

5. ผลการวิจัยของโครงการย่อยที่ 2

ผลการวิจัยจะสามารถแบ่งออกเป็นหัวข้อหลักๆได้ดังนี้

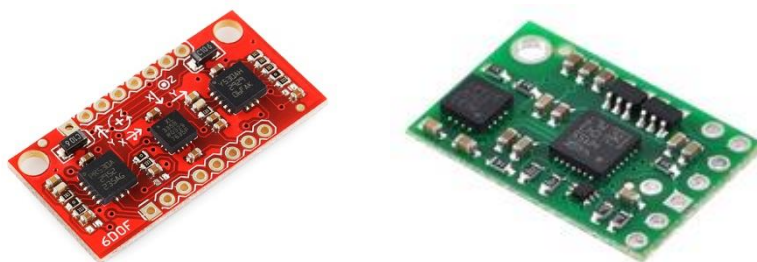
1. เพื่อพัฒนาระบบนำร่องอัตโนมัติของเรืออัตโนมัติ โดยใช้การบูรณาการของข้อมูลที่ได้จากเซนเซอร์ GPS และ IMU ที่ราคาไม่แพงหลายๆตัวเข้าด้วยกัน
2. เพื่อพัฒนาระบบตรวจวัดคุณภาพน้ำที่ราคาไม่แพงโดยใช้เซนเซอร์ชนิด ISFET เพื่อทดสอบความเป็นกรดต่าง (pH) ของน้ำ
3. การพัฒนาระบบตรวจวัดความลึกแหล่งน้ำโดยใช้เซนเซอร์ชนิด Echo Sounder ของ Garmin
4. การพัฒนาระบบส่งถ่ายข้อมูลผ่าน Internet โดยใช้บอร์ดไมโครคอนโทรลเลอร์ Netduino Plus หรือ บอร์ด Arduino Ethernet Shield

5.1 เพื่อพัฒนาระบบนำร่องอัตโนมัติของเรืออัตโนมัติ โดยใช้การบูรณาการของข้อมูลที่ได้จากเซนเซอร์ GPS และ IMU ที่ราคาไม่แพงหลายๆตัวเข้าด้วยกัน

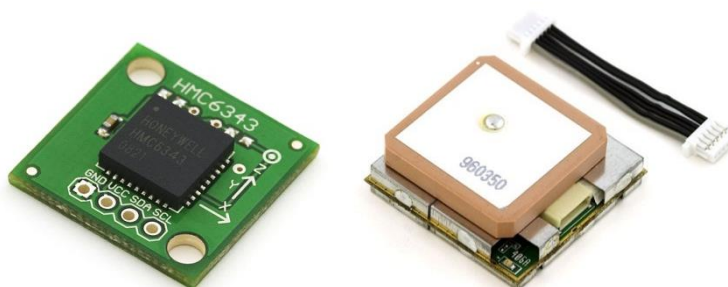
จุดประสงค์ของระบบนำร่องคือการประมาณตำแหน่งที่ดีที่สุดของเรือในทุกช่วงเวลา ดังนั้นเราจึงต้องการข้อมูลของการเคลื่อนที่ของเรือ ณ เวลาต่างๆกัน เช่น ความเร็ว, ทิศทางของหัวเรือ, ความเร่ง และ ตำแหน่งในระนาบบนพื้นผิวโลก

5.1.1 เซนเซอร์ที่ราคาไม่แพงชนิดต่างๆและคุณสมบัติที่นำมาประยุกต์ใช้ในงานวิจัยนี้

ข้อมูลที่ได้จากเซนเซอร์วัดมุมเอียงและเซนเซอร์วัดความเร่งในงานวิจัยนี้ จะมาจากบอร์ด IMU 6DOF Razor โดยที่มีความไว (Sensitivity) และช่วงการตอบสนอง (response range) ของเซนเซอร์วัดความเร่งเป็น 300 mV/g และ $\pm 3g$ ตามลำดับ โดยความถี่ของค่าที่วัดได้จาก IMU 6 DOF Razor [19] จะเป็น 100 Hz หรือ จะมาจากบอร์ด Pololu - MiniIMU-9 DOF Gyro, Accelerometer, and Compass โดยที่มีความไว (Sensitivity) และ ความละเอียดของสัญญาณเอาต์พุต (response range) สำหรับเซนเซอร์วัดความมุมเอียงเป็น $500^\circ/g$ และ 16-bit ตามลำดับ และ สำหรับเซนเซอร์วัดความเร่งเป็น $\pm 8g$ และ 12-bit ตามลำดับ โดยความถี่ของค่าที่วัดได้จาก Pololu - MiniIMU-9 DOF Gyro, Accelerometer, and Compass [20] จะเป็น 100 Hz ส่วนข้อมูลที่ได้จากเซนเซอร์ GPS จะมาจากบอร์ด 20 Channel EM-406 A SiRF III Receiver with Antenna [22] โดยที่สามารถรับสัญญาณดาวเทียมได้ภายใน 42 วินาที หลังจากเริ่มเปิดเครื่องเมื่อไม่ได้ใช้งานนานๆ (Cold Start) และรับสัญญาณดาวเทียมได้ภายใน 1 วินาที หลังจากปิดและเปิดเครื่องใหม่โดยทันที (Hot Start) โดยที่มีความแม่นยำในการวัดค่าตำแหน่งหรือพิกัดในระนาบเป็นระยะ 10 เมตร ส่วนข้อมูลที่ได้จากเซนเซอร์เข็มทิศ จะมาจากบอร์ด Compass Module with Tilt Compensation HMC6343 [21] ซึ่งจะให้ความแม่นยำในการวัดองศาในแต่ละแกนเป็น 2 องศา โดยจะมีค่าแก้ไของศาการเอียงในแนวราบ



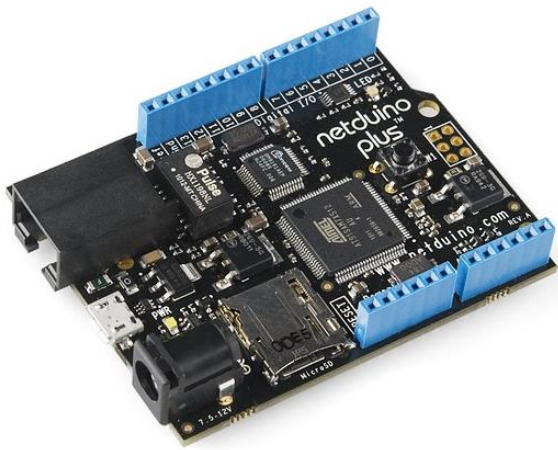
รูปที่ 5.1 เซนเซอร์วัดมุมเอียงและเซนเซอร์วัดความเร่ง (IMU 6DOF Razor) [19] (ซ้ายมือ), เซนเซอร์วัดมุมเอียงและเซนเซอร์วัดความเร่งและเข็มทิศ (Pololu - MiniIMU-9 DOF Gyro, Accelerometer, and Compass) [20] (ขวามือ)



