



การปรับปรุงรีโมทสำหรับสั่งงานโทรทัศน์และสวิตช์คาบิเน็ตด้วยเซ็นเซอร์
แบบคาปาซิทีฟ

กรณีศึกษา บริษัทพานาโซนิค เอวีซีเน็ตเวิร์คส์ (ประเทศไทย) จำกัด

IMPROVEMENT OF REMOTE CONTROL FOR TELEVISION
AND CABINET SWITCH USING CAPACITIVE TOUCH SENSING
TECHNOLOGY

CASE STUDY: PANASONIC AVC NETWORKS (THAILAND) CO.,
LTD.

นายธราเทพ เปรมประสาธสิทธิ์

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีไทย – ญี่ปุ่น

พ.ศ. 2554

การปรับปรุงรีโมทสำหรับสั่งงานโทรทัศน์และสวิตช์คาบิเน็ตด้วยเซ็นเซอร์แบบคาปาซิทีฟ
กรณีศึกษา บริษัทพานาโซนิค เอวีซีเน็ตเวิร์คส์(ประเทศไทย) จำกัด

IMPROVEMENT
OF REMOTE CONTROL FOR TELEVISION AND CABINET SWITCH USING CAPACITIVE
TOUCH SENSING TECHNOLOGY
CASE STUDY: PANASONIC AVC NETWORKS (THAILAND) CO., LTD.

นายธราเทพ เปรมประสาธสิทธิ์

โครงการสหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีไทย – ญี่ปุ่น

พ.ศ. 2554

คณะกรรมการสอบ

.....ประธานกรรมการสอบ

(อาจารย์ ดร.วิมล แสนอ้อม)

.....กรรมการสอบและอาจารย์ที่ปรึกษา

(อาจารย์ปรีวัตร คงกำเนิด)

.....กรรมการ

(อาจารย์ ดร.วรากรศรีเชวงทรัพย์)

ลิขสิทธิ์ของสถาบันเทคโนโลยีไทย-ญี่ปุ่น

บทสรุป

ชื่อโครงการ	การปรับปรุงรีโมทสำหรับสั่งงานโทรทัศน์ และสวิตช์คาบิเน็ตด้วยเซ็นเซอร์แบบคาปาซิทีฟ IMPROVEMENT OF REMOTE CONTROL FOR TELEVISION AND CABINET SWITCH USING CAPACITIVE TOUCH SENSING TECHNOLOGY
ผู้เขียน	นายธราเทพ เปรมประสาทสิทธิ์
คณะวิชา	วิศวกรรมศาสตร์ สาขาวิชา วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ปรีวัตร คงกำเนิด
พนักงานที่ปรึกษา	นายชาติรี เมตตาคิจตกุล
ชื่อบริษัท	พานาโซนิค เอวีซีเน็ดเวิร์ค (ประเทศไทย) จำกัด
ประเภทธุรกิจ/สินค้า	โรงงานผลิตโทรทัศน์สี

งานที่ปฏิบัติ

1. ออกแบบรูปร่างลักษณะทั่วไปของรีโมทและวางแนวทางของผลิตภัณฑ์รีโมทชนิดใหม่และสวิตช์บนตัวทีวี
2. ศึกษาเรื่องสิทธิบัตรที่เกี่ยวข้องกับผลิตภัณฑ์ที่กำลังจะทำขึ้นมาใหม่เพื่อไม่ให้สิ่งที่ทำขึ้นมาใหม่มีความซ้ำซ้อนกับสิ่งที่เคยมีคนทำขึ้นมาแล้ว
3. ศึกษาเกี่ยวกับโปรโตคอลที่ใช้ในการสั่งการรีโมท
4. ลงมือทำผลิตภัณฑ์ขึ้นมาใหม่ตั้งแต่การออกแบบ PCB ของผลิตภัณฑ์ การกัดแผ่น PCB เพื่อทำเป็นลายวงจรของผลิตภัณฑ์ ตลอดจนเขียนโปรแกรมลงในไมโครคอนโทรลเลอร์ด้วยภาษาซีเพื่อไปสั่งการให้ infrared LED ส่งสัญญาณไปสู่ทีวีต่อไป (สำหรับรีโมท) และแสดงผลบน Computer

ผลที่ได้รับจากการดำเนินงานและประโยชน์ที่ได้รับ

ได้ผลิตภัณฑ์ที่ทำให้ได้ความรู้สึกในการใช้งานที่แตกต่างไปจากเดิมและสร้างประสบการณ์ในการใช้งานใหม่ๆ ให้กับผู้ใช้งาน

กิตติกรรมประกาศ

ในการปฏิบัติงานสหกิจศึกษาที่บริษัท พานาโซนิค เอวีซีเน็ดเวิร์คส์ (ประเทศไทย) จำกัด เป็นระยะเวลาทั้งสิ้น 4 เดือนนั้น ได้รับความกรุณาและการดูแลจากหลาย ๆ ท่าน ทำให้การปฏิบัติงานครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี และเป็นประสบการณ์ที่ดี ที่ได้เรียนรู้หลาย ๆ เรื่องที่ไม่เคยรู้

ขอขอบคุณ คุณชาติรี เมตตาจิตสกุล พนักงานที่ปรึกษา ซึ่งให้การดูแลและให้คำปรึกษาต่าง ๆ ในการทำงาน รวมถึงตอบข้อสงสัยต่าง ๆ อย่างกระจ่าง พร้อมทั้งแนะนำเทคนิคในการทำงานซึ่งสามารถนำไปใช้ได้จริง รวมไปถึงพนักงานท่านอื่น ๆ ที่ดูแลและตอบข้อสงสัย รวมถึงเป็นที่ปรึกษาและคอยรับฟังให้ข้อมูลในด้านต่าง ๆ ในการทำงาน ทำให้ตลอด 4 เดือนที่ผ่านมาเป็นความทรงจำที่ประทับใจ ขอขอบพระคุณทุกท่านไว้ ณ โอกาสนี้ด้วย

ธราเทพ เปรมประสาธสิทธิ์



สารบัญ

หน้า

บทสรุป	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
รายการตาราง	ช
รายการรูปประกอบ	ณ

บทที่

1. บทนำ	1
1.1 ชื่อและที่ตั้งของสถานประกอบการ	1
1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร	2
1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร	2
1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย	3
1.5 พนักงานที่ปรึกษาและตำแหน่งของพนักงานที่ปรึกษา	4
1.6 ระยะเวลาที่ปฏิบัติงาน	4
1.7 วัตถุประสงค์หรือจุดมุ่งหมายของการปฏิบัติงานหรือโครงการที่ได้รับมอบหมายให้ปฏิบัติงานสหกิจศึกษา	4
1.8 ผลที่คาดว่าจะได้รับการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย	4
2. ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน	5
2.1 เทคโนโลยี Capacitive Touch Sensing	5
2.1.1 Capacitive Sensing Module (CSM)	5
2.1.2 Capacitive Voltage Divider (CVD)	6
2.1.3 Charge Time Measurement Unit (CTMU)	8
2.2 โปรโตคอลที่ใช้ในการสื่อสารระหว่างรีโมตกับทีวี	
2.2.1 家製協フォーマット (Kaseikyou format)	9

สารบัญ(ต่อ)

บทที่

หน้า

2. ทฤษฎีและเทคโนโลยีที่ใช้ในการปฏิบัติงาน(ต่อ)

2.3 ส่วนช่วยในการพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์

2.3.1 ชุดพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์

2.3.1.1 MPLAB IDE

10

2.3.1.2 ARDUINO IDE

23

2.3.2 คอมพิวเตอร์ที่ใช้ในการพัฒนาผลิตภัณฑ์

28

2.3.2.1 HI-TECH C

28

2.3.3 เครื่องมือที่ใช้ในการเขียนข้อมูลลงไปในไมโครคอนโทรลเลอร์

33

2.3.3.1 PICKIT 3

33

2.4 สายเชื่อมต่อเพื่อสื่อสารระหว่างไมโครคอนโทรลเลอร์และคอมพิวเตอร์

35

2.4.3 RS-232 Recommended Standard-232

35

2.4.2 USB (Universal Serial Bus)

36

2.5 ไมโครคอนโทรลเลอร์และอุปกรณ์เสริมที่ใช้ในการพัฒนาผลิตภัณฑ์

36

2.5.1 ไมโครคอนโทรลเลอร์ที่ใช้ในการพัฒนาผลิตภัณฑ์

36

2.5.1.1 PIC16f1826

36

2.5.1.2 ARDUINO ATMEGA 328

37

2.5.2 อุปกรณ์เสริมที่ช่วยในการพัฒนาผลิตภัณฑ์

37

2.5.2.1 Capacitive touch shield

37

2.5 ภาษาที่ใช้ในการเขียนโปรแกรมเพื่อพัฒนาผลิตภัณฑ์

38

2.5.1 C language Programming

38

3. แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

61

3.1 แผนการปฏิบัติงาน

61

3.2 รายละเอียดโครงการที่ได้รับมอบหมาย

61

3.3 ขั้นตอนการดำเนินงานโครงการ

62

3.3.1 วางแนวทางผลิตภัณฑ์ที่ต้องการ

62

3.3.2 ศึกษาเรื่องสิทธิบัตร

63

3.3.3 ศึกษาโปรโตคอลที่ใช้สื่อสารระหว่างรีโมตและทีวี

63

สารบัญ(ต่อ)

บทที่

3. แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน (ต่อ)	หน้า
3.3 ขั้นตอนการดำเนินงานโครงการ(ต่อ)	
3.3.4 ศึกษาและเข้าอบรมเกี่ยวกับหลักการทำงานของCapacitive Touch Sensing	63
3.3.5 จัดทำ Part List ที่ที่เป็นไปได้ที่จะใช้ในรีโมต	66
3.3.6 คิด interface บนทีวีที่ใช้ร่วมกันกับรีโมตและสวิตช์แบบใหม่	66
3.3.7 ทำผลิตภัณฑ์ต้นแบบขึ้นมาใหม่ทั้งหมด	66
3.3.8 เขียนโปรแกรมลงในไมโครคอนโทรลเลอร์ที่ใช้กับตัวผลิตภัณฑ์	66
3.3.9 ทดลองใช้งานและปรับปรุงแก้ไขเพิ่มเติม	71
4. สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ	73
4.1 สรุปการดำเนินงาน	73
4.2 การวิเคราะห์และสรุปผลต่าง ๆ	73
4.2.1 ผลจากการพัฒนารีโมต	73
4.2.2 ผลจากการพัฒนาสวิตช์บนส่วนของตัวทีวี	74
5. บทสรุปและข้อเสนอแนะ	75
5.1 บทสรุป	75
5.2 สรุปผลการปฏิบัติงาน	75
5.2.1 สรุปผลจากการศึกษาการวางแผนทางผลิตภัณฑ์ที่ต้องการ	76
5.2.2 สรุปผลจากการศึกษาเรื่องสิทธิบัตร	76
5.2.3 สรุปผลจากการลงมือสร้างผลิตภัณฑ์ใหม่	76
5.2.4 ทดลองใช้งานและปรับปรุงแก้ไขเพิ่มเติม	76
5.3 ปัญหาและอุปสรรค	77
5.4 ข้อเสนอแนะ	77

สารบัญ(ต่อ)

หน้า

เอกสารอ้างอิง

78

ภาคผนวก

81

ประวัติผู้วิจัย

90



รายการตาราง

ตาราง	หน้า
2.1 รายละเอียดของตัวแปรภาษาซีแต่ละชนิด	57
2.2 ตารางแสดง คำสงวน ที่ห้ามใช้ตั้งเป็นชื่อตัวแปร	59
2.3 ตารางแสดงตัวอย่างที่ถูกต้องและไม่ถูกต้องในการตั้งชื่อตัวแปรในภาษาซี	59
3.1 ตารางแสดงงาน วัน และรายละเอียด	61
3.2 ตารางเปรียบเทียบข้อดีของแต่ละระบบของ Capacitive Touch sensing	64
3.3 ตารางเปรียบเทียบข้อเสีย ของแต่ละระบบของ Capacitive Touch Sensing	65
3.4 ตารางเปรียบเทียบคุณสมบัติ ของแต่ละ Chip	65

รายการรูปประกอบ

รูป	หน้า
1.1	1
1.2	1
1.3	2
1.4	2
2.1	5
2.2	6
2.3	7
2.4	8
2.5	8
2.6	9
2.7	9
2.8	10
2.9	10
2.10	11
2.11	12
2.12	12
2.13	13
2.14	13
2.15	14
2.16	14
2.17	15
2.18	15
2.19	16
2.20	16
2.21	17

รายการรูปประกอบ(ต่อ)

รูป	หน้า
2.21 MPLAB IDE Project Wizard	17
2.22 MPLAB IDE Select Device	18
2.23 MPLAB IDE Select a language tool suite	18
2.24 MPLAB IDE Create a new Project	19
2.25 MPLAB IDE Save Project as	19
2.26 MPLAB IDE Add existing file to your project	20
2.27 MPLAB IDE Summary	20
2.28 MPLAB IDE New File	21
2.29 MPLAB IDE Blank space	21
2.30 MPLAB IDE File	22
2.31 MPLAB IDE Save As	22
2.32 MPLAB IDE Directory	22
2.33 ARDUINO IDE Download Link	23
2.34 Arduino-0015 Setup Wizard	23
2.35 ARDUINO IDE Select Destination Location	24
2.36 ARDUINO IDE Select Start Menu Folder	24
2.37 ARDUINO IDE Select Additional Task	25
2.38 ARDUINO IDE Ready to install	25
2.39 ARDUINO IDE Installing	26
2.40 ARDUINO IDE Completing the arduino-0015 Setup Wizard	26
2.41 หน้าต่างโปรแกรม Arduino IDE	27
2.42 ARDUINO IDE Compile button	27
2.43 ARDUINO IDE Status	27
2.44 ARDUINO IDE Write button	28
2.45 ARDUINO IDE Serial Monitoring button	28
2.46 Microchip's Compiler list	29

รายการรูปประกอบ(ต่อ)

รูป	หน้า
2.47 HI-TECH C installer	29
2.48 HI-TECH C mode selection	30
2.49 HI-TECH License agreement	30
2.50 HI-TECH C Install destination selection	31
2.51 HI-TECH C language selection	31
2.52 HI-TECH C Installing	32
2.53 HI-TECH C Finish installation	33
2.54 PICkit3	33
2.55 PICkit3 Component	34
2.56 PICkit3 Pin Description	35
2.57 สายRS-232	35
2.58 สายUSB	36
2.59 PIC16f1516	37
2.60 ARDUINO ATMEGA 328	37
2.61 Capacitive touch shield	38
3.1 Multi-Touch Gesture Dictionary Patent	62
3.2 Multi-Touch Gesture Dictionary Patent What is claimed is	63
3.3 Flow Chart แสดงขั้นตอนการทำงานของโปรแกรมรีโมต1	67
3.4 Flow Chart แสดงขั้นตอนการทำงานของโปรแกรมรีโมต2	68
3.5 Flow Chart แสดงขั้นตอนการทำงานของโปรแกรมรีโมต3	69
3.6 Flow Chart แสดงขั้นตอนการทำงานของโปรแกรมรีโมต4	70
3.7 Overview Block Diagram	71
3.8 Capacitance Equation	71
3.9 Example pad size and shape	72

บทที่ 1

บทนำ

1.1 ชื่อและที่ตั้งของสถานประกอบการ

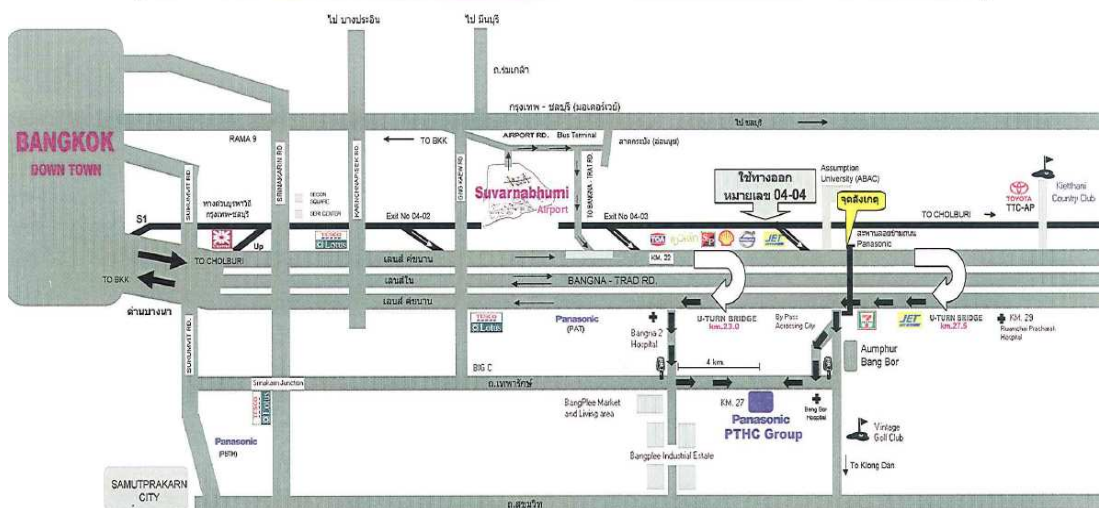
บริษัท พานาโซนิค เอวีซีเนตเวิร์คส์ (ประเทศไทย) จำกัด

Panasonic
ideas for life

รูปที่ 1.1 โลโก้บริษัทพานาโซนิค เอวีซีเนตเวิร์คส์ (ประเทศไทย) จำกัด

ที่ตั้ง : 101 หมู่ 2 ถนนเทพารักษ์ ตำบลบางเสาธง อำเภอบางเสาธง สมุทรปราการ

แผนที่ กลุ่มบริษัท พานาโซนิค PTHC



รูปที่ 1.2 แผนที่บริษัท พานาโซนิค เอวีซีเนตเวิร์คส์ (ประเทศไทย) จำกัด

1.2 ลักษณะธุรกิจของสถานประกอบการ หรือการให้บริการหลักขององค์กร

ในปัจจุบันทางบริษัทฯ ได้เป็นฐานการผลิตที่สำคัญในการผลิตเครื่องรับโทรทัศน์สี , จอมอนิเตอร์ , โมดูลสำหรับจอภาพแสดงผลแบบพลาสติกขนาด 42,46,50,55,60 และ 65 นิ้ว เพื่อจำหน่ายไปยังประเทศต่าง ๆ ทั่วโลกกว่า 30 ประเทศ

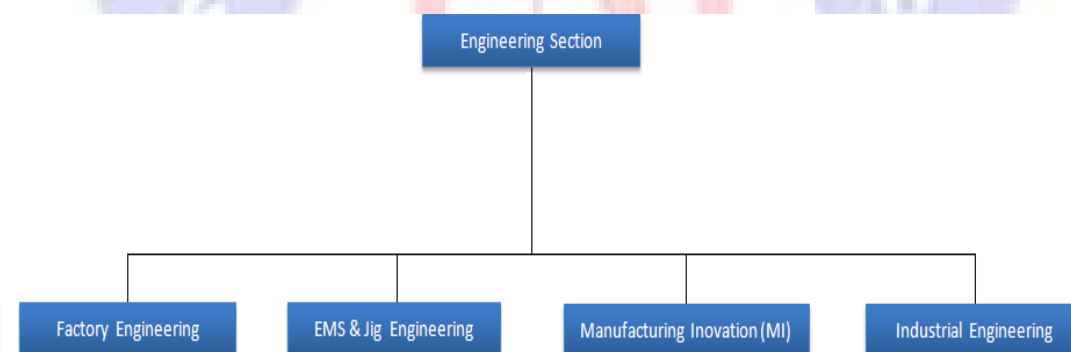
ตลอดระยะเวลากว่า 40 ปีในการผลิตเครื่องรับโทรทัศน์ในประเทศไทย บริษัทมุ่งมั่นสู่ความเป็นเลิศในด้านมาตรฐานระดับสูงผ่านการดำเนินธุรกิจของบริษัท บริษัทตั้งปณิธานอันแน่วแน่ในการรักษาสีเงาตลอด การนำเสนอเทคโนโลยีขั้นสูง, การพัฒนาผลิตภัณฑ์ และการดำเนินกิจกรรมเพื่อตอบแทนสู่สังคม

ปรัชญาพื้นฐานในการดำเนินธุรกิจของ PAVCTH คือ “ลูกค้าสำคัญมาเป็นอันดับหนึ่ง” ดังนั้นเราจึงพยายามที่จะสร้างสรรค์เทคโนโลยีอันทันสมัยสู่ตลาดและผลิตภัณฑ์ที่มีความน่าเชื่อถือ อันเป็นการสนองตอบความต้องการของลูกค้า

1.3 รูปแบบการจัดองค์กรและการบริหารองค์กร

แผนผังองค์กร

โครงสร้างการบริหารบริษัท พานาโซนิก เอวิซีเนตเวิร์คส์ (ประเทศไทย) จำกัด ฝ่ายวิศวกรรม

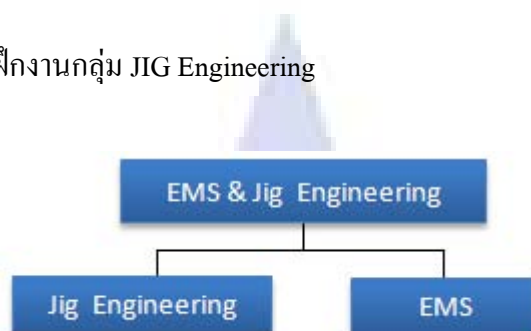


รูปที่ 1.3 โครงสร้างการบริหารบริษัท พานาโซนิก เอวิซีเนตเวิร์คส์ (ประเทศไทย) จำกัด

ฝ่ายวิศวกรรม

1.4 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่งงาน : นักศึกษาฝึกงานกลุ่ม JIG Engineering



รูปที่ 1.4 โครงสร้างการบริหารภายในแผนก EMS & Jig Engineering

หน้าที่ : การปฏิบัติงานจะเป็นการเขียนซอฟต์แวร์ต่าง ๆ เพื่อใช้ในสายการผลิต และพัฒนาผลิตภัณฑ์เกี่ยวกับรีโมตและสวิตช์บนทีวี

1.5 พนักงานที่ปรึกษาและตำแหน่งของพนักงานที่ปรึกษา

นายชาติ เมตตาจิตสกุล ตำแหน่ง ผู้จัดการแผนก Factory Engineering

1.6 ระยะเวลาที่ปฏิบัติงาน

1 มิถุนายน พ.ศ. 2554 สิ้นสุดงานสหกิจศึกษา 30 กันยายน พ.ศ. 2554

1.7 วัตถุประสงค์หรือจุดมุ่งหมายของการปฏิบัติงานหรือโครงการที่ได้รับมอบหมายให้ปฏิบัติงานสหกิจศึกษา

- 1) เพื่อเป็นการทำผลิตภัณฑ์ใหม่ที่ทำให้ความรู้สึกที่แตกต่างแก่ผู้ใช้งาน
- 2) เพื่อฝึกให้นักศึกษามีความรับผิดชอบต่องานที่ได้รับมอบหมาย

- 3) เพื่อนำความรู้ที่ได้จากชั่วโมงเรียนมาประยุกต์ใช้ในการปฏิบัติงานจริง
- 4) เพื่อฝึกฝนทักษะในการปฏิบัติงานร่วมกันกับผู้อื่น
- 5) เพื่อเพิ่มพูนทักษะเคล็ดลับ และการแก้ปัญหาในการปฏิบัติงานจริง

1.8 ผลที่คาดว่าจะได้รับจากการปฏิบัติงานหรือโครงการที่ได้รับมอบหมาย

- 1) ได้รับความรู้และประสบการณ์ในการปฏิบัติงานจริง
- 2) เรียนรู้เคล็ดลับและทักษะในการปฏิบัติงานในด้านต่าง ๆ เพิ่มขึ้น
- 3) ทดลองปฏิบัติงานเกี่ยวกับเรื่องที่ตนเองมีความสนใจหรือตั้งใจที่จะทำในอนาคต
- 4) นักศึกษาหรือผู้ที่สนใจสามารถนำข้อมูลเพื่อนำไปประกอบการทำรายงานหรืออ้างอิงได้



บทที่ 2

ทฤษฎีและเทคโนโลยีที่ใช้ปฏิบัติงาน

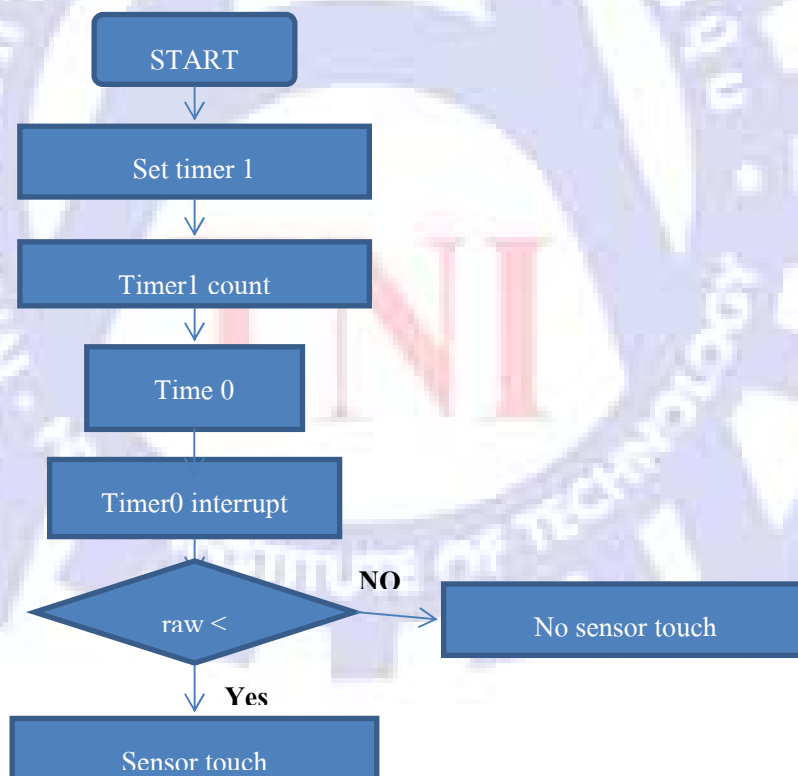
2.1 เทคโนโลยี capacitive touch sensing

2.1.1 Capacitive Sensing Module (CSM)

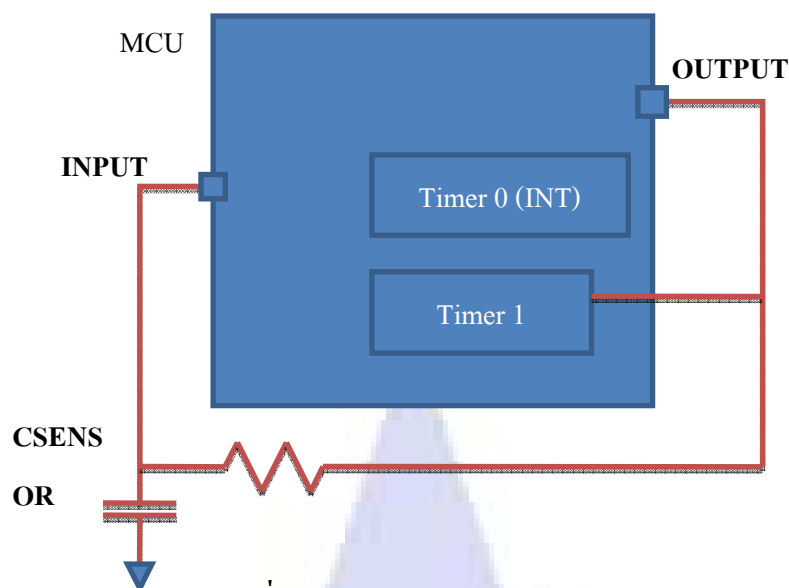
เป็นการตรวจจับการสัมผัส ด้วยการใช้คลื่นจากเซนเซอร์โดยโครงสร้างของเซนเซอร์ก็จะประกอบไปด้วยพื้นที่ที่เป็นทองแดงบนแผ่นPCB เมื่อมีการสัมผัสบนเซนเซอร์ กราฟค่า input ที่ได้ก็จะเปลี่ยนไป

โดยมีขั้นตอนการทำงานดังนี้

1. ตั้งค่า Timer1
2. สั่งให้ Timer1 นับสร้างค่าที่ขึ้นมา จนกระทั่งเกิด interrupt จาก timer0
3. ตรวจจับการสัมผัสของเซนเซอร์โดย ถ้าหากค่า raw (ค่าที่นับได้) น้อยกว่าค่า average – trip แสดงว่ามีการถูกสัมผัสอยู่ (เมื่อมีการสัมผัส ค่าที่รับจะลดลง)
4. เคลียร์ Timer1 เพื่อรอการนับต่อไป



รูปที่ 2.1 Flow Chart การทำงานของ CSM



รูปที่ 2.2 ภาพอธิบายการทำงานของ CSM

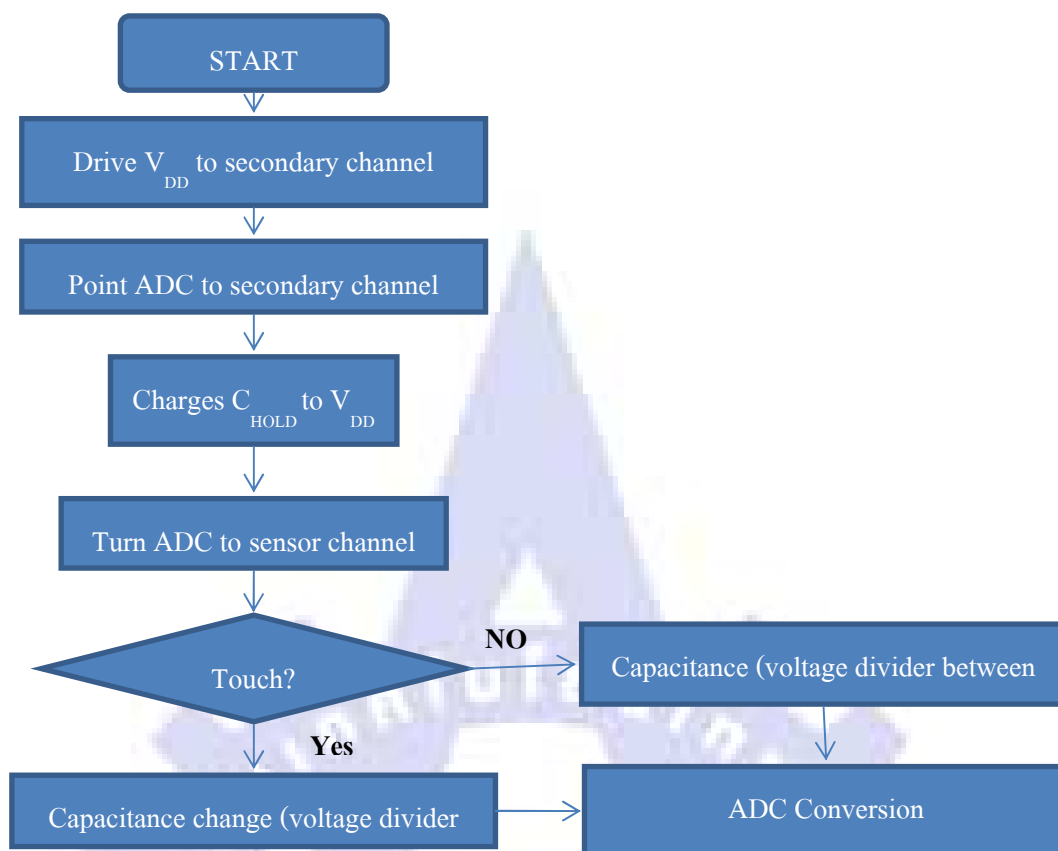
ทั้งนี้ทั้งนั้นการใช้คลื่นมาเป็นสัญญาณในการวัดนั้น มีข้อเสียตรงที่มีความเสี่ยงที่จะเกิดสัญญาณรบกวนได้สูงทำให้เกิดการทำงานที่ผิดพลาดขึ้นได้ อีกทั้งตัวชิปยังมีราคาแพงอีกด้วย

2.1.2 Capacitive Voltage Divider (CVD)

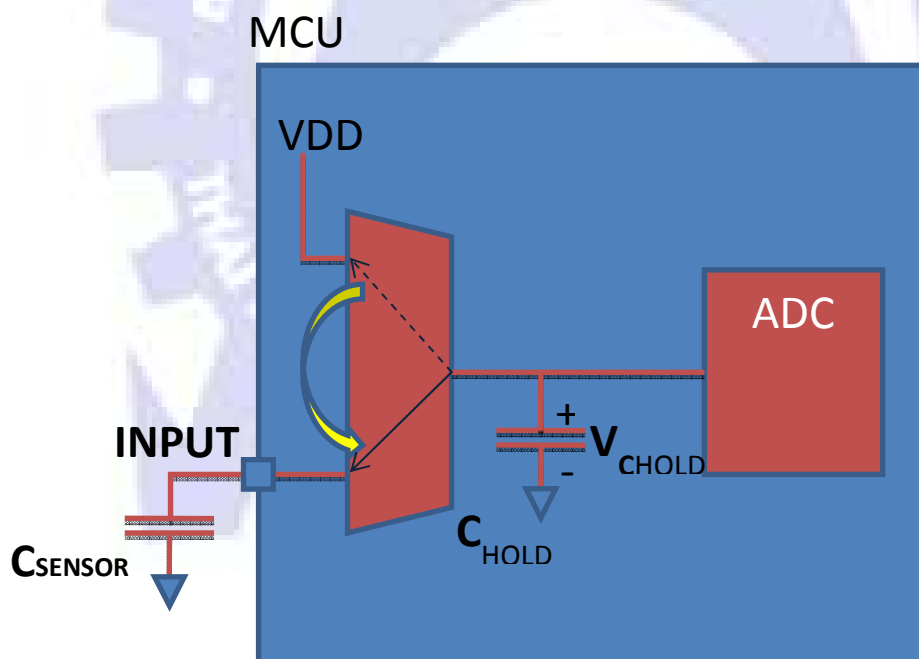
เป็นเทคโนโลยีที่อาศัยหลักการของการเก็บประจุของตัวนำมาใช้ส่งสัญญาณให้กับตัวไมโครคอนโทรลเลอร์ให้เอาไปใช้ประโยชน์ต่อ โดยโครงสร้างของเซนเซอร์ก็จะประกอบไปด้วยพื้นที่ที่เป็นทองแดงบนแผ่น PCB หรือพุดได้อีกอย่างหนึ่งว่าเป็นแผ่นเก็บประจุสำหรับสัมผัส เซนเซอร์จะต่อเข้าโดยตรงกับช่อง Analog to digital converter โดยการที่จะตรวจจับได้ จำเป็นต้องมี Analog to digital converter อย่างน้อยสองช่อง คือ ช่องหนึ่งมีไว้อ้างอิง อีกช่องมีไว้เพื่อเป็นตัวเซนเซอร์ หากช่องไหนยังไม่ได้สแกนเราควรจะให้ช่องนั้นเป็น ground หรือ VDD

