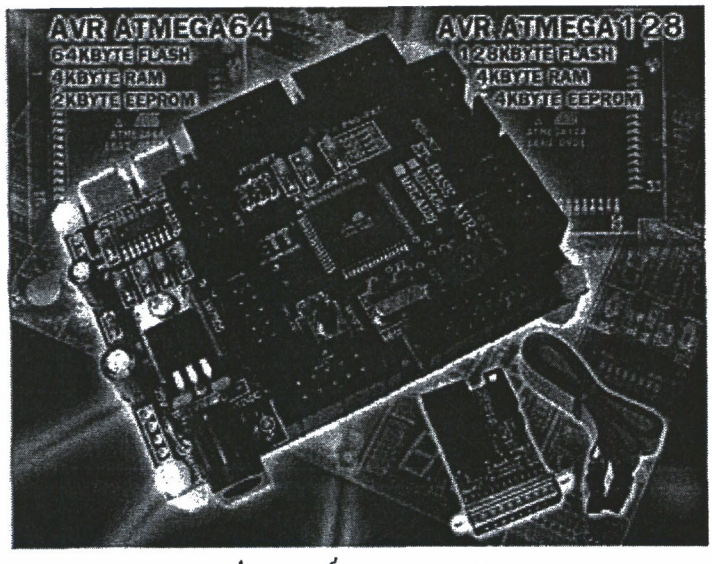
บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 บทนำ

ความเจริญก้าวหน้าของเทคโนโลยีในปัจจุบัน ทำให้ปัจจุบันแนวคิดเกี่ยวกับธรรมชาติที่หาได้ยากและสวยงามดั่งศิลปะถูกพัฒนาขึ้นไปในแนวทางของการวาดรูปภาพหรือเขียนข้อความด้วยน้ำ ซึ่งผลลัพธ์ที่ได้จาก Graphical waterfall ทำให้มาซึ่งลายเส้นที่เคลื่อนไหว รูปทรง และการพริ้วไหว ของแถบริบบิ้นที่ถูกสร้างขึ้นจากศิลปะและธรรมชาติ เมื่อ Graphical waterfall ถูกโปรแกรมขึ้นจะประกอบด้วยภาพกราฟฟิกเคลื่อนไหวมากมาย ไม่ว่าจะเป็นข้อความ ตัวอักษร และ โลโก้ต่างๆซึ่งสร้างความตื่นตาตื่นใจและน่าพิศวง จนทำให้ผู้คนที่ผ่านไปผ่านมา ต้องหยุดชื่นชมและจดจ่อรอดูว่าจะมีอะไรเกิดขึ้นต่อไป

2.2 ไมโครคอนโทรลเลอร์ AVR

ไมโครคอนโทรลเลอร์ AVR เป็นหนึ่งในไมโครคอนโทรลเลอร์ ที่ผลิตโดยบริษัท ATMEL (ผู้นำทางด้านไมโครคอนโทรลเลอร์ตระกูล MCS-51) AVR จัดเป็นไมโครคอนโทรลเลอร์ตระกูลใหม่จาก ATMEL มีสถาปัตยกรรมแบบ RISC(Advanced RISC architecture) คือหนึ่งคำสั่งทำงานใช้สัญญาณนาฬิกาเพียง 1 ลูก (instructions in a single clock cycle) เป็นไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพและความสามารถสูง แบ่งออกเป็นหลายอนุกรม ในแต่ละอนุกรมยังแบ่งออกเป็นหลายเบอร์ เพื่อรองรับความต้องการที่แตกต่างของผู้ใช้งาน ในขณะที่ยังคงความประสิทธิภาพที่เท่ากัน สำหรับไมโครคอนโทรลเลอร์ AVR ที่นำมาใช้ คือ Atmega128

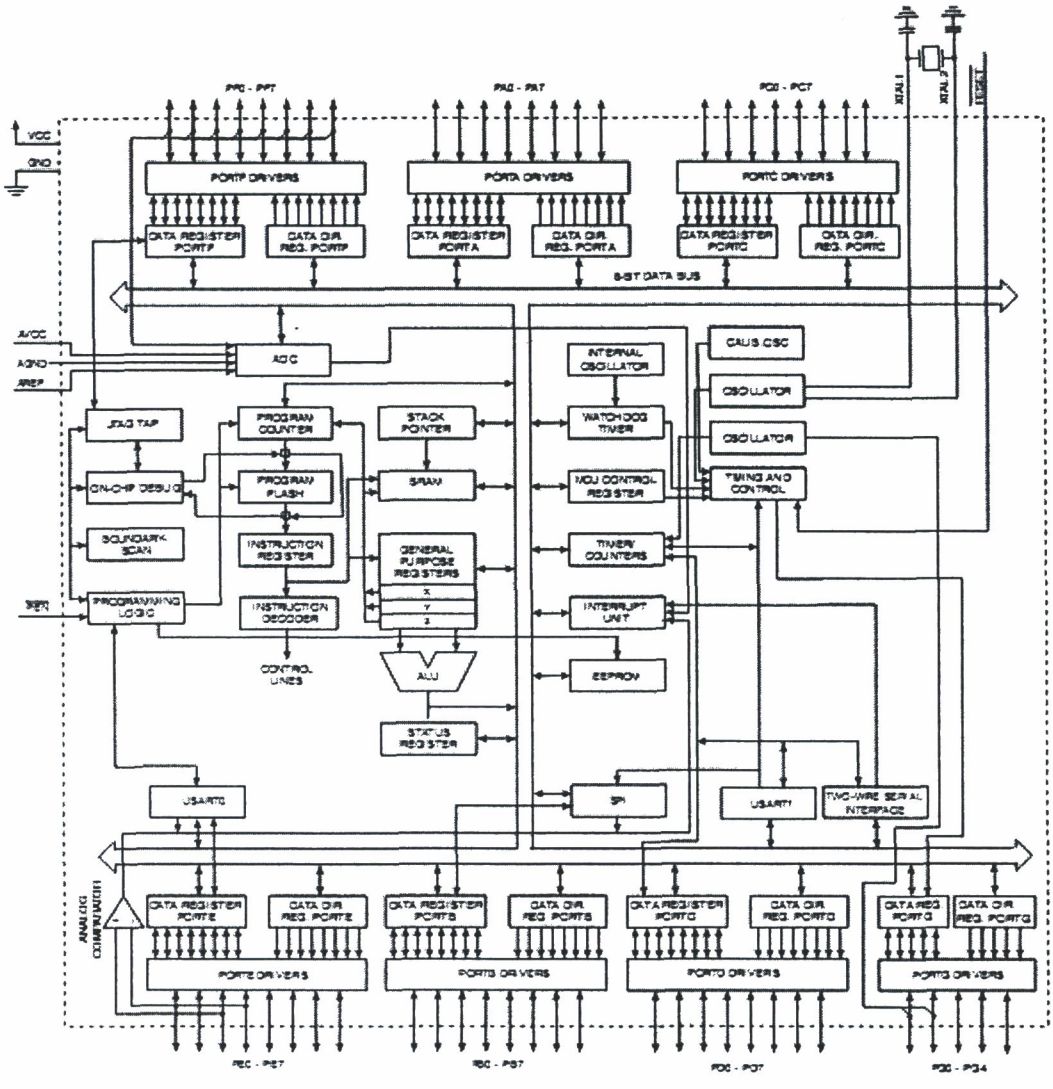


รูป 2.1 บอร์ด ATmega128

2.2.1 คุณสมบัติที่สำคัญ

1) สถาปัตยกรรมภายในเป็นแบบ Advanced RISC
2) มีคำสั่งควบคุมการทำงานมากถึง 133 คำสั่ง โดยมีความเร็วในการประมวลผล 1 คำสั่งต่อ 1 สัญญาณนาฬิกา (1MIP/1MHz)
3) มีรีจิสเตอร์ใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว (ทำให้สะดวกต่อการพัฒนาโปรแกรมด้วยภาษา C เป็นอย่างมาก)
4) ความเร็วในการทำงาน 1 MIPS ต่อ 1 MHz และมากถึง 16 MIPS เมื่อใช้ความถี่ที่ 16 MHz
5) หน่วยความจำ ROM แบบ Flash (มีโหมดป้องกันหน่วยความจำ) ขนาด 128 กิโลไบต์ (เขียน/ลบได้ 10,000 ครั้ง)
6) หน่วยความจำข้อมูลแบบอีอีพรอม ขนาด 4 กิโลไบต์ (เขียน/ลบได้ 100,000 ครั้ง)
7) หน่วยความจำข้อมูลแบบ SRAM 4 กิโลไบต์
8) ไทเมอร์/เคาน์เตอร์ทั้ง 8 บิตและ 16 บิต พร้อมปรีสเกลเลอร์
9) มีระบบตรวจสอบความผิดพลาดในการทำงานของซอฟแวร์ (Watchdog Timer with On-Chip Oscillator)
10) โมดูลสร้างสัญญาณ PWM (Pulse Width Modulator) มีจำนวน 6 ช่อง
11) มีโมดูลแปลงสัญญาณอะนาล็อกเป็นดิจิตอล (ADC) ขนาด 10 บิต มากถึง 8 ช่อง
12) โมดูลเปรียบเทียบแรงดันอะนาล็อก(Analog Comparator)
13) การสื่อสารข้อมูลอนุกรมมีทั้งแบบ UART (Universal Asynchronous Receiver Transmitters) หรือแบบ RS232, SPI (Serial Peripheral Interface) และแบบ I^2C