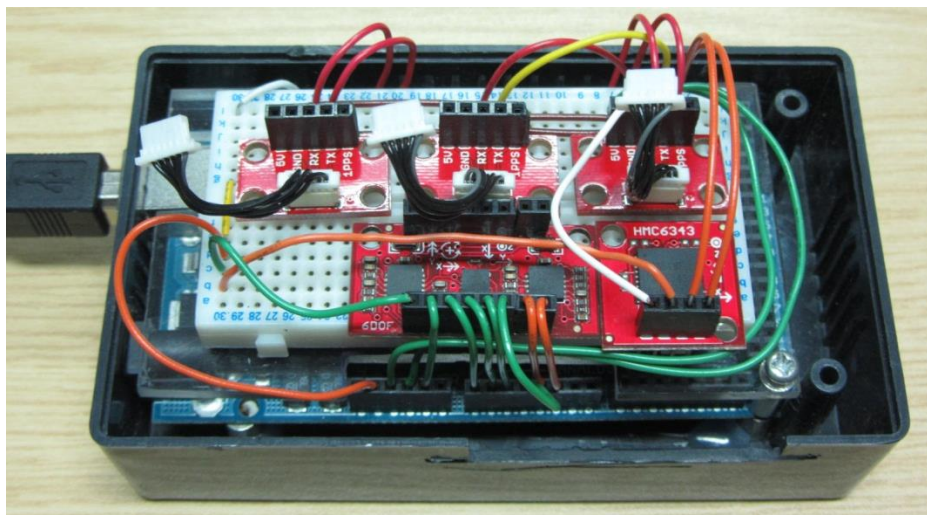
รูปที่ 5.2 เซนเซอร์เข็มทิศ (Digital Compass HMC6343) [21] (ซ้ายมือ), เซนเซอร์ GPS (EM-406 A SiRF III Receiver) [22] (ขวามือ)

5.1.2 การเชื่อมโยงข้อมูลจากเซนเซอร์ชนิดต่างๆ โดยใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino

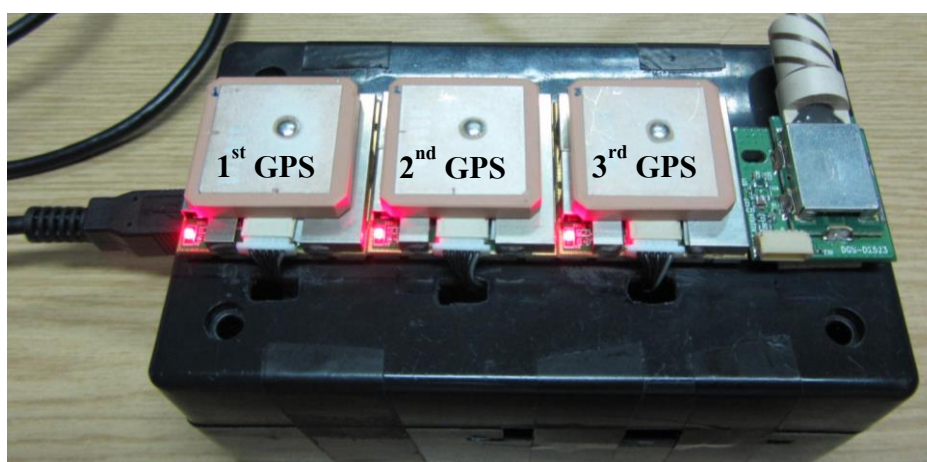
ในการบูรณาการข้อมูลจากเซนเซอร์ทั้งหมดจะเชื่อมต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ Arduino รุ่น MEGA 2560 [16] ดังแสดงในรูปที่ 5.3 ซึ่งมีช่องรับสัญญาณแบบทั้ง analog และ digital จำนวนหลาย pin มาก และมีช่องรับสัญญาณ Serial (Tx,Rx) ทั้งหมด 4 ช่อง โดยที่สัญญาณที่ได้จากเซนเซอร์วัดมุมเอียงและเซนเซอร์วัดความเร่งจะต่อเข้าช่องแบบ analog จำนวน 6 ช่อง ส่วนสัญญาณที่ได้รับจากเซนเซอร์เข็มทิศแบบดิจิตอลจะต่อเข้าช่องแบบ I²C และ สัญญาณที่ได้รับจากเซนเซอร์ GPS EM-406 จะต่อเข้าช่อง Serial จำนวน 3 ช่อง

 รูปที่ 5.3 บอร์ด Arduino รุ่น MEGA 2560 [16]	 รูปที่ 5.4 บอร์ด Netduino Plus [17]
คุณสมบัติของบอร์ด Arduino รุ่น MEGA 2560 ● ใช้ ATMEGA2560 เป็นไมโครคอนโทรลเลอร์ประจำบอร์ด ● แรงดันไฟฟ้า 5 V แรงดันไฟฟ้า Input ที่ควรใช้ 7-12 V ● แรงดันไฟฟ้า Input จำกัดที่ 6-20 V ● Pins ดิจิตอล I/O Input 54 ● Pins อนาล็อก Input 16 ● ไฟฟ้ากระแสตรงต่อ I/O Pin 40 mA ● ไฟฟ้ากระแสตรงสำหรับ Pin 3.3 V 50 mA ● หน่วยความจำแฟลช 256 กิโลไบต์ (ที่ 8 KB ที่ใช้โดย Bootloader) SRAM 8 KB และ EEPROM 4 KB ● ความเร็วสัญญาณนาฬิกา (Clock Speed) 16 MHz	คุณสมบัติของบอร์ด Netduino Plus ● เป็นไมโครคอนโทรลเลอร์ 32 บิต ● ความเร็ว 48 MHz, ARM7. Code Storage: 128KB. RAM: 60KB. ● รองรับ Micro SD card มากกว่า 2 GB (SD via SPI, FAT16/FAT32) ● Netduino Plus is 100% open source - networking is lwIP (BSD license) and file system/sd card support is from Microsoft (Apache 2.0). ● USB Cable Included ● 14 Digital I/O Pins ● 6 Analog Inputs ● 100mbps ethernet ● Power specifications: Input: 7.5 - 12.0 VDC or USB powered. Output: 5 VDC and 3.3 VDC regulated.

รูปที่ 5.3 และ 5.4 แสดงการเชื่อมต่อของเซนเซอร์ทุกตัวเข้ากับบอร์ดไมโครคอนโทรลเลอร์ Arduino MEGA ที่อยู่ทางด้านล่าง โดยมีเซนเซอร์วัดมุมเอียงและเซนเซอร์วัดความเร่ง และ เซนเซอร์เข็มทิศ ติดตั้งอยู่บนในกล่อง และมีเซนเซอร์ GPS ติดอยู่ทางด้านบนของกล่องระบบนำร่อง



