช้ในการสื่อสารเท่าๆกัน มีทุกจุดเชื่อมต่อในเครือข่ายส่งข้อมูลได้ตลอดเวลาที่ช่องสัญญาณว่างอยู่ในเครือข่ายแบบ Beacon จุดเชื่อมต่อจะสามารถส่งข้อมูลได้ในช่วงเวลาที่ถูกกำหนดล่วงหน้าไว้เท่านั้น

2.4 ประสิทธิภาพของระบบ

การประเมินประสิทธิภาพการตรวจจับการล้ม ของ ฮาร์ดแวร์ วัดได้จากค่า Sensitivity และ ค่า Specificity ซึ่งคำนวณจากเหตุการณ์ที่เป็นไปได้ 4 เหตุการณ์ คือ

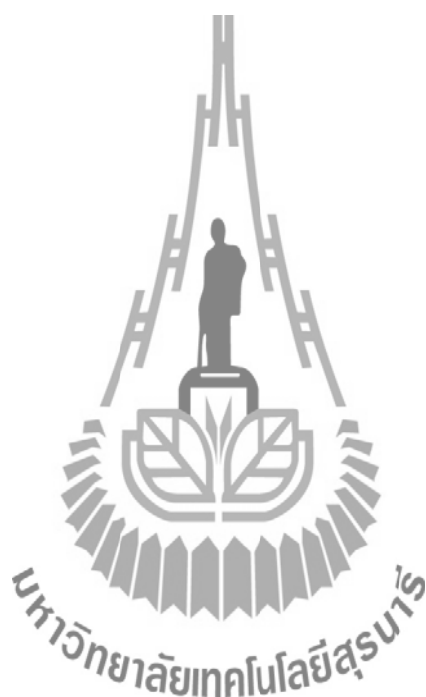
1. TP (True Positive) เกิดการล้มจริง และอุปกรณ์ฮาร์ดแวร์ ตรวจจับว่าล้มจริง
2. FP (False Positive) ไม่เกิดการล้ม แต่อุปกรณ์ฮาร์ดแวร์ ตรวจจับว่าล้ม
3. TN (True Negative) ไม่เกิดการล้ม และอุปกรณ์ฮาร์ดแวร์ ตรวจจับได้ถูกต้องว่าไม่ล้ม

4. FN (False Negative) เกิดการล้ม แต่ อุปกรณ์ฮาร์ดแวร์ ไม่สามารถตรวจจับได้ว่าล้ม โดยที่ค่า Sensitivity คือ ความสามารถในการตรวจจับอย่างถูกต้องว่าล้มจริง คำนวณได้จากสมการที่ 2.1

$$Sensitivity = \frac{TP}{TP + FN} \quad (2.1)$$

ค่า Specificity คือ ความสามารถในการตรวจจับอย่างถูกต้องว่าไม่ล้ม คำนวณได้จากสมการที่ 2.2

$$Specificity = \frac{TN}{TN+FP} \quad (2.2)$$



บทที่ 3

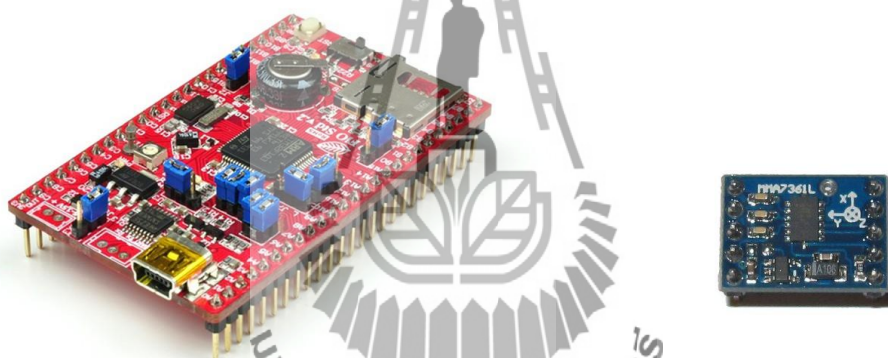
การออกแบบและการทำงานของระบบ

3.1 บทนำ

ในบทนี้จะกล่าวถึงขั้นตอนการออกแบบระบบการทำงานของระบบ การเขียนโปรแกรมควบคุมและการทำงานของ FiO Std Board, เซ็นเซอร์วัดความเร่ง 3 แกน, X-bee กับโปรแกรม MATLAB และโปรแกรม X-CTU และการเลือกใช้ค่า Threshold

3.2 การออกแบบระบบ

ระบบการตรวจรู้ของ FiO Std Board, เซ็นเซอร์วัดความเร่ง 3 แกน, X-bee และ คอมพิวเตอร์โน้ตบุ๊กแสดงเป็นแผนภาพโดยรวมดังรูปที่



FiO Std Board เซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L)



X-bee



คอมพิวเตอร์โน้ตบุ๊ก Acer Aspire 4740 332G32Mn

รูปที่ 3.1 อุปกรณ์ระบบ

โดยอุปกรณ์ที่สำคัญ คือ FiO Std Board, เซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L), X-bee และคอมพิวเตอร์โน้ตบุ๊ก โดยมีรายละเอียดดังนี้

3.2.1 FiO Std Board



รูปที่ 3.2 FiO Std Board

FiO Std Board เป็นชุดทดลองโดยใช้ไมโครคอนโทรลเลอร์ STM32TM ARM 32-bits Cortex TM M3 processors บอร์ด FiO Std นี้เป็นชุดการทดลองที่สามารถทำงานร่วมกับ MATLAB Simulink ได้ FiO Std เป็นบอร์ดทดลอง อิเล็กทรอนิกส์ และเป็นบอร์ดที่ถูกออกแบบมาเพื่อการเรียนการสอนของนักเรียนระดับมหาวิทยาลัย และมีชม ในรายวิชาด้านวิศวกรรมหลายๆ ด้าน เช่น ระบบควบคุม, ระบบอัตโนมัติ, หุ่นยนต์, ประมวลผลสัญญาณ Digital(DSP) ฯลฯ จุดเด่นของ FiO Std Board คือการใช้งานที่ง่ายโดยเฉพาะด้านการเขียนโปรแกรม เนื่องจากการเขียนโปรแกรมแบบ Graphic Programming ผ่าน Simulink ซึ่งติดตั้งมาพร้อมกัน MATLAB ซึ่งเป็น Module หนึ่งที่อยู่ในโปรแกรม MATLAB ทำให้สามารถทำความเข้าใจการทำงานของโครงงานทั้งในส่วนการทำงานของอุปกรณ์ต่างๆ การติดต่อสื่อสารระหว่างอุปกรณ์ อัลกอริทึม การเขียนโปรแกรม และอื่นๆ

คุณสมบัติหลักของบอร์ด FiO Std

1. Built-in RapidSTM Native-Support Bootloader
2. ARM 32-bit CortexTM-M3 Processor(STM32F103RET6)
3. ออสซิลเลเตอร์ 2 ตัว
4. แรงดันภายในบอร์ด 3.3 V regulator upto 800 mA

5. หน่วยความจำ 496Kbytes available flash memory
6. คอมไฟล์ และดาวโหลดอัตโนมัติเมื่อใช้ Blockset
7. หลอด LED 3 ตัว คือ สีแดง สีเหลือง สีเขียว
8. 2 user login (H/L)input jumpers Mode selection switch (see details under OPERATING MODES section)
9. ความต้านทานปรับค่าได้ 10K
10. IC เทียบสัญญาณเวลามาตรฐาน (0.33F) capacitor as RTC backup battery
11. ช่องใส่ MicroSD
12. Four built-in operating modes selectable via jumper setting
13. RoSH compliant(Real Time Operating System)

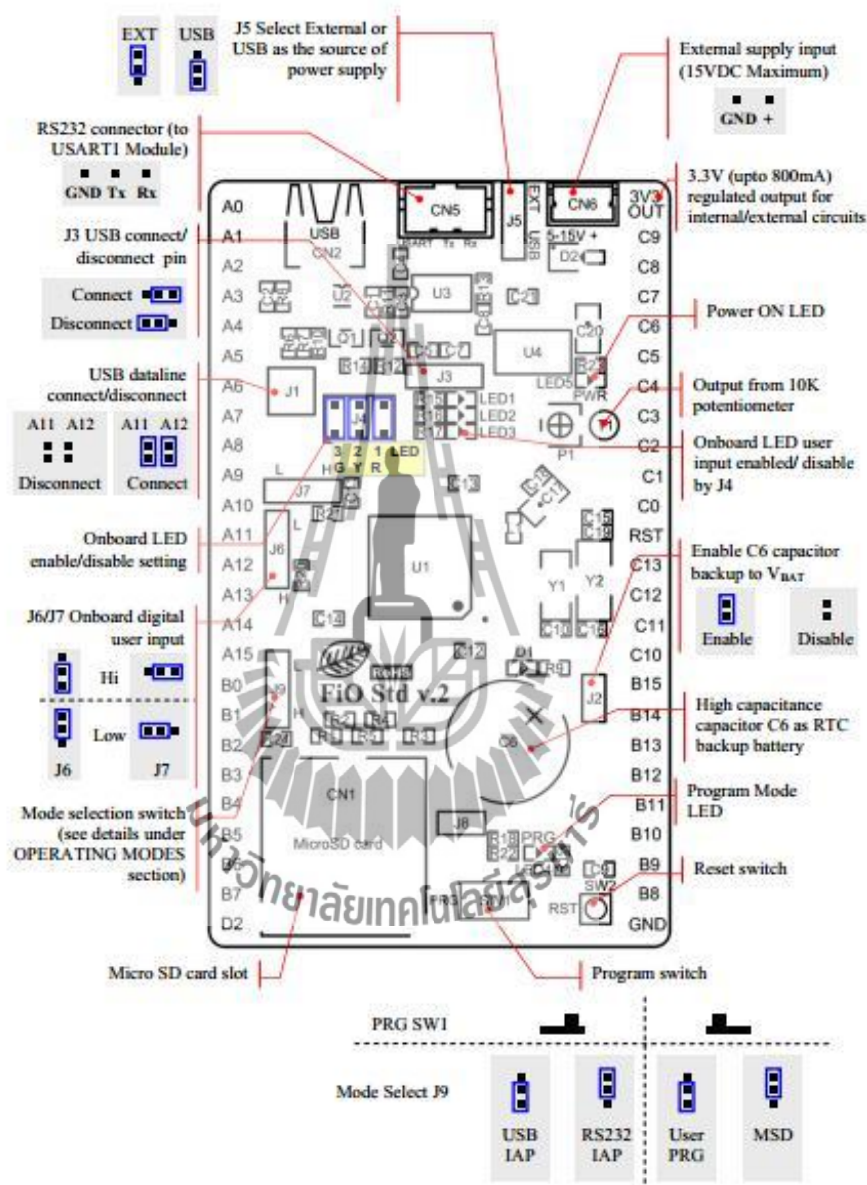
จุดเด่นของ FiO Std Board และ RapidSTM32blockset

1. ง่าย(Graphical Programming)
2. เร็ว (Rapid Prototyping, Simulation & Model-Based design)
3. ราคาไม่แพง
4. ซอฟต์แวร์มาตรฐานอุตสาหกรรม (MATLAB, Keil)
5. ฮาร์ดแวร์มาตรฐานอุตสาหกรรม (ARM-Architecture Processor)

ซอฟต์แวร์ที่ใช้งาน

1. RapidSTM32 Blockset version 0.3.6.1beta หรือเวอร์ชันที่ใหม่กว่า
2. Matlab 32-bits 2009 (version 7.8) หรือเวอร์ชันที่ใหม่กว่า
3. Simulink 2009 (version 7.3) หรือเวอร์ชันที่ใหม่กว่า
4. Real-Time Workshop 2009 (version 7.3) หรือ Simulink Coder หรือเวอร์ชันที่ใหม่กว่า
5. Real-Time Workshop Embedded Coder 2009 (version 5.3) , Simulink Coderหรือเวอร์ชันที่ใหม่กว่า
6. Real View MDK for ARM version 4.0 หรือเวอร์ชันที่ใหม่กว่า

7. Microsoft Windows XP SP2 หรือเวอร์ชันที่ใหม่กว่า
8. Microsoft .Net Framework version 3.5 หรือเวอร์ชันที่ใหม่กว่า



รูปที่ 3.3 FiO Std PCB LAYOUT

3.2.2 เซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L)

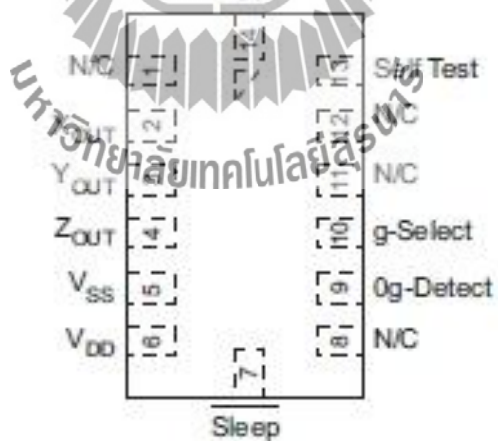


รูปที่ 3.4 เซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L)

รายละเอียด :

MMA7361L 3-Axis Accelerometer $\pm 1.5/6g$ with Voltage Regulator เป็น เซ็นเซอร์วัดสัญญาณความเร่งแบบ 3 แกน (X , Y , Z) สามารถเลือกกระดัดค่า Gravity ได้ $\pm 1.5g$ และ $\pm 6g$ ให้สัญญาณ Output เป็น Analog โดย PCB บอร์ดได้เพิ่มชุด Voltage Regulator 3.3 Volt ทำให้สามารถ Supply Voltage ได้ 2.2-16 Volt

การจัดขาเซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L) และฟังก์ชันการใช้งาน

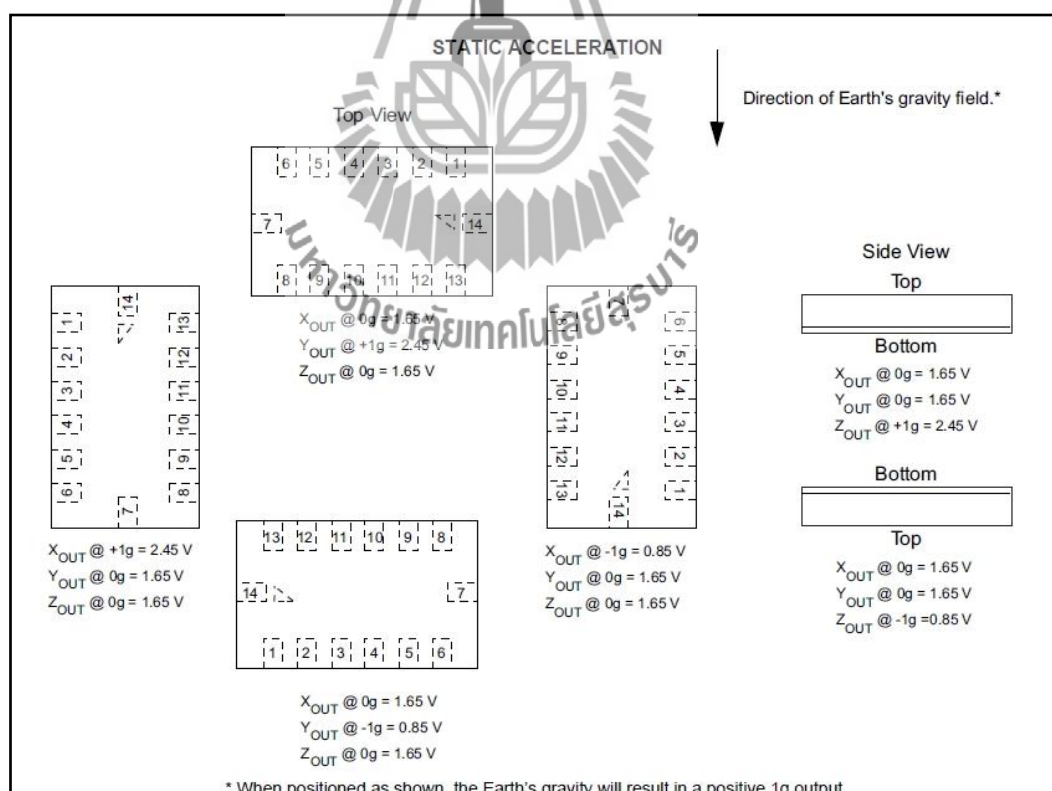
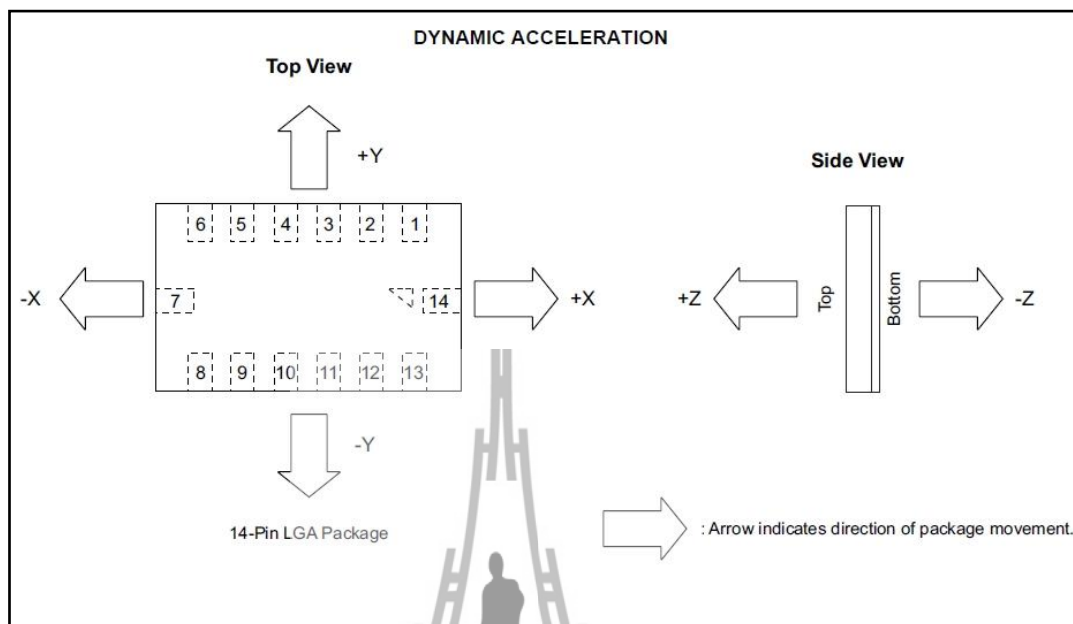


รูปที่ 3.5 การจัดขาของเซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L)

ตารางที่ 3.1 การจัดการขาเซ็นเซอร์วัดความเร่ง 3 แกน และฟังก์ชันในการทำงาน

Pin No.	Pin Name	Description
1	N/C	No internal connection Leave unconnected
2	XOUT	X direction output voltage
3	YOUT	Y direction output voltage
4	ZOUT	Z direction output voltage
5	VSS	Power Supply Ground
6	VDD	Power Supply Input
7	Sleep	Logic input pin to enable product or Sleep Mode
8	NC	No internal connection Leave unconnected
9	0g-Detect	Linear Freefall digital logic output signal
10	g-Select	Logic input pin to select g level
11	N/C	Unused for factory trim Leave unconnected
12	N/C	Unused for factory trim Leave unconnected
13	Self Test	Input pin to initiate Self Test
14	N/C	Unused for factory tri

เซ็นเซอร์วัดความเร่ง 3 แกน ตัวนี้ มีเอาต์พุตในการวางตัวแต่ละทิศทางดังนี้



รูปที่ 3.6 เอาต์พุตในทิศทางการวางตัวของเซ็นเซอร์วัดความเร่ง 3 แกน (MMA7361L)

3.2.3 X-bee (ZigBee)



รูปที่ 3.7 X-bee

ลักษณะการทำงานของ Zigbee

แบ่งได้เป็น 3 แบบ ดังนี้

1. Coordinator มีหน้าที่สร้างการสื่อสาร เชื่อมโยงเครือข่าย ระหว่าง End device กับ Router หรือ Coordinator กับ Coordinator ด้วยกัน หรือ Coordinator กับ Router กำหนดตำแหน่งที่อยู่ให้กับอุปกรณ์ที่อยู่ในวงเครือข่าย ไม่ให้ซ้ำกัน ดูแลจัดการเรื่องการจัดการเส้นทาง ซึ่งเทียบได้กับ FFD
2. End-Device เป็นอุปกรณ์ปลายทางสุด ซึ่งจะรับใช้สัญญาณจากเซ็นเซอร์ที่ปลายทาง โดยที่ใช้พลังงานต่ำในการทำงาน เทียบได้กับ RFD หรือ FFD บางกรณีขึ้นอยู่กับเซ็นเซอร์ที่ใช้
3. Router มีหน้าที่รับส่งข้อมูล ในเส้นทางต่างๆของเครือข่ายและหาเส้นทางที่ใกล้ที่สุด ซึ่งจะต้องใช้เวลาน้อยที่สุด

คุณสมบัติโดยทั่วไป

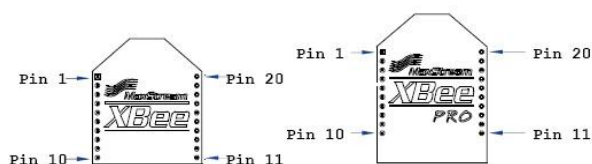
- ความถี่ในการทำงาน : 2.4 GHz
- สายอากาศ : มีสายอากาศแบบ Whip
- ระยะทำการในร่ม : สูงสุด 300 ฟุต หรือประมาณ 100 เมตร
- ระยะทำการกลางแจ้ง (แบบ line-of-sight) : สูงสุดถึง 1 ไมล์ หรือประมาณ 1,500 เมตร
- กำลังส่ง : 60 mW (18dBm)
- ความไวในการรับส่งสัญญาณ : -100dBm (1% packet error rate)
- การทำงานของขาพอร์ต : สามารถกำหนดผ่านทางซอฟต์แวร์ X-CTU เพื่อให้ทำงานเป็น
 - อินพุตอะนาล็อกสำหรับวงจรแปลงสัญญาณอะนาล็อกเป็นดิจิทัล ความละเอียด 10 บิต
 - อินพุตเอาต์พุตดิจิทัล

- ขนาด : 0.96 x 1.297 นิ้ว หรือ 2.438 x 3.294 เซนติเมตร
- ไฟเลี้ยง : 2.8 ถึง 3.3 V
- กระแสไฟฟ้า : เมื่อส่งข้อมูล 215mA รับข้อมูล 55mA น้อยกว่า 10 μ A ในโหมดลดพลังงานที่ไฟเลี้ยง +3.3V
- อุณหภูมิใช้งาน : -40 ถึง 85 °C

คุณสมบัติด้านการสื่อสารข้อมูล

- สามารถทำงานเป็นอุปกรณ์มาสเตอร์และสเลฟได้
- อัตราถ่ายทอข้อมูลผ่านคลื่นวิทยุ : 250,000 บิตต่อวินาที
- อัตราการถ่ายทอข้อมูลอนุกรม (บอดเรต) : 1,200 ถึง 115,200 บิตต่อวินาที
- รูปแบบโครงข่ายข้อมูลที่รองรับ : จุดต่อจุด (Point-to-Point) จุดต่อหลายจุด (Point-to-multipoint) และเข้ากันได้กับอุปกรณ์ตามมาตรฐาน IEEE 802.15.4
- ทางเลือกแอดเดรส : PAN ID ช่อง (Channel) และแอดเดรส (Addresses) สำหรับแอดเดรสสามารถกำหนดรหัสแอดเดรสได้มากถึง 65,000 รหัส
- เทคโนโลยีในการกระจายคลื่น : DSSS (Direct Sequence Spread Spectrum)
- รองรับการทำงานทั้งแบบ API และ AT command สามารถกำหนดได้ผ่านทางซอฟต์แวร์ X-CTU
- การรับรองมาตรฐาน : สหรัฐอเมริกา (FCC Part 15.247) OUR-XBEEPRO
แคนาดา (IC) 4214A XBEEPRO
ยุโรป (CE) ETSI (ที่กำลังส่งสูงสุด 10 dBm)
ญี่ปุ่น 005NYCA0378 (ที่กำลังส่งสูงสุด 10 dBm)

การจัดขาโมดูล X-bee และฟังก์ชันในการทำงาน



รูปที่ 3.8 X-bee และ การจัดขาของ X-bee

ตารางที่ 3.2 การจัดขาโมดูล X-bee และฟังก์ชันในการทำงาน

ขาที่	ชื่อขา/การทำงาน
1	Vcc ขาต่อไฟเลี้ยง +3.3V
2	DOUT : ขาเอาต์พุตส่งข้อมูลอนุกรม
3	DIN : ขาอินพุตรับข้อมูลอนุกรม
4	D08 : ขาเอาต์พุตดิจิตอล ช่อง 8
5	RESET : ขารีเซตหลัก (Active “0”)
6	PWM0/RSSI : ขาเอาต์พุต PWM ช่อง 0 และขาเอาต์พุตแสดงแรงของการรับสัญญาณ
7	PWM1 : ขาเอาต์พุต PWM ช่อง 1
8	ไม่ใช้งาน
9	DTR/SLEEP_RQ/DI8 : ขาอินพุตรับสัญญาณให้หยุดทำงานเข้าโหมดสลีป หรือ เป็นขาอินพุตดิจิตอลช่อง 8
10	GND : ขาต่อกราวด์
11	AD4/DIO4 : ขาอินพุตอะนาล็อก 4 หรือขาอินพุตเอาต์พุตดิจิตอล 4
12	CTS/DIO7 : อินพุตรับสัญญาณแจ้งการส่งข้อมูลจากโฮสต์ (Clear-To-Send) ใช้ในการควบคุมจังหวะการรับส่งข้อมูล หรือเป็นขาอินพุตเอาต์พุตดิจิตอล 7
13	ON/SLEEP : ขาแสดงสถานะการทำงาน “1” : อยู่ในโหมดทำงานปกติ “2” : อยู่ในโหมดสลีป
14	VREF : ขาต่อแรงดันอ้างอิงสำหรับโมดูลแปลงสัญญาณอะนาล็อกเป็นดิจิตอลภายใน X-Bee
15	Associated/AD5/DIO5 : ขาแสดงสถานะการเชื่อมต่อ หรือขาอินพุตอะนาล็อก 5 หรือ ขาอินพุตเอาต์พุตดิจิตอล 5
16	RTS/AD6/DIO6 : ขาเอาต์พุตแจ้งความพร้อมในการส่งข้อมูล (Ready-To-Send) ใช้ควบคุมจังหวะการรับส่งข้อมูล หรือเป็นขาอินพุตอะนาล็อก 6 หรือเป็นขาอินพุตดิจิตอล 6
17	AD3/DIO3 : ขาอินพุตอะนาล็อก 3 หรือ ขาอินพุตเอาต์พุตดิจิตอล 3
18	AD3/DIO2 : ขาอินพุตอะนาล็อก 2 หรือ ขาอินพุตเอาต์พุตดิจิตอล 2
19	AD3/DIO1 : ขาอินพุตอะนาล็อก 1 หรือ ขาอินพุตเอาต์พุตดิจิตอล 1
20	ADO/DIO0 : ขาอินพุตอะนาล็อก 0 หรือ ขาอินพุตเอาต์พุตดิจิตอล 0