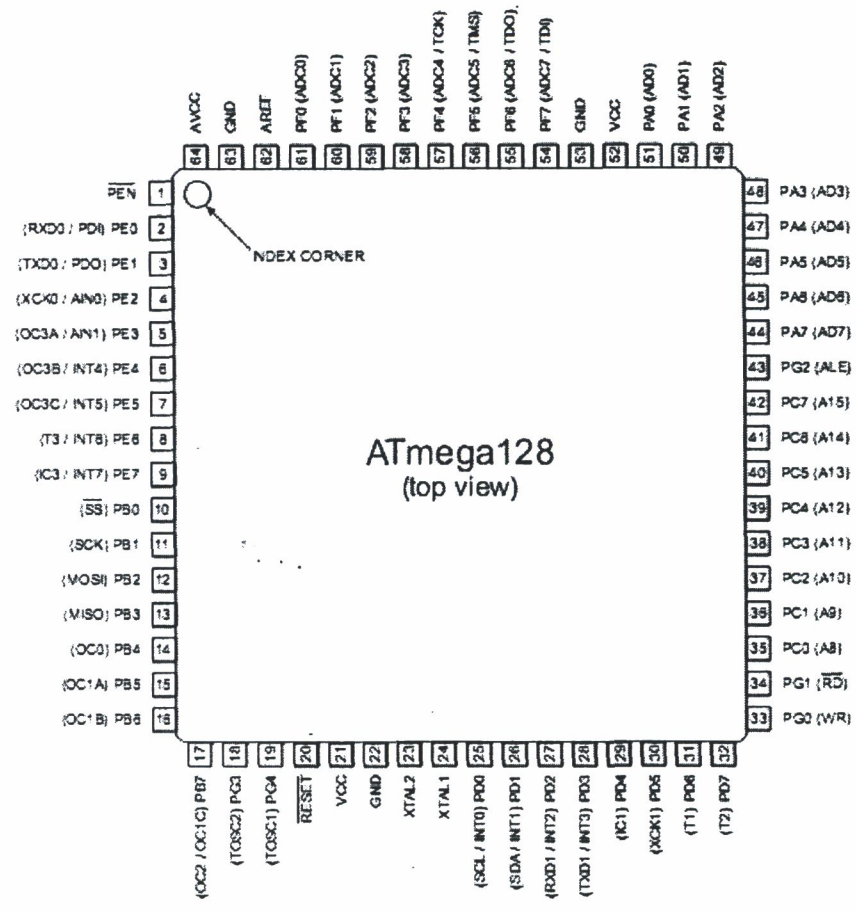
14) มี 6 อินพุตเอาท์พุตพอร์ต



รูป 2.2 บล็อกไดอะแกรม AVR(ATmega128)

2.2.2 ขาพอร์ตอินพุตเอาต์พุต

ขาพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ ATmega 128 มีจำนวน 64 ขา โดยแบ่งขาพอร์ตอินพุตเอาต์พุตอิสระ จำนวน 48 ขา ประกอบไปด้วย PA, PB, PC, PD, PE, PF ขนาด 8 บิต รายละเอียดขาพอร์ตทั้งหมดแสดงดังรูป 2.3



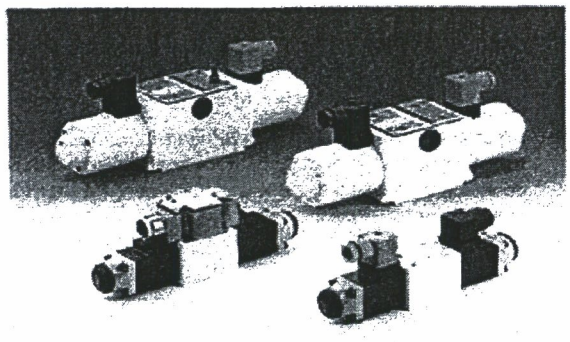
รูป 2.3 ขาพอร์ต AVR(ATmega128)

2.2.3 AVR Studio 4 คอมไพลเลอร์ (AVR Studio 4 C Complier)

เป็นซอฟแวร์สำหรับการแปลโปรแกรมภาษา C ของไมโครคอนโทรลเลอร์ AVR เป็นรหัสเครื่อง การใช้งานพอร์ตอินพุตเอาต์พุตดิจิตอลของไมโครคอนโทรลเลอร์ AVR ด้วยโปรแกรมภาษาซี

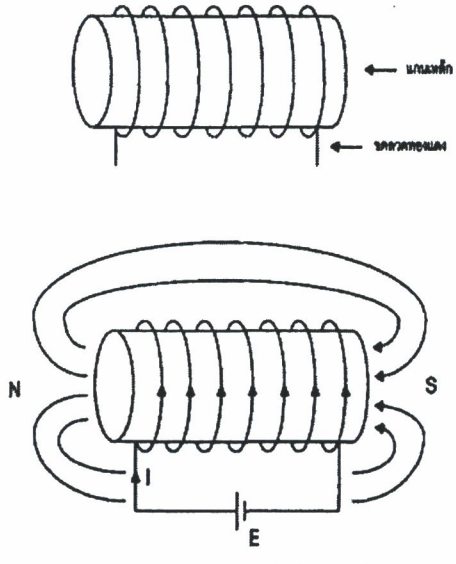
พอร์ตของไมโครคอนโทรลเลอร์ AVR สามารถทำงานเป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุตดิจิตอลและเป็นพอร์ตอินพุตวงจรแปลงสัญญาณอะนาล็อกเป็นดิจิตอล โดยที่การแปลงสัญญาณอะนาล็อกนี้จะขึ้นอยู่กับคุณสมบัติของขาพอร์ตนั้น

2.3 การตัดน้ำด้วยโซลินอยด์วาล์ว



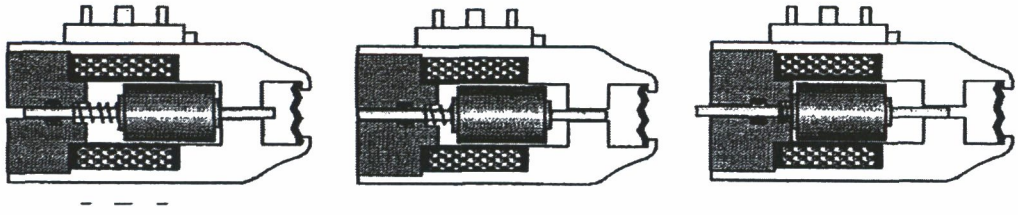
รูป 2.4 ลักษณะของโซลินอยด์วาล์ว

2.3.1 หลักการทำงานของขดลวดโซลินอยด์วาล์ว (Solenoid)



รูป 2.5 ขดลวดโซลินอยด์

ขดลวดโซลินอยด์ คือ ขดลวดที่พันรอบแกนเหล็ก เพื่อสร้างสนามแม่เหล็ก เมื่อป้อนกระแสไฟฟ้าเข้าที่ขดลวดจะเกิดสนามแม่เหล็กขึ้นรอบขดลวด แล้วรวมตัวกันเป็นสนามแม่เหล็กที่ใหญ่ขึ้นโดยมีทิศทางวิ่งจากขั้ว N ไปขั้ว S จากหลักการทำงานขดลวดโซลินอยด์ เมื่อป้อนกระแสไฟฟ้าเข้าที่ขดลวด จะทำให้เกิดสนามแม่เหล็กขึ้นรอบขดลวด ทำให้เกิดการดูดแกน Armature ซึ่งสามารถเอาชนะแรงสปริงที่ดันแกน Armature ไว้ ทำให้แกน Armature เคลื่อนที่ไปอยู่ตรงกลางของขดลวด ดังแสดงในรูปที่ 2.6



ก) ข) ค)

รูป 2.6 สภาวะต่างๆ ของโซลินอยด์

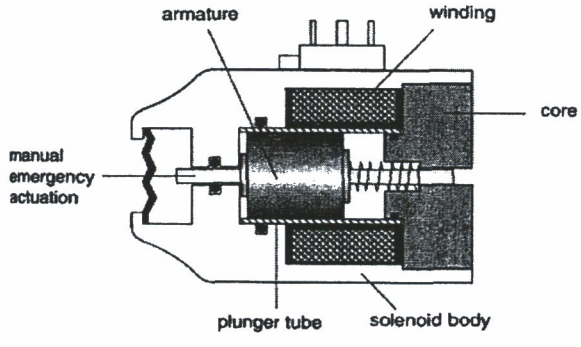
ก) สภาวะปกติ (ไม่ป้อนกระแสไฟฟ้าเข้าขดลวด)

ข) สภาวะทำงาน (ระหว่างป้อนกระแสไฟฟ้าเข้าขดลวด)

ค) สภาวะสุดท้าย (ป้อนกระแสไฟฟ้าเข้าขดลวด)

ขดลวดโซลินอยด์ที่ใช้ในงานควบคุมระบบไฮดรอลิกสามารถจำแนกออกได้ 2 ประเภท คือ

2.3.1.1 ขดลวดโซลินอยด์ชนิดเปียก (Wet solenoid)



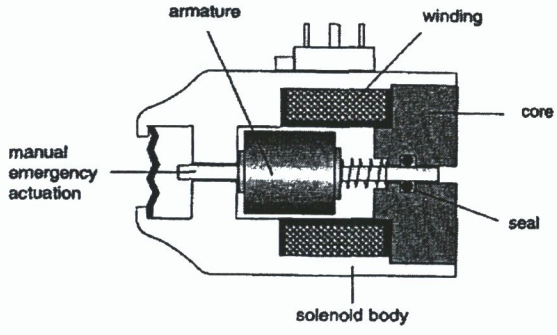
รูป 2.7 ขดลวดโซลินอยด์ชนิดเปียก (Wet solenoid)

ขดลวดโซลินอยด์ชนิดนี้ จะเห็นได้ว่าภายในห้องเลื่อนของแกน Armature จะมีน้ำมันไฮดรอลิกอยู่ด้วย พร้อมกับมีการต่อช่องทางระบายน้ำมันดังกล่าว ซึ่งสามารถไหลกลับสู่ถังพักน้ำมันได้ด้วย โดยส่วนใหญ่ ขดลวดโซลินอยด์ชนิดนี้จะใช้กับขดลวดโซลินอยด์สำหรับ ไฟฟ้ากระแสตรง (DC) เท่านั้น

ส่วนข้อดีของขดลวดโซลินอยด์ชนิดนี้ คือ

1) ใช้น้ำมันเป็นซีล (Seal) บาง ๆ ทำให้ลดการเสียดสีระหว่างแท่ง Armature กับผนังห้องเลื่อน
2) ป้องกันการเกิดสนิมภายในห้องเลื่อนแกน Armature
3) น้ำมันทำหน้าที่เป็นเบาะกันกระแทกในจังหวะทำงานได้

2.3.1.2 ขดลวดโซลินอยด์ชนิดแห้ง (Dry solenoid)



รูป 2.8 ขดลวดโซลินอยด์ชนิดแห้ง (Dry solenoid)