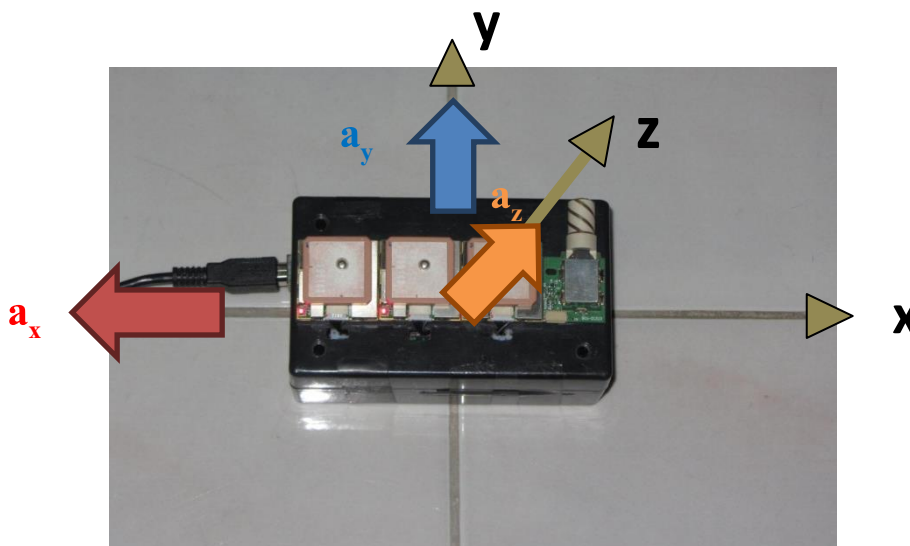
รูปที่ 5.5 เซนเซอร์วัดมุมเอียงและเซนเซอร์วัดความเร่งและเข็มทิศ ที่ติดตั้งภายในกล่องระบบนำร่อง



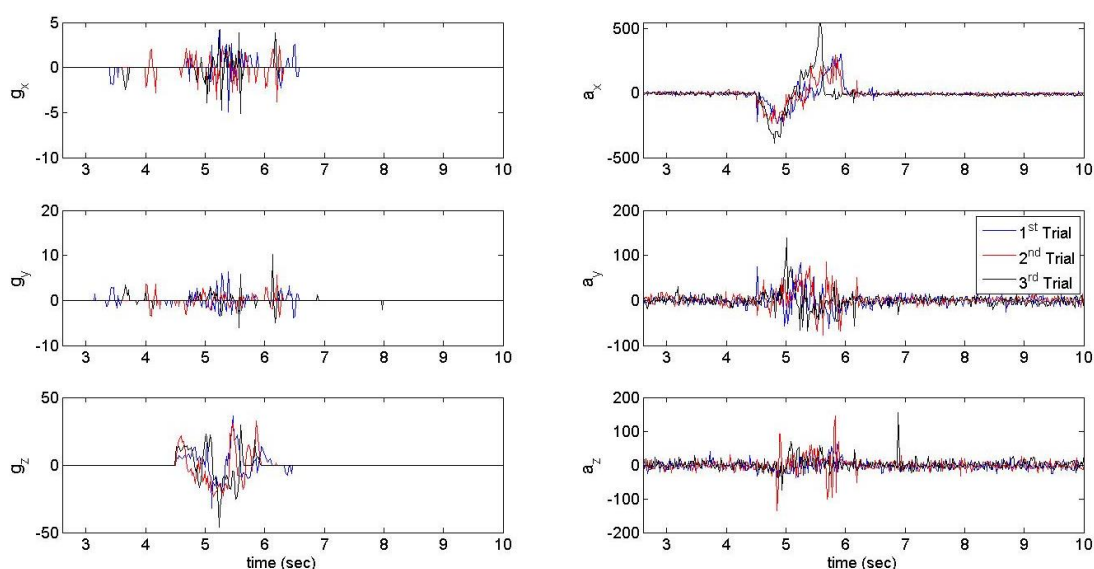
รูปที่ 5.6 เซนเซอร์ GPS EM-406 A สามตัว ที่ติดตั้งภายนอกกล่องระบบนำร่อง

5.1.3 การ calibrate ข้อมูลจากเซนเซอร์ที่ได้จากเซนเซอร์วัดความเร่ง (accelerometer) และวัดความเร็วเชิงมุม (gyroscope) เพื่อให้ได้อัตราขยายที่ถูกต้อง

เพื่อทดสอบความแม่นยำของเซนเซอร์วัดมุมเอียงและวัดค่าความเร่ง ว่าข้อมูลที่ได้มีความถูกต้องมากน้อยเพียงใดได้ทำการทดสอบเพื่อ calibrate ค่าความเร็วเชิงมุม และ ค่าความเร่งในแต่ละแกนจากเซนเซอร์ Pololu MiniIMU ในการ calibrate ค่าความเร่งในแนวแกน x, y และ z ดังแสดงในรูปที่ 5.7 จะทำการเคลื่อนที่ในแนวเส้นตรงเป็นระยะ 70 cm โดยเริ่มออกตัวจากจุดศูนย์แล้วหยุดที่ระยะ 70 cm ผลที่ได้จากการ calibrate ความเร่งในแกน x (a_x) แสดงในรูป 5.8 มีค่าเป็นลบในช่วงเริ่มออกตัว แล้วจึงมีค่าเป็นบวกเนื่องจากความหน่วงในตอนหยุด เมื่อทำการหาปริพันธ์ในช่วงเวลาที่ a_x ติดลบ จะได้เป็นความเร็วเฉลี่ยในการเคลื่อนที่ แล้วจึงนำมาหารกับความเร่งเฉลี่ยจากการหารด้วยเวลาที่เฉลี่ยจากการทดลอง 3 ครั้ง จะได้อัตราส่วนเป็น 0.495, 0.442 และ 0.359 เพราะฉะนั้นค่าความเร่งในแนวแกน x ที่วัดได้ ควรจะต้องคูณด้วยค่าอัตราขยาย 0.432 จึงจะได้ค่าความเร่งในแกน x ที่ถูกต้อง

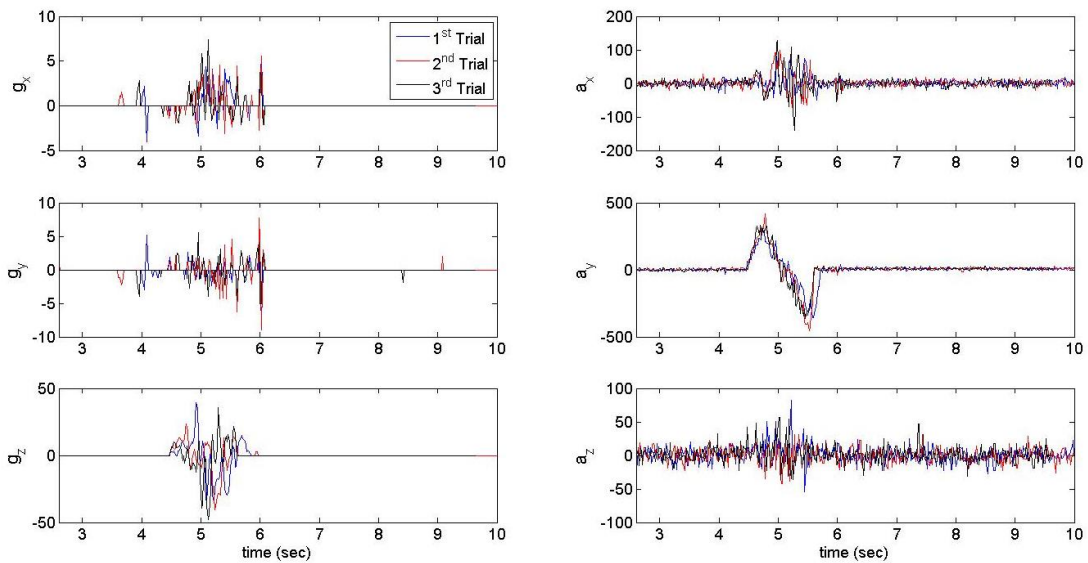


รูปที่ 5.7 การทดสอบการ calibration ค่าความเร่งในแนวแกน x, y, z เป็นจำนวน 3 ครั้ง/แกน

