

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการพัฒนาระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยอุปกรณ์มือถือแบบ Wi-Fi ได้ทำการศึกษาทฤษฎีและหลักการต่างๆที่สามารถนำมาประยุกต์เข้ากันได้ โดยแบ่งออกเป็นหัวข้อต่างๆ ได้ดังต่อไปนี้

- 2.1 บ้านอัจฉริยะ (Smart Home)
- 2.2 การควบคุมอุปกรณ์ไฟฟ้าแบบไร้สาย
- 2.3 ระบบโทรศัพท์มือถือที่ใช้ในการศึกษาค้นคว้าอิสระ
- 2.4 Arduino
- 2.5 Wi-Fi Shield for Arduino
- 2.6 การพัฒนาซอฟต์แวร์โดยใช้แบบจำลองแบบอไจล์ (Agile Development Model)
- 2.7 กระบวนการพัฒนาซอฟต์แวร์โดยใช้มาตรฐาน ISO 29110
- 2.8 การพัฒนาซอฟต์แวร์โดยใช้ Eclipse

2.1 บ้านอัจฉริยะ (Smart Home)

ระบบบ้านอัจฉริยะ (Smart Home) [1] มีระบบควบคุมอุปกรณ์ไฟฟ้า ระบบป้องกันขโมย ระบบเตือนภัยต่าง ๆ ซึ่งสามารถควบคุมผ่านได้หลายช่องทาง เช่น จาก ระบบบ้านอัจฉริยะจากระบบ Internet หรือผ่านทาง keypad ของกล่องควบคุม ระบบที่จะเลือกทำการศึกษาค้นคว้าจะเป็นระบบบ้านอัจฉริยะที่ใช้การเชื่อมต่อจากมือถืออัจฉริยะ (Smart Phone) ผ่านทาง Wi-Fi เพื่อเปิดปิดอุปกรณ์อำนวยความสะดวกภายในบ้าน รูปที่ 2 เป็นรูปบ้านอัจฉริยะที่มีระบบอัตโนมัติหลากหลายรูปแบบ ทั้งระบบการสื่อสารระบบความปลอดภัย ระบบไฟฟ้าอัตโนมัติ ระบบรดน้ำอัตโนมัติและอื่นๆ อีกหลากหลายซึ่งเป็นระบบอำนวยความสะดวกภายในบ้านทั้งสิ้น

บ้านอัจฉริยะ [2] คือการใช้เทคโนโลยีมาควบคุมอุปกรณ์ต่างๆภายในบ้าน เพื่ออำนวยความสะดวกแก่ผู้อยู่อาศัย มีระบบการจัดการพลังงาน ระบบรักษาความปลอดภัยอัตโนมัติทั้งภายใน

และรอบตัวบ้าน ส่วนใหญ่จะควบคุมด้วยระบบคอมพิวเตอร์ โดยทั่วไปเรียกว่า home automation ซึ่งสามารถถูกจำแนกความสามารถและความซับซ้อนในการควบคุมออกเป็น

- 2.1.1 ระบบควบคุมไฟฟ้าแสงสว่าง เช่น เปิด/ปิด หรือปรับระดับความสว่าง
- 2.1.2 ระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน เช่น สั่งงานเครื่องปรับอากาศ หรือการเปิดปิดม่าน
- 2.1.3 ระบบความบันเทิงภายในบ้าน เช่น สั่ง Internet radio ให้ทำงานในห้องที่ ผู้ใช้อยู่ และปิดเมื่อผู้ใช้ออกจากห้อง
- 2.1.4 ระบบบริหารพลังงาน และพลังงานสำรอง เช่น การปิด/เปิด เครื่องใช้ไฟฟ้าต่างๆโดยขึ้นกับสิ่งแวดล้อม
- 2.1.5 ระบบสื่อสาร เช่น รับ/ส่ง ข้อความหรือคำสั่งระหว่างผู้ใช้
- 2.1.6 ระบบรักษาความปลอดภัย เช่น เชื่อมต่อระบบกันขโมย/กล้อง กับ บ.รักษาความปลอดภัย



รูปที่ 2 ระบบควบคุมบ้านอัจฉริยะ

ที่มา: <http://www.homedecordream.com/smart-home-design/>

จากรูปที่ 2 แสดงภาพรวมของระบบควบคุมบ้านอัจฉริยะจะประกอบไปด้วยระบบต่าง ๆ ไม่ว่าจะเป็นระบบการสื่อสาร ระบบรดน้ำต้นไม้อัตโนมัติ ระบบกล้องวงจรปิด ระบบรักษาความปลอดภัย ระบบสำรองไฟฟ้าฉุกเฉิน ระบบเปิดปิดไฟฟ้าอัตโนมัติ ระบบความบันเทิงภายในบ้านซึ่งระบบระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยอุปกรณ์มือถือแบบไร้สาย เป็นส่วนหนึ่งในระบบบ้านอัจฉริยะ และอยู่ในส่วนของระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน

2.2 การควบคุมอุปกรณ์ไฟฟ้าแบบไร้สาย

การควบคุมอุปกรณ์ไฟฟ้าแบบไร้สาย [3] จะมีรูปแบบของการควบคุมแบบต่าง ๆ โดยมีการควบคุมผ่านช่องทาง Zigbee ผ่านทาง Bluetooth และผ่านทาง Wi-Fi แต่ละรูปแบบก็มีวิธีการที่แตกต่างกันแต่การใช้งานจะคล้าย ๆ กัน

2.2.1 การควบคุมอุปกรณ์ไฟฟ้าไร้สายแบบ Zigbee

ควบคุมผ่านทาง Zigbee เป็นเทคโนโลยีสื่อสารไร้สายที่ถูกพัฒนาขึ้นให้มีราคาต่ำ ใช้พลังงานน้อย จึงสามารถติดไว้ได้นาน และสามารถสร้างเครือข่ายได้ ซึ่งเหมาะกับการใช้งานด้านเซนเซอร์ไร้สาย ตรวจสอบตำแหน่งของวัตถุ และตรวจสอบสภาพแวดล้อม

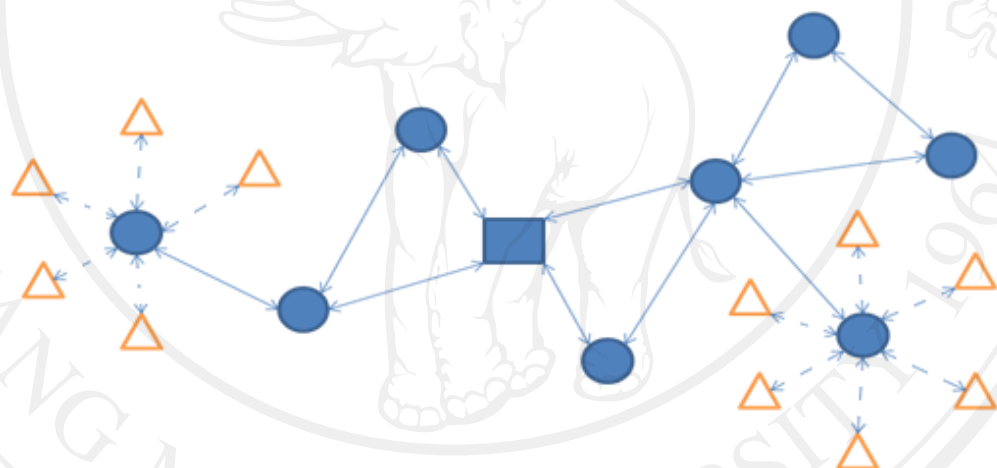
คุณสมบัติของ Zigbee

1. อัตราการส่งข้อมูล 250 Kbps (2.4 GHz), 40 Kbps (915 MHz), and 20 Kbps (866 MHz)
2. High throughput และ Low Latency Duty Cycle ต่ำ (< 0.1%)
3. มีการเข้าถึง Channel แบบ Channel Access Using Carrier Sense Multiple Access With Collision Avoidance (CSMA – CA)
4. สามารถรองรับ Address ได้ถึง 64 Bit IEEE Address (65535 Network)
5. สามารถส่งแบบ Full Hand Shaked Protocol
6. เชื่อมต่อ Topology ได้หลายแบบ เช่น Star, Peer To Peer, Mesh
7. ใช้พลังงานต่ำ (สามารถใช้ได้หลายเดือนจนถึงปี)
8. ระยะทางการส่งพื้นฐาน 5 – 500 เมตร

ส่วนประกอบของ Zigbee

1. Zigbee Coordinator ทำหน้าที่สร้างโครงข่าย จัดการโหนด (node) ในโครงข่าย และเก็บข่าวสารของโหนดในโครงข่าย
2. Zigbee Router ทำหน้าที่จัดการเส้นทางของข้อมูลที่ส่งผ่านภายในโครงข่าย ระหว่างโหนด

Zigbee End Device เป็นจุดปลายของโครงข่ายโครงข่าย อยู่ในส่วนของผู้ใช้งานโดยสามารถเป็นได้ทั้งแบบ FFD (Full Function Device)ซึ่งหมายถึงอุปกรณ์ที่สามารถทำงานได้ทุกอย่างในโครงข่าย และ RFD (Reduce Function Device) ซึ่งหมายถึงอุปกรณ์ที่ถูกลดความสามารถการทำงานในโครงข่าย



△ Zigbee End Device (RF or FFD)

■ Zigbee Coordinator (FFD)

● Zigbee Router

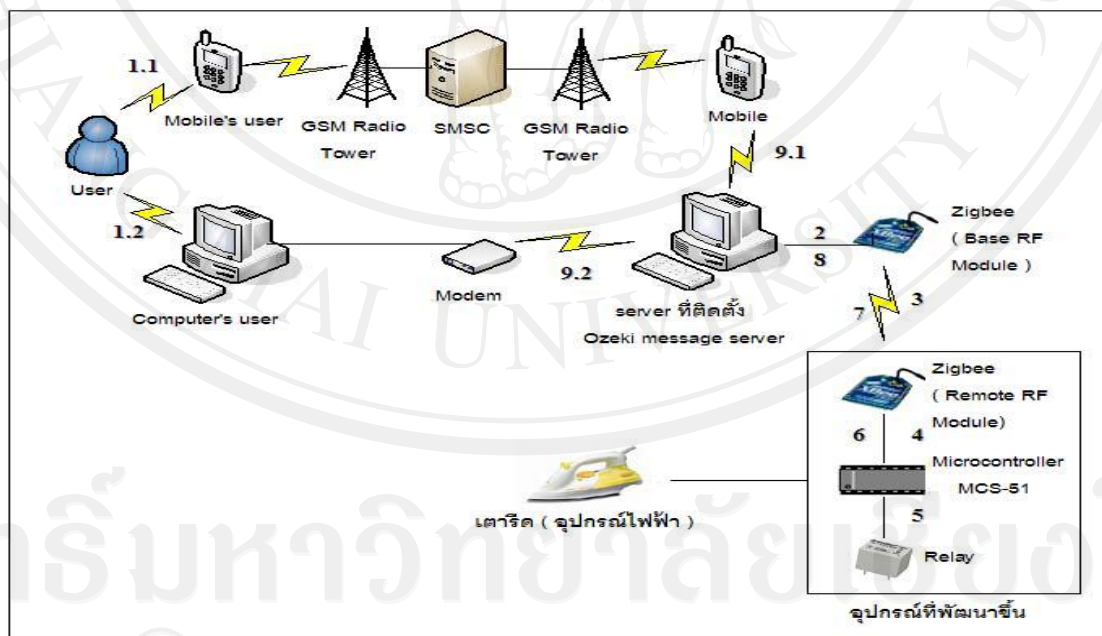
↔ Mesh Link

↔ Star Link

รูปที่ 3 ตัวอย่างการสร้างโครงข่ายของอุปกรณ์ Zigbee

ที่มา: <http://www.thaieasyelec.com/electronics-in-chapter/what-is-zigbee.html>

จากรูปที่ 3 แสดงผังการเชื่อมต่อระบบเครือข่ายของอุปกรณ์ Zigbee โดยมีการเชื่อมต่อกันในแบบต่าง ๆ โดยใช้ Zigbee เป็นอุปกรณ์ปลายทาง เป็นอุปกรณ์เชื่อมต่อระหว่างกลาง และเป็นอุปกรณ์ต้นทาง โดยมีการเชื่อมต่อโครงข่ายแบบ Mesh Link และ แบบ Star Link ตามรูปที่ 3 จะแสดงรูปแบบของการเชื่อมต่อในแบบต่าง ๆ ทั้ง 3 รูปแบบและยังแสดงการเชื่อมต่อในรูปแบบของโครงข่ายแบบ Mesh Link และ แบบ Star Link ในกรณีที่เป็นตัวอย่างของการศึกษาในส่วนของงานวิจัยที่นำมาเป็นข้อมูลประกอบการศึกษาค้นคว้าอิสระในครั้งนี้ เป็นรูปแบบการเชื่อมต่อในแบบอุปกรณ์ต้นทาง และเป็นแบบ Star Link ซึ่งเขาจะศึกษาการนำอุปกรณ์มือถือไปควบคุมอุปกรณ์ไฟฟ้าผ่านทางอุปกรณ์ Zigbee โดยมีการเชื่อมต่อได้ 2 ทาง เป็นการเชื่อมต่อผ่านระบบเครือข่ายโทรศัพท์ และการเชื่อมต่อผ่านระบบอินเทอร์เน็ต แต่สามารถเข้าควบคุมอุปกรณ์ไฟฟ้าผ่านช่องทาง Zigbee ได้เหมือนกันทั้ง 2 ทางโดยมีตัวกลางสำหรับการเชื่อมต่อในครั้งนี้ก็เป็นระบบ Server และเชื่อมต่อไปยัง Zigbee ก่อนที่จะส่งสัญญาณไปควบคุมอุปกรณ์ไฟฟ้าผ่านไมโครคอนโทรลเลอร์ และไมโครคอนโทรลเลอร์ส่งสัญญาณไปสั่ง รีเลย์เปิดปิดอุปกรณ์ไฟฟ้าต่อไป



รูปที่ 4 สถาปัตยกรรมของระบบควบคุมไฟฟ้าโดยใช้ Zigbee

ที่มา: นายสรกฤษ สิริปรีดากุล, นางสาวพิมลลักษณ์ จิรกุลกนก, รายงานการพัฒนาระบบควบคุมไฟฟ้าโดยใช้ Zigbee, หน้า 24-25

จากรูปที่ 4 แสดงการทำงานในการควบคุมอุปกรณ์ไฟฟ้าโดยใช้ Zigbee [4] มีการสั่งการจาก User ผ่านระบบได้ 2 ทาง โดยผ่านทางโทรศัพท์ และ ผ่านทางคอมพิวเตอร์ที่เชื่อมต่อผ่านระบบ Internet ทั้ง 2 ทางจะเชื่อมต่อเข้ากับ Server และ จะเชื่อมต่อไปยัง Zigbee จากนั้น Zigbee ก็จะทำการส่งสัญญาณควบคุมไปยังอุปกรณ์ปลายทาง ที่เป็น Zigbee ซึ่งเป็นอุปกรณ์ปลายทางที่เชื่อมต่อกับ Microcontroller และ ไปควบคุมการเปิดปิดอุปกรณ์ไฟฟ้าผ่าน Relay ตามรูปที่ 4

