เซ็นเซอร์วัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1 ด้วย Zigbee

(Wireless RPM Monitoring For SUT F1 Race Car with Zigbee)

โดย

นางสาวหนึ่งฤทัย	ปู่สสวงษ์	รหัส B5408047
นายทินกร	ไชยรักษา	รหัส B5409563
นายกษม	สุขเกษม	รหัส B5624652

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 527499 โครงงานศึกษาวิศวกรรมโทรคมนาคม

ประจำภาคการศึกษาที่ 3 ปีการศึกษา 2558

หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2554

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

เซ็นเซอร์วัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1 ด้วย Zigbee

คณะกรรมการสอบโครงงาน

____

(ผู้ช่วยศาสตราจารย์ ดร. วิภาวี หัตถกรรม)

กรรมการ/อาจารย์ที่ปรึกษาโครงงาน

____

(ผู้ช่วยศาสตราจารย์ เรืออากาศเอก ดร. ประโยชน์ คำสวัสดิ์)

กรรมการ

____

(อาจารย์ เศรษฐวิทย์ ภูฉายา)

กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับรายงานโครงงานฉบับนี้ เป็นส่วนหนึ่งของการศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม วิชา 527499 โครงงานวิศวกรรมโทรคมนาคม ประจำปีการศึกษา 2558

โครงงาน	เครื่องแสดงผลวัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1 (Wireless RPM Monitoring For SUT F1 Race Car with Zigbee)

ผู้ดำเนินงาน	นางสาวหนึ่งฤทัย	ปุสสวงษ์	B5408047
	นายทินกร	ไชยรักษา	B5409563
	นายกษม	สุขเกษม	B5624652
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.วิภาวี หัตถกรรม		
สาขาวิชา	วิศวกรรมโทรคมนาคม		

ภาคการศึกษา	1/2558

บทคัดย่อ

เนื่องจากในปัจจุบันมีการใช้เซ็นเซอร์แบบเหล็กและแบบแม่เหล็กตรวจจับการหมุนวงล้อของฟันเฟืองในรถยนต์ ส่วนมากจะเป็นการต่อแบบสายพ่วงไปยังจอแสดงผลโดยตรง ซึ่งวิธีดังกล่าวทำให้ไม่สามารถนำข้อมูลมาแสดงผลหรือวิเคราะห์แบบเวลาจริง (Real Time Monitoring) ในขณะที่ตัวรถมีการเคลื่อนที่ เพื่อดูรอบและวัดประสิทธิภาพการทำงานของเครื่องยนต์ได้ โครงงานนี้จึงเสนอการออกแบบและสร้างเครื่องมือสำหรับตรวจจับรอบออกมา ประกอบด้วยเซนเซอร์ตรวจจับโลหะซึ่งส่งสัญญาณที่ได้ผ่านเข้าโปรแกรมที่เขียนไว้ในบอร์ด Microcontroller แล้วส่งต่อข้อมูลออกไปโดยใช้ Zigbee 2.4 GHz. ไปยัง Base Station ข้อมูลที่อยู่ติดกับ Laptop ในขณะเดียวกัน Laptop จะรับข้อมูลที่ได้แสดงผลเป็นจำนวนรอบต่อเวลาแบบ Real Time – Plot

มหาวิทยาลัยเทคโนโลยีสุรนารี

กิตติกรรมประกาศ

จากการที่คณะจัดทำรายงานได้รับมอบหมายให้ทำโครงงานเรื่อง เครื่องแสดงผลวัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1 (Wireless RPM Monitoring For SUT F1 Race Car with Zigbee) ส่งผลให้คณะผู้จัดทำรายงานได้รับความรู้ความเข้าใจและประสบการณ์ต่างๆ จากการสร้างชุดควบคุมอุปกรณ์ดังกล่าว ด้วยการเขียนโปรแกรมซอร์ฟแวร์ Processing ที่ใช้เขียนโค้ดเพื่อดูความสัมพันธ์ระหว่างรอบและความเร็ว ณ ขณะนั้นซึ่งทำให้ได้ข้อมูลแบบ Real Time โดยวิธีการประมวลผลผ่านบอร์ด Arduino โดยใช่ Xbee เป็นตัวเชื่อมต่อไร้สายส่งข้อมูลหากัน และการทำโครงงานดังกล่าวนี้ได้สำเร็จลงแล้ว ทั้งนี้ด้วยความร่วมมือและได้รับการสนับสนุนจากผู้มีพระคุณ จึงทำให้สำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำจึงใคร่ขอขอบพระคุณอย่างยิ่ง คือ

1. ผศ.ดร.วิภาวี หัตถกรรม (อาจารย์ที่ปรึกษาโครงงาน)
2. ผศ.ดร.สุดเขตต์ พจน์ประไพ (อาจารย์สาขาวิชาวิศวกรรมเซรามิก)
3. ดร.เอกรงค์ สุขจิต (อาจารย์สาขาวิชาวิศวกรรมเครื่องกล)
4. นายอรรคชัย ภูผานิล
5. ชมรม Formula One SUT

ผู้ที่คอยให้ความช่วยเหลือและความรู้ต่างๆที่เกี่ยวกับโครงงาน การดูแลเอาใจใส่ติดตามโครงงานของเราและให้คำปรึกษาในทุกๆด้านแก่คณะผู้จัดทำมาโดยตลอด

คณะผู้จัดทำใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่มีส่วนร่วมในการให้ข้อมูลและเป็นที่ปรึกษาในการทำโครงงานฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้การดูแลและให้ความเข้าใจเกี่ยวกับพื้นฐานในการเขียนโปรแกรมและการทำงานของอุปกรณ์ที่ถูกต้องบนพื้นฐานความปลอดภัยเป็นหลักของผู้ใช้งาน ซึ่งคณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ
นางสาวหนึ่งฤทัย ปุสสวงษ์
นายทินกร ไชยรักษา
นายกษม สุขเกษม

สารบัญ

เรื่อง **หน้า**

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูปภาพ	จ
สารบัญตาราง	ญ
บทที่ 1 บทนำ	
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตการศึกษา	2
1.4 ขั้นตอนการดำเนินงาน	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้องและอุปกรณ์	
2.1 กล่าวนำ	5
2.2 ทฤษฎีการวัดรอบเครื่องยนต์แบบไร้สาย	5
2.3 อุปกรณ์	6
2.3.1 Arduino UNO	6
2.3.2. XBee	22
2.3.3. Reflective Optical Sensor with transistor Output	44
บทที่ 3 ชุดอุปกรณ์ต้นแบบและผลการทดลอง	
3.1 กล่าวนำ	46
3.2 เครื่องมือวัดความเร็วรอบมาตรฐาน	46
3.3 ชุดอุปกรณ์ต้นแบบ	48

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ(ต่อ)

เรื่อง **หน้า**

3.4 หลักการทำงานของอุ ปกรณ์ต้นแบบ 50

3.5 ผลการทดลอง 70

3.6 สรุปผลการทดลอง 73

บทที่ 4 ข้อสรุปโครงงาน

4.1 บทนำ 74

4.2 บทสรุปของโครงงาน 74

4.3 ปัญหาที่พบในขณะดำเนินการ 75

4.4 ข้อเสนอแนะ 76

4.5 แนวทางการพัฒนาต่อ 77

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูปภาพ

รูปภาพ	หน้า
รูปที่ 1.1 แสดงรูปแบบการทำงานทั้งระบบ	2
รูปที่ 2.1 บอร์ด Arduino UNO ด้านหน้าและหลัง	7
รูปที่ 2.2 Download the Arduino Software	8
รูปที่ 2.3 Download Arduino 1.0.1	9
รูปที่ 2.4 Install the Arduino Windows Driver	9
รูปที่ 2.5 แสดงขั้นตอน : Click ไปที่ Device Manager เพื่อ Start device manager	10
รูปที่ 2.6 แสดงขั้นตอน : Click ไปที่ Device Manager เพื่อ Start device manager	10
รูปที่ 2.7 แสดงขั้นตอน : Click ไปที่ Device Manager เพื่อ Start device manager	11
รูปที่ 2.8 แสดงขั้นตอน : Installing the Device Driver	12
รูปที่ 2.9 แสดงขั้นตอน : Update Driver Software	12
รูปที่ 2.10 แสดงขั้นตอน : Update Driver Software	13
รูปที่ 2.11 แสดงขั้นตอน : Update Driver Software	13
รูปที่ 2.12 แสดงขั้นตอน : Update Driver Software	14
รูปที่ 2.13 แสดงขั้นตอน : Update Driver Software	14
รูปที่ 2.14 แสดงขั้นตอน : Update Driver Software	15
รูปที่ 2.15 แสดงขั้นตอน : Setup Arduino Software	16
รูปที่ 2.16 แสดงขั้นตอน : Setup Arduino Software	17
รูปที่ 2.17 แสดงการเชื่อมต่อแบตเตอรี่	18
รูปที่ 2.18 หน้าต่างเริ่มต้นของโปรแกรม Arduino 1.0.5	19
รูปที่ 2.19 วิธีตั้งค่า Board	19
รูปที่ 2.20 วิธีการตั้งค่า Serial Port	20
รูปที่ 2.21 วิธีดูพอร์ท	20

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูปภาพ(ต่อ)

รูปภาพ หน้า

รูปที่ 2.22 แสดงหน้าต่างของโปรแกรม Arduino 1.0.5 21

รูปที่ 2.23 Xbee 22

รูปที่ 2.24 แสดงหน้าต่างของโมดูล Xbee-Pro และบอร์ดสำหรับต่อใช้งาน 23

รูปที่ 2.25 แสดงส่วนประกอบชุดอุปกรณ์ทั้งหมดในชุด Xbee 3V/5V 24

รูปที่ 2.26 ZX-Xbee บอร์ดเชื่อมต่อโมดูล Xbee-PRO กับคอมพิวเตอร์ 27

รูปที่ 2.27 วงจร ZX-Xbee บอร์ดเชื่อมต่อโมดูล Xbee/Xbee-PRO กับ คอมพิวเตอร์ 28

รูปที่ 2.28 รายละเอียดADX-Xbeeบอร์ดเชื่อมต่อโมดูล Xbee-PRO กับอุปกรณ์ภายนอก 29

รูปที่ 2.29 แสดงวงจรของบอร์ด ADX-Xbee ทั้งสองรุ่นคือสำหรับ ADX-XBee 5V/3V 30

รูปที่ 2.30 ขาของ Module 31

รูปที่ 2.31 แสดงการเชื่อมต่อบอร์ด ZX-Xbeeในกรณีที่คอมพิวเตอร์ไม่มีพอร์ตอนุกรม 32

รูปที่ 2.32 หน้าต่างการเชื่อมต่อ X-CTU 33

รูปที่ 2.33 Comtest/Query Modem 34

รูปที่ 2.34 X-CTU [com1] 35

รูปที่ 2.35 Xbee-Interface Board 38

รูปที่ 2.36 ส่วนประกอบ Xbee-Interface Board 38

รูปที่ 2.37 การเชื่อมต่อ USB 39

รูปที่ 2.38 การเชื่อมต่อแบบ TTL Serial 39

รูปที่ 2.39 Xbee Interface board 40

รูปที่ 2.40 Xbee Interface board 41

รูปที่ 2.41 การตั้งค่าใช้งาน 41

รูปที่ 2.42 การตั้งค่าใช้งาน 42

รูปที่ 2.43 การตั้งค่าใช้งาน 43

รูปที่ 2.44 Xbee Interface board 43

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูปภาพ(ต่อ)

รูปภาพ	หน้า
รูปที่ 2.45 Sensor TCRT5000(L)	44
รูปที่ 2.46 Sensor TCRT5000(L)	45
รูปที่ 2.47 การกำหนดค่าหน้าจอบนโปรแกรม Processing	46
รูปที่ 2.48 การดาวน์โหลดซอฟต์แวร์	47
รูปที่ 2.49 การแตกไฟล์ Processing	47
รูปที่ 2.50 การเปิดไฟล์ Processing.exe	48
รูปที่ 2.51 คลิกเปิดโปรแกรม Processing เพื่อใช้งาน	48
รูปที่ 2.52 หน้าต่างของโปรแกรม Processing	49
รูปที่ 3.1 เครื่องมือวัดความเร็วรอบ	51
รูปที่ 3.2 แสดงระบบเซนเซอร์ไร้สายเพื่อวัดเร็วรอบ	52
รูปที่ 3.3 แสดงการเชื่อมต่อระบบของอุปกรณ์ต้นแบบ	53
รูปที่ 3.4 อุปกรณ์ที่ใช้ในการอัพโหลด	54
รูปที่ 3.5 นำเซนเซอร์ติดตั้งบนแท่นทดลองเพื่อวัดรอบ	55
รูปที่ 3.6 ต่อไฟ 5 V จากบอร์ด Arduino Uno R3	55
รูปที่ 3.7 ทดสอบปรับค่าให้ตรงกับอุปกรณ์วัดความเร็วรอบแบบมาตรฐาน	56
รูปที่ 3.8 เปิดโปรแกรม Arduino	57
รูปที่ 3.9 ขั้นตอนการอัพโหลดโค้ดลงบอร์ด	57
รูปที่ 3.10 เขียนโค้ดลงโปรแกรมแล้วตรวจสอบ	58
รูปที่ 3.11 เปิดมอนิเตอร์เพื่อดูค่าที่ได้จากการวัด	58
รูปที่ 3.12 เปิดโปรแกรม Processing 2.0	59
รูปที่ 3.13 เขียนโค้ดลงในโปรแกรมเพื่อคำนวณค่า	59
รูปที่ 3.14 ตรวจสอบความถูกต้องของโค้ด	60
รูปที่ 3.15 กราฟที่แสดงการวัดรอบเมื่อผ่านการส่งสัญญาณแบบไร้สาย	60

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูปภาพ(ต่อ)

รูปภาพ	หน้า
รูปที่ 2.45 Sensor TCRT5000(L)	44
รูปที่ 2.46 Sensor TCRT5000(L)	45
รูปที่ 2.47 การกำหนดค่าหน้าจอบนโปรแกรม Processing	46
รูปที่ 2.48 การดาวน์โหลดซอฟต์แวร์	47
รูปที่ 2.49 Unzip File Processing	47
รูปที่ 2.50 การเปิดไฟล์ Processing.exe	48
รูปที่ 2.51 คลิกเปิดโปรแกรม Processing เพื่อใช้งาน	48
รูปที่ 2.52 หน้าต่างของโปรแกรม Processing	49
รูปที่ 3.1 เครื่องมือวัดความเร็วรอบ	51
รูปที่ 3.2 แสดงระบบเซนเซอร์ไร้สายเพื่อวัดเร็วรอบ	52
รูปที่ 3.3 แสดงการเชื่อมต่อระบบของอุปกรณ์ต้นแบบ	53
รูปที่ 3.4 อุปกรณ์ที่ใช้ในการอัพโหลด	54
รูปที่ 3.5 นำเซนเซอร์ติดตั้งบนแท่นทดลองเพื่อวัดรอบ	55
รูปที่ 3.6 ต่อไฟ 5 V จากบอร์ด Arduino Uno3	55
รูปที่ 3.7 ทดสอบปรับค่าให้ตรงกับอุปกรณ์วัดความเร็วรอบแบบมาตรฐาน	56
รูปที่ 3.8 เปิดโปรแกรม Arduino	57
รูปที่ 3.9 ขั้นตอนการอัพโหลดโค้ดลงบอร์ด	57
รูปที่ 3.10 เขียนโค้ดลงโปรแกรมแล้วตรวจสอบ	58
รูปที่ 3.11 เปิดมอนิเตอร์เพื่อดูค่าที่ได้จากการวัด	58
รูปที่ 3.12 เปิดโปรแกรม Processing 2.0	59
รูปที่ 3.13 เขียนโค้ดลงในโปรแกรมเพื่อคำนวณค่า	59
รูปที่ 3.14 ตรวจสอบความถูกต้องของโค้ด	60
รูปที่ 3.15 กราฟที่แสดงการวัดรอบเมื่อผ่านการส่งสัญญาณแบบไร้สาย	60

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญรูปภาพ(ต่อ)

รูปภาพ	หน้า
รูปที่ 3.39 แสดงการตั้งค่า XBee (Router)	71
รูปที่ 3.40 แสดงการอัพเดท โปรแกรมสำเร็จ	72
รูปที่ 3.41 ทดสอบการส่งผ่านข้อมูลระหว่าง Router และ Coordinator	73

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 1.1 ขั้นตอนการดำเนินงาน	3
ตารางที่ 2.1 การจัดขา PIN	26
ตารางที่ 3.5.1 การคำนวณค่าเปอร์เซ็นความผิดพลาดจากการเก็บค่าของอุปกรณ์ต้นแบบ	74
ตารางที่ 3.5.2 การทดสอบส่งค่าความเร็วรอบแบบไร้สายแปรกับระยะทางและช่วงเวลา	76
ตารางที่ 4.1 ปัญหาที่พบขณะดำเนินการ สาเหตุของปัญหาและวิธีการแก้ปัญหา	79

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

ปัจจุบันมีการใช้เซ็นเซอร์ทางด้านงานอุตสาหกรรมเข้ามาช่วยทำงานร่วมกับ อุปกรณ์ต่างๆเซ็นเซอร์แม่เหล็กจึงมีความจำเป็นอย่างยิ่งสำหรับการตรวจจับโลหะที่ไหลผ่านหรือหมุนผ่านตัว เซนเซอร์ แม้ในรถยนต์ก็ใช้เซ็นเซอร์ประเภทที่ทำงานในลักษณะนี้เหมือนกัน โดยการหมุนของข้อเหวี่ยง ฟันเฟืองและการหมุนของล้อ ส่วนมากจะเป็นแบบสาย ส่งค่าที่ได้ เข้ามายังหน้าจอแสดงผลแก่ผู้ขับรถเท่านั้น โครงงานนี้จึงสร้างตัวตรวจรู้ไร้สายโดยใช้ Zigbee เพื่อส่งค่าการวัดรอบ (Revolutions Per Minute , RPM) โดยทำการเปลี่ยนแปลงค่า RPM ของทีมวิศวกรยานยนต์ข้างสนามแข่งพร้อมกับผู้ขับขี่โดยจะใช้บอร์ด Microcontroller กับ Zigbee และมี Hall Effect Sensor ในการสร้างอุปกรณ์ตัวนี้ขึ้นมา ขณะเซ็นเซอร์ตรวจจับการเคลื่อนไหวส่งข้อมูลไปให้บอร์ด Arduino กับ Xbee ไปยัง Xbee ปลายทางอีกตัวที่อยู่กับ Laptop จะอ่านค่าเพื่อประมวลผลในการตรวจจับนี้สามารถนำข้อมูลที่ได้ไปประยุกต์ใช้ประโยชน์ตามความต้องการในรูปแบบอื่นๆได้

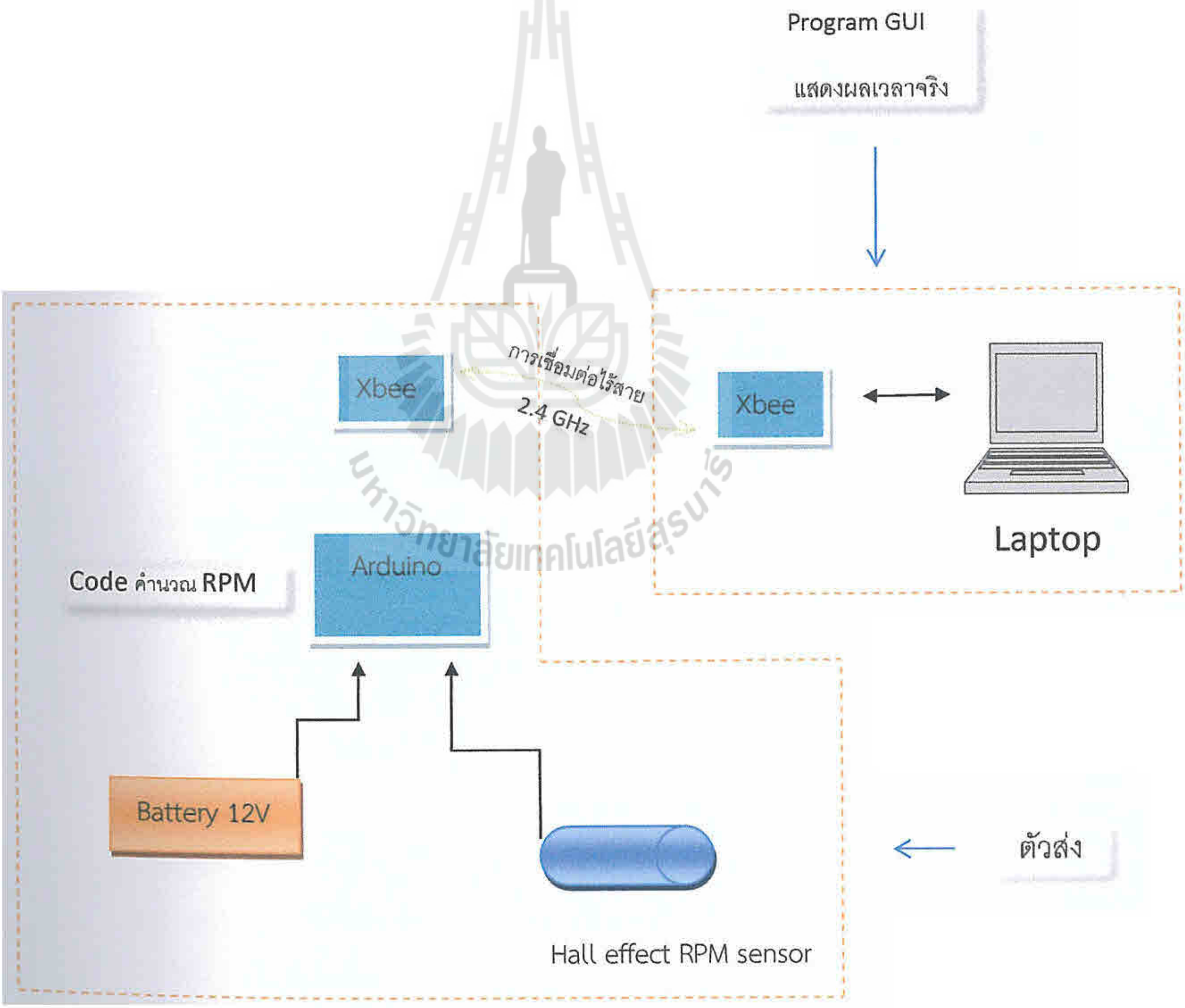
1.2 วัตถุประสงค์

1. เพื่อสร้างอุปกรณ์ต้นแบบตัวตรวจรู้ไร้สายเพื่อวัดค่า RPM
2. เพื่อศึกษาการเขียนโปรแกรมแสดงผลข้อมูล ด้วย Processing Version 2.0
3. เพื่อศึกษาขีดจำกัดของอุปกรณ์
4. การแก้ไขปัญหาด้าน Hardware ในสภาพการใช้งานจริง

มหาวิทยาลัยเทคโนโลยีสุรนารี

1.3 ขอบเขตของงาน

1. ออกแบบวงจรไฟฟ้า ที่จะส่งให้ Arduino Board รองรับได้
2. เขียนโปรแกรมประมวลผลค่าของเซนเซอร์ เป็นวินาทีหรือนาที เพื่อแสดงผลผ่านจอคอมพิวเตอร์
3. ศึกษารายละเอียดและวิเคราะห์ หาขีดจำกัดการทำงานของอุปกรณ์ที่เชื่อมต่อให้มีประสิทธิภาพสูงสุดในการรับส่งข้อมูล
4. ทำการสร้างอุปกรณ์ต้นแบบใช้ในการทดลองให้ได้ตามวัตถุประสงค์ที่ตั้งไว้



รูปที่1.1 แสดงรูปแบบการทำงานทั้งระบบระบบ

1.4 ขั้นตอนการดำเนินงาน

กิจกรรม	พ.ศ.2557 – พ.ศ.2558											
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.
1. ศึกษา ค้นคว้าหาข้อมูล	↔	↔										
2. เขียนโครงการเพื่อเสนอโครงการกับอาจารย์ที่ปรึกษาโครงการ			↔									
3. จัดซื้ออุปกรณ์ในการโครงการนี้				↔	↔							
4. เขียนโปรแกรมควบคุมเซนเซอร์ และกำหนดการเชื่อมต่อสัญญาณระหว่างอุปกรณ์					↔	↔						
5. เขียนโปรแกรมเพื่อแสดงผลที่ได้จากการเชื่อมต่อบนคอมพิวเตอร์							↔	↔				
6. สร้างอุปกรณ์ต้นแบบเพื่อใช้ในการทดสอบให้ได้ตามวัตถุประสงค์								↔	↔			
7. ทดสอบและแก้ไขปัญหาที่เกิดขึ้น									↔	↔		
8. สรุปผลการทดลองและเขียนรายงาน											↔	↔
9. นำเสนอโครงงาน												↔

ตารางที่ 1.1 ตารางขั้นตอนการดำเนินงาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจการทำงานของระบบตัวตรวจรู้ไร้สาย
2. นำไปประยุกต์ใช้วัดรอบอุปกรณ์ที่มีการหมุนแบบต่างๆได้
3. สามารถทำงานเป็นทีมได้

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 2
ทฤษฎีที่เกี่ยวข้องและอุปกรณ์

2.1 กล่าวนำ

เนื้อหาในบทนี้จะกล่าวถึงการทำงานที่เกี่ยวข้องกับการใช้งานของระบบโปรแกรมซอฟต์แวร์ต่างๆรวมไปถึงการเปรียบเทียบอุปกรณ์ที่ใช้งานให้มีประสิทธิภาพการทำงานเพิ่มมากยิ่งขึ้นและอีกทั้งยังกล่าวถึงขั้นตอนพื้นฐานในการเขียนโปรแกรม Arduino, Processing และการตั้งค่าโปรแกรม และกล่าวถึงการใช้งาน Xbee เบื้องต้นในการส่งสัญญาณแบบไร้สาย นอกจากนี้ยังสามารถแสดงถึงอุปกรณ์ในการใช้งานอีกด้วย

2.2 ทฤษฎีการวัดรอบเครื่องยนต์แบบไร้สาย

การวัดรอบเครื่องยนต์แบบไร้สายนั้นเป็นการวัดรอบโดยใช้ความถี่ตามขนาดของ Xbee เป็นตัวกำหนดความถี่และระยะทางในการรับสัญญาณ การวัดรอบ เครื่องยนต์นั้นคือการวัดรอบการหมุนของเครื่องยนต์ โดยใช้หน่วยในการนับเป็น รอบต่อนาที จะวัดจากรอบของเพลาข้อเหวี่ยงที่หมุน เพื่อให้เกิดกำลังงานมาขับเคลื่อนรถยนต์ ถ้ายิ่งกดคันเร่งมาก เครื่องยนต์ก็ยิ่งมีรอบสูง ทำงานมาก วัดรอบก็จะมีเลขเป็นหลักหน่วย จาก 0ไล่ไปตั้งแต่ 1-8 ตัวเลขนี้จะคูณด้วยหนึ่งพัน จะเท่ากับรอบของการทำงานเครื่องยนต์ต่อนาที เช่น เลข 3 ก็คือ เครื่องยนต์ทำงาน 3,000 รอบ/นาที เป็นต้น สำหรับในรอบเดินเบา ไม่มีการเร่ง เข็มจะชี้ไม่ถึงเลข 1 ก็เฉลี่ยประมาณ 700-800 รอบ แต่ถ้ารอบเดินเบาเกินเลข 1 แสดงว่าผิดปกติ ควรรีบเข้าเช็คเครื่องยนต์ ยกเว้นตอนสตาร์ทเครื่องเย็น จะมีระบบออโตเมติกโช้ค (Automatic Choke) ดึงรอบเดินเบาขึ้นสูงเพื่อไม่ให้เครื่องยนต์ดับ แต่ถ้าเครื่องยนต์ร้อนแล้วรอบจะตกลงเอง รถสมัยใหม่จะมีการตัดรอบเพื่อกันเครื่องยนต์พัง พอถึงโซนนี้เครื่องยนต์ก็จะตัดการทำงาน ไปต่อเกินกว่านี้ไม่ได้ เหยียบคันเร่งยังไงก็ไม่ไปเพื่อป้องกันการเสียหาย

อย่างไรก็ตาม แม้จะมีการตัดรอบ แต่ก็ไม่ควรจะลากรอบไปถึงบริเวณนี้บ่อยๆ เครื่องยนต์จะเกิดความเสียหายและอายุการใช้งานสั้นลง รอบเครื่องยนต์ที่เหมาะสมในการใช้งานเช่น เครื่องยนต์เบนซินโดยปกติแล้ว การขับขี่แบบใช้งานปกติ รอบเปลี่ยนเกียร์ที่เหมาะสมที่สุด จะอยู่

ในช่วงระหว่าง 2,500-3,000 รอบในช่วงนี้เครื่องยนต์จะมีกำลังพอดี สามารถพารถให้วิ่งไปได้อย่างต่อเนื่อง

2.3 อุปกรณ์

2.3.1 Arduino UNO

Arduino UNO R3 ดังรูปที่ 2.1 เป็น Platform ของ Input/Output บอร์ดอย่างง่ายที่มี Input/Output ขั้นพื้นฐานที่เพียงพอต่อการใช้งานและการเรียนรู้โดยตัวบอร์ดนี้จะมาพร้อมกับชุดคำสั่งที่ใช้ควบคุม Port Input/Output ไม่ว่าจะเป็น Digital Port, Analog Port, PWM และ Serial Port ซึ่งเชื่อมต่ออุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าใช้ชิพไมโครคอนโทรลเลอร์เพียงตัวเดียวในการสั่งงาน อีกทั้งมีโปรแกรมถูกพัฒนามาสำหรับเขียนลงบนบอร์ดเพื่อเรียกใช้การทำงาน โดยสามารถดาวน์โหลดได้ฟรี จึงเหมาะสมในการนำมาเรียนรู้ถึงการทำงานและประยุกต์ใช้ในการสร้างอุปกรณ์พื้นฐานอย่างยิ่ง Arduino UNO R3 มีจุดเด่นในเรื่องของความเข้าใจง่ายต่อการเรียนรู้และใช้งานเนื่องจากมีการออกแบบคำสั่งต่างๆขึ้นมาสนับสนุนการใช้งานด้วยรูปแบบที่ง่ายไม่ซับซ้อนและมีข้อดีจากบอร์ดสำเร็จรูปตัวอื่นคือ

1. มีราคาไม่แพงเนื่องจากมีรหัสต้นทาง (Source Code) และวงจรแจกฟรีสามารถต่อวงจรขึ้นมาใช้งานได้เองสะดวก
2. ใช้งานง่ายมีโปรแกรมพัฒนาที่ไม่ซับซ้อนมีโปรแกรมพัฒนา Arduinoใช้งานง่ายสำหรับมือใหม่และมีความสามารถครบตามความต้องการของนักพัฒนามืออาชีพ

Arduino UNO R3 เป็นบอร์ดไมโครคอนโทรลเลอร์ชนิดหนึ่ง ที่ใช้ชิพ AVR ขนาดเล็กเป็นตัวประมวลผลและสั่งงานให้เหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์และนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์ Input/Output ต่างๆได้มากมาย Chip และ IC ภายในบอร์ด Arduino UNO R3 ที่สำคัญได้แก่