3.2.4 คอมพิวเตอร์โน้ตบุ๊ก



รูปที่ 3.9 คอมพิวเตอร์โน้ตบุ๊ก Acer Aspire 4740 332G32Mn

คุณสมบัติของคอมพิวเตอร์โน้ตบุ๊ก

CPU – หน่วยประมวลผล :	Intel Core i3-330M processor (2.13GHz, 3MB L2 Cache)
GPU - การ์ดจอ แสดงผล :	Nvidia GeForce 310M with up to 1243MB TurboCache
Chipset :	Mobile Intel HM55 Express Chipset
Memory :	2048 MB DDR3-1066 (สูงสุด 4 GB)
Hard disk :	320 GB HDD 5400 (รอบต่อนาที)
Optical Drive :	DVD-SuperMulti Double Layer Drive
ระบบปฏิบัติการ :	Linpus Linux BE
Battery :	6 Cell Lithium-ion
หน้าจอแสดงผล :	14 นิ้ว แบบ TFT LCD HD Acer CineCrystal LED- Backlit ความละเอียด 1366x768 Pixel (อัตราส่วน 16:9)
GPU - การ์ดจอ แสดงผล :	Nvidia GeForce 310M with up to 1243MB TurboCache
Wireless Lan :	Intel Wireless WiFi Link 5100
Bluetooth :	Integrated Bluetooth 2.1+EDR
USB :	3 ช่อง : USB 2.0

หมายเหตุ สามารถใช้คอมพิวเตอร์รุ่นอื่นได้

3.3 การทำงานของระบบ

1. ภาคส่งทำการส่งค่าความเร่งจากฮาร์ดแวร์ที่ประกอบไปด้วย เซ็นเซอร์วัดความเร่ง 3 แกนเชื่อมต่อกับFiO Std BoardและX-bee โดยส่งผ่าน X-Bee ตัวที่เป็น Router ไปยัง X-bee ตัวรับที่เป็น Coordinator

2. การประมวลผลจะผ่าน โปรแกรมที่สร้างไว้ใน FiO Std Board โดยใช้ค่า Threshold ในการเปรียบเทียบและประมวลผล

3. ภาครับ จะรับโดย X-bee ตัวที่เป็น Coordinator โดยจะทำการแสดงผลรายงานผ่านที่ หน้าจอคอมพิวเตอร์ว่ามีการหกล้มเกิดขึ้น

3.4 การเขียนโปรแกรมควบคุมระบบการทำงานของฮาร์ดแวร์และการสร้างอุปกรณ์ฮาร์ดแวร์

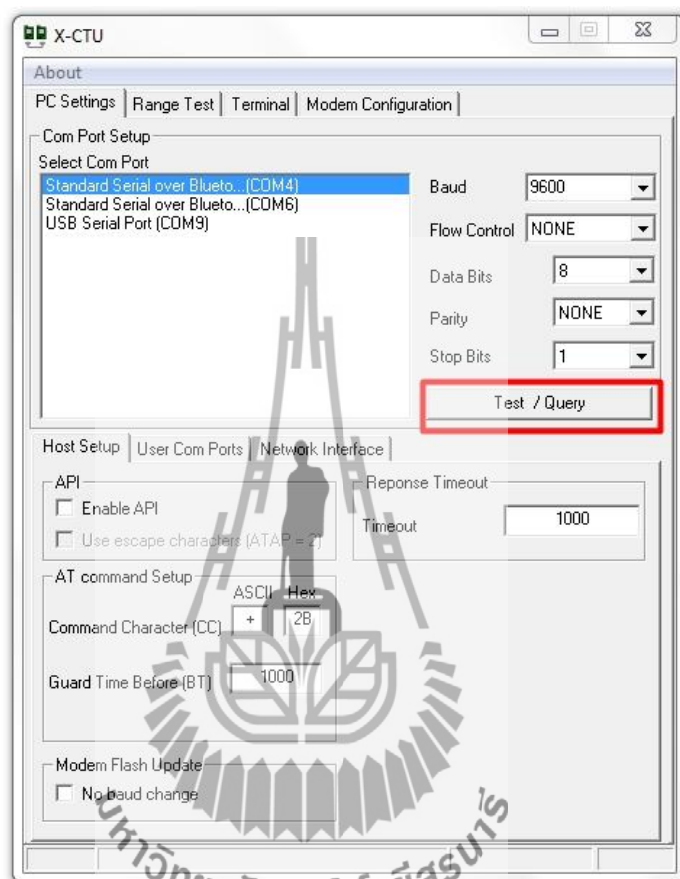
ในการเขียนโปรแกรมนั้น ในภาครับและภาคส่งเราใช้ X-bee ในการส่งข้อมูลหากัน ดังนั้นเราจะต้องทำให้ X-bee ภาครับและภาคส่งสามารถติดต่อกันได้ โดยใช้โปรแกรม X-CTU ทำได้ดังนี้

1. ติดตั้งโปรแกรม X-CTU เวอร์ชันล่าสุดลงในคอมพิวเตอร์
2. ติดตั้งโมดูล X-bee ลงบนบอร์ด X-bee USB Adapter

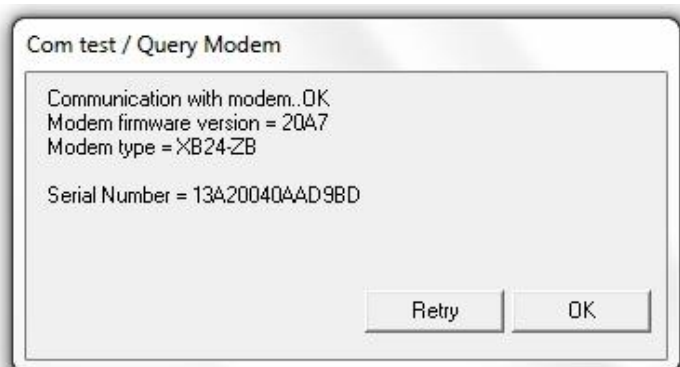


รูปที่ 3.10 การติดตั้งโมดูล X-bee ลงบนบอร์ด X-bee USB Adapter

3. เปิดโปรแกรม X-CTU และกดปุ่ม Test เพื่อทดสอบการเชื่อมต่อระหว่าง X-bee กับโปรแกรม X-CTU ว่าสามารถติดต่อกันได้หรือไม่ หากติดต่อกันได้จะปรากฏหน้าต่างแจ้งผลการติดต่อและข้อมูลของ X-bee เบื้องต้นดังรูปที่ 3.12

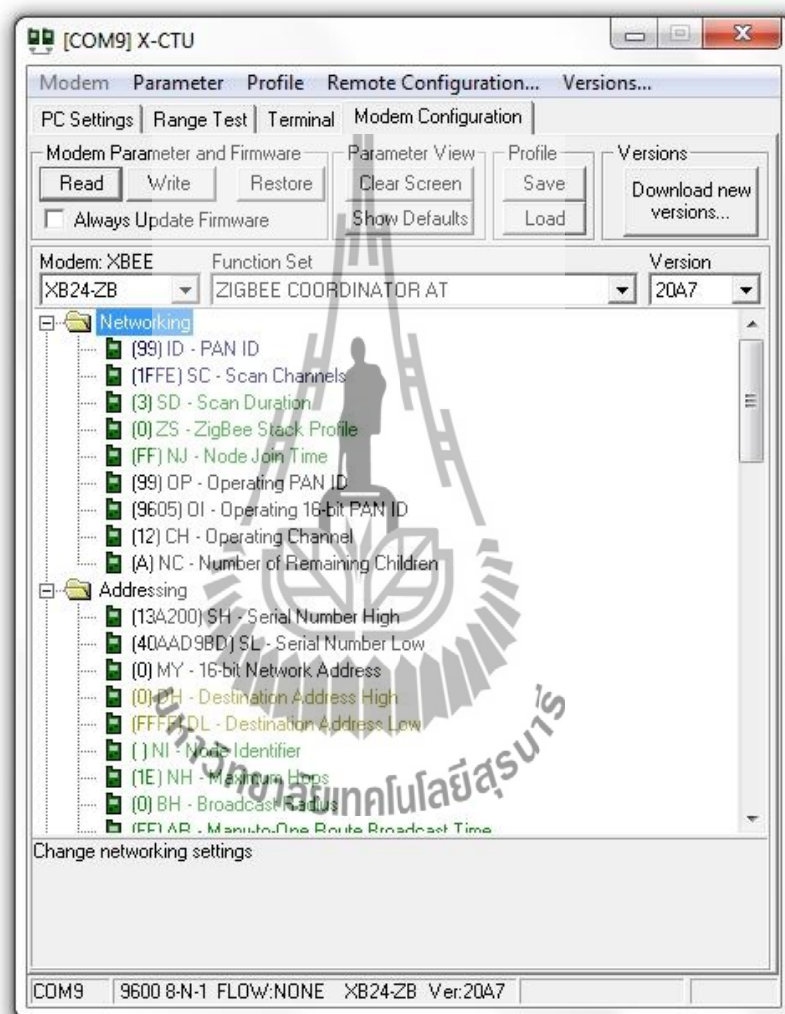


รูปที่ 3.11 หน้าต่างโปรแกรม X-CTU



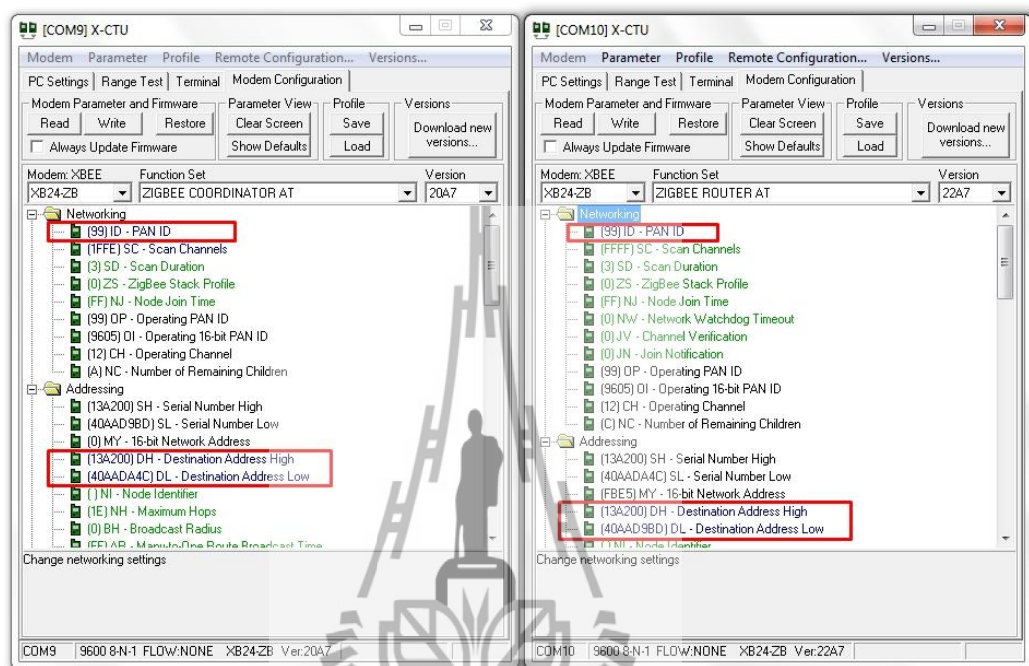
รูปที่ 3.12 หน้าต่างแจ้งผลการติดต่อ

4. หลังจากนั้นคลิกไปที่แท็บ Modem Configuration แล้วกดปุ่ม Read ในกรอบ Modem Parameter and Firmware จะปรากฏข้อมูลชื่อรุ่นของโมดูล X-bee (Modem Type) ชื่อฟังก์ชัน หมายเลขเวอร์ชันของเฟิร์มแวร์ และค่าพารามิเตอร์ต่างๆ ดังรูปที่ 3.13



รูปที่ 3.13 ผลการอ่านค่าจากโมดูล X-bee ก่อนการตั้งค่า

5. ตั้งค่าการรับส่งข้อมูลให้กับ X-bee โดยทำการตั้งค่าให้ 1 ตัวเป็นโหนด Coordinator (Rx) และ อีก 1 ตัวเป็นโหนด Router (Tx) โดยมีการเชื่อมต่อกันแบบเมช

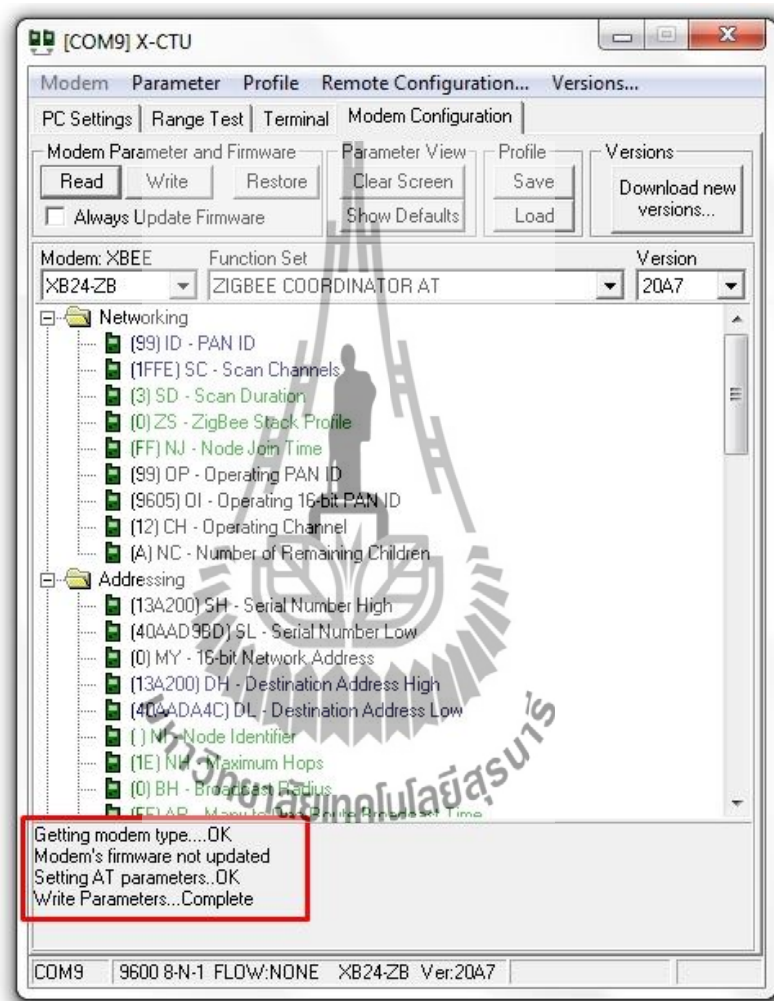


Coordinator

router

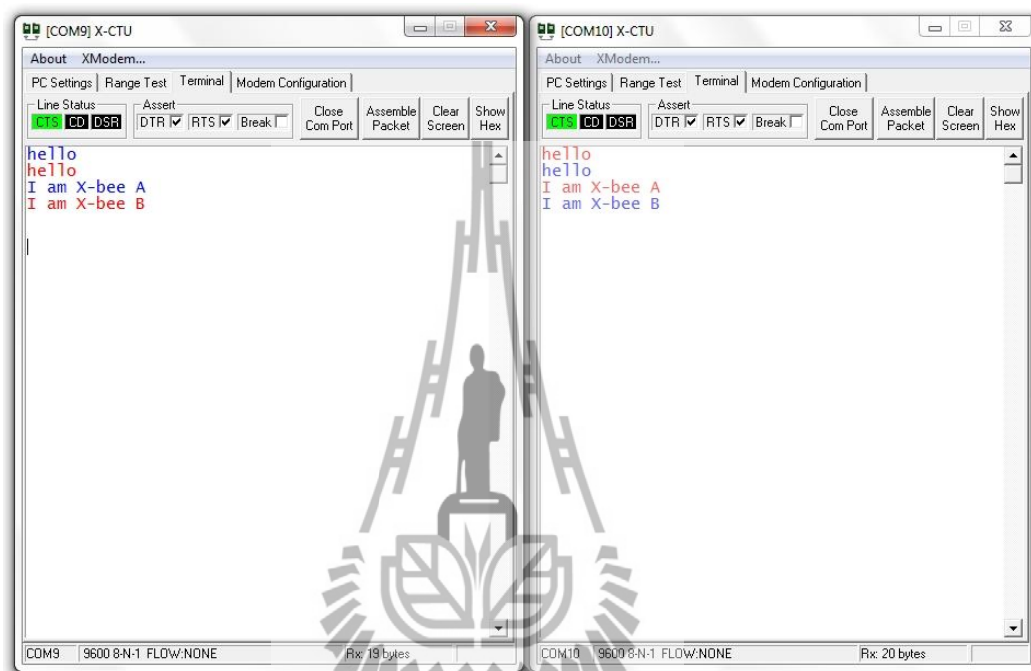
รูปที่ 3.14 การตั้งค่า X-bee

6. จากนั้นกดปุ่ม Write รอสักครู่ สังเกตด้านล่างของหน้าต่าง Modem Configuration จะแสดงข้อความเพื่อแจ้งสถานการณ์ทำงาน หากเป็นดังรูปที่ 3.15 แสดงว่าการกำหนดค่าเสร็จสมบูรณ์



รูปที่ 3.15 แสดงข้อความแจ้งสถานการณ์ทำงาน

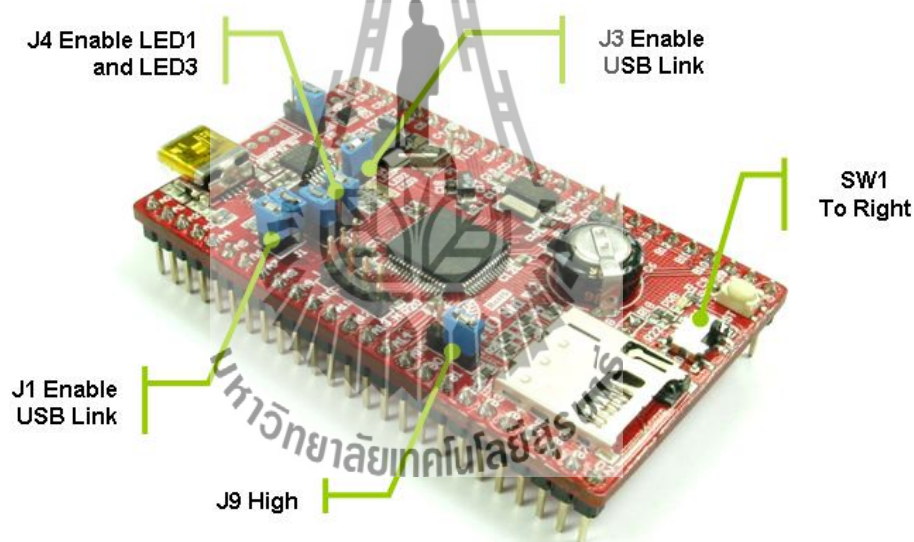
7. เราสามารถทดสอบว่า X-bee สามารถรับส่งข้อมูลหากันได้หรือไม่ โดยคลิกไปที่แท็บ Terminal ของทั้งสองตัว แล้วพิมพ์ข้อความ หากรับส่งข้อมูลกันได้ จะปรากฏดังรูปที่ 3.16 โดยที่ข้อความที่ส่งจะเป็นสีแดง และข้อความที่ได้รับเป็นสีน้ำเงิน



รูปที่ 3.16 X-bee มีการรับส่งข้อมูลผ่านโปรแกรม X-CTU

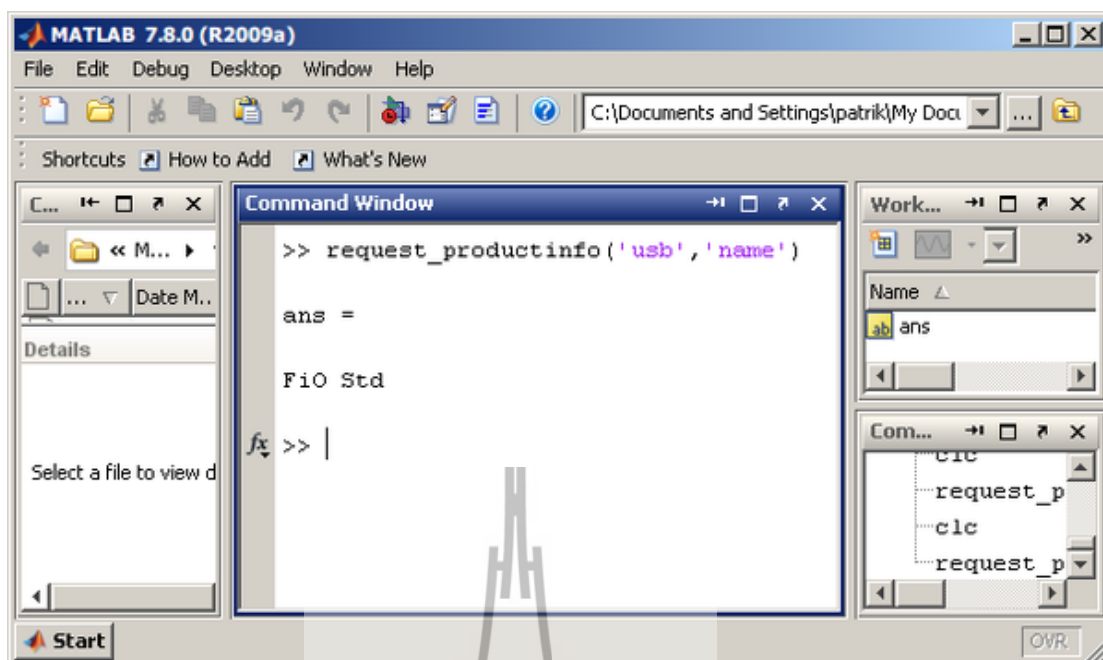
เนื่องจากเราใช้ FiO Std Board ในการสร้างฮาร์ดแวร์ เราจึงใช้ MATLAB ในการเขียนโปรแกรม โดยใช้ Block ของ Simulink โดยที่อันดับแรกเราจะทำการเชื่อมต่อ FiO Std Board กับคอมพิวเตอร์โน้ตบุ๊กให้สามารถเชื่อมต่อกันได้ก่อน แล้วจากนั้นจึงทำการเชื่อมต่อ FiO Std Board เข้ากับเซ็นเซอร์วัดความเร่ง 3 แกน และ X-bee เมื่อเชื่อมต่อเสร็จก็ทำการสร้างบล็อกในโปรแกรม Simulink MATLAB สามารถทำได้โดย

1. ทำการเชื่อมต่อ FiO Std Board ไปที่เครื่องคอมพิวเตอร์โน้ตบุ๊ก และทำการตั้งค่า FiO Std Board ให้อยู่ในโหมดโปรแกรมโดยในโหมดนี้ FiO Std Board สามารถตั้งค่าโปรแกรมผ่านทาง USB และตั้งค่า FiO Std ให้อยู่ในโหมดโปรแกรม โดยตั้งค่า Switch (SW1) และ jumper (J1, J2, J9) ดังรูปที่ 3.17 และต้องกดปุ่ม Reset เมื่อตั้งค่าทุก Jumpers และ Switches แล้ว



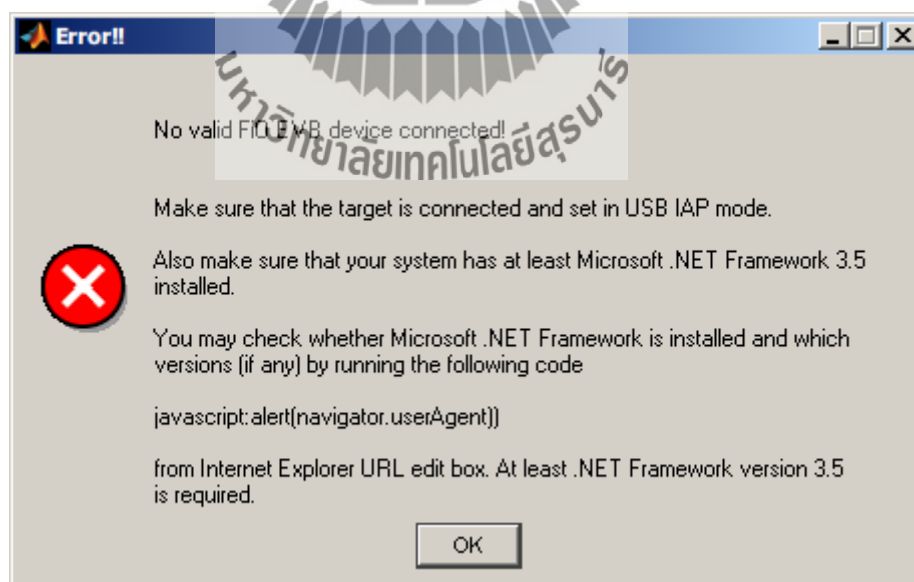
รูปที่ 3.17 การตั้งค่า Jumper และ Switches

2. เปิดโปรแกรม MATLAB ทดสอบการเชื่อมต่อ โดยตรวจสอบว่ามีการเชื่อมต่อ โดยพิมพ์คำสั่งลงบนหน้า Command Window ว่า `request_productinfo('usb','name')` หากบอร์ดมีการเชื่อมต่อจะปรากฏ FiO Std ดังรูปที่ 3.18



รูปที่ 3.18 การทดสอบการเชื่อมต่อ FiO Sta Board

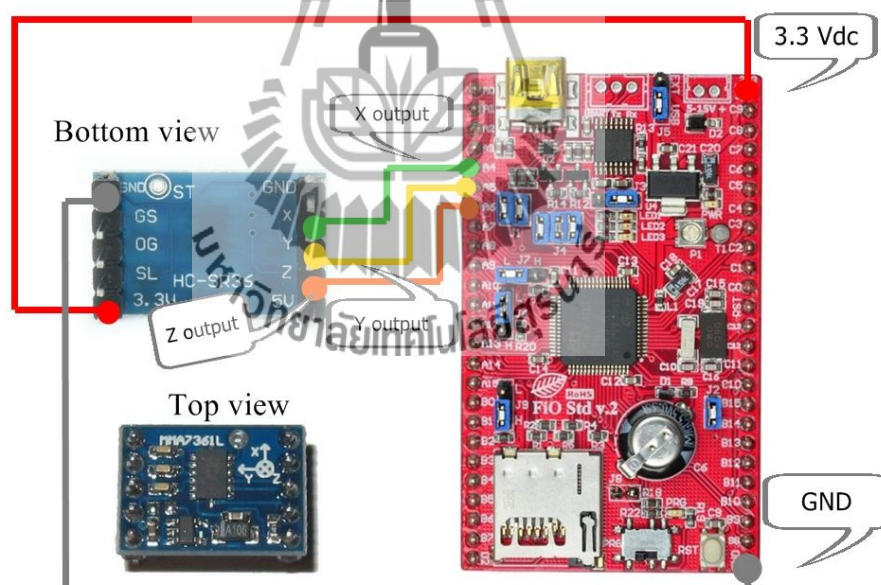
หากไม่พบบอร์ด จะปรากฏหน้าต่าง Error ดังรูปที่ 3.19 ให้ทำซ้ำในขั้นตอนที่ 1-2 ใหม่