จากรูปที่ 4 ระบบควบคุมไฟฟ้าโดยใช้ Zigbee มีขั้นตอนการทำงาน ดังนี้

- (1) ผู้ใช้งานระบบจะสามารถติดต่อกับเครื่อง Server ที่ทำการติดตั้ง Ozeki message server โดยผู้ใช้งานระบบสามารถติดต่อระบบได้ 2 ทาง ดังนี้
 - (1.1) ทางโทรศัพท์มือถือผู้ใช้งานระบบจะสามารถติดต่อและสั่งงาน โดยการส่ง SMS ไปยังโทรศัพท์มือถือที่เชื่อมต่อกับเครื่อง Server ที่ติดตั้งโปรแกรม Ozeki message server ไว้
 - (1.2) ทางเครื่องคอมพิวเตอร์ที่เชื่อมต่ออินเทอร์เน็ตผู้ใช้งานระบบจะสามารถติดต่อและสั่ง งานผ่านทางหน้า Web Page เพื่อส่งข้อมูลไปยังเครื่อง Server ที่ติดตั้งโปรแกรม Ozeki message server
- (2) เครื่อง Server ที่ติดตั้งโปรแกรม Ozeki message server จะทำงานบน software ที่พัฒนาจากโปรแกรม Microsoft Visual Studio.NET (C#) ซึ่งจะติดต่อสื่อสาร เพื่อสั่งงาน Base RF Module (Zigbee ที่เชื่อมต่อกับเครื่อง Server ที่ติดตั้งโปรแกรม Ozeki message server) ผ่านทาง USB Cable
- (3) Base RF Module จะมีการติดต่อสื่อสารกับอุปกรณ์ Remote RF Module (Zigbee ที่ติดตั้ง อยู่ในอุปกรณ์ที่พัฒนาขึ้น) ผ่านทางเครือข่ายไร้สาย โดยใช้ IEEE 802.15.4 เป็นมาตรฐานในการติดต่อ
- (4) Remote RF Module จะส่งข้อมูลไปยัง Microcontroller เพื่อให้ Microcontroller ใช้ประมวลผลในการควบคุมการสั่งเปิด-ปิดอุปกรณ์ไฟฟ้า
- (5) Microcontroller ที่ติดตั้งอยู่ในอุปกรณ์ที่พัฒนาขึ้นจะส่งข้อมูลไปยัง Relay เพื่อควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้า
- (6) เมื่อทราบสถานะการเปิด-ปิดอุปกรณ์ไฟฟ้าหรือสถานะการสั่งปิดอุปกรณ์ไฟฟ้าที่เลือกตัว Microcontroller จะส่งข้อมูลกลับไป Remote RF Module เพื่อรายงานสถานะของอุปกรณ์

- (7) เมื่อ Remote RF Module ได้ข้อมูลสถานะการเปิด-ปิดอุปกรณ์ไฟฟ้า หรือ สถานะการสั่งปิดอุปกรณ์ไฟฟ้าจาก Microcontroller แล้ว จะส่งข้อมูลผ่านทางเครือข่ายไร้สาย โดยใช้ IEEE 802.15.4 เป็นมาตรฐานในการติดต่อ กลับไปที่อุปกรณ์ Base RF Module
- (8) เมื่อ Base RF Module ได้ข้อมูลสถานะการเปิด-ปิดอุปกรณ์ไฟฟ้า หรือ สถานะการสั่งปิดอุปกรณ์ไฟฟ้าจาก Remote RF Module แล้ว จะส่งข้อมูลดังกล่าว กลับไปที่เครื่อง Server ที่ติดตั้ง Ozekimessage server ผ่านทาง USB Cable เพื่อประมวลผล และส่งข้อมูลกลับไปยังผู้ใช้งานระบบ
- (9) เครื่อง Server ที่ติดตั้ง Ozeki message server ติดต่อกับผู้ใช้งานระบบ โดยเครื่อง Server สามารถติดต่อได้ 2 ทาง ดังนี้
 - (9.1) ทางโทรศัพท์มือถือเครื่อง Server ที่ติดตั้งโปรแกรม Ozeki message server จะทำการประมวลผล และตรวจสอบความถูกต้อง และส่งผลการปฏิบัติงานผ่านทาง SMS กลับไปยังโทรศัพท์มือถือของผู้ใช้งาน
 - (9.2) ทางเครื่องคอมพิวเตอร์ที่เชื่อมต่ออินเทอร์เน็ตเครื่อง Server ที่ติดตั้งโปรแกรม Ozeki message server จะทำการประมวลผล และตรวจสอบความถูกต้องและส่งผลการปฏิบัติงานผ่านทางหน้า Web Page

ปัญหาของ Zigbee คือมี RF data rate อยู่ที่ 250 Kbps ต่ำกว่า Wi-Fi ที่มี RF data rate อยู่ที่ 1 Mbps ซึ่งถ้ามีการพัฒนาขีดความสามารถของระบบจะทำให้เกิดปัญหาในเรื่องของการสื่อสารขึ้นได้ ซึ่ง Zigbee เหมาะสำหรับการทำ ในส่วนของ Sensor Network มากกว่า

2.2.2 การควบคุมอุปกรณ์ไฟฟ้าไร้สายแบบ Bluetooth

ควบคุมผ่าน Bluetooth [5] เป็นข้อกำหนดสำหรับระบบเครือข่ายส่วนบุคคล(Personal Area Networks – PAN) แบบไร้สาย บลูทูธช่วยให้อุปกรณ์อิเล็กทรอนิกส์ สามารถเชื่อมต่อกันได้ เช่น โทรศัพท์มือถือ พีดีเอ คอมพิวเตอร์ส่วนบุคคล โดยผ่านทางคลื่นวิทยุ

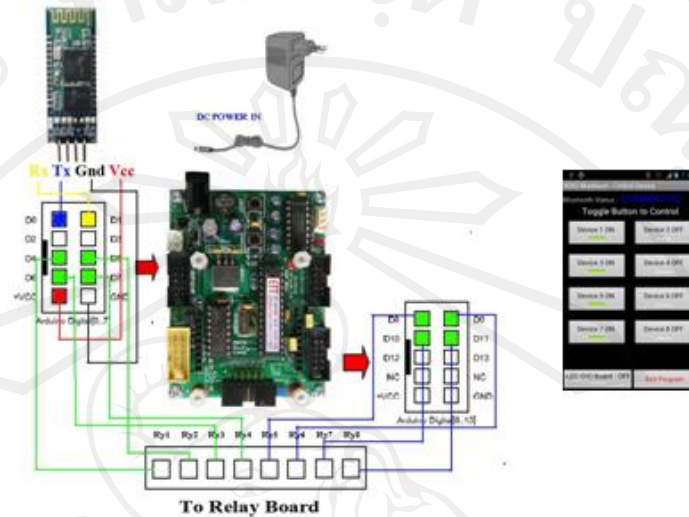
ระยะทำการ

ความสามารถในการส่งข้อมูลของบลูทูธนั้นแบ่งเป็นแต่ละ class ที่ใช้ ซึ่งมี 3 class ดังนี้

- Class 1 กำลังส่ง 100 มิลลิวัตต์ ระยะประมาณ 100 เมตร
- Class 2 กำลังส่ง 2.5 มิลลิวัตต์ ระยะประมาณ 10 เมตร
- Class 3 กำลังส่ง 1 มิลลิวัตต์ ระยะประมาณ 1 เมตร

รุ่น Bluetooth

- ปัจจุบัน พ.ศ. 2555 ข้อกำหนด Bluetooth มี 5 รุ่น
- Bluetooth 1.0 เป็นบลูทูธรุ่นแรกได้รับการพัฒนาขึ้นในปี พ.ศ. ๒๕๔๒ (ค.ศ. 1999) แต่ยังคงมีปัญหาอยู่มาก
- Bluetooth 1.1 ได้รับการพัฒนาขึ้นในปี พ.ศ. ๒๕๔๔ (ค.ศ. 2001) โดยได้แก้ปัญหที่เกิดขึ้นจากบลูทูธรุ่น 1.0 ทำให้บลูทูธรุ่นนี้ทำงานได้ดีขึ้น ใช้มาตรฐาน IEEE Standard 802.15.1 รองรับช่องสัญญาณที่ไม่มีการเข้ารหัส และมีเครื่องมือบอกระดับความแรงของสัญญาณด้วย
- Bluetooth 1.2 z ได้รับการพัฒนาขึ้นในปี พ.ศ. ๒๕๔๖ (ค.ศ. 2003) สามารถทำงานร่วมกับบลูทูธ 1.1 ได้ ซึ่งคุณสมบัติที่สำคัญของบลูทูธรุ่นนี้ได้แก่ การค้นหาสัญญาณและการเชื่อมต่อที่เร็วขึ้น ปรับปรุงความสามารถในการส่งข้อมูลโดยลดสัญญาณรบกวน นอกจากนี้เมื่อเกิดข้อผิดพลาดในการส่งข้อมูล ระบบก็จะส่งข้อมูลนั้นใหม่อีกครั้ง
- Bluetooth 2.0 บลูทูธรุ่นนี้สามารถทำงานร่วมกับบลูทูธ 1.1 ได้เช่นกัน ถูกพัฒนาขึ้นในปี พ.ศ. ๒๕๔๗ (ค.ศ. 2004) คุณสมบัติสำคัญเน้นในเรื่องของความเร็วในการรับ – ส่งข้อมูล ซึ่งสามารถส่งข้อมูลได้เร็วถึง ๒.๑ เมกะบิตต่อวินาที ด้วยความเร็วสูงสุดของช่องสัญญาณ ๓.๐ เมกะบิตต่อวินาที
- Bluetooth 2.0 EDR
- Bluetooth 2.1 EDR มีนวัตกรรมใหม่คือ การเชื่อมต่อระบบสัมผัส (Near Field Communication)
- Bluetooth 3.0 ความน่าสนใจของ Bluetooth 3.0 คือความเร็ว ซึ่งเหมือนกับ USB 3.0 แต่ Bluetooth มีอีกอย่างก็คือเรื่องของการประหยัดพลังงาน นั่นหมายความว่า เราสามารถพูดคุยโทรศัพท์ด้วยหูฟังที่ผ่าน Bluetooth 3.0 ได้นานขึ้น เหมาะสำหรับคนที่ใช้เวลาในการโทรนาน ๆ โดยเฉพาะ สำหรับมาตรฐานของ Bluetooth 3.0 นี้ใช้มาตรฐานเดียวกับ Wi-Fi คือ IEEE 802.11
- ระบบ EDR : Enhanced Data Rate เพิ่มความเร็วในการส่งข้อมูลสูงสุดเป็น 3 Mbps.
EDR (Enhanced Data Rate) คือ ระบบอิเล็กทรอนิกส์หรือคอมพิวเตอร์ที่ได้รับการปรับปรุงให้สามารถสื่อสารข้อมูลผ่านเครือข่ายได้เร็วขึ้น ทั้งนี้พัฒนาจากรุ่น 1.1 ที่ส่งข้อมูลด้วยอัตราความเร็ว 1 เมกะบิตต่อวินาที จนถึงรุ่น 1.2 ที่ปรับปรุงสัญญาณและคลื่นความถี่ บลูทูธ 2.0+ EDR ส่งข้อมูลด้วยอัตราความเร็ว 3 เมกะบิตต่อวินาที



รูปที่ 5 ระบบควบคุมอุปกรณ์ไฟฟ้าโดยใช้ Bluetooth

ที่มา: <http://microcontrollerkits.blogspot.com/2012/02/arduino-control-8-devices-via-android.html>

จากรูปที่ 5 แสดงระบบควบคุมอุปกรณ์ไฟฟ้าผ่านทาง Bluetooth [6] มีการเชื่อมต่อสัญญาณการควบคุมจากอุปกรณ์มือถือผ่านทาง Bluetooth โดยที่จะมีซอฟต์แวร์ที่เป็นตัวควบคุมอุปกรณ์ไฟฟ้าที่มือถือเขียนขึ้นมาใน ระบบ Android และส่งสัญญาณการควบคุมอุปกรณ์ไฟฟ้าผ่านทาง Bluetooth และผ่าน Microcontroller ผ่าน Relay และส่งสัญญาณการควบคุมไปยังอุปกรณ์ไฟฟ้า ตามรูปที่ 5

ปัญหาของ Bluetooth คือระยะทางของอุปกรณ์ที่ได้รับอนุญาตให้ใช้งานจะมีระยะทางในการติดต่อสื่อสาร ในระยะ ไม่เกิน 10 เมตรซึ่งเมื่อนำมาใช้งานจะเกิดปัญหาเรื่องของการควบคุมอุปกรณ์

2.2.3 การควบคุมอุปกรณ์ไฟฟ้าไร้สายแบบ Wi-Fi

Wi-Fi [7] หรือ เทคโนโลยีเครือข่ายแบบไร้สาย มาตรฐาน IEEE 802.11 ถือกำเนิดขึ้นในปี พ.ศ. 2528 จัดตั้งโดยองค์การสถาบันวิศวกรรมทางด้านไฟฟ้าและอิเล็กทรอนิกส์ (IEEE) มีความเร็ว 1 Mbps ในยุคเริ่มแรกนั้นให้ประสิทธิภาพการทำงานที่ค่อนข้างต่ำ ทั้งไม่มีการรับรองคุณภาพของการให้บริการที่เรียกว่า QoS (Quality of Service) และมาตรฐานความปลอดภัยต่ำ จากนั้นทาง IEEE

จึงจัดตั้งคณะทำงานขึ้นมาปรับปรุงหลายกลุ่มด้วยกัน โดยที่กลุ่มที่มีผลงานเป็นที่น่าพอใจและได้รับการยอมรับอย่างเป็นทางการว่า ได้มาตรฐานได้แก่กลุ่ม 802.11a, 802.11b และ 802.11g

มาตรฐาน IEEE 802.11b เสร็จสมบูรณ์เมื่อปี พ.ศ. 2542 ใช้เทคโนโลยีที่เรียกว่า CCK (Complimentary Code Keying) ผสมกับ DSSS (Direct Sequence Spread Spectrum) เพื่อปรับปรุงความสามารถของอุปกรณ์ให้รับส่งข้อมูลได้ด้วยความเร็วสูงสุดที่ 11 Mbps ผ่านคลื่นวิทยุความถี่ 2.4 GHz (เป็นย่านความถี่ที่เรียกว่า ISM (Industrial Scientific and Medical) ซึ่งได้รับการจัดสรรไว้อย่างสากลสำหรับการใช้งานอย่างสาธารณะด้านวิทยาศาสตร์ อุตสาหกรรม และการแพทย์ โดยอุปกรณ์ที่ใช้ความถี่ย่านนี้ก็เช่น IEEE 802.11, Bluetooth, โทรศัพท์ไร้สาย, และเตาไมโครเวฟ) มีระยะการส่งสัญญาณได้ไกลมาก ถึง 100 เมตร ปัจจุบันผลิตภัณฑ์อุปกรณ์เครือข่ายไร้สายภายใต้มาตรฐานนี้ได้รับการผลิตออกมาเป็นจำนวนมาก และที่สำคัญแต่ละผลิตภัณฑ์มีความสามารถทำงานร่วมกันได้ อุปกรณ์ของผู้ผลิตทุกยี่ห้อต้องผ่านการตรวจสอบจากสถาบัน Wi-Fi Alliance เพื่อตรวจสอบมาตรฐานของอุปกรณ์และความเข้ากันได้ของแต่ละผู้ผลิต ปัจจุบันนี้นิยมใช้อุปกรณ์ WLAN ที่มาตรฐาน 802.11b ไปใช้ในองค์กรธุรกิจ สถาบันการศึกษา สถานที่สาธารณะ และกำลังแพร่เข้าสู่สถานที่พักอาศัยมากขึ้น มาตรฐานนี้มีระบบเข้ารหัสข้อมูลแบบ WEP ที่ 128 บิต

มาตรฐาน IEEE 802.11a เสร็จสมบูรณ์เมื่อปี พ.ศ. 2542 โดยออกเผยแพร่ช้ากว่าของมาตรฐาน IEEE 802.11b ใช้เทคโนโลยีที่เรียกว่า OFDM (Orthogonal Frequency Division Multiplexing) เพื่อปรับปรุงความเร็วในการส่งข้อมูลให้สูงได้ถึง 54 Mbps บนความถี่ 5Ghz ซึ่งจะมีคลื่นรบกวนน้อยกว่าความถี่ 2.4 Ghz ที่มาตรฐานอื่นใช้กัน ที่ความเร็วนี้สามารถทำการแพร่ภาพและข่าวสารที่ต้องการความละเอียดสูงได้ อัตราความเร็วในการรับส่งข้อมูลสามารถปรับระดับให้ช้าลงได้ เพื่อเพิ่มระยะทางการเชื่อมต่อให้มากขึ้น แต่ข้อเสียก็คือ ความถี่ 5 Ghz นั้น หลายๆ ประเทศไม่อนุญาตให้ใช้ เช่นประเทศไทย เพราะได้จัดสรรให้อุปกรณ์ประเภทอื่นไปแล้ว และยิ่งไปกว่านั้น ระยะการส่งข้อมูลของ IEEE 802.11a ยังสั้นเพียง 30 เมตรเท่านั้น อีกทั้งอุปกรณ์ของ IEEE 802.11a ยังมีราคาสูงกว่า IEEE 802.11b ด้วย ดังนั้นอุปกรณ์ IEEE 802.11a จึงได้รับความนิยมน้อยกว่า IEEE 802.11b มาก จึงทำให้ไม่ค่อยเป็นที่ได้รับความนิยมเท่าที่ควร

