	ใบเนื้อหา		หน้าที่ 1
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พิชณนิตบูลิน และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

## หน่วยที่ 2 พิชณนิตบูลิน และการลดรูปสมการ

### ภาษาซีกับไมโครคอนโทรลเลอร์


#### 1. โครงสร้างของการเขียนโปรแกรมภาษาซี

ภาษาซีถือได้ว่าเป็นภาษาโปรแกรมที่มีความยืดหยุ่นในการเขียนโปรแกรมสูง และมีความเร็วในการประมวลผลที่ใกล้เคียงกับภาษาแอสเซมบลี โดยการเขียนโปรแกรมภาษาซีจะมีรูปแบบของโครงสร้างในการเขียนโปรแกรมอยู่ 2 รูปแบบใหญ่ดังนี้


1. โครงสร้างของการเขียนโปรแกรมภาษาซีรูปแบบที่ 1

```
#include <headerfile.h>           //Preprocessor Directives
void function()                   //Subroutine Function
{
    .....;
}
void main()                       //Main Function
{
    .....;
}
```

โครงสร้างของการเขียนโปรแกรมภาษาซีรูปแบบที่ 1 นี้จะประกอบด้วยส่วนแรกๆที่เรียกว่า Preprocessor Directives ซึ่งเป็นการอินคลูทไฟล์ที่เป็นนามสกุล .h หรือ .c เข้ามาในโปรแกรม เพื่อให้ผู้เขียนโปรแกรมเรียกใช้งานฟังก์ชันที่ผู้ออกแบบคอมพิวเตอร์ได้สร้างฟังก์ชันสำเร็จไว้รองรับให้เรียบร้อยแล้ว ส่วนในการเขียนโปรแกรมภาษาซีสำหรับไมโครคอนโทรลเลอร์ส่วนนี้ที่ขาดไม่ได้ก็จะเป็นในส่วนของไฟล์ที่เก็บรวบรวมรีจิสเตอร์พิเศษและรีจิสเตอร์ควบคุมของไมโครคอนโทรลเลอร์ตระกูลนั้น ๆ ส่วนไฟล์อื่น ๆ ที่จะทำการอินคลูทเอาเข้ามาเพิ่มเติมก็จะขึ้นอยู่กับอุปกรณ์ที่ไมโครคอนโทรลเลอร์ทำการเชื่อมต่อ หรือถ้าเป็นระบบคอมพิวเตอร์ก็จะเปรียบได้กับการเพิ่ม Library ของ Driver ของอุปกรณ์ ส่วนต่อมาที่ต่อจาก Preprocessor Directives ก็จะเป็น Subroutine Function ซึ่งก็คือส่วนของโปรแกรมย่อยที่ผู้พัฒนาโปรแกรมเขียนขึ้นเองเพื่อให้ทำหน้าที่อย่างใดอย่างหนึ่ง โดยอาจจะถูกเรียกใช้งานจากฟังก์ชันย่อยอื่น ๆ หรือฟังก์ชันหลักก็ได้ และส่วนสุดท้ายคือ Main Function ซึ่งในหนึ่งโปรแกรมจะมี Main Function ได้เพียงฟังก์ชันเดียวเท่านั้น ส่วนข้อกำหนดอื่นที่ไม่ได้ระบุในรูปแบบการเขียนโปรแกรมในโครงสร้างของการเขียนโปรแกรมภาษาซีรูปแบบที่ 1 ก็คือการประกาศตัวแปร โดยการประกาศตัวแปรเราสามารถประกาศใช้งานได้ทุกที่ แต่ความหมายของการใช้งานอาจจะต่างกัน ซึ่งจะได้อีกครั้งหนึ่งในเรื่องของการประกาศตัวแปร

	ใบเนื้อหา	หน้าที่ 2
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิชณนิตบูลิน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์
<div>2. โครงสร้างของการเขียนโปรแกรมภาษาซีรูปแบบที่ 2</div> <pre>#include &lt;headerfile.h&gt; //Preprocessor Directives void function();          // Function Prototype void main()                //Main Function {     .....; } void function()            //Function Body {     .....; }</pre> <p>โครงสร้างของการเขียนโปรแกรมภาษาซีรูปแบบที่ 2 นี้จะแตกต่างจากรูปแบบที่ 1 ในส่วนของ Subroutine Function โดย Subroutine Function จะถูกแบ่งออกเป็น 2 ส่วน โดยส่วนแรกจะเรียกว่า Function Prototype ซึ่งจะต้องเขียนขึ้นมาในลักษณะที่คล้ายกับการประกาศตัวแปรและจะต้องเขียนก่อน Main Function เท่านั้น ส่วนที่ 2 ก็คือส่วนของ Subroutine Function ทั้งหมดที่เรียกว่า Function Body ซึ่งจะเขียนต่อท้ายจาก Main Function</p> <p>การเขียนโปรแกรมโดยใช้โครงสร้างของการเขียนโปรแกรมภาษาซีในรูปแบบที่ 1 ส่วนใหญ่จะนิยมเขียนโปรแกรมภาษาซีที่ใช้งานกับไมโครคอนโทรลเลอร์ และสามารถเพิ่มความซับซ้อนของโปรแกรมได้ด้วยการนำฟังก์ชันที่เป็น Subroutine Function ทั้งหมดเขียนเป็นไฟล์ใหม่ขึ้นมาแล้วบันทึกเป็นไฟล์นามสกุล .c แล้วทำการอินคลูดไฟล์เข้ามาใช้งานในส่วนของ Preprocessor Directives ส่วนการเขียนโปรแกรมโดยใช้โครงสร้างของการเขียนโปรแกรมภาษาซีในรูปแบบที่ 1 ส่วนใหญ่จะนิยมเขียนโปรแกรมภาษาซีที่ใช้งานกับคอมพิวเตอร์เพราะสามารถดัดแปลงแก้ไขเพื่อสร้างเป็นไฟล์ Library หรือ Driver ของอุปกรณ์ได้โดยการสร้างไฟล์ใหม่ขึ้นมาอีก 2 ไฟล์ คือ ไฟล์นามสกุล .c คือไฟล์ที่เก็บ Subroutine Function หรือ Function Body ทั้งหมด ส่วนไฟล์ต่อมาคือ ไฟล์นามสกุล .h คือไฟล์ที่เก็บ Function Prototype ทั้งหมด เมื่อต้องการนำมาใช้งานก็ทำการอินคลูดไฟล์เข้ามาในส่วนของ Preprocessor Directives</p>

	ใบเนื้อหา		หน้าที่ 3
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พิชณนิบูลีน และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

2. ตัวแปร (Variable)

ตัวแปร (variable) คือสิ่งที่ผู้พัฒนาโปรแกรมกำหนดหรือสร้างขึ้นมาตามหลักการเขียนโปรแกรมของภาษานั้น ๆ เพื่อใช้ในการเก็บข้อมูลในลักษณะต่าง ๆ โดยการสร้างตัวแปรเพื่อใช้งานจะเป็นการจองพื้นที่หน่วยความจำข้อมูล หรือหน่วยความจำโปรแกรม ตามชนิดของตัวแปรที่ผู้พัฒนาสร้างขึ้น

**กฎในการตั้งชื่อตัวแปร**

1. ต้องขึ้นต้นด้วยตัวอักษร ตัวต่อไปจะเป็นตัวอักษรหรือตัวเลขก็ได้
2. ห้ามใช้สัญลักษณ์อื่นใด ยกเว้น \$ และขีดล่าง
3. ตัวอักษรพิมพ์เล็ก และพิมพ์ใหญ่มีความหมายต่างกัน
4. ห้ามเว้นวรรคระหว่างตัวแปร
5. ห้ามตั้งชื่อซ้ำกับคำสงวน

**รูปแบบของการประกาศตัวแปร**

```
type    name;
```


โดย type = ชนิดของข้อมูล  
name = ชื่อของตัวแปร


ตารางที่ 1.1 ตารางชนิดข้อมูลของตัวแปร

ชนิดตัวแปร	จำนวนบิต	ค่าข้อมูลที่เก็บได้
char	8	-128 ถึง 127
unsigned char	8	0 ถึง 255
int	16	-32768 ถึง 32767
unsigned int	16	0 ถึง 65535
long	32	-2147483648 ถึง 2147483648
unsigned long	32	0 ถึง 4294967295
float	32	3.4E-38 ถึง 3.4E+38
double	64	1.7E-308 ถึง 1.7E+308
bit (MCS-51)	1	0 ถึง 1

สำหรับการเขียนโปรแกรมภาษาซี MCS-51 ด้วยโปรแกรม Keil uVision Compiler ชนิดของข้อมูลสามารถกำหนดได้อีก 6 รูปแบบ คือ

1. sbit ใช้สำหรับประกาศตัวแปรที่อ้างถึงแหล่งข้อมูลระดับบิตในพื้นที่ SFR
2. sfr ใช้สำหรับประกาศตัวแปรขนาด 1 byte ที่แทนตำแหน่งหน่วยความจำ SFR
3. bdata ใช้สำหรับประกาศตัวแปรขนาด 1 byte ที่มีพื้นที่อยู่บนหน่วยความจำ RAM ที่สามารถเข้าถึงข้อมูลได้ในระดับบิต
4. data ใช้สำหรับประกาศตัวแปรที่มีพื้นที่อยู่บนหน่วยความจำ RAM ที่ใช้งานทั่วไป

	ใบเนื้อหา	หน้าที่ 4
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิชณนิตยลีน และการลดรูปสมการ	
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์		
<p>5. xdata ใช้สำหรับประกาศตัวแปรที่มีพื้นที่อยู่บนหน่วยความจำ RAM ภายนอก</p> <p>6. code ใช้สำหรับประกาศตัวแปรที่ใช้พื้นที่หน่วยจำโปรแกรม</p> <p>และสำหรับการเขียนโปรแกรมภาษาซีด้วย Compiler AvrGCC ชนิดของข้อมูลที่ยอมรับใช้งานสามารถกำหนดได้อีก 3 รูปแบบ คือ</p> <ol style="list-style-type: none"><li>1. uint8_t คือการประกาศตัวแปรแบบ unsigned char</li><li>2. uint16_t คือการประกาศตัวแปรแบบ unsigned int</li><li>3. uint32_t คือการประกาศตัวแปรแบบ unsigned long</li></ol> <p>การประกาศตัวแปรแบบปกติที่ได้กล่าวข้างต้นเป็นการใช้งานตัวแปรแบบทั่วไปที่ใช้สำหรับการเขียนโปรแกรมปกติทั่ว ๆ ไป แต่ในกรณีที่มีการประกาศตัวแปรเพื่อใช้งานแบบพิเศษ จะต้องมีความกำกับขึ้นต้นก่อนการประกาศตัวแปร ซึ่งมักจะพบเจอในการเขียนโปรแกรมของภาษาซีสำหรับไมโครคอนโทรลเลอร์ประมาณ 3 รูปแบบได้แก่</p> <ol style="list-style-type: none"><li>1. การระบุค่า extern โดยมีรูปแบบ extern type name; หมายถึงตัวแปรแบบโกลบอลที่ทุกฟังก์ชันสามารถมองเห็น และเรียกใช้งานได้ แต่จะถูกประกาศไว้ในไฟล์อื่น ๆ ซึ่งการระบุ extern ไว้หน้าตัวแปรเพื่อทำการอ้างอิงไปยังตัวแปรนั้น ๆ ในไฟล์อื่น เช่น extern unsigned pinIO8; เป็นต้น</li><li>2. การระบุค่า static โดยมีรูปแบบ static type name; หมายถึงตัวแปรที่ประกาศแบบโลคอล หรือโกลบอล ก็ได้ โดยตัวแปรแบบ static จะถูกสร้างเมื่อโปรแกรมเริ่มทำงาน และจะถูกทำลายเมื่อโปรแกรมจบการทำงาน (จบการทำงานของโปรแกรม ไม่ใช่จบการทำงานของฟังก์ชัน) ดังนั้นเมื่อโปรแกรมกลับมาทำงานที่ฟังก์ชันเดิมอีกครั้ง ค่าของตัวแปรที่เป็นค่าสุดท้ายของการประมวลผลก็จะยังคงอยู่ ส่วนใหญ่จะใช้กับตัวแปรแบบโลคอล</li><li>3. การระบุค่า volatile โดยมีรูปแบบ volatile type name; หมายถึงตัวแปรที่ประกาศเพื่อให้ Compiler รู้ว่าตัวแปรนี้ มีการเปลี่ยนแปลงค่าอยู่ตลอดเวลา และทุก ๆ ที่ในโปรแกรม ทั้งจากโปรแกรมหลัก โปรแกรมย่อย หรือ Interrupt เพื่อป้องกันไม่ใช้ Compiler มาทำการลดขนาดหรือทำลายค่าข้อมูลของตัวแปรให้มีค่าเป็นศูนย์ ซึ่งส่วนใหญ่สำหรับการเขียนโปรแกรมภาษาซีของไมโครคอนโทรลเลอร์จะใช้นี้สำหรับตัวแปรที่ใช้งานกับฟังก์ชันของการ Interrupt</li></ol> <p><b>ชนิดของตัวแปร</b></p> <p>ตัวแปรจะแบ่งออกเป็น 2 ชนิดใหญ่ ๆ ได้แก่</p> <ol style="list-style-type: none"><li>1. Global variable คือตัวแปรที่ประกาศไว้ภายนอกฟังก์ชันทุกฟังก์ชัน ซึ่งฟังก์ชันอื่น ๆ สามารถเรียกใช้งานตัวแปรแบบนี้ได้</li><li>2. Local variable คือ ตัวแปรที่ประกาศภายในฟังก์ชันหลัก หรือฟังก์ชันย่อย การใช้งานจะสามารถใช้งานได้เฉพาะฟังก์ชันที่ประกาศไว้เท่านั้น</li></ol> <p><b>ตัวอย่างการประกาศตัวแปร</b></p> <p>char a,b=5; //คือการประกาศตัวแปรที่มีการแก้ค่าข้อมูลได้ในช่วง -128 ถึง 127 ชื่อ a และ b โดย b มีค่าข้อมูลเริ่มต้นเท่ากับ 5</p>		

	ใบเนื้อหา		หน้าที่ 5
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พิชณนิตบูลิน และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

3. การเขียนฟังก์ชัน

การเขียนโปรแกรมฟังก์ชันย่อในภาษาซีจะมีรูปแบบและองค์ประกอบในการเขียนดังนี้

```
function-type function-name (Argument Variable)
{
    type variable;
    statement instruction;
    return value;
}
```

โดย function-type หมายถึงชนิดข้อมูลของฟังก์ชันเมื่อมีการคืนค่ากลับมาเมื่อฟังก์ชันที่ถูกเรียกใช้งานประมวลผลเรียบร้อยแล้ว ซึ่งชนิดข้อมูลของฟังก์ชันจะเหมือนกับชนิดข้อมูลของตัวแปร แต่จะมีในกรณีที่ฟังก์ชันไม่มีการคืนค่ากลับ ในส่วนของ function-type จะระบุด้วยคำว่า void

function-name หมายถึงชื่อของฟังก์ชันที่ต้องการจะสร้าง ซึ่งใช้ข้อกำหนดเดียวกับการประกาศตัวแปรในการตั้งชื่อ

Argument Variable หมายถึงส่วนที่ใช้ในการประกาศตัวแปรเพื่อรับค่าข้อมูลเข้ามาประมวลผลในฟังก์ชัน ซึ่งสามารถประกาศตัวแปรเพื่อรับค่าข้อมูลเข้ามาประมวลผลได้มากกว่า 1 ตัว

type variable; คือส่วนที่ใช้สำหรับการประกาศตัวแปรแบบ Local variable ที่ใช้สำหรับฟังก์ชันนั้น ๆ

statement instruction; คือส่วนสำหรับการเขียนชุดคำสั่งภาษาซีสำหรับใช้ในการประมวลผลของฟังก์ชันนั้น ๆ


return value; ในส่วนนี้จะเขียนก็ต่อเมื่อฟังก์ชันในส่วนของ function-type มีการระบุชนิดข้อมูลของฟังก์ชันที่ไม่ใช่ void


จากรูปแบบและองค์ประกอบในการเขียนฟังก์ชันดังกล่าวในโปรแกรมภาษาซีสามารถจำแนกรูปแบบการเขียนฟังก์ชันได้ 4 รูปแบบดังนี้

3.1 ฟังก์ชันที่ไม่มีการให้ค่าเมื่อทำการเรียกใช้งานและไม่มีการคืนค่าเมื่อออกจากการทำงานของฟังก์ชัน


รูปแบบในการเขียน

```
void function-name( )
{
    type variable;
    statement instruction;
}
```

