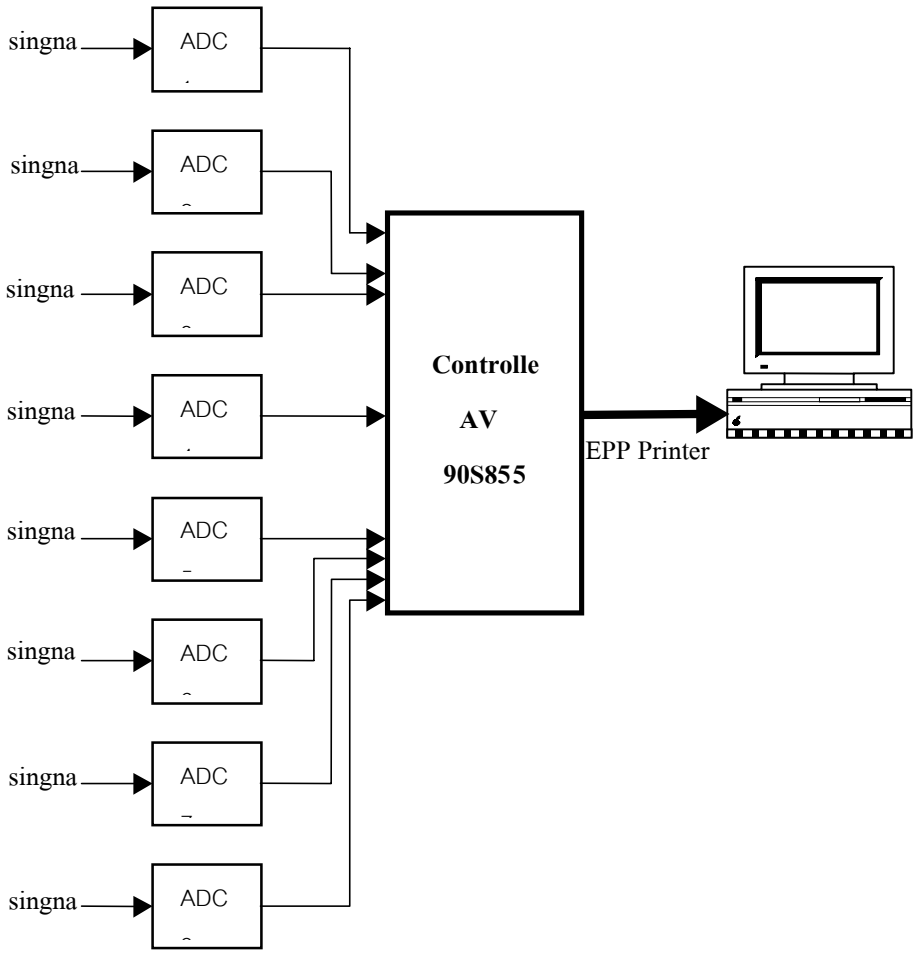
บทที่4

ตัวควบคุมการบันทึกเสียงเต้นหัวใจ

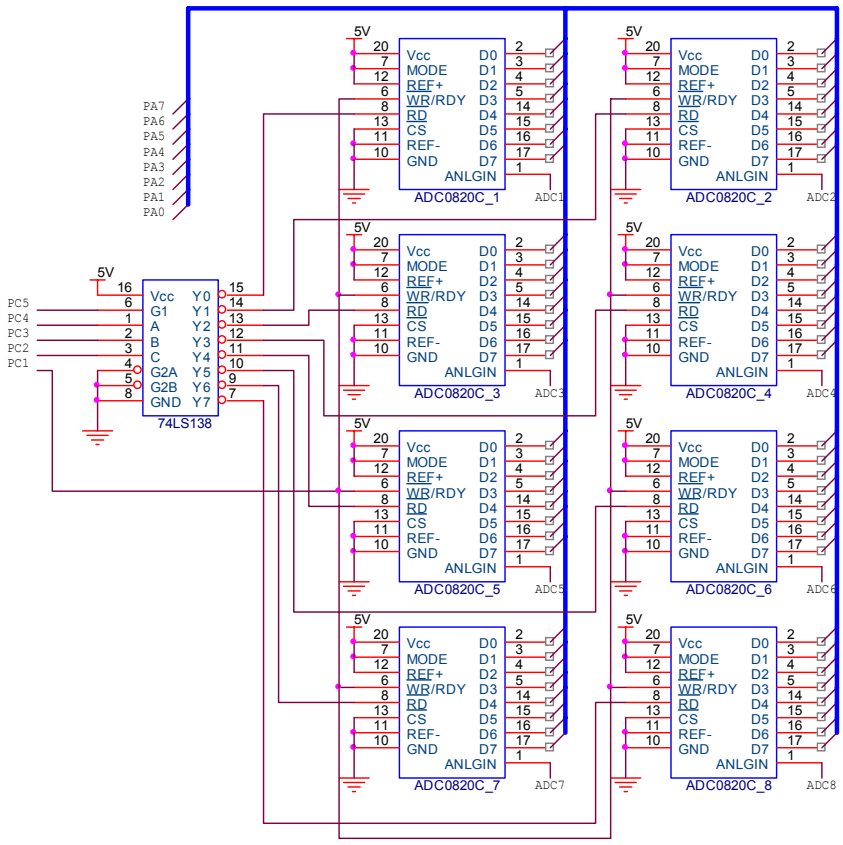
ในบทนี้จะกล่าวถึงการออกแบบตัวควบคุมการเก็บบันทึกเสียงเต้นหัวใจ ซึ่งควบคุมการรับส่งข้อมูลไปยังคอมพิวเตอร์ ตัวควบคุมจะประกอบด้วยส่วนหลักทั้งหมด 4 ส่วนคือ ส่วนควบคุมการแปลงสัญญาณแอนาลอกเป็นดิจิตอล ส่วนควบคุมการอ่านและส่งข้อมูล ส่วนการอินเตอร์เฟสกับพอร์ตขนานในโหมดการทำงานแบบ EPP และส่วนคอมพิวเตอร์ การทำงานของแต่ละส่วนหลักของตัวควบคุมดังภาพประกอบที่ 4-1



ภาพประกอบที่ 4-1 แสดงไดอะแกรมโดยรวมของตัวควบคุม

4.1 วงจรแปลงสัญญาณอานาลอกเป็นดิจิตอล(Analog to Digital converter,ADC)

ในส่วนนี้จะทำหน้าที่แปลงสัญญาณเสียงเต้นหัวใจซึ่งเป็นสัญญาณแอนาลอกให้เป็นสัญญาณดิจิตอล เพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ซึ่งอินเตอร์เฟสกับคอมพิวเตอร์เพื่อควบคุมการบันทึกข้อมูล วงจรแปลงสัญญาณแอนาลอกเป็นดิจิตอลได้เลือกใช้เบอร์ADC0820 ซึ่งมีเวลาในการแปลงสัญญาณ(Conversion time) ประมาณ 1.5 μs ขนาด 8 บิท



ภาพประกอบที่ 4-2 แสดงวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล

สัญญาณเสียงจะเข้ามาทางสัญญาณ Vin ซึ่งมีทั้งหมด 8 ช่องสัญญาณด้วยกันสำหรับขาสัญญาณเขียน(WR) ซึ่งเชื่อมต่อกับ PC1 และสัญญาณอ่าน(RD) ของแต่ละตัวจะเชื่อมต่อกับ IC ถอดรหัส(Decoders/Demultiplexers) เนื่องจากจำนวนพอร์ตของไมโครคอนโทรลเลอร์ไม่เพียงพอ

สำหรับการคอนโทรล โดยได้เลือกใช้ IC ถอดรหัสเบอร์ 74LS138 ซึ่งได้ทำการเชื่อมต่อเข้าที่ขา Y0-Y7 เข้ากับขาของตัว ADC ตัวที่ 1-7 ตามลำดับ โดยที่ขา A B C และขา G เชื่อมต่อกับขา PC2-PC5 ของไมโครคอนโทรลเลอร์ ซึ่งจะถูกควบคุมโดยไมโครคอนโทรลเลอร์ในการสั่งให้เริ่มแปลงสัญญาณ และสั่งอ่านข้อมูลเข้าทางขา PA0-PA7 ของไมโครคอนโทรลเลอร์

4.2 ตัวควบคุม(Controller)

ส่วนประกอบที่มีความสำคัญที่สุดในการควบคุมการบันทึกเสียงการเต้นของหัวใจคือ ตัวควบคุมซึ่งจะทำหน้าที่ในการควบคุมการแปลงสัญญาณอนาลอกเป็นดิจิตอลดังที่ได้กล่าวมาแล้ว ยังเป็นตัวสำคัญในกระบวนการรับส่งข้อมูล ซึ่งถูกออกแบบให้รับและส่งข้อมูลโดยใช้เทคนิค การมัลติเพล็กซิ่งทางเวลา และการควบคุมการสร้างสัญญาณให้พอร์ตขนานในโหมดการทำงานแบบ EPP ในงานวิจัยชิ้นนี้ได้เลือกใช้ตัวควบคุมไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ 90S8535 ซึ่งสามารถจะอธิบายถึงลักษณะโครงสร้างและการทำงานต่างๆ ดังต่อไปนี้