มาตรฐาน IEEE 802.11g เสร็จสมบูรณ์ในปี พ.ศ. 2546 ทางคณะทำงาน IEEE 802.11g ได้นำเอาเทคโนโลยี OFDM ของ 802.11a มาพัฒนาบนความถี่ 2.4 Ghz จึงทำให้ใช้ความเร็ว 36-54 Mbps ซึ่งเป็นความเร็วที่สูงกว่ามาตรฐาน 802.11b ซึ่ง 802.11g สามารถปรับระดับความเร็วในการสื่อสารลงเหลือ 2 Mbps ได้ตามสภาพแวดล้อมของเครือข่ายที่ใช้งาน มาตรฐานนี้เป็นที่ยอมรับจาก

ผู้ใช้เป็นจำนวนมากและกำลังจะเข้ามาแทนที่ 802.11b ในอนาคตอันใกล้ นอกจากที่กล่าวมาข้างต้นนี้ มีบางผลิตภัณฑ์ใช้เทคโนโลยีเฉพาะตัวเข้ามาเสริม ทำให้ความเร็วเพิ่มขึ้นจาก 54 Mbps เป็น 108 Mbps แต่ต้องทำงานร่วมกันเฉพาะอุปกรณ์ที่ผลิตจากบริษัทเดียวกันเท่านั้น ซึ่งความสามารถนี้เกิดจากชิป (Chip) กระจายสัญญาณของตัวอุปกรณ์ที่ผู้ผลิตบางรายสามารถเพิ่มประสิทธิภาพการรับส่งสัญญาณเป็น 2 เท่าของการรับส่งสัญญาณได้ แต่ปัญหาของการกระจายสัญญาณนี้จะมีผลทำให้อุปกรณ์ ไร้สายในมาตรฐาน 802.11b มีประสิทธิภาพลดลงด้วยเช่นกัน

มาตรฐาน IEEE 802.11e คณะทำงานชุดนี้ได้รับมอบหมายให้ปรับปรุง MAC Layer ของ IEEE 802.11 เพื่อให้สามารถรองรับการใช้งานหลักการ Quality of Service สำหรับ application เกี่ยวกับมัลติมีเดีย (Multimedia) เนื่องจาก IEEE 802.11e เป็นการปรับปรุง MAC Layer ดังนั้นมาตรฐานเพิ่มเติมนี้จึงสามารถนำไปใช้กับอุปกรณ์ IEEE 802.11 WLAN ทุกเวอร์ชันได้ แต่อย่างไรก็ตามการทำงานของคณะทำงานชุดนี้ยังไม่แล้วเสร็จในขณะนี้

มาตรฐาน IEEE 802.11i คณะทำงานชุดนี้ได้รับมอบหมายให้ปรับปรุง MAC Layer ของ IEEE 802.11 ในด้านความปลอดภัย เนื่องจากเครือข่าย IEEE 802.11 WLAN มีช่องโหว่อยู่มาก โดยเฉพาะอย่างยิ่งการเข้ารหัสข้อมูล (Encryption) ด้วย key ที่ไม่มีการเปลี่ยนแปลง คณะทำงานชุด IEEE 802.11i จะนำเอาเทคนิคขั้นสูงมาใช้ในการเข้ารหัสข้อมูลด้วย key ที่มีการเปลี่ยนค่าอยู่เสมอ และการตรวจสอบผู้ใช้ที่มีความปลอดภัยสูง มาตรฐานเพิ่มเติมนี้จึงสามารถนำไปใช้กับอุปกรณ์ IEEE 802.11 WLAN ทุกเวอร์ชันได้ แต่อย่างไรก็ตามการทำงานของคณะทำงานชุดนี้ยังไม่แล้วเสร็จในขณะนี้

มาตรฐาน IEEE 802.11n เป็นมาตรฐานใหม่ที่ทาง Wi-Fi Alliance กำลังอยู่ในช่วงการทดสอบ โดยคาดว่าจะมีความเร็วในการถ่ายโอนข้อมูลอยู่ที่ 74 Mbps และสูงสุดที่ 248 Mbps ซึ่งหมายถึงว่าความเร็วกว่ารุ่นก่อนถึงประมาณ 5 เท่า นอกจากนี้ก็ยังมีรัศมีทำการภายในอาคารที่ 70 เมตร และนอกอาคารที่ 160 เมตร เพิ่มความสามารถในการกันสัญญาณกวนจากอุปกรณ์อื่นๆ ที่ใช้ความถี่ 2.4GHz เหมือนกัน และสามารถรองรับอุปกรณ์มาตรฐาน IEEE 802.11b และ IEEE 802.11g ได้ มาตรฐาน IEEE 802.11n นี้ได้เสร็จสมบูรณ์ในปี พ.ศ. 2552

Wi-Fi ได้กำหนดลักษณะการเชื่อมต่อของอุปกรณ์ภายในเครือข่ายแลน ไว้ 2 ลักษณะคือ โหมด Infrastructure และ โหมด Ad-Hoc หรือ Peer-to-Peer

โหมด Infrastructure โดยทั่วไปแล้วอุปกรณ์ในเครือข่าย Wi-Fi จะเชื่อมต่อกันในลักษณะของโหมด Infrastructure ซึ่งเป็นโหมดที่อนุญาตให้อุปกรณ์ภายใน LAN สามารถเชื่อมต่อกับเครือข่ายอื่นได้ ในโหมด Infrastructure นี้จะประกอบไปด้วยอุปกรณ์ 2 ประเภทได้แก่ สถานีผู้ใช้ (Client Station) ซึ่งก็คืออุปกรณ์คอมพิวเตอร์ (Desktop, แล็ปท็อป, หรือ PDA ต่างๆ) ที่มีอุปกรณ์ Client Adapter เพื่อใช้รับส่งข้อมูลผ่าน Wi-Fi และสถานีแม่ข่าย (Access Point) ซึ่งทำหน้าที่ต่อเชื่อมสถานีผู้ใช้เข้ากับเครือข่ายอื่น (ซึ่งโดยปกติจะเป็นเครือข่าย IEEE 802.3 Ethernet LAN) การทำงานในโหมด Infrastructure มีพื้นฐานมาจากระบบเครือข่ายโทรศัพท์มือถือ กล่าวคือสถานีผู้ใช้จะสามารถรับส่งข้อมูลโดยตรงกับสถานีแม่ข่ายที่ให้บริการ แก่สถานีผู้ใช้นั้นอยู่เท่านั้น ส่วนสถานีแม่ข่ายจะทำหน้าที่ส่งต่อ (forward) ข้อมูลที่ได้รับจากสถานีผู้ใช้ไปยังจุดหมายปลายทางหรือส่งต่อข้อมูลที่ได้รับจากเครือข่ายอื่นมายังสถานีผู้ใช้

โหมด Ad-Hoc หรือ Peer-to-Peer เครือข่าย Wi-Fi ในโหมด Ad-Hoc หรือ Peer-to-Peer เป็นเครือข่ายที่ปิดคือไม่มีสถานีแม่ข่ายและไม่มีการเชื่อมต่อกับเครือข่ายอื่น บริเวณของเครือข่าย Wi-Fi ในโหมด Ad-Hoc จะเรียกว่า Independent Basic Service Set (IBSS) ซึ่งสถานีผู้ใช้หนึ่งสามารถติดต่อสื่อสารข้อมูลกับสถานีผู้ใช้อื่นๆในเขต IBSS เดียวกันได้โดยตรงโดยไม่ต้องผ่านสถานีแม่ข่าย แต่สถานีผู้ใช้จะไม่สามารถรับส่งข้อมูลกับเครือข่ายอื่นๆได้

2.3 ระบบโทรศัพท์มือถือที่ใช้ในการศึกษาค้นคว้าอิสระ

เป็นระบบปฏิบัติการ Android แพลตฟอร์มสำหรับโทรศัพท์มือถือและอุปกรณ์พกพา ซึ่งประกอบด้วยระบบปฏิบัติการ ไกลบริรี เฟรมเวิร์ค และซอฟต์แวร์อื่นๆ ที่จำเป็นในการพัฒนา มันเทียบเท่ากับ Windows Mobile, Palm OS, Symbian, OpenMoko และ Maemo ของโนเกีย โดยจับตลาดมือถือรุ่นใหม่ๆ ความสามารถสูงๆ ถึงจะเป็นแพลตฟอร์มใหม่ แต่ Android ก็ใช้องค์ประกอบที่เป็นโอเพนซอร์สหลายอย่าง เช่น Linux Kernel, SSL, OpenGL, FreeType, SQLite, WebKit และเขียนไกลบริรี+เฟรมเวิร์คของตัวเองเพิ่มเติม ซึ่งทั้งหมดจะโอเพนซอร์ส (ใช้ Apache License)

จุดเด่นของ Android นั้นอยู่ที่การออกแบบระบบปฏิบัติการมาให้มีความสามารถได้ใกล้เคียงกับคอมพิวเตอร์พกพาคัดย่อกที่ทำงานได้บนโทรศัพท์มือถือ โดยยังคงเรื่องความคล่องตัวในการใช้งานมาก นอกจากนี้ Android นั้นเป็นแบบระบบเปิด คือ เปิดโอกาสให้บริษัทและนักพัฒนาสามารถมีส่วนร่วมในการสร้างระบบปฏิบัติการได้ และยังไม่จำกัดการใช้งานเฉพาะบน

โทรศัพท์มือถือจากค่ายใดค่ายหนึ่ง นั่นทำให้มีระบบ Android จากผู้ผลิตโทรศัพท์มือถือเกือบทุกราย ต่างจาก iPhone หรือ BlackBerry ที่ใช้ระบบปิด

2.4 Arduino

Arduino [8] คือ ไมโครคอนโทรลเลอร์ ชนิดหนึ่ง ซึ่งเป็นแบบที่เรียกว่า Open Hardware กล่าวคือ Arduino อุปกรณ์ที่มีแบบส่วนประกอบเป็นมาตรฐานเปิดเผยข้อมูล หมายความว่า เราสามารถทำเองโดยใช้ต้นแบบที่มีการเปิดเผยทั่วไปได้ มีซอฟต์แวร์ให้ใช้งานฟรี สามารถนำไปใช้งานทั่วไปหรือแบบธุรกิจได้โดยไม่ต้องเสียค่าลิขสิทธิ์ เป็นรูปแบบที่มีข้อมูลมากที่สุดบนอินเทอร์เน็ต การพัฒนาง่าย เพราะมีตัวอย่างมากมาย และไม่ต้องเขียนโปรแกรมในรูปแบบ Low Level หมายความว่า เราสามารถใช้คำสั่งเขียนโปรแกรมได้เสมือน โปรแกรมภาษาชั้นสูงทั่วไป



รูปที่ 6 แผงวงจร ET-EASY MEGA1280

ที่มา: http://www.etteam.com/product2009/ET-AVR/ET-EASY_MEGA1280.html

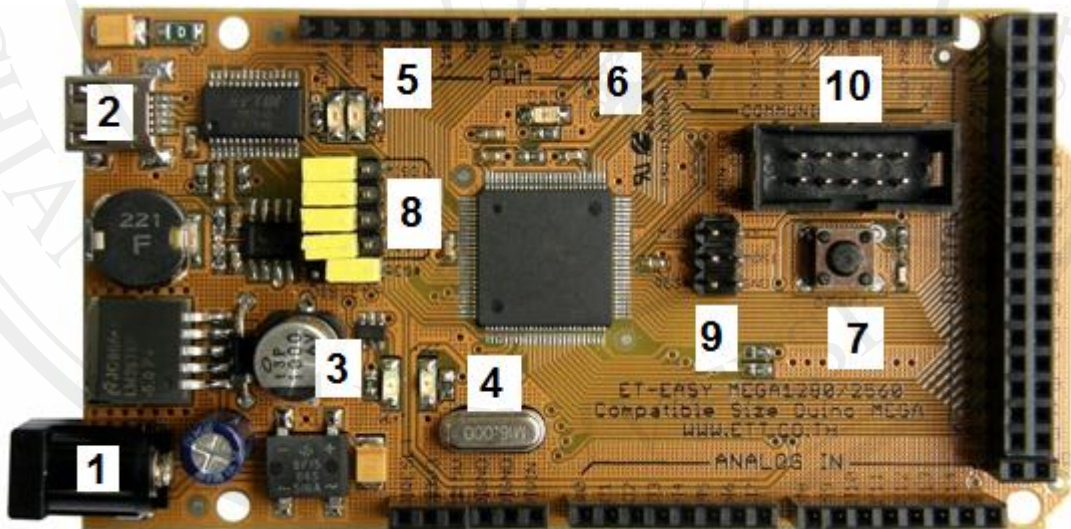
จากรูปที่ 6 แสดงรูปแผงวงจร ET-EASY MEGA1280 จากการที่ Arduino ที่เป็นโครงการพัฒนาระบบ MCU ของ AVR แบบ Open Source ได้รับการแนะนำเผยแพร่ออกมา ซึ่งได้รับความนิยมกันอย่างแพร่หลายจากผู้คนทั่วโลก ภายในระยะเวลาอันรวดเร็ว ทางด้านของ Software ก็มีการพัฒนาอย่างต่อเนื่อง ซึ่งเมื่อหลายปีที่ผ่านมา (สิงหาคม 2552) โปรแกรมของ Arduino ได้รับการปรับปรุงเป็น Version “arduino-0016” โดยทางด้าน Hardware เองก็ได้มีการพัฒนาปรับปรุงอย่างต่อเนื่องควบคู่กันไปด้วยเช่นเดียวกัน ซึ่งจากเดิมที่มีการพัฒนาโปรแกรมให้รองรับกับการใช้งานได้

กับชิพ MCU รุ่นเล็ก 28 ขา อย่าง ATMEGA8, ATMEGA88/ATMEGA168/ATMEGA328 สำหรับเป็นจุดเริ่มต้นให้ผู้สนใจได้ใช้เป็นเครื่องมือและอุปกรณ์ในการศึกษาทดลองเรียนรู้ ตลอดจนถึงนำไปประยุกต์ใช้งานกันนั้น มาถึงวันนี้ ขนาดของทรัพยากรต่างๆ รวมทั้งขนาดของหน่วยความจำสำหรับเขียนโปรแกรม ที่มีอยู่ในชิพ AVR รุ่นเล็กที่มีอยู่เริ่มไม่เพียงพอต่อการประยุกต์ใช้งานในงานบางประเภทแล้ว ทาง Arduino เอง จึงได้ทำการพัฒนาให้ Arduino สามารถรองรับการใช้งานขนาดใหญ่ขึ้นอีก โดยปรับปรุงโปรแกรมให้ใช้กับชิพ AVR รุ่นใหญ่ขึ้น เพื่อให้มีจำนวน I/O ทั้ง Digital, Analog, PWM, UART และ ขนาดหน่วยความจำเพิ่มมากขึ้นกว่าเดิม แต่ก็ยังคงใช้แนวทางในการพัฒนาโปรแกรม แบบเดียวกันกับรุ่นเล็กทุกประการ โดยได้เลือกใช้ชิพ AVR เบอร์ ATMEGA1280 และออกแบบพัฒนา Hardware บอร์ดขึ้นมารองรับโดยใช้รหัสชื่อยี่ห้อว่า “Arduino Mega” ออกวางจำหน่าย และได้มีการเผยแพร่รายละเอียดทาง Hardware ต่างๆ ให้ผู้สนใจนำไปพัฒนาต่อยอดกันตัวเอง แต่เนื่องจากชิพดังกล่าวมีโครงสร้างตัวถังเป็นแบบ SMD จึงทำให้เป็นอุปสรรคสำหรับผู้ใช้งานจำนวนมากไม่น้อยในการที่จะผลิต หรือ สร้างบอร์ดขึ้นใช้งานเองทางบริษัท อีทีที จึงได้นำ ATMEGA1280 มาพัฒนาเป็นบอร์ด โดยให้มีโครงสร้างการทำงานเช่นเดียวกันกับ Arduino Mega ขึ้นมา โดยใช้ชื่อว่า “ET-EASY MEGA1280” โดยได้ออกแบบให้มีการจัดสรร Pin I/O ต่างๆ รวมทั้งขนาดให้ตรงตามมาตรฐานของบอร์ด “Arduino Mega” เพียงแต่ได้มีการปรับปรุงข้อจำกัดบางอย่างให้ดีขึ้นกว่า Arduino Mega รุ่นมาตรฐาน เพื่อเพิ่มความสะดวกกับผู้ใช้งานมากยิ่งขึ้น