- ATmega328P-PU
- หน่วยความจำโปรแกรมแบบ FLASH ขนาด 32 กิโลไบต์
- หน่วยความจำข้อมูลแบบ SRAM ขนาด 2 กิโลไบต์
- หน่วยความจำข้อมูลแบบ EEPROM ขนาด 1 กิโลไบต์
- สนับสนุนการเชื่อมต่อแบบ I2C bus
- พอร์ตอินพุตเอาต์พุตจำนวน 23 บิต
- วงจรแปลงอนาลอกเป็นดิจิตอลขนาด 10 บิตในตัวจำนวน 8 ช่อง

ในช่วงระหว่าง 2,500-3,000 รอบในช่วงนี้เครื่องยนต์จะมีกำลังพอดี สามารถพารถให้วิ่งไปได้อย่างต่อเนื่อง

2.3 อุปกรณ์

2.3.1 Arduino UNO

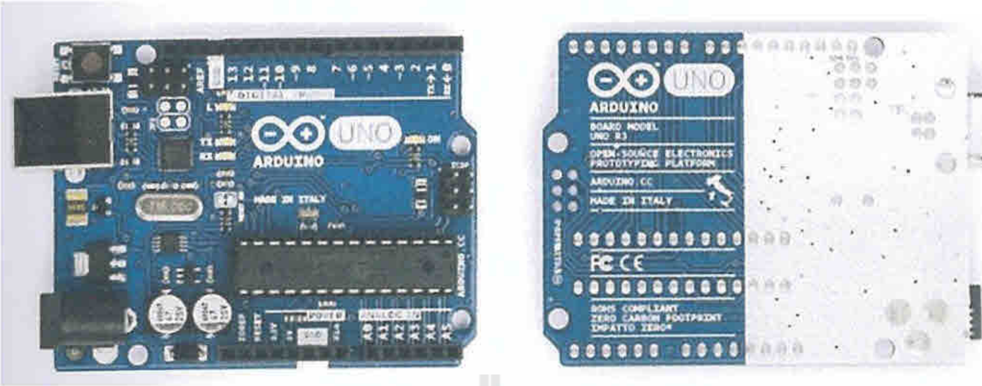
Arduino UNO R3 ดังรูปที่ 2.1 เป็น Platform ของ Input/Output บอร์ดอย่างง่ายที่มี Input/Output ขั้นพื้นฐานที่เพียงพอต่อการใช้งานและการเรียนรู้โดยตัวบอร์ดนี้จะมาพร้อมกับชุดคำสั่งที่ใช้ควบคุม Port Input/Output ไม่ว่าจะเป็น Digital Port, Analog Port, PWM และ Serial Port ซึ่งเชื่อมต่ออุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าใช้ชิพไมโครคอนโทรลเลอร์เพียงตัวเดียวในการสั่งงาน อีกทั้งมีโปรแกรมถูกพัฒนามาสำหรับเขียนลงบนบอร์ดเพื่อเรียกใช้การทำงาน โดยสามารถดาวน์โหลดได้ฟรี จึงเหมาะสมในการนำมาเรียนรู้ถึงการทำงานและประยุกต์ใช้ในการสร้างอุปกรณ์พื้นฐานอย่างยิ่ง Arduino UNO R3 มีจุดเด่นในเรื่องของความเข้าใจง่ายต่อการเรียนรู้และใช้งานเนื่องจากมีการออกแบบคำสั่งต่างๆขึ้นมาสนับสนุนการใช้งานด้วยรูปแบบที่ง่ายไม่ซับซ้อนและมีข้อดีจากบอร์ดสำเร็จรูปตัวอื่นคือ

1. มีราคาไม่แพงเนื่องจากมีรหัสต้นทาง (Source Code) และวงจรแจกฟรีสามารถต่อวงจรขึ้นมาใช้งานได้เองสะดวก
2. ใช้งานง่ายมีโปรแกรมพัฒนาที่ไม่ซับซ้อนมีโปรแกรมพัฒนา Arduinoใช้งานง่ายสำหรับมือใหม่และมีความสามารถครบตามความต้องการของนักพัฒนามืออาชีพ

Arduino UNO R3 เป็นบอร์ดไมโครคอนโทรลเลอร์ชนิดหนึ่ง ที่ใช้ชิพ AVR ขนาดเล็กเป็นตัวประมวลผลและสั่งงานให้เหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์และนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์ Input/Output ต่างๆ ได้มากมาย Chip และ IC ภายในบอร์ด Arduino UNO R3 ที่สำคัญได้แก่

- ATmega328P-PU
- หน่วยความจำโปรแกรมแบบ FLASH ขนาด 32 กิโลไบต์
- หน่วยความจำข้อมูลแบบ SRAM ขนาด 2 กิโลไบต์
- หน่วยความจำข้อมูลแบบ EEPROM ขนาด 1 กิโลไบต์
- สนับสนุนการเชื่อมต่อแบบ I2C bus
- พอร์ตอินพุตเอาต์พุตจำนวน 23 บิต
- วงจรแปลงอนาลอกเป็นดิจิตอลขนาด 10 บิตในตัวจำนวน 8 ช่อง

- ทำงานได้ตั้งแต่ย่านแรงดัน 1.8-5.5 โวลต์
- ความถี่ใช้งานสูงสุด 20 MHz



รูปที่ 2.1 บอร์ด Arduino UNO R3 ด้านหน้าและด้านหลัง

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ชนิดหนึ่งในตระกูลAVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ซึ่งตัวบอร์ดถูกออกแบบมาให้ใช้งานได้ง่าย จึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้ยังสามารถดัดแปลงเพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ดหรือโปรแกรมต่อได้อีกด้วย ดังนั้นจึงเป็นเรื่องง่ายที่จะใช้คอมพิวเตอร์บอร์ดเดียวที่มีประสิทธิภาพและได้รับการพิจารณาทั้งในตลาดงานอดิเรกและที่เป็นมืออาชีพ Arduino เป็นการพัฒนาแบบ Open Source ซึ่งหมายความว่าเป็นฮาร์ดแวร์ที่ราคาเหมาะสมกับการพัฒนาซอฟแวร์ฟรี โดยโครงการของ Arduino เริ่มต้นที่ประเทศอิตาลี เป็นสถานที่ซึ่งต้องการจะพัฒนาฮาร์ดแวร์ต้นทุนในราคาที่ต่ำสำหรับการออกแบบการทำงานร่วมกันและสถานที่ที่ดีที่สุดสำหรับฮาร์ดแวร์อยู่ในประเทศสหรัฐอเมริกา

การเขียนโปรแกรมภาษา Arduino เป็นรุ่นที่เรียบง่ายของภาษาซีC/C++ ถ้าผู้ใช้สามารถใช้และเข้าใจในภาษาซี การเขียนโปรแกรม Arduino จะเป็นที่คุ้นเคยและเป็นไปได้ง่าย แต่ถ้าไม่ทราบวิธีในการเขียนก็ไม่ต้องกังวล เพียงแค่เรามีคำสั่งที่มีความจำเป็นในการปฏิบัติหน้าที่ของสิ่งที่เราต้องการจะทำหรือรันมันออกมาเท่านั้น

คุณลักษณะที่สำคัญของ Arduino คือการที่คุณสามารถสร้างโปรแกรมควบคุมในเครื่องคอมพิวเตอร์หลักที่โหลดข้อมูลโดยตรงได้กับ Arduino และมันจะเริ่มการทำงานอัตโนมัติ โดยลดการเชื่อมต่อสาย USB เข้ากับเครื่องคอมพิวเตอร์โดยที่โปรแกรมจะยังคงทำงานทุกครั้งที่ผู้ใช้กดปุ่มรีเซตและเอาแบตเตอรี่ออก แต่เมื่อเริ่มเชื่อมต่อกับแบตเตอรี่ โปรแกรมสุดท้ายที่เก็บไว้จะทำงาน ซึ่งหมายความว่ามันสามารถเชื่อมต่อไปที่เครื่องคอมพิวเตอร์หลักได้ด้วยเช่นกันอีกทั้งยังสามารถพัฒนาหรือแก้ไขปัญหาโปรแกรมได้ทัน

2.3.1.1 สิ่งที่ต้องการสำหรับระบบการทำงาน

1. Arduino Board

2. สาย USB สำหรับการเขียนโปรแกรม

3. แบตเตอรี่หรือแหล่งจ่ายไฟภายนอก 9V (สำหรับการดำเนินงานStand-Alone)

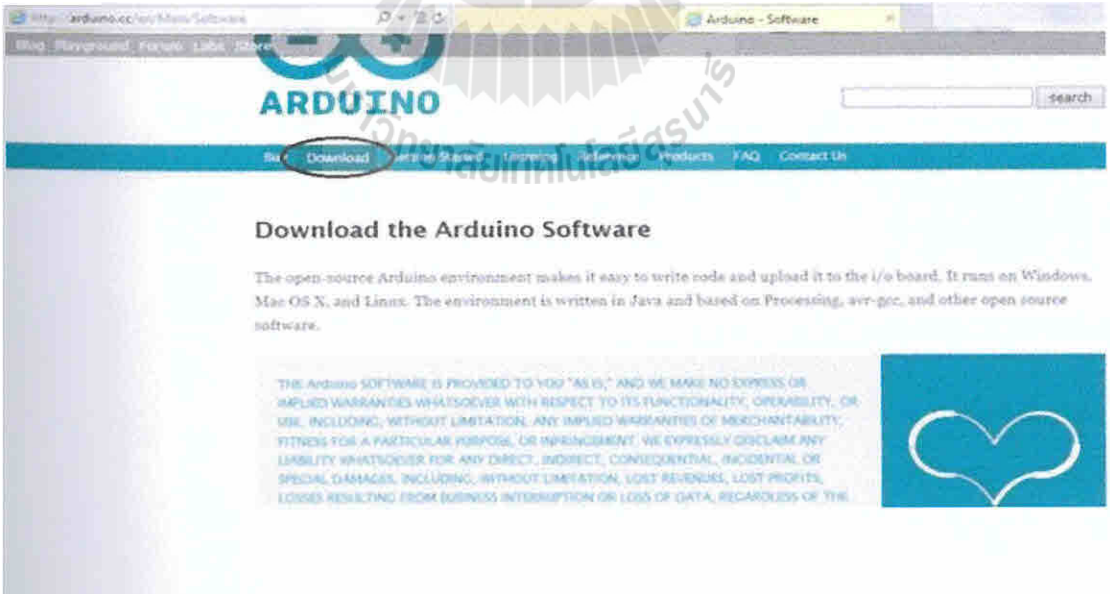
4. บัดกรีทั่วไปสำหรับวงจรภายนอกและลวด 22 กรัมสำหรับการบัดกรีเชื่อมต่อ

5. เครื่องคอมพิวเตอร์หลักที่ใช้สำหรับพัฒนา Arduinoมีรุ่นที่รองรับสำหรับ Windows,Mac และ Linux

2.3.1.2 การติดตั้งซอฟแวร์

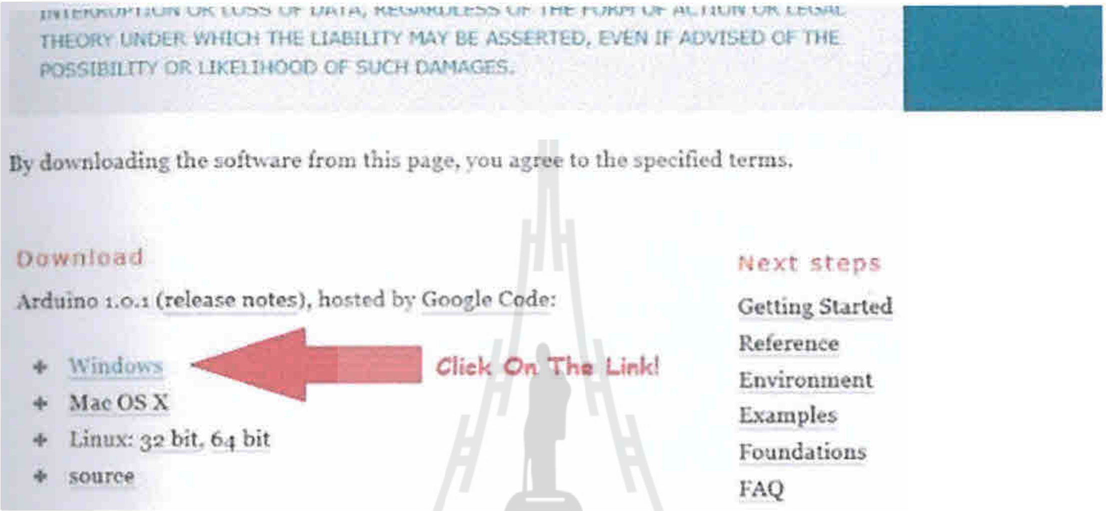
2.3.1.2.1 Download

ให้ไปที่ http://arduino.cc/en/Main/Softwareและกดคลิกที่ download เพื่อเข้าสู่ the download page.



รูปภาพที่ 2.2 Download the Arduino Software

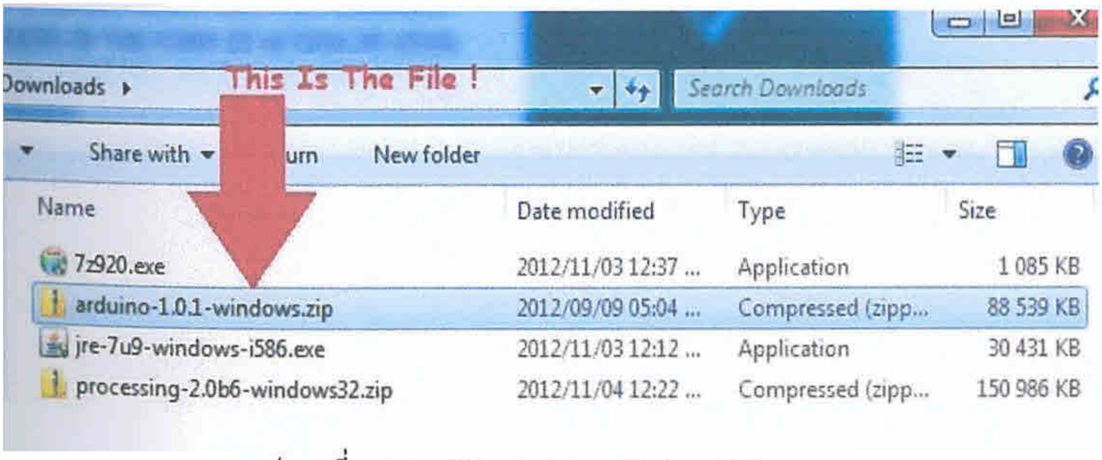
ที่หน้าต่าง Download ให้ Click ไปที่ลิ้ง Windows เพื่อที่จะเริ่มต้น Download Arduino software สำหรับระบบปฏิบัติการWindowsซึ่งซอฟต์แวร์นี้พัฒนาขึ้นมาเพื่อการเขียนโปรแกรมลงบนบอร์ด Arduino UNO โดยจะสามารถรองรับได้ทั้งระบบปฏิบัติการ Window Linux และ Mac OS X และภายในเว็ปไซต์ก็จะมีตัวอย่างซอฟต์แวร์ระบบเปิด (open source)แต่ในที่นี้เราจะยกตัวอย่างระบบปฏิบัติการที่เราใช้คือแบบWindows ตามรูปภาพด้านล่างนี้



รูปภาพที่ 2.3 Download Arduino 1.0.1

2.3.1.2.2 InstallFile

หลังจาก Download สำเร็จแล้วให้เลือกที่อยู่ของ Downloaded file ไว้บนระบบ System และทำการ Extract the folder จาก Zipped file วางลงบนตำแหน่งที่เหมาะสม



รูปภาพที่ 2.4 Install the Arduino Windows Drivers

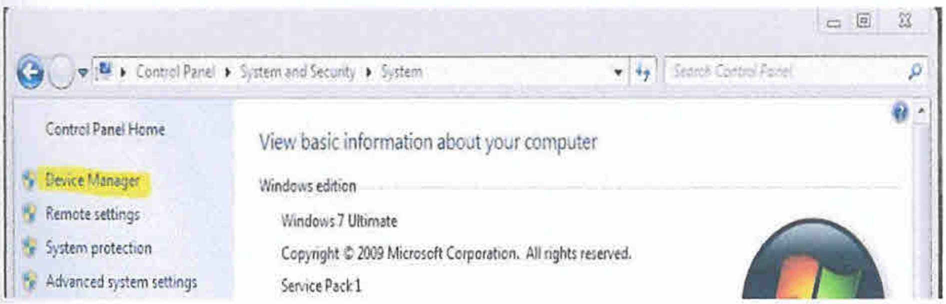
1. ทำการPlug the Arduino Board ลงบนเครื่อง PC

ทำการ Plug the Arduino board ลงบนเครื่อง PC จากนั้น Windows จะพยายาม Install drivers แต่สุดท้ายแล้วก็จะ Install ไม่สำเร็จ

2. ทำการ Start the Windows Device Manager

ให้ Click ไปที่ปุ่มหน้าต่าง Start menu คลิกขวาที่ My Computer บน Start menu จากนั้นก็ Click ไปที่ Properties or Manage จาก Pop-up Menu เพื่อทำการเปิด Device Manager

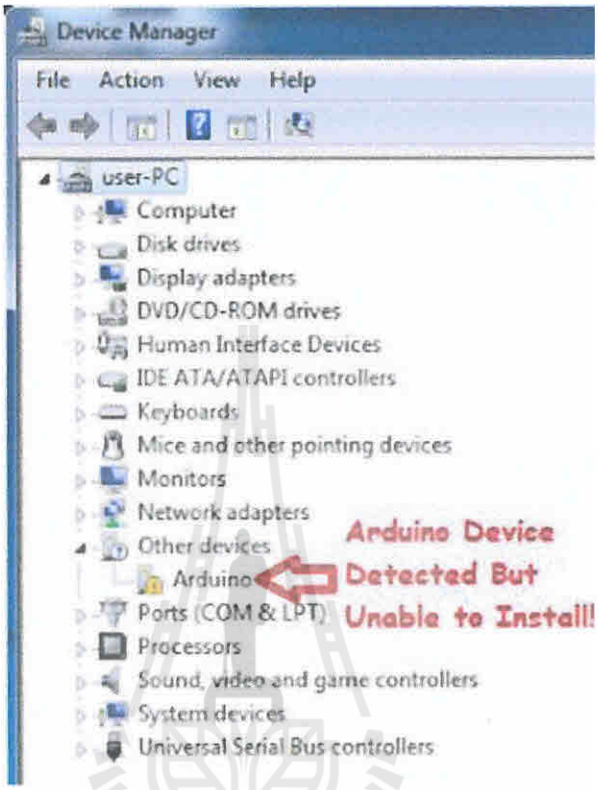
รูปภาพที่ 2.5 Click ไปที่ Device Managerเพื่อ Start device manager



รูปภาพที่ 2.6 Click ไปที่ Device Manager เพื่อ Start device manager

Device Manager จะเปิดและแสดง Arduino Device ที่เราทำการ Connect ไว้

(ขึ้นอยู่กับว่าใช้ board ชนิดไหน ชื่อก็จะแสดงขึ้นมาให้เห็น)

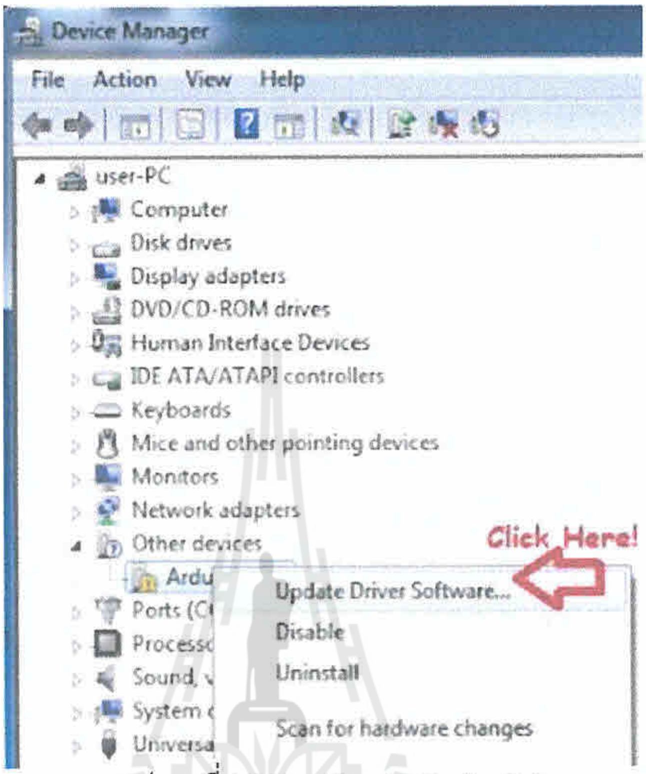


รูปภาพที่ 2.7 Click ไปที่ Device Manager เพื่อ Start device manager

จะเห็นว่าเครื่องหมายตกใจที่แสดงขึ้นสีเหลือง แสดงว่าอุปกรณ์ Arduino นั้นไม่สามารถทำการ Install ได้ ถ้ามาถึงขั้นตอนนี้ก็ต้องทำการ Installing the Device Driver

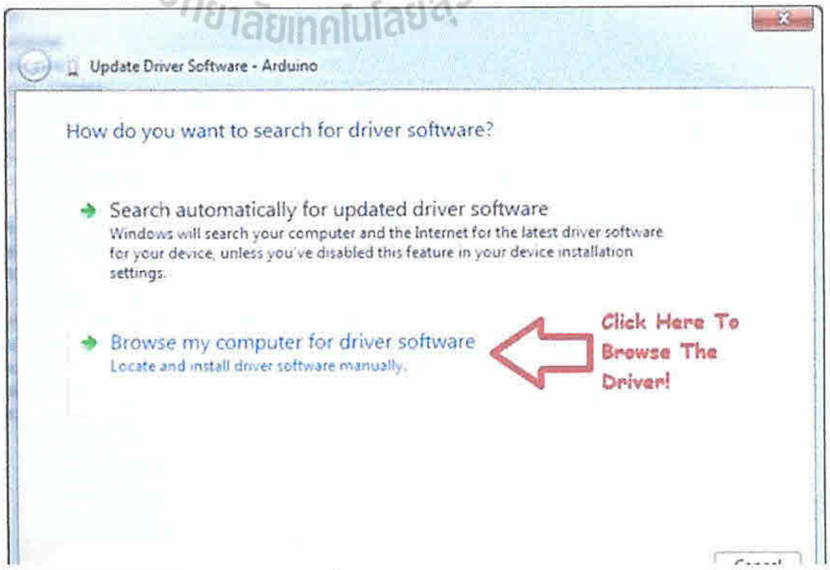
2.3.1.2.3 Installing the Device Driver

ในหน้าต่างของ Device Manager, ให้คลิกขวา ไปที่ Arduino board แล้วก็ click **Update Driver Software** บน pop-up menu



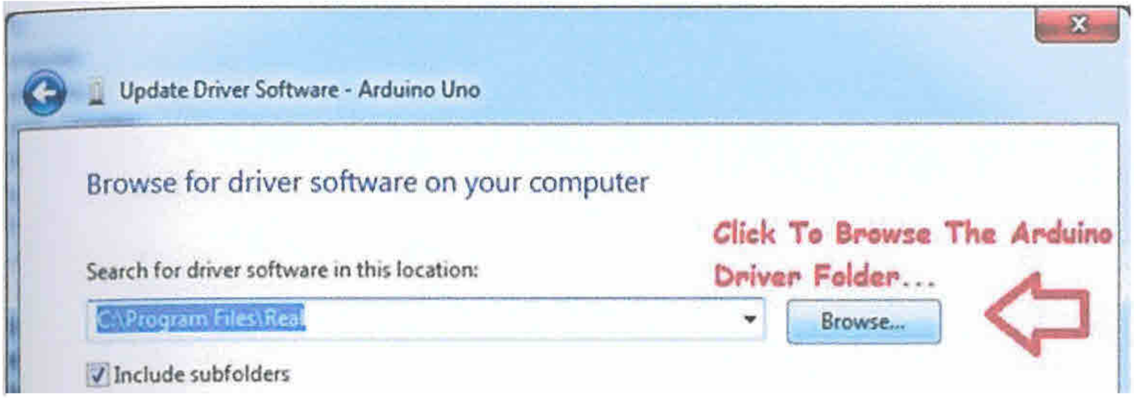
รูปภาพที่ 2.8 Installing the Device Driver

จะมีกล่องขึ้น pop-up มาโชว์ว่า **Update Driver Software** ให้คลิกที่ **Browse my computer for driver software** เพื่อที่จะ install Driver Software Manually



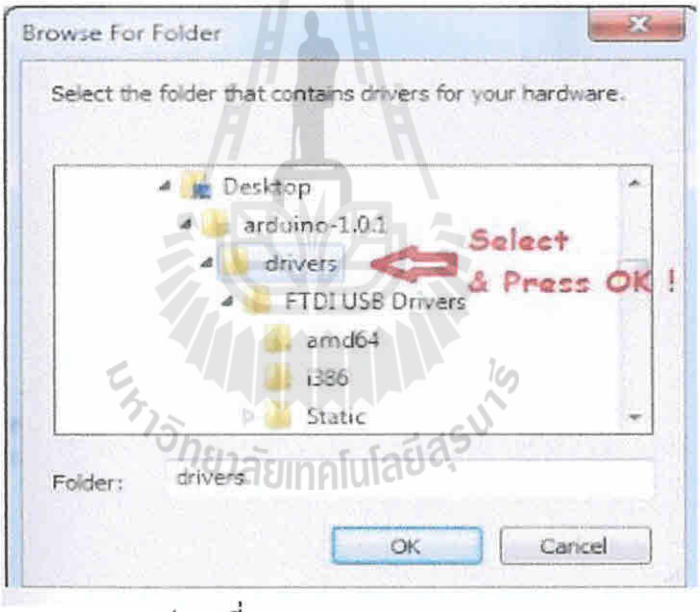
รูปภาพที่ 2.9 Update Driver Software

ต่อไปเลือก click the **Browse...** button



รูปภาพที่ 2.10 Update Driver Software

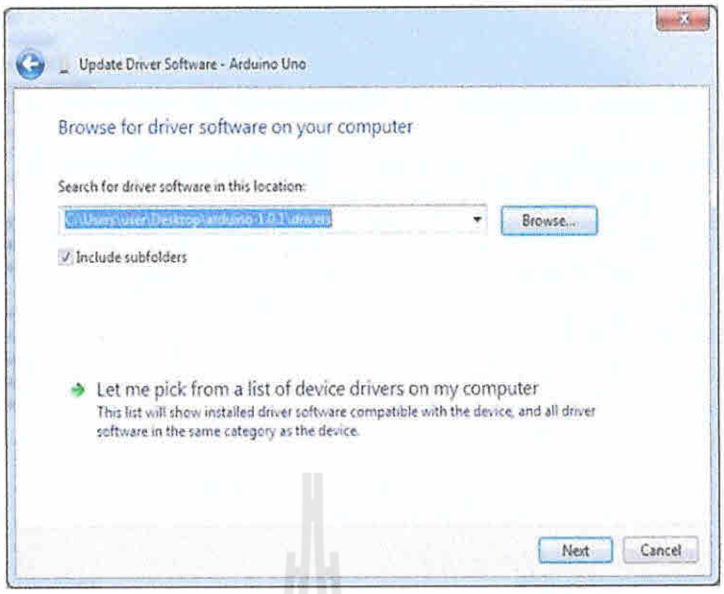
เลือกไปที่ drivers folder ใน Arduino folder ที่ได้ download มา



รูปภาพที่ 2.11 Update Driver Software

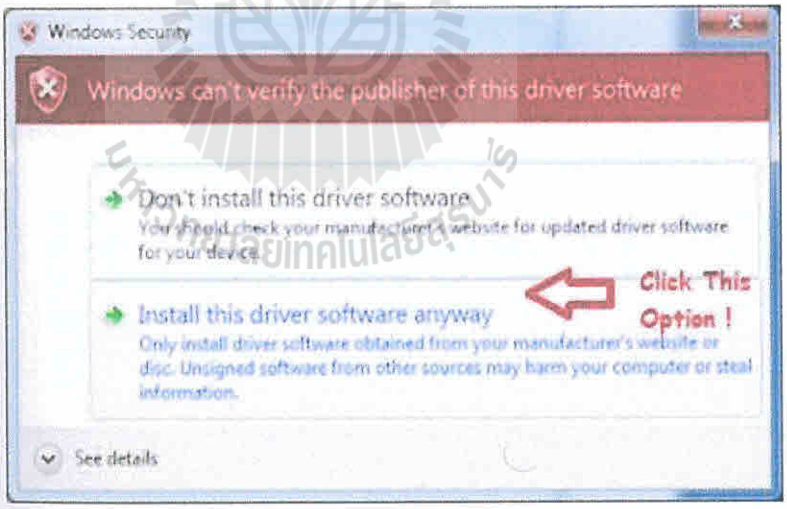
มหาวิทยาลัยเทคโนโลยีสุรนารี

หลังจากเลือกไปที่ driver folder เรียบร้อยแล้ว ให้ click **Next**



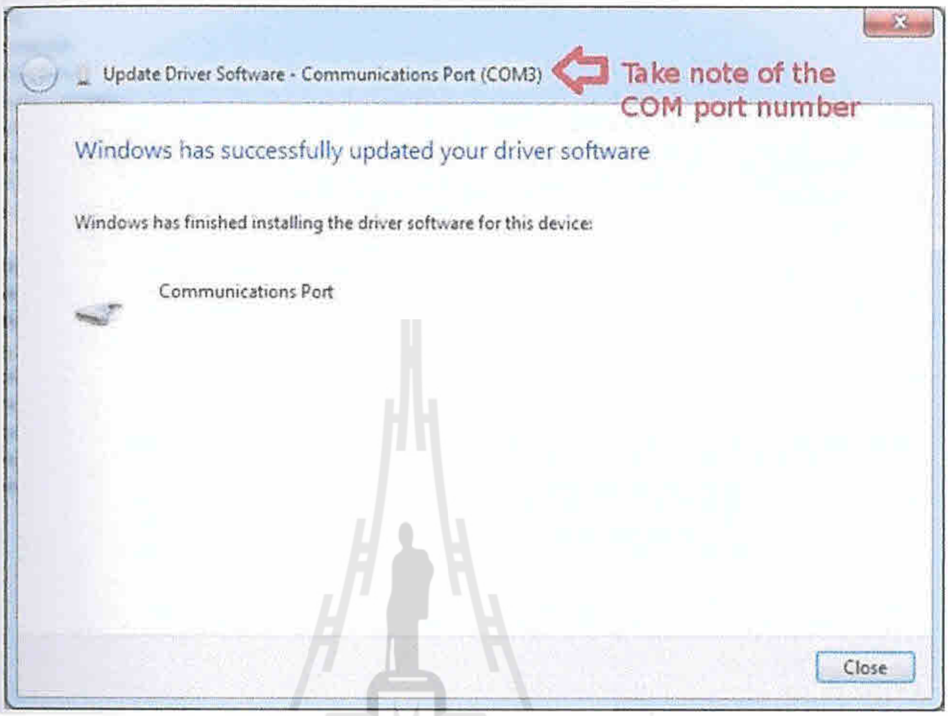
รูปภาพที่ 2.12 Update Driver Software

จะมี กล่อง pops up ขึ้นมา Click **Install this driver software anyway** to continue the Arduino Board Drivers....



