

การศึกษาเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO
A Study of Automatic Blood Pressure Monitor by PICASO



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

การศึกษาเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO
A Study of Automatic Blood Pressure Monitor by PICASO



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ปีการศึกษา 2556 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A Study of Automatic Blood Pressure Monitor by PICASO



Wipawan Ketkarn
Sornsawan Wiroonphol
Nichaluk Wasanapongsopon

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LARDKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2556
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

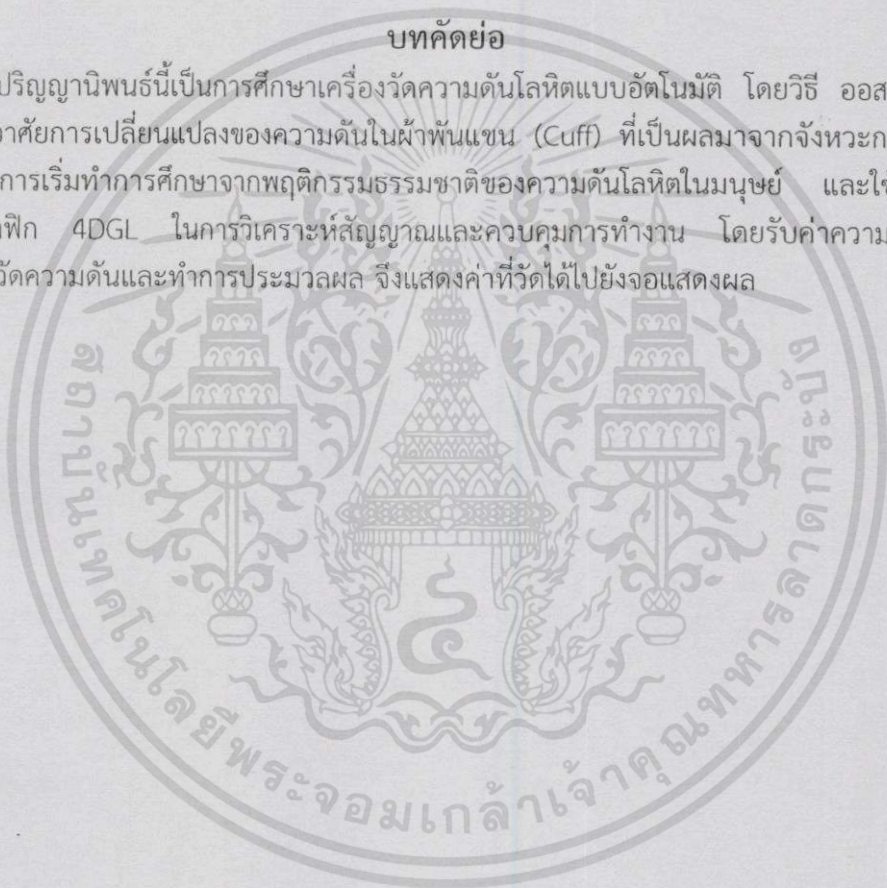
หัวข้อปริญญาานิพนธ์ การศึกษาเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO
A Study of Automatic Blood Pressure Monitor by PICASO
นักศึกษาผู้จัดทำ นางสาววิภาวรรณ เกตุการณ์ รหัสนักศึกษา 53011486
นางสาวศรสวรรค์ วิรุฬห์ผล รหัสนักศึกษา 53011525
นางสาวณิชาลักษณ์ วาสนาพงศ์โสภณ รหัสนักศึกษา 53011576
ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2556

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์วิริยะ กองรัตน์	

หัวข้อปริญญานิพนธ์	การศึกษาเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO		
	A Study of Automatic Blood Pressure Monitor by PICASO		
นักศึกษาผู้จัดทำ	นางสาววิภาวรรณ	เกตุการณ์	รหัสนักศึกษา 53011486
	นางสาวศรสวรรค์	วิรุฬห์ผล	รหัสนักศึกษา 53011525
	นางสาวณิชาลักษณ์	วาสนาพงศ์โสภณ	รหัสนักศึกษา 53011576
อาจารย์ที่ปรึกษา ปีการศึกษา	รองศาสตราจารย์วิริยะ	กองรัตน์	2556

บทคัดย่อ

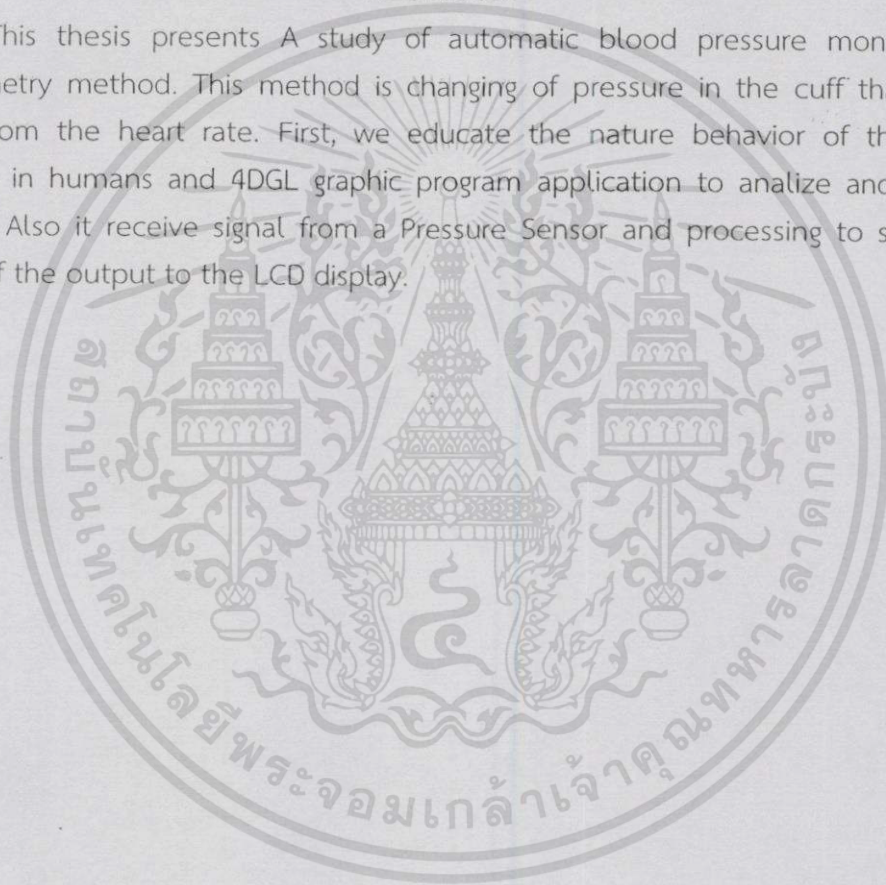
ปริญญานิพนธ์นี้เป็นการศึกษาเครื่องวัดความดันโลหิตแบบอัตโนมัติ โดยวิธี ออสซิลอเมตริก วิธีการนี้อาศัยการเปลี่ยนแปลงของความดันในผ้าพันแขน (Cuff) ที่เป็นผลมาจากจังหวะการเต้นของหัวใจโดยการเริ่มทำการศึกษากจากพฤติกรรมธรรมชาติของความดันโลหิตในมนุษย์ และใช้โปรแกรมภาษากราฟิก 4DGL ในการวิเคราะห์สัญญาณและควบคุมการทำงาน โดยรับค่าความดันมาจากเซนเซอร์วัดความดันและทำการประมวลผล จึงแสดงค่าที่วัดได้ไปยังจอแสดงผล



Thesis Title	A Study of Automatic Blood Pressure Monitor by PICASO	
Authors	Ms. Wipawan	Ketkarn
	Ms. Sornsawan	Wiroonphol
	Ms. Nichaluk	Wasanapongsopon
Thesis Advisor	Assoc. Prof. Viriya	Kongrat
Year	2013	

ABSTRACT

This thesis presents A study of automatic blood pressure monitor with Oscillometry method. This method is changing of pressure in the cuff that it is a result from the heart rate. First, we educate the nature behavior of the blood pressure in humans and 4DGL graphic program application to analyze and control working. Also it receive signal from a Pressure Sensor and processing to show the values of the output to the LCD display.



กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงได้ด้วยดีอย่างสมบูรณ์ โดยได้รับความอนุเคราะห์อย่างยิ่งจากรองศาสตราจารย์วิริยะ กองรัตน์ ที่ให้ความอนุเคราะห์ในการให้คำปรึกษาแนะนำอีกทั้งประโยชน์อื่นๆที่เป็นประโยชน์ต่อโครงการ รวมทั้งตรวจสอบแก้ไขข้อบกพร่องต่างๆในการทำงาน ทางคณะผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณบิดา มารดา ที่ท่านได้เลี้ยงดูและให้กำลังใจในการศึกษาค้นคว้าปริญญานิพนธ์

ขอขอบคุณเพื่อนที่คอยช่วยเหลือและให้คำปรึกษา ในการทดสอบการวัดความดันโลหิต

ขอขอบคุณเจ้าของบทความ ตำรา หนังสือทุกท่านที่ใช้ในการสืบค้นข้อมูล

สุดท้ายคุณค่าและประโยชน์จากการค้นคว้าของปริญญานิพนธ์เล่มนี้ ผู้จัดทำขอมอบเพื่อทดแทนบุญคุณต่อบิดามารดา และครูบาอาจารย์ทุกท่านที่ได้อบรมสั่งสอนและให้คำแนะนำตลอดมาตลอดจนผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ



สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII

บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	1
1.3 ขอบเขตของปริญญาโท.....	1
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎี.....	3
2.1 การวัดความดันโลหิต.....	3
2.1.1 การปฏิบัติก่อนการวัดความดันโลหิต.....	4
2.1.2 เทคนิคการวัดความดันที่ถูกต้อง.....	4
2.2 ทฤษฎีการวัดความดันโลหิต.....	5
2.2.1 วิธีการวัดความดันโลหิต.....	5
2.2.2 การวัดความดันโลหิตด้วยวิธีการออสซิลเลทหรือออสซิลโลเมตริก (Oscillometric).....	8
2.3 ส่วนของซอฟต์แวร์.....	10
2.3.1 PICASO – GFX2.....	10
2.4 คุณสมบัติของอุปกรณ์แต่ละตัว.....	12
2.4.1 ทรานสดิวเซอร์.....	12
2.4.2 อุปกรณ์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล MCP3424.....	14
2.4.3 ระบบรวบรวมข้อมูล และวิเคราะห์ข้อมูลโดยอาศัยคอมพิวเตอร์.....	15
2.4.3.1 คอมพิวเตอร์ส่วนบุคคล (PC).....	15
2.3.3.1.1 รับโปรแกรมและข้อมูล.....	15
2.3.3.1.2 ประมวลผล.....	15
2.3.3.1.3 แสดงผลลัพธ์.....	15
2.4.3.2 ทรานสดิวเซอร์.....	15
2.4.3.3 สัญญาณ.....	16
2.3.3.3.1 สัญญาณอนาล็อก.....	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.4.3.3.2 สัญญาณดิจิทัล.....	16
2.4.3.3.3 Signal Processing/Conditioning.....	16
2.4.3.4 Analog to Digital Conversion (A/D).....	16
2.4.3.4.1 DAQ Software.....	16
2.4.4 จอแสดงผล μ LCD – 43PCT (GFX).....	16
2.4.4.1 รายละเอียดของจอแสดงผล μ LCD – 43PCT (GFX).....	18
2.4.4.2 การสื่อสารระหว่างขาของอุปกรณ์.....	20
2.4.4.2.1 พอร์ตอนุกรม COM0, COM1 UARTS.....	20
2.4.4.2.2 GPIO – General Purpose IO Interface.....	22
2.4.4.2.3 System Pins.....	22
2.4.4.3 หน่วยความจำ.....	23
2.4.4.4 เสียง (Audio).....	23
2.4.4.5 ข้อความระวังสำหรับหน้าจอแสดงผล.....	23
2.4.5 4D Programming Cable.....	24
2.4.5.1 ข้อมูลเกี่ยวกับสายเคเบิล.....	24
2.4.5.2 ชนิดของฟังก์ชันการทำงาน.....	25
2.5 I ² C Bus (Inter Integrate Circuit Bus).....	25
2.5.1 การเชื่อมต่ออุปกรณ์แบบ I ² C Bus.....	25
2.5.2 การเขียน – อ่านข้อมูลกับอุปกรณ์แบบ I ² C Bus.....	26
2.5.3 สถานะบัสว่าง.....	26
2.5.4 การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I ² C Bus.....	26
2.5.5 รหัสควบคุมของ I ² C Bus.....	26
2.5.6 ช่วงเวลารับส่งบิตข้อมูลของ I ² C Bus.....	27
2.6 การสื่อสารแบบ SPI.....	27
บทที่ 3 วิธีการดำเนินงาน.....	29
3.1 การทำงานร่วมกับอุปกรณ์.....	29
3.2 ขั้นตอนการทำงานของโปรแกรม.....	30
3.2.1 โปรแกรมหลัก.....	31
3.2.2 โปรแกรมตรวจวัดความดันซิสโตลิก.....	34
3.2.3 โปรแกรมตรวจวัดความดันไดแอสโตลิก.....	35
3.2.4 โปรแกรมตรวจวัดอัตราการเต้นของหัวใจ.....	35
3.2.5 โปรแกรมตรวจวัดค่าเฉลี่ยความดันโลหิต.....	35
3.2.6 โปรแกรมแสดงค่าบนจอแสดงผล.....	35

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

3.3 วิธีการทดสอบการทำงานของเครื่องวัดความดันโลหิต..... 40

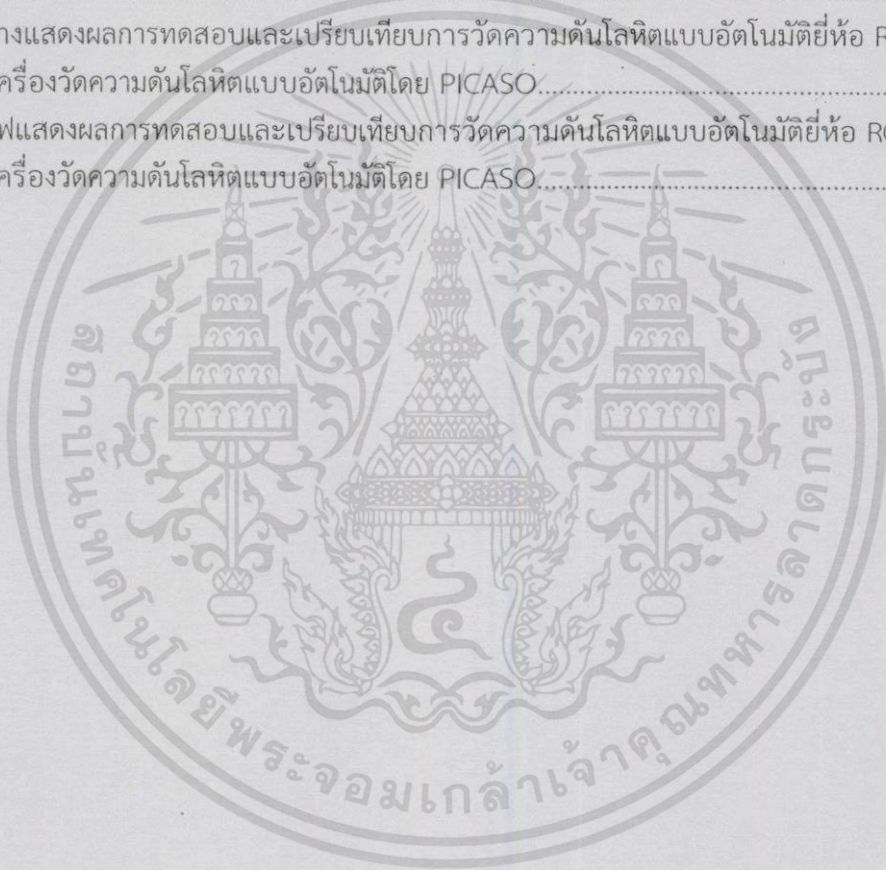
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง.....	41
4.1 ผลการทดสอบเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO.....	41
4.2 ผลการทดสอบและเปรียบเทียบการวัดความดันโลหิตยี่ห้อ ROSSMAX กับ เครื่องวัดความดันโลหิตโดย PICASO.....	42
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	45
5.1 สรุปผลการทดลอง.....	45
5.2 ปัญหาและอุปสรรค.....	45
5.3 ข้อเสนอแนะ.....	45
บรรณานุกรม.....	47
ภาคผนวก ก.....	49
ภาคผนวก ข.....	63

