บทที่ 3 การเขียนโปรแกรมภาษาซีสำหรับไมโครคอนโทรลเลอร์ MCS-51

3.1 บทนำ

ภาษาซีเป็นภาษาที่นักพัฒนางานทางด้านไมโครคอนโทรลเลอร์นำมาประยุกต์ใช้งานอย่าง แพร่หลาย ซึ่งจะเห็นได้ว่าบริษัทที่ผลิตไมโครคอนโทรลเลอร์ส่วนใหญ่เลือกใช้ภาษาซีในการพัฒนา โปรแกรมและมีนักพัฒนาได้เขียนโค้ดต้นฉบับภาษาซีรองรับการประยุกต์ใช้งานไว้เป็นจำนวนมาก ส่งผล ให้การพัฒนางานด้านที่เกี่ยวข้องกับการเชื่อมต่อกับอุปกรณ์ภายนอกในรูปแบบต่าง ๆ เช่น พอร์ต อนุกรม พอร์ตยูเอสบี หรือเครือข่ายคอมพิวเตอร์ าลา มีความคล่องตัวในการนำไปประยุกต์ใช้งานเป็น อย่างมาก ในอนาคตหากฮาร์ดแวร์ของไมโครคอนโทรลเลอร์มีการพัฒนาสมรรถนะให้สูงขึ้น เช่น มี ความเร็วสูงขึ้นหรือมีไมโครคอนโทรลเลอร์ตระกูลอื่น ๆ ที่อาจได้รับความนิยมเพิ่มขึ้นมา แต่ภาษาซียังคง เป็นพื้นฐานในการนำไปประยุกต์ใช้ได้อย่างต่อเนื่อง ในบทนี้จึงขออธิบายส่วนสำคัญที่เกี่ยวข้องกับการ เขียนโปรแกรมภาษาซีสำหรับไมโครคอนโทรลเลอร์

3.2 พื้นฐานภาษาซีสำหรับไมโครคอนโทรลเลอร์ MCS-51

การเขียนโปรแกรมภาษาซีสำหรับไมโครคอนโทรลเลอร์ MCS-51 ยังคงลักษณะตามมาตรฐาน ANSI-C (American National Standards Institute) และยังมีส่วนเพิ่มเติมสำหรับการเขียน โปรแกรมควบคุมไมโครคอนโทรลเลอร์โดยเฉพาะ ช่วยให้ผู้ที่มีความคุ้นเคยกับภาษาซีสามารถเข้าใจได้ โดยง่าย เนื้อหาในหัวข้อนี้อธิบายพื้นฐานโปรแกรมภาษาซี เพื่อเป็นแนวทางในการศึกษาการโปรแกรม ภาษาซีสำหรับไมโครคอนโทรลเลอร์ MCS-51 ในระดับที่สูงขึ้นต่อไป

3.2.1 โครงสร้างภาษาซี

โครงสร้างของโปรแกรมที่เขียนด้วยภาษาซีประกอบด้วยรายละเอียด ดังนี้

 คำสั่งชี้แนะตัวประมวลผลก่อน (Preprocessor directive) เป็นชุดคำสั่งที่ใช้ในการ จัดเตรียมข้อมูลไว้สำหรับการประมวลผลซึ่งทำหน้าที่ระบุให้ตัวแปรโปรแกรมไปดึงแฟ้มข้อมูลอื่นที่ กำหนดมาแปลร่วมด้วย แฟ้มข้อมูลเหล่านี้อาจเป็นแฟ้มข้อมูลมาตรฐานที่มีให้แล้วในภาษาซี หรือเป็น แฟ้มข้อมูลที่เขียนขึ้นมาใหม่ก็ได้ โดยส่วนใหญ่มักเป็นแฟ้มข้อมูลที่มีนามสกุล .h

2. การประกาศ (Declarations) เป็นส่วนที่ใช้ประกาศค่าตัวแปรหรือฟังก์ชัน และค่า อื่นๆ ที่ต้องการ โดยต้องมีการประกาศค่าเหล่านี้ก่อนการใช้งาน 3. ส่วนคำสั่ง ประกอบด้วยส่วนต่างๆ ดังนี้

1) นิพจน์ (Expression) เป็นการกระทำระหว่างตัวดำเนินการ (Operators) กับตัว ถูกดำเนินการ (Operands) เพื่อให้เกิดค่าใดค่าหนึ่ง

2) ข้อความสั่ง (Statement) คือคำสั่งที่ใช้ในการทำงานตามความต้องการของ ผู้เขียนโปรแกรม

 3) ฟังก์ชัน เป็นส่วนประกอบของโปรแกรมที่กำหนดให้ทำงานอย่างใดอย่างหนึ่งจน เสร็จสิ้น

 4. ฟังก์ชัน main เป็นฟังก์ชันที่ต้องมีการประกาศทุกครั้งในการเขียนโปรแกรม เพราะ โปรแกรมจะเริ่มต้นทำงานที่ฟังก์ชันหลักนี้และเป็นฟังก์ชันที่ใช้ในการเรียกฟังก์ชันอื่นๆ ในการทำงาน ในที่นี้ขอยกตัวอย่างโปรแกรมที่เขียนด้วยภาษาซี แสดงดังภาพที่ 3.1

#include <stdio.h></stdio.h>
char a;
int main (void)
{
a = 23;
printf ("Hello World");
return a;
}

ภาพที่ 3.1 ตัวอย่างโปรแกรมภาษาซี

จากโปรแกรมดังกล่าวผลผลลัพธ์ของการรันโปรแกรมจะได้ข้อความบนหน้าจอว่า "Hello World" โดยมีคำอธิบายประกอบโปรแกรมในแต่ละส่วนดังนี้

1) #include <stdio.h> คือส่วนของคำสั่งชี้แนะตัวประมวลผลก่อนเพื่อระบุตัวแปร โปรแกรมให้นำแฟ้มข้อมูล stdio.h มารวมด้วย

2) char a คือส่วนของการประกาศตัวแปร a โดยกำหนดให้เป็นตัวแปรที่มีชนิดเป็น ตัวอักษร

3) main คือส่วนของฟังก์ชัน main ซึ่งเป็นฟังก์ชันหลักของโปรแกรมโดยโปรแกรมจะ เริ่มทำงานที่ฟังก์ชันนี้