รูปภาพที่ 2.13 Update Driver Software

เมื่อทำการ Install Driveเป็นที่เรียบร้อยแล้ว จะมี Dialog Box ตามรูปข้างล่างเราต้องทราบ Port Number ที่ บอร์ด Arduino เราทำการเชื่อมต่ออยู่ในตัวอย่างนี้คือ Port COM3 แต่บอร์ดที่ใช้งานจริงของคุณ อาจจะมาเป็น Port COM อื่นๆ



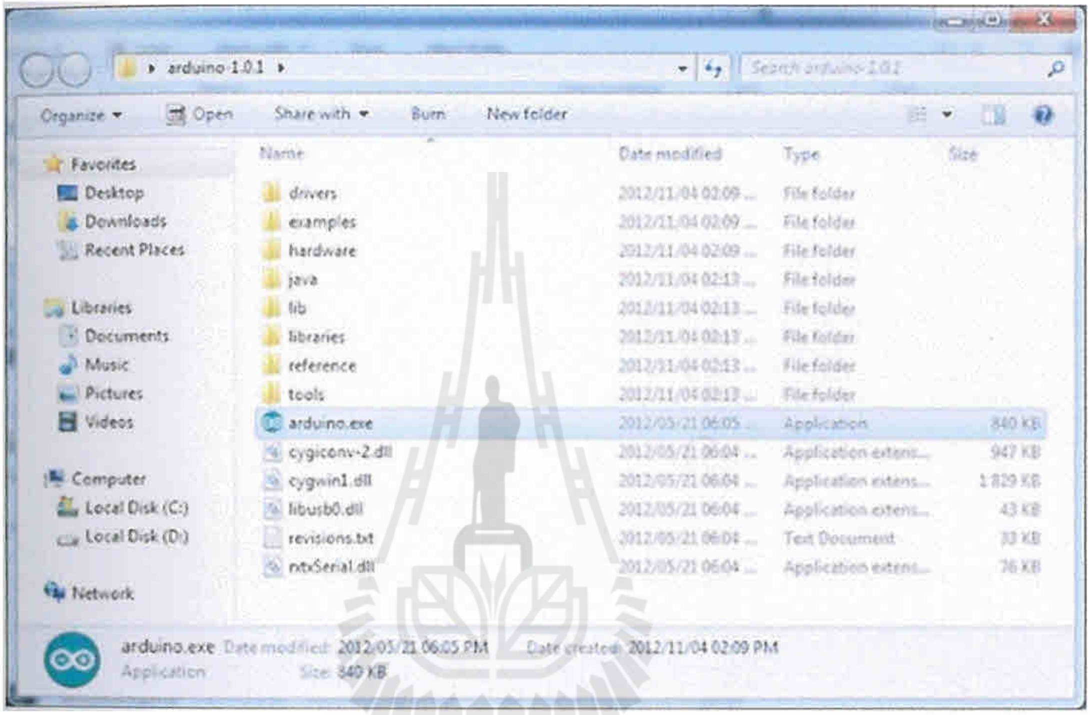
รูปภาพที่ 2.14 Update Driver Software

ถ้าเห็นข้อความนี้แสดงว่าคุณ ได้ทำการ Install Driver เป็นที่เรียบร้อยแล้วก็ทำการ "Close" ได้เลย

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.3.1.2.4 ทำการ Setup Arduino Software หลังจาก Installation แล้ว

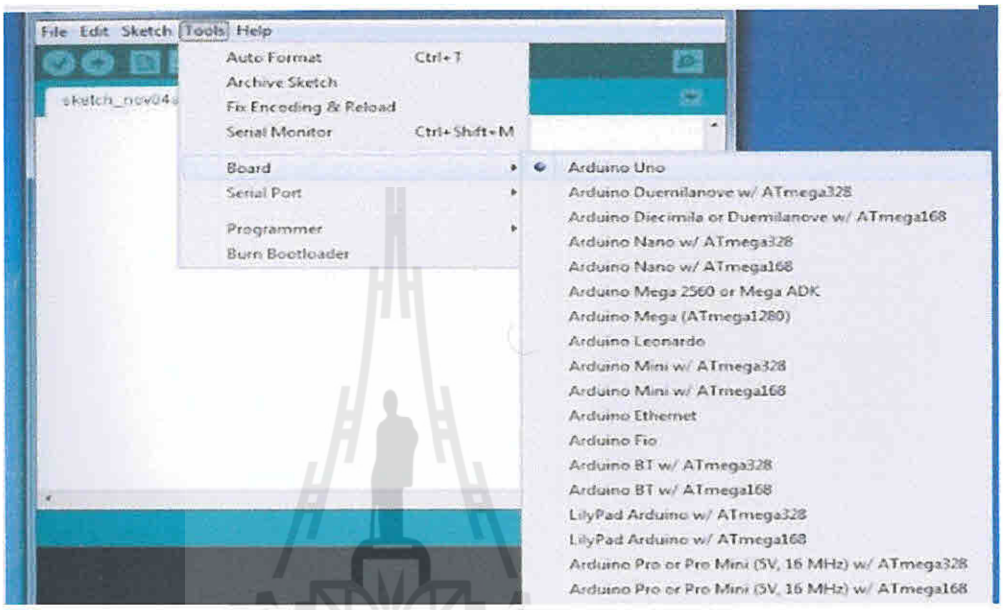
การเซตนี้ทำแค่ครั้งเดียวก็เสร็จสิ้น นอกจากผู้ใช้จะต้องการเปลี่ยนรุ่นของ Board หรือเปลี่ยน Port ที่ทำการเชื่อมต่อกับ Boardเท่านั้น เพียงแค่เลือกไปที่ Folder ที่คุณได้เก็บ Arduino IDE ไว้ และทำการ StartSoftware ของ Arduino IDE โดยทำการ Double-clickingไปที่ Arduino Application



รูปภาพที่ 2.15 Setup Arduino Software

มหาวิทยาลัยเทคโนโลยีสุรนารี

Double Click ที่ "Arduino" เพื่อ Start Arduino IDE จากนั้นเลือกรุ่นของ Arduino board ที่ทำการเชื่อมต่อให้ถูกต้องจากใน List และทำการเปลี่ยนถ้าต้องการเชคว่า Serial port ที่ต่ออยู่ได้ถูกเลือกอย่างถูกต้องหรือไม่ ให้เลือกไปที่ Computer Serial Port ที่ Arduino ทำการต่อเชื่อมอยู่เพื่อตรวจสอบว่าถูกต้องหรือไม่

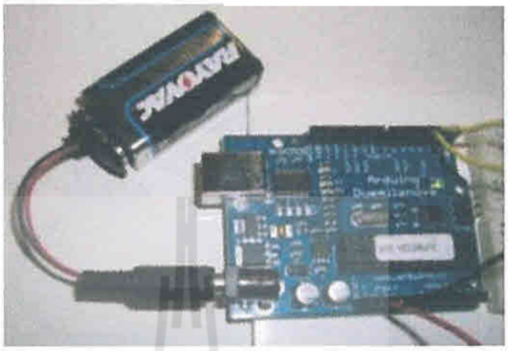


รูปภาพที่ 2.16 Setup Arduino Software

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.3.1.3 การเชื่อมต่อแบตเตอรี่

สำหรับการดำเนินงานแบบสแตนด์อโลนจะใช้พลังงานจากแบตเตอรี่มากกว่าพลังงานที่ผ่าน USB ออกการเชื่อมต่อกับคอมพิวเตอร์ ในขณะที่พอร์ตไฟฟ้าภายนอกสามารถใช้ได้ช่วง 6V ถึง 24V (ตัวอย่างเช่นคุณสามารถใช้แบตเตอรี่รถยนต์) ซึ่งเป็นมาตรฐานที่ 9V แบตเตอรี่มีความสะดวก



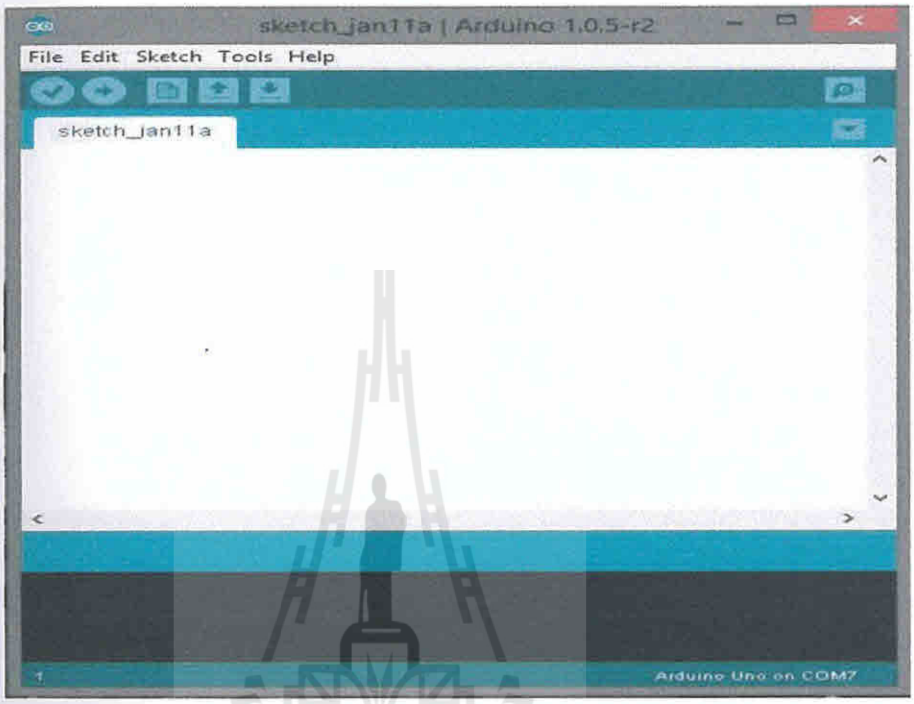
รูปภาพที่ 2.17 แสดงการเชื่อมต่อแบตเตอรี่

คำเตือน: ดูขั้วในขณะที่คุณเชื่อมต่อแบตเตอรี่ ถ้าต่อผิดขั้วกระแสอาจจะเกิดการย้อนกลับสามารถทำให้บอร์ดอาจเกิดการระเบิดได้

วิธีการคือ ทำการถอด Arduino จากคอมพิวเตอร์ และเชื่อมต่อแบตเตอรี่ 9 โวลต์กับช่องเสียบ Arduinoใช้แบตเตอรี่อะแดปเตอร์สแน็ป ตรวจยืนยันว่าโปรแกรมทำงานหรือไม่โดยดูจากไฟกระพริบที่บอร์ดซึ่งตรงนี้แสดงให้เห็นว่าเราสามารถใช้พลังงานจากแบตเตอรี่และรันโปรแกรมที่เราโหลดลงบอร์ดโดยไม่จำเป็นต้องเชื่อมต่อกับเครื่องคอมพิวเตอร์หลัก

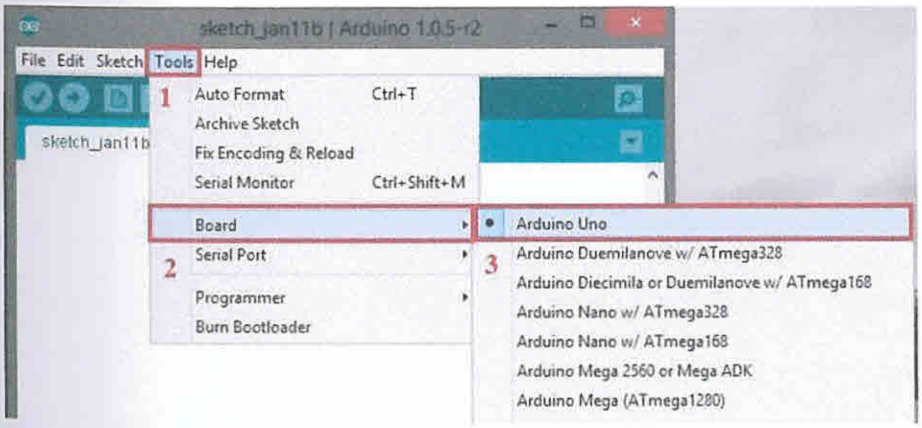
2.3.1.4 การตั้งค่าและการเริ่มต้นใช้งานโปรแกรมArduino 1.0.5

1. เชื่อมต่อบอร์ดArduino UNO R3 เข้ากับคอมของเราผ่านทางพอร์ท USB แล้วเปิดโปรแกรม Arduino 1.0.5 จะปรากฏหน้าต่างดังรูปที่ 2.18



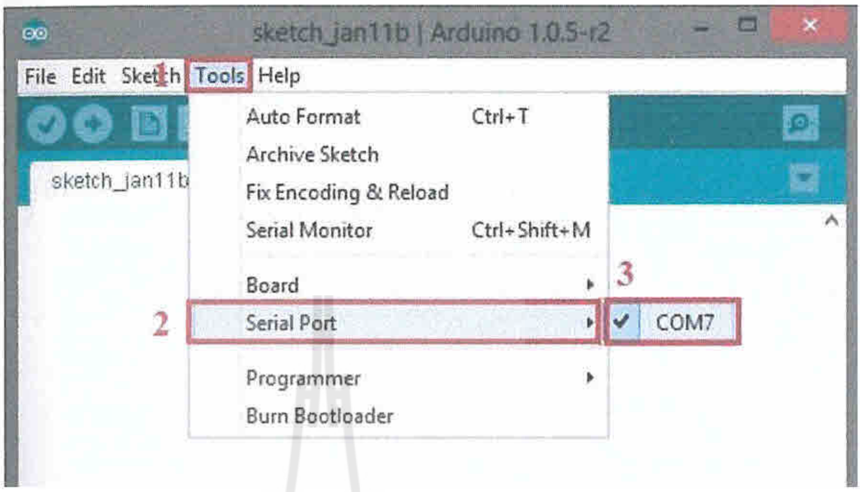
รูปที่ 2.18 หน้าต่างเริ่มต้นของโปรแกรมArduino 1.0.5

2. การตั้งค่าบอร์ดให้ไปที่ Tools >> Board >>เลือกArduino UNO กรณีใช่รุ่นอื่นก็เลือกให้ตรงกับรุ่นที่ใช้งานดังรูปที่ 2.19



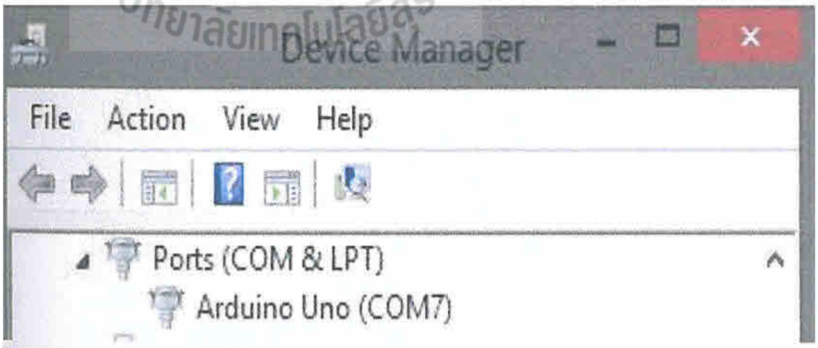
รูปที่ 2.19 วิธีตั้งค่า Board

3. การตั้งค่าพอร์ตใช้งานให้ไปที่ Tools >> Serial Port >>เลือก COM....
ดังรูปที่2.20



รูปที่ 2.20 วิธีการตั้งค่า Serial Port

หากไม่ทราบว่าบอร์ดArduino UNO R3 ใช้พอร์ทอะไรอยู่สามารถเข้าไปดูได้โดยคลิกขวาที่ My Computer เลือกProperties >>เลือกที่ Device Manager >> Ports (COM &LPT) จะพบดังรูปที่ 2.21



รูปที่ 2.21 วิธีดูพอร์ท

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.3.1.5 หน้าต่างของโปรแกรม Arduino 1.0.5

หน้าต่างของโปรแกรม Arduino 1.0.5 ดังรูปที่ 2.40ควรเรียนรู้ในขั้นพื้นฐานจะมีอยู่ทั้งหมด 9 ส่วน ดังนี้

หมายเลข 1 ปุ่ม Verify ใช้สำหรับตรวจสอบโค้ดที่เขียนเพื่อหาข้อผิดพลาด (Ctrl + R)

หมายเลข 2 ปุ่ม Upload ใช้สำหรับตรวจสอบแล้วทำการเขียนโค้ดไปยังบอร์ด Arduino (Ctrl + U)

หมายเลข 3 ปุ่ม New ใช้สำหรับสร้างไฟล์ Sketch ขึ้นมาใหม่ (Ctrl + N)

หมายเลข 4 ปุ่ม Open ใช้สำหรับเปิดไฟล์ Sketch ที่เราบันทึกเอาไว้ (Ctrl + O)

หมายเลข 5 ปุ่ม Save ใช้สำหรับบันทึกไฟล์ Sketch (Ctrl + S)

หมายเลข 6 ปุ่ม Serial Monitor ใช้สำหรับเรียกใช้ตัวสื่อสารแบบ Serial (Ctrl + Shift + M)

หมายเลข 7 ช่อง Code Program

หมายเลข 8 ช่องแสดง Status Program

หมายเลข 9 ช่องแสดง Error

รูปที่ 2.22 หน้าต่างของโปรแกรม Arduino 1.0.5

มหาวิทยาลัยเทคโนโลยีสุรนารี

เมื่อทำการตั้งค่าการใช้งานโปรแกรมArduino 1.0.5 พร้อมทั้งเรียนรู้ส่วนใช้งานหลักๆ แล้ว ต่อไปเป็นวิธีอัพโหลดโค้ดโปรแกรมลงบอร์ด Arduino UNO

1. เชื่อมต่อบอร์ดArduino UNO เข้ากับคอมผ่านทางพอร์ท USB แล้วเปิดโปรแกรมArduino 1.0.5 ขึ้นมา
2. พิมพ์โค้ดโปรแกรมคำสั่งที่ใช้งาน
3. ทำการ Verify โค้ดที่เขียนเพื่อตรวจสอบดูว่าโปรแกรมผิดพลาดหรือไม่
4. ทำการอัพโหลดโค้ดลงบอร์ด Arduino UNO R3

เมื่อทำการอัพโหลดโค้ดเสร็จแล้วจะสังเกตได้ว่า LED ที่ขา D13 กระพริบ ติด ดับ สลับกันเป็น ระยะเวลาห่างกัน 1 วินาที ไปเรื่อยๆจนกระบวนการเสร็จสิ้น LED ก็จะดับไป

2.3.2 XBEE

รูปภาพที่ 2.23 XBEE

อีกหนึ่งโมดูลสื่อสารข้อมูลอนุกรมไร้สายที่ใช้คลื่นวิทยุความถี่ 2.4 GHz สามารถสื่อสารกับ Xbee ตัวอื่นๆ ด้วยเครือข่าย ZigBee Mesh อินเตอร์เฟสผ่าน UART Serial Port (TTL Level) และเครื่องมือเพื่อสนับสนุนใช้งานโมดูล Xbee-Pro สามารถสรุปได้ดังนี้

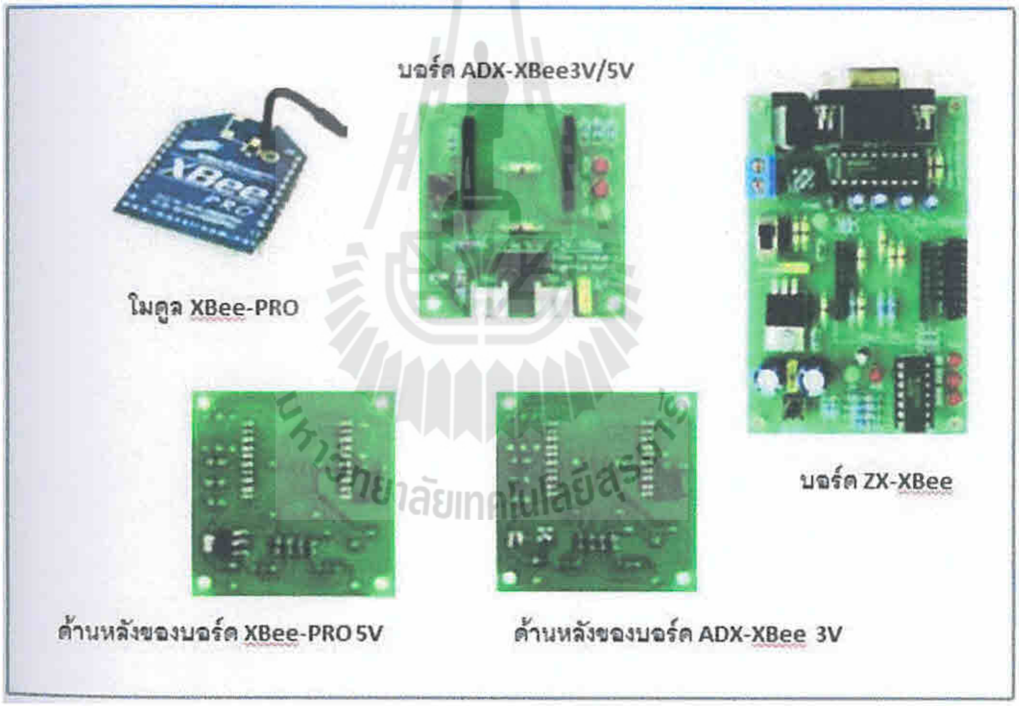
มหาวิทยาลัยเทคโนโลยีสุรนารี

1. Xbee-PRO โมดูลสื่อสารข้อมูลอนุกรมไร้สาย2.4GHz

2. ZX-Xbeeบอร์ดเชื่อมต่อโมดูล Xbee กับพอร์ตอนุกรมของคอมพิวเตอร์ ใช้ในการสื่อสารข้อมูลและตั้งค่าเพื่อใช้งานโมดูล Xbee สามารถรองรับการทำงานทั้งโมดูล Xbeeและ Xbee-PRO

3. ADX-Xbee 5V บอร์ดใช้งานโมดูล Xbee กับอุปกรณ์ที่ใช้ไฟเลี้ยง +5v

4. ADX-Xbee3 V บอร์ดใช้งานโมดูล Xbee กับอุปกรณ์ที่ใช้ไฟเลี้ยง +3.3V



รูปภาพที่ 2.24 แสดงหน้าตาของโมดูล Xbee-Pro และบอร์ดสำหรับต่อใช้งาน

รูปที่ 2.25 แสดงส่วนประกอบของชุดอุปกรณ์ทั้งหมดในชุด Xbee 3V/5V

2.3.2.1 XBee-PRO

2.3.2.1.1 คุณสมบัติทางเทคนิค

- ความถี่ในการทำงาน 2.4GHz
- มีสายอากาศแบบ Whip
- ระยะการทำงานในร่มสูงสุด 300 ฟุต หรือประมาณ 100เมตร
- ระยะการทำงานกลางแจ้ง (line-of-sight) สูงสุดถึง1,600 เมตร
- กำลังในการส่ง 60mW(18dBm)
- ความไวในการรับสัญญาณอยู่ที่ -100dBm(1% packet error rate)
- การทำงานของขาพอร์ตสามารถกำหนดผ่านทางซอฟต์แวร์ X-CTU เพื่อให้ทำงานเป็น

    -อินพุตอนาล็อกสำหรับวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลความละเอียด 10 บิต

    -อินพุตเอาท์พุตดิจิตอล

มหาวิทยาลัยเทคโนโลยีสุรนารี

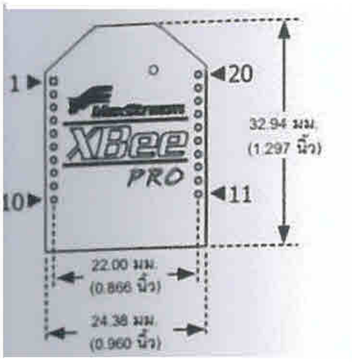
- มีขนาด 0.96×1.297 นิ้ว หรือ 2.438×3.294 เซนติเมตร
- ไฟเลี้ยง 2.8 ถึง 3.4V
- กระแสไฟฟ้า : เมื่อส่งข้อมูล 215mA,รับข้อมูล 55mA และจะน้อยกว่า 10μA ในโหมดลดพลังงานที่ไฟเลี้ยง +3.3V
- อุณหภูมิใช้งานระหว่าง -40 ถึง 85 องศาเซลเซียส

2.3.2.1.2คุณสมบัติด้านการสื่อสารข้อมูล

- สามารถทำงานเป็นอุปกรณ์มาสเตอร์และสเลฟได้
- อัตราการส่งข้อมูลผ่านคลื่นวิทยุ : 250,000บิตต่อวินาที
- อัตราการถ่ายทอดข้อมูลอนุกรม(บอทเรต): 1,200ถึง115,200บิตต่อวินาที
- รูปแบบโครงข่ายข้อมูลที่รองรับ : จุดต่อจุด(Point-to-Point),จุดต่อหลายจัด(Point-to-Multipoint)และเข้ากันได้กับอุปกรณ์ตามมาตรฐาน
- ทางเลือกแอดเดรส : PAN ID,ช่อง(Channel),แอดเดรส (Addresses) สำหรับแอดเดรสสามารถกำหนดรหัสแอดเดรสได้มากถึง 65,000รหัส
- เทคโนโลยีในการกระจายคลื่น : DSSS (Direct Sequence Spread Spectrum)
- รองรับการทำงานทั้งแบบ API และ AT Command สามารถกำหนดได้ผ่านทางซอร์ฟแวร์ X-CTU
- การรองรับมาตรฐาน : สหรัฐอเมริกา (FCC) OUR-XBEEPRO
  : แคนาดา (IC) 42141A XBEEPRO
  : ยุโรป (CE) กำลังส่งสูงสุด 10dBm
  : ญี่ปุ่น กำลังส่งสูงสุด 10dBm

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.3.2.2 การจัดขา PIN

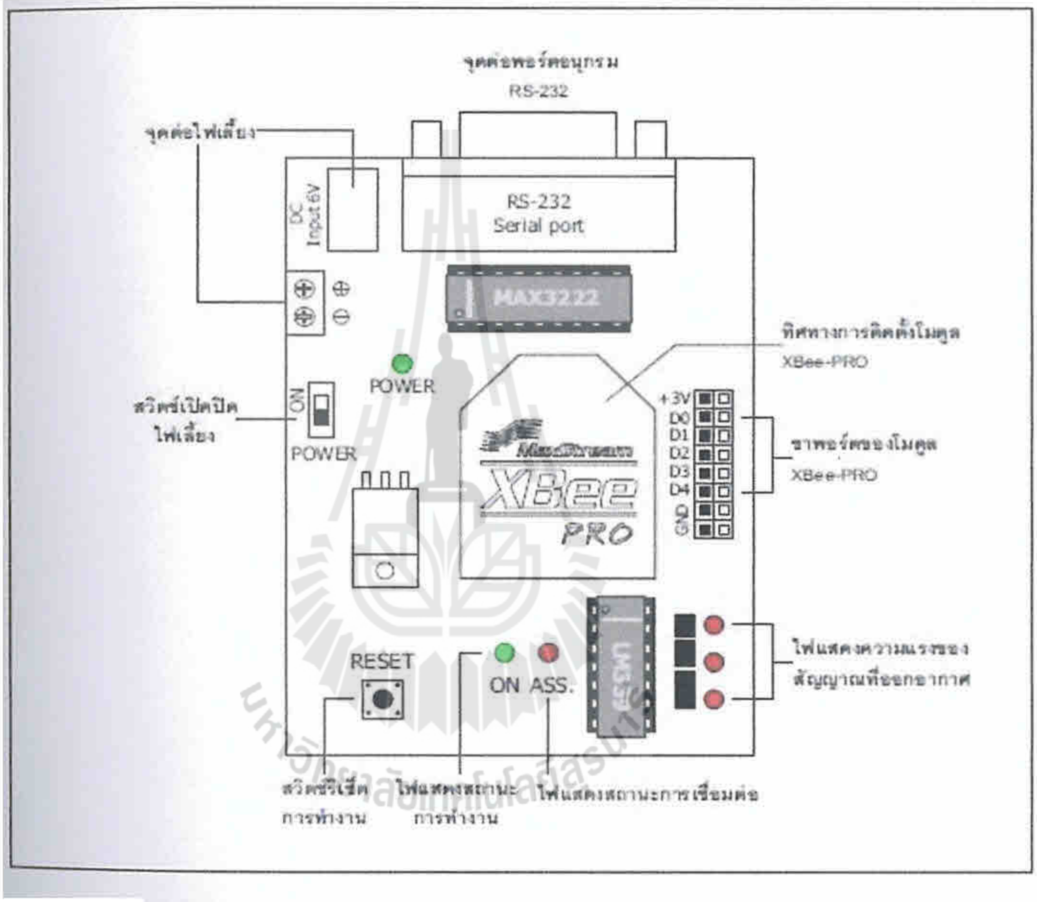


ขาที่	ชื่อขา/การทำงาน
1	Vcc : ขาต่อไฟเลี้ยง +3.3V
2	DOUT : ขาเอาต์พุตส่งข้อมูลอนุกรม
3	DIN : ขาอินพุตรับข้อมูลอนุกรม
4	DO8 : ขาเอาต์พุตดิจิตอล ช่อง 8
5	RESET : ขารีเซตหลัก (แอกตีฟ "0")
6	PWM0/RSSI : ขาเอาต์พุต PWM ช่อง 0 และขาเอาต์พุตแสดงความแรงของการรับสัญญาณ
7	PWM1 : ขาเอาต์พุต PWM ช่อง 1
8	ไม่ใช้งาน
9	DTR/SLEEP_RQ/DI8 : ขาอินพุตรับสัญญาณให้หยุดทำงานเข้าสู่โหมดสลีป หรือเป็นขาอินพุตดิจิตอลช่อง 8
10	GND : ขาต่อกราวด์
11	AD4/DIO4 : ขาอินพุตอะนาลอก 4 หรือ ขาอินพุตเอาต์พุตดิจิตอล 4
12	CTS/DIO7 : อินพุตรับสัญญาณแจ้งการส่งข้อมูลจากโฮสต์ (Clear-To-Send) ใช้ในการควบคุมจังหวะการรับส่งข้อมูล หรือเป็น ขาอินพุตเอาต์พุตดิจิตอล 7
13	ON/SLEEP : ขาแสดงสถานะการทำงาน "1" : อยู่ในโหมดทำงานปกติ "0" : อยู่ในโหมดสลีป
14	VREF : ขาต่อแรงดันอ้างอิงสำหรับโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอล ภายใน XBee-PRO
15	Associated/AD5/DIO5 : ขาแสดงสถานะการเชื่อมต่อ หรือ ขาอินพุตอะนาลอก 5 หรือ ขาอินพุตเอาต์พุตดิจิตอล 5
16	RTS/AD6/DIO6 : ขาเอาต์พุตแจ้งความพร้อมในการส่งข้อมูล (Ready-To-Send) ใช้ควบคุมจังหวะการรับส่งข้อมูล หรือเป็นขาอินพุตอะนาลอก 6 หรือเป็นขาอินพุตเอาต์พุตดิจิตอล 6
17	AD3/DIO3 : ขาอินพุตอะนาลอก 3 หรือ ขาอินพุตเอาต์พุตดิจิตอล 3
18	AD2/DIO2 : ขาอินพุตอะนาลอก 2 หรือ ขาอินพุตเอาต์พุตดิจิตอล 2
19	AD1/DIO1 : ขาอินพุตอะนาลอก 1 หรือ ขาอินพุตเอาต์พุตดิจิตอล 1
20	AD0/DIO0 : ขาอินพุตอะนาลอก 0 หรือ ขาอินพุตเอาต์พุตดิจิตอล 0