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางแสดงระดับความดันโลหิต.....	3
2.2 แสดงคุณลักษณะการทำงานของทรานสดิวเซอร์ความดัน MPX 5050.....	13
2.3 แสดงรายละเอียดของขาเอาต์พุต J2.....	19
2.4 แสดงรายละเอียดของขาอินพุต J1.....	19
2.5 แสดงการทำงานของขาแต่ละขา ที่เป็นการสื่อสารแบบ SPI.....	28
4.1 ตารางแสดงผลการทดสอบการวัดค่าความดันโลหิตแบบอัตโนมัติโดย PICASO.....	41
4.2 ตารางแสดงผลการทดสอบและเปรียบเทียบการวัดความดันโลหิตแบบอัตโนมัติยี่ห้อ ROSSMAX กับเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO.....	43
4.3 กราฟแสดงผลการทดสอบและเปรียบเทียบการวัดความดันโลหิตแบบอัตโนมัติยี่ห้อ ROSSMAX กับเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO.....	44



สารบัญรูป

รูปที่	หน้า
2.1 แสดงตำแหน่งของเส้นเลือดที่วัดความดันโลหิต.....	5
2.2 การวัดความดันโลหิตโดยการวัดโดยตรง.....	6
2.3 การวัดความดันโลหิตโดยใช้วิธีการทางอ้อม.....	6
2.4 แสดงการวัดการคล้ำซีพจร.....	7
2.5 แสดงการวัดความดันโลหิตวิธีออสซิลโลเมตริก.....	9
2.6 แสดงกราฟความดันภายในคัพและกราฟเสียง.....	9
2.7 แสดงไมโครคอนโทรลเลอร์ PICASO.....	10
2.8 แสดง PICASO – GFX2 ของ 4DGL.....	11
2.9 รายละเอียดขาของ PICASO.....	12
2.10 แสดงความสัมพันธ์ระหว่างความดันอินพุตกับแรงดันไฟฟ้าเอาต์พุตของทรานซิสเตอร์ความดัน MPX 5050.....	13
2.11 แสดงแบบของทรานซิสเตอร์ความดัน MPX 5050.....	14
2.12 แสดงอุปกรณ์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล MCP-3424.....	15
2.13 จอแสดงผล μ LCD – 43PCT (GFX).....	17
2.14 แสดงวงจรของจอแสดงผล μ LCD – 43PCT (GFX).....	18
2.15 แสดงรายละเอียดของวงจรของจอแสดงผล μ LCD – 43PCT (GFX).....	18
2.16 พอร์ตอนุกรม COM0, COM1 UARTS.....	21
2.17 แสดง Timing Diagram ในการส่งข้อมูลไบต์เดียว.....	21
2.18 หน่วยความจำ SD Card.....	23
2.19 แสดงส่วนหัวและส่วนปลายของสาย Programming Cable.....	24
2.20 ลักษณะการเชื่อมต่ออุปกรณ์แบบ I ² C Bus.....	25
2.21 รูปแบบการเขียน/อ่านข้อมูลแบบ I ² C Bus.....	26
2.22 Start and Stop Conditions.....	26
2.23 I ² C Bus (Control Byte).....	27
2.24 ช่วงเวลารับส่งบิตข้อมูลของ I ² C Bus.....	27
2.25 การสื่อสารแบบ SPI.....	28
3.1 แสดงแผนภาพการเชื่อมต่ออุปกรณ์แต่ละตัว.....	29
3.2 แสดงท่อลมเข้ากับอุปกรณ์ต่างๆ.....	29
3.3 เครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO.....	30
3.4 แสดงแผนผังการทำงานสำหรับเขียนโปรแกรม.....	31

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันเทคโนโลยีต่างๆ ได้เข้ามามีบทบาทในการทำงานของมนุษย์เรามากขึ้น เพื่อที่จะช่วยอำนวยความสะดวกในการทำงานมากขึ้น แม้กระทั่งเครื่องมือทางการแพทย์ก็ได้มีการพัฒนามากยิ่งขึ้น อย่างเช่นเครื่องวัดความดันโลหิต จากอดีตจะเห็นได้ว่า เครื่องมือวัดความดันโลหิตนั้นมีความยุ่งยากในการวัดมาก อีกทั้งต้องให้แพทย์หรือพยาบาลเป็นคนวัดให้เท่านั้น แต่ในปัจจุบัน เครื่องมือวัดความดันโลหิตมีวางจำหน่ายตามท้องตลาดมากมาย โดยทุกเครื่องที่มีวางจำหน่ายนั้นล้วนเป็นแบบอัตโนมัติทั้งสิ้น ซึ่งง่ายและสะดวกต่อการใช้งาน อีกทั้งสามารถวัดได้ด้วยตนเอง ดังนั้นโครงการนี้จัดทำขึ้นเพื่อศึกษาเครื่องวัดความดันโลหิตแบบอัตโนมัติ โดยใช้ PICASO จากทางการแพทย์ได้รับความหมายของความดันโลหิต คือ แรงดันของกระแสเลือดที่กระทบต่อผนังหลอดเลือด ซึ่งเกิดจากการสูบฉีดของหัวใจ (คล้ายแรงลมที่ต้นผืนยางเวลาสูบลมเข้า) ซึ่งสามารถวัดโดยใช้เครื่องวัดความดันโลหิต (Sphygmomanometer) วัดที่แขน และมีค่าที่วัดได้ 2 ค่า คือ ความดันซิสโตลิก (Systolic blood pressure) หมายถึง แรงดันเลือดขณะที่หัวใจบีบตัว ซึ่งอาจสูงตามอายุ ความดันช่วงบนในคนเดียวกัน อาจมีค่าแตกต่างกันบ้างเล็กน้อย ตามท่าทางของร่างกาย การเปลี่ยนแปลงของอารมณ์ และปริมาณของการออกกำลังกาย และ ความดันไดแอสโตลิก (Diastolic blood pressure) หมายถึง แรงดันเลือดขณะที่หัวใจคลายตัว ซึ่งโครงการนี้ศึกษาแนวทางองค์ความรู้การวัดความดันโลหิตโดยใช้วิธีการออสซิลโลเมตริก (Oscillometry) เพื่อมาทำวิเคราะห์ให้ได้ ซึ่งค่าความดันโลหิตที่สูง - ต่ำออกมา โดยสามารถควบคุมได้จากคอมพิวเตอร์ในซอฟต์แวร์ที่เรียกว่า 4DGL พร้อมทั้งศึกษาไมโครคอนโทรลเลอร์ที่เรียกว่า PICASO และนำมาใช้งาน โดยเป็นอุปกรณ์ที่ใช้สำหรับควบคุมการประมวลผลค่าที่ได้จากอุปกรณ์ตรวจวัดความดัน (Pressure Sensor) และแสดงผลลัพธ์ที่ออกมาซึ่งจอแสดงผล

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาวิธีการวัดความดันโลหิตแบบออสซิลโลเมตริก
2. ศึกษาการทำงานของเครื่องวัดความดันโลหิตแบบอัตโนมัติ
3. ศึกษาและทำความเข้าใจหลักการทำงานของอุปกรณ์ตรวจวัดความดัน MPX 5050
4. ศึกษาหลักการทำงานและการใช้งานของ PICASO
5. ศึกษาวิธีการเขียนโปรแกรมควบคุมการทำงานของ PICASO โดยใช้ภาษา 4DGL เพื่อนำข้อมูลไปแสดงผลยังจอแสดงผล
6. ศึกษาการทำเครื่องวัดความดันโลหิตแบบ Manual ให้เป็นแบบอัตโนมัติ

1.3 ขอบเขตของปริญญานิพนธ์

1. ศึกษาหลักการวัดความดันโลหิตและวิธีการวัดความดันโลหิตแบบออสซิลโลเมตริก
2. ศึกษาหลักการทำงานของอุปกรณ์ตรวจวัดความดัน MPX 5050

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ศึกษาการเขียนโปรแกรมด้วยภาษา 4DGL และออกแบบฟังก์ชันสำหรับวิเคราะห์สัญญาณ และแสดงผล

1.4 ขั้นตอนการศึกษา

1. ศึกษาประวัติความเป็นมา ความหมายของความดันโลหิต ระบบการหมุนเวียนเลือด
2. ศึกษาขั้นตอนการวัดความดันโลหิต
3. ศึกษาวิธีการวัดความดันโลหิตในแบบต่างๆ และวิธีการวัดความดันโลหิตแบบออสซิลโลเมตริกอย่างละเอียด
4. ศึกษาหลักการทำงาน และการนำไปใช้งานของอุปกรณ์ตรวจวัดความดัน MPX 5050
5. ศึกษาหลักการทำงาน และการนำไปใช้งานของไมโครคอนโทรลเลอร์ PICASO
6. ศึกษาวิธีการเขียนโปรแกรมโดยใช้ภาษา 4DGL จากนั้นทำการออกแบบฟังก์ชันสำหรับวิเคราะห์สัญญาณ และส่งผลมาแสดงผลบนจอ LCD

1.5 ประโยชน์ที่คาดว่าจะได้รับ

โครงการนี้เป็นการศึกษาเครื่องมือวัดความดันโลหิตแบบอัตโนมัติ โดยการนำไมโครคอนโทรลเลอร์ PICASO มาทำการควบคุมการทำงานและทำการประมวลผลที่ได้จากอุปกรณ์ตรวจวัดความดัน MPX5050 จากนั้นนำผลที่ได้มาทำการแสดงผลมายังจอแสดงผล โดยใช้หลักการวัดความดันโลหิตแบบออสซิลโลเมตริก คือ อาศัยการเปลี่ยนแปลงของค่าความดันภายใน cuff ที่เป็นผลมาจากจังหวะการเต้นของหัวใจ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 การวัดความดันโลหิต

การวัดความดันโลหิตด้วยวิธีออสซิลโลเมตริก เป็นวิธีการวัดความดันโลหิตโดยทางอ้อม ที่มีความนิยมมากที่สุดวิธีหนึ่ง ซึ่งการวัดความดันโลหิตแบบออสซิลโลเมตริกหลักการคือจะใช้ผ้าพันแขนทำการปิดกั้นการไหลของหลอดเลือดแดงเป็นการชั่วคราวโดยความดันที่สูงกว่าแล้วค่อยๆ ลดความดันที่ปิดกั้นลงมาจนถึงบางช่วงขณะที่ความดันในหลอดเลือดสูงกว่าความดันปิดกั้น หลอดเลือดจะเปิดและเลือดสามารถไหลผ่านได้ ซึ่งจะมีความสัมพันธ์กับความดันซิสโตลิกและความดันไดแอสโตลิก วิธีการวัดความดันเลือดแบบออสซิลโลเมตริกนี้จะให้ค่าเฉลี่ยได้ถูกต้องมากที่สุด แต่มีข้อเสียคือให้ค่าไดแอสโตลิกที่ถูกต้องน้อยกว่า

ความดันโลหิต หมายถึง แรงดันของกระแสเลือดที่กระทบต่อผนังหลอดเลือด ซึ่งเกิดจากการสูบฉีดของหัวใจ (คล้ายแรงลมที่ตีบนั่งยางรถเวลาสูบลมเข้า) ซึ่งสามารถวัดโดยใช้เครื่องวัดความดันเครื่องวัดความดัน วัดที่แขนและมีค่าที่วัดได้ 2 ค่าคือ

ความดันช่วงบนหรือความดันซิสโตลิก (Systolic blood pressure) หมายถึง แรงดันเลือดขณะที่หัวใจบีบตัว ซึ่งอาจจะสูงตามอายุ ความดันช่วงบนในคน ๆ เดียวกันอาจมีค่าแตกต่างกันบ้างเล็กน้อย ตามท่าของร่างกาย การเปลี่ยนแปลงของอารมณ์ และปริมาณของการออกกำลังกาย

ความดันช่วงล่างหรือความดันไดแอสโตลิก (Diastolic blood pressure) หมายถึง แรงดันเลือดขณะที่หัวใจคลายตัว ในปัจจุบันได้มีการกำหนดค่าความดันโลหิต และระดับความรุนแรงของโรคความดันโลหิตสูง สำหรับผู้ที่มีอายุตั้งแต่ 18 ปีขึ้นไป (โดยการวัดในท่านั่ง วัดอย่างน้อย 2 ครั้งขึ้นไปแล้วคิดเป็นค่าเฉลี่ย) ดังนี้

ตารางที่ 2.1 ตารางแสดงระดับความดันโลหิต

ระดับความดันเลือด	ความดันตัวบน (ม.ม.ปรอท)	ความดันตัวล่าง (ม.ม.ปรอท)	การแก้ไข
ปกติ	น้อยกว่า 120	น้อยกว่า 80	
ระยะก่อนความดันโลหิตสูง	120-139	หรือ 80-89	วัดซ้ำทุก 3-6 เดือน ร่วมกัน การปรับพฤติกรรม
ระดับ 1 ความดันโลหิตสูงอย่าง อ่อน	140-159	หรือ 90-99	ปรับพฤติกรรมร่วมกับการใช้ ยา
ระดับ 2 ความดันโลหิตสูง ปานกลาง	160-179	หรือ 100-109	เหมือนระดับ 1
ระดับ 3 ความดันโลหิตสูงรุนแรง	เท่ากับมากกว่า	เท่ากับมากกว่า	เหมือนระดับ 1