ขั้นตอนการตรวจจับ

1. ไดรฟ์ VDD ไปที่ช่องอ้างอิงของ sensor
2. สับขา Analog to digital converter ไปที่ช่องอ้างอิง
3. Capacitor จะชาร์จประจุจาก VDD ผ่านช่องอ้างอิง
4. สับขา Analog to digital converter ไปที่ช่องที่มีเซนเซอร์อยู่
5. ตรวจสอบว่ามีการสัมผัสอยู่หรือไม่ โดยใช้ Analog to digital converter วัดค่าความต่างศักย์ออกมา หากไม่ถูกกดอยู่ ค่าความต่างศักย์ที่ได้ จะมาจากการแบ่งแรงดันของ C จากเซนเซอร์และ C Hold แต่ถ้าหากถูกกดอยู่ก็就会有ความต่างศักย์จากการแบ่งแรงดันระหว่าง C จากเซนเซอร์และ C Hold และ C จากวัตถุที่มาสัมผัส ทำให้ค่าความต่างศักย์ที่ได้จากการที่มีการสัมผัสนั้นน้อยกว่าการที่ตัวเซนเซอร์ไม่มีการถูกสัมผัสอยู่



รูปที่ 2.3 Flow Chart การทำงานของ CVD



รูปที่ 2.4 ภาพอธิบายการทำงานของ CVD

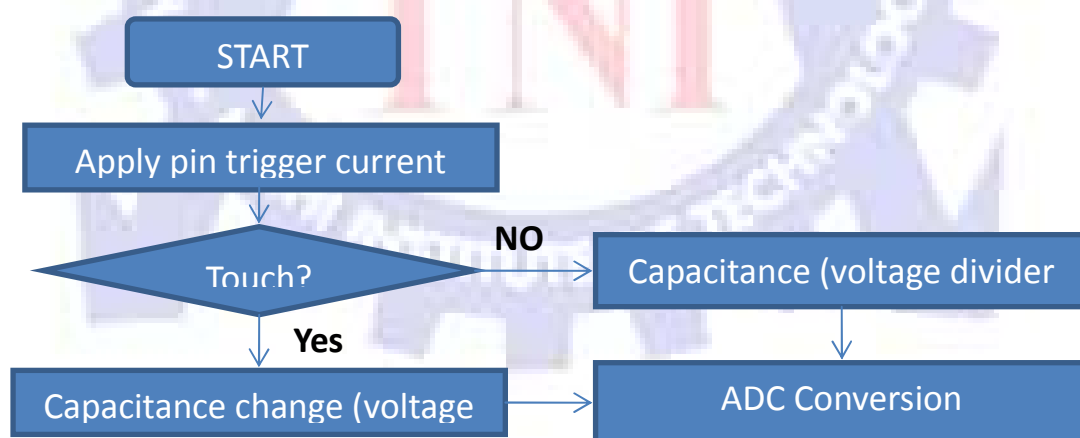
ทั้งนี้ทั้งนั้นวงจรนี้มีข้อดีที่ออกแบบวงจรได้ง่าย ใช้ทรัพยากรน้อยแต่ยังมีข้อเสียในส่วนเนื่องมาจากการที่ใช้แหล่งจ่าย V_{DD} หากแรงดันไฟฟ้าเกิดตก แล้วเกิดเขียนโปรแกรมรองรับไว้ไม่ดี การแปลค่าที่วัดได้จะเกิดการคลาดเคลื่อนทำให้เกิดการทำงานที่ผิดพลาดขึ้นได้

2.2.3 Charge Time Measurement Unit (CTMU)

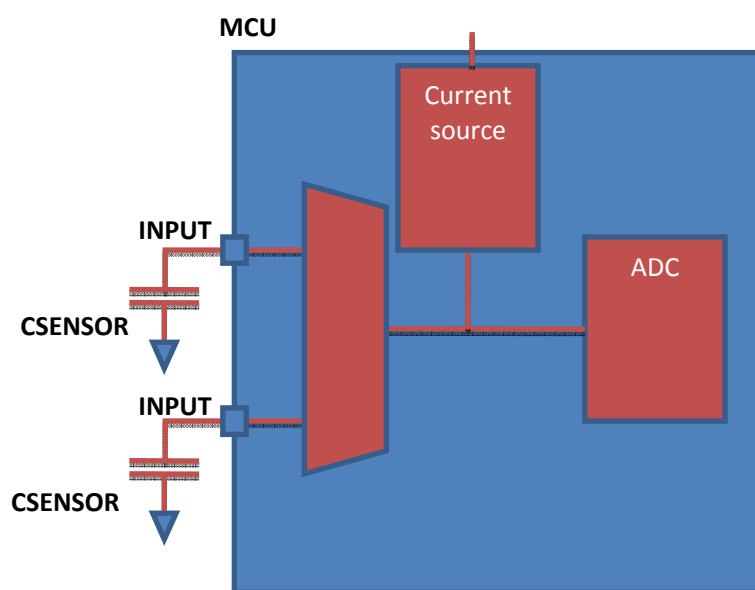
เป็นการพัฒนาขึ้นมาจาก CVD โดยเปลี่ยนข้อเสียที่มีของ CVD ที่มาจากการที่ใช้แหล่งจ่าย VCC ซึ่งถ้าหากตัวแหล่งจ่ายเป็นพวกถ่านไฟฉาย หากความต่างศักย์มีการเปลี่ยนแปลงไปอาจทำให้ค่าที่ได้มานั้นคลาดเคลื่อนไปได้ จึงมีการเพิ่มแหล่งจ่ายกระแส (Current Source) เพราะกระแสที่ได้จะเท่าเดิมเสมอ เนื่องจาก ความต้านทานคงที่ กระแสคงที่ ดังนั้นจึงได้ความต่างศักย์ที่คงที่การใช้งานทำให้เสถียรมากขึ้นแต่ก็ต้องแลกมากับชิปที่ราคาแพงขึ้น

ขั้นตอนการทำงาน

1. จ่ายแหล่งจ่ายที่ได้จากแหล่งจ่ายกระแสเข้าสู่ช่องอ้างอิงของเซนเซอร์
2. สับขา Analog to digital converter ไปที่ช่องอ้างอิง
3. Capacitor จะชาร์จประจุจาก VDD ผ่านช่องอ้างอิง
4. สับขา Analog to digital converter ไปที่ช่องที่มีเซนเซอร์อยู่
5. ตรวจสอบว่ามีการสัมผัสอยู่หรือไม่ โดยใช้ Analog to digital converter วัดค่าความต่างศักย์ออกมา หากไม่ถูกคอยู่ ค่าความต่างศักย์ที่ได้ จะมาจากการแบ่งแรงดันของ C จากเซนเซอร์และ C Hold แต่ถ้าหากถูกกดอยู่ก็จะมีค่าความต่างศักย์จากการแบ่งแรงดันระหว่าง C จากเซนเซอร์และ C Hold และ C จากวัตถุที่มาสัมผัส ทำให้ค่าความต่างศักย์ที่ได้จากการที่มีการสัมผัสนั้นน้อยกว่าการที่ตัวเซนเซอร์ไม่มีการถูกสัมผัสอยู่



รูปที่ 2.5 Flow Chart การทำงานของ CTMU

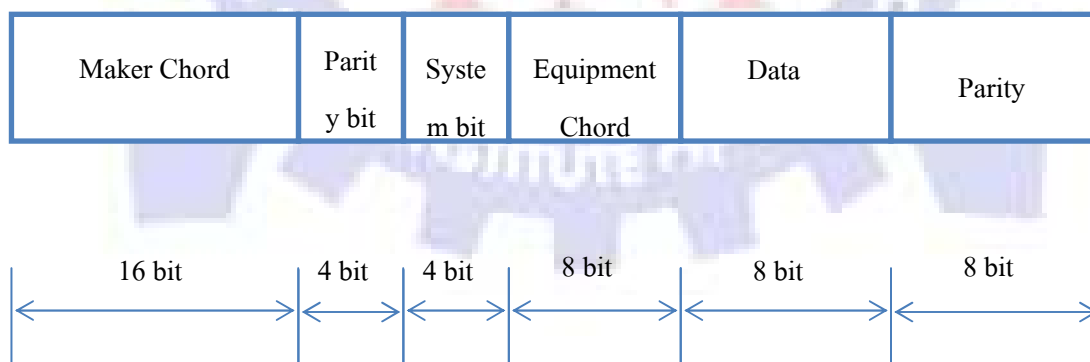


รูปที่ 2.6 ภาพอธิบายการทำงานของ CTMU

2.2 โปรโตคอลที่ใช้ในการสื่อสารระหว่างรีโมตกับทีวี

2.2.1 家製協フォーマット (Kaseikyou format)

เป็นโปรโตคอลที่จัดการโดย Associate for Electric Home Appliance (AEHA) 「家電製品協会」 ซึ่งเป็นโปรโตคอลที่ใช้กับการสื่อสารผ่านสัญญาณอินฟราเรดในเครื่องใช้ไฟฟ้าต่าง ๆ ซึ่งถูกใช้โดยผู้ผลิตรายใหญ่อื่น ๆ ยกเว้น SONY และ NEC อย่างแพร่หลาย ซึ่งขนาดของข้อมูลนั้นจะมีขนาด 20 bit แต่นอกจากนั้นก็ยังมียข้อมูลของผู้ผลิตอยู่ในข้อมูลที่ส่งจากรีโมตไปด้วย ซึ่งข้อมูลผู้ผลิตจะถูกจัดการโดย AEHA ทั้งนี้โปรโตคอลที่ใช้กับทีวีพานาโซนิคที่ใช้ในการปฏิบัติงานในครั้งนี้นรวมข้อมูลทั้งหมดแล้วมีขนาด 48 bit โดยรายละเอียดแต่ละบิตนั้นเป็นไปตามรูปที่ 2.7



รูปที่ 2.7 Panasonic's Kaseikyou format data bit

การที่จะสื่อสารระหว่างรีโมตกับอุปกรณ์นั้น รีโมตจะส่งข้อมูลดังนี้

1. ส่งสัญญาณว่ากำลังจะส่งข้อมูล
2. ส่งข้อมูลทั้งหมด 48 บิต โดยแต่ละบิตมีรายละเอียดดังนี้
 - บิตที่ 1 ถึงบิตที่ 16 คือบิตที่ระบุผู้ผลิต
 - บิตที่ 17 ถึงบิตที่ 20 คือ Parity bit Manufacturer
 - บิตที่ 21 ถึงบิตที่ 24 คือ System bit ที่เป็นตัวระบุว่าใช้ระบบไหน
 - บิตที่ 25 ถึงบิตที่ 32 คือ บิตที่ระบุว่าสื่อสารข้อมูลกับอุปกรณ์อะไร
 - บิตที่ 33 ถึงบิตที่ 40 คือบิตข้อมูลคำสั่ง
 - บิตที่ 41 ถึงบิตที่ 48 คือ Parity Bit สำหรับข้อมูลคำสั่ง
3. ส่ง End bit เพื่อเลิกการติดต่อ

2.3 ส่วนช่วยในการพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์

2.3.1 ชุดพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์

2.3.1.1 MPLAB IDE เป็นโปรแกรมที่ใช้ในการเขียนและแก้ไขซอฟต์แวร์ก่อนที่จะเขียนลงไปบนไมโครคอนโทรลเลอร์ตระกูล PIC และ dsPIC ของบริษัทไมโครชิพ

MPLAB SIM เป็นโปรแกรมที่ใช้ในการจำลองการทำงานของไมโครคอนโทรลเลอร์ที่เราได้เขียนลงไป ใน MPLAB IDE 8.73 โดยสามารถดูได้ตั้งแต่ลอจิกที่ออกมาจากพินต่างๆ ค่าที่อ่านได้จาก Analog to digital converter เป็นต้น

วิธีการติดตั้ง MPLAB IDE

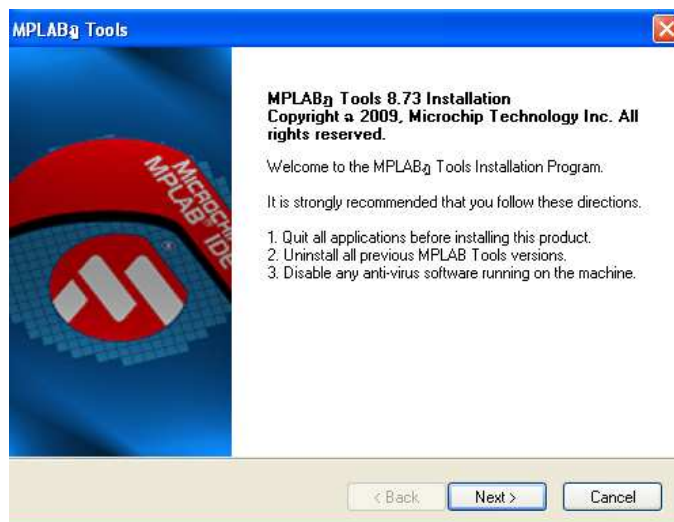
1. ดาวน์โหลด MPLAB IDE 8.76 จาก website

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002 แล้วคลิก MPLAB IDE

Downloads			
Title	Date Published	Size	DL
Advanced Debugging Techniques- Lab 1 of 3	11/29/2010 11:39:00 AM	21587 KB	
MATLAB Device Blocks for MPLAB IDE	3/29/2011 9:42:01 AM	36 KB	
MPASM/MPLINK User's Guide	4/8/2009 3:52:41 PM	2896 KB	
MPLAB Assembler, Linker and Utilities for PIC24 MCUs and dsPIC DSCs User's Guide	6/15/2011 9:23:04 PM	1527 KB	
MPLAB IDE Current Release Notes	8/11/2011 8:23:36 AM	261 KB	
MPLAB IDE User's Guide	1/20/2009 12:09:31 PM	4232 KB	
MPLAB IDE v8.76	8/11/2011 8:02:24 AM	125973 KB	
MPLAB® X IDE Product Overview	7/11/2011 4:17:30 PM	182 KB	
Quick Guide to Microchip Development Tools	3/4/2011 10:09:50 AM	582 KB	
Software Solutions and Tools for the 16-bit and 32-bit Designer	6/6/2011 2:52:57 PM	3138 KB	

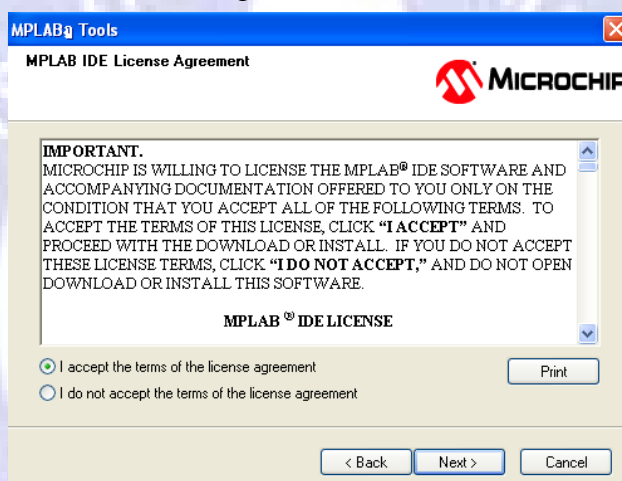
รูปที่ 2.8 ลิงค์ดาวน์โหลด MPLAB IDE

2. คลิก Next>



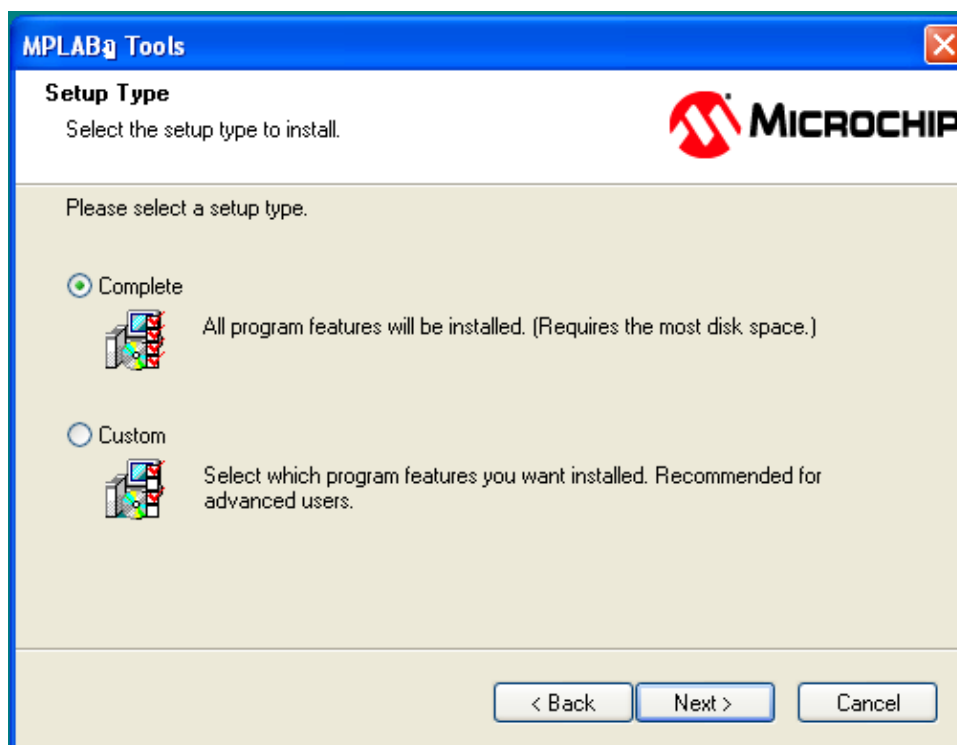
รูปที่ 2.9 MPLAB Tools Installation Program

3. เลือก “I accept the term of the license agreement” แล้วจึงคลิก Next>



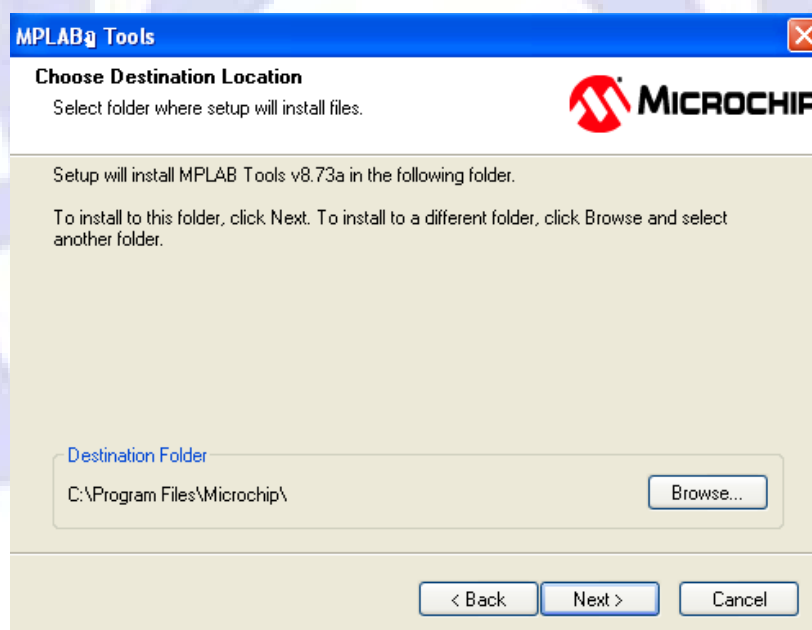
รูปที่ 2.10 MPLAB IDE License Agreement

4. เลือก “Complete” แล้วจึงคลิก Next>

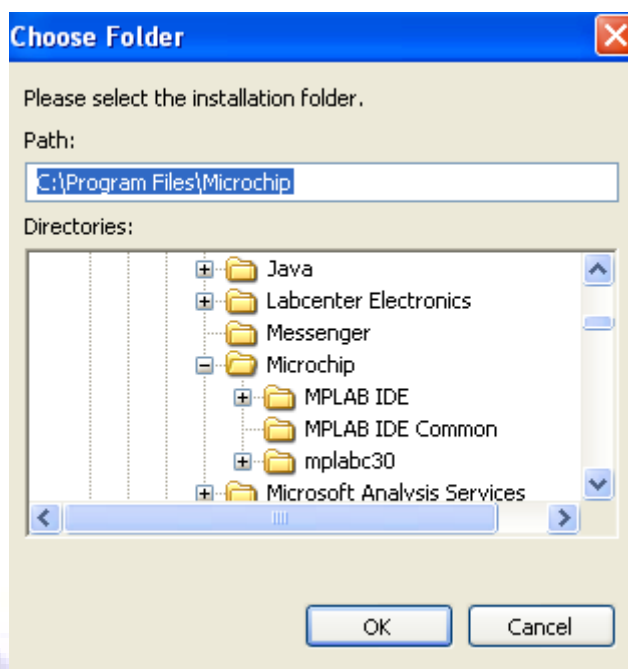


รูปที่ 2.11 MPLAB Setup type selection to install

5.เลือกโฟลเดอร์ที่ต้องการลง MPLAB IDE โดยคลิกที่ Browse... แล้วจึงเลือกโฟลเดอร์ปลายทาง แล้วจึงคลิก OK แล้ว จึงคลิก Next

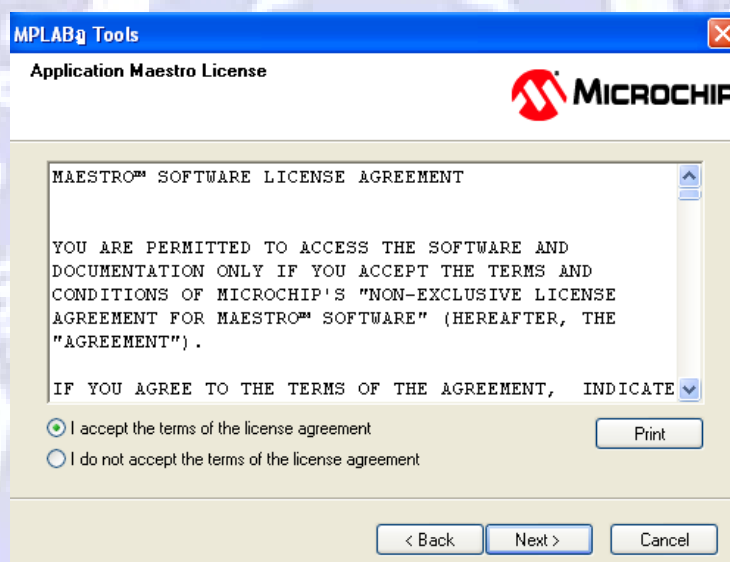


รูปที่ 2.12 MPLAB Choose Destination Location



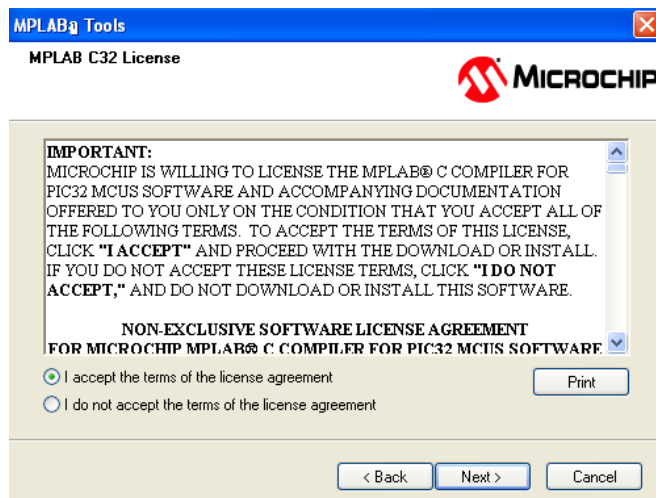
รูปที่ 2.13 MPLAB Choose Folder

6. เลือก “I accept the terms of the license agreement” แล้วคลิก Next>



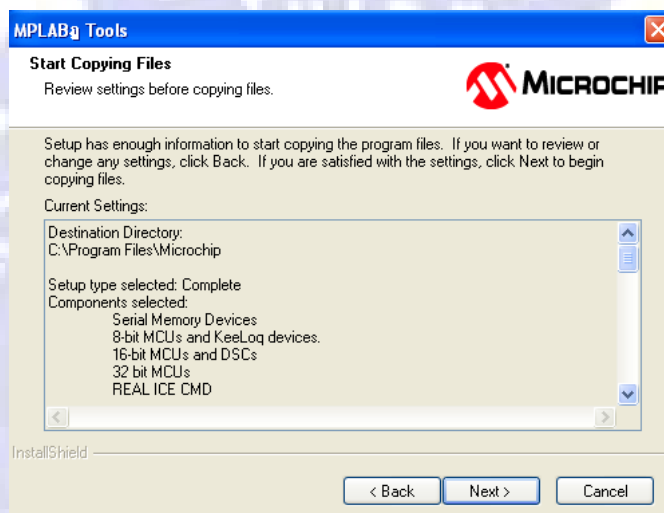
รูปที่ 2.14 MPLAB Tools Application Maestro License

7. เลือก “I accept the terms of the license agreement” แล้วคลิก Next>



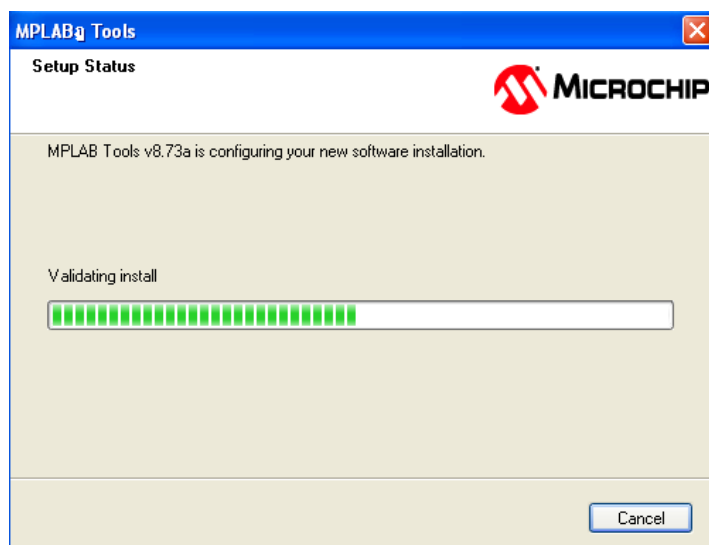
รูปที่ 2.15 8 MPLAB Tools C32 License

8.คลิก Next>



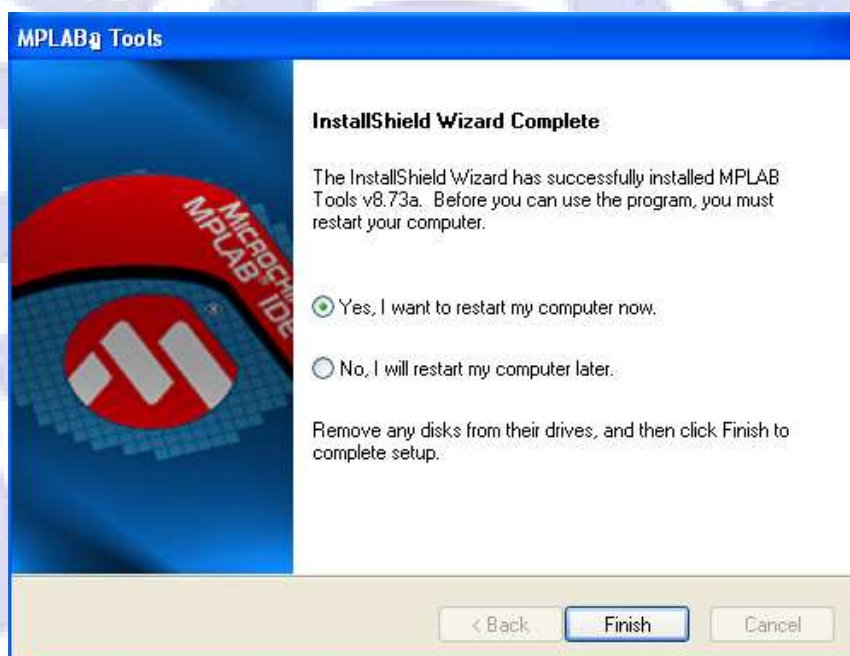
รูปที่ 2.16 MPLAB Tools Start Copying Files

9.รอนจนกระทั่งติดตั้งโปรแกรมเสร็จ



รูปที่ 2.17 MPLAB Tools Setup Status

10. รอจนกระทั่งติดตั้งโปรแกรมเสร็จจะมีกล่องข้อความขึ้นมาว่าจะติดตั้งคอมไพเลอร์ Hi-TECH C หรือไม่ ให้เลือก NO จากนั้นถ้ายังไม่ต้องการให้เครื่องคอมพิวเตอร์ restart ให้เลือก No, I will restart my computer later. แล้วจึงคลิก Finish แต่ถ้าหากต้องการให้ restart เครื่องเลย ให้เลือก Yes, I want to restart my computer now. แล้วจึงคลิก Finish จบการติดตั้ง



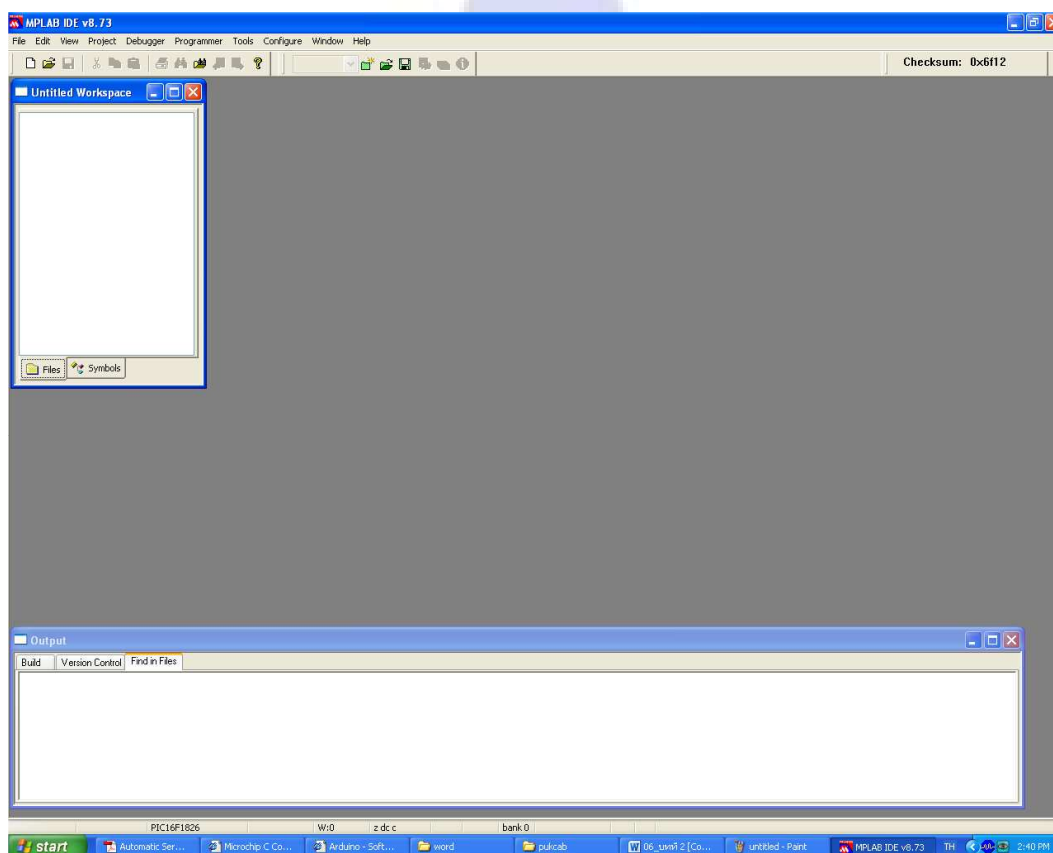
รูปที่ 2.18 MPLAB Tools InstallShield Wizard Complete

การใช้งาน MPLAB IDE

1.เปิดโปรแกรมขึ้นมาโดยการดับเบิลคลิกที่ไอคอน MPLAB IDE ดังรูปที่ 2.18 จะได้หน้าต่างโปรแกรมขึ้นมาดังรูปที่ 2.19

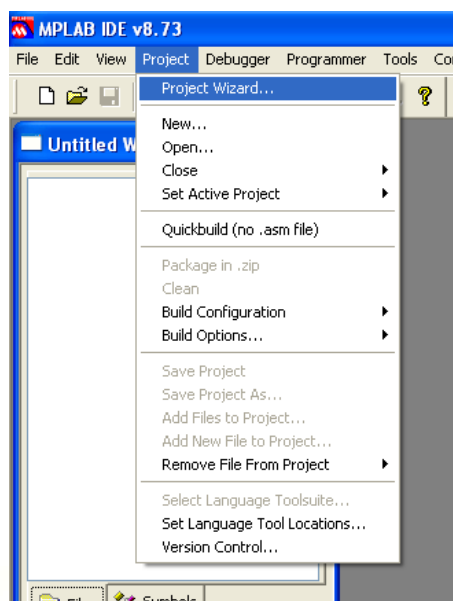


รูปที่ 2.19 MPLAB IDE Icon



รูปที่ 2.20 MPLAB IDE Interface

2.สร้างโปรเจกขึ้นมาโดยคลิกที่ Project -> Project Wizard. . .