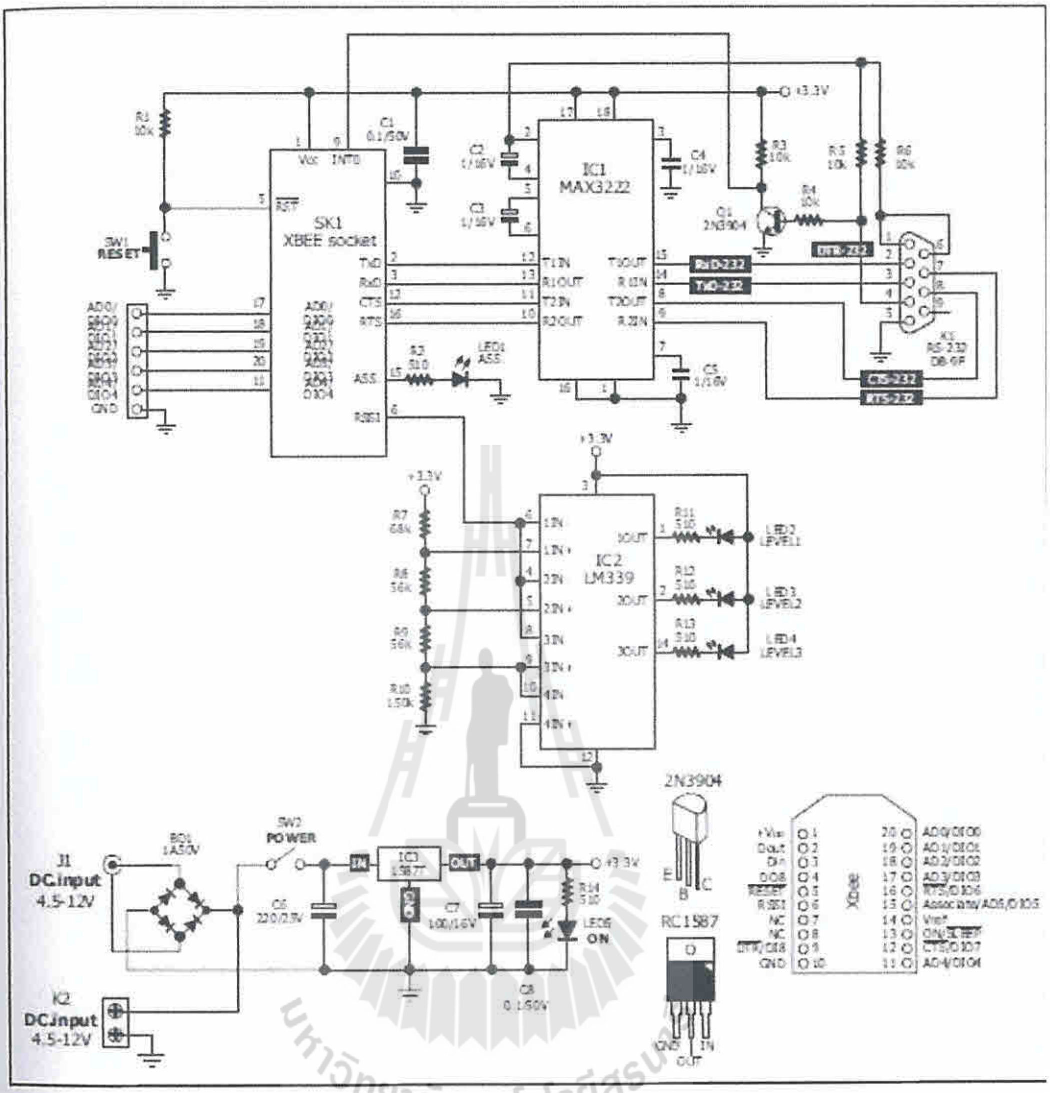
(ตารางที่ 2.1 การจัดขา PIN)

2.3.2.3 ZX-Xbeeบอร์ดเชื่อมต่อโมดูล Xbee-PRO กับคอมพิวเตอร์

เป็นบอร์ดสำหรับติดตั้งโมดูล Xbee-PRO เพื่อเชื่อต่อกับคอมพิวเตอร์สำหรับตั้งค่าพารามิเตอร์ต่างๆ ดังรูปที่ 3 แสดงรายละเอียดของบอร์ด ZX-Xbee และการติดตั้งโมดูล Xbee-PRO ลงบนบอร์ด สำหรับวงจรของบอร์ด ZX-Xbee แสดงในรูปที่ 2.45



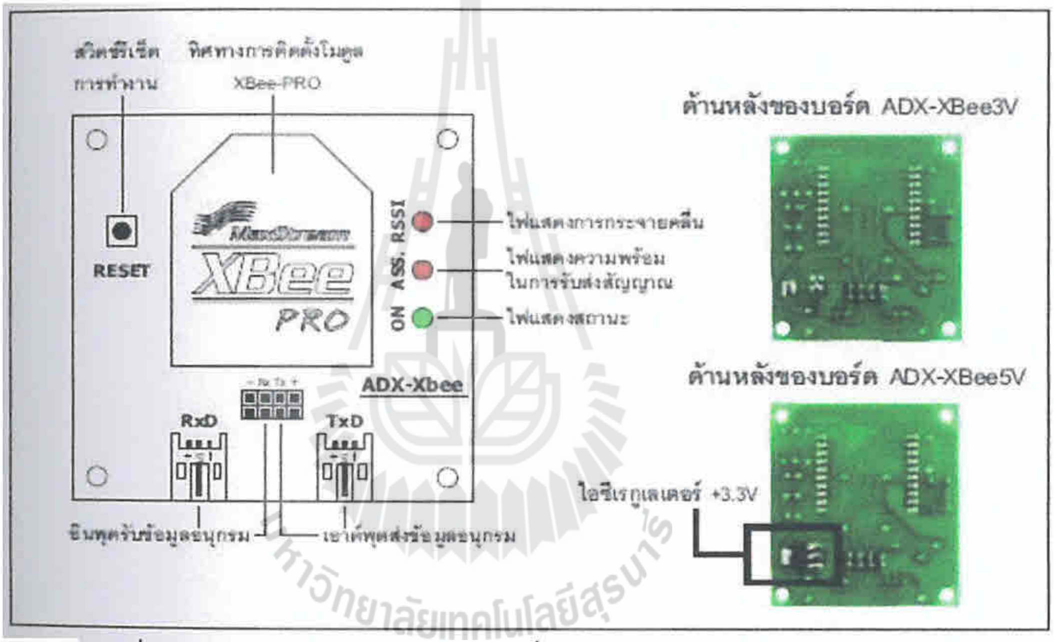
รูปภาพที่ 2.26 ZX-Xbeeบอร์ดเชื่อมต่อโมดูล Xbee-PRO กับคอมพิวเตอร์



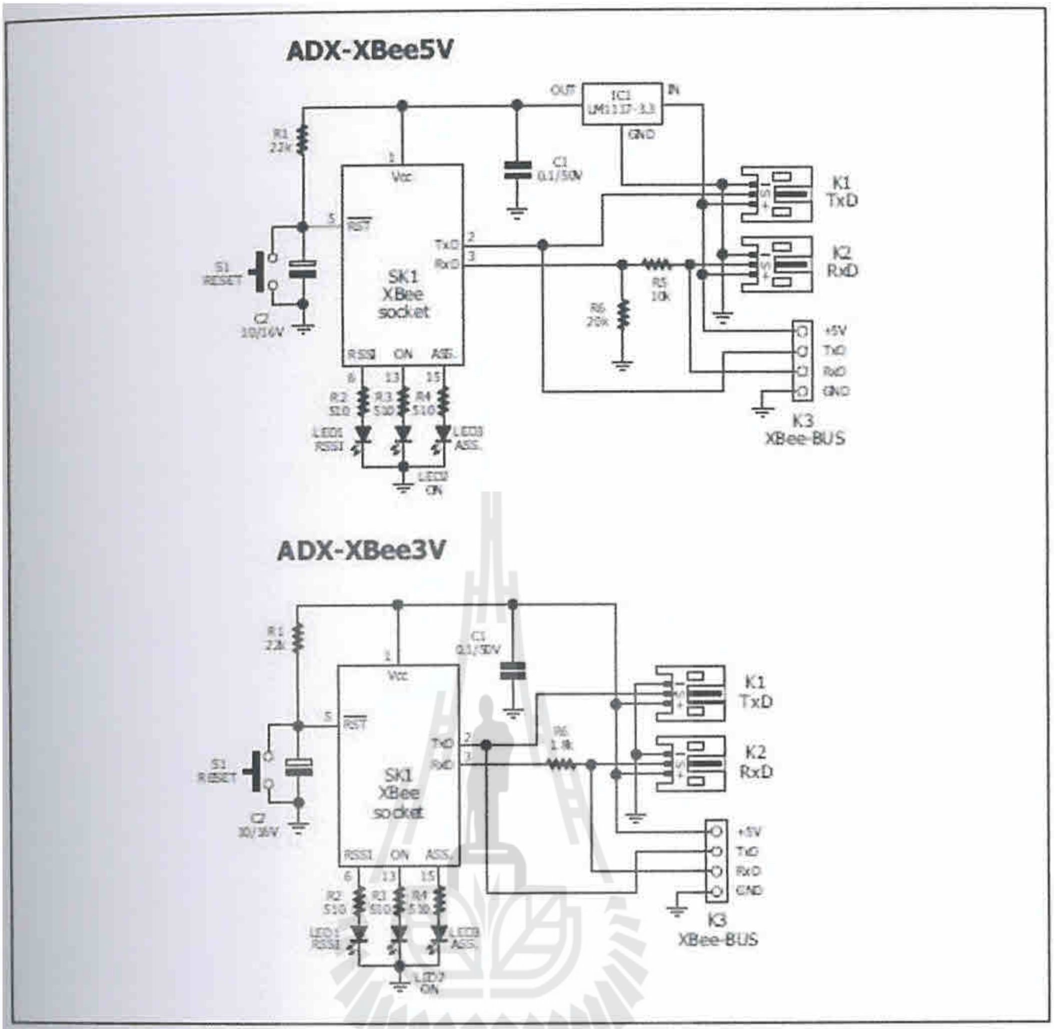
รูปที่ 2.27 วงจรของ ZX-bee บอร์ดเชื่อมต่อโมดูล Xbee/Xbee-PRO กับคอมพิวเตอร์

2.3.2.4 ADX-Xbeeบอร์ดเชื่อมต่อโมดูล Xbee-PRO กับอุปกรณ์ภายนอก

เป็นบอร์ดสำหรับติดตั้งโมดูล Xbee-PRO เพื่อนเชื่อมต่อกับอุปกรณ์ภายนอก เช่น ไมโครคอลโทรลเลอร์ หรือ FPGA เพื่อนช่วยให้ไมโครคอนโทรลเลอร์และ FPGA มีระบบสื่อสารข้อมูลอนุกรมไร้สาย 2.4GHz ในรูปที่ 4แสดงรายละเอียกของบอร์ด ADX-Xbeeและการติดตั้งโมดูล Xbeeลงบนบอร์ด ส่วนวงจรของบอร์ด ADX-Xbee ทั้ง 2 รุ่น คือ รุ่น 5V และ รุ่น 3V แสดงในรูปที่ 2.47



รูปที่ 2.28 รายละเอียด ADX-bee บอร์ดเชื่อมต่อโมดูล Xbee-PRO กับอุปกรณ์ภายนอก



รูปภาพที่ 2.29 แสดงวงจรของบอร์ด ADX-bee ทั้ง 2 รุ่นคือ สำหรับADX-Xbee 5V,3V

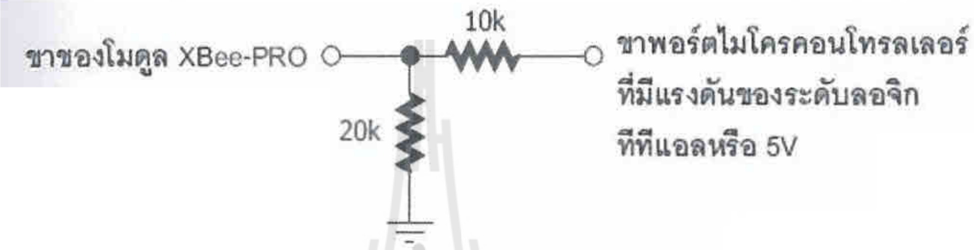
2.3.2.5 การใช้โมดูล Xbee-PRO เบื้องต้น

เมื่อเราจัดการซื้อโมดูล Xbee-PRO มาแล้ว เราสามารถใช้งานได้ทันที โดยไม่ต้องมีการตั้งค่าใดๆ ทั้งสิ้น โดยค่าตั้งต้นที่มาจากโรงงานผู้ผลิตคือ อัตราบอดเรตเป็น 9,600 บิตต่อวินาที 8 บิตข้อมูล ไม่มีการตรวจสอบพาริตี้ และ 1 บิต หยุด หรือ 9600 8N1 นั่นเอง

การต่อใช้งานก็ง่ายๆ เพียงแค่ต่อขา TxDของ XBee-PRO เข้ากับขา RxDของไมโครคอนโทรลเลอร์หรือคอมพิวเตอร์ และต่อขา RxDของ XBee-PRO เข้ากับขา TxD

ของไมโครคอลโทรลเลอร์หรือคอมพิวเตอร์ จ่ายไฟเลี้ยง +3V และต่อกราวด์ ก็สามารถใช้งานได้แล้ว

แต่เนื่องจาก Xbee-PRO ต้องการไฟเลี่ยงในย่าย2.8 ถึง 3.4V และขาสัญญาณั้วหมดทำงานในระบบบัส 3Vดังนั้นหากนำไปเชื่อมต่อกับไมโครคอนโทรลเลอร์หรืออุปกรณ์ภายนอกที่ใช้ระบบบัส 5V จะต้องมีการลดแรงดันที่ขาพอร์ตลงด้วย



รูปภาพที่ 2.30 ขาของ Module

2.3.2.6 การตั้งค่าคอนฟิกูเรชั่นให้กับโมดูล Xbee-PRO

จะต้องใช่บอร์ด ZX-Xbeeและซอฟต์แวร์ X-CTU มาช่วยในการกำหนดค่าคอนฟิกูเรชั่น หรือค่ากำหนดการทำงานทางฮาร์ดแวร์

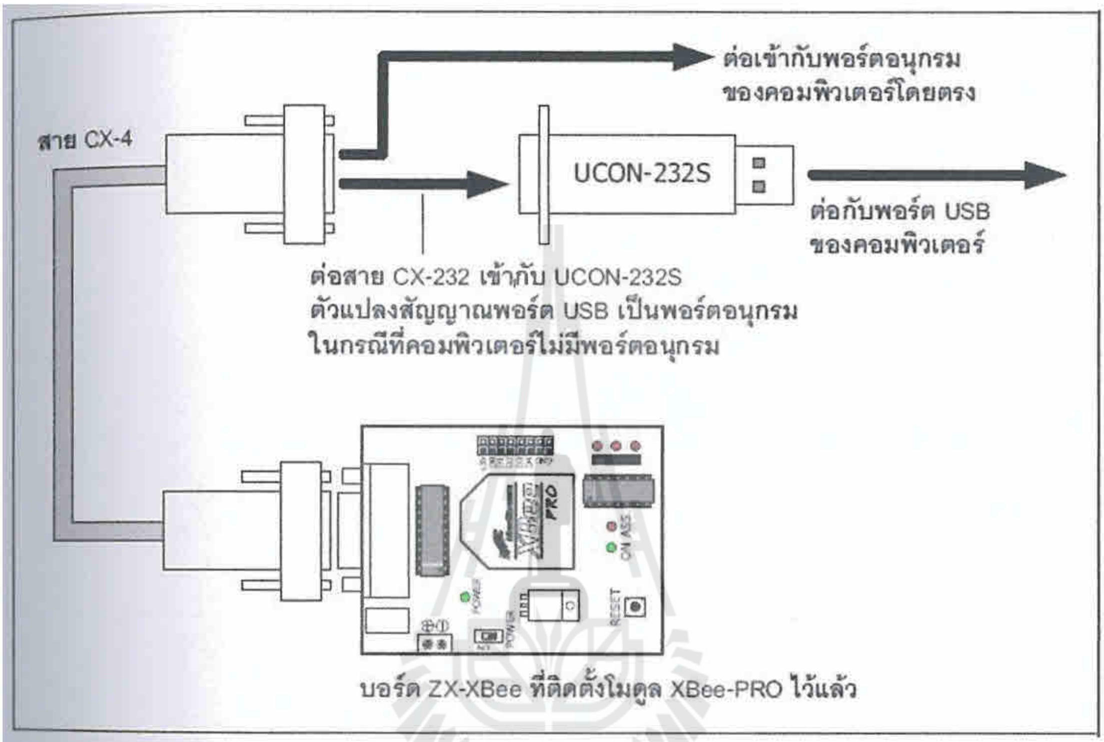
2.3.2.6.1 ความต้องการของระบบ

สำหรับคอมพิวเตอร์ที่จะนำมาใช้ในการกำหนดค่าคอนฟิกูเรชั่นของ Xbee-PRO นั้น ควรมีคุณสมบัติดังนี้

- ซีพียู ความเร็ว 500MHz ขึ้นไป
- หน่วยความจำแรม 256 MB
- ฮาร์ดดิสก์มีเนื้อที่ว่าง 100 MB เป็นอย่างน้อย
- ติดตั้งระบบปฏิบัติการ Windows XP Service Pack 2

มหาวิทยาลัยเทคโนโลยีสุรนารี

- มีพอร์ตอนุกรมว่างอย่างน้อย 1 พอร์ตถ้ามีพอร์ตเฉพาะ USB เป็นพอร์ตอนุกรม RS-232 แบบที่ให้สายสัญญาณครบทั้ง 9 เส้น แนะนำรุ่น UCON-232S ของ INEX ซึ่งหากตัวแปลง UCON-232S จะต้องทำการติดตั้งไดรเวอร์ให้เรียบร้อยเสียก่อน

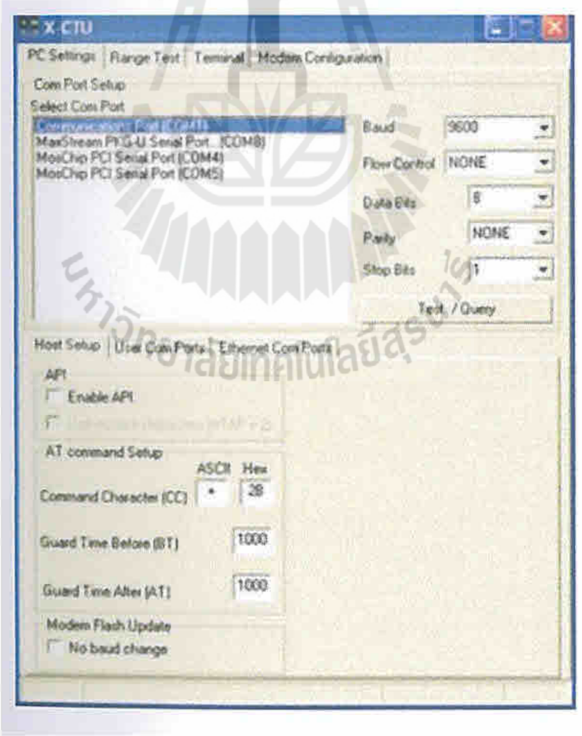


รูปภาพที่ 2.31 แสดงการเชื่อมต่อบอร์ด ZX-Xbee ในกรณีที่คอมพิวเตอร์ไม่มีพอร์ตอนุกรม

2.3.2.6.2 ขั้นตอน

1. ติดตั้งโปรแกรม X-CTU เวอร์ชั่นล่าสุด ดาวน์โหลดได้ฟรีจากเว็บไซต์ของผู้ผลิตที่ www.maxstrewam.net ซึ่งการติดตั้งเหมือนกับการติดตั้งโปรแกรมประยุกต์ทั่วไป
2. ติดตั้งโมดูล Xbee-PRO ลงบนบอร์ด ZX-Xbee232 ต้องระวังเรื่องตำแหน่งขาและทิศทางของโมดูลต้องติดตั้งให้ถูกต้อง และไม่เกิดการเหลื่อมกันเด็ดขาด เพราะหากติดตั้งผิดแล้ว เมื่อจ่ายไฟจะทำให้โมดูลเสียหายทันที

3. ต่อสายเชื่อมต่อพอร์ตอนุกรมระหว่างบอร์ด ZX-Xbee กับพอร์ตอนุกรมของคอมพิวเตอร์หรือตัวแปลง USB เป็นพอร์ตอนุกรม
4. จ่ายไฟให้แก่บอร์ด สำหรับไฟเลี้ยงที่เหมาะสมอยู่ในช่วง +6 ถึง +12V กระแสไม่น้อยกว่า 300mA แนะนำให้ใช้ +6V จะเห็น LED ในตำแหน่ง Power และ ON ติด และLED ในตำแหน่ง ASS.กระพริบ หากไม่เป็นไปตามนี้ ให้รีบปิดสวิตท์ ปลดไฟเลี้ยง แล้วตรวจสอบการติดตั้งโมดูล Xbee-PRO ทันที รวมทั้งตรวจสอบไฟเลี้ยงที่ขา Vcc ของ Xbee-PRO ว่าต้องอยู่ในช่วย +2.2 ถึง +3.3V โดยในการตรวจสอบนั้นต้องถอดโมดูล XBee-PRO ออกมาก่อน แล้ววัดแรงดันที่คอนเนคเตอร์ที่ใช้สำหรับติดตั้งโมดูล
5. เปิดโปรแกรม X-CTU หน้าต่างการเชื่อมต่อจะปรากฎขึ้นมาดัง รูปภาพที่ 2.32



รูปภาพที่ 2.32 X-CTU

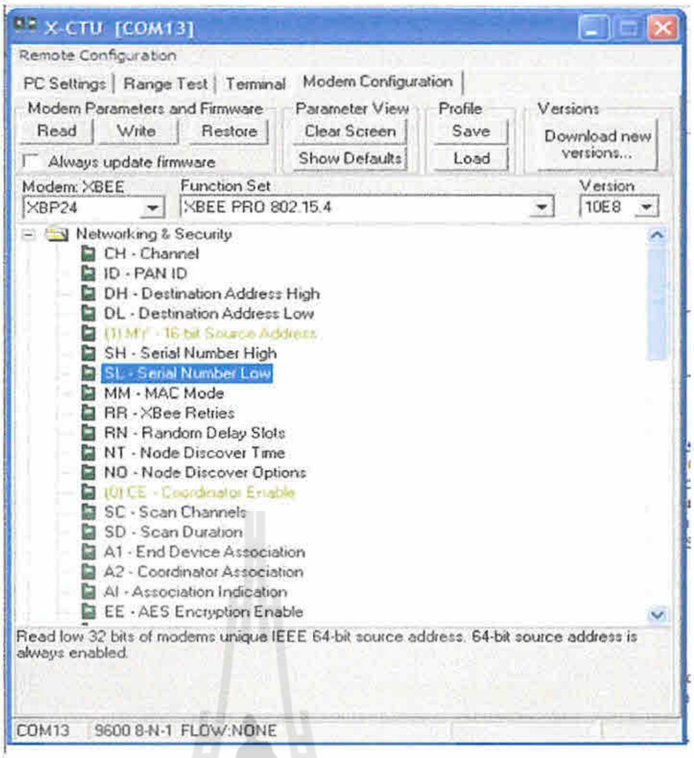
ให้ทำการเลือกพอร์ตที่ทำการเชื่อมต่อ เลือกอัตราBaudrateเป็น 9,600, Data 8บิต, Parity ไม่มีการตรวจสอบ และ Stop เป็น 1 บิต

6. กดปุ่ม Test เพื่อทดสอบการติดต่อระหว่าง XBee-PRO กับโปรแกรม X-CTU หากติดต่อกันได้จะปรากฏหน้าต่างแจ้งผลการติดต่อและข้อมูลทางฮาร์ดแวร์เบื้องต้นของโมดูลXBee-PRO ดังรูปภาพที่ 2.33

รูปภาพที่ 2.33 Com test/ Query Modem

หากมีการแจ้งความผิดพลาดใดๆเกิดขึ้นในขั้นตอนนี้ให้รีบปิดสวิตซ์ ปลดไฟเลี้ยง แล้วตรวจสอบการติดตั้งโมดูล XBee-PRO และการเชื่อมต่ออีกครั้ง รวมทั้งตำแหน่งขาพอร์ตด้วย และถ้าจำเป็นอาจต้องทดลองเลือกอัตราบอดเรทใหม่

7. หลังจากนั้นคลิกไปที่แท็ปModern Configuration แล้วกดปุ่ม Read ในกรอบ Modem Parameters and Firmware จะปรากฏข้อมูลชื่อรุ่นของโมดูล XBee-PRO ชื่อฟังก์ชั่น หมายเลขเวอร์ชั่นของเฟิร์มแวร์และค่าพารามิเตอร์ต่างๆ



รูปภาพที่ 2.34 X-CTU [COM13]

8. เมื่อมาถึงขั้นนี้ ผู้ใช้งานสามารถเปลี่ยนค่าคอนฟิกูเรชั่นได้ตามต้องการ ไม่ว่าจะเป็นการเลือกอัตราบอดเรตใหม่ กำหนดรูปแบบการทำงานของขาพอร์ตของ Xbee-PRO เนื่องจากขาพอร์ตของ Xbee-PRO สามารถใช้งานเป็นอินพุตอนาล็อกเพื่อนเชื่อมต่อกับตัวตรวจจับที่ให้ผลการทำงานเป็นแรงดัน ส่งไปยังวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลความละเอียด 10 บิตภายในโมดูล Xbee-Pro ซึ่งผู้ใช้งานสามารถอ่านค่าออกไปได้ ดังนั้นจึงสามารถใช้งาน Xbee-PRO ร่วมกับตัวตรวจจับทำงานเป็นโครงข่ายตัวตรวจจับไร้สาย (Wireless Sensor Network) ได้

2.3.2.7 รีจิสเตอร์ที่ควรทราบของโมดูล XBee-PRO

การส่งข้อมูลและควบคุมระหว่างไมโครคอนโทรลเลอร์หรือคอมพิวเตอร์กับโมดูล Xbee-PRO นั้น จะใช้การสื่อสารแบบอนุกรม UART ซึ่งโมดูล XBee-PRO สามารถใช้ความเร็วในการส่งข้อมูล (Baud rate) ได้ตั้งแต่ 1,200 จนถึง 115,200 บิตต่อวินาที (bit per second) โดยค่าที่กำหนดเป็นค่าเริ่มต้นคือ 9,600 บิตต่อวินาที และสามารถเปลี่ยนความเร็วในการส่งข้อมูลที่ได้รีจิสเตอร์ BD

ในการติดต่อสื่อสารระหว่างโมดูลXBee-Pro สามารถจัดเครือข่ายได้หลายรูปแบบ โดยการแยกช่องสัญญาณและเครือข่าย รีจิสเตอร์ที่ใช้สำหรับจัดการเกี่ยวกับเครือข่าย มีดังนี้

2.3.2.7.1 รีจิสเตอร์ CH(Channel) ใช้กำหนดช่องสัญญาณ

เลือกได้ตั้งแต่ช่องที่0*0C ถึง 0*17 แต่ละช่องไม่สามารถ ส่งข้อมูลข้ามช่องสัญญาณกันได้

2.3.2.7.2 รีจิสเตอร์ ID (PAN ID / Personal Area Network ID)ใช้กำหนดหมายเลขเครือข่าย

เลือกค่าได้ตั้งแต่ 0*0000 จนถึง 0*FFFE โดยแต่ละเครือข่ายจะไม่สามารถส่งข้อมูลข้ามเครือข่ายกันได้ ยกเว้นถ้ากำหนดด้วยค่า 0*FFFF จะสามารถส่งข้อมูลไปทุกเครือข่ายได้ แต่จะไม่สามารถรับข้อมูลจากเครือข่ายอื่นได้

2.3.2.7.3 รีจิสเตอร์ MY (16-bit Source Address) ใช้กำหนดแอดเดรส 16 บิต

ของแต่ละโมดูล เลือกค่าได้ตั้งแต่ 0*0000 ถึง 0*FFFD และสามารถยกเลิกแอดเดรส 16 บิตนี้เพื่อไปใช้แอดเดรสขนาด 64 บิตที่รีจิสเตอร์ SH และ SL แทนได้ เพื่อขยายให้จำนวนโมดูลลูกข่ายได้มากขึ้น โดยกำหนด MY เป็น 0*FFFE และ 0*FFFF

2.3.2.7.4 รีจิสเตอร์ SH และ SL (Serial Number High / Low) เป็นรีจิสเตอร์เก็บค่าหมายเลข

เฉพาะหรือ Serial Number ของแต่ละ โมดูล สามารถใช้เป็นแอดเดรส 64 บิต (SH รวมกับ SL) โดยต้องยกเลิกแอดเดรส 16 บิตที่รีจิสเตอร์ MY ก่อน ค่าในรีจิสเตอร์ SH และ SL ไม่สามารถเปลี่ยนแปลงได้

2.3.2.7-5 รีจิสเตอร์ DH และDL (Destination Address High / Low) ใช้กำหนดแอดเดรสของโมดูลตัวรับ

7-5.1ถ้าโมดูลตัวรับใช้รับจิสเตอร์ MY (แอดเดรส 16 บิต) ให้กำหนดค่ารีจิสเตอร์ DH เป็น 0*0000 และ เป็นค่า MY ของโมดูลตัวรับ

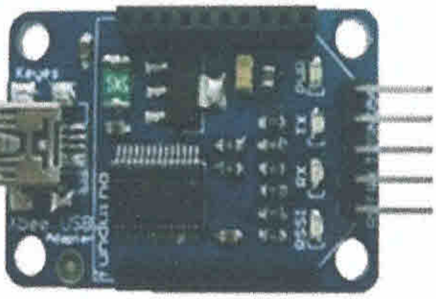
7-5.2ถ้าโมดูลตัวรับใช้รีจิสเตอร์ SH ร่วมกับ SL (แอดเดรส 64 บิต) ให้กำหนดค่าของรีจีสเตอร์ DH เป็นค่าของ SH และค่าของรีจิสเตอร์ DL เป็นค่าของ SL ของโมดูลตัวรับ

การตั้งค่าของโมดูลXBee-PRO ทำได้ 2 ทางคือ ใช้โปรแกรม X-CTU กับบอร์ด ZX-XBeeต่อกับคอมพิวเตอร์ทางพอร์ตอนุกรม อีกทางหนึ่งคือใช้ AT Command ซึ่งสามารถดูคำสั่งต่างๆ เพิ่มเติมได้จากไฟล์ในแผ่นซีดีรอม