ความดันโลหิตสูง หมายถึงความดันช่วงบนมีค่าตั้งแต่ 140 มิลลิเมตรปรอทขึ้นไป หรือความดันช่วงล่างมีค่าตั้งแต่ 90 มิลลิเมตรปรอทขึ้นไป โดยมากผู้ป่วยจะมีความดันช่วงล่างสูง (Diastolic hypertension) โดยความดันช่วงบนจะสูงหรือไม่ก็ได้ บางคนอาจมีความดันช่วงบนสูงเพียงอย่างเดียวแต่ความดันช่วงล่างไม่สูง เรียกว่า ความดันช่วงบนสูงเดี่ยว (Isolated systolic hypertension) ซึ่งมักจะพบในผู้สูงอายุ, โรคคอพอกเป็นพิษ, ภาวะหลอดเลือดแดงเอออร์ตาตีบตัน เราจะวินิจฉัยโรคนี้แน่นอนต่อเมื่อวัดความดันแต่ละคราว อย่างน้อย 2 ครั้งขึ้นไป หากค่าเฉลี่ยของความดันและนับมาวัดซ้ำอีกอย่างน้อย 1-2 คราวภายใน 1 สัปดาห์ แล้วยังพบว่ามีค่าเฉลี่ยความดันสูงกว่าปกติ ในการวัดแต่ละคราวควรให้ ผู้ป่วยนั่งพักสัก 5-10 นาทีเสียก่อนโรคความดันโลหิตสูง พบได้ประมาณ 5-10% ของคนทั่วไป ส่วนมากจะเริ่มเป็นในคนที่มีอายุมากกว่า 40 ปี ขึ้นไปโดยไม่ทราบสาเหตุแน่ชัดส่วนน้อยที่อาจพบในคนอายุน้อยซึ่งมักจะมีความผิดปกติอื่น ๆ ร่วมด้วย

อัตราการเต้นของหัวใจ หมายถึงความเร็วของการบีบตัวของหัวใจในช่วงระยะเวลาหนึ่งๆ โดยทั่วไปนิยมใช้หน่วย "ครั้งต่อนาที" อัตราหัวใจเต้นสามารถเปลี่ยนแปลงได้ขึ้นกับสรีรวิทยาของร่างกาย เช่น ความต้องการออกซิเจนและการขับคาร์บอนไดออกไซด์ของร่างกาย สิ่งที่มีผลกับอัตราหัวใจเต้นได้แก่กิจกรรมของร่างกาย เช่น การออกกำลังกาย การนอนหลับ ความเจ็บป่วย การย่อยอาหาร

ค่าเฉลี่ยความดันเลือด หรือ Mean Arterial Pressure/MAP คำนวณด้วยสูตร $MAP = [(2 \times DBP) + SBP] / 3$ มาจากการบีบตัวและคลายตัวของหัวใจจะเป็นสัดส่วนระยะเวลาที่ไม่เท่ากัน ช่วงเวลาที่หัวใจใช้ในการบีบตัวจะกินเวลาแค่ 1 ใน 3 ในขณะที่ช่วงที่หัวใจคลายตัวจะกินระยะเวลาที่ยาวมากกว่า คือคิดเป็น 2/3 และการแปลผลค่า MAP ของคนปกติ ควรมีค่าอยู่ระหว่าง 70-110 มิลลิเมตรปรอท

2.1.1 การปฏิบัติก่อนการวัดความดันโลหิต

1. ผู้ป่วยต้องไม่สูบบุหรี่ ดื่มกาแฟ หรือออกกำลังกายภายใน 30 นาทีก่อนการวัด
2. ผู้ป่วยต้องพักผ่อนก่อนวัดอย่างน้อย 5 นาที ซึ่งอาจจะเกิดภาวะความดันโลหิตสูงจากความตื่นเต้น
3. ขนาดของผ้าพันแขน (cuff) ที่เหมาะสมกล่าวคือ ถ่วงภายในผ้าพัน (bladder) ควรยาวพันได้รอบแขนโดยไม่ทับกัน หรืออย่างน้อยพันรอบ 2/3 ของแขนหรือขา และความกว้างควรมากกว่า 20 % ของเส้นผ่าศูนย์กลางตรงจุดกึ่งกลางของแขนและขา
4. ใส่เสื้อผ้าหลวมๆ ถ้าใส่เสื้อแขนยาวห้ามม้วนจนรัดต้นแขน เพราะจะทำให้วัดได้ค่าต่ำกว่าปกติ
5. ไม่สูบบุหรี่ ดื่มสุรา ชา กาแฟ ก่อนวัดความดันโลหิต 30 นาที
6. ไม่รับประทานยากระตุ้นระบบประสาท เช่น ยาลดน้ำมูก ยาขยายรูม่านตา เป็นต้น

2.1.2 เทคนิคการวัดความดันที่ถูกต้อง

1. ตำแหน่งของเส้นเลือดที่วัดความดันโลหิตควรอยู่ในระดับเดียวกับหัวใจ และถ้าระดับแขนอยู่ต่ำกว่าหัวใจจะทำให้มีความดันโลหิตที่เพิ่มขึ้นอันเกิดจากแรงโน้มถ่วงของโลกซึ่งอาจเพิ่มขึ้น 10-12 มิลลิเมตรปรอท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ปล่อยลมออกจากผ้าพันแขนให้เหมาะสม คือไม่เร็วหรือช้าเกินไป อัตราที่เหมาะสม คือ 2 มิลลิเมตรปรอทต่อจังหวะหัวใจเต้นหนึ่งครั้ง

3. ความดันโลหิตที่วัดครั้งแรกมักจะสูงกว่าความเป็นจริง จากการศึกษาวิจัยพบว่าค่าความดันโลหิตที่ได้จากการวัดครั้งที่ 3 มีความแม่นยำถูกต้องมากที่สุด

4. ในการรัดผ้าพันแขนไม่ควรรัดแน่นหรือหลวมเกินไปจะทำให้ค่าที่ได้มีความคลาดเคลื่อนจากความเป็นจริง

5. หลีกเลี่ยงการพูดคุยขณะวัดความดัน เพราะสามารถทำให้ความดันเพิ่มขึ้นได้ถึง 8 - 15 มิลลิเมตรปรอท และควรวัดความดันในห้องที่มีอุณหภูมิเหมาะสม ไม่หนาวหรือร้อนเกินไป



รูปที่ 2.1 แสดงตำแหน่งของเส้นเลือดที่วัดความดันโลหิตควรอยู่ในระดับเดียวกับหัวใจ

2.2 ทฤษฎีการวัดความดันโลหิต

ความดันโลหิตเกิดจากการบีบตัวของหัวใจ คำว่าความดันโลหิต มักจะหมายถึงความดันโลหิตแดง ซึ่งตามความเป็นจริงแล้วสามารถวัดความดันโลหิตที่อื่นได้ด้วย เช่น ความดันโลหิตดำ และความดันในห้องหัวใจ

ความดันโลหิตในส่วนต่างๆของอวัยวะในร่างกายของระบบไหลเวียนเลือดไม่เท่ากันทุกจุด โดยทั่วไปความดันโลหิตแดงที่ส่งจากหัวใจจุดแรกจะมีความดันสูงสุด ต่อจากนั้นจะค่อยๆลดลงจนถึงหลอดเลือดดำใหญ่ที่จะเข้าสู่หัวใจซึ่งจะมีความดันต่ำที่สุด

ความดันโลหิตแดงจะมีลักษณะเป็น pulsatile คือสูงสุดหัวใจบีบตัวและต่ำสุดขณะหัวใจคลายตัวและเมื่อถึงหลอดเลือดเล็กๆลักษณะการเกิด pulsatile จะค่อยๆลดลงและหมดไปที่ละน้อยอันเกิดจากการยืดหยุ่นและความต้านทานของหลอดเลือด

การที่หลอดเลือดต้องมีความดันก็เพราะหลอดเลือดมีหน้าที่เป็นท่อส่งหรือท่อลำเลียงโลหิตที่ส่งออกจากหัวใจไปเลี้ยงอวัยวะต่างๆ ทัวร่างกายความดันโลหิตแดงในระบบไหลเวียนทัวร่างกายสูงกว่าระบบไหลเวียนผ่านปอดถึง 5 เท่า

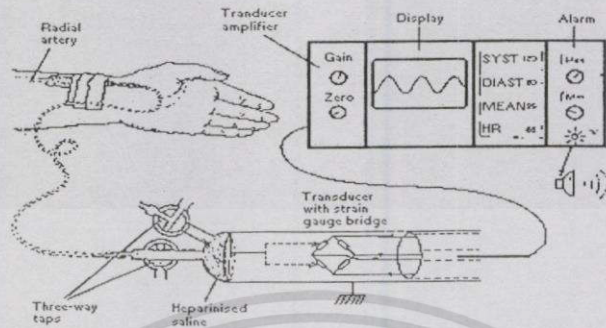
2.2.1 วิธีการวัดความดันโลหิต

ในการวัดความดันโลหิตกระทำได้ 2 วิธีคือ

1. การวัดโดยตรง เป็นการใช้เข็มแทงเข้าไปในหลอดเลือด แล้วนำมาต่อกับเครื่องวัด (manometer) โดยตรง แต่วิธีนี้ก็ยังไม่สามารถให้ค่าที่แท้จริงทีเดียวเพราะยังมีความเฉื่อยของปรอท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

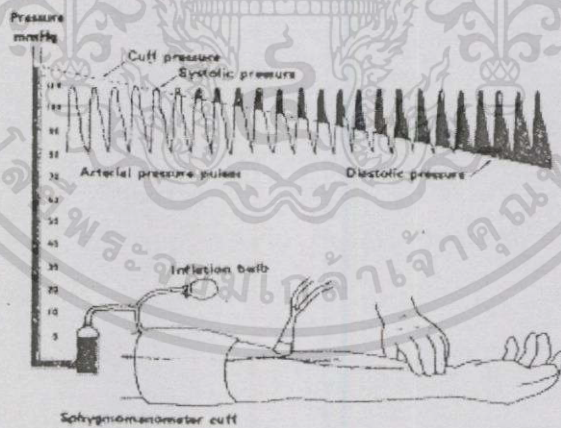
จนทำให้ pulse pressure แคบไปถ้าจะให้ค่าที่แน่นอนกว่าจะต้องใช้เครื่องวัดที่เป็นระบบ อิเล็กทรอนิกส์



Pressure transducer, bridge amplifier, display and alarm

รูปที่ 2.2 การวัดความดันโลหิตโดยการวัดโดยตรง

2. การวัดโดยทางอ้อม วิธีนี้สะดวกและผู้ถูกวัดไม่เจ็บตัว เพราะไม่ต้องวัดจากหลอดเลือดโดยตรงแต่จะใช้คัพพันที่ข้อมือบนหลอดเลือด แล้วเพิ่มความดันในคัพจนสูงกว่าความดันหลอดเลือด แล้วค่อยๆลดความดันลงและใช้เครื่องฟังตรวจฟังเสียงของหลอดเลือดที่อยู่ถัดจากคัพลงมา ค่าความดันที่อ่านได้เมื่อเริ่มได้ยินเสียง คือค่า ซิสโตลิกและค่าความดันที่อ่านได้ตรงจุดที่เสียงหายไป คือค่า ไดแอสโตลิก



รูปที่ 2.3 การวัดความดันโลหิตโดยใช้วิธีทางอ้อม

การตรวจหรือวัดความดันโลหิตของหลอดเลือดแดง สามารถกระทำได้ 5 วิธีด้วยกันคือ

1. การดูสีผิวเมื่อมีเลือด (flush)

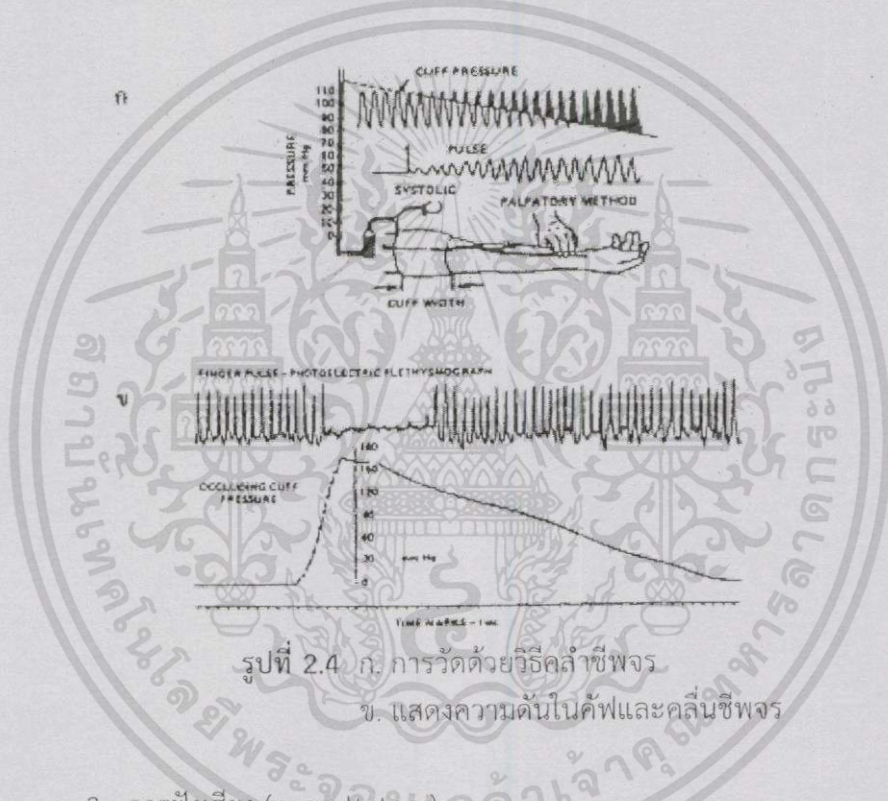
เป็นวิธีที่ใช้ผ้ายึด (elastic bandage) พันปลายขาขึ้นมาถึงลำตัว เหมือนเป็นการบีบเลือดออกจากเนื้อเยื่อ นำคัพมาพันเหนือผ้ายึดแล้วบีบลมเข้าไปในคัพให้มากพอแล้วคลายผ้ายึดออกต่อมาค่อยๆปล่อยลมออกจากผ้าพันแขนซ้าย และเมื่อถึงความดันซิสโตลิก ผิวหนังส่วนที่อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล่างลงมาจะแดงเร็วขึ้นเนื่องจากมีเลือดไหลผ่านเข้ามา การวัดวิธีนี้จะได้เฉพาะค่าซิสโตลิก ส่วนค่าอื่นๆไม่สามารถวัดได้ แม่วิธีนี้จะค่อนข้างยุ่งยาก

2. การคลำชีพจร (palpatory)

จะใช้นิ้วมือคลำชีพจรที่ข้อมือ ส่วนอีกมือหนึ่งบีบลูกยาง ฤงยางในผ้าพันแขนจะโป่งขณะเดียวกันต้องคอยสังเกตระดับความดันชีพจรจุดที่หายไป ให้บีบเลยขึ้นไปอีกเล็กน้อย แล้วจึงค่อยๆปล่อยลมออกทางวาล์ว ทำการสังเกตจุดที่เริ่มคลำชีพจรอีกครั้ง แล้วจึงนำค่าที่สังเกตได้ทั้งขึ้นและลงมาเฉลี่ยจะได้ค่าความดันซิสโตลิก ด้วยวิธีการนี้หากใช้คัพมาตรฐานจะได้ค่าความดันต่ำกว่าความเป็นจริง แต่จะไม่สามารถวัดค่าไดแอสโตลิกและค่าเฉลี่ยได้



รูปที่ 2.4 ก. การวัดด้วยวิธีคลำชีพจร

ข. แสดงความดันในคัพและคลื่นชีพจร

3. การฟังเสียง (auscultatory)

เป็นวิธีการฟังเสียง korotkoff ใช้ผ้าพันแขนพันที่แขน บีบลูกยางให้ความดันในผ้าพันแขนสูงกว่าค่าที่คาดว่าเป็นซิสโตลิกใช้ stethoscope ฟังที่หลอดเลือด brachial ตำแหน่งต่ำกว่าผ้าพันแขน และปล่อยลมออก เมื่อความดันในผ้าพันแขนต่ำกว่าซิสโตลิกหลอดเลือดจะเปิดตัวเล็กน้อยมีเลือดไหลผ่าน ฟังเสียงได้ ณ จุดนี้อ่านค่าเป็นความดันซิสโตลิกเมื่อปล่อยลมออกต่อไปจะยังคงได้ยินเสียงดังชัดเจนค่อยๆดังขึ้น, เสียงนุ่มลง, เสียงดังที่สุด, เสียงค่อยลงและหายไป