รูปที่ 3.19 แสดง Error เมื่อไม่พบบอร์ด

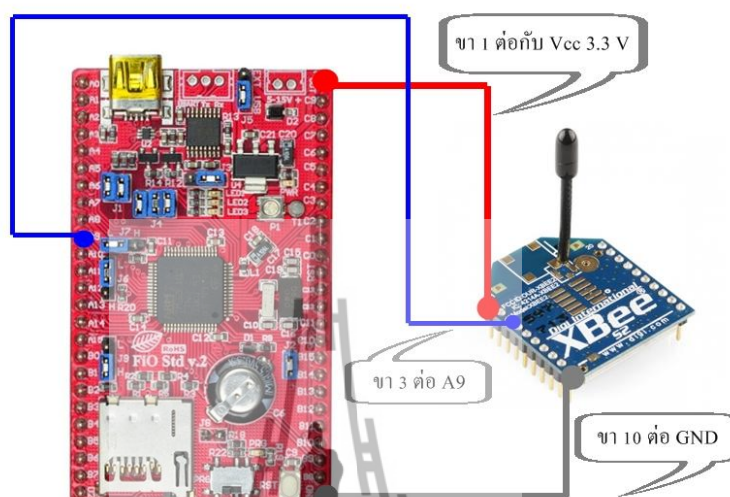
เมื่อ FiO Std Board สามารถทำการเชื่อมต่อกับคอมพิวเตอร์ได้แล้ว จากนั้นเราทำการเชื่อมต่อเซ็นเซอร์วัดความเร่ง 3 แกน และ X-bee เข้ากับ FiO Std Board และทำการสร้างบล็อกในโปรแกรม Simulink MATLAB เพื่อที่จะโปรแกรมลงไปบน FiO Std Board โดยที่ภาคส่งเราจะสร้างฮาร์ดแวร์ขึ้นมาเพื่อทำเป็นระบบตรวจจับพฤติกรรมการหกล้มโดยใช้เซ็นเซอร์วัดความเร่งแบบ 3 แกน และที่ภาครับเราจะใช้เพียง X-bee เชื่อมต่อกับคอมพิวเตอร์ที่รับข้อมูลว่ามีการหกล้มเกิดขึ้น แต่ขั้นต้นเราจะต้องมีการตรวจสอบว่าข้อมูลที่เรารับมานั้นตรงกับข้อมูลที่ส่งมา ซึ่งอธิบายไว้ใน ภาคผนวก ก เมื่อสามารถรับและส่งได้ตรงกันแล้ว จากนั้นจึงทำดังนี้

1. ทำการเชื่อมต่อเซ็นเซอร์วัดความเร่ง 3 แกน เข้ากับ FiO Std Board โดยสมมติเลือก ADC1_IN4, ADC1_IN5 และ ADC1_IN6 เป็นขาที่เชื่อมต่อกับ X output, Y output และ Z output จากเซ็นเซอร์ตามลำดับ ดังรูปที่ 3.20



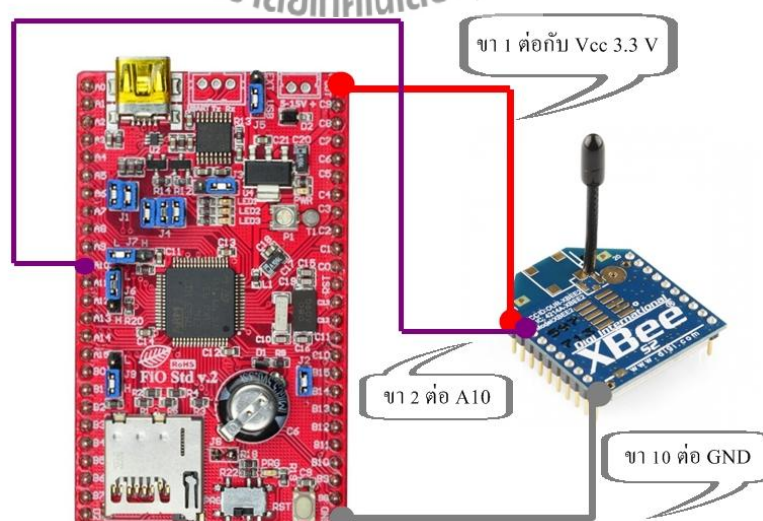
รูปที่ 3.20 การเชื่อมต่อเซ็นเซอร์วัดความเร่ง 3 แกน เข้ากับ FiO Std Board

2. ภาคล่งหลังจากเชื่อมต่อเซ็นเซอร์วัดความเร่งแบบ 3 แกน กับ FiO Std Board เรียบร้อยแล้ว จากนั้นเชื่อม X-bee เข้ากับ FiO Std Board โดยเลือก USART1 ให้ Tx เป็นขา A9 ของ FiO Std Board เพื่อเชื่อมต่อกับขาอินพุตของ X-bee ที่กำหนดให้เป็น Router ดังรูปที่ 3.21



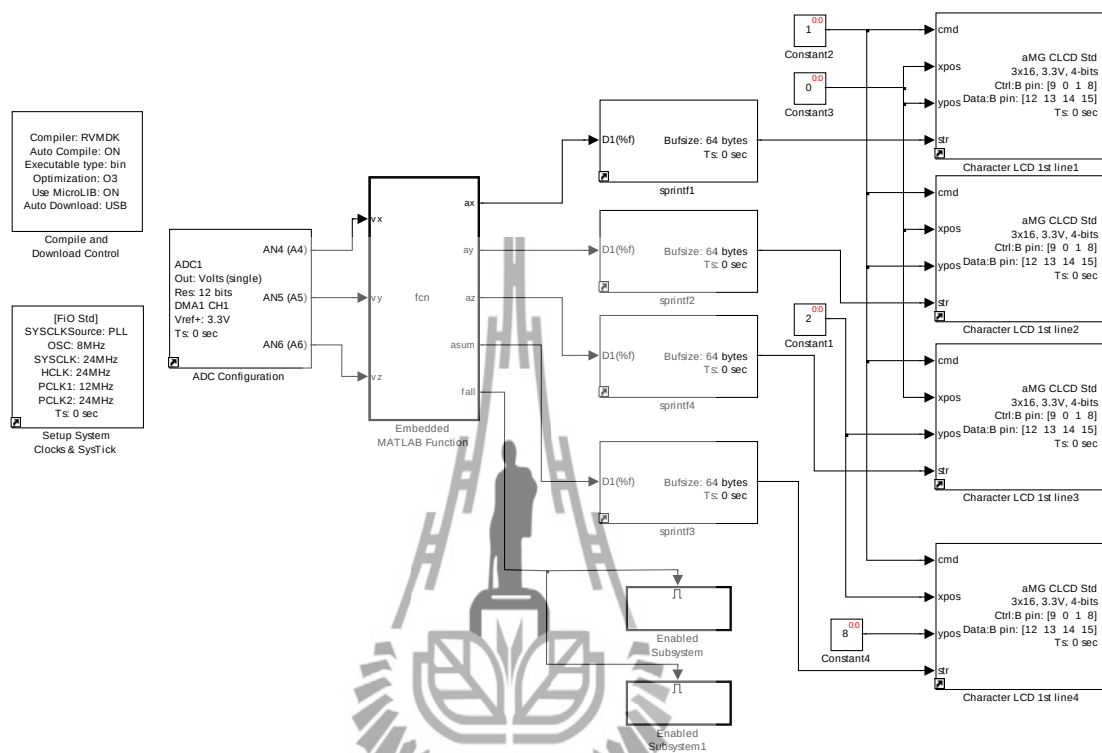
รูปที่ 3.21 การเชื่อมต่อX-bee ภาคล่ง เข้ากับ FiO Std Board

3. ส่วนภาครับ ทำการเชื่อม X-bee เข้ากับ FiO Std Board ที่เป็นบอร์ดรับ โดยเลือก USART1 ให้ Rx เป็นขา A10 ของ FiO Std Board เพื่อเชื่อมต่อกับขาอินพุตของ X-bee ที่กำหนดให้เป็น Coordinator ดังรูปที่ 3.22



รูปที่ 3.22 การเชื่อมต่อX-bee ภาครับ เข้ากับ FiO Std Board

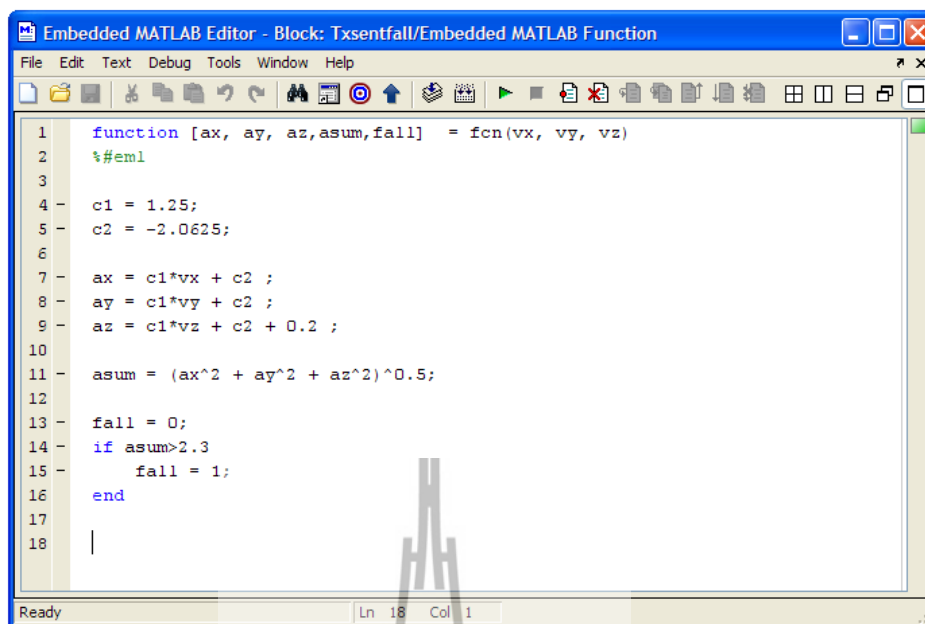
4. สร้างบล็อกใน Simulink ดังรูปที่ 3.23 เพื่อทำการโปรแกรมลง ฮาร์ดแวร์ภาคส่งที่สร้างขึ้นเป็นระบบตรวจจับพฤติกรรมหกล้มโดยเซ็นเซอร์วัดความเร่งแบบ 3 แกน และทำการโหลดโปรแกรมลง FiO Std Board



รูปที่ 3.23 บล็อก Simulink ของ ฮาร์ดแวร์ ภาคส่ง

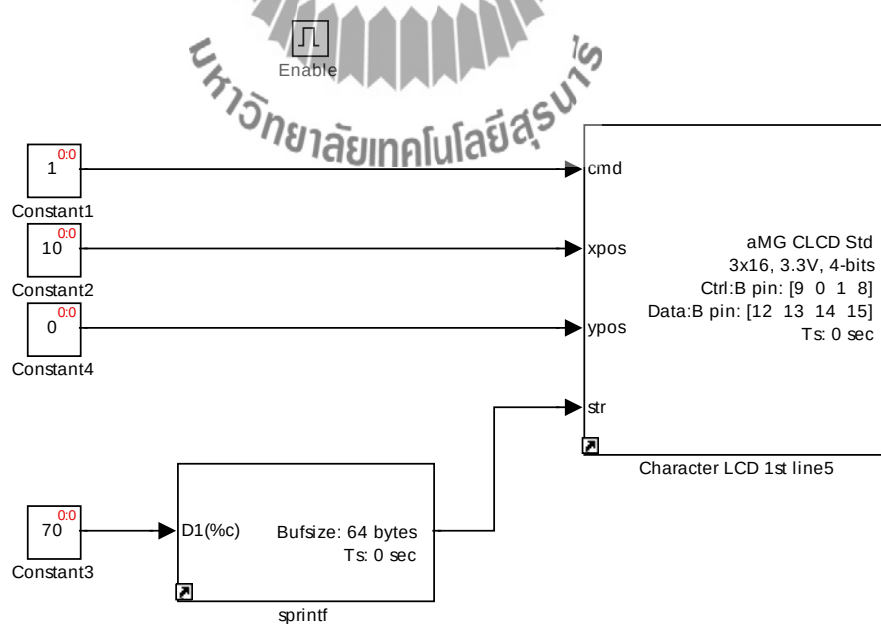
หมายเหตุ

- บล็อก Embedded MATLAB Function มีการกำหนดสมการ ดังรูปที่ 3.24
- บล็อก Enable Subsystem ได้มีการสร้างบล็อกภายในดังรูปที่ 3.25
- บล็อก Enable Subsystem 1 มีการสร้างบล็อกภายใน ดังรูปที่ 3.26

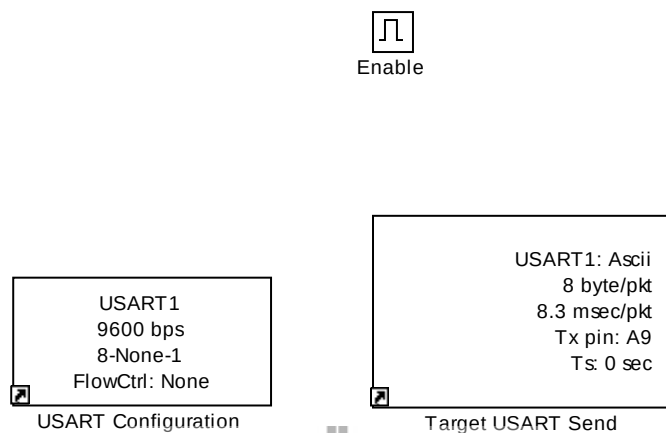


รูปที่ 3.24 สมการในบล็อก Embedded MATLAB Function

หมายเหตุ ในบรรทัดที่ 13-16 ของบล็อก Embedded MATLAB Function นี้ เราเพียงสมมติค่า Fall ซึ่งก็คือค่า Threshold เพื่อทดสอบการแจ้งเตือนของภาครับก่อน จากนั้นเราจะทำการทดสอบเพื่อหาค่า Threshold ที่เราใช้งานจริงมาใส่ลงไปแทน



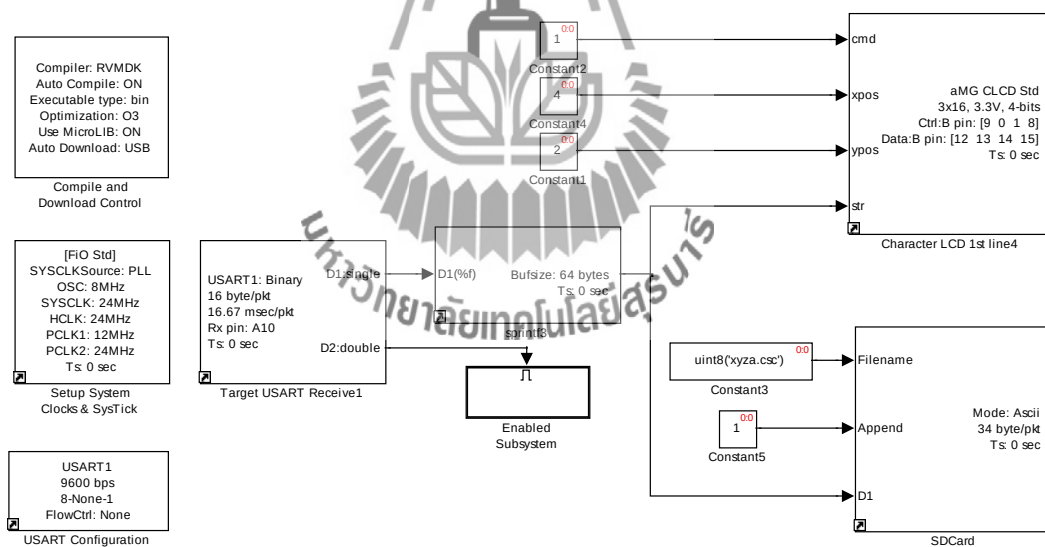
รูปที่ 3.25 บล็อกภายใน Enable Subsystem ของภาคลส่ง



รูปที่ 3.26 บล็อกภายใน Enable Subsystem ของภาคส่ง

5. สร้างบล็อกใน

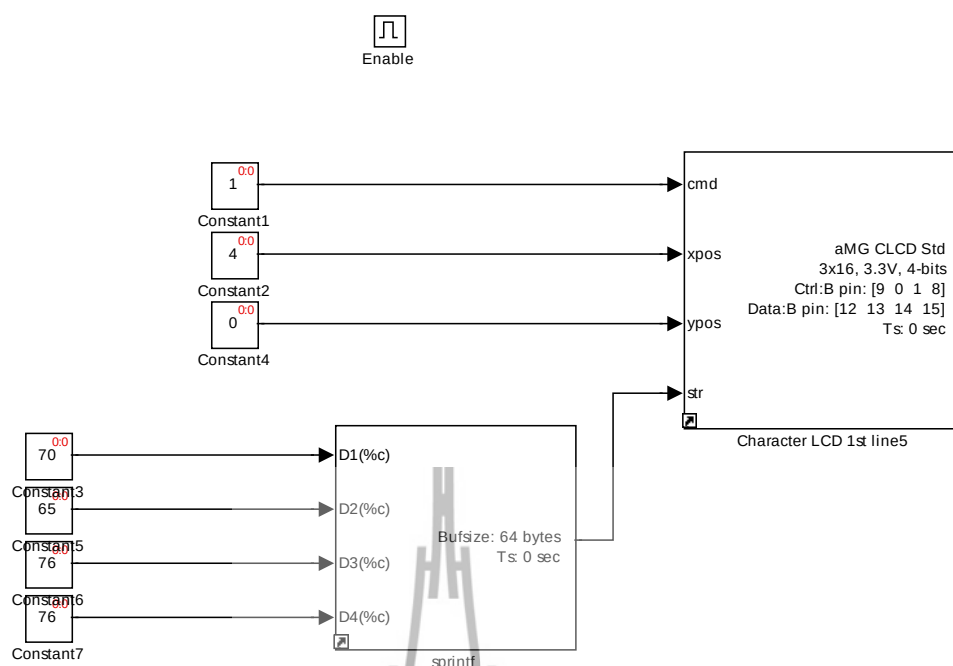
Simulink ดังรูปที่ 3.27 เพื่อทำการโปรแกรมลง ฮาร์ดแวร์ ภาครับ



รูปที่ 3.27 บล็อก Simulink ของ ฮาร์ดแวร์ ภาครับ

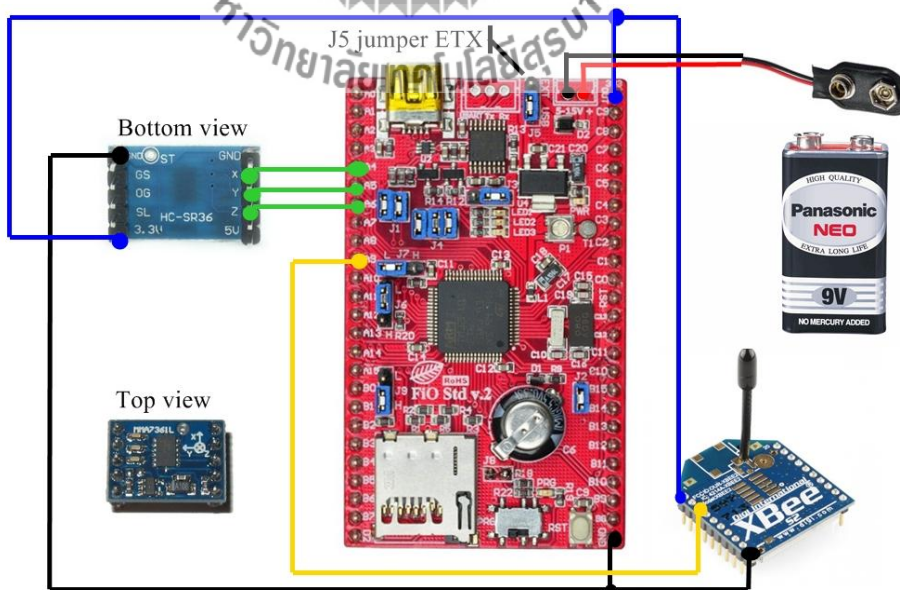
หมายเหตุ

- บล็อก Enable Subsystem ได้มีการสร้างบล็อกภายในดังรูปที่ 3.28



รูปที่ 3.28 บล็อกภายใน Enable Subsystem ของภาครับ

6. เมื่อทำการโหลดโปรแกรมลง FIO Std Board ลงทั้งภาครับและภาคส่ง แล้วทำการทดสอบว่าทั้งสองบอร์ดส่งข้อมูลได้ตรงกันแล้วหรือไม่ หากตรงกันแล้วจึงทำ ฮาร์ดแวร์ ของเราให้ไร้สายเพื่อทำการทดลองการหกล้ม โดยต่อถ่าน 9V เข้าเพื่อให้มีไฟเลี้ยงจ่ายเข้าบอร์ด แล้ว jumper J5 ไปที่ EXT เพื่อรับไฟเลี้ยงจากภายนอก ดังรูปที่ 3.29



รูปที่ 3.29 ฮาร์ดแวร์ไร้สาย

7. ทำการทดสอบการหกล้ม โดยเก็บค่าจาก Memory ที่ทำการบันทึกค่าขณะทดลองการหกล้ม โดยค่าเฉลี่ยที่ได้นั้น มาจากการหาผลรวมของค่าในแนวแกน x แกน y และแกน z ดังสมการ (3.1)

$$a_{sum} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (3.1)$$

เมื่อ a_x คือ ค่าความเร่งในแนวแกน x

a_y คือ ค่าความเร่งในแนวแกน y

a_z คือ ค่าความเร่งในแนวแกน z

โดยที่ค่าความเร่งในแนวแกน x แกน y และแกน z ได้มาจากเซ็นเซอร์วัดความเร่ง 3 แกน ที่ผ่านการคำนวณจากสมการในบล็อก Embedded MATLAB Function จากรูปที่ 3.23 และได้ทำการใส่สมการค่า a_{sum} เรียบร้อยแล้ว สามารถนำค่ามาวิเคราะห์ได้เลย

8. ทำการทดสอบการล้ม 4 ท่า คือ ล้มไปข้างหน้า ล้มไปข้างหลัง ล้มไปทางซ้าย และล้มไปทางขวา ทำซ้ำท่าละสิบครั้ง จากการทดลอง ทำให้ได้ค่าขณะหกล้ม ดังตารางที่ 3.3

ตารางที่ 3.3 ตารางค่าความเร่งขณะล้ม

ท่าล้ม ครั้งที่	ล้มไปข้างหน้า	ล้มไปข้างหลัง	ล้มไปทางซ้าย	ล้มไปทางขวา
1	2.29	2.06	2.24	1.76
2	2.16	1.36	1.58	1.57
3	1.74	1.34	2.91	2.42
4	2.35	1.33	1.49	1.16
5	1.09	1.44	1.64	1.26
6	1.55	1.37	1.83	1.89
7	2.37	1.52	1.6	1.29
8	2.00	2.27	1.07	1.88
9	2.03	1.34	2.43	3.14
10	2.29	2	1.25	1.81
ค่าเฉลี่ย	1.987	1.832	1.804	1.818

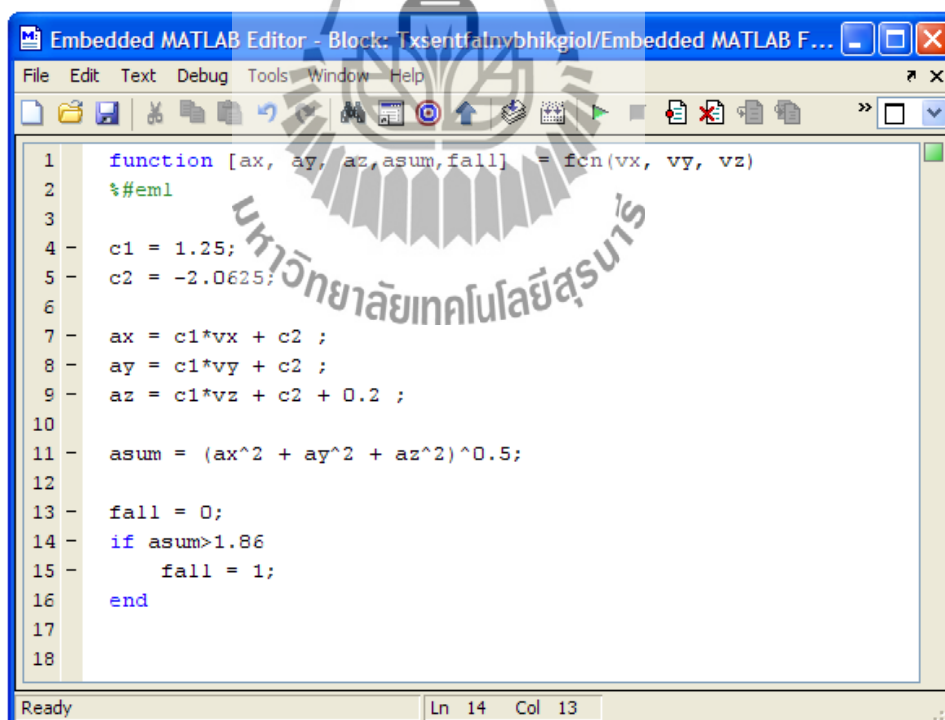
3.5 การเลือก Threshold

ทำการลี้มทั้งหมด 4 ทำ ได้แก่ ลี้มไปข้างหน้า ข้างหลัง ด้านข้างทางซ้าย และด้านข้างทางขวา ลี้มท่าละ 10 ครั้ง แล้วหาค่าเฉลี่ยของลี้ม ถ้าค่าที่รับมามากกว่า Threshold แสดงว่ามีการหกลี้ม ดังนั้นค่า Threshold ที่ใช้คือ

$$Threshold = \frac{\sum_{i=1}^N X_i}{N}$$

โดยที่ X_i คือ ค่าของลี้มในแต่ละครั้ง
 N คือ จำนวนครั้งที่ลี้ม

นำค่า Threshold ที่เราเลือกใช้มาใส่เงื่อนไขในบล็อก Embedded MATLAB Function จากรูปที่ 3.24 ได้เป็นดังรูปที่ 3.30



รูปที่ 3.30 ใส่เงื่อนไขค่า Threshold ใน บล็อก Embedded MATLAB Function

เมื่อเราได้ค่า Threshold และได้ฮาร์ดแวร์ที่เสร็จสมบูรณ์แล้ว จึงเปลี่ยนภาครับจาก FiO Std Board กับ X-bee ให้เป็นการรับค่าผ่าน X-bee โดยส่งข้อมูลโดยตรงมายังคอมพิวเตอร์โน้ตบุ๊ก ในการแจ้งเตือนว่ามีการหกล้มเกิดขึ้น ดังรูปที่ 3.31



รูปที่ 3.31 ภาครับส่งข้อมูลผ่าน X-bee มายัง คอมพิวเตอร์โน้ตบุ๊ก



บทที่ 4

ผลการทดลอง

4.1 บทนำ

ในบทนี้จะกล่าวถึงวิธีการทดลอง ผลการทดลองว่าหากหกล้มแล้ว ระบบมีการตรวจจับเหตุการณ์และ มีการแจ้งเตือนตามที่เกิดเหตุการณ์นั้นจริงหรือไม่ และหากไม่มีการหกล้มเกิดขึ้น ระบบจะสามารถตรวจจับเหตุการณ์ว่าไม่มีการหกล้มได้อย่างถูกต้องหรือไม่

4.2 ทดสอบการหกล้ม

4.2.1 ขั้นตอนการทดลอง

1. ติดตั้งอุปกรณ์ Hardware ไว้ที่ตัว และติดตั้ง X-bee ภากรับ ไว้ที่คอมพิวเตอร์โน้ตบุ๊ก