ก่อนการใช้โมดูล XBee-PRO ควรจะทำการตรวจสอบตั้งค่าแอดเดรสต่างๆและรูปแบบของการส่งข้อมูลก่อนนำไปติดตั้งกับบอร์ดไมโครคอนโทรลเลอร์

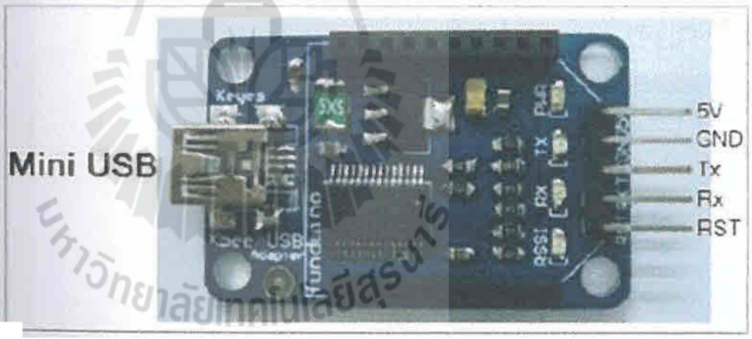
มหาวิทยาลัยเทคโนโลยีสุรนารี

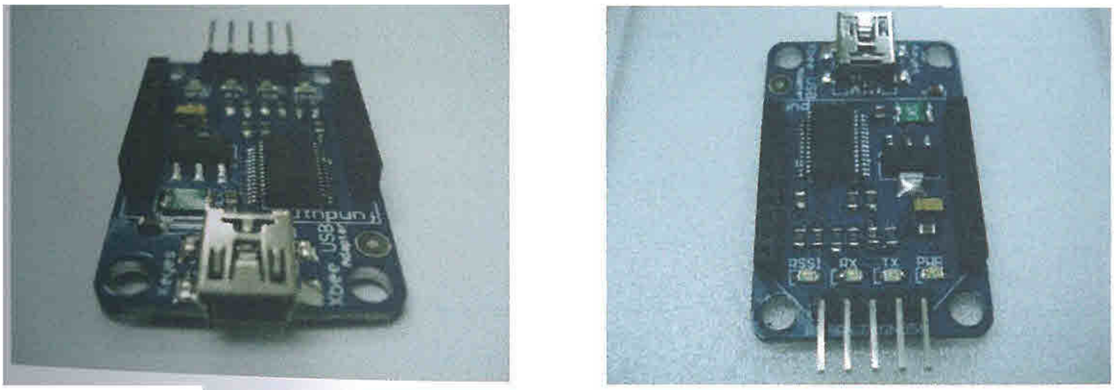
2.3.2.8 XBee Interface Board



รูปภาพที่ 2.35 XBee Interface Board

คือบอร์ดเชื่อมต่อ XBEE กับคอมพิวเตอร์ เพื่อแก้ไขค่าต่างๆ หรืออัพเดท Firmware ด้วยโปรแกรม X-CTU เพียงแค่ต่อโมดูล XBeeเข้ากับบอร์ดก็สามารถเชื่อมต่อกับคอมพิวเตอร์ด้วยสาย USB ได้ทันที และสามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์หรืออุปกรณ์อื่นๆ ผ่าน UART Serial Port ได้เช่นกัน





รูปภาพที่ 2.36 ส่วนประกอบXbee Interface Board

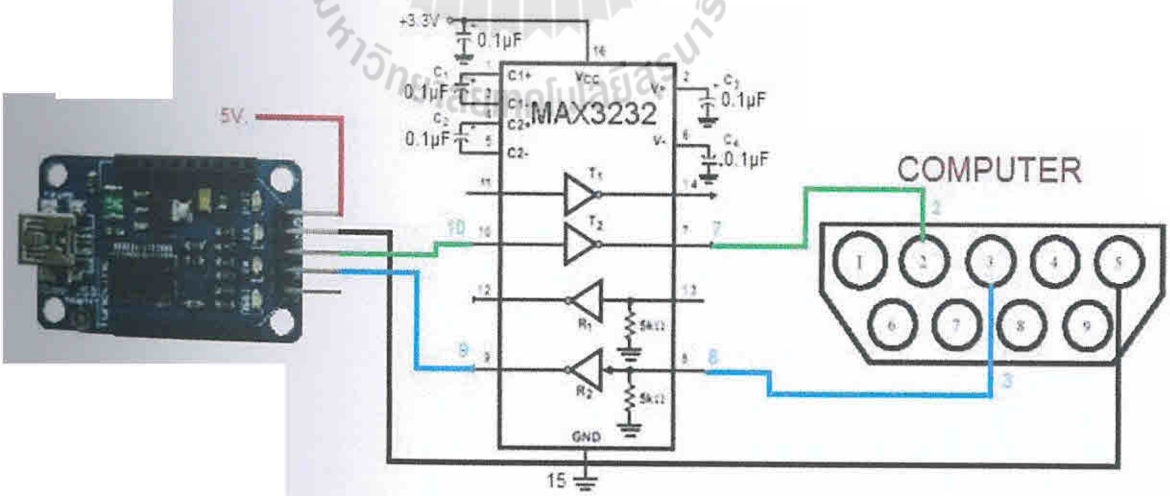
ซึ่งอุปกรณ์ตัวนี้สามารถเชื่อมต่อโดยใช้สาย USB (Mini USB) หรือจะใช้การเชื่อมต่อแบบ UART TTL Serial Port ก็ได้ทั้ง 2 วิธี

วิธีที่ 1 : เชื่อมต่อด้วย USB

รูปภาพที่ 2.37 การเชื่อมต่อ USB

ใช้สาย USB ต่อบอร์ดเข้ากับคอมพิวเตอร์ได้โดยตรง แล้วใช้งานได้เลย

วิธีที่ 2 : การเชื่อมต่อแบบ TTL Serial



รูปภาพที่ 2.38 การเชื่อมต่อแบบ TTL Serial

การต่อโดยใช้ TTL Serial Interface ต้องผ่านวงจร MAX3232 และต่อเข้าหัว DB9 ของคอมพิวเตอร์ ดังรูปที่ 2.38

2.3.2.8.1 คุณสมบัติ XBee Interface Board

1.ใช้ไฟ 5 V.

2.มีพอร์ท Mini USB และ 5 pin สำหรับเชื่อมต่อแบบ UART

3.ใช้ได้กับ XBeeทุกรุ่น (ที่ไม่ใช่แบบ Surface Mount)

2.3.2.8.2 อุปกรณ์และคำอธิบาย

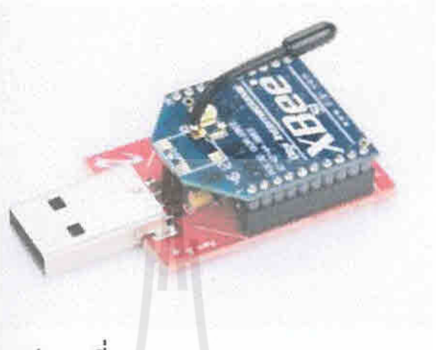
รูปภาพที่ 2.39 XBEE Interface board

คู่ Xbee อินเตอร์เฟซที่ได้รับการออกแบบมาเพื่อความสะดวกสบายโดยที่เราไม่ต้องใช้ USB ใดๆ อีกต่อไป เพื่อเชื่อมต่อแบบอนุกรม เมื่อทำการ FTDI ชิฟอินเตอร์เฟสอนุกรม USB จะถูกสร้างขึ้นแล้วบนบอร์ดนี้ ดังนั้นสิ่งที่เราต้องทำก็คือ เพิ่มโมดูล xbeeและปรับแต่งการเชื่อมโยงและเริ่มต้นการเชื่อมต่อ

อินเตอร์เฟส USB จะใช้ในการกำหนดค่าแรกของทั้งโมดูล Xbee ผ่านการตั้งค่า Xbee ซอร์ฟแวร์ X-CTU และหลังจากที่โมดูลทั้งสองได้รับการกำหนดค่าแล้ว มันจึงพร้อมที่จะเริ่มต้นการเชื่อมต่อไร้สายกับคอมพิวเตอร์หลัก

2.3.2.8.3 การกำหนดค่า

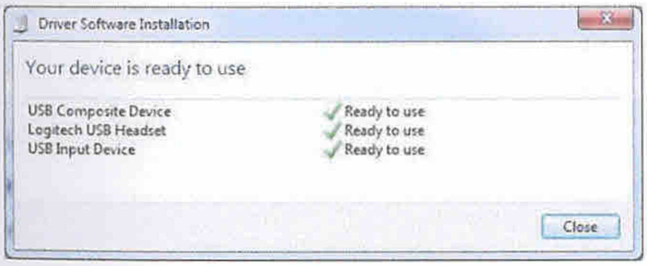
ใช้เชื่อมกับกระดานอินเตอร์เฟส โดยต่อ Xbee หรือ Xbee-PRO เข้าด้วยกัน ดังเช่นภาพด้านล่าง ก่อนจะเสียบเข้ากับคอมพิวเตอร์



รูปภาพที่ 2.40 XBEE Interface board

2.3.2.8 .4 การตั้งค่า

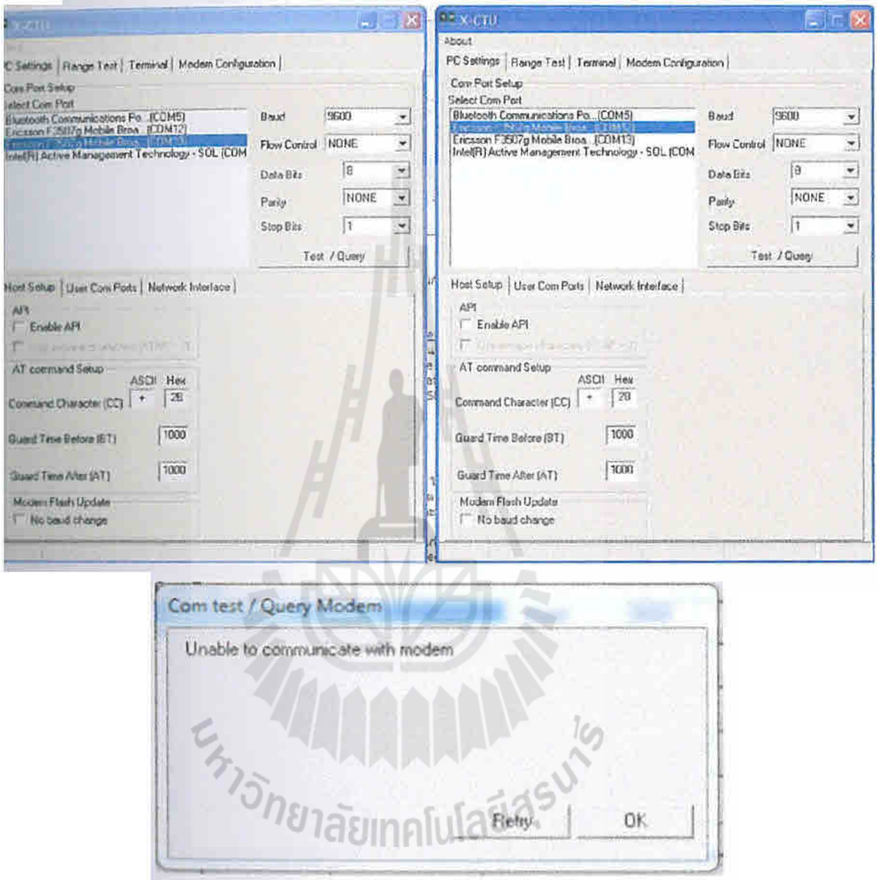
เนื่องจากเรามีตัวโมดูล ทั้งสองตัวที่ยังไม่เคยผ่านการใช้งานมาก่อน ดังนั้น เมื่อเราทำในส่วนก่อนหน้านี้ไปแล้ว ให้ใช้สาย USB เชื่อมต่อกับคอมพิวเตอร์ ติดตั้งโปรแกรมควบคุม FTDI และติดตั้งคอมพอร์ตที่จะทำงาน ทั้งนี้หมายเลขคอมพอร์ตจะเป็นเลขใดก็ได้ ระหว่าง 1 ถึง 256



รูปภาพที่ 2.41การตั้งค่าใช้งาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

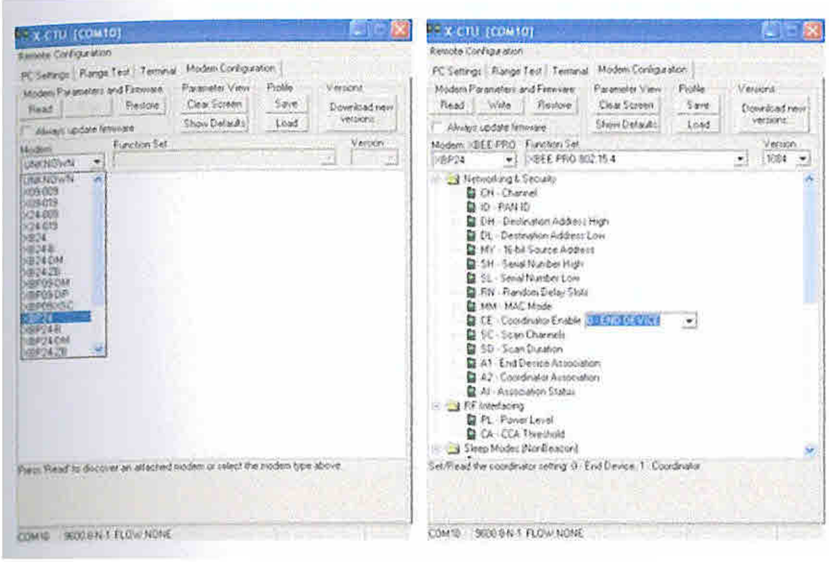
หลังจากมีการติดตั้งโปรแกรมควบคุมและติดตั้งซอร์ฟแวร์ X-CTU เรียบร้อยแล้ว เมื่อรัน พอร์ตคอมจะถูกเลือก แล้วทำการเลือกบอดเรตที่ 9600 bps สำหรับการสื่อสาร จากนั้นคลิกที่ปุ่มทดสอบ หากซอร์ฟแวร์ X-CTU เชื่อมต่อกับโมดูลสำเร็จเรียบร้อย เราจะเห็นหน้าต่างรายงานเกี่ยวกับประเภทของโมเด็มและรุ่นของโมดูลที่เราใช้



มหาวิทยาลัยเทคโนโลยีสุรนารี

รูปภาพที่ 2.42 การตั้งค่าใช้งาน

ขั้นตอนถัดไป คือการกำหนดอัตรารับส่งข้อมูลที่ถูกต้องสำหรับการสื่อสารกับ Xbee คลิกที่แท็ปสุดท้าย เรียกว่า การกำหนดค่าโมเด็ม (1), คลิกที่ปุ่มอ่าน(2), ค่าจะปรากฏขึ้นให้เลื่อนลงจนกว่าจะพบกับการเชื่อมต่อของอัตราข้อมูล, ให้เลือก(6) -57600 แล้วคลิกที่ปุ่ม เขียนบันทึกค่าของโมดูล



รูปภาพที่ 2.43 การตั้งค่าใช้งาน

เมื่อกำหนดค่าต่างๆเสร็จสมบูรณ์ เราสามารถปิดซอร์ฟแวร์และเอาโมดูลที่เชื่อมต่อออกจากคอมพิวเตอร์ได้เลย และทำการเชื่อมต่อโมดูลที่สอง ทำการกำหนดค่าX-CTU ตามสเต็ปที่กล่าวจากข้างต้นอีกครั้ง

รูปภาพที่ 2.44 Xbee Interface board

เมื่อมาถึงขั้นตอนนี้ เราต้องการที่จะเสียบโมดูลทั้งสองตัวนี้เข้าด้วยกัน แล้วทำการต่อสายเคเบิลออกมา อินเตอร์เฟสบอร์ดจะต่อเข้ากับ LED และต้องตรวจสอบให้แน่ใจว่า Pin 1 ที่ 1 เชื่อมต่อกับ Pin 1 ที่ 1 บทพอร์ตของ Xbee รวมถึง Interface Board จะมีหลอดไฟ

LED สีแดง ไม่ได้ใช้เฉพาะแค่คำสั่งบ่งชี้ แต่ใช้สำหรับเปิด Power ด้วยเช่นกัน มันจะทำงานได้ถ้า Xbee Module ถูกสวมใส่ไว้ เมื่อเชื่อมต่อถูกทุกพอร์ทถูกต้องแล้วสำหรับ ถ้าในกรณีอื่นไม่สามารถเชื่อมต่อสื่อสารได้ ให้เริ่มที่การทดสอบ Xbee ที่ความถี่ 900 MHz และ 2.45 GHz บน Xbee module จนกระทั่งสามารถรีโมตควบคุม Xbee ในความถี่ 2.45 Ghz

2.3.3 Reflective Optical Sensor with Transistor Output (TCRT5000(L))

รูปที่2.45 Sensor TCRT5000(L)

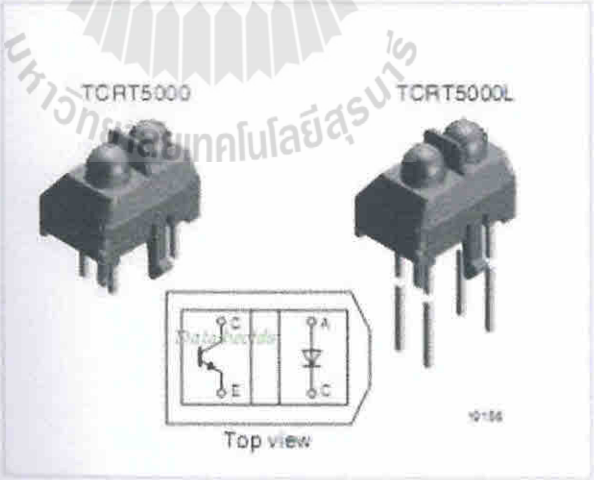
TCRT5000(L) มีการสร้างให้อยู่ในรูปแบบกะทัดรัด ส่วนที่เปล่งแสงและเครื่องตรวจจับจะถูกออกแบบจัดเรียงให้อยู่ในทิศทางเดียวกันกับการตรวจจับสัมผัสของวัตถุ โดยใช้ลำแสงอินฟาเรดสะท้อนจากวัตถุ ความยาวคลื่นที่ใช้ในการทำงาน 950 mm เครื่องตรวจจับประกอบด้วยโฟโต้ทรานซิสเตอร์

2.3.3.1 การประยุกต์ใช้งาน

- ตรวจสอบวัสดุสะท้อนแสง เช่นกระดาษ บัตร เทปแม่เหล็ก เป็นต้น
- วัตถุประสงค์ทั่วไป สำหรับพื้นที่ที่มีการจำกัด

2.3.3.2 คุณสมบัติในการทำงาน

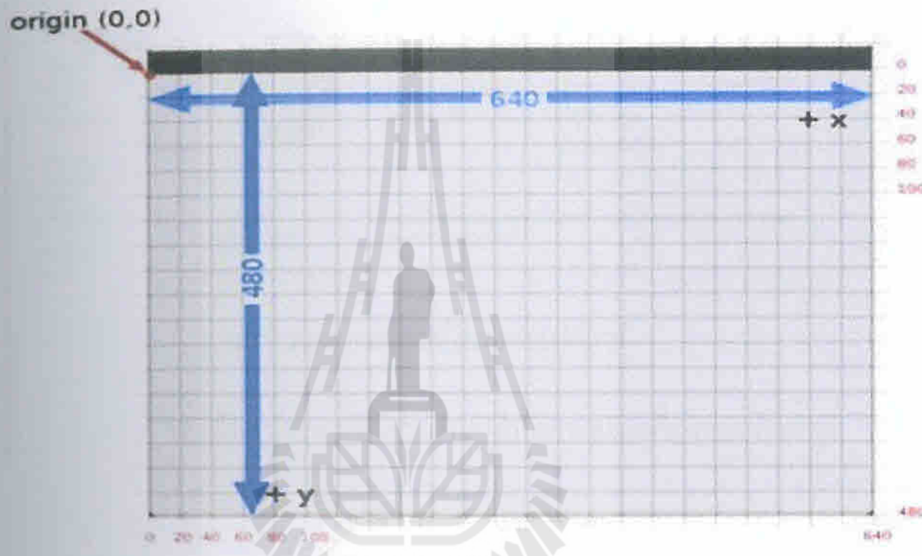
- ชนิดเครื่องตรวจจับ : Phototransistor
- ขนาด 10.2×5.8×7
- ระยะการดำเนินงานสูงสุด 2.5 mm
- การส่งออกทั่วไปภายใต้การทดสอบ : IC = 1 mA
- ความยาวคลื่น 950 nm
- เข้ากันได้กับ RoHs Directive 2022/95/EC และตาม WEEE 2002/96/EC
- มีส่วนประกอบของโพลีคาร์บอนเนตช่วยป้องกันการแทรกข้ามสัญญาณ
- Snap in สำหรับการติดตั้ง PCB
- แรงดันย้อนกลับ 5 V
- ใช้กระแสไฟฟ้า 60mA
- อุณหภูมิที่ใช้งาน -25 t0 85 องศาเซลเซียส



รูปที่ 2.46 Sensor TCRT5000(L)

2.3.4 โปรแกรม Processing

โปรแกรม Processing เหมาะสำหรับผู้ต้องการพัฒนาโปรแกรมเกี่ยวกับการสร้างภาพเคลื่อนไหวโปรแกรมที่เชื่อมโยงความรู้พื้นฐานทางคณิตศาสตร์ เช่น รูปทรงเรขาคณิต ตรีโกณมิติ ฟังก์ชันทางคณิตศาสตร์ รวมถึงการเขียนโปรแกรมแบบเรียกใช้ตัวเองโดยแสดงผลเป็นรูปกราฟิกอีกทั้ง Processing ใช้หลักการในการเขียนโปรแกรมโดยมีพื้นฐานมาจากภาษา C/C++



รูปที่ 2.47 การกำหนดค่าหน้าจอบนโปรแกรม Processing

จากรูป ดังกล่าวก่อนหน้าจอแสดงผล จะถูกกำหนดขนาดผ่านทาง Parameter วัดเป็น Pixels เมื่อเริ่มต้นวาดภาพเพื่อหน้าจอแสดงผลที่ต้องการบอกการประมวลผลในรูปแบบของพิกัด X และ Y ที่มาของหน้าต่างแสดงผลอยู่ในมุมซ้ายมือด้านบน และเป็นจุดเริ่มต้นก็จะเป็นพิกัด (0,0) หมายถึงศูนย์ X และ Y ตามลำดับใช้ได้กัน ระบบปฏิบัติการ Windows, Linux, Mac OSX เป็นซอฟต์แวร์ประเภท Open-source (เปิดเผยโค้ดต้นฉบับ) สามารถใช้สร้างรูปกราฟฟิกแบบ 2D และ 3D (2 และ 3 มิติ) หรือแบบมีปฏิสัมพันธ์ (Interactive) กับผู้ใช้ได้ เช่น ในการเรียนรู้การสร้างเกมส์คอมพิวเตอร์เบื้องต้นหรือการนำเสนอข้อมูลในรูปแบบต่างๆ (Data Visualization) ทำให้ผู้เรียนได้เห็นความเชื่อมโยงระหว่างการเขียนโค้ดและสิ่งที่ปรากฏเห็นได้อันเป็นผลมาจากการทำงานของโปรแกรม รองรับการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming)

2.3.4.1 การติดตั้งซอร์ฟแวร์

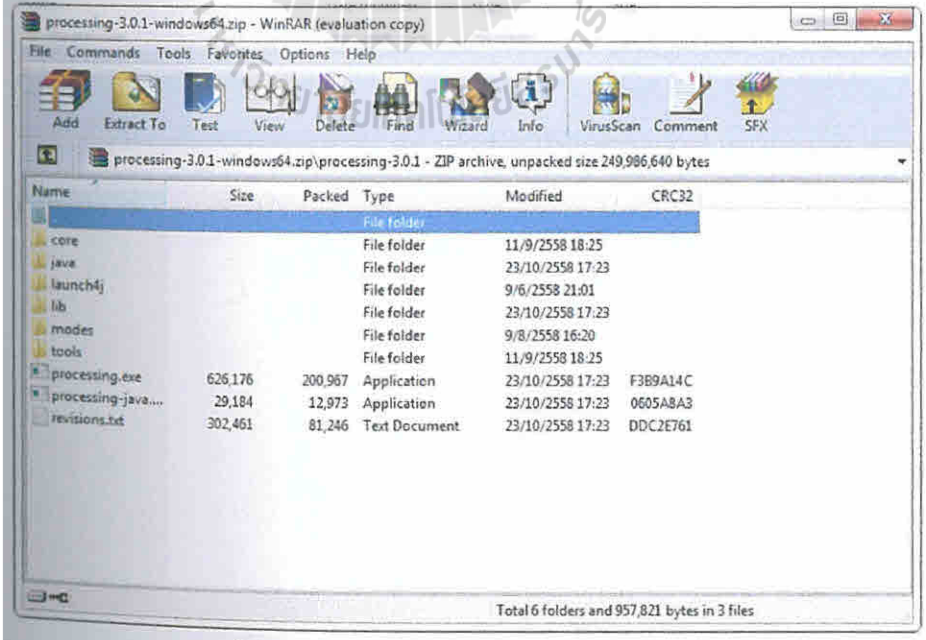
1.ทำการดาวน์โหลดซอฟต์แวร์ผ่านทาง

https://processing.org/download/?processing

รูปที่ 2.48 การดาวน์โหลดซอฟต์แวร์

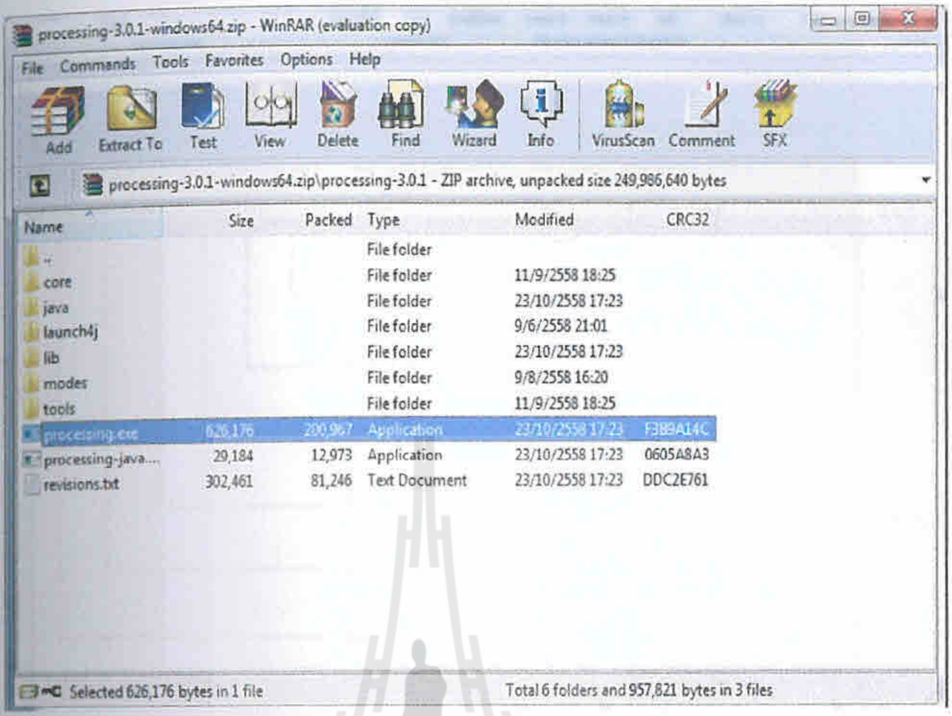
2. เลือกประเภทให้ตรงตามปฏิบัติการของเครื่องคอมพิวเตอร์

3. Unzip File จะได้โฟลเดอร์ Processing



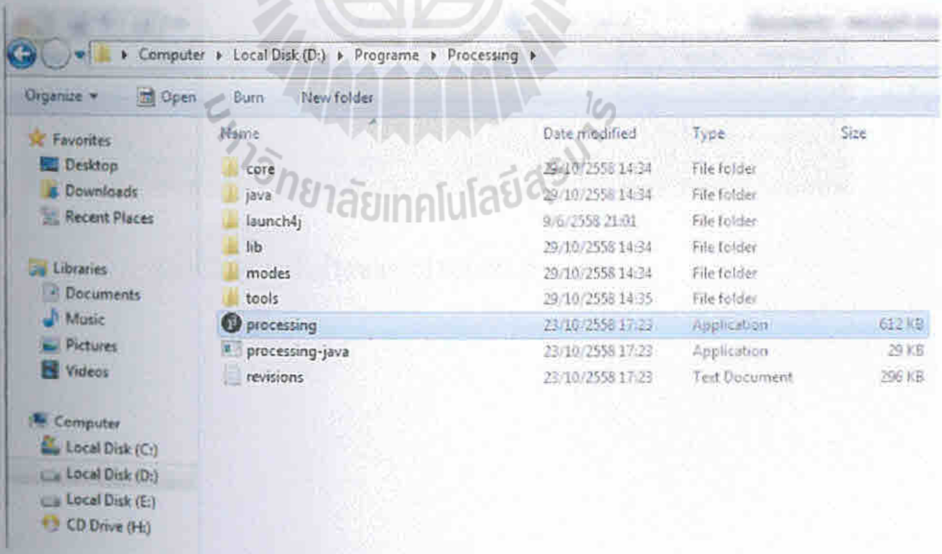
รูปที่ 2.49 Unzip File Processing

4. คลิกไปที่ไฟล์ Processing.exe ดังรูปที่ 2.50



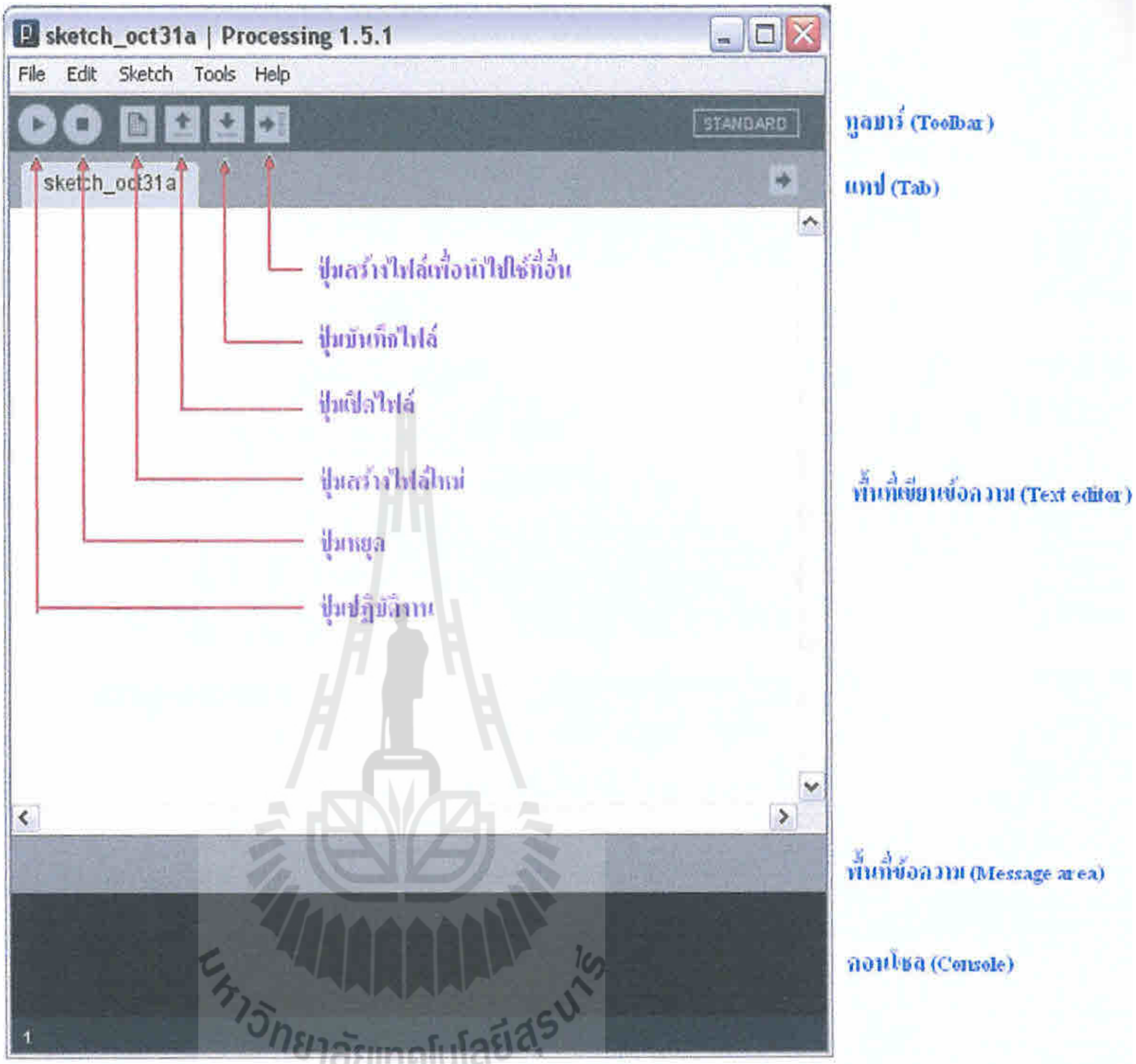
รูปที่ 2.50 การเปิดไฟล์ Processing.exe

5. ดับเบิ้ลคลิกที่โฟลเดอร์ Processing เพื่อเปิดใช้งาน



รูปที่2.51 คลิกเปิดโปรแกรม Processing เพื่อใช้งาน

6. หน้าต่างของโปรแกรม Processing



รูปที่ 2.52 หน้าต่างของโปรแกรม Processing

บทที่ 3

ชุดอุปกรณ์ต้นแบบและผลการทดลอง

3.1 กล่าวนำ

ในบทนี้จะกล่าวถึงในบทนี้จะกล่าวถึงอุปกรณ์ต้นแบบวัดรอบเครื่องยนต์แบบไร้สาย อุปกรณ์ต้นแบบนี้จะทำการรับค่าจากตัวตรวจรู้(Sensor) มายังบอร์ดโดยใช้ Xbee เป็นตัวส่งข้อมูลที่ได้มายัง Xbee อีกตัวที่อยู่กับ Laptop ทำการแสดงข้อมูลของโปรแกรม อุปกรณ์ต้นแบบนี้จะทำการทดลองในสถานที่ เวลา ระยะทาง และสภาพแวดล้อมที่แตกต่างกันไปในบทนี้ จะประกอบไปด้วย การใช้เครื่องมือวัดรอบความเร็วมาตรฐาน การทำงานของอุปกรณ์ต้นแบบ การทดสอบ การแสดงผลข้อมูลด้วย Processing 2.0 ผลการทดลองและสรุปผลการทดลอง