4) a = 23 คือส่วนคำสั่งกำหนดให้ค่าตัวแปร a มีค่าเป็น 23

5) printf ("Hello World") คือส่วนคำสั่งการใช้ฟังก์ชัน printf เพื่อพิมพ์ข้อความที่อยู่ ในเครื่องหมายอัญประกาศออกทางอุปกรณ์นำออกมาตรฐาน

3.2.2 การแปลโปรแกรมและโปรแกรมเชื่อมโยงในภาษาซี

การพัฒนาโปรแกรมภาษาซีแสดงดังภาพที่ 3.2 โดยประกอบด้วยขั้นตอน ดังนี้ 1. สร้างแฟ้มข้อมูลต้นฉบับ (Source file) โดยมีแฟ้มข้อมูลนามสกุล .c ขึ้นมา โดยใช้ โปรแกรมที่สามารถเขียนแฟ้มข้อมูลใดๆ ก็ได้ โดยอยู่ในรูปแบบของการเขียนโปรแกรมภาษาซี

2. ตัวแปลโปรแกรมของภาษาซี (C Compiler) จะทำการแปลงโปรแกรมต้นฉบับเก็บไว้ ในอีกแฟ้มข้อมูลหนึ่งเรียกว่าแฟ้มข้อมูลออบเจ็กต์ (Object file) ที่มีนามสกุล .obj

 3. โปรแกรมเชื่อมโยง (Linker) ทำการตรวจสอบว่าโปรแกรมที่เขียนขึ้นนั้น มีการเรียกใช้ งานฟังก์ชันมาตรฐานใดจากคลังโปรแกรมภาษาซี (C Library) หรือไม่ โดยโปรแกรมเชื่อมโยงทำการ รวมเอาฟังก์ชันเหล่านั้นเข้ากับแฟ้มข้อมูลออบเจ็กต์แล้วจะได้แฟ้มข้อมูลที่กระทำการได้ (Executable file) โดยมีนามสกุลเป็น .exe



ภาพที่ 3.2 แผนภาพบล็อกแสดงขั้นตอนการแปลโปรแกรมและโปรแกรมเชื่อมโยงในภาษาซี

3.2.3 โปรแกรม C51

สำหรับซอฟต์แวร์ในการเขียนโปรแกรมและแปลโปรแกรมภาษาซีของไมโครคอนโทรลเลอร์ MCS-51 หรือที่เรียกว่า C51 ที่ปัจจุบันได้รับความนิยมและมีการใช้งานอย่างแพร่หลายได้แก่ Keil C51 ซึ่งมีรูปแบบเป็นไปตามมาตรฐาน ANSI-C และได้เพิ่มความสามารถทางด้านการเขียนโปรแกรมควบคุม ไมโครคอนโทรลเลอร์โดยเฉพาะ ดังนั้นเอกสารประกอบการสอนเล่มนี้อ้างอิงการใช้งานตัวแปรโปรแกรม Keil C51 เป็นหลักในการอธิบาย โดย ในปัจจุบันผู้ผลิตซอฟต์แวร์ Keil C51 ได้พัฒนาเครื่องมือในพัฒนา โปรแกรมที่ได้รวมส่วนประกอบต่างๆ ได้แก่ แอสเซมเบลอร์ (Assembler) ตัวแปรโปรแกรม โปรแกรม เชื่อมโยง และโปรแกรมตรวจแก้จุดบกพร่อง (Debugger) ไว้ในตัวเดียวกันเพื่อความสะดวกในการ พัฒนาโปรแกรม ซึ่งเรียกว่า Integrated Development Environment (IDE) โดยซอฟต์แวร์ Keil ได้ พัฒนา IDE สำหรับไมโครคอนโทรลเลอร์ MCS-51 ชื่อว่า μ Vision ซึ่งจะได้แนะนำการใช้งานโปรแกรมนี้ ในหัวข้อต่อไป

3.3 แนะนำโปรแกรม Keil μ Vision

โปรแกรม Keil µVision ปัจจุบันได้ถูกพัฒนาเป็นรุ่นที่ 4 สามารถบรรจุลงมาทดลองใช้งานได้ที่ เวปไซต์ www.keil.com เมื่อทำการติดตั้งโปรแกรมเรียบร้อยแล้วเราสามารถใช้เครื่องมือ Keil µVisionช่วยในการพัฒนาโปรแกรม เนื้อหาในส่วนนี้ได้อธิบายการใช้งาน Keil µVisionในการ ประยุกต์ใช้งานสำหรับไมโครคอนโทรลเลอร์ MCS-51

3.3.1 กระบวนการพัฒนาโปรแกรม Keil µVision

การพัฒนาโปรแกรมโดย Keil μ Vision มีกระบวนการพัฒนาโปรแกรมแสดงดังภาพที่ 3.3 อธิบายเป็นขั้นตอนได้ ดังนี้

 เริ่มต้นสร้างโปรเจคโดยทำการเลือกบริษัทผู้ผลิตและเบอร์ของไมโครคอนโทรลเลอร์ที่ ต้องการใช้งานจากฐานข้อมูล

 สร้างแฟ้มข้อมูลต้นฉบับภาษา C/C++ หรือภาษาแอสเซมบลี เพื่อทำการเขียน โปรแกรมหรือแก้ไขโปรแกรม

3. จัดทำโปรแกรมประยุกต์โดยผ่านการจัดการโปรเจคของโปรแกรม Keil μ Vision

4. ตรวจแก้จุดบกพร่องของโปรแกรมข้อมูลต้นฉบับเพื่อทดสอบความถูกต้องของ

โปรแกรม

ฮาร์ดแวร์

5. บรรจุโปรแกรมลงหน่วยความจำแฟลชหรือเอสแรมพร้อมทั้งทดสอบการทำงานกับ

µVisi Integrated Develo	on IDE pment Environment
µVision Pro	ject Manager
1	<u> </u>
C/C++ Compiler	Macro Assembler
Linker	/ Locator
. Aliaian	Dobuggor
μνιsion	Debuggei