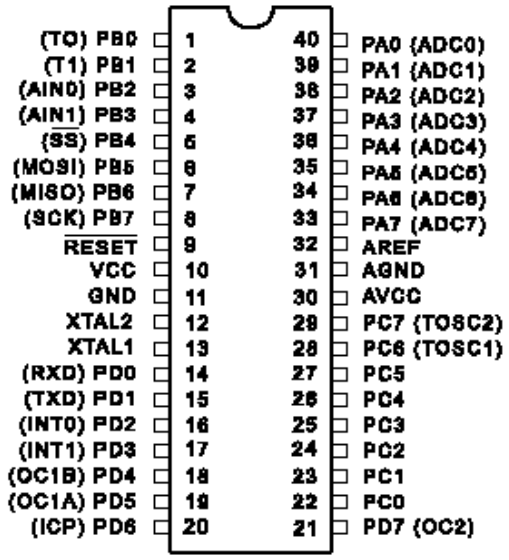
4.2.1 คุณสมบัติโครงสร้างและของตัวควบคุม

ตัวควบคุมที่ได้ออกแบบมีคุณสมบัติดังนี้

- สถาปัตยกรรมถูกออกแบบให้ใช้สถาปัตยกรรมแบบRISC(Reduce Instruction Set Computer)
- หน่วยความจำแบบ FLASH สำหรับบันทึก PROGRAM MEMORY ขนาด 8 Kbyte
- หน่วยความจำแบบ EEPROM สำหรับบันทึก DATA MEMORY ขนาด 512 Byte
- หน่วยความจำแบบ RAM ขนาด 512 Byte
- ระบบเปลี่ยนสัญญาณอนาลอกเป็นดิจิตอล ขนาด 10 บิท จำนวน 8 แชนแนล
- พอร์ตอินพุทและเอาท์พุท ขนาด 8 บิท จำนวน 4 พอร์ต
- ระบบการสื่อสารข้อมูลข้อมูลดิจิตอลแบบอะซิงโครนัส(SPI) 1 แชนแนล
- ความถี่สัญญาณนาฬิการ 0-8 MHz
- ระบบรีเซ็ตอัตโนมัติเมื่อเริ่มจ่ายกระแสไฟฟ้าเข้าไมโครคอนโทรลเลอร์(Power on Reset)
- ระบบกำเนิดความถี่สัญญาณแบบ PWM จำนวน 3 แชนแนล
- ระบบการป้องกันการก็อปปี้การ COPY ข้อมูลในหน่วยความจำ

- ระบบตรวจจับการทำงานผิดพลาดของ CPU
- ระบบการอินเตอร์รัพท์จากภายนอก
- Timer/Counter ขนาด 16 บิท 1 แชนแนล
- Timer/Counter ขนาด 8 บิท 2 แชนแนล

AVR 90S8535 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิท ที่มีสถาปัตยกรรมแบบ RISC(Reduce Instruction Set Computer) ซึ่งทำให้การประมวลผลมีความเร็ว 1 คำสั่ง/1 Clock ลักษณะโครงสร้างภายนอกและตำแหน่งขาดังภาพประกอบที่ 4-3

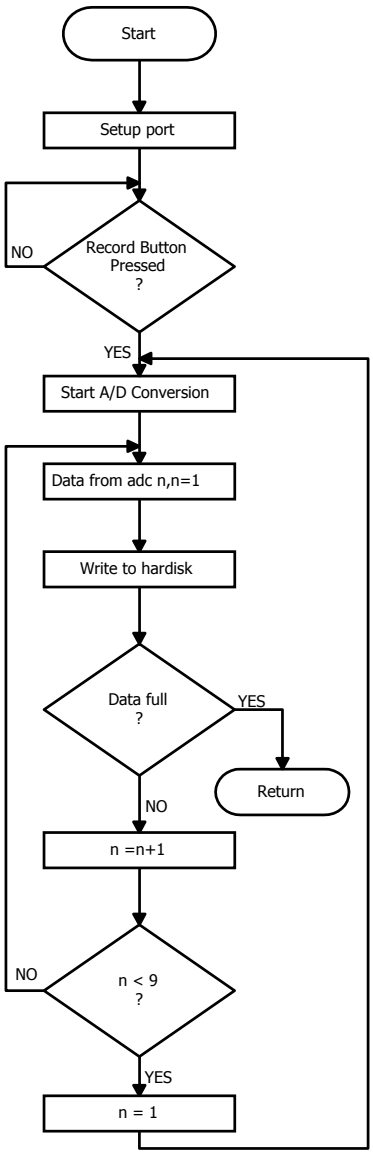


ภาพประกอบที่ 4-3 แสดงโครงสร้างภายนอกของ AVR 90S8535

โครงสร้างภายใน AVR จะมีหน่วยความจำสำหรับ Program Memory แบบ Flash ขนาด 8 kbyte มีหน่วยความจำแบบ RAM 512 Byte มีพอร์ตที่สามารถทำงานได้ 2 ทิศทาง จำนวน 4 พอร์ตมีจำนวน 32 เส้นสัญญาณ และระบบการแปลงสัญญาณอานาลอกเป็นดิจิตอล ขนาด 10 บิต จำนวน 8 ช่องสัญญาณ แต่เนื่องว่าในส่วนของการแปลงสัญญาณอานาลอกเป็นสัญญาณดิจิตอล ใช้เวลาค่อนข้างนานคือ ใช้เวลาสำหรับการแปลง(Conversion time) ประมาณ 65-250 μs และการสั่งให้ Convert จะเป็นแบบ Multi Channal คือไม่สามารถสั่งการ Convert พร้อมกันทั้ง 8 ช่องสัญญาณได้ ซึ่งจะทำให้อัตราการชักตัวอย่าง(Sampling rate) ค่อนข้างที่จะช้ามาก ดังนั้นผู้วิจัยจึงเลือกใช้ ADC ต่อภายนอกดังรายละเอียดที่กล่าวไปแล้วก่อนหน้านี้

4.2.2 การออกแบบโครงสร้างควบคุมการทำงาน

ในการออกแบบเขียนโปรแกรมเพื่อควบคุมการทำงานจะใช้โปรแกรมภาษาแอสแซมบลี้(Assembly Language Program) ในการควบคุมการทำงาน ซึ่งรายละเอียดเกี่ยวกับขั้นตอนการพัฒนางานออกแบบการเขียนโปรแกรมรวมทั้งรหัสต้นฉบับ(Source code) ของโปรแกรมได้แสดงในภาคผนวก ก และภาคผนวก ข ตามลำดับ โดยสามารถแสดงโฟล์วชาร์ตของการออกแบบตัวควบคุมดังนี้

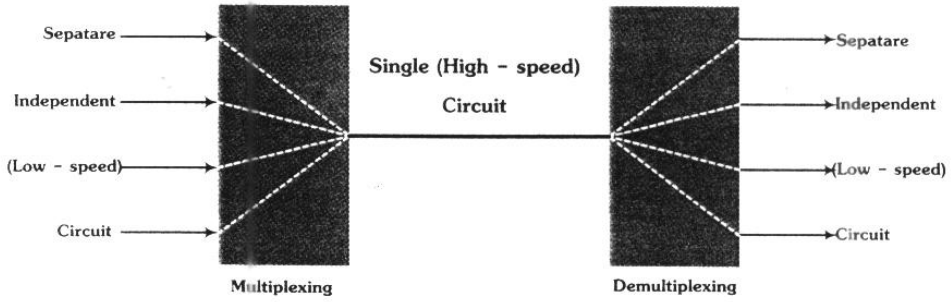


ภาพประกอบที่ 4-4 แสดงโฟล์วชาร์ตของโครงสร้างโปรแกรม

4.2.3 มัลติเพล็กเซอร์ (Multiplexr)

มัลติเพล็กเซอร์ เป็นอุปกรณ์สื่อสารข้อมูลตัวหนึ่งในงานวิจัยนี้คือตัวไมโครคอนโทรลเลอร์ ทำหน้าที่ในการรวมสัญญาณหลายๆ สัญญาณจากแหล่งข้อมูลหลายๆแหล่งเข้าไว้ด้วยกันเพื่อให้สามารถเดินทางไปในช่องทางการสื่อสารเพียงช่องทางเดียวได้ และที่ปลายทางก็จะมีดีมัลติเพล็กเซอร์ (Demultiplexer) ทำหน้าที่ในการแบ่งสัญญาณกลับไปเป็นหลายๆสัญญาณดังเดิม

หลักการทำงานของมัลติเพล็กเซอร์จะใช้เทคนิคที่เรียกว่ามัลติเพล็กซิ่ง (Multiplexing) ดังภาพประกอบ 4-5