คุณสมบัติของบอร์ด

- ใช้ ATMEGA1280 เป็น MCU ประจำบอร์ด Run ความถี่ 16MHz จาก Crystal Oscillator
- 128KByte Flash (สงวนไว้ 4KByte สำหรับ Bootloader) / 8KByte SRAM / 4KByte EEPROM
- รองรับการพัฒนาโปรแกรมด้วยภาษา C++ ของ Arduino ตามแบบ Arduino Mega ได้ 100%
- ใช้ USB Bridge ของ FTDI เบอร์ FT232RL พร้อม Over Current Protection สำหรับติดต่อสื่อสารและ Download Code จากคอมพิวเตอร์ให้บอร์ด พร้อม Jumper สำหรับปรับใช้งานบอร์ดเป็นการ Program Bootloader ให้กับ MCU จากพอร์ต USB ในบอร์ดได้เอง โดยไม่ต้องใช้เครื่องโปรแกรม AVRISP จากภายนอก
- 54 Pin Digital I/O โดยมี 14Pin สามารถโปรแกรมหน้าที่เป็น PWM ได้
- 16 Pin Analog Input (ADC ขนาด 10 บิต 16 ช่อง)

- 4 UART(Hardware Serial Port) แบบ TTL Logic
- ขนาดของ PCB บอร์ด และ ตำแหน่ง Pin Connector ต่างๆ ตรงกันกับ Arduino Mega ทั้งหมดทำให้สามารถนำไปติดตั้งใช้งานร่วมกับบอร์ด Shield แบบต่างๆที่มีการผลิตขึ้นมาใช้งานร่วมกับบอร์ด Arduino Mega ได้ทั้งหมด โดยบอร์ดมีขนาด PCB Size 5.3cm x 10.2cm
- มีขั้ว Header 10Pin IDE ของ 8 บิต Digital I/O(D22...D29) สำหรับเชื่อมต่อกับ LCD หรือ บอร์ด I/O แบบต่างๆ ของ อีทีที เพิ่มความสะดวกในการใช้งานรองรับการใช้งานกับ External Supply ทั้งแบบ AC และ DC ขนาด 7-20V โดยเลือกใช้ Regulate แบบ Switching ขนาด 1A (LM2575-5V) ลดปัญหาเรื่องความร้อนเมื่อมีการใช้กระแสสูงๆ สามารถใช้แหล่งจ่ายจากพอร์ต USB ได้ในกรณีใช้กระแสไม่เกิน 500mA โดยมีวงจรเลือกแหล่งจ่ายอัตโนมัติโดยจะตัดการใช้ไฟเลี้ยงจาก USB โดยอัตโนมัติ เมื่อมีการต่อแหล่งจ่ายจากภายนอกให้บอร์ด



รูปที่ 7 แสดงโครงสร้างของบอร์ด ET-EASY MEGA1280 (Duino MeGA)

ที่มา: http://www.etteam.com/product2009/ET-AVR/ET-EASY_MEGA1280.html

จากรูปที่ 7 แสดงโครงสร้างของบอร์ด ET-EASY MEGA1280 (Duino MeGA) ซึ่งจะมีองค์ประกอบของบอร์ดตามหมายเลขกำกับดังต่อไปนี้

หมายเลข 1 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงจากภายนอก สามารถใช้ได้กับแหล่งจ่ายทั้งแบบ AC และ DC พร้อมวงจร Bridge Rectifier และ Regulate แบบ Switching ช่วยลดความร้อนของ IC Regulate เมื่อมีการดึงกระแสมากๆ ได้เป็นอย่างดี สามารถใช้กับแรงดัน Input 7-20V

หมายเลข 2 เป็นขั้วต่อ USB สำหรับติดต่อสื่อสารกับคอมพิวเตอร์ PC โดยใช้ FT232RL เป็น USB Bridge ในการเชื่อมต่อระหว่างคอมพิวเตอร์ PC และ MCU ในบอร์ด และยังสามารถใช้ไฟจาก พอร์ต USB เป็นแหล่งจ่ายให้กับบอร์ดได้ด้วย โดยจะมี Poly Fuse ขนาด 500mA สำหรับป้องกันการดึงกระแสเกินจากพอร์ต USB ด้วย และที่พิเศษคือมีวงจรสำหรับตรวจสอบแหล่งจ่ายเพื่อสลับการใช้งานแหล่งจ่ายจาก USB ไปเป็น External Supply ได้เอง โดยอัตโนมัติ โดยเมื่อไม่ได้ต่อ External Supply บอร์ดจะใช้ไฟจากพอร์ต USB เป็นแหล่งจ่ายในการทำงาน แต่เมื่อ มีการต่อ External Supply วงจรจะสลับไปใช้แหล่งจ่ายจาก External Supply เองโดยอัตโนมัติ

หมายเลข 3 เป็น LED VEXT ใช้แสดงสถานะเมื่อมีการจ่ายไฟเลี้ยงจาก External Supply

หมายเลข 4 เป็น LED +VCC ใช้แสดงสถานะของแหล่งจ่ายไฟเลี้ยง (+VCC) ของบอร์ด โดยเมื่อบอร์ดใช้แหล่งจ่ายจาก External Supply จะแสดงสถานะโดยการให้ LED VEXT และ LED +VCC ติดสว่างพร้อมกันทั้งคู่ แต่ถ้าบอร์ดใช้แหล่งจ่ายจากพอร์ต USB จะแสดงสถานะโดยการให้ LED +VCC ติดสว่างเพียงดวงเดียว

หมายเลข 5 เป็น LED แสดงสถานะของ RX และ TX ใช้สำหรับแสดงการรับส่งข้อมูลระหว่างบอร์ด ET-EASY MEGA1280 กับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB

หมายเลข 6 เป็น LED D13 ใช้สำหรับทดสอบการทำงานของ Bootloader และ ใช้ทดสอบการทำงานของบอร์ดจากการควบคุมของ Pin Digital-13 ทำงานด้วย Logic “1” และ หยุดทำงานด้วย Logic “0”

หมายเลข 7 เป็นสวิตช์ Reset ใช้สำหรับสั่ง Reset การทำงานของบอร์ด

หมายเลข 8 เป็นชุด Jumper สำหรับเลือก การ Program Bootloader ผ่าน USB Port และการใช้งานตามปกติ

หมายเลข 9 เป็นขั้วต่อ AVRISP ใช้สำหรับ Download Code ให้กับ MCU โดยขั้วต่อ AVRISP นี้จะสามารถใช้งานได้กับเครื่องโปรแกรมทุกรุ่นที่รองรับการใช้งานกับ ATMEGA1280 และใช้ขั้วต่อตรงตามมาตรฐาน AVRISP

หมายเลข 10 เป็นขั้วต่อสัญญาณจาก D[22..29] สำหรับเชื่อมต่อกับบอร์ด I/O ของ อีทีที รวมทั้งจอแสดงผล LCD โดยใช้ร่วมกับ 10PIN LCD หรือ ET-CONV SPI TO LCD

คุณสมบัติของสัญญาณต่างๆของบอร์ด ET-EASY MEGA1280

RESET# เป็นสัญญาณ Input Logic Reset ของ MCU เมื่อเป็น Logic Low จะทำให้ MCU อยู่ในสภาวะรีเซ็ต เมื่อเป็น Logic High จะทำให้ MCU อยู่ในสภาวะทำงานตามปกติ โดยสัญญาณ RESET# นี้จะถูกควบคุมจาก 2 แหล่ง คือ จาก สวิตช์ RESET ภายในบอร์ด และ จากสัญญาณ DTR ของ FT232RL ถ้ามีการเลือก Enable Jumper ของ Auto Reset จาก DTR ไว้

+3V3 เป็นแหล่งจ่ายไฟขนาด +3.3V ที่ได้จากวงจร Regulate ภายในของ FT232RL สามารถจ่ายกระแสได้สูงสุด 50mA ซึ่งเมื่อต้องการนำแหล่งจ่าย +3.3V นี้ไปใช้งานเป็นแหล่งจ่ายให้กับอุปกรณ์ต่างๆ ต้องระมัดระวังเรื่องการดึงกระแสของโหลดด้วย ถ้าโหลดมีการดึงกระแสมากกว่า 50mA อาจทำให้ FT232RL เกิดความเสียหายได้

+VIN เป็นไฟ DC ที่รับมาจาก Jack VIN(External Supply) แต่ผ่านการ Rectifier และ Filter เป็น DC แล้ว มีขนาดแรงดันเฉลี่ยตามขนาดแรงดันที่ป้อนให้กับบอร์ดทาง Jack VIN

+5V เป็นจุดต่อแหล่งจ่ายไฟของบอร์ดออกไปใช้งาน ซึ่งมาจากแหล่งกำเนิด 2 แหล่ง คือ จากพอร์ต USB และจาก External Supply ซึ่งถ้าต่อแหล่งจ่ายให้บอร์ดจาก External Supply ผ่านทาง Jack VIN แหล่งจ่าย +5V นี้จะมาจาก Switching Regulate (LM2575-5V) สามารถจ่ายกระแสได้สูงสุดถึง 1A แต่ถ้าใช้แหล่งจ่ายจากพอร์ต USB แหล่งจ่าย +5V นี้จะมาจากพอร์ต USB โดยตรงโดยจะมีฟิวส์ แบบ Poly ขนาด 500mA ต่อป้องกันการดึงกระแสเกินเพื่อป้องกันความเสียหายของพอร์ต USB โดยจะจ่ายกระแสได้สูงสุดไม่เกิน 500mA ขึ้นอยู่กับคุณสมบัติในการจ่ายกระแสของพอร์ต USB และการ Configure ค่าให้กับ FT232RL ด้วย

A0-A15 เป็นขาสัญญาณ Analog Input แบบ ADC มีขนาดความละเอียด 10 บิต มี 16 Pin สามารถรับแรงดัน Analog Input ได้ 0-5VDC

D0-D53 เป็นขาสัญญาณ Digital Input/Output แบบ TTL มีทั้งหมด 54 Pin สามารถใช้ทำหน้าที่เป็น Input หรือ Output ตามการกำหนดจากโปรแกรม โดยมีบาง Pin สามารถกำหนดหน้าที่ใช้งานเป็นฟังก์ชันพิเศษต่างๆเพิ่มเติมได้อีก

D0-D1 ถูกสงวนไว้ใช้ทำหน้าที่เป็นพอร์ตสื่อสารอนุกรม RS232 (UART0) โดยได้ทำการเชื่อมต่อกับ USB Bridge ของ FT232RL เพื่อใช้ Upload Code ให้กับบอร์ด และยังสามารถใช้ทดสอบติดต่อสื่อสารรับส่งข้อมูลระหว่างบอร์ดกับคอมพิวเตอร์ PC

D2-D13 สามารถ โปรแกรมหน้าที่เป็น PWM ขนาด 8 บิต มี 14 Pin

D14 สามารถ โปรแกรมหน้าที่เป็น TX3 สำหรับ ส่งข้อมูลของ UART3

D15 สามารถ โปรแกรมหน้าที่เป็น RX3 สำหรับ รับข้อมูลให้กับ UART3

D16 สามารถ โปรแกรมหน้าที่เป็น TX2 สำหรับ ส่งข้อมูลของ UART2

D17 สามารถ โปรแกรมหน้าที่เป็น RX2 สำหรับ รับข้อมูลให้กับ UART2

D18 สามารถ โปรแกรมหน้าที่เป็น TX1 สำหรับ ส่งข้อมูลของ UART1

D19 สามารถ โปรแกรมหน้าที่เป็น RX1 สำหรับ รับข้อมูลให้กับ UART1

D20 สามารถ โปรแกรมหน้าที่เป็น SDA ของ I2C Bus สำหรับใช้สื่อสารกับ I2C

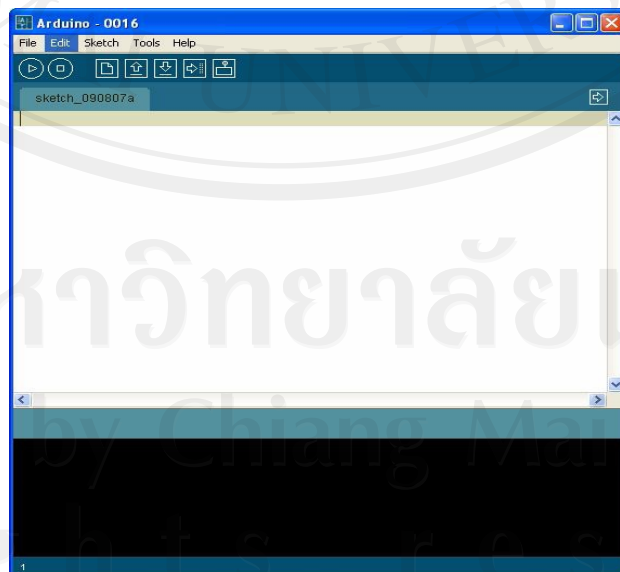
D21 สามารถ โปรแกรมหน้าที่เป็น SCL ของ I2C Bus สำหรับใช้สื่อสารกับ I2C

AREF เป็นสัญญาณ Analog Reference จากภายนอกที่ต้องการป้อนให้กับ MCU ซึ่งตามปรกติแล้ว ATMEGA1280 สามารถโปรแกรมให้เลือกใช้แรงดันอ้างอิงจากภายในได้อยู่แล้วโดยสามารถเลือกเป็น 1.1V หรือ 2.56V หรือ AVCC(+5V) โดยไม่จำเป็นต้องป้อนแรงดันอ้างอิงจากภายนอกให้กับบอร์ดอีก

ทดสอบเขียนโปรแกรมใช้งานด้วย Arduino

ทดสอบเขียนโปรแกรมใช้งานด้วย Arduino [9] หลังจากที่เราได้ทำการติดตั้งโปรแกรม Arduino เป็นที่เรียบร้อยแล้ว ก็เป็นอันเสร็จสิ้นขั้นตอนของการเตรียมการแล้ว ลำดับขั้นตอนต่อจากนี้เป็นต้นไป ก็เป็นเรื่องของการใช้งาน การเขียนโปรแกรม และการศึกษาเรียนรู้ต่างๆตามความต้องการแล้ว แต่ก่อนอื่นเราจะต้องทำการติดตั้งโปรแกรมของ Arduino เพื่อใช้เป็นโปรแกรมสำหรับศึกษาเรียนรู้ ซึ่งมีลำดับขั้นตอนดังต่อไปนี้

