คู่มือหุ่นยนต์ MICRO AVR

ระดับผู้เรียน ควรมีความรู้พื้นฐานทางด้านอิเล็กทรอนิกส์มาบ้างแล้ว

แนะนำอุปกรณ์ต่างๆ ที่สำคัญ

1. ไอซีไมโครคอนโทรลเลอร์

ไอซีไมโครคอนโทรลเลอร์ (Microcontroller) คืออะไร

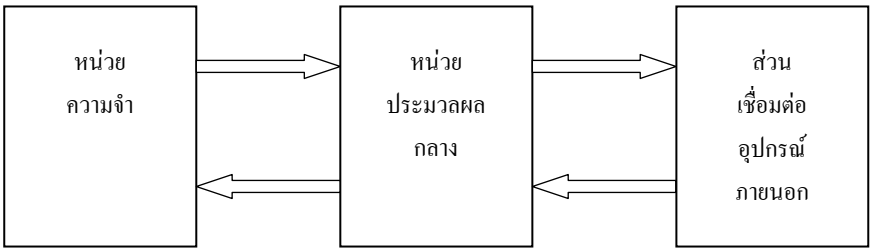
ไอซีไมโครคอนโทรลเลอร์ คือ อุปกรณ์อิเล็กทรอนิกส์ชนิดหนึ่งที่รวมเอาอุปกรณ์ทางด้านอิเล็กทรอนิกส์ต่างๆ เข้ามาไว้ภายในตัวมัน โดยข้อแตกต่างของไอซีคอนโทรลเลอร์กับไอซีโดยทั่วไปก็คือ ภายในตัวมันสามารถแก้ไขเปลี่ยนแปลงคำสั่งในการทำงานของมันได้ โดยอาศัยโปรแกรมภายในหน่วยความจำ ซึ่งเราสามารถเขียนขึ้นมาได้ด้วยตัวเราเอง ทำให้เราสามารถนำไอซีไมโครคอนโทรลเลอร์ไปประยุกต์ใช้งานในด้านต่างๆ ได้อย่างมากมาย เช่น เครื่องซักผ้าอัตโนมัติ, ตู้น้ำหยอดเหรียญ, วิทยุ, โทรทัศน์ เป็นต้น ซึ่งอุปกรณ์ต่างๆ เหล่านี้จะมีไอซีไมโครคอนโทรลเลอร์เป็นหัวใจหลักหรือเป็นสมองให้กับเครื่องใช้ไฟฟ้าต่างๆ เหล่านั้นคอยทำการสั่งให้กับอุปกรณ์ที่ต่อร่วมทำงานอย่างถูกต้อง

ความแตกต่างระหว่างไอซีไมโครคอนโทรลเลอร์กับไมโครโปรเซสเซอร์นั้น ก็คือ ไมโครโปรเซสเซอร์ (Microprocessor) จะเป็นอุปกรณ์ที่ทำหน้าที่ในการประมวลผลเท่านั้น ซึ่งตัวมันจะต้องอาศัยอุปกรณ์อื่นๆ เข้ามาร่วมด้วย เช่น หน่วยความจำ (Memory), ส่วนเชื่อมต่ออุปกรณ์ภายนอก (Interface Unit) เป็นต้น จึงจะทำงานได้อย่างสมบูรณ์ ตัวอย่างก็คือเครื่องคอมพิวเตอร์ที่เราใช้กันอยู่ในปัจจุบันนั่นเอง ส่วนไอซีไมโครคอนโทรลเลอร์นั้นจะมีลักษณะคล้ายกับไมโครโปรเซสเซอร์ก็คือจะมีส่วนประมวลผลเหมือนกัน แต่จะเพิ่มส่วนของหน่วยความจำและส่วนเชื่อมต่ออุปกรณ์เข้ามาไว้ในตัวมันด้วย ดังนั้นไมโครคอนโทรลเลอร์จึงเหมาะที่จะนำมาใช้ในงานควบคุมที่ไม่ต้องการความซับซ้อนมากนัก

โครงสร้างภายในไอซีไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกเป็น 3 ส่วนใหญ่ๆ ด้วยกัน ดังนี้

1. หน่วยประมวลผลกลาง (Central Processor Unit : CPU) เป็นส่วนที่เป็นเหมือนกับสมองของตัวไอซีไมโครคอนโทรลเลอร์เลยทีเดียว โดยหน้าที่ของมัน ก็คือ จะทำการประมวลผลข้อมูลต่างๆ ที่เข้ามา แล้วทำการส่งสัญญาณออกไปยังส่วนต่างๆ เพื่อควบคุมการทำงานให้ตรงตามข้อมูลนั้นๆ
2. หน่วยความจำ (Memory) เป็นส่วนที่ใช้ในการเก็บข้อมูลต่างๆ เอาไว้ เพื่อรอการส่งให้กับหน่วยประมวลผลกลางทำการประมวลผลอีกทีหนึ่ง โดยภายในไอซีไมโครคอนโทรลเลอร์นี้ จะมีหน่วยความจำอยู่ 3 แบบ คือ หน่วยความจำโปรแกรม (Program Memory), หน่วยความจำข้อมูลแรม (RAM data Memory) และหน่วยความจำข้อมูลอีอีพรอม (EEPROM data memory)
    - หน่วยความจำโปรแกรม จะเป็นหน่วยความจำที่ใช้ในการเก็บรักษาคำสั่งควบคุมต่างๆ ที่ผู้พัฒนาโปรแกรมเขียนขึ้น โดยหน่วยประมวลผลกลางจะทำการติดต่อกับส่วนนี้ เพื่อดึงไปประมวลผลและส่งคำสั่งไปควบคุมส่วนอื่นๆ ต่อไป โดยหน่วยความจำโปรแกรมนี้จะคงอยู่ ถึงแม้ว่าจะไม่มีไฟเลี้ยงให้กับตัวไอซีก็ตาม
    - หน่วยความจำข้อมูลแรม จะเป็นหน่วยความจำที่ไอซีไมโครคอนโทรลเลอร์ทุกตัวต้องมีเลยทีเดียว เพราะจะใช้ในการเก็บข้อมูลเกี่ยวกับการประมวลผลทั้งในระหว่างและหลังการประมวลผล แต่ข้อมูลต่างๆ เหล่านี้จะหายไป เมื่อไม่มีไฟเลี้ยงให้กับไอซี
    - หน่วยความจำข้อมูลอีอีพรอม จะเป็นหน่วยความจำพิเศษ ซึ่งไอซีไมโครคอนโทรลเลอร์บางตัวมีและบางตัวไม่มี มีไว้สำหรับเก็บข้อมูลที่ต้องการเก็บรักษาเป็นพิเศษ เมื่อไม่มีไฟเลี้ยงตัวไอซีข้อมูลเหล่านี้ก็จะยังคงอยู่จนกว่าจะมีการเขียนทับลงไปใหม่

3. ส่วนเชื่อมต่ออุปกรณ์ภายนอก (Interface Unit) จะเป็นส่วนที่ทำหน้าที่ในการติดต่อกับอุปกรณ์ภายนอก โดยการสั่งงานมาจากหน่วยประมวลผลกลางอีกทีหนึ่ง ซึ่งในส่วนนี้เราสามารถที่จะกำหนดให้เป็นแบบอินพุต (รับข้อมูล) หรือแบบเอาต์พุต (ส่งข้อมูล) ก็ได้ ตามการเขียนโปรแกรมของผู้พัฒนาโปรแกรม



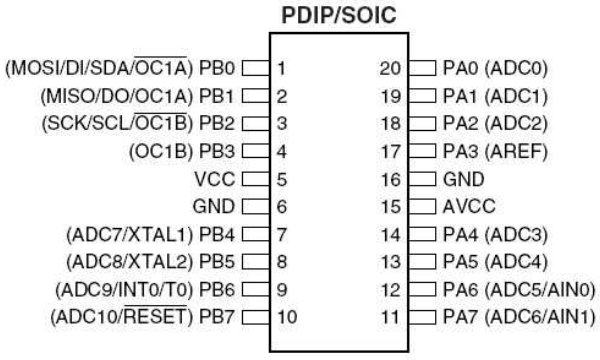
รูปแสดงลักษณะโครงสร้างภายในของไมโครคอนโทรลเลอร์

แนะนำไอซีไมโครคอนโทรลเลอร์ Attiny26

ไอซีไมโครคอนโทรลเลอร์ Attiny26 ตัวนี้ เป็นไอซีที่ทางบริษัท Atmel Corporation พัฒนาขึ้นมา เพื่อรองรับการใช้งานที่หลากหลายตามการเขียนโปรแกรมของผู้พัฒนา โดยที่ราคาไม่แพง มีชุดคำสั่งไม่มาก ทำให้เหมาะแก่การศึกษา

คุณสมบัติของไอซีไมโครคอนโทรลเลอร์ Attiny26

- ไอซีใช้ไฟเลี้ยง ตั้งแต่ 4.5 ถึง 5.5 โวลท์
- ความเร็วในการทำงานสูงสุด 16MHz
- ขนาดของหน่วยความจำโปรแกรม 2 กิโลเวิร์ด
- ขนาดของหน่วยความจำข้อมูลแรม 128 ไบต์
- ขนาดของหน่วยความจำข้อมูลอีอีพรอม 128 ไบต์
- มีจำนวนขาเชื่อมต่อกับอุปกรณ์ภายนอกได้ 16 ช่อง
- สามารถลบและเขียนคำสั่งลงในหน่วยความจำโปรแกรมได้ประมาณ 10,000 ครั้ง

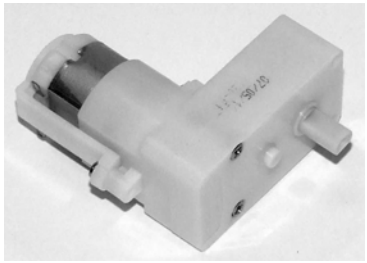


รูปแสดงตำแหน่งและหน้าที่แต่ละขาของไอซี Attiny26

2. มอเตอร์เกียร์ (Gear Motor)