ภาพที่ 3.3 กระบวนการในการพัฒนาโปรแกรมของ Keil μ Vision

ที่มา: ARMKIEL Microcontrollers Tools, 2014

3.3.2 แนวคิดการวางตำแหน่งพื้นที่หน้าต่างของ Keil µVision

เมื่อทำการติดตั้งโปรแกรมเรียบร้อยแล้ว ทำการเปิดโปรแกรม Keil μ Visionจะได้ โปรแกรมแสดงดังภาพที่ 3.4 จะเห็นว่า Keil μ Visionประกอบด้วยพื้นที่หลัก 3 พื้นที่ ได้แก่

 พื้นที่หน้าต่างการจัดการโปรเจค (Project windows area) เป็นหน้าต่างที่ระบุถึง รายละเอียดของโปรเจคในด้านต่างๆ เช่น หน้าต่างแสดงแฟ้มข้อมูลของโปรเจคหรือหน้าต่างแสดงค่า เรจิสเตอร์ต่างๆ ของซีพียู เป็นต้น

 พื้นที่หน้าต่างการตรวจแก้งาน (Editor windows area) เป็นหน้าต่างที่แสดง รายละเอียดของโปรแกรมที่กำลังพัฒนา ซึ่งนักพัฒนาโปรแกรมสามารถปรับแก้โปรแกรมต้นฉบับ ตรวจสอบประสิทธิภาพและวิเคราะห์การทำงานของโปรแกรม ตลอดจนตรวจสอบโปรแกรม ภาษาแอสเซมบลิในพื้นที่ส่วนนี้

 พื้นที่หน้าต่างการนำออก (Output windows area) เป็นหน้าต่างที่จัดเตรียมข้อมูล นำออกต่างๆ เช่น การแก้จุดบกพร่องของโปรแกรม การแสดงพื้นที่หน่วยความจำ การแสดงค่าตัวแปร ต่างๆ เป็นต้น

Project Windows		Editor Wind	dows			
C:\Users\Vostro\Desktop\Source code-MCS51\Chapter 3\Test.uvproj - @Vision4		-				_ 0 X
Hie Edit View Project Hash Debug Penpherais Tools SVCS Window Help	k time 💌	1 A # 1 Q I • 0	0 0 III 4			
	- *-					
Project 4 🖬 Disassembly						¢ (
Haraget 1 Horaget 2 Horaget 2	/* Delay var /* LED var */ /* Loop forev	*/ /* rer */				
*) Test.c						▼ ×
1 #include <regx51.h> 2 3⊖void main (void) { 4 unsigned int i; 5 unsigned char j; 6 7 0 70 while (1) { 80 for (j=0x0; j< 0x80; j<<==)</regx51.h>	/* D /* L () { /* B (/* D	elay var */ ED var */ oop forever * link LED 0, 1 utput to LED elay for 1000	/ , 2, 3, 4, 5, Port */ 0 Counts */	6 */		E
Command	4 🖬	Call Stack + Locals				¢ (c
Running with Code Sire Limit: 2K Load "C:\\Users\\Vostro\\Desktop\\Source code-MCSS1\\Chapter 3\\Test" < >	r F	Name MAIN i j	Location/Value C:0x0800 0x0000 0x00	Type uint uchar		
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet Break	Access	Call Stack + Loc	als Memory 1			
		Simulation	t1:0	.00019450 sec	L:8 C:1	CAP NUM SCRL OVR R/M
Output	Windows					

ภาพที่ 3.4 การวางตำแหน่งพื้นที่หน้าต่างของ Keil μ Vision

3.3.3 รูปแบบการทำงานของ Keil µVision

ในการพัฒนาโปรแกรมของ Keil μ Vision ประกอบด้วยรูปแบบการทำงานที่สำคัญ ดังนี้

 รูปแบบการพัฒนาโปรแกรม (Build mode) เป็นรูปแบบมาตรฐานการใช้งาน โปรแกรม Keil µVision ทำหน้าที่รองรับการเขียนและแก้ไขโปรแกรม การปรับแต่งการทำงานของ โปรเจค การกำหนดค่าด้านฮาร์ดแวร์ของอุปกรณ์ไมโครคอนโทรลเลอร์ ซึ่งสามารถนำโปรแกรมที่พัฒนา มาทำการแปลโปรแกรม สร้างโปรแกรมเชื่อมโยงและแปลภาษาแอสเซมบลี เพื่อตรวจสอบความ ความถูกต้องและข้อผิดพลาดของโปรแกรม

 รูปแบบการตรวจแก้จุดบกพร่องของโปรแกรม (Debug mode) ทำหน้าที่จำลองการ ทำงานของโปรแกรมที่พัฒนาขึ้นจากรูปแบบการพัฒนาโปรแกรม สามารถแสดงผลการนำออกได้อย่าง หลากหลาย เช่น แสดงการทำงานของช่องทางเข้า/ออกของไมโครคอนโทรลเลอร์ แสดงพื้นที่ หน่วยความจำ แสดงค่าตัวแปรต่างๆ เป็นต้น

3.3.4 แถบเครื่องมือของ Keil µVision

โปรแกรม Keil µVision มีแถบเครื่องมือที่สำคัญสำหรับการทำงานแต่ละรูปแบบ ดังนี้ 1. พื้นที่หน้าต่างแถบเครื่องมือเกี่ยวกับการจัดการแฟ้มข้อมูล (File toolbar) ประกอบด้วยเครื่องมือต่างๆ ที่สำคัญแสดงดังตารางที่ 3.1

ไอคอน	รายละเอียดของคำสั่ง
	[New file] เปิดหน้าต่างสำหรับเริ่มต้นเขียนโปรแกรม
2	[Open] เปิดแฟ้มข้อมูลที่ได้บันทึกไว้แล้วเพื่อนำมากระทำการต่าง ๆ เช่น นำแฟ้มข้อมูล
	มาแก้ไขโปรแกรม เป็นต้น
5	[Save] บันทึกแฟ้มข้อมูลที่เปิดอยู่ในปัจจุบัน
1	[Save all] บันทึกแฟ้มข้อมูลทั้งหมดที่เปิดอยู่ในปัจจุบัน
¥	[Cut] ตัดข้อความที่ถูกเลือกไปเก็บไว้ยังคลิปบอร์ด

ตารางที่ 3.1 แถบเครื่องมือที่ใช้สำหรับการจัดการแฟ้มข้อมูล

ตารางที่ 3.1 แถบเครื่องมือที่ใช้สำหรับการจัดการแฟ้มข้อมูล (ต่อ)

