

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

การวิจัยการแยกพยางค์เสียงพูดภาษาไทยด้วยไมโครคอนโทรลเลอร์ มีหัวข้อการศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง เพื่อศึกษาข้อมูลต่างๆ ดังนี้

- 2.1 การแปลงสัญญาณจากอนาลอกเป็นดิจิตอล
- 2.2 การออกแบบขั้นตอนการทำงานของโปรแกรม
- 2.3 วงจรสำหรับการแยกพยางค์เสียงพูดภาษาไทยด้วยไมโครคอนโทรลเลอร์
- 2.4 งานวิจัยที่เกี่ยวข้อง

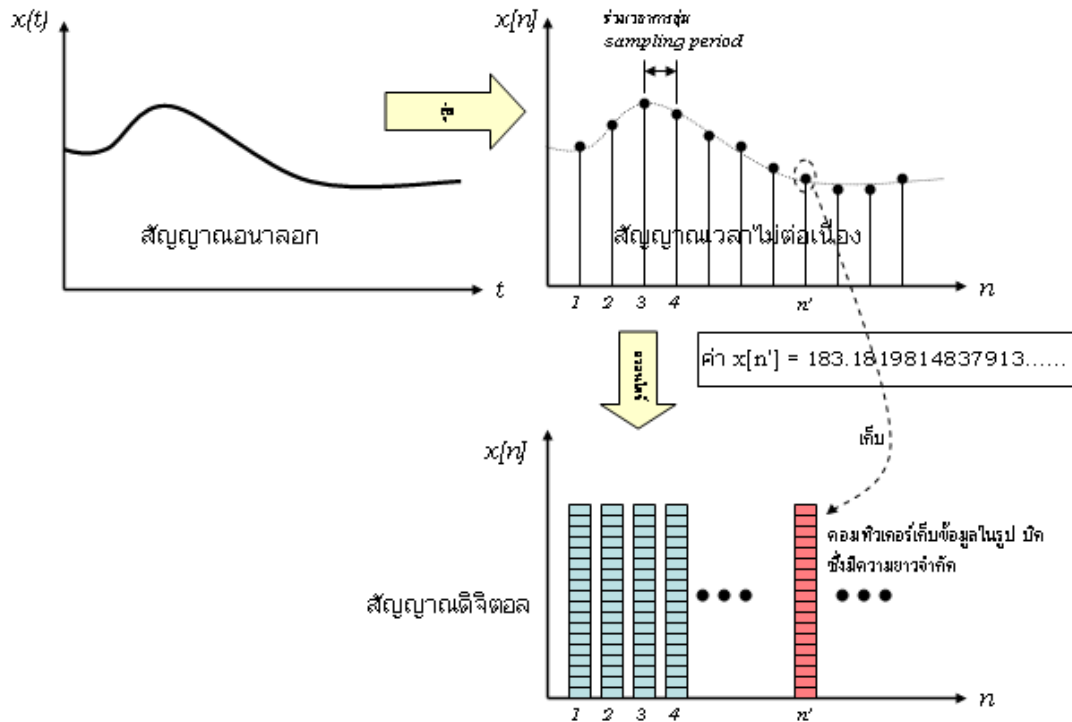
2.1 การแปลงสัญญาณจากอนาลอกเป็นดิจิตอล

คือระบบการแปลงสัญญาณ อนาลอกให้เป็นสัญญาณดิจิตอลซึ่งจะได้ค่าเท่ากับสัญญาณอนาลอก นั้นๆ

การแปลงสัญญาณจากอนาลอกเป็นดิจิตอลจะมีอุปกรณ์ที่ทำหน้าที่แปลงสัญญาณที่เรียกว่า A/D converter หรือ Analog to Digital converter โดยอุปกรณ์นี้จะตรวจจับสัญญาณที่ได้จากหัววัดสัญญาณ จากนั้นจะแปลงค่าต่างๆ ที่สุ่มได้เป็นค่านับวัดในระบบดิจิตอลตามระดับที่กำหนดด้วยค่า Bit ที่ใช้ เรียกการแปลงค่านี้ว่า Digitization และค่า Bit ที่ใช้ในการกำหนดจำนวนระดับของค่าดิจิตอลหรือรายละเอียดที่ต้องการเรียกว่า Bit depth ซึ่งเป็นค่าบ่งบอกความละเอียดของสัญญาณ (Spatial resolution)

2.1.1 Data sampling

การแปลงสัญญาณจากอนาลอกเป็นดิจิตอลจะถูกสุ่มค่าจากสัญญาณไฟฟ้าตามเวลาที่เปลี่ยนแปลงไปซึ่งตัวเลขสุ่มที่ได้แต่ละค่าจะกำหนดเป็นระดับค่าดิจิตอลตามระดับเทียบเท่ากับระบบ bit ที่ใช้ซึ่งหากสุ่มจำนวนที่เหมาะสมพอดีค่าดิจิตอลที่ได้เมื่อนำไปแปลงกลับเป็นสัญญาณอนาลอกได้สมบูรณ์มีรายละเอียดที่ครบถ้วน แต่หากสุ่มน้อยเกินไปสัญญาณอนาลอกที่ได้จะเกิดความคลาดเคลื่อนในการแปลง ทำให้สัญญาณอนาลอกผิดไปจากความเป็นจริง แต่ถ้าสุ่มจำนวนมากเกินไปแล้วเมื่อนำไปสร้างสัญญาณอนาลอกแม้จะได้สัญญาณที่มีรายละเอียดต่างๆ สมบูรณ์เหมือนสัญญาณต้นแบบแต่จะทำให้เวลาที่ใช้ในการประมวลผลมากเกินไป ซึ่งความถี่ที่ใช้สุ่มข้อมูล (Sampling frequency) ที่เหมาะสมเรียกว่า Nyquist frequency ซึ่งมีค่าเท่ากับ 2 เท่าของความถี่สัญญาณ input



ภาพที่ 2.1 การสุ่มสัญญาณ อนาลอก เพื่อให้ได้ สัญญาณไม่ต่อเนื่องทางเวลา

2.1.2 Aliasing

Sampling frequency เป็นความถี่ในการนับวัดสัญญาณที่เวลาใดๆ เพื่อแปลงค่าเป็น Digital signal จะต้องมีความถี่เป็น 2 เท่าหรือมากกว่าความถี่สูงสุด (f_{max}) ของสัญญาณ เรียกความถี่นี้ว่า Nyquist frequency ถ้า Sampling frequency มีค่าน้อยกว่า Nyquist frequency จะเกิดการสูญเสียสัญญาณมาก (S/N ต่ำ) สัญญาณอนาลอกที่ได้เกิดลักษณะที่เรียกว่า Aliasing artifact

2.1.3 Shannon sampling theorem

“ An analog signal containing components up to a maximum frequency f -Hz may be completely by regularly spaced samples of $2f$ or twice the signal frequency”

จากหลักการดังกล่าวจะพบว่าอัตราการนับวัดสัญญาณ (Sampling rate; T)

$$T = 1/2f$$

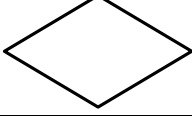
ความถี่ที่น้อยที่สุดในการเก็บข้อมูลคือ $2f$ (Nyquist frequency)

2.2 การออกแบบขั้นตอนการทำงานของโปรแกรม

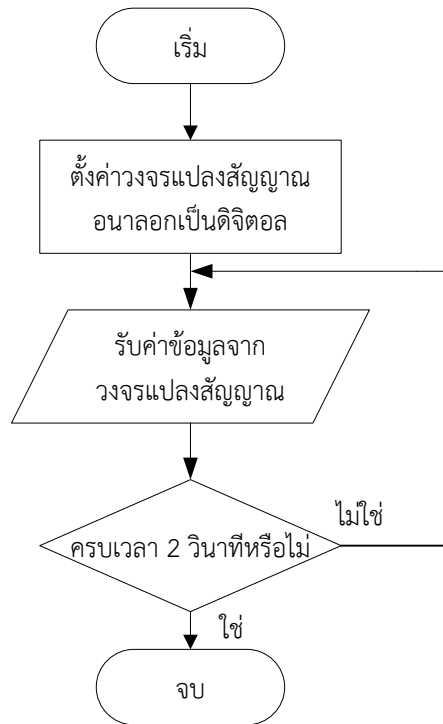
การออกแบบขั้นตอนการทำงานของโปรแกรมเป็นขั้นตอนที่สองการพัฒนาโปรแกรม ซึ่งจะให้ผลเป็นแนวทางการป้อนรหัสโปรแกรม ผู้ออกแบบขั้นตอนการทำงานของโปรแกรมอาจใช้เครื่องมือต่างๆ ช่วยในการออกแบบ เครื่องมือชนิดหนึ่งที่นิยมใช้ออกแบบโปรแกรม เช่น ผังงาน (Flow chart) เป็นต้น

ผังงาน (Flow chart) คือ วิธีการออกแบบโปรแกรมอย่างเป็นขั้นตอนและมีเหตุมีผล โดยการใช้สัญลักษณ์และ ตัวอักษรประกอบการอธิบายในแต่ละขั้นตอน เพื่อให้ผู้เขียนโปรแกรมและผู้สนใจทั่วไปสามารถเข้าใจความหมายได้ ดังตัวอย่างที่แสดงไว้ในตารางที่ 2.1