มอเตอร์เกียร์ถือว่า เป็นชุดส่งกำลังที่สำคัญอีกส่วนหนึ่งของหุ่นยนต์เลยทีเดียว โดยมอเตอร์เกียร์จะประกอบไปด้วยมอเตอร์และชุดเฟือง สาเหตุที่จำเป็นต้องมีมอเตอร์เกียร์ก็เนื่องมาจากมอเตอร์ที่มีจำนวนรอบการหมุนที่สูงเกินไป ทำให้ไม่เหมาะสมกับการนำมาใช้งานโดยตรง ดังนั้นจึงมีความจำเป็นที่จะต้องมีชุดเฟืองเกียร์ เพื่อลดจำนวนรอบในการหมุนลง และนอกจากจะลดจำนวนรอบในการหมุนแล้วยังเป็นตัวช่วยในการทำให้เกิดแรงบิดขึ้น ทำให้หุ่นยนต์สามารถขับเคลื่อนตัวไปได้ ซึ่งในการบอกคุณสมบัติของมอเตอร์

เกียร์นั้นจะมีการบอกอัตราทดของเฟืองเกียร์ด้วย เช่น 1:28 นั่นหมายความว่า เมื่อมอเตอร์หมุนไป 28 รอบ ตัวเฟืองเกียร์อันสุดท้าย จะหมุนเพียง 1 รอบ เป็นต้น



รูปลักษณะของมอเตอร์เกียร์ขนาดเล็ก แบบตัว L

ตัวหุ่นยนต์ประกอบด้วยอะไรบ้าง

1. วงจรอิเล็กทรอนิกส์ (Electronic Circuit)

วงจรอิเล็กทรอนิกส์ที่ใช้ในตัวหุ่นยนต์รุ่นนี้จะมีอยู่ด้วยกัน 3 บอร์ดใหญ่ๆ คือ บอร์ดเซ็นเซอร์ (Sensor Board), บอร์ดควบคุม (Control Board) และบอร์ดไดร์มอเตอร์ (Motor Driver Board)

บอร์ดเซ็นเซอร์ (Sensor Board)

บอร์ดเซ็นเซอร์ที่ใช้กับบอร์ดควบคุมชุดนี้จะมีอยู่ด้วยกัน 3 บอร์ด ได้แก่ บอร์ดสวิตซ์, บอร์ด LDR และบอร์ดออปโต้ทรานซิสเตอร์

หลักการทำงานของบอร์ดสวิตซ์ คือ ในสภาวะปกติที่ยังไม่มีการกดสวิตซ์ ที่ตำแหน่ง S จะมีไฟประมาณ 5 โวลท์ แต่เมื่อไรก็ตามที่มีการกดสวิตซ์ ที่ตำแหน่ง S จะไม่มีไฟเหลืออยู่เลย เนื่องมาจากไฟ 5 โวลท์ จะถูกต่อให้ไหลลงกราวน์ทันที

หลักการทำงานของบอร์ด LDR คือ ในกรณีที่ตัว LDR ไม่ได้รับแสง ความต้านทานภายในของตัว LDR จะมีค่าสูง ทำให้ที่ตำแหน่ง S จะมีไฟประมาณ 5 โวลท์ แต่เมื่อไรก็ตามที่ตัว LDR ได้รับแสง ความต้านทานภายในของตัว LDR จะมีค่าต่ำ ที่ตำแหน่ง S จะไม่มีไฟเหลืออยู่เลย เนื่องมาจากไฟ 5 โวลท์ จะถูกต่อให้ไหลลงกราวน์ทันที

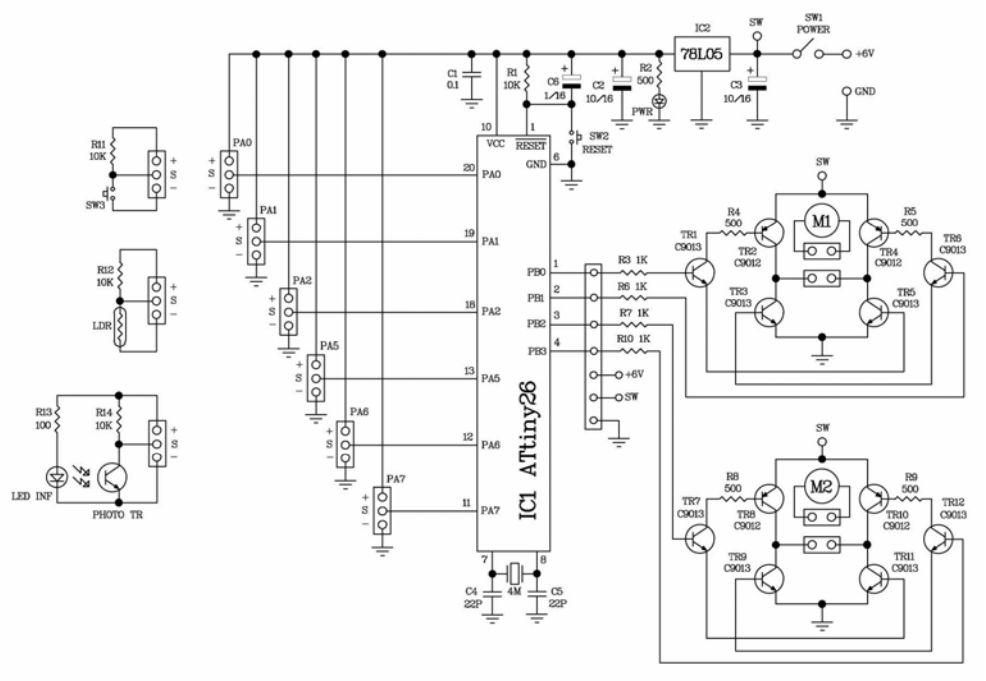
หลักการทำงานของบอร์ดออปโต้ทรานซิสเตอร์ คือ ตัว LED INF จะทำการส่งแสงอินฟาเรดตลอดเวลา สำหรับตัวโฟโต้ทรานซิสเตอร์นั้น เมื่อไม่ได้รับแสงอินฟาเรด ความต้านทานภายในของตัวโฟโต้ทรานซิสเตอร์จะมีค่าสูง ทำให้ที่ตำแหน่ง S จะมีไฟประมาณ 5 โวลท์ แต่เมื่อไรก็ตามที่ตัวโฟโต้ทรานซิสเตอร์ได้รับแสงอินฟาเรด ความต้านทานภายในของตัวโฟโต้ทรานซิสเตอร์ จะมีค่าต่ำ ที่ตำแหน่ง S จะไม่มีไฟเหลืออยู่เลย เนื่องมาจากไฟ 5 โวลท์ จะถูกต่อให้ไหลลงกราวน์ทันที

บอร์ดควบคุม (Control Board)

ลักษณะการทำงานของบอร์ดควบคุมนี้ จะขึ้นอยู่กับ IC1 Attiny26 ซึ่งถือว่าเป็นหัวใจของวงจรเลยทีเดียว ซึ่ง IC1 นี้เป็นไอซีไมโครคอนโทรลเลอร์ที่ใช้ในการเก็บชุดคำสั่งต่างๆ ที่เราเขียนขึ้นมา โดยอาศัยบอร์ดเซ็นเซอร์เพื่อเป็นตัวรับรู้ว่ามีการกระตุ้นให้ทำงาน จากนั้นก็จะไปสั่งให้บอร์ดไดร์มอเตอร์ส่งไฟไปให้กับมอเตอร์เพื่อทำการหมุนต่อไป สำหรับ IC2 เมื่อได้รับไฟจากแบตเตอรี่ประมาณ 6 โวลท์ ตัว IC2 จะทำการลดไฟให้เหลือเพียง 5 โวลท์ เพื่อให้เหมาะสมกับการจ่ายไฟให้กับ IC1

บอร์ดไดร์มอเตอร์ (Motor Driver Board)

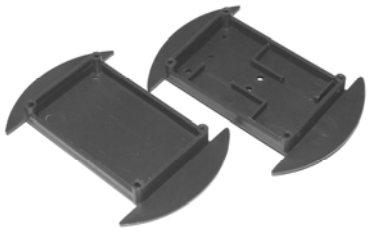
บอร์ดนี้จะมีวงจรที่เหมือนกันอยู่ 2 ชุด ดังนั้นจะขออธิบายเพียงชุดเดียว เมื่อ IC1 ทำการส่งแรงดันออกที่ขา 10 และหยุดส่งแรงดันที่ขา 11 จะมีผลทำให้ TR1 ทำงาน TR2 และ TR5 จึงทำงานไปด้วย มอเตอร์เกียร์ M1 จึงหมุนไปทางซ้าย แต่เมื่อไรก็ตามที่ IC1 ทำการหยุดส่งแรงดันออกที่ขา 10 และส่งแรงดันที่ขา 11 จะมีผลทำให้ TR6 ทำงาน TR3 และ TR4 จึงทำงานไปด้วย มอเตอร์เกียร์ M1 จึงหมุนไปทางขวา



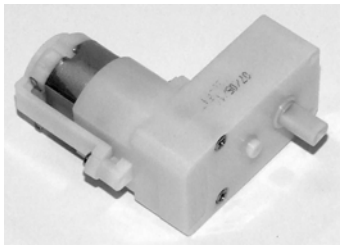
รูป แสดงวงจรหุ่นยนต์ MICRO AVR

2. ส่วนแมคคานิคส์ (Mechanic Part)

ในส่วนของแมคคานิคส์นั้นจะประกอบไปด้วยหลายส่วนด้วยกัน ได้แก่ ตัวบอดี้หุ่นยนต์, มอเตอร์เกียร์, ล้อใหญ่และล้อหลัง