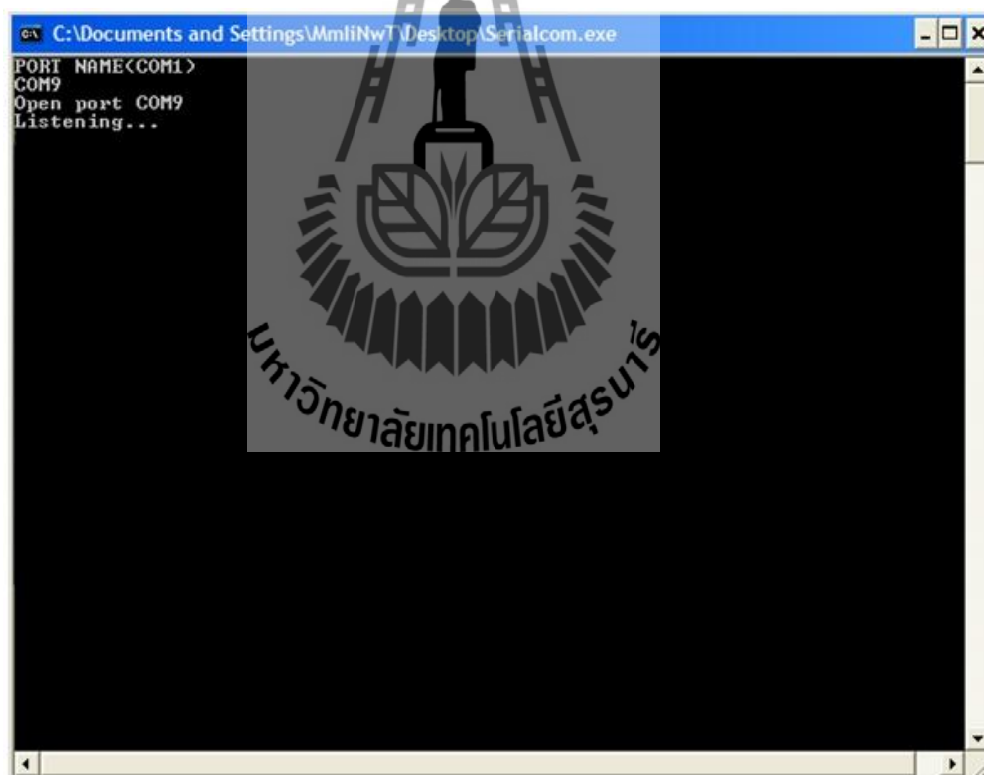


รูปที่ 4.1 ติดตั้งอุปกรณ์ Hardware ที่ตำแหน่งเอาจของผู้ทดสอบ



รูปที่ 4.2 ติดตั้ง X-bee ภาครับเข้ากับคอมพิวเตอร์โน้ตบุ๊ก

2. ทำการเปิดโปรแกรมรับค่าจากX-bee ขึ้นมา เปิดตัวอุปกรณ์ Hardware และเปิดพอร์ตของX-bee ตัวรับที่โปรแกรมรับค่า พร้อมทำการทดลอง



รูปที่ 4.3 เปิดโปรแกรมของภาครับ และเปิด Port ที่เราใช้สำหรับ X-bee ภาครับ

3. ให้ผู้ทดสอบยืนอยู่กับที่ประมาณ 10 วินาทีเพื่อ จากนั้นให้ผู้ทดสอบ ทดสอบด้วยการล้มไปข้างหน้า บันทึกผลที่ได้

4. ทำตามขั้นตอนที่ 1-3 เพื่อบันทึกค่าการล้มไปข้างหน้าอีก 9 ครั้ง จากนั้นทำการทดลองซ้ำโดยเปลี่ยนจากการล้มไปข้างหน้าเป็น ล้มไปข้างหลัง , ล้มไปทางซ้าย ล้มไปทางซ้าย ล้มไปทางขวา นอนหงาย นอนคว่ำ นอนตะแคงซ้าย นอนตะแคง ขวา เดิน และลุกขึ้นยืน อย่างละ 10 ครั้ง บันทึกผลการทดลอง

5. หาค่า Sensitivity และค่า Specificity

6. วิเคราะห์ผลการทดลอง

4.2.1 ผลการทดลอง

ผลการทดลองการตรวจจับการหกล้มทั้งหมด 10 พฤติกรรม ประกอบด้วย การล้มไปข้างหน้า ล้มไปข้างหลัง ล้มไปทางซ้าย ล้มไปทางขวา นอนหงาย นอนคว่ำ นอนตะแคงซ้าย นอนตะแคงขวา เดิน และลุกขึ้นยืน

ตารางที่ 4.1 แสดงผลการทดสอบของผู้ทดสอบคนที่ 1

Activity	Test	Accuracy
ล้มไปข้างหน้า	5	2
ล้มไปข้างหลัง	5	2
ล้มไปทางซ้าย	5	2
ล้มไปทางขวา	5	4
นอนหงาย	5	5
นอนคว่ำ	5	5
นอนตะแคงซ้าย	5	5
นอนตะแคงขวา	5	5
เดิน	5	5
ลุกขึ้นยืน	5	2

TP = 10 FP = 3 TN = 27 FN = 10

Sensitivity = $10/(10+10)$ = 0.5 = 50%

Specificity = $27/(27+3)$ = 0.9 = 90%

ตารางที่ 4.2 แสดงผลการทดสอบของผู้ทดสอบคนที่ 2

Activity	Test	Accuracy
ล้มไปข้างหน้า	5	3
ล้มไปข้างหลัง	5	2
ล้มไปทางซ้าย	5	3
ล้มไปทางขวา	5	4
นอนหงาย	5	5
นอนคว่ำ	5	5
นอนตะแคงซ้าย	5	5
นอนตะแคงขวา	5	5
เดิน	5	5
ลุกขึ้นยืน	5	4

$$\begin{aligned}
 TP &= 12 & FP &= 1 & TN &= 29 & FN &= 8 \\
 \text{Sensitivity} &= 8/(8+12) & &= 0.6 & &= 60\% \\
 \text{Specificity} &= 29/(29+1) & &= 0.97 & &= 97\%
 \end{aligned}$$

ตารางที่ 4.3 แสดงผลการทดสอบของผู้ทดสอบคนที่ 3

Activity	Test	Accuracy
ล้มไปข้างหน้า	5	2
ล้มไปข้างหลัง	5	2
ล้มไปทางซ้าย	5	4
ล้มไปทางขวา	5	3
นอนหงาย	5	5
นอนคว่ำ	5	5
นอนตะแคงซ้าย	5	5
นอนตะแคงขวา	5	5
เดิน	5	5
ลุกขึ้นยืน	5	3

$$\begin{aligned}
 TP &= 11 & FP &= 2 & TN &= 28 & FN &= 9 \\
 \text{Sensitivity} &= 10/(10+10) & &= 0.55 & &= 55\% \\
 \text{Specificity} &= 28/(28+2) & &= 0.93 & &= 93\%
 \end{aligned}$$

ตารางที่ 4.4 แสดงผลการทดสอบของผู้ทดสอบคนที่ 4

Activity	Test	Accuracy
ล้มไปข้างหน้า	5	3
ล้มไปข้างหลัง	5	4
ล้มไปทางซ้าย	5	4
ล้มไปทางขวา	5	3
นอนหงาย	5	5
นอนคว่ำ	5	5
นอนตะแคงซ้าย	5	5
นอนตะแคงขวา	5	5
เดิน	5	5
ลุกขึ้นยืน	5	2

$TP = 14$ $FP = 3$ $TN = 27$ $FN = 6$
 $Sensitivity = 12/(12+8) = 0.7 = 70\%$
 $Specificity = 27/(27+3) = 0.9 = 90\%$

4.3 วิเคราะห์และสรุปผลการทดลอง

จากการทดลองจะเห็นว่า การใช้ค่า Threshold ที่ได้จากการเฉลี่ย ขณะล้ม ให้ความแม่นยำในการตรวจสอบ ในระดับหนึ่ง 1 ใน 4 ผู้ทดสอบที่ได้ค่าดีที่สุดคือ ค่า Sensitivity (สามารถตรวจสอบได้ถูกต้องว่ามีการหกล้มเกิดขึ้นจริง) 70% และได้ค่า Specificity (ความสามารถในการตรวจสอบว่าไม่มีการหกล้มเกิดขึ้น) 90 %

พฤติกรรมการหกล้มทั้งหมดที่ได้ทำการทดสอบ เกิดการตรวจสอบผิดพลาดคือ การลุกขึ้นยืน เนื่องจากในขณะที่ลุกขึ้นยืน อาจมีการเปลี่ยนแปลงของความเร่งไวเช่นเดียวกับการหกล้ม และการล้มหกแต่ระบบไม่สามารถตรวจจับพฤติกรรมได้ว่ามีการหกล้มจึงไม่เกิดการแจ้งเตือน อาจเกิดจากค่าเฉลี่ยของความเร่งไม่เพียงพอกับค่า Threshold ที่เราเลือกไว้ เพราะในการเลือกค่า Threshold จากการทดสอบ ล้มทั้ง 4 ท่า แล้วนำค่าเฉลี่ยขณะล้มมาใช้ เราใช้ผู้ทดสอบเพียงคนเดียว ในการเก็บค่าเฉลี่ยนั้น และเมื่อนำมาทดสอบกับผู้ทดสอบคนอื่นอีก 4 คน จึงอาจเกิดการล้มได้ในระดับที่ไม่เท่ากัน ส่วนหนึ่งของความผิดพลาด ก็อาจเกิดขึ้นได้ด้วยประการนี้



บทที่ 5

สรุป อภิปรายผล และข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการทดลองพบว่าปัจจัยที่ทำให้ระบบตรวจจับการหกล้มแม่นยำ คือ

1. การเลือกค่า Threshold
2. ตำแหน่งการติดตั้งอุปกรณ์ฮาร์ดแวร์

จะเห็นได้ว่า เมื่อผู้ทดสอบติดตั้งอุปกรณ์ที่บริเวณ เอว เลือกค่า Threshold โดยการใช้เฉลี่ยขณะล้ม จากนั้นทำการทดสอบการล้มทั้ง 4 ท่า ได้ค่า Sensitivity 70% และ Specificity 90% ซึ่งเป็นค่าที่ดีที่สุด

5.2 สิ่งที่ได้รับจากการทำโครงการ

1. ได้รับความรู้เกี่ยวกับโปรแกรม Simulink MATLAB
2. ได้รับความรู้เกี่ยวกับการใช้งาน FiO Std Board, X-bee และเซ็นเซอร์วัดความเร่ง 3 แกน
3. ได้รู้จักการทำงานร่วมกับผู้อื่น
4. สามารถนำความรู้ที่ได้จากการศึกษาทฤษฎีมาปฏิบัติและประยุกต์ใช้ได้จริง

5.3 ปัญหาและอุปสรรค

1. การลงโปรแกรมสำหรับ FiO Std Board ต้องลงให้ครบทุกซอฟต์แวร์ และตรวจสอบเวอร์ชันที่ใช้งานให้ถูกต้องจึงจะสามารถใช้งานติดต่อกับคอมพิวเตอร์ได้
2. มีความเพี้ยนของสัญญาณความเร่ง ที่เกิดการหกล้มแล้วระบบไม่สามารถตรวจจับได้ว่าเกิดการหกล้มจึงไม่เกิดการแจ้งเตือน และไม่หกล้มแต่ระบบตรวจจับว่ามีการหกล้มจึงแจ้งเตือน
3. ฮาร์ดแวร์ที่เราสร้างด้วยตัวเอง มีข้อจำกัดหลายอย่าง

5.4 ข้อเสนอแนะ

1. ใช้เซ็นเซอร์วัดความเร่งแบบ 3 แกน ควบคู่กับเซ็นเซอร์อื่นๆ เพื่อการตรวจจับการหกล้มที่แม่นยำมากขึ้น เช่น Body Heat , Heart Rat
2. ควรติดตั้งอุปกรณ์ให้แนบชิดกับตัว ไม่ให้มีการแกว่งไปมา
3. การเลือกค่าThreshold มีผลต่อการตรวจจับพฤติกรรม
4. โปรแกรมที่ใช้สามารถนำไปพัฒนาให้ดียิ่งขึ้นได้
5. สามารถพัฒนาอุปกรณ์ให้ตรวจจับพฤติกรรมต่างๆ ได้ดีกว่านี้ เช่น การหกล้มเบาๆ หรือ สะดุดล้ม และมีการลุกขึ้นมาได้ ไม่ให้แจ้งเตือนว่าเกิดพฤติกรรมหกล้มที่รุนแรง
6. Package สามารถทำให้เล็กกว่านี้ และพกพาได้ง่ายขึ้น
7. ในการทดสอบนั้น แม้ว่าเราจะทดสอบบนที่นอน ก็ควรจะทดสอบด้วยความระมัดระวัง เพราะอาจเกิดอันตรายที่แท้จริงได้



เอกสารอ้างอิง

- [1] Stefano Abbate, Marco Avvenuti, Paolo Corsini, Janet Light and Alessio Vecchio (2010). Monitoring of Human Movements for Fall Detection and Activities Recognition in Elderly Care Using Wireless Sensor Network: a Survey, Wireless Sensor Networks: Application-Centric Design, Yen Kheng Tan (Ed.), ISBN: 978-953-307-321-7, InTech, Available from: <http://www.intechopen.com/books/wireless-sensor-networks-application-centricdesign/monitoring-of-human-movements-for-fall-detection-and-activities-recognition-in-elderly-care-using-wi>
- [2] ชีรศักดิ์ โชติกวนิชย์. 2555. ระบบสมองกลฝังตัวสำหรับตรวจจับการล้มขณะอาบน้ำด้วยกล้องที่รักษาความเป็นส่วนตัว . วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า มหาวิทยาลัยสงขลานครินทร์.
- [3] นางสาวสุภาวี มาท่ามา, นางสาวทิพย์พากร หงษ์คำ, นางสาวกมลชนก จิระสุโข. 2556. ระบบตรวจจับพฤติกรรมกรหกล้มโดยใช้เซ็นเซอร์ความเร่งแบบ 3 แกน. โครงการงานวิศวกรรมโทรคมนาคม สาขาวิชาโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี.
- [4] P. Jantaraprim, P. Phukpharanont, C. Limsakul and B. Wongkittisuksa “Improving the Accuracy of a Fall Detection Algorithm Using Free Fall Characteristics,” in: NECTEC Technical Journal, NECTEC-ACE2009 Special Edition, September, 2009, pp. 239-247.
- [5] “Threshold”, [ออนไลน์], เข้าถึงได้จาก: [http://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](http://en.wikipedia.org/wiki/Thresholding_(image_processing)).

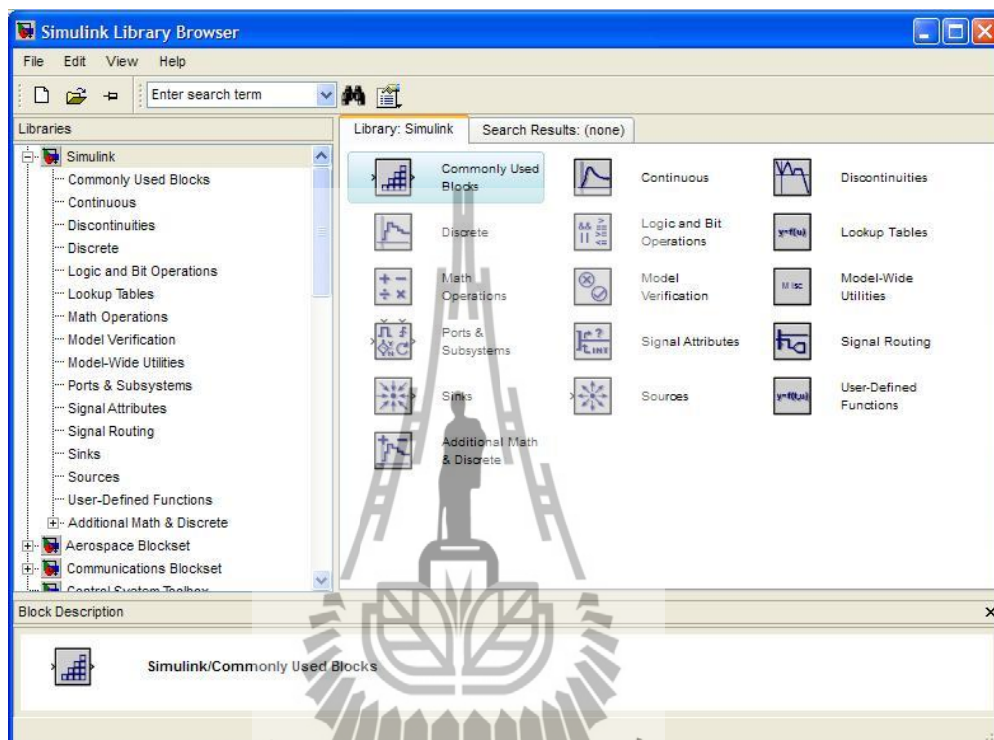


ภาคผนวก

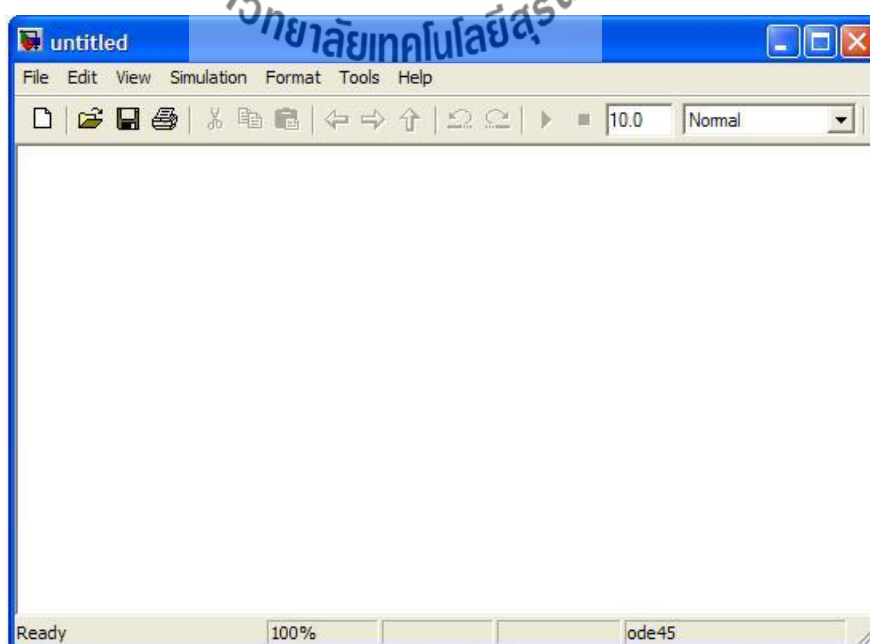
วิธีการใช้งานหน้าต่าง Simulink การสร้างบล็อก Simulink และการตั้งค่าบล็อก

วิธีการใช้งานหน้าต่าง Simulink

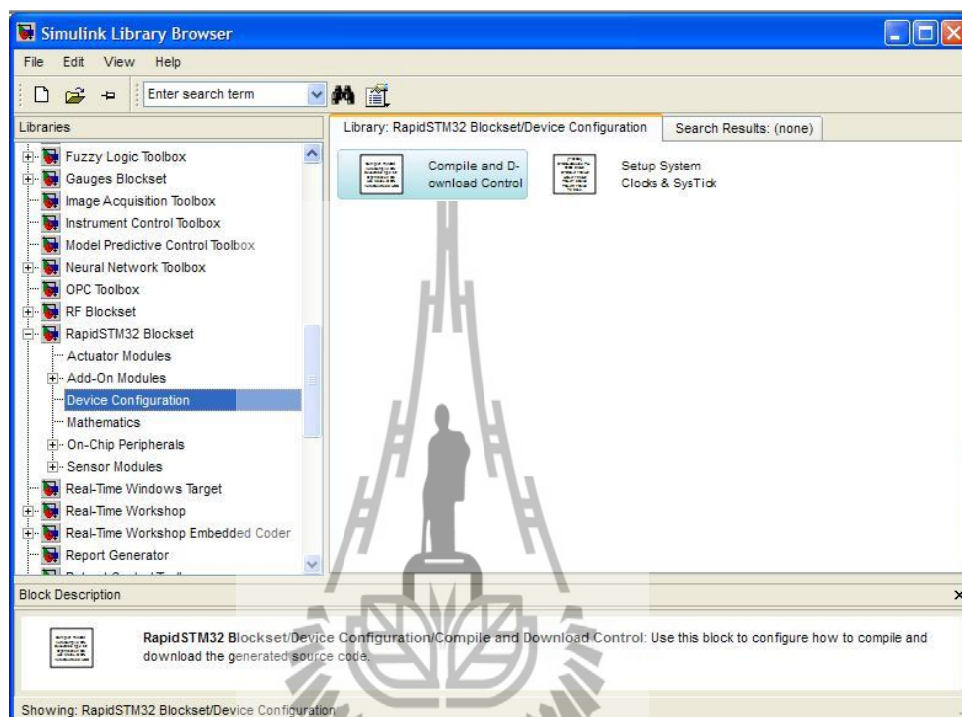
เปิดโปรแกรม MATLAB คลิกปุ่ม จะแสดงหน้าต่างดังรูป จากนั้นเลือก New Model จะได้หน้าต่างใหม่ขึ้นมาเพื่อสร้างบล็อก Simulink



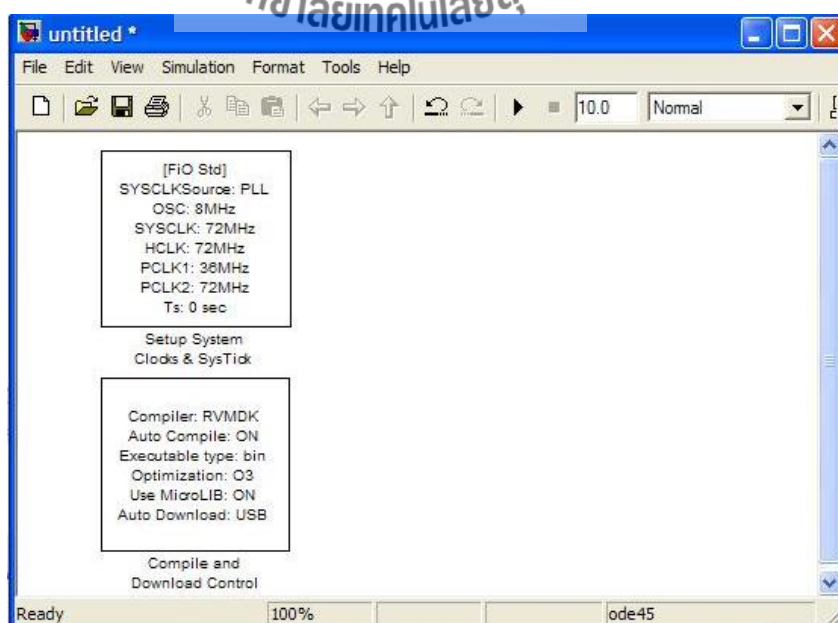
หน้าต่าง New Model



สร้างบล็อก Simulink เพื่อตั้งค่าการใช้งาน FiO Board การคอมไพล์และการควบคุมโหลด โดยคลิกที่ RapidSTM32 Blockset>> Device Configuration แล้วเลือกบล็อก Compile and Download Control แล้วบล็อก Setup System Clock & SysTickลงในหน้าต่างของ Model ที่ต้องการสร้างดังรูป



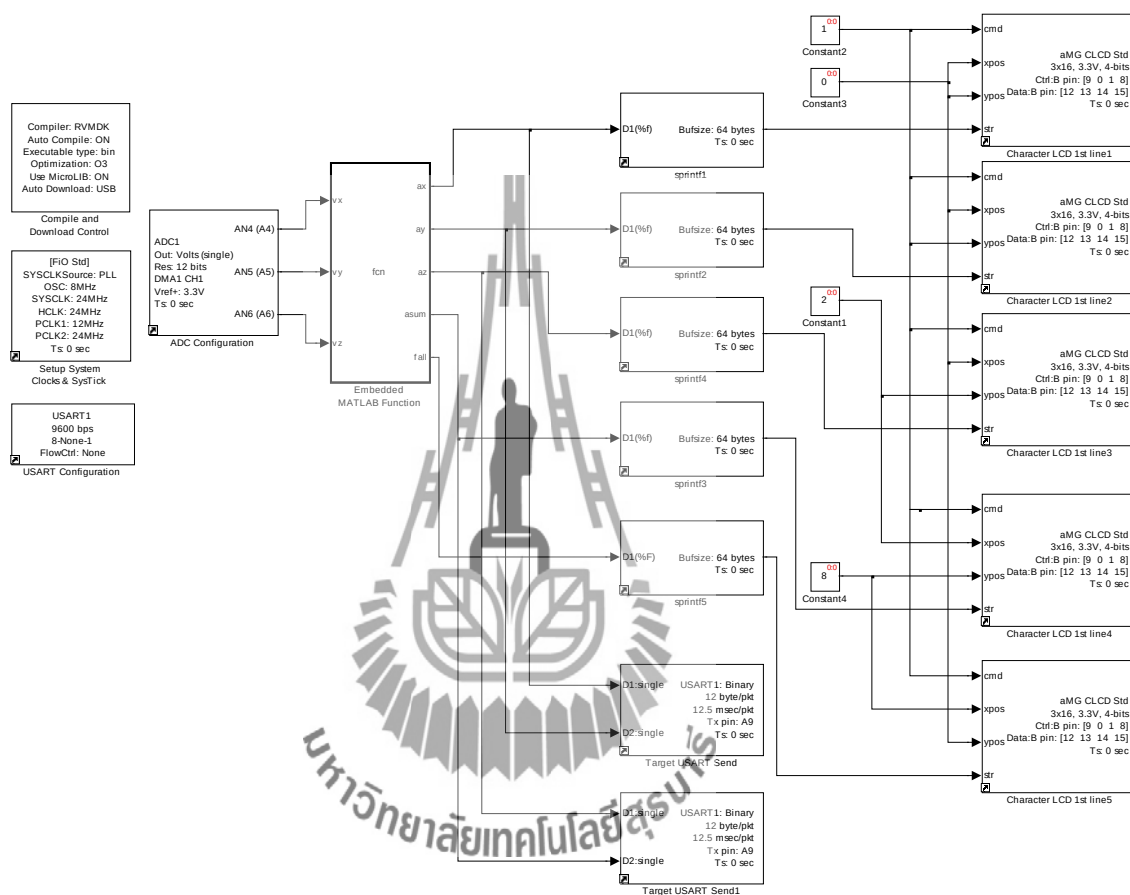
การสร้างบล็อก



การสร้างบล็อก Simulink และการตั้งค่าบล็อก

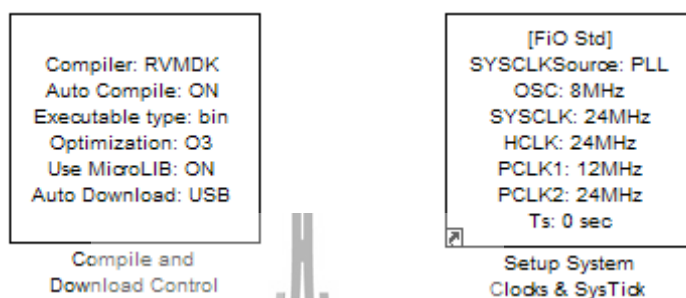
บล็อก Simulink ภา่งสำหรับการทดสอบ

การสร้างบล็อก Simulink และการตั้งค่าบล็อก สำหรับภา่งที่ทำการสร้างเพื่อทดสอบว่า ภา่งรับและภา่งส่งสามารถรับข้อมูลได้ตรงกันหรือไม่ สามารถสร้างบล็อก Simulink ได้ดังนี้

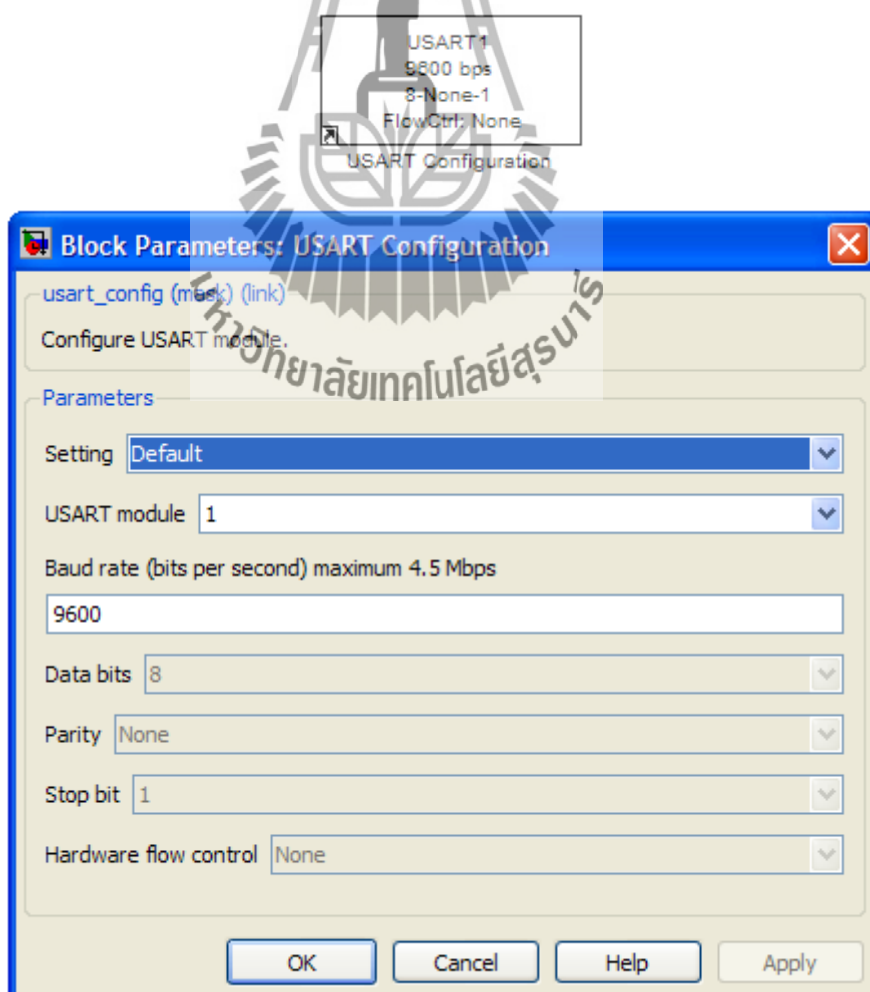


โดยอธิบายการการตั้งค่าบล็อกที่สำคัญดังนี้

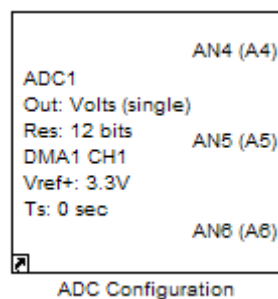
อันดับแรกในการสร้างบล็อกเลือก Compile and Download Control ก่อน แล้วจึงสร้างบล็อก Setup System Clock & SysTick เพราะสองบล็อกแรกนี้ ลำดับในการวางมีผลในการสร้างบล็อก Simulink ถัดไป ใช้สำหรับDownloadโปรแกรมลงบอร์ด



USART Configuration เป็นบล็อกสำหรับ X-bee



ADC Configuration เป็นบล็อกสำหรับเซ็นเซอร์ที่เป็น Analog



Source Block Parameters: ADC Configuration

adc1 (mask) (link)

Implement ADC1 using: Regular Group Channel Selection, Scan mode, and DMA1 data transfer.