1. ทำการสั่ง Run โปรแกรม “arduino.exe” จะได้ผลดังรูป

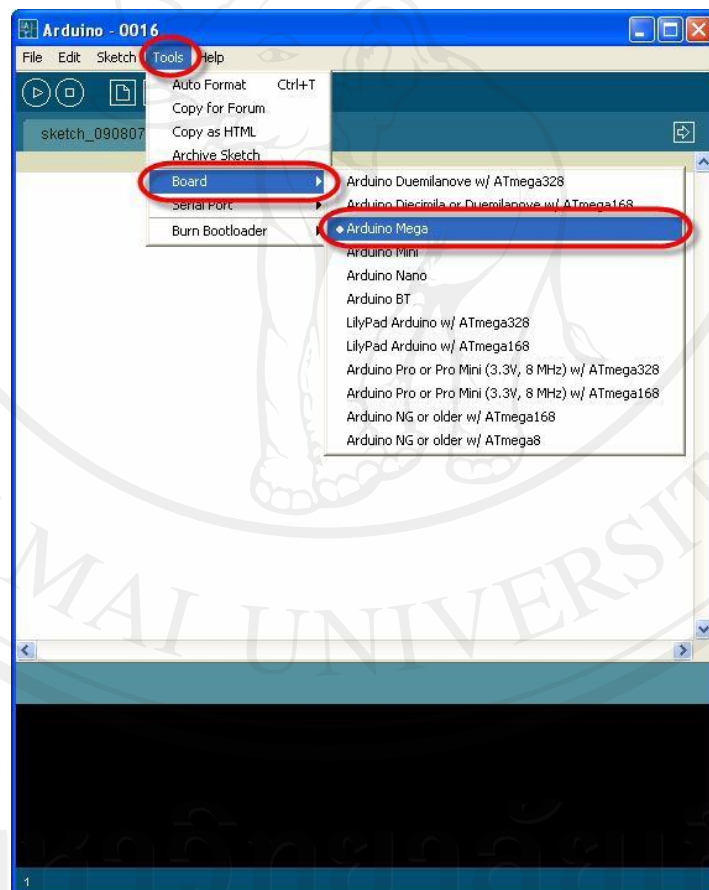


รูปที่ 8 รูปการสั่ง Run โปรแกรม “arduino.exe”

จากรูปที่ 8 แสดงรูปการตั้ง Run โปรแกรม “arduino.exe” จะได้หน้าต่างของโปรแกรมที่ใช้เขียนโปรแกรมในการใช้งาน Arduino ด้วยภาษาซี

2. ในครั้งแรกของการเรียกใช้งานโปรแกรม ให้ทำการกำหนดระบบฮาร์ดแวร์ที่จะใช้งานกับโปรแกรมของ Arduino ให้เรียบร้อยเสียก่อน เนื่องจากในปัจจุบันนี้ มีการออกแบบวงจรและ

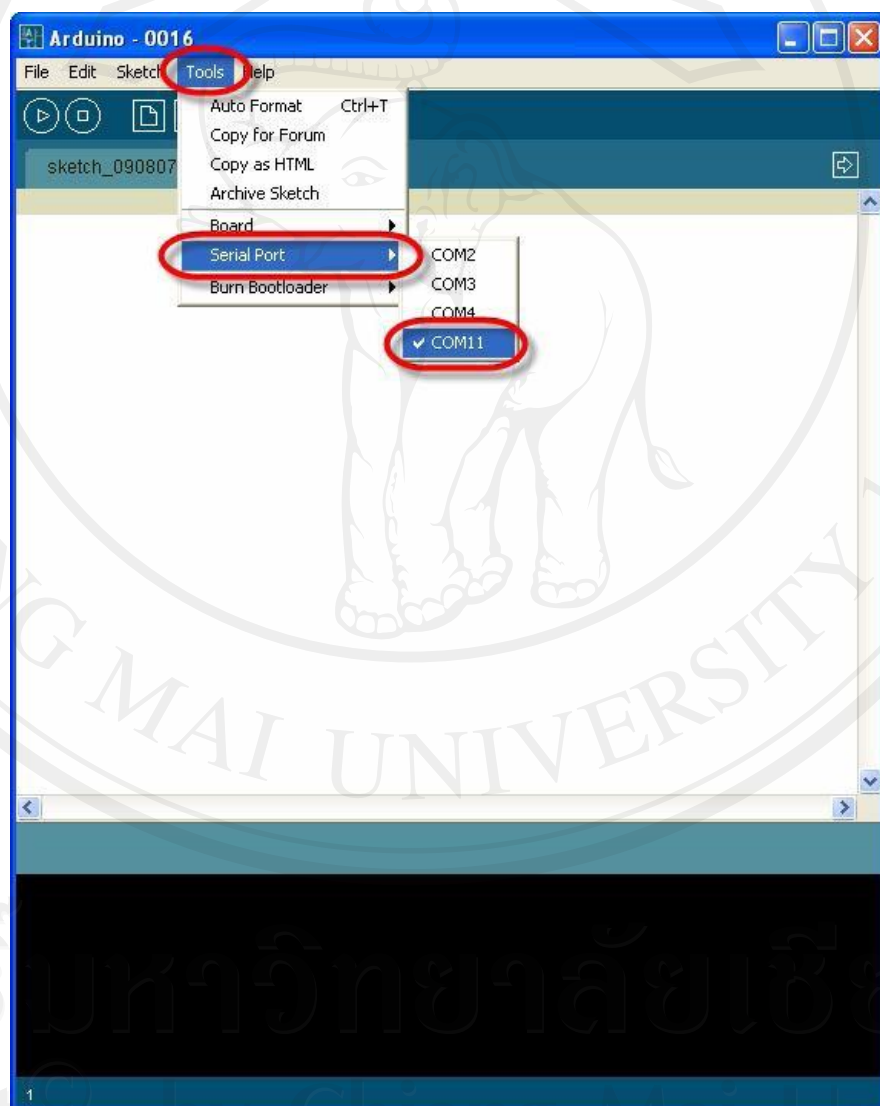
สร้างฮาร์ดแวร์บอร์ดแบบต่างๆสำหรับนำมาใช้งานร่วมกับโปรแกรมพัฒนาของ Arduino ใหม่มากมายหลายรุ่น โดยในกรณีของบอร์ด ET-EASY MEGA1280 ให้ทำการเลือกกำหนดชื่อบอร์ดเป็น “Arduino Mega” โดยคลิกเมาส์ที่ “Tools → Board → “Arduino Mega” ดังรูป



รูปที่ 9 รูปการตั้งค่าการกำหนดระบบฮาร์ดแวร์เป็น “Arduino Mega”

จากรูปที่ 9 แสดงรูปการตั้งค่าการกำหนดระบบฮาร์ดแวร์เป็น “Arduino Mega” โดยการเข้าไปในเมนู Tool แล้วเลือก Board เป็น Arduino Mega ก็จะพร้อมที่จะเขียนโปรแกรมไปยัง Board Arduino Mega ได้แล้ว

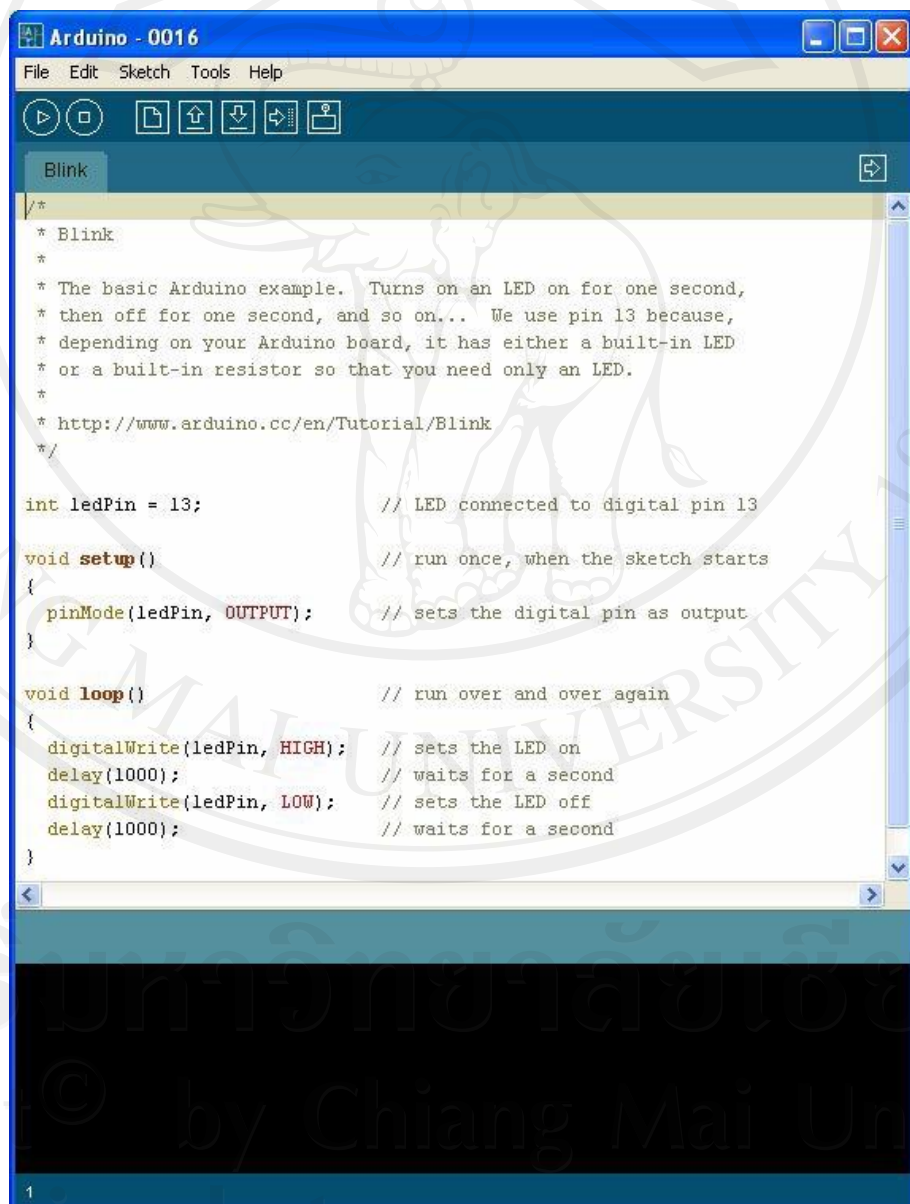
3. เลือกกำหนดหมายเลขพอร์ต สำหรับติดต่อสื่อสารกับบอร์ด ให้ตรงกับหมายเลข Comport ที่ต่อใช้งานไว้จริงในเครื่องคอมพิวเตอร์ PC เช่น ถ้าหมายเลข Comport ของเครื่องคอมพิวเตอร์ PC เป็น COM11 ให้คลิกเมาส์ที่ Tools → Serial Port → COM11 ดังรูป



รูปที่ 10 รูปการตั้งค่ากำหนดหมายเลขพอร์ต สำหรับติดต่อสื่อสารกับบอร์ด “Arduino Mega”

จากรูปที่ 10 แสดงรูปการตั้งค่ากำหนดหมายเลขพอร์ต สำหรับติดต่อสื่อสารกับบอร์ด “Arduino Mega” โดยเลือกเป็น Com11

4. ทดสอบเขียนโปรแกรม โดยคลิกเมาส์ที่ File → New แล้วพิมพ์โปรแกรมทดสอบ หรืออาจใช้การสั่งเปิดไฟล์ตัวอย่างที่สร้างไว้แล้วขึ้นมาแทนก็ได้ โดยในที่นี้ขอแนะนำให้ทดสอบ ด้วยโปรแกรมไฟกระพริบ โดยให้เลือก “File → sketchbook → Examples → Digital Blink” ซึ่งจะได้อีกดังรูป



The screenshot shows the Arduino IDE window titled "Arduino - 0016". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu bar is a toolbar with icons for running, saving, opening, and other functions. The main text area displays the "Blink" example code. The code is as follows:

```

/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()               // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

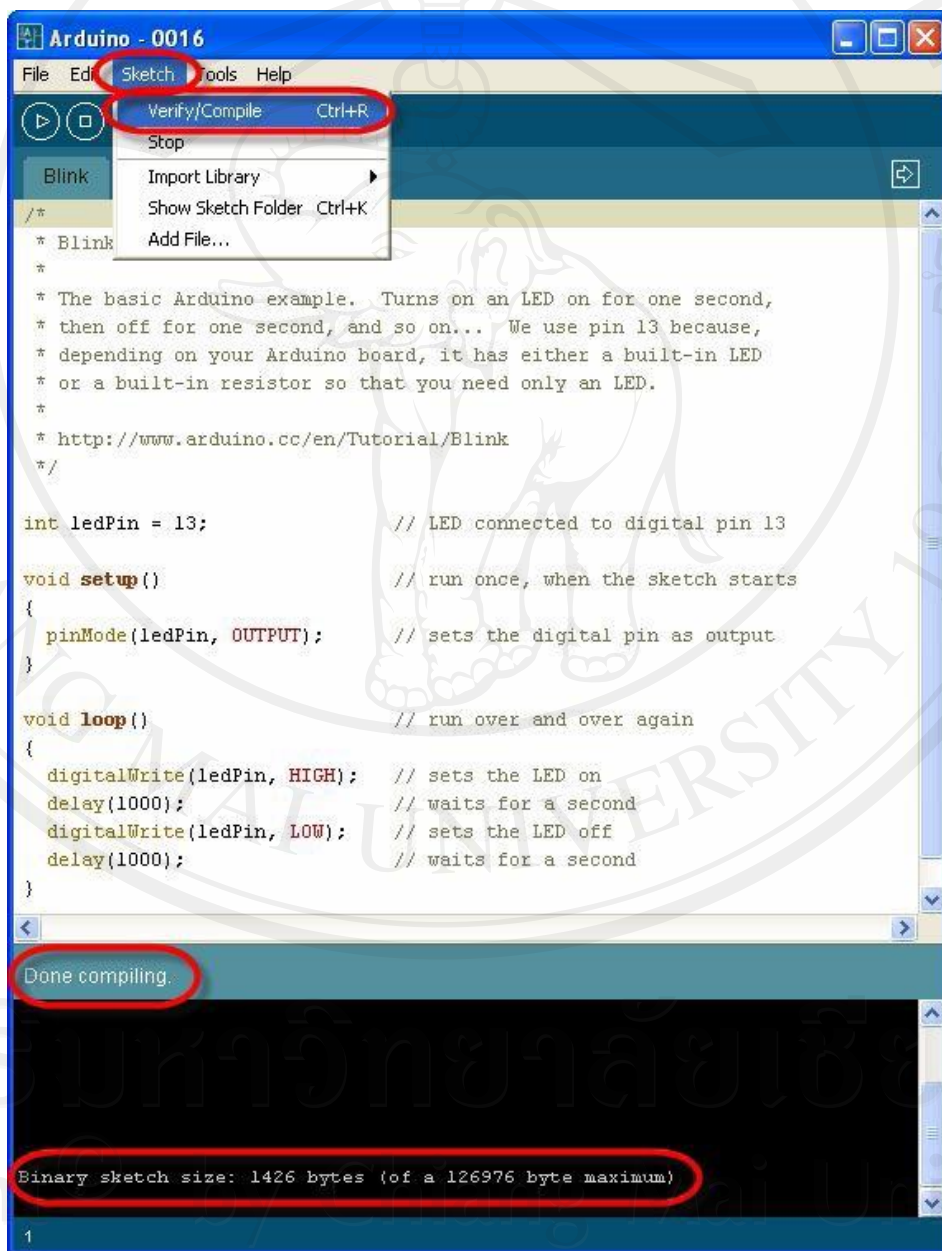
void loop()                // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}
  
```

The status bar at the bottom shows the line number 1.