รูปบอดี้ของหุ่นยนต์



รูปมอเตอร์เกียร์ แบบตัว L

รูปล้อใหญ่

รูปล้อหลัง

รายการอุปกรณ์ของหุ่นยนต์

วงจรอิเล็กทรอนิกส์

บอร์ด LDR

ตัวต้านทานขนาด 1/8 วัตต์

R12 - 10KΩ (น้ำตาล ดำ ส้ม ทอง) 1 ตัว

ตัว LDR 1 ตัว

บอร์ดสวิตช์

ตัวต้านทานขนาด 1/8 วัตต์

R11 - 10KΩ (น้ำตาล ดำ ส้ม ทอง) 1 ตัว

SW3 - สวิตช์กดติดปล่อยดับ 1 ตัว

บอร์ดออปโต้ทรานซิสเตอร์

ตัวต้านทานขนาด 1/8 วัตต์

R13 - 100Ω (น้ำตาล ดำ น้ำตาล ทอง) 1 ตัว

R14 - 10KΩ (น้ำตาล ดำ ส้ม ทอง) 1 ตัว

ชุดออปโต้ 1 ตัว

บอร์ดควบคุม

ตัวต้านทานขนาด 1/4 วัตต์

R1 - 10KΩ (น้ำตาล ดำ ส้ม ทอง) 1 ตัว

R2 - 500Ω (เขียว ดำ น้ำตาล ทอง) 1 ตัว

ตัวเก็บประจุชนิดเซรามิก

C1 – 0.1µF (104) 1 ตัว

C4, C5 – 22pF (22) 2 ตัว

ตัวเก็บประจุชนิดอิเล็กทรอไลต์

C2, C3 – 10µF 16V 2 ตัว

C6 – 1µF 16V 1 ตัว

ไอซี

IC1 – ATtiny26 1 ตัว

IC2 – 78L05 1 ตัว

คริสตอล ความถี่ 4MHz 1 ตัว

บอร์ดไดร์มอเตอร์

ตัวต้านทานขนาด 1/4 วัตต์

R4, R5, R8, R9 - 500Ω (เขียว ดำ น้ำตาล ทอง) 4 ตัว

R3, R6, R7, R10 - 1KΩ (น้ำตาล ดำ แดง ทอง) 4 ตัว

ทรานซิสเตอร์

TR1, TR3, TR5, TR6, TR7, TR9, TR11, R12 - C9013 8 ตัว

TR2, TR4, TR8, TR10 - C9012 4 ตัว

ชุดแมกคานิกส์

บอดี้หุ่นยนต์ 1 ชุด

ล้อใหญ่ 2 ล้อ

ชุดล้อหลัง 1 ชุด

มอเตอร์เกียร์ 2 ตัว

ลังถ่าน 4 ก้อน ขนาด AA 1 อัน

ถ่านขนาด AA จำนวน 4 ก้อน (ไม่มีในชุด)

เครื่องมือที่จำเป็นในการประกอบหุ่นยนต์ (ไม่มีในชุด)

- อุปกรณ์บัดกรี (หัวแร้งขนาด 40 วัตต์, ตะกั่วบัดกรี) 1 ชุด
- ไขควงแฉก 1 ด้าม
- คัตเตอร์ 1 ด้าม
- คีมตัด 1 อัน
- คีมจับ 1 อัน
- ที่ดูดตะกั่ว 1 อัน

แนะนำการบัดกรีเบื้องต้น

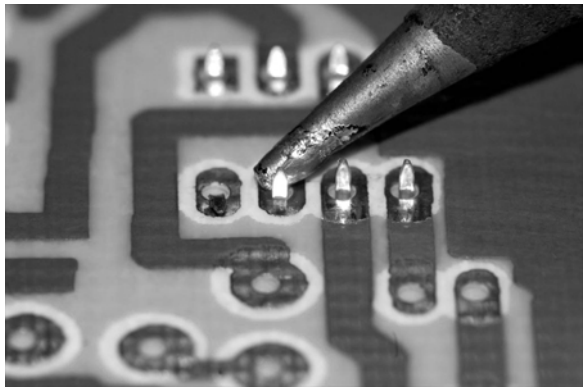
สำหรับเด็กที่อายุต่ำกว่า 8 ปี ไม่ควรทำการบัดกรีอุปกรณ์ต่างๆ เอง เพราะอาจจะเกิดอันตรายได้ ในกรณีที่ผู้ไม่มีพื้นฐานทางด้านอิเล็กทรอนิกส์มาก่อน ควรอยู่ในความดูแลของผู้ที่มีความรู้ทางด้านอิเล็กทรอนิกส์มาทำการสอน ก่อนการประกอบ เพื่อเป็นการป้องกันไม่ให้เกิดอันตรายกับผู้ปฏิบัติงานเอง

ในการบัดกรีนั้น จะต้องจำไว้เสมอว่า อุปกรณ์ที่ใช้ในการบัดกรีนั้นมีความร้อนที่สูงมาก ฉะนั้นไม่ควรนำหัวแร้งไปจี้ยังจุดที่ไม่ใช่จุดบัดกรีหรือนำไปแกล้งคนอื่น เพราะอาจจะเกิดอันตรายจากความร้อนได้ เช่น ไฟไหม้ เป็นต้น สำหรับลำดับขั้นการบัดกรีนั้นจะเริ่มจาก

1.ก่อนการใส่อุปกรณ์ทุกครั้งควรสังเกตที่จุดทองแดงกับขาอุปกรณ์อิเล็กทรอนิกส์ว่าสกปรกหรือไม่ สำหรับจุดทองแดงนั้น วิธีทำความสะอาดก็คือ ให้ใช้ยางลบดินสอลบบริเวณจุดทองแดงที่สกปรก จนกระทั่งเกิดเป็นเงาขึ้น และสำหรับขาอุปกรณ์ ให้ใช้คัตเตอร์ขูดขาของอุปกรณ์จนกระทั่งเกิดเป็นเงาที่ขาอุปกรณ์ สาเหตุที่ต้องทำความสะอาดก็เนื่องมาจากคราบสกปรกอาจจะทำให้การบัดกรียากลำบาก เพราะบัดกรีไม่ติดและยังเป็นสาเหตุที่วงจรไม่ทำงานอีกด้วย

2.ทำการใส่อุปกรณ์ลงในตำแหน่งของอุปกรณ์ตัวนั้นบนแผ่นวงจรพิมพ์ โดยในการใส่จะต้องอุปกรณ์ที่มีความสูงน้อยที่สุดก่อน เช่น ตัวต้านทาน เป็นต้น จากนั้นจึงทำการบัดกรี แล้วค่อยไล่ความสูงขึ้นไปเรื่อยๆ ทำอย่างนี้ไปเรื่อยจนใส่อุปกรณ์ครบทุกตัว แล้วใช้คีมตัด ตัดขาอุปกรณ์ออก

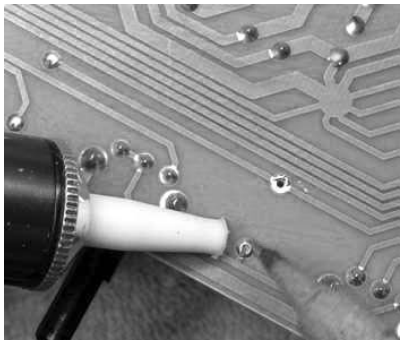
3.ในการบัดกรี จะต้องให้บริเวณปลายของหัวแร้งสัมผัสโดนจุดทองแดงกับขาของอุปกรณ์พร้อมกัน แล้วทิ้งไว้ประมาณ 3 วินาที จากนั้นจึงนำตะกั่วบัดกรีมาจี้ลงบริเวณดังกล่าวเล็กน้อย จะสังเกตเห็นว่าตะกั่วบัดกรีจะไหลมารวมกับจุดทองแดงและขาของอุปกรณ์จนเป็นหนึ่งเดียวกัน จากนั้นจึงดึงหัวแร้งออก

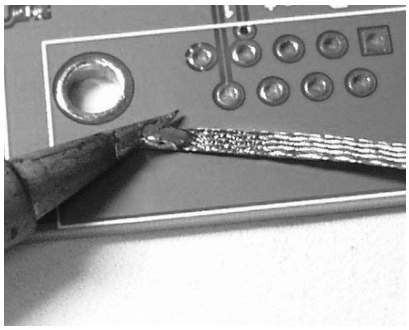


รูปการบัดกรีอุปกรณ์ที่ถูกต้อง

4.จะต้องจดจำไว้เสมอว่าอุปกรณ์อิเล็กทรอนิกส์ทุกชนิดนั้น สามารถทนความร้อนได้ในระดับหนึ่งเท่านั้น ดังนั้นจึงไม่ควรจี้หัวแร้งลงบนขาของอุปกรณ์เป็นเวลาจนเกินไป เพราะอาจจะทำให้อุปกรณ์ตัวนั้นเสียหาย เนื่องมาจากความร้อนได้

5.ในกรณีที่บัดกรีอุปกรณ์ผิดตำแหน่ง เราสามารถทำการถอดอุปกรณ์ตัวนั้นได้ โดยการใช้หัวแร้งจี้ลงบนขาของอุปกรณ์ตัวนั้นให้ร้อนจนกระทั่งตะกั่วเริ่มละลาย จากนั้นให้ใช้ที่ดูดตะกั่ว ดูดเอาตะกั่วที่ละลายกลายเป็นน้ำขึ้นมาหรือใช้ลวดซับตะกั่ว ซับเอาตะกั่วออก ทำอย่างนี้กับทุกขาของอุปกรณ์ตัวที่ใส่ผิดตำแหน่ง เพียงเท่านี้ก็จะสามารถดึงอุปกรณ์ออกจากแผ่นวงจรพิมพ์ได้ โดยไม่ทำให้แผ่นวงจรพิมพ์เสียหาย ข้อควรระวัง ไม่ควรใช้หัวแร้งจี้ลงบนจุดบัดกรีเป็นเวลานาน เพราะอาจจะทำให้จุดบัดกรีและอุปกรณ์เสียหายได้