ไอคอน	รายละเอียดของคำสั่ง
	[Copy] คัดลอกข้อความถูกเลือกไปเก็บไว้ยังคลิปบอร์ด
2	[Paste] วางข้อความที่อยู่ในคลิปบอร์ดไปยังตำแหน่งที่ตัวชี้ตำแหน่งกระพริบอยู่
5	[Undo changes] ย้ายกลับไปยังการกระทำก่อนหน้า
6	[Redo changes] กลับมายังการกระทำครั้งสุดท้ายของ Undo
P	[Bookmark] ใช้สำหรับเซตหรือเคลื่อนย้ายเครื่องหมายคั่นหน้าที่ตำแหน่งที่ตัวชี้ ตำแหน่งกระพริบอยู่
12	[Previous bookmark] กำหนดให้ตัวชี้ตำแหน่งเคลื่อนย้ายไปยังตำแหน่งที่ทำ เครื่องหมายคั่นหน้าไว้ก่อนหน้า
1	[Next bookmark] กำหนดให้ตัวซี้ตำแหน่งเคลื่อนย้ายไปยังตำแหน่งที่ทำเครื่องหมาย คั่นหน้าถัดไป
1	[Clear All bookmarks] ยกเลิกการทำเครื่องหมายคั่นหน้าเอกสารที่เปิดอยู่ในปัจจุบัน ทั้งหมด
#	[Indent selection] เลื่อนบรรทัดของข้อความที่สนใจไปทางขวาหนึ่งจุดตั้งระยะ
	[Unindent selection] เลื่อนบรรทัดของข้อความที่สนใจไปทางซ้ายหนึ่งจุดตั้งระยะ
//≣	[Set comment] กำหนดบรรทัดของโปรแกรมที่สนใจให้เป็นหมายเหตุ
// x	[Remove comment] เปลี่ยนหมายเหตุในบรรทัดที่สนใจกลับให้เป็นโปรแกรม
0	[Debug session] สลับรูปแบบระหว่างการเขียนโปรแกรมและการแก้จุดบกพร่องของ โปรแกรม
٠	[Breakpoint] กำหนดหรือยกเลิกจุดพักในตำแหน่งที่ตัวชี้ตำแหน่งทำงาน

ตารางที่ 3.1 แถบเครื่องมือที่ใช้สำหรับการจัดการแฟ้มข้อมูล (ต่อ)

ไอคอน	รายละเอียดของคำสั่ง
1	[Translate] แปลโปรแกรมหรือแปลภาษาแอสเซมบลีแฟ้มข้อมูลที่กำลังตรวจแก้ใน
	ปัจจุบัน
	[Build] แปลโปรแกรมและเชื่อมโยงโปรแกรมเพื่อสร้างแฟ้มข้อมูลของโปรเจค
[##]	[Rebuild] แปลโปรแกรม แปลภาษาแอสเซมบลี และเชื่อมโยงโปรแกรมใหม่อีกครั้ง
	เพื่อสร้างแฟ้มข้อมูลของโปรเจค
×.	[Target options] การปรับแต่งและกำหนดเบอร์ไมโครคอนโทรลเลอร์สำหรับการ
	โปรแกรม
	[File extensions, environments, and books] การปรับแต่งค่าแฟ้มข้อมูล การ
	กำหนดแฟ้มข้อมูลชนิดต่างๆ ฯลฯ

ที่มา: ARMKIEL Microcontrollers Tools, 2014

 แถบเครื่องมือสำหรับรูปแบบการพัฒนาโปรแกรม ประกอบด้วยเครื่องมือต่างๆ ที่สำคัญแสดงดังตารางที่ 3.2

				~ ~	
ตารางที่ 3.2	แถบเครือ	เงมือที่ไข่	ช้สำหรับเ	าารเขียนโบ	โรแกรม

ไอคอน	รายละเอียดของคำสั่ง
	[Translate] แปลโปรแกรมหรือแปลภาษาแอสเซมบลีแฟ้มข้อมูลที่กำลังตรวจแก้ใน
9	ปัจจุบัน
	[Build] แปลโปรแกรมและเชื่อมโยงโปรแกรมเพื่อสร้างแฟ้มข้อมูลของโปรเจค
	[Rebuild] แปลโปรแกรม แปลภาษาแอสเซมบลี และเชื่อมโยงโปรแกรมใหม่อีกครั้ง
	เพื่อสร้างแฟ้มข้อมูลของโปรเจค
S.	[Target options] การปรับแต่งและกำหนดเบอร์ไมโครคอนโทรลเลอร์สำหรับการ
a*a *	โปรแกรม
	[File extensions, environments, and books] การปรับแต่งค่าแฟ้มข้อมูล การ
	กำหนดแฟ้มข้อมูลชนิดต่างๆ ฯลฯ

ที่มา: ARMKIEL Microcontrollers Tools, 2014

 แถบเครื่องมือสำหรับการตรวจแก้จุดบกพร่องของโปรแกรม ประกอบด้วยเครื่องมือต่างๆ ที่สำคัญ แสดงดังตารางที่ 3.3

ตารางที่ 3.3 แถบเครื่องมือที่ใช้สำหรับการตรวจแก้จุดบกพร่องของโปรแกรม

ไอคอน	รายละเอียดของคำสั่ง
RST	[Reset CPU] ตั้งการทำงานของซีพียูใหม่
	[Run] การดำเนินงานโปรแกรมตั้งแต่บรรทัดปัจจุบัน
8	[Stop] หยุดการทำงานของซีพียูชั่วคราว
{ }	[Step one line] การดำเนินงานทีละคำสั่ง
0	[Step over] การดำเนินงานทีละคำสั่ง โดยจะมองโปรแกรมย่อยที่เรียกใช้เป็นหนึ่ง คำสั่ง
የት	[Step out] การดำเนินงานและกระโดดออกจากโปรแกรมย่อยปัจจุบัน โดยประมวลผล ทุกคำสั่งในโปรแกรมย่อย
*{}	[Run to cursor line] การดำเนินงานไปจนถึงตำแหน่งที่ตัวชี้ตำแหน่งชี้อยู่
	[Command window] เปิด/ปิดหน้าต่างคำสั่งโปรแกรม
Ø.	[Disassembly window] เปิด/ปิด หน้าต่างดิสแอสเซมเบลอร์ ซึ่งเป็นโปรแกรมแปล ภาษาแอสเซมบลีที่ถูกแปลมาจากภาษาซีที่เขียนขึ้น
G	[Symbol window] เปิด/ปิดหน้าต่างแสดงสัญลักษณ์ต่างๆ ที่ใช้ในโปรแกรม
	[Register window] เปิด/ปิดหน้าต่างแสดงค่าเรจิสเตอร์
¢2	[Call stack window] เปิด/ปิดหน้าต่างสำหรับดูค่าต่างๆ ของแสตก (Stack)
-	[Watch windows] เปิด/ปิดหน้าต่างที่ใช้สำหรับดูค่าต่างๆ ของตัวแปรที่ใช้ใน โปรแกรม