	ใบเนื้อหา		หน้าที่ 6
	ชื่อวิชา	ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย		
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์			
<div>3.2 ฟังก์ชันที่มีการส่งผ่านค่าเมื่อทำการเรียกใช้งานและไม่มีการคืนค่าเมื่อออกจากการทำงานของฟังก์ชัน</div> <div>รูปแบบในการเขียน</div> <div>void function-name(Argument Variable)</div> <div>{</div> <div>type variable;</div> <div>statement instruction;</div> <div>}</div> <div>3.3 ฟังก์ชันที่ไม่มีการส่งผ่านค่าเมื่อทำการเรียกใช้งานและมีการคืนค่าข้อมูลเมื่อออกจากการทำงานของฟังก์ชัน</div> <div>รูปแบบในการเขียน</div> <div>function-type function-name( )</div> <div>{</div> <div>type variable;</div> <div>statement instruction;</div> <div>return value;</div> <div>}</div> <div>หรือ</div> <div>function-type function-name (void)</div> <div>{</div> <div>type variable;</div> <div>statement instruction;</div> <div>return value;</div> <div>}</div> <div>3.4 ฟังก์ชันที่มีการส่งผ่านค่าเมื่อทำการเรียกใช้งานและมีการคืนค่าข้อมูลเมื่อออกจากการทำงานของฟังก์ชัน</div> <div>รูปแบบในการเขียน</div> <div>function-type function-name( Argument Variable)</div> <div>{</div> <div>type variable;</div> <div>statement instruction;</div> <div>return value;</div> <div>}</div>			

	ใบเนื้อหา		หน้าที่ 7																								
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2																								
	ชื่อหน่วย พิชณนิตบูลีน และการลดรูปสมการ																										
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์																											
<div>ตัวอย่างในการเขียนฟังก์ชันย่อย</div> <pre>int sum(int a,int b){ //สร้างฟังก์ชันชื่อว่า sum เมื่อมีการเรียกใช้งานจะต้องให้ค่าฟังก์ชัน 2 ค่า และ //เมื่อฟังก์ชันประมวลผลเรียบร้อยแล้วจะมีการคืนค่าข้อมูลออกมาเป็นชนิด int int c; //สร้างตัวแปรชื่อว่า c ให้เก็บค่าข้อมูลชนิด int c = a + b; //ให้นำค่าข้อมูลของตัวแปร a มาบวกกับค่าของข้อมูล b ผลลัพธ์ที่ได้เก็บไว้ที่ //ตัวแปร c return c; //ออกจากฟังก์ชันพร้อมกับคืนค่าข้อมูลของตัวแปร c ให้แก่ระบบที่เรียกใช้ //ฟังก์ชัน sum }</pre>																											
4. ตัวดำเนินการ และนิพจน์คณิตศาสตร์																											
<p>การเขียนโปรแกรมคือการเขียนภาษาโปรแกรมให้จัดการกับข้อมูลที่ผู้พัฒนาต้องการในลักษณะทางด้านคณิตศาสตร์ ตรรกะทางลอจิก การให้ค่าข้อมูล การเปรียบเทียบข้อมูล ผ่านตัวดำเนินการต่าง ๆ เพื่อให้ได้เอาต์พุตอย่างใดอย่างหนึ่ง โดยการเขียนโปรแกรมภาษาซีก็เช่นเดียวกัน โดยจะแบ่งลักษณะของการประมวลผลข้อมูลด้วยการกระทำต่าง ๆ ดังนี้</p>																											
4.1 ตัวดำเนินการนิพจน์ทางคณิตศาสตร์ (Arithmetic Operation)																											
<p>ตัวดำเนินการนิพจน์ทางคณิตศาสตร์ (Arithmetic Operation) คือการประมวลผลข้อมูลทางด้านคณิตศาสตร์ของตัวแปรผ่านตัวดำเนินการนิพจน์ทางคณิตศาสตร์ โดยตัวดำเนินการนิพจน์ทางคณิตศาสตร์ของภาษาซีแสดงดังรูปตารางที่ 1.1</p>																											
<table><tr><th>ตัวดำเนินการ</th><th>ความหมาย</th><th>ตัวอย่าง</th></tr><tr><td>+</td><td>บวก (addition)</td><td>x + y</td></tr><tr><td>-</td><td>ลบ(subtraction)</td><td>x - y</td></tr><tr><td>*</td><td>คูณ(multiplication)</td><td>x * y</td></tr><tr><td>/</td><td>หาร(division)</td><td>x / y</td></tr><tr><td>%</td><td>หารเอาผลลัพธ์เฉพาะเศษ</td><td>x % y</td></tr><tr><td>++</td><td>เพิ่มค่าครั้งละ 1</td><td>x++</td></tr><tr><td>--</td><td>ลดค่าครั้งละ 1</td><td>x--</td></tr></table>				ตัวดำเนินการ	ความหมาย	ตัวอย่าง	+	บวก (addition)	x + y	-	ลบ(subtraction)	x - y	*	คูณ(multiplication)	x * y	/	หาร(division)	x / y	%	หารเอาผลลัพธ์เฉพาะเศษ	x % y	++	เพิ่มค่าครั้งละ 1	x++	--	ลดค่าครั้งละ 1	x--
ตัวดำเนินการ	ความหมาย	ตัวอย่าง																									
+	บวก (addition)	x + y																									
-	ลบ(subtraction)	x - y																									
*	คูณ(multiplication)	x * y																									
/	หาร(division)	x / y																									
%	หารเอาผลลัพธ์เฉพาะเศษ	x % y																									
++	เพิ่มค่าครั้งละ 1	x++																									
--	ลดค่าครั้งละ 1	x--																									
รูปที่ 1.1 รูปตารางแสดงตัวดำเนินการนิพจน์ทางคณิตศาสตร์ของภาษาซี																											



	ใบเนื้อหา	หน้าที่ 8
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิชณนิตบูลีน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

#### 4.2 ตัวดำเนินการเปรียบเทียบ (Comparative Operation)

ตัวดำเนินการเปรียบเทียบ (Comparative Operation) คือการประมวลผลข้อมูลในลักษณะของการเปรียบเทียบค่าข้อมูลในตัวแปรกับค่าข้อมูลคงที่ หรือค่าข้อมูลของตัวแปรกับค่าข้อมูลของอีกตัวแปรหนึ่งผ่านตัวดำเนินการเปรียบเทียบ ซึ่งจะใช้กับชุดคำสั่งของภาษาซี เช่น คำสั่ง if() , for() , while และ do-while() โดยตัวดำเนินการเปรียบเทียบของภาษาซีแสดงดังรูปตารางที่ 1.2

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
>	มากกว่า	$x > y$
<	น้อยกว่า	$x < y$
>=	มากกว่าหรือเท่ากับ	$x \geq y$
<=	น้อยกว่าหรือเท่ากับ	$x \leq y$
==	เท่ากับ	$x == y$
!=	ไม่เท่ากับ	$x != y$

รูปที่ 1.2 รูปตารางแสดงตัวดำเนินการเปรียบเทียบของภาษาซี


#### 4.3 ตัวดำเนินการทางตรรกะ (Logical Operation)

ตัวดำเนินการทางตรรกะ (Logical Operation) คือการประมวลผลข้อมูลให้ได้ผลลัพธ์ในลักษณะของลอจิก ซึ่งจะมีการใช้งานอยู่ 2 ลักษณะ โดยลักษณะแรกจะเป็นการดำเนินการทางตรรกะเพื่อต้องการได้ผลลัพธ์เพื่อนำไปใช้ในขบวนการตัดสินใจ เช่นนำไปใช้งานร่วมกับชุดคำสั่ง if(),for(),while() และ do-while() เป็นต้น ดังรูปตารางที่ 1.3 ส่วนการใช้งานในลักษณะที่สองจะเป็นการดำเนินการกับข้อมูลในลักษณะลอจิก และจะมีผลต่อชุดของข้อมูลนั้น ๆ ที่นำมากระทำกันผ่านรูปแบบของตัวดำเนินการทางตรรกะดังตารางที่ 1.2

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
&&	และ(and)	$mark \geq 80 \&\& mark \leq 100$
	หรือ(or)	$score < 0    score > 100$
!	ไม่(not)	$!x \&\& !y$

รูปที่ 1.3 รูปตารางแสดงตัวดำเนินการทางตรรกะที่ใช้งานร่วมกับชุดคำสั่ง if(),for(),while() และ do-while()



	ใบเนื้อหา		หน้าที่ 9
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พิชณนิตบูลีน และการลดรูปสมการ		

### ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

ตารางที่ 1.2 ตารางตัวดำเนินการทางตรรกะของภาษาซีเพื่อใช้ในการแปลงค่าข้อมูล


ตัวดำเนินการ	ความหมาย	ตัวอย่าง
&	And Data	A & 0xff
	Or Data	A   0x03
! หรือ ~	Not Data	!A หรือ ~A
^	Xor Data	A ^ 0xff


#### 4.4 ตัวดำเนินการกำหนดค่า


ตัวดำเนินการกำหนดค่า คือ สัญลักษณ์ทางคณิตศาสตร์ที่มีความหมายในการให้ค่าข้อมูล หรือ หมายถึงสัญลักษณ์ทางด้านซ้ายมือของตัวดำเนินการกำหนดค่า มีค่าเท่ากับการกระทำ หรือข้อมูลที่อยู่ทางด้านขวามือ โดยการเขียนโปรแกรมภาษาซี นอกจากจะหมายถึงสัญลักษณ์เครื่องหมาย “=” แล้วยังสามารถที่จะลดรูปเครื่องหมาย “=” ให้ใช้งานร่วมกับตัวดำเนินการทางคณิตศาสตร์และลอจิกได้ดังรูปตารางที่ 1.4

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
=	กำหนดค่าให้เท่ากับ	x = y
+=	การเพิ่มค่า	x += y มาจาก x = x + y
-=	การลบค่า	x -= y มาจาก x = x - y
*=	การคูณ	x *= y มาจาก x = x * y
/=	หาร ได้ผลลัพธ์จำนวนเต็ม	x /= y มาจาก x = x / y
%=	การหาร ได้ผลลัพธ์เศษ	x %= y มาจาก x = x % y
&=	ดำเนินการ	x &= y มาจาก x = x & y
=	ดำเนินการ	x  = y มาจาก x = x   y
^=	ดำเนินการ	x ^= y มาจาก x = x ^ y
<<=	การเลื่อนบิตไปทางซ้าย	x <<= 2 มาจาก x = x << 2
>>=	การเลื่อนบิตไปทางขวา	x >>= 2 มาจาก x = x >> 2


รูปตารางที่ 1.4 รูปตารางแสดงการใช้งานตัวดำเนินการกำหนดค่าของภาษาซีในรูปแบบต่าง ๆ

	ใบเนื้อหา		หน้าที่ 10
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พิชณนิทูลีน และการลดรูปสมการ		
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์			
<p><b>5. คำสั่งพื้นฐานในเขียนโปรแกรมภาษาซี</b></p> <p>การเขียนโปรแกรมภาษาซีเพื่อประมวลผลข้อมูลในรูปแบบต่าง ๆ จะเป็นการเขียนโปรแกรมโดยใช้ตัวดำเนินการทางคณิตศาสตร์ และลอจิก เพื่อกระทำกับข้อมูลเพื่อให้ได้ผลลัพธ์ที่ผู้พัฒนาต้องการ โดยการประมวลผลข้อมูลนอกจากจะใช้รูปแบบทางทางคณิตศาสตร์ และลอจิกแล้ว ก็จำเป็นที่จะต้องมีการตัดสินใจในการประมวลผล ซึ่งการเขียนโปรแกรมภาษาซีได้จัดชุดคำสั่งพื้นฐานในการเขียนโปรแกรมเพื่อประมวลผลข้อมูลรวมกับการดำเนินการทางคณิตศาสตร์ และลอจิกไว้ 2 กลุ่มดังนี้</p> <p><b>5.1 คำสั่งการตรวจสอบเงื่อนไข</b></p> <p>คำสั่งการตรวจสอบเงื่อนไข คือชุดคำสั่งพื้นฐานในการเขียนโปรแกรมภาษาซีเพื่อใช้ในการตัดสินใจซึ่งจะกระทำร่วมกับตัวดำเนินการเปรียบเทียบและตัวดำเนินการทางตรรกะในลักษณะแรก ซึ่งคำสั่งการตรวจสอบเงื่อนไขจะแบ่งออกเป็น 2 กลุ่มดังนี้</p> <p><b>5.1.1 กลุ่มคำสั่งตรวจสอบเงื่อนไขในรูปแบบคำสั่ง if()</b></p> <p><b>5.1.1.1 คำสั่งตรวจสอบเงื่อนไข 1 ทางเลือก</b></p> <p>รูปแบบ if ( เงื่อนไข )</p> <pre>{     ประโยคคำสั่งเมื่อเงื่อนไขเป็นจริง; }</pre> <p>ตัวอย่าง</p> <pre>a = 3;          //ให้ตัวแปร a มีค่าเท่ากับ 3 if(a &gt;= 6){     //เงื่อนไขจะเป็นจริงเมื่อตัวแปร a มีค่ามากกว่าหรือเท่ากับ 6     a += 3;     //ถ้าเงื่อนไขเป็นจริงตัวแปร a จะมีค่าเท่ากับ a + 3 }              //จบบล็อกเงื่อนไขของ if a = a + 3;     //เมื่อประมวลผลบล็อกคำสั่ง if แล้ว ค่าของตัวแปร a จะมีค่าเท่ากับ a + 3</pre> <p>จากตัวอย่างจะได้ผลลัพธ์ค่าตัวแปร a สุดท้ายคือ 6</p> <p><b>5.1.1.2 คำสั่งตรวจสอบเงื่อนไข 2 ทางเลือก</b></p> <p>รูปแบบ if ( เงื่อนไข )</p> <pre>{     ประโยคคำสั่งเมื่อเงื่อนไขเป็นจริง; } else {     ประโยคคำสั่งเมื่อเงื่อนไขเป็นเท็จ; }</pre>			

	<b>ใบเนื้อหา</b>	หน้าที่ 11
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิชณนิตบูลิน และการลดรูปสมการ	
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์		
<p>ตัวอย่าง</p> <pre> a = 3;           //ให้ตัวแปร a มีค่าเท่ากับ 3 if(a &gt;= 6){      //เงื่อนไขจะเป็นจริงเมื่อตัวแปร a มีค่ามากกว่าหรือเท่ากับ 6     a += 3;      //ถ้าเงื่อนไขเป็นจริงตัวแปร a จะมีค่าเท่ากับ a + 3 }else{           //เงื่อนไขเป็นเท็จให้ทำชุดคำสั่งในบล็อกของ else     a += 2;      //เงื่อนไขเป็นเท็จตัวแปร a จะมีค่าเท่ากับ a + 2 }               //จบบล็อกเงื่อนไขของ if a = a + 3;      //เมื่อประมวลผลบล็อกคำสั่ง if-else แล้ว ค่าของตัวแปร a จะมีค่าเท่ากับ                 // a + 3 </pre> <p>จากตัวอย่างจะได้ผลลัพธ์ค่าตัวแปร a สุดท้ายคือ 8</p> <p><b>5.1.1.3 คำสั่งตรวจสอบเงื่อนไขหลายทางเลือกโดยใช้ if</b></p> <p>รูปแบบ if ( เงื่อนไข 1 )</p> <pre> {     ประโยคคำสั่งเมื่อเงื่อนไขที่ 1 เป็นจริง; } else if ( เงื่อนไขที่ 2 ) {     ประโยคคำสั่งเมื่อเงื่อนไข 2 เป็นจริง; } else if ( เงื่อนไขที่ n ) {     ประโยคคำสั่งเมื่อเงื่อนไข n เป็นจริง; } else {     ประโยคคำสั่งเมื่อเงื่อนไขทั้งหมดเป็นเท็จ; } </pre> <p>ตัวอย่าง</p> <pre> a = 3;           //ให้ตัวแปร a มีค่าเท่ากับ 3 if(a &lt;= 2){      //เงื่อนไขจะเป็นจริงเมื่อตัวแปร a มีค่าน้อยกว่าหรือเท่ากับ 2     a += 3;      //ถ้าเงื่อนไขเป็นจริงตัวแปร a จะมีค่าเท่ากับ a + 3 }else if(a &lt;= 3){ //เงื่อนไขจะเป็นจริงเมื่อตัวแปร a มีค่าน้อยกว่าหรือเท่ากับ 3     a += 2;      //ถ้าเงื่อนไขเป็นจริงตัวแปร a จะมีค่าเท่ากับ a + 2 }else{           //เงื่อนไขเป็นเท็จให้ทำชุดคำสั่งในบล็อกของ else     a += 1;      //เงื่อนไขเป็นเท็จตัวแปร a จะมีค่าเท่ากับ a + 1 }               //จบบล็อกเงื่อนไขของ if a = a + 3;      //เมื่อประมวลผลบล็อกคำสั่ง if-else-if แล้ว ค่าของตัวแปร a จะมีค่าเท่ากับ                 // a + 3 </pre> <p>จากตัวอย่างจะได้ผลลัพธ์ค่าตัวแปร a สุดท้ายคือ 8</p>		