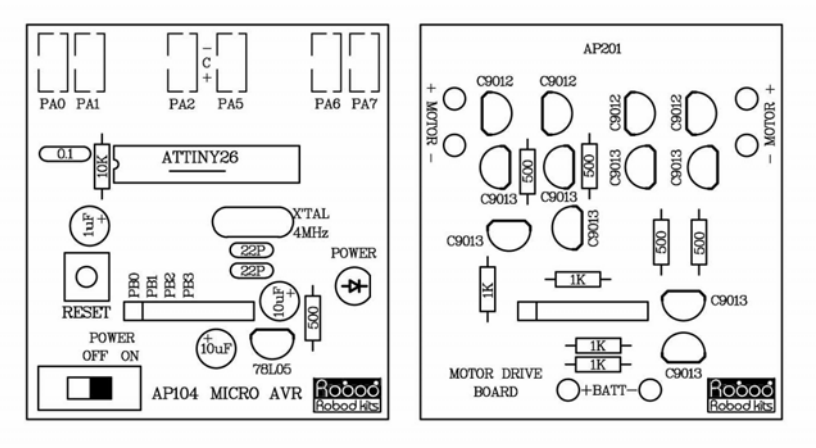


รูปการใช้ที่ดูดตะกั่วและลวดซับตะกั่ว

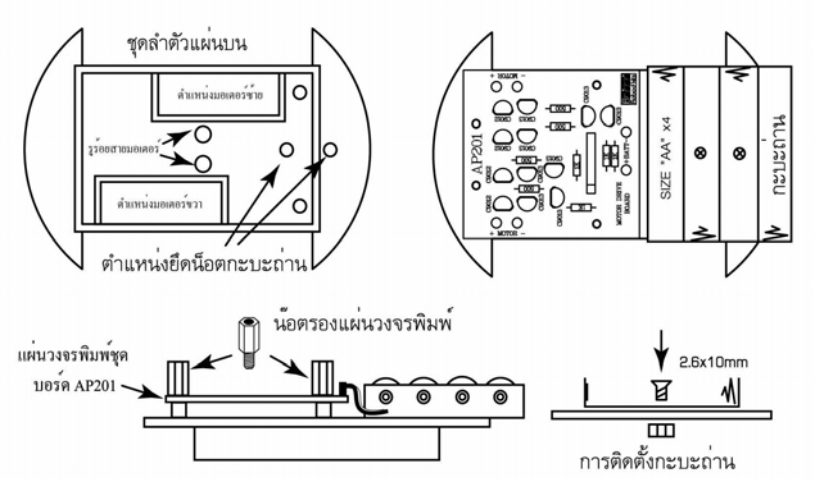
6.เมื่อบัดกรีจนครบทุกตัวแล้วควรทำการตรวจสอบอีกครั้งหนึ่ง ว่าจุดบัดกรีนั้นไปโดนกับตำแหน่งอื่นหรือไม่ ถ้ามีให้ใช้ที่ดูดตะกั่วดูดเอาตะกั่วที่เกินออก เพราะถ้าจุดบัดกรีสองจุดต่อถึงกันอาจจะทำให้เกิดการลัดวงจรและวงจรเสียหายได้

การประกอบ

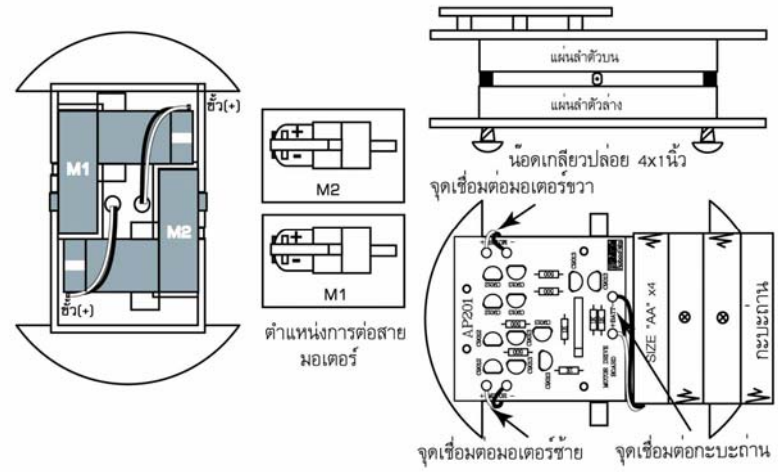
ในการประกอบหุ่นยนต์นี้จะแบ่งออกเป็น 2 ส่วนด้วยกัน คือ ส่วนวงจรอิเล็กทรอนิกส์และส่วนแมคคานิคส์



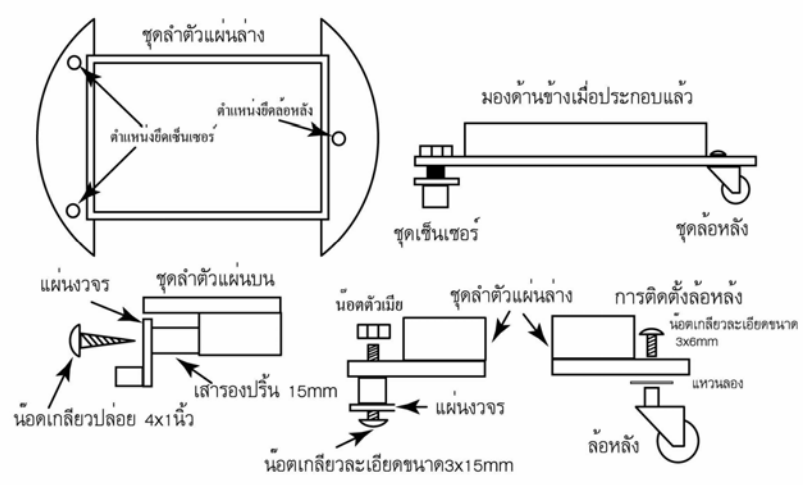
ตำแหน่งการใส่อุปกรณ์อิเล็กทรอนิกส์ลงบนแผ่นวงจรพิมพ์



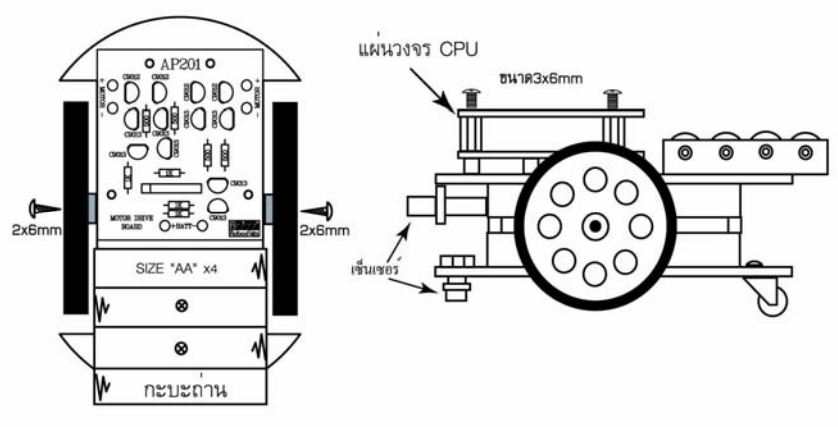
รูปการประกอบกะบะถ่านและแผ่นวงจรไดร์ฟมอเตอร์



รูปการประกอบมอเตอร์ ลำตัวและการเชื่อมต่อสาย



รูปการประกอบชุดเซ็นเซอร์และล้อหลัง



รูปการประกอบชุดล้อข้างและแผ่นวงจรไอซีไมโครคอนโทรลเลอร์

Software ที่ใช้กับตัวหุ่นยนต์

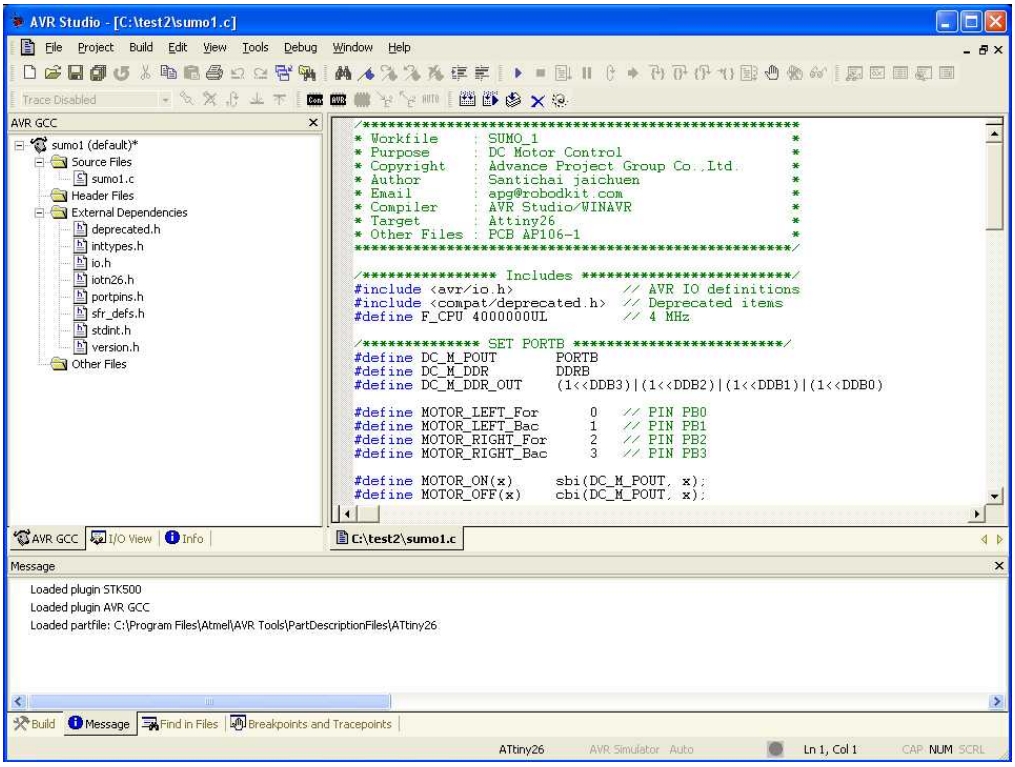
ในส่วนของ Software จะมีอยู่ด้วยกัน 2 โปรแกรม คือ โปรแกรมกำหนดการทำงานของหุ่นยนต์ และ โปรแกรมบันทึกข้อมูลลงในไมโครคอนโทรลเลอร์