ตารางที่ 3.3 แถบเครื่องมือที่ใช้สำหรับการตรวจแก้จุดบกพร่องของโปรแกรม (ต่อ)

ไอคอน	รายละเอียดของคำสั่ง
	[Memory window] เปิด/ปิด หน้าต่างหน่วยความจำ ซึ่งเป็นหน้าต่างที่แสดงค่าข้อมูล
	ในหน่วยความจำตำแหน่งต่างๆ
	[Serial windows] เปิด/ปิด หน้าต่างพอร์ตอนุกรม ซึ่งเป็นหน้าต่างสำหรับแสดงข้อมูล
	ที่ส่งมาทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์
-	[Analysis windows] เปิด/ปิด หน้าต่างสำหรับวิเคราะห์การทำงานต่างๆ ของ
2222	โปรแกรม ประกอบด้วย
	1. Logic analyzer ทำหน้าที่ที่วิเคราะห์เปอร์เซ็นต์ของการทำงานในแต่ละฟังก์ชัน
	2. Performance analyzer ทำหน้าที่วิเคราะห์ประสิทธิภาพการทำงานของโปรแกรม
	3. Code coverage window ทำหน้าที่แสดงเปอร์เซ็นต์การประมวลผลในส่วนต่าง ๆ
	ของโปรแกรม

ที่มา: ARMKIEL Microcontrollers Tools, 2014

3.4 การพัฒนาโปรแกรม Keil μ Vision

การพัฒนาโปรแกรมของ Keil µVisionโดยทั่วไปต้องพัฒนาภายใต้โปรเจค ดังนั้นในการพัฒนา โปรแกรมจะต้องสร้างโปรเจคขึ้นมาก่อนแล้วนำแฟ้มข้อมูลภาษาซีขึ้นที่เขียนขึ้นเพิ่มเข้ามาในโปรเจค ก่อนที่จะทำการแปลโปรแกรมและตรวจสอบการทำงานของโปรแกรมต่อไป ขั้นตอนในการเขียน โปรแกรม Keil µVisionมีดังนี้

1. เปิดโปรแกรม Keil μ Vision โดยคลิกที่ Start -> All Program -> Keil μ Vision 4 จะ ปรากฏโปรแกรมแสดงดังภาพที่ 3.5

Wision4	
File Edit View Project Flash Debug Peripheralis Tools SVCS Window Help	
664A.	
Functions a 🔛	
Build Output	a 🛛
	*
C	
	CAP NUM SCR., OVR R/M

ภาพที่ 3.5 โปรแกรม Keil μ Vision

2. เริ่มสร้างโปรเจคใหม่โดยการคลิกที่ Project -> New μ Vision Project จะปรากฏหน้าต่าง ดังภาพ 3.6

🛂 ตVision4			_ C _X
File Edit View	Pro	ject Flash Debug Peripherals Tools SVCS Window Help	
1000 - 000 00 - 00 - 00 - 00 - 00 - 00 -		New #Vision Project New Multi-Project Workspace	
Functions		Open Project	
		Export Final Finae	
		Select Device for Target Remove Item	
	15	Options Alt+F7	
	(E) (E)	Clean target Build target F7 Rebuild all ranget files	
		Batch Build Translate	
E P (38 {)	1 C/Usen/Vostro/Desktop/Source code-MCS51/Chapter 3/Test.uvproj 2 O/ASSIST-PROP/umarnalsenauminasul fuilesTweamark 1 (Source-code/Chapter 3/Test.uvproj	
Build Output		3 C:\Keil_C51_9_51\C51\Examples\BLINKY\BLINKY.uvproj	Q
		4 C:\Keil_C51_9_51\C51\Examples\HELLO\Hello.uvproj	
		5 C:\Users\Vostro\Desktop\test MCU\IEST.uvproj	
		7 D\SOURCE-CODE\ARM-7(TC\TTSA\3.1 TTC-TTAS1 Add-Remove tasks to system-PONT\ARM1.uvproj	
		8 D:\SOURCE-CODE\ARM-7\TTC\MTIs\3.7.2 Add-Remove tasks to system- Change loop of Index in task interrupt COOL COOL\ARM1.uvproj	
		9 D:\SOURCE-CODE\ARM-7\BASIC PROGRAM\GPIO\1. Blinkly LED\Blinkly LED.uvproj	
		10 D:\SOURCE-CODE\ARM-7\LPC2378\LPC23xx_24xxSampleSoftware.r6\Keil\SPIO\fio.uvproj	

ภาพที่ 3.6 การสร้างโปรเจคใหม่โดยโปรแกรม Keil μ Vision

 ทำการเลือกตำแหน่งที่บันทึกโปรเจคและตั้งชื่อโปรเจคตามความต้องการ เมื่อเรียบร้อยให้ให้ คลิก Save ดังภาพที่ 3.7

Create New Project	2
COC - Local Disk (C) + Keil_C51_9_51 + C51 + Examples + BLINKY	- 4+ Search BLINKY
Organize - New folder	III • 📀
Dropbox Places BLINKY	
© Libraries S Documents Music S Pictures	
■ Videos III ■ Homegroup ■ Computer	
tocal Disk (C:)	
Iccal Disk (D:)	
File name: TEST Save as type: Project Files (*uvproj)	•
(in Hide Folders	Save Cancel