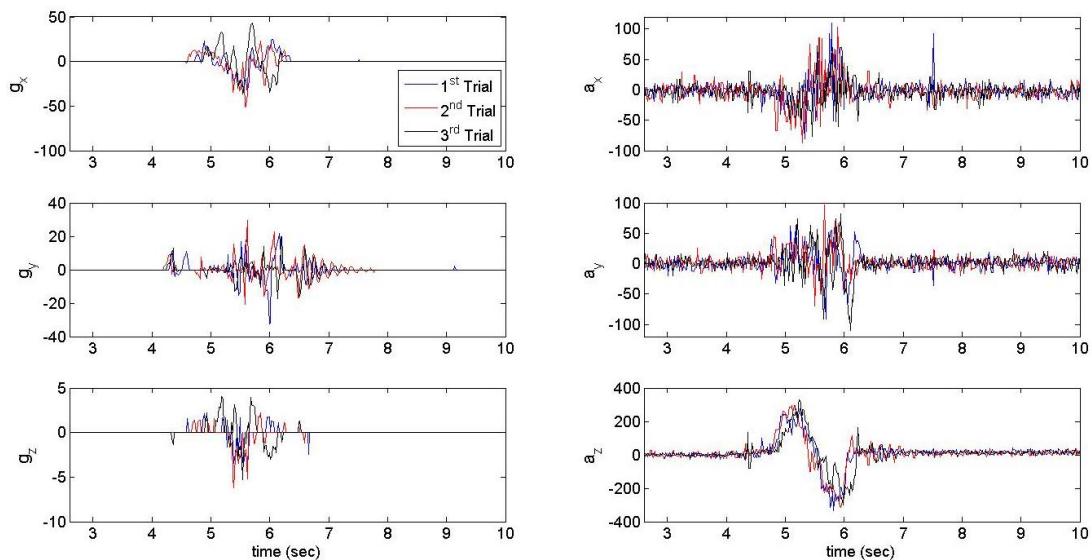


รูปที่ 5.8 : การ Calibrate ค่าความเร่งในแกน x ของ Polulu IMU จากการทดลอง 3 ครั้ง

ผลที่ได้จากการ calibrate ความเร่งในแกน y (a_y) แสดงในรูป 5.9 มีค่าเป็นบวกในช่วงเริ่มออกตัว แล้วจึงมีค่าเป็นลบเนื่องจากความหน่วงในตอนหยุด เมื่อทำการหาปริพันธ์ในช่วงเวลาที่ a_y เป็นบวก จะได้เป็นความเร็วเฉลี่ยในการเคลื่อนที่ แล้วจึงนำมารวบรวมกับความเร่งเฉลี่ยจากการหารด้วยเวลาที่เฉลี่ยจากการทดลอง 3 ครั้ง จะได้อัตราส่วนเป็น 0.418, 0.379 และ 0.359 เพราะฉะนั้นค่าความเร่งในแนวแกน y ที่วัดได้ ควรจะต้องคูณด้วยค่าอัตราขยาย 0.386 จึงจะได้ค่าความเร่งในแกน y ที่ถูกต้อง



รูปที่ 5.9 : การ Calibrate ค่าความเร่งในแกน y ของ Polulu IMU จากการทดลอง 3 ครั้ง

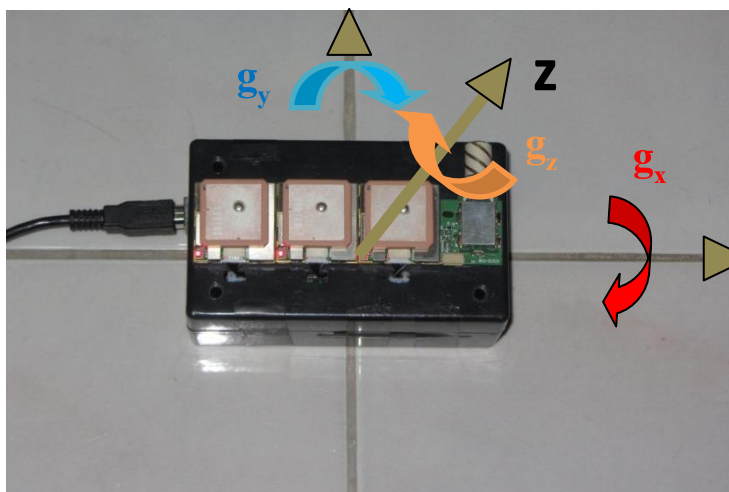


รูปที่ 5.10 : การ Calibrate ค่าความเร่งในแกน z ของ Polulu IMU จากการทดลอง 3 ครั้ง

ในทำนองเดียวกัน ผลที่ได้จากการ calibrate ความเร่งในแกน z (a_z) แสดงในรูป 5.10 มีค่าเป็นบวกในช่วงเริ่มออกตัว แล้วจึงมีค่าเป็นลบเนื่องจากความหน่วงในตอนหยุด เมื่อทำการหาปริพันธ์ในช่วงเวลาที่ a_z เป็นบวก จะได้เป็นความเร็วเฉลี่ยในการเคลื่อนที่ แล้วจึงนำมารวมกับความเร็วเฉลี่ยจากการหารด้วยเวลาที่เฉลี่ยจากการทดลอง 3 ครั้ง จะได้อัตราส่วนเป็น 0.336, 0.330 และ 0.340 เพราะฉะนั้นค่าความเร่งในแนวแกน z ที่วัดได้ ควรจะต้องคูณด้วยค่าอัตราขยาย 0.335 จึงจะได้ค่าความเร่งในแกน z ที่ถูกต้อง

เพื่อทำการ calibrate ค่าความเร็วเชิงมุมรอบแกน x, y และ z ดังแสดงในรูปที่ 5.8 จะทำการหมุนรอบแต่ละแกนเป็นมุม 90° โดยเริ่มจากหยุดนิ่งและทำการหมุนรอบแต่ละแกนด้วยความเร็วเชิงมุมคงที่ ในการทดสอบค่าความเร็วเชิงมุมในแกน x (g_z) ดังแสดงในรูป 5.0 มีค่าเป็นบวกในช่วงเริ่มออกตัว แล้วจึงมีค่าลดลงจนเป็นศูนย์ในตอนท้ายเมื่อหยุด เมื่อทำการหาปริพันธ์ในช่วงเวลาที่ g_z เป็นบวก จะได้เป็นองศาการหมุนเฉลี่ย แล้วจึงนำค่าที่ได้มาหารกับองศาการหมุน $= 90^\circ$ จากการทดลอง 3 ครั้ง จะได้อัตราส่วนเป็น 1.287,