Channel 16 is on-chip temperature sensor (units in volts). Use the supplied 'Convert Onboard AN16 Volts to Temp C' to convert output from volts to Celsius.

Click help below for more information.

Parameters

Setting: Custom

Channel Selection (AN0 - AN17), maximum 16 channels at a time: [4 5 6]

Resolution (bits): 12 bits (Standard)

Output Data Type: Volts (single - 6 bits)

ADC (DMA1) Priority Level: High

ADCCLK = 4 MHz for Prescaler: PCLK2/6

Sample time (in seconds): 1

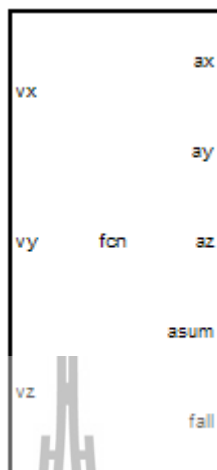
AN4 Sample time: 7.5 cycles

AN5 Sample time: 7.5 cycles

Positive Reference Voltage (Vref+), Volts: 3.3

OK Cancel Help

Embedded MATLAB Function เป็นบล็อกสำหรับเขียนสมการเซ็นเซอร์



Embedded
MATLAB Function

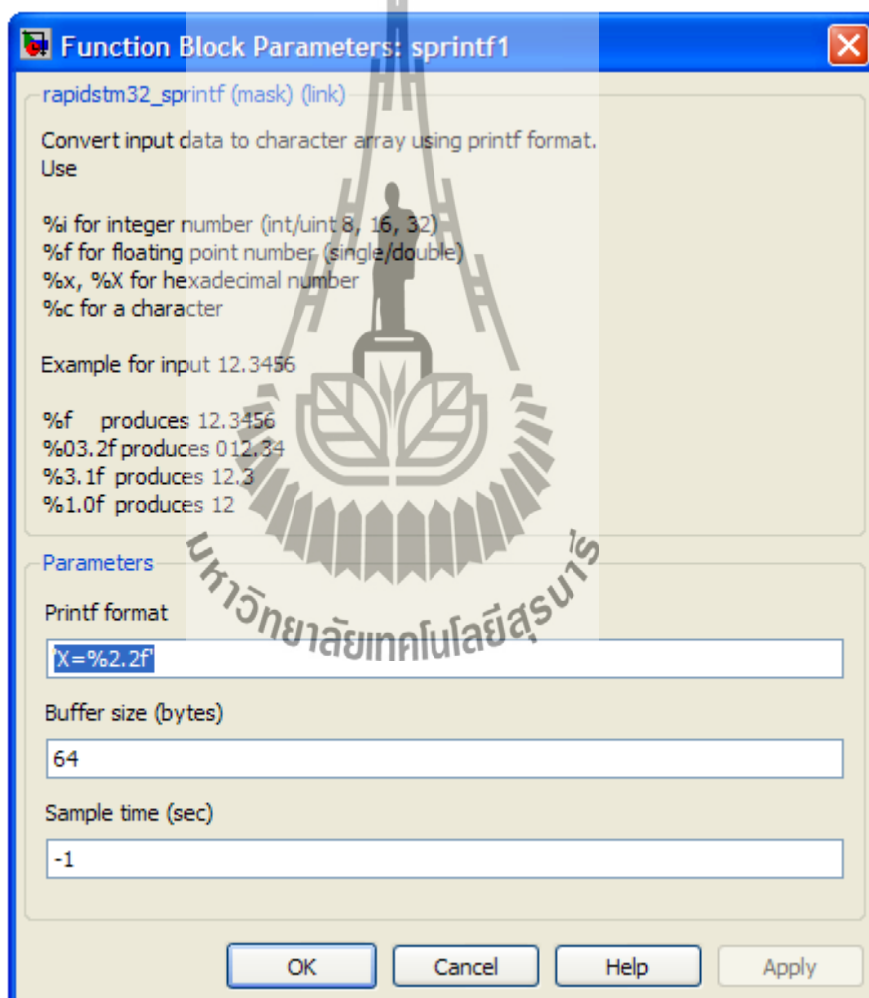
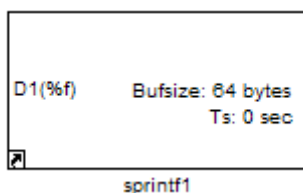
```

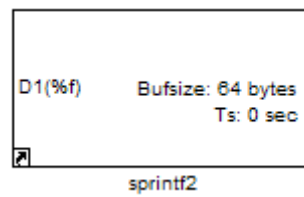
1  function [ax, ay, az, asum, fall] = fcn(vx, vy, vz)
2  %#eml
3
4  c1 = 1.25;
5  c2 = -2.0625;
6
7  ax = c1*vx + c2 ;
8  ay = c1*vy + c2 ;
9  az = c1*vz + c2 + 0.2 ;
10
11  asum = (ax^2 + ay^2 + az^2)^0.5;
12
13  fall = 0;
14  if asum>2.3
15      fall = 1;
16  end
17
18

```

Ready Ln 18 Col 1

Sprintf เป็นบล็อกสำหรับแสดงผล โดยในที่นี้จะแสดงผลที่ได้จากบล็อก Embedded MATLAB Function ตามลำดับดังนี้ Sprintf1 แสดงค่า X, Sprintf1 แสดงค่า X, Sprintf2 แสดงค่า Y, Sprintf4 แสดงค่า Z, Sprintf3 แสดงค่า a และ Sprintf5 แสดงค่า FALL





Function Block Parameters: sprintf2

rapidstm32_sprintf (mask) (link)

Convert input data to character array using printf format.
Use

- %i for integer number (int/uint 8, 16, 32)
- %f for floating point number (single/double)
- %x, %X for hexadecimal number
- %c for a character

Example for input 12.3456

- %f produces 12.3456
- %03.2f produces 012.34
- %3.1f produces 12.3
- %1.0f produces 12

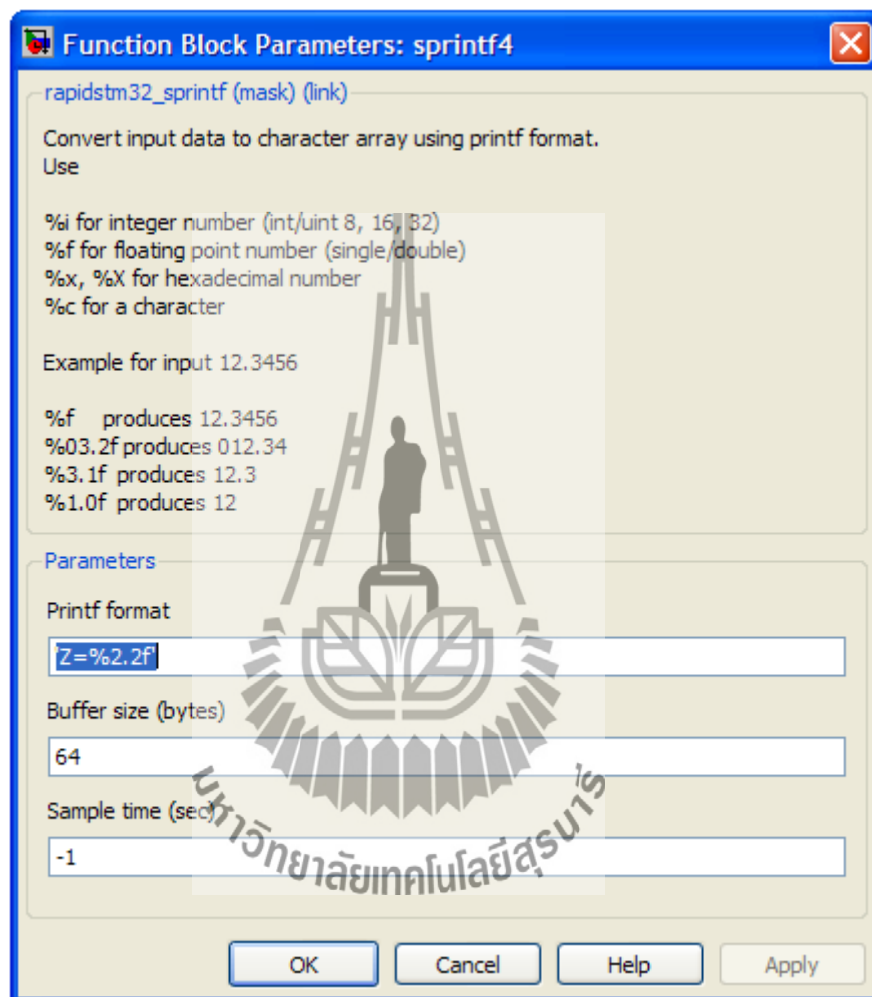
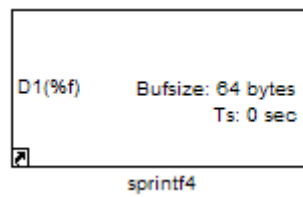
Parameters

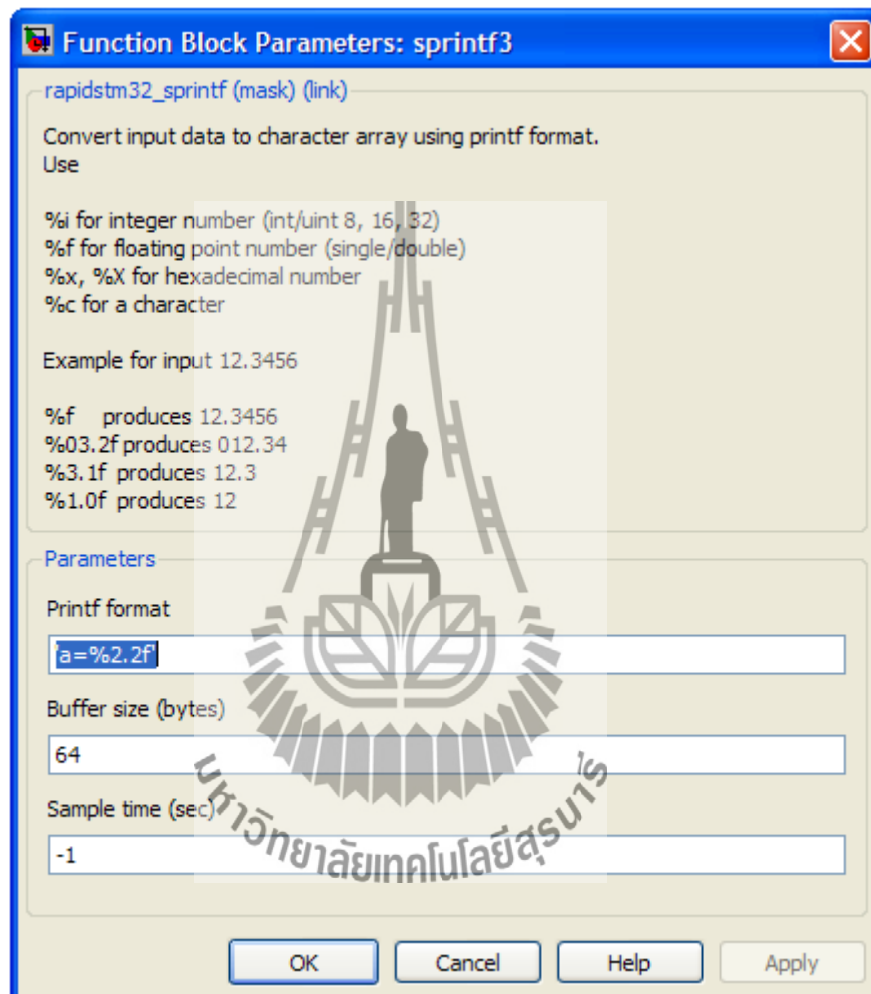
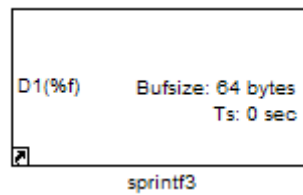
Printf format

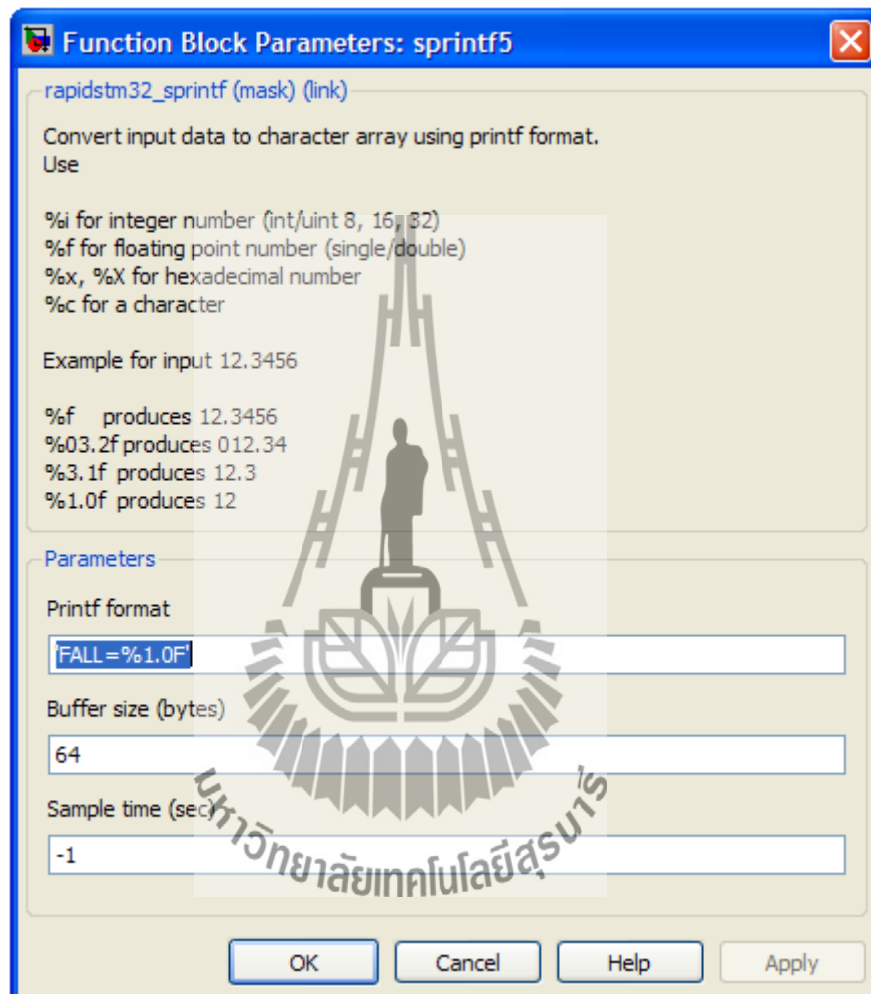
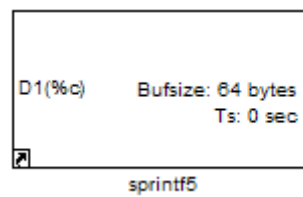
Buffer size (bytes)

Sample time (sec)

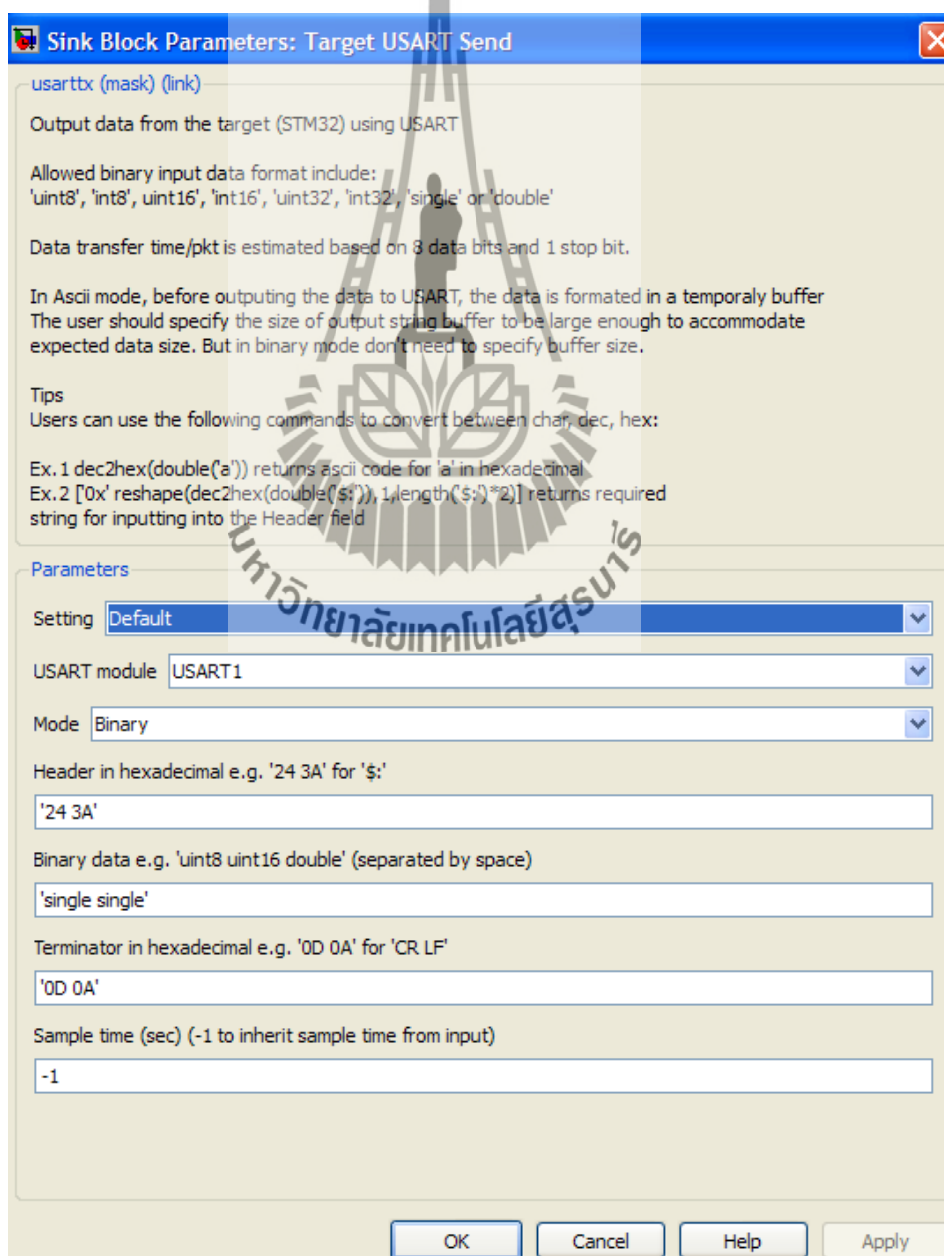
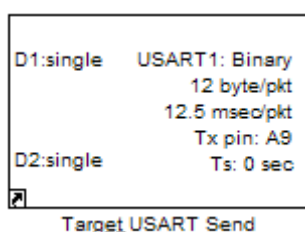
OK Cancel Help Apply

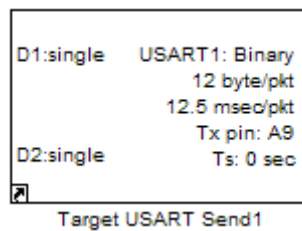






Target USART Send เป็นบล็อกสำหรับการส่งข้อมูลผ่าน X-bee โดยหนึ่งบล็อกส่งข้อมูลได้ 2 ข้อมูล ในที่นี้เราใช้ Target USART Send 2 บล็อกเพื่อส่งข้อมูลคือ Target USART Send ส่งข้อมูล X และ Y และ Target USART Send1 ส่งข้อมูล Z และ a





Sink Block Parameters: Target USART Send1

usarttx (mask) (link)

Output data from the target (STM32) using USART

Allowed binary input data format include:
'uint8', 'int8', 'uint16', 'int16', 'uint32', 'int32', 'single' or 'double'

Data transfer time/pkt is estimated based on 8 data bits and 1 stop bit.

In Ascii mode, before outputting the data to USART, the data is formatted in a temporary buffer
The user should specify the size of output string buffer to be large enough to accommodate expected data size. But in binary mode don't need to specify buffer size.