รูปที่ 11 รูปการทดสอบเขียนโปรแกรม

จากรูปที่ 11 แสดงรูปการทดสอบเขียนโปรแกรม เป็นการเขียนโปรแกรมให้ทำงานเป็นไฟกะพริบที่ขา 13 ของบอร์ด ET-EASY MEGA1280

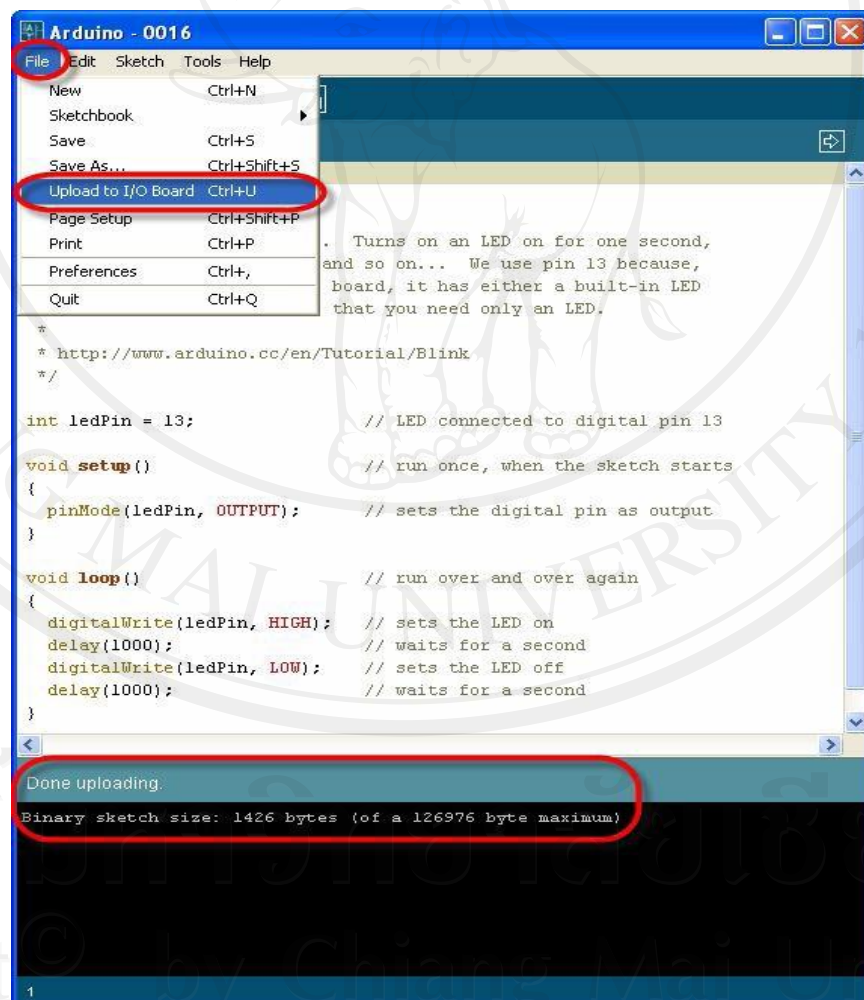
5. สั่งแปลโปรแกรมโดยคลิกเมาส์ที่ “Sketch → Verify/Compile” เพื่อตรวจสอบคำสั่งต่างๆในโปรแกรมว่าถูกต้องหรือไม่ ดังตัวอย่าง



รูปที่ 12 รูปการสั่งแปลโปรแกรมเพื่อตรวจสอบคำสั่งต่าง ๆ

จากรูปที่ 12 แสดงการสั่งแปลโปรแกรมเพื่อตรวจสอบคำสั่งต่าง ๆ สั่งแปลโปรแกรมโดยคลิกเมาส์ที่ Sketch → Verify/Compile เพื่อตรวจสอบคำสั่งต่างๆในโปรแกรมว่าถูกต้องหรือไม่ดังตัวอย่าง

6. สั่ง Download Code ให้กับบอร์ด โดยคลิกเมาส์เลือกที่ “File → Upload to I/O Board” แล้วรอสักครู่จนโปรแกรมทำงานเสร็จ หลังจากที่ทำกร Upload Code ให้กับบอร์ดเป็นที่เรียบร้อยแล้วบอร์ดก็จะเริ่มต้นทำงานตามคำสั่งที่เขียนไว้ในโปรแกรมทันที โดยจะสังเกตเห็น LED กระพริบติดและดับ สลับกันไปมา ด้วยความเร็วประมาณ 1 วินาที ตลอดเวลา ซึ่งควรได้ผลดังรูป



รูปที่ 13 รูปการสั่ง Download Code ให้กับบอร์ด “Arduino Mega”

จากรูปที่ 13 แสดงการสั่ง Download Code ให้กับบอร์ด “Arduino Mega” โดยคลิกเมาส์เลือกที่ “File Upload to I/O Board” แล้วรอสักครู่จนโปรแกรมทำงานเสร็จ หลังจากนั้นทำการ Upload Code ให้กับบอร์ดเป็นที่เรียบร้อยแล้วบอร์ดก็จะเริ่มต้นทำงานตามคำสั่งที่เขียนไว้ในโปรแกรมทันที

ในการพัฒนาระบบระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยอุปกรณ์มือถือแบบ Wi-Fi เป็นระบบที่ใช้มือถืออัจฉริยะในการควบคุมอุปกรณ์ไฟฟ้าผ่านระบบไร้สายที่เป็น Wi-Fi ใช้อุปกรณ์มือถือระบบแอนดรอยด์ ในการระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยอุปกรณ์มือถือแบบ Wi-Fi ในครั้งนี้ ส่วนประกอบของการสร้างระบบจะประกอบด้วย มือถือระบบแอนดรอยด์ Wi-Fi Shield for Arduino และ Arduino

2.5 Wi-Fi Shield for Arduino

Wi-Fi Shield for Arduino [10] ใช้ โมดูล RN171 Wi-Fi เพื่อให้ใช้งานกับ Arduino / Seeeduno ด้วยฟังก์ชันอนุกรมของอิเธอร์เน็ต ใช้เพียงสองขาที่จะต่ออุปกรณ์ Arduino / Seeeduno ด้วยมาตรฐาน 802.11b/g นอกจากนี้ยังมีเสาอากาศที่เป็นอิสระซึ่งสามารถครอบคลุมช่วงกว้างและส่งสัญญาณได้อย่างมีประสิทธิภาพ และสนับสนุนสำหรับ TCP, UDP และโปรโตคอล FTP การสื่อสารด้วย Wi-Fi Shield สามารถตอบสนองความต้องการของส่วนใหญ่เป็นโครงการเครือข่ายไร้สายเช่นเดียวกับเครือข่ายภายในบ้านอัจฉริยะ, การควบคุมหุ่นยนต์หรือสถานีตรวจอากาศส่วนบุคคล ฯลฯ มีการเตรียมคำสั่งที่ง่ายสำหรับการตั้งค่านี้เพื่อให้คุณสามารถใช้รหัสที่มีความรัดกุมในการเรียกใช้ฟังก์ชัน

คุณสมบัติ

- ใช้ได้กับ Arduino, Seeeduno, Arduino Mega ,Seeeduno Mega
- ความเร็วอัตราการรับส่งข้อมูลได้ถึง 1 Mbps สำหรับ UART, 2 Mbps ในโหมด SPI Slave
- UART พอร์ตอนุกรมของ RN171 ต่อกับ Shield สามารถเชื่อมต่อกับ Arduino โดยจัมเปอร์ SPI ขาของ RN171 ออกแบบให้เชื่อมต่อกับ Arduino ได้โดยตรงเมื่อต่อกับ Shield
- มีขั้วต่อสองขั้วที่ทำให้ง่ายต่อการเชื่อมต่อ I2C และดิจิทัลโมดูล IO ของตัว Arduino
- การตรวจสอบความปลอดภัย Wi-Fi WEP-128, WPA-PSK (TKIP), WPA2-PSK (AES)
- มีโครงสร้างภายในที่ใช้งานเครือข่าย: DHCP client, DNS Client, ARP, ICMP ping, FTP, Telnet,

ที่มา: Razvan Turiae, RN-171 Data Sheet : page 4

จากรูปที่ 14 แสดงรูป Wi-Fi Shield for Arduino มีการทำงานในส่วนหลัก ๆ จะเป็นส่วนของเสาอากาศที่ใช้ในการเชื่อมต่อสัญญาณกับอุปกรณ์มือถือทาง Wi-Fi หรืออุปกรณ์อื่น ๆ ที่สามารถเชื่อมต่อกับอุปกรณ์ได้ มีส่วนของการเชื่อมต่อกับไมโครคอนโทรลเลอร์โดยสื่อสารกันระหว่างไมโครคอนโทรลเลอร์กับ Wi-Fi Shield for Arduino ด้วยการสื่อสารแบบ RS232 ส่วนของการเชื่อมต่อกับอุปกรณ์เสริมในส่วนที่เป็นเซนเซอร์ ส่วนที่เชื่อมต่อกับไฟเลี้ยงวงจรการทำงานของแต่ละขาของ Wi-Fi Shield for Arduino มีดังต่อไปนี้

- ขาที่ 1 เป็นขา GND ที่ต่อกับกราวด์ หรือ 0 โวลท์
- ขาที่ 2 เป็นขา Not Used ไม่สามารถต่อใช้งานได้
- ขาที่ 3 เป็นขา Not Used ไม่สามารถต่อใช้งานได้
- ขาที่ 4 เป็นขา GPIO 9 มีข้อต่อให้ทำงานในโหมด Adhoc
- ขาที่ 5 เป็นขา GPIO 8 เป็นขาอินพุต เอาท์พุต
- ขาที่ 6 เป็นขา GPIO 7 เป็นขาอินพุต เอาท์พุต
- ขาที่ 7 เป็นขา GPIO 6 เป็นขาอินพุต เอาท์พุต

- ขาที่ 8 เป็นขา GPIO 5 เป็นขาอินพุต เอาท์พุต
- ขาที่ 9 เป็นขา GPIO 4 เป็นขาอินพุต เอาท์พุต
- ขาที่ 10 เป็นขาไฟเลี้ยง 3.3 โวลต์
- ขาที่ 11 เป็นขา GPIO 3 เป็นขาอินพุต เอาท์พุต
- ขาที่ 12 เป็นขา GPIO 2 เป็นขาอินพุต เอาท์พุต
- ขาที่ 13 เป็นขา GPIO 1 เป็นขาอินพุต เอาท์พุต
- ขาที่ 14 เป็นขา GND ขากราวด์
- ขาที่ 15 เป็นขา Not Used ไม่สามารถต่อใช้งานได้
- ขาที่ 16 เป็นขา Not Used ไม่สามารถต่อใช้งานได้
- ขาที่ 17 เป็นขา Not Used ไม่สามารถต่อใช้งานได้
- ขาที่ 18 เป็นขา Not Used ไม่สามารถต่อใช้งานได้
- ขาที่ 19 เป็นขา Not Used ไม่สามารถต่อใช้งานได้
- ขาที่ 20 เป็นขา GND ขากราวด์
- ขาที่ 21,22,23 เป็นขา GND ขากราวด์
- ขาที่ 24 เป็นขาต่อเสาอากาศ
- ขาที่ 25,26,27,28 เป็นขา GND ขากราวด์
- ขาที่ 29 Sensor 0 เป็นขาต่ออุปกรณ์ตรวจจับสัญญาณภายนอก
- ขาที่ 30 Sensor 1 เป็นขาต่ออุปกรณ์ตรวจจับสัญญาณภายนอก
- ขาที่ 31 Sensor 2 เป็นขาต่ออุปกรณ์ตรวจจับสัญญาณภายนอก
- ขาที่ 32 Sensor 3 เป็นขาต่ออุปกรณ์ตรวจจับสัญญาณภายนอก
- ขาที่ 33 Sensor Power เป็นขาแรงดันออกของอุปกรณ์ตรวจจับสัญญาณ
- ขาที่ 34 VDD_3.3V_RF เป็นขาต่อไฟเลี้ยง 3.3 โวลต์ส่วนที่ใช้เลี้ยงส่วนของ RF
- ขาที่ 35 Sensor 4 เป็นขาต่ออุปกรณ์ตรวจจับสัญญาณภายนอก
- ขาที่ 36 Sensor 5 เป็นขาต่ออุปกรณ์ตรวจจับสัญญาณภายนอก
- ขาที่ 37 Sensor 6 เป็นขาต่ออุปกรณ์ตรวจจับสัญญาณภายนอก
- ขาที่ 38 Sensor 7 เป็นขาต่ออุปกรณ์ตรวจจับสัญญาณภายนอก
- ขาที่ 39 เป็นขา GND ขากราวด์
- ขาที่ 40 เป็นขา Reset ที่ใช้ในการรีเซ็ตการทำงานของโมดูล
- ขาที่ 41 เป็นขา Force_Awake เป็นขาอินพุต เป็นขาต่ออุปกรณ์เสริม Module Awake

- ขาที่ 42 เป็นขา GPIO 14 เป็นขาอินพุต เอาท์พุต
- ขาที่ 43 เป็นขา UART_RTS เป็นขาเอาท์พุต มีกระแสขับได้ 8 mA
- ขาที่ 44 เป็นขา UART_CTS เป็นขาอินพุต
- ขาที่ 45 เป็นขา UART_RX เป็นขาอินพุต
- ขาที่ 46 เป็นขา UART_TX เป็นขาเอาท์พุต
- ขาที่ 47 เป็นขา GND ขากราวด์
- ขาที่ 48 เป็นขา SREG_3V3_CTRL เป็นขาควบคุมแรงดัน
- ขาที่ 49 เป็นขา VDD-BATT เป็นขาต่อแรงดันต่ำและต้องการขยายแรงดัน

การตั้งค่าให้บอร์ด Wi-Fi Shield for Arduino พร้อมทำงานมีวิธีการในการเตรียมการให้พร้อมใช้งานโดยการทำการตั้งค่าการทำงานต่าง ๆ มีการเข้าโหมดการตั้งค่าโดยการตั้งค่าของบอร์ด โดยขั้วต่อ J6 อยู่ในสถานะต่อเข้ากับ +5 โวลต์ บอร์ดก็จะอยู่ในโหมด Adhoc การเข้าโหมดการตั้งค่าของบอร์ดจะมีวิธีการโดยที่การเชื่อมต่อผ่าน เครื่องคอมพิวเตอร์ แล้วทำการเชื่อมต่อกับบอร์ด ใช้คำสั่ง ipconfig จะได้ IP Address ของบอร์ด ทำการเชื่อมต่อคอมพิวเตอร์จาก IP Address นั้นและใช้คำสั่ง telnet 169.254.1.1 2000 จะมีคำตอบรับจากบอร์ดว่า “HELLO” ให้เราส่งรหัส \$\$\$ ดิ๊งกันอย่างรวดเร็ว จะเราก็จะสามารถทำการตั้งค่าต่าง ๆ ของบอร์ดได้แล้วโดยมีคำสั่งในการเตรียมการหลายคำสั่งแต่เราจะทำการตั้งค่าเฉพาะบางส่วนที่จำเป็นเท่านั้น

1. set ip address <192.168.12.25> ตั้งค่า server IP address
2. set ip netmask 255.255.0.0
3. set ip dhcp 0 ไม่ให้ทำงานในโหมด DHCP
4. set broadcast port <5555> ตั้งค่า Server Port
5. set dns name <mynetwork> ตั้งชื่อ Wi-Fi
6. set uart baud <9600> ตั้งความเร็วการสื่อสารอนุกรมกับ Arduino เป็น 9600 B/S

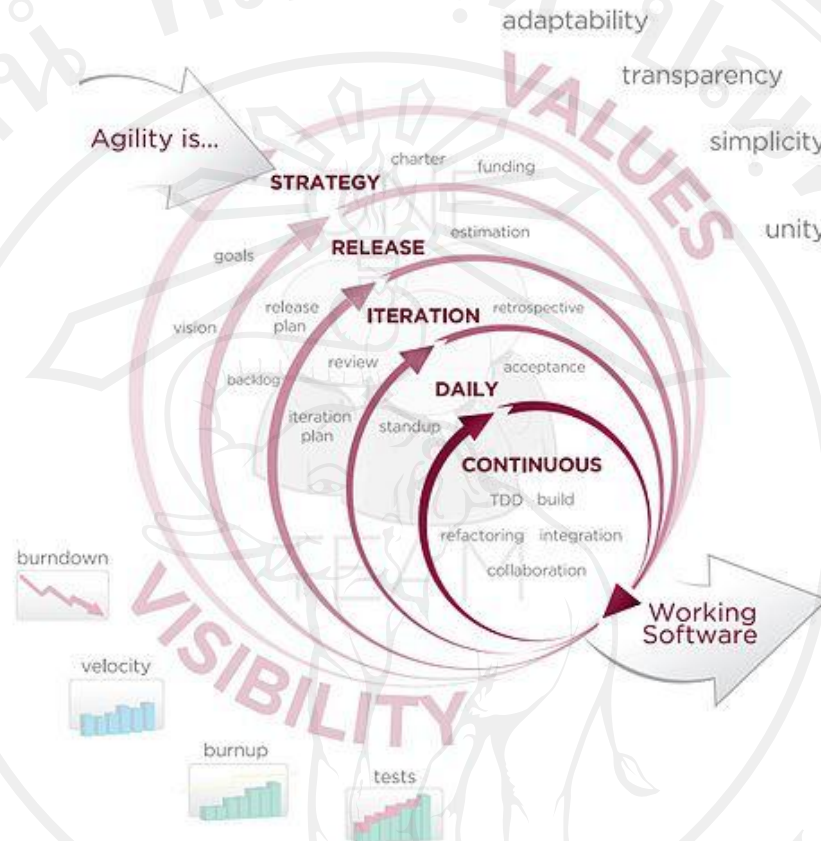
เมื่อทำการตั้งค่าต่าง ๆ ตามลำดับ บอร์ด ก็พร้อมที่จะทำการเชื่อมต่อกับ อุปกรณ์มือถือและเชื่อมต่อกับ Arduino ซึ่งจะใช้ในการศึกษาในหัวข้อคั่นคว่ำอิสระในครั้งนี้โดยจะมีการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยอุปกรณ์มือถือแบบ Wi-Fi การเชื่อมต่อระหว่างมือถือกับบอร์ด Wi-Fi Shield for Arduino จะเป็น IP Address 192.168.12.25 และ Port 5555 ตัว Wi-Fi Shield for Arduino