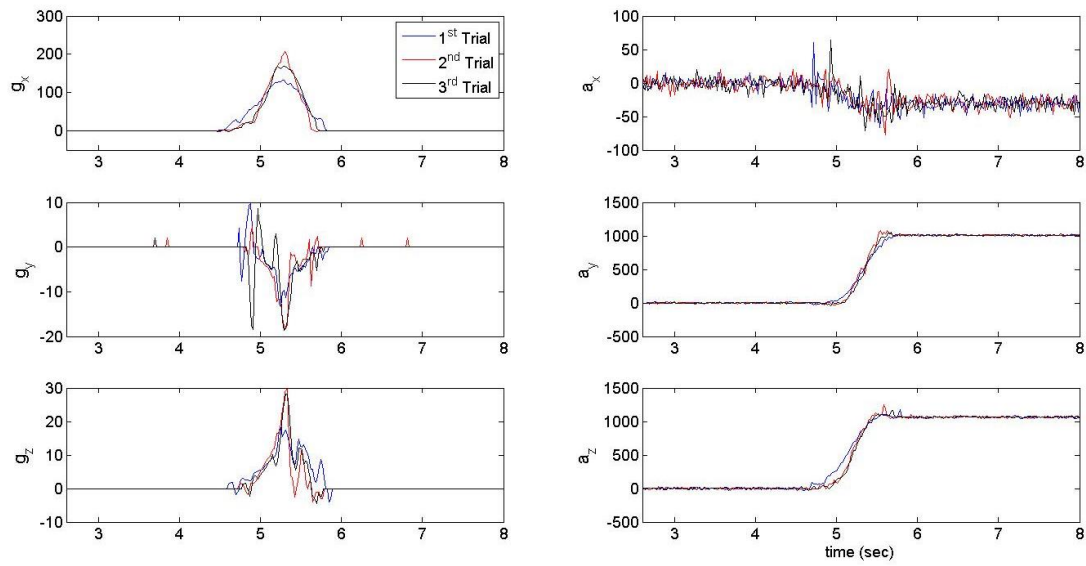
1.194 และ 1.235 เพราะฉะนั้นค่าความเร่งในแนวแกน z ที่วัดได้ ควรจะต้องคูณด้วยค่าอัตราขยาย 1.239 จึงจะได้ค่าความเร็วเชิงมุมรอบแกน x ที่ถูกต้อง



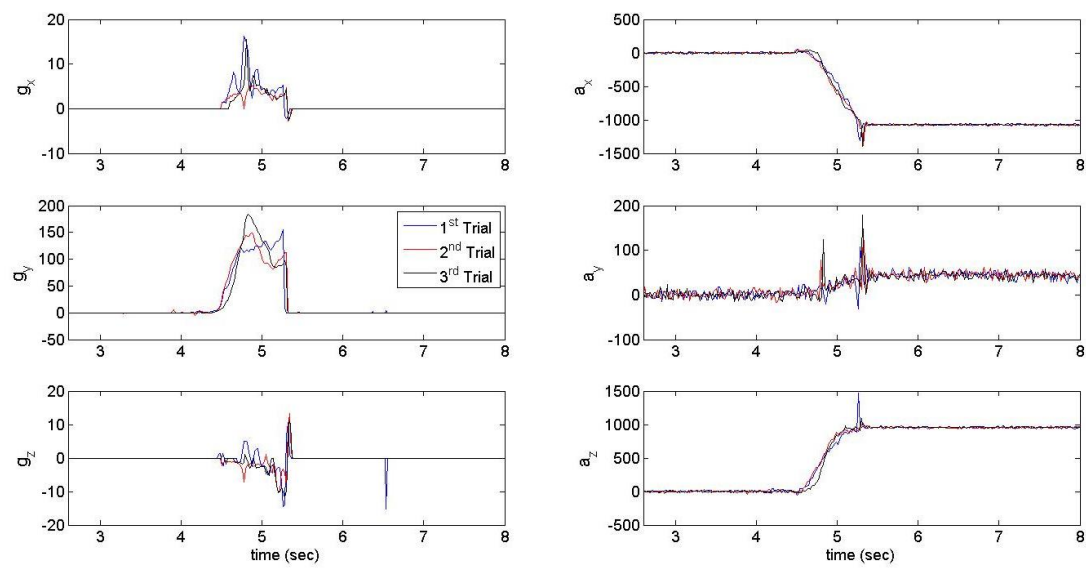
รูปที่ 5.11 การทดสอบการ calibration ค่าความเร็วเชิงมุมรอบแกน x, y, z เป็นจำนวน 3 ครั้ง/แกน

เพื่อทดสอบหาอัตราขยายของค่าความเร็วเชิงมุมในแกน y (g_y) ดังแสดงในรูป 4.44 มีค่าเป็นบวก ในช่วงเริ่มออกตัว แล้วจึงมีค่าลดลงจนเป็นศูนย์ในตอนท้ายเมื่อหยุด เมื่อทำการหาปริพันธ์ในช่วงเวลาที่ g_y เป็นบวก จะได้เป็นองศาการหมุนเฉลี่ย แล้วจึงนำค่าที่ได้มาหารกับองศาการหมุน=90° จากการทดลอง 3 ครั้ง จะได้อัตราส่วนเป็น 0.952, 0.952 และ 0.992 เพราะฉะนั้นค่าความเร่งในแนวแกน y ที่วัดได้ ควรจะต้องคูณด้วยค่าอัตราขยาย 0.965 จึงจะได้ค่าความเร็วเชิงมุมรอบแกน y ที่ถูกต้อง

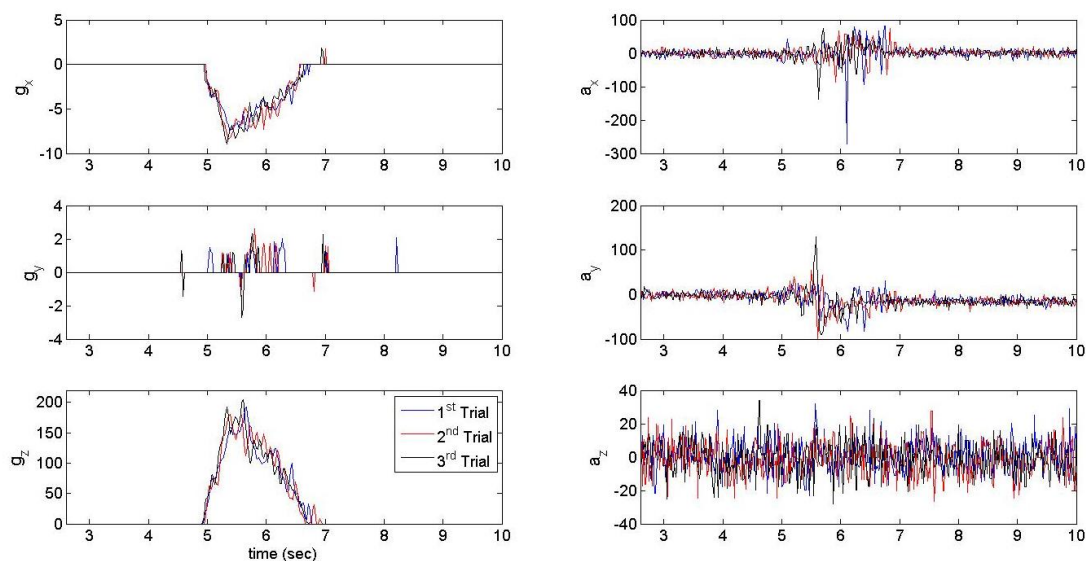
เพื่อทดสอบหาอัตราขยายของค่าความเร็วเชิงมุมในแกน z (g_z) ดังแสดงในรูป 4.45 มีค่าเป็นบวก ในช่วงเริ่มออกตัว แล้วจึงมีค่าลดลงจนเป็นศูนย์ในตอนท้ายเมื่อหยุด เมื่อทำการหาปริพันธ์ในช่วงเวลาที่ g_z เป็นบวก จะได้เป็นองศาการหมุนเฉลี่ย แล้วจึงนำค่าที่ได้มาหารกับองศาการหมุน=90° จากการทดลอง 3 ครั้ง จะได้อัตราส่วนเป็น 1.786, 1.893 และ 1.690 เพราะฉะนั้นค่าความเร่งในแนวแกน z ที่วัดได้ ควรจะต้องคูณด้วยค่าอัตราขยาย 1.789 จึงจะได้ค่าความเร็วเชิงมุมรอบแกน z ที่ถูกต้อง