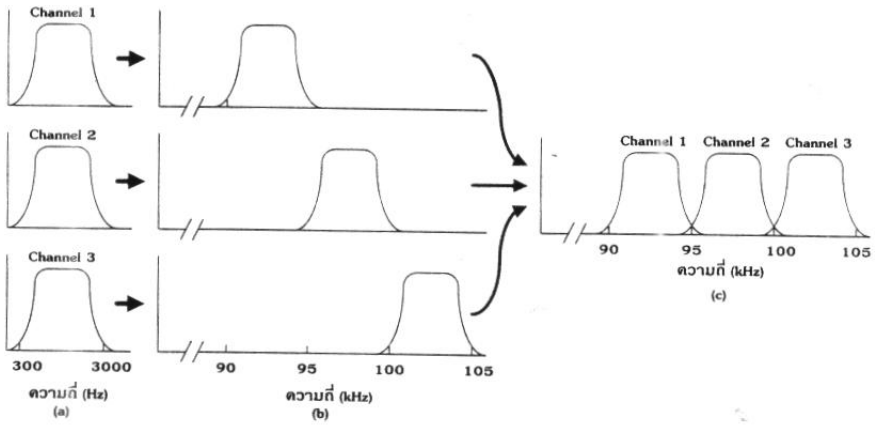
ภาพประกอบ 4-5 แสดงการทำมัลติเพล็กซิ่ง และดิมัลติเพล็กซิ่ง
(ที่มา: ฟ้าฟื้น เบญจคล, 2542)

วิธีการมัลติเพล็กสัญญาณที่สำคัญมีอยู่ 2 วิธี ดังนี้

- การมัลติเพล็กแบบแบ่งความถี่ (FDM:FrequencyDivisionMultiplexing)

FDM เป็นการรวมเอาสัญญาณที่มีความถี่แตกต่างกันมาไว้ด้วยกัน แล้วส่งออกพร้อมๆกัน ซึ่งเป็นแบบที่นิยมใช้กันมากในด้านวิทยุและโทรทัศน์ ระบบเอฟ. ดี.เอ็ม ไม่จำกัดเฉพาะสาย จะปรากฏกับการส่งกระจายเสียงทางวิทยุและโทรทัศน์ แต่จละอันมีคลื่นพาห์ของตนเอง ตลอดย่านความถี่เพื่อหลีกเลี่ยงการรบกวน

วงจร เอฟ.ดี.เอ็ม ใช้สัญาณต่ำและกำลังต่ำในการขยายสัญญาณ ทำให้สัญาณมัลติเพล็กสมบูรณ์ และส่งต่อไปยังภาคสุดท้าย เพื่อส่งออกอากาศ เอฟ.ดี.เอ็ม สามารถทำได้ทั้งความสูงของสัญญาณ(Amplitude) ความถี่(Frequency) และมุม(Phase) ดังภาพประกอบ 4-6

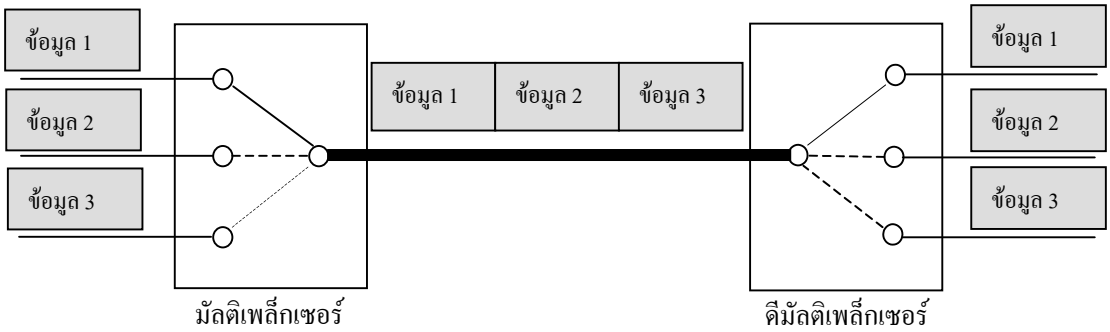


ภาพประกอบ 4-6 แสดงการมัลติเพล็กแบบแบ่งความถี่ (a) ข้อมูลก่อนการมัลติเพล็กอยู่ในช่วงความถี่เดียวกัน (b) ข้อมูลถูกจัดให้อยู่ในช่วงความถี่ที่แตกต่างกัน (c) ข้อมูลหลังการมัลติเพล็กถูกส่งไปพร้อมกัน

(ที่มา: ฟ้าฟื้น เบญจดล, 2542)

- การมัลติเพล็กแบบแบ่งตามเวลา (TDM: Time DivitionMultiplexing)

TDM เป็นการแบ่งช่วงเวลาในการส่งสัญญาณออกเป็นช่วงเล็กๆ แล้วส่งข้อมูลจากแต่ละแหล่งไปในแต่ละช่วงเวลานั้น เช่น ข้อมูลจากแหล่งที่ 1 ส่งไปในช่วงเวลา t_1 ข้อมูลจากแหล่งที่ 2 ส่งไปในช่วงเวลา t_2 เป็นต้น เพื่อให้สามารถส่งสัญญาณข้อมูลที่มาจากหลายๆ แหล่งออกไปได้ในสายสัญญาณเส้นเดียว ดังภาพประกอบ 4-7 วิธี TDM นี้ใช้ได้กับสัญญาณดิจิตอลเท่านั้น



ภาพประกอบ 4-7 แสดงการมัลติเพล็กแบบแบ่งตามเวลา

ในการออกแบบการรับส่งข้อมูลในการจำนวน 8 ช่องสัญญาณจะใช้วิธีการมัลติเพล็กซิ่ง วิธีการดังกล่าวเป็นกระบวนการส่งสัญญาณแต่ละชุดไปบนช่องสัญญาณเดียวกัน ในงานวิจัยนี้จะใช้วิธีการมัลติเพล็กซิ่งทางเวลา นั้นคือจะส่งข้อมูลแต่ละชุดในช่วงเวลาสั้นๆ ซึ่งกำหนดเวลาต่อเนื่องกันเป็นรอบๆ ในการส่งข้อมูลเป็นดิจิตอลข้อมูลจะจัดให้อยู่ในรูปแบบซีเรียลดาต้าเวิร์ด(Serial Data Words) คือหนึ่งช่องสัญญาณส่งข้อมูล 8 บิตและหยุดทำงานเพื่อส่งข้อมูลช่องสัญญาณถัดไป ส่งจนครบทุกช่องสัญญาณแล้ววนกลับมาส่งช่องสัญญาณที่หนึ่งใหม่

4.2.4 ความรู้เบื้องต้นเกี่ยวกับพอร์ตขนาน

การประมวลผลข้อมูลเพื่องานควบคุมนั้นสิ่งแรกจะต้องมีส่วนของสัญญาณอินพุทซึ่งอาจจะมาจากการตรวจจับต่างๆ ผ่านวงจรภาคหน้าเพื่อเปลี่ยนรูปสัญญาณอินพุทให้เหมาะสมกับการเชื่อมต่อกับคอมพิวเตอร์ เมื่อข้อมูลอินพุทถูกส่งเข้าสู่คอมพิวเตอร์แล้ว คอมพิวเตอร์จะทำการประมวลผลข้อมูลที่ได้มาเหล่านั้นให้อยู่ในรูปแบบที่เหมาะสมก่อนที่จะส่งออกไปยังภายนอกผ่านอุปกรณ์เอาต์พุท ซึ่งอาจจะเป็นการส่งออกไปยังจอภาพ หรือส่งออกไปยังจุดเชื่อมต่ออื่นๆ เพื่อควบคุมอุปกรณ์เอาท์พุทต่อไป

การเชื่อมต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอกทั้งส่วนของภาคอินพุทและภาคเอาท์พุท สามารถทำได้หลายวิธีดังนี้

- เชื่อมต่อผ่านทางการ์ดอินพุทเอาท์พุท ซึ่งใช้วิธีการเสียบหรือติดตั้งการ์ดลงในสล็อตภายในเครื่องคอมพิวเตอร์
- เชื่อมต่อผ่านทางพอร์ตอนุกรม
- เชื่อมต่อผ่านทางพอร์ตขนาน
- เชื่อมต่อผ่านระบบมาตราฐานอื่นๆ เช่น พอร์ต USB(Universal Serial Bus), พอร์ต SCSI หรือพอร์ต GAME เป็นต้น

4.2.4.1 เหตุผลการเลือกใช้พอร์ตขนาน

เมื่อเทียบกับการใช้งานการ์ดอินพุตเอาท์พุทที่ต้องติดตั้งอยู่ภายในเครื่องคอมพิวเตอร์แล้วพอร์ตขนานมีข้อได้เปรียบอยู่หลายประการดังนี้