Tips
Users can use the following commands to convert between char, dec, hex:

Ex.1 `dec2hex(double('a'))` returns ascii code for 'a' in hexadecimal
Ex.2 `[0x' reshape(dec2hex(double($:')),1,length($:'))*2]` returns required string for inputting into the Header field

Parameters

Setting **Default**

USART module **USART1**

Mode **Binary**

Header in hexadecimal e.g. '24 3A' for '\$':
'24 3A'

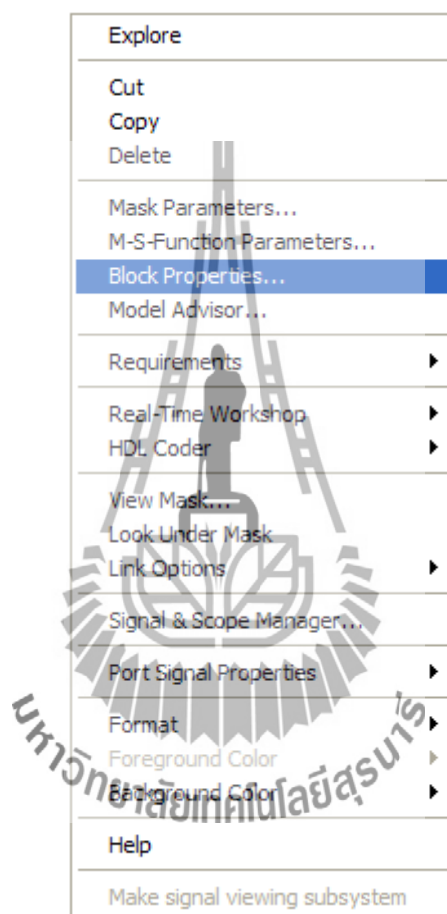
Binary data e.g. 'uint8 uint16 double' (separated by space)
'single single'

Terminator in hexadecimal e.g. '0D 0A' for 'CR LF'
'0D 0A'

Sample time (sec) (-1 to inherit sample time from input)
-1

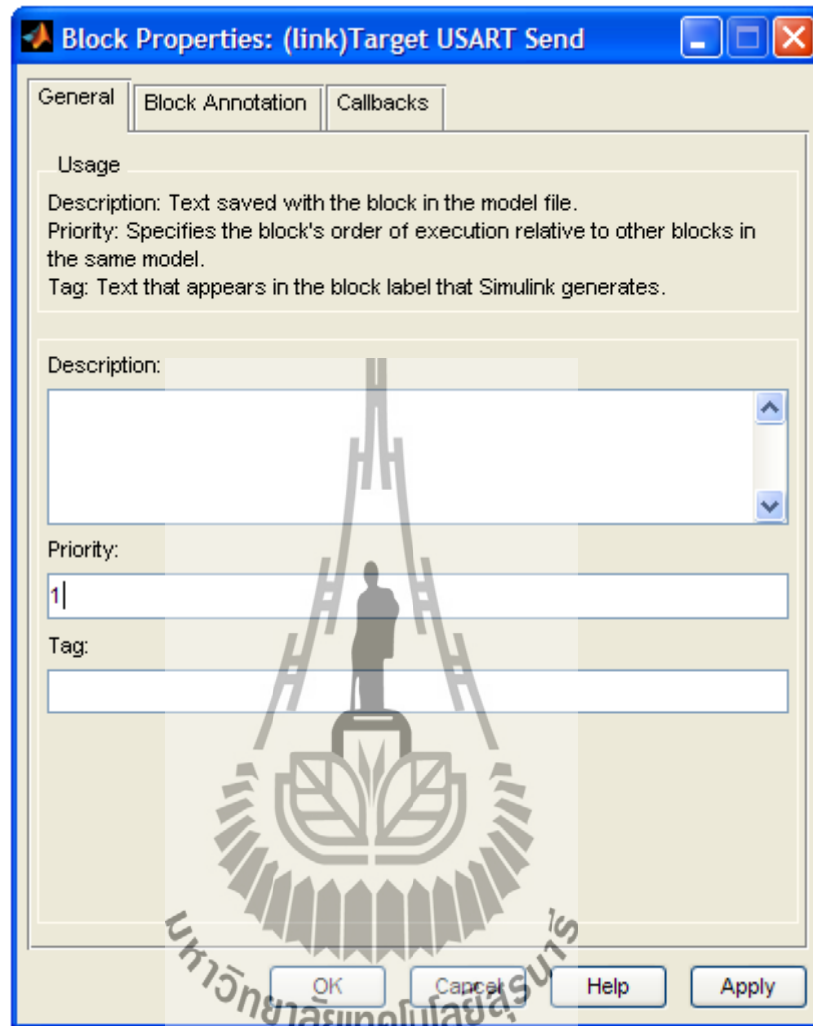
OK Cancel Help Apply

สำหรับบล็อก Target USART Send เราพบว่าต้องมีการกำหนด Priority ให้กับบล็อก เพื่อให้ส่งข้อมูลได้ถูกต้อง เพราะเป็นการจัดลำดับความสำคัญในการส่งข้อมูล การกำหนดค่า Priority ทำได้โดยคลิกขวาที่บล็อก Target USART Send แล้วเลือก Block Properties

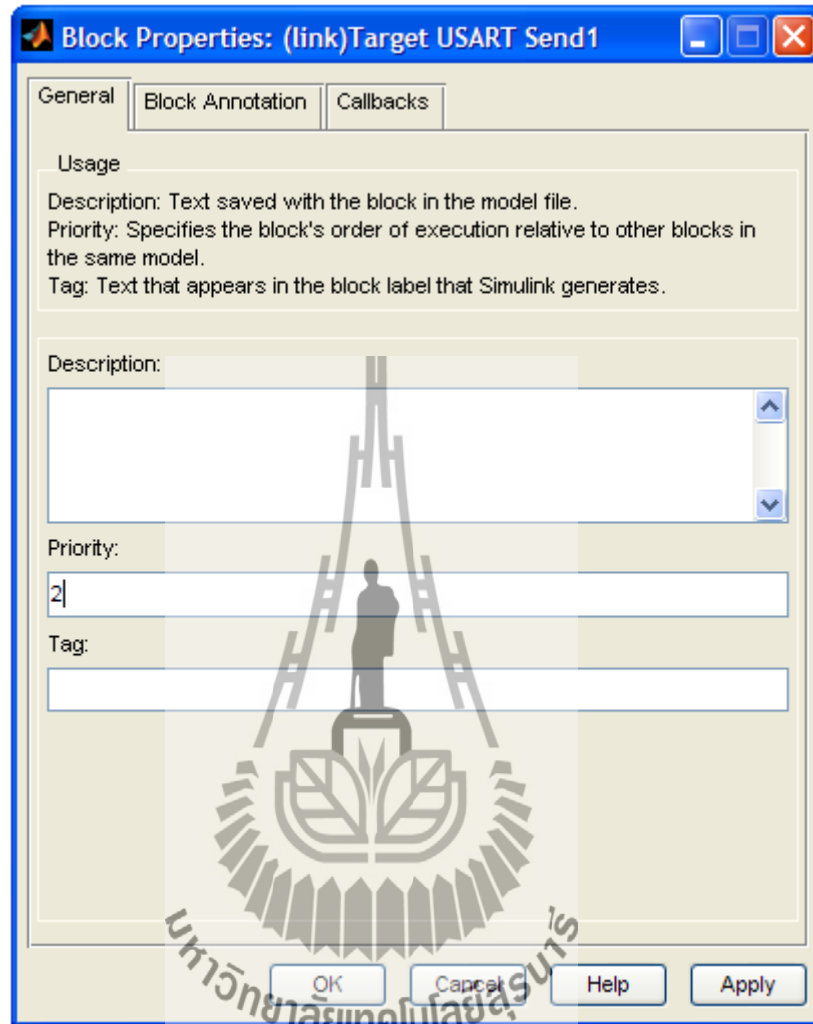


จากนั้นทำการตั้งค่าตามลำดับ โดยให้ Target USART Send ที่ส่งค่า X และ Y เป็น Priority 1 และ Target USART Send1 ที่ส่งค่า Z และ a เป็น Priority 2

Target USART Send : Priority 1

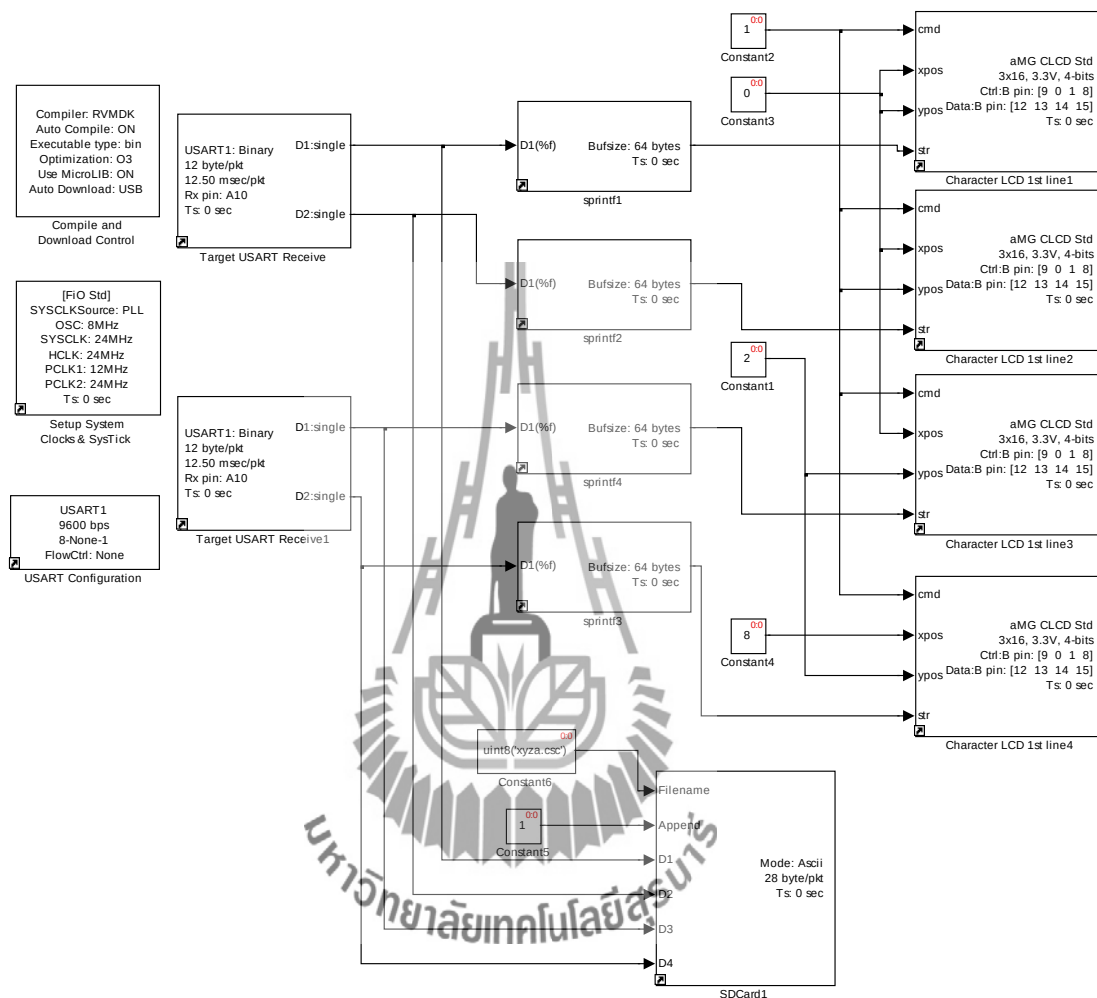


Target USART Send1 : Priority 2



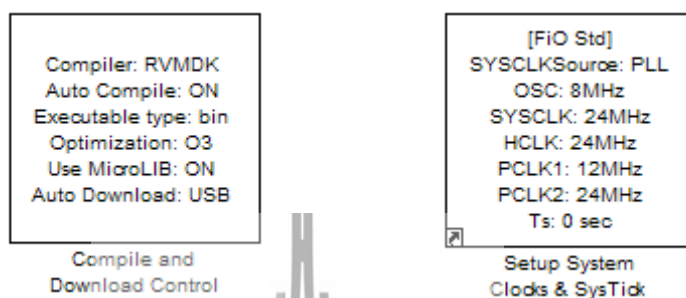
บล็อก Simulink ภาครับสำหรับการทดสอบ

การสร้างบล็อก Simulink และการตั้งค่าบล็อก สำหรับภาครับ ที่ทำการสร้างเพื่อทดสอบว่า
ภาครับและภาคส่งสามารถรับข้อมูลได้ตรงกันหรือไม่ สามารถเขียนสร้างบล็อก Simulink ได้ดังนี้

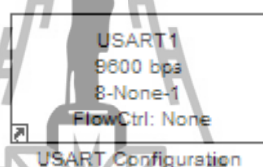


โดยอธิบายการตั้งค่าบล็อกที่สำคัญดังนี้

อันดับแรกในการสร้างบล็อก ให้เลือกสร้าง Compile and Download Control ก่อน แล้วจึงสร้างบล็อก Setup System Clock & SysTick เพราะสองบล็อกแรกนี้ ลำดับในการวางมีผลในการสร้างบล็อก Simulink ถัดไป ใช้สำหรับDownloadโปรแกรมลงบอร์ด



USART Configuration เป็นบล็อกสำหรับ X-bee



Block Parameters: USART Configuration

usart_config (mask) (link)
Configure USART module.

Parameters

Setting: Default

USART module: 1

Baud rate (bits per second) maximum 4.5 Mbps
9600

Data bits: 8

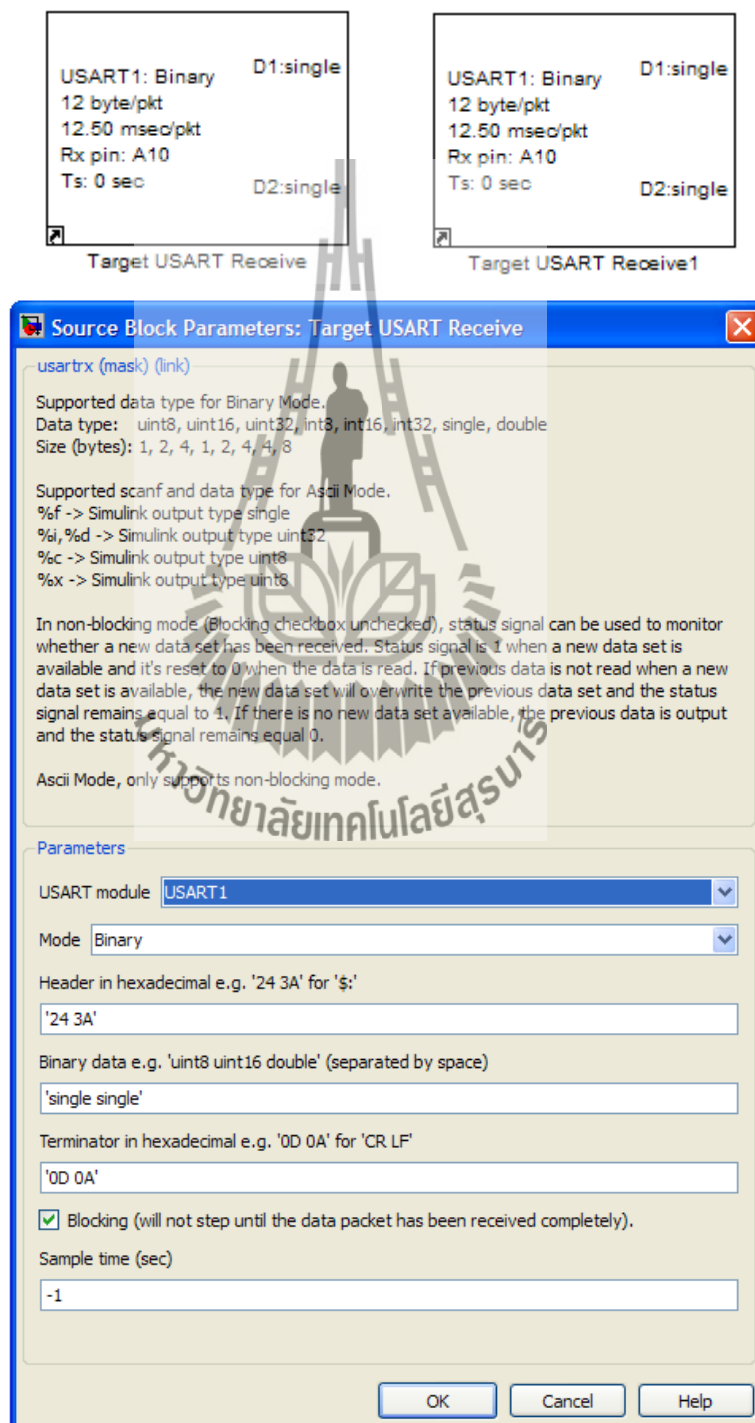
Parity: None

Stop bit: 1

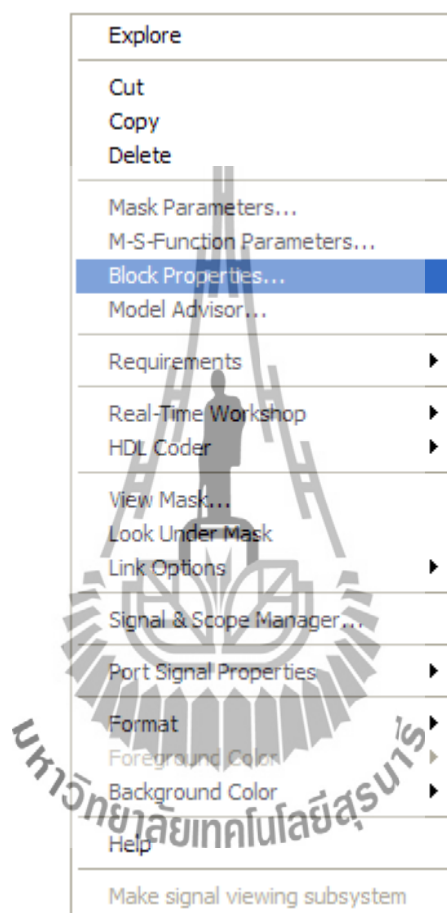
Hardware flow control: None

OK Cancel Help Apply

Target USART Receive เป็นบล็อกสำหรับการรับข้อมูลจาก X-bee ภาครับ โดยหนึ่งบล็อก
 รับได้ 2 ข้อมูล ในที่นี้เราใช้ Target USART Receive 2บล็อกเพื่อรับข้อมูลคือ Target USART
 Receive รับข้อมูล X และ Y และ Target USART Receive1 รับข้อมูล Z และ a

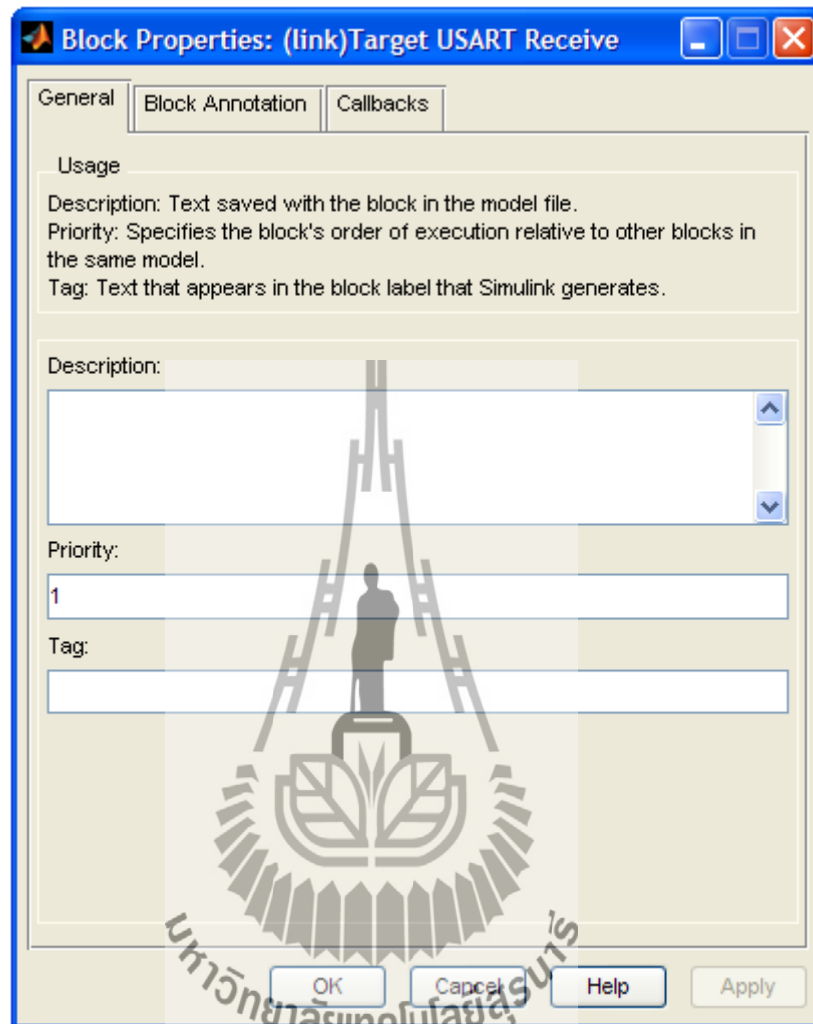


สำหรับบล็อก Target USART Receive เราพบว่าการกำหนด Priority ให้กับบล็อก เพื่อให้ส่งข้อมูลได้ถูกต้อง เพราะเป็นการจัดลำดับความสำคัญในการรับข้อมูล การกำหนดค่า Priority ทำได้โดยคลิกขวาที่บล็อก Target USART Receive แล้วเลือก Block Priority

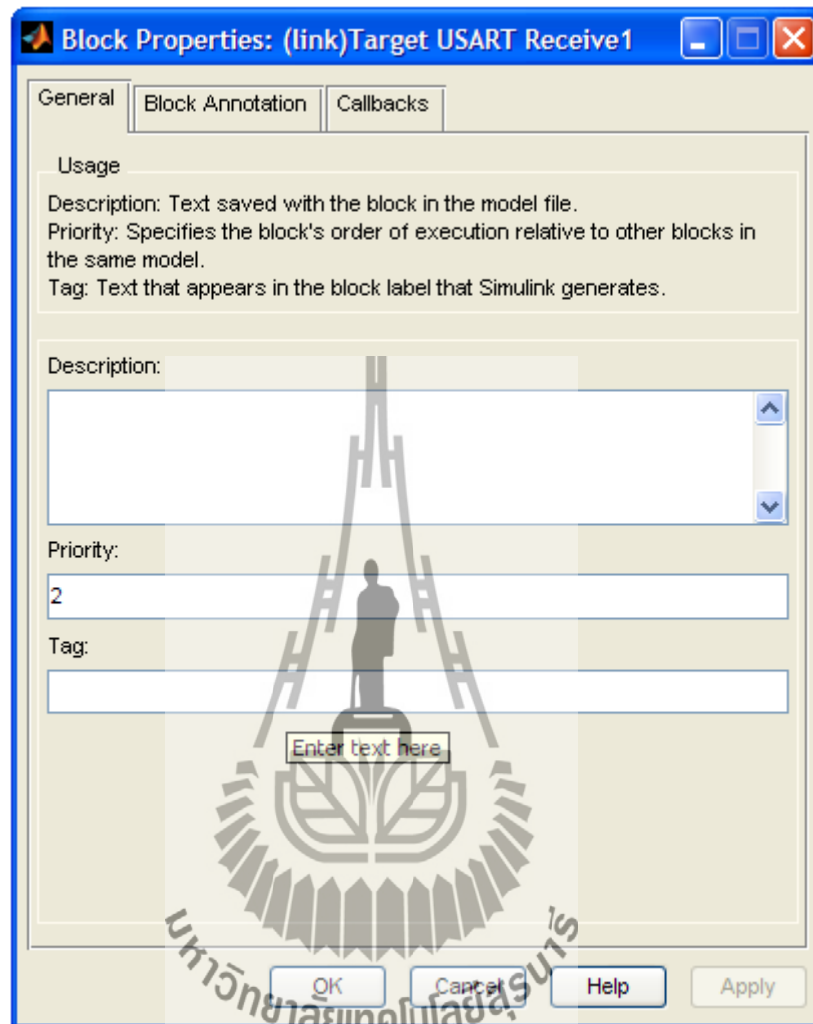


จากนั้นทำการตั้งค่าตามลำดับ โดยให้ Target USART Receive ที่รับค่า X และ Y เป็น Priority 1 และ Target USART Receive1 ที่รับค่า Z และ a เป็น Priority 2

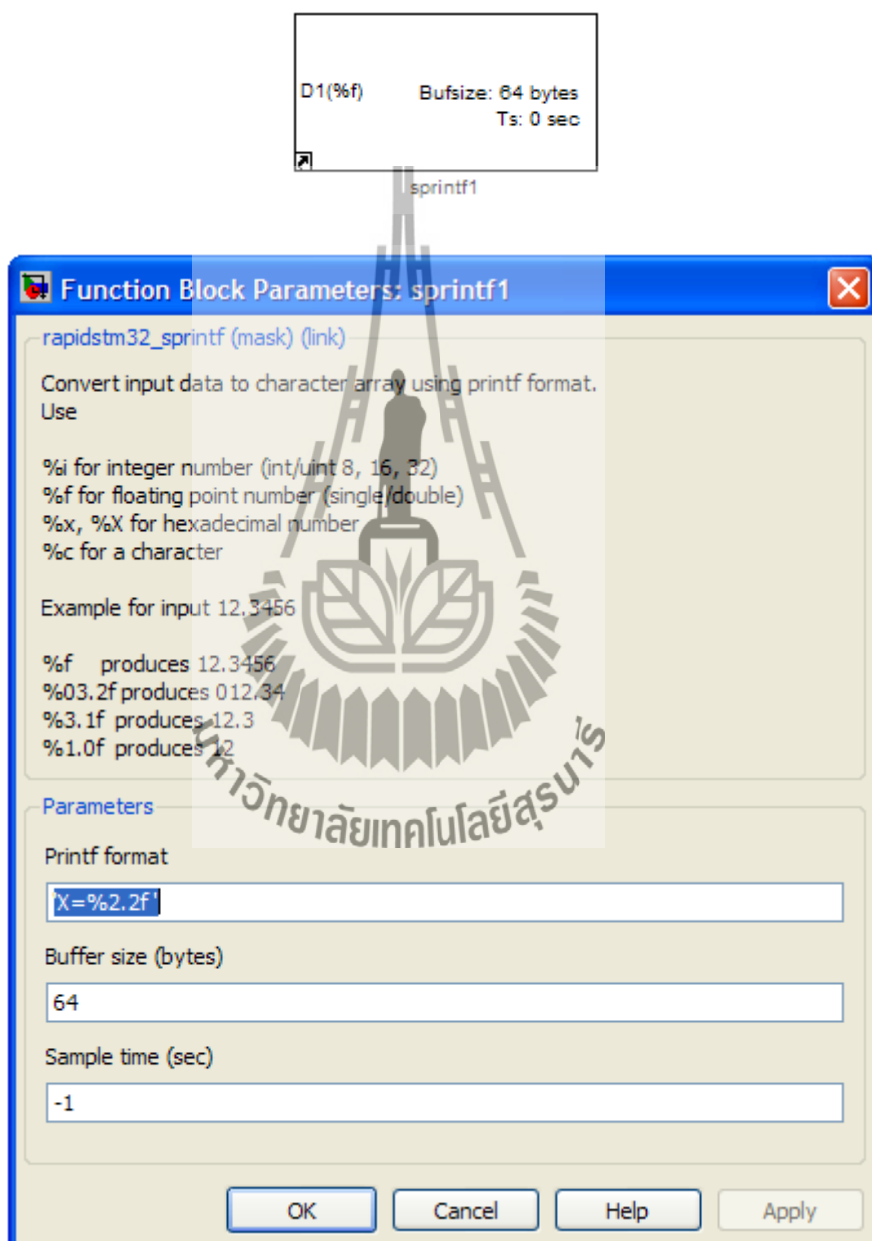
Target USART Receive : Priority 1

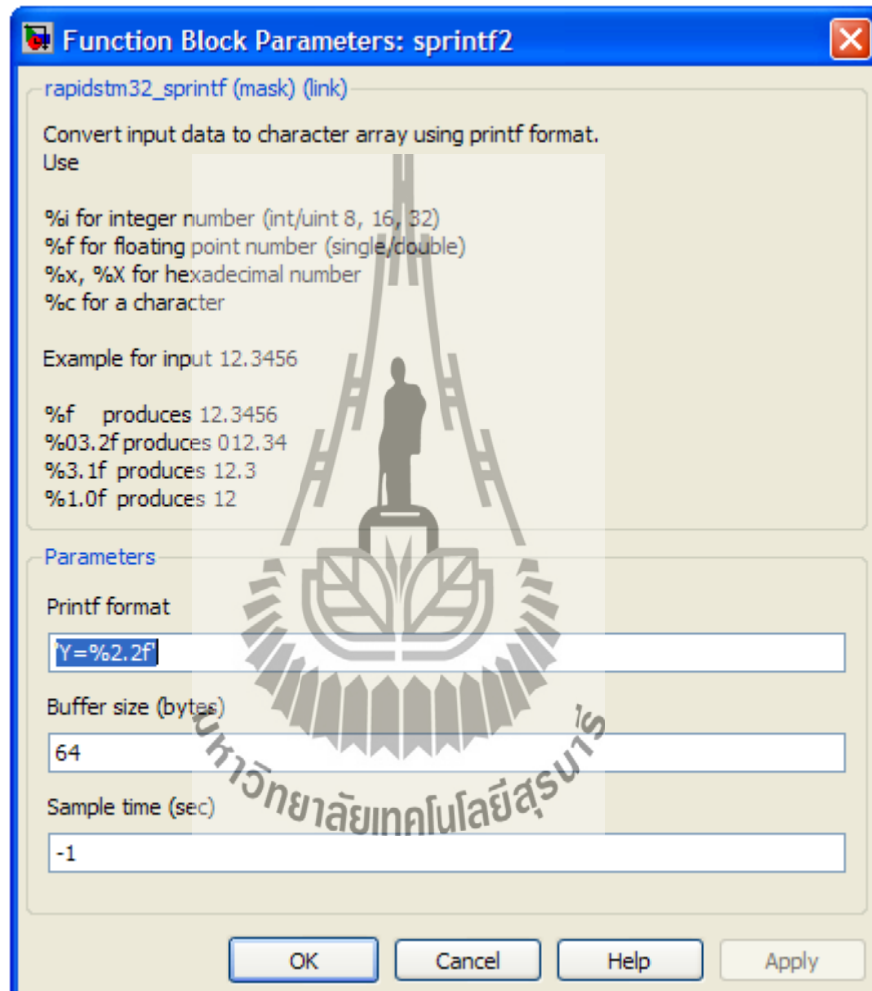
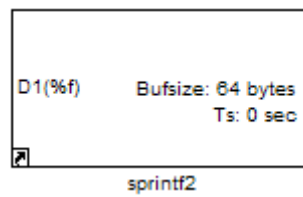