รูปที่ 5.12 : การ Calibrate ค่าความเร็วเชิงมุมรอบแกน x ของ Polulu IMU จากการทดลอง 3 ครั้ง



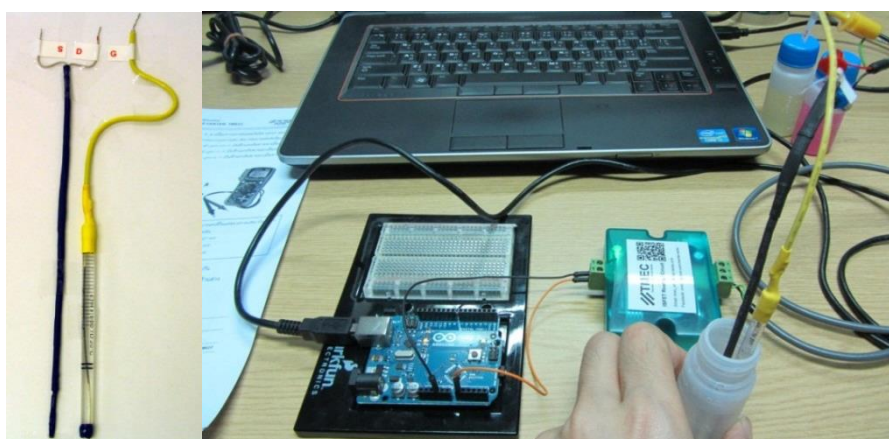
รูปที่ 5.13: การ Calibrate ค่าความเร็วเชิงมุมรอบแกน y ของ Polulu IMU จากการทดลอง 3 ครั้ง



รูปที่ 5.14 : การ Calibrate ค่าความเร็วเชิงมุมรอบแกน z ของ Polulu IMU จากการทดลอง 3 ครั้ง

5.2. การพัฒนาระบบตรวจวัดคุณภาพน้ำที่ราคาไม่แพงโดยใช้เซนเซอร์ชนิด ISFET เพื่อทดสอบความเป็นกรดต่าง (pH) ของน้ำ

เซนเซอร์วัดคุณภาพน้ำรุ่น ISFET ฟีเอช เซนเซอร์ [15] ซึ่งผลิตในประเทศไทย ทางคณะวิศวกรรมศาสตร์ มหาวิทยาลัยราชภัฏวชิรสุพรรณบุรีได้รับการบริจาคจากศูนย์เทคโนโลยีไมโครอิเล็กทรอนิกส์ (TMEC) และ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) เพื่อนำมาใช้วัดคุณภาพน้ำ ซึ่งจะช่วยให้ประหยัดต้นทุนในการซื้อเครื่องมือวัดคุณภาพน้ำจากต่างประเทศ ความไว (Sensitivity) ในการวัดค่าความเป็นกรดเป็นด่าง จะเป็น $\text{Si}_3\text{N}_4 > 45 \text{ mV/pH}$, ช่วงของค่าความเป็นกรดเป็นด่างที่สามารถวัดได้ (Linear range) 1 – 14, ช่วงอุณหภูมิที่สามารถทำงานได้ (Operating Temp) 5°C ถึง 100°C , ค่าสัมประสิทธิ์ของการเปลี่ยนแปลงตามอุณหภูมิ (Temperature coefficient) -1.76 mV/oC (เมื่อ $\text{pH} = 7$) ค่าพีเอชสามารถอ่านได้โดยการเชื่อมต่อหัววัดผ่านวงจรอ่านค่าและส่งข้อมูลผ่านไปยังช่องสัญญาณแบบ analog ของบอร์ดไมโครคอนโทรลเลอร์ Arduino ดังแสดงในรูปที่ 4.66 นอกจากนั้นได้พัฒนาโปรแกรมการแสดงผลค่าพีเอชที่อ่านได้บนหน้าจอคอมพิวเตอร์แบบ real time อีกด้วย

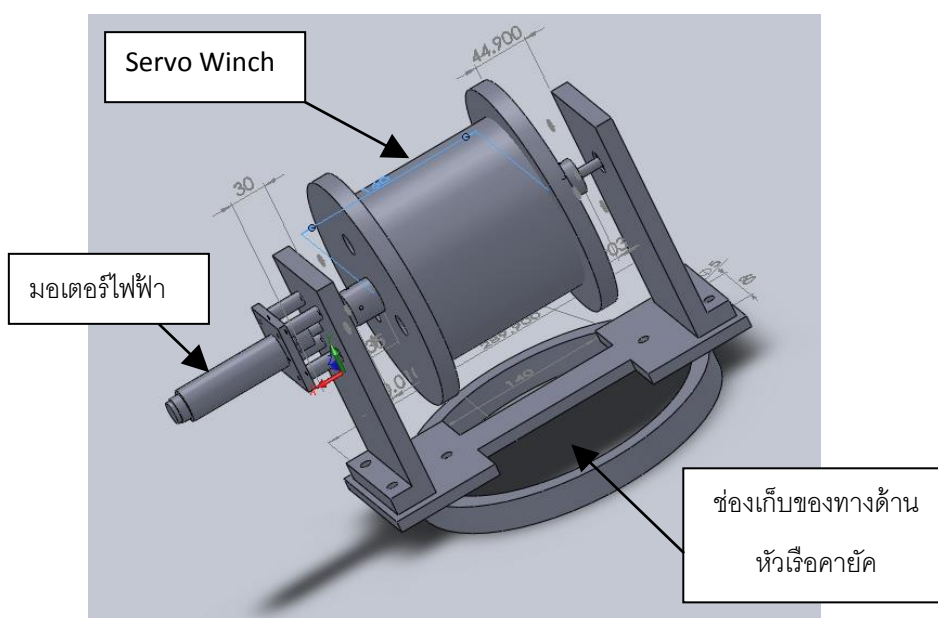


รูปที่ 5.15 : หัววัดค่า pH (ด้านซ้าย) เชื่อมต่อเข้ากับวงจรอ่านค่าในกล่องสี่เหลี่ยมซึ่งเชื่อมไปยังไมโครคอนโทรลเลอร์ Arduino (ด้านขวา)



รูปที่ 5.16 : โปรแกรมที่ใช้ในการบ่งบอกค่าพีเอชของน้ำที่ได้ถูกพัฒนาขึ้นด้วยโปรแกรม Processing ที่เชื่อมต่อข้อมูลมาจาก Arduino microcontroller

เพื่อที่หย่อนหัววัดคุณภาพน้ำลงไปยังระดับความลึกที่ต้องการนั้น ได้ทำการออกแบบระบบ servo winch ที่ขับเคลื่อนด้วยมอเตอร์ไฟฟ้าที่จะนำมาติดตั้งกับช่องที่เก็บของด้านหัวเรือคายัคซึ่งจะไว้ใช้เก็บอุปกรณ์อิเล็กทรอนิกส์ได้อีกด้วย โดยมีเส้นผ่านศูนย์กลางของตัว winch = 15 cm ฉะนั้นในการเก็บสายหัววัดคุณภาพน้ำที่มีเส้นผ่านศูนย์กลาง 10 cm ที่ยาว 6 เมตร จะต้องม้วนสายรอบ winch ประมาณ 13 รอบ



รูปที่ 5.17 ระบบ servo winch ที่จะใช้ในการหมุนเก็บสายของหัววัดคุณภาพน้ำที่ยาว 6 เมตร

5.3 การพัฒนาระบบตรวจวัดความลึกแหล่งน้ำโดยใช้เซนเซอร์ชนิด Echo Sounder ของ Garmin

นอกจากนั้นเรือคายัคยังสามารถติดอุปกรณ์เสริมที่สามารถติดตั้งได้อีกชนิด คือ หัววัดความลึกท้องน้ำ (echo sounder) รุ่น Garmin 421s [14] Garmin 421s เป็นเครื่องจีพีเอสที่มาพร้อมกับ หัวขานเดอร์ของ Airmar จากสหรัฐอเมริกาแบบสองความถี่สามารถแสดงลักษณะพื้นทะเลและปลาได้ชัดเจน ไม่ว่าจะใช้ในทะเลหรือน้ำจืด โดยวัดความลึกได้ถึง 1500 ฟุต/457 เมตร (สำหรับรุ่นสองความถี่) สำหรับเรือที่ออกแบบมาให้ใช้ได้สะดวก มีขนาดกะทัดรัด ประหยัดพื้นที่ที่มีจำกัดบนเรือ มีเมนูแบบใหม่ใช้งานง่าย นอกจากนี้ยังมีจุดเด่นเรื่อง การแสดงแผนที่แบบ 3-มิติ หากมีเกาะหรือแผ่นดินอยู่ใกล้แผนที่จะแสดงภาพเกาะหรือแผ่นดินส่วนนั้นเสมือนจริง มีการแสดงระดับความลึกและร่องน้ำที่ดูได้ชัดและง่ายดาย มี หน้าจอสีขนาด 4 นิ้ว แบบ QVGA ที่เห็นได้ชัดแม้ดูกลางแดด

ในงานวิจัยนี้ได้ติดตั้งหัววัดความลึกท้องน้ำไว้ทางด้านหัวเรือคายัค โดยออกแบบให้สามารถปรับระดับได้อย่างโดยใช้ลูมิเนียมโปรไฟล์ ดังแสดงในรูปที่ 5.18 หัววัดความลึกท้องน้ำสามารถติดต่อกับคอมพิวเตอร์ประจำเรือเพื่อการรับส่งข้อมูลผ่านเครือข่าย WLAN (wireless local area network) หรือเครือข่ายการสื่อสารแบบ 3G ได้อีกด้วย โดยผู้ใช้งานจะต้องคำนึงถึงน้ำหนักบรรทุกรวมของเรือที่ติดตั้งอุปกรณ์เสริมทั้งหมดไม่ควรเกิน 100 กิโลกรัม



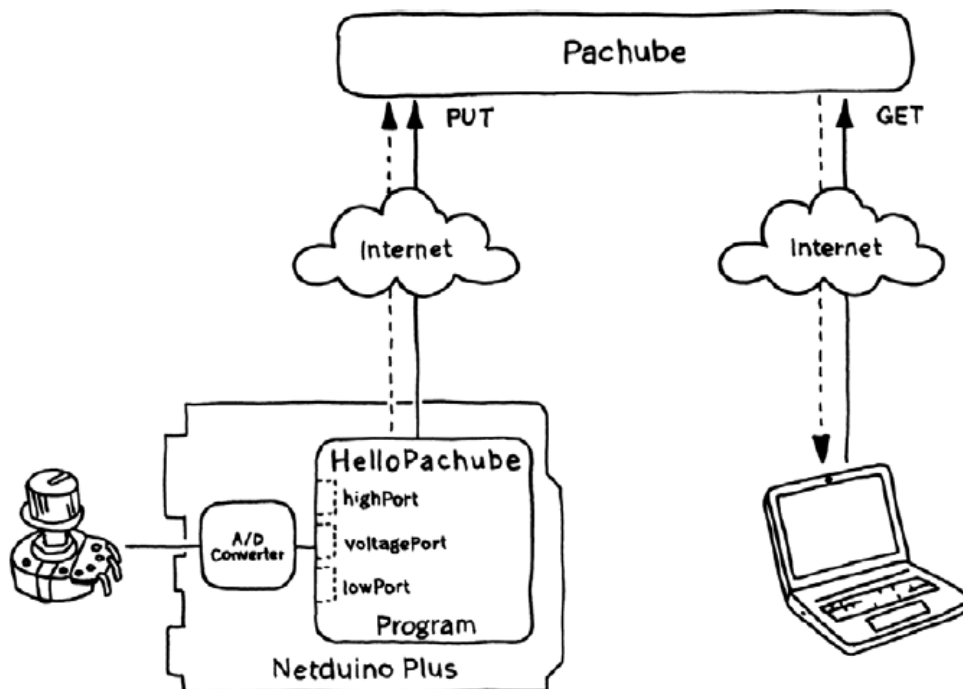
รูปที่ 5.18 หัววัดความลึกท้องน้ำและการติดตั้งทางด้านหัวเรือคายัคเมื่อนำเรือออกปฏิบัติการ (ซ้าย) และ เมื่อนำเรือเข้าเทียบท่า (ขวา)

5.4 การพัฒนาระบบส่งถ่ายข้อมูลผ่าน Internet โดยใช้บอร์ดไมโครคอนโทรเลอร์ Netduino Plus หรือบอร์ด Arduino Ethernet Shield

Netduino Plus เป็นบอร์ดไมโครคอนโทรเลอร์อิเล็กทรอนิกส์ที่มีรูปแบบการใช้งานคล้ายกับบอร์ดไมโครคอนโทรเลอร์แต่มีความสามารถพิเศษในการส่งข้อมูลผ่านอินเทอร์เน็ตแบบ Real-time โดยจะใช้โปรแกรม .NET Micro Framework ที่ใช้งานร่วมกับโปรแกรม Visual Studio 2010 และ การทำงานเชื่อมต่อกับอินเทอร์เน็ต

การทำงานของบอร์ด Netduino Plus กับโปรแกรม Visual Studio 2010 ร่วมกับ Website Pachube [23] ซึ่งเป็น Web Server แบบ Real-time Feed โดย Website ของ Pachube นั้นเราจะต้องมีการตั้งค่า ID เพื่อที่จะให้สามารถส่งข้อมูลจาก บอร์ด Netduino Plus ไปยังเว็บ Pachube และสามารถเรียก