4. การออสซิลเลท (oscillometric)

เป็นการวัดแอมพลิจูดของการออสซิลเลทชันของความดันภายในผ้าพันแขน นำมาเป็นตัวบ่งชี้ตำแหน่งของความดันซิสโตลิกและความดันเฉลี่ย เนื่องจากการออสซิลเลชันจะมีขนาดของแอมพลิจูดที่มีขนาดเล็ก ในวิธีการสร้างเครื่องจึงต้องมีการขยายสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. อัลตราซาวด์ (ultrasonic)

เป็นการวัดการเคลื่อนของผนังหลอดเลือด จะได้ค่าสัมพันธ์กันดีมากทั้งค่าซิสโตลิกและไดแอสโตลิกเลือดคนผนังเส้นเลือด เพราะหูของคนปกติจะได้ยินแต่เสียง ไม่ใช่การสัมผัสเพื่อน

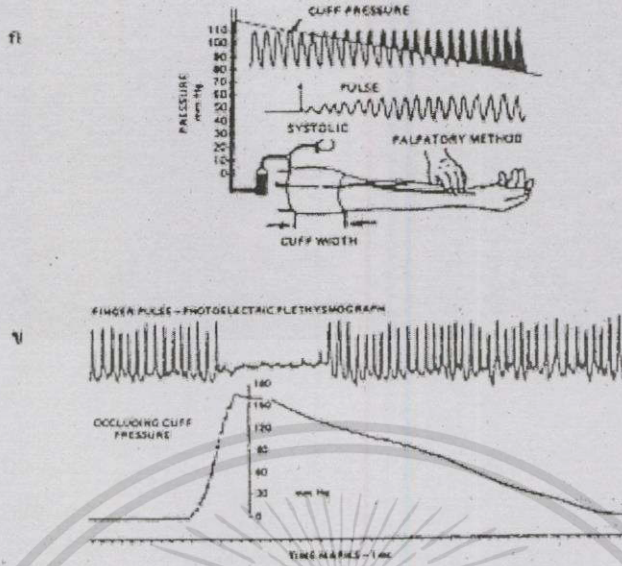
วิธีการที่ดีที่สุดในการยืนยันความถูกต้องของความแม่นยำของเครื่องมือวัดแบบการตรวจจับแรงดันก็คือ ให้แพทย์ฟังเสียงด้วยหูฟังสเตโกลโคปร่วมกับเครื่องมือวัดแบบการตรวจจับแรงดัน โดยการทำการทดลองที่คนไข้เดียวกัน และทำการเปรียบเทียบค่าที่ได้จากเครื่องกับแพทย์ผู้ฟังหูฟัง

2.2.2 การวัดความดันโลหิตด้วยวิธีการออสซิลเลทหรือออสซิลโลเมตริก (Oscillometric)

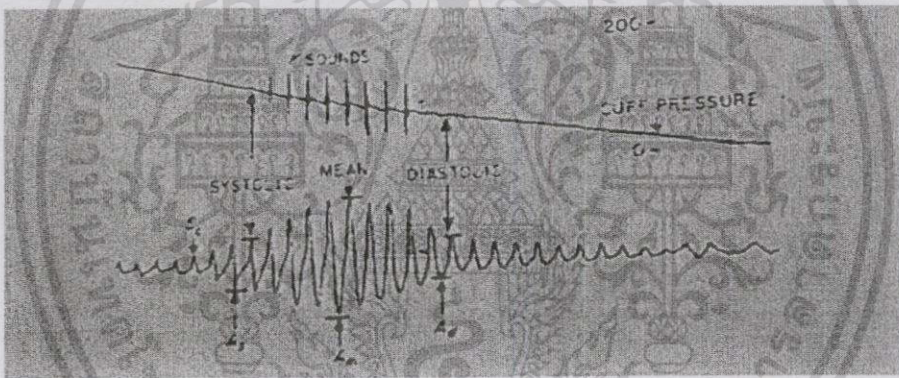
วิธีการออสซิลเลชัน หรือ ออสซิลโลเมตริก เป็นการวัดแอมพลิจูดของการออสซิลเลชันของความดันภายในผ้าพันแขน นำมาเป็นตัวบ่งชี้ตำแหน่งของความดันซิสโตลิกและความดันเฉลี่ย เนื่องจากการออสซิลเลชันจะมีขนาดของแอมพลิจูดที่มีขนาดเล็ก ในวิธีการสร้างเครื่องจึงต้องมีการขยายสัญญาณ

ค่าความดันภายในคัพที่สูงกว่าค่าซิสโตลิก จะเป็นตำแหน่งเดียวกันที่คลื่นชีพจรหายไป เมื่อปล่อยลมออกช้าๆ (ประมาณ 3 มิลลิเมตรปรอท ต่อหนึ่งจังหวะหัวใจ) และในช่วงที่ค่าความดันต่ำกว่าค่าความดันเลือด หลอดเลือดจะเปิดและเลือดจะไหลได้ ทำให้ความดันในผ้าพันแขนแกว่ง (ออสซิลเลท) ดังแสดงในรูป 2.4 ก ตำแหน่ง s การออสซิลเลชันจะค่อยๆ เติบโตขึ้นเมื่อปล่อยลมออกจากคัพจนถึงจุดสูงสุดที่ตำแหน่ง M ซึ่งตำแหน่งนี้จะใกล้เคียงกับค่าความดันเฉลี่ย ส่วนค่าไดแอสโตลิกนี้ยังไม่มีตัวบ่งชี้ที่แน่นอน

วิธีการออสซิลโลเมตริก จะสามารถวัดค่าความดันเฉลี่ยได้ดี จากการศึกษาเปรียบเทียบกับวิธีการอื่น เช่น การศึกษาของ Posey และคณะเมื่อปี ค.ศ. 1969, Mauck และเมื่อปี ค.ศ. 1980 ปัจจุบันมีเครื่องมือวัดความดันเลือดอิเล็กทรอนิกส์ ที่ใช้วิธีนี้และแสดงค่าความดันเฉลี่ยออกมาเป็นตัวเลข จากการศึกษาค่าความดันเลือดในคนของ Ramsey เมื่อปี ค.ศ. 1979 และ Yalderman ได้แสดงให้เห็นว่าค่าที่ได้จากการวัดวิธีออสซิลโลเมตริก กับการวัดโดยวิธีทางตรงมีความสัมพันธ์กันสูงมาก ดังรูปที่ 1.2 ข้อมูลของ Ramsey และจากการศึกษาของ Kimble และคณะ เมื่อปี ค.ศ. 1981 ก็ได้ผลเช่นเดียวกัน เมื่อเขาวัดเปรียบเทียบกับการวัดค่าความดันเลือดแดงเฉลี่ย โดยทางอ้อมใช้ Dynamap กับการวัดทางตรงที่หลอดเลือดสายสะดือในเด็กแรกเกิดที่มีน้ำหนักตัว 700 - 3,600 กรัม อัตราส่วนระหว่างความกว้างของผ้าพันแขนต่อความยาวรอบแขนทารกระหว่าง 0.45 ถึง 0.7 การวัดโดยทางอ้อม (IND) มีความสัมพันธ์กับการวัดทางตรง (DIR) โดยความสัมพันธ์ของ $IND = 0.822 DIR + 7.48$ สัมประสิทธิ์ความสัมพันธ์ 0.853 ช่วงความดันเฉลี่ยที่วัด 20 - 65 มิลลิเมตรปรอท



รูปที่ 2.5 แสดงการวัดความดันโลหิตวิธีการออสซิลโลเมตริก



รูปที่ 2.6 เส้นบนเป็นการบันทึกเสียง Korotkoff เส้นล่างเป็นเส้นความดันในคัพที่ออสซิลเลท

SO เป็นจุดที่เริ่มมีการออสซิลเลทเพิ่มขึ้น

AS เป็นค่าแอมพลิจูดค่าความดันในคัพที่สอดคล้องกับการฟังเสียงที่เป็นค่าความดันซิสโตลิก

Ad เป็นค่าแอมพลิจูดความดันในคัพที่สอดคล้องกับการฟังเสียงที่เป็นค่าความดันไดแอสโตลิก

Am เป็นแอมพลิจูดของค่าความดันสูงสุดในคัพ ซึ่งหมายถึงค่าความดันเฉลี่ย

ด้วยวิธีการออสซิลโลเมตริก เรามีหลักเกณฑ์ในการกำหนดค่า ซิสโตลิก ไดแอสโตลิก และค่าเฉลี่ย จากการทดลองของ Geddes และคณะ เมื่อปี ค.ศ. 1983 โดยทำการทดลอง เปรียบเทียบกับวิธีการฟังเสียง Korotkoff (วิธีการ Auscultatory) ในผู้ใหญ่จำนวน 23 คน ใช้ผ้าพัน แขนมาตรฐานกว้าง 12 เซนติเมตร ที่ถูกวางจะติดไมโครโฟนรับเสียง ทำการบันทึกเสียงและค่าความดันในผ้าพันแขนที่ออสซิลเลท ณ จุดที่ได้ยินเสียงแรก (phase I) ดังขึ้นซึ่งเป็นค่าความดันออสซิลโตลิก และจุดที่สัญญาณเสียงหายไป (phase V) เป็นค่าความดันซิสโตลิก ข้อมูลที่บันทึกทำในขณะที่ให้ผู้ถูกทดสอบออกกำลังกายเกร็งกล้ามเนื้อขาเพื่อเพิ่มความต้านทานรอบนอกของระบบไหลเวียนเลือด

เป็นการแสดงผลการบันทึกที่พบจุดสังเกตที่สำคัญ เช่น ความดันซิสโตลิกที่ได้มาจากการแกว่งของ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความดันในผ้าพันแขน (SO) จะมีค่าสูงกว่าค่าที่วัดได้ด้วยวิธีการฟังเสียง Korotkoff ซึ่งจากการฟังเสียงนี้ เราทราบว่าต่ำกว่าความเป็นจริงเล็กน้อย คือหากค่าความดันซิสโตลิก มีค่า 120 มิลลิเมตรปรอท วิธีการวัดแบบออสซิลโลเมตริกจะอ่านค่าได้ประมาณ 129.60 มิลลิเมตรปรอท หรือสูงกว่าประมาณ 8%

ในจุดที่ฟังได้ค่าความดันซิสโตลิก เมื่อวัดขนาดแอมพลิจูดของออสซิลเลทในผ้าพันแขน (AS) แล้วนำไปเปรียบเทียบหรือหาสัดส่วนกับแอมพลิจูดสูงสุด (Am) ซึ่งเป็นค่าความดันเฉลี่ย จะได้สัดส่วน AS/Am เท่ากับ 0.55 ที่ความดัน 120 มิลลิเมตรปรอท และจะมีค่า 0.45 – 0.57 ในช่วงความดัน 100 – 190 มิลลิเมตรปรอท

ในจุดที่ฟังได้ค่าความดันไดแอสโตลิก เมื่อวัดขนาดของแอมพลิจูดในผ้าพันแขน (Ad) แล้วนำไปเปรียบเทียบหรือหาสัดส่วนกับค่าแอมพลิจูดสูงสุด (Am) จะได้ค่า Ad/Am มีค่า 0.82 ที่ความดัน 80 มิลลิเมตรปรอท และจะมีค่า 0.74 – 0.82 ในช่วงความดัน 55 – 115 มิลลิเมตรปรอท ข้อได้เปรียบประการหนึ่งของการวัดด้วยวิธีออสซิลโลเมตริก คือเราสามารถใช้อุปกรณ์ที่ไม่ยุ่งยากเพียงแค่ว่าใช้ผ้าพันแขนกับผู้ป่วยที่ถูกทดสอบ และสามารถวัดได้ทั้งใน สัตว์ทดลอง เด็ก และทารก ที่ใช้วิธีการฟังเสียงไม่ได้ผล และสามารถวัดได้สำเร็จแม้ความดันเลือดจะต่ำมาก (Hypotention)

2.3 ส่วนของฮาร์ดแวร์

โปรแกรม 4D เป็นภาษาสำหรับควบคุมการแสดงผลแบบกราฟิก ที่ให้ผู้พัฒนาเขียนโปรแกรมแบบ High level language เหมือนกับพวกภาษาที่รู้จักกัน เช่น ภาษา BASIC, C, Pascal และสั่งการทำงานได้โดยตรงบนตัวโปรเซสเซอร์ GOLDELOX ที่ใส่ลงมาบนบอร์ด μ OLED-128-G1 นี้ ด้วยคำสั่งและไลบรารีมากมายที่นำมา และภาษาโปรแกรมแบบ High level language ทำให้ผู้พัฒนาสามารถควบคุมสั่งการฮาร์ดแวร์ ส่วนต่างๆที่นำมาได้แบบสมบูรณ์อย่างเช่น Serial port ,Graphic Display, micro-SD, I/O pins และอื่นๆและอาจไม่จำเป็นต้องใช้ไมโครคอนโทรลเลอร์จากภายนอกมาต่อร่วมเลยเพราะผู้พัฒนาสามารถพัฒนางานได้บนตัวมันเอง



รูปที่ 2.7 แสดงไมโครคอนโทรลเลอร์ PICASO

2.3.1 PICASO-GFX2

PICASO - GFX2 เป็นชิพควบคุมการออกแบบกราฟิก เชื่อมต่อกับจอแสดงผลกราฟิก OLED และ μ LCD - 43PCT (GFX) เพื่อแสดงภาพกราฟิก ข้อความ ภาพเคลื่อนไหว มีการเชื่อมต่อแบบ Plug-n-Play ขนาด 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างภายในของ PICASO - GFX2 สร้างให้มีบล็อกการทำงานระดับสูงที่มีการควบคุมและดูแลโดย EVE(Extensible Virtual Engine)



รูปที่ 2.8 แสดง PICASO-GFX2 ของ 4DGL

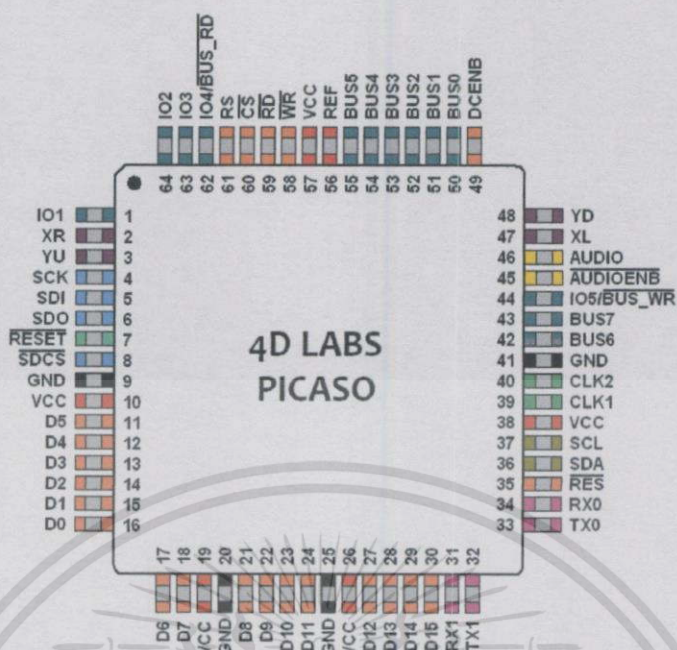
คุณสมบัติ

- เชื่อมต่อ CPU 16 บิต 80-Series กับ จอแสดงผลกราฟิก OLED/ μ LCD-43PCT(GFX)
- หน่วยความจำแฟลต 14 KB และ RAM 14 KB
- ต่อพอร์ตอนุกรมกับอุปกรณ์แบบ 2 Asynchronous
- พอร์ตการเชื่อมต่อฮาร์ดแวร์ SPI สำหรับ Micro-SD Card
- ใช้ Micro-SD Card/SDHC Card
- เข้าถึงไฟล์ DOS
- ไฟล์เสียง PWM ขนาด 16 บิต
- การสื่อสารแบบ I2C Bus
- รองรับพอร์ต GPIO ต่อขนานความละเอียด 8 บิต
- ตัวจับเวลา ขนาด 8x16 บิต ด้วยความละเอียด 1 Millisecond