	ใบเนื้อหา	หน้าที่ 12
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิชณนิตบูลิน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์
<p>5.1.2 กลุ่มคำสั่งตรวจสอบเงื่อนไขในรูปแบบคำสั่ง switch-case</p> <p>รูปแบบ switch ( ตัวแปร )</p> <pre>{     case 1 : ประโยคคำสั่งเมื่อตัวแปรมีค่าเท่ากับ 1 ;             break;     case 2 : ประโยคคำสั่งเมื่อตัวแปรมีค่าเท่ากับ 2 ;             break;     case n : ประโยคคำสั่งเมื่อตัวแปรมีค่าเท่ากับ n ;             break;     default : break; }</pre> <p>ตัวอย่าง</p> <pre>a = 3;          //ให้ตัวแปร a มีค่าเท่ากับ 3 switch (a){     //ตรวจสอบค่าของตัวแปร a     case 1 : a += 6 ; //ถ้าตัวแปร a มีค่าเท่ากับ 1 ให้ทำคำสั่ง a = a + 6             break;   //ออกจากบล็อกคำสั่ง switch-case     case 2 : a += 5 ; //ถ้าตัวแปร a มีค่าเท่ากับ 2 ให้ทำคำสั่ง a = a + 5             break;   //ออกจากบล็อกคำสั่ง switch-case     case 3 : a += 4 ; //ถ้าตัวแปร a มีค่าเท่ากับ 3 ให้ทำคำสั่ง a = a + 4             break;   //ออกจากบล็อกคำสั่ง switch-case     default : a = 0; //ถ้าตัวแปร a มีค่าไม่ตรงกับ case ใดเลยให้ทำคำสั่ง a = 0             break;   //ออกจากบล็อกคำสั่ง switch-case }  a = a + 3;      //เมื่อประมวลผลบล็อกคำสั่ง switch-case แล้ว ค่าของตัวแปร a จะมีค่า                 //เท่ากับ a + 3</pre> <p>จากตัวอย่างจะได้ผลลัพธ์ค่าตัวแปร a สุดท้ายคือ 10</p>

	ใบเนื้อหา		หน้าที่ 13
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พิชณนิตบูลิน และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

## 5.2 คำสั่งวนรอบการทำงาน

คำสั่งวนรอบการทำงานของโปรแกรมภาษาซี คือการประมวลผลแบบวนลูปตามเงื่อนไขที่ผู้พัฒนา กำหนดไว้ โดยชุดคำสั่งวนรอบการทำงานของการเขียนโปรแกรมด้วยภาษาซีจะมี 3 รูปแบบ ได้แก่

### 5.2.1 คำสั่ง for()

คำสั่ง for() ส่วนใหญ่จะใช้สำหรับการวนรอบการทำงานแบบที่ผู้พัฒนา กำหนดการวนรอบแบบแน่นอน คาดเดาจำนวนรอบได้ ซึ่งมีรูปแบบการใช้งานดังนี้

รูปแบบ for ( ให้ค่าเริ่มต้นแก่ตัวแปร ; ตรวจสอบเงื่อนไข ; เพิ่มค่าหรือลดค่าตัวแปร )

```
{
    ประโยคคำสั่งเมื่อเงื่อนไขเป็นจริง;
}
```

ในการทำงานของชุดคำสั่ง for ในส่วนของการให้ค่าเริ่มต้นแก่ตัวแปร จะกระทำเพียงครั้งเดียวของการวนรอบ หลังจากนั้นจะทำการตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็นจริง CPU จะเข้าไปประมวลผลในบล็อกคำสั่งของ for() จนถึงวงเล็บปีกกาปิดของ for() ก็จะมาทำในส่วนของการเพิ่มค่าหรือลดค่าตัวแปรในบรรทัด for() หลังจากนั้นก็จะทำการตรวจสอบเงื่อนไขอีกครั้งหนึ่ง ถ้าเงื่อนไขเป็นจริง CPU จะเข้าไปประมวลผลในบล็อกคำสั่งของ for() และกลับมาเพิ่มค่าหรือลดค่าตัวแปรในบรรทัด for() และจะเป็นเช่นนี้จนกว่า ส่วนของการตรวจสอบเงื่อนไขจะมีผลเป็นเท็จ ก็จะจบการทำงานของคำสั่ง for


ตัวอย่าง

```
a = 1;
for(b=2;b<5;b++){
    a += b;
}
```

จากตัวอย่างการทำงานของชุดคำสั่ง for() จะทำการวนรอบทำคำสั่ง a += b; จำนวน 3 รอบ ได้แก่ รอบที่ b = 2 , b = 3 และ b = 4 โดยในรอบที่ b = 2 จะได้ค่า a = 1 + 2 รอบที่ b = 3 จะได้ค่า a = 3 + 3 และรอบที่ b = 4 จะได้ค่า a = 6 + 4 ดังนั้นหลังจากที่ CPU ประมวลผลชุดคำสั่ง for() เรียบร้อยแล้วจะได้ค่า a สุดท้ายคือ a = 10

### 5.2.2 คำสั่ง while()

คำสั่ง while() ส่วนใหญ่จะใช้สำหรับการวนรอบการทำงานแบบที่ผู้พัฒนา กำหนดให้ทำการวนรอบเมื่อเงื่อนไขเป็นจริง และจะทำการออกจากบล็อกของ while() เมื่อเงื่อนไขของการวนรอบเป็นเท็จ โดยรูปแบบการใช้งานดังนี้

	ใบเนื้อหา		หน้าที่ 14
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พิชคณิตบูลีน และการลตรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

รูปแบบ while ( เงื่อนไข )

```

{
    ประโยคคำสั่งเมื่อเงื่อนไขเป็นจริง;
}

```

การทำงานของชุดคำสั่ง while(เงื่อนไข) เริ่มแรกของการทำงานจะทำการตรวจสอบเงื่อนไขก่อน ถ้าผลเป็นจริง CPU จะประมวลผลชุดคำสั่งที่อยู่ภายในวงจรมีปีกกาของบล็อกคำสั่ง while จนกระทั่งพบวงเล็บปีกกาปิดของบล็อกคำสั่ง while CPU จะกลับไปตรวจสอบเงื่อนไขของคำสั่ง while อีกครั้งหนึ่ง ถ้าผลเป็นจริง CPU จะประมวลผลชุดคำสั่งที่อยู่ภายในวงจรมีปีกกาของบล็อกคำสั่ง while และจะเป็นเช่นนี้จนกว่าการตรวจสอบเงื่อนไขของคำสั่ง while จะมีผลเป็นเท็จ

ตัวอย่าง

```

a = 1;
while(a<10){
    a += 2;
}

```

จากตัวอย่างการทำงานของชุดคำสั่ง while() จะทำการวนรอบจนกว่าค่าของตัวแปร a จะมีค่ามากกว่าหรือเท่ากับ 10 โดยจากตัวอย่างจะมีการวนรอบจำนวน 5 ครั้ง ซึ่งครั้งแรก a = 1 แล้วมีผลทำให้ a = 1 + 2 ทำให้ในรอบต่อมา a = 3 แล้วมีผลทำให้ a = 3+2 ทำให้ในรอบต่อมา a = 5 แล้วมีผลทำให้ a = 5 + 2 ทำให้ในรอบต่อมา a = 7 แล้วมีผลทำให้ a = 7 + 2 ทำให้ในรอบต่อมา a = 9 แล้วมีผลทำให้ a = 9 + 2 ซึ่งเท่ากับ 11 เมื่อ while ทำการตรวจสอบจะทำให้ผลของเงื่อนไขเป็นเท็จ จึงเป็นการจบการทำงานของคำสั่ง while ส่วน a ค่าสุดท้ายจะมีค่าเท่ากับ 11

### 5.2.3 คำสั่ง do-while()

คำสั่ง do-while() จะมีลักษณะการทำงานของชุดคำสั่งคล้ายกับคำสั่ง while() ซึ่งคำสั่ง while() จะทำการวนรอบก็ต่อเมื่อทำการตรวจสอบคำสั่งแล้วมีผลเป็นจริงก่อนถึงจะทำการวนรอบ แต่คำสั่ง do-while() จะทำการวนรอบในบล็อกของคำสั่ง do-while() ก่อนแล้วทำการตรวจสอบเงื่อนไขถ้าเงื่อนไขเป็นจริงก็จะทำการวนรอบประมวลผลใหม่อีกครั้งหนึ่ง และจะทำเช่นนี้จนกว่าเงื่อนไขของคำสั่ง do-while() จะมีผลเป็นเท็จถึงจะออกจากการประมวลผลคำสั่ง do-while() ส่วนรูปแบบการเขียนคำสั่ง do-while() เป็นดังนี้


รูปแบบ do {

```

    ประโยคคำสั่งก่อนที่จะทำการตรวจสอบเงื่อนไข;
} while ( เงื่อนไข );


```




	ใบเนื้อหา	หน้าที่ 15
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิชณนิตบูลีน และการลดรูปสมการ	
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์		
<div>ตัวอย่าง</div> <div>a = 1;</div> <div>do{</div> <div>    a += 2;</div> <div>} while(a&lt;10);</div> <div>จากตัวอย่างการทำงานของชุดคำสั่ง do-while() โปรแกรมจะทำการประมวลผล a += 2 ก่อนหลังจากนั้นถึงจะทำการตรวจสอบว่า a &lt; 10 หรือไม่ ถ้าผลเป็นจริงก็จะกลับไปประมวลผลคำสั่ง a += 2 ใหม่ แต่ถ้าผลเป็นเท็จก็จะจบการทำงานของคำสั่ง do-while() ทันที โดยตัวอย่างโปรแกรมจะมีการวนรอบจำนวน 5 ครั้ง โดยในครั้งแรก a = 1 เป็นผลทำให้ a = 1 + 2 ในรอบต่อมา a = 3 เป็นผลทำให้ a = 3 + 2 ในรอบต่อมา a = 5 เป็นผลทำให้ a = 5 + 2 ในรอบต่อมา a = 7 เป็นผลทำให้ a = 7 + 2 ในรอบต่อมา a = 9 เป็นผลทำให้ a = 9 + 2 ซึ่งได้ผลลัพธ์สุดท้ายของ a เป็น 11 เมื่อ do-while() ทำการตรวจสอบจะได้ผลเป็นเท็จ ทำให้สิ้นสุดการทำงานของ do-while()</div>		
<h3>6. การใช้งานขาไมโครคอนโทรลเลอร์ด้วยภาษาซี</h3> <p>ภาษาซีกับการใช้งานขาของไมโครคอนโทรลเลอร์นั้นจะคล้ายคลึงกับการใช้งานภาษาแอสเซมบลีในส่วนของไมโครคอนโทรลเลอร์ตระกูล PIC16F และ AVR คือจะต้องมีการเรียกใช้รีจิสเตอร์ที่ควบคุมทิศทางการทำงานของขาสัญญาณพอร์ตนั้น ๆ ก่อนเพื่อกำหนดทิศทางของขาพอร์ตว่าจำเป็นต้องทำหน้าที่เป็นขาสัญญาณอินพุตหรือเอาต์พุต ส่วนการเขียนโปรแกรมเพื่อติดต่อพอร์ตก็จะกระทำคล้ายกับการให้ค่าทางคณิตศาสตร์ ส่วนการตรวจสอบค่าข้อมูลก็จะกระทำผ่านตัวดำเนินการเปรียบเทียบกับคำสั่งพื้นฐานภาษาซี ซึ่งการใช้งานขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51,PIC16F และ AVR ในลักษณะขาสัญญาณดิจิทัลจะมีการใช้งาน 2 รูปแบบคือ</p> <div><div>6.1 การใช้งานขาไมโครคอนโทรลเลอร์ตระกูล MCS-51,PIC16F และAVR เป็นขาสัญญาณเอาต์พุตด้วยภาษาซี</div><div>6.2 การใช้งานขาไมโครคอนโทรลเลอร์ตระกูล MCS-51,PIC16F และ AVR เป็นขาสัญญาณอินพุตด้วยภาษาซี</div></div> <p>โดยการใช้งานขาของไมโครคอนโทรลเลอร์ตระกูล MCS-51,PIC16F และ AVR ให้ทำหน้าที่เป็นขาสัญญาณอินพุตเอาต์พุตเพื่อใช้งานรับส่งสัญญาณดิจิทัลนั้นจะมีข้อแตกต่างในเรื่องของการเข้าถึงขาสัญญาณผ่านรีจิสเตอร์ตัวควบคุม และโครงสร้างของพอร์ตตั้งที่กล่าวมาแล้วในเรื่องของพื้นฐานไมโครคอนโทรลเลอร์ และข้อกำหนดของ compiler ที่ผู้พัฒนาเลือกใช้งานในการพัฒนาโปรแกรมภาษาซีของไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะอ้างอิงกับ C51 compiler ส่วนไมโครคอนโทรลเลอร์ตระกูล PIC16F จะอ้างอิงกับ XC8 compiler และไมโครคอนโทรลเลอร์ตระกูล AVR จะอ้างอิงกับ AvrGCC และการใช้งานขาไมโครคอนโทรลเลอร์ด้วยภาษาซีของตระกูลทั้งสามสามารถสรุปเป็นขั้นตอน และเทียบเคียงกันได้ดังนี้</p>		

	ใบเนื้อหา		หน้าที่ 16
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พิชณนิบูลีน และการลดรูปสมการ		
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์			
การเขียนโปรแกรมภาษาซีเพื่อกำหนด และใช้งานพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ตระกูล MCS-51 , PIC16F และ AVR			
1. การประกาศหัวไฟล์ Preprocessor Directives ที่เป็นการนำข้อมูลเกี่ยวกับไมโครคอนโทรลเลอร์เบอร์นั้น ๆ มาร่วมประมวลผล เพื่อให้การประกาศตัวแปรที่อ้างถึงรีจิสเตอร์ภายในไมโครคอนโทรลเลอร์เบอร์นั้น ๆ ตัวคอมไพเลอร์รู้จัก และจะไม่แสดงการคอมไพล์ Error			
ตัวอย่างการ include หัวไฟล์			
MCS51-51	PIC16F	AVR	
#include <reg51.h> /*สำหรับไมโครคอนโทรลเลอร์ที่อ้างอิงไมโครโปรเซสเซอร์ 8051 */ //หรือ #include <reg52.h> /*สำหรับไมโครคอนโทรลเลอร์ที่อ้างอิงไมโครโปรเซสเซอร์ 8052 */	#include <xc.h> /*เป็นการเรียกไฟล์ IO มาตรฐานของ PIC ซึ่งจะจบที่บรรทัดนี้เมื่อตอนเริ่มต้นใช้งาน IDE ได้กำหนดเบอร์ไมโครคอนโทรลเลอร์ตระกูล PIC เรียบร้อยแล้ว*/	#include <avr/io.h> /*เป็นการเรียกไฟล์ IO มาตรฐานของ AVR ซึ่งจะจบที่บรรทัดนี้หรือเพิ่มการ include ไปอีก 1 บรรทัดเพื่อระบุเบอร์ของ AVR*/  #include <avr/iom32.h> /*เป็นการเรียกไฟล์ IO ของเบอร์ ATMEGA32*/	
2. การประกาศตัวแปรในภาษาซีที่ใช้สำหรับอ้างอิงถึงขาสัญญาณของไมโครคอนโทรลเลอร์ที่เชื่อมต่ออยู่กับอุปกรณ์อินพุตและเอาต์พุต			
ตัวอย่างการประกาศตัวแปรที่อ้างถึงขาไมโครคอนโทรลเลอร์			
MCS-51	PIC16F	AVR	
sbit sw1 = P1^0; /*กำหนดตัวแปรชื่อ sw1 เพื่อใช้ในอ้างอิงถึงข้อมูลที่ขา P1.0*/ sbit led1 = P1^1; /*กำหนดตัวแปรชื่อ led1 เพื่อใช้ในการอ้างอิงถึงข้อมูลที่ขา P1.1*/  หมายเหตุ MCS51 ไม่จำเป็นต้องกำหนด Direction ของขาที่สามารถใช้งานเป็นขาอินพุตหรือเอาต์พุตได้	#define sw1 RB0 /*กำหนดตัวแปรชื่อ sw1 เพื่อใช้ในอ้างอิงถึงข้อมูลที่ขา RB0 ที่ต้องการให้ขา RB0 ทำหน้าที่เป็นอินพุต*/ #define led1 RB1 /*กำหนดตัวแปรชื่อ led1 เพื่อใช้ในการอ้างอิงถึงข้อมูลที่ขา RB1 ที่ต้องการให้ขา RB1 ทำหน้าที่เป็นเอาต์พุต*/ /*เมื่อประกาศตัวแปรแล้วในส่วนฟังก์ชัน main จะต้องทำการกำหนด Direction ให้ขาพอร์ตเหล่านี้ให้ทำหน้าที่เป็นขาอินพุตหรือเอาต์พุตโดยใช้คำสั่ง TRISx=yy โดย x คือชื่อพอร์ต และ yy คือค่าข้อมูลที่กำหนดคุณสมบัติของขาพอร์ต*/	#define sw1 PINB0 /*กำหนดตัวแปรชื่อ sw1 เพื่อใช้ในอ้างอิงถึงข้อมูลที่ขา PB0 ที่ต้องการให้ขา PB0 ทำหน้าที่เป็นอินพุต*/ #define led1 PB1 /*กำหนดตัวแปรชื่อ led1 เพื่อใช้ในการอ้างอิงถึงข้อมูลที่ขา PB.1 ที่ต้องการให้ขา PB.1 ทำหน้าที่เป็นเอาต์พุต*/ /*เมื่อประกาศตัวแปรแล้วในส่วนฟังก์ชัน main จะต้องทำการกำหนด Direction ให้ขาพอร์ตเหล่านี้ให้ทำหน้าที่เป็นขาอินพุตหรือเอาต์พุตโดยใช้คำสั่ง DDRx=yy โดย x คือชื่อพอร์ต และ yy คือค่าข้อมูลที่กำหนดคุณสมบัติของขาพอร์ต*/	

	ใบเนื้อหา		หน้าที่ 17						
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2						
	ชื่อหน่วย พืชคณิตบูลีน และการลดรูปสมการ								
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์									
<p>3. การกำหนดคุณสมบัติขาพอร์ตของไมโครคอนโทรลเลอร์ให้ทำหน้าที่เป็นขาอินพุตหรือเอาต์พุต ซึ่งคำสั่งนี้จะต้องเขียนหรือประกาศในฟังก์ชัน main หรือสร้างฟังก์ชันที่ใช้ในการกำหนดคุณสมบัติขาพอร์ตของไมโครคอนโทรลเลอร์ขึ้นมาแล้วทำการเรียกใช้งานโดยฟังก์ชัน main</p> <p>ตัวอย่างการกำหนดคุณสมบัติขาพอร์ตของไมโครคอนโทรลเลอร์ให้ทำหน้าที่เป็นอินพุตหรือเอาต์พุต</p> <table><tr><th>MCS-51</th><th>PIC16F</th><th>AVR</th></tr><tr><td>MCS51 ไม่จำเป็นต้องกำหนด Direction ของขาที่สามารถใช้งานเป็นขาอินพุตหรือเอาต์พุตได้</td><td>TRISB = 0b00000001; /*ค่าที่กำหนดให้รีจิสเตอร์ TRISB เป็นข้อมูลเลขฐานสองโดยเรียงจากซ้ายมาขวาเป็นค่าข้อมูลที่กำหนดการทำงานของขา RB7-RB0 ซึ่งค่า 0 หมายถึงให้ทำงานเป็นขาเอาต์พุต และค่า 1 หมายถึงให้ทำงานเป็นขาอินพุต ดังนั้นจากคำสั่ง RB7-RB1 จึงทำงานเป็นขาเอาต์พุต ส่วน RB0 ทำงานเป็นขาอินพุต*/</td><td>DDRB = 0b00000001; /*ค่าที่กำหนดให้รีจิสเตอร์ DDRB เป็นข้อมูลเลขฐานสองโดยเรียงจากซ้ายมาขวาเป็นค่าข้อมูลที่กำหนดการทำงานของขา PB7-PB0 ซึ่งค่า 0 หมายถึงให้ทำงานเป็นขาอินพุต และค่า 1 หมายถึงให้ทำงานเป็นขาเอาต์พุต ดังนั้นจากคำสั่ง PB7-PB1 จึงทำงานเป็นขาอินพุตส่วน PB0 ทำงานเป็นขา เอาต์พุต*/</td></tr></table>				MCS-51	PIC16F	AVR	MCS51 ไม่จำเป็นต้องกำหนด Direction ของขาที่สามารถใช้งานเป็นขาอินพุตหรือเอาต์พุตได้	TRISB = 0b00000001; /*ค่าที่กำหนดให้รีจิสเตอร์ TRISB เป็นข้อมูลเลขฐานสองโดยเรียงจากซ้ายมาขวาเป็นค่าข้อมูลที่กำหนดการทำงานของขา RB7-RB0 ซึ่งค่า 0 หมายถึงให้ทำงานเป็นขาเอาต์พุต และค่า 1 หมายถึงให้ทำงานเป็นขาอินพุต ดังนั้นจากคำสั่ง RB7-RB1 จึงทำงานเป็นขาเอาต์พุต ส่วน RB0 ทำงานเป็นขาอินพุต*/	DDRB = 0b00000001; /*ค่าที่กำหนดให้รีจิสเตอร์ DDRB เป็นข้อมูลเลขฐานสองโดยเรียงจากซ้ายมาขวาเป็นค่าข้อมูลที่กำหนดการทำงานของขา PB7-PB0 ซึ่งค่า 0 หมายถึงให้ทำงานเป็นขาอินพุต และค่า 1 หมายถึงให้ทำงานเป็นขาเอาต์พุต ดังนั้นจากคำสั่ง PB7-PB1 จึงทำงานเป็นขาอินพุตส่วน PB0 ทำงานเป็นขา เอาต์พุต*/
MCS-51	PIC16F	AVR							
MCS51 ไม่จำเป็นต้องกำหนด Direction ของขาที่สามารถใช้งานเป็นขาอินพุตหรือเอาต์พุตได้	TRISB = 0b00000001; /*ค่าที่กำหนดให้รีจิสเตอร์ TRISB เป็นข้อมูลเลขฐานสองโดยเรียงจากซ้ายมาขวาเป็นค่าข้อมูลที่กำหนดการทำงานของขา RB7-RB0 ซึ่งค่า 0 หมายถึงให้ทำงานเป็นขาเอาต์พุต และค่า 1 หมายถึงให้ทำงานเป็นขาอินพุต ดังนั้นจากคำสั่ง RB7-RB1 จึงทำงานเป็นขาเอาต์พุต ส่วน RB0 ทำงานเป็นขาอินพุต*/	DDRB = 0b00000001; /*ค่าที่กำหนดให้รีจิสเตอร์ DDRB เป็นข้อมูลเลขฐานสองโดยเรียงจากซ้ายมาขวาเป็นค่าข้อมูลที่กำหนดการทำงานของขา PB7-PB0 ซึ่งค่า 0 หมายถึงให้ทำงานเป็นขาอินพุต และค่า 1 หมายถึงให้ทำงานเป็นขาเอาต์พุต ดังนั้นจากคำสั่ง PB7-PB1 จึงทำงานเป็นขาอินพุตส่วน PB0 ทำงานเป็นขา เอาต์พุต*/							
<p>4. การส่งค่าข้อมูลออกที่ขาพอร์ตในรูปแบบข้อมูลแบบบิตผ่านตัวแปรที่กำหนด</p> <p>ตัวอย่างการส่งค่าข้อมูลออกที่ขาพอร์ตในรูปแบบข้อมูลแบบบิตผ่านตัวแปรที่กำหนด</p> <table><tr><th>MCS-51</th><th>PIC16F</th><th>AVR</th></tr><tr><td>led1 = 1; /*เป็นการส่งค่าข้อมูล 1 หรือให้แรงดัน VCC ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/  led1 = 0; /*เป็นการส่งค่าข้อมูล 0 หรือให้แรงดัน GND ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/</td><td>led1 = 1; /*เป็นการส่งค่าข้อมูล 1 หรือให้แรงดัน VDD ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/  led1 = 0; /*เป็นการส่งค่าข้อมูล 0 หรือให้แรงดัน VSS ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/</td><td>PORTB  = _BV(led1); หรือ PORTB  = (1&lt;&lt;led1); /*เป็นการส่งค่าข้อมูล 1 หรือให้แรงดัน VCC ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/ PORTB &amp;= ~_BV(led1); หรือ PORTB &amp;= ~(1&lt;&lt;led1); /*เป็นการส่งค่าข้อมูล 0 หรือให้แรงดัน GND ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/</td></tr></table>				MCS-51	PIC16F	AVR	led1 = 1; /*เป็นการส่งค่าข้อมูล 1 หรือให้แรงดัน VCC ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/  led1 = 0; /*เป็นการส่งค่าข้อมูล 0 หรือให้แรงดัน GND ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/	led1 = 1; /*เป็นการส่งค่าข้อมูล 1 หรือให้แรงดัน VDD ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/  led1 = 0; /*เป็นการส่งค่าข้อมูล 0 หรือให้แรงดัน VSS ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/	PORTB  = _BV(led1); หรือ PORTB  = (1<<led1); /*เป็นการส่งค่าข้อมูล 1 หรือให้แรงดัน VCC ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/ PORTB &= ~_BV(led1); หรือ PORTB &= ~(1<<led1); /*เป็นการส่งค่าข้อมูล 0 หรือให้แรงดัน GND ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/
MCS-51	PIC16F	AVR							
led1 = 1; /*เป็นการส่งค่าข้อมูล 1 หรือให้แรงดัน VCC ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/  led1 = 0; /*เป็นการส่งค่าข้อมูล 0 หรือให้แรงดัน GND ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/	led1 = 1; /*เป็นการส่งค่าข้อมูล 1 หรือให้แรงดัน VDD ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/  led1 = 0; /*เป็นการส่งค่าข้อมูล 0 หรือให้แรงดัน VSS ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/	PORTB  = _BV(led1); หรือ PORTB  = (1<<led1); /*เป็นการส่งค่าข้อมูล 1 หรือให้แรงดัน VCC ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/ PORTB &= ~_BV(led1); หรือ PORTB &= ~(1<<led1); /*เป็นการส่งค่าข้อมูล 0 หรือให้แรงดัน GND ออกไปยังขาพอร์ตของไมโครคอนโทรลเลอร์ที่ถูกอ้างถึงด้วยตัวแปร led1*/							

	ใบเนื้อหา		หน้าที่ 18
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พืชคณิตบูลีน และการลดรูปสมการ		
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์			
5. การอ่านค่าข้อมูลจากขาพอร์ตในรูปแบบข้อมูลแบบบิต ผ่านตัวแปรที่กำหนด มาเก็บไว้ในตัวแปรอีกตัวแปรหนึ่ง			
ตัวอย่างการอ่านค่าข้อมูลจากขาพอร์ตในรูปแบบข้อมูลแบบบิตมาเก็บไว้ในตัวแปรที่กำหนด			
MCS-51	PIC16F	AVR	
bit dat; dat = sw1; /*เป็นนำค่าข้อมูลขนาด 1 บิตที่ขาพอร์ตที่อ้างถึงด้วยตัวแปร sw1 มาเก็บไว้ในตัวแปร dat*/	unsigned char dat; dat = sw1; /*เป็นนำค่าข้อมูลขนาด 1 บิตที่ขาพอร์ตที่อ้างถึงด้วยตัวแปร sw1 มาเก็บไว้ในตัวแปร dat โดยตัวแปร dat จะเก็บค่าข้อมูลขนาด 1 byte */	unsigned char dat; dat = PINx & (1<<sw1); /*เป็นนำค่าข้อมูลขนาด 1 บิตที่ขาพอร์ตที่อ้างถึงด้วยตัวแปร sw1 มาเก็บไว้ในตัวแปร dat โดยตัวแปร dat จะเก็บค่าข้อมูลขนาด 1 byte*/	
6. การอ่านค่าข้อมูลจากขาพอร์ตเพื่อเปรียบเทียบค่าข้อมูลในรูปแบบข้อมูลแบบบิต ที่ใช้งานร่วมกับคำสั่งควบคุมของภาษาซีได้แก่คำสั่ง if			
ตัวอย่างการอ่านค่าข้อมูลจากขาพอร์ตในรูปแบบข้อมูลแบบบิตเพื่อเปรียบเทียบกับค่าข้อมูลที่กำหนด			
MCS-51	PIC16F	AVR	
If(sw1 == 0) { led1 = 1; }else{ led1 = 0; } /*เป็นการตรวจสอบข้อมูลที่ขาของพอร์ตผ่านตัวแปร sw1 ว่ามีค่าเท่ากับ 0 หรือไม่ ถ้าใช่ให้ทำคำสั่งส่งข้อมูลออกขาพอร์ตผ่านตัวแปร led1 มีค่าเป็น 1 แต่ถ้าไม่ใช่ให้ส่งค่าข้อมูล 0 ออกไปที่ขาพอร์ตผ่านตัวแปร led1*/	If(sw1 == 0) { led1 = 1; }else{ led1 = 0; } /*เป็นการตรวจสอบข้อมูลที่ขาของพอร์ตผ่านตัวแปร sw1 ว่ามีค่าเท่ากับ 0 หรือไม่ ถ้าใช่ให้ทำคำสั่งส่งข้อมูลออกขาพอร์ตผ่านตัวแปร led1 มีค่าเป็น 1 แต่ถ้าไม่ใช่ให้ส่งค่าข้อมูล 0 ออกไปที่ขาพอร์ตผ่านตัวแปร led1*/	If((PINx & (1<<sw1)) == 0) { PORTx  = _BV(led1); }else{ PORTx &= ~_BV(led1); } /*เป็นการตรวจสอบข้อมูลที่ขาของพอร์ตผ่านตัวแปร sw1 ว่ามีค่าเท่ากับ 0 หรือไม่ ถ้าใช่ให้ทำคำสั่งส่งข้อมูลออกขาพอร์ตผ่านตัวแปร led1 มีค่าเป็น 1 แต่ถ้าไม่ใช่ให้ส่งค่าข้อมูล 0 ออกไปที่ขาพอร์ตผ่านตัวแปร led1 โดย x ให้เติมด้วยชื่อของพอร์ตที่เรากำหนดเป็นอินพุตหรือเอาต์พุต*/	

	ใบเนื้อหา	หน้าที่ 19
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พืชคณิตบูลีน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

## 7. การใช้งานซอฟต์แวร์เพื่อเขียนโปรแกรมภาษาซีของไมโครคอนโทรลเลอร์

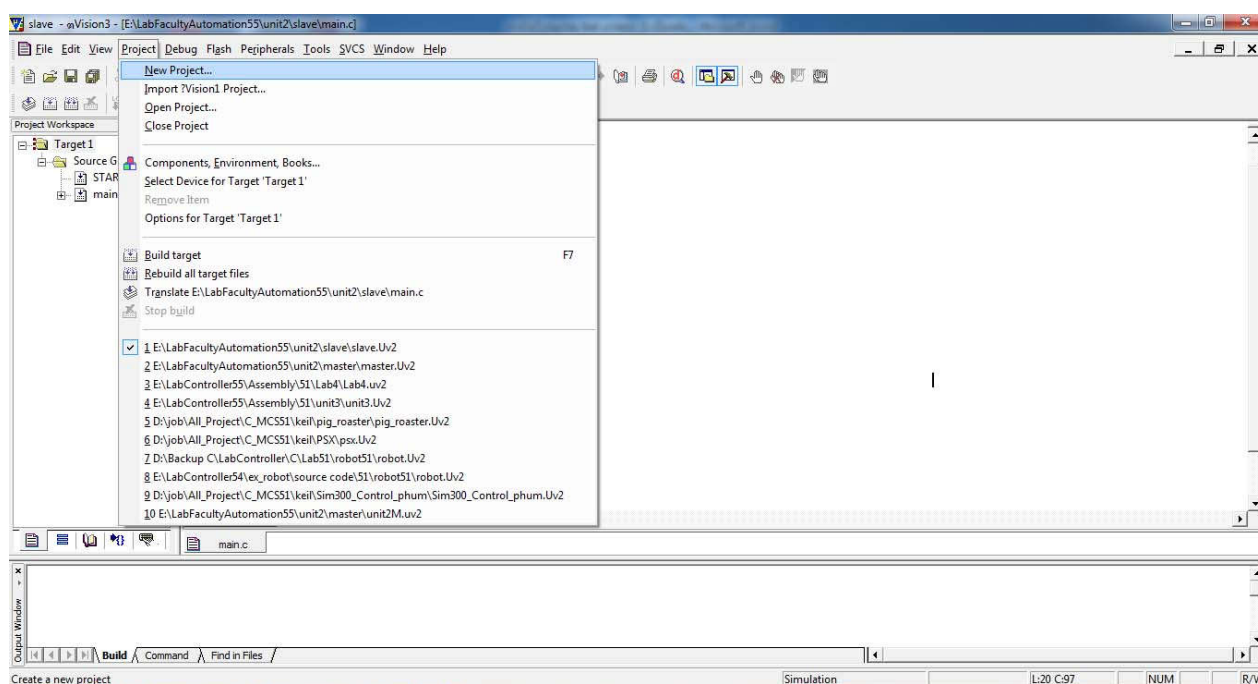
### 7.1 การใช้งานโปรแกรม Keil uVision3 สำหรับเขียนโปรแกรมภาษาซีของไมโครคอนโทรลเลอร์ตระกูล MCS51

1. ให้ทำการดับเบิลคลิกที่ไอคอนโปรแกรม Keil uVision3 ด้านหน้า Desktop ตามรูปที่ 1.5




รูปที่ 1.5 รูปแสดงไอคอนของโปรแกรม Keil uVision3

2. คลิกที่เมนู Project > New Project ตามรูปที่ 1.6

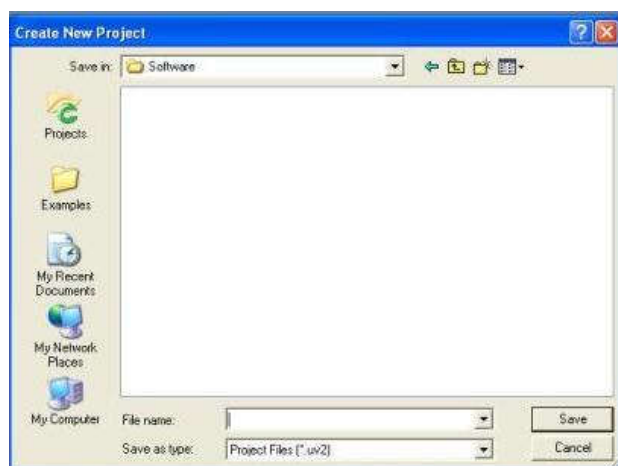


รูปที่ 1.6 รูปแสดงโปรแกรม Keil uVision3

3. เมื่อปรากฏรูปตามรูปที่ 1.7 ให้ทำการเลือก Drive และ Folder ที่ต้องการจะทำการเก็บโปรเจกต์ไฟล์ที่กำลังจะสร้างขึ้น จากนั้นให้ทำการตั้งชื่อโปรเจกต์ไฟล์ในช่อง File name เมื่อตั้งชื่อไฟล์เรียบร้อยแล้วให้ทำการคลิกที่ปุ่ม Save

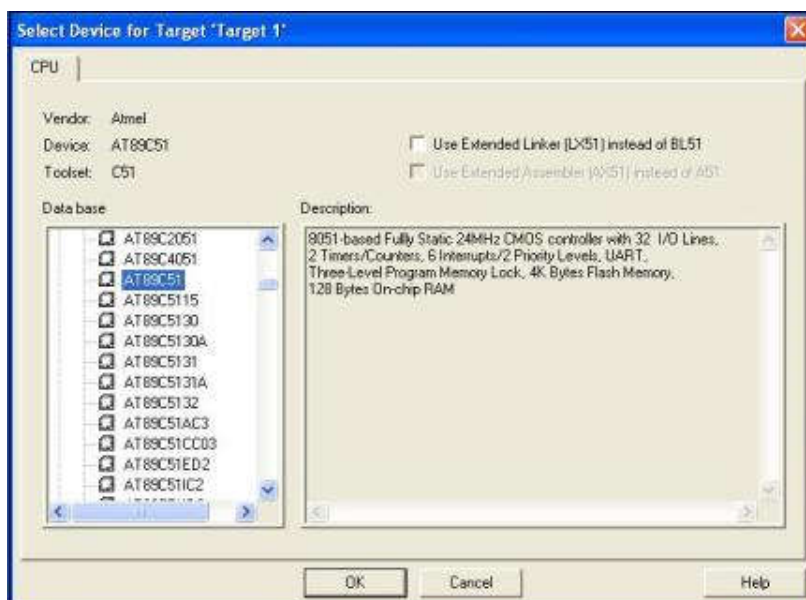
	ใบเนื้อหา	หน้าที่ 20
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิชณนิบูลีน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์



รูปที่ 1.7 รูปแสดงหน้าต่าง Create New Project


4. หลังจากที่ยืนยันโปรเจกต์แล้ว โปรแกรม uVision 3 จะแสดงหน้าต่าง Select Device for Target 'Target1' ขึ้นมาตามรูปที่ 1.8 เพื่อให้เราเลือกชิพที่จะใช้งานจาก Device Database โดยให้เลือกชิพที่ต้องการ ซึ่งในตัวอย่างเลือกชิพ AT89C51 ของบริษัท ATMEL



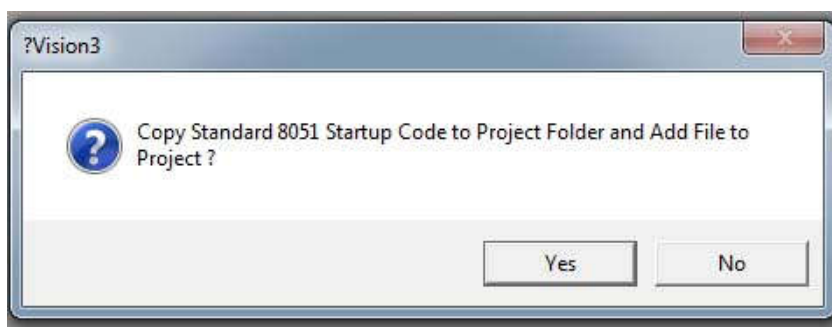
รูปที่ 1.8 รูปแสดงหน้าต่าง Select Device for Target 'Target1'

5. เมื่อทำการเลือกเบอร์ CPU เรียบร้อยแล้วจะปรากฏหน้าต่างตามรูปที่ 1.9 ซึ่งถ้าเขียนโปรแกรมภาษาแอสเซมบลีให้เลือกตอบ No แต่ถ้าเขียนโปรแกรมภาษาซีให้ตอบ Yes



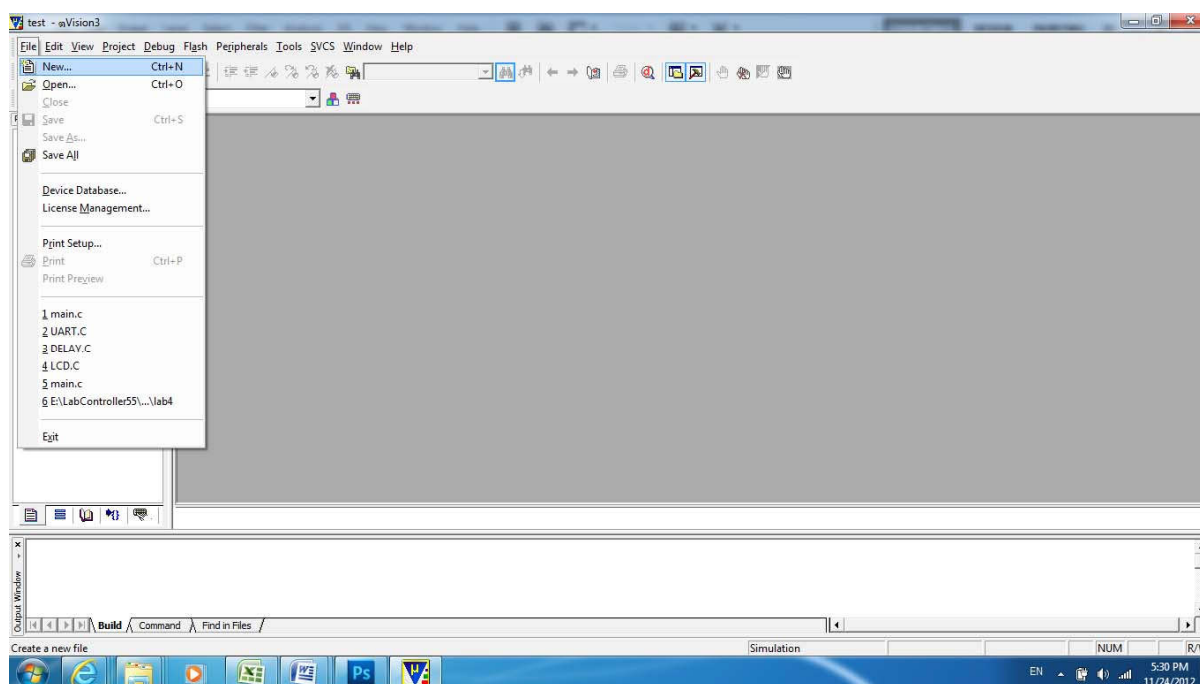
	ใบเนื้อหา	หน้าที่ 21
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พืชคณิตมูล และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์




รูปที่ 1.9 รูปแสดงหน้าต่างให้เลือกสร้างไฟล์ 8051 Startup Code

6. เมื่อเข้าสู่โปรแกรมให้ทำการเลือกเมนู File > new ตามรูปที่ 1.10

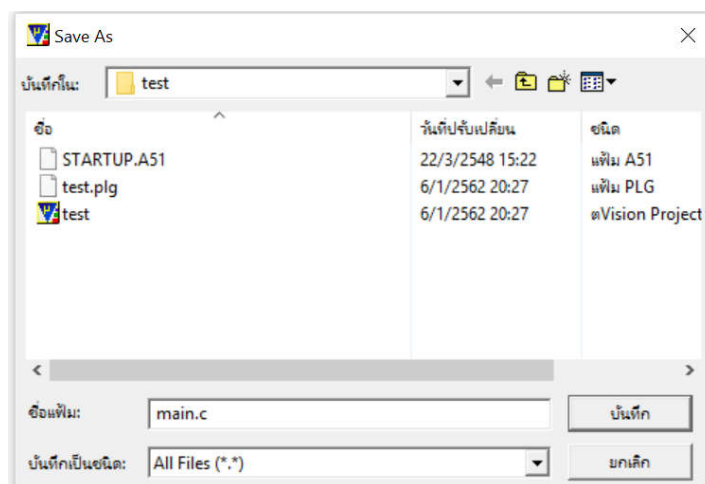


รูปที่ 1.10 รูปแสดงขั้นตอนสร้าง new file

7. ให้ทำการขยายหน้าจอ work sheet แล้วทำการเลือกเมนู File > Save จะปรากฏหน้าต่าง Save as ตามรูปที่ 1.11 ให้ทำการตั้งชื่อไฟล์ในช่อง File name โดยให้ต่อท้ายด้วยนามสกุล .c

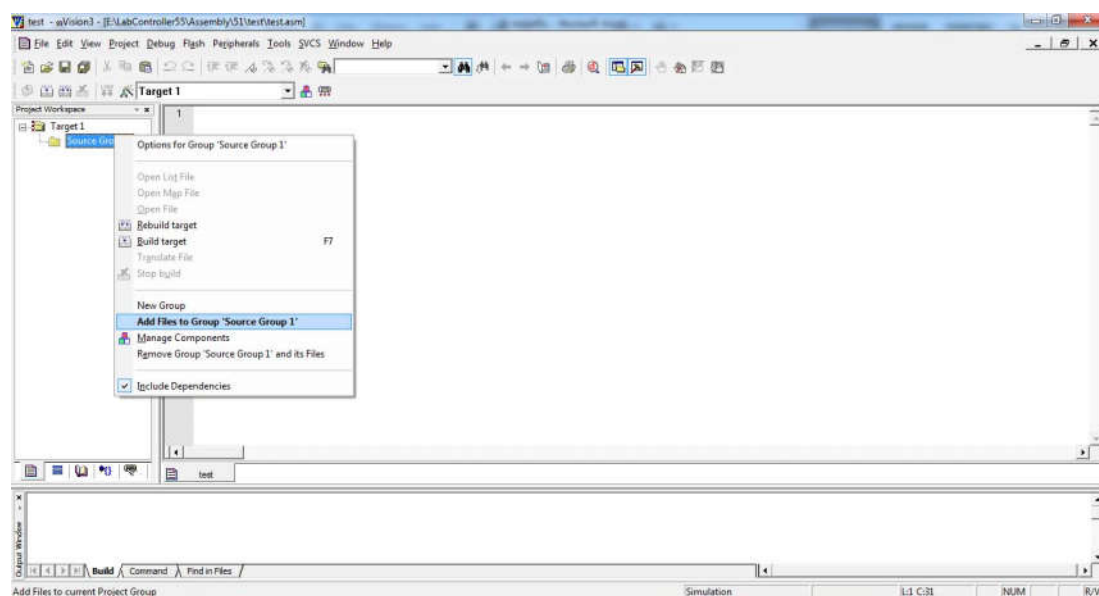
	ใบเนื้อหา	หน้าที่ 22
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พืชคณิตบูลีน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์




รูปที่ 1.11 แสดงรูปหน้าต่างการตั้งชื่อไฟล์นามสกุล .c

8. บริเวณหน้าต่างโปรแกรมทางด้านซ้ายมือให้ทำการคลิกที่เครื่องหมายถูกหน้าคำว่า Target1 แล้วทำการคลิกขวาที่คำว่า Source Group 1 จะปรากฏหน้าต่างดังรูปที่ 1.12 ให้ทำการเลือก Add File to Group “Source Group 1”

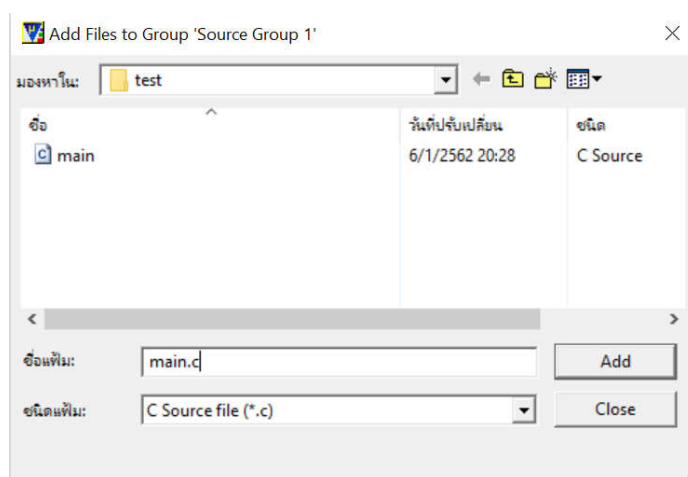


รูปที่ 1.12 แสดงหน้าต่างการเลือกการเลือก Add File to Group “Source Group 1”

9. เมื่อปรากฏหน้าต่าง Add File to Group “Source Group 1” ตามรูปที่ 1.13 ในช่อง File name ให้พิมพ์ชื่อไฟล์พร้อมนามสกุล .c ที่ได้ทำการบันทึกไว้แล้ว หลังจากนั้นให้ทำการคลิกที่ปุ่ม Add ก็จะปรากฏไฟล์ที่ได้ทำการเลือกไว้ปรากฏที่หน้าต่างโปรแกรมทางด้านซ้ายมือซึ่งจะอยู่ในซับของ Source Group 1 โดยขั้นตอนนี้ให้ทำเพียงครั้งเดียวในการเขียนโปรแกรมหนึ่งโปรแกรม

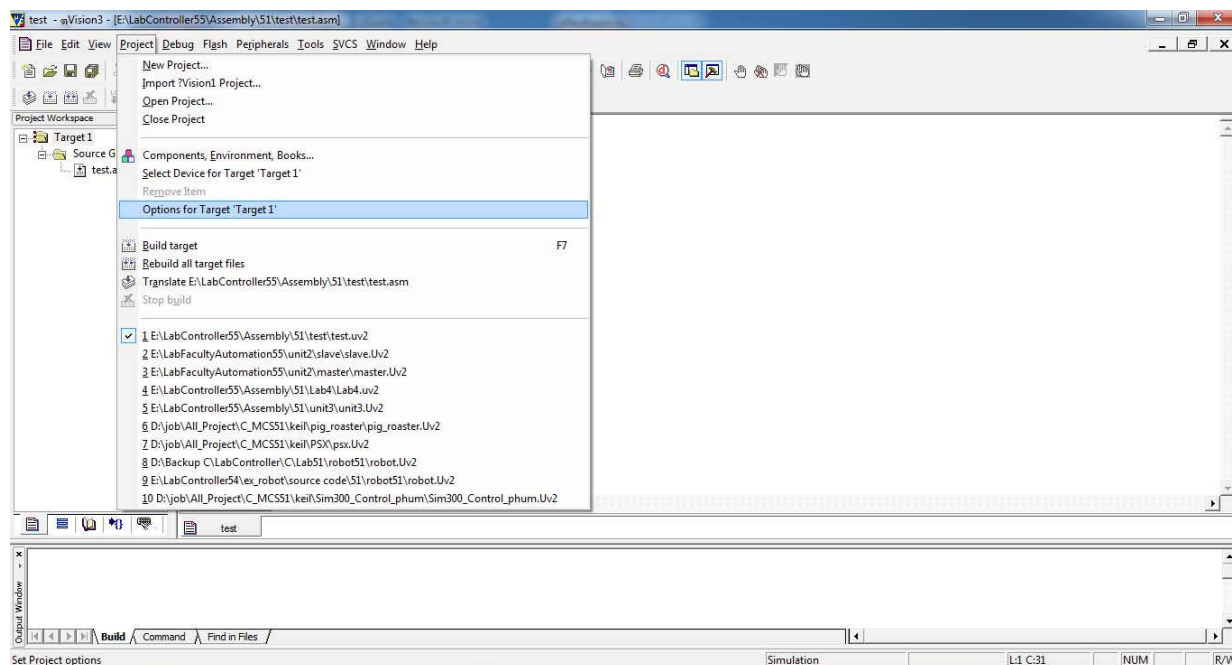
	ใบเนื้อหา		หน้าที่ 23
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พืชคณิตบูลีน และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์