- ในด้านความปลอดภัย การที่ต้องถอดฝาเครื่องคอมพิวเตอร์ออกมาเสียบการ์ดเชื่อมต่อลงในสล็อตของคอมพิวเตอร์ อาจจะทำให้เกิดความเสียหายกับส่วนอื่นๆ ของเครื่องคอมพิวเตอร์ได้ ถ้าผู้ใช้งานไม่มีความชำนาญหรือเกิดการต่อวงจรผิดพลาดในด้านการเข้ากันได้กับคอมพิวเตอร์ส่วนใหญ่ การเชื่อมต่อโดยใช้การ์ดที่เสียบลงในสล็อตไม่สามารถใช้กับคอมพิวเตอร์ ในปัจจุบันได้ทุกรุ่น ยกตัวอย่าง คอมพิวเตอร์โน้ตบุ๊ก จะไม่มีสล็อตเสียบแต่จะมีที่

เสียบการ์ด PCMCIA แทน ในขณะที่พอร์ตขนานจะมีการติดตั้งอยู่ภายในคอมพิวเตอร์ทุกเครื่องทั้งนี้เพื่อใช้ในการติดต่อกับเครื่องพิมพ์

- ข้อจำกัดด้านพื้นที่ คอมพิวเตอร์บางเครื่องมีการเสียบการ์ดเชื่อมต่อตัวอื่นๆ อยู่แล้ว อาทิ การ์ดเสียง การ์ดโมเด็ม จนไม่มีสล็อตเหลือพอสำหรับการเสียบการ์ดเชื่อมต่อเพิ่มเติม

- ความสะดวกในการใช้งาน การเชื่อมต่อพอร์ตขนานสามารถทำได้ง่ายๆ เพียงต่อสายสำหรับเชื่อมต่อเข้ากับคอนเน็กเตอร์ DB25 ของพอร์ตขนาน

- ความเร็วในการสื่อสารข้อมูลกับพอร์ตขนาน มีความเร็วเท่ากับการติดต่อกับระบบบัสโดยตรง และมีความเร็วมากกว่าการติดต่อผ่านทางพอร์ตอนุกรม

- อะไหล่และชิ้นส่วนประกอบ คอนเน็กเตอร์และสายเชื่อมต่อผ่านทางพอร์ตขนาน หาได้ง่ายและราคาไม่แพง หรือจะสร้างขึ้นเองก็สามรถทำได้ง่าย

จากคุณสมบัติดังที่ได้กล่าวมาแล้วนั้นทำให้พอร์ตขนานเหมาะอย่างยิ่งที่จะนำมาใช้ในการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกเพื่อควบคุมหรือรับสัญญาณข้อมูล นอกจากนั้นหากนำคุณสมบัติของการเขียนโปรแกรมง่ายๆ ผ่านระบบปฏิบัติการวินโดวส์ ก็สามารถสร้างระบบการเชื่อมต่อที่สมบูรณ์และใช้งานง่ายได้ไม่ยาก

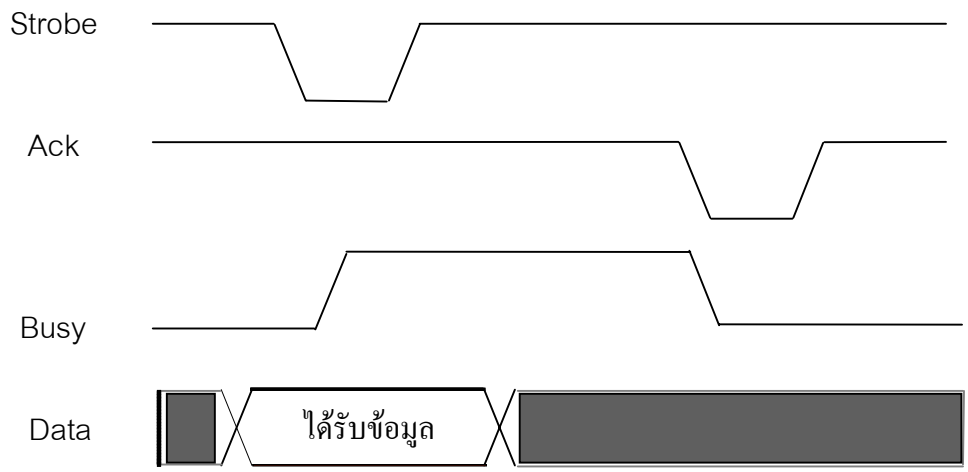
4.2.4.2 ลักษณะทางกายภาพ

พอร์ตขนาน(Palallel port) สาเหตุที่มีชื่อนี้ เนื่องจากการถ่ายทอดข้อมูลของพอร์ตนี้เป็นแบบขนาน สำหรับชื่อเรียกอีกชื่อหนึ่งของพอร์ตขนานคือ พอร์ตเครื่องพิมพ์ (Printer port) เนื่องจากพอร์ตนี้ใช้สำหรับต่อเครื่องพิมพ์นั่นเอง

ด้วยการถ่ายทอดข้อมูลแบบนี้เอง ทำให้พอร์ตขนานมีอัตราการถ่ายทอดข้อมูลสูงกว่าการถ่ายทอดข้อมูลแบบอนุกรมประมาณ 8-10 เท่า และการประมวลผลข้อมูลส่วนใหญ่จะมีขนาด 8 บิต ดังนั้นพอร์ตขนานจึงสามารถรองรับการถ่ายทอดข้อมูล 8 บิตได้โดยไม่ต้องต่อส่วนเพิ่มเติมใดๆ

เพื่อให้เข้าใจถึงการนำเอาพอร์ตขนานไปใช้งาน ก่อนอื่นต้องมาทำความเข้าใจก่อนว่า ปกตินั้นการสั่งพิมพ์งานจากคอมพิวเตอร์ไปยังพอร์ตขนานนั้นมีรูปแบบการทำงานภายในอย่างไร ในภาพประกอบ 4-8 แสดงไดอะแกรมเวลาของการติดต่อระหว่างพอร์ตขนานกับเครื่องพิมพ์ ซึ่งจะเห็นว่ามีสัญญาณที่ใช้งานจริงๆ มีไม่มาก เริ่มจากสัญญาณพอร์ต Data ถูกส่งออกไปยังเครื่องพิมพ์ พร้อมทั้งส่งสัญญาณ Strobe ออกไปด้วย เพื่อให้เครื่องพิมพ์รับรู้ว่ามีการส่งข้อมูลมาที่ขา Data แล้ว จากนั้นคอมพิวเตอร์จะต้องรอตอบกลับจากเครื่องพิมพ์ นั่นคือเครื่องพิมพ์จะสร้างสัญญาณ Busy หรือเพื่อบอกว่าเครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูลใหม่ จนกระทั่งเมื่อเครื่อง

พิมพ์พร้อม เครื่องพิมพ์จะสร้างสัญญาณ ACK ส่งไปยังคอมพิวเตอร์เพื่อแจ้งว่า พร้อมที่จะรับข้อมูลใหม่แล้ว



ภาพประกอบ 4-8 แสดงไดอะแกรมเวลาของการส่งข้อมูลไปยังเครื่องพิมพ์
(ที่มา: สุพจน์ แซ่เตีย, 2542)

สัญญาณข้อมูลขนาด 8 บิท, สัญญาณ Strobe และสัญญาณ ACK (acknowledge) เป็นสัญญาณที่สำคัญในการส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องพิมพ์ นอกจากสัญญาณทั้งสามแล้วส่วนใหญ่การติดต่อกับเครื่องพิมพ์ยังต้องมีสัญญาณอื่นๆร่วมด้วย เนื่องจากเครื่องพิมพ์ต้องทำหน้าที่ถึง 3 อย่างด้วยกันคือ รับข้อมูลจากคอมพิวเตอร์, พิมพ์ข้อมูลที่รับเข้ามาและตอบสนองต่อการใช้งานของผู้ใช้ เช่น การเปลี่ยนฟอนต์ เป็นต้น บางครั้งอาจเกิดเหตุการที่ไม่ปกติ เช่น บัฟเฟอร์สำหรับรับข้อมูลเต็ม(เนื่องจากเครื่องพิมพ์เป็นอุปกรณ์ที่ทำงานทางกลย่อมทำงานได้ช้ากว่าการส่งข้อมูลของเครื่องคอมพิวเตอร์) เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์ว่าให้หยุดส่งข้อมูลชั่วคราว เนื่องจากไม่สามารถรับข้อมูลมากว่านี้ได้แล้ว สัญญาณที่ส่งจากเครื่องพิมพ์ไปยังคอมพิวเตอร์คือ สัญญาณ Busy และเมื่อเครื่องพิมพ์เกิดข้อผิดพลาด เช่น กระดาษติดเครื่องพิมพ์เครื่องพิมพ์จะต้องแจ้งไปยังคอมพิวเตอร์เช่นกัน โดยสัญญาณที่แจ้งไปยังคอมพิวเตอร์เรียกว่าสัญญาณ Error นอกจากนี้เมื่อคอมพิวเตอร์ต้องการรีเซ็ตเครื่องพิมพ์ คอมพิวเตอร์จะส่งสัญญาณ Reset ไปยังเครื่องพิมพ์เพื่อรีเซ็ตเครื่องพิมพ์ด้วย สามารถสรุปขาสัญญาณที่จำเป็นสำหรับการติดต่อดังตาราง 4-1