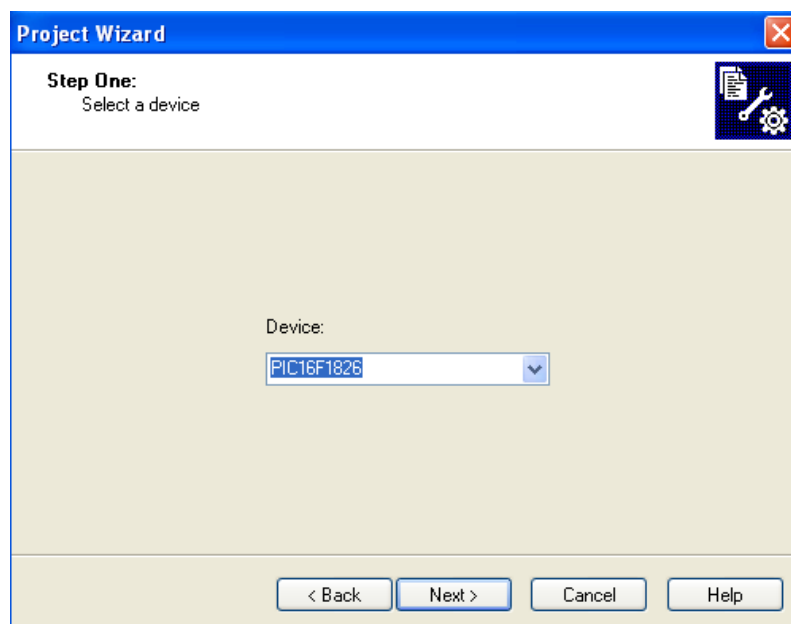
รูปที่ 2.21 MPLAB IDE Project

3.คลิกที่ Next >



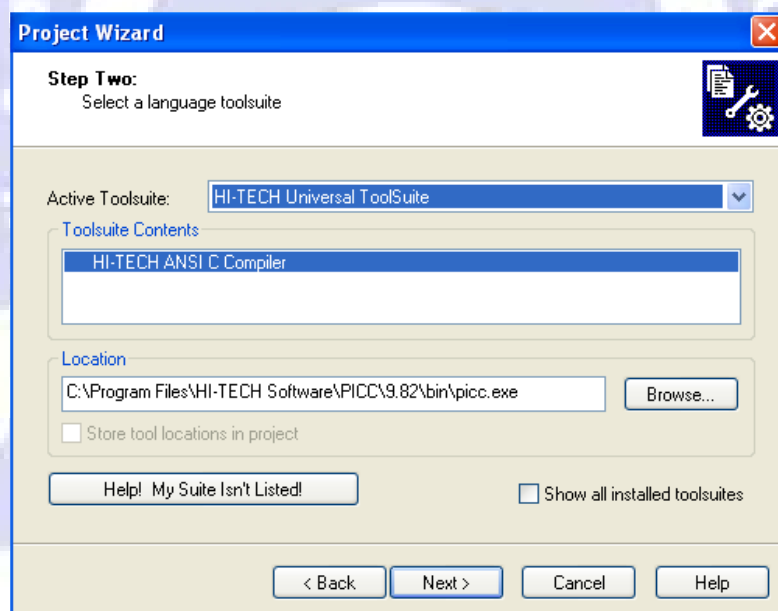
รูปที่ 2.22 MPLAB IDE Project Wizard

4.คลิกเลือกเบอร์ microcontroller ที่เราต้องการจะใช้ ให้เลือกเบอร์ “PIC16F1826” จากนั้นคลิก Next>



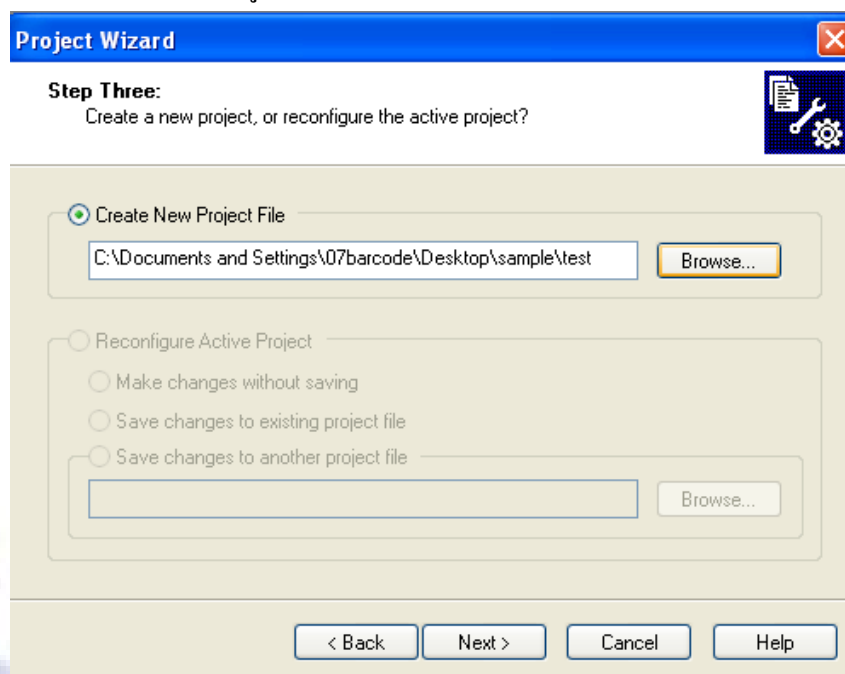
รูปที่ 2.23 MPLAB IDE Select Device

5.ตรงช่องของ Active Tool suite ให้คลิกเลือกที่ “HI-TECH Universal ToolSuite” จากนั้นในส่วน
ของ Toolsuite Contents ให้คลิกเลือก HI-TECH ANSI C Compiler และเลือก Location ที่เราได้ลง
คอมไพเลอร์ไว้แล้วจึงคลิก Next >

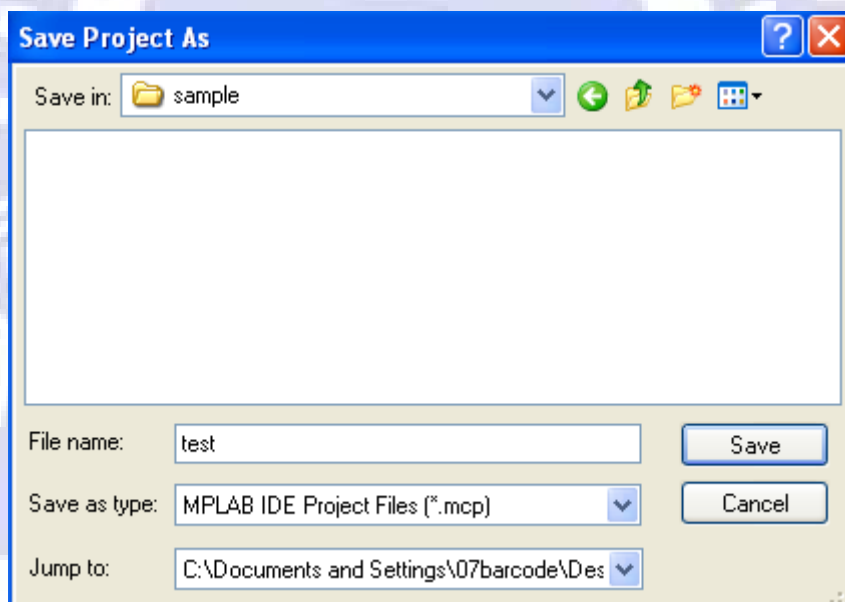


รูปที่ 2.24 MPLAB IDE Select a language tool suite

6.เลือกโฟลเดอร์ที่จะเก็บโปรเจกต์ที่กำลังจะเขียนขึ้นมาโดยคลิกที่ Browse แล้วเลือกโฟลเดอร์ที่ต้องการจากนั้นพิมพ์ชื่อโปรเจกต์ลงไปในช่อง File name แล้วจึงคลิก Save จากนั้นจึงคลิก Next เพื่อเข้าสู่ขั้นตอนต่อไป

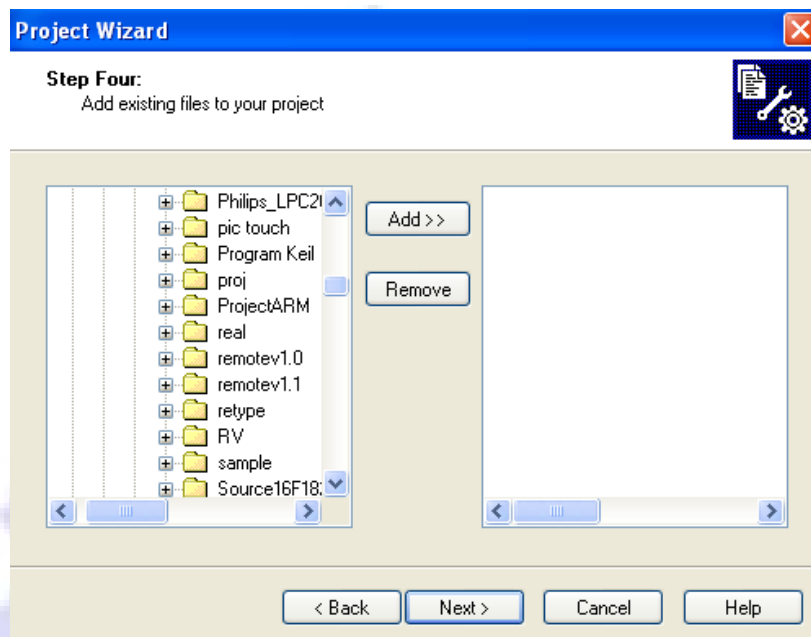


รูปที่ 2.25 MPLAB IDE Create a new Project



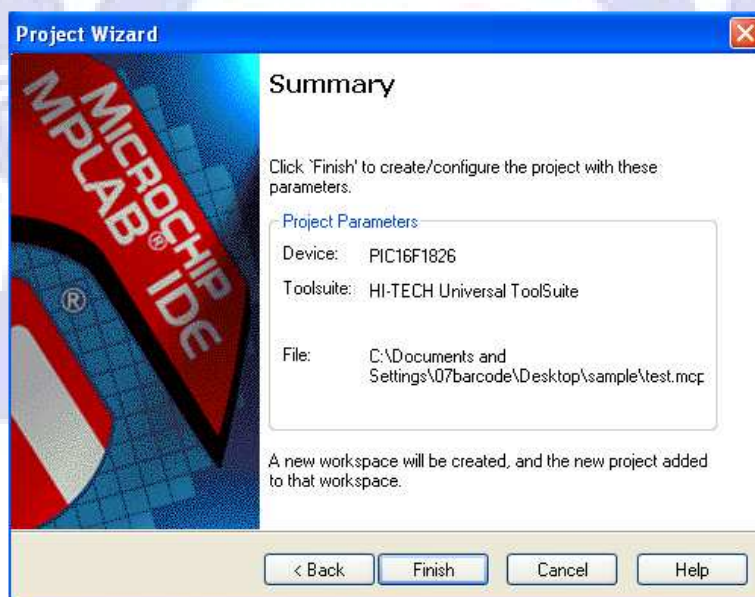
รูปที่ 2.26 MPLAB IDE Save Project as

7. ถ้าหากเรามีไฟล์ที่ต้องการ นำไปใช้ในโปรเจกต์อยู่แล้วก็ให้เลือกไฟล์เข้าไปใช้ได้เลยโดยการคลิกเลือกที่ไฟล์ที่ต้องการแล้วจึงคลิกที่ Add >> แต่ถ้าต้องการที่จะ นำไฟล์ออกจากโปรเจกต์ก็ให้เลือกที่ไฟล์ที่ต้องการนำออกแล้วจึงคลิกที่ Remove แต่ถ้าไม่มีไฟล์อะไรที่ต้องการเพิ่มเข้าไปในโปรเจกต์ก็คลิก Next >> ได้เลย (เรายังสามารถเพิ่มไฟล์เข้าไปภายหลังได้)



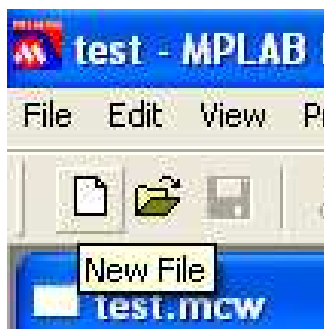
รูปที่ 2.27 MPLAB IDE Add existing file to your project

8. จากนั้นจะมีเมสเสจบอกสรุปข้อมูลว่าอุปกรณ์ที่เลือกนั้นเบอร์อะไร ใช้คอมไพเลอร์ตัวไหน เมื่อเห็นว่าถูกต้องแล้วให้คลิกที่ Finish เพื่อจบการสร้างโปรเจกต์

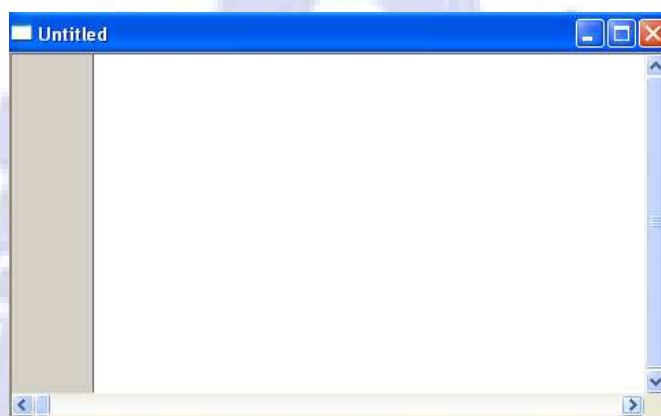


รูปที่ 2.28 MPLAB IDE Summary

9.คลิกที่รูปกระดาดังรูปที่ 2.28 เพื่อสร้างไฟล์ใหม่จะได้หน้าต่างใหม่มาดังรูปที่ 2.29

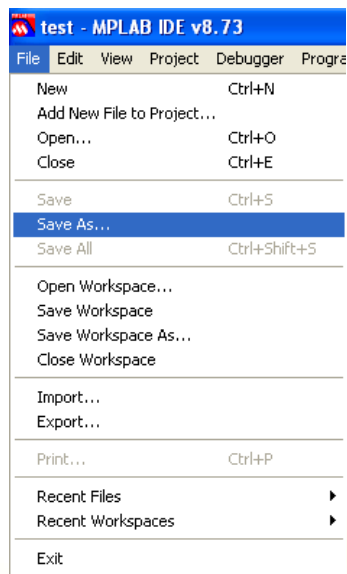


รูปที่ 2.29 MPLAB IDE New File



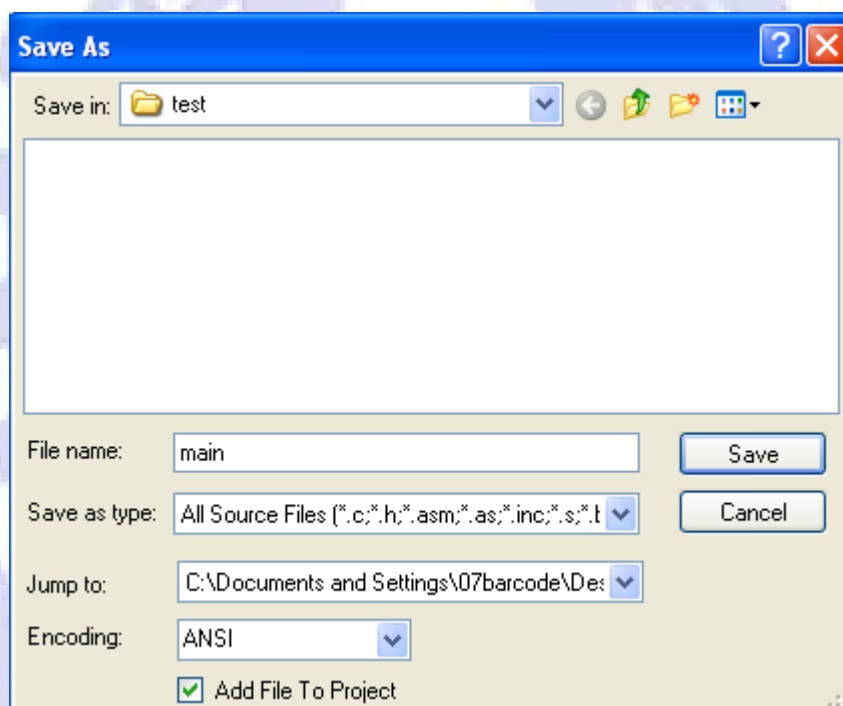
รูปที่ 2.30 MPLAB IDE Blank space

10.จากนั้นคลิกที่ File -> Save As



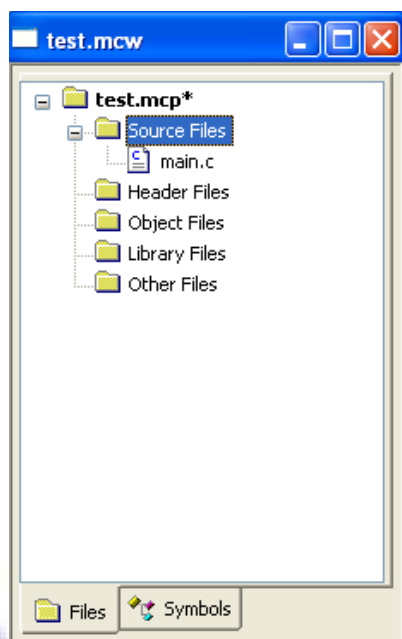
รูปที่ 2.31 MPLAB IDE File

10. เลือกโฟลเดอร์ที่จะให้ไฟล์ของเราอยู่ตามต้องการแล้วตั้งชื่อว่า “main.c” และคลิกที่ “Add File To Project” เพื่อใส่ไฟล์เข้าไปในโปรเจก



รูปที่ 2.32 MPLAB IDE Save As

11. ให้นำแนชดว่าไฟล์ของเราเข้าไปอยู่ในโปรเจกแล้วดังรูป



รูปที่ 2.33 MPLAB IDE Directory

2.3.1.2 ARDUINO IDE

เป็นโปรแกรมใช้ในการเขียนและแก้ไขซอฟต์แวร์ก่อนที่จะเขียนลงไปบนไมโครคอนโทรลเลอร์และเขียนข้อมูลลงบนไมโครคอนโทรลเลอร์ตระกูล ARDUINO ของบริษัท ATMEAL โดยมีคอมพิวเตอร์ในตัวเอง

การติดตั้ง ARDUINO IDE

1.ดาวน์โหลด ARDUINO IDE ได้ที่ <http://www.arduino.cc/en/Main/Software> คลิกที่ Window

Download

Arduino 0022 ([release notes](#)), hosted by [Google Code](#):

***Windows**

*Mac OS X

*Linux: [32 bit](#), [64 bit](#)

*[source](#)

Also available from [Arduino.cc](#): [Windows](#), [Mac OS X](#),
[Linux \(32bit\)](#) ([64bit](#)), [Source](#)

Next steps

[Getting Started](#)

[Reference](#)

[Environment](#)

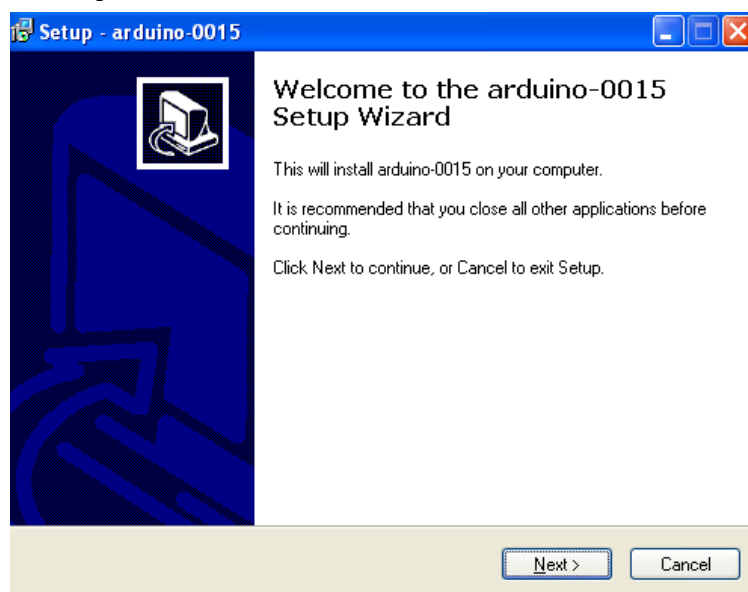
[Examples](#)

[Foundations](#)

[FAQ](#)

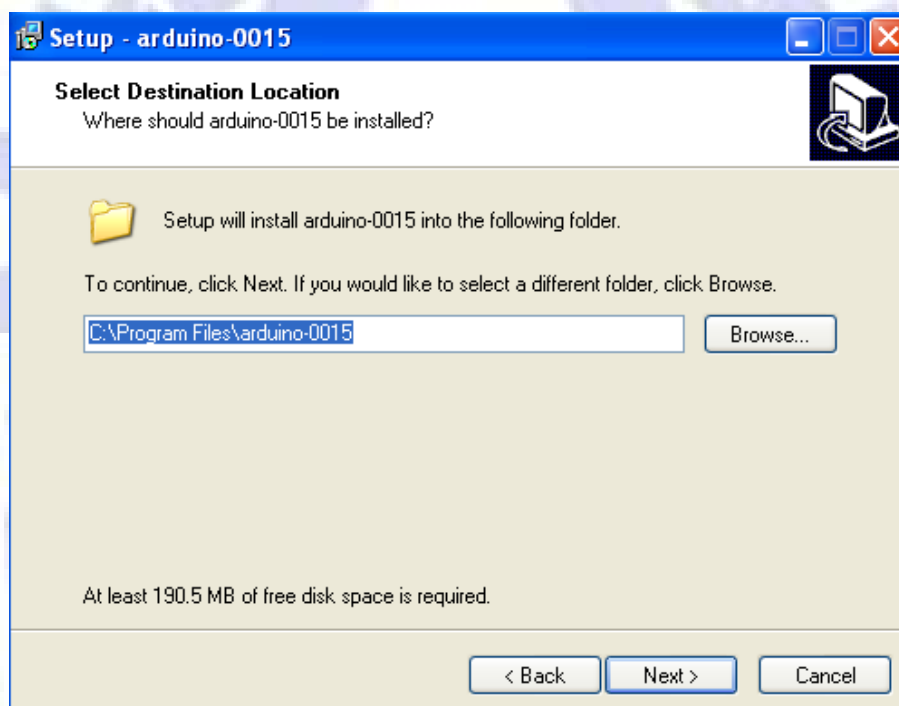
รูปที่ 2.34 ARDUINO IDE Download Link

2.เปิดไฟล์ ArduinoSetup ขึ้นมาแล้วคลิก Next>



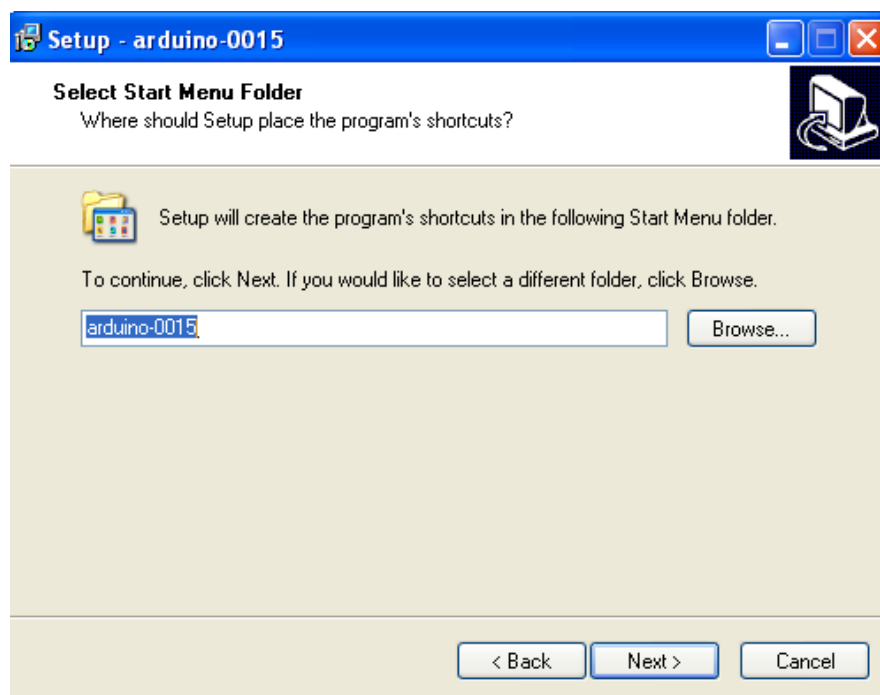
รูปที่ 2.35 Arduino-0015 Setup Wizard

3.เลือกโฟลเดอร์ปลายทางที่ต้องการติดตั้ง Arduino IDE



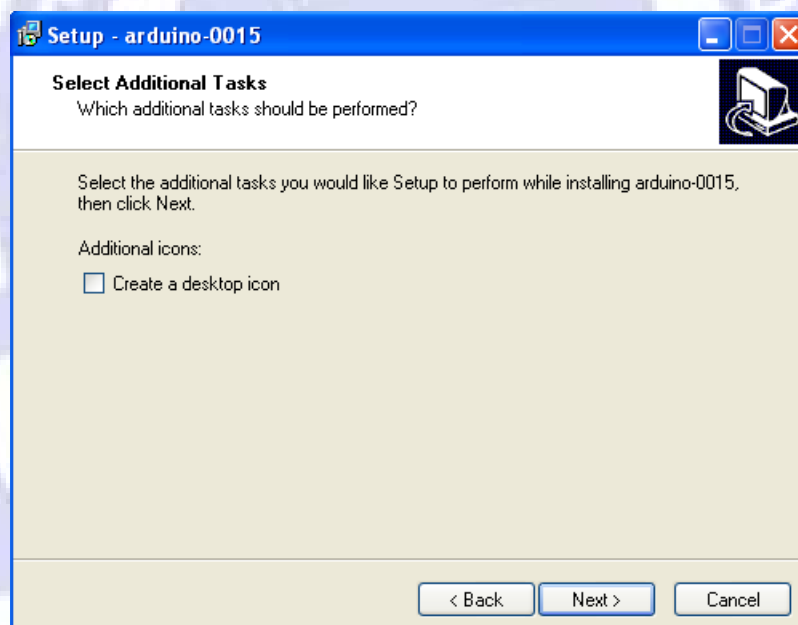
รูปที่ 2.36 ARDUINO IDE Select Destination Location

4.เลือกโฟลเดอร์ปลายทางที่ต้องการให้มีช็อตคัต



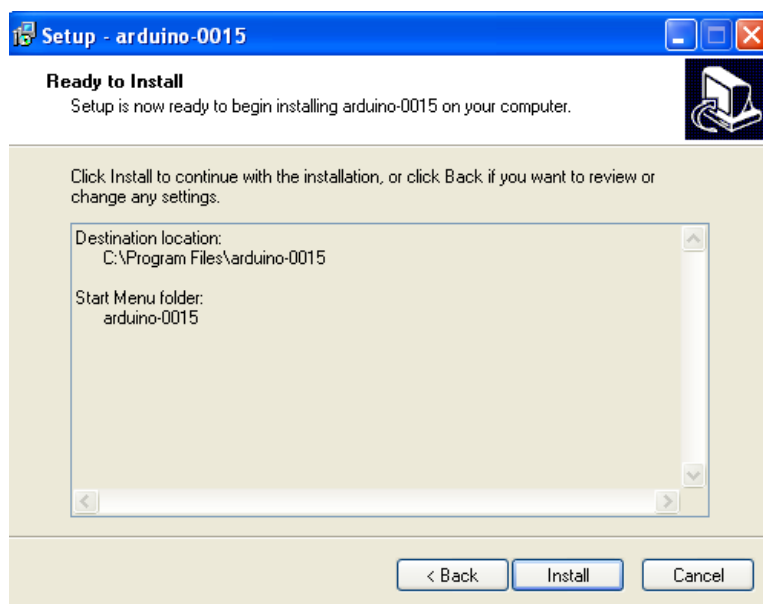
รูปที่ 2.37 ARDUINO IDE Select Start Menu Folder

5.ถ้าหากต้องการให้มีชื่อย่อติดตั้งตรงหน้า Desktop ด้วย ก็ให้ติ๊กเลือก “Create a desktop icon” แล้วจึงคลิก Next> แต่ถ้าไม่ต้องการ ให้คลิก Next ได้เลย



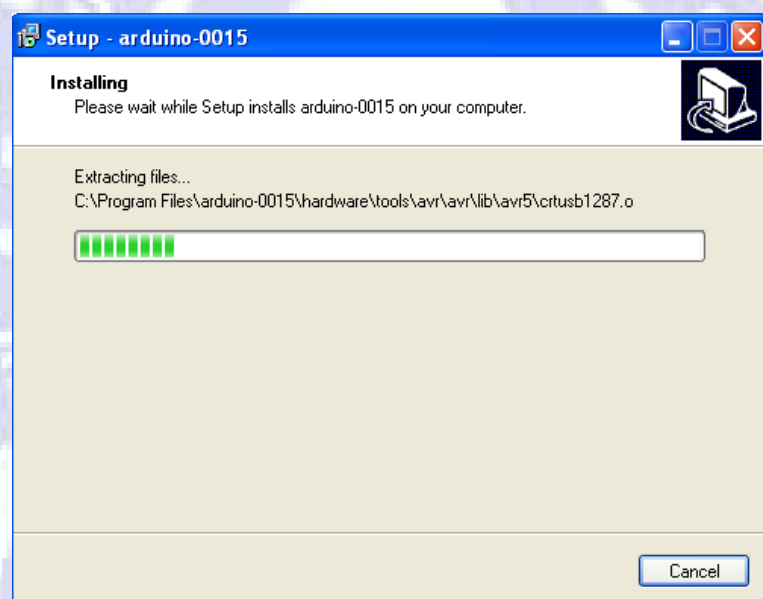
รูปที่ 2.38 ARDUINO IDE Select Additional Task

6.คลิก Install



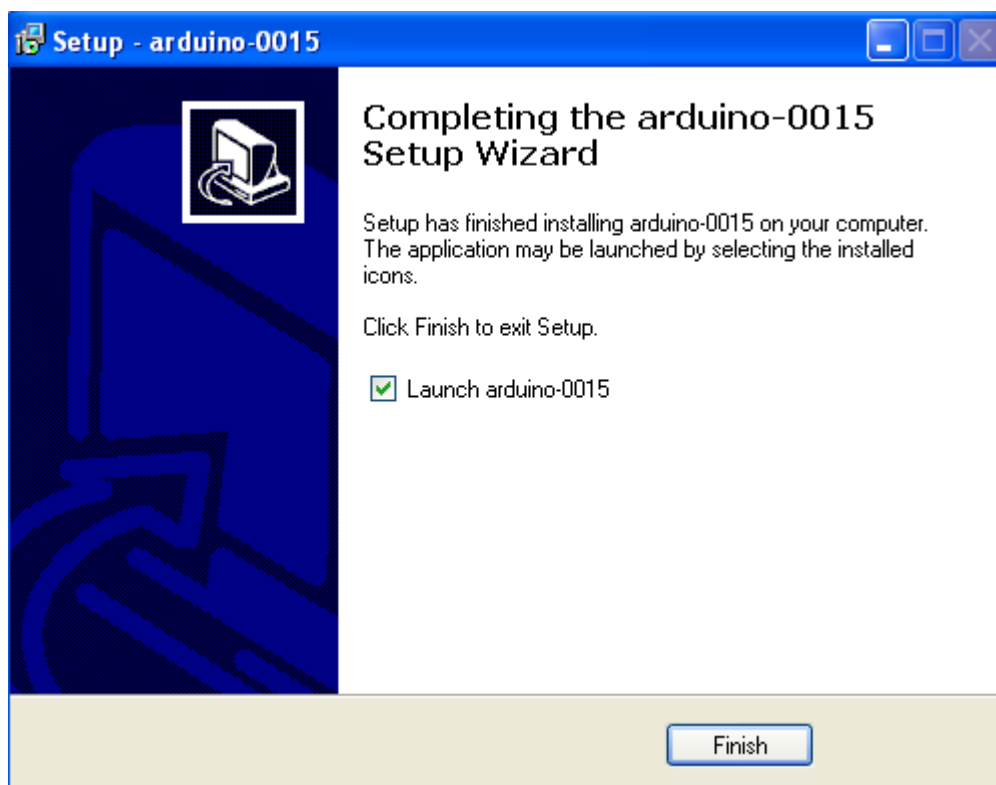
รูปที่ 2.39 ARDUINO IDE Ready to install

7.รอนกว่าการติดตั้งจะเสร็จสิ้น

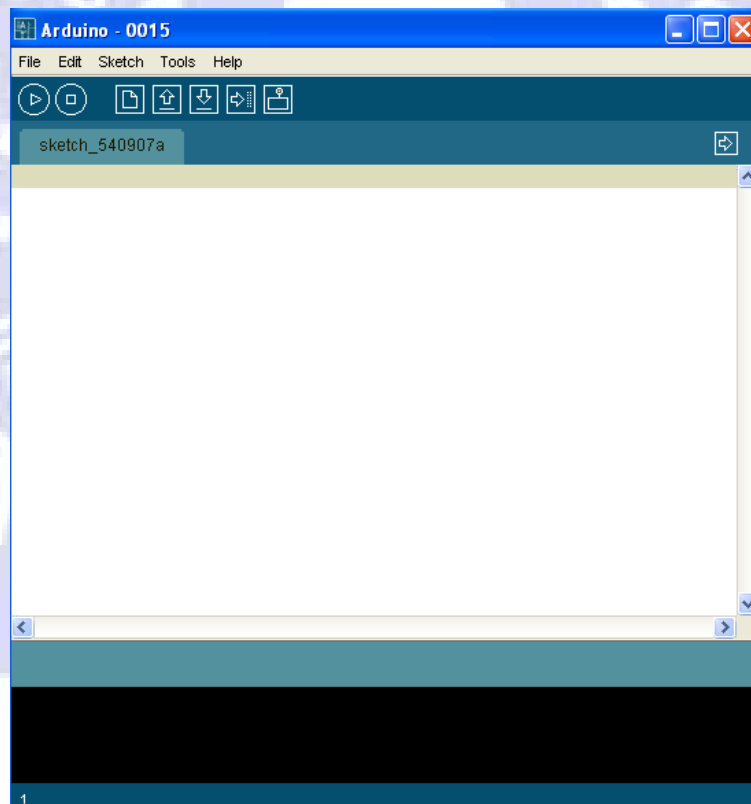


รูปที่ 2.40 ARDUINO IDE Installing

8.เสร็จสิ้นการติดตั้ง ถ้าหากต้องการเปิดโปรแกรมArduino IDE ขึ้นมาเลย ให้เลือก “Launch arduino” แล้วจึง คลิก Finish แต่ถ้าไม่ต้องการ ก็ให้คลิก Finish เพื่อจบการติดตั้งได้เลย



รูปที่ 2.41 ARDUINO IDE Completing the arduino-0015 Setup Wizard

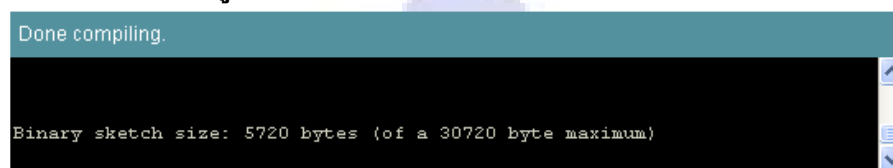


รูปที่ 2.42 หน้าต่างโปรแกรม ARDUINO IDE

เมื่อเขียนโปรแกรมแล้วต้องการทำการ Compile ให้คลิกที่สัญลักษณ์ ดังรูปที่ 2.42 โปรแกรมจะทำการคอมไพล์แล้วสรุปออกมาว่าโปรแกรมที่เขียนไปใช้หน่วยความจำไปเท่าไรดังรูปที่ 2.43



รูปที่ 2.43 ARDUINO IDE Compile button



รูปที่ 2.44 ARDUINO IDE Status

จากนั้นถ้าหากว่าเราต้องการเขียนข้อมูลลงไปในบอร์ด Arduino ให้คลิกที่สัญลักษณ์ดังรูปที่ 2.44 และเราสามารถดูการสื่อสารระหว่างสาย Serial ได้ด้วยการคลิกที่สัญลักษณ์ดังรูปที่