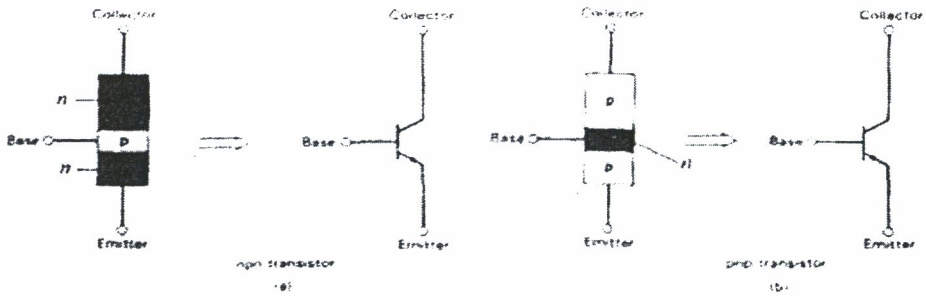
ขดลวดโซลินอยด์ชนิดนี้ จะมีซีล (Seal) รีดน้ำมันที่บริเวณช่วงปลายแกน Armature เพื่อป้องกันไม่ให้น้ำมันไฮดรอลิกซึมผ่านเข้าไปภายในห้องเลื่อนแกน Armature ได้ ขดลวดโซลินอยด์ชนิดนี้ สามารถใช้ได้กับไฟฟ้ากระแสตรง และกระแสสลับก็ได้

2.4 ทรานซิสเตอร์

ทรานซิสเตอร์แบ่งได้เป็นสองประเภทคือ ทรานซิสเตอร์แบบรอยต่อคู่ (Bipolar Junction Transistor, BJTs) และทรานซิสเตอร์แบบสนามไฟฟ้า (Field Effect Transistors,FETs) ทรานซิสเตอร์จะมีขาเชื่อมต่อสามจุด อธิบายโดยย่อคือเมื่อมีการปรับเพิ่มแรงดันไฟฟ้าที่ขาหนึ่งจะส่งผลให้ความนำไฟฟ้าระหว่างขาที่เหลือสูงขึ้นอันทำให้สามารถควบคุมการไหลของกระแสไฟฟ้าได้ อย่างไรก็ตามหลักทางฟิสิกส์ในการทำงานของทรานซิสเตอร์ทั้งสองแบบ (ชนิดรอยต่อคู่และชนิดสนามไฟฟ้า) มีความแตกต่างกันอยู่มาก ซึ่งจะกล่าวถึงเฉพาะทรานซิสเตอร์แบบรอยต่อคู่

2.4.1 ประเภทของทรานซิสเตอร์ (Type of Transistors)

ทรานซิสเตอร์แบ่งตามโครงสร้างได้ 2 ประเภท คือ ทรานซิสเตอร์แบบ npn (npn Transistor) และทรานซิสเตอร์แบบ pnp (pnp Transistor)



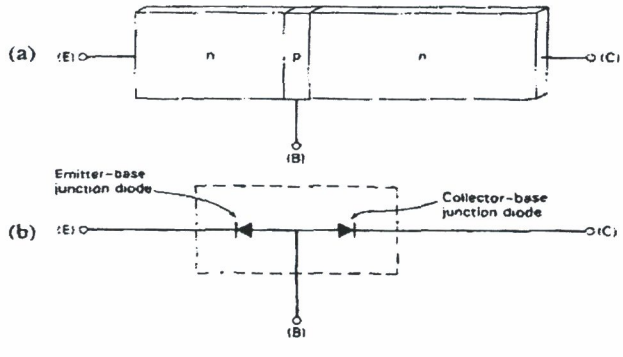
รูป 2.9 โครงสร้างทรานซิสเตอร์

ทรานซิสเตอร์แบบ npn ประกอบด้วยสารกึ่งตัวนำชนิด n จำนวน 2 ชิ้นต่อเชื่อมกับสารกึ่งตัวนำชนิด p จำนวน 1 ชิ้น

ทรานซิสเตอร์แบบ pnp ประกอบด้วยสารกึ่งตัวนำชนิด p จำนวน 2 ชิ้นต่อเชื่อมกับสารกึ่งตัวนำชนิด n จำนวน 1 ชิ้น

2.4.2 โครงสร้างและการทำงานของทรานซิสเตอร์ (Transistor Construction and Operation)

ได้กล่าวมาแล้วว่าทรานซิสเตอร์ประกอบด้วยสารกึ่งตัวนำ 3 ชิ้นต่อเชื่อมกัน ดังนั้นจึงมีรอยต่อ pn จำนวน 2 ตำแหน่ง



รูป 2.10 โครงสร้างของทรานซิสเตอร์

ตำแหน่งที่อิมิตเตอร์กับเบสเชื่อมกันเป็นรอยต่อ PN เรียกว่า รอยต่ออิมิเตอร์-เบส (Emitter Base Junction) ส่วนตำแหน่งที่คอลเลคเตอร์กับเบสต่อเชื่อมกันเรียกว่า รอยต่อคอลเลคเตอร์-เบส (Collector Base Junction) เขียนแทนได้ด้วย ค่าเทียบเคียงของไดโอด เมื่อนำหลักการมาร่วมพิจารณา ทำให้ทราบว่าการที่จะนำทรานซิสเตอร์ไปใช้งานได้นั้นต้องต่อแรงดันไฟฟ้าเพื่อทำการไบอัสที่รอยต่อหรือไดโอดเทียบเคียงทั้งสอง เนื่องจากทรานซิสเตอร์ มี 3 ขั้ว การต่อแรงดันไฟฟ้าที่ขั้วเพื่อให้ทรานซิสเตอร์ทำงานจึงเป็นไปได้ 3 แบบคือ

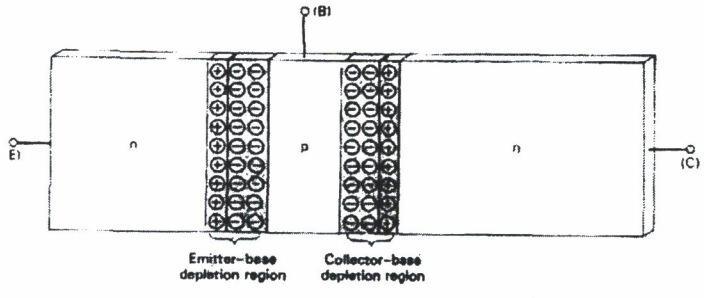
1) การให้ทรานซิสเตอร์ทำงานที่บริเวณคัตออฟ (Cut-off Region)

2) การให้ทรานซิสเตอร์ทำงานที่บริเวณอิ่มตัว (Saturation Region)

3) การให้ทรานซิสเตอร์ทำงานที่บริเวณแอกตีฟ (Active Region)

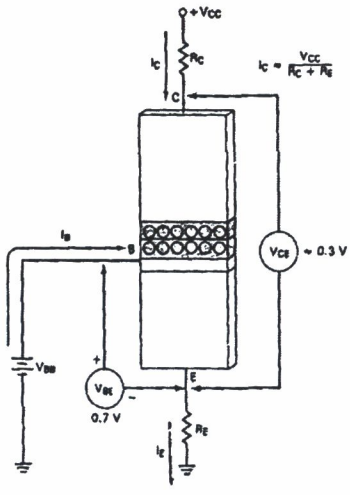
ในการอธิบายถึงการทำงานที่บริเวณต่าง ๆ ของทรานซิสเตอร์นั้น จะเริ่มต้นจากกรณีไม่มีการต่อแรงดันที่ขั้วของทรานซิสเตอร์ หรือกรณีไม่ได้รับการไบอัส

กรณีไม่ได้รับการไบอัส ขณะทรานซิสเตอร์ไม่ได้รับการไบอัส จะเกิดบริเวณปลอดพาหะ (Depletion Region) ที่รอยต่อทั้งสอง การต่อแหล่งจ่ายไฟฟ้าให้ทรานซิสเตอร์ทำงานในบริเวณคัตออฟเป็นการไบอัสกลับที่รอยต่อทั้ง 2 ตำแหน่ง ซึ่งจะทำให้กระแสที่ไหลผ่านขั้วทั้งสามมีค่าใกล้ศูนย์ จากการต่อวงจรในลักษณะดังกล่าวบริเวณปลอดพาหะทั้งสองบริเวณจะขยายกว้างขึ้น จึงมีเพียงกระแสย้อนกลับ (Reverse Current) กระแสรั่วไหลปริมาณต่ำมากเท่านั้นที่ไหลจากคอลเลคเตอร์ไปยังอิมิตเตอร์ได้



รูป 2.11 การทำงานที่บริเวณคัตออฟ

การทำงานที่บริเวณอิ่มตัว เมื่อทราบว่าถ้าค่า IB เพิ่มขึ้น IC ก็จะเพิ่มขึ้นด้วย เมื่อ IC เพิ่มขึ้นจนถึงค่าสูงสุด หรือ เรียกว่า ทรานซิสเตอร์เกิดการอิ่มตัว ณ ตำแหน่งนี้ค่า IC จะเพิ่มตามค่า IB ไม่ได้อีกแล้ว การหาค่า IC ทำได้โดยใช้ VCC หารด้วยผลรวมของความต้านทานที่ขั้วคอลเลคเตอร์ (RC) กับความต้านทาน ที่ขั้วอิมิตเตอร์(RE) ดังรูป 2.13

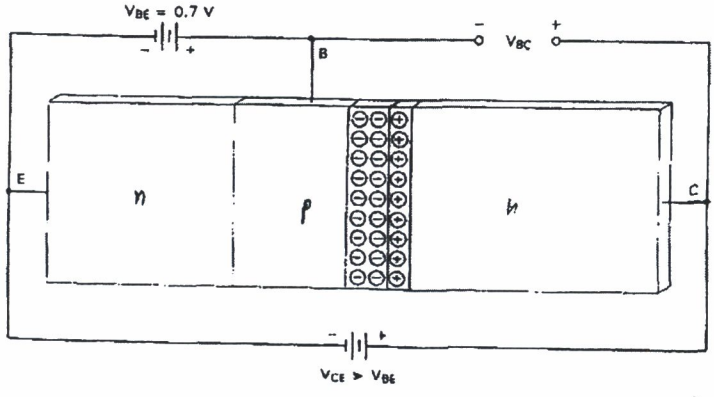


รูป 2.12 วงจรสมมุติ

สมมติขณะที่ V_{CE} ของทรานซิสเตอร์มีค่า 0 V (สภาพในอุดมคติ) I_C จะขึ้นอยู่กับค่า V_{CC}, R_C และ R_E ดังนี้

$$I_C = V_{CC} / (R_C + R_E) \qquad (2.1)$$

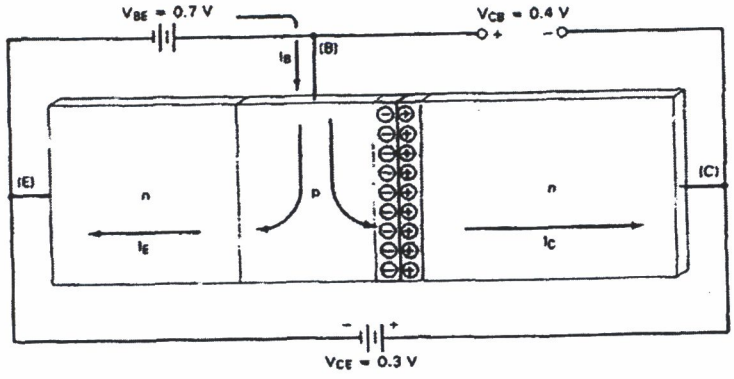
การต่อแหล่งจ่ายไฟฟ้าให้ทรานซิสเตอร์ทำงานในบริเวณอิ่มตัว เป็นการไบอัสตรงที่รอยต่อทั้ง 2 ตำแหน่ง ของทรานซิสเตอร์



รูป 2.13 การต่อแหล่งจ่ายไฟฟ้าให้ทรานซิสเตอร์ทำงานในบริเวณอิ่มตัว

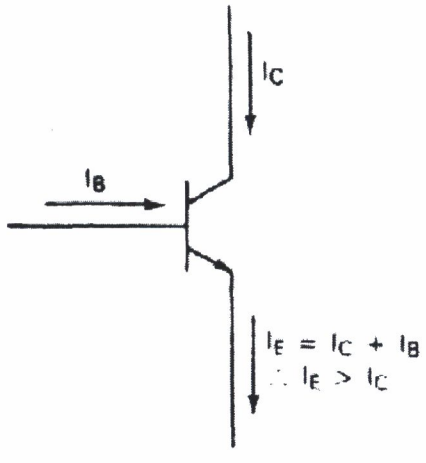
สมมติค่า V_{CE} ของทรานซิสเตอร์ขณะอิ่มตัว มีค่า 0.3 V (ซึ่งต่ำกว่า V_{BE} ที่มีค่า0.7 V) บริเวณรอยต่อคอลเลคเตอร์-เบส จะได้รับการไบอัสตรงด้วยผลต่างระหว่างแรงดัน V_{BE} กับ V_{CE} (เท่ากับ 0.4 V) กระแสไฟฟ้า I_E, I_C และ I_B จะมีทิศทาง

สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุดงานวิจัย



รูป 2.14 ทิศทางการไหลของกระแสไฟฟ้า

การทำงานที่บริเวณแอกตีฟ การต่อแหล่งจ่ายไฟฟ้าให้ทรานซิสเตอร์ทำงานในบริเวณแอกตีฟเป็นการแอกตีฟเป็นการไบอัสตรงที่รอยต่อ อิมิตเตอร์-เบส และไบอัสกลับที่รอยต่อคอลเลคเตอร์-เบส



รูป 2.15 การต่อแหล่งจ่ายไฟฟ้าให้ทรานซิสเตอร์ในบริเวณแอกตีฟ

การอธิบายหลักการทำงานของทรานซิสเตอร์ในบริเวณนี้จะง่ายขึ้น ถ้าพิจารณาเฉพาะรอยต่ออิมิตเตอร์-เบส สมมติ V_{BE} มีค่ามากพอที่จะทำให้ไดโอดทำงาน (Si ประมาณ 0.7 V และGe ประมาณ 0.3 V)] รอยต่อคอลเลคเตอร์-เบสได้รับการไบอัสกลับ ทำให้บริเวณปลอดพาหะกว้างกว่าที่รอยต่ออิมิตเตอร์-เบสซึ่ง ได้รับการไบอัสตรง ดังนั้น ความต้านทานที่เบส (R_B) จึงมีค่าสูง เมื่อพิจารณาในรูปของไดโอดจะเห็นว่า IB เป็นกระแสที่มีค่าต่ำมาก เมื่อเทียบกับกระแสคอลเลคเตอร์ (I_C) และเป็นส่วนหนึ่งของ I_E ดังนั้น I_E ส่วนใหญ่จึงเป็นกระแส I_C ซึ่งผ่านรอยต่อคอลเลคเตอร์- เบสของทรานซิสเตอร์

สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุดงานวิจัย
วันที่ 7 ส.ค. 2555
เลขทะเบียน 190943
เลขเรียกหนังสือ

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย

ค่าพิกัดของทรานซิสเตอร์มีหลายประเภท ในหัวข้อนี้จะกล่าวถึงค่าพิกัดเฉพาะบางประเภทอันเป็นพื้นฐาน สำคัญสำหรับการนำทรานซิสเตอร์ไปใช้ให้ได้ประสิทธิภาพสูงสุด และหลีกเลี่ยงไม่ให้เกิดความเสียหายใดๆซึ่งได้แก่ พิกัดเบตาไฟฟ้ากระแสตรง, พิกัดอัลฟาไฟฟ้ากระแสตรง, พิกัดกระแสไฟฟ้าสูงสุด และพิกัดแรงดันไฟฟ้าสูงสุด

เบตาไฟฟ้ากระแสตรง (DC BETA) พิกัดเบตาไฟฟ้ากระแสตรงของทรานซิสเตอร์ซึ่งมักเรียกสั้น ๆ ว่าเบตา เป็นอัตราส่วนของ I_C ต่อ I_B เขียน เป็นสมการได้ดังนี้ คือ

$$\text{Beta} = I_C / I_B \qquad (2.2)$$

วงจรทรานซิสเตอร์ส่วนมากมีสัญญาณอินพุตจ่ายให้ขั้วเบส และสัญญาณเอาต์พุตออกจากขั้วคอลเลคเตอร์ เบตาของทรานซิสเตอร์จึงเป็นสัญลักษณ์แทนอัตราขยายกระแส dc (dc Current Gain) ของทรานซิลเตอร์

หาค่ากระแสอิมิตเตอร์ได้ ดังนี้

$$I_C = \text{Beta} * I_B \qquad (2.3)$$

$$I_E = I_B + I_C$$

$$= I_B + \text{Beta} * I_B$$

$$I_E = I_B (1+\text{Beta}) \qquad (2.4)$$

เราใช้เบตาและกระแสไฟฟ้าที่ขั้วใดขั้วหนึ่งหาค่ากระแสไฟฟ้าที่ขั้วอื่น ๆ ได้

อัลฟาไฟฟ้ากระแสตรง (DC Alpha) พิกัดอัลฟาของทรานซิสเตอร์ ซึ่งมักเรียกสั้น ๆ ว่าอัลฟา คือ อัตราส่วน I_C ต่อ I_E เขียนเป็น สมการได้ ดังนี้

$$\text{Alpha} = IC / IE \qquad (2.5)$$

เมื่อนำกฎกระแสไฟฟ้าของเคอร์ชอฟฟ์มาร่วมพิจารณา จะเห็นได้ว่าความสัมพันธ์ระหว่างกระแสไฟฟ้าที่ ขั้วทั้งสามของทรานซิสเตอร์เป็นดังสมการ 1 คือ

$$I_E = I_B + I_C \qquad (2.6)$$

$$I_C = I_E - I_B \qquad (2.7)$$

2.5 สายสัญญาณสำหรับการเชื่อมต่อบอร์ดคอนโทรลเลอร์กับเครื่องคอมพิวเตอร์

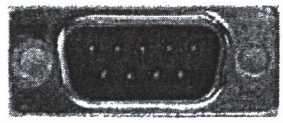
สายสัญญาณ ที่เลือกใช้คือ RS-232

2.5.1 พอร์ตอนุกรม RS232 (Serial Port)

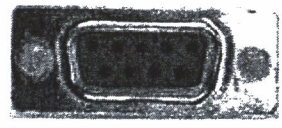
RS-232 ย่อมาจาก Recommended Standard-232 (มาตรฐานแนะนำรุ่น 232) เป็นมาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรม (Serial Port) กำหนดโดย EIA (Electronics Industry Association) หรือ สมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกา ใช้กับการสื่อสารแบบจุดต่อจุด โดยใช้สายเชื่อมต่อ DB แบบ 25 และ 9 เข็ม ที่ไม่ประสานจังหวะระหว่างคอมพิวเตอร์กับอุปกรณ์ต่อพ่วง มีการทำงานแบบสองทางพร้อมกัน (Full-duplex) ลักษณะโดยทั่วไปของการเชื่อมต่อข้อมูลแบบอนุกรมตามมาตรฐาน RS-232 คือเป็นการสื่อสารข้อมูลแบบจุดต่อจุด ซึ่งเดิมทีเป็นการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับโมเด็ม ซึ่งจริงๆ แล้วทั้งสองฝั่งจะเป็นอะไรก็ได้ การสื่อสารเป็นแบบสองทางพร้อมกัน (Full-duplex) โดยอาจใช้สายสัญญาณอื่นร่วมเพื่อทำแฮนด์เชค (Hand-shake) หรือไม่ก็ได้ มาตรฐาน RS-232 จำกัดความยาวสายไว้ที่ 50 ฟุต (หรือประมาณ 15 เมตร) สำหรับการส่งสัญญาณที่ความเร็ว 19,200 บิทต่อวินาที โดยที่ความยาวสายจะต้องสั้นลงถ้าต้องการสื่อสารที่ความเร็วสูงขึ้น และถ้ามีสัญญาณรบกวนมากๆ เช่น ในโรงงานหรือบริเวณใกล้เครื่องจักรที่เป็นแบบมีการสวิทช์สัญญาณไฟฟ้าที่กระแสสูงๆ ก็จะทำให้ต้องมีการลดความเร็วในการส่งสัญญาณลงหรือใช้สายที่สั้นลง

RS-232 มีจุดเริ่มต้นจากความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่างคอมพิวเตอร์กับโมเด็มในสมัยนั้น ตัวมาตรฐานจะกำหนดสิ่งที่เกี่ยวข้องกับการเชื่อมต่อนี้ด้วยกันทั้งหมด 4 หัวข้อหลักๆ ด้วยกันคือ

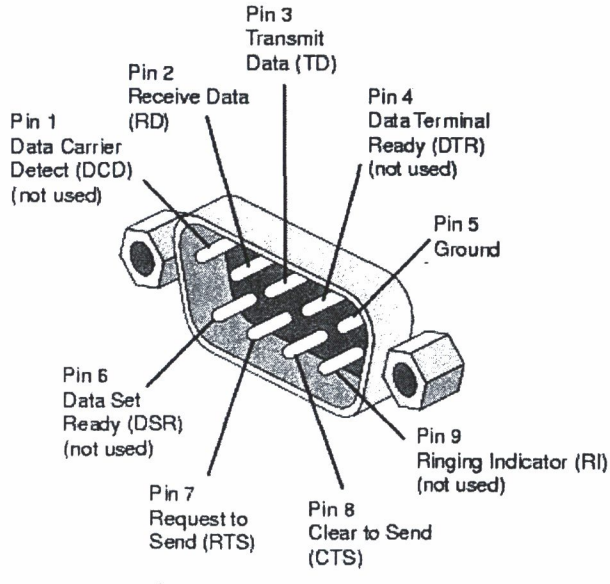
1) คุณสมบัติทางไฟฟ้าของสัญญาณ
2) คุณสมบัติทางกลของการเชื่อมต่อ ซึ่งหมายถึงตัวคอนเน็กเตอร์
3) หน้าที่การทำงานของวงจรสำหรับแลกเปลี่ยนข้อมูล
4) มาตรฐานการเชื่อมต่อสำหรับระบบสื่อสารเฉพาะอย่าง



รูป 2.16 DB9-male ตัวผู้



รูป 2.17 DB9-Female ตัวเมีย

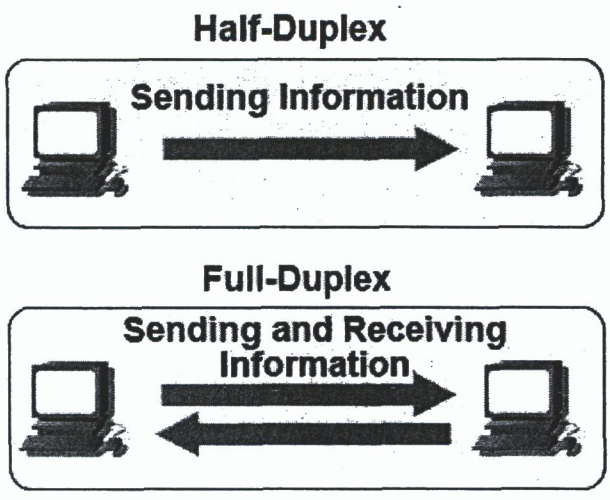


รูป 2.18 ขาสัญญาณของ RS232

ตาราง 2.1 โครงสร้างขาของพอร์ตอนุกรม DB9

ขา	คำอธิบาย	ชนิด
1	Data Carrier Detect (DCD)	Input
2	Received Data (RXD)	Input
3	Transmitted Data (TXD)	Output
4	Data Terminal Ready (DTR)	Output
5	Signal Ground (GND)	Input
6	Data Set Ready (DSR)	Input
7	Request To Send (RTS)	Output
8	Clear to Send (CTS)	Input
9	Ring Indicator (RI)	Input

สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย



รูป 2.19 การทำแฮนด์เชค

2.5.1.1 การทำงานของขาสัญญาณ DB9

1) TXD เป็นขาที่ใช้ส่งข้อมูล
2) RXD เป็นขาที่ใช้รับข้อมูล
3) DTR แสดงสภาวะพอร์ตว่าเปิดใช้งาน, DSR ตรวจสอบว่าพอร์ตที่ติดต่อด้วยเปิดอยู่หรือไม่ เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าต้องการติดต่อด้วย ในขณะเดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่
4) RTS แสดงสภาวะพอร์ตว่าต้องการส่งข้อมูล, CTS ตรวจสอบว่าพอร์ตที่ติดต่ออยู่ต้องการส่งข้อมูลหรือไม่ เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF ในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่
5) GND ขากราวด์ (Ground)

2.6 ระบบบัสไอสแควร์ซี

ไอสแควร์ซี(I^2C – Inter-IC Communication) เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสารระหว่าง ไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อสั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ serial data (SDA) และสาย serial clock (SCL) เป็นสายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน ซึ่งสามารถเชื่อมต่ออุปกรณ์จำนวนหลายๆ ตัว เข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น

อัตราการถ่ายทอดข้อมูลบนบัส I2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (standard mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (fast mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส I^2C ใช้ข้อมูลสำหรับการเข้าถึง 2 แบบ คือ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing)

ข้อเด่นอีกประการหนึ่งของบัส I^2C คือ สามารถเชื่อมต่ออุปกรณ์ ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ โดยอุปกรณ์บนบัส I^2C ตัวหนึ่งอาจใช้ไฟเลี้ยง +5V ในขณะที่อีกตัวหนึ่งใช้ไฟเลี้ยง +12V การต่อร่วมกันบนบัส I2C สามารถกระทำได้ในลักษณะเดียวกันการกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน กล่าวคือ ให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทานพูลอัป (Rp) เข้ากับแรงดัน +5V ไว้ด้วยเสมอ ในกรณีที่อาจมีแรงดันไฟกระชากขนาดใหญ่ปะปนเข้ามาในบัส I2C ที่ขา SDA และ SCL ของอุปกรณ์แต่ละตัวต้องต่อตัวต้านทานอนุกรมกับขา SDA และ SCL เรียกว่า RS ก่อนต่อเข้าสู่บัส I^2C

2.6.1 หลักการของบัสไอสแควร์ซี

บัสไอสแควร์ซีประกอบด้วยสายสัญญาณ 2 เส้น ดังที่ได้กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือ เรียกว่า โปรโตคอล (protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูลเรียกว่า ตัวส่ง (transmitter) อุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (receiver) อุปกรณ์บนบัส I^2C สามารถเป็นได้ทั้งตัวรับและตัวส่งบางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I^2C ที่ทำหน้าที่เป็นตัวส่งอย่างเดียว อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I^2C เรียกว่า มาสเตอร์ (master) อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I^2C เรียกว่า สเลฟ (slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I^2C คือ

1) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น
2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้นสัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาควบคุมแทน

2.6.2 สภาวะที่เกิดขึ้นบนบัสไอสแควร์ซี

มีด้วยกัน 5 สภาวะ ดังนี้

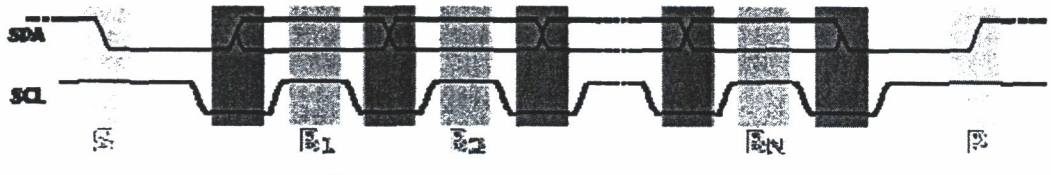
1) บัสว่าง (Bus not busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ หมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้

2) เริ่มต้นการถ่ายทอดข้อมูล (start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่าสภาวะ เริ่มต้น (START)

3) หยุดการถ่ายทอดข้อมูล (stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (STOP)

4) ข้อมูลดำรงอยู่บนบัส (data valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่า เป็น "0" หรือ "1" ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุด หรือ สภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้น เกิดความผิดพลาดขึ้น

5) รับข้อมูล (acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่าง สมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูล มา 1 บิตเรียกว่า บิตรับรู้ (acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์ จะทำการส่งสัญญาณรับรู้พิเศษ ซึ่งสัมพันธ์กับสัญญาณนาฬิกาเพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะ กำเนิดบิตรับรู้เพื่อตอบสนอง ให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว



รูป 2.20 ไดอะแกรมเวลาแสดงสถานะต่างๆ ในบัส I^2C

2.6.3 การเขียนโปรแกรมติดต่อบัสไอสแควร์ซี

เริ่มต้นด้วยการสร้างสภาวะมาตรฐานของบัสไอสแควร์ซี อันประกอบด้วย สภาวะเริ่มต้น,สภาวะสิ้นสุดการส่งข้อมูล,สภาวะหยุด, สัญญาณนาฬิกาบนขาสายสัญญาณนาฬิกาอนุกรม, การเขียนและอ่านข้อมูลกับอุปกรณ์บนระบบบัสไอสแควร์ซี

2.6.3.1 การสร้างสภาวะเริ่มต้น

1) เมื่อต้องการติดต่อกับบัสไอสแควร์ซีสิ่งแรกที่ต้องทำสำหรับไมโครคอนโทรลเลอร์ซึ่งถือว่าเป็นอุปกรณ์มาสเตอร์คือ การทำให้บัสว่างด้วยการกำหนดให้สายสัญญาณนาฬิกาอนุกรมและขาสายข้อมูลอนุกรมมีลอจิกเป็น “1” ทั้งคู่
2) จากนั้นทำให้ขาสายข้อมูลอนุกรมมีลอจิก “0” โดยที่ขาสายสัญญาณนาฬิกาอนุกรมยังคงเป็นลอจิก “1” อยู่
3) กำหนดให้ขาสายสัญญาณนาฬิกาอนุกรมมีลอจิกเป็น “0” ถึงตอนนี้ทั้งสายสัญญาณนาฬิกาอนุกรมและสายข้อมูลอนุกรมจะมีลอจิกเป็น “0” ทั้งคู่พร้อมที่จะติดต่อได้แล้ว

2.6.3.2 การสร้างสภาวะหยุด

1) เมื่อต้องการหยุดส่งข้อมูล ต้องส่งสภาวะหยุดออกไปโดยในตอนแรกต้องกำหนดให้ขาสายสัญญาณนาฬิกาอนุกรม และสายข้อมูลอนุกรมเป็นลอจิก “0” ทั้งคู่ก่อน
2) กำหนดให้ขาสายสัญญาณนาฬิกาอนุกรมมีลอจิกเป็น “1” โดยสายข้อมูลอนุกรมยังคงมีลอจิกเป็น “0”
3) กำหนดให้ขาสายข้อมูลอนุกรมมีลอจิกเป็น “1” ทำให้กลับเข้าสู่บัสว่างอีกครั้งพร้อมที่จะรับหรือส่งข้อมูลต่อไป