1. โปรแกรมกำหนดการทำงานของหุ่นยนต์

โปรแกรมที่ใช้ในการเขียน เพื่อกำหนดการทำงานของหุ่นยนต์นั้น เราจะใช้โปรแกรมที่เขียนด้วยภาษาซี เนื่องมาจากในการเขียนโปรแกรมนั้นสามารถเข้าใจได้ง่ายและไม่มีความยุ่งยากสำหรับผู้เรียน ในที่นี้เราจะใช้โปรแกรมที่มีชื่อว่า AVR Studio ของบริษัท Atmel Corporation ซึ่งทางบริษัทได้เปิดให้ทำการดาวน์โหลดใช้ได้ที่เว็บไซต์ www.atmel.com

สำหรับคุณสมบัติขั้นต่ำของเครื่องคอมพิวเตอร์ที่แนะนำให้ใช้กับโปรแกรม AVR Studio มีดังนี้

- ซีพียู ขนาด 200 MHz ขึ้นไป Intel Pentium หรือ AMD K-6
- หน่วยความจำแรม 64MB
- เนื้อที่ว่างบนฮาร์ดดิสก์ อย่างน้อย 50MB
- CD-ROM
- หน้าจอขนาด 1200x768 (ต่ำสุด 800x600)



รูปหน้าตาของโปรแกรม AVR Studio

2. โปรแกรมบันทึกข้อมูลลงในไมโครคอนโทรลเลอร์

ขั้นตอนการติดตั้งโปรแกรม AVR Studio

ในการลงโปรแกรม AVR Studio นั้น จะเหมือนกับการลงโปรแกรมอื่นๆ ทั่วไป ซึ่งขั้นตอนที่ง่ายมาก ดังนี้

1.ทำการดับเบิ้ลคลิกไฟล์ aStudio4b460.exe จะเป็นการเริ่มเข้าสู่ขั้นตอนการติดตั้งโปรแกรม

2.เมื่อเริ่มเข้าสู่กระบวนการติดตั้งโปรแกรม ให้ทำการคลิกปุ่ม Next โปรแกรมจะแสดงข้อความประกาศสิทธิ ให้ทำการติ๊กที่ I accept the terms of the license agreement แล้วคลิก NEXT โปรแกรมจะแสดงแสดงหน้าต่างใหม่ขึ้นมา ซึ่งในหน้าต่างนี้จะถามถึงตำแหน่งที่เราต้องการติดตั้ง โดยปกติตัวติดตั้งจะกำหนดมาให้อยู่แล้ว เมื่อเลือกตำแหน่งที่ต้องการติดตั้งโปรแกรมได้แล้วก็ให้ทำการกดปุ่ม Next

3.โปรแกรมจะถามว่า เราต้อง Install/upgrade USB driver หรือไม่ ให้ทำการติ๊กที่หน้าหัวข้อ จากนั้นให้คลิก NEXT แล้วคลิก Install

4.โปรแกรมก็จะทำการติดตั้งจนกระทั่งเสร็จ ก็ให้ทำการกดปุ่ม Finish ก็เป็นอันเสร็จ

เริ่มต้นใช้งานโปรแกรม AVR Studio

การเปิดโปรแกรม

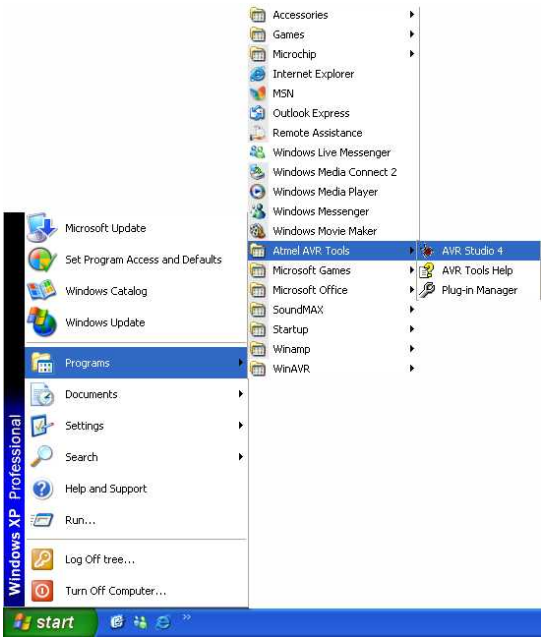
การเปิดโปรแกรม AVR Studio ทำได้โดย

1.คลิกที่ปุ่ม START ของ Windows

2.เลื่อนไปที่ All Programs

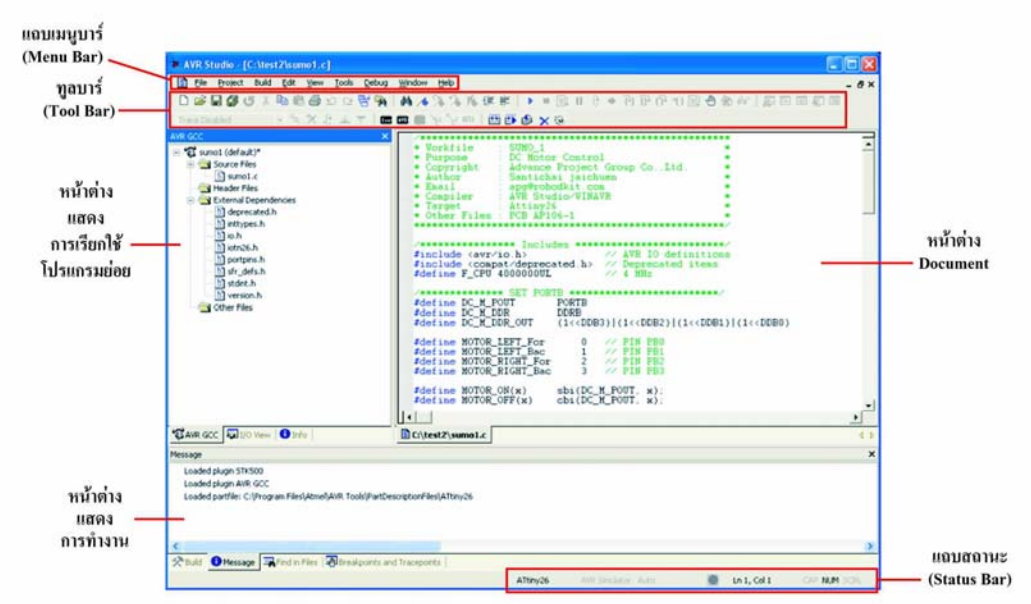
3.เลื่อนไฮไลท์เมาส์ไปที่กลุ่มไอคอน Atmel AVR Tools

4.คลิกที่ไอคอน AVR Studio 4



รูปแสดงลำดับขั้นการเปิดโปรแกรม AVR Studio

ส่วนประกอบของหน้าจอโปรแกรม



1.หน้าต่าง Document

หน้าต่าง Document คือส่วนที่ใช้สำหรับใส่คำสั่งที่เราต้องการเขียน วิธีใช้งานหน้าต่างนี้จะคล้ายกับการใช้งานโปรแกรมเวิร์ดโปรเซสเซอร์ทั่วไป เช่น การพิมพ์ข้อความ, การคัดลอก, การลบข้อความ เป็นต้น

2.แถบเมนูบาร์ (Menu Bar)

คือ ส่วนที่เก็บคำสั่งสำหรับการทำงานต่างๆ เอาไว้ ซึ่งบางคำสั่งสามารถเรียกใช้จากทูลบาร์ได้ แต่บางคำสั่งจะมีลักษณะเฉพาะในแถบเมนูนั้น เราสามารถเปิดเมนูต่างๆ ขึ้นมาใช้งานโดยการคลิกที่ชื่อเมนูและเลื่อนเมาส์ไปคลิกยังคำสั่งที่ต้องการ หากคำสั่ง

นั้นมีเมนูย่อย (สังเกตจากลูกศรที่อยู่ด้านขวาของเมนูนั้น) ให้เลื่อนเมาส์ไปที่คำสั่งนั้น จะปรากฏกรอบเมนูย่อยแสดงขึ้นมา จากนั้นก็คลิกเลือกคำสั่งที่ต้องการ

3.ทูลบาร์ (Toolbar)

คือแถบเครื่องมือที่เก็บปุ่มคำสั่งต่างๆ ที่จำเป็นต้องใช้งานบ่อยๆ ไว้ เมื่อนำเมาส์ไปชี้ตามปุ่มต่างๆ เหล่านั้น จะมีข้อความขึ้นมาเพื่ออธิบายถึงหน้าที่ของปุ่มต่างเหล่านั้น เช่น ปุ่ม Create New File, ปุ่ม Open File, ปุ่ม Save File เป็นต้น

4.แถบสถานะ (Status Bar)

คือแถบแสดงสถานะที่อยู่ด้านล่างของหน้าต่าง Document โดยทางด้านซ้ายจะเป็นตัวบอกว่าใช้ไอซีเบอร์อะไร ถัดมาจะเป็นตัวบอกตำแหน่งของเคอร์เซอร์ว่าอยู่บรรทัดและตัวอักษรที่เท่าไร เช่น Ln 1, Col 1 หมายความว่า ตอนนี้เคอร์เซอร์อยู่บรรทัดที่ 1 ตัวอักษรที่ 1 เป็นต้น

5.หน้าต่างแสดงการเรียกใช้โปรแกรมย่อย จะเป็นหน้าต่างแสดงโปรแกรมย่อยต่างๆ ที่เราเรียกใช้ขึ้นมา เพื่อสะดวกในการไล่โปรแกรม

6.หน้าต่างแสดงการทำงาน จะเป็นหน้าต่างแสดงลำดับการทำงานของโปรแกรม เช่น ตำแหน่งที่เรียกใช้หรือเก็บโปรแกรมที่เรากำลังเขียนอยู่ เป็นต้น

หลักในการเขียนโปรแกรม

ขั้นตอนที่ 1 เราจะต้องทำการกำหนดการทำงานของวงจรเสียก่อน เพื่อจะได้ทราบถึงการกำหนดการทำงานของโปรแกรมได้ โดยปกติแล้ว ตัวไมโครคอนโทรลเลอร์จะทำงานในลักษณะของวงจรดิจิตอล คือจะเป็น 1 (มีไฟ) และ 0 (ไม่มีไฟ) เท่านั้น ไม่ว่าจะเป็นทางด้านอินพุทหรือเอาต์พุทก็ตาม จะเป็นในลักษณะนี้ทั้งหมด โดยสัญญาณอินพุทและเอาต์พุตนี้จะต้องมีแรงดันไม่เกิน 5 โวลท์ เพราะถ้าแรงดันเกินจากนี้ ตัวไอซีไมโครคอนโทรลเลอร์อาจจะเสียหายได้

ขั้นตอนที่ 2 ทำการเขียนโฟลว์ชาร์ตขึ้น เพื่อเป็นแนวทางในการเขียนโปรแกรมให้ง่ายขึ้นและทำให้เราสามารถทราบถึงลำดับขั้นตอนการทำงานของวงจรได้อีกด้วย ซึ่งเป็นการลดขั้นตอนการทำงานเมื่อมีการแก้ไขโปรแกรมลงได้

สัญลักษณ์ของโฟลว์ชาร์ตและลักษณะการทำงาน

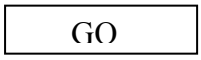
1.สัญลักษณ์เริ่มต้นและสิ้นสุดโปรแกรม

สัญลักษณ์นี้จะมีรูปร่างลักษณะที่เหมือนกัน คือ มีลักษณะเป็นสี่เหลี่ยมผืนผ้าและมีมุมที่โค้ง ในกรณีที่ใช้เป็นตัวเริ่มต้นการทำงานของโปรแกรม นั่นหมายความว่า เป็นการเริ่มต้นจ่ายไฟเข้าวงจรให้ทำงาน สำหรับกรณีที่ใช้เป็นตัวสิ้นสุดการทำงานของโปรแกรม นั่นก็คือจบการทำงานนั่นเอง

รูปแสดงลักษณะสัญลักษณ์เริ่มต้นและสิ้นสุดโปรแกรม

2.สัญลักษณ์ขั้นตอนการทำงาน

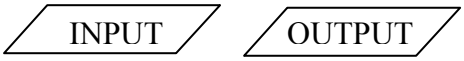
สัญลักษณ์นี้จะมีรูปร่างเป็นลักษณะสี่เหลี่ยมผืนผ้า โดยสัญลักษณ์ตัวนี้จะเป็นตัวกำหนดลักษณะการทำงานของโปรแกรมว่าจะทำงานในลักษณะใด เช่น เดินไปข้างหน้า, เดินถอยหลัง เป็นต้น



รูปแสดงลักษณะสัญลักษณ์ขั้นตอนการทำงาน

3.สัญลักษณ์รับข้อมูลและส่งข้อมูล

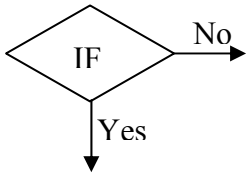
สัญลักษณ์นี้จะมีลักษณะเป็นสี่เหลี่ยมด้านขนาน โดยสัญลักษณ์ตัวนี้จะเป็นตัวบอกถึงการรับรับข้อมูลเข้ามาหรือส่งข้อมูลออกไป เช่น การรับข้อมูลจากตัวเซ็นเซอร์, การส่งไฟออกไปขับมอเตอร์ เป็นต้น



รูปแสดงลักษณะสัญลักษณ์รับข้อมูลและส่งข้อมูล

4.สัญลักษณ์การตัดสินใจ

สัญลักษณ์นี้จะมีลักษณะเป็นสี่เหลี่ยมข้าวหลามตัด หน้าที่ของสัญลักษณ์นี้ก็คือจะเป็นตัวกำหนดการตัดสินใจของโปรแกรมว่าจะให้ไปทางใด โดยมีการกำหนดใน 2 ลักษณะ คือ ใช่ (Yes) และ ไม่ใช่ (No)



รูปแสดงลักษณะสัญลักษณ์การตัดสินใจ

5.สัญลักษณ์ทางเดินของข้อมูล

สัญลักษณ์นี้จะมีลักษณะเป็นลูกศร เพื่อกำหนดทิศทางในการทำงานของโปรแกรม ว่าจะให้ไปในทิศทางใด

ขั้นตอนที่ 3 ทำการเขียนภาษาซี ตามลำดับขั้นตอนที่เราได้เขียนโฟลว์ชาร์ตเอาไว้ ในส่วนนี้จำเป็นต้องมีความรู้พื้นฐานในการเขียนโปรแกรมด้วยภาษาซี จึงจะทำการเขียนโปรแกรมได้

พื้นฐานการเขียนโปรแกรมภาษาซีและ AVR Studio

1.โครงสร้างในการเขียนภาษาซี

โครงสร้างของภาษาซีนั้น จะประกอบด้วยการทำงานอยู่ 2 ส่วน คือ ส่วนของโปรแกรมหลัก (Main Code Programming) และส่วนของโปรแกรมย่อยหรือฟังก์ชั่น (Function Code Programming)

การทำงานของโปรแกรมโดยหลักๆ แล้วจะอยู่ในส่วนของโปรแกรมหลัก ซึ่งอาจจะมีการเรียกใช้งานในส่วนของโปรแกรมย่อยบ้าง โดยโปรแกรมย่อยนี้จะเป็นโปรแกรมที่ถูกเรียกใช้งานบ่อยๆ เช่น โปรแกรมเกี่ยวกับเวลา, เงื่อนไขที่กระทำเหมือนๆ กัน เป็นต้น ถ้าเราไม่ทำการแยกโปรแกรมต่างๆ เหล่านี้ออกจากโปรแกรมหลักแล้ว อาจจะทำให้โปรแกรมรวมทั้งหมดมีขนาดใหญ่เกินกว่าที่จะสามารถนำไปใช้งานได้ ดังนั้นโปรแกรมย่อยจึงมีประโยชน์อย่างมากในการลดขนาดของโปรแกรมรวมทั้งหมด ให้ลดลงนั่นเอง

ลักษณะในการเขียนโปรแกรมหลักนี้ เราจะขึ้นต้นด้วย main ตามด้วยปีกกาใหญ่ { } ส่วนโปรแกรมที่เขียนจะเขียนอยู่ภายในปีกกาใหญ่นั้นจะเป็นตัวกำหนดการทำงานของโปรแกรม เช่น

ตัวอย่างโปรแกรม

```
#include <stdio.h>            // Preprocessor directives (header file)
void main(void) {
        printf ( “\nHello World\n” );   //แสดงผลลัพธ์ด้วยฟังก์ชั่นมาตรฐาน printf
}
```

คำอธิบายโปรแกรม

โดยโปรแกรมที่ยกตัวอย่างนี้ เมื่อทำการรันโปรแกรม ตัวโปรแกรมจะทำการแสดงข้อความ “Hello World” บนหน้าจอคอมพิวเตอร์ แต่สำหรับโปรแกรม AVR Studio แล้วจะเป็นการแสดงผลผ่านตัว LED โดยการติดและดับ ส่วนโปรแกรมย่อยนั้นจะมีลักษณะการเขียนที่เหมือนกับโปรแกรมหลัก แต่ชื่อของโปรแกรมเราสามารถเปลี่ยนได้ตามความต้องการ เพื่อสะดวกในการใช้งาน เช่น Delay_time, Go_left เป็นต้น ตัวอย่างการเขียนจะเป็นดังนี้

ตัวอย่างโปรแกรม

```
#include <stdio.h>          // Preprocessor directives (header file)

void time_delay(void) {
    Delay_ms(1000);                 //หน่วงเวลาเป็นเวลา 1 วินาที
}

void main(void) {
    while(TRUE) {
            printf ( “\nHello \n” );        //แสดงผลลัพธ์ด้วยฟังก์ชั่นมาตรฐาน printf
            time_delay();                   //เรียกใช้งานโปรแกรมย่อย time_delay
            printf (“\nWorld\n” );          //แสดงผลลัพธ์ด้วยฟังก์ชั่นมาตรฐาน printf
            time_delay();                   //เรียกใช้งานโปรแกรมย่อย time_delay
    }
}
```

ตัวอย่างโปรแกรม

จากโปรแกรมดังกล่าว เมื่อทำการรันโปรแกรม ที่หน้าจอคอมพิวเตอร์จะทำการแสดงข้อความ “Hello” ก่อน แล้วทิ้งช่วงประมาณ 1 วินาที แล้วจึงทำการแสดงข้อความ “World” แล้วทิ้งช่วงประมาณ 1 วินาที แล้วก็จะกลับไปแสดงข้อความ “Hello” ใหม่ จะเป็นอย่างนี้ไปเรื่อย เนื่องมาจากคำสั่ง while

จะสังเกตเห็นว่า ในการเขียนโปรแกรมนั้นจะเขียนโปรแกรมหลักไว้ด้านล่างและเขียนโปรแกรมย่อยไว้ข้างบน สาเหตุที่เป็นอย่างนี้ เนื่องมาจากว่าภายในโปรแกรมจะทำการไล่ลำดับการทำงานจากบนลงล่าง แต่เมื่อมีการเรียกใช้งานโปรแกรมย่อยจะเรียกจากข้างบนนั่นเอง

2.การกำหนดค่าของตัวแปรต่างๆ (Declaration) ในโปรแกรมภาษาซีที่ใช้กับ AVR Studio

- ชนิดของข้อมูล(data type) จะสามารถกำหนดได้หลายรูปแบบไม่ว่าจะเป็นตัวเลขหรือตัวอักษรก็ได้ เช่น int, char, float, short, void เป็นต้น ซึ่งแต่ละตัวจะมีช่วงของค่าของข้อมูลที่ต่างกันออกไปดังนี้

int	เป็นตัวแปรขนาด 8 บิต (ตัวเลขจำนวนเต็ม) มีช่วงค่าของข้อมูลอยู่ที่ 0 ถึง 255
int1	เป็นตัวแปรขนาด 1 บิต มีช่วงค่าของข้อมูลอยู่ที่ 0 ถึง 1
int8	เป็นตัวแปรขนาด 8 บิต มีช่วงค่าของข้อมูลอยู่ที่ 0 ถึง 255
int16	เป็นตัวแปรขนาด 16 บิต มีช่วงค่าของข้อมูลอยู่ที่ 0 ถึง 65,535
float	เป็นตัวแปรขนาด 32 บิต (ตัวเลขทศนิยม) มีช่วงค่าของข้อมูลอยู่ที่ $3.4X10^{-38}$ ถึง $3.4X10^{38}$
char	เป็นตัวแปรขนาด 8 บิต (ตัวอักขระ) มีค่าของข้อมูลเป็นตัวอักขระรหัสแอสกี้
void	เป็นการไม่กำหนดค่าใดๆ

- การคำนวณทางคณิตศาสตร์ สัญลักษณ์ในการคำนวณทางคณิตศาสตร์จะเหมือนกับที่เราเคยเรียนกันมา จะต่างเพียงบางตัวเท่านั้น เช่น ตัวหาร ในภาษาซีจะใช้เป็นเครื่องหมาย / ในการคำนวณแต่ละครั้งเราจะต้องกำหนดค่าของตัวแปรแต่ละตัวก่อนแล้วจึงนำไปคำนวณ

ตัวอย่างเช่น

```
int x, y, z;          //กำหนดให้ x, y และ z เป็นตัวแปรขนาด 8 บิต
x = 10; y = 5;        //กำหนดให้ x มีค่าเท่ากับ 10 และ y มีค่าเท่ากับ 5
z = x + y             //ค่าของ z มีค่าเท่ากับ x+y นั่นคือ 15
```

3.การทำงานแบบมีเงื่อนไข

การทำงานในลักษณะนี้จะเป็นการตรวจสอบเงื่อนไขตามที่เราตั้งเอาไว้ ถ้าเป็นไปตามเงื่อนไขที่เราตั้งเอาไว้ ก็จะไปทำงานในเงื่อนไขนั้น ถ้าไม่ใช่ก็จะกระโดดข้ามไป คำสั่งในลักษณะนี้จะมีอยู่ 2 คำสั่ง คือ

- คำสั่ง if…else… เป็นคำสั่งที่ทำการตรวจสอบเงื่อนไขที่เรากำหนด ถ้าเป็นจริงก็จะทำงานในปีกกาของ if แต่ถ้าไม่ใช่ก็จะทำงานในปีกกาของ else เช่น

ตัวอย่างโปรแกรมที่ 1

```
if (a=1) {              //ถ้า a มีค่าเท่ากับ 1 จะทำให้ b มีค่าเท่า 2 ถ้าไม่ใช่ก็จะกระโดดข้ามไป
        b = 2;
}
```

ตัวอย่างโปรแกรมที่ 2

```
if (a=1) {              //ถ้า a มีค่าเท่ากับ 1 จะทำให้ b มีค่าเท่า 2
        b = 2;
} else {                //ถ้า a ไม่เท่ากับ 1 จะทำให้ b มีค่าเท่ากับ 0
        b = 0;
}
```

ตัวอย่างโปรแกรมที่ 3

```
if (a=1) {              //ถ้า a มีค่าเท่ากับ 1 จะทำให้ b มีค่าเท่า 2
        b = 2;
} else if (a=2) {               //ถ้า a มีค่าเท่ากับ 2 จะทำให้ b มีค่าเท่ากับ 0
        b = 0;
}
```

- คำสั่ง switch เป็นคำสั่งที่ใช้ตรวจสอบหลายๆ เงื่อนไขในครั้งเดียว ซึ่งถ้าเราใช้คำสั่ง if…else… อาจจะทำให้คำสั่งนั้นยาวเกินความจำเป็น ตัวอย่างเช่น

```
switch (a) {                    //ตรวจสอบ a
        case 0 : b = 1;         //ถ้า a มีค่าเท่ากับ 0 จะทำให้ b มีค่าเท่ากับ 1 แล้วจึงหยุดการทำงานด้วย break
        break;
        case 1 : b = 2;         //ถ้า a มีค่าเท่ากับ 1 จะทำให้ b มีค่าเท่ากับ 2 แล้วจึงหยุดการทำงานด้วย break
        break;
```

```
        case 2 : b = 3;          //ถ้า a มีค่าเท่ากับ 2 จะทำให้ b มีค่าเท่ากับ 3 แล้วจึงหยุดการทำงานด้วย break
        break;
}
```

4.การทำงานแบบวนลูป

เป็นคำสั่งที่สั่งให้โปรแกรมทำงานอยู่ภายในลูป ซึ่งจะทำงานไปเรื่อยๆ จนกระทั่งเงื่อนไขจะเป็นเท็จ ก็จะหลุดออกจากลูปไป

- คำสั่ง while เป็นคำสั่งที่จะทำการตรวจสอบเงื่อนไขว่าเป็นจริงหรือไม่ ถ้าเป็นจริงก็จะทำงานในคำสั่ง while ไปเรื่อยๆ จนกว่าจะเป็นเท็จจึงจะหลุดออกจากลูป ตัวอย่างโปรแกรม

```
while (a<10) {          //ถ้า a มีค่าน้อยกว่า 10 ก็ให้ทำการบวกค่าไปทีละ 1 จนมีค่าเท่ากับ 10 จึงหลุดไป
        a++;            //บวก a ทีละหนึ่งเรื่อยๆ
        a = a+1;        //นำค่า a บวกหนึ่ง
}
```

- คำสั่ง do…while… เป็นคำสั่งที่แตกต่างจากคำสั่ง while ตรงที่จะกระทำคำสั่งภายในลูปก่อน 1 ครั้ง แล้วจึงจะตรวจสอบเงื่อนไข ถ้าเป็นจริงก็จะทำงานในลูป แต่ถ้าเป็นเท็จจึงจะหลุดออกจากลูป ตัวอย่างโปรแกรม

```
do {
        a++;            //บวก a ทีละหนึ่งเรื่อยๆ
        a = a+1;        //นำค่า a บวกหนึ่ง
} while (a<10);         //ถ้า a มีค่าน้อยกว่า 10 ก็ให้ทำการบวกค่าไปทีละ 1 จนมีค่าเท่ากับ 10 จึงหลุดไป
```

- คำสั่ง for เป็นคำสั่งที่จะทำงานตามจำนวนที่กำหนดเอาไว้ เมื่อครบแล้วก็จะหลุดออกจากลูปไป ตัวอย่างโปรแกรม

```
for (i=0; i<10; i++) {  //กำหนดให้ i เท่ากับศูนย์ แล้วบวกไปเรื่อยๆ จนเท่ากับ 10 จึงหลุดไป
        a++;            //บวก a ทีละหนึ่งเรื่อยๆ
        a = a+1;        //นำค่า a บวกหนึ่ง
}
```

เริ่มต้นเรียนรู้การเขียนโปรแกรมภาษาซีบนโปรแกรม AVR Studio

ก่อนที่เราจะเริ่มทำการเขียนโปรแกรมนั้น เราจะต้องวางแผนเสียก่อนว่าโปรแกรมจะทำงานในลักษณะใดบ้าง ดังนี้

ขั้นแรก เราจะต้องทราบถึงลักษณะการทำงานของโปรแกรมก่อนว่าจะให้เริ่มแบบไหนและสิ้นสุดยังไง

ขั้นที่สอง ทำการเขียนโฟลว์ชาร์ตตามลักษณะการทำงานของโปรแกรมของโปรแกรมที่เรากำหนดไว้ในตอนแรก

ขั้นที่สาม เมื่อเราได้ข้อมูลตั้งแต่ลักษณะการทำงานและโฟลว์ชาร์ตมาแล้ว เราก็เริ่มทำการเขียนภาษาซี

ขั้นที่สี่ ทำการคอมไพล์ภาษาซีที่เราเขียนขึ้นมาให้เป็นไฟล์ HEX เพื่อนำไปโหลดลงบนตัวไอซีไมโครคอนโทรลเลอร์ต่อไป ในกรณีที่มีข้อผิดพลาดเกี่ยวกับโปรแกรมที่เราเขียนขึ้น โปรแกรมจะไม่สามารถคอมไพล์ได้และทำการแจ้งเตือนว่าจุดใดผิดพลาด