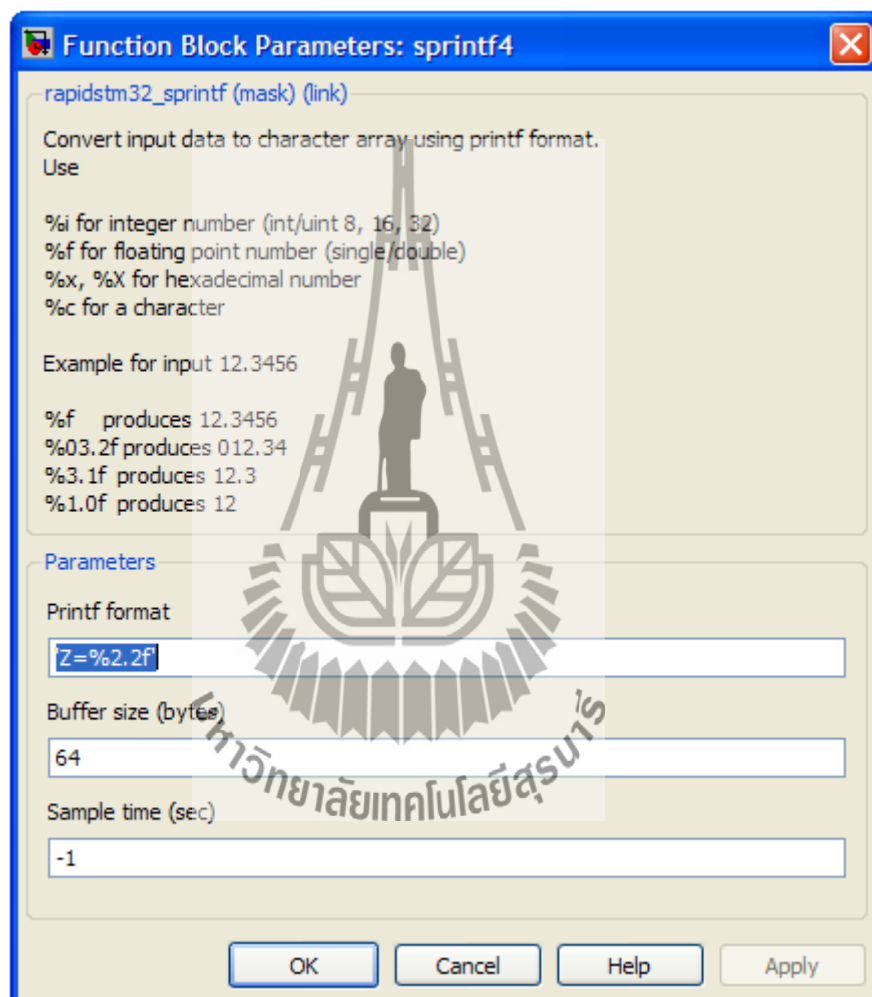
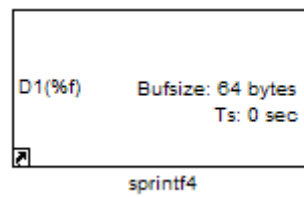
Target USART Receive1 : Priority 2

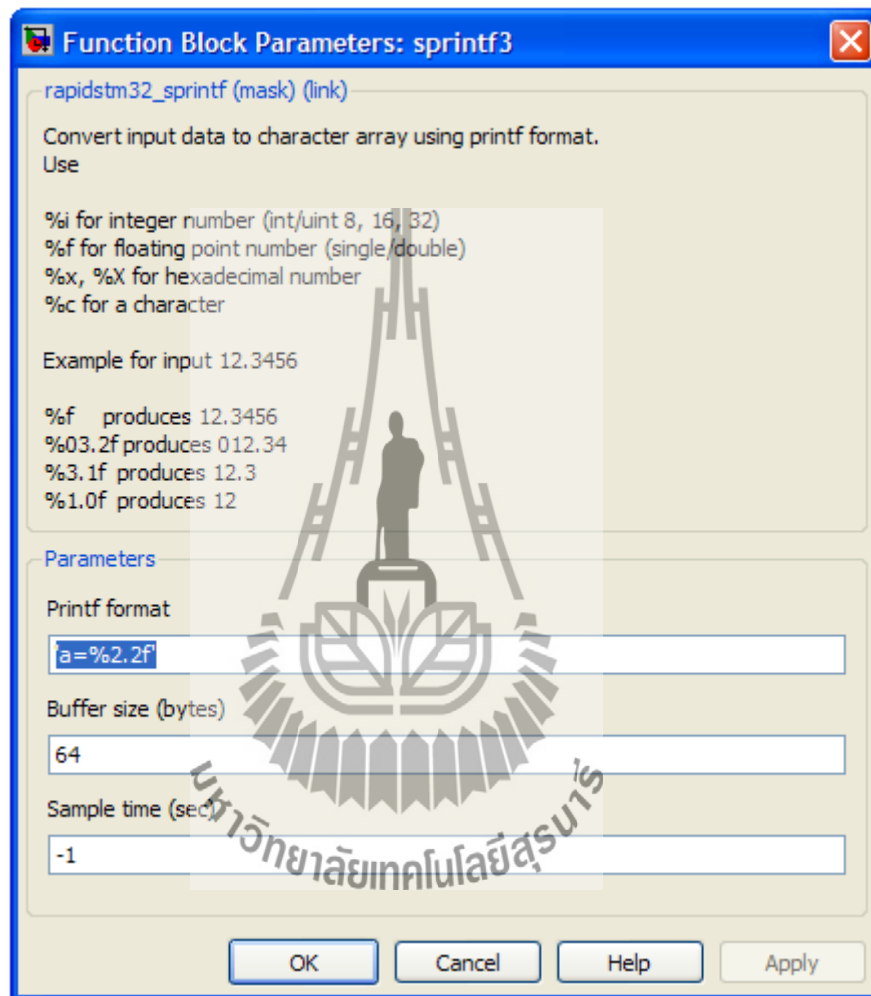
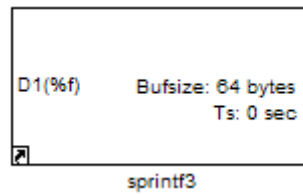


Sprintf เป็นบล็อกสำหรับแสดงผล โดยในที่นี้จะแสดงผลที่ได้รับจาก X-bee ภาดส่งตามลำดับดังนี้ Sprintf1 แสดงค่า X, Sprintf1 แสดงค่า X, Sprintf2 แสดงค่า Y, Sprintf4 แสดงค่า Z และ Sprintf3 แสดงค่า a

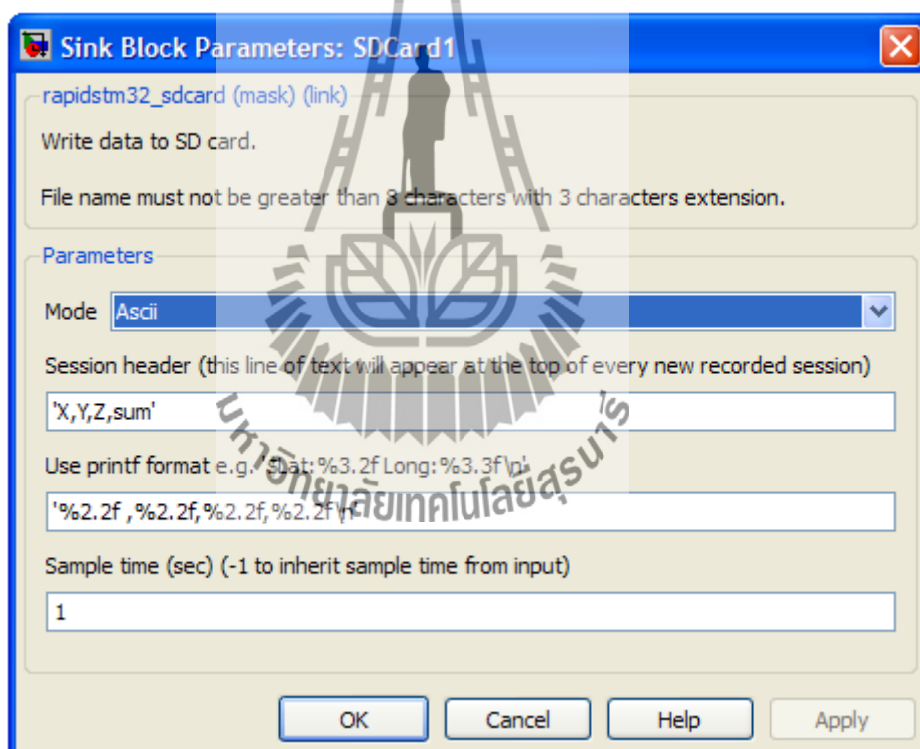
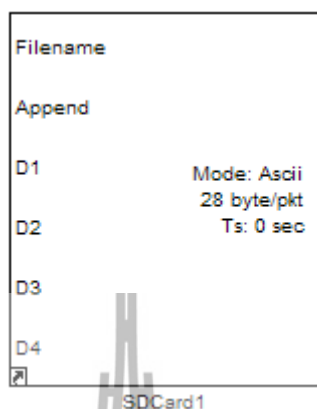




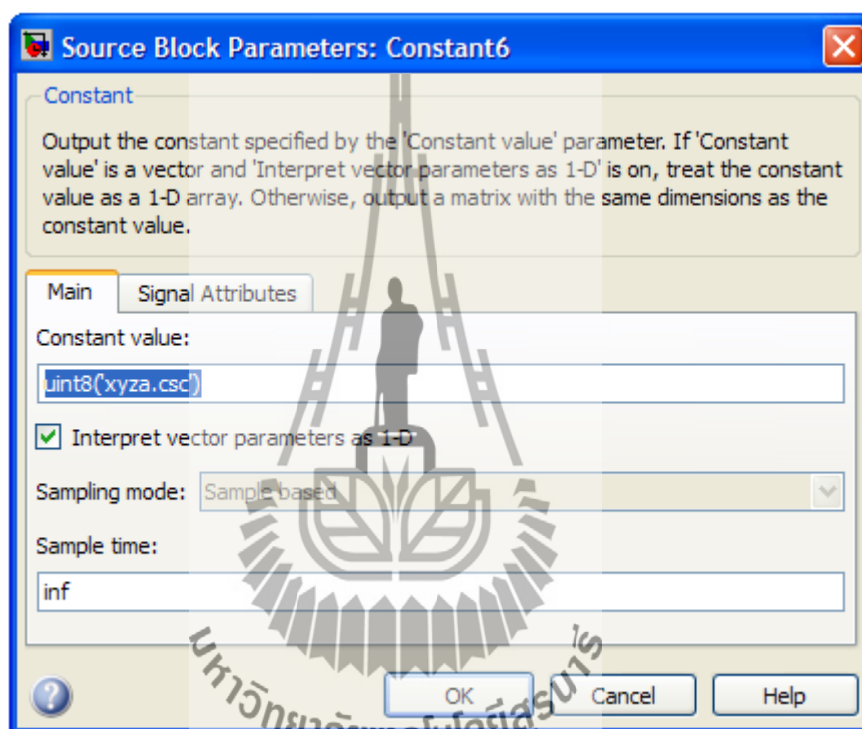




SDcard1 เป็นบล็อกสำหรับเก็บค่าที่รับได้หรือค่าที่ส่งไว้ในหน่วยความจำ (SD card)



บล็อก Constant นี้เราใช้สำหรับตั้งชื่อไฟล์ที่เรารับค่าเก็บไว้ใน SD card โดยตั้งชื่อไฟล์ว่า xyza และ กำหนดให้เป็นไฟล์ Excel

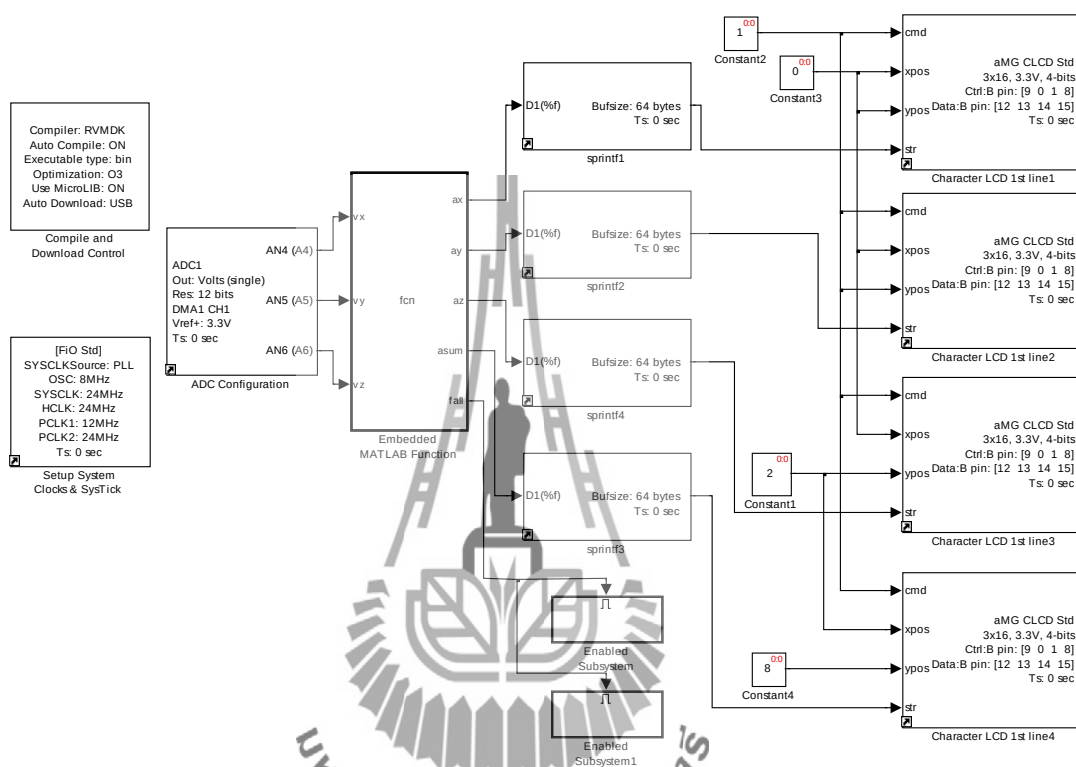


บล็อก Simulink ภาคส่งสำหรับการใช้งานจริงกับฮาร์ดแวร์

เมื่อเราทำสารทดสอบแล้วว่า ที่ภาครับและภาคส่งสามารถรับส่งข้อมูลได้ตรงกัน จึงสร้าง

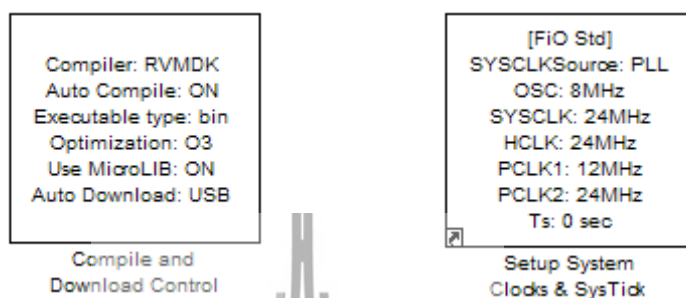
บล็อก Simulink สำหรับการส่งข้อมูลว่ามีการหกล้ม เพื่อให้ภาครับรับข้อมูลและมีการแจ้งเตือน

เกิดขึ้นที่โปรแกรมภาครับ สามารถสร้างบล็อกได้ดังนี้

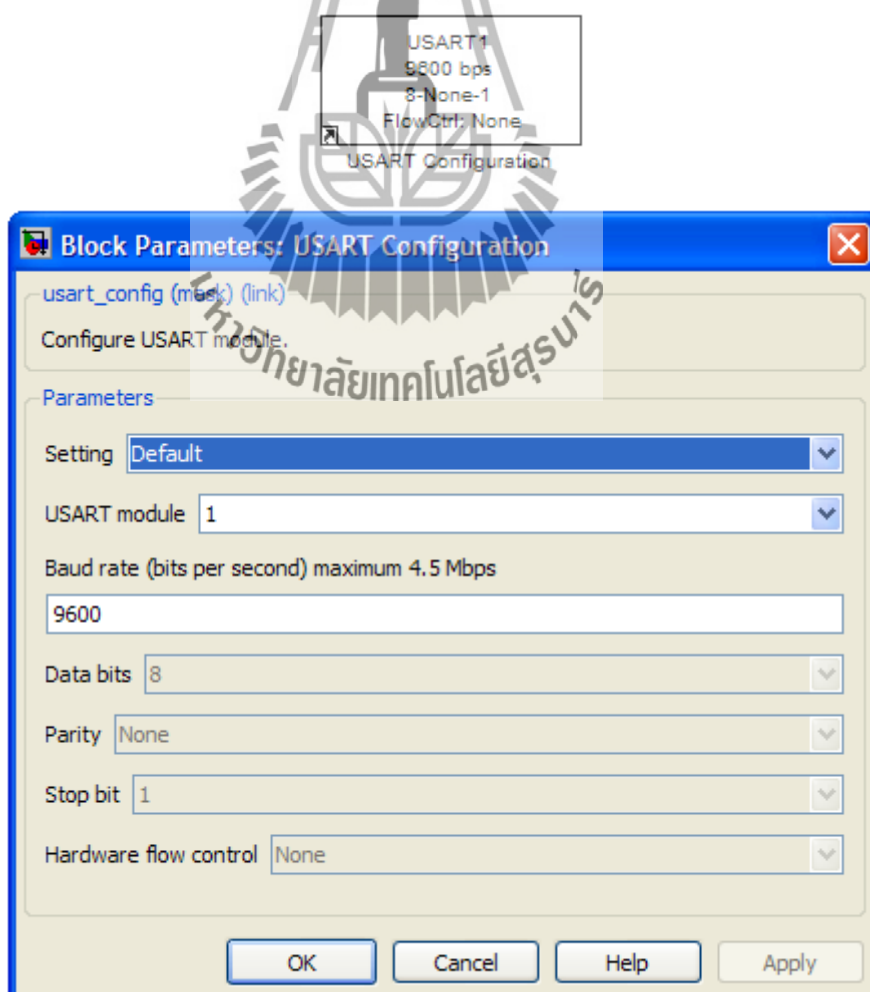


โดยอธิบายการการตั้งค่าบล็อกที่สำคัญดังนี้

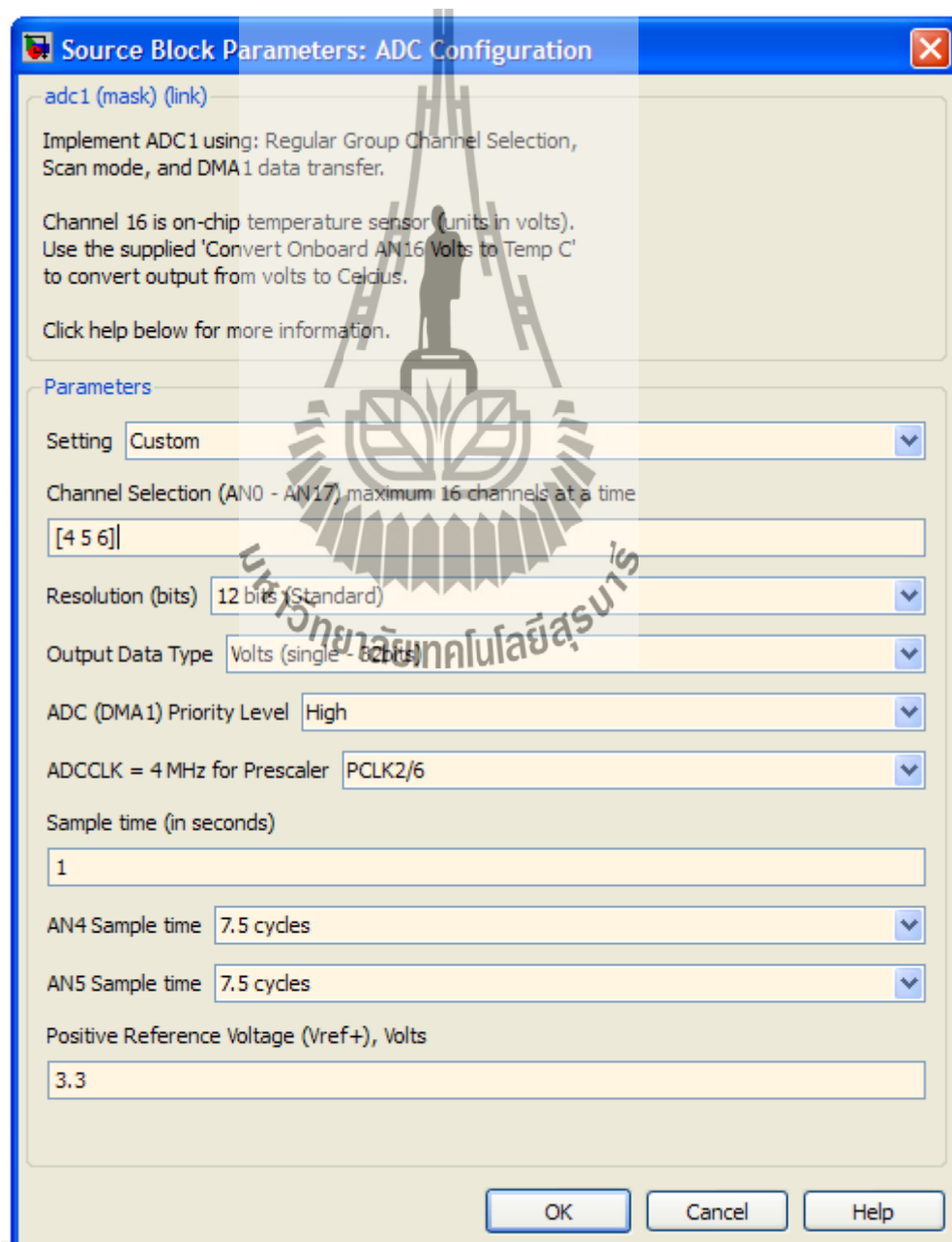
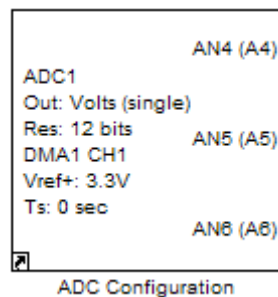
อันดับแรกในการสร้างบล็อก เลือก Compile and Download Control ก่อน แล้วจึงสร้างบล็อก Setup System Clock & SysTick เพราะสองบล็อกแรกนี้ ลำดับในการวางมีผลในการสร้างบล็อก Simulink ถัดไป ใช้สำหรับDownloadโปรแกรมลงบอร์ด



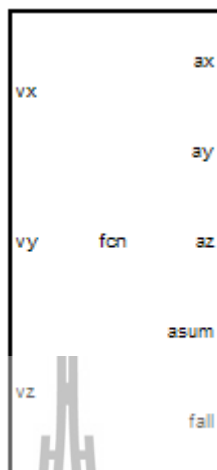
USART Configuration เป็นบล็อกสำหรับ X-bee



ADC Configuration เป็นบล็อกสำหรับเซ็นเซอร์ที่เป็น Analog



Embedded MATLAB Function เป็นบล็อกสำหรับเขียนสมการเซ็นเซอร์

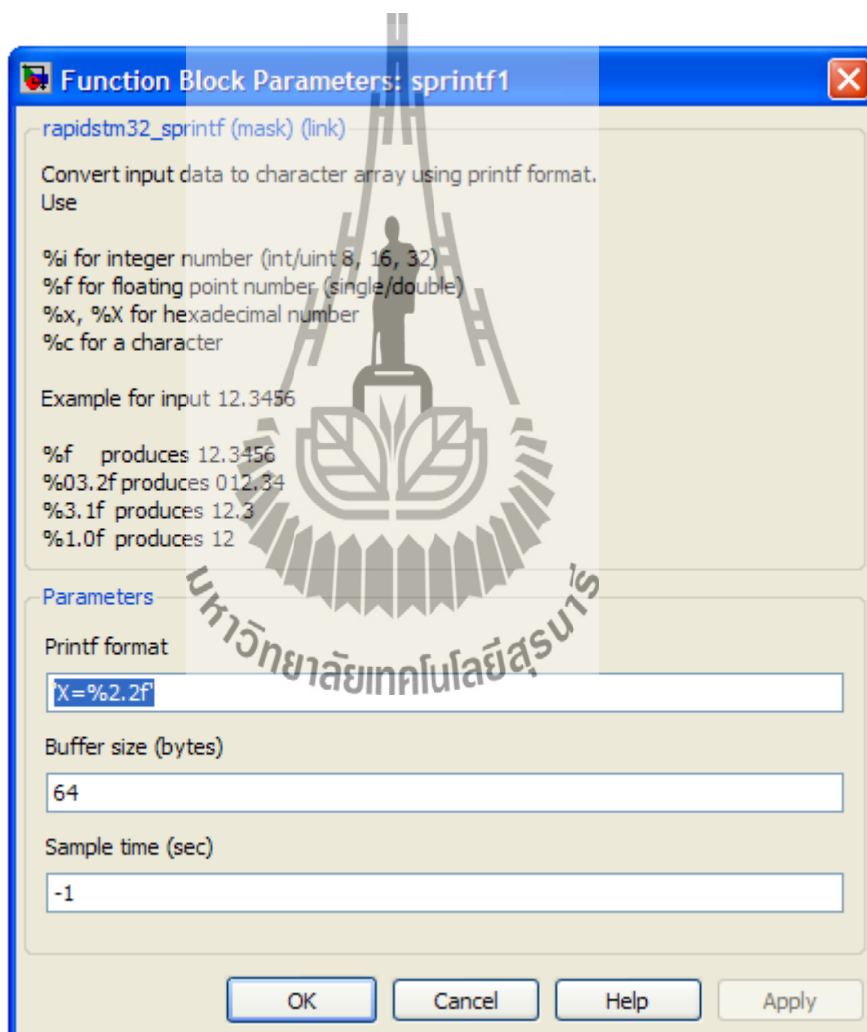
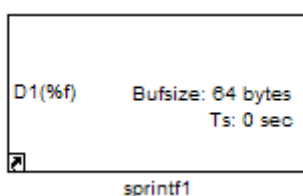


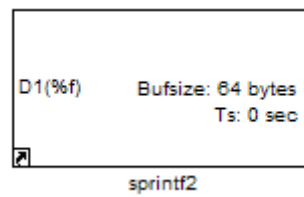
```

1  function [ax, ay, az, asum, fall] = fcn(vx, vy, vz)
2  %#eml
3
4  c1 = 1.25;
5  c2 = -2.0625;
6
7  ax = c1*vx + c2;
8  ay = c1*vy + c2;
9  az = c1*vz + c2 + 0.2;
10
11  asum = (ax^2 + ay^2 + az^2)/0.5;
12
13  fall = 0;
14  if asum>1.86
15      fall = 1;
16  end
17
18

```

Sprintf เป็นบล็อกสำหรับแสดงผล โดยในที่นี้จะแสดงผลที่ได้จากบล็อก Embedded MATLAB Function ตามลำดับดังนี้ Sprintf1 แสดงค่า X, Sprintf1 แสดงค่า X, Sprintf2 แสดงค่า Y, Sprintf4 แสดงค่า Z, Sprintf3 แสดงค่า a, Sprintf5 แสดงค่า FALL





Function Block Parameters: sprintf2

rapidstm32_sprintf (mask) (link)

Convert input data to character array using printf format.
Use

- %i for integer number (int/uint 8, 16, 32)
- %f for floating point number (single/double)
- %x, %X for hexadecimal number
- %c for a character

Example for input 12.3456

- %f produces 12.3456
- %03.2f produces 012.34
- %3.1f produces 12.3
- %1.0f produces 12