2.6.3.3 การส่งข้อมูลลอจิก “0” และลอจิก “1”

หลังจากส่งบิตเริ่มต้นแล้วลำดับต่อไปคือ ส่งข้อมูลควบคุมซึ่งจะเป็นขบวนของลอจิก “0” และ “1” ดังนี้

1) กำหนดให้ขาสายข้อมูลอนุกรมเป็น “0” สำหรับการส่งข้อมูลลอจิก “0”
2) กำหนดให้ขาสายสัญญาณนาฬิกาอนุกรมเป็น “1” สำหรับการป้อนสัญญาณนาฬิกาในขณะที่ขาสายข้อมูลอนุกรมยังคงเป็น “0” อยู่
3) จากนั้นทำให้ขาสายสัญญาณนาฬิกาอนุกรมกลับมามีสถานะเป็นลอจิก “0” เหมือนเดิม

2.6.3.4 ในขณะที่การส่งข้อมูลลอจิก "1" มีขั้นตอนดังนี้

1) กำหนดให้ขาสายข้อมูลอนุกรมมีลอจิกเป็น "1" สำหรับการส่งข้อมูลลอจิก "1"
2) กำหนดให้ขาสายสัญญาณนาฬิกาอนุกรมเป็น "1" สำหรับการส่งสัญญาณนาฬิกาโดยที่ขาสายข้อมูลอนุกรมยังคงเป็น "1"
3) จากนั้นทำให้ขาสายสัญญาณนาฬิกาอนุกรมกลับมามีสถานะเป็นลอจิก "0" เหมือนเดิม

ข้อมูลที่ใช้ในการส่งไปยังขาสายข้อมูลอนุกรมนั้นจะกำหนดที่แอกคิวมูเลเตอร์ (Accumulator) แล้วทำการส่งออกไปยังแฟลกทดด้วยการใช้คำสั่งหมุนข้อมูล (RLC A) เพื่อถ่ายทอดต่อไปยังขาสายข้อมูลอนุกรมต่อไป

2.6.4 บิตแสดงแอดเดรสอุปกรณ์

บนระบบบัส I^2C สามารถที่จะต่ออุปกรณ์ได้มากกว่าหนึ่งตัว การติดต่อกับอุปกรณ์แต่ละตัวนั้นจะอ้างอิงด้วยแอดเดรสของอุปกรณ์ตัวนั้น โดยมีแอดเดรสทั้งแบบ 7 บิต และ 10 บิต

2.6.4.1 การอ้างอิงแอดเดรสแบบ 7 บิตและ 10 บิต

แอดเดรสแบบ 7 บิต จะอ้างถึงอุปกรณ์แต่ละตัวด้วยแอดเดรสบิตที่ 1 ถึง 7ส่วนบิตที่ 0 จะเป็นบิตกำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟ โดยบิตที่ 0 ถ้ากำหนดให้เป็น 0 จะเป็นการเขียนข้อมูลให้กับอุปกรณ์สเลฟและหากเป็น 1 จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ

2.7 การสื่อสารข้อมูลอนุกรมผ่านโมดูล USART

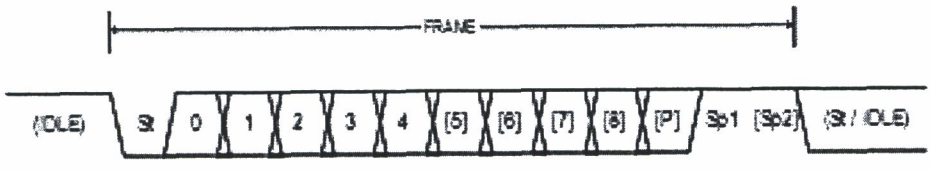
ชุดควบคุมนั้นถูกพัฒนาบนไมโครคอนโทรลเลอร์ ซึ่งในการสื่อสารข้อมูลอนุกรมนั้น สามารถกระทำผ่านโมดูล USART (Universal Synchronous and Asynchronous) เพื่อสื่อสารข้อมูลผ่านทางพอร์ตอนุกรมได้ทั้งแบบซิงโครนัส (ข้อมูลมีความต่อเนื่อง มีการกำหนดสัญญาณมาตรฐานที่เหมือนกันทั้งทางด้านรับและด้านส่ง เพื่อให้การรับส่งมีความสัมพันธ์กัน) และอะซิงโครนัส (ข้อมูลไม่จำเป็นต้องต่อเนื่องมีบิตเริ่มต้น (Start bit) บิตข้อมูล (Data bit) และบิตหยุด (Stop bit) มีบิตพาริตี้ (Parity bit) หรือไม่มีก็ได้)

2.7.1 โมดูล UASRT แบ่งออกเป็น 3 ส่วน

1) ส่วนสร้างสัญญาณนาฬิกา (Clock Generator) เพื่อใช้ในการกำหนดอัตราบอดในการรับส่งข้อมูล โดยสามารถกำหนดได้ทั้งภายในและภายนอก ผ่านทางขา XCK (Transfer Clock)
2) ส่วนส่งข้อมูลอนุกรม (Transmitter) โดยส่งข้อมูลออกทางขาพอร์ต TxD
3) ส่วนรับข้อมูลอนุกรม (Receiver) โดยการรับข้อมูลจากขาพอร์ต RxD

มีรีจีสเตอร์ควบคุมการทำงาน 3 ตัวประกอบไปด้วย UCSRA, UCSRB และ UCSRC การส่งข้อมูลอนุกรมในรูปแบบอะซิงโครนัส จะเป็นการส่งข้อมูลเป็นเฟรมลักษณะของเฟรมข้อมูลอนุกรมนี้ประ กอบไปด้วย

1) บิตเริ่มต้นข้อมูล (Start bit)
2) บิตข้อมูล (Data bit)
3) พาริตี้บิต (Parity bit)
4) บิตหยุดข้อมูล (Stop bit)



รูป 2.21 รูปแบบเฟรมข้อมูลในการส่งข้อมูลแบบ UASRT

2.7.2 คุณสมบัติที่สำคัญของโมดูล USART

1) การสื่อสารข้อมูลแบบฟูลดูเพลกซ์ (Full Duplex) ตัวรับและตัวส่งแยกอิสระต่อกัน สามารถรับและส่งได้ในเวลาเดียวกัน (พร้อมกัน)
2) ทำงานได้ทั้งในรูปแบบซิงโครนัสและอะซิงโครนัส
3) มีคุณสมบัติของพอร์ตอนุกรมครบถ้วน เช่น การกำหนดบิตข้อมูล การกำหนดบิตหยุด และการกำหนดพาริตี้ เป็นต้น
4) มีส่วนตรวจสอบความผิดพลาดของเฟรมข้อมูลและข้อมูลโอเวอร์รัน (Framing Error and Data Overrun Detection)
5) โหมดการสื่อสารแบบมัลติโปรเซสเซอร์
6) โหมดทวีคูณความเร็วในการสื่อสารแบบอะซิงโครนัส

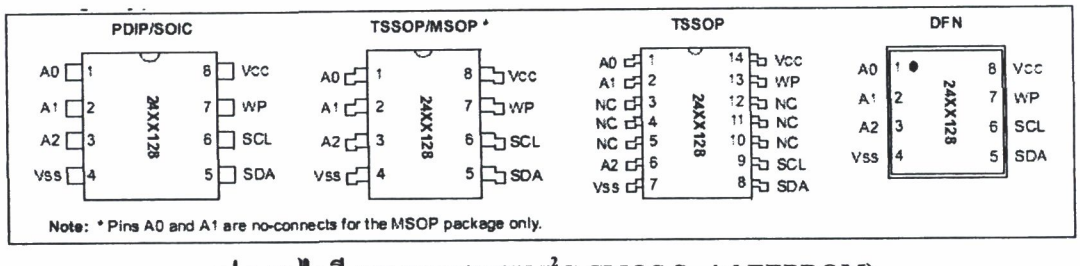
2.7.3 รีจิสเตอร์

1) รีจีสเตอร์ UDR (USART I/O Data Register) รีจีสเตอร์อ่านเขียนข้อมูลขนาด 8 บิต โดยแบ่งออกเป็น 2 ตัวคือ RXB ใช้รับข้อมูลจากภายนอกเข้ามาในไมโครคอนโทรลเลอร์ และTXB ใช้สำหรับส่งข้อมูลออกจากไมโครคอนโทรลเลอร์การอ่านเขียนข้อมูลจะทำกับรีจีสเตอร์ UDR โดยตรง
2) รีจีสเตอร์ UCSRA (USART Control and Status Register A) รีจีสเตอร์ควบคุมการทำงานและแสดงสถานะการทำงานของโมดูล USART ชุด A เกี่ยวข้องกับสถานะของการสื่อสารข้อมูล

3) รีจีสเตอร์ UCSRB (USART Control and Status Register B) รีจีสเตอร์ควบคุมการทำงานและแสดงสถานะการทำงานของโมดูล USART ชุด B เกี่ยวข้องกับบิตกำหนดอินเตอร์รัปต์และการกำหนดขนาดของข้อมูลแบบ 0 บิตข้อมูล (Data Bit)
4) รีจีสเตอร์ UCSRC (USART Control and Status Register C) รีจีสเตอร์ควบคุมการทำงานและแสดงสถานะการทำงานของโมดูล USART ชุด C เกี่ยวข้องกับการกำหนดอัตราบอดในการรับส่งข้อมูล
5) รีจีสเตอร์ UBRRL และ UBRRH (USART Baud Rate Registers) รีจีสเตอร์กำหนดอัตราบอด

2.8 อีอีพรอม (EEPROM:Ectrically Erasable Programmable Read-Only Memory)