รายละเอียดขาของ PICASO

- PICASO มีขนาด 16 บิต มีขาสัญญาณตั้งแต่ D0 ถึง D15 รวมทั้งขา RES, CS, RS, และ RD/RW ในการเชื่อมต่อกับจอแสดงผล
- PICASO ตั้งค่าให้รองรับกับคุณสมบัติของอุปกรณ์ อินพุต และ เอาท์พุต มีการสื่อสารแบบ SPI, I²C และการอนุกรมกับอุปกรณ์ มีการเชื่อมต่อโดยใช้พอร์ต GPIO 13 พอร์ต ซึ่งจะใช้สาย Pmmc ในการเชื่อมต่อ แต่ละ GPIO สามารถต่อกับอุปกรณ์อินพุต เอาท์พุต และBUS0 ถึง BUS7 ขนาด 8 บิต สามารถใช้ 8 บิต ต่อขนานกับการสื่อสารอุปกรณ์ อินพุต และ เอาท์พุต
- ไฟล์เสียงถูกสร้างภายในหน่วยประมวลผล PICASO เพื่อให้มีการรวมเสียงได้ง่าย แม้มีการทำงานพร้อมกับการทำงานอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 รายละเอียดขาของ PICASO

2.4 คุณสมบัติของอุปกรณ์แต่ละตัว

2.4.1 ทรานสดิวเซอร์ตรวจวัดความดัน MPX 5050

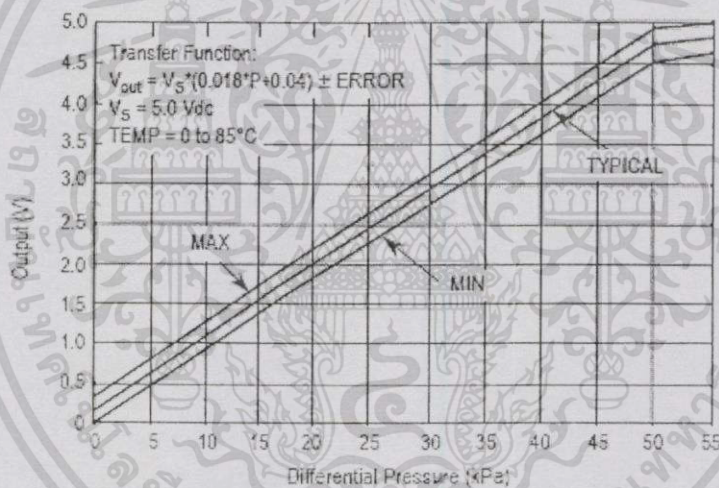
ทรานสดิวเซอร์ความดัน คือ อุปกรณ์ที่ทำหน้าที่แปลงสัญญาณความดันไปเป็นสัญญาณทางไฟฟ้าแบบอนาล็อก โดยทั่วไปทรานสดิวเซอร์ความดันจะมีหลายชนิดด้วยกัน ส่วนที่นิยมใช้งานมากที่สุด คือ ชนิดที่ใช้สเตรนเกจ (Strain-gage base transducer)

ทรานสดิวเซอร์ความดัน MPX 5050 นี้ เป็นชนิดเอาต์พุตที่มีหน่วยเป็น โวลท์ จะประกอบด้วย

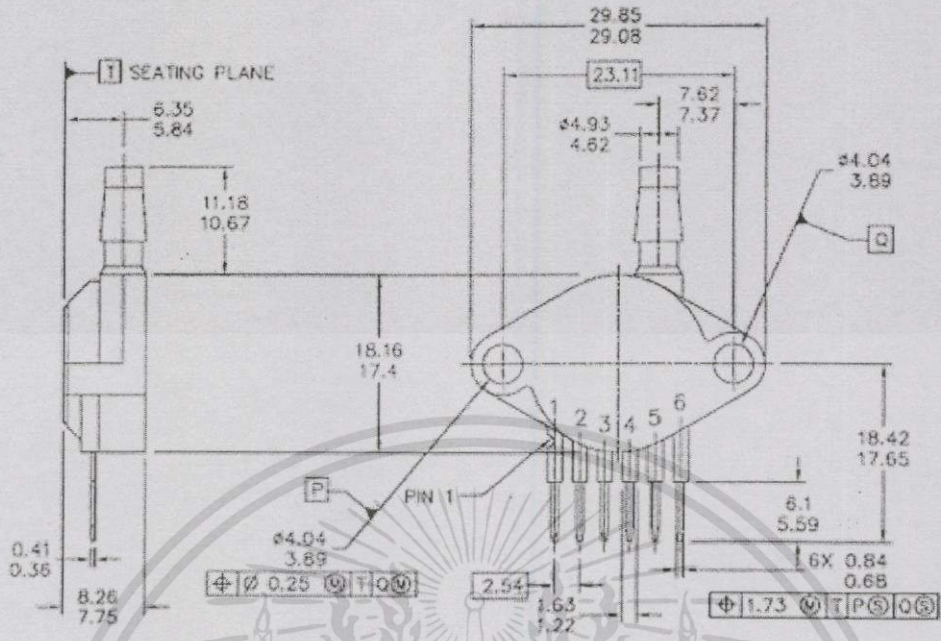
- วงจรปรับสภาวะสัญญาณอยู่ในตัวเอง integral signal conditioning
- ภายในวงจรเป็น Patented Silicon Shear Stress Strain Gauge ซึ่งเป็นซิลิคอน
- มีวงจรปรับสภาวะสัญญาณในตัว และสามารถชดเชยอุณหภูมิได้ในช่วงใช้งาน -40°C ถึง $+125^{\circ}\text{C}$
- สัญญาณเอาต์พุตที่ได้จะเป็นแบบ Analog โดยเป็นสัดส่วนโดยตรงกับความดันที่ป้อนเข้าไป
- ความคลาดเคลื่อนสูงสุด 2.5% เมื่ออุณหภูมิไม่ได้อยู่ในช่วง 0°C ถึง 85°C
- ใช้ส่วนประกอบของ Epoxy Unibody ที่ให้ความทนทาน

ตารางที่ 2.2 แสดงคุณลักษณะการทำงานของทรานสดิวเซอร์ความดัน MPX 5050

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range ⁽¹⁾	P _{OP}	0	—	50	kPa
Supply Voltage ⁽²⁾	V _S	4.75	5.0	5.25	Vdc
Supply Current	I _o	—	7.0	10	mAdc
Minimum Pressure Offset ⁽³⁾ ⊗ V _S = 5.0 Volts	V _{off}	0.088	0.2	0.313	Vdc
Full Scale Output ⁽⁴⁾ ⊗ V _S = 5.0 Volts	V _{FSO}	4.587	4.7	4.813	Vdc
Full Scale Span ⁽⁵⁾ ⊗ V _S = 5.0 Volts	V _{FSS}	—	4.5	—	Vdc
Accuracy ⁽⁶⁾	—	—	—	±2.5	%V _{FSS}
Sensitivity	V/VP	—	90	—	mV/kPa
Response Time ⁽⁷⁾	t _R	—	1.0	—	ms
Output Source Current at Full Scale Output	I _o	—	0.1	—	mAdc
Warm-Up Time ⁽⁸⁾	—	—	20	—	ms
Offset Stability ⁽⁹⁾	—	—	±0.5	—	%V _{FSS}



รูปที่ 2.10 แสดงความสัมพันธ์ระหว่างความดันอินพุตกับแรงดันไฟฟ้าเอาต์พุตของทรานสดิวเซอร์ความดัน MPX 5050



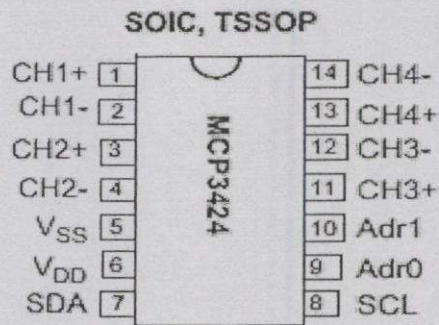
รูปที่ 2.11 แสดงแบบของทรานสดิวเซอร์ความดัน MPX 5050

2.4.2 อุปกรณ์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล MCP3424

เป็นอุปกรณ์แปลงสัญญาณจากสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัลที่มีสัญญาณรบกวนต่ำและมีความแม่นยำสูง อยู่ในตระกูลของ MCP342X จากบริษัท Microchip Technology ซึ่งสามารถแปลงสัญญาณได้ละเอียดมากถึง 18 บิต

คุณสมบัติของอุปกรณ์มีดังนี้

1. แรงดันไฟฟ้าอ้างอิงมีพิสัยคือ -2.048 V. คือ +2.048 V.
2. มีช่องการเชื่อมต่อ (Channel) 4 ช่องด้วยกัน
3. สามารถปรับอัตราขยายก่อนที่จะรับการแปลงสัญญาณได้ที x1, x2, x4 หรือ x8 ซึ่งทำให้สามารถแปลงสัญญาณที่อ่อนมากๆ ให้เป็นสัญญาณที่มีความละเอียดสูง
4. สามารถให้เอาต์พุตที่มีอัตราการเปลี่ยนแปลงของสัญญาณที่ 3.75, 15, 60 หรือ 240 ตัวอย่างต่อหนึ่งวินาที (โดยปรับตั้งค่าได้จากการตั้งค่าบิตผ่านทาง I2C อินเทอร์เฟซ)
5. การรับส่งข้อมูลผ่าน I²C อินเทอร์เฟซ ซึ่งสามารถทำงานได้ตั้งแต่โหมดปกติ (100 kHz) โหมดเร็ว (400 kHz) และโหมดความเร็วสูง (3.4 MHz)
6. I²C แอดเดรสสำหรับ MCP3424 สามารถกำหนดได้โดยขา Adr0 และ Adr1 ผู้ใช้สามารถตั้งค่าได้อุปกรณ์นี้ให้เป็นหนึ่งในแปดแอดเดรสโดยต่อขาแอดเดรสด้วย V_{DD}, V_{SS} หรือปล่อยลอยก็ได้



รูปที่ 2.12 แสดงอุปกรณ์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล MCP3424

2.4.3 ระบบรวบรวมข้อมูล และวิเคราะห์ข้อมูล โดยอาศัยคอมพิวเตอร์ (DAQ System)

กระบวนการเก็บข้อมูล หรือ สัญญาณ จากแหล่งที่ต้องการวัด ทั้งใน รูปของ อนาล็อก และ ดิจิตอล แล้วนำข้อมูล หรือสัญญาณมาจัดเก็บเพื่อนำมาใช้วิเคราะห์ หรือนำเสนอข้อมูลใน ภายหลังเครื่อง คอมพิวเตอร์ เป็นการใช้นวัตกรรมของทั้ง ฮาร์ดแวร์ และ ซอฟต์แวร์ เพื่อนำมา จัดทำเป็นระบบการวัด หรือเก็บข้อมูล ประกอบไปด้วย

2.4.3.1 คอมพิวเตอร์ส่วนบุคคล (PC)

เครื่องอุปกรณ์อิเล็กทรอนิกส์ชนิดหนึ่งที่มีการทำงานแบบอัตโนมัติ ทำหน้าที่ เหมือนสมองกล สามารถแก้ปัญหาต่าง ๆ ทั้งที่ง่ายและซับซ้อนตามคำสั่งของโปรแกรม จาก ความหมายจะเห็นว่าเครื่องคอมพิวเตอร์ มีอุปกรณ์ที่สามารถทำงานได้ 3 อย่าง คือ

2.4.3.1.1 รับโปรแกรมและข้อมูล

โปรแกรมในที่นี้ หมายถึง ชุดของคำสั่งที่จะให้คอมพิวเตอร์ ทำงาน ซึ่งเราเรียกว่าโปรแกรมคอมพิวเตอร์ ส่วนข้อมูลนั้นอาจจะเป็นตัวเลข หรือตัวอักษรที่ต้องการ ให้คอมพิวเตอร์ทำการประมวลผล

2.4.3.1.2 ประมวลผล

หมายถึง การจัดระเบียบแบบแผนของข้อมูล เพื่อให้ได้ผลลัพธ์ ตามที่ต้องการ ซึ่งทำได้โดยการคำนวณ เปรียบเทียบ วิเคราะห์โดยใช้สูตรทางวิทยาศาสตร์หรือ คณิตศาสตร์ วิธีการต่างๆ เหล่านี้ ทำได้โดยอาศัยชุดคำสั่งหรือโปรแกรมที่เขียนขึ้น

2.4.3.1.3 แสดงผลลัพธ์

คือการนำผลลัพธ์ที่ได้จากการประมวลผลเสร็จเรียบร้อยแล้ว แสดงออกในรูปแบบต่างๆ ที่ผู้ใช้เข้าใจ และนำไปใช้ประโยชน์ได้

2.4.3.2 ทรานสดิวเซอร์

ทรานสดิวเซอร์ คือ อุปกรณ์ที่ทำหน้าที่เปลี่ยนสัญญาณในรูปแบบหนึ่งไปเป็น อีกรูปแบบหนึ่ง เช่น การเคลื่อนที่, การไหล ฯลฯ ให้เป็นสัญญาณทางไฟฟ้า หน้าที่ของทรานสดิวเซอร์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้แก่ ตรวจจับค่าการวัดที่เกิดขึ้น เช่น ขนาด การเปลี่ยนแปลงความถี่ อุปกรณ์ทรานสดิวเซอร์ส่วนใหญ่ใช้อุปกรณ์และวงจรวจ็เล็กทรอนิกส์เป็นองค์ประกอบ

2.4.3.3 สัญญาณ

2.4.3.3.1 สัญญาณอนาลอก

เป็นสัญญาณที่มีค่าใด ๆ เทียบกับเวลา ตัวอย่างของสัญญาณอนาลอก เช่นค่าความต่างศักย์, อุณหภูมิ, ความดัน, เสียง เป็นต้น สัญญาณ อนาลอก มีคุณลักษณะแตกต่างกันอยู่ 3 อย่างคือ ระดับ, รูปร่าง และ ความถี่