ตัวอย่างโปรแกรม

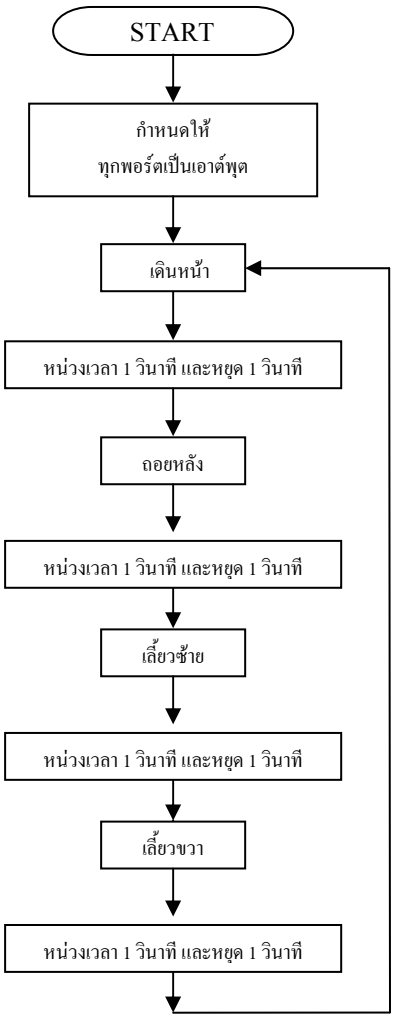
ชื่อโปรแกรม ทดสอบการเดินของหุ่นยนต์

ลักษณะการทำงาน

การทำงานของโปรแกรม เมื่อทำการเปิดสวิตซ์ หุ่นยนต์จะทำการเดินหน้าไป 1 วินาทีและหยุด 1 วินาที, ถอยหลัง 1 วินาทีและหยุด 1 วินาที, เลี้ยวซ้าย 1 วินาทีและหยุด 1 วินาที, เลี้ยวขวา 1 วินาทีและหยุด 1 วินาที และจะเป็นแบบนี้ไปเรื่อยๆ จนกว่าจะปิดสวิตซ์ โดยกำหนดให้ทุกขา เป็นเอาต์พุตเสมอ

การทำงานแบบโฟลว์ชาร์ต

เมื่อทำการกำหนดการทำงานได้เรียบร้อยแล้ว เราก็จะสามารถเขียนโฟลว์ชาร์ตได้ ดังต่อไปนี้



รูปโฟลว์ชาร์ตการทำงานของหุ่นยนต์

เปลี่ยนโฟลว์ชาร์ตเป็นภาษาซี

เมื่อเราได้โฟลว์ชาร์ตมาแล้ว เราก็เริ่มทำการเขียนภาษาซีในโปรแกรม AVR Studio ดังต่อไปนี้

1.เปิดโปรแกรม AVR Studio 4 โปรแกรมจะขึ้นหน้าต่างให้เรียกใช้งานไฟล์ โดยให้เราเลือกที่ New Project ตรงหน้าต่าง Project Type ให้เลือก AVR GCC แล้วให้ทำการตั้งชื่อที่ช่อง Project name ในที่นี้ให้ตั้ง sumo1 จากนั้นให้ทำการเลือกสถานที่จัดเก็บในช่อง Location แล้วกด NEXT ทำการเลือก Debug platform โดยให้เลือก AVR simulator แล้วเลือก Device เป็น Attiny26 แล้วกดปุ่ม Finish

2.จะขึ้นหน้าต่าง Document ขึ้นมา ซึ่งอยู่ภายใต้ชื่อ Sumo1แล้วทำการเขียนโปรแกรมลงในหน้าต่าง Document ดังต่อไปนี้

```
/*************** Includes ****************/
#include <avr/io.h>                    // AVR IO definitions
#include <compat/deprecated.h>         // Deprecated items
#define F_CPU 4000000UL                // กำหนดให้ไอซีไมโครคอนโทรลเลอร์ใช้ความถี่ 4 MHz

/************* SET PORTB **************/
#define DC_M_POUT          PORTB
#define DC_M_DDR           DDRB
#define DC_M_DDR_OUT   (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)

#define MOTOR_LEFT_For          0       // กำหนดให้ขา PB0 เป็นมอเตอร์ซ้าย เดินหน้า
#define MOTOR_LEFT_Bac          1       // กำหนดให้ขา PB1 เป็นมอเตอร์ซ้าย ถอยหลัง
#define MOTOR_RIGHT_For         2       // กำหนดให้ขา PB2 เป็นมอเตอร์ขวา เดินหน้า
#define MOTOR_RIGHT_Bac         3       // กำหนดให้ขา PB3 เป็นมอเตอร์ซ้าย ถอยหลัง

#define MOTOR_ON(x)     sbi(DC_M_POUT, x);
#define MOTOR_OFF(x)    cbi(DC_M_POUT, x);
/************ delay **************/
void delay(int i)                  //ส่วนหน่วงเวลา
{
        int j,k;
                for (;i > 0; i--)
                for(j=100; j>0; j--)
                for(k=1000; k>0; k--);
}

/*************** Motor Forward ********************/
void Motor_Forward(int delay_m_f)          //ส่วนควบคุมให้หุ่นยนต์เดินหน้า
{
        MOTOR_ON(MOTOR_LEFT_For);          // กำหนดให้มอเตอร์ซ้าย เดินหน้า
        MOTOR_ON(MOTOR_RIGHT_For);         // กำหนดให้มอเตอร์ขวา เดินหน้า
        delay(delay_m_f);
}
```

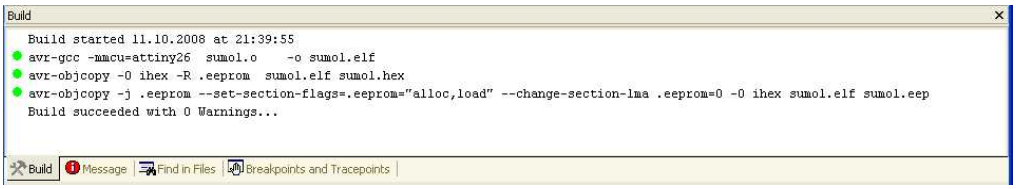
```
/************** Motor Backword ********************/
void Motor_Backword(int delay_m_f)          //ส่วนควบคุมให้หุ่นยนต์ถอยหลัง
{
        MOTOR_ON(MOTOR_LEFT_Bac);           // กำหนดให้มอเตอร์ซ้าย ถอยหลัง
        MOTOR_ON(MOTOR_RIGHT_Bac);          // กำหนดให้มอเตอร์ขวา ถอยหลัง
        delay(delay_m_f);
}

/************** Motor Stop ***********************/
void Motor_Stop(int delay_m_s)       //ส่วนควบคุมให้มอเตอร์หยุดการทำงานทั้งหมด
{
        MOTOR_OFF(MOTOR_LEFT_For);          // Motor right stop
        MOTOR_OFF(MOTOR_RIGHT_For);         // Motor left stop
        MOTOR_OFF(MOTOR_LEFT_Bac);          // Motor right stop
        MOTOR_OFF(MOTOR_RIGHT_Bac);         // Motor left stop
   delay(delay_m_s);
}

/************** Motor Turn Left ***************/
void Motor_Turn_Left(int delay_m_f)          //ส่วนควบคุมให้หุ่นยนต์เลี้ยวซ้าย
{
        MOTOR_ON(MOTOR_LEFT_Bac);           // กำหนดให้มอเตอร์ซ้าย ถอยหลัง
        MOTOR_ON(MOTOR_RIGHT_For);          // กำหนดให้มอเตอร์ขวา เดินหน้า
        delay(delay_m_f);
}

/************** Motor Turn Right ******************/
void Motor_Turn_Right(int delay_m_f)        //ส่วนควบคุมให้หุ่นยนต์เลี้ยวขวา
{
        MOTOR_ON(MOTOR_LEFT_For);           // กำหนดให้มอเตอร์ซ้าย เดินหน้า
        MOTOR_ON(MOTOR_RIGHT_Bac);          // กำหนดให้มอเตอร์ขวา ถอยหลัง
        delay(delay_m_f);
}

/************** Main Functions ***********************/
int main(void)
{
        DC_M_DDR   = DC_M_DDR_OUT;          // Set Port Output
   while (1)
        {
                Motor_Forward(1); Motor_Stop(1);
                Motor_Backword(1); Motor_Stop(1);
                Motor_Turn_Left(1); Motor_Stop(1);
```

```
                Motor_Turn_Right(1); Motor_Stop(1);
        }
        return 0;
}
/************************จบโปรแกรม***************************/
```

คอมไพล์จากภาษาซีไปสู่ HEX FILE

หลังจากที่เราเขียนโปรแกรมเสร็จเรียบร้อยแล้ว เราจะต้องทำการเปลี่ยนจากภาษาซีให้เป็น HEX FILE ก่อน เพื่อที่เราจะได้โหลดโปรแกรมลงบนตัวไอซีไมโครคอนโทรลเลอร์ได้ โดยขั้นตอนมีดังนี้

เมื่อทำการเขียนโปรแกรมจนเสร็จเรียบร้อยแล้ว ให้เข้าไปที่เมนู Build จากนั้นเลือก build (หรือจะกดปุ่ม F7 ก็ได้) โปรแกรมจะแสดงรายละเอียดเกี่ยวกับการคอมไพล์ทั้งหมดที่หน้าต่าง แสดงการทำงาน โดยไพล์ที่คอมไพล์ไม่ผ่านหรือมีข้อผิดพลาดใด โปรแกรมจะแจ้งให้ทราบถึงข้อผิดพลาดนั้น แต่ถ้าคอมไพล์ผ่าน โปรแกรมจะสร้างไฟล์อื่นๆ ขึ้นมา โดยหนึ่งในนั้นก็คือ ไฟล์ HEX (ไฟล์ HEX จะอยู่ในโฟลเดอร์ default) ที่เราจะนำไปโหลดลงตัวไอซีไมโครคอนโทรลเลอร์นั่นเอง



รูปแสดงผลการคอมไพล์ไฟล์ SUMO1.C ที่คอมไพล์ผ่าน