หน่วยความจำรอมที่ผู้ใช้สามารถลบหรือแก้ไขหรือเขียนซ้ำข้อมูลที่บรรจุอยู่ภายในได้ และสามารถกระทำซ้ำได้หลายครั้ง โดยอาศัยแอพพลิเคชั่นที่ใช้กำลังไฟฟ้าสูงกว่าปกติ อีอีพรอมจะต่างจากอีพรอมตรงที่ไม่จำเป็นต้องถอดออกจากเครื่องคอมพิวเตอร์เพื่อทำการแก้ไขข้อมูล การลบข้อมูลในอีอีพรอมจะเป็นการลบข้อมูลทั้งหมด ไม่สามารถเลือกลบเฉพาะบางส่วนได้ อย่างไรก็ตามมันมีอายุการใช้งานจำกัดขึ้นอยู่กับจำนวนครั้งในการลบหรือแก้ไขข้อมูล เช่น 10 ครั้ง หรือ 100 ครั้ง รูปแบบพิเศษของอีอีพรอม คือ หน่วยความจำแฟลช (Flash Memory) ซึ่งใช้ระดับไฟปกติในเครื่องพีซีสำหรับการลบหรือเขียนหรือแก้ไขข้อมูล ซึ่งในที่นี้ใช้อีอีพรอม เบอร์ 24LC128



รูป 2.22 ไอซี 24LC128(128K I^2C CMOS Serial EEPROM)

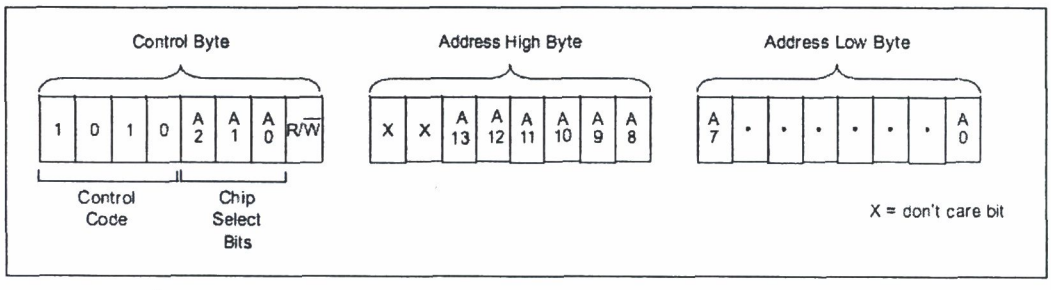
2.8.1 ขั้นตอนการติดต่อไอซี 24LC128

การติดต่อกับไอซี 24LC128 จะใช้การติดต่อแบบ I^2C เริ่มต้นโดยการสร้างสัญญาณติดต่อหรือที่เรียกว่า สภาวะเริ่มต้น (START Condition) จากนั้นตามด้วยแอดเดรสของไอซี 24LC128 (Control Byte) มีค่าเท่ากับ 0xA0 (10100000_2) จากนั้นจึงเริ่มต้นเข้าสู่กระบวนการอ่านหรือเขียนข้อมูลในตำแหน่งแอดเดรสของไอซี 24LC128 และหยุดการติดต่อด้วยการสร้างสัญญาณที่เรียกว่า สภาวะหยุด (STOP Condition) กระบวนการสร้างสัญญาณติดต่อจะเป็นหน้าที่ของ

อุปกรณ์ที่เรียกว่า มาสเตอร์ ในที่นี้คือ ไมโครคอนโทรลเลอร์ AVR รายละเอียดเกี่ยวกับการเขียน 24LC128 มีดังนี้

2.8.1.1 รูปแบบไบต์ควบคุมและแอดเดรสที่อ่าน

จากรูป 2.23 รายละเอียดของไบต์ควบคุม (Control Byte) และไบต์กำหนดแอดเดรส (Address High Byte/Low Byte) ที่จะอ่านของอุปกรณ์สเลฟ โดยไบต์ควบคุมจะประกอบไปด้วย 4 บิตบน คือ 1010_2 (CONTROL CODE) ซึ่งเป็นบิตคงที่จากทางผู้ผลิตชิพ และตำแหน่งบิตที่ใช้งานสามารถกำหนดได้อีก 3 บิต คือ A2,A1,A0 (CHIP SELECT BITS) สำหรับบิตสุดท้ายจะใช้เป็นบิตกำหนดการอ่าน/เขียนบัส I^2C (R/W) ไบต์ถัดมาคือไบต์กำหนดแอดเดรส ที่ใช้ในการอ้างถึงเพื่อไปอ่านหรือเขียนข้อมูลไอซี 24LC128 ประกอบด้วย 2 ไบต์ เพื่อให้สามารถอ้างอิงแอดเดรสได้ถึง 128 Kbit โดยบิตที่ 6 และ 7 ของ ADDRESS HIGH BYTE จะเป็นบิตที่ผู้ใช้งานไม่ต้องสนใจ (Don't Care bit) เพราะจะใช้งานเพียง 14 บิตเท่านั้นประกอบไปด้วยบิต A0 ถึงบิต A13

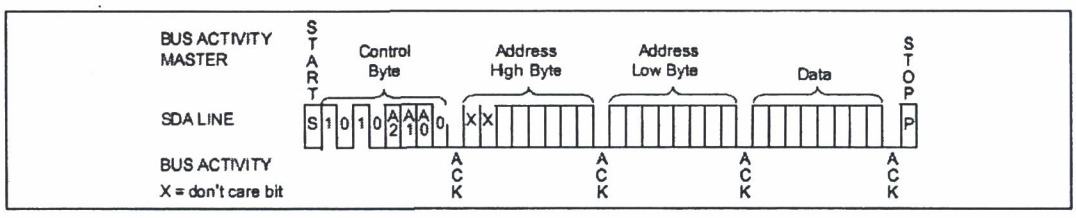


รูป 2.23 แอดเดรสของ Control Byte และ Address High Byte/Low Byte

2.8.1.2 กระบวนการเขียนแบบไบต์

จากรูป 2.24 ขั้นตอนการเขียนข้อมูลไปที่ไอซี 24LC128 อีอีพรอมแบบไบต์ (เขียนข้อมูลเพียง 1 ไบต์) เริ่มต้นด้วยกระบวนการสร้างสัญญาณเริ่มต้น (START) ตามด้วยไบต์ควบคุม (CONTROL BYTE) ตามด้วยตำแหน่งแอดเดรสที่จะเขียน (ADDRESS HIGH BYTE และ ADDRESS LOW BYTE) จากนั้นตามด้วยข้อมูลที่จะเขียน 1 ไบต์ (DATA) แล้วจบการเขียนข้อมูลด้วยสัญญาณหยุด (STOP) ในทุกๆ ขั้นตอนการเขียนข้อมูลจะกำหนดสัญญาณตอบรับ

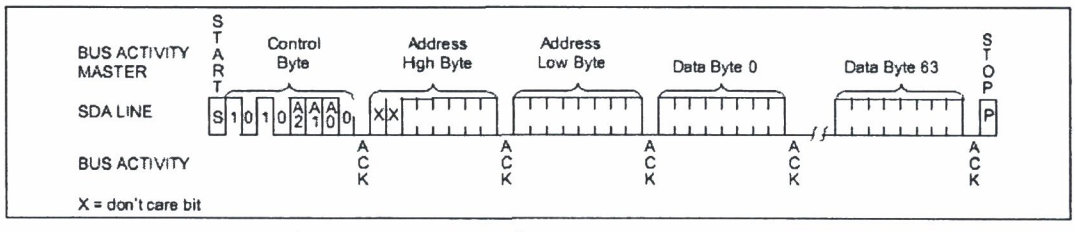
สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดงานวิจัย



รูป 2.24 กระบวนการเขียนแบบไบต์

2.8.1.3 กระบวนการเขียนแบบเพจ

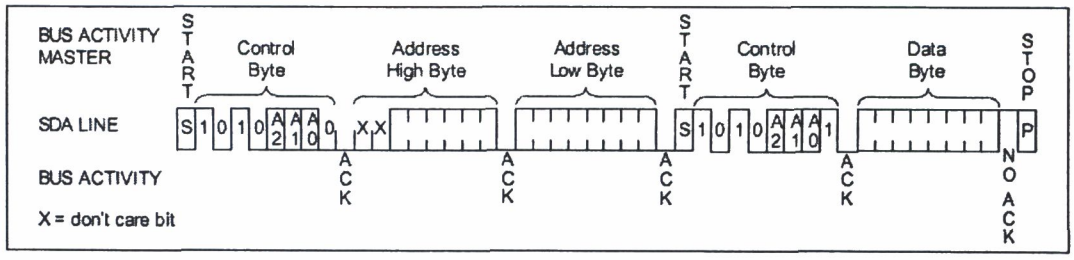
สำหรับขั้นตอนการเขียนข้อมูลไปที่ไอซี 24LC128 อีอีพรอม แบบเพจ(เขียนข้อมูลต่อเนื่อง) แสดงในรูป 2.25 โดยเริ่มต้นด้วยกระบวนการสร้างสัญญาณเริ่มต้น (START) ตามด้วยไบต์ควบคุม (CONTROL BYTE) ตามด้วยตำแหน่งแอดเดรสที่จะเขียน (ADDRESS HIGH BYTE และ ADDRESS LOW BYTE) จากนั้นตามด้วยข้อมูลที่จะเขียนไบต์แรก (DATA BYTE 0) จนถึงไบต์สุดท้าย (DATA BYTE 63) แล้วจบการเขียนข้อมูลด้วยสัญญาณหยุด(STOP) และในทุกๆ ขั้นตอนการเขียนข้อมูลจะกำหนดสัญญาณตอบรับ(ACK) ด้วย



รูป 2.25 กระบวนการเขียนแบบเพจ (Page Write)