2.4.3.3.2 สัญญาณดิจิตอล

เป็นสัญญาณที่มีค่าเป็นไปได้อยู่ 2 ค่าคือ สูงและต่ำ (หรือ 1/0) มีค่าต่ำ เมื่อระดับของสัญญาณอยู่ในช่วง 0 ถึง 0.8 V, และมีค่าสูงเมื่อระดับสัญญาณอยู่ระหว่าง 2-5 V สัญญาณ ดิจิตอล ที่วัดมาได้ใช้แสดงสถานะ(State) และอัตรา (Rate) การเปลี่ยนสถานะของระบอบ

2.4.3.3.3 SIGNAL PROCESSING/CONDITIONING

เครื่องมือวัดจะส่งสัญญาณทางไฟฟ้าในรูปแบบมาตรฐานต่างๆซึ่งสัญญาณทางไฟฟ้าส่วนมากจะอยู่ในรูปของ ความต่างศักย์ไฟฟ้า (Voltage)หรือ กระแสไฟฟ้า (Ampere) โดยทั่วไปแล้ว มีสัญญาณมาตรฐาน เช่น 0-5V หรือ 4-20mA

2.4.3.4 Analog to Digital Conversion (A/D)

ทำหน้าที่แปลงสัญญาณข้อมูลที่ได้รับ, สัมผัสได้ เป็นข้อมูลทางไฟฟ้า เพื่อป้อนเข้าสู่การประมวลผล จึงเป็นขบวนการหนึ่งของการรับข้อมูล (Input Unit)เป็นกระบวนการอิเล็กทรอนิกส์ ที่สัญญาณแปรผันต่อเนื่อง (analog) ได้รับการแปลงให้เป็นสัญญาณดิจิตอล โดยไม่มีการลบข้อมูล

2.4.3.4.1 DAQ Software

ได้แก่ LabVIEW, Visual basic/Visual C++, C/C++,HP-VEE, MATLAB

2.4.4 จอแสดงผล μ LCD - 43PCT(GFX)

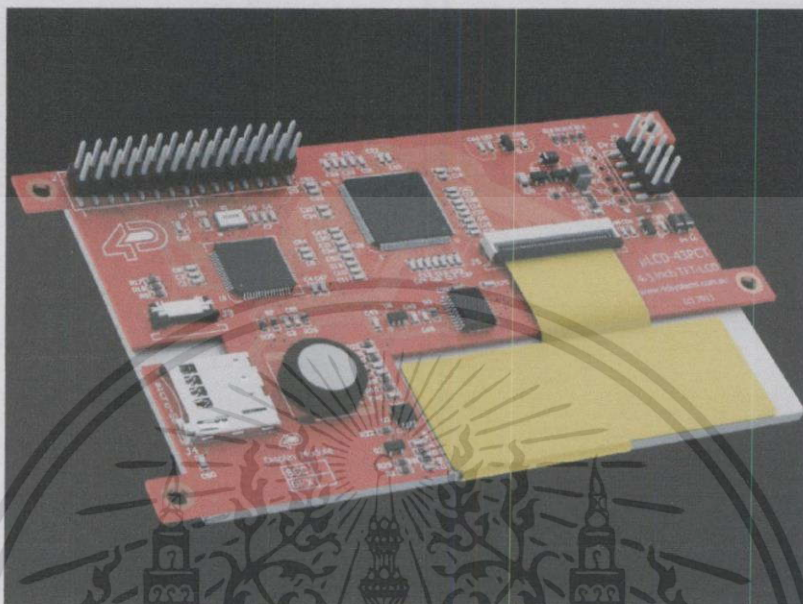
เป็นจอแสดงผลลำสมัยซึ่งมีช่วงคุณสมบัติหลากหลาย มีประสิทธิภาพและมีขนาดกะทัดรัด ส่วนสำคัญในการออกแบบคือ หน่วยประมวลผล PICASO-GFX2 ซึ่งทำงานโดยเครื่องยนต์เสมือนที่มีประสิทธิภาพสูงสุด(EVE) ซึ่งมีความหลากหลายของโปรแกรมและอุปกรณ์ ซึ่งเป็นอุปกรณ์ต่อพ่วงที่ถูกใส่เข้าไปในการออกแบบให้ใช้งานได้อย่างอิสระและปรับให้เข้ากับการทำงานต่างๆ ซึ่งมีคุณสมบัติประกอบไปด้วย

- จอสัมผัส 4.3" TFT 480x272

- โฟล์เสียน

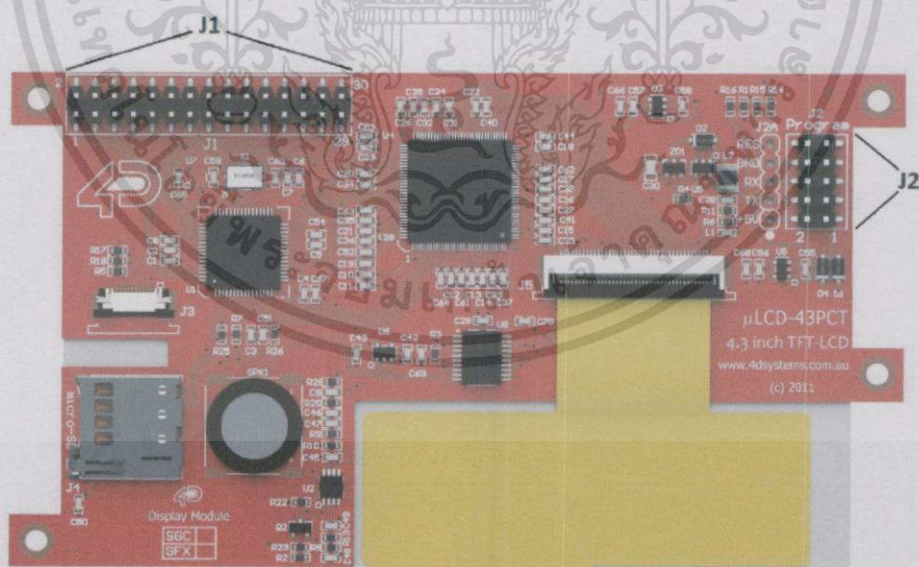
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถใช้ตัวอักษรได้หลากหลายของ Windows
- การทำงานในช่วง 4 ถึง 5.5 V
- มาตรฐาน ROHS



รูปที่ 2.14 แสดงวงจรของจอแสดงผล μ LCD-43PCT(GFX)

2.4.4.1 รายละเอียดของจอแสดงผล μ LCD-43PCT(GFX)



รูปที่ 2.15 แสดงรายละเอียดของวงจรของจอแสดงผล μ LCD-43PCT(GFX)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ตารางแสดงรายละเอียดของขาเอาต์พุต J2

ขาที่	สัญลักษณ์	อินพุต/ เอาต์พุต	รายละเอียด
1,2	Vin	P	ขาอินพุตรับแหล่งจ่ายหลัก ป้องกันการย้อนกลับอยู่ในช่วง 4 ถึง 5.5 โวลต์(ปกติถึง 5 โวลต์)
3,4	TX	O	ขาที่มีการสื่อสารแบบซิงโครนัส โดยเชื่อมกับไมโครคอนโทรลเลอร์ที่ขา Rx ไมโครคอนโทรลเลอร์ รับข้อมูลจากจอแสดงผลผ่านขานี้ ใช้ไฟ 5 โวลต์ขึ้นไป
5,6	RX	I	ขาที่มีการสื่อสารแบบซิงโครนัส โดยเชื่อมกับไมโครคอนโทรลเลอร์ที่ขา Tx ไมโครคอนโทรลเลอร์ ส่งคำสั่งและข้อมูลไปยังจอแสดงผลผ่านขานี้ ใช้ไฟ 5 โวลต์ขึ้นไป
7,8	GND	P	ขากราวด์
9,10	RES	I	ไมโครคอนโทรลเลอร์ ส่งสัญญาณรีเซต ตัวต้านทานภายใน 3.3 โวลต์ ถึง 4.7 โวลต์ สัญญาณพัลส์กระทำที่น้อยกว่า 2 μ s จะที่กักรีเซต ถ้าต้องการทำการรีเซตจากภายนอก เพียงใช้วงจรคอนเสดเตอร์ ขานี้จะไม่ทำงานสภาวะต่ำดดยเงื่อนไขภายในไมโครคอนโทรลเลอร์ ควบคุมขานี้โดยใช้ขาใดขาหนึ่งในการเปิดคอนเสดเตอร์

ตารางที่ 2.4 ตารางแสดงรายละเอียดของขาอินพุต J1

ขาที่	สัญลักษณ์	อินพุต/ เอาต์พุต	รายละเอียด
1	IO1	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต 1
3	IO2	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต 2
3	SCL	เอาต์พุต	ส่งสัญญาณนาฬิกาเอาต์พุต I2C
5	IO3	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต 3
6	SDA	อินพุต/ เอาต์พุต	สัญญาณข้อมูล I2C สองทิศทาง
7	IO4/BUS_RD	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต 4 และ สายBUSอ่าน
9	IO5/BUS_WR	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต 5 และ สายBUSเขียน
11	GND	แหล่งจ่าย	กราวด์ของแหล่งจ่ายไฟ
12	VCC	แหล่งจ่าย	แหล่งจ่ายไฟเข้าด้านขาอินพุต 4.5 โวลต์ ถึง 5.5 โวลต์
13	GND	แหล่งจ่าย	กราวด์ของแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15	BUS7	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต BUS บิตที่ 7
16	RESET	อินพุต	รีเซ็ตจากไมโครคอนโทรลเลอร์ ขานี้สัญญาณต่ำสำหรับ 2 μ s หรือยาวกว่า ในการทำการการรีเซ็ต
17	BUS6	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต BUS บิตที่ 6
19	BUS5	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต BUS บิตที่ 5
20	3.3V	แหล่งจ่าย	กำหนดเอาต์พุต 3.3 โวลต์ กระแสผ่านสูงสุด 400 มิลลิแอมแปร์
21	BUS4	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต BUS บิตที่ 4
23	BUS3	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต BUS บิตที่ 3
24	RX0	อินพุต	การสื่อสารแบบซิงโครนัส ที่ขารับข้อมูล พอร์ต 0 COM0 Rx
25	BUS2	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต BUS บิตที่ 2
26	TX0	เอาต์พุต	การสื่อสารแบบซิงโครนัส ที่ขาส่งข้อมูล พอร์ต 0 COM0 Rx
27	BUS1	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต BUS บิตที่ 1
28	TX1	เอาต์พุต	การสื่อสารแบบซิงโครนัส ที่ขารับข้อมูล พอร์ต 1 COM1 Rx
29	BUS0	อินพุต/ เอาต์พุต	อินพุต เอาต์พุต BUS บิตที่ 0
30	RX1	อินพุต	การสื่อสารแบบซิงโครนัส ที่ขาส่งข้อมูล พอร์ต 1 COM1 Rx
NC	NC	--	ขาที่ไม่ใช้งานได้แก่ 2,8,10,14,18,22

2.4.4.2 การติดต่อสื่อสารระหว่างขาของอุปกรณ์

2.4.4.2.1 พอร์ตอนุกรม COM0, COM1 UARTS

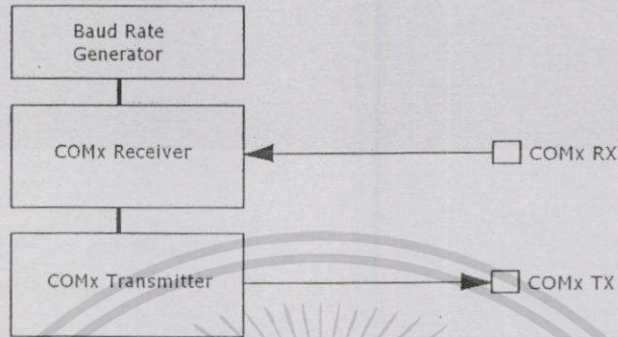
PICASO-GFX2 มีอุปกรณ์สองชนิดที่เป็นพอร์ตอนุกรมสื่อสารแบบซิงโครนัส ที่ซึ่งสามารถสื่อสารกับ อุปกรณ์ภายนอกโดยการต่ออนุกรม โดยอ้างถึงพอร์ตอนุกรมได้แก่ COM0 และ COM1

คุณสมบัติพื้นฐาน

- ส่งข้อมูลและรับข้อมูล แบบทางคู่ (Full-Duplex) ขนาด 8 บิต
- รูปแบบของข้อมูล 8 บิต ไม่มีความเท่าเทียม มี 1 บิตหยุด
- อัตราบอर्डิสระจาก 300 บอर्डขึ้นไป ถึง 256 กิโลบอर्ड

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีไบต์เดียวในการส่งข้อมูลและรับข้อมูล หรือ การบริการพักข้อมูลระหว่างส่ง (Buffer) คุณสมบัติของการพักข้อมูลคือจะรันไปยังการจับพื้นหลัง และการพักข้อมูลแบบอนุกรมปราศจากการทำงานผู้ใช้ที่กำลังสำรวจอย่างต่อเนื่องของพอร์ตอนุกรม



รูปที่ 2.16 พอร์ตอนุกรม COM0, COM1 UARTS

ไบต์เดียวในการส่งข้อมูลและรับข้อมูลแบบอนุกรม ประกอบด้วย บิตเริ่มต้น บิตหยุดขนาด 8 บิตของข้อมูล โดยบิตเริ่มต้นจะเริ่มที่ 0 เสมอ ขณะที่บิตหยุดจะเริ่มที่ 1 ซึ่ง 2 บิตที่มีนัยสำคัญต่ำสุด (Least Significant Bit, Bit0) ถูกส่งออกอันดับแรกที่บิตเริ่มต้น



รูปที่ 2.17 แสดง Timing Diagram ในการส่งข้อมูลไบต์เดียว

- ขา TX0 (การส่งข้อมูลผ่านพอร์ตอนุกรม COM0)

การสื่อสารอนุกรมแบบซิงโครนัสโดยการส่งข้อมูล(TX) ผ่านพอร์ต COM0 โดยขาทำการเชื่อมต่ออนุกรมกับขารับสัญญาณข้อมูลของอุปกรณ์ภายนอก(RX) ใช้แรงดัน 5.0 โวลต์

- ขา RX0 (การรับข้อมูลผ่านพอร์ตอนุกรม COM0)

การสื่อสารอนุกรมแบบซิงโครนัสโดยการรับข้อมูล(RX) ผ่านพอร์ต COM0 โดยขาทำการเชื่อมต่ออนุกรมกับขาส่งสัญญาณข้อมูลของอุปกรณ์ภายนอก(TX) ใช้แรงดัน 5.0 โวลต์

- ขา TX1(การส่งข้อมูลผ่านพอร์ตอนุกรม COM1)

การสื่อสารอนุกรมแบบซิงโครนัสโดยการส่งข้อมูล(TX) ผ่านพอร์ต COM1 โดยขาทำการเชื่อมต่ออนุกรมกับขารับสัญญาณข้อมูลของอุปกรณ์ภายนอก(RX) ใช้แรงดัน 5.0 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา RX1 (การรับข้อมูลผ่านพอร์ตอนุกรม COM1)

การสื่อสารอนุกรมแบบซิงโครนัสโดยการรับข้อมูล(RX) ผ่านพอร์ต COM0 โดยขาทำการ เชื่อมต่ออนุกรมกับขาส่งสัญญาณข้อมูลของอุปกรณ์ภายนอก(TX) ใช้แรงดัน 5.0 โวลต์