รูปที่ 1.13 แสดงการเพิ่มไฟล์นามสกุล .c เข้าสู่ Source Group 1

10. ให้ทำการเลือกเมนู Project > Options for Target 'Target 1' ดังรูปที่ 1.14

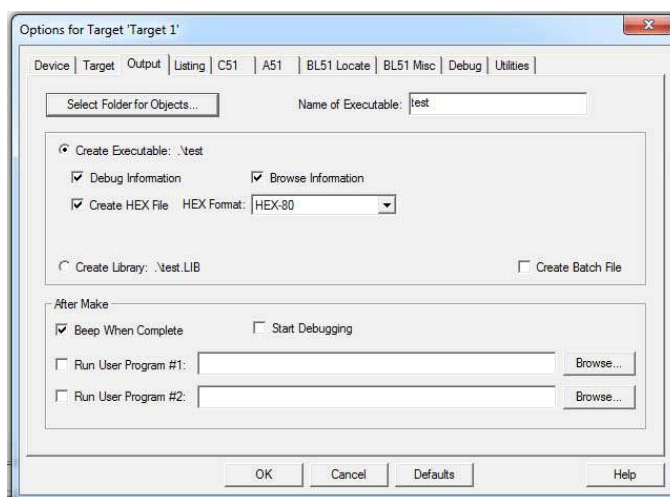


รูปที่ 1.14 แสดงขั้นตอนการเลือกเมนู Options for Target 'Target 1'

11. จากข้อ 10 จะปรากฏหน้าต่าง Options for Target 'Target1' ดังรูปที่ 1.15 ให้ทำการเลือกแท็บ Output แล้วทำการคลิกตรงช่อง Create Hex File ให้มีเครื่องหมายถูก จากนั้นให้คลิกที่ปุ่ม Ok โดยขั้นตอนนี้จะกระทำเพียงครั้งเดียวต่อ 1 โปรแกรม

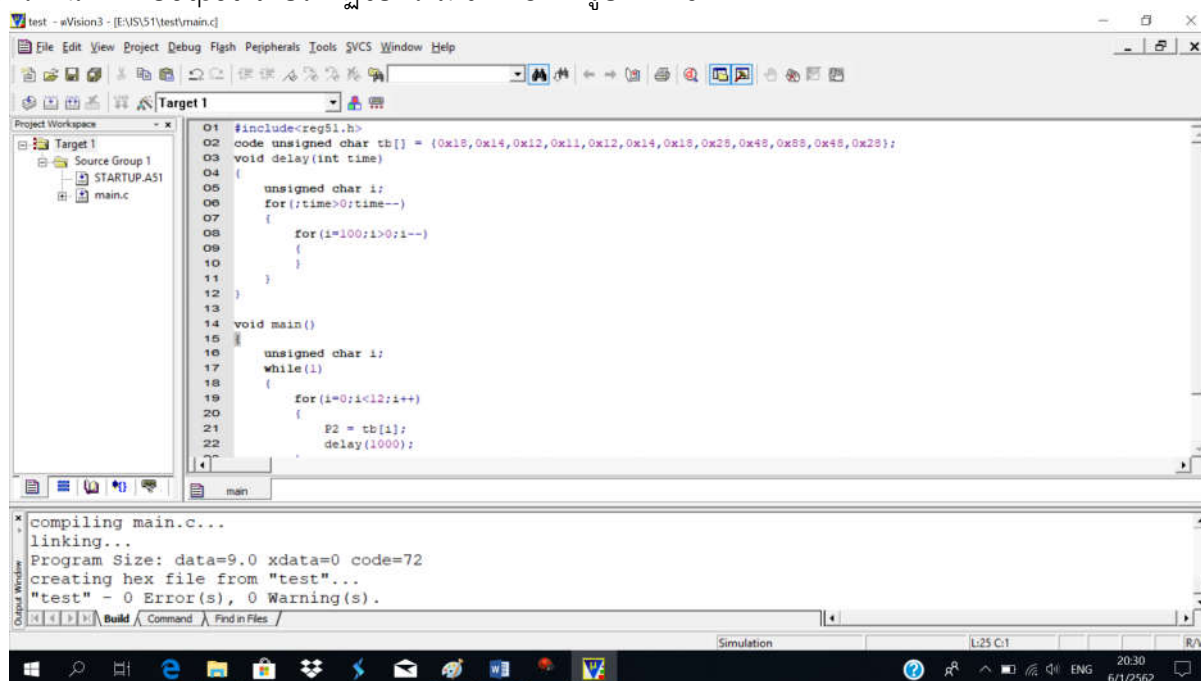
	ใบเนื้อหา	หน้าที่ 24
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิษคณิตบูลีน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์




รูปที่ 1.15 รูปแสดงการเลือกเมนู Create HEX File

12. เมื่อทำขั้นตอน 1 – 11 เรียบร้อยแล้วให้ทำการเขียนโปรแกรมภาษา C บริเวณ Work Sheet เมื่อทำการเขียนโปรแกรมเรียบร้อยแล้วให้ทำการ Compile โปรแกรมโดยเข้าเมนู Project > Build target แล้วให้สังเกตบริเวณหน้าต่างโปรแกรมด้านล่างว่ามีข้อความ 0 Error หรือไม่ ถ้าไม่ให้ทำการแก้ไขโปรแกรมแล้วทำการ Build target ใหม่จนกว่าหน้าต่าง Output จะปรากฏข้อความ 0 Error ดังรูปที่ 1.16



รูปที่ 1.16 รูปแสดงการตัวอย่างการเขียนโปรแกรมภาษาซีและ Build target ได้ 0 Error

	ใบเนื้อหา	หน้าที่ 25
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พืชคณิตมูล และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

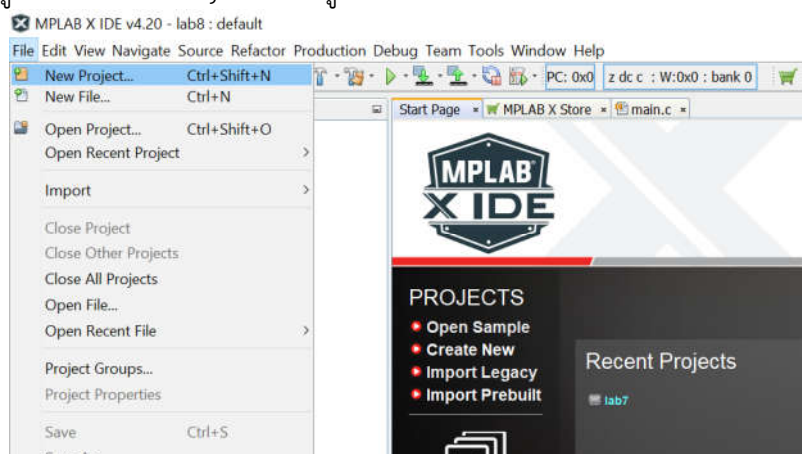
## 7.2 การใช้งานโปรแกรม MPLAB X สำหรับเขียนโปรแกรมภาษาซีของไมโครคอนโทรลเลอร์ตระกูล PIC16F

1. ให้ทำการดับเบิลคลิกที่ไอคอนโปรแกรม MPLAB X IDE ด้านหน้า Desktop ตามรูปที่ 1.17



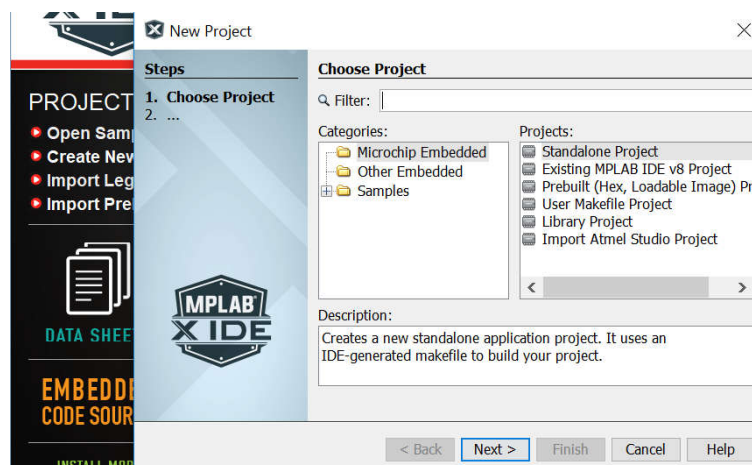
รูปที่ 1.17 แสดงรูปไอคอนของโปรแกรม MPLAB X IDE V4.20

2. คลิกที่เมนู File > New Project ตามรูปที่ 1.18




รูปที่ 1.18 แสดงวิธีการสร้าง New project

3. เมื่อปรากฏหน้าต่างต่าง New project Choose Project ตามรูปที่ 1.19 ให้คลิกที่ปุ่ม Next หลังจากนั้นจะปรากฏหน้าต่างให้เลือก Select Device ให้ทำการเลือกไมโครคอนโทรลเลอร์เบอร์ PIC16F887 ตามรูปที่ 1.20 แล้วคลิกปุ่ม Next

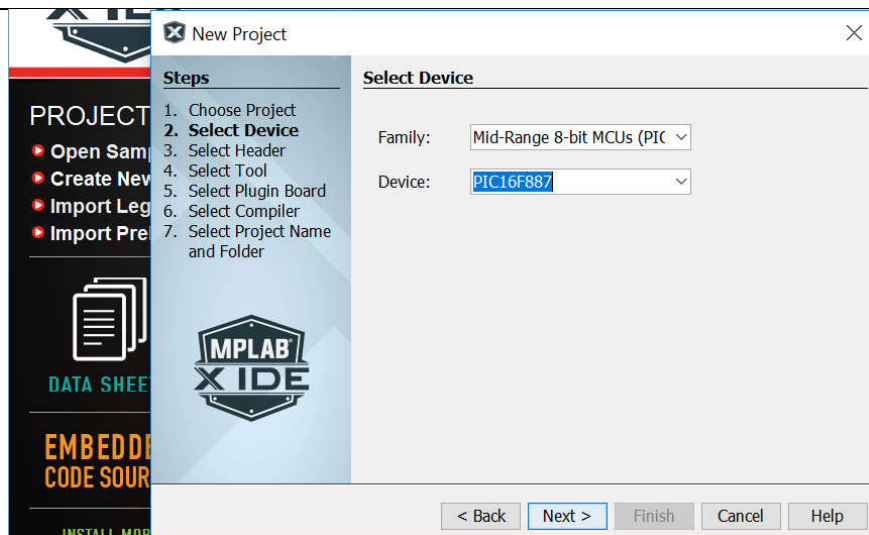


รูปที่ 1.19 แสดงหน้าต่าง New project Choose Project



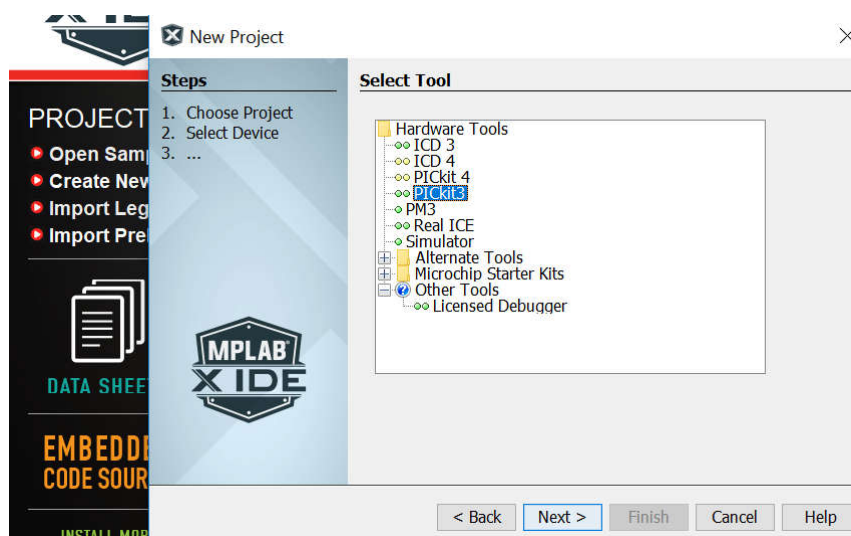
	ใบเนื้อหา	หน้าที่ 26
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย ฟิชคณิตบูลีน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์



รูปที่ 1.20 แสดงหน้าต่าง Select Device


4. จากข้อ 3 เมื่อคลิกที่ปุ่ม Next แล้วจะปรากฏหน้าต่าง Select Tool ให้ทำการเลือก PICkit3 แล้วทำการคลิกที่ปุ่ม Next ดังรูปที่ 1.21



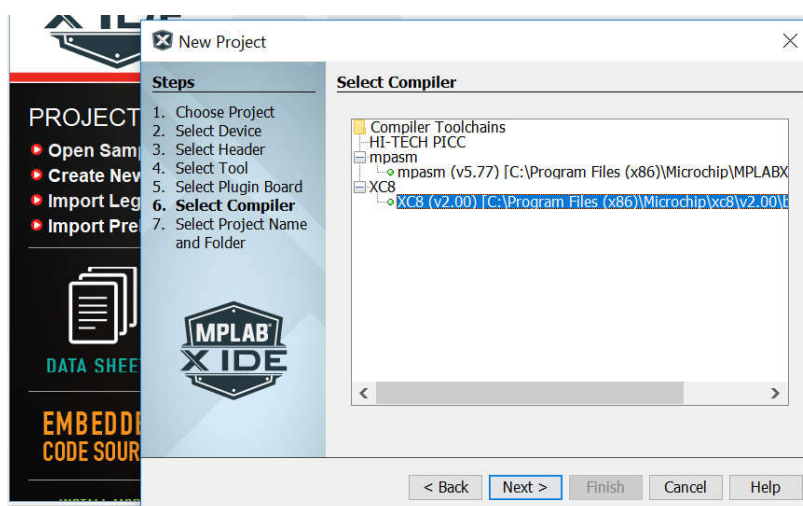
รูปที่ 1.21 แสดงหน้าต่าง Select Tool

5. จากข้อ 4 เมื่อคลิกที่ปุ่ม Next แล้วจะปรากฏหน้าต่าง Select Compiler ให้ทำการเลือก XC8 แล้วทำการคลิกที่ปุ่ม Next ดังรูปที่ 1.22



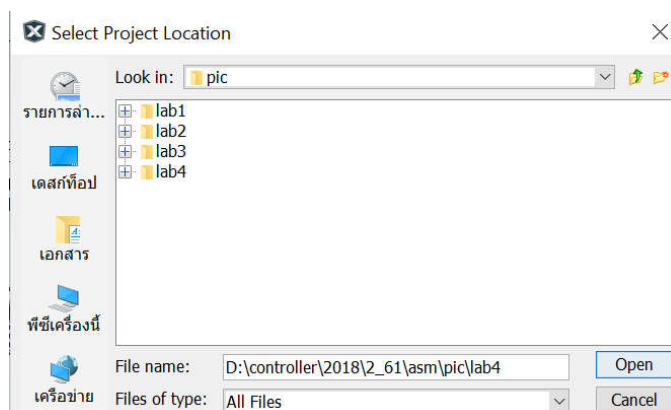
	ใบเนื้อหา	หน้าที่ 27
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พืชคณิตบูลีน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์




รูปที่ 1.22 หน้าต่างแสดง Select Compiler

6. จากข้อ 5 เมื่อคลิกที่ปุ่ม Next แล้วจะปรากฏหน้าต่าง Select Project Name and Folder ซึ่งในช่อง Project Location ให้ทำการคลิกที่ปุ่ม Browse แล้วจะปรากฏหน้าต่าง Select Project Location ดังรูปที่ 1.23 ให้ทำการเลือก Drive เลือก Folder ที่ต้องการจะบันทึก Project File ซึ่งในช่อง File name จะปรากฏ path ที่ต้องการจะบันทึก Project File แล้วคลิกปุ่ม Open เพื่อกลับเข้าหน้าต่าง Select Project Name and Folder