สัญญาณ	หน้าที่การทำงาน	ทิศทาง
ข้อมูล 8 บิท	ข้อมูลที่ส่งจากคอมพิวเตอร์ไปยังเครื่องพิมพ์	คอมพิวเตอร์
Strobe	แจ้งเครื่องพิมพ์ถึงข้อมูลที่ส่งมาใหม่	คอมพิวเตอร์
Acknowledge	เครื่องพิมพ์แจ้งมายังคอมพิวเตอร์ว่าได้รับข้อมูลแล้ว	คอมพิวเตอร์
Busy	แจ้งสถานะว่าเครื่องพิมพ์ไม่ว่างที่จะรับข้อมูลใหม่	คอมพิวเตอร์
Eror	แจ้งสถานะว่าเครื่องพิมพ์เกิดข้อผิดพลาด	คอมพิวเตอร์
Reset	รีเซตเครื่องพิมพ์	คอมพิวเตอร์

ตาราง 4-1 แสดงสัญญาณสำคัญๆ ของพอร์ตขนานที่ใช้ติดต่อกับเครื่องพิมพ์
(ที่มา: สุพจน์ แซ่เตีย, 2542)

จากตาราง 4-1 จะเห็นได้ว่าพอร์ตขนานของคอมพิวเตอร์ยังแยกออกเป็นอีก 3 พอร์ตได้แก่ พอร์ตเอาท์พุทที่ทำหน้าที่ส่งข้อมูลจากเครื่องคอมพิวเตอร์ไปยังเครื่องพิมพ์ พอร์ตเอาท์พุทสำหรับสัญญาณ Strobe และ Reset พอร์ตอินพุทสำหรับการอ่านค่าสัญญาณ Acknowledge, Busy และสัญญาณ Eror จากเครื่องพิมพ์

โดยปกติพอร์ตขนานออกแบบมาให้มีสายสัญญาณเหล่านั้นจะมีรีจีสเตอร์ 3 ตัวควบคุมการทำงานดังนี้

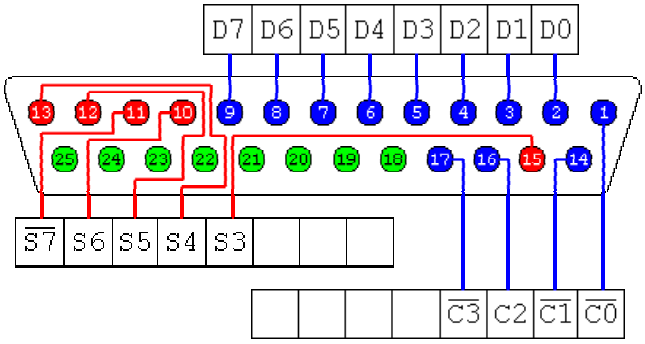
- พอร์ตเอาท์พุทสำหรับสัญญาณข้อมูล 8 เส้น มีรีจีสเตอร์ Data ควบคุม

-พอร์ตอินพุทสำหรับการอ่านค่าสถานะต่างๆ จากภายนอกมีอยู่ด้วยกัน 5 เส้น ใช้รีจีสเตอร์ Status ในการควบคุม

-พอร์ตเอาท์พุทสำหรับส่งสัญญาณควบคุมไปยังอุปกรณ์ภายนอกมีอยู่ด้วยกัน 4 เส้น ใช้รีจีสเตอร์ Control ในการควบคุม

สำหรับคอมพิวเตอร์ส่วนใหญ่ในปัจจุบันพอร์ตขนานจะมีมาพร้อมกับเมนบอร์ด ไม่จำเป็นต้องใช้การ์ดเสียบเพิ่มเติมเหมือนในอดีต พร้อมทั้งมีฟังก์ชันการทำงานที่ซับซ้อนขึ้น แต่ยังคงสนับสนุนการทำงานของพอร์ตขนานในรูปแบบมาตราฐาน(SPP) อยู่

ลักษณะทางกายภาพของพอร์ตขนานตัวเมีย ดังภาพประกอบ 4-9

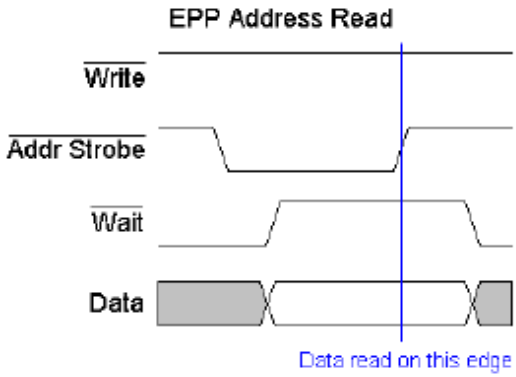


ภาพประกอบที่ 4-9 ลักษณะทางกายภาพของพอร์ตขนานตัวเมีย

(ที่มา: Interfacing to the IBM-PC Parallel Printer Port, 1998)

4.2.5 ส่วนควบคุมการสร้างสัญญาณให้พอร์ตขนานในโหมดการทำงานแบบ EPP(EPP Singnal Control)

เนื่องจากตัวควบคุมจะรอรับคำสั่งการเริ่มต้นทำงานจากซอฟต์แวร์ผ่านพอร์ตขนานในโหมดการทำงาน EPP โดยใช้สัญญาณ Address Strobe ดังนั้นจำเป็นจะต้องสร้างสัญญาณ Wait จากภายนอกเป็นสัญญาณตอบรับเพื่อเป็นไปตามข้อตกลง(Hand shaking) ของพอร์ตขนานในโหมดการทำงานแบบ EPP ซึ่งรายละเอียดทั้งหมดเกี่ยวกับพอร์ตขนานจะกล่าวในหัวข้อถัดไป ดังนั้นในส่วนนี้ตัวควบคุมจะทำหน้าที่สร้างสัญญาณ Wait เพื่อตอบรับในรอบการอ่านซึ่งจะประกอบไปด้วย Write และ Address Strobe จากพอร์ตขนานดังภาพประกอบที่ 4-10



ภาพประกอบที่ 4-10 แสดงสัญญาณ Wait เมื่อมีสัญญาณ Write และ Address Strobe

(ที่มา: Craig Peacock, Interfacing the Enhanced Palalell Port version 1.0)

4.3 การอินเตอร์เฟสพอร์ตขนานในโหมดการทำงานแบบ EPP

ปัจจุบันการใช้พอร์ตขนานของเครื่องคอมพิวเตอร์ในการเชื่อมต่อกับอุปกรณ์ภายนอกที่ไม่ใช่อุปกรณ์รอบนอกจำพวก เครื่องพิมพ์ พล็อตเตอร์ กำลังเป็นที่นิยมมากขึ้นเนื่องจากเครื่องคอมพิวเตอร์สมัยใหม่มีขีดความสามารถที่สูงขึ้น มีอัตราการถ่ายโอน(Tranfer rate) ข้อมูลที่เร็วมากขึ้น ในขณะที่ขาสัญญาณข้อมูลของพอร์ตขนานรุ่นใหม่สามารถถ่ายโอนข้อมูลได้สองทิศทาง ทำให้มีการนำพอร์ตขนานมาใช้ประโยชน์มากขึ้น

สำหรับงานวิจัยนี้เลือกการอินเตอร์เฟสพอร์ตขนานในโหมดการทำงานแบบ EPP ซึ่งมีรายละเอียดดังต่อไปนี้

โหมดการทำงานแบบ EPP เป็นโหมดการทำงานแบบหนึ่งของพอร์ตขนานซึ่งถูกออกแบบโดยความร่วมมือกันของบริษัท Intel, Xircom และ Zenith Data Systems เริ่มแรกโหมดการทำงานแบบ EPP จะถูกกำหนดเป็นมาตราฐาน 1.7 (EPP 1.7 Standard) ต่อมาในปี 1994 โหมดการทำงานแบบ EPP ก็ได้ถูกรวมในมาตรฐานของ IEEE 1284 ดังนั้นโหมดการทำงานแบบ EPP จึงมี 2 มาตรฐานคือ โหมดการทำงานแบบ EPP มาตรฐาน 1.7(EPP 1.7 Standard) ต่อมาในปี 1994 โหมดการทำงานแบบ EPP ก็ได้ถูกรวมในมาตรฐาน 1.7 และโหมดการทำงานแบบ EPP มาตรฐาน 1.9(EPP 1.9 Standard) ซึ่งมีข้อแตกต่างที่ส่งผลกระทบต่อการทำงานของอุปกรณ์รอบนอก โดยปกติโหมดการทำงานแบบ EPP จะมีอัตราการถ่ายโอนข้อมูลประมาณ 500 กิโลไบต์ต่อวินาที(kbyte/s) ถึง 2 เมกกะไบต์ต่อวินาที(Mbyte/s) ซึ่งสามารถทำได้โดยฮาร์ดแวร์ที่มีอยู่ในพอร์ตขนานเพื่อการ

Handshaking หรือการ Strobing มากกว่าจะใช้ซอฟต์แวร์เป็นผู้ทำเหมือนในกรณีโหมดการทำงานแบบ Centronics