2.4.4.2.2 GPIO - General Purpose IO Interface

มี 13 ขาเนกประสงค์ (GPIO) ที่ง่ายต่อการใช้งาน ประกอบไปด้วยกลุ่มของ IO1 ถึง IO5 และ BUS0 ถึง BUS7 ในกลุ่มของขาอินพุต เอาท์พุต 5 ขา(IO1 ถึง IO5) ให้การทำงานของแต่ละขาเป็นไปอย่างยืดหยุ่น ขณะที่อีก 8 ขา(BUS0 ถึง BUS7) ถูกรู้จักที่ GPIO BUS ให้ร่วมการทำงานสำหรับไบต์ IO4 ถึง IO5 ยังทำหน้าที่เป็นสัญญาณในการควบคุม GPIO BUS ที่ซึ่ง GPIO BUS สามารถอ่านและเขียนโดยสัญญาณพัลส์ต่ำ คือ IO4/BUS_RD หรือ IO5/BUS_WR สำหรับการอ่านและการเขียนตามลำดับ

- ขา IO1 ถึง IO3

ขาอินพุต เอาท์พุตเนกประสงค์ แต่ละขาสามารถตั้งค่าอินพุต หรือค่าเอาท์พุต โดยแหล่งจ่าย ทำการรีเซ็ตที่ขาอินพุตทั้งหมด

- ขา IO4/BUS_RD (GPIO IO4 หรือ ขาBUS_RD)

ขาเนกประสงค์ IO4 ใช้สำหรับ สัญญาณ BUS_RD ในการอ่านและล๊อคข้อมูลเข้าไปในพอร์ตเนกประสงค์(GPIO)แบบขนาน BUS0 ถึง BUS7

- ขา IO5/BUS_WR (GPIO IO5 หรือ ขาBUS_WR)

ขาเนกประสงค์ IO5 ใช้สำหรับ สัญญาณ BUS_WP ในการเขียนและล๊อคข้อมูลเข้าไปในพอร์ตเนกประสงค์(GPIO)แบบขนาน BUS0 ถึง BUS7

- ขาBUS0 ถึง BUS7 (GPIO 8 บิต BUS)

BUS ในการสื่อสารพอร์ตเนกประสงค์แบบขนานขนาด 8 บิต

หมายเหตุ ขา GPIO ทั้งหมดใช้ไฟขนาด 5. โวลต์

2.4.4.2.3 System Pins

- V_{CC} (แรงดันไฟฟ้าอินพุต) J1 ขาที่ 1/2, J2 ขาที่ 12

แหล่งจ่ายแรงดันไฟฟ้าด้านขาอินพุต ขานี้ต้องเชื่อมต่อการควบคุมแรงดันในช่วง 4.0 โวลต์ ถึง 5.5 โวลต์ DC ปกติจะทำงานในช่วงแรงดัน 5 โวลต์

- 3.3 V_{out} (3.3V ควบคุมด้านเอาท์พุต)

วงจรรภายนอก ที่ซึ่งเรียกร่องให้ควบคุมแหล่งจ่าย 3.3 V สามารถขับเคลื่อนขึ้นโดยขานี้ กระแสสูงสุดอยู่ที่ 400 มิลลิแอมแปร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- GND (กราวด์) J1 ขาที่ 7/8, J2 ขาที่ 11/13
ขากราวด์ของอุปกรณ์ ขาเหล่านี้ต้องเชื่อมต่อกับกราวด์
- RESET (ขารีเซ็ตของตัวไมโครคอนโทรลเลอร์) J1 ขาที่ 9/10, J2 ขาที่ 16
ขารีเซ็ตของตัวไมโครคอนโทรลเลอร์ กระทำที่สัญญาณพัลส์ต่ำ ที่มากกว่า 2.0 μs จะทำการรีเซ็ต
ภายในดึงขึ้นถึง 3.3 V โดยตัวต้านทาน 4.7K

2.4.4.3 หน่วยความจำ

บนบอร์ดมีช่องสำหรับเสียบหน่วยความจำ SD Card หน่วยความจำถูกถ่ายโอนข้อมูลมัลติมีเดียจำพวก ภาพ ภาพการเคลื่อนไหว นอกจากนี้ยังใช้สำหรับการบันทึกข้อมูลหน่วยความจำ FAT 16 Format ไมโครคอนโทรลเลอร์ PICASO-GFX2 ถูกออกแบบให้มีมาตรฐานให้สามารถใช้ได้กับ หน่วยความจำขนาด 2 GB สามารถจัดรูปแบบระบบไฟล์แบบ FAT 16 สำหรับการดำเนินการที่เกี่ยวข้องกับไฟล์ FAT หน่วยความจำจะต้องทำการจัดรูปแบบ FAT16 ทำโดยระบบในคอมพิวเตอร์ซึ่งจะเป็นตัวอ่านเมื่อเลือกให้จัดรูปแบบเป็น FAT ดังนั้นหน่วยความจำจึงสามารถใช้งานได้กับ PICASO - GFX2



รูปที่ 2.18 หน่วยความจำ SD CARD

2.4.4.4 เสียง (Audio)

ไฟล์เสียงที่มีความเฉพาะตัวรองรับกับ PICASO-GFX2 ซึ่งจะมีผลในช่วงของการประมวลผลแบบกราดิเอนท์ PWM เป็นสัญญาณเสียงที่รับรองคุณภาพเสียงที่ระดับเสียงอยู่ในช่วง 8 ถึง 127 โคจรสร้างพื้นฐานอนุญาตให้ผู้ใช้ดำเนินการกับไฟล์เสียง การทำงานของเสียงสามารถดำเนินการพร้อมกับการดำเนินการตามคำแนะนำอื่นๆ

2.4.4.5 ข้อควรระวังสำหรับจอแสดงผล

- หลีกเลี่ยงการแสดงรูปภาพ หรือวัตถุที่เหมือนกันบนหน้าจอเป็นเวลานานๆ สิ่งนี้เป็นสาเหตุให้เกิดการเผาไหม้ภายในจอแสดงผล ที่ซึ่งเป็นปัญหาพื้นฐานกับเทคโนโลยีจอแสดงผลทุกชนิด หากหน้าจอว่างเปล่าหรือมืดให้หยุดการเชื่อมต่อ
- ความชื้นและน้ำสามารถทำลายจอแสดงผล ความชื้นบนพื้นผิวของจอแสดงผลจะทำให้ขั้วไฟฟ้าเป็นสนิม แก้โดยการปัดความชื้นเบาๆ หรือปล่อยให้จอแสดงผลแห้งก่อนที่จะใช้งาน
- ฝุ่นและคราบความมันจากรอยนิ้วมือสามารถเปื้อนพื้นผิวของจอแสดงผลได้อย่างง่าย ดังนั้นใช้ผ้าที่ไม่เป็นขลุ่ย เช็ดอย่างเบาๆบนผิวจอแสดงผล

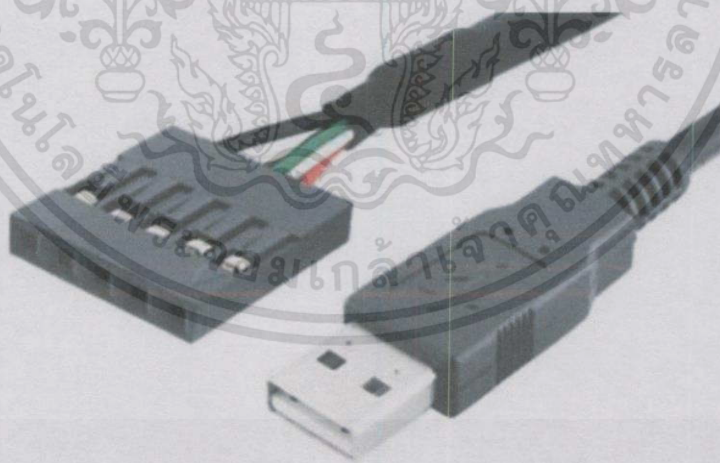
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การแสดงของจอแสดงผลจะต้องไม่อยู่ภายใต้อุณหภูมิที่สูงและไม่มีความชื้น
- ไม่ยุ่งกับสายเคเบิลที่ซึ่งต่อกับบอร์ดคอนโทรล ซึ่งอาจมีผลกระทบต่อการทำงานของจอแสดงผลกับวงจรที่กำลังทำงานทำให้เกิดการล้มเหลวได้
- จอแสดงผลอาจอ่อนไหวจากการกระตูกทางกล แรงที่กระทำต่อโมดูลมีผลทำให้แถบบนจอแสดงผลเสียหายได้รวมทั้งเซลล์บนหน้าจอก็เกิดการแตกหัก

2.4.5 4D Programming Cable

4D Programming Cable เป็น USB ที่ต่ออนุกรมกับ สายเคเบิล TTL UART ประกอบไปด้วย Silabs CP2102 USB ต่อกับ Serial UART bridge IC ที่ซึ่งจัดการการส่งสัญญาณและโปรโตคอล สายเคเบิลมีความเร็วและใช้ในการสื่อสารได้ง่ายของอุปกรณ์ที่ซึ่งเรียกร่องการสื่อสารอนุกรม TTL กับ USB โดย Programming Cable เป็นอุปกรณ์ที่มีความจำจึงถูกนำไปใช้สำหรับไฟล์ programming PmmC และ สำหรับการดาวโหลด 4DGL ไปยัง หน่วยประมวลผล PICASO ที่ซึ่งติดตั้งอยู่ใน 4D โมดูล

สายเคเบิลประกอบด้วยบอร์ดวงจรไฟฟ้าอิเล็กทรอนิกส์ขนาดเล็กภายใน และใช้ชิป CP2102 อีกทั้งใช้วงจรรีเซตพิเศษ ที่ซึ่งถูกห่อหุ้มภายในหัวเชื่อมต่อUSBส่วนปลายของสายเคเบิล สายเคเบิลได้รับรองให้เป็นสินค้าสำเร็จรูป ที่อยู่ภายใต้ขอบเขตของระเบียบ RoHS และยังใช้หัวตัวเมีย 5 ขา ที่ระดับ TTL ของ 3.3 โวลต์ สำหรับขา RX และขา TX สำหรับส่วนหัวที่มี 5 ขา ได้ระบุชื่อสัญญาณ อาทิเช่น +5V, RX, TX, GND, RESET ที่ต้องสัมพันธ์กับหัว programming บนอุปกรณ์ 4D สายเคเบิลใช้แหล่งจ่าย +5V DC ถึงการขับเคลื่อนไปยังอุปกรณ์เป้าหมาย สายเคเบิลความยาว 1 เมตร และวงจรกำลัง USB สนับสนุนอัตราข้อมูล 300bps ถึง 1 Mbps



รูปที่ 2.19 แสดงส่วนหัวและส่วนปลายของสาย Programming Cable

2.4.5.1 ข้อมูลเกี่ยวกับสายเคเบิล

- USB 2.0 ความเร็วสูงสุดที่ 12Mbps
- อุปกรณ์ หรือ Xon/Xoff ร่วมกันสนับสนุนอัตราข้อมูล 300bps ถึง 1 Mbps
- UART สนับสนุนข้อมูล 5 ถึง 8 บิต, บิตหยุด 1-2, คู่/คี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการรวบรวม EEPROM สำหรับ รหัสของ Vendor รหัสสินค้า หมายเลข และหมายเลขรุ่น

- พอร์ต COM ของอุปกรณ์ อนุญาตให้ดำเนินการร่วมกับฟังก์ชันการทำงานของพอร์ต COM ของคอมพิวเตอร์

- ขนาดเล็ก วงจรไฟฟ้าภายในหัวเชื่อมต่อ

2.4.5.2 ชนิดของฟังก์ชันการทำงาน

- เชื่อมต่อ UART ของไมโครคอนโทรลเลอร์ กับ USB

- USB ต่อกับ เครื่องแปลงแบบ RS-232

- USB ต่อกับ เครื่องแปลงแบบ RS-422/ RS-485

- อีเทอร์เน็ตอุปกรณ์แบบ RS-232

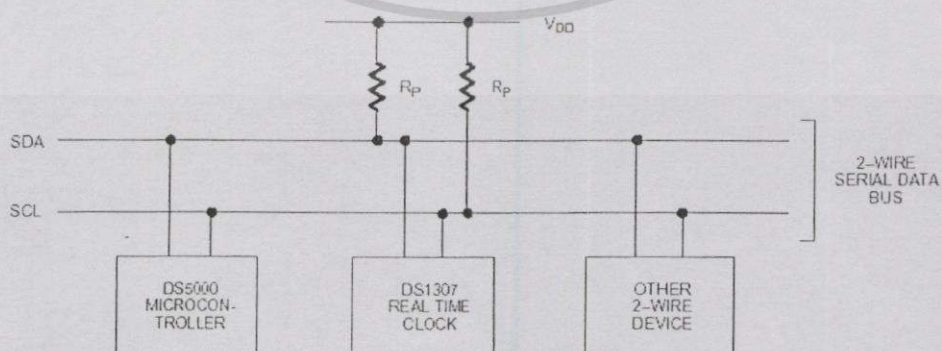
สายเคเบิลถูกออกแบบให้มีความเฉพาะเจาะจงในการเชื่อมต่อสื่อสารกับ อุปกรณ์ 4D ที่ซึ่งใช้หัว programming 5 ขา ถ้าหากไม่ใช้ฟังก์ชันการทำงานของอุปกรณ์ 4D จะใช้ TX และ RX จากสายเคเบิลเชื่อมต่อไปยังอุปกรณ์ต่อพ่วง นั่นคือ TX ของสายเคเบิล เชื่อมต่อไปยัง RX ของอุปกรณ์เป้าหมาย และจาก RX ของสายเคเบิลเชื่อมต่อไปยัง TX ของอุปกรณ์เป้าหมาย

2.5 I²C Bus (Inter Integrate Circuit Bus)

การติดต่อสื่อสารกันระหว่างไมโครคอนโทรลเลอร์ PICASO กับ ชิพ MCP3424 สามารถใช้วิธีการติดต่อแบบ I²C ซึ่งเป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้สื่อสารระหว่างไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก โดยใช้สายสัญญาณเพียง 2 เส้นคือ serial data (SDA) และสาย serial clock (SCL) ซึ่งเชื่อมต่ออุปกรณ์ จำนวนหลายๆ ตัว เข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น

2.5.1 การเชื่อมต่ออุปกรณ์แบบ I²C BUS

I²C BUS ใช้สายสัญญาณ 2 เส้น คือ SCL, SDA สำหรับติดกับอุปกรณ์แบบ 2 ทิศทาง โดยที่ขาสัญญาณทั้ง 2 จะต้องต่อกับตัวต้านทานแบบ pull up 2-10K เนื่องจากเอาต์พุตมีลักษณะเป็น แบบ Open Darin หรือเป็นแบบ Open Collector เพื่อให้เอาต์พุตเชื่อมต่อกันได้หลายตัว



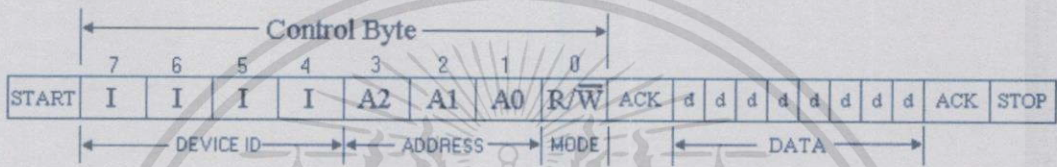
รูปที่ 2.20 ลักษณะการเชื่อมต่ออุปกรณ์แบบ I²C BUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 การเขียน – อ่านข้อมูลกับอุปกรณ์แบบ I²C BUS