2.8.1.4 กระบวนการอ่านแบบสุ่ม

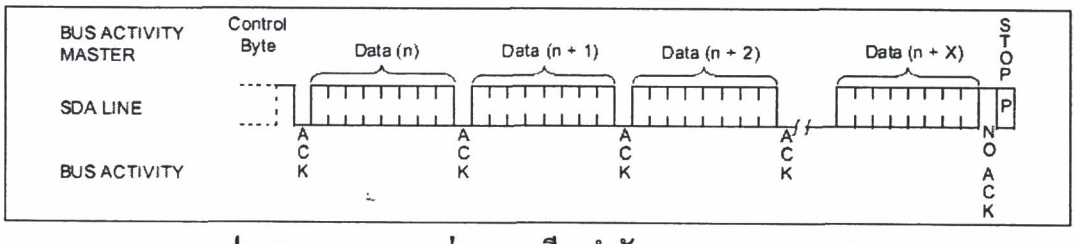
สำหรับขั้นตอนการอ่านข้อมูลไปที่ไอซี 24LC128 อีอีพรอม แบบสุ่ม (อ่านข้อมูลเพียง 1 ไบต์) แสดงตามรูป 2.26 เริ่มต้นด้วยกระบวนการสร้างสัญญาณเริ่มต้น (START) ตามด้วยไบต์ควบคุม (CONTROL BYTE) ตามด้วยแอดเดรสที่จะอ่าน (ADDRESS HIGH BYTE และ ADDRESS LOW BYTE) และเริ่มกระบวนการสร้างสัญญาณเริ่มต้นอีกครั้ง จากนั้นตามด้วยไบต์ควบคุม (เพื่อเปลี่ยนจากโหมดเขียนเป็นโหมดอ่าน สังเกตได้จากบิตต่ำสุดของไบต์ควบคุมจะเปลี่ยนจาก 0 เป็น 1) และอ่านข้อมูล 1 ไบต์ (DATA) แล้วจบการอ่านข้อมูลด้วยการสร้างสัญญาณหยุด (STOP) ในทุกๆ ขั้นตอนของการอ่านข้อมูลจะกำหนดสัญญาณตอบรับ (ACK) ด้วย สำหรับไบต์ข้อมูล อาจไม่ต้องการสัญญาณตอบรับก็ได้ (NO ACK) เนื่องจากจะเข้าสู่ขั้นตอนการส่งสัญญาณหยุดการอ่านข้อมูลแล้ว



รูป 2.26 กระบวนการอ่านแบบสุ่ม (Random Read)

2.8.1.5 กระบวนการอ่านแบบเรียงลำดับ

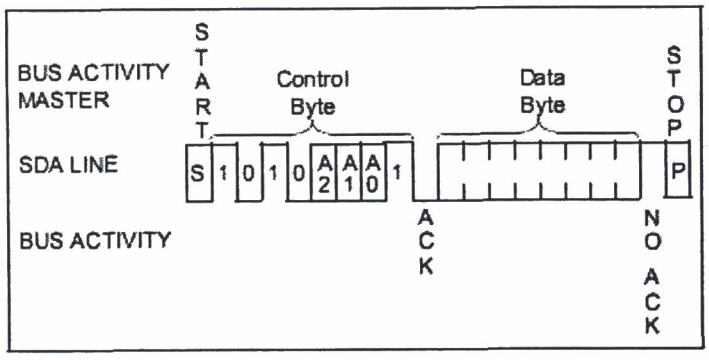
ในรูป 2.27 แสดงขั้นตอนการอ่านข้อมูลจากไอซี 24LC128 อีอีพรอมแบบเรียงลำดับ (อ่านข้อมูลต่อเนื่อง) เริ่มต้นด้วยกระบวนการสร้างสัญญาณเริ่มต้น (START) ตามด้วยไบต์ควบคุม (CONTROL BYTE) ตามด้วยแอดเดรสที่จะอ่าน (ADDRESS HIGH BYTE และ ADDRESS LOW BYTE) และเริ่มกระบวนการสร้างสัญญาณเริ่มต้นอีกครั้ง จากนั้นตามด้วยไบต์ควบคุม แล้วอ่านข้อมูลต่อเนื่องจนกว่าจะส่งสัญญาณหยุด (STOP) ในทุกๆ ขั้นตอนการเขียนข้อมูลจะกำหนดสัญญาณตอบรับ (ACK) ด้วย ไบต์ข้อมูลสุดท้าย อาจไม่ต้องการสัญญาณตอบรับก็ได้ (NO ACK) เนื่องจากจะเข้าสู่ขั้นตอนการส่งสัญญาณหยุดเพื่อหยุดการอ่านข้อมูลแล้ว



รูป 2.27 กระบวนการอ่านแบบเรียงลำดับ (Sequential Read)

2.8.1.6 กระบวนการอ่านข้อมูลแอดเดรสปัจจุบัน

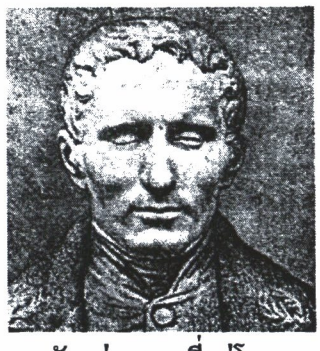
หากไม่ได้กำหนดตำแหน่งแอดเดรสที่จะอ่านข้อมูล กระบวนการอ่านข้อมูลจะมีรูปแบบดังรูป 2.28 คือ จะเริ่มด้วยกระบวนการสร้างสัญญาณเริ่มต้น (START) ตามด้วยไบต์ควบคุม (CONTROL BYTE) จากนั้นอ่านข้อมูลที่แอดเดรสตำแหน่งปัจจุบัน โดยจะไม่มีขั้นตอนการสร้างสัญญาณเริ่มต้น (START) ตามด้วยไบต์ควบคุม (CONTROL BYTE) ตามด้วยแอดเดรสที่จะอ่าน (ADDRESS HIGH BYTE และ ADDRESS LOW BYTE) ซึ่งต่างไปจากกระบวนการอ่านแบบสุ่มและกระบวนการอ่านแบบเรียงลำดับ



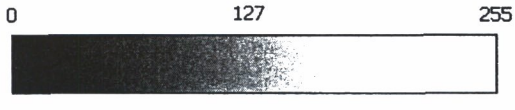
รูป 2.28 กระบวนการอ่านข้อมูลแอดเดรสปัจจุบัน (Current Address Read)

2.9 การแปลงภาพสีไปเป็นภาพขาวดำ

ในการแปลงภาพสีไปเป็นภาพขาวดำนั้น สามารถทำได้มากกว่า 1 วิธี ความแตกต่างของน้ำหนักของสีต่างๆ ทำให้เกิดผลกระทบในการจับภาพทั่วไปของกล้องถ่ายรูป วิธีการทั่วไป จะเน้นการใกล้เคียงกันของความสว่างของภาพขาวดำกับความสว่างของภาพสี



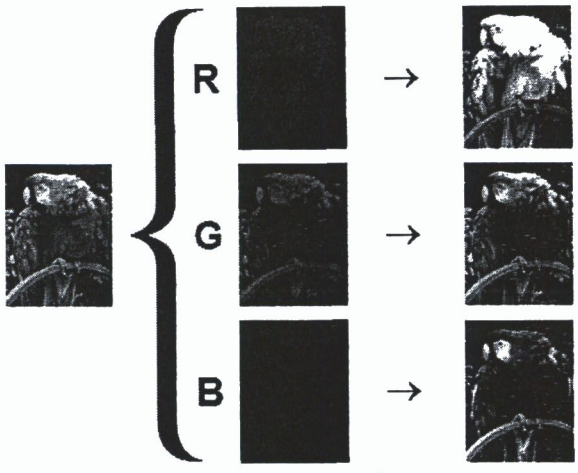
รูป 2.29 ตัวอย่างภาพที่อยู่โหมดขาวดำ



รูป 2.30 ระดับเฉดสีในโหมดภาพขาวดำ ตั้งแต่สีดำไปจนสีขาว

ในการแปลงสีใดๆเป็นสีขาว-ดำนั้น ในขั้นแรก จะทำการรับค่าสีของภาพนั้นๆเข้ามา โดยแบ่งเป็น 3 เฉดสี คือ แดง เขียว และฟ้า (RGB) หรือสีที่เป็นแม่สีทั้ง 3 โดยจะนำ 30% ของสีแดง, 59% ของสีเขียว และ 11%ของสีฟ้า (โดยน้ำหนักนี้ จะขึ้นอยู่กับสีที่เราต้องการให้เป็นสีหลักในการแปลง) และเมื่อนำค่าสีทั้ง 3 เข้าไปคูณกับอัตราแล้ว จึงนำทั้งหมดมารวมกันโดยเมื่อรวมกันก็จะครบ

100% โดยถ้าเทียบเฉดสีก็คือ จะมีเฉดสีทั้งหมด 255 สี และนำค่าสีที่ได้ ใส่คืนให้ค่าสีทั้ง 3 ซึ่ง ถ้าใน 3 ค่านั้น ถ้าหากเท่ากันหมด ก็จะแสดงออกมาในรูปแบบสีขาวดำ



รูป 2.31 ตัวอย่างการแปลงภาพสีเป็นภาพขาวดำ

2.10เครื่องมือที่ใช้พัฒนา

ภาษาC# คือ Visual C# เนื่องจากเป็นโปรแกรมที่เขียนในรูปแบบของอ็อปเจ็ค ทำให้เขียนง่าย มี tool ที่ใช้ในการทำงานหลากหลาย และสามารถติดต่อ Serial Port ได้ง่ายโดยผ่านฟังก์ชั่นภายใน มีหน้าตา Interface ที่สวยงาม และมีฟังก์ชันการทำงานที่ง่ายต่อการใช้งานของผู้ใช้งาน

เหตุผลสำหรับการเลือกใช้ภาษา C# ในการเขียนส่วนติดต่อผู้ใช้งานคือ ภาษา C# นั้น เป็นภาษาที่มีรูปร่างหน้าตาและโครงสร้างในแบบที่เรามักจะเรียกว่า "C-Style Language" หรือ ภาษาที่มีลักษณะคล้ายคลึงกับภาษา C ซึ่งแม้แต่ภาษาที่โปรแกรมเมอร์ชาวไทยคุ้นเคยกันดีอย่าง Java และ PHP นั้นก็จัดอยู่ในภาษากลุ่มนี้เช่นกัน นั่นก็เพราะว่า "C-Style" เป็นรูปแบบภาษาที่โปรแกรมเมอร์ส่วนใหญ่ที่มักมีพื้นฐานมาจากภาษา C คุ้นเคย แต่ก็อาจจะเป็นภาษาที่ดูแปลกตา สำหรับผู้ที่ไม่มีพื้นฐานการเขียนโปรแกรมมาก่อน หรือผู้ที่คุ้นเคยกับภาษาที่ค่อนข้างดูคล้ายกับภาษาพูดอย่าง Visual Basic ไปเลยก็เป็นได้ ดังนั้นถ้าคุณมีพื้นฐานจากภาษาในกลุ่ม C-Style อยู่ก่อน ก็อาจจะข้ามในส่วนของการแนะนำโครงสร้างภาษานี้และไปเริ่มอ่านในส่วนของการแนะนำฟีเจอร์เฉพาะของภาษา C#