3.2 เครื่องมือวัดความเร็วรอบมาตรฐาน

เครื่องมือวัดความเร็วรอบมาตรฐาน ในการทดลองจะใช้ยี่ห้อ Victorรุ่น DM6234P+ดังแสดงในรูปที่ 3.1 เพื่อใช้ในการเชื่อมค่ารอบ RPM Sensor Model สามารถเก็บค่ารอบความเร็วรอบสูงสุดต่ำสุด และค่าเฉลี่ยได้ เพื่อความแม่นยำ เพื่อเปรียบเทียบค่าของอุปกรณ์ต้นแบบ การทำงานของเครื่องมือวัดความเร็วรอบมาตรฐาน ถ้าจะทำการวัดรอบ ให้กดปุ่ม Test จะมีแสงเลเซอร์นำแสงนั้นไปวางตัดขวาง กับตัวที่เราจะทำการวัดรอบ สามารถกดปล่อยสลับกันไปตามต้องการ ในการดูค่าต่ำสุดสูงสุด และค่าเฉลี่ย ให้กดปุ่ม MEM สามารถอ่านค่าได้เลย เพราะอุปกรณ์ มีหน้าจอขนาดใหญ่และสามารถอ่านค่าได้ง่าย

มหาวิทยาลัยเทคโนโลยีสุรนารี

<u>รายละเอียด</u>

- ขนาด: 5 หลักแอลซีดี LCD ขนาดใหญ่แสดงความสูงเป็นหลัก 18mm
- ความละเอียด: 0.1rpm (2.5 ~ 99999RPM); 1rpm (1000RPM)
- วัดช่วง: 2.5 ~ 99999 รอบ / นาที
- ความถูกต้อง: (0.05% 1D)

ตัวอย่างเวลา: 0.5 วินาที (120RPM)

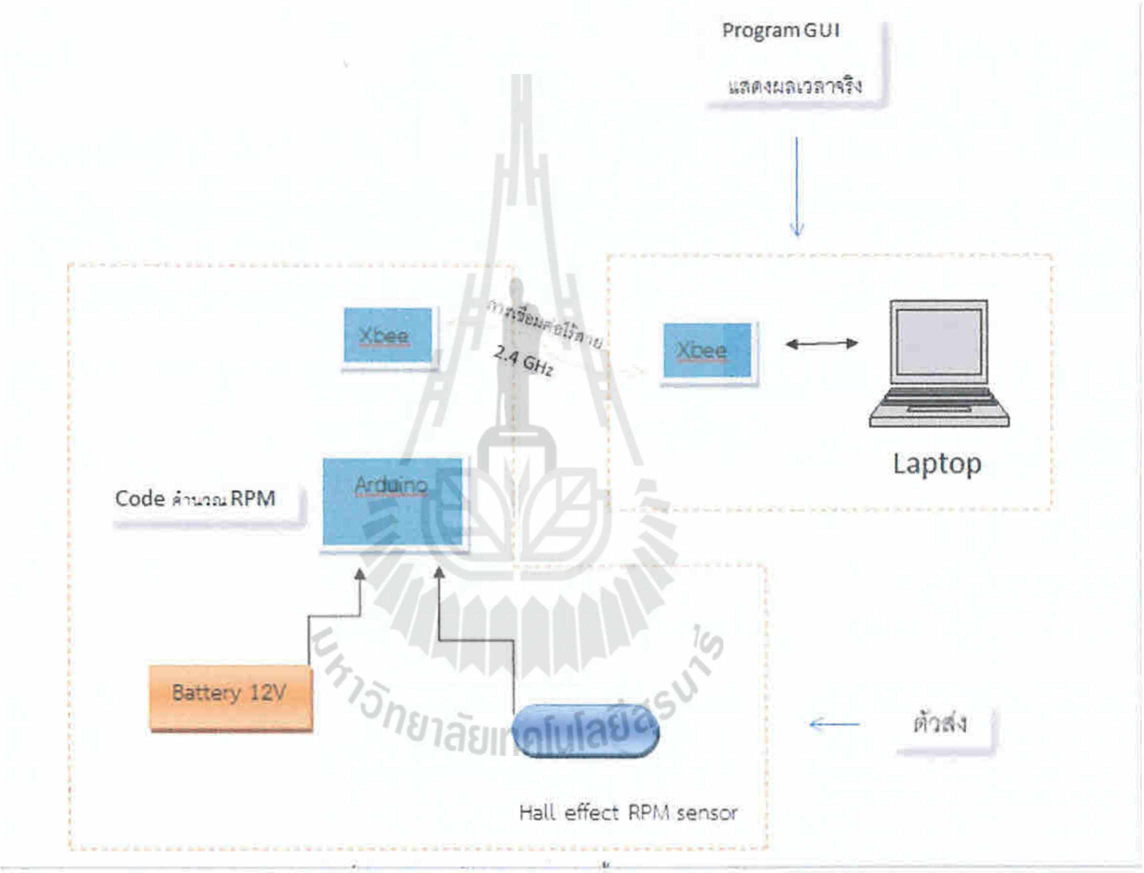
- ช่วงเลือกอัตโนมัติเลือก (เลเซอร์)
- ระยะทางที่มีประสิทธิภาพ: 50mm ~ 500mm
- ทำงานปัจจุบัน: max. 40mA.
- ฐานเวลา: 6MHz.
- ไฟ 9v (6F22) แบตเตอรี่
- ขนาดประมาณ: 130 × 70 × 29mm.

รูปที่ 3.1 เครื่องมือวัดความเร็วรอบ

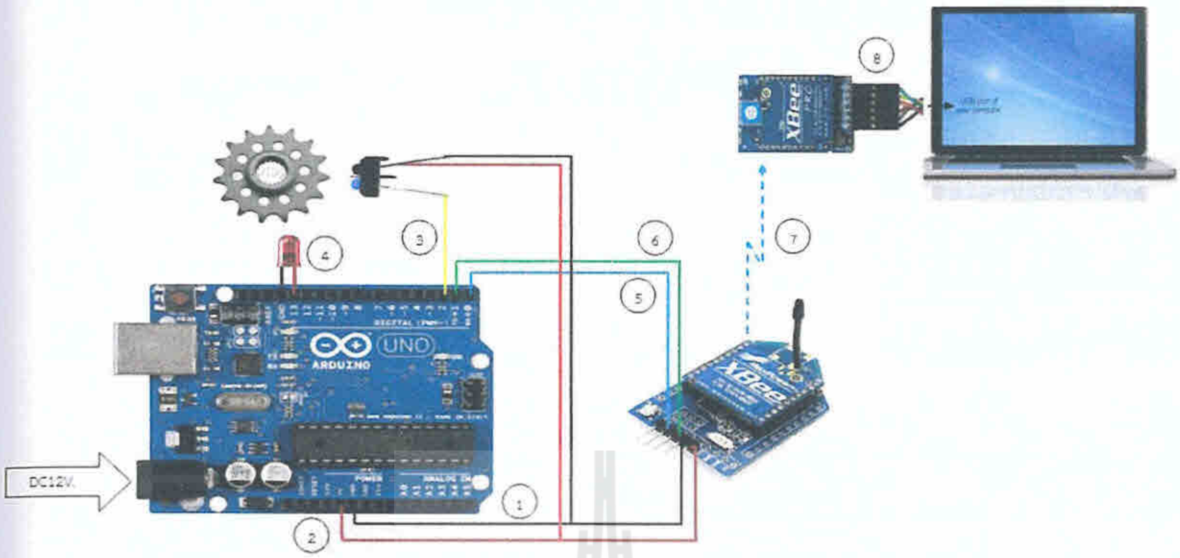
มหาวิทยาลัยเทคโนโลยีสุรนารี

3.3 ชุดอุปกรณ์ต้นแบบ

ชุดโปรแกรมต้นแบบเซ็นเซอร์วัดรอบเครื่องยนต์แบบไร้สาย ประกอบด้วย Auduino uno3 xbee pro 900 Mhz. หรือ 2.4 Ghz., Xbee Shield, IR sensor ,Battery 12 V และ Laptop ที่ทำงานร่วมกัน ดังแสดงรูปที่ 3.2 และการเชื่อมต่อดังแสดงรูปที่ 3.3ชุดอุปกรณ์ที่ใช้ทำอุปกรณ์ต้นแบบวัดความเร็วรอบเครื่องยนต์แบบไร้สาย การเชื่อมต่อของ Arduino Board ,ชุด xbee , IRsensor Battery 12 V และ laptop



(รูปที่ 3.2 แสดงระบบเซนเซอร์ไร้สายเพื่อวัดค่าความเร็วรอบ)



รูปที่ 3.3 แสดงการเชื่อมต่อระบบของอุปกรณ์ต้นแบบ

หมายเลข 1 ที่ GND ของบอร์ด Arduino UNO R3 เชื่อมต่อกับ GND ของ Xbee Shieldและ GND ของ Sensor

หมายเลข 2ที่ 5V ของบอร์ด Arduino UNO R3 เชื่อมต่อกับ 5V ของ Xbee Shield และ 5V ของ Sensor

หมายเลข 3 ที่ Output signal ของ IRsensor เชื่อมต่อกับพอร์ต Digital 2 ของ Arduino UNO R3

หมายเลข 4 ที่ไฟแอลอีดีแสดงสถานการณ์ทำงานส่งค่าที่วัดได้ของ IR Sensor บนArduino UNO R3

หมายเลข 5ที่พอร์ต Tx ของ Arduino UNO R3 เชื่อมต่อกับ Rx ของ Xbee Shield

หมายเลข 6ที่พอร์ต Rx ของ Arduino UNO R3 เชื่อมต่อกับ Txของ Xbee Shield

หมายเลข 7ที่การเชื่อมต่อของระบบไร้สาย ระหว่าง Xbee ภาคส่งกับ Xbee ภาครับ

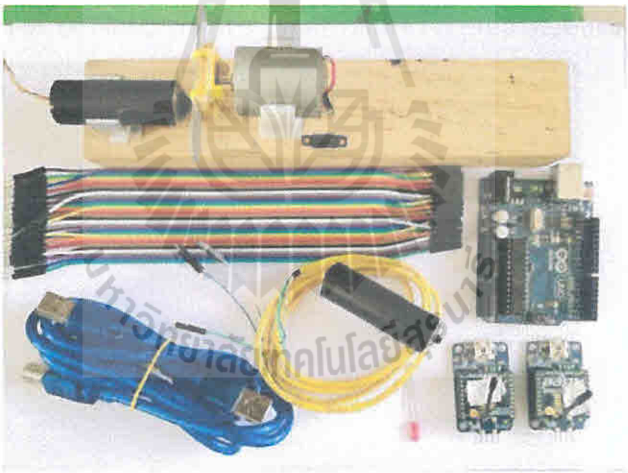
หมายเลข 8ที่การเชื่อมต่อระหว่าง Xbee ภาครับกับ Laptop แสดงผลออกทางหน้าจอ

3.4 หลักการทำงานของอุปกรณ์ต้นแบบ

อุปกรณ์ต้นแบบวัดรอบเครื่องยนต์แบบไร้สาย ซึ่งมีหลักการทำงาน ดังรูปที่ 3.4 จะเริ่มการทำงานรับข้อมูลจากขา Output Signal ของ IR Sensor เข้ามายัง Board Arduino ที่ Port Digital 2 ทำการประมวลผล เสร็จจะส่งข้อมูลไปยังขา Tx ต่อกับขา Rx ของ Xbee แล้ว Xbee ทำการส่งข้อมูลแบบผ่านสัญญาณไร้ ไปยัง Xbee อีกตัวที่ติดอยู่กับ Laptop ในขณะเดียวกัน Laptop จะอ่านข้อมูลต่อที่ได้มาแสดงผลเป็นจำนวนรอบต่อเวลาแบบ Real Time - Port

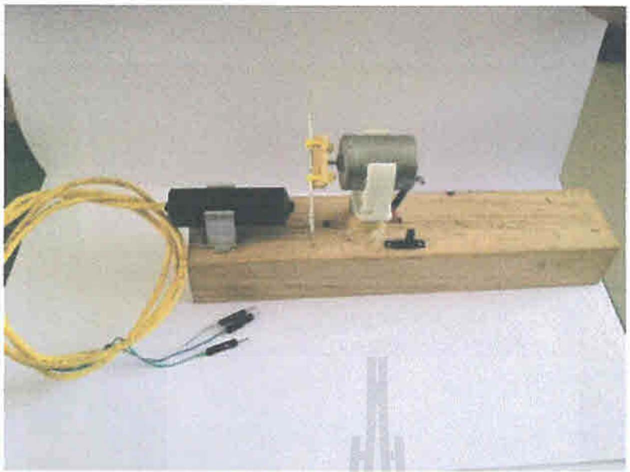
ขั้นตอนการทดสอบเบื้องต้น

1. อัพโหลดโค้ดลงไปในบอร์ด Arduino UNO R3 โดยใช้โปรแกรม Arduino



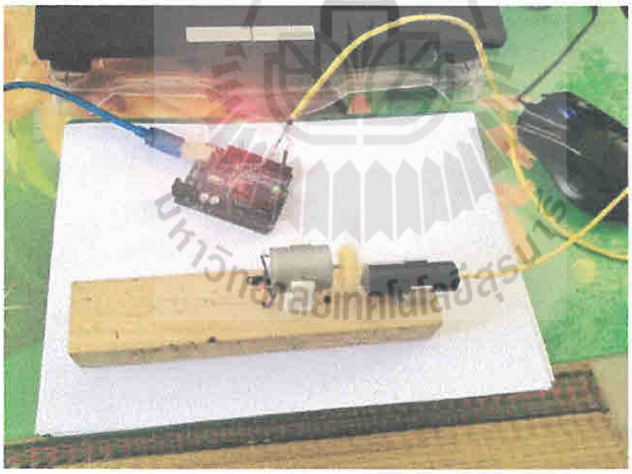
รูปที่ 3.4 อุปกรณ์ที่ใช้ในการอัพโหลด

2. นำเซนเซอร์ติดตั้งบนแท่นทดลองพร้อมกับมอเตอร์เพื่อวัดรอบ



รูปที่ 3.5 นำเซ็นเซอร์ติดตั้งบนแท่นทดลองเพื่อวัดรอบ

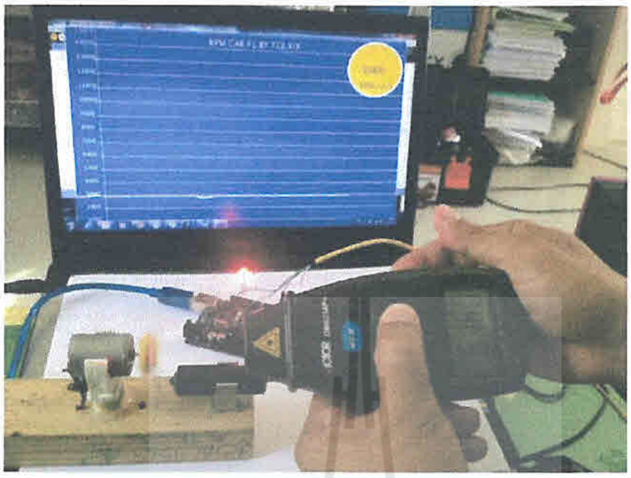
3. นำเซนเซอร์ IR Sensor ต่อไฟ 5 โวลต์กระแสตรงจากบอร์ด Arduino UNO R3



รูปที่ 3.6 ต่อไฟ 5 โวลต์จากบอร์ด Arduino UNO R3

มหาวิทยาลัยเทคโนโลยีสุรนารี

4. ทำการปรับค่าให้ตรงกับอุปกรณ์วัดความเร็วรอบแบบมาตรฐาน



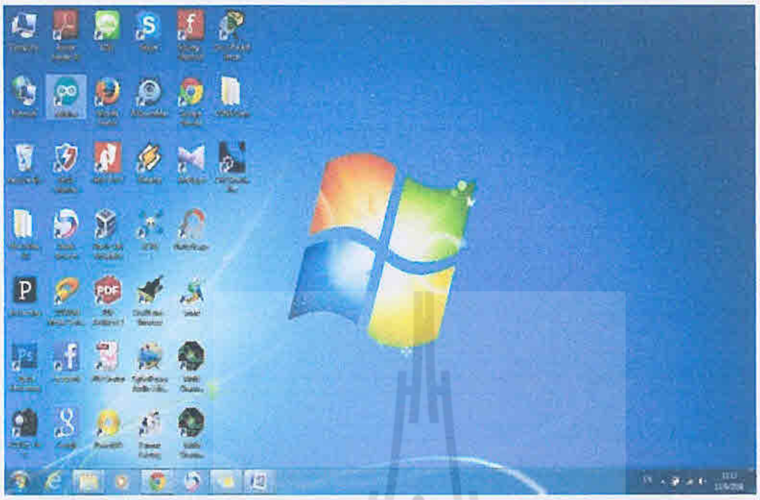
รูปที่ 3.7 ทดสอบ,ปรับค่าให้ตรงกับอุปกรณ์วัดความเร็วรอบแบบมาตรฐาน

5. นำ Xbee จำนวน 2 ตัวที่จะการส่งค่าแบบไร้สายมาทำการตั้งค่าเพื่อให้ Xbee ทั้ง 2ตัวเชื่อมกันได้ โดยการใช้โปรแกรม XCTU
6. ทำการวัดค่าความเร็วรอบได้เลย

มหาวิทยาลัยเทคโนโลยีสุรนารี

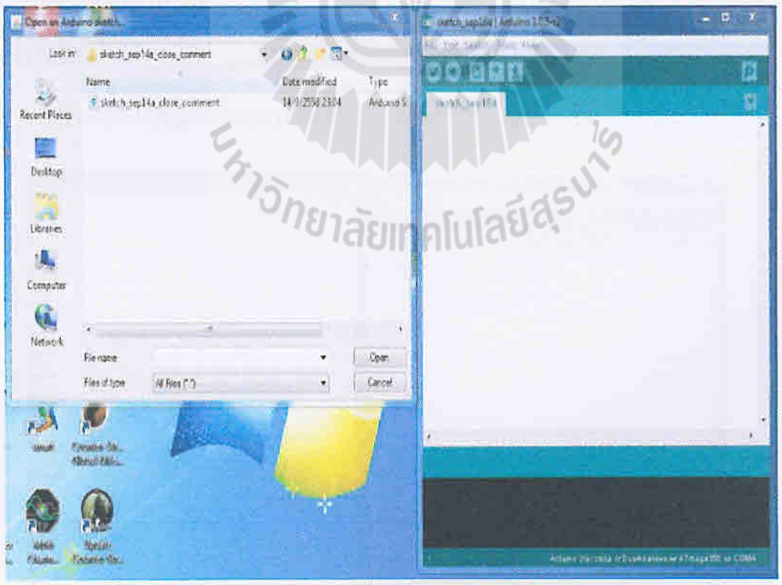
ขั้นตอนการอัพโหลดโค้ดลงในบอร์ด Arduino UNO R3

1. เลือกโปรแกรม Arduino เพื่อทำการเขียนโค้ดการนับรอบ



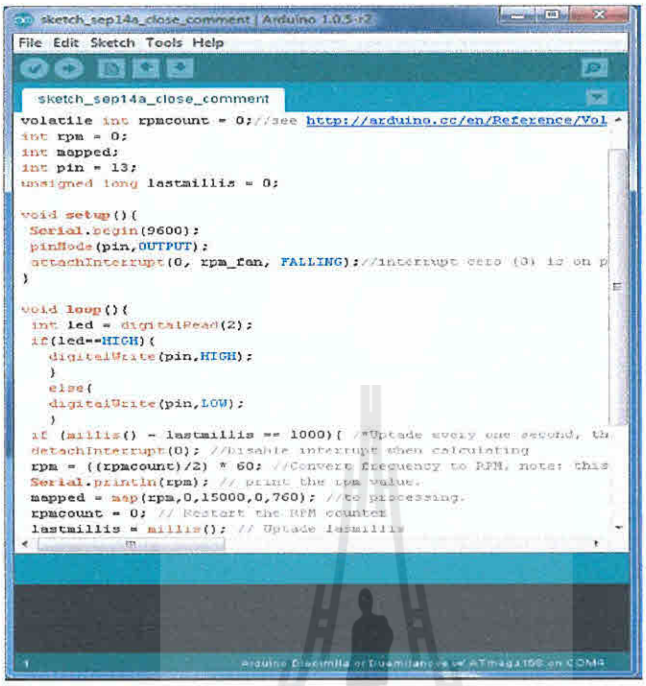
รูปที่ 3.8 เปิดโปรแกรม Arduino

2. เป็นขั้นตอนการเปิดโค้ดที่จะอัพโหลดลงบน Arduino UNO R3



รูปที่ 3.9 ขั้นตอนการอัพโหลดโค้ดลงบอร์ด

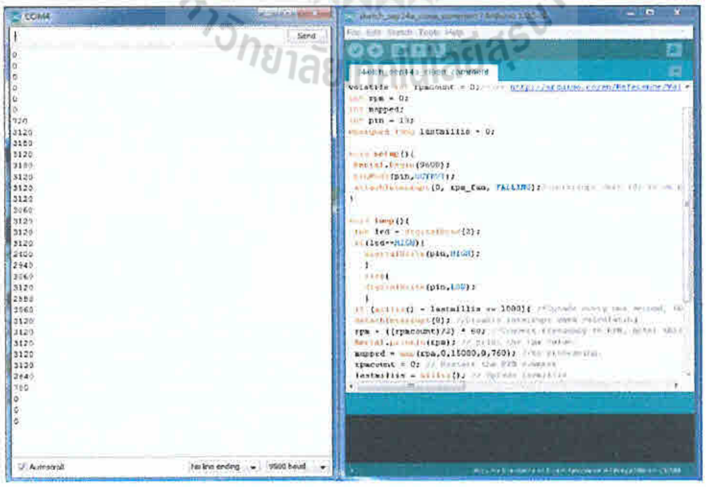
3. การตรวจสอบโค้ดบนโปรแกรม Arduino



รูปที่ 3.10 เขียนโค้ดลงในโปรแกรมแล้วตรวจสอบ

4. การอัพโหลดโค้ดที่ถูกต้องแล้วลงบน Arduino UNO R3

5. การเปิด Monitor เพื่อดูผลของการวัดรอบของเซนเซอร์



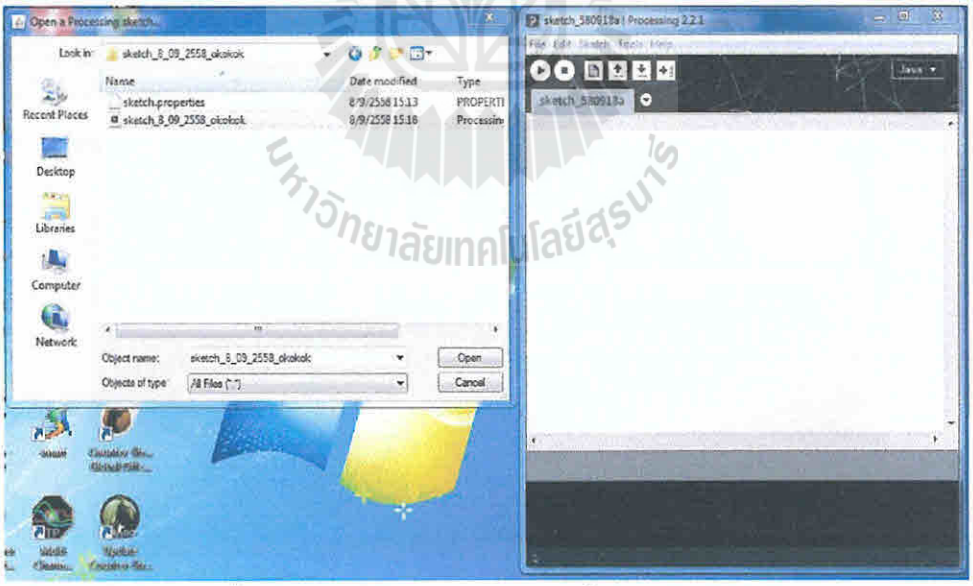
รูปที่ 3.11 เปิดมอนิเตอร์เพื่อดูค่าที่ได้จากการวัด

ขั้นตอนการแสดงผลเป็นจำนวนรอบต่อเวลาแบบ Real Time - Port โดยใช้โปรแกรม Processing 2.0

1. การเข้าโปแกรม Processing 2.0

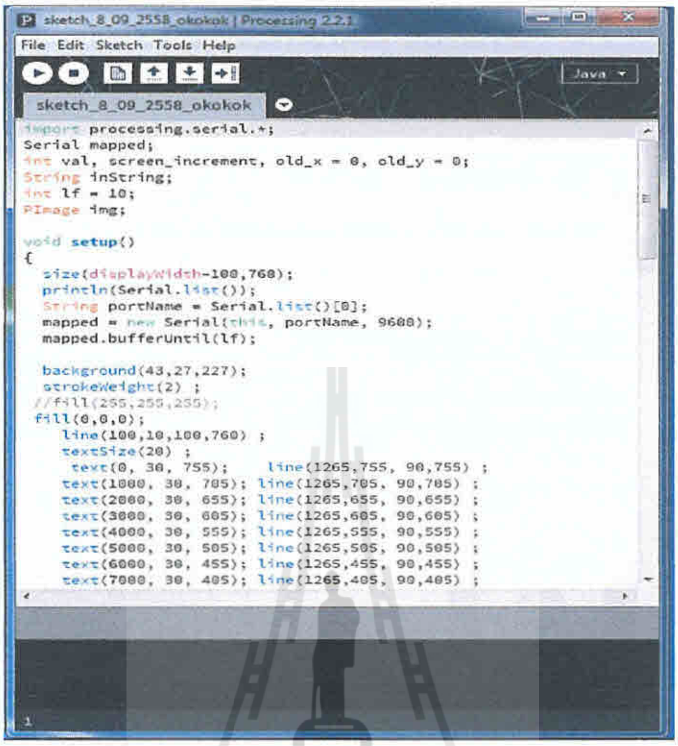
รูปที่ 3.12 เปิดโปรแกรม Processing 2.0

2. การเปิดโค้ดที่จะแสดงเป็นกราฟบนโปรแกรม Processing 2.0



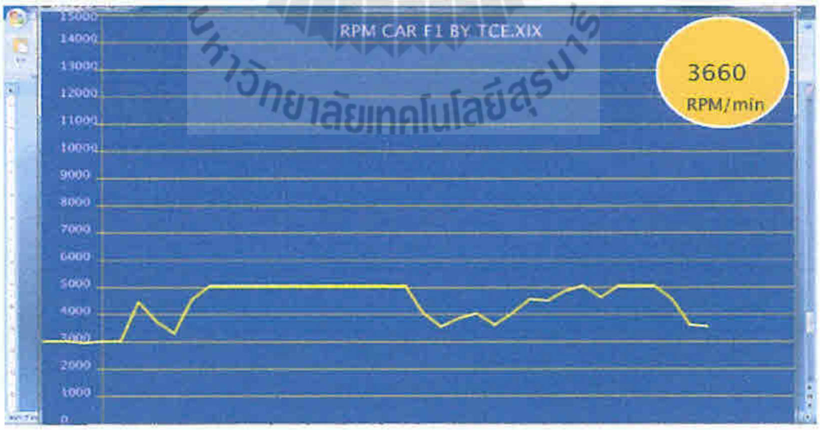
รูปที่ 3.13 เขียนโค้ดลงในโปรแกรมเพื่อคำนวณค่า

3. ทำการตรวจสอบและรันโค้ดแสดงเส้นกราฟของการวัดรอบ



รูปที่ 3.14 ตรวจสอบความถูกต้องของโค้ด

4. เส้นกราฟแสดงการวัดรอบที่รับข้อมูลผ่านไร้สายระหว่าง Xbee ทั้ง 2 ตัว มายัง Laptop