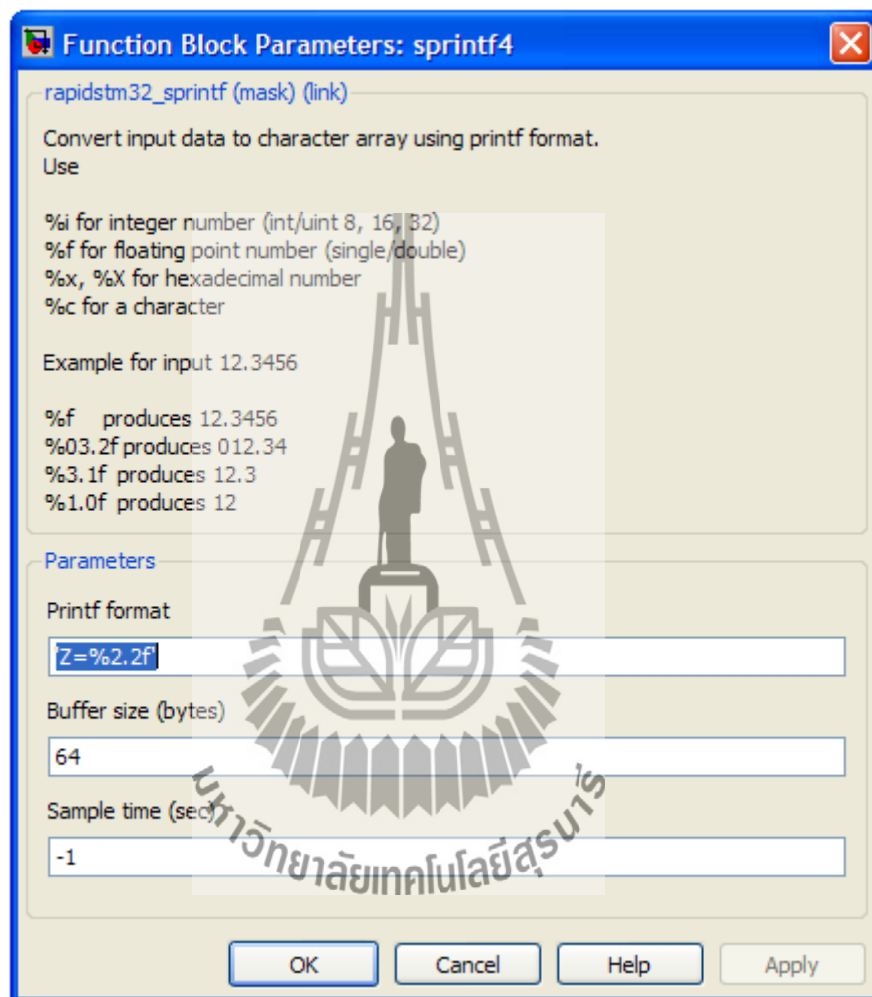
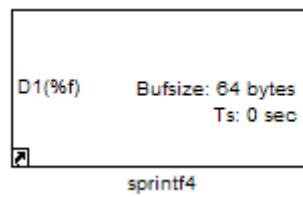
Parameters

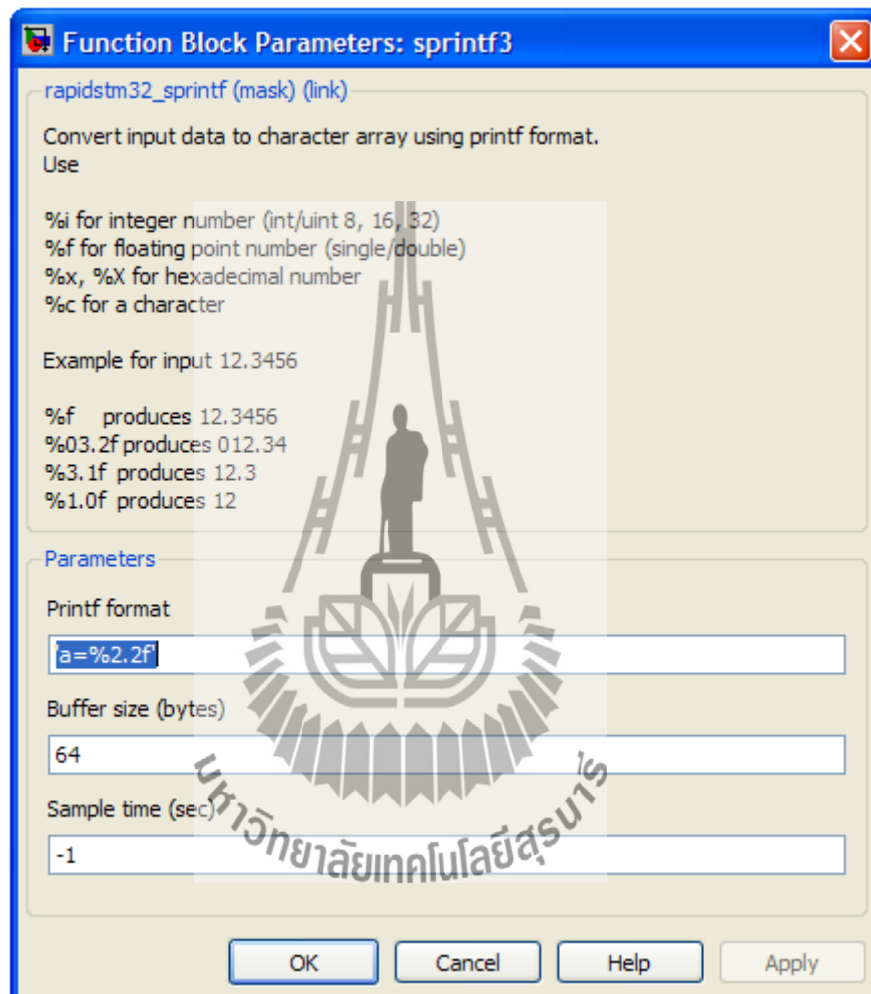
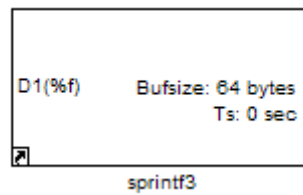
Printf format

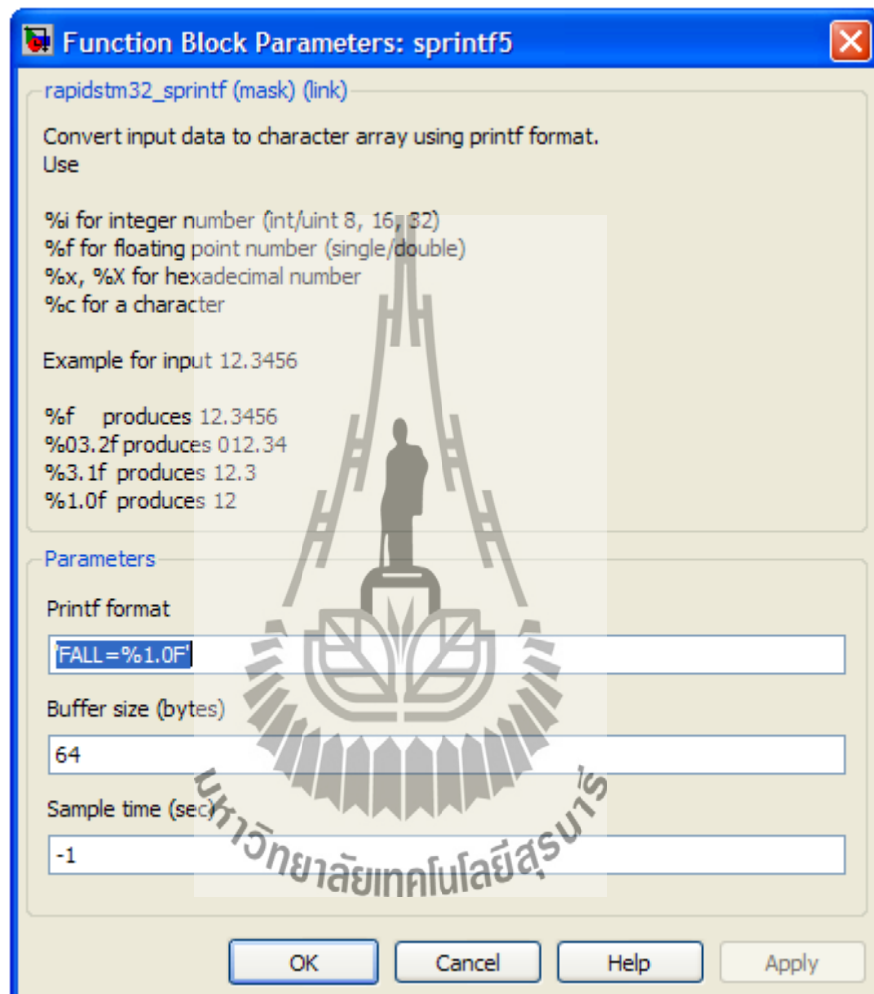
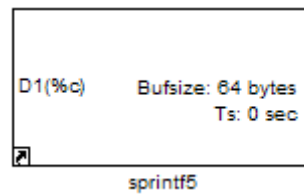
Buffer size (bytes)

Sample time (sec)

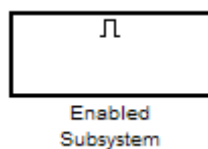
OK Cancel Help Apply



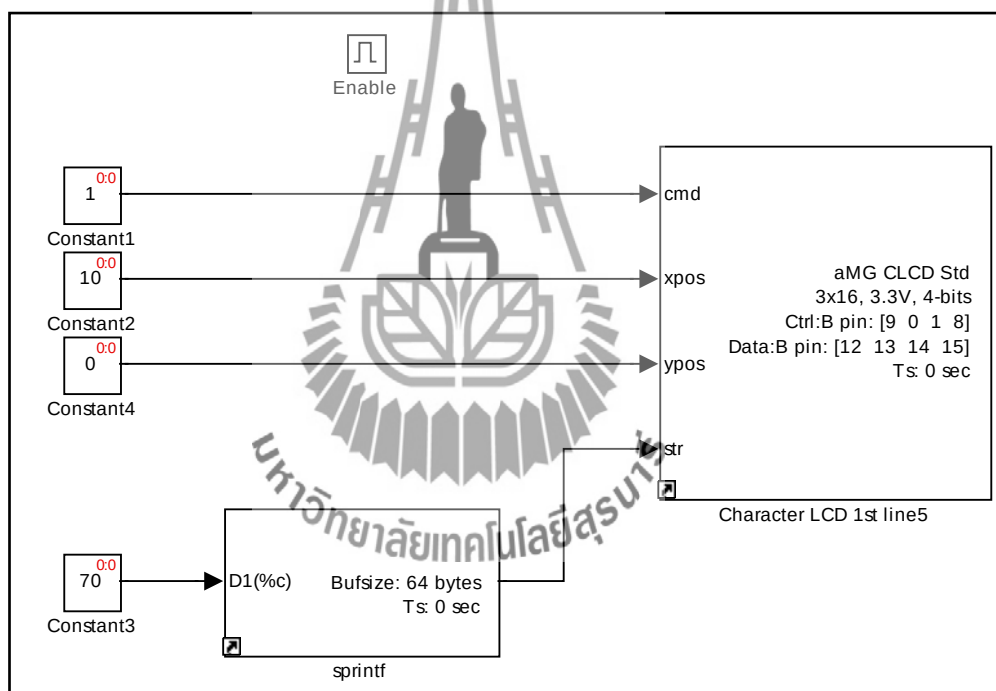




Enabled Subsystem เป็นบล็อกที่ใช้สำหรับให้ทำงานเมื่อตรงตามเงื่อนไขที่เรากำหนดไว้
 ในบล็อก Embedded MATLAB Function คือ ถ้า Fall มีค่า มากกว่า 1.86 ให้เข้ามาที่บล็อก Enabled
 Subsystem นี้

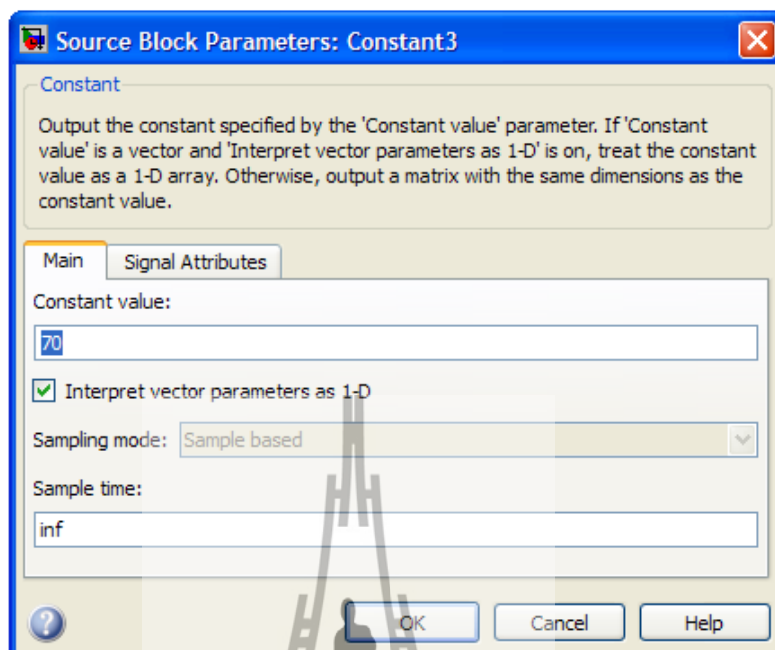


บล็อก Enabled Subsystem นี้จะสามารถเข้าไปสร้างบล็อกย่อยข้างในได้อีก โดยเราสร้าง
 เพื่อให้ส่งข้อมูลตัวอักษร F ดังนี้



โดยที่บล็อก

Constant3 กำหนดค่าดังนี้

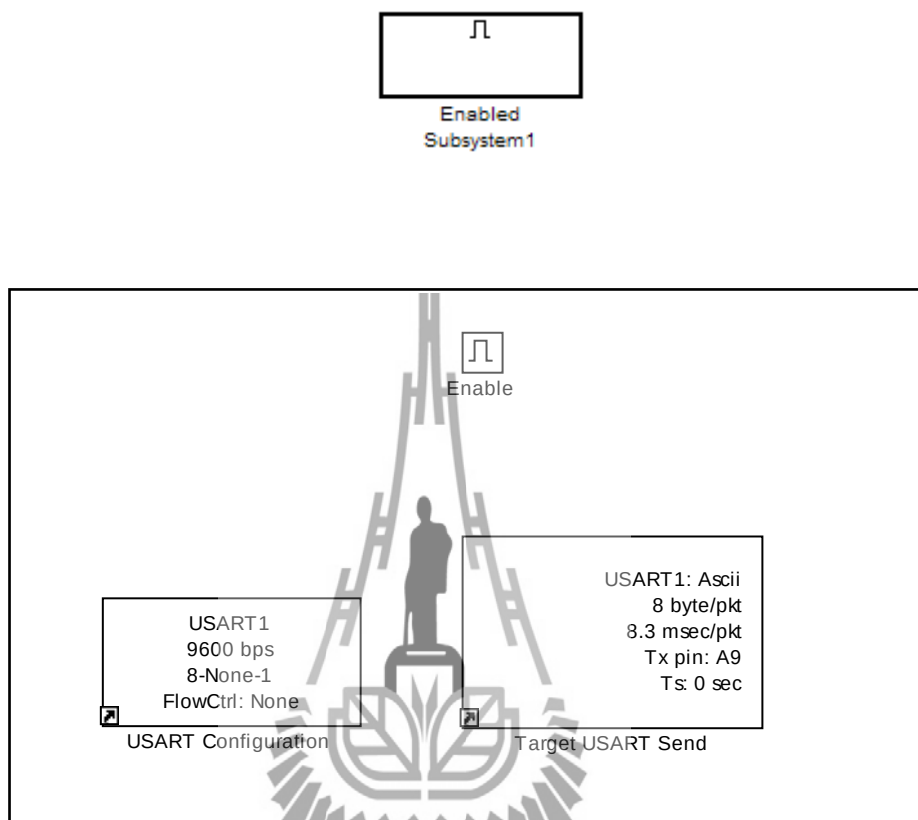


บล็อก

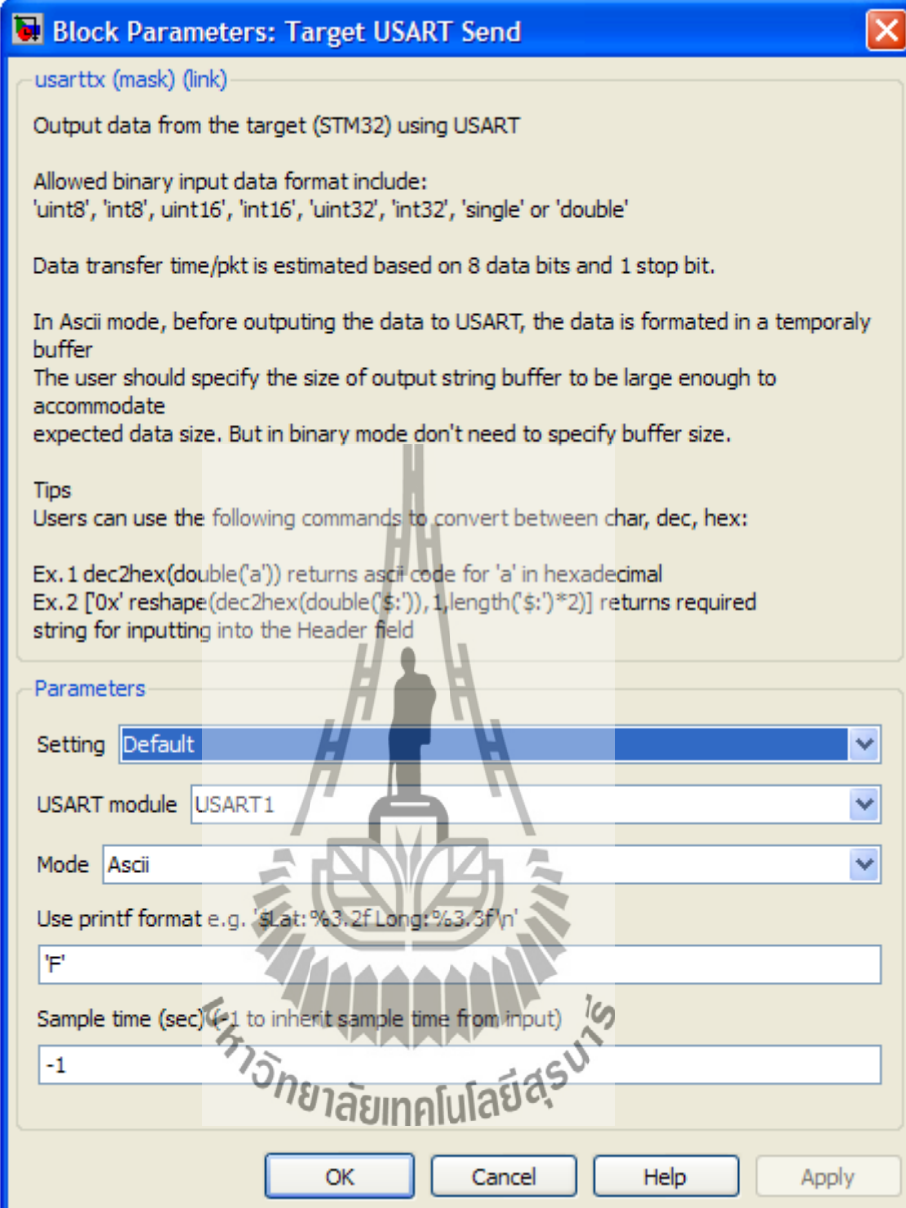
Sprintf กำหนดดังนี้



บล็อก Enabled Subsystem1 นี้เรากำหนดให้ตัวส่ง ส่งข้อมูลตัวอักษร F ผ่าน X-bee ตัวส่งไปยัง X-bee ตัวรับ โดยกำหนดบล็อกย่อยใน Enabled Subsystem1 ดังนี้



USART Configuration เป็นบล็อกสำหรับ X-bee และ Target USART Send เป็นบล็อกสำหรับการส่งข้อมูลผ่าน X-bee x บล็อกของ Target USART Send เรากำหนดให้ X-bee ส่งตัวอักษร F ดังนี้



Block Parameters: Target USART Send

usarttx (mask) (link)

Output data from the target (STM32) using USART

Allowed binary input data format include:
'uint8', 'int8', 'uint16', 'int16', 'uint32', 'int32', 'single' or 'double'

Data transfer time/pkt is estimated based on 8 data bits and 1 stop bit.

In Ascii mode, before outputting the data to USART, the data is formatted in a temporary buffer
The user should specify the size of output string buffer to be large enough to accommodate expected data size. But in binary mode don't need to specify buffer size.

Tips
Users can use the following commands to convert between char, dec, hex:
Ex.1 `dec2hex(double('a'))` returns ascii code for 'a' in hexadecimal
Ex.2 `[0x' reshape(dec2hex(double($:')),1,length('$:')*2)]` returns required string for inputting into the Header field

Parameters

Setting **Default**

USART module **USART1**

Mode **Ascii**

Use printf format e.g. 'Lat:%3.2f Long:%3.3f\n'

'F'

Sample time (sec) (-1 to inherit sample time from input)

-1

OK Cancel Help Apply

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ข

การเขียนโปรแกรมที่ใช้งานสำหรับภาครับ

สำหรับโปรแกรมในภาครับใช้ Dev C++ ในการเขียนโปรแกรมเพื่อรับข้อมูลจากภาคส่งมายัง X-bee ภาครับที่เชื่อมต่อกับคอมพิวเตอร์โน้ตบุ๊ก โดยสามารถเขียนโปรแกรมดังนี้

```
#include<windows.h>
#include<string>
#include<iostream>
using namespace std;

int main(int argc, char** argv) {
    string portn="COM1"; //ประกาศตัวแปรประเภทสตริงสำหรับเก็บชื่อพอร์ต

    bool bRead; //ประกาศตัวแปรประเภทบูลีนสำหรับเก็บค่าที่ส่งกลับมาจากฟังก์ชัน ReadFile()

    HANDLE hSerial; //ประกาศ handle สำหรับเก็บค่าที่ส่งกลับมาจากฟังก์ชัน CreateFile()

    cout<<"PORT NAME(COM1)"<<endl; //แสดงข้อความ "PORT NAME(COM1)"
    cin>>portn; //รับค่าชื่อพอร์ต แล้วเก็บลงในตัวแปร portn

    hSerial = CreateFile(portn.c_str(),
        GENERIC_READ, 0, 0, OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL, 0); //ใช้ฟังก์ชัน CreateFile() เพื่อเปิดพอร์ต

    if(hSerial!=INVALID_HANDLE_VALUE){ //เช็คค่าในตัวแปร hSerial เท่ากับ
        INVALID_HANDLE_VALUE หรือไม่
        ถ้าไม่เท่ากับแสดงว่าเปิดพอร์ตได้

        DCB config; //ประกาศ struct DCB เพื่อทำการตั้งค่าการส่งข้อมูล
```

```

config.DCBlength=sizeof(config);
config.BaudRate=CBR_9600;
config.ByteSize=8;
config.StopBits=ONESTOPBIT;
config.Parity=NOPARITY;
SetCommState(hSerial, &config);
char szBuff[1];
DWORD myBytesRead = 0;
cout<<"Open port "+portn<<endl;

cout<<"Listening..."<<endl;
while(szBuff[0]!='F'){
    bRead=ReadFile(hSerial, szBuff, sizeof(szBuff),
    &myBytesRead, NULL);

    if(bRead==FALSE){cout<<"Error to read port"<<endl;}

    CloseHandle(hSerial);

    int i1;
    while(1){
        cout<<"FALL\a"<<endl;
        for(i1=0;i1<20000000;i1+=1){}
    }
}

```

//อัตราการส่งข้อมูลเป็น 9600 บิตต่อวินาที

//ขนาดของข้อมูล 8 บิต

//มีบิตหยุด 1 บิต

//ในรูปแบบไม่มีพาริตีบิต

//ทำการตั้งค่า

//ประกาศตัวแปรเพื่อเก็บข้อมูลจากพอร์ต

//แสดงข้อความ "Open port " และชื่อพอร์ตที่อยู่ในตัวแปร portn

//แสดงข้อความ "Listening..."

//ทำการวนลูปรับข้อมูลจากพอร์ตแบบต่อเนื่องจนกว่าจะเจอตัว 'F'

//คำสั่ง ReadFile() อ่านข้อมูลจากพอร์ตแล้วเก็บไว้ในตัวแปร szBuff พร้อมทั้ง return ค่าเก็บไว้ในตัวแปร bRead เพื่อนำไปเช็คต่อไปในบรรทัดถัดไป

//เช็คว่าการอ่านข้อมูลจากพอร์ตสำเร็จหรือไม่ ถ้าไม่สำเร็จ จะแสดงข้อความว่า "Error to read port"

//ทำการปิดพอร์ต

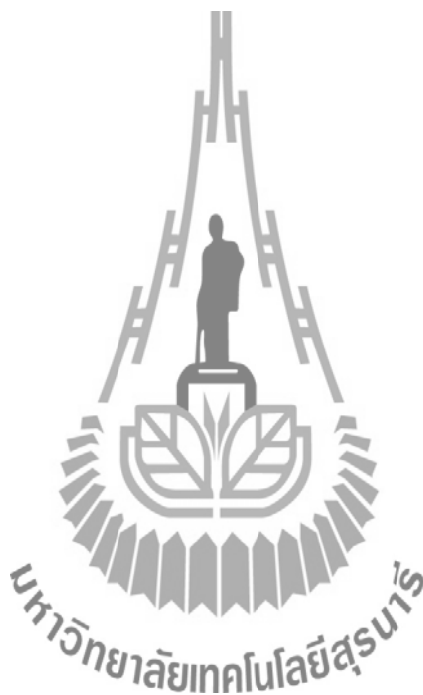
//ประกาศตัวแปรเพื่อใช้ในการวนลูปหน่วยเวลา

//วนลูปแบบไม่สิ้นสุดจนกว่าจะปิดโปรแกรม

//แสดงข้อความ "FALL" พร้อมทั้งส่งเสียงเตือน

//วนลูปเพื่อหน่วยเวลาการเตือน

```
}else{cout<<"Error to open port"<<endl; } //ถ้าค่าในตัวแปร hSerial เท่ากับ  
INVALID_HANDLE_VALUE จะแสดง  
ข้อความ "Error to open port" แสดงว่าไม่  
สามารถเปิดพอร์ตได้  
  
system("pause");  
return 0;  
}
```



ประวัติผู้เขียน



นางสาวจันทนิภา กาญจนนพวงศ์ เกิดเมื่อวัน ที่ 11
กันยายน พ.ศ. 2535 ภูมิลำเนาอยู่ที่ ตำบล ในเมือง อำเภอ
เมือง จังหวัด นครราชสีมา สำเร็จการศึกษาระดับ
มัธยมศึกษาตอน ปลายจากโรงเรียน สุรนารีวิทยา 2 อำเภอ
เมือง จังหวัดนครราชสีมา เมื่อปี พ.ศ. 2553 ปัจจุบันเป็น
นักศึกษาชั้นปีที่ 4 สาขาวิชา วิศวกรรมโทรคมนาคม
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี



นางสาวกรชนก พุทธะ เกิดเมื่อวันที่ 10 กุมภาพันธ์
พ.ศ. 2535 ภูมิลำเนาอยู่ที่ ตำบล ในเมือง อำเภอ เมือง
จังหวัดหนองคาย สำเร็จการศึกษาระดับมัธยม ศึกษา ตอน
ปลายจากโรงเรียนปทุมเทพวิทยาคาร อำเภอเมือง จังหวัด
หนองคาย เมื่อปี พ.ศ. 2553 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4
สาขา วิชา วิศวกรรมโทรคมนาคม สำนักวิชา
วิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี