ภาคผนวก

ภาคผนวก ก ขั้นตอนการออกแบบพัฒนาวงจรตัวควบคุมไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ 90S8535

สำหรับการออกแบบวงจรตัวควบคุมไมโครคอนโทรลเลอร์ จะใช้ภาษาบรรยายฮาร์ดแวร์หรือภาษาแอสแซมบลี้(Assembly Language Program) ซึ่งเป็นภาษาระดับต่ำมีความยุ่งยากเมื่อเทียบกับภาษาซี แต่มีข้อดีคือ ควบคุมการทำงานในระดับฮาร์ดแวร์ได้ง่าย มีความเร็วสูง และโปรแกรมมีขนาดเล็ก สำหรับการออกแบบในส่วนวงจรตัวควบคุม ในการบรรยายลักษณะการทำงานของดิจิตอลฮาร์ดแวร์จะบรรยายเชิงพฤติกรรม (Behavioral description) หรืออธิบายความสัมพันธ์ระหว่างอินพุตกับเอาท์พุท โดยผู้ออกแบบไม่จำเป็นต้องรู้ถึงรายละเอียดของวงจรว่าประกอบด้วยเกตชนิดใดเพียงแต่รู้หน้าที่การทำงานและความสัมพันธ์ระหว่างอินพุตและเอาท์พุตของวงจรที่ต้องการออกแบบ ขั้นตอนการพัฒนาออกแบบตัวควบคุมประกอบด้วยขั้นตอนหลักคือ ขั้นตอนการการใช้ซอฟต์แวร์หรือตัวคอมไพเลอร์ (Compiler) และขั้นตอนการดาวน์โหลดลงอุปกรณ์เป้าหมาย (Target device) คือชิพ AVR เพื่อโปรแกรมการทำงาน (Configuration) ซึ่งทั้งสองขั้นตอนนี้มีรายละเอียดและวิธีการพัฒนาดังนี้

1. แนะนำการใช้โปรแกรม Astudio4 (AVR assembler)

สำหรับในหัวข้อนี้เราจะอธิบายถึงวิธีการใช้งานโปรแกรม Astudio4 เวอร์ชั่น 4.05 อย่างง่าย นั่นคือเราจะอธิบายเฉพาะการใช้ text editer และวิธีการคอมไพล์โปรแกรมเท่านั้น แต่ในส่วนอื่นเช่นการ debug program, chip simulator หากท่านใดสนใจก็สามารถศึกษาได้จากเมนู Help

สำหรับโปรแกรมนี้มันสามารถทำงานได้บนวินโดวส์ 9x/ME/NT/2000/XP ดังนั้นจึงไม่มีปัญหาใดๆสำหรับผู้ใช้ระบบปฏิบัติการที่ต่างชนิดกัน การติดตั้งโปรแกรมก็ทำได้ไม่ยาก ซึ่งสามารถดาวน์โหลดได้ที่ www.Atmel.comโดยให้ท่านคลายไฟล์ Astudio4(zip file)ออกมา จากนั้นก็รันไฟล์ setup.exe และก็ทำตามขั้นตอนที่แนะนำไปเรื่อยๆ จนการติดตั้งเสร็จสมบูรณ์ ซึ่งหลังการติดตั้งโปรแกรมนี้มันก็พร้อมที่จะทำงานทันทีโดยเราไม่ต้องเสียเวลารีสตาร์ตเครื่องใหม่

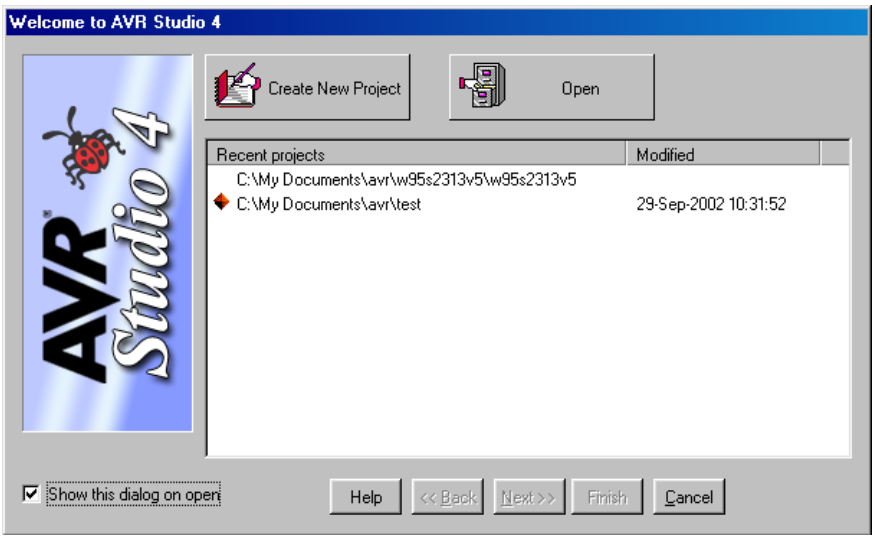
System requirement

- โปรแกรมนี้รันบนระบบปฏิบัติการ windows 9x/ME/NT/2000/XP
- ความเร็วของ CPU Intel Pentium 200 MHz
- ความละเอียดหน้าจอไม่ควรต่ำกว่า 800x600 pixels
- แรม 64 MB

- ที่ว่างฮาร์ดดิสก์ 15 MB ขึ้นไป

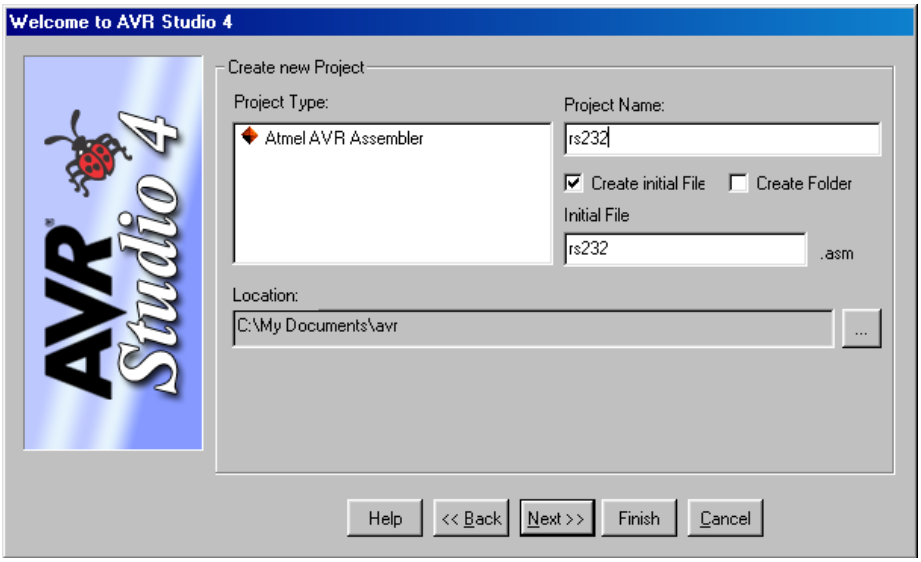
1.1 การสร้างไฟล์โปรเจ็กต์และไฟล์อื่น ๆ

ถ้าการติดตั้งถูกต้อง เมื่อเราเปิดโปรแกรมนี้ขึ้นมา เราจะพบหน้าต่าง Welcome to AVR Studio4 ดังนี้



ภาพประกอบ ก1 หน้าต่าง Welcome to AVR Studio 4

หน้าต่างนี้จะมีไว้สำหรับให้เราสามารถทำการสร้างโปรเจ็กต์ไฟล์ใหม่หรือสามารถเปิดโปรเจ็กต์ไฟล์ที่มีอยู่แล้วได้ โปรเจ็กต์ไฟล์จะมีนามสกุลเป็น .aps สำหรับหน้าต่างนี้มันจะแสดงเป็นอันดับแรกของการเปิดโปรแกรมทุกครั้ง หากเราไม่ต้องการให้มันแสดงหน้าต่างนี้ เราสามารถทำได้โดยการคลิกเมาส์ 1 ครั้งที่เช็คบล็อกShow this dialog on open ที่อยู่ทางด้านล่างซ้ายมือของหน้าต่างนี้ เพื่อเอาเครื่องหมายถูกที่กาอยู่ในเช็คบล็อกออกไป ซึ่งในการเปิดโปรแกรมครั้งต่อไปท่านจะไม่พบหน้าต่างนี้อีก และการเปิดหรือสร้างโปรเจ็กต์ใหม่เราสามารถทำได้ผ่านทางออปชั่น Project ที่อยู่ในเมนูบาร์แทน แต่สำหรับการอธิบายในที่นี้เราจะใช้หน้าต่างนี้ในการสร้างโปรเจ็กต์ใหม่ โดยให้ท่านคลิกที่ปุ่ม Create New Project ซึ่งท่านจะพบกับหน้าต่างใหม่ดังรูปต่อไปนี้



ภาพประกอบ ก2 หน้าต่างที่ใช้ในการสร้างโปรเจ็กต์ไฟล์

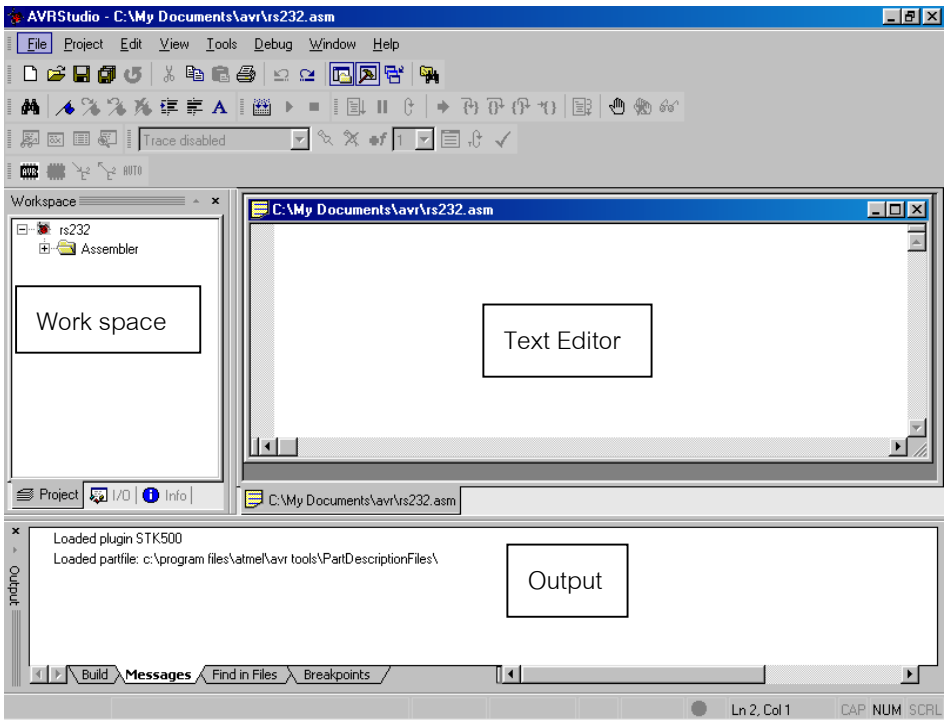
ในขณะที่ทำการสร้างไฟล์โปรเจ็กต์ โปรแกรมสามารถที่จะสร้างไฟล์ .asm ซึ่งเป็นไฟล์โปรแกรมภาษาแอสเซมบลีได้พร้อมกันไปในตัวโดยการกาเครื่องหมายถูกที่เช็คบ็อก Create Initial File จากนั้นให้เราพิมพ์ชื่อไฟล์โปรเจ็กต์ที่ช่องข้อความ Project Name ในตัวอย่างเราจะใช้ชื่อไฟล์โปรเจ็กต์ rs232 และในระหว่างที่เราพิมพ์ชื่อไฟล์โปรเจ็กต์ เราจะสังเกตุเห็นว่าโปรแกรมจะพิมพ์ชื่อไฟล์โปรเจ็กต์ลงที่ช่องข้อความ Initial File ไปด้วย ซึ่งจะเป็นการสร้างไฟล์โปรแกรม .asm (ไฟล์แอสเซมบลี) และหากเราต้องการให้โปรแกรมสร้างไฟล์ .asm ที่มีชื่อต่างจากชื่อไฟล์โปรเจ็กต์ เราก็สามารถแก้ไขชื่อไฟล์ .asm ได้ที่ช่องข้อความ Initial File นี้

ข้อควรระวังในการตั้งชื่อไฟล์แอสเซมบลีต่างจากชื่อไฟล์โปรเจ็กต์นั่นก็คือ เมื่อเราทำการคอมไพล์แล้วไฟล์ต่างๆ ที่โปรแกรมทำการสร้างขึ้นมามันจะตั้งชื่อตามชื่อไฟล์โปรเจ็กต์เช่น ถ้าเราตั้งชื่อไฟล์โปรเจ็กต์เป็น Testproject และเราตั้งชื่อไฟล์แอสเซมบลีชื่อ rs232 หลังการคอมไพล์เราจะได้ไฟล์ Testproject.hex ดังนี้เป็นต้น

หากเราไม่ต้องการให้มันสร้างไฟล์โปรแกรม .asm ขึ้นในขณะที่เราสร้างไฟล์โปรเจ็กต์ เราสามารถทำได้โดยการคลิกที่เช็คบ็อก Create Initial File เพื่อเอาเครื่องหมายถูกที่กาอยู่ออก ส่วนที่เช็คบ็อก Create Folder จะเป็นการสร้างโฟลเดอร์ขึ้นมาเพื่อเก็บไฟล์โปรเจ็คไว้นั่นเอง ถัดลงมาที่ช่องข้อความ Location จะเป็นตำแหน่งของไฟล์โปรเจ็คที่จะถูกนำไปเก็บ ซึ่งเราสามารถเลือกที่จะเก็บไฟล์ที่ใหม่ได้โดยการคลิกที่ปุ่ม ... ที่อยู่ท้ายช่องรับความหรือพิมพ์ชื่อ path ลงไปเลยก็ได้เช่น

เดียวกัน สำหรับในตัวอย่างนี้เราจะเก็บไฟล์โปรเจ็กต์ไว้ที่ C:\My Documents\avr และทำการสร้างไฟล์ .asm ขึ้นพร้อมกับการสร้างไฟล์โปรเจ็กต์ด้วย

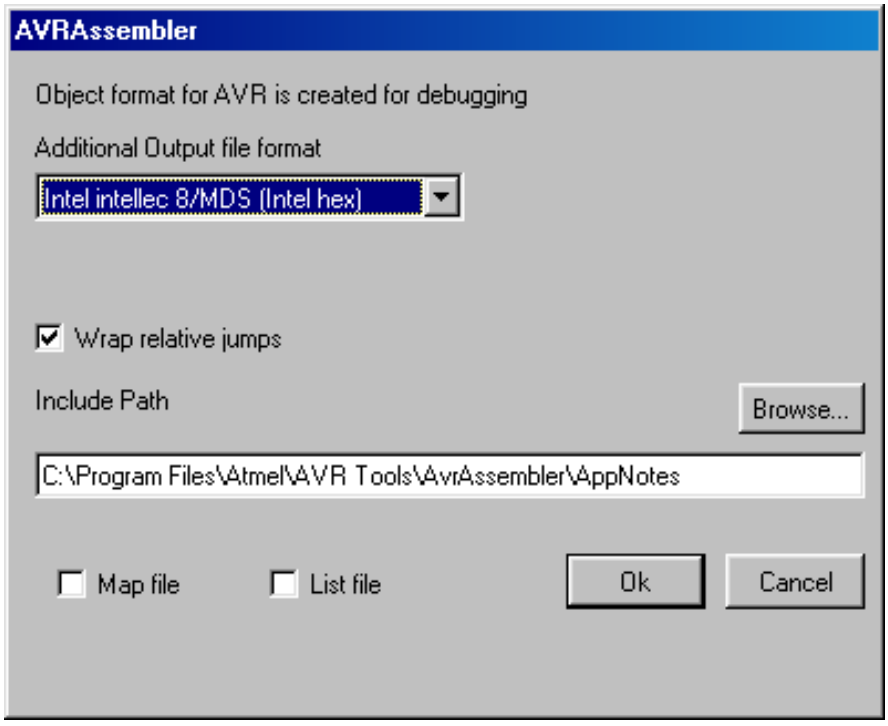
เมื่อทุกอย่างถูกต้องดีแล้วก็ให้เราทำการคลิกที่ปุ่ม Finish เพื่อจะเข้าหน้าจอ text editor สำหรับเขียนโปรแกรมต่อไป หน้าจอที่เกิดขึ้นใหม่หลังการกดปุ่ม Finish แสดงได้ดังรูปต่อไปนี้ (ถ้าเรากดปุ่ม Next โปรแกรมจะให้เราเลือก paltform และ MCU ที่เราจะใช้ในการซิมมูเลตและ debug โปรแกรม ต่อไป)



ภาพประกอบ ก3 หน้าจอที่เกิดขึ้นหลังการกดปุ่ม Finish

จากภาพประกอบ ก3 จะประกอบไปด้วยหน้าต่าง Work space ซึ่งมีไว้สำหรับจัดการกับไฟล์ในโปรเจ็กต์เช่นการ Add existing file, Remove File หรือ การ Set as Entry File เป็นต้น ส่วนหน้าต่างถัดมาคือ Text Editor มีไว้สำหรับให้เราเขียนโปรแกรมลงไป และสุดท้ายเป็นหน้าต่าง Output ที่ทำหน้าที่แสดงข้อความบอกให้เราทราบถึงผลจากการคอมไพล์โปรแกรม ซึ่งหากท่านไม่พบหน้าต่างทั้งสามนี้ท่านสามารถเปิดมันขึ้นมาได้เอง โดยการเข้าไปเลือกที่เมนู View

เมื่อเราทำการเขียนโปรแกรมลงไปที่ text editor เรียบร้อยแล้วก็ให้ทำการบันทึกไฟล์โดยการเลือกไปที่เมนู File>save และทำการคอมไพล์โปรแกรมโดยการกดปุ่ม F7 หรือเลือกที่เมนู Project>Build หากเราเขียนโปรแกรมถูกต้อง โปรแกรมจะไม่ฟ้อง Error ที่หน้าต่าง Output แต่หากเขียนโปรแกรมไม่ถูกต้องมันจะฟ้อง Error หรือ Warning ก็ให้เราแก้ไขมันให้ถูกต้องและทำการคอมไพล์ใหม่ซึ่งจะทำให้ได้ไฟล์ rs232.aps, rs232.hex, rs232.lst, rs232.map, rs232.asm และ rs232.obj ขึ้นมา หากเราไม่ต้องการไฟล์ .map และ .lst ให้เราเข้าไปที่เมนู Project>AVR Asembler Setup จากนั้นจะพบหน้าต่าง AVRAssembler ให้เราคลิกเอาเครื่องหมายถูกที่ช่อง Map File กับ Lst File ออกและจากนั้นก็กดปุ่ม Ok ก็เป็นอันเสร็จสิ้น สำหรับไฟล์นามสกุลต่างๆ ดังที่เรากล่าวมาข้างต้นท่านสามารถเปิดเข้าไปดูเนื้อหาข้างในไฟล์ได้ด้วยการเลือกไปที่เมนู File>Open File จากนั้นก็เลือกไฟล์ที่ท่านต้องการเปิดและกดปุ่ม Open



ภาพประกอบ ก4 หน้าต่าง AVRAssembler

1.2 การเปิดไฟล์โปรเจ็กต์และไฟล์แอสเซมบลี

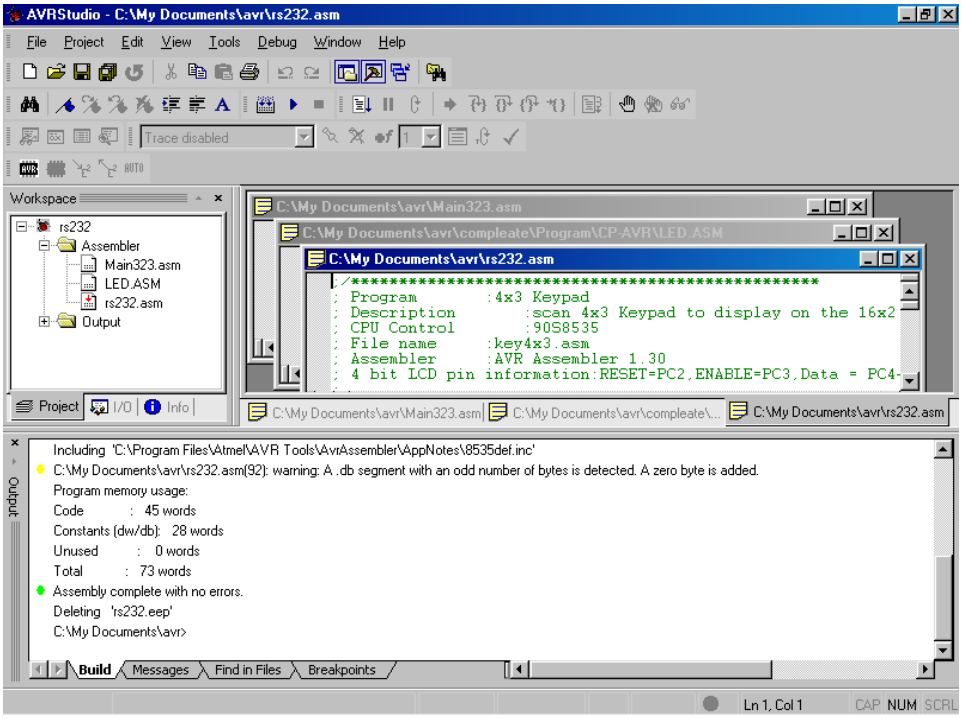
เมื่อเราต้องการเปิดโปรแกรมที่มีอยู่แล้วขึ้นมาแก้ไขเราสามารถทำได้โดยการคลิกเมนู Project>Open Project เพื่อเป็นการเปิดไฟล์โปรเจ็กต์(*.aps) หลังจากที่เราเปิดไฟล์โปรเจ็กต์แล้วจากนั้นก็คลิกเลือกไปที่โฟลเดอร์ Assembler ที่อยู่ในหน้าต่าง Work space เราจะพบไฟล์ .asm ที่เราต้องการแก้ไขอยู่ในนั้น แต่ถ้าหากเราเปิดไฟล์โดยใช้เมนู File>Open File เราก็สามารถที่จะเปิดไฟล์ .asm(หรือไฟล์อื่นๆ ที่ text editor รองรับ) ได้เช่นเดียวกัน แต่เราจะไม่สามารถทำการคอมไพล์มันได้ เป็นได้แต่เพียงการแก้ไขและบันทึกเท่านั้น อย่างไรก็ตามการเปิดไฟล์โปรเจ็กต์(*.aps) ด้วยเมนู File>Open File นี้ก็สามารถทำได้และให้ผลการทำงานเช่นเดียวกับการเปิดมันด้วยเมนู Project>Open Project

1.2.1 การ Add existing File, Remove File และ Set as Entry File

- **Add existing File** ก็คือการเพิ่มไฟล์ .asm เข้ามาในโปรเจ็กต์ โดยเราสามารถทำได้ 2 วิธีด้วยกันคือการเลือกไปที่เมนู Project>Add existing File จากนั้นก็ให้เราเลือกไฟล์ .asm ที่เราต้องการนำเข้ามา ส่วนอีกวิธีหนึ่งทำได้โดยการคลิกเมาส์ปุ่มขวาไปที่โฟลเดอร์ Assembler ที่อยู่ในหน้าต่าง Work space จากนั้นเราจะเห็นหน้าต่างออปชั่นเล็กๆ ปรากฏขึ้นมาก็ให้เราเลื่อนเมาส์ไปคลิกเลือกที่ Add existing File จะทำให้เราสามารเลือกไฟล์ .asm ที่เราต้องการได้

- **Remove File** จะทำงานตรงกันข้ามกับ Add existing File นั่นคือมันจะลบไฟล์ .asm ที่เราไม่ต้องการออกจากโปรเจ็กต์ไฟล์ของเรา เราสามารถ Removeไฟล์ได้โดยการคลิกเมาส์ปุ่มขวาเลือกไปที่ไฟล์ที่เราต้องการลบ จากนั้นจะปรากฏหน้าต่างขึ้นมาให้เราเลือกไปที่ Remove File form Project เพียงเท่านี้ไฟล์ที่เราเลือกมันก็จะถูกลบออกไปจากโปรเจ็กต์ที่เรากำลังทำงานอยู่

- **Set as Entry File** จะเป็นการเลือกไฟล์ .asm ที่อยู่ในโปรเจ็กต์ไฟล์ขณะนั้นให้มันพร้อมที่จะถูกคอมไพล์ ฟังก์ชั่นนี้จะมีความจำเป็นมากเมื่อเรามีไฟล์ .asm หลายไฟล์อยู่ในโปรเจ็กต์เดียวกัน ซึ่งโปรแกรม Astudio4 นี้จะไม่ทำการคอมไพล์ไฟล์ .asm ทั้งหมดพร้อมกัน แต่จะเลือกคอมไพล์เฉพาะไฟล์ที่โดน Set as Entry File เท่านั้น เราสามารถรู้ได้ว่าไฟล์ .asm ตัวใดที่ถูก set ไว้โดยการสังเกตไอคอนที่อยู่หน้าชื่อไฟล์จะมีรูปลูกศรสีแดงหันหัวลงดังนี้ หรือดูได้จากภาพประกอบ ก5



ภาพประกอบ ก5 หน้าจอที่เกิดจากการ Add existing File และหน้าต่าง Output ที่รายงานผลการคอมไพล์โปรดสังเกตการ Warning (วงกลมสีเหลือง) ต่างๆด้วย ถึงแม้การคอมไพล์ไม่ผิดพลาดแต่อาจจะมีผลต่อการทำงานของ CPU ได้

สำหรับการ Set as Entry File สามารถทำได้โดยการคลิกเมาส์ปุ่มขวาเลือกไปที่ชื่อไฟล์ที่เราจะต้องการคอมไพล์จากนั้นจะปรากฏเมนูเล็กๆ ขึ้นมาให้เราเลือกไปที่ Set as Entry File ก็เป็นอันเสร็จ เราควรจำไว้อย่างหนึ่งว่าหลังการคอมไพล์แล้วไม่ว่าเราจะทำการคอมไพล์ไฟล์ .asm ตัวใดก็ตามในโปรเจ็กต์นั้น ไฟล์เอาพุตที่โปรแกรมสร้างขึ้นมานั้นจะมีชื่อตามชื่อไฟล์โปรเจ็กต์นั้นเสมอ มันจะไม่มีชื่อไฟล์ตามไฟล์ .asm ที่ถูกคอมไฟล์ ดังนั้นเพื่อป้องกันการสับสนท่านอาจจะให้มีเพียงไฟล์ .asm เพียงไฟล์เดียวอยู่ในไฟล์โปรเจ็กต์แต่ละโปรเจ็กต์ก็ได้และตั้งชื่อให้ทั้งสองไฟล์นี้ตรงกัน

อีกอย่างหนึ่งที่อยากจะให้สังเกต นั่นก็คือผลที่ได้จากการคอมไพล์ในแต่ละครั้งว่าเป็นเช่นไรบ้างเช่น ขนาดของไฟล์ที่คอมไพล์ได้มีขนาดเท่าไรซึ่งบางทีเราเขียนโปรแกรมเพลินมันอาจจะมีขนาดโตกว่าหน่วยความจำที่จะเก็บมันก็ได้ หรือที่สำคัญอีกอย่างหนึ่งก็คือข้อความ Warning ที่โปรแกรมเตือนดังนี้

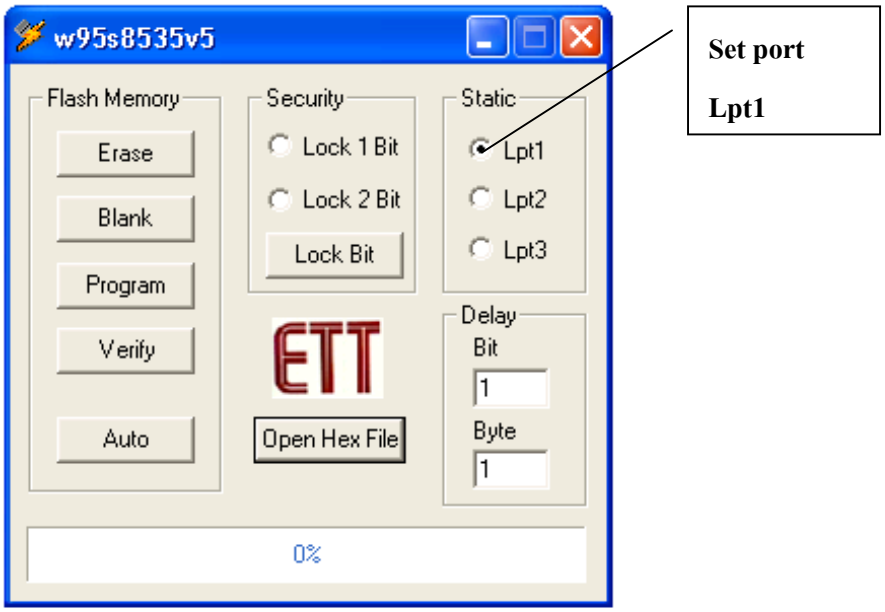
“C:\My Documents\avr\rs232.asm(92): warning: A .db segment with an odd number of bytes is detected. A zero byte is added.”

หากท่านใดเคยเล่น AVR และใช้ฟังก์ชั่นที่เกี่ยวกับพอร์ตอนุกรม RS232 คงพอจะทราบดีว่าบางครั้ง AVR มันก็ไม่สามารถแสดงข้อความที่อยู่หลังบรรทัดนี้(บรรทัดที่แสดงข้อความ Warning) ต่อไปได้ ทั้งที่ข้อความยังไม่สิ้นสุด นั่นเป็นเพราะว่ามันจะทำการเขียนข้อมูลลงหน่วยความจำโปรแกรมเป็นทีละ word(2 ไบต์) ดังนั้นเมื่อข้อความที่เราต้องการจะเขียนลงบนหน่วยความจำมันมีจำนวนเป็นเลขคี่ มันก็จะเพิ่มเลข 0 เข้าไปท้ายข้อความในบรรทัดนั้นๆ เพื่อให้ครบจำนวนเป็นเลขคู่เสมอ ทีนี้เมื่อเราเขียนโปรแกรมให้ตรวจสอบข้อความว่าถ้าพบเลข 0 เมื่อใดก็ให้ถือว่าสิ้นสุดข้อความทันที ดังนั้นเมื่อโปรแกรมทำการอ่านและตรวจสอบข้อมูลมาถึงไบต์ที่ AVR เพิ่มเลข 0 ขึ้นมาเอง โปรแกรมก็เข้าใจว่าเราต้องการสิ้นสุดข้อความที่บรรทัดนี้ดังนั้นจึงทำให้ข้อความที่เหลือไม่ถูกตรวจพบนั่นเอง และวิธีแก้ง่ายๆ นั่นก็คือให้เราเพิ่มอักขระอะไรก็ได้เข้าไปอีกตัวหนึ่งในบรรทัดนี้ยกเว้นเลข 0 เท่านี้ก็สามารถแก้การ warning ตรงนี้ได้(เลข 0 ที่เรากำลังพูดถึงกันอยู่ขณะนี้ไม่ใช่เลข “0” ที่เป็นรหัส ASCII นะครับหากแต่เป็นเลข 0 ที่เป็นเซตของเลขจำนวนเต็ม)

2. แนะนำการใช้โปรแกรม w95s8535v5

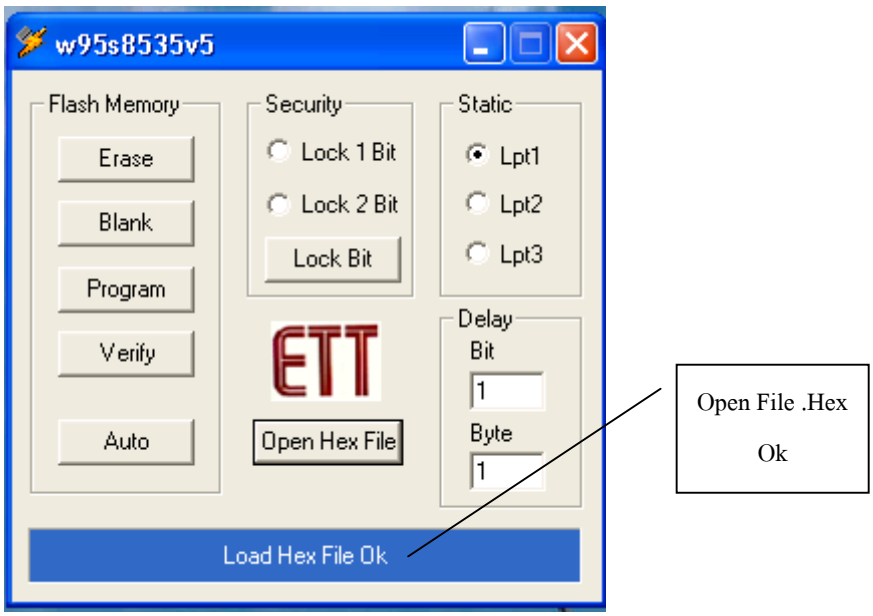
สำหรับในหัวข้อนี้จะอธิบายถึงวิธีการใช้งานโปรแกรม w95s8535v5 เวอร์ชั่น 5 โดยจะอธิบายวิธีการดาวน์โหลดโปรแกรมที่ผ่านการคอมไพล์โปรแกรมแล้วลงบนชิปเพื่อใช้งานตามความต้องการของผู้ใช้

สำหรับโปรแกรมนี้สามารถทำงานได้บนวินโดวส์ 9x/ME/NT/2000/XP ดังนั้นจึงไม่มีปัญหาใดๆสำหรับผู้ใช้ระบบปฏิบัติการที่ต่างชนิดกัน ซึ่งโปรแกรมดังกล่าวเป็นของบริษัทอีทีที สามารถดาวน์โหลดได้ที่ www.ett.co.th การติดตั้งโปรแกรมก็ทำได้ไม่ยาก หลังจากดาวน์โหลดไฟล์เสร็จแล้ว สามารถรันโปรแกรมใช้ได้ทันที โดยในการรันโปรแกรมใช้ครั้งแรก จะต้องเซ็ตเลือกตำแหน่งพอร์ตสำหรับการโหลดโปรแกรมใน Frame Static ในภาพประกอบ ก6 ได้เลือกใช้พอร์ต Lpt1



ภาพประกอบ ก6 แสดงหน้าต่างโปรแกรม w95s8535v5

สำหรับหน้าต่างนี้มันจะแสดงเป็นอันดับแรกของการเปิดโปรแกรมทุกครั้ง หลังจากนั้นเลือกคลิกปุ่ม Open Hex File ซึ่งจะแสดง Path ของ window ขึ้นมาจากนั้นทำการเลือกไฟล์โปรแกรมที่คอมไฟล์แล้ว ซึ่งไฟล์จะมีนามสกุลเป็น .HEX หลังจาก Open Hex File แล้วหน้าต่างโปรแกรมในส่วน Text ด้านล่างสุดจากที่แสดงข้อความ 0% จะแสดงข้อความ Load Hex File Ok ซึ่งหมายความว่า โปรแกรมพร้อมที่จะทำการโหลดไฟล์ลงบนอุปกรณ์แล้ว



ภาพประกอบ ก7 แสดงหน้าต่างเมื่อทำการ Open File .Hex

หลังจากที่ทำการเปิดไฟล์ .Hex เสร็จเรียบร้อยแล้วก็มาถึงขั้นตอนการโหลดโปรแกรมลงบนชิพ ซึ่งจะต้องทำการตรวจสอบการต่อพอร์ตและเปิดไฟเลี้ยงจ่ายแก่บอร์ดอุปกรณ์ให้เรียบร้อย หลังจากนั้นกดปุ่ม Program เพื่อทำการโหลดโปรแกรมลงบนชิพเพื่อนำไปใช้งานต่อไป ในขบวนการโหลดเมื่อโปรแกรมโหลดข้อมูลเสร็จจะแสดงข้อความบอกดังภาพประกอบ

ภาพประกอบ ก8 แสดงหน้าต่างเมื่อโหลดโปรแกรมลงบนชิพ

ภาคผนวก ข โปรแกรมรหัสต้นฉบับภาษาแอสแซมบลี้สำหรับการออกแบบควบคุม

ลำดับการแสดงโปรแกรมรหัสต้นฉบับของการออกแบบมีดังนี้

1. ส่วนควบคุมการสร้างสัญญาณให้พอร์ตขนานในโหมดการทำงานแบบ EPP (EPPSigCntl)
2. ส่วนการสร้างสัญญาณควบคุมโหมดการเขียนและอ่าน ADC
3. ส่วนการรับและส่งข้อมูลดิจิตอลเข้าสู่เครื่องคอมพิวเตอร์ผ่านพอร์ตขนาน

การนิยามตำแหน่งรีจิสเตอร์อินพุทเอาท์พุท (I/O Register Definitions)

ในการพัฒนาโปรแกรมจะมีการอ้างชื่อไฟล์ (File name) เริ่มต้นก่อนที่จะเขียนโปรแกรมพัฒนางานตามความต้องการของผู้ใช้คือไฟล์ "8535def.inc" โดยในการพัฒนาโปรแกรมจะเริ่มต้นโปรแกรมคือ .include "8535def.inc" ซึ่งหมายถึง การรวมไฟล์ "8535def.inc" ทั้งหมดเข้าในโปรแกรมนี้ โดยในไฟล์นี้จะประกอบไปด้วย การนิยามค่าตำแหน่งต่างๆของรีจีสเตอร์อินพุทเอาท์พุท โดยตำแหน่งของชื่อรีจิสเตอร์จะนิยามค่าเป็นเลขฐาน 16 (hexadecimal) และตำแหน่งของบิทจะนิยามค่าเป็นเลข 0-7 โดยรายละเอียดการนิยามตำแหน่งรีจิสเตอร์อินพุทเอาท์พุทมีดังนี้

```
;***** I/O Register Definitions******
.equ    SREG        =       $3f
.equ    SPH         =       $3e
.equ    SPL         =       $3d
.equ    GIMSK       =       $3b
.equ    GIRF        =       $3a
.equ    TIMSK       =       $39
.equ    TIFR        =       $38
.equ    MCUCR       =       $35
.equ    MCUSR       =       $34
.equ    TCCR0       =       $33
.equ    TCNT0       =       $32
.equ    TCCR1A      =       $2f
.equ    TCCR1B      =       $2e
.equ    TCNT1H      =       $2d
.equ    TCNT1L      =       $2c
.equ    OCR1AH      =       $2b
.equ    OCR1AL      =       $2a
.equ    OCR1BH      =       $29
.equ    OCR1BL      =       $28
.equ    ICR1H       =       $27
.equ    ICR1L       =       $26
```

```
.equ    TCCR2       =       $25
.equ    TCNT2       =       $24
.equ    OCR2        =       $23
.equ    ASSR        =       22
.equ    WDTCR       =       $21
.equ    EEARH       =       $1f
.equ    EEARL       =       $1e
.equ    EEDR        =       $1d
.equ    EECR        =       $1c
.equ    PORTA       =       $1b
.equ    DDRA        =       $1a
.equ    PINA        =       $19
.equ    PORTB       =       $18
.equ    DDRB        =       $17
.equ    PINB        =       $16
.equ    PORTC       =       $15
.equ    DDRC        =       $14
.equ    PINC        =       $13
.equ    PORTD       =       $12
.equ    DDRD        =       $11
.equ    PIND        =       $10
.equ    SPDR        =       $0f
.equ    SPSR        =       $0e
.equ    SPCR        =       $0d
.equ    UDR         =       $0c
.equ    USR         =       $0b
.equ    UCR         =       $0a
.equ    UBRR        =       $09
.equ    ACSR        =       $08
.equ    ADMUX       =       $07
```

```
.equ    ADCSR       =       $06
.equ    ADCH        =       $05
.equ    ADCL        =       $04

;***** Bit Definitions*****
.equ    INT1        =       7
.equ    INT0        =       6

.equ    TOIE1       =       7
.equ    OCIE1A      =       6
.equ    OCIE1B      =       5
.equ    TICIE       =       3
.equ    TOIE0       =       1

.equ    TOV1        =       7
.equ    OCF1A       =       6
.equ    OCF1B       =       5
.equ    ICF1        =       3
.equ    TOV0        =       1

.equ    SRE         =       7
.equ    SRW         =       6
.equ    SE          =       5
.equ    SM          =       4
.equ    ISC11       =       3
.equ    ISC10       =       2
.equ    ISC01       =       1
.equ    ISC00       =       0

.equ    CS02        =       2
```

```
.equ    CS01        =    1
.equ    CS00        =    0

.equ    COM1A1      =    7
.equ    COM1A0      =    6
.equ    COM1B1      =    5
.equ    COM1B0      =    4
.equ    PWM11       =    1
.equ    PWM10       =    0

.equ    ICNC1       =    7
.equ    ICES1       =    6
.equ    CTC1        =    3
.equ    CS12        =    2
.equ    CS11        =    1
.equ    CS10        =    0

.equ    PWM2        =    6
.equ    COM21       =    5
.equ    COM20       =    4
.equ    CTC2        =    3
.equ    CS22        =    2
.equ    CS21        =    1
.equ    CS20        =    0

.equ    AS2         =    3
.equ    TCN2UB      =    2
.equ    OCR2UB      =    1
.equ    TCR2UB      =    0
```

```
.equ    WDTOE       =       4
.equ    WDE         =       3
.equ    WDP2        =       2
.equ    WDP1        =       1
.equ    WDP0        =       0

.equ    EEAR8       =       0

.equ    EEWE        =       1
.equ    EERE        =       0

.equ    PA7         =       7
.equ    PA6         =       6
.equ    PA5         =       5
.equ    PA4         =       4
.equ    PA3         =       3
.equ    PA2         =       2
.equ    PA1         =       1
.equ    PA0         =       0

.equ    DDA7        =       7
.equ    DDA6        =       6
.equ    DDA5        =       5
.equ    DDA4        =       4
.equ    DDA3        =       3
.equ    DDA2        =       2
.equ    DDA1        =       1
.equ    DDA0        =       0

.equ    PINA7       =       7
```

```
.equ    PINA6       =       6
.equ    PINA5       =       5
.equ    PINA4       =       4
.equ    PINA3       =       3
.equ    PINA2       =       2
.equ    PINA1       =       1
.equ    PINA0       =       0

.equ    PB7         =       7
.equ    PB6         =       6
.equ    PB5         =       5
.equ    PB4         =       4
.equ    PB3         =       3
.equ    PB2         =       2
.equ    PB1         =       1
.equ    PB0         =       0

.equ    DDB7        =       7
.equ    DDB6        =       6
.equ    DDB5        =       5
.equ    DDB4        =       4
.equ    DDB3        =       3
.equ    DDB2        =       2
.equ    DDB1        =       1
.equ    DDB0        =       0

.equ    PINB7       =       7
.equ    PINB6       =       6
.equ    PINB5       =       5
.equ    PINB4       =       4
```

```
.equ    PINB3       =       3
.equ    PINB2       =       2
.equ    PINB1       =       1
.equ    PINB0       =       0

.equ    PC7         =       7
.equ    PC6         =       6
.equ    PC5         =       5
.equ    PC4         =       4
.equ    PC3         =       3
.equ    PC2         =       2
.equ    PC1         =       1
.equ    PC0         =       0

.equ    DDC7        =       7
.equ    DDC6        =       6
.equ    DDC5        =       5
.equ    DDC4        =       4
.equ    DDC3        =       3
.equ    DDC2        =       2
.equ    DDC1        =       1
.equ    DDC0        =       0

.equ    PINC7       =       7
.equ    PINC6       =       6
.equ    PINC5       =       5
.equ    PINC4       =       4
.equ    PINC3       =       3
.equ    PINC2       =       2
.equ    PINC1       =       1
```

```
.equ    PINC0       =       0

.equ    PD6         =       6
.equ    PD5         =       5
.equ    PD4         =       4
.equ    PD3         =       3
.equ    PD2         =       2
.equ    PD1         =       1
.equ    PD0         =       0

.equ    DDD6        =       6
.equ    DDD5        =       5
.equ    DDD4        =       4
.equ    DDD3        =       3
.equ    DDD2        =       2
.equ    DDD1        =       1
.equ    DDD0        =       0

.equ    PIND6       =       6
.equ    PIND5       =       5
.equ    PIND4       =       4
.equ    PIND3       =       3
.equ    PIND2       =       2
.equ    PIND1       =       1
.equ    PIND0       =       0

.equ    SPIF        =       7
.equ    WCOL        =       6

.equ    SPIE        =       7
```

```
.equ    SPE             =       6
.equ    DORD            =       5
.equ    MSTR            =       4
.equ    CPOL            =       3
.equ    CPHA            =       2
.equ    SPR1            =       1
.equ    SPR0            =       0

.equ    RXC             =       7
.equ    TXC             =       6
.equ    UDRE            =       5
.equ    FE              =       4
.equ    OR              =       3

.equ    RXCIE           =       7
.equ    TXCIE           =       6
.equ    UDRIE           =       5
.equ    RXEN            =       4
.equ    TXEN            =       3
.equ    CHR9            =       2
.equ    RXB8            =       1
.equ    TXB8            =       0

.equ    ACD             =       7
.equ    ACO             =       5
.equ    ACI             =       4
.equ    ACIE            =       3
.equ    ACIC            =       2
.equ    ACIS1           =       1
.equ    ACIS0           =       0
```

```
.equ    MUX0            =       0
.equ    MUX1            =       1
.equ    MUX2            =       2

.equ    ADPS0           =       0
.equ    ADPS1           =       1
.equ    ADPS2           =       2
.equ    ADIE            =       3
.equ    ADIF            =       4
.equ    ADFR            =       5
.equ    ADSC            =       6
.equ    ADEN            =       7

.def    XL              =       r26
.def    XH              =       r27
.def    YL              =       r28
.def    YH              =       r29
.def    ZL              =       r30
.def    ZH              =       r31

.equ    RAMEND          =       $20+$40+$1ff    ;Adjust for registers and I/O

.equ    INT0addr        =       $001    ;External Interrupt0 Vector Address
.equ    INT1addr        =       $002    ;External Interrupt1 Vector Address
.equ    ICP1addr        =       $003    ;Input Capture1 Interrupt Vector Address
.equ    OC1Aaddr        =       $004    ;Output Compare1A Interrupt Vector Address
.equ    OC1Baddr        =       $005    ;Output Compare1B Interrupt Vector Address
.equ    OVF1addr        =       $006    ;Overflow1 Interrupt Vector Address
.equ    OC0addr         =       $007    ;Output Compare0 Interrupt Vector Address
.equ    OVF0addr        =       $008    ;Overflow0 Interrupt Vector Address
```

```
.equ    SPIaddr     =    $009    ;SPI Interrupt Vector Address
.equ    URXCaddr    =    $00a    ;UART Receive Complete Interrupt Vector Address
.equ    UDREaddr    =    $00b    ;UART Data Register Empty Interrupt Vector Address
.equ    UTXCaddr    =    $00c    ;UART Transmit Complete Interrupt Vector Address
.equ    ACIaddr     =    $00d    ;Analog Comparator Interrupt Vector Address
```

```
;/*****************************************************************************
; Program          :Phonocardiogram the external ADC
; Description      :convert analog input signal to 8 bit digital output.
;                  :The digital signal will be send to palalell port.
; Pin description :Digital input pin = PA7-PA0 : Output pin = PD7-PD0 : PC0 = wait : PC1 = WR
;                  :PB0 = Address strobe : PC2-PC5 Control signal decoder/demultiplexer to
;                  :G,A,B,C with RD1-RD8
; CPU Control      :90S8535
; File name        :Multiplex 8 channal.asm
; Assembler        :AVR Studio 4.05
;*****************************************************************************
.include "8535def.inc"
                   .org     $000
                   rjmp     RESET

                   .def     temp  = r19
                   .equ     koo =    0
                   .equ      kai =    1

RESET:             ldi      temp,low(RAMEND)
                   Out      SPL,temp
                   Ldi      temp,high(RAMEND)
                   Out      SPL+1,temp
;/***********************
;          set port
;/***********************
                   ser      r16
                   out      DDRC,r16
                   cbi      PORTC,koo
                   out      DDRD,r16
```

```
out     PORTB,r16
out     PORTA,r16
ldi     r16,0x00
out     DDRA,r16
out     DDRB,r16
sbi     PORTC,kai
ldi     r17,0b11000010
out     PORTC,r17

 ldi    r16,0b10100010      ;ch1 start
 mov    r1,r16
 ldi    r16,0b10100110      ;ch2
 mov    r2,r16
 ldi    r16,0b10101010      ;ch3
 mov    r3,r16
 ldi    r16,0b10101110      ;ch4
 mov    r4,r16
 ldi    r16,0b10110010      ;ch5
 mov    r5,r16
 ldi    r16,0b10110110      ;ch6
 mov    r6,r16
 ldi    r16,0b10111010      ;ch7
 mov    r7,r16
 ldi    r16,0b10111110      ;ch8
 mov    r8,r16

ldi     r16,0b01000010      ;ch1 end
ldi     r17,0b11000010      ;ch2
ldi     r21,0b11010110      ;ch3
ldi     r22,0b11011110      ;ch4
```

```
                ldi     r23,0b11001110          ;ch5
                ldi     r24,0b11011110          ;ch6
                ldi     r25,0b11011110          ;ch7
                ldi     r26,0b11011110          ;ch8
                ldi     r27,0xab

;/***********************************
;           start convert multiplex 8 chanall
;/***********************************
a:              in      r18,PINB                ;wait logic 0 from palalell port
                cpi     r18,0xfe
                brne    a
                cbi     PORTC,kai               ;ch1
                sbi     PORTC,kai               ;write complet
                rcall   delay                   ;delay time 1.5us

b:              in      r18,PINB
                cpi     r18,0xfe
                brne    b
                out     PORTC,r1                ;start read
                nop
                nop
                nop
                In      r20,PINA                ;read data r20
                out     PORTC,r16
                out     PORTD,r20               ;sent data to palalell ch1
                sbi     PORTC,koo
                cbi     PORTC,koo

c:              in      r18,PINB
```

```
        cpi     r18,0xfe
        brne    c
        out     PORTC,r2            ;ch2
        nop
        nop
        nop
        in      r20,PINA
        out     PORTC,r17
        out     PORTD,r20
        sbi     PORTC,koo
        cbi     PORTC,koo

d:      in      r18,PINB
        cpi     r18,0xfe
        brne    d
        out     PORTC,r3            ;ch3
        nop
        nop
        nop
        in      r20,PINA
        out     PORTC,r21
        out     PORTD,r20
        sbi     PORTC,koo
        cbi     PORTC,koo

e:      in      r18,PINB
        cpi     r18,0xfe
        brne    e
        out     PORTC,r4            ;ch4
        nop
```

```
        nop
        nop
        in      r20,PINA
        out     PORTC,r22
        out     PORTD,r20
        sbi     PORTC,koo
        cbi     PORTC,koo

f:      in      r18,PINB
        cpi     r18,0xfe
        brne    f
        out     PORTC,r5        ;ch5
        nop
        nop
        nop
        in      r20,PINA
        out     PORTC,r23
        out     PORTD,r20
        sbi     PORTC,koo
        cbi     PORTC,koo

g:      in      r18,PINB
        cpi     r18,0xfe
        brne    g
        out     PORTC,r6        ;ch6
        nop
        nop
        nop
        in      r20,PINA
        out     PORTC,r24
```

```
        out     PORTD,r20
        sbi     PORTC,koo
        cbi     PORTC,koo

h:      in      r18,PINB
        cpi     r18,0xfe
        brne    h
        out     PORTC,r7        ;ch7
        nop
        nop
        nop
        in      r20,PINA
        out     PORTC,r25
        out     PORTD,r20
        sbi     PORTC,koo
        cbi     PORTC,koo

i:      in      r18,PINB
        cpi     r18,0xfe
        brne    i
        out     PORTC,r8        ;ch8
        nop
        nop
        nop
        in      r20,PINA
        out     PORTC,r26
        out     PORTD,r20
        sbi     PORTC,koo
        cbi     PORTC,koo
```

```
                out     PORTD,r27               ;sent ab
                sbi     PORTC,koo
                cbi     PORTC,koo
                rjmp    a
;/************************
;       delay time
;/************************
delay   :       nop
                nop
                nop
                nop
                nop
                ret
```

ภาคผนวก ค.
ผลงานทางวิชาการ

บทความเรื่อง

การพัฒนาเครื่องต้นแบบเพื่อเก็บบันทึกและวิเคราะห์เสียงการเต้นของหัวใจจากหลายตำแหน่งบริเวณหน้าอก

Development of a Prototype System for Recording and Analysising of Simultaneous Phonocardiograms from Multiple chest sites.

นำเสนอในการประชุมเสนอผลงานวิจัยระดับบัณฑิตศึกษา มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 2