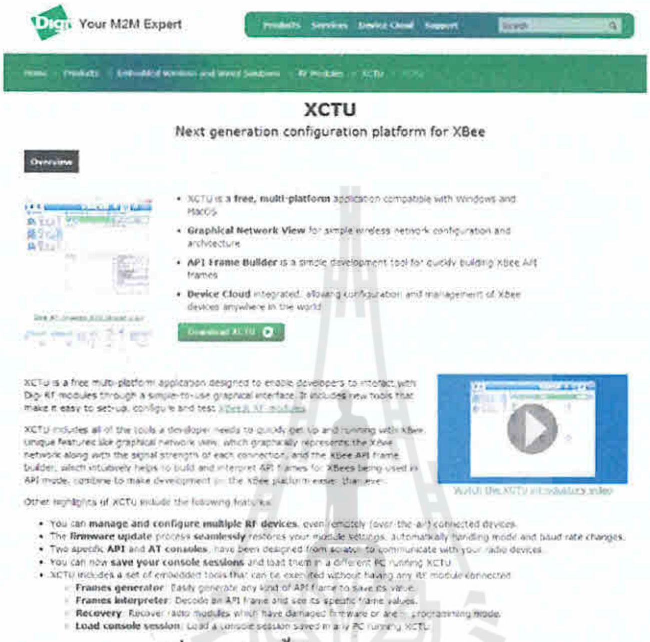
รูปที่ 3.15 กราฟที่แสดงการวัดรอบเมื่อผ่านการส่งสัญญาณแบบไร้สาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

การเชื่อมต่อ Xbee Pro โดยการใช้ Program XCTU

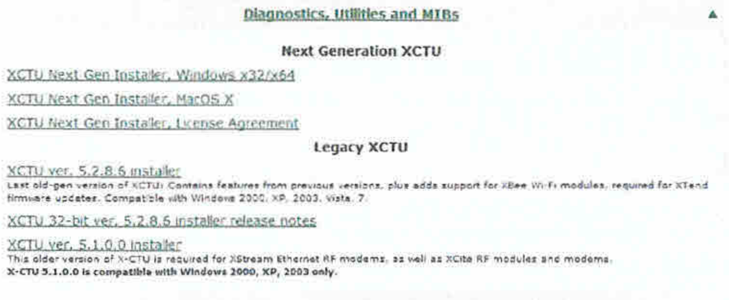
1. ติดตั้งโปรแกรม X-CTU โดยดาวน์โหลดไฟล์ติดตั้งได้

   จาก http://www.digi.com/products/wireless-wired-embedded-solutions/ZigBee-rf-modules/xctu



รูปที่ 3.16ติดตั้งโปรแกรม XCTU

จากนั้นเลือกไฟล์ติดตั้งที่ตรงกับ OS ที่ติดตั้ง ในตัวอย่างนี้เลือกเป็น XCTU NEXT Gen Installer, Windowsx32/x64ดังรูปที่ 3.17



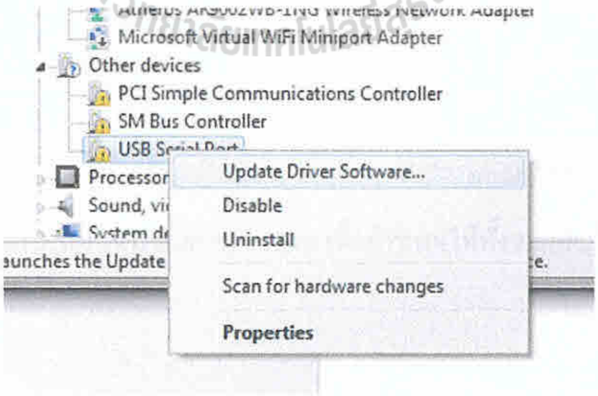
รูปที่ 3.17 เลือกไฟล์ที่ตรงกับ OS ที่ติดตั้ง

2. ติดตั้ง Driver USB to Serial สำหรับบอร์ด Mini Xbee USB Dongle โดย download ไฟล์ติดตั้ง ได้ที่http://www.ftdichip.com/Drivers/VCP.htmจากนั้นเลือกไฟล์ติดตั้งที่ตรงกับ OS ที่ติดตั้ง ในตัวอย่างนี้เลือกเป็น Windows X86 (32-bit)

		Processor Architecture							
Operating System	Release Date	x86 (32-bit)	x64 (64-bit)	PPC	ARM	MIPSII	MIPSIV	SH4	Comments
Windows*	2015-07-28	2.12.06	2.12.06	-	-	-	-	-	2.12.06 WHQL Certified Available as setup executable Release Notes
Linux	2009-05-14	1.5.0	1.5.0	-	-	-	-	-	All FTDI devices now supported in Ubuntu 11.10, kernel 3.0.0-19 Refer to TN-101 if you need a custom VCP VID/PID in Linux
Mac OS X 10.3 to 10.8	2012-08-10	2.2.18	2.2.18	2.2.18	-	-	-	-	Refer to TN-105 if you need a custom VCP VID/PID in MAC OS
Mac OS X 10.9 and above	2015-04-15	-	2.3	-	-	-	-	-	This driver is signed by Apple
Windows CE 4.2-5.2**	2012-01-06	1.1.0.20	-	-	1.1.0.20	1.1.0.10	1.1.0.10	1.1.0.10	
Windows CE 6.0/7.0	2012-01-06	1.1.0.20 CE 6.0 CAT CE 7.0 CAT	-	-	1.1.0.20 CE 6.0 CAT CE 7.0 CAT	1.1.0.10	1.1.0.10	1.1.0.10	For use of the CAT files supplied for ARM and x86 builds refer to AN_319
Windows CE 2013	2015-03-06	BETA			BETA				BETA VCP Driver Support for WinCE2013

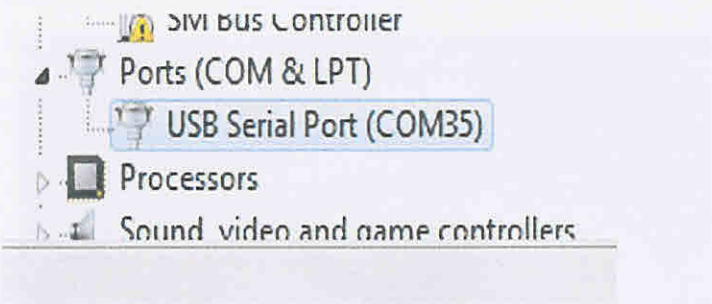
รูปที่ 3.18 ติดตั้ง Driver USB to Serial โดยเลือกไฟล์ที่ติดตั้งให้ตรงกับ OS

3. ติดตั้ง Driver Mini Xbee USB Dongle โดยเสียบสาย USB เข้ากับบอร์ด และ PC เมื่อเข้าไปที่ Device Manager ในหัวข้อ Other Device คลิกขวา ที่ USB Serial Port เลือกหัวข้อ Update Driver Software … จากนั้นเลือกที่อยู่ไฟล์ Driver ในข้อ 2



รูปที่ 3.19ขั้นตอนการติดตั้งDriver Mini Xbee USB Dongle

หลังจากติดตั้ง Driver เสร็จเรียบร้อยแล้ว PC จะเห็นบอร์ดเป็น USB Serial Port หนึ่งพอร์ตในระบบ



รูปที่ 3.20แสดงพอร์ตที่ใช้งาน

4. ตรวจสอบ Serial Number โดยใน Xbee ทุกตัวจะมีหมายเลข Serial Number อยู่ มีตัวเลขอยู่ 2 ชุด คือ SH (Serial Number High) และ SL (Serial Number Low) ใช้ค่านี้เพื่อกำหนดให้ Xbee ทั้งสองติดต่อกัน

รูปที่ 3.21ตรวจสอบหมายเลขของ Xbee เพื่อกำหนดให้ทั้งสองตัวสื่อสารกันได้

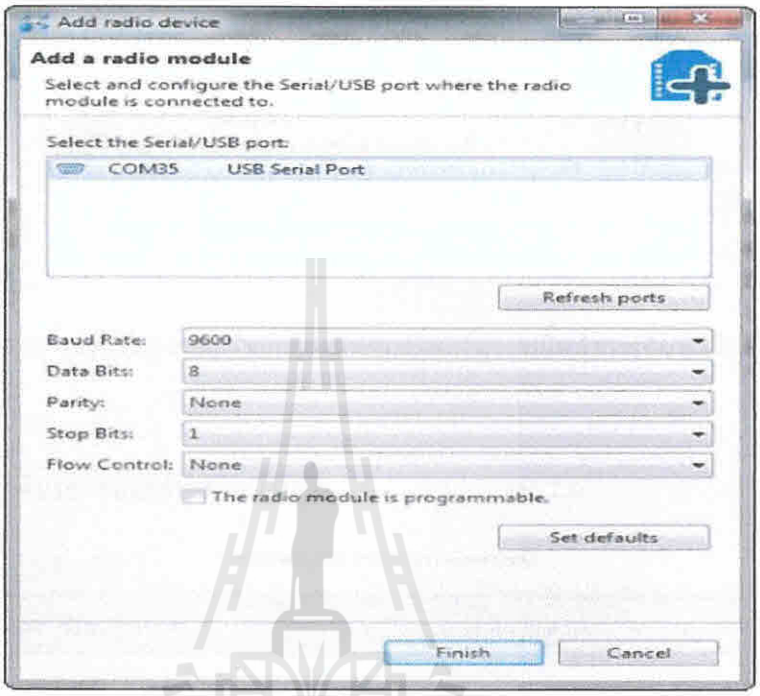
5. กำหนดค่าให้กับ Xbee ที่ต้องการเซตเป็น Coordinator ก่อน โดยต่อโมดูล Xbee เข้ากับบอร์ด Mini Xbee USB Dongle และต่อเข้ากับ PC

รูปที่ 3.22 ทำการกำหนดค่าโดยต่อเข้ากับ PC

6. เปิดโปรแกรม XCTU กดปุ่ม Add a Radio Module … ดังภาพ

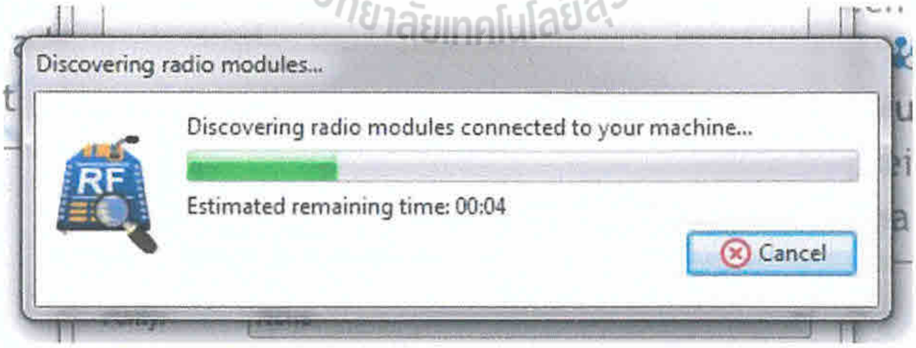
รูปที่ 3.23 เปิดโปรแกรม XCTU เพื่อเช็คว่าทั้งสองตัวติดต่อกันได้หรือไม่

7. เลือกหมายเลข Serial Port ของ Mini Xbee USB Dongle กำหนด Baud Rate เป็น 9600 bps Data Bits เท่ากับ 8 Parity เป็น None Stop Bits เท่ากับ 1 และ Flow Control เป็น None จากนั้นกดปุ่ม Finish



รูปที่ 3.24 กำหนดค่า โดยเลือกอัตราบอดเรทที่9600

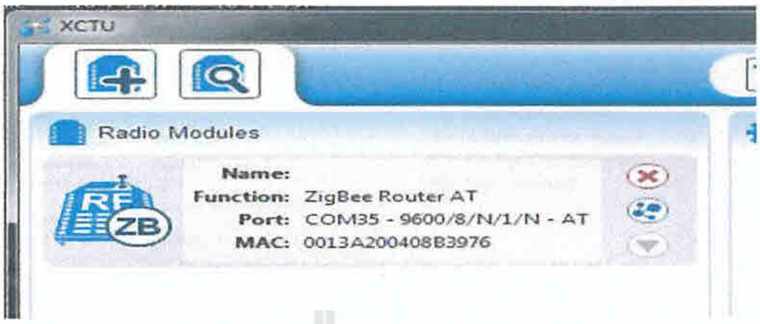
8. รอจนกว่าหน้าต่างจะทำการค้นหาโมดูล Xbee เสร็จ



รูปที่ 3.25 โปรแกรมกำลังค้นหาโมดูล

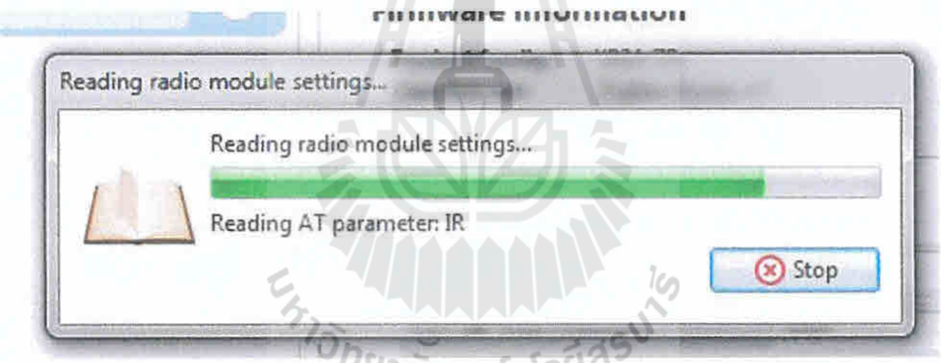
มหาวิทยาลัยเทคโนโลยีสุรนารี

9. จากนั้นโปรแกรมจะแสดงชื่อรุ่น Xbee ขึ้นมาดังภาพ ให้คลิกที่โมดูล เพื่อดูรายละเอียดพารามิเตอร์ต่างๆ ของโมดูล



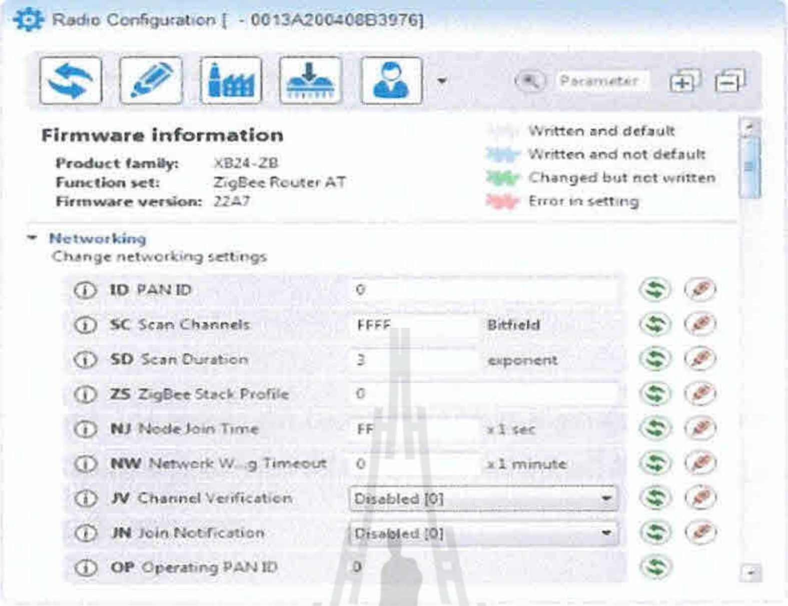
รูปที่ 3.26 แสดงชื่อรุ่นและรายละเอียดพารามิเตอร์ต่างๆเมื่อทำการค้นหาสำเร็จ

10. รอจนกว่าโปรแกรมจะอ่านค่าบนโมดูล



รูปที่ 3.27 ทำการค้นหาเพื่ออ่านค่าบน โมดูล

11. จากนั้นที่ด้านขวาของหน้าต่างโปรแกรมในหัวข้อ Radio Configuration จะแสดงค่าพารามิเตอร์ของ Xbee ดังภาพ

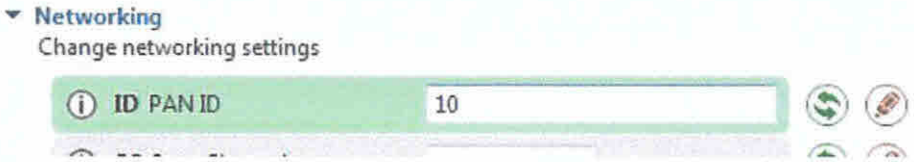


รูปที่ 3.28 แสดงค่าพารามิเตอร์ของ Xbee

12. กดปุ่ม Collapse All Setting Section เพื่อแสดงเฉพาะหัวข้อใหญ่เท่านั้น

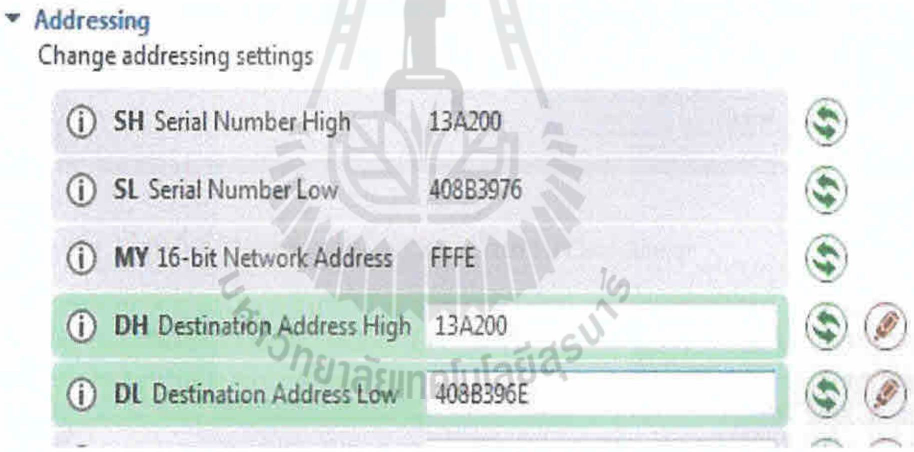
รูปที่ 3.29 แสดงหัวข้อต่างๆในโปรแกรม

13. ไปที่หัวข้อ Networking > PAN ID (Personal Area Network Identifier) แสดงหมายเลข ID ของ Network ที่ Xbee ใช้ ในตัวอย่างนี้กำหนดเป็น 10



รูปที่ 3.30 แสดงหมายเลข ID

14. ไปที่หัวข้อ Addressing > DH Destination Address High และ DL Destination Address Low กำหนด Serial Number Xbee ที่ต้องการติดต่อในตัวอย่างนี้ กำหนดเป็น DH เป็น 13A200 และ DL เป็น 408B396E



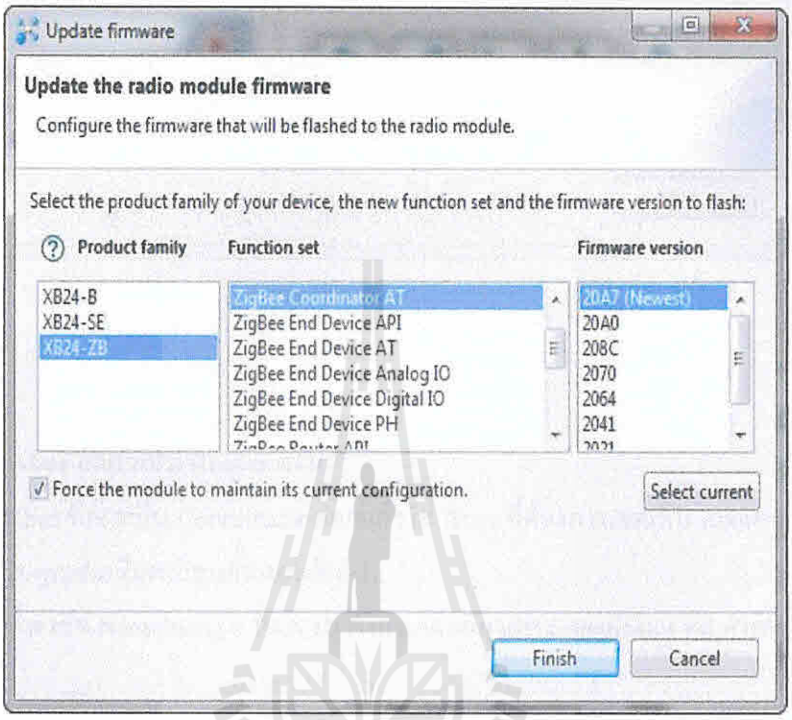
รูปที่ 3.31แสดงการกำหนด Serial Number Xbee

15. กดปุ่ม Update Firmware ดังภาพ

รูปที่ 3.32 รายละเอียดปุ่มต่างๆ

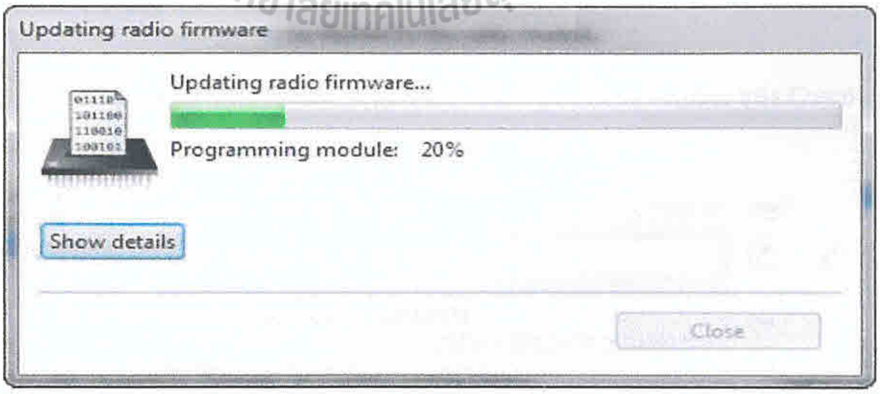
มหาวิทยาลัยเทคโนโลยีสุรนารี

16. เลือกรุ่น (Product Family) ของ Xbee เป็น XB24-ZB ที่ช่อง Function Set กำหนดให้ Xbee เป็น Coordinator โหมดการสื่อสารแบบ AT และเลือก Version เป็นรุ่นล่าสุด (20A7) ดังภาพ แล้วกดปุ่ม Finish



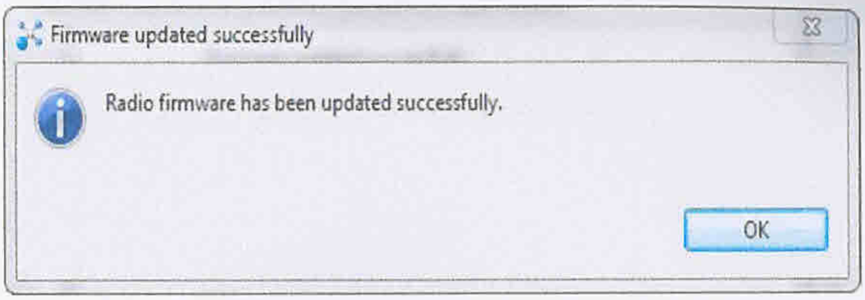
รูปที่ 3.33 แสดงการตั้งค่า Xbee ในส่วน Coordinator

17. รอจนกว่า โปรแกรมจะอัพเดท Firmware และเซต Parameter ตามที่เราได้กำหนดก่อนหน้านี้



รูปที่ 3.34 รอโปรแกรมอัพเดทและเซตค่าตามที่เรากำหนด

18. ถ้าการอัพเดทสำเร็จ โปรแกรมจะแสดงข้อความ "Radio Firmware Has Been Update Successfully" เสร็จขั้นตอนการเซตโมดูล XBee เป็น Coordinator

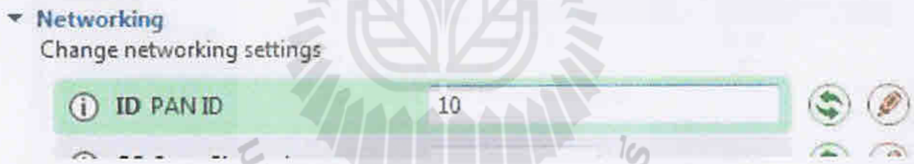


รูปที่ 3.35 แสดงหน้าต่างการอัพเดทโปรแกรมสำเร็จ

ต่อไปการเซต Xbee อีกตัวเป็น Router ดังนี้

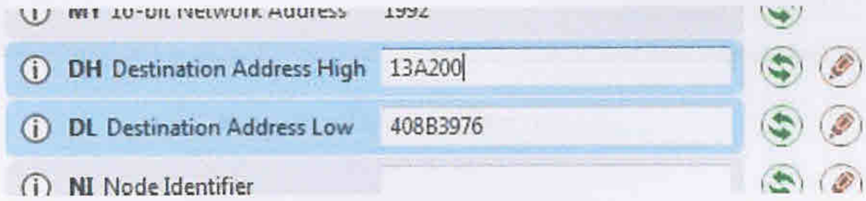
ถอด Xbee ที่เซตเป็น Coordinator ออกแล้วใส่ Xbee ที่ต้องการเซตเป็น Router เข้าไปแทน จากนั้นค้นหาโมดูลเช่นเดิมเหมือนขั้นตอนที่ 6-12

19. ไปที่หัวข้อ Networking > PAN ID กำหนดค่าเดียวกับ Coordinator คือ กำหนดเป็น 10



รูปที่ 3.36แสดงการกำหนดค่า PAN ID

20. ในหัวข้อ Addressing ส่วนของ DH และ DL กำหนดเลข Serial Number ของ Coordinator ลงไป กำหนดเป็น DH เป็น 13A200 และ DL เป็น 408B3976

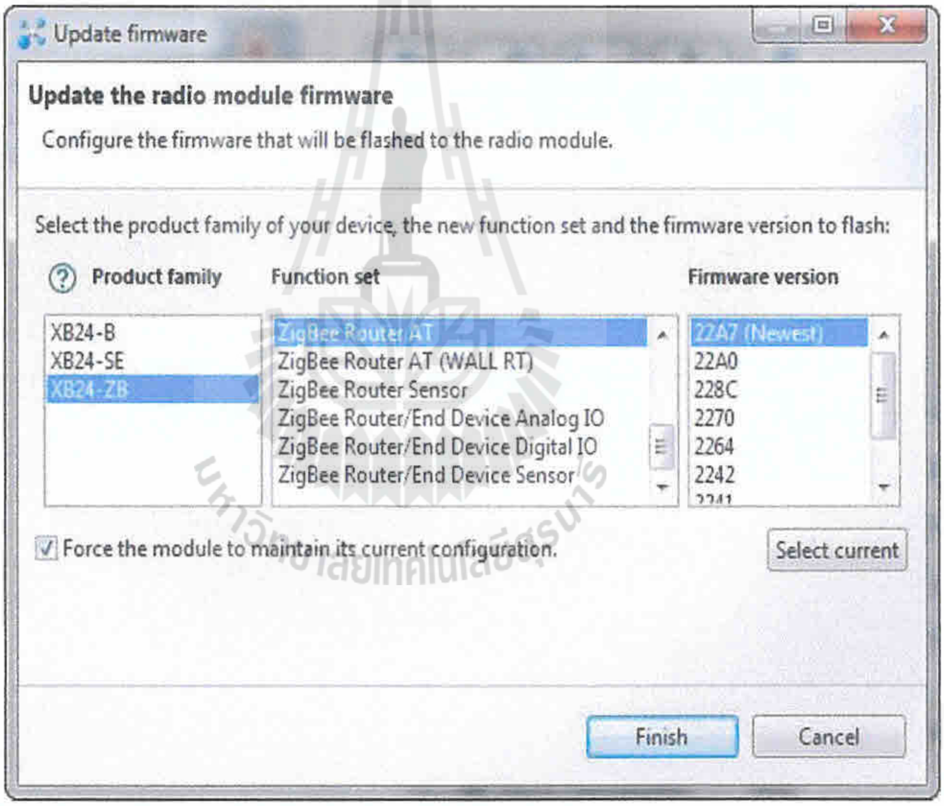


รูปที่ 3.37แสดงการกำหนดซีเรียลนัมเบอร์

21. กดปุ่ม Update Firmware ดังภาพ

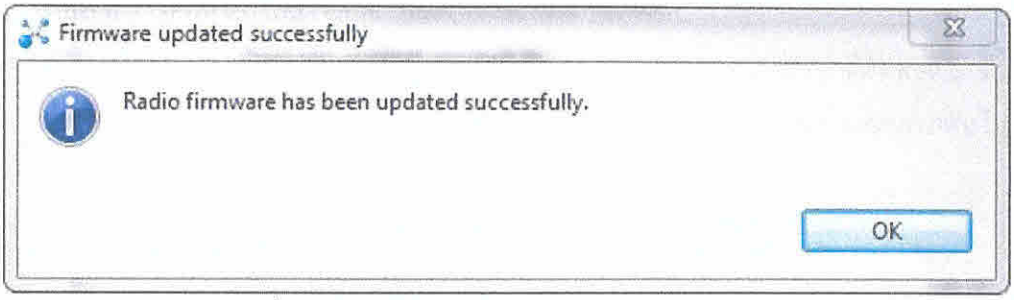
รูปที่ 3.38 แสดงรายละเอียดปุ่มต่างๆ

22. เลือกรุ่น (Product Family) ของ Xbeeเป็น XB24-ZB ที่ช่อง Function set กำหนดให้ Xbee เป็น Router โหมดการสื่อสารแบบ AT และเลือก Version เป็นรุ่นล่าสุด 22A7 ดังภาพ



รูปที่ 3.39แสดงการตั้งค่า Xbee ในส่วนRouter

23. ถ้าการอัพเกรดสำเร็จโปรแกรมจะแสดงข้อความ “Radio Firmware Has Been Update Successfully” เสร็จขั้นตอนการเซตโมดูล Xbee เป็น Router



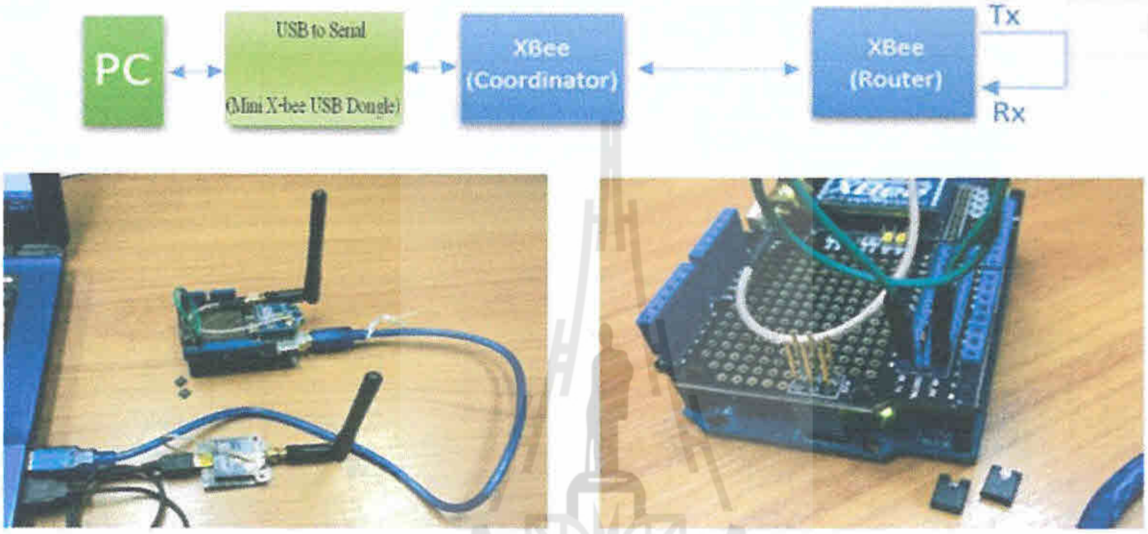
รูปที่ 3.40 แสดงการอัพเกรดโปรแกรมสำเร็จ

มหาวิทยาลัยเทคโนโลยีสุรนารี

ทดสอบการเชื่อมต่อเบื้องต้น

ถ้าการเซต Xbee ไม่มีข้อผิดพลาด Coordinator สามารถส่งข้อมูลไปยัง Router ได้ และในทางกลับกัน Router ก็สามารถส่งข้อมูลไปยัง Coordinator ได้