รูปที่ 2.45 ARDUINO IDE Write button



รูปที่ 2.46 ARDUINO IDE Serial Monitoring button

2.3.2 คอมไพเลอร์ที่ใช้ในการพัฒนาผลิตภัณฑ์

2.3.2.1 Hi-Tech C Compiler

เป็นคอมไพเลอร์ที่ทำหน้าที่แปลงโปรแกรมที่เราเขียนแปลงจากภาษาซีไปเป็นภาษาแอสเซมบลี เป็นโปรแกรมที่ใช้ร่วมกันกับ MPLAB IDE โดยเวอร์ชันที่ใช้จะเป็นเวอร์ชันไลท์ (Lite) ซึ่งสามารถใช้ได้ฟรี

วิธีการติดตั้ง

1.ดาวน์โหลด Hi-Tech C Compiler จาก

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en534868 โดยตัวที่ดาวน์โหลดนี้จะเป็นเวอร์ชัน Lite ซึ่งสามารถดาวน์โหลดได้ฟรี

All Compilers

	PRO	Standard	Lite	Evaluation
PIC10/12/16 MCUs	 HI-TECH C	 HI-TECH C	 HI-TECH C	 HI-TECH C
PIC18 MCUs	 HI-TECH C	 MPLAB C	 MPLAB C	 MPLAB C
PIC24 MCUs and dsPIC DSCs		 MPLAB C	 MPLAB C	 MPLAB C
PIC32 MCUs		 MPLAB C	 MPLAB C	 MPLAB C

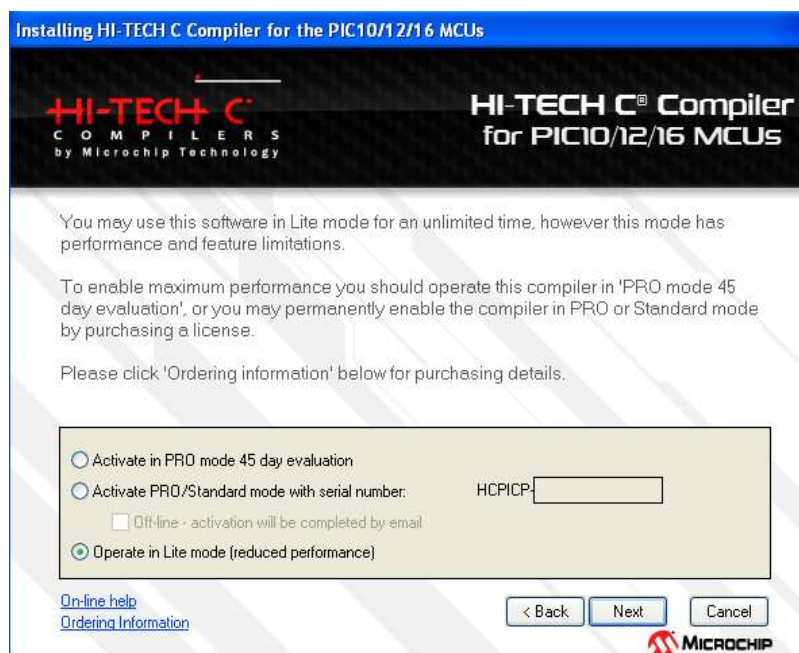
รูปที่ 2.47 Microchip's Compiler list

2.เปิดตัวติดตั้ง picc-9_82-win แล้วจึงคลิก Next



รูปที่ 2.48 HI-TECH C installer

3.เลือก “Operate in Lite mode (reduced performance)”



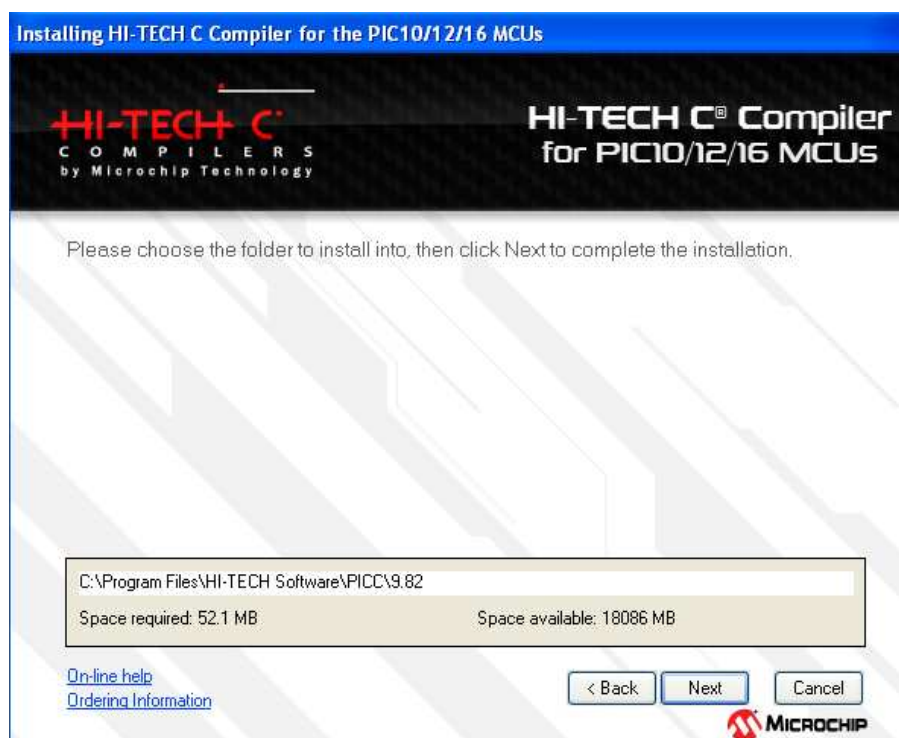
รูปที่ 2.49 HI-TECH C mode selection

4.เลือก I accept the terms of the licence agreement แล้วจึงคลิก Next



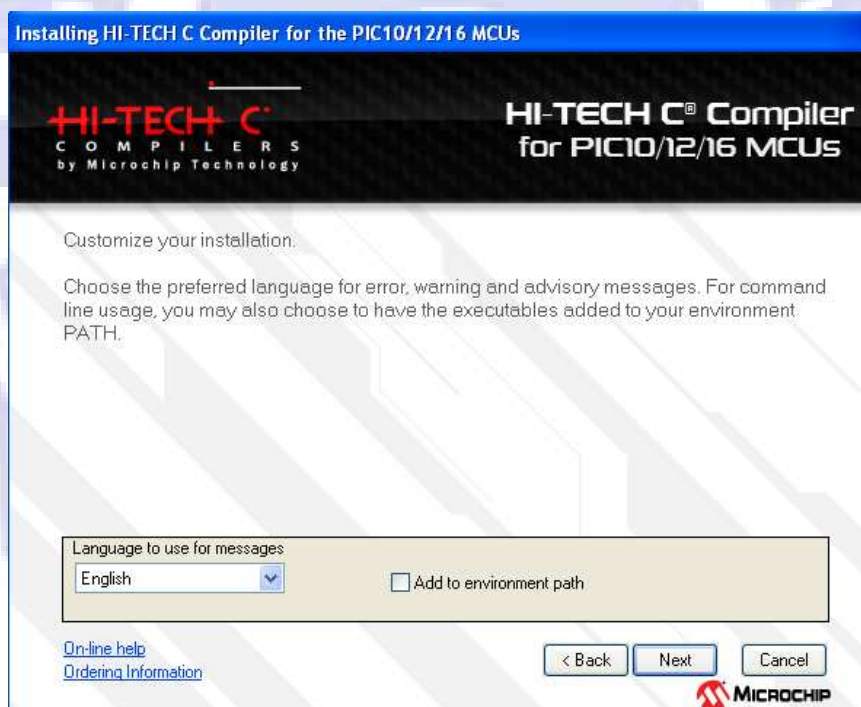
รูปที่ 2.50 HI-TECH License agreement

5.เลือกไฟล์เคอร์ที่ต้องการจะติดตั้ง จากนั้นจึงคลิก Next



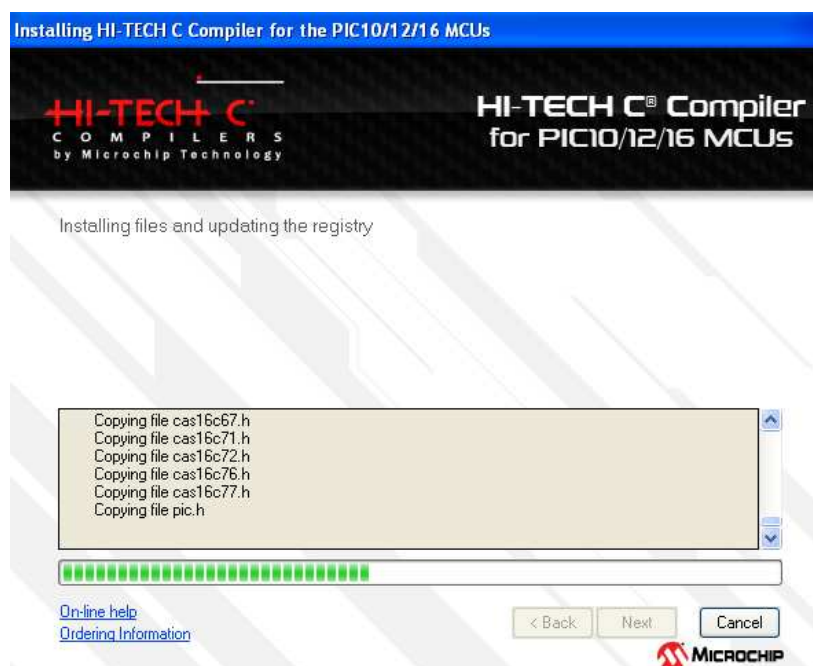
รูปที่ 2.51 HI-TECH C Install destination selection

6.เลือกภาษาที่ใช้สำหรับการเตือนในกรณีที่เกิด error ขึ้น ให้เลือก English แล้วจึงคลิก Next



รูปที่ 2.52 HI-TECH C language selection

7. รอให้การติดตั้งเสร็จสิ้น



รูปที่ 2.53 HI-TECH C Installing

8. คลิก Finish เพื่อเสร็จสิ้นการติดตั้งโปรแกรม



รูปที่ 2.54 HI-TECH C Finish installation

2.3.3 เครื่องมือที่ใช้ในการเขียนข้อมูลลงไปบนไมโครคอนโทรลเลอร์

2.2.3.1 PICKIT3 อุปกรณ์ที่สามารถใช้เขียนโปรแกรมลงไปบนไมโครคอนโทรลเลอร์ตระกูล PIC ได้ทุกรุ่นที่มีผลิตในปัจจุบัน อีกทั้งยังสามารถทำหน้าที่เป็น Debugger ช่วยให้เราสามารถดูการทำงานที่ละจังหวะๆ ได้ว่าแต่ละจังหวะการทำงาน โปรแกรมที่เขียนขึ้นสามารถไปสั่งการไมโครคอนโทรลเลอร์ได้ถูกต้องหรือไม่ สามารถเชื่อมต่อกับคอมพิวเตอร์ได้ผ่านทางสาย USB อีกทั้งยังสามารถดึงไฟฟ้าจากช่อง USB ของคอมพิวเตอร์มาเป็นขั้วพลาซให้กับตัวไมโครคอนโทรลเลอร์ได้ด้วย จึงสะดวกมาสำหรับการใช้งาน



รูปที่ 2.55 PICKit3



รูปที่ 2.56 PICKit3 Component

2.2.1.1 ส่วนประกอบต่างๆภายนอกของ PICKit3

1. Lanyard Connection

มีรูให้คล้องเชือกทำ

2. USB Port Connection

มีช่องเชื่อมต่อ USB mini-B connector และยังสามารถใช้แหล่งจ่ายจากสาย USB ได้อีกด้วย

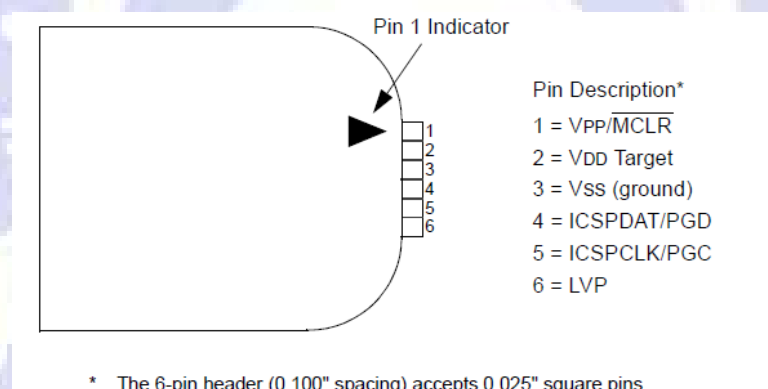
3. Pin 1 Marker

คือจุดบอกว่าPinไหนคือPinที่1

4. Programming Connector

คือช่องที่ใช้เสียบเพื่อเขียนโปรแกรมลงไมโครคอนโทรลเลอร์เป็นแบบเฮดเดอร์สำหรับ 6 Pin

โดยแต่ละช่องใช้สำหรับเสียบPin ต่างๆดังรูปที่ 2.56



รูปที่ 2.57 PICKit3 Pin Description

5. Status LED

จะมีหลอด LED แสดงสถานะ อยู่สามหลอด

หลอดที่1 Power (สีเขียว) แสดงสถานะว่า PICKIT3 ได้รับแหล่งจ่ายผ่านทาง USB Port

หลอดที่2 Active(สีฟ้า) แสดงว่า PICKit3 เชื่อมต่ออยู่กับคอมพิวเตอร์ผ่านทางสาย USB และสามารถสื่อสารกันระหว่างตัว PICKit3 และตัวคอมพิวเตอร์ได้

หลอดที่3 Status (สีเหลือง) แสดงว่า PICKit3 อยู่ในสถานะไม่ว่าง(Busy) เช่นขณะกำลังเขียนโปรแกรมลงบนไมโครคอนโทรลเลอร์ หรือแสดงว่าเกิดการผิดพลาด(สีแดง) error ในตัว PICKit3

2.4 สายเชื่อมต่อเพื่อสื่อสารระหว่างไมโครคอนโทรลเลอร์และคอมพิวเตอร์

2.4.1 RS-232 Recommended Standard-232

เป็นมาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรม กำหนดโดย EIA (Electronics Industry Association) หรือ สมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกา ใช้กับการสื่อสารแบบจุดต่อจุด โดยใช้สายเชื่อมต่อ DB แบบ 25 และ 9 เจ็ม ที่ไม่ประสานจังหวะระหว่างคอมพิวเตอร์กับอุปกรณ์ต่อพ่วง มีการทำงานแบบสองทางพร้อมกัน (Full-duplex) โดยอาจใช้สายสัญญาณอื่นร่วมเพื่อทำแฮนด์เชก (Hand-shake) หรือไม่ได้กะ ทั้งนี้มาตรฐาน RS-232 จำกัดความยาวสายไว้ที่ 50 ฟุต (หรือประมาณ 15 เมตร) สำหรับการส่งสัญญาณที่ความเร็ว 19,200 บิตต่อวินาที โดยที่ความยาวสายจะต้องสั้นลงถ้าต้องการสื่อสารที่ความเร็วสูงขึ้น เรานำสายตัวนี้มาเชื่อมระหว่างบอร์ด ARDUINO กับ คอมพิวเตอร์ และยังใช้เพื่อสื่อสารส่งข้อมูลจากไมโครคอนโทรลเลอร์ไปยังคอมพิวเตอร์อีกด้วย



รูปที่ 2.58 สายRS-232

2.4.2 USB (Universal Serial Bus)

เป็นข้อกำหนดมาตรฐานของบัสการสื่อสารแบบอนุกรม เพื่อใช้ในการเชื่อมต่อกับอุปกรณ์ ซึ่งโดยทั่วไปจะใช้กับคอมพิวเตอร์ แต่สามารถใช้ได้กับอุปกรณ์อื่น เรานำสายชนิดนี้มาใช้เชื่อมต่อระหว่างPICKIT3และคอมพิวเตอร์เพื่อให้คอมพิวเตอร์เขียนข้อมูลลงบนไมโครคอนโทรลเลอร์ โดยผ่านสายUSB เข้าPICKIT3 แล้วจึงเข้าสู่ไมโครคอนโทรลเลอร์ต่อไป



รูปที่ 2.59 สายUSB

2.5 ไมโครคอนโทรลเลอร์และอุปกรณ์เสริมที่ใช้ในการพัฒนาผลิตภัณฑ์

2.5.1 ไมโครคอนโทรลเลอร์ที่ใช้ในการพัฒนาผลิตภัณฑ์

2.5.1.1 PIC16f1516

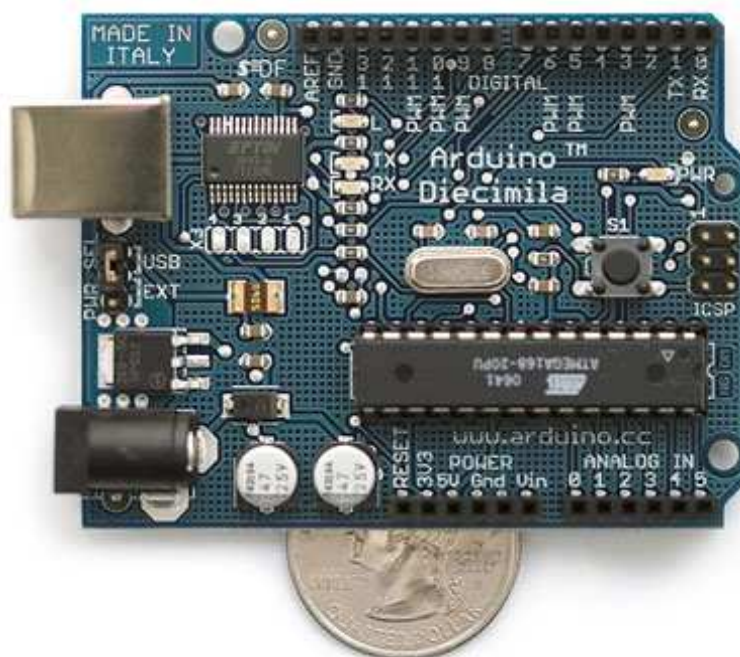
เป็นไมโครคอนโทรลเลอร์ของบริษัทไมโครชิพ ที่มีคุณสมบัติในส่วนของ capacitive sensing อยู่ในตัวเองจึงสามารถนำมาพัฒนาภายใต้คอนเซ็ปต์ระบบสัมผัสได้ จึงเลือกที่จะนำมาเป็นตัวที่จะทำขึ้นมาเป็นตัวจริง



รูปที่ 2.60 PIC16F1516

2.5.1.2 ARDUINO ATMEGA 328

เป็นไมโครคอนโทรลเลอร์ที่เป็นโอเพ่นซอร์สทางฮาร์ดแวร์ จึงเป็นที่นิยมทำให้มีนักพัฒนาอุปกรณ์เสริมสำหรับบอร์ดตัวนี้มาก ทางผู้จัดทำจึงได้ใช้ นำมาเป็นตัวทดลองที่จะใส่ฟังก์ชันการใช้งานและลูกเล่นลงไปก่อนที่จะนำไปใส่ลงในอุปกรณ์ที่จะใช้ทำขึ้นมาจริงๆ

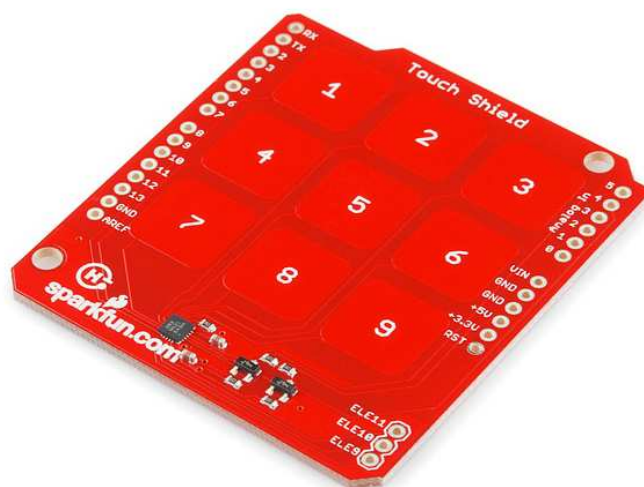


รูปที่ 2.61 ARDUINO ATMEGA 328

2.5.2 อุปกรณ์เสริมที่ช่วยในการพัฒนาผลิตภัณฑ์

2.5.2.1 Capacitive touch shield

จากบริษัท SPARKFUN เป็นอุปกรณ์เสริมสำหรับ ARDUINO ATMEGA 328 ที่ทำหน้าที่เป็นแป้นตัวเลขโดยใช้เทคโนโลยี capacitive sensing



รูปที่ 2.62 Capacitive touch shield

2.6 ภาษาที่ใช้ในการเขียนโปรแกรมเพื่อพัฒนาผลิตภัณฑ์

2.6.1 C language Programming

ภาษาซี (C) เป็นภาษาโปรแกรมบนคอมพิวเตอร์ที่มีวัตถุประสงค์ทั่วไป พัฒนาขึ้นเมื่อ พ.ศ. 2515 (ค.ศ. 1972) โดย เดนนิส ริตชี ที่เบลล์เทเลโฟนแลบอราทอรีส์ (Bell Telephone Laboratories) เกิดขึ้นเพื่อสร้างระบบปฏิบัติการยูนิกซ์ในขณะนั้น

นอกจากภาษาซีออกแบบขึ้นมาเพื่อสร้างซอฟต์แวร์ระบบแล้ว^[4] ภาษาซียังสามารถใช้อย่างแพร่หลายเพื่อพัฒนาซอฟต์แวร์ประยุกต์ที่เคลื่อนย้าย (portable) ไปบนระบบอื่นได้อีกด้วย

ภาษาซีเป็นภาษาโปรแกรมหนึ่งที่มีความนิยมมากที่สุดตลอดกาล^{[5][6]} มีสถาปัตยกรรมคอมพิวเตอร์เพียงส่วนน้อยเท่านั้นที่ไม่มีตัวแปลโปรแกรมของภาษาซี ภาษาซีมีอิทธิพลอย่างมากต่อภาษาโปรแกรมที่นิยมอื่น ๆ ที่เด่นชัดที่สุดก็คือภาษาซีพลัสพลัส ซึ่งเดิมเป็นส่วนขยายของภาษาซี

การออกแบบ

ภาษาซีเป็นภาษาเขียนโปรแกรมระบบเชิงคำสั่ง (หรือเชิงกระบวนการ) ถูกออกแบบขึ้นเพื่อใช้แปลด้วยตัวแปลโปรแกรมแบบการเชื่อมโยงที่ตรงไปตรงมา สามารถเข้าถึงหน่วยความจำในระดับล่าง เพื่อสร้างภาษาที่จับคู่อย่างมีประสิทธิภาพกับชุดคำสั่งเครื่อง และแทบไม่ต้องการ

สนับสนุนใด ๆ ขณะทำงาน ภาษาซีจึงเป็นประโยชน์สำหรับหลายโปรแกรมที่ก่อนหน้านี้เคยเขียนในภาษาแอสเซมบลีมาก่อน

หากไม่คำนึงถึงความสามารถในระดับล่าง ภาษานี้ถูกออกแบบขึ้นเพื่อส่งเสริมการเขียนโปรแกรมที่ไม่ขึ้นอยู่กับเครื่องใดเครื่องหนึ่ง (machine-independent) โปรแกรมภาษาซีที่เขียนขึ้นตามมาตรฐานและเคลื่อนย้ายได้ สามารถแปลได้บนแพลตฟอร์มคอมพิวเตอร์และระบบปฏิบัติการต่าง ๆ อย่างกว้างขวาง โดยแก้ไขรหัสต้นฉบับเพียงเล็กน้อยหรือไม่ต้องแก้ไขเลย ภาษานี้สามารถใช้ได้บนแพลตฟอร์มได้หลากหลายตั้งแต่ไมโครคอนโทรลเลอร์ฝังตัวไปจนถึงซูเปอร์คอมพิวเตอร์

แนวคิดการทำให้เล็กที่สุด

การออกแบบของภาษาซีถูกผูกมัดอยู่กับจุดประสงค์การใช้คือเป็นภาษาเขียนโปรแกรมระบบที่เคลื่อนย้ายได้ ภาษาซีจึงจัดเตรียมการเข้าถึงวัตถุใด ๆ ที่สามารถระบุตำแหน่งได้โดยตรงและง่ายดาย (ตัวอย่างเช่น รีจิสเตอร์ควบคุมอุปกรณ์ซึ่งจับคู่อยู่กับหน่วยความจำ) และนิพจน์ในรหัสต้นฉบับสามารถแปลงอย่างตรงไปตรงมาตามพฤติกรรมเป็นคำสั่งเครื่องดั้งเดิมที่ทำงานได้ ตัวแปลภาษาซีรุ่นก่อนบางตัวทำงานได้บนหน่วยประมวลผลพีซี-11 ซึ่งมีบิตอ้างตำแหน่งเพียง 16 บิตได้อย่างสบาย ตัวแปลภาษาซีเช่นแอซเท็กซี (Aztec C) สำหรับแพลตฟอร์ม 8 บิตทั่วไปก็สามารถทำงานได้เช่นกัน

ลักษณะเฉพาะ

ภาษาซีมีสิ่งอำนวยความสะดวกสำหรับการเขียนโปรแกรมเชิงโครงสร้าง และสามารถกำหนดขอบข่ายตัวแปรและเรียกซ้ำ เช่นเดียวกับภาษาโปรแกรมเชิงคำสั่งส่วนใหญ่ในสายตระกูลภาษาอัลกอล ในขณะที่ระบบชนิดตัวแปรแบบอพลวัตช่วยป้องกันการดำเนินการที่ไม่ได้ตั้งใจ รหัสที่ทำงานได้ทั้งหมดในภาษาซีถูกบรรจุอยู่ในฟังก์ชัน พารามิเตอร์ของฟังก์ชันส่งผ่านด้วยค่าของตัวแปรเสมอ ส่วนการส่งผ่านด้วยการอ้างอิงจะถูกจำลองขึ้นโดยการส่งผ่านค่าตัวชี้ ชนิดข้อมูลรวมแบบแตกต่าง (struct) ช่วยให้ผู้รวมข้อมูลที่เกี่ยวข้องกันสามารถรวมกันและจัดการได้ในหน่วยเดียว รหัสต้นฉบับของภาษาซีเป็นรูปแบบอิสระ ซึ่งใช้อักขระ (;) เป็นตัวจบคำสั่ง (มิใช่ตัวแบ่ง)

ภาษาซียังมีลักษณะเฉพาะต่อไปนี้เพิ่มเติม

- ตัวแปรอาจถูกซ่อนในบล็อกซ้อนใน
- ชนิดตัวแปรไม่เคร่งครัด เช่นข้อมูลตัวอักษรสามารถใช้เป็นจำนวนเต็ม

- เข้าถึงหน่วยความจำคอมพิวเตอร์ในระดับต่ำโดยแปลงที่อยู่ลงในเครื่องด้วยชนิดตัวแปรตัวชี้ (pointer)
- ฟังก์ชันและตัวชี้ข้อมูลรองรับการทำงานในภาวะหลายรูปแบบ (polymorphism)
- การกำหนดดัชนีแถวลำดับสามารถทำได้ด้วยวิธีรอง คือนิยามในพจน์ของเลขคณิตของตัวชี้
- ตัวประมวลผลก่อนสำหรับการนิยามแมโคร การรวมไฟล์รหัสต้นฉบับ และการแปลโปรแกรมแบบมีเงื่อนไข
- ความสามารถที่ซับซ้อนเช่น ไอ/โอ การจัดการสายอักขระ และฟังก์ชันทางคณิตศาสตร์รวมอยู่ในไลบรารี
- คำหลักที่สงวนไว้มีจำนวนค่อนข้างน้อย
- ตัวดำเนินการแบบประสมจำนวนมาก อาทิ +=, -=, *=, ++ ฯลฯ

โครงสร้างการเขียน คล้ายภาษาบีมากกว่าภาษาอัลกอล ตัวอย่างเช่น

- ใช้วงเล็บปีกกา { ... } แทนที่จะเป็น begin ... end ในภาษาอัลกอล 60 หรือวงเล็บโค้ง (...) ในภาษาอัลกอล 68
- เท่ากับ = ใช้สำหรับกำหนดค่า (คัดลอกข้อมูล) เหมือนภาษาฟอร์แทรน แทนที่จะเป็น := ในภาษาอัลกอล
- เท่ากับสองตัว == ใช้สำหรับเปรียบเทียบความเท่ากัน แทนที่จะเป็น .EQ. ในภาษาฟอร์แทรนหรือ = ในภาษาเบสิกและภาษาอัลกอล
- ตรรกะ "และ" กับ "หรือ" แทนด้วย && กับ || ตามลำดับ แทนที่จะเป็นตัวดำเนินการ $\wedge$ กับ $\vee$ ในภาษาอัลกอล แต่ตัวดำเนินการดังกล่าวจะไม่ประเมินค่าตัวถูกดำเนินการทางขวา ถ้าหากผลลัพธ์จากทางซ้ายสามารถพิจารณาได้แล้ว เหตุการณ์เช่นนี้เรียกว่าการประเมินค่าแบบลัดวงจร (short-circuit evaluation) และตัวดำเนินการดังกล่าวก็มีความหมายต่างจากตัวดำเนินการระดับบิต & กับ |

คุณลักษณะที่ขาดไป

ธรรมชาติของภาษาในระดับต่ำช่วยให้โปรแกรมเมอร์ควบคุมสิ่งที่คอมพิวเตอร์กระทำได้อย่างใกล้ชิด ในขณะที่อนุญาตให้มีการปรับแต่งพิเศษและการทำให้เหมาะสมที่สุดสำหรับแพลตฟอร์มหนึ่งใดโดยเฉพาะ สิ่งนี้ทำให้รหัสสามารถทำงานได้อย่างมีประสิทธิภาพบนฮาร์ดแวร์ที่มีทรัพยากรจำกัดมาก ๆ ได้เช่นระบบฝังตัว

ภาษาซีไม่มีคุณลักษณะบางอย่างที่มีในภาษาอื่น อาทิ

- ไม่มีการนิยามฟังก์ชันซ้อนใน
- ไม่มีการกำหนดค่าแฉวลำดับหรือสายอักขระโดยตรง (การคัดลอกข้อมูลจะกระทำผ่านฟังก์ชันมาตรฐาน แต่การรับการกำหนดค่าวัตถุที่มีชนิดเป็น struct หรือ union)
- ไม่มีการเก็บข้อมูลขยะโดยอัตโนมัติ
- ไม่มีข้อกำหนดเพื่อการตรวจสอบขอบเขตของแฉวลำดับ
- ไม่มีการดำเนินการสำหรับแฉวลำดับทั้งหมดในระดับตัวภาษา
- ไม่มีวากยสัมพันธ์สำหรับช่วงค่า (range) เช่น A..B ที่ใช้ในบางภาษา
- ก่อนถึงภาษาซี99 ไม่มีการแบ่งแยกชนิดข้อมูลแบบบูล (ค่าศูนย์หรือไม่ศูนย์ถูกนำมาใช้แทน) ^[7]
- ไม่มีส่วนปิดคลุมแบบรูปนัย (closure) หรือฟังก์ชันในรูปแบบพารามิเตอร์ (มีเพียงตัวชี้ของฟังก์ชันและตัวแปร)
- ไม่มีตัวสร้างและโครูทีน การควบคุมกระแสการทำงานภายในเทรคมีเพียงการเรียกใช้ฟังก์ชันซ้อนลงไป เว้นแต่การใช้ฟังก์ชัน longjmp หรือ setcontext จากไลบรารี
- ไม่มีการจัดการสิ่งผิดปกติ (exception handling) ฟังก์ชันไลบรารีมาตรฐานจะแสดงเงื่อนไขข้อผิดพลาดด้วยตัวแปรส่วนกลาง errno และ/หรือค่ากลับคืนพิเศษ และฟังก์ชันไลบรารีได้เตรียม goto แบบไม่ใช่เฉพาะที่ไว้ด้วย
- การเขียนโปรแกรมเชิงมอดูลรองรับแค่ระดับพื้นฐานเท่านั้น
- การโอเวอร์โหลดฟังก์ชันหรือตัวดำเนินการไม่รองรับภาวะหลายรูปแบบขณะแปลโปรแกรม
- การเขียนโปรแกรมเชิงวัตถุรองรับในระดับที่จำกัดมาก โดยพิจารณาจากภาวะหลายรูปแบบกับการรับทอด (inheritance)
- การซ่อนสารสนเทศ (encapsulation) รองรับในระดับที่จำกัด
- ไม่รองรับโดยพื้นฐานกับการทำงานแบบมัลติเทรคและเครือข่ายคอมพิวเตอร์
- ไม่มีไลบรารีมาตรฐานสำหรับคอมพิวเตอร์กราฟิกส์และความจำเป็นหลายอย่างในการเขียนโปรแกรมประยุกต์

คุณลักษณะเหล่านี้จำนวนหนึ่งมิให้ใช้ได้จากส่วนขยายในตัวแปลโปรแกรมบางตัว หรือจัดสรรไว้แล้วในสภาพแวดล้อมของระบบปฏิบัติการ (เช่น โพสซิกซ์) หรือจัดเตรียมโดยไลบรารี

ภายนอก หรือสามารถจำลองโดยดัดแปลงแก้ไขรหัสที่มีอยู่ หรือบางครั้งก็ถูกพิจารณาว่าไม่ใช่รูปแบบการเขียนโปรแกรมที่เหมาะสม

พฤติกรรมไม่นิยาม

การดำเนินการหลายอย่างในภาษาซีมีพฤติกรรมไม่นิยามซึ่งไม่ถูกกำหนดว่าต้องตรวจสอบขณะแปลโปรแกรม ในกรณีของภาษาซี "พฤติกรรมไม่นิยาม" หมายถึงพฤติกรรมเฉพาะอย่างที่เกิดขึ้นโดยมาตรฐานมิได้ระบุไว้ และสิ่งที่จะเกิดขึ้นก็ไม่มีในเอกสารการใช้งานของภาษาซี หนึ่งในชุดคำสั่งที่มีชื่อเสียงและน่าขบขันจากกลุ่มข่าว comp.std.c และ comp.lang.c นั้นทำให้โปรแกรมเกิดปัญหาที่เรียกว่า "ปีศาจที่ออกมาจากจมูกของคุณ" (demons to fly out of your nose)^[8] บางครั้งสิ่งที่เกิดขึ้นในทางปฏิบัติอันเป็นผลมาจากพฤติกรรมไม่นิยามทำให้เกิดจุดบกพร่องที่ยากต่อการตรวจสอบและอาจทำให้ข้อมูลในหน่วยความจำผิดแปลกไป ตัวแปลโปรแกรมบางชนิดช่วยสร้างการดำเนินงานที่ทำให้พฤติกรรมนั้นดีขึ้นและมีเหตุผล ซึ่งแตกต่างจากการแปลโดยตัวแปลชนิดอื่นที่อาจดำเนินงานไม่เหมือนกัน สาเหตุที่พฤติกรรมบางอย่างยังคงไว้ว่าไม่นิยามก็เพื่อให้ตัวแปลโปรแกรมบนสถาปัตยกรรมชุดของคำสั่งเครื่องที่หลากหลาย สามารถสร้างรหัสที่ทำงานได้ในพฤติกรรมที่นิยามอย่างมีประสิทธิภาพมากขึ้น ซึ่งเชื่อว่าเป็นบทบาทหนึ่งที่สำคัญของภาษาซีในฐานะภาษาสำหรับสร้างระบบ ดังนั้นภาษาซีจึงส่งผลให้เกิดความรับผิดชอบของโปรแกรมเมอร์เพื่อหลีกเลี่ยงพฤติกรรมไม่นิยาม โดยอาจใช้[[# 1/8