ตารางที่ 2.1 ตัวอย่างของสัญลักษณ์ต่างๆ ที่มีใช้ในผังงาน

สัญลักษณ์	รูปภาพ
	จุดเริ่มต้นหรือจุดจบของโปรแกรม (Terminal)
	ข้อมูลเข้า/ออก (Input/Output)
	การประมวลผล (Process)
	การตัดสินใจ (Decision หรือ Selection)
	ข้อมูลออก ทางจอภาพ (Display)
	ทิศทางของ ขั้นตอนการดำเนินงาน (Flow line)

ตัวอย่างผังงานของโปรแกรมการแยกพยางค์เสียงพูดภาษาไทยด้วยไมโครคอนโทรลเลอร์ ในส่วนของการนำสัญญาณเสียงเข้ามาเก็บไว้ในหน่วยความจำ จะแสดงไว้ในภาพที่ 2.1



ภาพที่ 2.2 ผังงานของการนำสัญญาณเสียงเข้ามาเก็บไว้ในหน่วยความจำ

2.2.1 การป้อนรหัสโปรแกรม

การป้อนรหัสโปรแกรมเป็นการนำผลลัพธ์ของการออกแบบโปรแกรม ไปเปลี่ยนให้เป็นโปรแกรมภาษาคอมพิวเตอร์ ภาษาใดภาษาหนึ่งผู้เขียนโปรแกรมจะต้องให้ความสนใจต่อรูปแบบคำสั่งและกฎเกณฑ์ ของภาษาที่ใช้เพื่อให้การประมวลผลเป็นไปตามผลลัพธ์ที่ได้ออกแบบไว้นอกจากนั้นผู้เขียนโปรแกรมควรแทรกคำอธิบายการทำงานต่างๆ ลงในโปรแกรมเพื่อให้โปรแกรมนั้น มีความกระชับ และง่ายต่อการตรวจสอบและโปรแกรมนี้อย่างยังเป็นส่วนหนึ่งของเอกสารประกอบ

2.2.2 การทดสอบและการแก้ไขโปรแกรม

การทดสอบโปรแกรมเป็นการนำโปรแกรมที่ป้อนรหัสแล้ว เข้าคอมพิวเตอร์เพื่อตรวจสอบรูปแบบกฎเกณฑ์ของภาษา และผลการทำงานของโปรแกรมนั้นถ้าพบว่ายังไม่ถูกต้องก็แก้ไขให้ถูกต้องต่อไป

2.2.3 การทำเอกสารประกอบโปรแกรม

การทำเอกสารประกอบโปรแกรมเป็นงานที่สำคัญของการพัฒนาโปรแกรม เอกสารประกอบโปรแกรมช่วยให้ผู้ใช้โปรแกรมเข้าใจ วัตถุประสงค์ ข้อมูลที่จะต้องใช้กับโปรแกรม ตลอดจนผลลัพธ์ที่จะได้จากโปรแกรม การพัฒนาโปรแกรมทุกโปรแกรมจึงควรต้องทำ เอกสารกำกับด้วยเสมอ

2.2.4 Keil uVision 3

คอมไพเลอร์ที่ใช้ในการพัฒนาคือ โปรแกรม Keil uVision 3 หรืออาจเรียกว่า Keil C51 เป็นคอมไพเลอร์ที่สร้างขึ้นโดยบริษัท Keil Software สามารถดาวน์โหลดได้ที่ <http://www.keil.com> ซึ่งทำให้ดาวน์โหลดเป็นเวอร์ชันทดลองใช้ โปรแกรม Keil uVision 3 จะช่วยให้เขียนโปรแกรมด้วยภาษา C กับไมโครคอนโทรลเลอร์ตระกูล MCS51 ได้ง่ายและรวดเร็ว โดยสามารถแปลงภาษา C51 เป็นโค้ด HEX ได้เลย

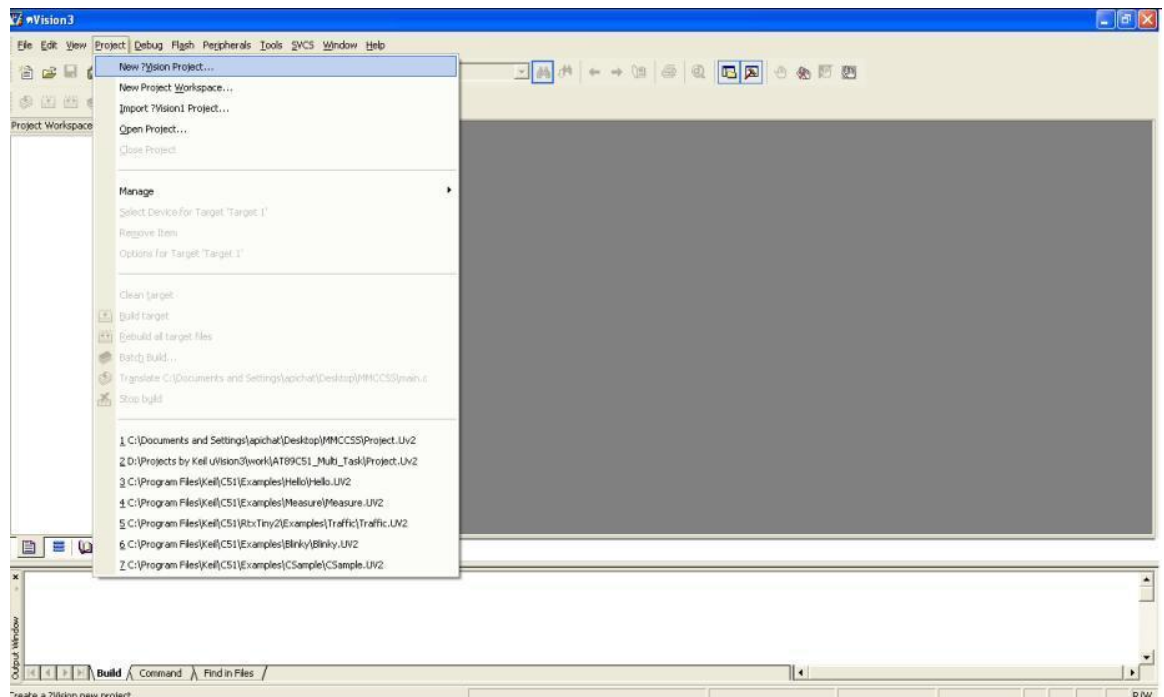


ภาพที่ 2.3 เป็นไอคอนของโปรแกรม Keil uVision 3

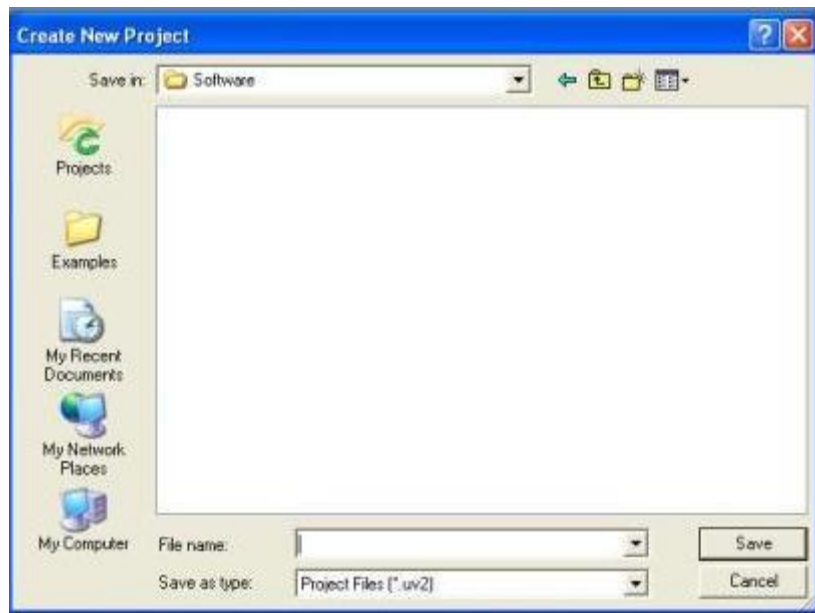
2.2.5 ขั้นตอนการพัฒนาโปรแกรมด้วย Keil uVision 3

ขั้นตอนที่ 1 การสร้างโปรเจ็ค

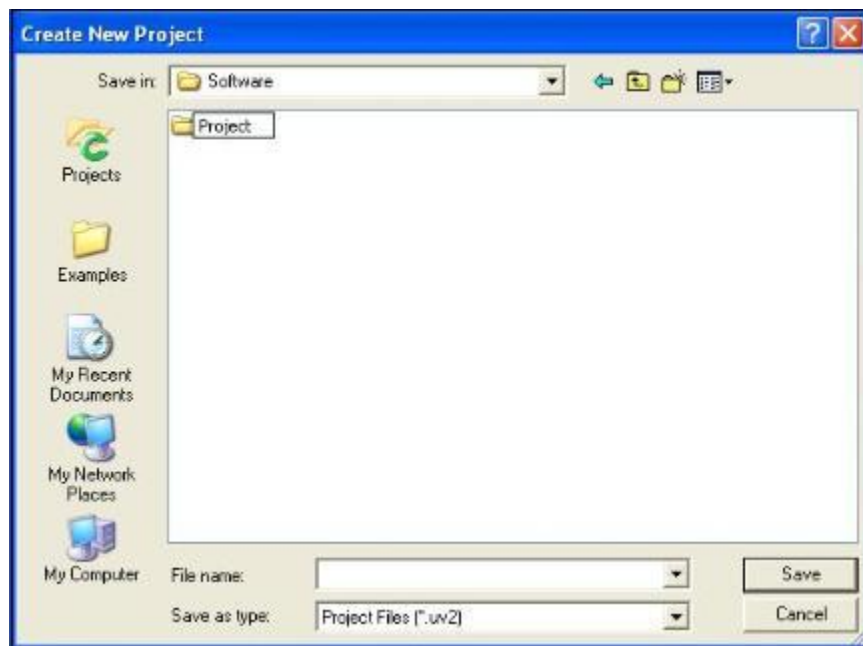
1. คลิกเมนู Project > New ?Vision Project



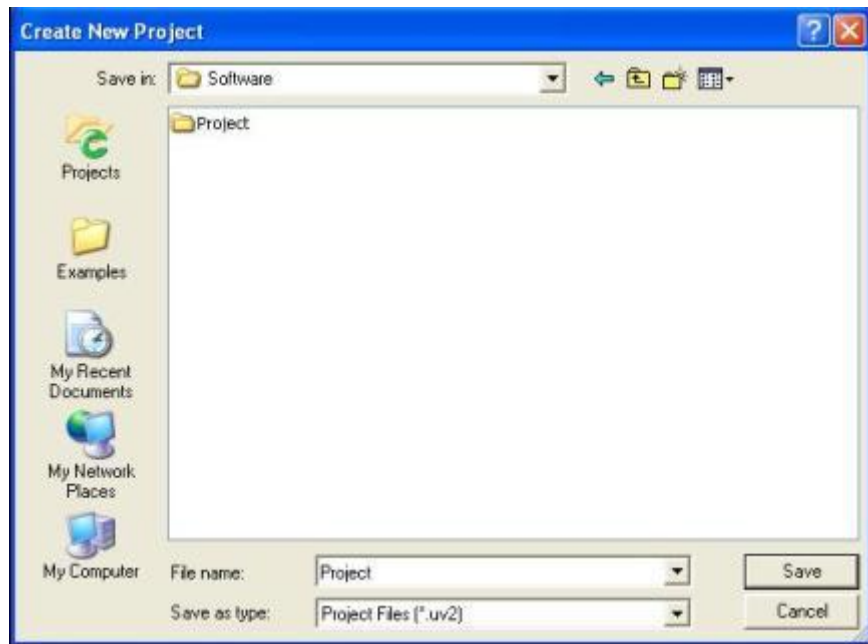
เมื่อเลือกคำสั่ง New ?Vision Project จะปรากฏหน้าต่าง Create New Project ขึ้นมา โดยหน้าต่าง Create New Project จะกำหนดให้เราใส่ชื่อโปรเจ็ค สำหรับใช้งาน ซึ่งในการทำงานอาจมีหลายโปรเจ็คเพื่อความสะดวกในการใช้งานควรสร้างโฟลเดอร์ (folder) ขึ้นมาใหม่แต่ละโปรเจ็คโดยเฉพาะ



2. คลิกปุ่ม  เพื่อสร้างโฟลเดอร์ใหม่ ในตัวอย่างตั้งชื่อโฟลเดอร์ว่า Project

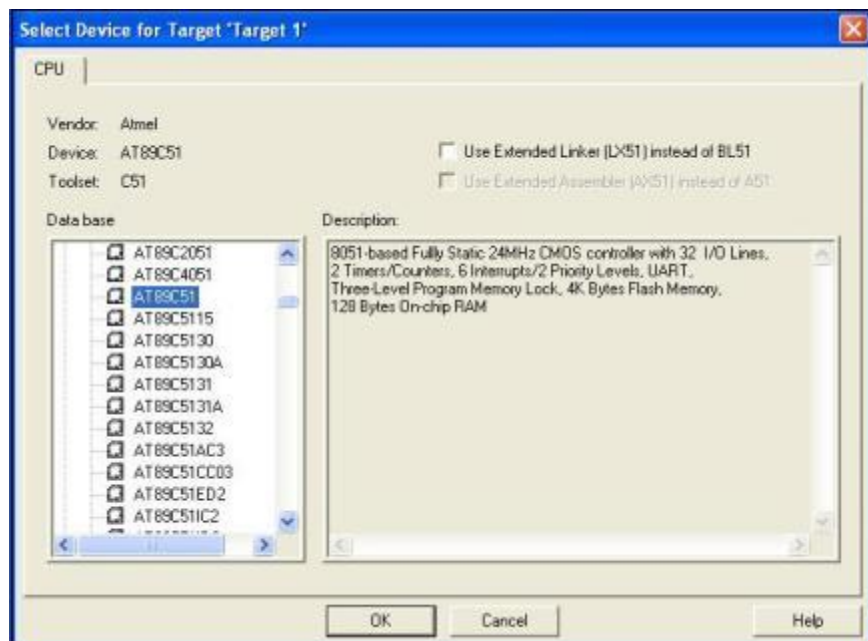


3. คลิกเข้าไปในโฟลเดอร์ที่สร้างใหม่ เพื่อสร้างโปรเจกต์ในโฟลเดอร์นั้น
4. กรอกชื่อโปรเจกต์ที่ช่อง File name:
5. คลิกปุ่ม  เพื่อบันทึกไฟล์โปรเจกต์



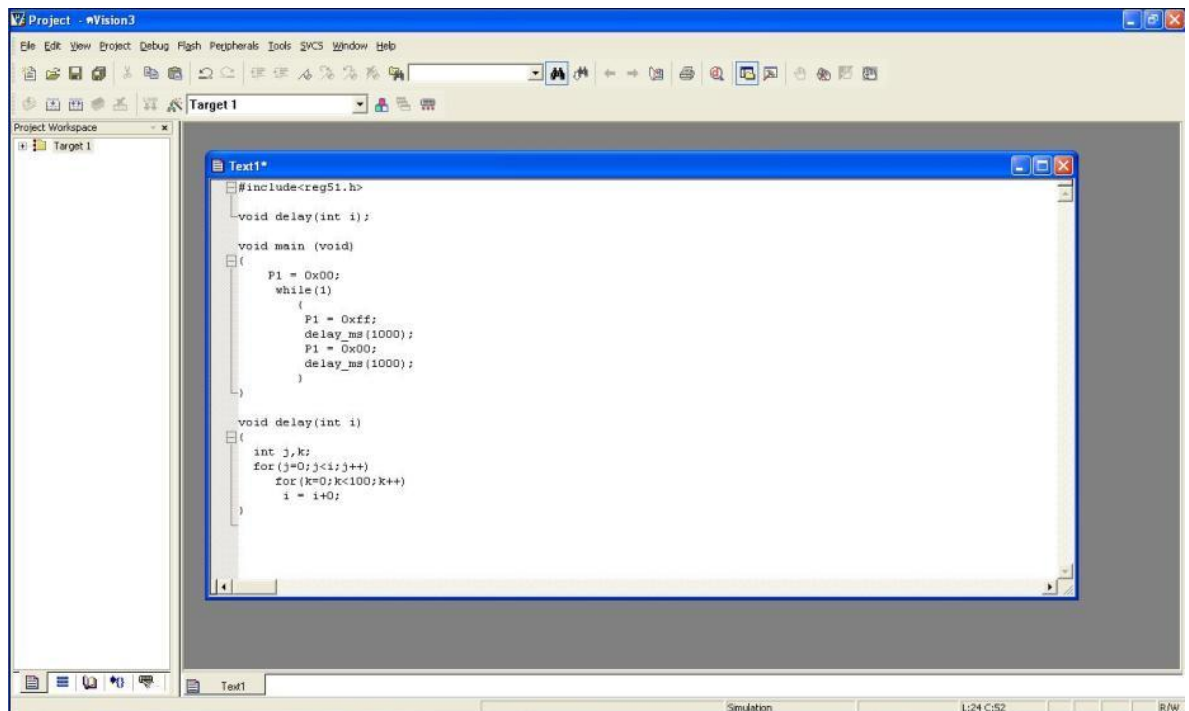
หลังจากที่บันทึกโปรเจกต์แล้ว โปรแกรม uVision 3 จะแสดงหน้าต่าง Select Device for Target 'Target1' ขึ้นมา เพื่อให้เราเลือก ชิปที่จะใช้งานจาก Device Database

6. เลือกชิปที่ต้องการ โดยในตัวอย่างเลือกชิป AT89C51 ของบริษัท ATMEL

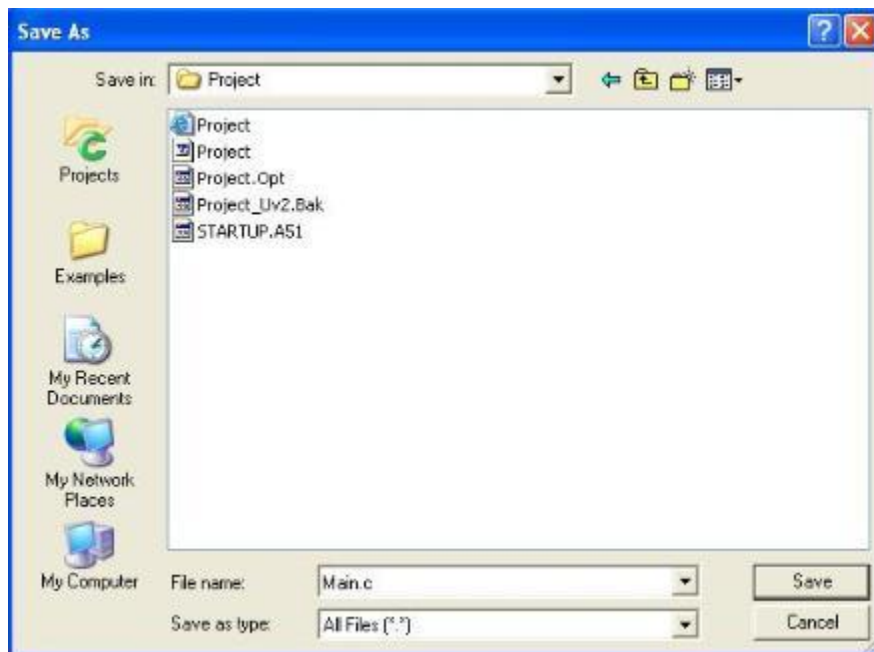


ขั้นตอนที่ 2 เขียนโค้ดโปรแกรม

1. คลิกเมนู File > New หรือคลิกปุ่ม ที่แถบเมนู เพื่อสร้างไฟล์ที่จะใช้เขียนโค้ดโปรแกรม
2. เขียนโค้ดโปรแกรมที่หน้าต่างว่างตรงกลางของโปรแกรม uVision 3 (หน้าต่าง Text 1*)



3. คลิกเมนู File > Save หรือคลิกปุ่ม เพื่อบันทึกโค้ดโปรแกรม ถ้าเป็นการบันทึกครั้งแรกโปรแกรม uVision 3 จะโชว์หน้าต่าง Save As ขึ้นมา เพื่อให้ตั้งชื่อไฟล์โค้ดโปรแกรม



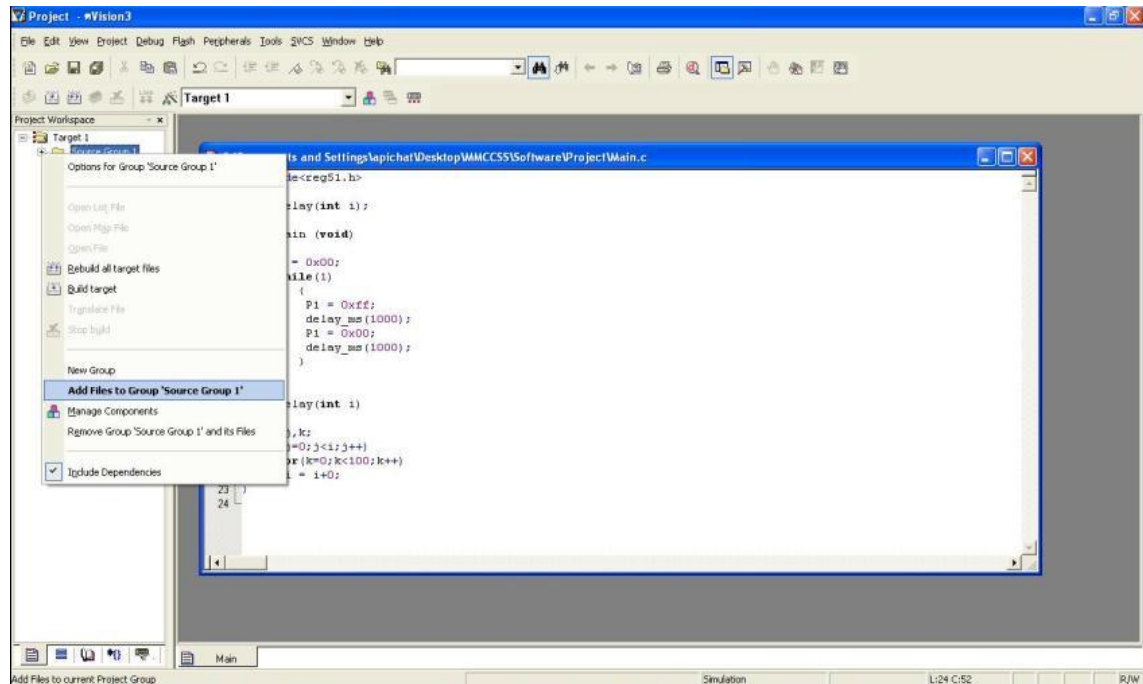
4. กรอกชื่อไฟล์ในช่อง File Name: โดยในตัวอย่างตั้งชื่อไฟล์เป็น Main.c

5. คลิกปุ่ม **Save** เพื่อบันทึกโปรแกรม

เมื่อสร้างโค้ดโปรแกรมเสร็จเรียบร้อยแล้ว จากนั้นให้เพิ่มไฟล์เข้าไปในโปรเจกต์ของเรา

6. คลิกขวาที่ Source Group1 ตรงหน้าต่าง Project Workspaces ถ้าหา Source Group1 ไม่เจอให้คลิกเครื่องหมาย + ข้างหน้า Target 1

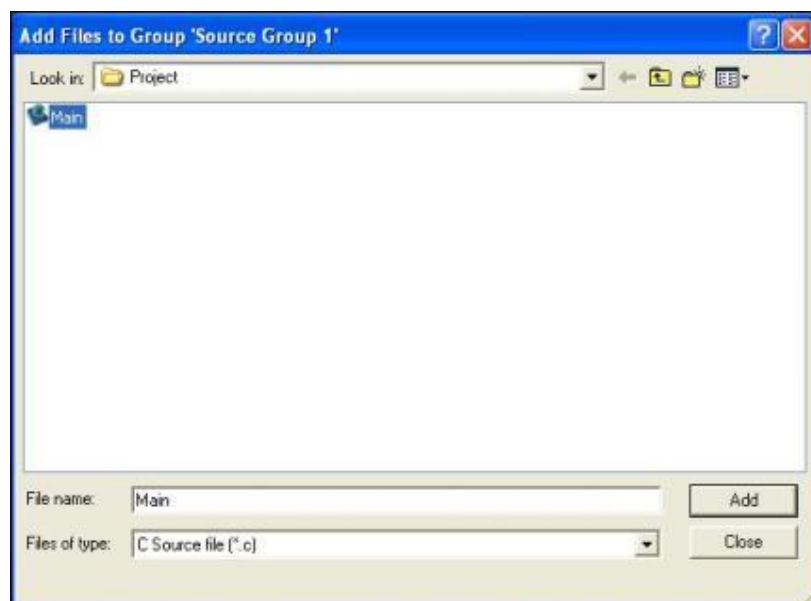
7. คลิกเลือกเมนู Add Files to Group 'Source Group1'



จากนั้นหน้าต่าง Add Files to Group 'Source Group1' จะปรากฏขึ้นมาให้เราเลือกไฟล์ที่จะเพิ่มเข้าไปในโปรเจกต์

8. คลิกเลือกไฟล์ที่จะเพิ่มเข้าไปในโปรเจกต์

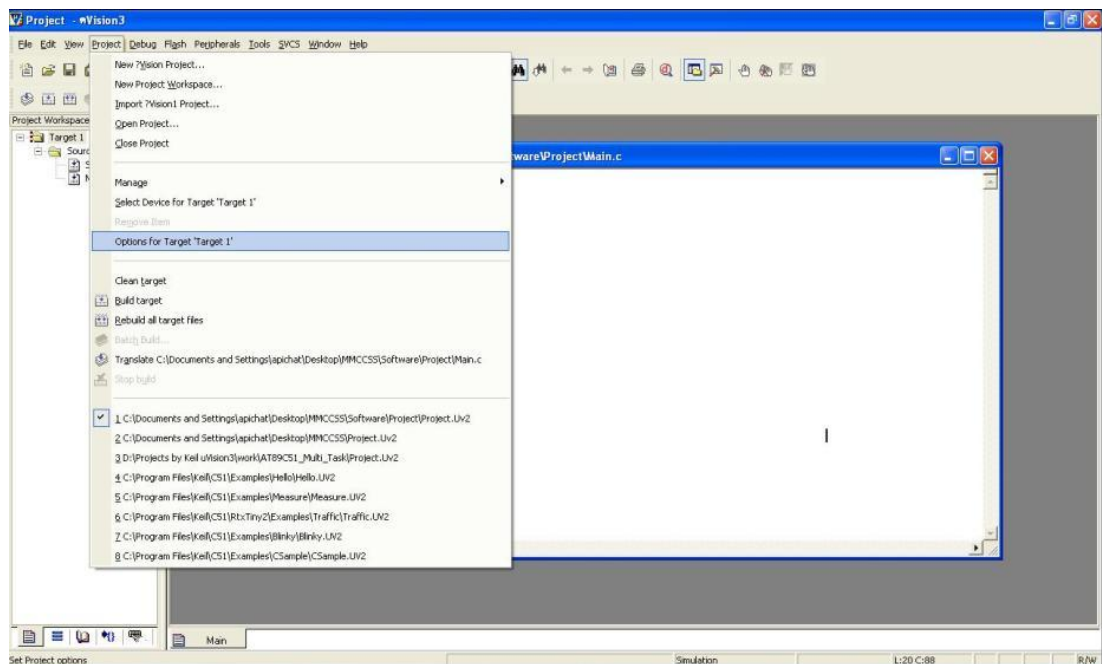
9. คลิกปุ่ม **Add** เพื่อเพิ่มไฟล์



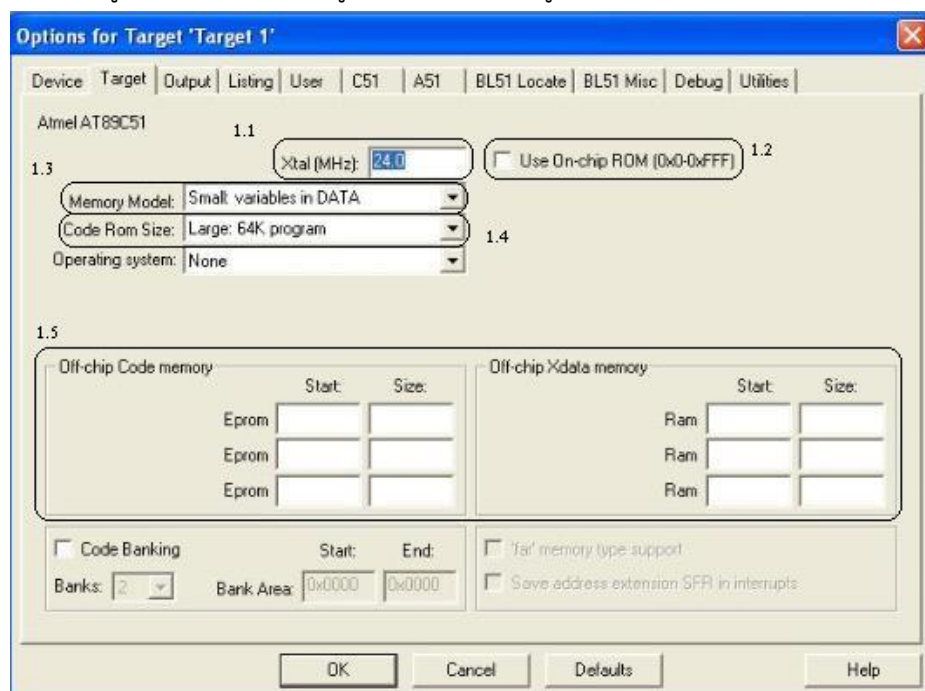
ขั้นตอนที่ 3 เลือก Tool Option สำหรับชิพ

โปรแกรม uVision 3 ให้เราสามารถกำหนดค่าพารามิเตอร์ต่างๆ ที่จำเป็นสำหรับชิพที่เราจะใช้งาน โดยผ่านทางหน้าต่าง Option for Target ก่อนอื่นต้องเลือกชิพที่ต้องการใช้งานเสียก่อน ดังแสดงต่อไปนี้

1. คลิกเมนู Project > Option for Target 'Target 1' เพื่อเปิดหน้าต่าง Option for Target 'Target1'



หน้าต่าง Option for Target 'Target 1' จะปรากฏขึ้นมาให้เรากำหนดพารามิเตอร์ต่างๆ ตามต้องการ (แถบเมนูในหน้าต่างจะต้องอยู่ที่ Target ถ้าไม่อยู่ให้คลิกแถบ Target)



การกำหนดพารามิเตอร์สำหรับชิพในส่วนตัวต่างๆ มีรายละเอียดดังแสดงต่อไปนี้

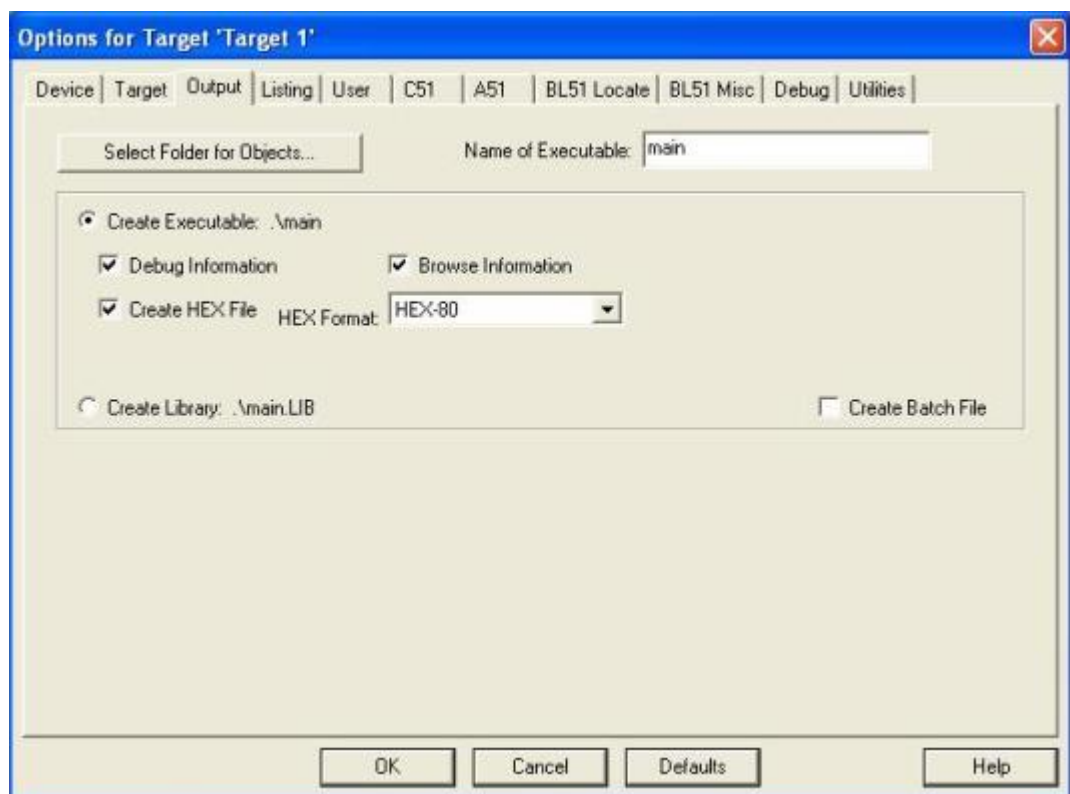
- 1.1 กำหนดสัญญาณนาฬิกาของชิพ ซึ่งจะมีค่าเดียวกับค่าความถี่ Xtal ที่ใช้
- 1.2 กำหนดว่าจะให้ใช้พื้นที่หน่วยความจำเก็บโปรแกรมภายในชิพหรือไม่ ปกติจะกำหนดให้ภายในชิปไว้อัตโนมัติอยู่แล้ว ซึ่งกำหนดภายในไฟล์

STARTUP.A51

- 1.3 กำหนด Memory Model หรือโมเดลพื้นที่หน่วยความจำ ปกติจะกำหนดเป็น Small
- 1.4 กำหนด Code Rom Size สำหรับขนาดพื้นที่หน่วยความจำที่จะเก็บโค้ดโปรแกรม ซึ่งเราอาจจะเก็บไว้สูงสุด 64K
- 1.5 กำหนดพื้นที่ใช้งานหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลไว้ภายนอกชิพทั้งหมด โดยให้เราใส่แอดเดรสเริ่มต้นและขนาดพื้นที่หน่วย

ความจำที่ใช้ และใช้งานได้ถูกต้องควรทำการ Enable การใช้หน่วยความจำภายนอกในไฟล์ STARTUP.A51 ด้วย

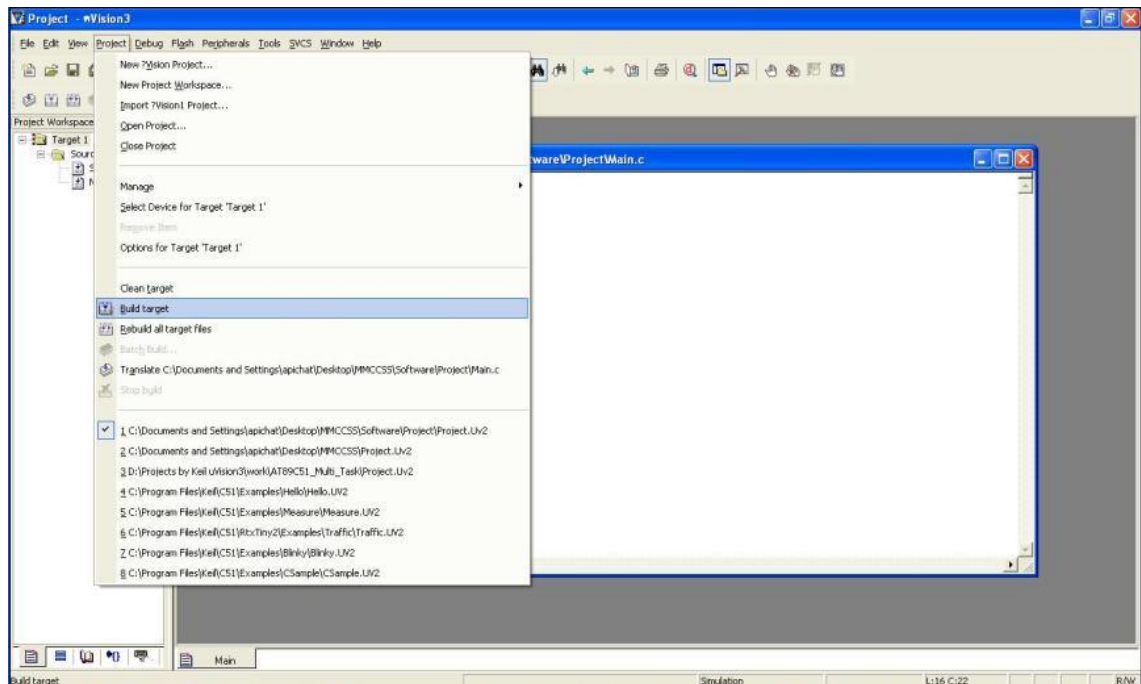
2. คลิกแถบ Output เพื่อกำหนดค่าเอาต์พุต
3. ใส่ชื่อไฟล์เอาต์พุต (ไฟล์นามสกุล.hex) ในช่อง Name of Executable เพื่อให้ง่ายควรใส่ชื่อให้เหมือนชื่อไฟล์โค้ดโปรแกรม ถ้าไม่กำหนด โปรแกรม uVision 3 จะตั้งชื่อเดียวกับชื่อโปรเจกต์
4. กำหนดว่าจะให้โปรแกรม uVision 3 สร้างไฟล์ Hex หรือไม่ ถ้าต้องการให้คลิกที่ช่องว่างข้างหน้า



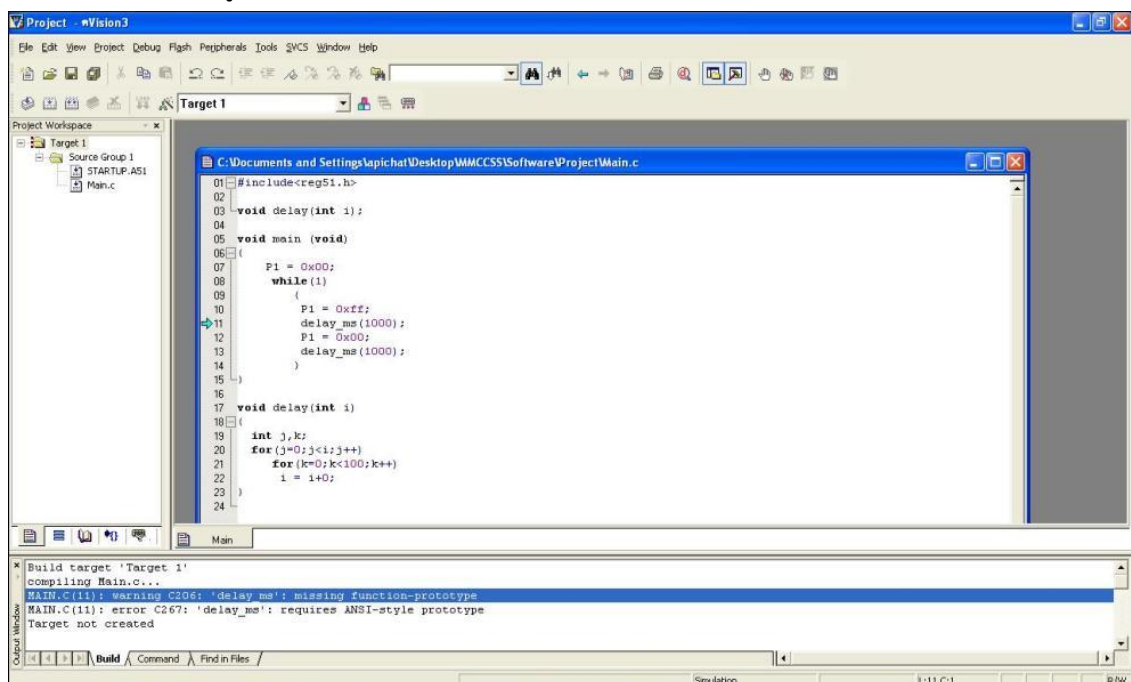
ขั้นตอนที่ 4 Build โปรเจ็คเพื่อให้ได้ไฟล์ Hex

ในขั้นตอนนี้เป็นการแปลงโค้ดโปรแกรมไปเป็นไฟล์ Hex ซึ่งมีขั้นตอนดังต่อไปนี้

1. คลิกเมนู Project > Build target หรือคลิกปุ่ม 



ถ้ามีข้อผิดพลาดเกิดขึ้นโปรแกรม uVision3 จะแสดงข้อผิดพลาดและข้อความเตือนให้รับทราบที่หน้า Build ของหน้าต่าง Output Window และถ้าดับเบิลคลิกบรรทัดที่แสดงข้อผิดพลาดก็จะกระโดดไปยังโค้ดโปรแกรมที่ผิดพลาดนั้นทันที จากภาพตัวอย่างแสดงว่าบรรทัดนั้นมีข้อผิดพลาดสังเกตได้จากเครื่องหมาย  ที่อยู่บรรทัดนั้น



จากตัวอย่างข้อผิดพลาดที่เกิดขึ้นก็คือ ฟังก์ชัน `delay_ms(1000)`; ไม่มีในการประกาศไว้ เมื่อแก้เป็น `delay(1000)`; ให้ถูกต้องแล้ว ให้ทำการ Build target เหมือนขั้นตอนที่ 1 อีกครั้ง โปรแกรม uVision 3 จะรายงานว่ามีข้อผิดพลาดเกิดขึ้น

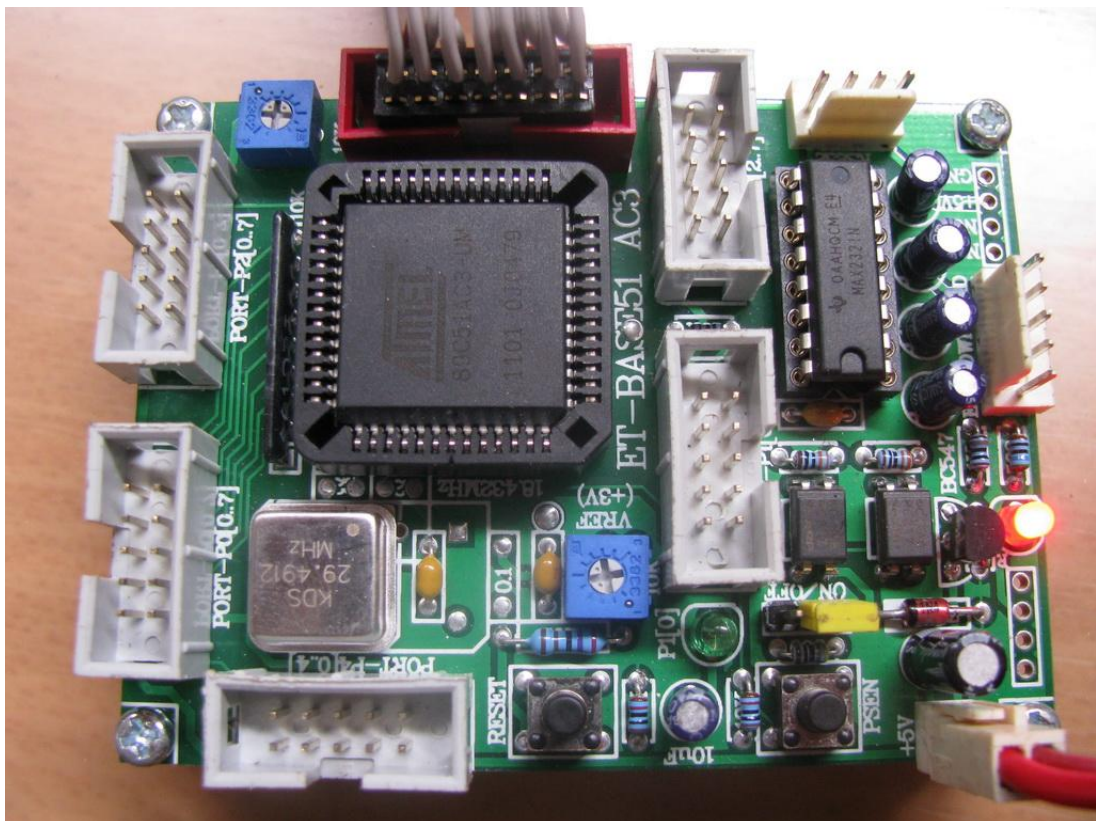
```
Build target 'Target 1'
compiling Main.c...
linking...
Program Size: data=9.0 xdata=0 code=78
creating hex file from "main"...
"main" - 0 Error(s), 0 Warning(s).
```