ภาพที่ 3.7 การบันทึกโปรเจคของโปรแกรม Keil μ Vision

4. เมื่อทำการบันทึกโปรเจคเรียบร้อยแล้ว จะได้หน้าต่าง Select Device for Target 'Target
 1'... เพื่อเลือกบริษัทผู้ผลิตไมโครคอนโทรลเลอร์ที่ต้องการใช้งาน ยกตัวอย่างจากภาพที่ 3.8 ทำการ
 เลือกไมโครคอนโทรลเลอร์ บริษัท Atmel

lect Device for Target 'Target 1'		
Vendor: Atmel		
Device:		
Toolset:		
Data base contents:	Description:	
Acer Labs Acer Labs Actel Actel Actel Actel Action Construction Analog Devices Analog Devices Analog Devices Analog Devices Analog Devices Analog Devices Active Construction Active Construction California Eastern Laboratorie CAST, Inc. CML Microcircuits CORERIVER Cubernatic Micro Systems Cubernatic Micro Systems Cubernatic Micro Systems Active Construction Active	4	
	OK Cancel	Help

ภาพที่ 3.8 ขั้นตอนการเลือกบริษัทผู้ผลิตไมโครคอนโทรลเลอร์ที่ต้องการใช้งาน

5. ทำการเลือกเบอร์ไมโครคอนโทรลเลอร์ที่ต้องการใช้งาน จากภาพที่ 3.9 ทำการเลือก ไมโครคอนโทรลเลอร์เบอร์ AT89C52 หลังจากนั้นคลิก OK

PU				
Vendor:	Atmel			
Device:	AT89C51		Use Extended Linker (LX51) instead of BL51	
Toolset	C51		Use Extended Assembler (AX51) instead of A51	
Data base	contents:		Description:	
0	AT89C2051 AT89C4051 AT89C5115 AT89C5115 AT89C5130 AT89C5130A AT89C5131A AT89C5131A AT89C5131A AT89C5132 AT89C51AC3 AT89C51C03 AT89C51C02 AT89C51ID2	•	8051-based Fullly Static 24MHz CMOS controller with 32 I/O Lines, 2 Timers/Counters, 6 Interrupts/2 Priority Levels, UART, Three-Level Program Memory Lock, 4K Bytes Flash Memory, 128 Bytes On-chip RAM	

ภาพที่ 3.9 การเลือกเบอร์ไมโครคอนโทรลเลอร์ที่ต้องการใช้งาน

6. ในขั้นตอนนี้โปรแกรมจะถามว่าต้องการเพิ่มแฟ้มข้อมูล Startup เข้าไปยังโปรเจคหรือไม่ แสดงดังภาพที่ 3.10 โดยให้เลือก No ไปก่อนเนื่องจากเรายังไม่ต้องการใช้แฟ้มข้อมูลนี้ เพราะแฟ้มข้อมูล ตัวนี้เป็นแฟ้มข้อมูลที่จัดการเกี่ยวกับฮาร์ดแวร์ที่เป็นลักษณะอื่น เช่น มีการใช้รอมหรือแรมภายนอก เป็น ต้น



ภาพที่ 3.10 การเลือกไฟล์ Startup เข้าไปยังโปรเจค

 7. ทำการเพิ่มแฟ้มข้อมูลเข้าไปยังโปรเจค โดยแฟ้มข้อมูลที่เพิ่มต้องเป็นแฟ้มข้อมูลนามสกุล .c สามารถทำได้โดยคลิกเลือกที่ File -> New... ดังภาพที่ 3.11 ซึ่งจะเห็นพื้นที่สำหรับใช้ในการเขียน โปรแกรมต้นฉบับ

🛯 Т	est - জ	vision4													-	-				
File	Edit	View Pro	oject	Flash	Debug	Perip	herals	Tools	SVCS	Windo	w ł	Help								
	New				Ctrl+N	1 7	n 19.	四限		- //= // ₁		Task_tir	ne	-	a de	0	0 0	0	-	6
2	Open				Ctrl+O	-	N.	48											 	
	Close						100													
	Save				Ctrl+S	12														
	Save As	i				12														
0	Save Al	Ľ.																		
	Device	Database.																		
	License	Managen	nent			12														
	Print Se	tup																		
9	Print				Ctrl+P	12														
	Print Pr	eview																		
	1 BLINK	(Y.m51																		
	2 C:\Ke	il_C51_9_5	51\\	BLINKY	r															
	3 C:\Kei	il_C51_9_5	51\\	ABSTRA	ACT	12														
	4 C:\Ke	il_C51_9_5	51\\	HELLO		12														
	5 C:\Us	ers\\test	MCL	J\test		12														
	6 C:\Us	ers\\STA	RTUF	P.A51		12														
	7 D:\SC	URCE-CC	DE\	.\Task_c	cal	12														
	8 D:\SC	URCE-CC	DE\	.\systen	n_init	12														
	9 D:\SC	URCE-CC	DE\	.\sch_co	оор															
	10 D:\S	OURCE-C	ODE	\\Delay	y_T0															
	Exit																		 	

ภาพที่ 3.11 การสร้างแฟ้มข้อมูลนามสกุล .c

8. ทำการบันทึกเป็นแฟ้มข้อมูลใหม่นี้เป็นแฟ้มข้อมูลนามสกุล .c ดังภาพที่ 3.12 โดยคลิกที่ File -> Save As....

۲ 🕙	est - @Vision4		Conception and the local division of the loc
File	Edit View Project Flash	Debug	Peripherals Tools SVCS Window Help
	New Open Close	Ctrl+N Ctrl+O	作 熱 熱 秋 洋 洋 /// /// 図 Task_time ▼ 論 4* Q ● ○ 今 歳 回▼ ≪ ▼ ぶ 品 号
	Save	Ctrl+S	1
	Save As		
	Device Database License Management		
	Print Setup		
8	Print Print Preview	Ctrl+P	
-	1 BLINKY.m51 2 C:\Keil_C51_9_51\\BLINK') 3 C:\Keil_C51_9_51\\BBSTR, 4 C:\Keil_C51_9_51\\HELLO 5 C:\Users\\STARTUP.A51 7 D:\SOURCE-CODE\\Stast_ 8 D:\SOURCE-CODE\\Seta_0 10 D:\SOURCE-CODE\\Dela	cal m_init oop y_T0	
	Exit		