วากยสัมพันธ์

รหัสต้นฉบับของภาษาซีมีรูปแบบอิสระ ซึ่งสามารถใช้อักขระช่องว่างเท่าใดก็ได้ในรหัสมากกว่าที่จะถูกจำกัดด้วยคอลัมน์หรือบรรทัดข้อความอย่างภาษาฟอร์แทรน 77 ข้อความหมายเหตุจะปรากฏระหว่างตัวกัน /* และ */ (แบบดั้งเดิม) หรือตามหลัง // จนกว่าจะจบบรรทัด (ภาษาซี99 เป็นต้นไป)

รหัสต้นฉบับแต่ละไฟล์ประกอบด้วยการประกาศและการนิยามฟังก์ชันต่าง ๆ และการนิยามฟังก์ชันก็ประกอบด้วยการประกาศและข้อความสั่งต่าง ๆ ภายในอีกด้วย การประกาศอาจกำหนดชนิดข้อมูลใหม่โดยใช้คำหลักเช่น struct, union และ enum หรือกำหนดค่าของชนิดข้อมูลและอาจสงวนเนื้อที่สำรองให้กับตัวแปรใหม่ โดยการเขียนชื่อของชนิดข้อมูลตามด้วยชื่อตัวแปร คำหลักอาทิ char และ int เป็นชนิดข้อมูลพื้นฐานที่มากับภาษา ส่วนต่าง ๆ ของรหัสถูกคลุมด้วย

วงเล็บปีกกา { กับ } เพื่อจำกัดขอบเขตของการประกาศ และเพื่อกระทำเสมือนข้อความสั่งเดียว สำหรับโครงสร้างการควบคุม

ภาษาซีใช้ *ข้อความสั่ง* (statement) ในการระบุการกระทำเช่นเดียวกับภาษาเชิงคำสั่งอื่น ข้อความสั่งที่สามัญที่สุดคือ *ข้อความสั่งนิพจน์* (expression statement) ซึ่งประกอบด้วยนิพจน์ที่จะถูกนำไปประเมินค่า ตามด้วยอัฒภาค ; จากผลข้างเคียงของการประเมินค่า ฟังก์ชันหลายฟังก์ชันอาจถูกเรียกใช้และตัวแปรหลายตัวอาจถูกกำหนดค่าใหม่ ภาษาซีได้เตรียมข้อความสั่งสำหรับควบคุมการไหลของโปรแกรมไว้หลายข้อความซึ่งดูได้จากคำสงวนต่าง ๆ ตัวอย่างเช่น การใช้ if-else เพื่อการทำงานแบบมีเงื่อนไข และการใช้ do-while, while และ for เพื่อการทำงานแบบวนรอบ เพื่อปรับเปลี่ยนการทำงานอันเป็นลำดับปกติ เป็นสิ่งที่รองรับสำหรับการเขียนโปรแกรมเชิงโครงสร้างสำหรับข้อความสั่ง for นั้นมีนิพจน์ของการกำหนดค่าเริ่มต้น การทดสอบเงื่อนไข และการกำหนดค่ารอบใหม่ทั้งสามอย่างในตัวเอง ซึ่งสามารถละเว้นนิพจน์ใดก็ได้ ข้อความสั่ง break และ continue สามารถใช้ภายในการทำงานแบบวนรอบ เพื่อหยุดการวนรอบ หรือข้ามไปยังการกำหนดค่ารอบใหม่ทันทีตามลำดับ นอกจากนี้ยังมีข้อความสั่งที่ไม่เป็นเชิงโครงสร้างคือ goto ซึ่งจะทำให้การไหลของโปรแกรมข้ามไปยังป้าย (label) ที่ตั้งชื่อไว้ทันทีภายในฟังก์ชัน ข้อความสั่ง switch และ case ใช้สำหรับพิจารณาทางเลือกของการทำงานโดยพิจารณาจากนิพจน์ที่เป็นจำนวนเต็ม

นิพจน์ต่าง ๆ สามารถใช้ตัวดำเนินการที่มีมากับภาษาได้หลากหลาย (ดูด้านล่าง) และอาจมีการเรียกใช้ฟังก์ชัน อาร์กิวเมนต์ของฟังก์ชันและตัวถูกดำเนินการของตัวดำเนินการส่วนใหญ่ที่จะถูกประเมินค่านั้นไม่มีการระบุลำดับ การประเมินค่าจึงอาจแทรกซ้อนกันก็ได้ อย่างไรก็ตาม ผลกระทบที่เกิดขึ้นทั้งหมด (รวมทั้งที่เก็บข้อมูลตัวแปร) จะปรากฏก่อน *จุดลำดับ* (sequence point) ถัดไป จุดลำดับนั้นคือจุดสิ้นสุดของข้อความสั่งของแต่ละนิพจน์ และจุดที่เข้าและออกจากการเรียกใช้ฟังก์ชัน จุดลำดับก็ยังเกิดขึ้นระหว่างการประเมินค่านิพจน์ที่มีตัวดำเนินการบางชนิด (เช่น &&, ||, ?: และตัวดำเนินการจุดภาค) สิ่งนี้ทำให้การปรับแต่งรหัสจุดหมายให้เหมาะสมทำได้ในระดับสูง ซึ่งไม่จำเป็นต้องให้โปรแกรมเมอร์ภาษาซีใส่ใจมากนักเพื่อให้ได้ผลลัพธ์ที่เชื่อถือได้ ในขณะที่จำเป็นสำหรับภาษาโปรแกรมอื่น

ถึงแม้ว่าวากยสัมพันธ์ของภาษาซีจะถูกเลียนแบบโดยภาษาอื่นหลายภาษาเพราะว่าความเคยชินอย่างกว้างขวาง แต่ก็ถูกวิพากษ์วิจารณ์บ่อยครั้ง ตัวอย่างเช่น เคอร์นิกันและริตชีได้กล่าวในบทนำของ *เดอะซีโปรแกรมมิงแลงกวิจ* ไว้ว่า "ภาษาซีก็มีคำนิยามของมันเหมือนภาษาอื่นใด ตัวดำเนินการบางตัวมีสิทธิการมาก่อนที่ผิด วากยสัมพันธ์บางส่วนสามารถทำให้ดีกว่านี้"

ปัญหาเฉพาะบางอย่างที่ควรหมายเหตุไว้มีดังนี้

- ไม่มีการตรวจสอบจำนวนและชนิดของอาร์กิวเมนต์ เมื่อการประกาศฟังก์ชันมีรายการพารามิเตอร์ว่าง (สิ่งนี้เพื่อความเข้ากันได้ย้อนหลังกับภาษาเคแอลดีอาร์ซี ซึ่งไม่มีโพโทไทป์)
- ทางเลือกที่น่าสงสัยของสิทธิการมาก่อนของตัวดำเนินการ ดังที่กล่าวถึงโดยเคอร์นิกันและริตชีข้างต้น เช่น $=$ ที่วางอยู่ติดกับ $\&$ และ $|$ ในนิพจน์ดังตัวอย่าง $x \& 1 == 0$ ตัวดำเนินการ $=$ จะทำก่อนซึ่งไม่ใช่ผลที่คาดไว้ จำเป็นต้องใส่วงเล็บเพิ่ม $(x \& 1) == 0$ เพื่อให้ $\&$ ทำก่อนตามต้องการ
- ตัวดำเนินการ $=$ ซึ่งใช้แสดงภาวะเท่ากันในคณิตศาสตร์ แต่ในภาษาซีใช้เพื่อกำหนดค่าของตัวแปร โดยใช้ตามแบบที่มีอยู่ก่อนในภาษาฟอร์แทรน ภาษาพีแอล/วัน และภาษาเบสิก ไม่เหมือนภาษาอัลกอลและภาษาต่อของของมัน ริตชีตั้งใจเลือกรูปแบบนี้ด้วยเหตุผลหลักว่า อาร์กิวเมนต์ของการกำหนดค่าเกิดขึ้นบ่อยกว่าการเปรียบเทียบ
- ความคล้ายกันของตัวดำเนินการกำหนดค่าและการเปรียบเทียบภาวะเท่ากัน ($=$ และ $==$) ทำให้เกิดความผิดพลาดจากการใช้เครื่องหมายผิดได้ง่าย ในหลายกรณีเครื่องหมายถูกใช้ในบริบทของอีกอันหนึ่งโดยไม่มีความผิดพลาดขณะแปล (แม้ว่าตัวแปลโปรแกรมปกติจะสร้างข้อความเตือนขึ้นมา) ตัวอย่างเช่น นิพจน์เงื่อนไขภายใน `if (a = b)` จะเป็นจริงถ้า `a` มีค่าไม่เป็นศูนย์หลังจากการกำหนดค่า^[12] อย่างไรก็ตาม ข้อบกพร่องนี้อาจมีประโยชน์สำหรับการเขียนรหัสอย่างย่อในบางกรณี
- การขาดตัวดำเนินการเติมกลางสำหรับวัตถุซับซ้อนหลายชนิด โดยเฉพาะการดำเนินการสายอักขระ ทำให้โปรแกรมที่ขึ้นอยู่กับการดำเนินการเหล่านี้มีขนาดใหญ่กว่าที่ควรเป็น (เพราะต้องสร้างฟังก์ชันขึ้นเอง) และทำให้รหัสอ่านยากขึ้นด้วย
- รูปแบบของการประกาศที่บางครั้งไม่เป็นไปตามสามัญสำนึก โดยเฉพาะตัวชี้ฟังก์ชัน (แนวคิดของริตชีคือการประกาศตัวระบุในบริบทที่สัมพันธ์กับการใช้งานของมัน)

ตัวดำเนินการ

ภาษาซีรองรับตัวดำเนินการหลายประเภท ซึ่งเป็นสัญลักษณ์ที่ใช้ในนิพจน์เพื่อระบุการจัดการที่จะถูกทำให้เกิดผล ระหว่างการประเมินค่าของนิพจน์นั้น ภาษาซีมีตัวดำเนินการต่อไปนี้

- พิชคณิต (+, -, *, /, %)
- การกำหนดค่า (=)

- การกำหนดค่าแต่งเติม (+, -, *, /, %, &, |, ^, <<=, >>=)
- ตรรกะระดับบิต (~, &, |, ^)
- การเลื่อนระดับบิต (<<, >>)
- ตรรกะแบบบูล (!, &&, ||)
- การประเมินค่าเชิงเงื่อนไข (?:)
- การทดสอบภาวะเท่ากัน (==, !=)
- การรวมอาร์กิวเมนต์ฟังก์ชัน (())
- การเพิ่มค่าและการลดค่า (++ , --)
- การเลือกสมาชิกในวัตถุ (., ->)
- ขนาดของวัตถุ (sizeof)
- ความสัมพันธ์เชิงอันดับ (<, <=, >, >=)
- การอ้างอิงและการถูกอ้างอิง (&, *, [D])
- การลำดับ (,)
- การจัดกลุ่มนิพจน์ย่อย (())
- การแปลงชนิดข้อมูล (())

ภาษาซีมีไวยากรณ์รูปนัยซึ่งระบุโดยมาตรฐานภาษาซี

การแปลงจำนวนเต็ม จำนวนจุดลอยตัว และการปิดเศษ

วากยสัมพันธ์ของการแปลงชนิดข้อมูลสามารถใช้แปลงค่าต่าง ๆ ระหว่างชนิดข้อมูลจำนวนเต็มและจำนวนจุดลอยตัว (จำนวนทศนิยม) หรือระหว่างจำนวนเต็มสองจำนวน หรือระหว่างจำนวนจุดลอยตัวสองจำนวนที่มีขนาดแตกต่างกัน ตัวอย่างเช่น (long int)sqrt(1000.0), (double)(256*256) หรือ (float)sqrt(1000.0) เป็นต้น การแปลงชนิดข้อมูลเป็นภาวะปริยายในหลายบริบท อาทิ เมื่อกำหนดค่าให้กับตัวแปรหรือพารามิเตอร์ของฟังก์ชัน หรือเมื่อใช้จำนวนจุดลอยตัวเป็นดัชนีของเวกเตอร์ หรือในการดำเนินการทางเลขคณิตที่มีตัวถูกดำเนินการเป็นข้อมูลคนละชนิดกัน

การแปลงค่าระหว่างจำนวนเต็มและจำนวนจุดลอยตัวโดยทั่วไป จะเกิดการเปลี่ยนแปลงการเข้ารหัสระดับบิตไปยังขอบเขตที่เป็นไปได้เพื่อสงวนค่าจำนวนของตัวถูกดำเนินการนั้น ไม่เหมือนกับการแปลงชนิดข้อมูลกรณีอื่น (ซึ่งการเข้ารหัสระดับบิตของตัวถูกดำเนินการจะถูกตีความใหม่ตามชนิดเป้าหมายเพียงเท่านั้น) โดยเฉพาะอย่างยิ่ง การแปลงชนิดข้อมูลจากจำนวนเต็มไปเป็น

จำนวนจุดลอยตัวจะคงไว้ซึ่งค่าจำนวนได้อย่างถูกต้อง เว้นแต่ถ้าจำนวนบิตในชนิดเป้าหมายมีไม่เพียงพอ กรณีดังกล่าวจะทำให้บิตที่มีนัยสำคัญน้อยที่สุดสูญหายไป

ส่วนการแปลงชนิดข้อมูลจากจำนวนจุดลอยตัวไปเป็นจำนวนเต็มจะเกิดการตัดค่าหลังจุดทศนิยมอย่างหลีกเลี่ยงไม่ได้ (ค่าถูกปัดเศษเข้าหาศูนย์) สำหรับการปัดเศษชนิดอื่น ภาษซี99ได้ระบุไว้แล้วในฟังก์ชันดังนี้ (ใน `<math.h>`)

- `round()`: ปัดเศษไปยังจำนวนเต็มที่ใกล้ที่สุด
- `rint()`, `nearbyint()`: ปัดเศษตามทิศทางของจำนวนจุดลอยตัวปัจจุบัน
- `ceil()`: ค่าจำนวนเต็มน้อยสุดที่ไม่น้อยกว่าอาร์กิวเมนต์ (ปัดขึ้น) ดูเพิ่มที่ฟังก์ชันเพดาน
- `floor()`: ค่าจำนวนเต็มมากที่สุดที่ไม่มากกว่าอาร์กิวเมนต์ (ปัดลง) ดูเพิ่มที่ฟังก์ชันพื้น
- `trunc()`: ปัดเศษเข้าหาศูนย์ (เหมือนกับการแปลงชนิดข้อมูลเป็นจำนวนเต็ม)

ฟังก์ชันทั้งหมดนี้รับอาร์กิวเมนต์ `double` และคืนค่าเป็น `double` ซึ่งต่อจากนั้นก็อาจแปลงชนิดข้อมูลเป็นจำนวนเต็มอีกทีหากจำเป็น

การแปลงชนิดข้อมูลจาก `float` ไปเป็น `double` จะคงไว้ซึ่งค่าจำนวนได้อย่างถูกต้อง ในขณะที่การแปลงกลับ ค่าจะถูกปัดเศษซึ่งมักเป็นการปัดเศษเข้าหาศูนย์ เพื่อให้พอดีกับจำนวนบิตที่น้อยลง (เนื่องจาก `float` ก็มีช่วงเลขชี้กำลังที่น้อยกว่าด้วย การแปลงชนิดข้อมูลอาจให้ผลเป็นค่าอนันต์แทน) ตัวแปลโปรแกรมบางโปรแกรมจะแปลงค่าของ `float` ไปเป็น `double` โดยเบื้องหลังในบางบริบท เช่น พารามิเตอร์ของฟังก์ชันที่ประกาศเป็น `float` ตามความเป็นจริงอาจส่งค่าเป็น `double` ก็ได้

เครื่องมือที่ทำตามมาตรฐานจำนวนจุดลอยตัวของ IEEE เหตุการณ์การปัดเศษบางเหตุการณ์มีผลมาจากสถานะการปัดเศษปัจจุบัน (ได้แก่การปัดเศษเลขคู่ การปัดเศษขึ้น การปัดเศษลง และการปัดเศษเข้าหาศูนย์) ซึ่งอาจเรียกดูหรือตั้งค่าสถานะโดยใช้ฟังก์ชัน `fegetround()/fesetround()` ที่นิยามไว้ใน `<fenv.h>`

ตัวอย่างโปรแกรม "เฮลโลเวิลด์"

ตัวอย่างโปรแกรม "เฮลโลเวิลด์" ซึ่งปรากฏอยู่ในหนังสือ *เดอะซีโปรแกรมมิงแลงกวิจ* ที่พิมพ์ครั้งแรก กลายเป็นตัวแบบของโปรแกรมเกริ่นนำในตำราการเขียนโปรแกรมส่วนใหญ่หากไม่คำนึงถึงภาษาที่ใช้เขียน โปรแกรมดังกล่าวจะแสดงผล "hello, world" ทางอุปกรณ์ส่งออกมาตรฐาน ซึ่งมักจะเป็นเครื่องปลายทางหรือหน่วยแสดงผลจอภาพ

รหัสโปรแกรมรุ่นดั้งเดิมเป็นดังนี้

```
main()
{
    printf("hello, world\n");
}
```

และหลังจากการปรับเปลี่ยนรหัสให้เข้ากับมาตรฐาน รหัสจึงเป็นดังนี้^[14]

```
#include <stdio.h>
```

```
int main(void)
{
    printf("hello, world\n");
    return 0;
}
```

บรรทัดแรกของโปรแกรมเป็นคำสั่งชี้แนะตัวประมวลผลก่อน (preprocessing directive) แสดงไว้โดย `#include` ทำให้ตัวประมวลผลก่อน (อันเป็นเครื่องมืออย่างแรกที่พิจารณาห้สดับฉบับขณะแปล) นำเนื้อหาข้อความทั้งหมดของไฟล์ส่วนหัวมาตรฐาน `stdio.h` เข้ามาแทนที่บรรทัดนั้น ซึ่งไฟล์ดังกล่าวมีการประกาศฟังก์ชันสำหรับอุปกรณ์นำเข้าและส่งออกมาตรฐานอาทิ `printf` วงเล็บแหลมที่คลุมชื่อไฟล์ `stdio.h` (ซึ่งความจริงคือเครื่องหมายน้อยกว่า-มากกว่า) เป็นการแสดงว่า `stdio.h` ถูกกำหนดที่ตั้งโดยใช้กลยุทธการค้นหาที่ให้ความสำคัญต่อไฟล์ส่วนหัวมาตรฐาน มากกว่าไฟล์ส่วนหัวอื่นที่มีชื่อเดียวกัน อนุญาตอาจจะใช้ได้ในกรณีที่ต้องการนำไฟล์ส่วนหัวที่อยู่ใกล้เคียงหรือเจาะจงโครงการเข้ามารวม

บรรทัดถัดมาเป็นการนิยามฟังก์ชันชื่อว่า `main` ฟังก์ชัน `main` เป็นฟังก์ชันที่มีจุดประสงค์พิเศษในโปรแกรมภาษาซี สภาพแวดล้อมขณะทำงานจะเรียกใช้ฟังก์ชัน `main` เพื่อเริ่มต้นการทำงานของโปรแกรม ตัวระบุชนิด `int` เป็นตัวแสดงว่า ค่าส่งคืน ที่ถูกส่งคืนโดยตัวที่เรียกใช้ (กรณีนี้คือสภาพแวดล้อมขณะทำงาน) จะเป็นจำนวนเต็มค่าหนึ่ง อันเป็นผลจากการประเมินค่าของฟังก์ชัน `main` คำหลัก `void` ในรายการพารามิเตอร์แสดงว่าฟังก์ชัน `main` ไม่ต้องใช้อาร์กิวเมนต์

วงเล็บปีกกาเปิดหมายถึงจุดเริ่มต้นของการนิยามฟังก์ชัน main

บรรทัดถัดมาเป็นการ *เรียกใช้* ฟังก์ชันที่ชื่อว่า printf ซึ่งประกาศไว้ใน stdio.h และจัดเตรียมขึ้นจากไลบรารีของระบบ ในการเรียกใช้ครั้งนี้ ฟังก์ชัน printf จะถูก *ผ่านค่า* ด้วยอาร์กิวเมนต์หนึ่งตัวคือตำแหน่งหน่วยความจำของอักขระตัวแรกในสายอักขระ "hello, world\n" สายอักขระดังกล่าวคือแถวลำดับที่ไม่มีชื่ออันประกอบด้วยชนิดข้อมูล char จะถูกสร้างขึ้นโดยอัตโนมัติโดยตัวแปลโปรแกรม และแถวลำดับจะมีอักขระค่าศูนย์ (null) เป็นสิ่งที่บ่งบอกจุดสิ้นสุดของสายอักขระ (printf จำเป็นต้องทราบสิ่งนี้) \n ที่ปรากฏในสายอักขระคือ *ลำดับการหลีกเลี่ยง* (escape sequence) ภาษาซีจะตีความว่าเป็นอักขระ newline (ขึ้นบรรทัดใหม่) ซึ่งจะทำให้อุปกรณ์ส่งออกทราบว่าถึงจุดสิ้นสุดของบรรทัดปัจจุบัน ค่าส่งคืนจากฟังก์ชัน printf คือชนิด int แต่มันถูกละทิ้งไปอย่างเงียบ ๆ เนื่องจากไม่มีการใช้ (โปรแกรมที่ระมัดระวังมากกว่าอาจทดสอบค่าส่งคืน เพื่อพิจารณาว่าผลจากการทำงานของฟังก์ชัน printf สำเร็จหรือไม่) อัฒภาค ; เป็นจุดสิ้นสุดข้อความสั่ง

ข้อความสั่ง return เป็นการสิ้นสุดการทำงานของฟังก์ชัน main และทำให้ฟังก์ชันส่งกลับเป็นจำนวนเต็มค่า 0 ซึ่งสภาพแวดล้อมขณะทำงานจะตีความว่าเป็นรหัสออกจากโปรแกรมที่แสดงว่าการทำงานประสบผลสำเร็จ

วงเล็บปีกกาปิดหมายถึงจุดสิ้นสุดของการนิยามฟังก์ชัน main

ชนิดข้อมูล

ภาษาซีมีระบบชนิดตัวแปรแบบไม่เคร่งครัด ซึ่งมีความคล้ายคลึงบางประการร่วมกับภาษาลูกของภาษาอัลกอล อาทิ ภาษาปาสคาล ภาษาซีมีชนิดตัวแปรที่เตรียมไว้แล้วสำหรับจำนวนเต็มหลายขนาด แบบทั้งมีเครื่องหมายและไม่มีเครื่องหมาย จำนวนจุดลอยตัว ตัวอักขระ และชนิดข้อมูลแฉก (enum) ในภาษาซี99 ได้เพิ่มชนิดตัวแปรแบบบูลเข้าไปด้วย ภาษาซีก็ยังมีชนิดตัวแปรที่รับทอดมาด้วยเช่นแถวลำดับ ตัวชี้ ระเบียบ (struct) และยูเนียน (union)

ภาษาซีมักใช้กับการเขียนโปรแกรมระบบในระดับต่ำ ซึ่งอาจหลีกเลี่ยงการใช้ระบบชนิดตัวแปรเมื่อจำเป็น ตัวแปลโปรแกรมจะพยายามทำให้แน่ใจว่า ชนิดตัวแปรถูกใช้อย่างถูกต้องในนิพจน์ส่วนใหญ่ แต่โปรแกรมเมอร์ก็สามารถลบเลือนการตรวจสอบเช่นนั้นได้หลายทาง อาทิ การโยนชนิดข้อมูล (type cast) เพื่อแปลงค่าจากชนิดหนึ่งไปเป็นชนิดหนึ่งอย่างชัดเจน หรือการใช้ตัวชี้หรือยูเนียนเพื่อแปลความหมายบิตของค่าที่อยู่ภายในไปเป็นอีกชนิดหนึ่ง

ตัวชี้

ภาษาซีรองรับการใช้งานตัวชี้ (pointer) ซึ่งเป็นชนิดข้อมูลสำหรับการอ้างอิงอย่างง่ายชนิดหนึ่ง ที่เก็บบันทึกที่อยู่หรือตำแหน่งของวัตถุหรือฟังก์ชันในหน่วยความจำ ตัวชี้สามารถ อ้างอิงกลับ (dereference) เพื่อเข้าถึงข้อมูลที่บันทึกในตำแหน่งที่ถูกชี้อยู่ หรือเพื่อเรียกใช้ฟังก์ชันที่ถูกชี้อยู่ ตัวชี้สามารถจัดดำเนินการกำหนดค่าและเลขคณิตของตัวชี้ได้ด้วย ค่าของตัวชี้ขณะโปรแกรมทำงานมักจะเป็นตำแหน่งมูลฐานในหน่วยความจำ (ซึ่งอาจเสริมด้วยค่าออฟเซตในหน่วยเวิร์ด) แต่เนื่องจากตัวชี้มีการระบุชนิดตามข้อมูลที่ชี้ไป ตัวแปลโปรแกรมจึงสามารถตรวจสอบชนิดตัวแปรในนิพจน์ต่าง ๆ รวมทั้งตัวชี้ด้วยกันเองขณะแปลได้ เลขคณิตของตัวชี้จะแปรสัดส่วนของขนาดโดยอัตโนมัติตามชนิดข้อมูลที่ชี้ไป (ดูเพิ่มที่ส่วนความใช้แทนกันได้ระหว่างตัวชี้และแถวลำดับ) จุดประสงค์ของการใช้ตัวชี้มีหลากหลายในภาษาซีเช่น สายอักขระมักจัดดำเนินการโดยใช้ตัวชี้ไปยังแถวลำดับของตัวอักขระ การจัดสรรหน่วยความจำพลวัต (dynamic memory allocation) สามารถกระทำได้ด้วยตัวชี้ ชนิดข้อมูลชนิดอื่นเช่น ต้นไม้ ปกติจะถูกพัฒนาขึ้นโดยการจัดสรรวัตถุ struct โดยพลวัต ซึ่งเชื่อมโยงแต่ละหน่วยเข้ากันด้วยตัวชี้ ตัวชี้ของฟังก์ชันใช้เพื่อการเรียกกลับ (callback) สำหรับชุดคำสั่งจัดการเหตุการณ์ เป็นต้น

ตัวชี้ว่าง (null pointer) คือตัวชี้ที่ชี้ไปยังตำแหน่งที่ใช้งานไม่ได้ ซึ่งจะมีค่าเป็น 0^[16] การอ้างอิงกลับของตัวชี้ว่างจึงไม่มีความหมาย และโดยทั่วไปให้ผลเป็นข้อผิดพลาดขณะทำงาน อย่างไรก็ตามตัวชี้ว่างก็มีประโยชน์สำหรับกรณีพิเศษเช่น ใช้เป็นจุดสิ้นสุดหน่วยสุดท้ายของรายการโยง ซึ่งหมายความว่าไม่มีตัวชี้ไปหน่วยอื่นแล้ว หรือใช้แจ้งข้อผิดพลาดจากฟังก์ชันที่คืนค่าเป็นตัวชี้ ตัวชี้ว่างในการลงรหัสมักจะนำเสนอด้วย 0 หรือ NULL

ตัวชี้วอยด์ (void *) คือตัวชี้ของวัตถุที่ไม่ทราบชนิดตัวแปร ดังนั้นจึงสามารถใช้เป็นตัวชี้ "ทั่วไป" ก็ได้ แต่เนื่องจากขนาดและชนิดของวัตถุที่ถูกชี้ไม่เป็นที่ทราบ ตัวชี้วอยด์จึงไม่สามารถอ้างอิงกลับได้ และเลขคณิตของตัวชี้ก็ใช้กับตัวชี้วอยด์ไม่ได้ แม้ว่าตัวชี้ของวัตถุชนิดหนึ่งอาจแปลงเป็นตัวชี้ชนิดอื่นได้โดยง่าย (และในหลายบริบทก็แปลงได้อย่างคลุมเครือ)

การใช้งานตัวชี้อย่างไม่ระมัดระวังอาจเกิดอันตรายได้ เนื่องจากตัวแปรตัวชี้สามารถชี้ไปที่ตำแหน่งใดก็ได้โดยไม่มีกฎเกณฑ์ และปกติก็ไม่มีการตรวจสอบ ซึ่งอาจทำให้เกิดผลกระทบที่ไม่พึงปรารถนา ถึงแม้ตัวชี้ที่ใช้งานอย่างถูกต้องได้ชี้ไปยังตำแหน่งที่ปลอดภัยอยู่แล้ว แต่มันก็อาจถูกทำให้ชี้ไปยังตำแหน่งที่ไม่ปลอดภัยโดยการดำเนินการเลขคณิตที่ไม่ถูกต้อง หรือตัวชี้ไปยังวัตถุที่อาจเรียกคืนการจัดสรรไปแล้วแต่ถูกเรียกใช้ใหม่ (ตัวชี้อย่างหลวม dangling pointer) หรือตัวชี้ที่อาจใช้

งานโดยไม่กำหนดค่าเริ่มต้น (ตัวชี้ตัวแทน wild pointer) หรือตัวชี้ที่อาจถูกกำหนดด้วยค่าที่ไม่ปลอดภัยโดยตรง ด้วยวิธีโยนชนิดตัวแปร ยูเนียน หรือผ่านค่ามาจากตัวชี้อื่นที่เสีย เป็นต้น โดยทั่วไปภาษาซีอนุญาตให้จัดดำเนินการและแปลงชนิดตัวแปรของตัวชี้ได้ แม้ว่าตัวแปลโปรแกรมก็มีตัวเลือกสำหรับการตรวจสอบอยู่หลายระดับก็ตาม ภาษาโปรแกรมอื่นบางภาษาจัดการปัญหานี้โดยกำหนดให้ใช้ชนิดตัวแปรอ้างอิงที่เคร่งครัดมากกว่า

แถวลำดับ

ชนิดข้อมูลแถวลำดับ (array) ในภาษาซีแบบดั้งเดิมมีขนาดคงที่และสถิต ซึ่งจะถูกกำหนดตอนแปลโปรแกรม (ในเวลาถัดมา มาตรฐานภาษาซี99 อนุญาตให้สร้างแถวลำดับที่มีความยาวแปรได้) อย่างไรก็ตามแถวลำดับสามารถกำหนดให้จัดสรรเนื้อที่หน่วยความจำขนาดใดก็ได้ขณะทำงาน โดยใช้ฟังก์ชัน malloc จากไลบรารีมาตรฐาน แล้วทำให้เป็นแถวลำดับ การทำให้เป็นหนึ่งเดียวระหว่างแถวลำดับและตัวชี้ของภาษาซี ทำให้หมายความว่าแถวลำดับที่แท้จริงและแถวลำดับที่จัดสรรอย่างพลวัตเสมือนใช้แทนกันได้ เนื่องด้วยแถวลำดับเข้าถึงผ่านตัวชี้เสมอ (ในทางปฏิบัติ) การเข้าถึงแถวลำดับจึงไม่มีการตรวจสอบขนาดภายในแถวลำดับ แม้ว่าตัวแปลโปรแกรมอาจมีตัวเลือกสำหรับตรวจสอบขอบเขตก็ตาม การใช้งานเกินขอบเขตของแถวลำดับจึงยังคงสามารถเป็นไปได้ ซึ่งเกิดขึ้นค่อนข้างเป็นปกติในรหัสที่เขียนอย่างไม่ระมัดระวัง และนำไปสู่ผลสะท้อนกลับหลายอย่าง อาทิ การเข้าถึงหน่วยความจำที่ไม่อนุญาต การทำให้ข้อมูลผิดแปลกไป บัฟเฟอร์ส่วนเกิน และสิ่งผิดปรกติขณะทำงาน

ถึงแม้ภาษาซีรองรับแถวลำดับแบบสถิต แต่ก็ไม่จำเป็นว่าดัชนีของแถวลำดับจะต้องมีผล (การตรวจสอบขอบเขต) ตัวอย่างเช่น เราสามารถลบออกบันทึกค่าสมาชิกตัวที่ตกลงในแถวลำดับที่มีสมาชิกห้าตัวได้ ซึ่งจะทำให้เกิดผลที่ไม่คาดคิด ความผิดพลาดเช่นนี้เรียกว่า *บัฟเฟอร์ส่วนเกิน* (buffer overflow/overflow) เป็นสาเหตุที่สำคัญอย่างหนึ่งของปัญหาด้านความปลอดภัย เนื่องจากเทคโนโลยีการจัดการตรวจสอบขอบเขต (bounds-checking elimination) ไม่มีอยู่เลยเมื่อภาษาซีถูกนิยามขึ้น การตรวจสอบขอบเขตจึงลดทอนประสิทธิภาพอย่างรุนแรง โดยเฉพาะกับการคำนวณเชิงจำนวน เมื่อสองสามปีก่อนหน้านั้น ตัวแปลภาษาฟอร์แทรนมีตัวเลือกให้เปิดหรือปิดการตรวจสอบขอบเขตได้ แต่ตัวเลือกเช่นนี้ไม่มีประโยชน์ต่อภาษาซี เพราะอาร์กิวเมนต์ของแถวลำดับถูกผ่านค่าด้วยตัวชี้ธรรมดา

ภาษาซีไม่มีข้อกำหนดพิเศษสำหรับการประกาศแถวลำดับหลายมิติ แต่อาจจะขึ้นอยู่กับ การเรียกซ้ำภายในระบบชนิดตัวแปร เพื่อประกาศแถวลำดับของแถวลำดับ ซึ่งสามารถบรรลุผล

สำเร็จได้เหมือนกัน ค่าดัชนีของ "แถวลำดับหลายมิติ" ที่สร้างขึ้นสามารถพิจารณาว่าเพิ่มขึ้นตามอันดับเรียงตามแถว (row-major order)

โดยปกติแถวลำดับหลายมิติถูกใช้งานในขั้นตอนวิธีเชิงจำนวนเพื่อเก็บข้อมูลเมทริกซ์ (ซึ่งประยุกต์มาจากพีชคณิตเชิงเส้นเป็นหลัก) โครงสร้างของแถวลำดับในภาษาซีเหมาะสมเป็นอย่างดีสำหรับงานนี้ แต่เนื่องจากแถวลำดับถูกผ่านค่าด้วยตัวชี้ ขอบเขตของแถวลำดับจึงต้องเป็นค่าที่ทราบและตายตัว หรือไม่เช่นนั้นก็ต้องผ่านค่าไปพร้อมกับซับรูทีนที่จำเป็นต้องทราบ นอกจากนี้ แถวลำดับของแถวลำดับที่จัดสรรขนาดแบบพลวัต ไม่สามารถเข้าถึงได้โดยใช้ดัชนีสองชั้น (ตัวอย่างกรณีนี้ เช่นการจัดสรรแถวลำดับด้วย "เวกเตอร์แถว" ของตัวชี้ไปยังสมรรถ)