ผลลัพธ์ที่ได้ก็คือไฟล์ Hex หรือไฟล์นามสกุล .Hex นั้นเอง โดยไฟล์ Hex ที่ได้จะเก็บอยู่ในโฟลเดอร์โปรเจกต์ที่ใช้งานปัจจุบัน สามารถนำไฟล์นี้ไปโหลดเข้าชิปไมโครคอนโทรลเลอร์ได้เลย

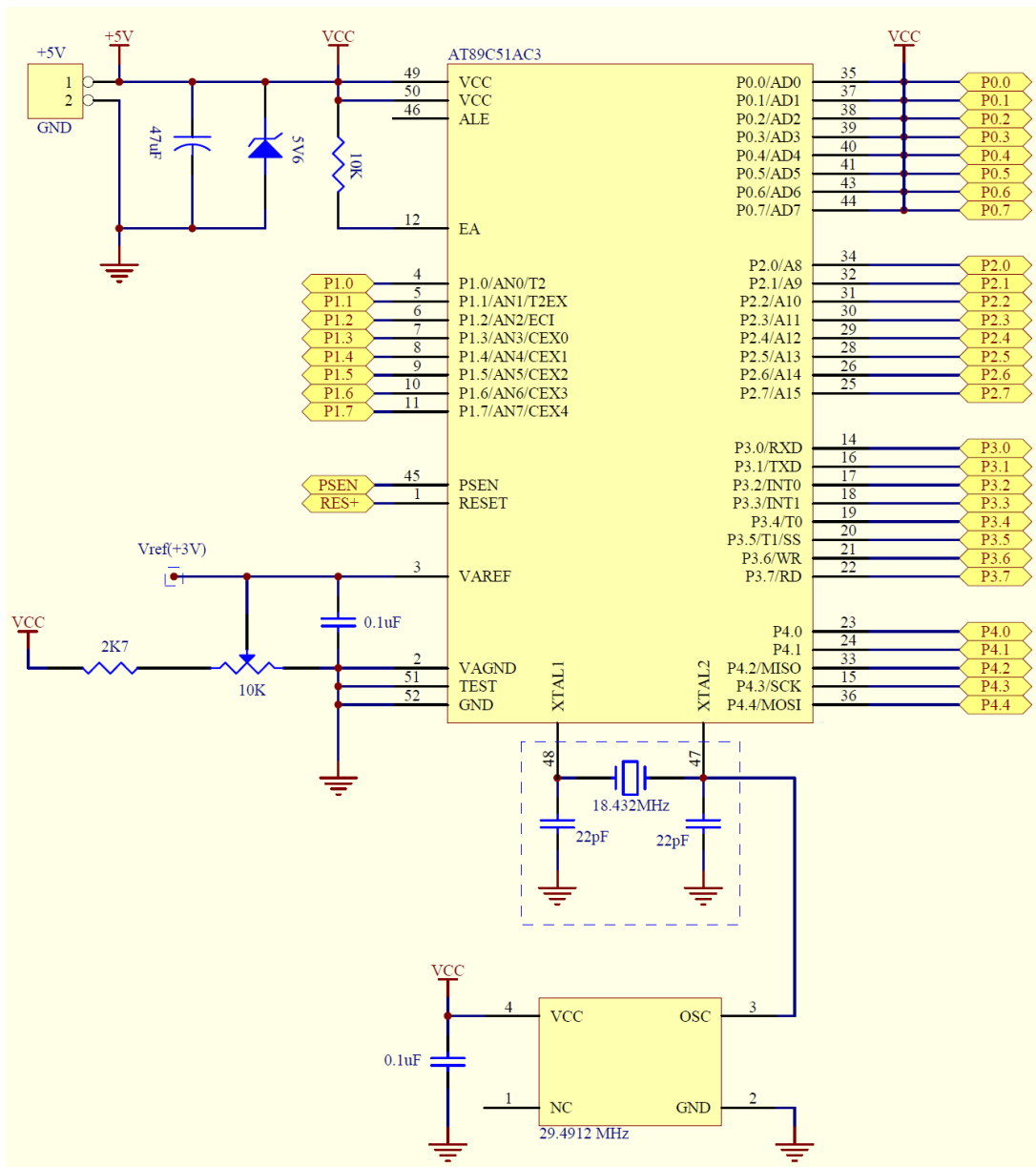
2.3 วงจรสำหรับการแยกพยางค์เสียงพูดภาษาไทยด้วยไมโครคอนโทรลเลอร์

2.3.1 วงจรบอร์ดประมวลผลหลัก

วงจรสำหรับการแยกพยางค์เสียงพูดภาษาไทยด้วยไมโครคอนโทรลเลอร์ในงานวิจัยนี้ ได้เลือกใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C51AC3 ซึ่งมีฟังก์ชันการทำงานสำหรับการแปลงสัญญาณอนาล็อกเป็นดิจิทัลในตัว จึงไม่ต้องมีวงจรส่วนนี้เพิ่มแต่อย่างใด วงจรที่ใช้แสดงดังรูปที่ 3



ภาพที่ 2.4 บอร์ดตัวประมวลผลที่ใช้ในการแยกพยางค์เสียงพูดภาษาไทยด้วยไมโครคอนโทรลเลอร์



ภาพที่ 2.5 วงจรส่วนตัวประมวลผลที่ใช้ในการแยกพยางค์เสียงพูดภาษาไทยด้วย ไมโครคอนโทรลเลอร์

2.3.2 คุณสมบัติของ MCU ตระกูล MCS51 เบอร์ AT89C51AC3 ของ ATMEL

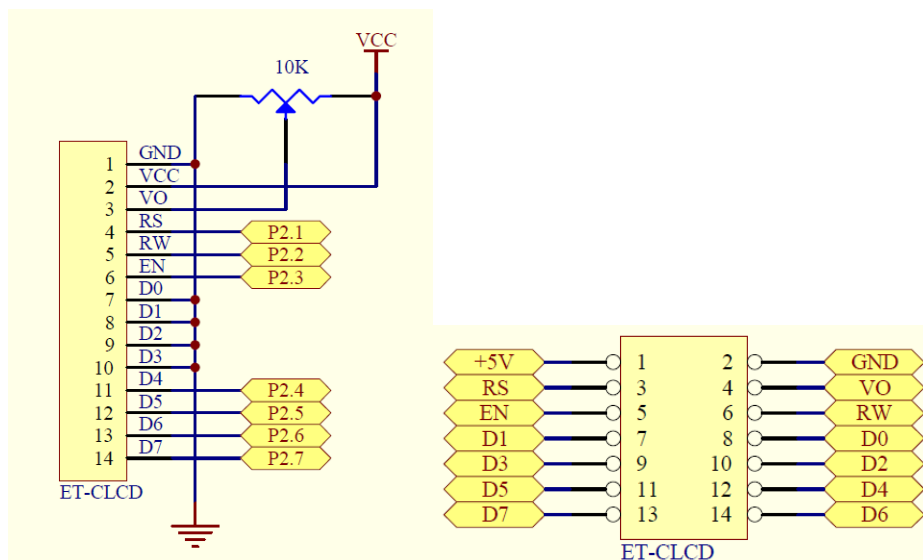
ใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ Oscillator Module ค่า 29.4912 MHz ซึ่งสามารถกำหนดการทำงานของ MCU ให้ทำงานในโหมดความเร็ว 2 เท่า (X2 Mode) ได้ ทำให้ MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 58.9824 MHz โดยคุณสมบัติเด่นๆของ MCU ได้แก่

- มีหน่วยความจำ Flash สำหรับเขียนโปรแกรมขนาด 64KByte
- มี EEPROM ขนาด 2KByte สำหรับเก็บข้อมูล และสามารถเขียนซ้ำได้กว่า 1 ล้านครั้ง
- มีพอร์ต I/O ขนาด 8 บิต จำนวน 5 พอร์ต (P0,P1,P2,P3 และ P4(5Bit))

- มี RAM ใช้งาน 2304 Byte (ERAM 2048 Byte + IRAM 256 Byte)
 - มีวงจรสื่อสารอนุกรม UART จำนวน 1 พอร์ต และมีวงจรสื่อสาร SPI จำนวน 1 พอร์ต
 - มีวงจร Timer/Counter ขนาด 16 บิต จำนวน 3 ชุด
 - มีวงจร ADC ขนาด 10บิต จำนวน 8 ช่อง (ใช้ Port-P1 โดยกำหนดจากโปรแกรม)
 - มีวงจร Watchdog, Power-ON Reset, Capture/Compare ,PWM
- ☐ มีขั้วต่อสัญญาณ I/O แบบ TTL แบบ Header 2x5 จำนวน 5 ชุด (P0,P1,P2,P3 และ P4)
 - ☐ มีขั้วต่อ LCD แบบ Header 2x7 รองรับการเชื่อมต่อกับ LCD Character (เชื่อมต่อแบบ 4 บิต)
 - ☐ มีขั้วต่อใช้งาน RS232 สำหรับใช้งาน และ ET-DOWNLOAD สำหรับ Download ผ่าน RS232
 - ☐ มี LED แสดงสถานะแหล่งจ่าย Power และ Self-Test สำหรับใช้ทดสอบการทำงานของบอร์ด
 - ☐ ใช้แหล่งจ่ายไฟขนาด +5VDC
 - ☐ ขนาด PCB Size เล็กเพียง 8 x 6 cm.