ภาพที่ 3.12 การบันทึกแฟ้มข้อมูลนามสกุล .c

9. ทำการเลือกตำแหน่งที่บันทึกแฟ้มข้อมูลและตั้งชื่อแฟ้มข้อมูลโดยมีนามสกุล.c เมื่อเรียบร้อย ให้คลิก Save เพื่อทำการบันทึกแฟ้มข้อมูล ดังภาพที่ 3.13

🗹 Save As		X
Source	code-MCS51 + Chapter 3	- 4 Search Chapter 3
Organize • New fol	der	III • 🔞
 Favorites Desktop Downloads Dropbox Recent Places Libraries Documents Music Pictures Videos 	 Test.uvgui.Vostro Test.uvpti Test Test_uvproj.bak 	
Computer	*	
File name: Tes	tc	
Save as type: All f	illes (**)	•
Hide Folders		Save

ภาพที่ 3.13 การเลือกตำแหน่งในการบันทึกแฟ้มข้อมูลนามสกุล .c

10. ทำการเพิ่มแฟ้มข้อมูลเข้ามายังโปรเจค ดังภาพที่ 3.14 โดยการคลิกขวาที่ Source Group 1 (ที่หน้าต่าง Project Workspace) แล้วคลิกเลือกที่ Add Files to Group 'Source Group1'...



ภาพที่ 3.14 การเพิ่มแฟ้มข้อมูลต้นฉบับเข้ามายังโปรเจค

11. ทำการเลือกตำแหน่งของแฟ้มข้อมูลที่บันทึกไว้ก่อนหน้านี้ จากนั้นคลิกเลือกที่แฟ้มข้อมูลที่ ต้องการเพิ่มเข้ามายังโปรเจคและคลิก Add และ Close ตามลำดับ ดังภาพที่ 3.15 ซึ่งสามารถสังเกตได้ จากหน้าต่างโปรเจคจะมีแฟ้มข้อมูลดังกล่าวเพิ่มเข้ามา



ภาพที่ 3.15 การเลือกตำแหน่งของแฟ้มข้อมูลเพื่อเพิ่มแฟ้มข้อมูลเข้ามายังโปรเจค

12. เมื่อต้องการเรียกแฟ้มข้อมูลนามสกุล .c มาทำการแก้ไข ให้ดับเบิลคลิกที่ไอคอนของ แฟ้มข้อมูล (Test.c) ในหน้าต่างโปรเจค ดังภาพที่ 3.16 เพื่อทำการเขียนโปรแกรมหรือแก้ไขโปรแกรม

Project	P 🔝
⊡ 🔁 Target 1 ⊡ 🔁 Source ⊕ 🔛 Te	ce Group 1 est.c
<mark>је Р</mark> 🍪 В 1	() F (), T

ภาพที่ 3.16 การแก้ไขแฟ้มข้อมูลนามสกุล .c

13. ทดลองเขียนโปรแกรมเพื่อทดสอบการแปลโปรแกรม แสดงดังภาพที่ 3.17

Test.c	
1 #includ	le <regx51.h></regx51.h>
2	
3 main	()
4 🖓 {	
5 unsig	ned char i;
6 while	ė (1)
7 白 {	
8	<pre>for (i=0x01; i<0x80; i<<=1)</pre>
9 白	{
10	P1 = i;
11 -	}
12 }	1
13 13	

ภาพที่ 3.17 การเขียนโปรแกรมเพื่อทดสอบการแปลโปรแกรม

 14. ทำการบันทึกแฟ้มข้อมูลที่เขียนขึ้น หลังจากนั้นให้คลิกที่แถบเครื่องมือ Build Target เพื่อ ทำการแปลโปรแกรม สังเกตที่หน้าต่างนำออกจะรายงานผลการแปลโปรแกรมออกมาแสดงดังภาพที่
 3.18 ซึ่งหากไม่มีข้อผิดพลาดเกิดขึ้น แสดงว่าโปรแกรมที่เขียนขึ้นถูกต้องต้องตามรูปแบบของข้อกำหนด ต่าง ๆ ของการเขียนโปรแกรม เราสามารถตรวจสอบผลการทำงานของโปรแกรมได้โดยการใช้ ความสามารถของ Keil µVision ในรูปแบบการตรวจแก้จุดบกพร่องของโปรแกรมซึ่งจะกล่าวถึงใน หัวข้อต่อไป

```
Build Output

Rebuild target 'Target 1'

compiling Test.c...

linking...

Program Size: data=9.0 xdata=0 code=75

"Test" - 0 Error(s), 0 Warning(s).
```

ภาพที่ 3.18 หน้าต่างนำออกแสดงการแปลโปรแกรมจากแฟ้มข้อมูลที่เขียนขึ้น

3.5 การตรวจแก้จุดบกพร่องของโปรแกรม Keil µVision

หลังจากทำการพัฒนาโปรแกรมและทำการแปลโปรแกรมเรียบร้อยแล้ว ให้คลิกที่แถบเครื่องมือ Start/Stop Debug Session เพื่อทำการสลับรูปแบบจากการเขียนโปรแกรมไปสู่รูปแบบการตรวจแก้ จุดบกพร่องของโปรแกรม โดยมีขั้นตอนการทำงานดังนี้

 เมื่อทำการสลับรูปแบบจากการเขียนโปรแกรมไปสู่รูปแบบการตรวจแก้จุดบกพร่องของ โปรแกรม Keil µVision จะแสดงข้อความระบุให้ทราบว่า โปรแกรมที่ใช้งานเป็นรุ่นทดลอง สามารถ รองรับขนาดของรหัสคำสั่งหรือแฟ้มข้อมูลนามสกุล .hex ได้ไม่เกิน 2 กิโลไบต์ แสดงดังภาพที่ 3.19 จากนั้นให้คลิก OK เพื่อเข้าสู่รูปแบบการตรวจแก้จุดบกพร่องของโปรแกรม



ภาพที่ 3.19 Keil μ Vision แสดงข้อความโปรแกรมที่ใช้งานเป็นรุ่นทดลอง

 2. ทดลองดำเนินงานโปรแกรมที่เขียนขึ้น โดยการคลิกที่แถบเครื่องมือ Step one line เพื่อ ดำเนินงานโปรแกรมทีละคำสั่ง โดยในการดำเนินงานจะสังเกตเห็นตัวชี้ตำแหน่งเคลื่อนที่ไปที่ละหนึ่ง บรรทัด แสดงดังภาพที่ 3.20