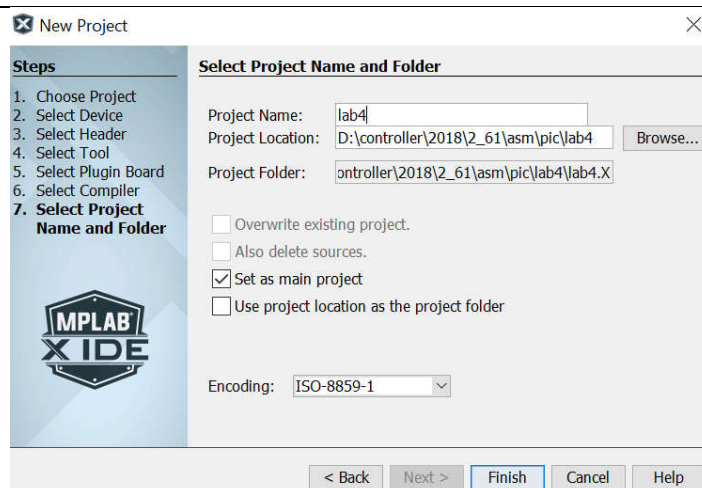


รูปที่ 1.23 แสดงหน้าต่าง Select Project Location

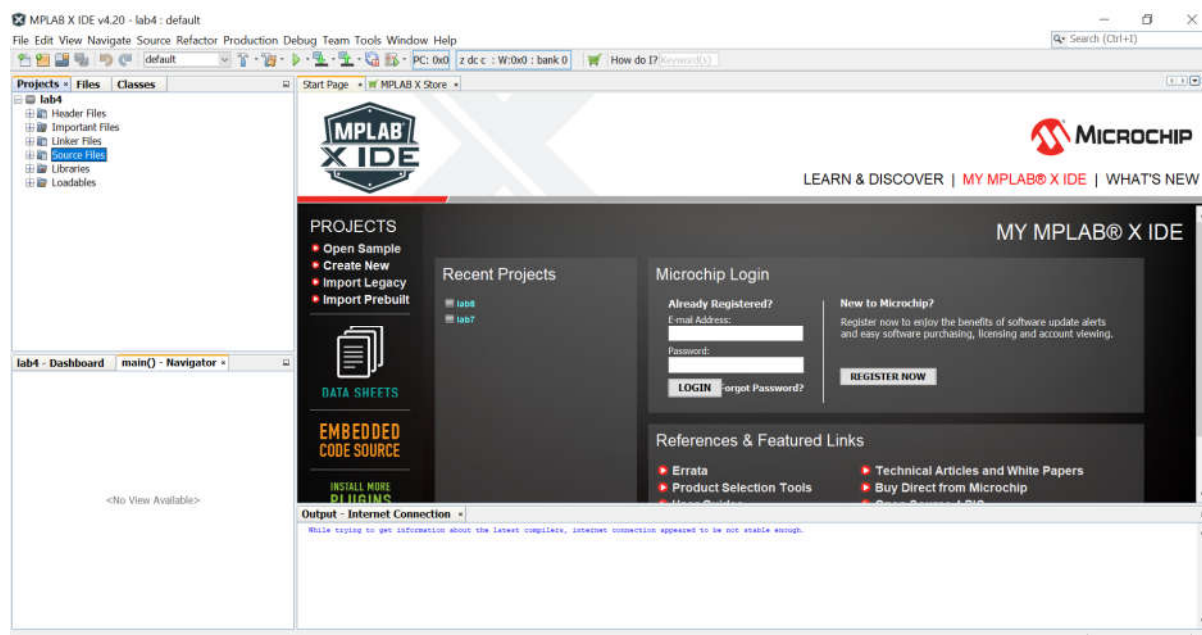
7. จากข้อ 6 เมื่อกลับเข้าสู่หน้าต่าง Select Project Name and Folder ในช่อง Project Name ให้ทำการพิมพ์ชื่อ Project Name ที่เราต้องการโดยที่ไม่ต้องใส่นามสกุลของไฟล์ แล้วให้คลิกที่ปุ่ม Finish เพื่อสิ้นสุดการสร้างโปรเจกต์ไฟล์ดังรูปที่ 1.24 และ 1.25

	ใบเนื้อหา		หน้าที่ 28
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย ฟิชคณิตบูลีน และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์




รูปที่ 1.24 แสดงหน้าต่าง Select Project Name and Folder

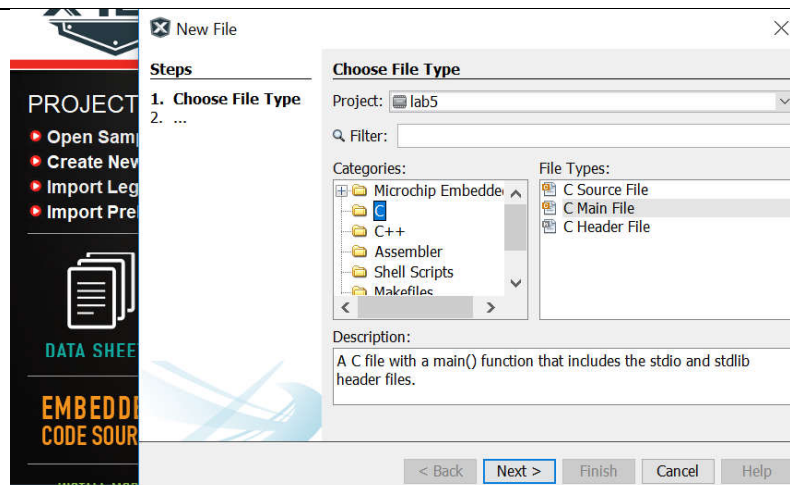


รูปที่ 1.25 แสดงการเตรียมการสร้างไฟล์ main.c

8. เมื่อปรากฏหน้าจอโปรแกรม MPLAB X IDE ให้คลิกที่ข้อความ Source Files แล้วให้เข้าเมนู File > New File จะปรากฏหน้าต่าง New File ดังรูปที่ 1.26 แล้วให้ทำการเลือก C ในช่อง Categories และเลือก C Main File ในช่อง File Types แล้วทำการคลิกที่ปุ่ม Next

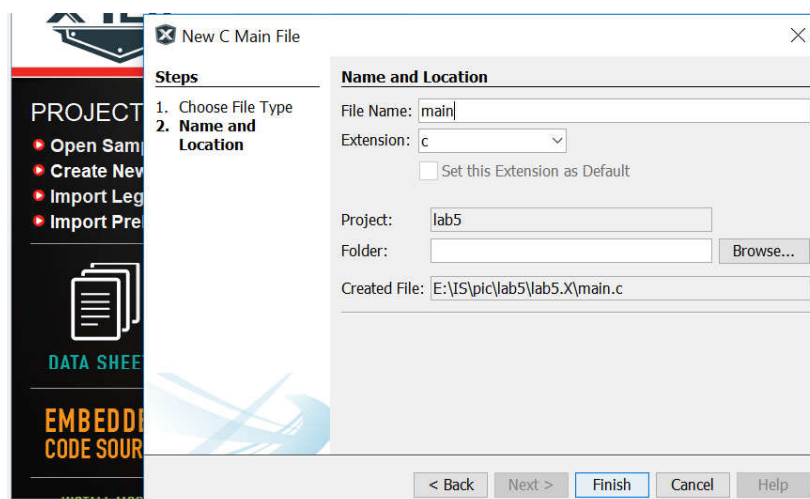
	ใบเนื้อหา		หน้าที่ 29
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พิชณนศิลป์ และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์




รูปที่ 1.26 หน้าต่างแสดงให้เลือกชนิดของไฟล์ที่ต้องการสร้าง

9. จากข้อ 8 เมื่อปรากฏหน้าต่าง Name and Location ในช่อง File Name ให้ทำการพิมพ์ชื่อ File Name ชื่อ main โดยที่ไม่ต้องใส่นามสกุลของไฟล์ แล้วให้คลิกที่ปุ่ม Finish เพื่อสิ้นสุดการสร้างไฟล์ ภาษา C ดังรูปที่ 1.27

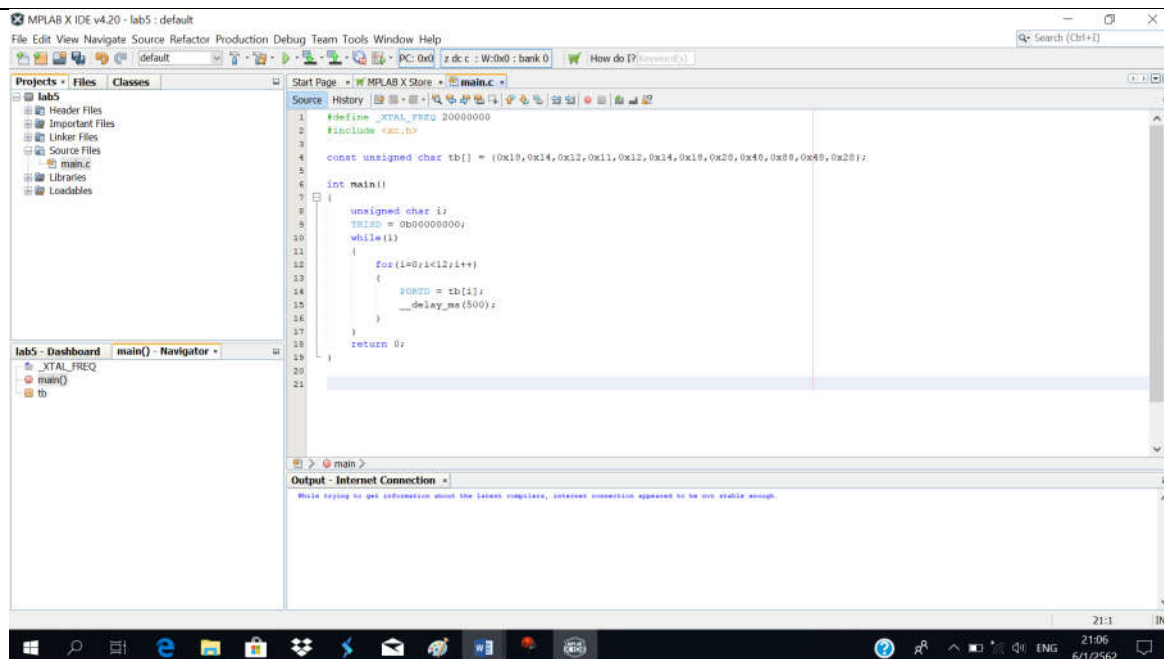


รูปที่ 1.27 รูปหน้าต่างแสดงการสร้างไฟล์ main.c

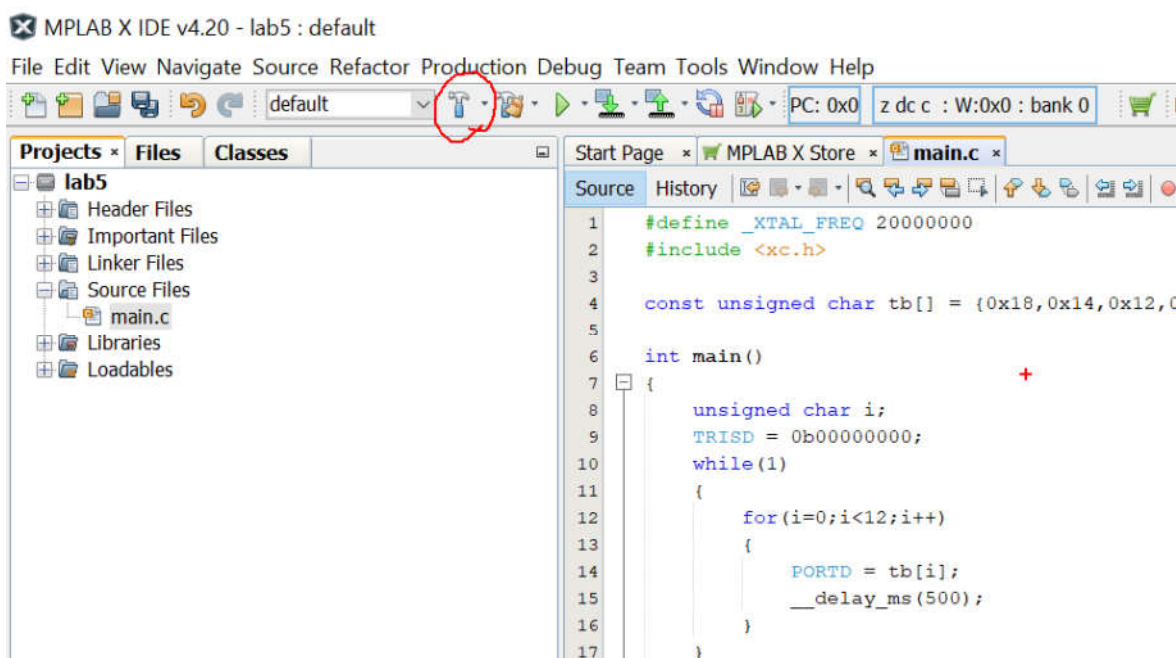
10. เมื่อกลับเข้าสู่โปรแกรม MPLAB X IDE ให้ทำการเขียนโปรแกรมภาษา C บริเวณ Work Sheet ดังรูปที่ 1.28 เมื่อทำการเขียนโปรแกรมเรียบร้อยแล้วให้ทำการ Compile โปรแกรมโดยคลิกที่ไอคอนเมนู Build Main Project ดังรูปที่ 1.29 แล้วให้สังเกตบริเวณหน้าต่าง Output ของโปรแกรมด้านล่างว่ามีข้อความ Loading Completed หรือไม่ ถ้าไม่ให้ทำการแก้ไขโปรแกรมแล้วทำการ Build Main Project ใหม่จนกว่าหน้าต่าง Output จะปรากฏข้อความ Loading Completed ดังรูปที่ 1.30

	ใบเนื้อหา	หน้าที่ 30
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย ฟิชคณิตบูลีน และการลดรูปสมการ	


ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์



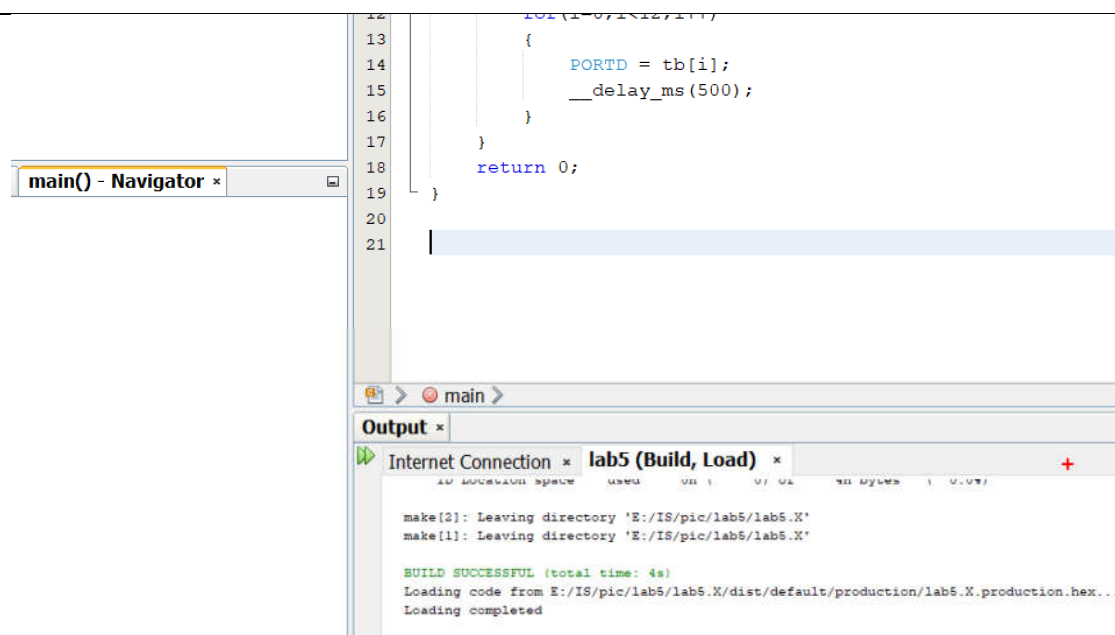
รูปที่ 1.28 รูปแสดงการตัวอย่างการเขียนโปรแกรมภาษาซีของไมโครคอนโทรลเลอร์ PIC16F887



รูปที่ 1.29 รูปแสดงตำแหน่งของเครื่องมือ Build Main Project

	ใบเนื้อหา		หน้าที่ 31
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย ฟิชคณิตบูลีน และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์



รูปที่ 1.30 รูปแสดงการ Build Main Project แล้วหน้าต่าง Output แสดงการ Build Success

### 7.3 การใช้งานโปรแกรม AVR Studio 6.2 สำหรับเขียนโปรแกรมภาษาซีของไมโครคอนโทรลเลอร์ตระกูล AVR


1. ให้ทำการดับเบิลคลิกที่ไอคอนโปรแกรม AVR Studio 6.2 ด้านหน้า Desktop ตามรูปที่ 1.31