จะติดต่อกับบอร์ด ET-EASY MEGA1280 ผ่านทางการสื่อสารอนุกรม ความเร็ว 9600 bps คำสั่งในการสั่งการควบคุมอุปกรณ์ไฟฟ้า ด้วยมือถือก็จะส่งสัญญาณผ่านช่องทางดังกล่าวตามลำดับ

2.6 การพัฒนาซอฟต์แวร์โดยใช้แบบจำลองแบบอไจล์ (Agile Development Model)

Agile model (Agile Software Development Model, 2009) [11] เป็นโมเดลที่ออกแบบให้มีความรวดเร็ว ยืดหยุ่น พร้อมทั้งจะรับกับความเปลี่ยนแปลงความต้องการของระบบตลอดเวลา เพื่อลดความเสี่ยงในการพัฒนาซอฟต์แวร์ โดยการแบ่งการพัฒนาออกเป็นเวลาออกเป็นช่วง ๆ แต่ละช่วงยาวนานไม่มากนัก ไม่เกิน 1 เดือน หรือ 4 สัปดาห์ การพัฒนาจะดำเนินการอย่างต่อเนื่อง ถึงแม้ว่าเมื่อไรมากระทบก็ไม่สนใจ เมื่อมีความเปลี่ยนแปลง ก็จะพัฒนาให้สามารถรองรับกับความเปลี่ยนแปลงนั้นได้ อย่างไม่มีข้อจำกัดตายตัว ในการพัฒนาจะเน้นการพูดคุยกันในทีมงาน และผู้ใช้มากกว่าเน้นกระบวนการหรือ เครื่องมือ การทำงานจะยึดที่ผลิตหรือตัวซอฟต์แวร์เป็นหลัก ไม่ค่อยเน้นการจัดทำเอกสาร เน้นที่ความสัมพันธ์ของทีมงานและการสื่อสารเป็นหลัก เพื่อให้ได้ความต้องการมาครบถ้วนและพร้อมที่จะยอมรับความเปลี่ยนแปลงเพิ่มเติมของความต้องการ หัวใจของ Agile model ได้แก่ เน้นความพึงพอใจของลูกค้า โดยการส่งมอบซอฟต์แวร์ให้ลูกค้าอย่างต่อเนื่องทุก 2 สัปดาห์ พร้อมรับกับความต้องการที่เปลี่ยนแปลงเสมอ ทีมพัฒนาระบบจะดำเนินโครงการที่ปฏิบัติงานของลูกค้ามีการพบหน้ากันทุกวันจนกว่า โครงการจะเสร็จ มีการประชุมพบหน้ากันสม่ำเสมอ ทีมงานมีอำนาจในการตัดสินใจเต็มที่ วัดความก้าวหน้าของงานกันที่ตัวซอฟต์แวร์ การทำงานใช้กระบวนการที่เรียบง่าย เน้นความคุณภาพชีวิตของทีมงาน มีเทคนิคต่าง ๆ ก็นำมาแลกเปลี่ยนกัน เน้นเทคนิคการออกแบบที่ง่าย ไม่ซับซ้อน ทำให้บำรุงรักษาปรับเปลี่ยนระบบได้ง่ายอาจจะมองว่า Agile model เป็นส่วนขยายของกระบวนการพัฒนาซอฟต์แวร์อื่น ๆ ที่มีอยู่เดิม โดยใช้ Agile เข้าไปกำกับโดยเลือกเอาส่วนที่สำคัญ ๆ กิจกรรมไหนควรทำ ไม่ควรทำ แล้วนำมาจัดลำดับให้เหมาะสม สำหรับวิธีของ Agile การที่ควรรู้จัก ได้แก่ Agile UP, XP-eXtream programming, FDD-Feature Driven Development, และ Scrum

AGILE DEVELOPMENT



ACCELERATE DELIVERY

รูปที่ 15 การพัฒนาซอฟต์แวร์โดยใช้โมเดล

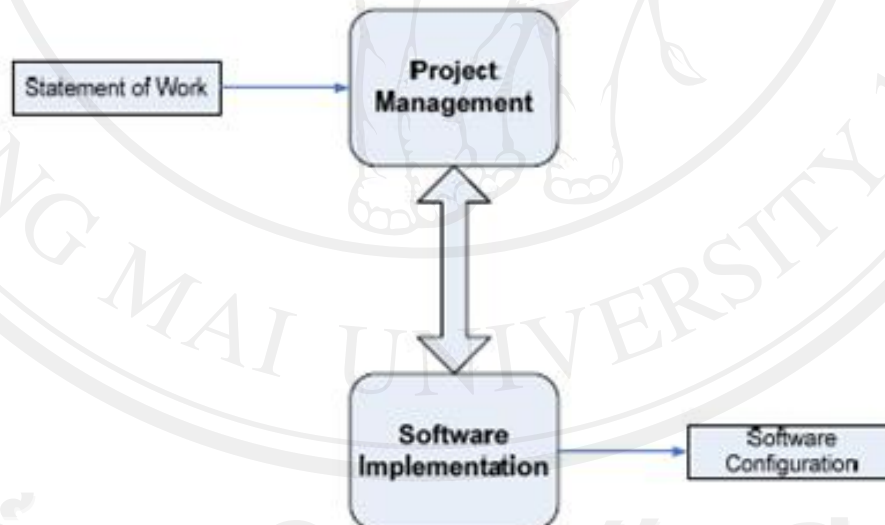
(Agile software development, 2010)

สำหรับระบบที่จัดทำขึ้น ได้มีการนำ Agile Model Development มาใช้ คือ ระบบที่จัดสร้าง ขึ้น มีการเปลี่ยนแปลงความต้องการของระบบก่อนข้างบอຍอีกทั้งยังเป็นการเพิ่มคุณภาพให้กับซอฟต์แวร์ด้วย โดยการติดต่อกับผู้ให้หรือผู้ให้ความต้องการของระบบให้เห็นผลอยู่ทุกขั้นตอนในการพัฒนา ทำการประชุม หรือพูดคุย อีกทั้งยังเป็นระบบที่มีระยะเวลาในพัฒนาที่สั้น จึงมีการใช้ Agile Model Development เป็นโมเดลในการพัฒนาระบบที่มีประสิทธิภาพมากยิ่งขึ้น

2.7 กระบวนการพัฒนาซอฟต์แวร์โดยใช้มาตรฐาน ISO 29110

กระบวนการของ ISO 29110 [12] จะเน้นให้ผู้ประกอบการซึ่งอาจจะเป็นผู้ประกอบการอิสระผู้ประกอบการขนาดเล็กที่มีบุคลากรไม่เกิน 25 คน หรือหน่วยงานทางด้านซอฟต์แวร์ที่อยู่ในองค์กรขนาดใหญ่ให้มีกระบวนการในการพัฒนาซอฟต์แวร์ที่เป็นระบบ และเข้าสู่กระบวนการสากล โดยจะเป็นการเริ่มต้นในเชิงกิจกรรมของการปรับปรุงกระบวนการ หรือ SPI (Software Process Improvement) ทาง ISO 29110 ได้ให้ความสำคัญในกระบวนการที่จะต้องทำการปรับปรุงให้เป็นระบบและเป็นสากล 2 กระบวนการหลัก คือ กระบวนการด้านการบริหารโครงการ (Project Management) และกระบวนการด้านการสร้างซอฟต์แวร์ (Software Implementation) ซึ่งจะประกอบด้วยกระบวนการย่อยๆภายในอีกทั้งสองกระบวนการได้ ถูกออกแบบให้เหมาะสมกับผู้ประกอบการขนาดเล็กจึงมีความเหมาะสมในการประยุกต์ใช้ได้ทันที โดยได้กำหนดขนาดของกระบวนการให้เหมาะสมกับองค์กรขนาดเล็ก จึงไม่สร้างปัญหาในการปรับใช้งานให้เข้ากับองค์กร

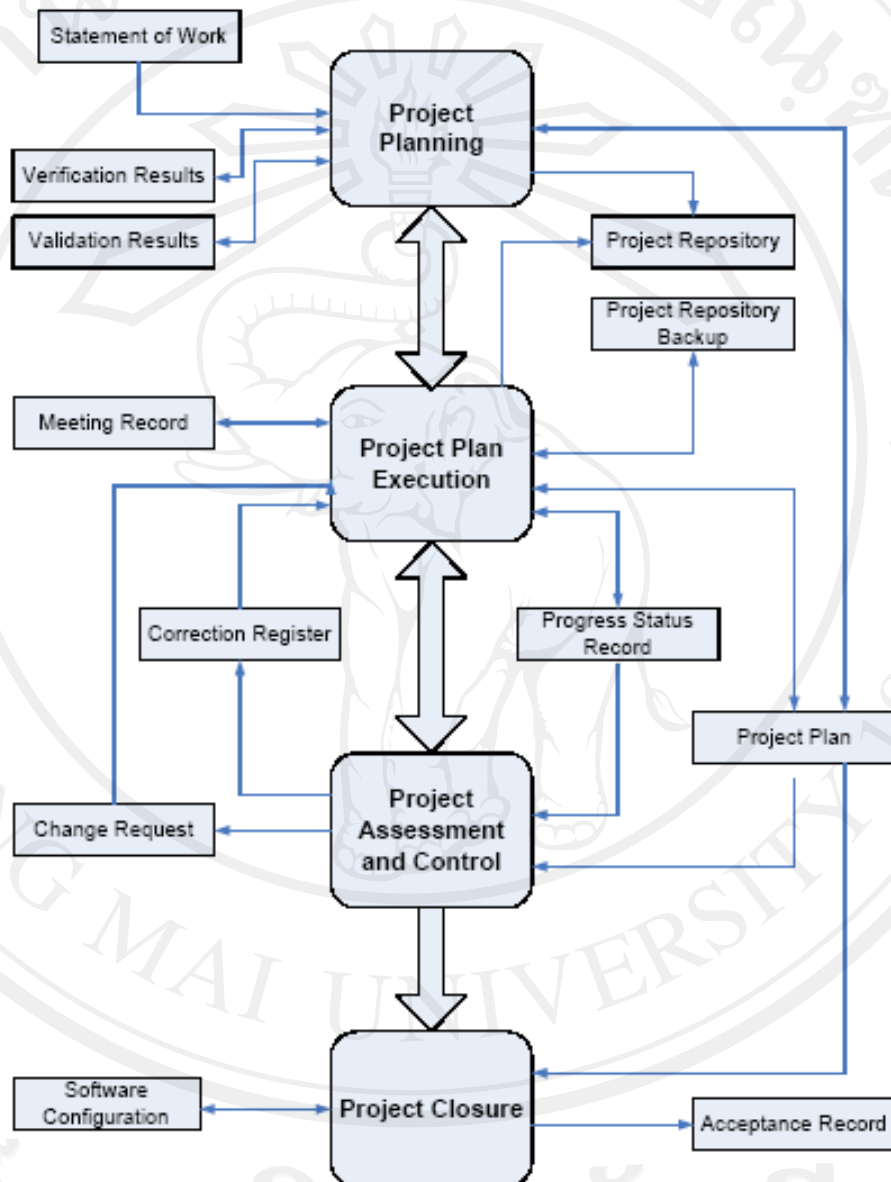
1) กระบวนการด้านการบริหารโครงการ (Project Management)



รูปที่ 16 แผนภาพแสดงกิจกรรมหลัก (Basic Profile) ตามมาตรฐาน ISO/IEC 29110

ที่มา: <http://www.center4vse.net>

จากรูปที่ 16 แสดงแผนภาพแสดงกิจกรรมหลัก (Basic Profile) ตามมาตรฐาน ISO/IEC 29110 โดยมีองค์ประกอบหลักอยู่ 2 ส่วนคือ Project Management และ Software Implementation



รูปที่ 17 แผนภาพแสดงกิจกรรมของกระบวนการบริหารโครงการ ตามมาตรฐาน ISO 29110

ที่มา: <http://www.center4vse.net>

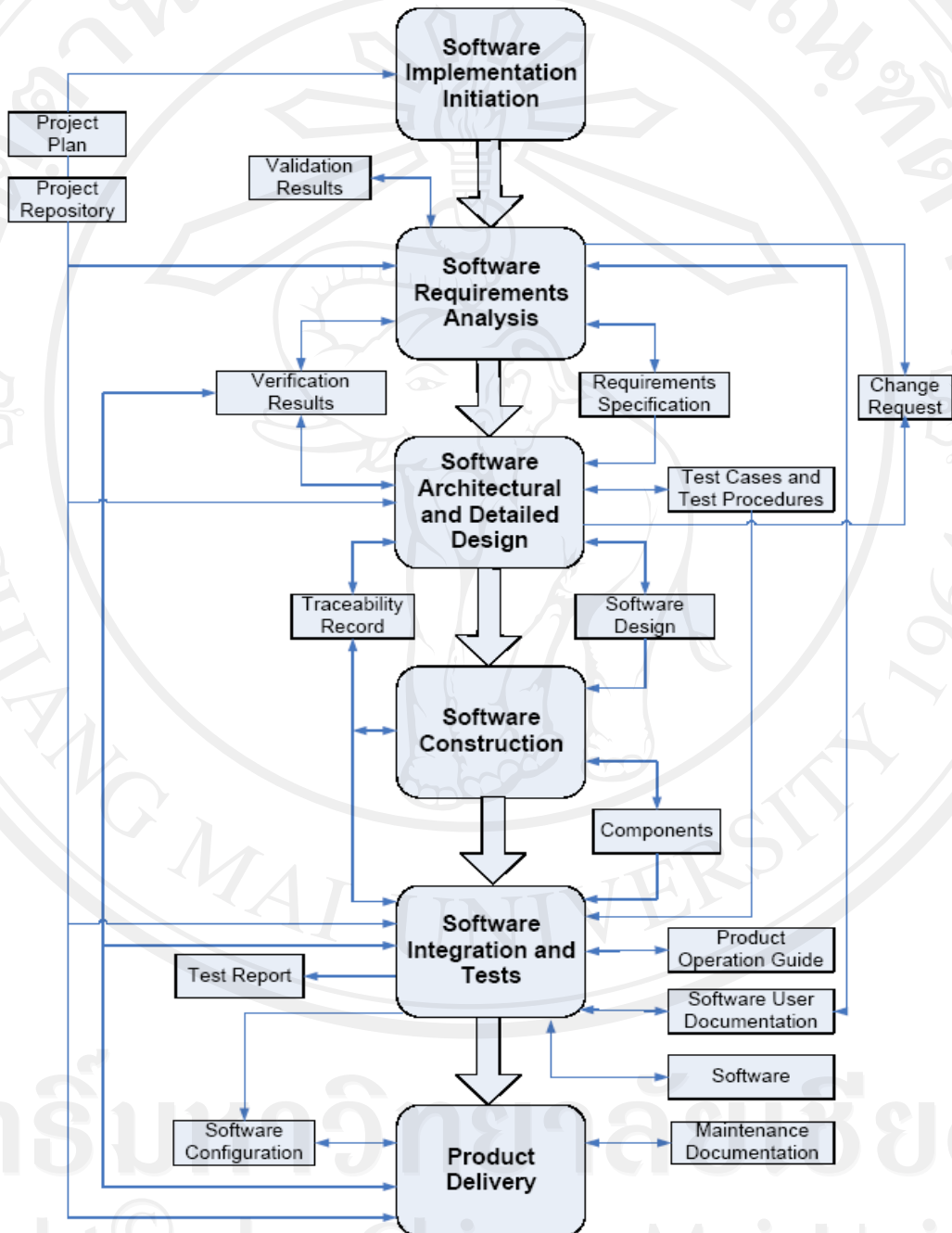
สำหรับกระบวนการบริหารโครงการตามมาตรฐาน ISO29110 นั้นมีกิจกรรมหลักทั้งสิ้น 4 กิจกรรมหลักซึ่งประกอบไปด้วย