ทดสอบโดยเชื่อมต่อ Coordinator เข้ากับ PC ส่งข้อมูลผ่านโปรแกรม XCTU ที่ฝั่ง Router จั๊มขา Tx กับ Rx ของโมดูลเข้าด้วยกันหมายความว่าข้อมูลทั้งหมดที่ส่งจาก Coordinator จะถูกส่งกลับไปยัง Coordinator ดังภาพ



รูปที่ 3.41ทดสอบการส่งผ่านข้อมูลระหว่าง Router และ Coordinator

ฝั่ง Coordinator ต่อเข้ากับ PC

ฝั่ง Router ต่อเข้ากับ Arduino

ฝั่ง Router ต่อขา Txกับ Rx เข้าหากัน

ใช้โปรแกรม XCTU ส่งข้อมูลเข้าไปที่ Coordinator จากภาพจะเห็นได้ว่าข้อมูลทั้งส่งและรับ เป็นข้อมูลเดียวกัน

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.5 ผลการทดลอง

ตารางผลการทดลอง

ตารางที่ 3.5.1 ในการคำนวณค่าเปอร์เซ็นต์ความผิดพลาด จากการเก็บค่า RPM (Revolutions Per Minute)ของอุปกรณ์ต้นแบบทั้งหมด 150 ค่า

ค่าที่วัดได้	ค่าจริง	ค่าที่วัดได้	ค่าจริง	ค่าที่วัดได้	ค่าจริง
840	862	860	885	1440	1441
840	862	840	863	1440	1446
840	862	840	862	1440	1454
900	862	960	1024	1500	1509
840	863	960	1024	1500	1508
840	862	960	1101	1500	1508
840	863	1080	1101	1500	1511
840	864	1080	1101	1500	1513
840	862	1080	1101	1800	1800
840	862	1140	1152	1800	1820
900	864	1140	1152	1810	1842
840	863	1140	1152	1680	1691
840	862	1140	1138	1680	1602
840	862	1140	1138	1260	1223
900	865	1139	1138	1260	1223
900	863	1139	1138	1200	1212
840	863	1200	1194	1200	1212
840	863	1200	1194	1260	1240
840	862	1200	1196	1260	1238
840	863	1200	1163	1260	1230
840	862	1080	1197	1200	1238
900	862	1080	1197	1300	1251

ค่าที่วัดได้	ค่าจริง	ค่าที่วัดได้	ค่าจริง	ค่าที่วัดได้	ค่าจริง
840	863	1080	1197	1320	1325
900	862	1080	1158	1320	1418
840	863	1080	1158	1380	1477
900	863	1200	1187	1620	1551
840	862	1200	1187	1620	1557
840	863	1200	1187	1560	1525
840	863	1200	1205	1560	1525
840	862	1200	1205	1560	1503
840	863	1200	1205	1500	1497
900	863	1200	1205	1500	1529
840	863	1200	1199	1500	1523
900	863	1200	1199	1140	1152
840	863	1200	1231	1140	1138
840	863	1300	1290	1140	1138
840	863	1300	1290	1139	1138
900	865	1320	1290	1139	1138
840	864	1440	1145	1200	1194
840	864	1440	1145	1080	1101
840	864	1440	1145	1080	1101
900	865	1440	1145	1080	1097
840	863	1500	1500	1080	1158
840	862	1500	1560	1200	1205
840	864	1500	1564	1200	1225
900	891	1500	1500	1080	1101
900	890	1500	1526	960	1024
900	901	1500	1500	840	863
900	902	1500	1422		
900	902	1500	1420		

ค่าที่วัดได้	ค่าจริง	ค่าที่วัดได้	ค่าจริง	ค่าที่วัดได้	ค่าจริง
900	887	1440	1433		

$$\%Error = \left| \frac{\text{ค่าที่วัดได้} - \text{ค่าจริง}}{\text{ค่าจริง}} \right| \times 100\%$$

ค่าที่วัดได้ (ค่าเฉลี่ย) = 1140.967 รอบต่อนาที

ค่าจริงที่ได้จากเครื่องวัดรอบความเร็วแบบมาตรฐาน (ค่าเฉลี่ย) = 1148.695 รอบต่อนาที

$$\%Error = \left| \frac{1140.573 - 1148.36}{1148.36} \right| \times 100\%$$

%Error = 0.68 %

ดังนั้น ค่าเปอร์เซ็นต์ความผิดพลาด จากการเก็บค่าของอุปกรณ์ต้นแบบ เท่ากับ 0.68 เปอร์เซ็นต์ (%)

<u>ตารางที่ 3.5.2</u> การทดสอบส่งค่าความเร็วรอบแบบไร้สาย ที่ขึ้นระยะทางกับช่วงเวลา

ระยะทาง (m) / ช่วงเวลา	100	200	300	400	500	600	700	800	900	1,000	1,100
เช้ากลาง	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
กลางวัน	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
เย็น	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗

โดยที่ ✗ คือสัญญาณส่งไม่ถึงกัน

✓ คือสัญญาณส่งถึงกัน

***ค่าความเข้มแสงมีผลต่อการส่งสัญญาณ จึงทำให้ช่วงเวลาการส่งและรับค่าได้ในระยะที่ใกล้กว่าช่วงเวลาอื่นๆ

3.6 สรุปผลการทดลอง

ในบทที่ 3 เราได้ทำการทดลองวัดค่าความเร็วรอบจากอุปกรณ์ต้นแบบพร้อมทั้งใช้เครื่องมือวัดความเร็วรอบมาตรฐานในการทดสอบควบคู่กันไปด้วย เพื่อให้ผลการทดสอบออกมาตรงตามค่ามาตรฐานไม่ผิดเพี้ยนไปแต่อย่างใด ซึ่งค่าที่ได้จากการทดลองจากอุปกรณ์ต้นแบบนั้นตรงค่าที่ได้จากการใช้เครื่องมือวัดความเร็วรอบมาตรฐานโดยจำลองจากเฟืองเกียร์รถที่ใช้ในการทำงานจริง เราจึงได้ย่อขนาดเพื่อให้สะดวกต่อการทดสอบ ซึ่งการทดสอบนี้เราทำเพื่อช่วยให้ผู้ที่ต้องการทราบค่าความเร็วของรถ รู้ค่าได้โดยที่ตัวเองไม่ได้ไปขับรถคันเพียงแค่รถคันดังกล่าวติดอุปการณ์ที่เราได้ทำขึ้นมานี้โดยอุปกรณ์นี้ทำการส่งสัญญาณค่าความเร็วที่ได้จากรถ ส่งมายังตัวรับสัญญาณเพื่อที่จะแสดงผลให้ได้เห็นค่าความเร็วของรถคันดังกล่าว หรือถ้าเราอยากรู้ความเร็วต่อรอบแบบเป็นกราฟเราก็สามารถทำได้เช่นกัน ทั้งนี้เพื่ออำนวยความสะดวกในการดูรอบความเร็วรถแข่งในสนามเพื่อทำเวลาต่อรอบให้ได้ดีและทำให้เรารู้ข้อบงพร้องของความเร็วรถได้เพื่อที่แก้ไขในขั้นตอนต่อไปได้โดยเร็ว แหละยังสามารถประหยัดเวลาในการที่จะต้องมาเช็คความเร็วของรถอีกครั้งด้วย

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 4

ข้อสรุปของโครงงาน

4.1 บทนำ

เนื้อหาในบทนี้เป็นการกล่าวถึงบทสรุปของโครงงานเครื่องแสดงผลวัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1ซึ่งเนื้อหาในบทนี้แบ่งออกเป็นหัวข้อย่อยดังนี้

4.1 กล่าวนำ

4.2 บทสรุปของโครงงาน

4.3 ปัญหาที่พบในขณะดำเนินการ

4.4 ข้อเสนอแนะ

4.5 แนวทางในการพัฒนาต่อ

4.2 บทสรุปของโครงงาน

โครงงานเครื่องแสดงผลวัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1 ได้จัดทำขึ้นเพื่อส่งข้อมูลไร้สาย สามารถวัดความเร็วรอบ ส่งค่าที่วัดได้นอกจากคนขับที่รับรู้ว่ามีความเร็วรอบที่เท่าไรแล้ว บุคคลภายนอกรถยนต์จะรู้ว่ารถยนต์คันดังกล่าวที่วิ่งอยู่นั้น มีความเร็วรอบเครื่องยนต์ที่เท่าไร เพื่อนำข้อมูลที่ได้ไปทำการวิเคราะห์ พฤติกรรมการขับรถยนต์ของแต่ละบุคคลว่ามีลักษณะแตกต่างกันอย่างไร สู่การขับขี่ที่มีประสิทธิภาพในการแข่งขันให้ได้มากที่สุด ในเครื่องแสดงผลวัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1 นี้จะมีส่วนประกอบ 2 ส่วนด้วยกัน แต่ละส่วนมีหน้าที่การทำงานดังนี้ ส่วนแรกจะทำหน้าที่นับรอบการหมุนของเครื่องยนต์ คือตัวตรวจจับแบบแสง เป็นเซ็นเซอร์ IR Sensor TCRT5000 เมื่อได้ค่าจากเซ็นเซอร์ส่งเข้าสู่บอร์ด Arduino UNO R3 ประมวลผลที่ได้ส่งให้ Xbeeที่จะส่งค่าแบบไร้สาย ต่อไปยังส่วนที่สอง ในส่วนที่สองจะประกอบด้วย ชุดรับค่าความเร็วรอบเครื่องยนต์เป็น Xbeeอีกตัวโดยเชื่อมต่ออยู่กับคอมพิวเตอร์หรือแล็ปท็อปเพื่อแสดงค่าที่รับได้ออกทางหน้าจอทั้งเป็นค่าจริงและเส้นกราฟ จะใช้โปรแกรมทั้งหมด 3 โปรแกรม ได้แก่ Program

Arduino 1.0.5-r2 หน้าที่แสดงค่าจำนวนจริง Program Processing Version 2.0 หน้าแสดงค่าเป็นเส้นกราฟจำนวนจริง และ Program XCTU ช่วยในการปรับตั้งค่าของ Xbee ที่ใช้ส่งข้อมูลถึงกันได้ถูกต้อง โดยชุดเซ็นเซอร์ที่ทำงานนั้นจะมีค่าความผิดพลาด (%Error) = 0.58% การนำเซ็นเซอร์ไปติดตั้งกับรถยนต์บริเวณดังกล่าวต้องมีค่าความเข้มของแสงอยู่ในช่วง 0-3000 Lux. ถ้าความเข้มของแสงสว่างเกิน และเครื่องยนต์ควรไม่มีการรั่วไหลของไฟฟ้าในเครื่องยนต์ จะทำให้เซ็นเซอร์ตรวจวัดผิดเพี้ยน โครงงานเครื่องแสดงผลวัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1 ได้ทำอุปกรณ์ต้นแบบออกมา เป็นอุปกรณ์ที่สามารถส่งค่าความเร็วรอบเครื่องยนต์ ในการส่งสัญญาณไร้สายเราใช้ Xbee ในย่านความถี่ 2.4GHz.และมี RSMA 50mW. ทำหน้าที่ขยายสัญญาณให้แรงขึ้น สามารถรองรับการส่งสัญญาณที่ระยะทางได้เพิ่มขึ้น ทำให้ระบบวัดรอบเครื่องยนต์ นำมาแสดงผลได้ตามวัตถุประสงค์

4.3 ปัญหาที่พบในขณะดำเนินการ

ในการทำโครงงานเครื่องแสดงผลวัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1 ที่ปรากฏให้เห็นปัญหาต่างๆ ดังแสดงไว้ในตารางที่ 4.1 ซึ่งประกอบด้วยปัญหาที่พบขณะดำเนินการ สาเหตุของปัญหาและรวมทั้งวิธีการแก้ปัญหา

ตารางที่ 4.1 ปัญหาที่พบขณะดำเนินการ สาเหตุของปัญหาและรวมทั้งวิธีการแก้ปัญหา

ปัญหาที่พบขณะดำเนินการ	สาเหตุของปัญหา	วิธีการแก้ปัญหา
แบตเตอรี่หมดเร็ว	เกิดจากเซ็นเซอร์และตัวส่งไร้สาย Xbee จะใช้ไฟฟ้าจากบอร์ด Arduino ทั้งหมด โดย IR Sensor TCRT5000 ใช้กระแสไฟฟ้า 60mA. และตัว Xbee ใช้กระแสไฟฟ้า 295mA.	เปลี่ยนจากใช้ถ่าน 9 โวลต์ เป็นแบตเตอรี่ 12 โวลต์
เซ็นเซอร์ (Sensor)	ตอนแรกจะเป็นการใช้ IR Sensor แบบแยกตัวส่งกับตัวรับ	จึงเลือกใช้ IR Sensor TCRT5000 แทนที่มีทั้งตัวส่งและตัวรับอยู่ด้วยกัน

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปัญหาที่พบขณะดำเนินการ	สาเหตุของปัญหา	วิธีการแก้ปัญหา
ความเข้มของแสงสว่าง	ความเข้มของแสงต้องมีค่าอยู่ในช่วง 0-3000 Lux. แต่ถ้าเกินจะที่ให้เซ็นเซอร์ตรวจจับผิดเพี้ยนไป	ที่ติดตั้งเซ็นเซอร์ ต้องทำตัวครอบบริเวณดังกล่าว
คลื่นแม่เหล็กไฟฟ้า	ไฟฟ้าที่เกิดขึ้นในการทำงานของเครื่องยนต์ไม่สม่ำเสมอ ทำให้รบกวนอุปกรณ์ที่ติดตั้งกับเครื่องยนต์	โดยการทำกล่องโลหะใส่อุปกรณ์

4.4 ข้อเสนอแนะ

1. Arduino มีหลายรุ่นหลายราคา จะลองรับการทำงานได้เหมือนกันในพื้นฐานแต่จะต่างตามรุ่นราคาที่มีฟังก์ชันเสริมเพิ่มขึ้น
2. Xbee มีหลายรุ่นแต่ละรุ่น มีความถี่ที่ใช้ ระยะการส่งแตกต่างกัน ถ้าต้องการส่งสัญญาณที่เร็วและแรง แนะนำเป็น Xbee Pro S2B 2.4 GHz. RPSMA Connector
3. Sensor โดย IR Sensor TCRT5000 มีราคาต่ำและสามารถตรวจวัดที่ความเร็วรอบสูงๆ ได้ดี
4. Digital Tachometer Victor DM6234P+ ใช้งานง่าย มีความผิดพลาดน้อยมาก รองรับความเร็วรอบได้ถึง 99,999 รอบ และสามารถบอกค่าสูงสุด ต่ำสุด และค่าเฉลี่ย
5. Program Arduino 1.0.5-r2 การใช้งานต้องมีพื้นฐานจากภาษาซี ในเวลาทำงานจะมีรายละเอียดบางส่วนที่เกี่ยวกับคำสั่งเฉพาะโปรแกรม Arduino 1.0.5-r2
6. Program Processing 2.0 เน้นการใช้งานเกี่ยวกับการแสดงรูปภาพและเส้นกราฟ
7. Program XCTU เพื่อการปรับตั้งค่าของ Xbee ให้เชื่อมต่อกันได้ส่วนการตั้งค่าในแต่ละโหมดโดยอธิบายอย่างละเอียดตามบทที่ 3 แต่ถ้าผู้สนใจจะใช้งานควรศึกษาหาข้อมูลเพิ่มเติมเพื่อเหมาะสมกับการใช้งาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

4.5 แนวทางในการพัฒนาต่อ

เนื่องจากเครื่องแสดงผลวัดรอบเครื่องยนต์แบบไร้สายสำหรับรถแข่ง SUT F1 ที่มี Xbee เป็นตัวส่งสัญญาณไร้สายยังมีระยะใกล้ สามารถดูค่าความเร็วรอบไร้สายได้เพียงหน้าจอเดียว ในอนาคตควรเพิ่มอุปกรณ์ส่งไร้สายเป็นแบบ GPRS/GSM Module กับการทำเว็บไซต์ให้คนที่มีส่วนร่วมสามารถเข้าดูได้หลายๆ เครื่องพร้อมกัน หรือสามารถเพิ่มการวัดความเร็ว อุณหภูมิของเครื่องยนต์ และสามารถเพิ่มกราฟการแสดงผลในหน้าจอเดียวกัน

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก

Code Program Arduino

```
volatile int rpmcount = 0;
int rpm = 0;
int mapped;
int pin = 13;
unsigned long lastmillis = 0;
        void setup()
                {
Serial.begin(9600);
pinMode(pin,OUTPUT);
attachInterrupt(0, rpm_fan, FALLING);                   //Interrupt cero (0) is on pin two(2).
                }
void loop()
        {
int led = digitalRead(2);                               // Sensor detect.
        if(led==HIGH){
        digitalWrite(pin,HIGH);}
                else{
        digitalWrite(pin,LOW);                          // Not Sensor detect.
                        }
        // Serial.println("  rpm ");
                if (millis() - lastmillis == 1000);{    //* Update every one second, this will be
equal to reading frequency (Hz).
detachInterrupt(0);                                     //Disable interrupt when calculating.
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
rpm = ((rpmcount)/2) * 60;                          //Convert frequency  to RPM note: this
works for one interruption per full rotation. For two interrupt per full rotation use rpmcount * 30.
Serial.println(rpm);                                 // print the rpm value.
        mapped = map(rpm,0,15000,0,760);             //To processing.
        rpmcount = 0;                                // Restart the RPM counter.
        lastmillis = millis();                       // Update last millis.
        attachInterrupt(0, rpm_fan, FALLING);        //Enable interrupt.
        }
        }
void rpm_fan();{                                     // This code will be executed every time
the interrupt 0 (pin2) gets low.
        rpmcount++;
                }
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

Code Program Processing

```
import processing.serial.*;
Serial mapped;
int val, screen_increment, old_x = 0, old_y = 0;
String inString;
int lf = 10;
PImage img;
void setup() {
size(displayWidth-100,760);
println(Serial.list());
String portName = Serial.list()[0];
mapped = new Serial(this, portName, 9600);
mapped.bufferUntil(lf);
background(43,27,227);                      // Background Color  Blue
strokeWeight(2) ;
fill(0,0,0);
line(100,10,100,760) ;                      // Line axis x and axis y.
    textSize(20) ;                          // Set scale  text size
    text(0, 30, 755);    line(1265,755, 90,755) ;
    text(1000, 30, 705); line(1265,705, 90,705) ;
    text(2000, 30, 655); line(1265,655, 90,655) ;
    text(3000, 30, 605); line(1265,605, 90,605) ;
    text(4000, 30, 555); line(1265,555, 90,555) ;
    text(5000, 30, 505); line(1265,505, 90,505) ;
```

```
        text(6000, 30, 455); line(1265,455, 90,455) ;
        text(7000, 30, 405); line(1265,405, 90,405) ;
        text(8000, 30, 355); line(1265,355, 90,355) ;
        text(9000, 30, 305);  line(1265,305, 90,305) ;
        text(10000, 30, 255); line(1265,255, 90,255) ;
        text(11000, 30, 205); line(1265,205, 90,205) ;
        text(12000, 30, 155); line(1265,155, 90,155) ;
        text(13000, 30, 105); line(1265,105, 90,105) ;
        text(14000, 30, 55);  line(1265,55, 90,55) ;
        text(15000, 30, 15);   line(1265,5, 90,5) ;
}void draw(){
        }
        void serialEvent(Serial myport)
        {
        inString = myport.readString();
inString = trim(inString);
val = int(inString);
strokeWeight(5);
stroke(255,255,255);
        color c = color(255, 204, 0);
        fill(c);
        ellipse(displayWidth-220, 110, 220, 200);           // Ellipse RPM the real numbers.
        fill(50);
textSize(40);
        text(val,displayWidth-280, 125);                    // Text size RPM.
        textSize(30);
```

```
        text("RPM/min",displayWidth-280,180);
strokeWeight(5) ;
stroke(242,253,3);                                  // Background Color Yellow monitor.
line(old_x, old_y, screen_increment, (760-val/20)); //Plot of the rpm.
        old_x = screen_increment;
        old_y = (760-val/20);
screen_increment = screen_increment+30;
        strokeWeight(0);
        textSize(20);
if (screen_increment > displayWidth-100)
        { background(43,27,227);
strokeWeight(2) ;
fill(255,255,255);
line(100,10,100,760) ;
        textSize(20) ;
        text(0, 30, 755);   line(1265,755, 90,755) ;
        text(1000, 30, 705); line(1265,705, 90,705) ;
        text(2000, 30, 655); line(1265,655, 90,655) ;
        text(3000, 30, 605); line(1265,605, 90,605) ;
        text(4000, 30, 555); line(1265,555, 90,555) ;
        text(5000, 30, 505); line(1265,505, 90,505) ;
        text(6000, 30, 455); line(1265,455, 90,455) ;
        text(7000, 30, 405); line(1265,405, 90,405) ;
        text(8000, 30, 355); line(1265,355, 90,355) ;
        text(9000, 30, 305);  line(1265,305, 90,305) ;
        text(10000, 30, 255); line(1265,255, 90,255) ;
```

```
text(11000, 30, 205); line(1265,205, 90,205) ;
text(12000, 30, 155); line(1265,155, 90,155) ;
text(13000, 30, 105); line(1265,105, 90,105) ;
text(14000, 30, 55);  line(1265,55, 90,55) ;
text(15000, 30, 15);   line(1265,5, 90,5) ;
textSize(30);
text("RPM CAR F1 BY TCE.XIX ",500,45);         //Text name project on processing.
        textSize(20) ;
        strokeWeight(5) ;
screen_increment = -50 ;
old_x = -50 ;
old_y = 0 ;
 } }
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

Structure

Setup ()

The setup() function จะถูกเรียกเมื่อเริ่มต้น Sketch โปรแกรม ใช้เพื่อเริ่มต้นตัวแปร, Pin โหมด, เริ่มใช้ Libraries และอื่นๆ การตั้งค่าฟังก์ชั่นจะทำงานครั้งเดียวหลังจากที่แต่ละ powerup หรือการรีเซ็ทของบอร์ด Arduino

ตัวอย่าง

```
Int buttonPin = 3 ;

void setup()

{

        Serial.begin(9600) ;

        pinMode (buttonPin, INPUT) ;

}

void loop ( )

{

        // ...

}
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

Loop ()

หลังจากสร้างตัว setup() function ขึ้นมา,ซึ่งเป็นจุดเริ่มต้นและตั้งค่าตัวแปร, The loop() function ไม่ได้ทำงานแม่นยำอย่างที่ชื่อตัวมันบอก และ loop สามารถทำงานได้ต่อเนื่อง ทำให้โปรแกรมของคุณสามารถตอบสนองต่อการเปลี่ยนแปลง ใช้ loop() เพื่อควบคุมการทำงานของบอร์ด Arduino

ตัวอย่าง

```
const int buttonPin = 3;
// setup initializes serial and the button pin
void setup()
{
 Serial.begin(9600);
 pinMode(buttonPin, INPUT);
}
// loop checks the button pin each time,
// and will send serial if it is pressed
void loop()
{
 if (digitalRead(buttonPin) == HIGH)
   Serial.write('H');
 else
   Serial.write('L');
 delay(1000);
}
```

Control Struction

<u>if (conditional) and ==, !=, <, > (comparison operators)</u>

เป็นคำสั่งที่ใช้เปรียบเทียบการทดสอบว่าเงื่อนไขบางอย่างที่ได้รับ เช่น การป้อนข้อมูลเป็นจำนวนหนึ่ง รูปแบบการทดสอบคือ

<u>ตัวอย่าง</u>

```
if (someVariable > 50)
{
 // do something here
}
```

โปรแกรมทดสอบเพื่อดูว่า ค่า Some Variable มีค่ามากกว่า 50 ถ้าเป็นจริงโปรแกรมจะเริ่มดำเนินการ ถ้าเป็นเท็จโปรแกรมจะข้ามคำสั่งหรือโค้ดนี้ไปทำคำสั่งถัดไป

<u>ตัวอย่าง</u>

```
if (x > 120) digitalWrite(LEDpin, HIGH);

if (x > 120)
digitalWrite(LEDpin, HIGH);


if (x > 120){ digitalWrite(LEDpin, HIGH); }


if (x > 120){
 digitalWrite(LEDpin1, HIGH);
 digitalWrite(LEDpin2, HIGH);
}                    // all are correct
```

Statement ดังกล่าวต้องการจะใช้ 1 หรือมากกว่า 1 คำสั่ง

มหาวิทยาลัยเทคโนโลยีสุรนารี

กิตติกรรมประกาศ
(Acknowledgement)

โครงงานเครื่องขยายกำลังส่ง ฉบับนี้สำเร็จลุล่วงด้วยดี ผู้ทำโครงงานขอกราบขอบพระคุณ ผ.ศ.ดร.ชาญชัย ทองโสภา อาจารย์ที่ปรึกษาโครงงานที่กรุณาให้คำปรึกษา ชี้แนะ และให้การช่วยเหลือในการศึกษาโครงงานอย่างดีตลอดมา รวมถึงให้คำแนะนำในการเขียนและตรวจแก้โครงงานจนเสร็จสมบูรณ์

ขอบคุณเจ้าหน้าที่ประจำศูนย์เครื่องมือ 3 มหาวิทยาลัยเทคโนโลยีสุรนารี ทุกท่านที่สละเวลาให้ความช่วยเหลือ และอำนวยความสะดวกแก่ผู้ทำโครงงาน

ข้าพเจ้าใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่มีส่วนร่วมในการให้ข้อมูล และเป็นที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้การดูแลและให้ความเข้าใจเกี่ยวกับพื้นฐานการใช้งานโปรแกรม ซึ่งข้าพเจ้าขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

นายพิพัชรพล จอมพลาพล
นายนายอติชาต เรียวเรืองแสงกุล
นายอภิวัฒน์ บัวเบิก

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ (ต่อ)

เรื่อง หน้า

บทที่ 4 การออกแบบและสร้างวงจรขยายกำลังส่งสัญญาณคลื่นวิทยุ
ในย่านความถี่เอฟเอ็ม กำลัง 1000 วัตต์ 37
4.1 การออกแบบวงจรขยายสัญญาณวิทยุ ที่ความถี่ 98 เมกะเฮิรตซ์ 37
4.2 การสร้างวงจรขยายสัญญาณวิทยุ ที่ความถี่ 98 เมกะเฮิรตซ์ 62
บทที่ 5 ผลการทดลอง 70
5.1 การวัดผลการทดลองวงจรขยายกำลังส่งสัญญาณคลื่นวิทยุในย่านความถี่เอฟเอ็ม 70
5.2 สรุปผลการทดลอง 98
บทที่ 6 สรุปผลและข้อเสนอแนะ 99
6.1 สรุปผล 99
6.2 ปัญหาและข้อเสนอแนะ 100
6.3 แนวทางการพัฒนา 100
ภาคผนวก 101
ประวัติผู้เขียน 116
บรรณานุกรม 117

มหาวิทยาลัยเทคโนโลยีสุรนารี

เอกสารอ้างอิง

1. โปรแกรมที่ใช้ในการเขียนโค้ดลงบน Arduino. [ออนไลน์]

   1.1 https://www.arduino.cc/en/Main/Software

2. โปรแกรม XCTU ที่ใช้ในการเชื่อมต่อและการตั้งค่าของ Xbee. [ออนไลน์]

   2.1 http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu

3. โปรแกรม Processing ที่ใช้เขียนโค้ดเพื่อแสดงผลเป็นรูปแบบของกราฟบน laptop. [ออนไลน์]

   3.1 https://processing.org/download/

4. การเขียนโค้ดเบื้องต้นของโปรแกรม Proceesing. [ออนไลน์]

   4.1 https://processing.org/handbook/

   4.2 http://www.processing.lyndondaniels.com/17visual.php

5. คู่มือเกี่ยวกับ XBee. [ออนไลน์]

   5.1 http://thaieasyelec.com/article-wiki/embedded-electronics-application/learning-xbee-with-xbee-series-2-starter-kit.html

6. คู่มือการต่อ Zigbee and Xbee BASIC กับไมโครคอนโทรลเลอร์เบื้องต้น. [ออนไลน์]

   6.1 http://www.thaieasyelec.com/article-wiki/review-product-article/xbee-with-microcontroller-pic16f877.html

   6.2 http://examples.digi.com/get-started/basic-xbee-802-15-4-chat/

7. Datasheet 2.4GHz Duck Antenna 5 dBi with RP-SMA. [ออนไลน์]

   7.1 http://www.thaieasyelec.net/archives/Manual/EWLM025-TA040163.pdf

8. Datasheet IR Sensor TCRT5000 and TCRT5000L. [ออนไลน์]

   8.1 http://www.vishay.com/optical-sensors/list/product-83760/

9. Datasheet Victor DM6234P+ Digital Tachometer. [ออนไลน์]

   9.1 http://www.china-victor.com/english/en/product_data.aspx?ClassID=212&ID=246

ประวัติผู้เขียน

นางสาวหนึ่งฤทัย ปุสสวงษ์ เกิดเมื่อวันพฤหัสบดีที่ 7 มกราคม พ.ศ. 2536 ภูมิลำเนาอยู่ที่ตำบลโรงเข้ อำเภอบ้านแพ้ว จังหวัดสมุทรสาคร สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนสายธรรมจันทร์ อำเภอดำเนินสะดวก จังหวัดราชบุรี เมื่อปี พ.ศ. 2553 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี โทร 08-7416-6993

นายกษม สุขเกษม เกิดเมื่อวันศุกร์ที่ 30 ตุลาคม พ.ศ. 2535 ภูมิลำเนาอยู่ที่ตำบลวังหว้า อำเภอตะพานหิน จังหวัดพิจิตร สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนตะพานหิน อำเภอตะพานหิน จังหวัดพิจิตร เมื่อปี พ.ศ. 2553 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 3 สาขาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี โทร 08-2484-2869

มหาวิทยาลัยเทคโนโลยีสุรนารี

นายทินกร ไชยรักษา เกิดเมื่อวันศุกร์ที่ 10 กุมภาพันธ์ พ.ศ. 2535 ภูมิลำเนาอยู่ที่ตำบลนาแค อำเภอนายูง จังหวัดอุดรธานี สำเร็จการศึกษาระดับมัธยมศึกษาจากโรงเรียนยูงทองพิทยาคม อำเภอนายูง จังหวัดอุดรธานี เมื่อปี พ.ศ. 2553 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 5 สาขาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี โทร 09-0374-2418