ภาษาซี99 ได้แนะนำ "แถวลำดับความยาวแปรได้" เพิ่มเข้ามา แต่ก็ยังมีปัญหาบางประการที่เหมือนกับปัญหาแถวลำดับของภาษาซี

ความใช้แทนกันได้ระหว่างตัวชี้และแถวลำดับ

คุณลักษณะเด่นชัดของภาษาซี (ซึ่งอาจทำให้สับสนด้วย) คือการปฏิบัติต่อแถวลำดับและตัวชี้ สัญกรณ์แถวลำดับ $x[i]$ สามารถใช้กับตัวชี้ x ได้ โดยแปลความหมายว่าเป็นการเข้าถึงวัตถุตัวที่ $i + 1$ ของวัตถุข้อมูลที่อยู่ติดกันถัดจากตำแหน่งที่ x ซอยู่ ซึ่งถือว่าเป็นสมาชิกตัวแรกของแถวลำดับ ($x[0]$)

$x[i]$ มีความหมายเทียบเท่า $*(x + i)$ ตามรูปแบบ และเนื่องจากชนิดตัวแปรของตัวชี้เป็นที่ทราบขณะแปล ตำแหน่ง $x + i$ ที่ชี้ไปไม่ได้หมายความว่าจากตำแหน่ง x แล้วเพิ่มไปอีก i ไบต์ แต่หมายถึงเพิ่มไปอีก i คูณด้วยขนาดของสมาชิกที่ตำแหน่ง x ขนาดของสมาชิกนี้ได้มาจากการใช้ตัวดำเนินการ `sizeof` บนสมาชิกที่อ้างอิงกลับตัวใดตัวหนึ่งของ x ดังเช่น $n = \text{sizeof } *x$ หรือ $n = \text{sizeof } x[0]$

นอกจากนี้ในบริบทส่วนใหญ่ของนิพจน์ ชื่อของแถวลำดับจะถูกแปลงเป็นตัวชี้ที่ชี้ไปยังสมาชิกตัวแรกของแถวลำดับนั้น สิ่งนี้ออกเป็นนัยว่าแถวลำดับจะไม่ถูกคัดลอกข้อมูลไปทั้งหมดเมื่อนำไปตั้งชื่ออาร์กิวเมนต์ของฟังก์ชัน แต่จะมีเพียงแค่ตำแหน่งของสมาชิกตัวแรกเท่านั้นที่ส่งผ่านไป ดังนั้นถึงแม้ว่าการเรียกใช้ฟังก์ชันในภาษาซีจะตีความว่าส่งโดยให้ค่า (pass-by-value) แต่แถวลำดับนั้นส่งโดยอ้างอิง (pass-by-reference) ในทางปฏิบัติ

จำนวนสมาชิกของแถวลำดับ x ที่ได้ประกาศไว้แล้ว สามารถคำนวณได้จาก $\text{sizeof } x / \text{sizeof } x[0]$

การสาธิตอย่างหนึ่งที่น่าสนใจต่อความใช้แทนกันได้ระหว่างตัวชี้และแวลวลำดับแสดงไว้ด้านล่าง
การกำหนดค่าทั้งสี่มีความหมายเทียบเท่ากันและเป็นรหัสที่ใช้งานได้ในภาษาซี

/ x เป็นแวลวลำดับหรือตัวชี้, i เป็นจำนวนเต็ม */*

x[i] = 1; / เทียบเท่ากับ *(x + i) */*

**(x + i) = 1;*

**(i + x) = 1;*

i[x] = 1; / เทียบเท่ากับ *(i + x) */*

แม้ว่าการกำหนดค่าทั้งสี่เทียบเท่ากัน แต่มีเพียงแบบแรกเท่านั้นที่แสดงรูปแบบการลงรหัสที่ดี กรณีอื่นอาจพบได้ในรหัสภาษาซีที่ยุ่งเหยิง

ถึงอย่างไรก็ตามแวลวลำดับและตัวชี้ก็ยังมีความแตกต่างแม้ว่ามันจะเทียบเท่ากัน ตัวชี้ไปยังสมาชิกตัวแรกซึ่งแปลงมาจากแวลวลำดับ ไม่มีเนื้อที่เก็บข้อมูลตำแหน่งของมันเอง ต่างจากตัวแปรตัวชี้ซึ่งมีเมื่อเป็นเช่นนั้นแล้วสิ่งที่แวลวลำดับ "ชี้ไป" จึงไม่สามารถเปลี่ยนแปลงได้ และไม่สามารถกำหนดค่าใหม่ให้กับตัวแปรแวลวลำดับ (ค่าต่าง ๆ ของแวลวลำดับอาจคัดลอกได้ โดยใช้ฟังก์ชัน memcpy เป็นต้น)

การจัดการหน่วยความจำ

ฟังก์ชันการทำงานหนึ่งที่สำคัญที่สุดของภาษาโปรแกรมคือ การให้บริการการจัดการหน่วยความจำและวัตถุที่บันทึกอยู่ในหน่วยความจำ ภาษาซีมีสามแนวทางที่ต่างกันเพื่อจัดสรรหน่วยความจำสำหรับวัตถุ

- การจัดสรรหน่วยความจำสถิต ที่ว่างสำหรับวัตถุในรหัสฐานสองซึ่งเว้นไว้ขณะแปลโปรแกรม วัตถุเหล่านี้มีอายุขัย (extent) ครบเท่าที่รหัสฐานสองที่มีวัตถุนั้นบรรจุลงในหน่วยความจำ
- การจัดสรรหน่วยความจำอัตโนมัติ วัตถุชั่วคราวสามารถเก็บบันทึกในกองซ้อน (stack) และที่ว่างนี้จะถูกเรียกคืนและใช้ใหม่หลังจากวัตถุที่ประกาศเลิกการทำงานโดยอัตโนมัติ
- การจัดสรรหน่วยความจำพลวัต บล็อกต่าง ๆ ของหน่วยความจำในขนาดที่ต้องการสามารถร้องขอได้ขณะทำงาน โดยใช้ฟังก์ชันไลบรารีอาทิ malloc ของเนื้อที่หน่วยความจำที่เรียกว่า

ฮีป (heap) บล็อกเหล่านี้คงอยู่จนกว่าจะถูกเรียกคืนเพื่อใช้ใหม่โดยใช้ฟังก์ชัน free ในภายหลัง

แนวทางสามอย่างนี้เหมาะสมในสถานการณ์และข้อแลกเปลี่ยนที่ต่างกันไป ตัวอย่างเช่น การจัดสรรหน่วยความจำสถิตไม่มีการดำเนินงานสิ้นเปลือง (overhead) เพื่อการจัดสรร การจัดสรรหน่วยความจำอัตโนมัติอาจมีการสิ้นเปลืองน้อย และการจัดสรรหน่วยความจำพลวัตอาจเป็นไปได้ว่ามีความสิ้นเปลืองอย่างมากทั้งการจัดสรรและการเรียกคืน ในทางตรงข้าม ที่ว่างในกองซ้อนโดยทั่วไปมีขนาดจำกัดและไม่คงทนถาวรไปกว่าหน่วยความจำแบบสถิตหรือที่ว่างในฮีป และการจัดสรรหน่วยความจำพลวัตสามารถจัดสรรวัตถุที่ทราบขนาดเฉพาะขณะทำงานได้ โปรแกรมภาษาซีส่วนใหญ่จึงใช้งานทั้งสามแนวทางอย่างกว้างขวาง

การจัดสรรมักให้ความสำคัญแก่แบบอัตโนมัติหรือแบบสถิตมากกว่า เพราะตัวแปลโปรแกรมเป็นส่วนจัดการหน่วยเก็บบันทึก ทำให้โปรแกรมเมอร์ไม่ต้องจัดสรรและเรียกคืนหน่วยเก็บบันทึก จุกจิกด้วยตนเองซึ่งเสี่ยงต่อการเกิดข้อผิดพลาด อย่างไรก็ตาม โครงสร้างข้อมูลหลายชนิดสามารถขยายขนาดได้ในขณะทำงาน และเนื่องจากการจัดสรรสถิต (และการจัดสรรอัตโนมัติในภาษาซี89 และซี90) จะต้องมีความตายตัวขณะแปลโปรแกรม หลายสถานการณ์จึงจำเป็นต้องใช้การจัดสรรพลวัต ก่อนที่จะมีมาตรฐานซี99 แถวลำดับความยาวแปรได้เป็นตัวอย่างปัญหาหนึ่งของกรณีนี้

วัตถุที่จัดสรรแบบอัตโนมัติและพลวัตจะถูกกำหนดค่าเริ่มต้นถ้าได้ระบุไว้ หรือมิฉะนั้นมันจะมีค่าที่ไม่แน่นอน (ไม่ว่ารูปแบบรหัสฐานสองบนหน่วยเก็บบันทึกจะเป็นอะไรก็ตาม ซึ่งอาจไม่เป็นค่าที่ใช้งานได้สำหรับชนิดตัวแปรนั้น) ถ้าโปรแกรมพยายามเข้าถึงค่าที่ไม่กำหนดนี้ ผลลัพธ์จะไม่สามารถนิยามได้ ตัวแปลโปรแกรมสมัยใหม่หลายโปรแกรมพยายามตรวจสอบและแจ้งเตือนเกี่ยวกับปัญหานี้ แต่ก็เกิดทั้งผลบวกและผลลบลง

ปัญหาอีกประการหนึ่งคือ การจัดสรรหน่วยความจำในฮีปจำเป็นต้องกระทำพร้อมกับการใช้งานจริงด้วยตนเองในโปรแกรมใด ๆ ก็ตาม เพื่อให้มันสามารถนำกลับมาใช้ใหม่มากที่สุดเท่าที่เป็นไปได้ ตัวอย่างเช่น ถ้ามีตัวชี้ไปยังฮีปที่ถูกจัดสรรนอกขอบเขต หรือค่าของตัวชี้ถูกเขียนทับก่อนเรียกใช้ free จะทำให้หน่วยความจำที่ตำแหน่งนั้นไม่สามารถเรียกคืนเพื่อใช้ใหม่ภายหลังและสูญเสียไปกับโปรแกรม อันเป็นปรากฏการณ์ที่เรียกว่า หน่วยความจำรั่ว (memory leak) ในทางกลับกัน การปลดปล่อยหน่วยความจำเร็วเกินไปแล้วยังคงใช้งานอยู่ซึ่งเป็นไปได้ แต่เนื่องจากระบบจัดสรรหน่วยความจำสามารถจัดสรรอีกครั้งหรือใช้หน่วยความจำที่ถูกทำให้ว่าง พฤติกรรมที่คาดเดาไม่ได้ก็อาจเกิดขึ้น โดยปกติอาการจะปรากฏในส่วน of โปรแกรมที่อยู่ไกลจากจุดที่ทำให้เกิด

ความผิดพลาดจริง ทำให้ตรวจแก้ปัญหาดูอย่างยากลำบาก ปัญหาเช่นนี้ได้รับการปรับปรุงแก้ไขในภาษาโปรแกรมที่มีการเก็บกวาดข้อมูลขยะอัตโนมัติ

ไลบรารี

ภาษาซีใช้ไลบรารีเป็นวิธีการหลักสำหรับส่วนขยาย ไลบรารีคือกลุ่มของฟังก์ชันที่บรรจุอยู่ในไฟล์เดียวกันโดย "ถาวร" ไลบรารีแต่ละชนิดจะมีไฟล์ส่วนหัว ซึ่งรวบรวมต้นแบบ (prototype) ตามฟังก์ชันที่มีอยู่ในไลบรารีซึ่งอาจถูกเรียกใช้โดยโปรแกรม และมีการประกาศชนิดข้อมูลพิเศษ และสัญลักษณ์แมโครที่ใช้ในฟังก์ชันเหล่านั้น โปรแกรมจะต้องรวมไฟล์ส่วนหัวนี้เข้าไปเพื่อใช้งานไลบรารี และไลบรารีจะต้องเชื่อมโยงกับโปรแกรม ซึ่งในหลายกรณีอาจต้องใช้ตัวบ่งชี้คอมไพเลอร์ (compiler flag) (เช่น -lm สำหรับไลบรารีคณิตศาสตร์เป็นต้น)

ไลบรารีสามัญที่สุดคือไลบรารีมาตรฐานของภาษาซี ซึ่งระบุไว้โดยมาตรฐานไอโซและแอนซีซีและติดมากับทุกโปรแกรมที่พัฒนาด้วยภาษาซี (ส่วนการพัฒนามนสภาพแวดล้อมแบบฝังตัวอาจมีไลบรารีมาตรฐานเพียงส่วนย่อยส่วนหนึ่ง) ไลบรารีนี้รองรับกระแสข้อมูลรับเข้าและส่งออก การจัดสรรหน่วยความจำ คณิตศาสตร์ สายอักขระ และค่าของเวลา

ไลบรารีสามัญอีกกลุ่มหนึ่งเป็นฟังก์ชันที่เจาะจงใช้กับโปรแกรมที่ทำงานบนระบบปฏิบัติการยูนิกซ์หรือคล้ายยูนิกซ์ โดยเฉพาะฟังก์ชันที่มีส่วนต่อประสานเข้ากับเคอร์เนล ฟังก์ชันเหล่านี้ได้ให้รายละเอียดไว้ในมาตรฐานหลากหลายเช่นโพสซิกซ์หรือข้อกำหนดคุณลักษณะยูนิกซ์เชิงเดี่ยว (Single UNIX Specification)

เนื่องจากโปรแกรมหลายโปรแกรมถูกเขียนขึ้นด้วยภาษาซี ไลบรารีอื่น ๆ ที่หลากหลายในวงกว้างก็มีเช่นกัน บ่อยครั้งที่ไลบรารีเหล่านั้นเขียนด้วยภาษาซี เพราะตัวแปลภาษาซีจะจัดสร้างรหัสวัตถุ (object code) ที่มีประสิทธิภาพ จากนั้นโปรแกรมเมอร์จะสร้างส่วนต่อประสานไปยังไลบรารี จึงทำให้ภาษาระดับที่สูงกว่าอย่างภาษาจาวา ภาษาเพิร์ล และภาษาไพทอน สามารถใช้งานรูทีนในรหัสวัตถุได้

เครื่องมือที่ใช้กับภาษา

เครื่องมือหลายอย่างถูกสร้างขึ้นเพื่อช่วยเหลือโปรแกรมเมอร์ภาษาซี เพื่อหลีกเลี่ยงปัญหาบางประเภทที่มากับภาษา เช่น ข้อความสั่งที่มีพฤติกรรมไม่นิยาม หรือข้อความสั่งที่ปฏิบัติไม่ดีซึ่งอาจส่งผลให้เกิดพฤติกรรมที่ไม่ตั้งใจหรือความผิดพลาดขณะทำงาน

การตรวจสอบแก้ไขรหัสต้นฉบับอัตโนมัติเป็นประโยชน์สำหรับทุกภาษา และภาษาซีก็มีเครื่องมือเหล่านั้นเช่นกันเช่น lint การใช้ lint โดยปกติเพื่อตรวจจบบั๊กที่พบบ่อยเมื่อโปรแกรมเขียนขึ้นเป็นครั้งแรก เมื่อโปรแกรมผ่านการตรวจสอบจาก lint แล้ว มันจึงจะถูกแปลด้วยตัวแปลภาษาซี ตัวแปลภาษาหลายตัวก็สามารถเลือกได้เพื่อแข่งเดือน เกี่ยวกับโครงสร้างที่ถูกต้องตามวากยสัมพันธ์แต่อาจเกิดความผิดพลาดได้จริง มีสราซี เป็นกลุ่มแนวทางที่มีกรรมสิทธิ์เพื่อการหลีกเลี่ยงรหัสที่น่าสงสัยเช่นนั้น ซึ่งพัฒนาขึ้นสำหรับระบบฝังตัว

นอกจากนี้ยังมีตัวแปลโปรแกรม ไบบรารี และกลไกระดับระบบปฏิบัติการ เพื่อการตรวจสอบขอบเขตของแฉวลำดับ การตรวจจับบั๊กเฟอรัสส่วนล้น การทำให้เป็นอนุกรม (serialization) และการเก็บกวาดข้อมูลขยะอัตโนมัติ ซึ่งมีใช้ส่วนหนึ่งที่เป็นมาตรฐานของภาษาซี

เครื่องมืออื่นอย่างเช่น เพียวริฟาย แวลกรินด์ และการเชื่อมโยงกับไลบรารีที่มีฟังก์ชันจัดสรรหน่วยความจำแบบพิเศษ สามารถช่วยเปิดเผยข้อผิดพลาดในหน่วยความจำขณะทำงานได้

ภาษาที่เกี่ยวข้อง

ภาษาซีมีอิทธิพลต่อภาษาอื่นในยุคหลังทั้งในทางตรงและทางอ้อมเช่น ภาษาจาวา ภาษาเพิร์ล ภาษาพีเอชพี จาวาสคริปต์ ภาษาแอลพีซี ภาษาซีชาร์ป และซีเชลล์ของยูนิกซ์ อิทธิพลที่แพร่หลายมากที่สุดคือรูปแบบวากยสัมพันธ์ ทุกภาษาที่กล่าวมาได้รวมวากยสัมพันธ์ของข้อความตั้งกับนิพจน์ของภาษาซี พร้อมทั้งระบบชนิดตัวแปร อันเป็นตัวแทนข้อมูลและ/หรือโครงสร้างโปรแกรมขนาดใหญ่ที่ต่างไปจากของภาษาซี ซึ่งบางครั้งก็ต่างกันอย่างมาก

เมื่อแนวคิดภาษาเชิงวัตถุเป็นที่นิยม ภาษาซีพลัสพลัสและภาษาอ็อบเจกทีฟ-ซีเป็นส่วนขยายที่แตกต่างกันของภาษาซีที่ให้ความสามารถเชิงวัตถุได้ ภาษาทั้งสองแต่เดิมทำให้เกิดผลโดยใช้ตัวแปลภาษาแบบแปลรหัสต่อรหัส นั่นคือรหัสต้นฉบับของภาษาดังกล่าวจะถูกแปลเป็นรหัสภาษาซีก่อน จากนั้นจึงแปลด้วยคอมไพเลอร์อีกต่อหนึ่ง

ภาษาซีพลัสพลัสประดิษฐ์ขึ้นโดยเบียนอ สตราสตรูป (Bjarne Stroustrup) ให้เป็นภาษาที่มีการทำงานเชิงวัตถุโดยมีวากยสัมพันธ์คล้ายภาษาซี ภาษาซีพลัสพลัสเพิ่มเติมความรัดกุมต่อชนิด

ตัวแปร ขอบข่าย และเครื่องมืออื่น ๆ ที่เป็นประโยชน์ในการเขียนโปรแกรมเชิงวัตถุ และอนุญาตให้เขียนโปรแกรมเชิงทั่วไปผ่านแม่แบบ ภาษาซีพลัสพลัสรองรับรหัสส่วนใหญ่ของภาษาซีจนแทบจะครอบคลุมทั้งหมด แต่ก็มีข้อยกเว้นบางประการ (ดูเพิ่มที่ ความเข้ากันได้ระหว่างภาษาซีและภาษาซีพลัสพลัส สำหรับรายการความแตกต่างโดยละเอียด)

ภาษาอ็อบเจกทีฟ-ซีเดิมเป็นเพียง "ชั้นบาง ๆ" บนภาษาซีและยังคงครอบคลุมภาษาซีอย่างเข้มงวด ซึ่งอนุญาตให้เขียนโปรแกรมเชิงวัตถุโดยใช้กระบวนการที่ซับซ้อนตัวแปรผสมพลวัต/สถิต วากยสัมพันธ์ของภาษาอ็อบเจกทีฟ-ซีมาจากทั้งภาษาซีและภาษาสมอลล์ทอล์ก นั่นคือ วากยสัมพันธ์ที่เกี่ยวข้องกับการประมวลผลก่อน นิพจน์ การประกาศฟังก์ชัน และการเรียกใช้ฟังก์ชันรับมาจากภาษาซี ในขณะที่วากยสัมพันธ์สำหรับคุณลักษณะเชิงวัตถุนำมาจากภาษาสมอลล์ทอล์ก

ภาษาดีทำคุณลักษณะหลายอย่างให้ต่างออกไปแต่ยังคงไว้ซึ่งวากยสัมพันธ์ทั่วไปของภาษาซี ไม่เหมือนภาษาซีพลัสพลัสที่แทบจะเข้ากันได้แบบย้อนหลังกับภาษาซี ภาษาดีละทิ้งคุณลักษณะจำนวนหนึ่งของภาษาซีออกไป เนื่องจากวอลเตอร์ ไบรต์ (Walter Bright) ผู้ออกแบบภาษาดี พิจารณาว่าไม่มีความจำเป็นต้องใช้คุณลักษณะเหล่านั้น รวมทั้งตัวประมวลผลก่อนและไดร้อกเกอร์ ส่วนขยายบางอย่างของภาษาดีไปยังภาษาซี ทับซ้อนกับส่วนขยายไปยังภาษาซีพลัสพลัส

ภาษาลิมโบเป็นภาษาหนึ่งพัฒนาโดยทีมงานที่เบลล์แล็บส์ และในขณะที่ยังคงรักษาวากยสัมพันธ์และลักษณะทั่วไปบางอย่างของภาษาซี ก็ยังมีการเก็บกวาดข้อมูลขยะและภาวะพร้อมกันที่มีพื้นฐานบนกระบวนการสื่อสารแบบลำดับ (communicating sequential processes)

ภาษาไพทอนสืบทอดมาจากภาษาซีในแนวทางที่ต่างออกไป ในขณะที่วากยสัมพันธ์และความหมายของภาษาไพทอนแตกต่างกับภาษาซีอย่างสิ้นเชิง แต่เครื่องมือทำให้เกิดผลในภาษาไพทอนที่ใช้กันอย่างกว้างขวางที่สุดคือซีไพทอน ซึ่งเป็นโปรแกรมภาษาซีแบบโอเพนซอร์ซ สิ่งนี้ช่วยให้ผู้ใช้สามารถเขียนภาษาซีเป็นส่วนขยายของภาษาไพทอน หรือฝังภาษาไพทอนลงในโปรแกรมภาษาซี ความสัมพันธ์อย่างใกล้ชิดนี้เป็นปัจจัยหนึ่งที่นำไปสู่ความสำเร็จของภาษาไพทอนในฐานะภาษาพลวัตเพื่อการใช้งานทั่วไป

ภาษาเพิร์ลเป็นอีกตัวอย่างหนึ่งของภาษาโปรแกรมที่มีต้นกำเนิดจากภาษาซี โครงสร้างโดยรวมทั้งหมดของภาษาเพิร์ลมาจากภาษาซีอย่างมาก เครื่องมือทำให้เกิดผลของภาษาเพิร์ลมาตรฐานเขียนขึ้นด้วยภาษาซี และรองรับส่วนขยายที่เขียนในภาษาซีด้วย

ตัวแปรในภาษาซี

ตัวแปร (Variable) คือ การจองพื้นที่ในหน่วยความจำของคอมพิวเตอร์สำหรับเก็บข้อมูลที่ต้องใช้ในการทำงานของโปรแกรม โดยมีการตั้งชื่อเรียกหน่วยความจำในตำแหน่งนั้นด้วย เพื่อความสะดวกในการเรียกใช้ข้อมูล ถ้าจะใช้ข้อมูลใดก็ให้เรียกผ่านชื่อของตัวแปรที่เก็บเอาไว้

ชนิดของข้อมูล

ภาษาซีเป็นอีกภาษาหนึ่งที่มีชนิดของข้อมูลให้ใช้งานหลายอย่างด้วยกัน ซึ่งชนิดของข้อมูลแต่ละอย่างมีขนาดเนื้อที่ที่ใช้ในหน่วยความจำที่แตกต่างกัน และเนื่องจากการที่มีขนาดที่แตกต่างกันไป ดังนั้นในการเลือกใช้งานประเภทข้อมูลก็ควรจะคำนึงถึงความจำเป็นในการใช้งานด้วย สำหรับประเภทของข้อมูลมีดังนี้คือ

1. ข้อมูลชนิดตัวอักษร (Character) คือข้อมูลที่เป็นรหัสแทนตัวอักษรหรือค่าจำนวนเต็มได้แก่ตัวอักษร ตัวเลข และกลุ่มตัวอักษรพิเศษ ใช้พื้นที่ในการเก็บข้อมูล 1 ไบต์
2. ข้อมูลชนิดจำนวนเต็ม (Integer) คือข้อมูลที่เป็นเลขจำนวนเต็ม ได้แก่ จำนวนเต็มบวก จำนวนเต็มลบ ศูนย์ ใช้พื้นที่ในการเก็บ 2 ไบต์
3. ข้อมูลชนิดจำนวนเต็มที่มีขนาด 2 เท่า (Long Integer) คือข้อมูลที่มีเลขเป็นจำนวนเต็ม ใช้พื้นที่ 4 ไบต์
4. ข้อมูลชนิดเลขทศนิยม (Float) คือข้อมูลที่เป็นเลขทศนิยม ขนาด 4 ไบต์
5. ข้อมูลชนิดเลขทศนิยมอย่างละเอียด (Double) คือข้อมูลที่เป็นเลขทศนิยม ใช้พื้นที่ในการเก็บ 8 ไบต์

ชนิด	ขนาดความกว้าง	ช่วงของค่า	การใช้งาน
Char	8 บิต	ASCII character (-128 ถึง 127)	เก็บข้อมูลชนิดอักขระ
Unsignedchar	8 บิต	0-255	เก็บข้อมูลอักขระแบบไม่คิดเครื่องหมาย
Int	16 บิต	-32768 ถึง 32767	เก็บข้อมูลชนิดจำนวนเต็ม
long	32 บิต	-2147483648 ถึง 2147483649	เก็บข้อมูลชนิดจำนวนเต็มแบบ

			ยาว
Float	32 บิต	3.4E-38 ถึง 3.4E+38 หรือ ทศนิยม 6	เก็บข้อมูลชนิดเลขทศนิยม
Double	64 บิต	1.7E-308 ถึง 1.7E+308 หรือ ทศนิยม 12	เก็บข้อมูลชนิดเลขทศนิยม
Unsigned int	16 บิต	0 ถึง 65535	เก็บข้อมูลชนิดจำนวนเต็ม ไม่คิดเครื่องหมาย
Unsigned long	32 บิต	0 ถึง 4294967296	เก็บข้อมูลชนิดจำนวนเต็มแบบยาว ไม่คิดเครื่องหมาย

ตารางที่ 2.1 รายละเอียดของตัวแปรภาษาซีแต่ละชนิด

รูปแบบในการประกาศตัวแปรในภาษา C

การสร้างตัวแปรขึ้นมาใช้งานจะเรียกว่า การประกาศตัวแปร (Variable Declaration) โดยเขียนคำสั่งให้ถูกต้องตามแบบการประกาศตัวแปร แสดงดังนี้

type name;

type : ชนิดของตัวแปร

name : ชื่อของตัวแปร ซึ่งต้องตั้งให้ถูกต้องตามหลักของภาษา C

การเขียนคำสั่งเพื่อประกาศตัวแปร ส่วนใหญ่แล้วจะเขียนไว้ในส่วนหัวของโปรแกรมก่อนฟังก์ชัน main ซึ่งการเขียนไว้ในตำแหน่งดังกล่าว จะทำให้ตัวแปรเหล่านั้นสามารถเรียกใช้จากที่ใดก็ได้ในโปรแกรม ดังตัวอย่าง

```
#include <stdio.h>
```

```
int num;
```

```
float y;
```

```
char n;
```

```
void main()
```

สร้างตัวแปรชื่อ num เพื่อเก็บข้อมูลชนิดจำนวนเต็ม

สร้างตัวแปรชื่อ y เพื่อเก็บข้อมูลชนิดเลขทศนิยม

สร้างตัวแปรชื่อ n เพื่อเก็บข้อมูลชนิดตัวอักษร


```

{
    printf("Enter number : ")
    scanf("%d",&num);

    printf("Enter name : ");
    scanf("%f",&n);

    printf("Thank you");
}

```

หลักการตั้งชื่อตัวแปร

ในการประกาศสร้างตัวแปรต้องมีการกำหนดชื่อ ซึ่งชื่อนั้นไม่ว่าจะตั้งให้สื่อความหมายถึงข้อมูลที่เก็บอย่างเดียวยัง โดยไม่คำนึงถึงอย่างอื่น เนื่องจากภาษา C มีข้อกำหนดในการตั้งชื่อตัวแปรเอาไว้ แล้วถ้าตั้งชื่อผิดหลักการเหล่านี้ โปรแกรมจะไม่สามารถทำงานได้ หลักการตั้งชื่อตัวแปรในภาษา C แสดงไว้ดังนี้

1. ต้องขึ้นต้นด้วยตัวอักษร A-Z หรือ a-z หรือเครื่องหมาย _ (Underscore) เท่านั้น
2. ภายในชื่อตัวแปรสามารถใช้ตัวอักษร A-Z หรือ a-z หรือตัวเลข 0-9 หรือเครื่องหมาย _
3. ภายในชื่อห้ามเว้นช่องว่าง หรือใช้สัญลักษณ์นอกเหนือจากข้อ 2
4. ตัวอักษรเลขหรือใหญ่มีความหมายแตกต่างกัน
5. ห้ามตั้งชื่อซ้ำกับคำสงวน (Reserved Word) ดังนี้

auto	default	float	register	struct	volatile	break
Do	far	return	switch	while	Case	double
goto	short	typedef	char	else	If	signed
union	const	enum	int	sizeof	unsigned	continue
extern	long	static	void			

ตารางที่ 2.2 ตารางแสดง คำสงวน ที่ห้ามใช้ตั้งเป็นชื่อตัวแปร

ตัวอย่างการตั้งชื่อตัวแปรในภาษา C ทั้งที่ถูกต้องและไม่ถูกต้องตามหลักการ แสดงดังนี้

bath_room	ถูกต้อง
n-sync	ผิดหลักการ เนื่องจากมีเครื่องหมาย - ปรากฏในชื่อ
108dots	ผิดหลักการ เนื่องจากขึ้นต้นด้วยตัวเลข
Year#	ผิดหลักการ เนื่องจากมีเครื่องหมาย # อยู่ในชื่อ
_good	ถูกต้อง
Goto	ผิดหลักการ เนื่องจากเป็นคำสงวน
work	ถูกต้อง
break	ผิดหลักการ เนื่องจากเป็นคำสงวน

ตารางที่ 2.3 ตารางแสดงตัวอย่างที่ถูกต้องและไม่ถูกต้องในการตั้งชื่อตัวแปรในภาษาซี

ตัวแปรสำหรับข้อความ

ในภาษา C ไม่มีการกำหนดชนิดของตัวแปรสำหรับข้อความโดยตรง แต่จะใช้การกำหนดชนิดของตัวแปรอักขระ (char) ร่วมกับการกำหนดขนาดแทน และจะเรียกตัวแปรสำหรับเก็บข้อความว่า ตัวแปรสตริง (string) รูปแบบการประกาศตัวแปรสตริงแสดงได้ดังนี้

```
char name[n] = "str";
```

name ชื่อของตัวแปร

n ขนาดของข้อความ หรือจำนวนอักขระในข้อความ

str ข้อความเริ่มต้นที่จะกำหนดให้กับตัวแปรซึ่งต้องเขียนไว้ภายในเครื่องหมาย " "

ตัวอย่างการประกาศตัวแปรสำหรับเก็บข้อความ แสดงได้ดังนี้

```
char name[5] = "kwan" สร้างตัวแปร name สำหรับเก็บ ข้อความ kwan ซึ่งมี 4 ตัวอักษร ดังนั้น
;
```

name ต้องมีขนาด 5

char year[5] = "2549"; สร้างตัวแปร year สำหรับเก็บ ข้อความ 2549 ซึ่งมี 4 ตัวอักษร ดังนั้น
year ต้องมีขนาด 5

char product_id[4] = สร้างตัวแปร product_id สำหรับเก็บ ข้อความ A01 ซึ่งมี 3 ตัวอักษร
"A01"; ดังนั้น product_id ต้องมีขนาด 4



บทที่ 3

แผนงานการปฏิบัติงานและขั้นตอนการดำเนินงาน

3.1 แผนงานการปฏิบัติงาน

ตารางที่ 3.1 ตารางแสดงงาน วัน และรายละเอียด

รายละเอียด	เดือนที่1				เดือนที่2				เดือนที่3				เดือนที่4			
วางทิศทาง ของ ผลิตภัณฑ์ที่ จะทำ																
ศึกษาเรื่อง สิทธิบัตร เพื่อป้องกัน การซ้ำซ้อน																
ลงมือทำ ผลิตภัณฑ์ ใหม่																
ทดสอบ ผลิตภัณฑ์																
แก้ไขและ ปรับปรุง เพิ่มเติม																

3.2 รายละเอียดโครงการที่ได้รับมอบหมาย

- วางแนวทางของผลิตภัณฑ์ใหม่ร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย
- ศึกษาเกี่ยวกับสิทธิบัตรเพื่อป้องกันการมีปัญหาระหว่างสิทธิบัตรและปรับปรุงผลิตภัณฑ์ไม่ให้ทับซ้อนกับสิ่งที่ได้จดสิทธิบัตรไว้แล้ว ร่วมกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย
- ศึกษาโปรโตคอลที่ใช้สื่อสารระหว่างตัวรีโมตกับทีวี
- ศึกษาและเข้าอบรมหลักการการทำงานของ Capacitive Touch Sensing ไม่ว่าจะเป็น Capacitive Sensing Module (CSM), Capacitive Voltage Divider (CVD) และ Charge Time Measurement Unit (CTMU) ร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย

- จัดหา Part List ที่เป็นไปได้ที่จะใช้ในรีโมต
- คิด interface ในทีวีที่จะใช้ร่วมกันกับรีโมตและสวิตช์บนทีวี
- ทำผลิตภัณฑ์ต้นแบบทั้งหมดไม่ว่าจะเป็นกรอบรีโมต, PCB ที่ใช้ในรีโมต จัดทำชุดแสดงสวิตช์ในส่วนของทีวีร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย
- เขียนโปรแกรมลงในไมโครคอนโทรลเลอร์ที่ใช้กับตัวรีโมตโดยยึดตามหลักการโปรโตคอลที่ได้ศึกษาไว้ข้างต้น เขียนโปรแกรมที่ใช้สำหรับสวิตช์บนส่วนของทีวีและทดลองใช้งาน

3.3 ขั้นตอนการดำเนินงานโครงการงาน

3.3.1 วางแนวทางผลิตภัณฑ์ที่ต้องการ

เนื่องจากเราต้องการให้ผลิตภัณฑ์ใหม่ที่ทำขึ้นมาให้ผู้ได้ความรู้สึกที่แตกต่างเมื่อได้ใช้งาน จึงต้องคิดก่อนว่าทำอย่างไรให้ผู้ใช้งานได้อย่างสะดวกและง่ายที่สุด หลังจากนั้นจึงหาวิธีที่จะทำอย่างไรที่จะได้มาซึ่งผลิตภัณฑ์ที่เราตั้งเป้าไว้

3.3.2 ศึกษาเรื่องสิทธิบัตร

เนื่องจากปัจจุบันการทำผลิตภัณฑ์นั้นมีปัญหาฟ้องร้องเรื่องสิทธิบัตรมากมายหลายกรณี เพื่อหลีกเลี่ยงปัญหาที่จะเกิดขึ้นจึงต้องมีการศึกษาและปรับปรุงไม่ให้ผลิตภัณฑ์ที่ทำขึ้นมาใหม่นั้นเกิดปัญหาฟ้องร้องขึ้นได้ โดยศึกษาจากเว็บไซต์ <http://www.google.com/patents> ในหัวข้อที่สนใจและดูที่หัวข้อ What is claimed is ว่าทับซ้อนกับสิ่งเรากำลังจะทำหรือไม่



US 20070177804A1

(19) **United States**
 (12) **Patent Application Publication** (10) **Pub. No.: US 2007/0177804 A1**
 Elias et al. (43) **Pub. Date: Aug. 2, 2007**

(54) **MULTI-TOUCH GESTURE DICTIONARY** **Publication Classification**

(75) **Inventors:** John Greer Elias, Townsend, DE (US); Wayne Carl Westerman, San Francisco, CA (US); Myra Mary Haggerty, San Mateo, CA (US)

(51) **Int. Cl.** G06K 9/00 (2006.01)
 (52) **U.S. CL.** 382/188

(57) **ABSTRACT**
 A multi-touch gesture dictionary is disclosed herein. The gesture dictionary can include a plurality of entries, each corresponding to a particular chord. The dictionary entries can include a variety of motions associated with the chord and the meanings of gestures formed from the chord and the motions. The gesture dictionary may take the form of a dedicated computer application that may be used to look up the meaning of gestures. The gesture dictionary may also take the form of a computer application that may be easily accessed from other applications. The gesture dictionary may also be used to assign user-selected meanings to gestures. Also disclosed herein are computer systems incorporating multi-touch gesture dictionaries. The computer systems can include, desktop computers, tablet computers, notebook computers, handheld computers, personal digital assistants, media players, mobile telephones, and the like.