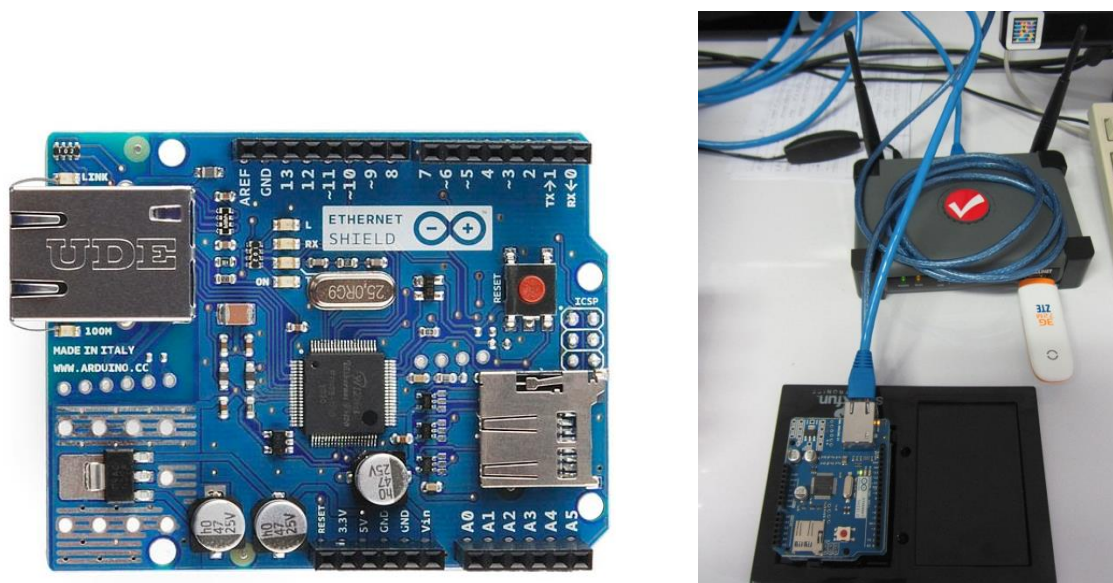
ข้อมูลเหล่านี้ไปยังคอมพิวเตอร์ หรือ บอร์ด Netduino Plus ตัวอื่นที่เชื่อมต่อกับอินเทอร์เน็ตได้ ดังตัวอย่างที่แสดงในรูปที่ 5.19 จากรูป จะเห็นการทำงานของ HTTP Client , Netduino ด้วยโปรแกรม HelloPachube ที่จะส่งข้อมูลไปยังเว็บ Pachube



รูปที่ 5.19 : ตัวอย่างการเชื่อมต่อของบอร์ด Netduino เพื่อส่งข้อมูลไปยัง Pachube Web Server และคอมพิวเตอร์สามารถเรียกดูข้อมูลได้แบบ Real time ผ่านอินเทอร์เน็ต

อีกวิธีหนึ่งในการส่งข้อมูลผ่านทางอินเทอร์เน็ตที่ทำงานคล้ายกับบอร์ด Netduino คือการใช้ Arduino Ethernet Shield [18] ดังแสดงในรูปที่ 5.20 ที่เชื่อมต่อเข้ากับบอร์ด Arduino โดยที่การทำงานของ Ethernet Shield จะอาศัยไอซีอินเทอร์เน็ตของ Wiznet W5100 จะให้ความสามารถในการระบุ network (IP) ทั้งในรูปแบบการเชื่อมต่อแบบ TCP และ UDP นอกจากนี้ Arduino Ethernet Shield จะรองรับการเชื่อมต่อการทำงานได้มากถึงสี่ช่องพร้อมๆกัน โดยจะต้องใช้หัวต่ออินเทอร์เน็ตมาตรฐานแบบ RJ45 เพื่อต่อสายอินเทอร์เน็ตเข้ากับ Ethernet Shield นอกจากนั้นแล้วหัวต่ออินเทอร์เน็ตยังสามารถที่ใช้ในการส่งผ่านกำลังไฟฟ้า (Power over Ethernet or PoE) ได้มากที่สุดถึง 15 W ผ่านสายเคเบิลแบบ Cat5 ที่ใช้มาตรฐานของ IEEE 802.3af ได้อีกด้วย

ข้อดีอีกอย่างหนึ่งของ Arduino Ethernet Shield คือจะมีช่องเก็บข้อมูลด้วยแผ่น SD card และสามารถใส่คำสั่งของ Arduino นำมาเรียกได้เลยอีกด้วย และ จะมีตัวควบคุมการเรียกกับสภาพเดิม (reset controller) ที่จะช่วยให้นึกได้ว่าชุดหน่วย W5100 Ethernet นั้นจะถูกรีเซ็ตอย่างถูกต้องเมื่อเริ่มใช้งาน และ ปุ่มรีเซ็ตบน Ethernet Shield จะทำหน้าที่รีเซ็ตทั้ง W5100 และ บอร์ด Arduino การเชื่อมต่อฟังก์ชันของ W5100 และ แผ่น SD card นั้นจะผ่าน SPI bus



รูปที่ 5.20 บอร์ด Arduino Ethernet Shield [18] (ซ้าย) และ การเชื่อมต่อบอร์ด Arduino Ethernet Shield เข้ากับ 3G wireless router เพื่อส่งสัญญาณข้อมูลผ่านอินเทอร์เน็ต

5.4.1 การกำหนดการตั้งค่า Network โดยทำงานผ่าน Pachube Web Server [23]

ก่อนที่โปรแกรมของ Netduino หรือ Ethernet Shield จะสามารถทำงานได้ คุณจะต้องแน่ใจเสียก่อนว่าบอร์ด Netduino Plus หรือ Ethernet Shield สามารถเชื่อมต่อหรือเข้าถึงอินเทอร์เน็ตได้ เช่น มันจะส่งข้อความไปยัง server ที่สามารถเห็นได้บนอินเทอร์เน็ต สมมุติว่า Netduino Plus หรือ Ethernet Shield ถูกเชื่อมต่อไปยังอินเทอร์เน็ตผ่าน router และ cable หรือ DSL Modem นั้นหมายความว่าต้องมี address ที่อยู่ของ network เพื่อที่จะเชื่อมต่อเข้ากับ board และคอมพิวเตอร์ PC โดยสามารถเชื่อมต่อ Netduino Plus หรือ Ethernet Shield ผ่านสาย USB เข้ากับคอมพิวเตอร์ได้เลย

สำหรับการทำงานของเว็บ Pachube หรือ Cosm ดังแสดงในรูปที่ 5.21 ตอนเริ่มต้นของดโปรแกรมของ Netduino Plus จะต้องมีการป้อนสองค่าหลักๆของ Pachube หรือ Cosm คือ API key (apiKey) และ ID Feed (feedID) ซึ่งสองค่านี้เราจะได้จาก www.pachube.com [23] ที่เราทำการสมัครไว้และจะต้องนำมาใส่ในโปรแกรม Visual Studio 2010 สำหรับ Netduino และ โปรแกรม Arduino สำหรับ Ethernet Shield เพื่อให้ค่านี้ตรงกับในเว็บ Pachube ก่อนที่เราจะสามารถทำงานของโปรแกรมได้ จะต้องแน่ใจเสียก่อนว่าบอร์ด Netduino Plus หรือ Ethernet Shield ของคุณเข้าถึงอินเทอร์เน็ต เช่นมันจะส่งข้อความไปยัง server ที่สามารถเห็นได้บนอินเทอร์เน็ต

รูปที่ 5.21 เว็บไซต์ของ Pachube [23]

รูปที่ 5.22 เว็บไซต์ของ Cosm [23]