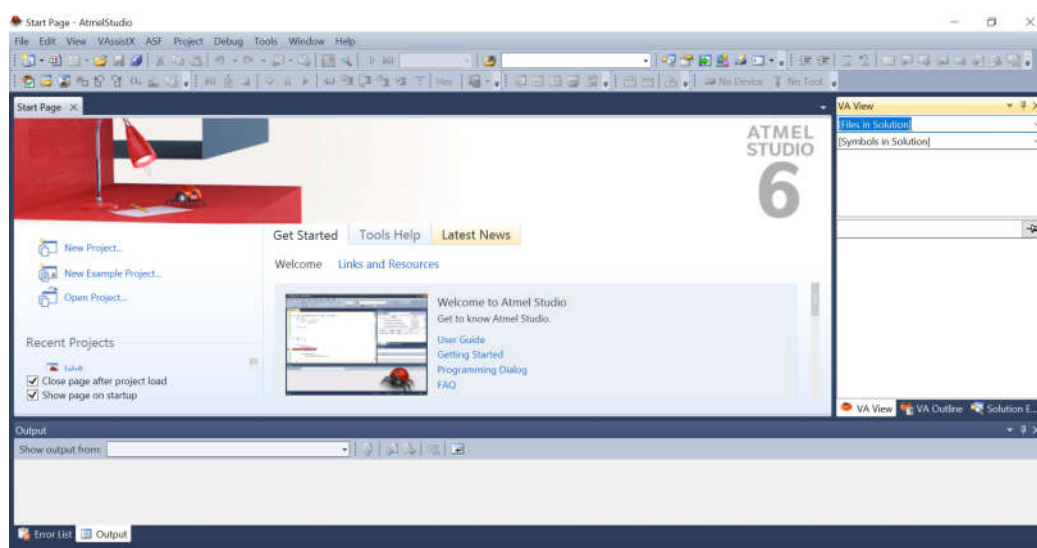
รูปที่ 1.31 รูปแสดงไอคอนของโปรแกรม AVR Studio 6.2

2. เมื่อปรากฏหน้าต่างโปรแกรมตามรูปที่ 1.32 ถ้าต้องการสร้างโปรเจกใหม่ให้ทำการเลือกเมนู File > New > Project จะปรากฏหน้าต่าง New Project ตามรูปที่ 1.33

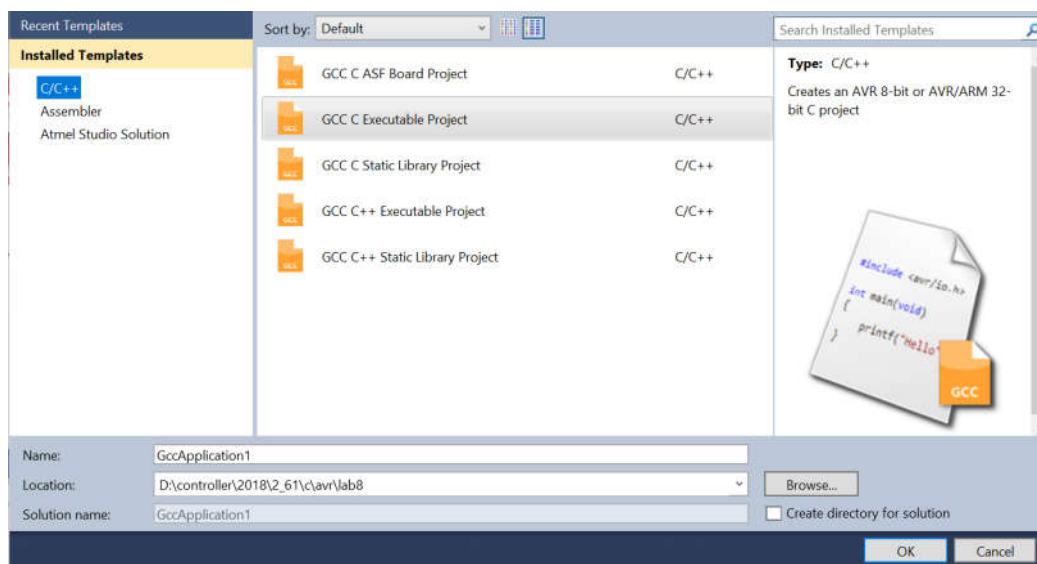


	ใบเนื้อหา		หน้าที่ 32
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย ฟิสิกส์มูลฐาน และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์




รูปที่ 1.32 รูปภาพแสดงหน้าต่างของโปรแกรม AVR Studio 6.2



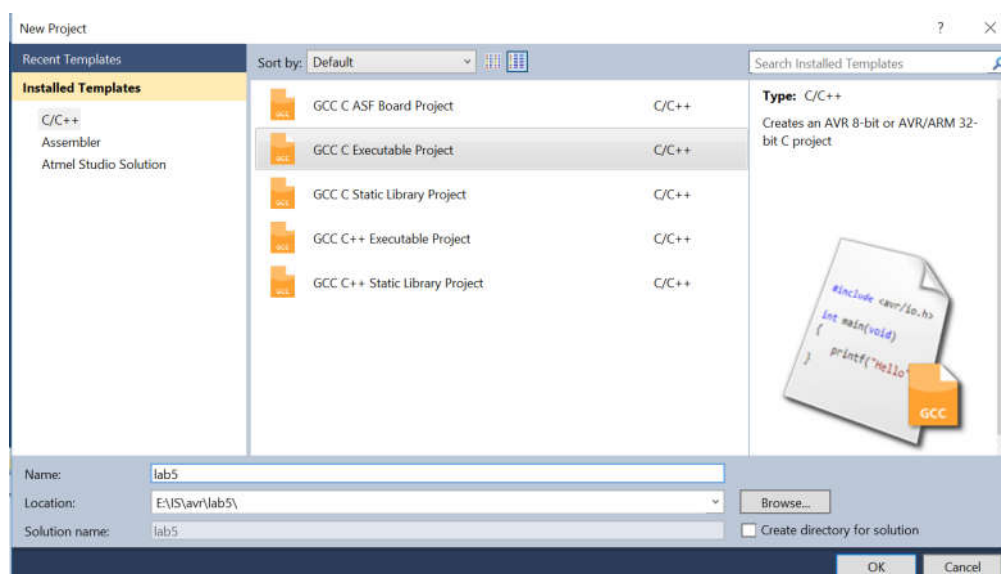
รูปที่ 1.33 แสดงรูปหน้าต่างของ New Project

3. จากขั้นตอนที่ 2 เมื่อปรากฏหน้าต่าง New Project ตามรูปที่ 1.33 ให้ทำการเลือกข้อความ C/C++ และ GCC C Executable Project โดยในช่อง Location ให้ทำการเลือก Drive เลือก Folder ที่ต้องการเก็บโปรเจกต์ไฟล์ที่ต้องการสร้างขึ้นโดยการพิมพ์ path หรือคลิกที่ปุ่ม Browse แล้วทำการทำการเลือก Drive เลือก Folder ที่ต้องการเก็บโปรเจกต์ไฟล์ที่ต้องการสร้างขึ้น ส่วนในช่อง Name ให้ทำการพิมพ์ชื่อโปรเจกต์ที่ต้องการจะบันทึกโดยไม่ต้องใส่นามสกุลของไฟล์ดังรูปที่ 1.34 แล้วทำการคลิกปุ่ม OK



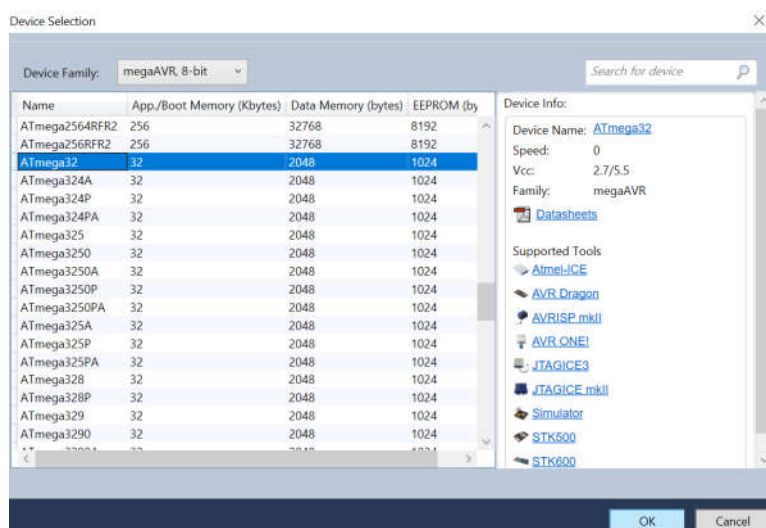
	ใบเนื้อหา	หน้าที่ 33
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิษคณิตบูลีน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์




รูปที่ 1.34 แสดงรูปหน้าต่างของ New Project ที่ตั้งชื่อในช่อง name และเลือก location เรียบร้อยแล้ว

4. จากข้อ 3 เมื่อคลิกที่ปุ่ม Ok แล้วจะปรากฏหน้าต่าง Device Selection โดยในส่วนของ Device Family ให้เลือก megaAVR, 8bit แล้วให้ทำการคลิกเลือกไมโครคอนโทรลเลอร์บอร์ด Atmega32 แล้วทำการคลิกที่ปุ่ม OK ดังรูปที่ 1.35

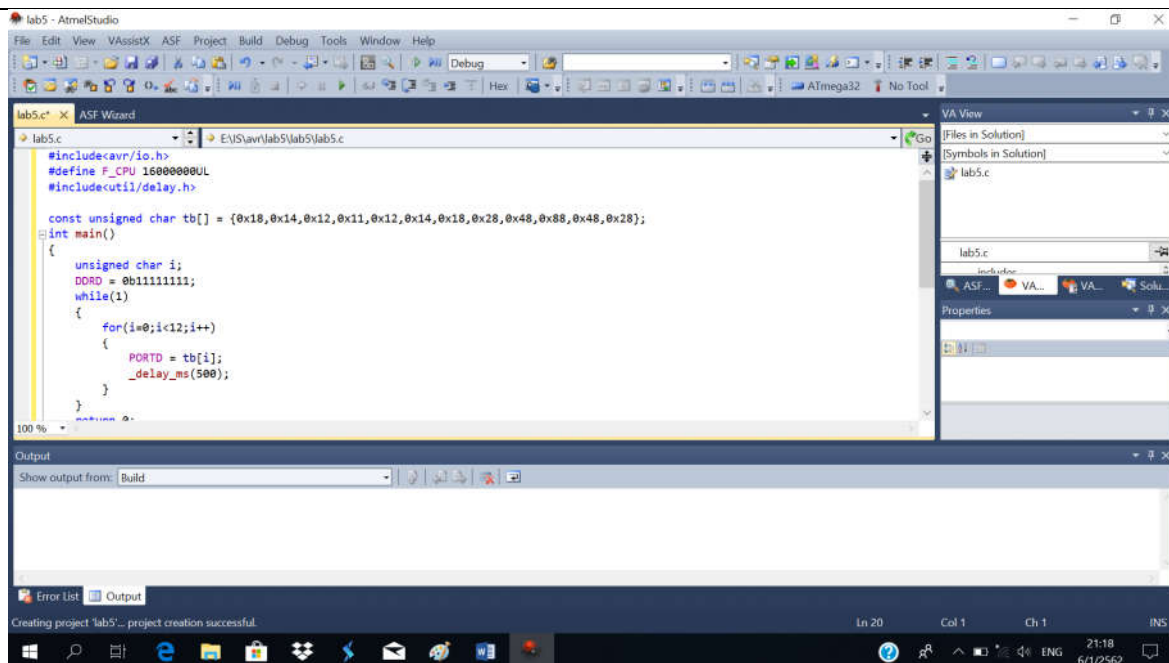


รูปที่ 1.35 รูปหน้าต่างแสดงการเลือกบอร์ดไมโครคอนโทรลเลอร์

5. เมื่อเข้าสู่โปรแกรม AVR Studio 6.2 ในส่วนของ Work Sheet ให้ทำการเขียนโปรแกรมภาษา C ของไมโครคอนโทรลเลอร์ AVR ให้เรียบร้อยดังรูปที่ 1.36

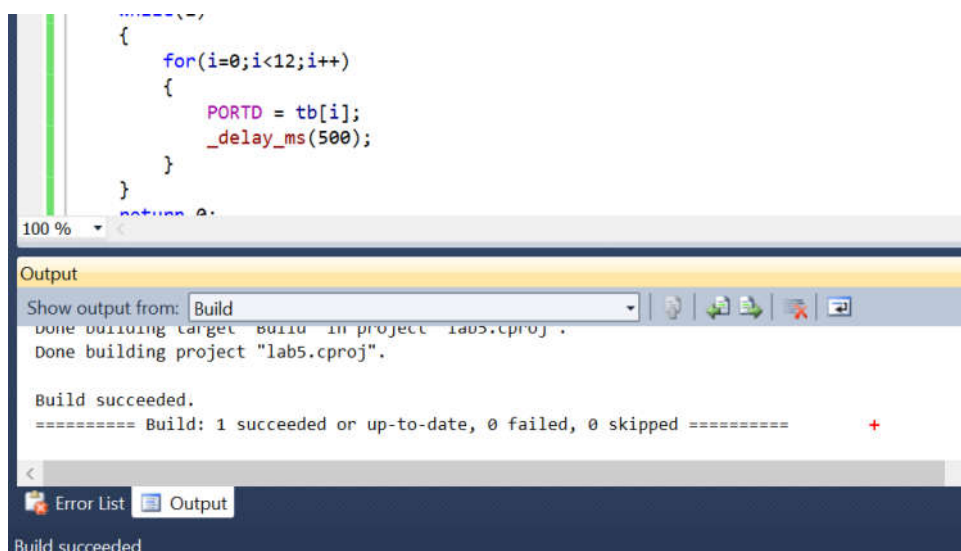
	ใบเนื้อหา		หน้าที่ 34
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 2
	ชื่อหน่วย พืชคณิตบูลีน และการลดรูปสมการ		

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์





รูปที่ 1.36 รูปแสดงตัวอย่างการเขียนโปรแกรมภาษาซีของไมโครคอนโทรลเลอร์ AVR

6. เมื่อทำขั้นตอนที่ 5 เรียบร้อยแล้วให้ทำการ Compile โปรแกรมโดยเข้าเมนู Build > Build Solution แล้วให้สังเกตบริเวณหน้าต่าง Output ของโปรแกรมด้านล่างว่ามีข้อความ Build succeeded. หรือไม่ ถ้าไม่ให้ทำการแก้ไขโปรแกรมแล้วทำการ Build ใหม่จนกว่าหน้าต่าง Output จะปรากฏข้อความ Build succeeded. ดังรูปที่ 1.37



รูปที่ 1.37 รูปแสดงการ Build แล้วไม่เกิดข้อผิดพลาด

	แบบฝึกหัด	หน้าที่ 1
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย พิชณณบูลีน และการลดรูปสมการ	
ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์		
<p><u>คำสั่ง</u> จงตอบคำถามต่อไปนี้ให้ถูกต้อง</p> <ol style="list-style-type: none"> <li>จงอธิบายโครงสร้างของการเขียนโปรแกรมภาษาซี</li> <li>จงอธิบายวิธีการประกาศตัวแปรในภาษาซี</li> <li>จงอธิบายความหมายของคำสั่ง unsigned int result;</li> <li>จงอธิบายข้อแตกต่างของการประกาศตัวแปรแบบ Global กับ Local</li> <li>การประกาศตัวแปรแบบ static มีความสำคัญอย่างไร</li> <li>การประกาศตัวแปรแบบ volatile มีความสำคัญอย่างไร</li> <li>การเขียนฟังก์ชันย่อยมีกี่รูปแบบอะไรบ้าง</li> <li>ถ้าเขียน Function Prototype เป็น float divData(float a,float b); มีความหมายว่าอย่างไร</li> </ol>		

	แบบฝึกหัด	หน้าที่ 2
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 2
	ชื่อหน่วย ฟิชคณิตบูลีน และการลดรูปสมการ	

ชื่อเรื่อง ภาษาซีกับไมโครคอนโทรลเลอร์

9. จงอธิบายคำสั่ง ++x;

.....

.....

.....

10. จงอธิบายคำสั่ง x++;

.....

.....

.....

11. จงอธิบายคำสั่ง  $y = x \mid z$ ;

.....

.....

.....

12. จงอธิบายคำสั่ง

If(a>5) b=3;

else b=2;

.....

.....

.....

13. จงอธิบายคำสั่ง

do{

x = x + 3;

}while(x <= 10);

.....

.....

.....

14. จงอธิบายคำสั่ง TRISB = 0x0f;

.....

.....

.....

15. จงอธิบายคำสั่ง DDRC = 0b11110000;

.....

.....

.....