4.3.1 คุณสมบัติทางฮาร์ดแวร์ของโหมดการทำงานแบบ EPP

พอร์ตแบบ EPP มีไบต์ข้อมูลแบบ 2 ทิศทาง และยังสามารถอ่านหรือเขียนข้อมูล 1 ไบต์ได้ในรอบการทำงานเดียว(one cycle) ของบัสขยายระบบแบบ ISA (ISA expansion bus) หรือใช้เวลาเพียง 1 ไมโครวินาที รวมการทำแฮนด์เช็ค (handshanking) ด้วย ซึ่งเร็วกว่าพอร์ตแบบ SPP หรือ PS/2 ซึ่งต้องใช้ 4 รอบการทำงาน พอร์ตการทำงานแบบ EPP สามารถกลับทิศทางการสื่อสารได้อย่างรวดเร็วซึ่งเป็นประโยชน์มากเมื่อใช้กับตัวอ่านเทปหรือดิสก์และอุปกรณ์อื่นๆ ที่มีการถ่ายเทข้อมูลแบบ 2 ทิศทาง และพอร์ต EPP บางแบบยังสามารถเรียนแบบการทำงานของพอร์ตแบบ PS/2 ได้ด้วยสำหรับชื่อและหน้าที่ของสัญญาณในโหมดการทำงานแบบ EPP โดยเปรียบเทียบกับโหมดการทำงานแบบมาตรฐาน(Standard Parallel Port, SPP) แสดงดังตาราง 4-2

ขา(Pin)	SPP Signal	EPP Signal	In/Out	หน้าที่(Funtion)
1	Strobe	Write	Out	ลอจิก “0” แสดงโหมดการเขียน ลอจิก “1” แสดงโหมดการอ่าน
2-9	Data 0-7	Data0-7	In-Out	บัสข้อมูลแบบ 2 ทิศทาง(Bidirection)
10	Ack	Interrupt	In	เส้นอินเตอร์รัปต์ โดยเกิดอินเตอร์รัปต์ที่ขอบขาขึ้นของสัญญาณ (Rising edge)
11	Busy	Wait	In	ใช้สำหรับการทำ Handshaking โดยรอบการทำงานของโหมด EPP จะเริ่มเมื่อสัญญาณเป็นลอจิอ “0” และสิ้นสุดเมื่อสัญญาณเป็นลอจิก “1”
12	Paper Out/End	Spare	In	สำรองไว้ (ไม่ใช้ในการ Handshaking)
13	Select	Spare	In	สำรองไว้ (ไม่ใช้ในการ Handshaking)
14	Auto Linefeed	Data Strobe	Out	ลอจิก “0” แสดงว่าเกิดการโอนย้ายข้อมูล
15	Error/Fault	Spare	In	สำรองไว้ (ไม่ใช้ในการ Handshaking)
16	Initialize	Reset	Out	ลอจิก “0” แสดงว่ารีเซ็ต
17	Select Printer	Address Strobe	Out	จิก “0” แสดงว่าเกิดการโอนย้ายแอดเดรส
18-25	Ground	Ground	Gnd	กราวด์

ตาราง 4-2 แสดงชื่อและหน้าที่ของสัญญาณในโหมดการทำงานแบบ EPP

(ที่มา : Craig Peacocock, Interfacing the Enhanced Parallel Port version 1.0)

หมายเหตุ : สัญญาณ Paper out, Select printer และ Error ไม่ได้ใช้ในการทำ Handshaking ดังนั้นผู้ใช้จึงสามารถใช้งานได้อย่างอิสระ ส่วนสถานะของสัญญาณเหล่านี้สามารถดูได้ที่ริสจิสเตอร์หรือพอร์ตแสดงสถานะ (Status port) ของโหมดการทำงานแบบมาตราฐาน

4.3.2 พอร์ตที่ใช้ในการโปรแกรมพอร์ตขนานในโหมดการทำงานแบบ EPP

โหมดการทำงานแบบ EPP จะมีพอร์ตที่เพิ่มเติมจากโหมดการทำงานแบบมาตรฐานดังแสดงในตาราง 4-3

แอดเดรส (Address)	ชื่อพอร์ต (Port name)	อ่าน/เขียน(Read/Write)
Base address + 0	พอร์ตข้อมูล (Data port) (SPP)	Write
Base address + 1	พอร์ตสถานะ (Status port) (SPP)	Read
Base address + 2	พอร์ตควบคุม (Control port) (SPP)	Write
Base address + 3	พอร์ตแอดเดรส (Address port) (EPP)	Read/Write
Base address + 4	พอร์ตข้อมูล (Data port) (EPP)	Read/Write
Base address + 5	ไม่ระบุ (16/32 bit Transfer)	-
Base address + 6	ไม่ระบุ (16/32 bit Transfer)	-
Base address + 7	ไม่ระบุ (16/32 bit Transfer)	-

ตาราง 4-3 แสดงพอร์ตที่ใช้ในการโปรแกรมพอร์ตขนานในโหมดการทำงานแบบ EPP

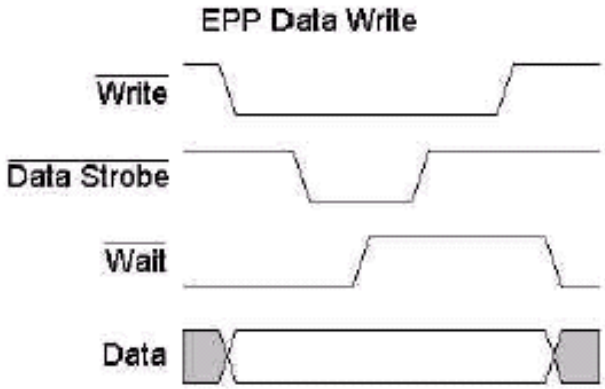
(ที่มา : Craig Peacocock, Interfacing the Enhanced Parallel Port version 1.0)

หมายเหตุ : Base address เป็นแอดเดรสเริ่มต้นสำหรับการติดต่อกับพอร์ตขนาน โดยปกติมักอยู่ที่แอดเดรส 0x378

4.3.3 หลักการโต้ตอบ(Handshaking) ของโหมดการทำงานแบบ EPP

การใช้งานพอร์ตขนานในโหมดการทำงานแบบ EPP เพื่อเคลื่อนย้ายข้อมูลนั้นต้องปฏิบัติตามหลักการ Handshaking ซึ่งต้องการเพียงแค่ให้ฮาร์ดแวร์เป็นตัวจัดการ โดยสัญญาณที่ใช้ในการทำ Handshaking จะมี 3 สัญญาณคือ Write, Data strobe หรือ Address strobe และ Wait ซึ่งการโอนย้ายนี้จะแบ่งรอบการทำงานออกเป็น 2 ลักษณะคือ รอบการเขียนข้อมูล(Data) หรือแอดเดรส(Address) และรอบการอ่านข้อมูลหรือแอดเดรส ในการนำไปประยุกต์ใช้งานผู้ใช้งานจะต้องทราบรายละเอียดต่างๆ เกี่ยวกับวิธีการติดต่อสื่อสารข้อมูล ไม่ว่าจะเป็นรอบการเขียนหรือรอบการอ่านข้อมูล ซึ่งวิธีการทำ Handshaking ดังรายละเอียดต่อไปนี้

4.3.3.1 รอบการเขียนข้อมูล(Data write cycle)



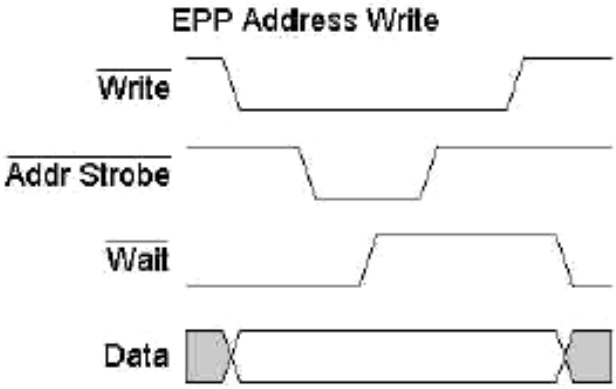
ภาพประกอบ 4-11 แสดงลักษณะการโต้ตอบของรอบการเขียนข้อมูล

(ที่มา : Craig Peacocock, Interfacing the Enhanced Parallel Port version 1.0)

จากภาพประกอบ 4-11 แสดงรายละเอียดดังนี้

- โปรแกรมเขียนไปที่พอร์ตข้อมูลที่ Base address+4
- สัญญาณ Write จะมีค่าเป็นลอจิก 0 (แสดงว่าอยู่ในโหมดการเขียน)
- ข้อมูลพร้อมอยู่ที่เส้นสัญญาณข้อมูลทั้ง 7 เส้น (Data line 0-7)
- สัญญาณ Data strobe จะถูกส่งออกมา (เป็นลอจิก 0) ถ้าสัญญาณ Wait เป็นลอจิก 0 (เริ่มรอบการเขียนข้อมูล)
- Host จะรอการตอบรับ (Acknowledgement) โดยรอจนกว่าสัญญาณ Wait จะเป็นลอจิก 1 (เพื่อหยุดรอบการทำงาน)
- สัญญาณ Data strobe จะกลับเป็นลอจิก 1
- สิ้นสุดรอบการเขียนข้อมูล