Test.c	
1 #include	e <regx51.h></regx51.h>
2	
3 main	()
4 🕀 {	
5 unsign	ned char i;
6 while	(1)
7 🗛 👔	
8	<pre>for (i=0x01; i<0x80; i<<=1)</pre>
9 🛱	{
10	P1 = i;
11	}
12 }	
13 ^L }	

ภาพที่ 3.20 การดำเนินงานโปรแกรมทีละคำสั่ง

 ทดสอบค่าตัวแปรที่ใช้ในโปรแกรมจากหน้าต่าง Call Stack จะได้หน้าต่างดังภาพที่ 3.21 ซึ่ง จะแสดงผลลัพธ์ของตัวแปร i ออกมา

Name	9	Location/Value	Туре
•	MAIN	C:0x0800	
	🤗 i	0x08	uchar

ภาพที่ 3.21 การตรวจสอบค่าตัวแปรที่ใช้ในโปรแกรมจากหน้าต่าง Call Stack

4. ทดสอบการทำงานของไมโครคอนโทรลเลอร์ สามารถทำได้โดยการคลิกที่รายการเลือก
 Peripherals หลังจากนั้นให้คลิกที่ Port 1 ซึ่งจะแสดงหน้าต่างการทำงานของ Port 1 ดังภาพที่ 3.22

Interrupt	Wind	ow Help		Parallel Port 1
I/O-Ports	•	Port 0		Port 1 7 Bits 0
Serial	V	Port 1 💻	\rightarrow	
Timer	•	Port 2	P	Pins: 0x08
8051 statemachine		Port 3		

ภาพที่ 3.22 การตรวจสอบการทำงานของไมโครคอนโทรลเลอร์

จะเห็นได้ว่าผู้เขียนโปรแกรมสามารถใช้งานความสามารถของ Keil μ Vision ในรูปแบบการตรวจ แก้จุดบกพร่องของโปรแกรมเพื่อดูผลการทำงานต่างๆ ของแต่ละคำสั่ง ซึ่งรูปแบบการตรวจแก้ จุดบกพร่องนี้จะช่วยเพิ่มความสะดวกในการแก้ไขความผิดพลาดต่างๆ ของโปรแกรมได้อย่างรวดเร็ว

5. เมื่อทำการตรวจแก้จุดบกพร่องของโปรแกรมเรียบร้อยแล้ว ขั้นตอนต่อไปคือการนำ โปรแกรมที่พัฒนาขึ้นไปอัพโหลดโปรแกรมลงบนไมโครคอนโทรลเลอร์เพื่อนำไปทดสอบการทำงานกับ ฮาร์ดแวร์ต่อไป โดยการอัพโหลดโปรแกรมลงบนไมโครคอนโทรลเลอร์ต้องนำแฟ้มข้อมูลนามสกุล .hex ไปโปรแกรม ซึ่งมีขั้นตอนในการสร้างแฟ้มข้อมูลนามสกุล .hex ดังนี้ 5.1 ให้คลิกขวาที่ Target 1 จากนั้นให้เลือกที่ Option for Target 'Target 1' ดังภาพที่



ภาพที่ 3.23 การสร้างแฟ้มข้อมูลนามสกุล .hex

3.23

5.2 ที่หน้าต่าง Option for Target 'Target 1 ' ให้เลือกที่ตัวเลือก Output และคลิกที่ Create HEX File ดังภาพที่ 3.24

vice Target Output Listing User	C51 A51 BL51 Locate BL51 Misc Debug Utilities	
Select Folder for Objects	Name of Executable: Test	
Create Executable: .\Test		
 Debug Information 	Browse Information	
Create HEX File HEX Fo	ormat HEX-80	
C Create Library: .\Test.LIB		Create Batch File
C Create Library: .\TestLIB		Create Batch File
C Create Library: .\TestLIB		Create Batch File
C Create Library: .\TestLIB		Create Batch File
C Create Library: .\TestLIB		Create Batch File
C Create Library: .\TestLIB		Create Batch File
C Create Library: .\TestLIB		Create Batch File

ภาพที่ 3.24 การสร้างแฟ้มข้อมูลนามสกุล .hex (ต่อ)

51

3.6 สรุป

เนื้อหาในบทนี้ได้กล่าวถึงการเขียนโปรแกรมภาษาซีสำหรับไมโครคอนโทรลเลอร์ MCS-51 โดยได้ อธิบายพื้นฐานภาษาซีสำหรับไมโครคอนโทรลเลอร์ MCS-51 และได้แนะนำโปรแกรม Keil μ Vision สำหรับการพัฒนางานไมโครคอนโทรลเลอร์ ซึ่งการโปรแกรมนี้รองรับการทำงานไมโครคอนโทรลเลอร์ MCS-51 ได้อย่างหลากหลายและสามารถเลือกใช้งานไมโครคอนโทรลเลอร์แต่ละบริษัทผู้ผลิตได้ตาม ต้องการ นอกจากนี้โปรแกรม Keil μ Vision ยังสามารถจำลองการทำงานเพื่อการตรวจแก้จุดบกพร่อง ของโปรแกรมได้อย่างมีประสิทธิภาพ

3.7 แบบฝึกหัดท้ายบท

แบบฝึกหัดมีทั้งหมด 3 ข้อ ให้นักศึกษาทำแบบฝึกหัดทุกข้อ

- 1. จงอธิบายขั้นตอนในการเขียนโปรแกรมโดย Keil μ Vision
- 2. จงอธิบายขั้นตอนในการตรวจแก้จุดบกพร่องของโปรแกรมโดย Keil μ Vision
- 3. จงเขียนโปรแกรมและหาผลลัพธ์ของการรันโปรแกรม ดังนี้

```
#include <REGX51.H>
 1
 2 main (void)
 3 ⊟ {
 4 unsigned int i;
 5 unsigned char j;
 6 while (1)
 7白
     {
        for (j=0x01; j< 0x80; j<<=1)
8
9白
10
        P1 = j;
11
          for (i = 0; i < 10000; i++)
12 白
          {
13
          }
14
        }
15
16
      for (j=0x80; j> 0x01; j>>=1)
17 白
        {
18
        P1 = j;
19
          for (i = 0; i < 10000; i++)
20 白
          {
21
          }
22
        }
23
      }
24
   }
```