Correspondence Address:
WONG, CABELLO, LUTSCH, RUTHERFORD & BRUCCULERI LLP
 20333 SH 249, SUITE 600
 HOUSTON, TX 77070

(73) **Assignee:** APPLE COMPUTER, INC., CUPERTINO, CA (US)

(21) **Appl. No.:** 11/619,571
 (22) **Filed:** Jan. 3, 2007

Related U.S. Application Data
 (60) Provisional application No. 60/763,605, filed on Jan. 3, 2006

รูปที่ 3.1 Multi-Touch Gesture Dictionary Patent

What is claimed is:

1. A method of providing to a user of a device a dictionary of multi-touch gestures usable to interact with the device, each of the gestures comprising a chord and a motion associated with the chord, wherein a chord comprises a predetermined number of hand parts in a predetermined configuration, the method comprising:

identifying a trigger presented by the user, wherein the trigger comprises a user interaction with the device; and

displaying a multi-touch gesture dictionary in response to the presented trigger; wherein

if the trigger uniquely corresponds to a chord, displaying a multi-touch gesture dictionary comprises displaying a dictionary entry associated with the chord; and

if the trigger does not uniquely correspond to a chord, displaying a multi-touch gesture dictionary comprises displaying a chord index, receiving one or more inputs indicating a selection of at least one chord from the chord index, and displaying a dictionary entry corresponding to the selected at least one chord.

รูปที่ 3.2 Multi-Touch Gesture Dictionary Patent What is claimed is

3.3.3 ศึกษาโปรโตคอลที่ใช้สื่อสารระหว่างรีโมตและทีวี

การที่จะสื่อสารกับตัวทีวีพานาโซนิคได้นั้นจำเป็นจะต้องใช้โปรโตคอลของพานาโซนิคเองจึงจะสามารถสื่อสารและสั่งการตัวทีวีพานาโซนิคได้ จึงจำเป็นที่จะต้องศึกษาโปรโตคอลของทีวีพานาโซนิค (ทั้งนี้ โปรโตคอลที่ใช้สื่อสารระหว่างรีโมตและทีวีของพานาโซนิคในแต่ละคำสั่งที่ใช้ในการสั่งการทีวีว่าปุ่มใดต้องส่งสัญญาณบิตแบบไหนนั้นเป็นข้อมูลความลับของบริษัท ทางผู้จัดทำจึงขอสงวนข้อมูลนี้ไว้เป็นความลับ)

3.3.4 ศึกษาและเข้าอบรมเกี่ยวกับหลักการทำงานของ Capacitive Touch Sensing

เนื่องจากเทคโนโลยี Capacitive Touch Sensing มีหลายเทคโนโลยีไม่ว่าจะเป็น Capacitive Sensing Module (CSM), Capacitive Voltage Divider (CVD) และ Charge Time Measurement Unit (CTMU) แต่ละเทคโนโลยีก็ต่างมีทั้งจุดดีและจุดด้อยในตัวเอง จึงต้องมีการศึกษาว่าควรจะนำเทคโนโลยีส่วนไหนมาใช้กับผลิตภัณฑ์ ซึ่งปัจจัยที่เรานำมาเลือกใช้ระบบ CVD เนื่องจาก มีราคาที่ถูกและต้องใช้ต้นทุนน้อยกว่าแบบ CTMU ซึ่งจะต้องอาศัย Current Source (แหล่งจ่ายกระแส) เพิ่มในส่วนของอุปกรณ์ภายนอกเอง อีกทั้งตัวชิปเอง ก็ยังมีราคาที่แพงกว่าแบบ CVD อีกทั้งยังกินพลังงานมากกว่าแบบ CVD ส่วนแบบ CSM นั้นมีปัญหาทางด้าน noise (สัญญาณรบกวน) ที่ถูก

รบกวนได้โดยง่าย เนื่องจากอาศัยคลื่นเป็นหลัก หากมีคลื่นอื่น ๆ แทรกเข้ามาจะทำให้การรับสัญญาณผิดเพี้ยนไปได้

ตารางที่ 3.2 ตารางเปรียบเทียบข้อดีของแต่ละระบบของ Capacitive Touch sensing

Advantage point

CVD	CTMU	CSM
1. Low power(nA) and noise	1. Low noise	1. Voltage drop by battery will not have any effect to result
2. Fast response time	2. Very fast response time	2. Lowest power and cost
3. Used in both systems (line, battery)	3. Used in both systems (line, battery)	
4. Easy to implement (Code)	4. Easy to implement (Code)	
5. Can detect under solution on surface	5. Can detect under solution on surface	
	6. Can endure frequency at +6 Vrms	

ตารางที่ 3.3 ตารางเปรียบเทียบข้อเสีย ของแต่ละระบบของ Capacitive Touch Sensing

Disadvantage point

CVD	CTMU	CSM
1. Voltage drop in circuit will cause sensing problem	1. Use power (uA) for constant current source	1. Frequency noise at sensor input and timer1 input will cause sensing problem
2. Cannot use in high bit architecture MCU	2. Voltage drop in circuit will cause sensing problem	2. Hard to implement (Code)
	3. 2 external pins are provided to trigger the current source.	3. Hard to detect under solution on surface
	4. Highest cost	4. Can use only battery-powered system.

ตารางที่ 3.4 ตารางเปรียบเทียบคุณสมบัติ ของแต่ละ Chip

MCU	PIC18F46J11	PIC18F1XK22	PIC16LF1826	PIC16LF707	PIC16LF1516
operating voltage	2.0-3.6	1.8-3.6	1.8-3.6	1.8-3.6	1.8-3.6
Deep sleep mode	13nA	-	-	-	
sleep mode	105nA	34nA	30nA	20nA	20nA
Idle	2.3uA	-	-		
Run	6.2uA	-	-		
Watchdog timer	813nA	460nA	500nA	500nA	600nA
I/O PORT	23 (10pin 5V)	17	15		35
SPI port	2	1	2	1	1
Cap CH	13	12	12	14	28
SYSTEM	CTMU	CTMU	CSM	CSM	CVD

หลังจากที่ได้เปรียบเทียบแล้วจึงเลือกที่จะใช้วิธี CVD เนื่องจากมีสัญญาณรบกวนน้อยกว่าแบบ CSM แต่ยังราคาถูกและกินไฟน้อยกว่าแบบ CTMU จากนั้นจึงเข้าอบรมที่บริษัท Arizona Microchip Technology (Thailand) Co., Ltd. เกี่ยวกับการทำ Capacitive Touch Sensing

3.3.5 จัดทำ Part List ที่ที่เป็นไปได้ที่จะใช้ในรีโมต

คิดชิ้นส่วนที่เป็นไปได้ทั้งหมดที่จะใช้ลงไปในรีโมตทำลงในตารางเพื่อเทียบราคาและอัตราของที่สั่งได้ต่อเดือน

3.3.6 คิด interface บนทีวีที่ใช้ร่วมกันกับรีโมตและสวิตช์แบบใหม่

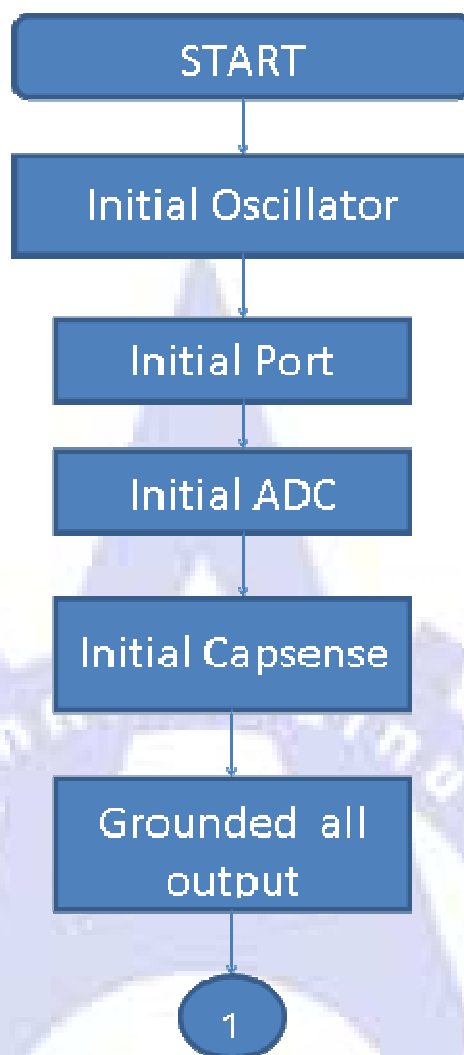
เนื่องจากการทำรีโมตและสวิตช์แบบใหม่นั้นจะทำให้ความรู้สึกของผู้ใช้งานไม่เหมือนเดิม interface ที่อยู่บนตัวทีวีนั้นจึงจำเป็นต้องสอดคล้องกันกับการใช้งานของผลิตภัณฑ์ที่ทำขึ้นใหม่ด้วย

3.3.7 ทำผลิตภัณฑ์ต้นแบบขึ้นมาใหม่ทั้งหมด

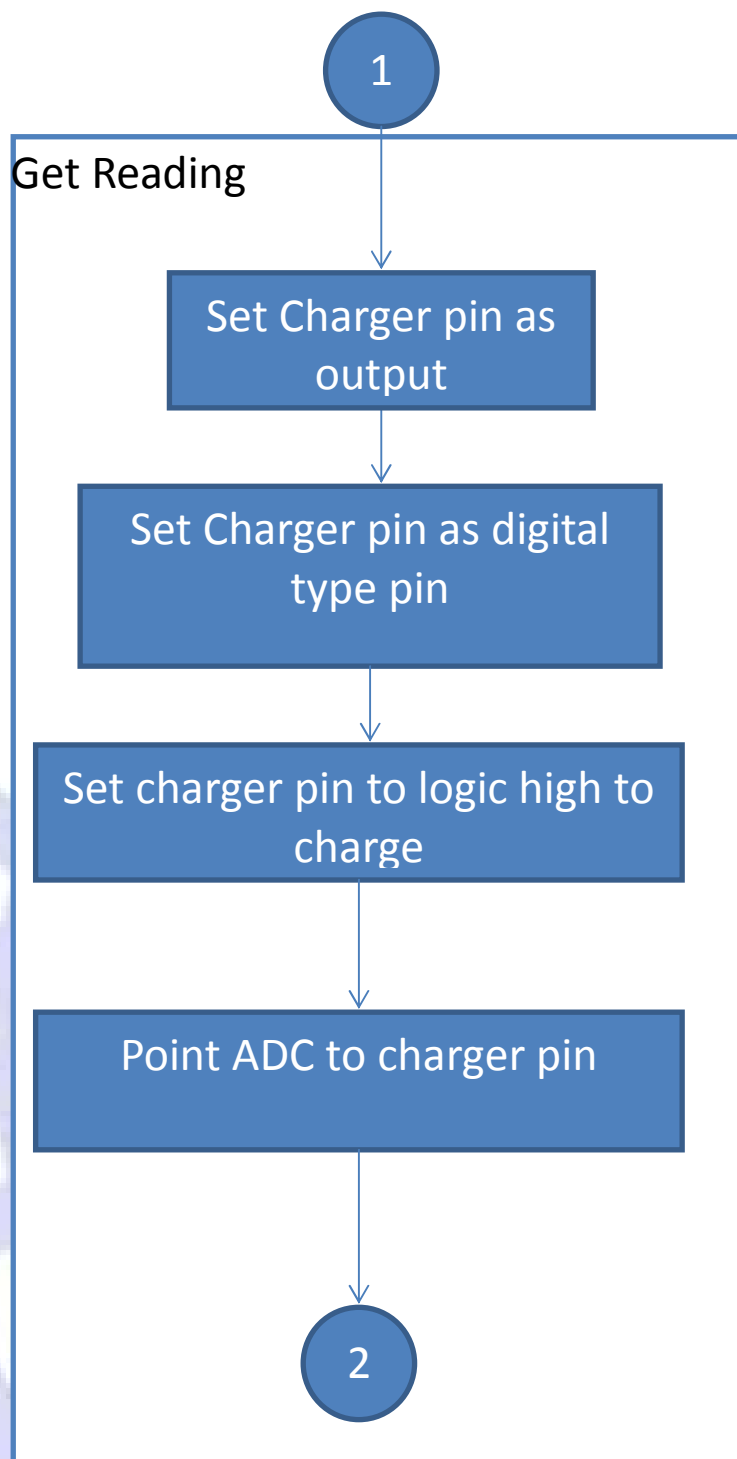
ไม่ว่าจะเป็นกรอบรีโมต, PCB ที่ใช้ในรีโมต จัดทำชุดแสดงสวิตช์ในส่วนของทีวีร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย

3.3.8 เขียนโปรแกรมลงในไมโครคอนโทรลเลอร์ที่ใช้กับตัวผลิตภัณฑ์

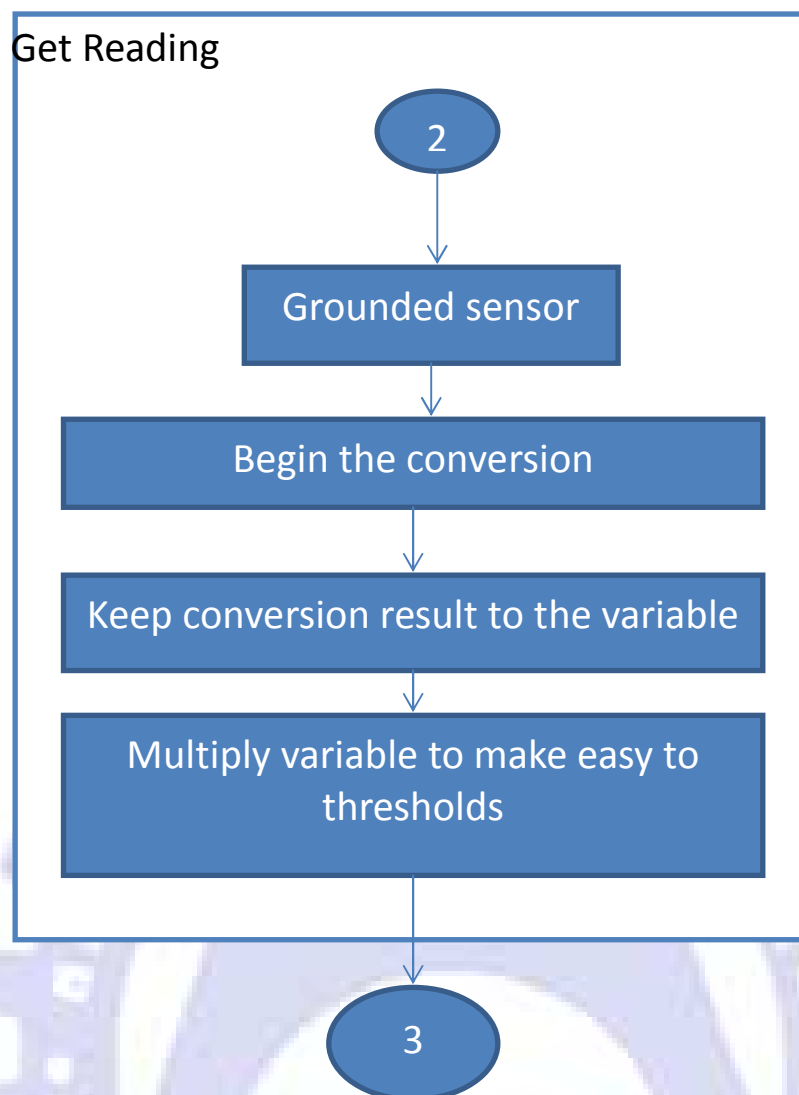
รีโมตโดยยึดตามหลักการโปรโตคอลที่ได้ศึกษาไว้ข้างต้น เขียนโปรแกรมที่ใช้สำหรับสวิตช์บนส่วนของทีวีด้วยภาษาซี และทดลองใช้งานผลิตภัณฑ์ โดยโปรแกรมมีส่วนการทำงานดังต่อไปนี้



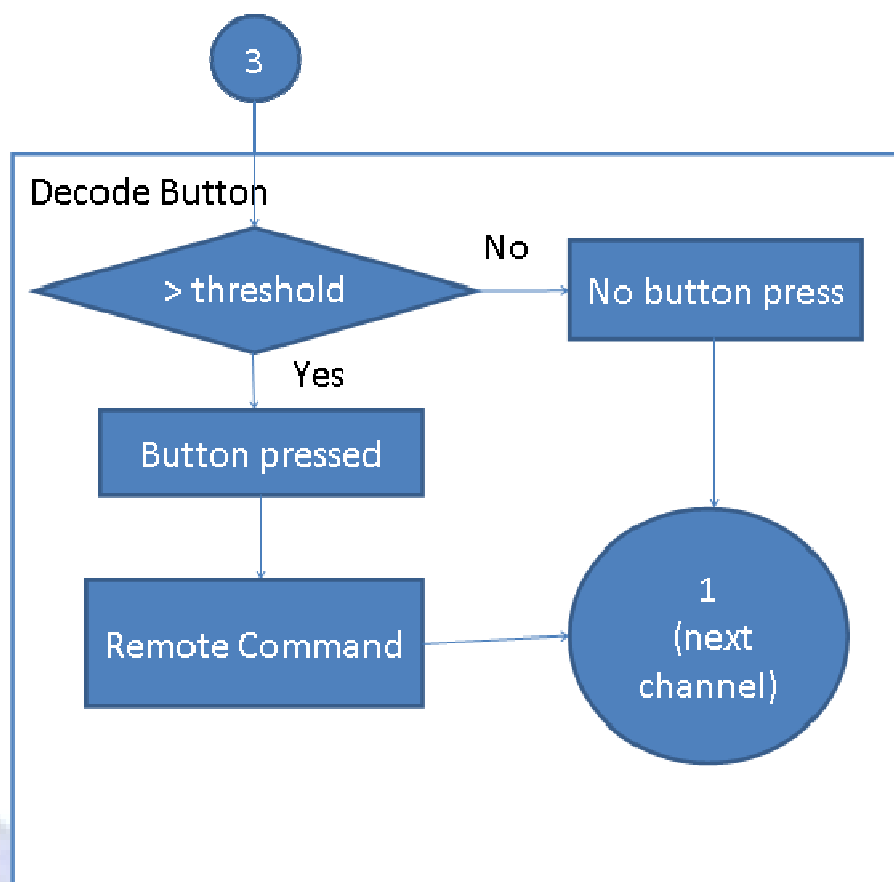
รูปที่ 3.3 Flow Chart แสดงขั้นตอนการทำงานของโปรแกรมรีโมต1



รูปที่ 3.4 Flow Chart แสดงขั้นตอนการทำงานของโปรแกรมรีโมต2

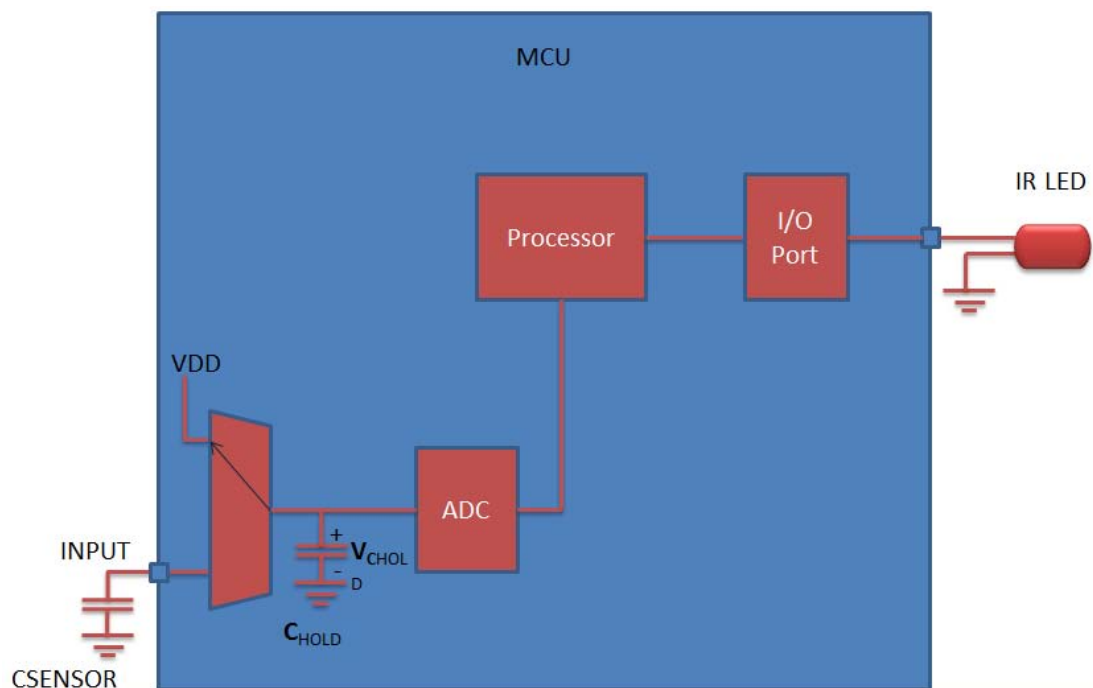


รูปที่ 3.5 Flow Chart แสดงขั้นตอนการทำงานของโปรแกรมรีโมต3



รูปที่ 3.6 Flow Chart แสดงขั้นตอนการทำงานของโปรแกรมรีโมต4

หมายเหตุ สำหรับสวิตช์บนตัวที่วินั้น ไฟล์ชาร์ตเหมือนกันกับรีโมตทั้งหมดแต่แทนที่จะส่งเป็นคำสั่งรีโมตก็ส่งเป็น UART เข้าคอมพิวเตอร์แทน



รูปที่ 3.7 Overview Block Diagram

3.3.9 ทดลองใช้งานและปรับปรุงแก้ไขเพิ่มเติม

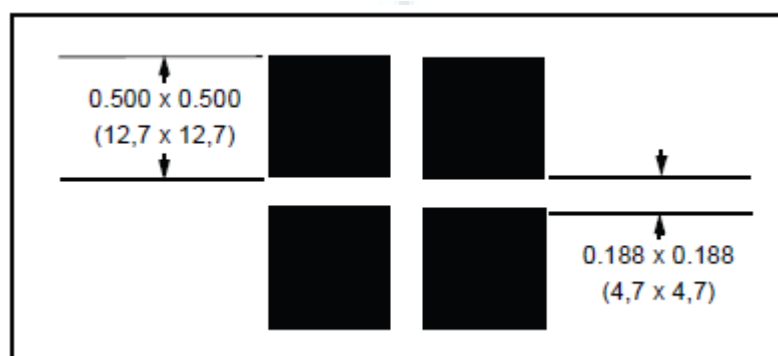
ทดลองใช้งานผลิตภัณฑ์ที่ทำขึ้นมาโดยทดลองใช้กับทีวีพลาสมาโดยใช้ฟังก์ชันปุ่มต่าง ๆ ได้เหมือนรีโมตปกติและแก้ไขส่วนผิดพลาดที่เกิดขึ้นในตัวผลิตภัณฑ์ โดยเริ่มจากการทดลองการทวน้ำของเซนเซอร์ ซึ่งจากการศึกษาแล้วพบว่า การที่หยดน้ำลงบนตัวเซนเซอร์นั้น ทำให้ค่าความต่างศักย์ที่ ADC อ่านค่าได้นั้น มีค่าสูงขึ้นกว่าตอนที่เซนเซอร์ไม่ได้ถูกกดและแห้ง ซึ่งจะให้ผลที่ต่างกับการที่ผู้ใช้งานกดลงไปที่เซนเซอร์ จะทำให้ได้ค่า ความต่างศักย์ที่ ADC อ่านได้นั้น น้อยลง จึงตั้งค่าการทำงานของวงจรส่งสัญญาณรีโมต ก็ต่อเมื่อ ADC อ่านค่าได้น้อยถึงค่าที่กำหนดเท่านั้นจึงจะทำงาน

ในการปรับปรุงในส่วนของการ Lay out (การออกแบบลายทองแดง) นั้นในส่วนของการรูปร่างของปุ่มนั้นไม่ได้สำคัญมากแต่สิ่งที่สำคัญคือพื้นที่บนตัวปุ่ม เพราะยังมีพื้นที่มากก็จะยิ่งเก็บประจุได้มาก ทำให้สามารถรับรู้การเปลี่ยนแปลงได้ดีกว่าพื้นที่น้อยซึ่งเป็นไปตามสมการ

$$C = \frac{\epsilon_0 \epsilon_r A}{d}$$

รูปที่ 3.8 Capacitance Equation

ในส่วนระยะห่างระหว่างเซนเซอร์นั้น เพื่อไม่ให้สัญญาณกวนกันเองจึงปรับให้ระยะห่างระหว่างตัวเซนเซอร์นั้น มีระยะห่างกันน้อยที่สุดอยู่ที่ 4.7 มิลลิเมตร และขนาดของปุ่มที่ต้องสอดคล้องกับขนาดของนิ้วคนโดยเฉลี่ย ซึ่งก็ได้ปรับให้เป็นขนาด 0.5 นิ้ว x 0.5 นิ้ว



รูปที่ 3.9 Example pad size and shape

ในการทำรีโมตและสวิตช์ในส่วนของคาบิเนตนั้นสวิตช์ที่เราใช้จะไม่ได้ปล่อยให้เห็นเซนเซอร์และแตะลงไปบนเซนเซอร์จริง แต่ต้องมีแผ่นพลาสติกมาคลุมอีกชั้นเพื่อความสวยงาม โดยทั่วไปแล้วไม่ว่าจะใช้วัสดุอะไรมาวางทับก็ตาม ไม่ควรจะมีความหนาเกิน 2 mm แต่ถ้าหากเป็น Pixiglas หรืออะคริลิคอื่น ๆ นั้นแม้ว่าจะมีขนาดหนาถึง 5 mm ก็ยังสามารถรับสัญญาณในระดับที่ยอมรับได้ สำหรับวัสดุที่เป็นตัวนำเช่น โลหะนั้น จะไม่สามารถนำมาเป็นแผ่นคลุมได้เนื่องจากวัสดุตัวนำนั้นสามารถที่จะดูดซับการแผ่กระจายที่ไม่โครคอนโทรลเลอร์ได้เซนเซอร์ได้ จึงทำให้การที่กดลงไปบนเซนเซอร์ ผ่านแผ่นตัวนำนั้น ไม่สามารถทำได้ หรือถ้าหากได้ ก็จะส่งสัญญาณไปถึงเซนเซอร์ทั้งหมดที่อยู่ภายใต้แผ่นคลุมตัวนำนั้น ๆ ทำให้เกิดการทำงานที่ผิดพลาดขึ้นได้

บทที่ 4

สรุปผลการดำเนินงาน การวิเคราะห์และสรุปผลต่าง ๆ

4.1 สรุปการดำเนินงาน

ในการปฏิบัติงานสหกิจศึกษา ณ สถานประกอบการเป็นเวลา 4 เดือน มีการดำเนินงาน โดยสรุปดังต่อไปนี้

- รับงานและฟังความต้องการที่เกี่ยวกับผลิตภัณฑ์ที่ต้องการจากหัวหน้างาน
- ศึกษาเกี่ยวกับสิทธิบัตรเพื่อป้องกันการมีปัญหาระหว่างสิทธิบัตรและปรับปรุงผลิตภัณฑ์ไม่ให้ทับซ้อนกับสิ่งที่ได้จดสิทธิบัตรไว้แล้ว ร่วมกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย
- ศึกษาและใช้งาน โปรโตคอลที่ใช้สื่อสารระหว่างตัวรีโมตกับทีวี
- ศึกษาและเข้าอบรมหลักการทำงานของ Capacitive Touch Sensing ร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย
- จัดทำ Part List ที่เป็นไปได้ที่จะใช้ในรีโมต
- คิด interface ในทีวีที่จะใช้ร่วมกันกับรีโมตและสวิตช์บนทีวีร่วมกันกับพนักงานที่ได้รับมอบหมาย
- ทำผลิตภัณฑ์ต้นแบบทั้งหมดไม่ว่าจะเป็นกรอบรีโมต, PCB ที่ใช้ในรีโมต จัดทำชุดแสดงสวิตช์ในส่วนของทีวีร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย
- เขียนโปรแกรมลงในไมโครคอนโทรลเลอร์ที่ใช้กับตัวรีโมตโดยยึดตามหลักการโปรโตคอลที่ได้ศึกษาไว้ข้างต้น เขียนโปรแกรมที่ใช้สำหรับสวิตช์บนส่วนของทีวีและทดลองใช้งาน

4.2 การวิเคราะห์และสรุปผลต่าง ๆ

จากการทำงานในแต่ละขั้นตอน สามารถสรุปและวิเคราะห์ผลการทำงานได้ดังนี้

4.2.1 ผลจากการพัฒนารีโมต

เริ่มจากการวางแผนทางของผลิตภัณฑ์นั้นร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมายจนเลือกได้ว่าจะใช้เทคโนโลยี Capacitive Touch Sensing มาใช้เป็นหลัก หลังจากนั้นจึงได้ไปศึกษาและเข้าอบรมกับบริษัทภายนอกในเรื่อง Capacitive Touch Sensing จากนั้นจึงได้ดีไซน์ตัวรีโมตร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย จากนั้นจึงจัดทำ Part List ทั้งหมดที่เราสามารถจะใส่ลงไปบนตัวรีโมต พร้อม ๆ กันนั้นได้ศึกษาในเรื่องสิทธิบัตรซึ่งก็พบว่าการพัฒนาผลิตภัณฑ์ที่กำลังจะทำนั้น

ทับซ้อนกับสิทธิบัตรหลายฉบับ จึงต้องปรับเปลี่ยนแนวทางที่จะทำผลิตภัณฑ์ขึ้นมาใหม่ ทำแผ่น PCB ที่จะนำมาเป็นเซนเซอร์บนตัวรีโมต หลังจากนั้นจึงลงมือเขียน Code ด้วยภาษาซีเพื่อใช้ในควบคุมหลอด IR LED ที่ใช้ในการส่งสัญญาณตามโปรโตคอลของพานาโซนิคที่มีรูปแบบการส่งของสัญญาณที่แตกต่างกันไปตามแต่ฟังก์ชัน โดยสามารถส่งงานโหมดต่างๆ ของทีวีได้ ไม่ว่าจะเป็น เปิด/ปิดทีวี เพิ่ม/ลด ช่อง เพิ่ม/ลด เสียง โหมด 3D (ในทีวีที่เป็นรุ่น 3 มิติ) และอื่น ๆ ทั้งนี้เนื่องจากข้อมูลของดีไซน์ ทั้งฮาร์ดแวร์ และ ซอฟต์แวร์ รวมทั้งโปรโตคอลของพานาโซนิคนั้นเป็นความลับของบริษัทพานาโซนิค ทางผู้จัดทำจึงต้องขอสงวนข้อมูลนี้ไว้เป็นความลับ

4.2.2 ผลจากการพัฒนาผลิตภัณฑ์ส่วนของตัวทีวี

เริ่มจากการวางแผนทางของผลิตภัณฑ์ร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมายจนเลือกได้ว่าจะใช้เทคโนโลยี Capacitive Touch Sensing มาใช้เป็นหลัก หลังจากนั้นจึงได้ไปศึกษาและเข้าอบรมกับบริษัทภายนอกในเรื่อง Capacitive Touch Sensing จากนั้นจึงได้ดีไซน์ส่วนของสวิตช์ที่จะใช้บนตัวทีวีร่วมกันกับพนักงานคนอื่น ๆ ที่ได้รับมอบหมาย ศึกษาในเรื่องสิทธิบัตรซึ่งก็พบว่าการพัฒนาผลิตภัณฑ์ที่กำลังจะทำนั้นทับซ้อนกับสิทธิบัตรหลายฉบับ เช่นเดียวกันกับรีโมต ทำแผ่น PCB ที่จะนำมาเป็นเซนเซอร์ หลังจากนั้นจึงลงมือเขียน Code ด้วยภาษาซีเพื่อใช้ในควบคุม จากนั้นพนักงานอีกคนจะเป็นคนทำส่วนติดต่อกับผู้ใช้นคอมพิวเตอร์ เนื่องจากข้อมูลของดีไซน์ ทั้งฮาร์ดแวร์ และ ซอฟต์แวร์ รวมทั้งโปรโตคอลของพานาโซนิคนั้นเป็นความลับของบริษัทพานาโซนิค ทางผู้จัดทำจึงต้องขอสงวนข้อมูลนี้ไว้เป็นความลับเช่นกัน