4.3.3.2 รอบการเขียนแอดเดรส(Address write cycle)



ภาพประกอบ 4-12 แสดงลักษณะการโต้ตอบของรอบการเขียนแอดเดรส

(ที่มา : Craig Peacocock, Interfacing the Enhanced Parallel Port version 1.0)

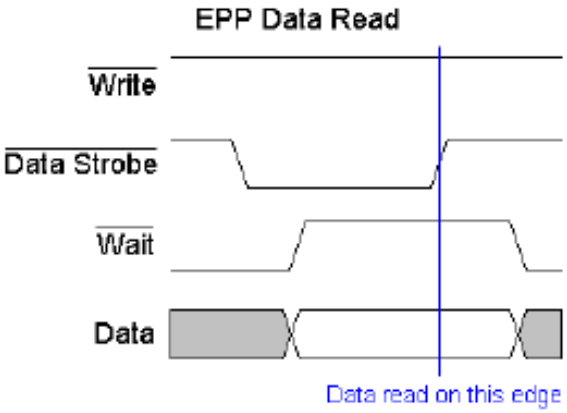
จากภาพประกอบ 4-12 แสดงรายละเอียดได้ดังนี้

- โปรแกรมเขียนไปที่พอร์ตข้อมูลที่ Base address+4
- สัญญาณ Write จะมีค่าเป็นลอจิก 0 (แสดงว่าอยู่ในโหมดการเขียน)
- แอดเดรสพร้อมอยู่ที่เส้นสัญญาณข้อมูลทั้ง 7 เส้น (Data line 0-7)
- สัญญาณ Address strobe จะถูกส่งออกมา (เป็นลอจิก 0) ถ้าสัญญาณ Waitเป็นลอจิก 0 (เริ่มรอบการเขียนแอดเดรส)

-Host จะรอการตอบรับ (Acknowledgement) โดยรอจนกว่าสัญญาณ Wait จะเป็นลอจิก 1 (เพื่อหยุดรอบการทำงาน)

- สัญญาณ Address strobe จะกลับเป็นลอจิก 1
- สิ้นสุดรอบการเขียนแอดเดรส

4.3.3.3 รอบการอ่านข้อมูล(Data read cycle)



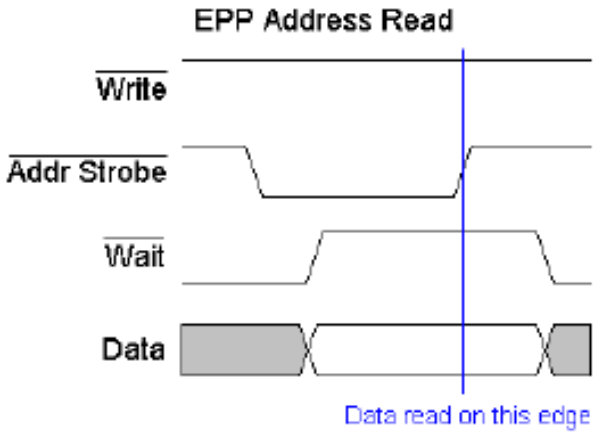
ภาพประกอบ 4-13 แสดงลักษณะการโต้ตอบของรอบการอ่านข้อมูล

(ที่มา : Craig Peacocock, Interfacing the Enhanced Parallel Port version 1.0)

จากภาพประกอบ 4-13 แสดงรายละเอียดได้ดังนี้

- โปรแกรมเขียนไปที่พอร์ตข้อมูลที่ Base address+4
- สัญญาณ Data strobe จะถูกส่งออกมา (เป็นลอจิก 0) ถ้าสัญญาณ Wait เป็นลอจิก 0 (เริ่มรอบการเขียนข้อมูล)
- Host จะรอการตอบรับ (Acknowledgement) โดยรอจนกว่าสัญญาณ Wait จะเป็นลอจิก 1 (เพื่อหยุดรอบการทำงาน)
- สัญญาณ Data strobe จะกลับเป็นลอจิก 1
- สิ้นสุดรอบการอ่านข้อมูล

4.3.3.4 รอบการอ่านแอดแดรส(Address read cycle)



ภาพประกอบ 4-14 แสดงลักษณะการโต้ตอบของรอบการอ่านข้อมูล

(ที่มา : Craig Peacocock, Interfacing the Enhanced Parallel Port version 1.0)

จากภาพประกอบ 4-14 แสดงรายละเอียดได้ดังนี้

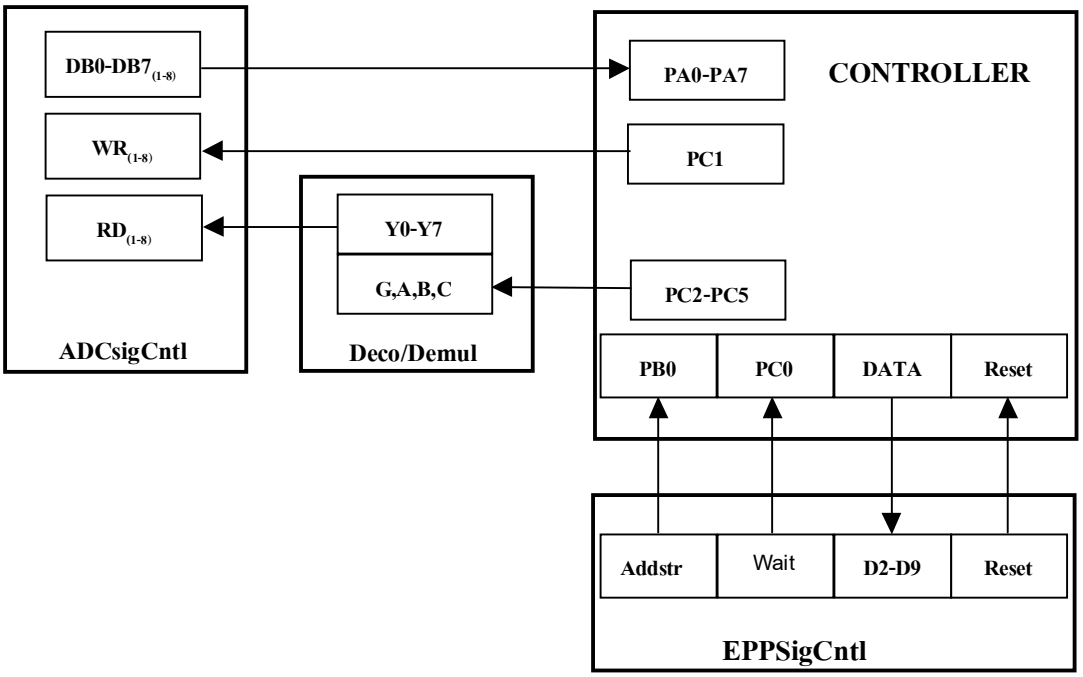
- โปรแกรมเขียนไปที่พอร์ตข้อมูลที่ Base address+4
- สัญญาณ Data strobe จะถูกส่งออกมา (เป็นลอจิก 0) ถ้าสัญญาณ Wait เป็นลอจิก 0 (เริ่มรอบการเขียนข้อมูล)
- Host จะรอการตอบรับ (Acknowledgement) โดยรอจนกว่าสัญญาณ Wait จะเป็นลอจิก 1 (เพื่อหยุดรอบการทำงาน)
- สัญญาณ Data strobe จะกลับเป็นลอจิก 1
- สิ้นสุดรอบการอ่านข้อมูล

สำหรับโหมดการทำงานแบบ EPP มาตรฐาน 1.7 สัญญาณ Data strobe และ Address strobe จะถูกส่งออกมาโดยไม่จำเป็นต้องรอให้สัญญาณ Wait เป็นลอจิก 0 แต่ทั้งโหมดการทำงานแบบ EPP มาตรฐาน 1.7 และโหมดการทำงานแบบ EPP มาตรฐาน 1.9 จำเป็นต้องรอจนกว่าสัญญาณ Wait เป็นลอจิก 1 สิ้นสุดรอบการทำงาน

ในงานวิจัยนี้จะใช้พอร์ตขนานในโหมดการทำงานแบบ EPP ในการอินเตอร์เฟสกับเครื่องคอมพิวเตอร์กับไมโครคอนโทรลเลอร์เพื่อนำข้อมูลดิจิตอลมาเก็บในฮาร์ดดิกส์ ดังนั้นจึ่งใช้เพียงรอบทำงานในการอ่านพอร์ตขนานเท่านั้น