- PM.1 Project Planning
- PM.2 Project Plan Execution
- PM.3 Project Assessment and Control
- PM.4 Project Closure

โดยมีวัตถุประสงค์ในการดำเนินการตามกิจกรรมทั้งสิ้น 7 ประการซึ่งประกอบไปด้วย

- PM.O1. Project Plan จะต้องสร้างมาจากเอกสารความต้องการโดยในแผนต้องมีการระบุทรัพยากรที่ต้องใช้และเวลาในการดำเนินการแต่ละขั้นตอนโดยประมาณ และแผนที่ได้จะต้องมีการรับทราบและยอมรับในแผน ก่อนจะเริ่มนำไปใช้งานในขั้นตอนถัดไป
- PM.O2. การดำเนินการโครงการจะต้องถูกติดตามความก้าวหน้าอย่างต่อเนื่องโดยเทียบกับแผนการดำเนินการโครงการและมีการบันทึกผลเก็บไว้ในแบบบันทึกการติดตามความก้าวหน้าโครงการ
- PM.O3. ขั้นตอนการร้องขอเปลี่ยนแปลง (Change Request) จะต้องมีการระบุวิธีการรับเรื่องและขั้นตอนการวิเคราะห์ผลกระทบ ทั้งในด้าน ต้นทุน เวลา และ ปัจจัยในทางเทคนิค
- PM.O4. มีการสรุปการประชุมระหว่างทีมพัฒนาและลูกค้า เพื่อกำหนดข้อตกลงในการติดตามความก้าวหน้าของโครงการ
- PM.O5. มีการระบุความเสี่ยง (Risk) ที่อาจจะเกิดในระหว่างดำเนินการและจัดการโครงการ
- PM.O6. มีการกำหนดแผนการจัดการรุ่นซอฟต์แวร์ (Version Control Plan) โดยมีการกำหนด Baseline และมีการควบคุมการแก้ไขของทีมพัฒนา และการนำออกจากระบบเพื่อส่งมอบลูกค้า โดยในแผนต้องระบุสถานที่เก็บ และ โครงสร้างของสารบบ (Directory Structure)
- PM.O7. มีแผนการประกันคุณภาพซอฟต์แวร์ (Software Insurance) เพื่อให้มั่นใจกระบวนการดำเนินการ และผลการดำเนินการมีความสอดคล้องกับแผนการดำเนินการและข้อตกลงความต้องการ

2) กระบวนการด้านการ สร้างซอฟต์แวร์ (Software Implementation)



รูปที่ 18 แผนภาพแสดงกิจกรรมในกระบวนการพัฒนาซอฟต์แวร์ตามมาตรฐาน ISO 29110

ที่มา: <http://www.center4vse.net>

สำหรับกระบวนการพัฒนาซอฟต์แวร์ตามมาตรฐาน ISO29110 นั้นมีกิจกรรมหลักทั้งสิ้น 6 กิจกรรมหลักซึ่งประกอบไปด้วย

- SL.1 Software Implementation Initiation
- SL.2 Software Requirements Analysis
- SL.3 Software Architectural and Detailed Design
- SL.4 Software Construction
- SL.5 Software Integration and Tests
- SL.6 Product Delivery

โดยมีวัตถุประสงค์ในการดำเนินการตามกิจกรรมทั้งสิ้น 7 ประการประกอบไปด้วย

- SLO1. งานในแต่ละกิจกรรมจะต้องได้รับการดำเนินการบรรลุเป้าหมายตามแผนการดำเนินการ
- SLO2. ข้อกำหนดความต้องการซอฟต์แวร์ (Software Requirement) จะต้องมีการกำหนดให้ชัดเจนถึงวิธีการวัดผลและข้อตกลงการทดสอบ โดยวิธีการวัดผลดังกล่าวจะต้องได้รับการยอมรับจากลูกค้าเพื่อจัดทำเป็นเอกสารและเก็บไว้
- SLO3. เอกสารสถาปัตยกรรมซอฟต์แวร์หรือเอกสารการออกแบบซอฟต์แวร์ที่จัดทำขึ้นจะต้องสามารถอธิบายรายละเอียดเกี่ยวกับซอฟต์แวร์ได้ดี รวมไปถึงจนถึงมีความสัมพันธ์ที่ตรวจสอบย้อนกลับไปยังเอกสารความต้องการซอฟต์แวร์ได้
- SLO4. ส่วนประกอบของซอฟต์แวร์ที่สร้างขึ้นจะต้องผ่านการทดสอบระดับหน่วย (Unit Test) และเป็นไปตามเอกสารความต้องการซอฟต์แวร์และเอกสารการออกแบบ
- SLO5. มีการทดสอบการทำงานร่วมกันของส่วนประกอบทั้งหมดของซอฟต์แวร์ ตาม Test Cases และ Test Procedures ที่ออกแบบไว้พร้อมบันทึกผลการทดสอบไว้ใน Test Report โดยข้อผิดพลาดทั้งหมดที่ตรวจพบจะต้องได้รับการแก้ไขให้เป็นไปตามเอกสารการออกแบบ
- SLO6. โครงร่างซอฟต์แวร์ (Software Configuration) จะต้องตรงกับเอกสารความต้องการซอฟต์แวร์ที่เป็นไปตามข้อตกลงกับลูกค้า ซึ่งรวมถึง คู่มือผู้ใช้ ซึ่งทั้งหมดจะต้องถูกจัดเก็บใน Project Repository และเมื่อมีความต้องการเปลี่ยนแปลงรายการใด Repository แล้วนั้นจะต้องมีการเปิดคำขอเปลี่ยนแปลง
- SLO7. การดำเนินงานทั้งหมดรวมถึงผลผลิตที่ได้จากงานทั้งหมดจะต้องได้รับการตรวจสอบว่าเป็นไปตามเกณฑ์มาตรฐานของ Input Product และ Output Product ของแต่ละกิจกรรมตามมาตรฐานกำหนด ข้อผิดพลาดที่พบจะต้องได้รับการแก้ไขและ

บันทึกไว้ใน Validation Result หรือ Verification Result และจากขั้นตอนดำเนินการตามวัตถุประสงค์ของมาตรฐาน ISO29110 จะได้เอกสารทั้งสิ้น 20 เอกสารโดยมีรายละเอียดดังนี้

เอกสารที่ได้จากขั้นการบริหารโครงการ 8 เอกสาร

- Project Plan คือ เอกสารและตารางเวลาโครงการ
- Progress Status Record คือ เอกสารบันทึกผลความก้าวหน้าโครงการ
- Correction Register คือ เอกสารบันทึกประเด็นปัญหาในโครงการพร้อมแนวทางการแก้ไข
- Meeting Record คือ เอกสารบันทึกการประชุม
- Change Request คือ เอกสารขอเปลี่ยนแปลงความต้องการที่เกิดขึ้นระหว่างดำเนินการ
- Verification Result คือ เอกสารบันทึกผลการทบทวนเอกสาร
- Validation Result คือ เอกสารบันทึกผลการทบทวนเอกสารกับลูกค้า
- Acceptance Record คือ เอกสารการส่งมอบผลิตภัณฑ์ของโครงการ

เอกสารที่ได้จากขั้นตอนการพัฒนาซอฟต์แวร์ 12 เอกสาร

- Requirement Specification คือ เอกสารความต้องการซอฟต์แวร์
- Software Design คือ เอกสารการออกแบบซอฟต์แวร์
- Traceability Record คือ เอกสารแสดงความสัมพันธ์ของผลิตภัณฑ์
- Software Component คือ ผลิตภัณฑ์ซอฟต์แวร์
- Software คือ ผลิตภัณฑ์สำหรับส่งมอบให้กับลูกค้า
- Test Cases and Test Procedures คือ เอกสารจัดทำทดสอบพร้อมวิธีการทดสอบ
- Test Report คือ เอกสารรายงานผลการทดสอบ
- Software User Document คือ เอกสารคู่มือการใช้งานระบบ
- Product Operation Guideline คือ เอกสารคู่มือการติดตั้งและการจัดการซอฟต์แวร์
- Maintenance Document คือ เอกสารกำหนดผลิตภัณฑ์และเครื่องมือที่ใช้ในโครงการ
- Verification Result คือ เอกสารบันทึกผลการทบทวนเอกสาร

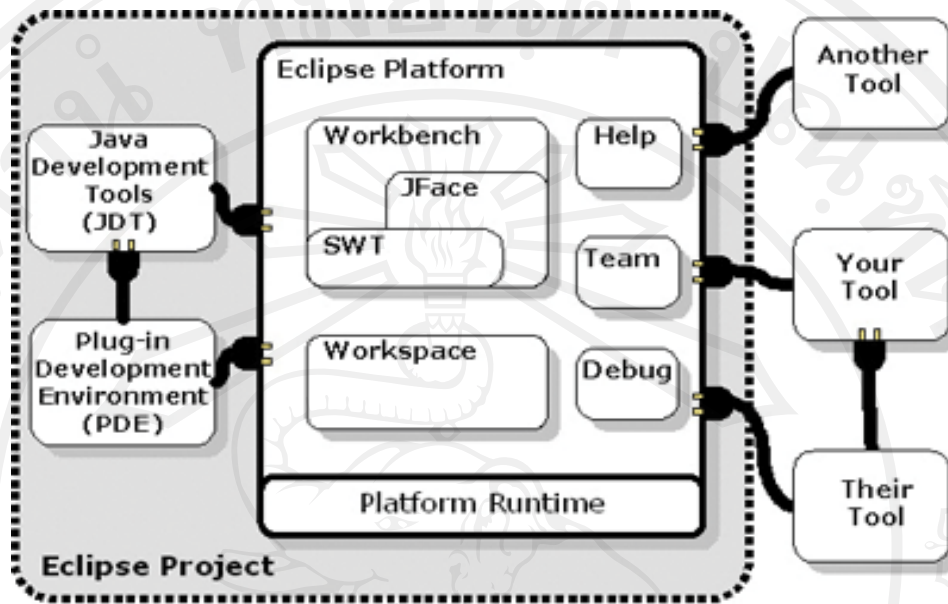
- Validation Result คือ เอกสารบันทึกผลการทบทวนเอกสารกับลูกค้า

2.8 การพัฒนาซอฟต์แวร์โดยใช้ Eclipse

Eclipse [13] คือโปรแกรมที่ใช้สำหรับพัฒนาภาษา Java ซึ่งโปรแกรม Eclipse เป็นโปรแกรมหนึ่งที่ใช้ในการพัฒนา Application Server ได้อย่างมีประสิทธิภาพ และเนื่องจาก Eclipse เป็นซอฟต์แวร์ Open Source ที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ Eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว

Eclipse มีองค์ประกอบหลักที่เรียกว่า Eclipse Platform ซึ่งให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่างๆจากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน และมีส่วนประกอบที่เรียกว่า Plug-in Development Environment (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Eclipse plug-ins ดังนั้นหากต้องการให้ Eclipse ทำงานใดเพิ่มเติม ก็เพียงแค่พัฒนา plugin สำหรับงานนั้นขึ้นมา และนำ Plug-in นั้นมาติดตั้งเพิ่มเติมให้กับ Eclipse ที่มีอยู่เท่านั้น Eclipse Plug-in ที่มีมาพร้อมกับ Eclipse เมื่อเรา download มาครั้งแรกก็คือองค์ประกอบที่เรียกว่า Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียนและ Debug โปรแกรมภาษา Java

ข้อดีของโปรแกรม Eclipse คือ ติดตั้งง่าย สามารถใช้ได้กับ J2SDK ได้ทุกเวอร์ชัน รองรับภาษาต่างประเทศอีกหลายภาษา มี plug in ที่ใช้เสริมประสิทธิภาพของโปรแกรม สามารถทำงานได้กับไฟล์หลายชนิด เช่น HTML, Java, C, JSP, EJB, XML และ GIF และที่สำคัญเป็นฟรีแวร์ (ให้ใช้งานได้ 90 วัน ถ้าจะใช้งานเต็มประสิทธิภาพต้องเสียค่าใช้จ่ายภายหลัง) ใช้งานได้กับระบบปฏิบัติการ Windows, Linux และ MacOS



รูปที่ 19 สถาปัตยกรรมของ Eclipse Platform และ Plug In
ที่มา : <http://www2.electronicproducts.com>

การพัฒนาระบบ เนื่องจากเป็นเครื่องมือที่สะดวกในการใช้งานสามารถที่จะทำการพัฒนาระบบได้อย่างรวดเร็ว ไม่มีปัญหาเรื่องของลิขสิทธิ์ และเครื่องมือยังสามารถปรับปรุงรุ่นใหม่ได้ตลอดเวลา

สรุป ในเนื้อหาในส่วนทฤษฎีและงานวิจัยที่เกี่ยวข้องมีเรื่องต่าง ๆ ในการนำมาประกอบการศึกษาค้นคว้าอิสระในครั้งนี้

2.1 บ้านอัจฉริยะ (Smart Home) จะมีองค์ประกอบของระบบต่าง ๆ ที่อยู่ในระบบบ้านอัจฉริยะเป็นหมวดหมู่อยู่

- 2.1.1 ระบบควบคุมไฟฟ้าแสงสว่าง
- 2.1.2 ระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน
- 2.1.3 ระบบความบันเทิงภายในบ้าน
- 2.1.4 ระบบบริหารพลังงาน และพลังงานสำรอง
- 2.1.5 ระบบสื่อสาร
- 2.1.6 ระบบรักษาความปลอดภัย

2.2 การควบคุมอุปกรณ์ไฟฟ้าแบบไร้สาย

2.2.1 การควบคุมอุปกรณ์ไฟฟ้าไร้สายแบบ Zigbee

2.2.2 การควบคุมอุปกรณ์ไฟฟ้าไร้สายแบบ Bluetooth

2.2.3 การควบคุมอุปกรณ์ไฟฟ้าไร้สายแบบ Wi-Fi

2.3 ระบบโทรศัพท์มือถือที่เลือกใช้ในการค้นคว้าอิสระ เป็นระบบ Android

2.4 Arduino ไมโครคอนโทรลเลอร์ ที่ใช้ในการเชื่อมต่อระบบ Wi-Fi กับอุปกรณ์ไฟฟ้าที่จะทำการควบคุม

2.5 Wi-Fi Shield for Arduino เป็นอุปกรณ์เชื่อมต่อระหว่างมือถือกับ Arduino

2.6 การพัฒนาซอฟต์แวร์โดยใช้แบบจำลองแบบอไจล์ (Agile Development Model) ในการพัฒนาซอฟต์แวร์

2.7 กระบวนการพัฒนาซอฟต์แวร์โดยใช้มาตรฐาน ISO 29110 เป็นกระบวนการควบคุมการผลิตซอฟต์แวร์

2.8 การพัฒนาซอฟต์แวร์โดยใช้ Eclipse เป็นเครื่องมือที่ใช้ในการเขียนโปรแกรมมือถือที่ใช้ในการศึกษาค้นคว้าอิสระในครั้งนี้