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

หลังจากที่ได้ทำโครงงานนี้แล้วพบว่ารีโมตสามารถทำงานด้วยระบบสัมผัสได้อย่างถูกต้อง สามารถสั่งการทีวีได้เหมือนรีโมตเดิมทุกประการ แต่เป็นระบบสัมผัสซึ่งก็สามารถสร้างความรู้สึกที่แตกต่างให้กับผู้ใช้ได้ แต่รีโมตที่ทำขึ้นมานั้น ยังสามารถที่จะใส่ลูกเล่นเข้าไปเพื่อสร้างความประทับใจให้กับผู้ใช้ได้อีก ซึ่งก็ไม่ใช่เรื่องยากที่จะทำเพราะคอนเซ็ปต์การทำงานนั้นยังอยู่ในลักษณะเดิม หากแต่เป็นวิธีการวาง Lay out ของ PCB ที่จะทำให้ลักษณะการใช้งานนั้น สนุกมากขึ้น และถ้าหากต้องการทำปุ่มให้เพิ่มมากขึ้นกว่านี้ เนื่องจาก คอมไพเลอร์นั้นเป็นตัวฟรี ทำให้ขนาดของไฟล์ที่จะทำการเบิร์นลงไปในไมโครคอนโทรลเลอร์นั้นใหญ่กว่าเวอร์ชันเต็มราวๆ สองเท่า จึงต้องทำการ Optimize Code เพื่อให้พอใช้งานในหน่วยความจำ

สำหรับสวิตช์บนตัวทีวีนั้นสามารถส่งข้อมูลผ่านทางสาย RS-232 เข้าสู่ PC และไปติดต่อกับ Program Interface TV ที่ทำด้วยโปรแกรม Visual Basic ได้หากแต่เมื่อนำไปใช้งานจริงเมื่อติดตั้งไปบนทีวี จะพบสัญญาณรบกวนเป็นจำนวนมากจากวงจรที่อยู่ในทีวีเองจึงต้องหาวิธีการที่จะป้องกันการรบกวนเพื่อให้สามารถส่งสัญญาณได้อย่างถูกต้อง

5.2 สรุปผลการปฏิบัติงาน

5.2.1 สรุปผลจากการศึกษาการวางแผนทางผลิตภัณฑ์ที่ต้องการ

ได้ผลการผลิตภัณฑ์ที่ใช้ได้ง่ายขึ้นและสร้างความรู้สึกที่แตกต่างให้ผู้ใช้ แต่ทั้งนี้ทั้งนั้น ทางผู้จัดทำยังคิดว่าผลิตภัณฑ์ตัวนี้ยังสามารถที่จะพัฒนาไปได้อีก ไม่ว่าจะเป็นด้วยการพัฒนาวิธีการใช้ หรือ การใส่เทคโนโลยีเพิ่มเข้าไปเพื่อให้มีผลิตภัณฑ์ที่มีความน่าสนใจมากยิ่งขึ้น

5.2.2 สรุปผลจากการศึกษาเรื่องสิทธิบัตร

ผลจากการที่ได้ศึกษาสิทธิบัตรพบว่าเรื่องสิทธิบัตรนั้น ทำให้การสร้างผลิตภัณฑ์นั้น ยากลำบากมากยิ่งขึ้น เนื่องจากผู้ที่จบสิทธิบัตรบางรายมีการจดทะเบียนโดยใช้ถ้อยคำที่คลุมเครือ จึง สุ่มเสี่ยงที่จะเกิดปัญหา ทำให้ผู้จัดทำเกิดความลังเลว่าจะทำแบบไหนได้บ้าง แบบไหนไม่ได้บ้าง

5.2.3สรุปผลจากการลงมือสร้างผลิตภัณฑ์ใหม่

จากผลการศึกษาโปรโตคอลที่ใช้สื่อสารระหว่างรีโมตและทีวีพานาโซนิค ทำให้ผู้จัดทำ สามารถเชื่อมต่อสื่อสารระหว่างรีโมตและทีวีได้อย่างถูกต้องตามคำสั่งที่ต้องการจะให้ เป็น เนื่องจากข้อมูลโปรโตคอลที่ใช้เป็นความลับของบริษัท จึงทำให้การศึกษาค้นหาข้อมูลนั้นลำบากขึ้น แต่ก็ไม่ใช่เรื่องใหญ่ ยังคงศึกษาได้

จากการศึกษาและเข้าอบรมเกี่ยวกับหลักการทำงานของ Capacitive Touch Sensingจากผล การศึกษาทำให้ผู้จัดทำสามารถทำผลิตภัณฑ์ขึ้นมาโดยมีลักษณะการใช้งานเป็นระบบสัมผัสได้และ เป็นไปตามแนวทางของผลิตภัณฑ์ที่ตั้งไว้ตอนแรกได้

จากการจัดทำ Part List ที่ที่เป็นไปได้ที่จะใช้ในรีโมตได้รายการชิ้นส่วนที่เป็นไปได้จะใส่ ลงไปในผลิตภัณฑ์ หากต้องการพัฒนาผลิตภัณฑ์ในรูปแบบใหม่ๆ ก็สามารถดึงชิ้นส่วนที่อยู่ใน PartList มาเปรียบเทียบ และตัดสินใจได้ ไม่ว่าจะเป็นเรื่องราคา spec เป็นต้น

จากการคิด interface บนทีวีที่ใช้ร่วมกันกับรีโมตและสวิตช์แบบใหม่ทำให้ได้ concept ของ interface อย่างคร่าวๆซึ่งสอดคล้องกับวิธีการใช้งานของผลิตภัณฑ์ที่ทำขึ้นมา

หลังจากลงมือสร้างผลิตภัณฑ์ขึ้นมาแล้วนั้น ส่วนของตัวรีโมตสามารถนำไปใช้งานได้แทน ผลิตภัณฑ์เดิมทุกประการได้ ในส่วนของสวิตช์นั้นก็สามารถส่งข้อมูลไปใช้งานสื่อสารกับ คอมพิวเตอร์ได้

5.2.4 ทดลองใช้งานและปรับปรุงแก้ไขเพิ่มเติม

จากการทดลองใช้งานในครั้งแรกพบว่าในบางครั้งมีการกวนสัญญาณกันเองระหว่างปุ่มทำ ให้สัญญาณที่ปล่อยออกมาจากตัวรีโมตนั้นไม่ถูกต้อง จึงต้องปรับแก้ไข Lay out บนตัว PCB ใหม่

เพื่อให้สัญญาณกวนกัน หลังจากนั้นได้ทดลองใช้งาน ก็พบว่ารีโมทที่ทำขึ้นมานั้น มีการกวนสัญญาณกันบริเวณสายไฟที่โยงออกมาจากแผ่น PCB จึงได้ทำการปรับการวางสายไฟใหม่ จึงสามารถทำงานได้อย่างปกติ

สำหรับในส่วนของสวิตช์บนตัวที่วินั้น ก็ได้มีการปรับแก้ไขตัวแผ่น PCB อยู่หลายครั้ง เนื่องจาก ขนาดที่ไม่เหมาะสมนิ้วมือบ้าง ข้อมูลที่ได้จากการส่งข้อมูลนั้นไม่ถูกต้องบ้าง จึงต้องเพิ่มลดขนาดของแผ่น PCB รวมทั้งลายทองแดงด้วย

5.3 ปัญหาและอุปสรรค

5.3.1 ในตอนแรก ผู้จัดทำไม่มีความรู้เรื่อง Capacitive Touch Sensing มาตั้งแต่ต้นทำให้ต้องใช้เวลาศึกษาเพิ่มเติมพอสมควร

5.3.2 ข้อจำกัดเรื่องสิทธิบัตร ทำให้การพัฒนาผลิตภัณฑ์มีข้อจำกัดมากขึ้น

5.3.3 มีการกวนกันของสัญญาณระหว่างปุ่ม ทำให้ต้องปรับแก้ Lay out PCB อยู่หลายครั้ง

5.4 ข้อเสนอแนะ

หลังจากการทำโครงการนี้ พบว่าผลิตภัณฑ์ที่ทำเสร็จในปัจจุบันนี้ ยังสามารถใส่เทคโนโลยีอื่นๆ และ รูปแบบวิธีการใช้ ให้มีความน่าสนใจมากยิ่งขึ้นได้อีก

โปรแกรมที่เขียนและบรรจุอยู่ในไมโครคอนโทรลเลอร์นั้นยังสามารถย่อขนาดของโปรแกรมได้อีก

เอกสารอ้างอิง

Tom Perme Microchip Technology Inc., 2007, Introduction to Capacitive Sensing. [Online].

เข้าถึงได้จาก

<http://ww1.microchip.com/downloads/en/AppNotes/01101a.pdf>

(วันที่สืบค้นข้อมูล : 11กรกฎาคม 2554).

Tom Perme Microchip Technology Inc., 2007, Layout and Physical Design Guidelines for Capacitive Sensing. [Online]. เข้าถึงได้จาก

<http://ww1.microchip.com/downloads/en/AppNotes/01102a.pdf>

(วันที่สืบค้นข้อมูล : 11กรกฎาคม 2554).

Thomas Perme and Dieter Peter Microchip Technology Inc., 2009, Capacitive Touch Using Only an ADC (“CVD”). [Online] เข้าถึงได้จาก

<http://ww1.microchip.com/downloads/en/AppNotes/01298A.pdf>

(วันที่สืบค้นข้อมูล : 11กรกฎาคม 2554).

Bruce Bohn Microchip Technology Inc. 2009, Microchip CTMU for Capacitive Touch Application.

[Online] เข้าถึงได้จาก <http://ww1.microchip.com/downloads/en/AppNotes/01250a.pdf>

(วันที่สืบค้นข้อมูล : 11กรกฎาคม 2554).

Thomas Perme and Steven Lin Microchip Technology Inc, 2009, Water-Resistant Capacitive

Sensing. [Online] เข้าถึงได้จาก <http://ww1.microchip.com/downloads/en/AppNotes/01286A.pdf>

(วันที่สืบค้นข้อมูล : 11กรกฎาคม 2554).

Keith Curtis and Dieter Peter Microchip Technology Inc, 2010, mTouch[®] Metal Over Cap Technology. [Online] เข้าถึงได้จาก

<http://ww1.microchip.com/downloads/en/AppNotes/01325A.pdf>

(วันที่สืบค้นข้อมูล : 11กรกฎาคม 2554).

Burke Davison Microchip Technology Inc, 2010, Techniques for Robust Touch Sensing Design.

[Online]

เข้าถึงได้จาก <http://ww1.microchip.com/downloads/en/AppNotes/01334A.pdf>

(วันที่สืบค้นข้อมูล : 11 กรกฎาคม 2554).

Padmaraja Yedamale and Jim Bartling Microchip Technology Inc, 2010, See What You Can Do with the CTMU. [Online] เข้าถึงได้จาก

<http://ww1.microchip.com/downloads/en/AppNotes/CTMU%2001375a.pdf>

(วันที่สืบค้นข้อมูล : 11 กรกฎาคม 2554).

Microchip Technology Inc, 2009, PICkit 3 Programmer/Debugger User's Guide. [Online]

เข้าถึงได้จาก

http://www.sparkfun.com/datasheets/Programmers/PICkit_3_User_Guide_51795A.pdf

(วันที่สืบค้นข้อมูล : 11 กรกฎาคม 2554).

Microchip Technology Inc, 2009, PIC16(L)F1826/27 Data Sheet. [Online] เข้าถึงได้จาก

<http://ww1.microchip.com/downloads/en/DeviceDoc/41391D.pdf>

(วันที่สืบค้นข้อมูล : 11 กรกฎาคม 2554).

Thaiio.com, 2000, USB Source. [Online] เข้าถึงได้จาก <http://www.thaiio.com/usb.html>

(วันที่สืบค้นข้อมูล : 7 กรกฎาคม 2554).

Thaiio.com, 2000, Serial Port Source. [Online] เข้าถึงได้จาก

<http://www.thaiio.com/serialport.html>

(วันที่สืบค้นข้อมูล : 7 กรกฎาคม 2554).

Banahan, M.; Brady, D.; Doran, M. (1991). *The C Book* (2nd ed.). Addison-Wesley. [Online]

เข้าถึงได้จาก http://publications.gbdirect.co.uk/c_book/.

(วันที่สืบค้นข้อมูล : 8 กรกฎาคม 2554).



ภาคผนวก ก.

ความรู้เพิ่มเติม



1. ประวัติภาษาซี

การพัฒนาช่วงแรก

การเริ่มต้นพัฒนาภาษาซีเกิดขึ้นที่เบลล์แล็บส์ของเอทีแอนด์ทีระหว่าง พ.ศ. 2512–2516^[2] แต่ตามข้อมูลของริตชี ช่วงเวลาที่เกิดความสร้างสรรค์มากที่สุดคือ พ.ศ. 2515 ภาษานี้ถูกตั้งชื่อว่า "ซี" เพราะคุณลักษณะต่าง ๆ ต่อยอดมาจากภาษาก่อนหน้าคือ "บี" ซึ่งจากข้อมูลของเคน ทอมป์สัน (Ken Thompson) กล่าวว่าภาษาบีเป็นรุ่นที่แยกตัวออกจากภาษาบีซีพีแอลอีกทอดหนึ่ง

จุดเริ่มต้นของภาษาซีผูกอยู่กับการพัฒนาระบบปฏิบัติการยูนิกซ์อย่างใกล้ชิด ซึ่งเดิมพัฒนาด้วยภาษาแอสเซมบลีบนหน่วยประมวลผลพีดีพี-7 โดยริตชีและทอมป์สัน โดยผสมผสานความคิดหลากหลายจากเพื่อนร่วมงาน ในตอนท้ายพวกเขาตัดสินใจที่จะย้ายระบบปฏิบัติการนั้นลงในพีดีพี-11 แต่ภาษาบีขาดความสามารถบางอย่างที่จะใช้คุณลักษณะอันได้เปรียบของพีดีพี-11 เช่น ความสามารถในการระบุตำแหน่งที่อยู่เป็น ไบต์ จึงทำให้เกิดการพัฒนาภาษาซีรุ่นแรกขึ้นมา

รุ่นดั้งเดิมของระบบยูนิกซ์บนพีดีพี-11ถูกพัฒนาขึ้นด้วยภาษาแอสเซมบลี เมื่อประมาณ พ.ศ. 2516 ภาษาซีเพิ่มชนิดข้อมูล struct ทำให้ภาษาซีเพียงพออย่างมีประสิทธิภาพ ซึ่งเคอร์เนलयูนิกซ์ส่วนใหญ่ถูกเขียนด้วยภาษาซี นี่ก็จะเป็นเคอร์เนลหนึ่งของระบบปฏิบัติการที่พัฒนาด้วยภาษาอื่นนอกเหนือจากภาษาแอสเซมบลี (ระบบอื่นเช่นมัลดิกซ์เขียนด้วยภาษาพีแอล/วัน เอ็มซีพีสำหรับเบอร์โรส์ ปี5000เขียนด้วยภาษาอัลกอล ในปี พ.ศ. 2504)

ภาษาเคแอนด์อาร์ซี

เมื่อ พ.ศ. 2521 ไบรอัน เคอร์นิกัน (Brian Kernighan) และเดนนิส ริตชี ได้ตีพิมพ์หนังสือเล่มแรกชื่อ *เดอะซีโปรแกรมมิงแลงกวิจ (The C Programming Language)*^[9] ซึ่งเป็นที่รู้จักในกลุ่มโปรแกรมเมอร์ภาษาซีว่า "เคแอนด์อาร์" (K&R อักษรย่อของผู้แต่งทั้งสอง) หนังสือเล่มนี้ทำหน้าที่เป็นข้อกำหนดของภาษาอย่างไม่เป็นทางการมาหลายปี ภาษาซีรุ่นดังกล่าวจึงมักถูกอ้างถึงว่าเป็น **ภาษาเคแอนด์อาร์ซี (K&R C)** ส่วนหนังสือที่ปรับปรุงครั้งที่สองครอบคลุมมาตรฐาน เอเนซีซี ที่มีขึ้นทีหลัง

ภาษาเคแอนด์อาร์ซีได้แนะนำคุณลักษณะหลายประการเช่น

- ไบบรีไอ/โอมาตรฐาน

- ชนิดข้อมูล long int (จำนวนเต็มขนาดยาว)
- ชนิดข้อมูล unsigned int (จำนวนเต็มไม่มีเครื่องหมาย)
- ตัวดำเนินการกำหนดค่าแบบประสมในรูปแบบ =ตัวดำเนินการ (เช่น -=) ถูกเปลี่ยนเป็น ตัวดำเนินการ= (เช่น -=) เพื่อลดปัญหาความกำกวมเชิงความหมาย อย่างเช่นกรณี i=-10 ซึ่งจะถูกต้องความว่า i = -10 แทนที่จะเป็นอย่างที่ตั้งใจคือ i = -10

แม้ว่าหลังจากการเผยแพร่มาตรฐานของภาษาซีเมื่อ พ.ศ. 2532 ภาษาเคแอนด์อาร์ซีถูกพิจารณาว่าเป็น "ส่วนร่วมต่ำสุด" อยู่เป็นเวลาหลายปี (ความสามารถในการแปลรหัสจำนวนหนึ่งเป็นคำสั่งซึ่งทำงานได้บนเครื่องใดก็ตามเป็นอย่างน้อย) ซึ่งโปรแกรมเมอร์ภาษาซีต้องจำกัดความสามารถของพวกเขาในกรณีที่ต้องการให้ระบบสามารถใช้ได้กับหลายเครื่องมากที่สุด เนื่องจากตัวแปลโปรแกรมเก่า ๆ ก็ยังคงมีการใช้งานอยู่ และการเขียนภาษาซีแบบเคแอนด์อาร์ซีอย่างระมัดระวังสามารถเข้ากันได้กับภาษาซีมาตรฐานเป็นอย่างดี

ในภาษาซีรุ่นแรก ๆ เฉพาะฟังก์ชันที่คืนค่าไม่เป็นจำนวนเต็ม จำเป็นต้องประกาศไว้ก่อนการนิยาม ฟังก์ชันหากมีการเรียกใช้ อีกนัยหนึ่งคือ ฟังก์ชันที่ถูกเรียกใช้โดยไม่มีการประกาศมาก่อน ถือว่าฟังก์ชันนั้นจะคืนค่าเป็นจำนวนเต็มหากค่าของมันถูกใช้งาน ตัวอย่างเช่น

```
long int SomeFunction();
/* int OtherFunction(); */

/* int */ CallingFunction()
{
    long int test1;
    register /* int */ test2;

    test1 = SomeFunction();
    if (test1 > 0)
        test2 = 0;
    else
        test2 = OtherFunction();
```



```

return test2;
}

```

จากตัวอย่างข้างต้น การประกาศ `int` ที่ถูกคัดลอก สามารถละเว้นได้ในภาษาเคแอลดีอาร์ซี แต่ `long int` จำเป็นต้องประกาศ

การประกาศฟังก์ชันของภาษาเคแอลดีอาร์ซีไม่มีการระบุข้อมูลเกี่ยวกับอาร์กิวเมนต์ที่ใช้ ดังนั้นจึงไม่มีการตรวจชนิดข้อมูลพารามิเตอร์ของฟังก์ชัน แม้ว่าตัวแปลโปรแกรมบางตัวจะแสดงข้อความเตือน ถ้าฟังก์ชันถูกเรียกใช้ภายใน โดยมีจำนวนอาร์กิวเมนต์ที่ผิด หรือถ้าฟังก์ชันถูกเรียกใช้หลายครั้งจากภายนอก โดยมีชนิดข้อมูลของอาร์กิวเมนต์ต่างกัน เครื่องมือภายนอกอาทิ ลินต์ (lint) ของยูนิกซ์ถูกพัฒนาขึ้นเพื่อให้สามารถตรวจสอบความคงเส้นคงวาของฟังก์ชันที่ใช้งานข้ามไฟล์รหัสต้นฉบับหลายไฟล์

หลายปีถัดจากการเผยแพร่ภาษาเคแอลดีอาร์ซี คุณลักษณะที่ไม่เป็นทางการหลายอย่างก็ถูกเพิ่มเข้ามาในภาษา ซึ่งรองรับโดยตัวแปลโปรแกรมจากเอทีแอนด์ทีและผู้ผลิตรายอื่น คุณลักษณะที่เพิ่มเหล่านี้เช่น

- ฟังก์ชัน `void`
- ฟังก์ชันที่คืนค่าเป็นชนิดข้อมูล `struct` หรือ `union` (แทนที่จะเป็นตัวชี้)
- การกำหนดค่าให้กับชนิดข้อมูล `struct`
- ชนิดข้อมูลแฉงนับ (enumerated type)

ส่วนขยายที่เพิ่มขึ้นอย่างมากและการขาดข้อตกลงในเรื่องไลบรารีมาตรฐาน อีกทั้งความนิยมในภาษาและข้อเท็จจริงที่ว่าไม่เพียงแต่ตัวแปลโปรแกรมยูนิกซ์เท่านั้นที่พัฒนาขึ้นตามข้อกำหนดของเคแอลดีอาร์ซี ทั้งหมดนำไปสู่ความสำคัญของการทำให้เป็นมาตรฐาน

ภาษาเอนซีซีและภาษาไอโซซี

ช่วงพุทธทศวรรษ 2520 ภาษาซีหลายรุ่นถูกพัฒนาขึ้นสำหรับเมนเฟรมคอมพิวเตอร์ มินิคอมพิวเตอร์ และ ไมโครคอมพิวเตอร์อย่างกว้างขวางรวมทั้งไอบีเอ็มพีซี ซึ่งความนิยมของมันเริ่มเพิ่มขึ้นอย่างมีนัยสำคัญ

เมื่อ พ.ศ. 2526 สถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา (ANSI) ได้ก่อตั้ง คณะกรรมการ เอกซ์เจ11 ขึ้นมาเพื่อกำหนดมาตรฐานของภาษาซี ต่อมา พ.ศ. 2532 มาตรฐานดังกล่าวได้รับการอนุมัติเป็น ANSI X3.159-1989 "Programming Language C" ซึ่งภาษารุ่นนี้มักถูกอ้างถึงว่าเป็นภาษาแอนซีซี (ANSI C) ภาษาซีมาตรฐาน หรือภาษาซี89 (C89) ในบางครั้ง

เมื่อ พ.ศ. 2533 องค์การระหว่างประเทศว่าด้วยการมาตรฐาน (ISO) ได้รับเอามาตรฐาน แอนซีซี (พร้อมการเปลี่ยนแปลงการจัดรูปแบบ) มาเป็น ISO/IEC 9899:1990 ซึ่งบางครั้งก็ถูกเรียกว่าภาษาไอโซซี (ISO C) หรือภาษาซี90 (C90) ดังนั้นคำว่า "ซี89" กับ "ซี90" จึงหมายถึงภาษาโปรแกรมเดียวกัน

แอนซีไม่ได้พัฒนามาตรฐานภาษาซีโดยเอกเทศอีกต่อไปแล้ว เหมือนเช่นองค์กรมาตรฐานแห่งชาติอื่น ๆ แต่ก็ค่อยตามมาตรฐานไอโซซี การรับเอามาตรฐานระดับชาติมาปรับปรุงเป็นมาตรฐานระดับสากล เกิดขึ้นภายในปีเดียวกับที่เผยแพร่มาตรฐานไอโซ

จุดมุ่งหมายหนึ่งของกระบวนการสร้างมาตรฐานให้ภาษาซีคือเพื่อสร้างซูเปอร์เซตของ ภาษาเคแอนด์อาร์ซี ผสมผสานคุณลักษณะต่าง ๆ ที่ยังไม่เป็นทางการซึ่งแนะนำต่อกันมา คณะกรรมการมาตรฐานได้รวมคุณลักษณะหลายประการเพิ่มเข้ามา อาทิ ฟังก์ชัน โพรโทไทป์ (ยืมมาจากภาษาซีพลัสพลัส), ตัวชี้ void, รองรับการจัดเรียงท้องถิ่น (locale) และชุดอักขระสากล, และการปรับปรุงตัวประมวลก่อนให้ดีขึ้น วากยสัมพันธ์สำหรับการประกาศพารามิเตอร์ถูกเพิ่มเข้ามาให้เหมือนกับรูปแบบที่ใช้ในภาษาซีพลัสพลัส แม้ว่าการเขียนแบบเคแอนด์อาร์ซีก็ยังสามารถใช้ได้เพื่อความเข้ากันได้กับรหัสต้นฉบับที่มีอยู่แล้ว

ภาษารุ่นนี้ยังคงรองรับในตัวแปลโปรแกรมในปัจจุบัน และรหัสภาษาซีส่วนใหญ่ที่เขียนขึ้นทุกวันนี้ก็ใช้พื้นฐานมาจากรุ่นนี้ โปรแกรมใด ๆ ที่เขียนขึ้นด้วยภาษาซีมาตรฐานโดยไร้สมมติฐานว่าขึ้นอยู่กับฮาร์ดแวร์ใด จะทำงานได้อย่างถูกต้องบนแพลตฟอร์มใดก็ตามด้วยการพัฒนาภาษาซีที่สอดคล้องกันภายในทรัพยากรที่จำกัด หากไม่ระมัดระวังเช่นนั้น โปรแกรมอาจแปลได้เฉพาะบนแพลตฟอร์มหนึ่งหรือด้วยตัวแปลตัวหนึ่งเท่านั้น อันเนื่องมาจากการใช้ไลบรารีไม่มาตรฐานเช่นไลบรารีส่วนต่อประสานกราฟิกกับผู้ใช้ที่ดี หรือความเชื่อมั่นต่อสมบัติเฉพาะของ

แพลตฟอร์มหรือตัวแปลหนึ่ง ๆ เช่นขนาดที่แท้จริงของชนิดข้อมูลหรือการลำดับข้อมูลไบต์ (endianness) ก็ดี

ในกรณีที่ต้องเลือกว่ารหัสต้องถูกแปลด้วยตัวแปลภาษาซีมาตรฐานหรือภาษาเคแอลดีอาร์ ซีอย่างใดอย่างหนึ่ง การใช้แมโคร `__STDC__` สามารถช่วยให้แบ่งแยกรหัสส่วนมาตรฐานและส่วนเคแอลดีอาร์ออกจากกัน ซึ่งเป็นคุณลักษณะที่ได้เปรียบอีกอย่างหนึ่งที่มีเฉพาะในภาษาซีมาตรฐาน

ภาษาซี99

หลังจากกระบวนการทำให้เป็นมาตรฐานของแอนซี/ไอโซแล้ว ข้อกำหนดภาษาซียังคงนิ่งอยู่ชั่วระยะเวลาหนึ่ง ในขณะที่ภาษาซีพลัสพลัสกำลังก่อตัวด้วยความพยายามทำให้เป็นมาตรฐานของมันเอง การเพิ่มเติมกฎเกณฑ์ครั้งที่ 1 สำหรับมาตรฐานภาษาซีเผยแพร่เมื่อ พ.ศ. 2538 เพื่อแก้ไขรายละเอียดบางจุดและเพิ่มการรองรับชุดอักขระสากลให้มากขึ้น ต่อมามาตรฐานภาษาซีถูกเรียบเรียงดัดแปลงใหม่และนำไปสู่การเผยแพร่ ISO/IEC 9899:1999 ออกสู่สาธารณะชนใน พ.ศ. 2542 ซึ่งมักถูกอ้างถึงว่า "ซี99" (C99) มาตรฐานนี้มีการเพิ่มเติมกฎเกณฑ์แล้วสามครั้ง โดย Technical Corrigenda ปัจจุบันมาตรฐานภาษาซีสากลดูแลและควบคุมโดยกลุ่ม ISO/IEC JTC1/SC22/WG14

ภาษาซี99ได้แนะนำคุณลักษณะใหม่หลายประการ อาทิ ฟังก์ชันแบบแทรก (inline function) ชนิดข้อมูลใหม่หลายชนิด (เช่น `long long int` และ `complex` สำหรับจำนวนเชิงซ้อน) แลวลำดับความยาวแปรได้ (variable-length array) แมโครอาร์กิวเมนต์แปรได้ (variadic macro) และหมายเหตุในหนึ่งบรรทัดที่ขึ้นต้นด้วย `//` เหมือนภาษาพีซีพีแอลหรือภาษาซีพลัสพลัส ซึ่งคุณลักษณะส่วนใหญ่เคยพัฒนาไว้แล้วเป็นส่วนขยายของตัวแปลภาษาซีหลายโปรแกรม

ภาษาซี99สามารถเข้ากันได้แบบย้อนหลังกับภาษาซี90เป็นส่วนใหญ่ แต่ก็จำกัดมากขึ้นในบางแง่มุม โดยเฉพาะการประกาศโดยไม่ระบุชนิด จะไม่ถูกสมมติว่าเป็น `int` อีกต่อไป แมโครมาตรฐาน `__STDC_VERSION__` ถูกนิยามขึ้นด้วยค่า 199901L เพื่อแสดงว่ารหัสนั้นรองรับภาษาซี99 ขณะนี้ จีซีซี ซันสตูดิโอ และตัวแปลโปรแกรมอื่น ๆ ก็รองรับคุณลักษณะใหม่ของภาษาซี99เป็นจำนวนมากหรือทั้งหมดแล้ว

ภาษาซี1เอกซ์

เมื่อ พ.ศ. 2550 มีกลุ่มทำงานหนึ่งเริ่มต้นขึ้นเพื่อปรับปรุงมาตรฐานภาษาซีอีกรุ่น ซึ่งเรียกชื่ออย่างไม่เป็นทางการว่า "ซี1เอกซ์" (C1X) คณะกรรมการนี้รับเอาแนวคิดต่าง ๆ เพื่อจำกัดการเลือกคุณลักษณะใหม่ที่ยังไม่เคยมีการทดสอบพัฒนามาก่อน

การใช้งาน

การเขียนโปรแกรมระบบเป็นการใช้งานหลักของภาษาซี ซึ่งรวมไปถึงการพัฒนา ระบบปฏิบัติการและโปรแกรมประยุกต์ระบบฝังตัว เนื่องจากลักษณะเฉพาะอันเป็นที่ต้องการถูกรวมเข้าไว้ด้วยกัน อย่างเช่น ความสามารถในการเคลื่อนย้ายได้กับประสิทธิภาพของรหัสต้นฉบับ ความสามารถในการเข้าถึงที่อยู่ของฮาร์ดแวร์ที่ระบุ ความสามารถเรื่อง *type punning* เพื่อให้เข้ากับความ ต้องการการเข้าถึงข้อมูลที่กำหนดไว้จากภายนอก และความต้องการทรัพยากรระบบขณะทำงานต่ำ ภาษาซีสามารถใช้เขียนโปรแกรมเว็บไซต์โดยใช้ซีจีไอเป็น "เกตเวย์" เพื่อแลกเปลี่ยนสารสนเทศระหว่างเว็บแอปพลิเคชัน เซิร์ฟเวอร์ และเบราว์เซอร์^[10] ปัจจุบันบางอย่างไม่ให้เลือก ภาษาซีแทนที่จะเป็นภาษาอินเทอร์พรีเตอร์ คือความเร็ว เสถียรภาพ และความอ่อนไหวต่อการเปลี่ยนแปลงในสภาพแวดล้อมของการดำเนินงาน เนื่องจากเป็นธรรมชาติของภาษาคอมไพเลอร์

ผลจากการยอมรับในระดับกว้างขวางและประสิทธิภาพของภาษาซี ทำให้ ตัวแปลโปรแกรม ตัวแปลคำสั่ง ไบรารีต่าง ๆ ของภาษาอื่น มักพัฒนาขึ้นด้วยภาษาซี ตัวอย่างเช่น ตัวแปลโปรแกรมภาษาไอเฟลหลายโปรแกรมส่งข้อมูลออกเป็นรหัสภาษาซีเป็นภาษากลาง เพื่อส่งต่อไปให้ตัวแปลโปรแกรมภาษาซีต่อไป การพัฒนาสายหลักของภาษาไพทอน ภาษาเพิร์ล 5 และภาษาพีเอชพี ทั้งหมดถูกเขียนขึ้นด้วยภาษาซี

ภาษาซีมีประสิทธิภาพสำหรับคอมพิวเตอร์เพื่องานคำนวณและวิทยาศาสตร์ เนื่องจาก ความสิ้นเปลืองต่ำ ธรรมชาติของภาษาระดับต่ำ ธรรมชาติของภาษาที่ถูกแปล และมีส่วน คณิตศาสตร์ที่ดีในไลบรารีมาตรฐาน ตัวอย่างของการใช้ภาษาซีในงานคำนวณและวิทยาศาสตร์ เช่นจีเอ็มพี ไลบรารีวิทยาศาสตร์ของgnu แมเทอแมติกา แมตแล็บ และแซต

ภาษาซีบางครั้งใช้เป็นภาษาระหว่างกลางในการทำให้เกิดผลของภาษาอื่น แนวคิดนี้อาจใช้ เพื่อความสะดวกต่อการเคลื่อนย้าย โดยให้ภาษาซีเป็นภาษาระหว่างกลาง ซึ่งไม่จำเป็นต้องพัฒนาตัว สร้างรหัสแบบเจาะจงเครื่อง ตัวแปลโปรแกรมที่ใช้ภาษาซีในทางนี้เช่น บิตซี เกมบิต จีเอชซี สควิก และวาลา เป็นต้น อย่างไรก็ตามภาษาซีถูกออกแบบมาเพื่อเป็นภาษาเขียนโปรแกรม ไม่ใช่

ภาษาเป้าหมายของตัวแปลโปรแกรม จึงเหมาะสมน้อยกว่าสำหรับการใช้เป็นภาษาระหว่างกลาง ด้วยเหตุผลนี้นำไปสู่การพัฒนาภาษาระหว่างกลางที่มีพื้นฐานบนภาษาซีเช่น ภาษาซีไมน์สไมน์ส

ผู้ใช้งานปลายใช้ภาษาซีอย่างแพร่หลายเพื่อสร้างแอปพลิเคชันของผู้ใช้เอง แต่เมื่อแอปพลิเคชันใหญ่ขึ้น การพัฒนาเช่นนั้นมักจะย้ายไปทำในภาษาอื่นที่พัฒนามาด้วยกัน เช่นภาษาซีพลัสพลัส ภาษาซีชาร์ป ภาษาวิวอลเบสิก เป็นต้น



ประวัติผู้วิจัย

ชื่อ - นามสกุล	นายธราเทพ เปรมประสาทสิทธิ์
วัน เดือน ปีเกิด	3 ตุลาคม พ.ศ. 2532
ประวัติการศึกษา	
ระดับมัธยมศึกษา	มัธยมศึกษาตอนปลาย โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า พ.ศ.2550
ประวัติการฝึกอบรม	นักศึกษาแลกเปลี่ยน กับ Institute Of Technologist (Monotsukuri University) TESA TOPGUN RALLY 2009 ของ Thai Embedded System Association Robot Design Contest 2009 ของ ศูนย์เทคโนโลยีโลหะและวัสดุแห่งชาติ และ จุฬาลงกรณ์มหาวิทยาลัย