4.4 การออกแบบตัวควบคุมการบันทึกสัญญาณเสียงเต้นของหัวใจ

ในงานวิจัยนี้ตัวควบคุมจะใช้ไมโครคอนโทรลเลอร์ตระกูล AVRเบอร์90S8535 (รายละเอียดและสถาปัตยกรรมได้กล่าวไปแล้วข้างต้น) สำหรับตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลได้เลือกใช้เบอร์ ADC0820 ในส่วนของการเขียนโปรแกรมเพื่อควบคุมการบันทึกสัญญาณข้อมูลจะใช้ภาษาแอสแซมบลี้(Assembly Language Program) ซึ่งรายละเอียดเกี่ยวกับขั้นตอนการพัฒนาออกแบบตัวควบคุมรวมทั้งรหัสต้นฉบับ(Source code) ของโปรแกรมได้แสดงในภาคผนวก ก และ ข ตามลำดับ โดยสามารถแสดงรายละเอียดของการออกแบบตัวควบคุมได้ดังภาพประกอบ 4-15



ภาพประกอบ 4-15 แสดงการเชื่อมต่อสัญญาณต่างๆของการออกแบบตัวควบคุม

สำหรับการออกแบบตัวควบคุมโดยใช้ไมโครคอนโทรลเลอร์ AVR90s8535 ร่วมกับส่วนหลักของวงจรต่างๆ ตมที่ได้กล่าวมาทั้งหมด สัญญาณที่ใช้ในการควบคุมการสั่งเริ่ม(Start), สั่งหยุด(Stop) จะเป็นสัญญาณที่มาจากพอร์ตขนานในโหมดการทำงานแบบ EPP สัญญาณเหล่านี้ ผู้ใช้

สามารถใช้งานได้อย่างอิสระ เพื่อใช้ประโยชน์จากพอร์ตขนานให้มากที่สุด สรุปชื่อสัญญาณที่ใช้ทั้งหมด ร่วมกับตัวควบคุมดังตาราง 4-4

ส่วนอุปกรณ์เชื่อมต่อ			ส่วนตัวควบคุม AVR90S8535	
ส่วน	**ชื่อสัญญาณ**	**ขา(Pin)**	**ชื่อสัญญาณ**	**ขา(Pin)**
1. วงจรADC	WR RD D0-D7	6 8 2,3,4,5,14,15,16,17	PC1 PA0-PA7	23 40,39,38,37,36, 35,34,33
2. IC 74LS138 เข้ารหัส/ถอดรหัส	G1 A,B,C Y0-Y7	6 1,2,3 15,14,13,12,11,10, 9,7	PC5 PC4,PC3,PC2	27 24,25,26
3. พอร์ตขนานใน โหมดการทำงาน แบบ EPP	D2-D9 Wait Reset Addrestrob	2,3,4,5,6,7,8,9 11 16 17	PD0-PD7 PC0 Reset PB0	14-20 22 9 1

ตารางที่ 4-4 แสดงการสรุปสัญญาณทั้งหมดที่ใช้ในการสร้างตัวควบคุมการบันทึก

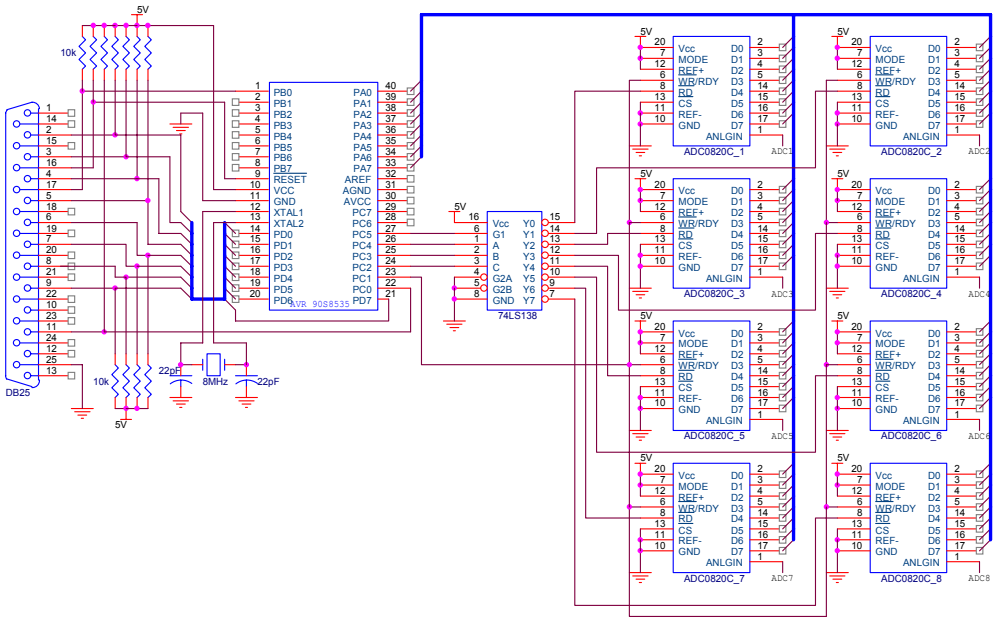
4.5 การจัดสร้างตัวควบคุมการบันทึกเสียงเต้นของหัวใจ

ผู้วิจัยได้สร้างบอร์ดสำหรับการทดสอบการทำงานของตัวควบคุมจำนวน 2 บอร์ด ดังนี้

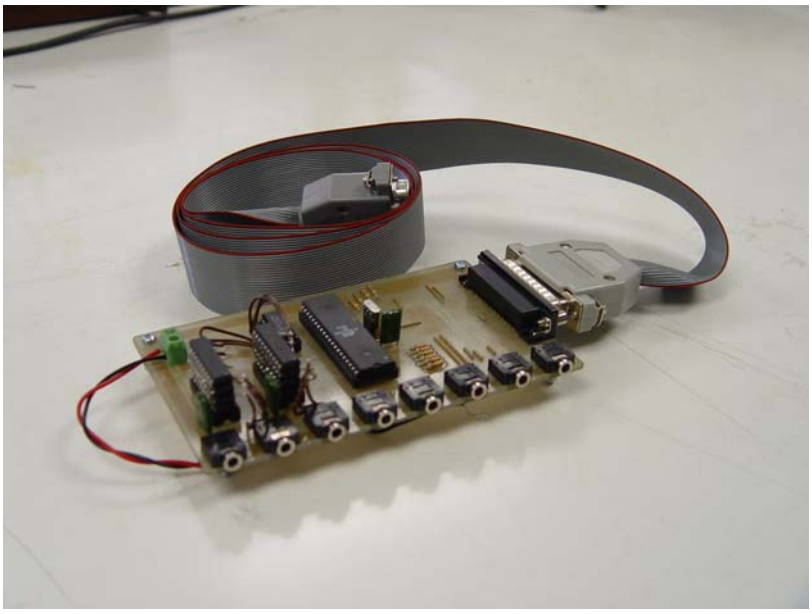
- บอร์ดรุ่นที่ 1 ประกอบด้วยบอร์ดสำเร็จรูปสำหรับการพัฒนาของไมโครคอนโทรลเลอร์ AVR90S8535 ได้ต่อวงจรในส่วน ADC จำนวน 6 ช่องสัญญาณ และคอนเน็คเตอร์ 25 PIN สำหรับเชื่อมต่อพอร์ตขนาน ดังแสดงในภาพประกอบ 4-16

ภาพประกอบ 4-16 แสดงบอร์ดรุ่นที่ 1 สำหรับการทดสอบ

- บอร์ดรุ่นที่ 2 ได้จัดสร้างแผ่นวงจรพิมพ์ ประกอบด้วย ตัวควบคุมAVR90S8535 ส่วน ADC จำนวน 8 ช่องสัญญาณ ซึ่งพัฒนาเพิ่มช่องสัญญาณขึ้นมาโดยใช้ตัวเข้ารหัส/ถอดรหัส เบอร์ 74LS138 เนื่องจากพอร์ตสัญญาณจากตัวควบคุมมีจำกัด และส่วนของพอร์ตขนาน โดยสามารถดู Schematic diagram ของวงจรได้ในภาพประกอบ 4-17 และบอร์ดที่พัฒนาขึ้นดังภาพประกอบ 4-18



ภาพประกอบ 4-17 แสดงรายละเอียดวงจรตัวควบคุมการบันทึกข้อมูล



ภาพประกอบ 4-18 แสดงบอร์ดรุ่นที่ 2 ที่พัฒนาขึ้นสำหรับควบคุมการบันทึกสัญญาณ

4.6 คอมพิวเตอร์

คอมพิวเตอร์เป็นส่วนที่มีความสำคัญมากส่วนหนึ่ง ซึ่งมีซอฟต์แวร์ที่ทำหน้าที่ควบคุมการสั่งให้เริ่มต้นหรือหยุดการทำงาน และนำข้อมูลมาเก็บบันทึกลงบนฮาร์ดดิสก์ เพื่อนำข้อมูลไปใช้ประโยชน์เกี่ยวกับสัญญาณตามแต่ผู้ใช้ต้องการ โดยรายละเอียดของซอฟต์แวร์จะนำเสนอในส่วนของการแสดงผลบทต่อไป