2.3.3 วงจรการแสดงผล

วงจรส่วนแสดงผลใช้ส่วนแสดงผลเป็นชนิด LCD ขนาด 20 ตัวอักษร 4 บรรทัด จะแสดงได้เฉพาะภาษาอังกฤษเท่านั้น พอร์ต CLCD ใช้กับ Character LCD เป็นการเชื่อมต่อแบบ 4 บิต โดยสัญญาณที่ใช้เชื่อมต่อกับ LCD จะเป็นสัญญาณชุดเดียวกับที่ต่อไปยังขั้วต่อของ PORT-P2 ในการเชื่อมต่อสายสัญญาณจากขั้วต่อของพอร์ต LCD ไปยังจอแสดงผล LCD นั้น ให้ยึดชื่อสัญญาณเป็นจุดอ้างอิง โดยให้ต่อสัญญาณที่มีชื่อตรงกันเข้าด้วยกันให้ครบทั้ง 14 เส้น



ภาพที่ 2.6 วงจรส่วนแสดงผล Character LCD

2.4 งานวิจัยที่เกี่ยวข้อง

ในการวิจัยครั้งนี้ ผู้วิจัยได้ศึกษางานวิจัยต่างๆ ที่เกี่ยวข้องเพื่อใช้เป็นแนวทางในการวิจัย โดยมีรายละเอียดดังต่อไปนี้

สัญญา เครือหงษ์ (2546 : บทคัดย่อ) วิจัยเรื่องการพัฒนาการแยกพยางค์จากข้อความภาษาไทยเพื่อการสังเคราะห์เสียงพูด การแยกพยางค์ของภาษาไทยเป็นกระบวนการหนึ่งเพื่อจัดเตรียมหน่วยเสียงที่เหมาะสม สำหรับการสังเคราะห์เสียงพูดจากข้อความ ในงานวิจัยนี้ผู้วิจัยได้เสนอการแยกพยางค์จากภาษาไทย โดยใช้กฎเพื่อเปรียบเทียบหาโครงสร้างพยางค์ที่เหมาะสม (Suited-Syllable-Structure-Mapping: 3S-Mapping) วิธีการนี้มีลักษณะที่สำคัญคือจะใช้โครงสร้างของพยางค์มาสร้างเป็นกฎ โดยกฎเหล่านี้จะเริ่มต้นจากการกำหนดโครงสร้างให้อยู่ในรูปแบบอย่างง่ายที่ประกอบด้วย C V และ T ซึ่ง แทน พยัญชนะ สระ และวรรณยุกต์ตามลำดับ เรียกว่า สัญลักษณ์อักขระ (Alphabet-Symbols) โครงสร้างเหล่านี้จะต้องทำงานร่วมกับการตรวจสอบกลุ่มอักขระไทย (Enhanced-Thai-Character- Cluster: ETCC) การพัฒนาด้วยวิธีการดังกล่าวจะเริ่มจากการแทนอักขระแต่ละตัวที่นำเข้ามาด้วย สัญลักษณ์อักขระ สิ่งที่ได้จากกระบวนการนี้จะถูกนำไปเปรียบเทียบกับกฎที่สร้างไว้และตรวจสอบ โครงสร้างที่เหมาะสมด้วยการตรวจสอบกลุ่มอักขระไทย นอกจากนี้งานวิจัยนี้ยังได้ใช้การแยก พยางค์ด้วยพจนานุกรมด้วยวิธีการเปรียบเทียบคำตามความยาวที่เหมาะสม (Suited-Length-Word Mapping: SL-word Mapping) เพื่อแยกพยางค์ที่อยู่นอกเหนือจากกฎที่ออกแบบไว้ จากวิธีดังกล่าว ในการทดลองแสดงผลให้เห็นว่าการแยกพยางค์ด้วยวิธีนี้สามารถแยกพยางค์ได้ความถูกต้องถึง 99.85 % จากตัวอย่างทั้งหมด 4,000 ข้อความ

เฉลิมวุฒิ ไวชนะ (2548: บทคัดย่อ) วิจัยเรื่องการชี้เฉพาะคำสำคัญเสียงพูดภาษาไทยบนพื้นฐานของการตรวจสอบหน่วยเริ่มและหน่วยตามของพยางค์ วัตถุประสงค์เพื่อพัฒนากฎวิธีในการชี้เฉพาะคำสำคัญเสียงพูดภาษาไทยแบบไม่ขึ้นกับผู้พูดโดยการประยุกต์ใช้หลักการของแบบจำลองฮิดเดนมาร์คอฟ ซึ่งเป็นแบบจำลองของเสียงระดับย่อยของพยางค์ ใช้วิธีการตรวจสอบหน่วยเริ่มและหน่วยตามของพยางค์ มีจำนวนคำสำคัญ 70 คำแบ่งออกเป็น 3 ชุดตามจำนวนคือ 20 คำ 40 คำและ 70 คำเสียงพูดที่นำมาเป็นต้นแบบและเป็นแบบทดสอบประกอบด้วยเสียงผู้ชาย 30 คนและเสียงผู้หญิง 20 คน ผลการทดสอบค่า FOM ของระบบการชี้เฉพาะแบบไม่ขึ้นกับผู้พูดมีอัตราเฉลี่ยร้อยละ สำคัญ 54.45 โดยค่า FOM เฉพาะชุดคำสำคัญ 20 คำมีค่าเฉลี่ยร้อยละ 64.59 ค่า FOM เฉพาะชุดคำสำคัญ 40 คำมีค่าร้อยละ 51.11 และค่า FOM เฉพาะชุดคำสำคัญ 70 คำมีค่าร้อยละ 47.65 ตามลำดับ

จรัสแสง ผิวอ่อน และ ชาญวิทย์ ตั้งสิริวรกุล (2545 : บทคัดย่อ) ทำวิจัยโปรแกรมฝีกออกเสียงสำหรับคนหูหนวก โครงงานนี้แบ่งการทำงานของโปรแกรมฝีกออกเสียงสำหรับคนหูหนวก ซึ่งเขียนด้วยโปรแกรม MATLAB สามารถแบ่งเป็น 3 ส่วนคือภาคอ่านไฟล์เสียงพูด ภาครู้จำเสียงพูด ภาคประมวลผลและแสดงผล โดยเสียงคำพูดเป็นแบบคำโดด (Isolated Speech) เสียงพูด

จะถูกบันทึกในรูปแบบเวฟ (Wave Format) โดยผ่านทางไมโครโฟนที่ต่ออยู่กับการ์ดเสียง (Sound Card) ด้วยความถี่แซมปลิง 11,025 Hz และความละเอียด 16 บิตต่อแซมปลิง (Bits/Sample) จากนั้นได้นำข้อมูลวิเคราะห์หาความถูกต้องของเสียงพูด โดยขั้นแรกคือภาคหาจุดเริ่มต้นและจุดสุดท้ายของเสียงคำพูด โดยอ่านจากไฟล์เสียงพูด เพื่อนำเอาเฉพาะข้อมูลตรงส่วนที่เป็นสัญญาณเสียงคำพูดไปวิเคราะห์หาพารามิเตอร์ของเสียงพูด จากนั้นนำสัญญาณเสียงมาแปลงด้วยฟูรีเยร์แบบเร็ว และลดจำนวนข้อมูลลง เพื่อให้เหมาะสมกับการนำไปใช้งานในภาครัฐจำเสียงพูดต่อไป โดยแบ่งแกนความถี่ของสเปกตรัมของสัญญาณเสียงออกเป็นช่วง ๆ และทำการหาค่าเฉลี่ยของแอมพลิจูดในแต่ละช่วงความถี่นั้น ๆ โดยค่าเฉลี่ยที่ได้นี้คือ ค่าพารามิเตอร์ที่ใช้แทนลักษณะเฉพาะของเสียงที่ต้องการที่มีทั้งหมด 46 จุด แล้วนำไปสู่กระบวนการตัดสินใจเสียงคำพูด โดยใช้นิวรัลเน็ตเวิร์คที่ได้รับการสอนแล้วอ่านค่าน้ำหนักการเชื่อมต่อ และใช้สเปกตรัมของเสียงคำพูดเป็นรูปแบบอินพุต จากนั้นใช้ส่วนการตัดสินใจแบบ logsig-moid มาใช้ในการตัดสินใจว่าเป็นเสียงคำพูดมีความถูกต้องหรือไม่พร้อมทั้งแสดงผลทางหน้าจอคอมพิวเตอร์ที่สร้างด้วยโปรแกรม MATLAB ในตอนท้ายได้มีการทดสอบการแยกแยะเสียง noise แต่ละเน็ตเวิร์คโดยมีการปรับค่าอัตราการสอนที่ 0.5 ได้ความถูกต้อง 64.5% และผลการทดสอบแต่ละเน็ตเวิร์คโดยมีการปรับค่าอัตราการสอนที่ 0.5 สำหรับผู้ทดสอบคนเดิมได้ผลเฉลี่ย 81.4% และผู้ทดสอบที่ไม่ได้ทำการสอนได้ผลเฉลี่ย 65.5% จากผลการทดลองดังกล่าวได้ผลเป็นที่น่าพอใจ