การรับ-ส่งข้อมูลแบบ BUS MCU จะเริ่มต้นการส่งข้อมูลด้วยการ

- ส่งสถานะเริ่มต้น (START Conditions) เพื่อแสดงการขอใช้บัส
- แล้วตามด้วย รหัสควบคุม (Control Byte) ซึ่งประกอบ ด้วยรหัส ประจำตัว อุปกรณ์ Device ID, Device Address และ Mode ในการเขียนหรืออ่านข้อมูล
- เมื่ออุปกรณ์ รับทราบว่า MCU ต้องการ จะติดต่อกับก็ต่อส่งสถานะรับรู้ (Acknowledge) หรือแจ้งให้ MCU รับรู้ว่าข้อมูลที่ได้ส่งมามีความถูกต้อง
- และเมื่อสิ้นสุดการส่งข้อมูล MCU จะต้องส่ง สถานะสิ้นสุด (STOP Conditions) เพื่อบอกกับอุปกรณ์ว่า สิ้นสุดการใช้บัส



รูปที่ 2.21 รูปแบบการเขียน/อ่านข้อมูลแบบ I²C BUS

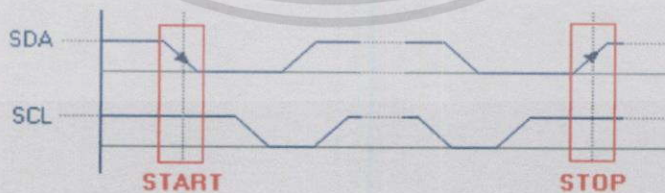
2.5.3 สถานะบัสว่าง

คือเมื่อบัสไม่ได้ถูกใช้งาน ทั้ง SCL และ SDA จะเป็น 1 ทั้งคู่

2.5.4 การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I²C BUS (START and STOP Conditions)

ลักษณะการกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I²C BUS

- เมื่อต้องการส่งข้อมูล MCU จะต้องส่งสถานะเริ่มต้น (START Conditions) คือให้ SDA เปลี่ยนจาก 1 มาเป็น 0 ในขณะที่ SCL มีค่าเป็น 1
- เมื่อสิ้นสุดการการใช้บัส MCU จะต้องส่งสถานะสิ้นสุด (STOP Conditions) คือให้ SDA เปลี่ยนจาก 0 มาเป็น 1 ในขณะที่ SCL มีค่าเป็น 1



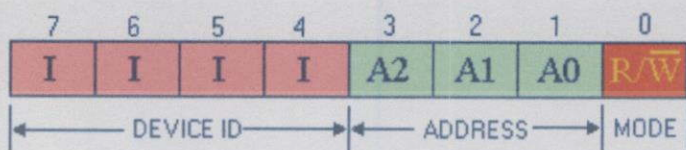
รูปที่ 2.22 START and STOP Conditions

2.5.5 รหัสควบคุมของ I²C BUS (Control Byte)

รหัสควบคุมของ I²C BUS ประกอบด้วยรหัสประจำตัวของอุปกรณ์ (Device ID) ประกอบด้วยบิต 1-7 และบิต 0 เป็นบิตควบคุมการเขียนอ่าน

- รหัสประจำตัวของอุปกรณ์ ประกอบด้วยรหัสประจำตัวจากผู้ผลิต Product ID 4 บิต (บิต 4-7) ที่เปลี่ยนแปลงแก้ไขไม่ได้ และ Device Address 3 บิต (บิต 1-3) ซึ่งผู้ใช้ สามารถ กำหนด เองได้ รวมแล้วเป็นรหัส 7 บิต ใช้ระบุตัวอุปกรณ์ ที่ต่ออยู่บนบัส จะมีค่าซ้ำกันไม่ได้

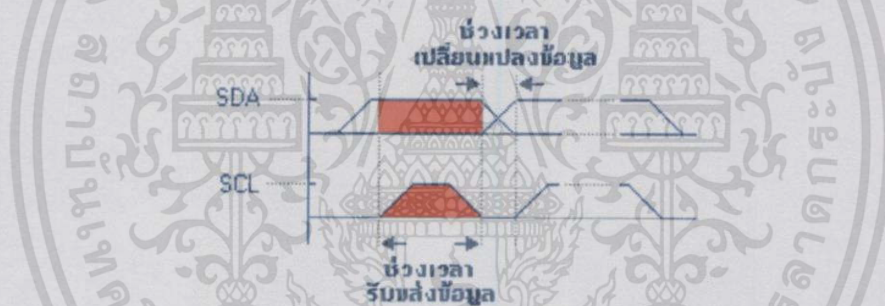
- บิตควบคุมการเขียนอ่าน (Mode) บิต 0 เมื่อ MCU ต้องการเขียนข้อมูลไป ยังอุปกรณ์ก็กำหนดให้บิตนี้เป็น 0 และเมื่อต้องการ อ่านข้อมูล จากอุปกรณ์ ก็กำหนดให้บิตนี้เป็น 1



รูปที่ 2.23 I²C BUS (Control Byte)

2.5.6 ช่วงเวลารับส่งบิตข้อมูลของ I²C BUS

- สถานะการรับ-ส่งข้อมูล จะกระทำในขณะที่ขา SCL เป็น 1
- สถานะการเปลี่ยนแปลงข้อมูล จะกระทำในขณะที่ขา SCL เป็น 0



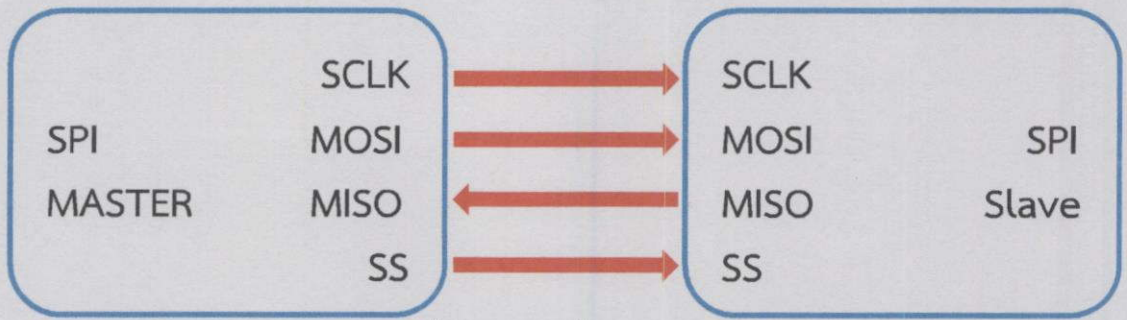
รูปที่ 2.24 ช่วงเวลารับส่งบิตข้อมูลของ I²C BUS

2.6 การสื่อสารแบบ SPI

เป็นการติดต่อสื่อสารกันระหว่างไมโครคอนโทรลเลอร์ PIXASO กับ SD Card เลือกใช้วิธีการติดต่อแบบ SPI เพราะการติดต่อสื่อสารเป็นแบบ ซิงค์โครนัส (Synchronous) เหมาะสำหรับใช้ใน ระยะสั้นๆ ซึ่ง ไมโครคอนโทรลเลอร์ส่วนใหญ่จะติดต่อแบบนี้ อีกทั้ง ยังมีความเร็วในการรับส่ง-ข้อมูล มากกว่าการสื่อสารแบบอื่น

SPI (Serial Peripheral Interface) คือ การสื่อสารอนุกรมแบบซิงค์โครนัส (Synchronous) เป็นรูปแบบที่ใช้วิธีส่ง ข้อมูล โดยใช้สัญญาณ Clock มาเป็นตัวกำหนดจังหวะในการรับส่งข้อมูล

SPI ทำงานในรูปแบบที่ให้อุปกรณ์ตัวหนึ่งทำหน้าที่เป็นมาสเตอร์ ในขณะที่อีกตัวหนึ่งทำหน้าที่เป็นสลาฟ และส่งข้อมูลในโหมด Full-duplex ซึ่งหมายความว่า สัญญาณสามารถส่งหากัน ระหว่าง มาสเตอร์ และ สลาฟ ได้อย่างต่อเนื่อง ในการสื่อสารแบบ SPI นี้ไม่มีมาตรฐานกำหนดตายตัวว่าข้อมูลที่ส่งหากันต้องอยู่ในรูปแบบหรือฟอร์แมตแบบไหนเป็นการคิดโปรโตคอลการสื่อสารกันเอง



รูปที่ 2.25 การสื่อสารแบบ SPI

ตารางที่ 2.5 แสดงการทำงานของขาแต่ละขา ที่เป็นการสื่อสารแบบ SPI

เส้น	ชื่อ	การทำงาน
SCLK	Serial Clock	สัญญาณนาฬิกาที่มาสเตอร์เป็นผู้ส่ง
MOSI	มาสเตอร์เอาต์พุต	เอาต์พุตจากมาสเตอร์ที่ส่งให้สลาฟ
MISO	มาสเตอร์อินพุต	เอาต์พุตจากสลาฟที่ส่งให้มาสเตอร์
SS	Slave Select	เป็นสัญญาณที่มาสเตอร์ใช้เป็นตัวเลือกว่าจะติดต่อกับสลาฟใด

master ก็ ทำหน้าที่เป็นตัวควบคุมการสื่อสารทั้งหมด โดยควบคุมการสื่อสารตามสัญญาณนาฬิกา ตัวมาสเตอร์จะเป็นตัวที่ตัดสินใจ รับ หรือ ส่งข้อมูล ภายในการสื่อสาร จะเป็นการสื่อสารแบบ full duplex ในการนำส่งข้อมูล

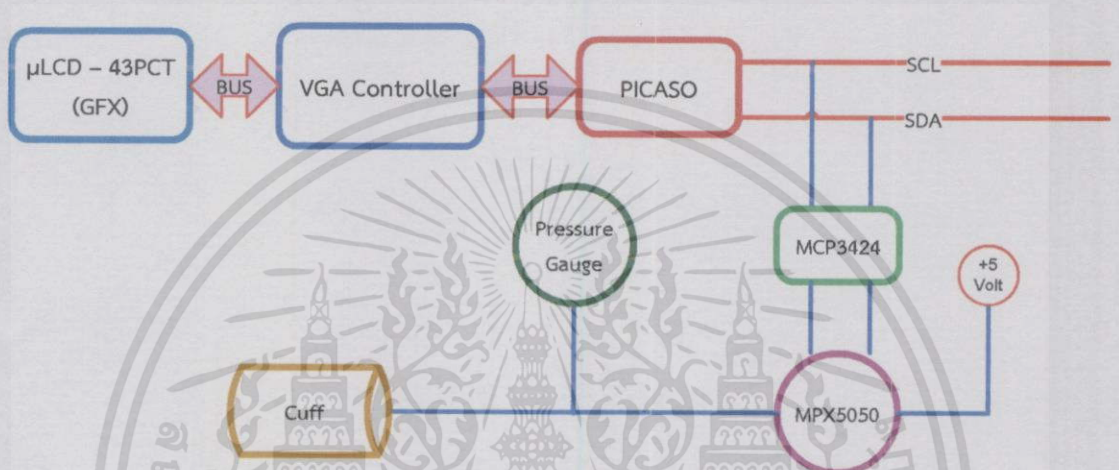
สัญญาณเส้น SS หรือ Slave select ในกรณี ที่เรามีตัว slave มากกว่า 1 ตัว โดยการทำให้เส้น SS มีระดับสัญญาณเป็น Low เมื่อต้องการติดต่อกับ Slave ตัวใด จากรูปด้านล่างหากเราต้องการติดต่อสื่อสารกับ Slave ตัวใด ก็เพียงทำให้สัญญาณ SS ของ Slave ตัวนั้น มีระดับสัญญาณเป็น Low

บทที่ 3

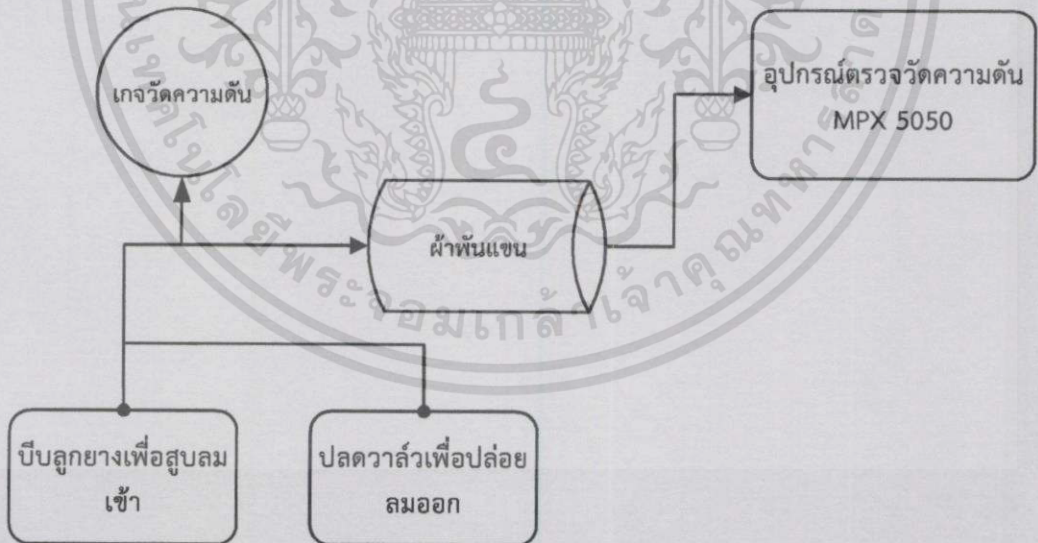
วิธีการดำเนินงาน

3.1 การทำงานร่วมกันของอุปกรณ์

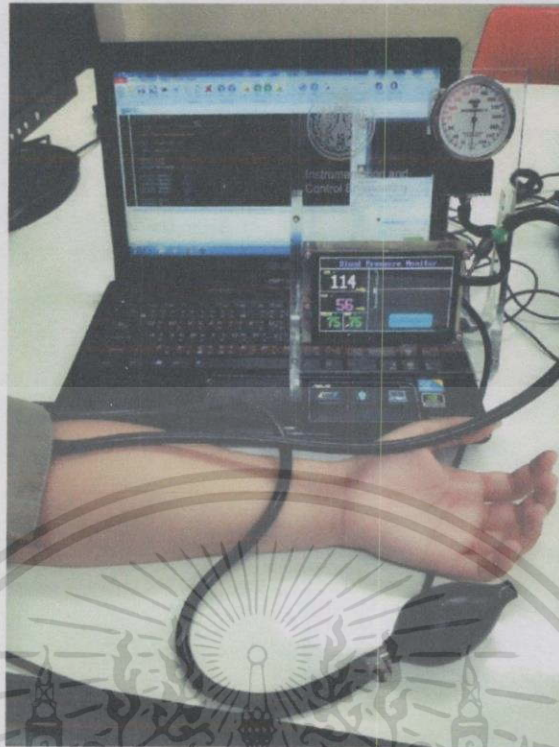
ทำการเชื่อมต่ออุปกรณ์แต่ละตัวลงบนแผงวงจรอิเล็กทรอนิกส์และเชื่อมต่ออุปกรณ์อื่นๆ ที่เกี่ยวกับแรงดันลมโดยมีแผนภาพดังนี้



รูปที่ 3.1 แสดงแผนภาพการเชื่อมต่ออุปกรณ์แต่ละตัว



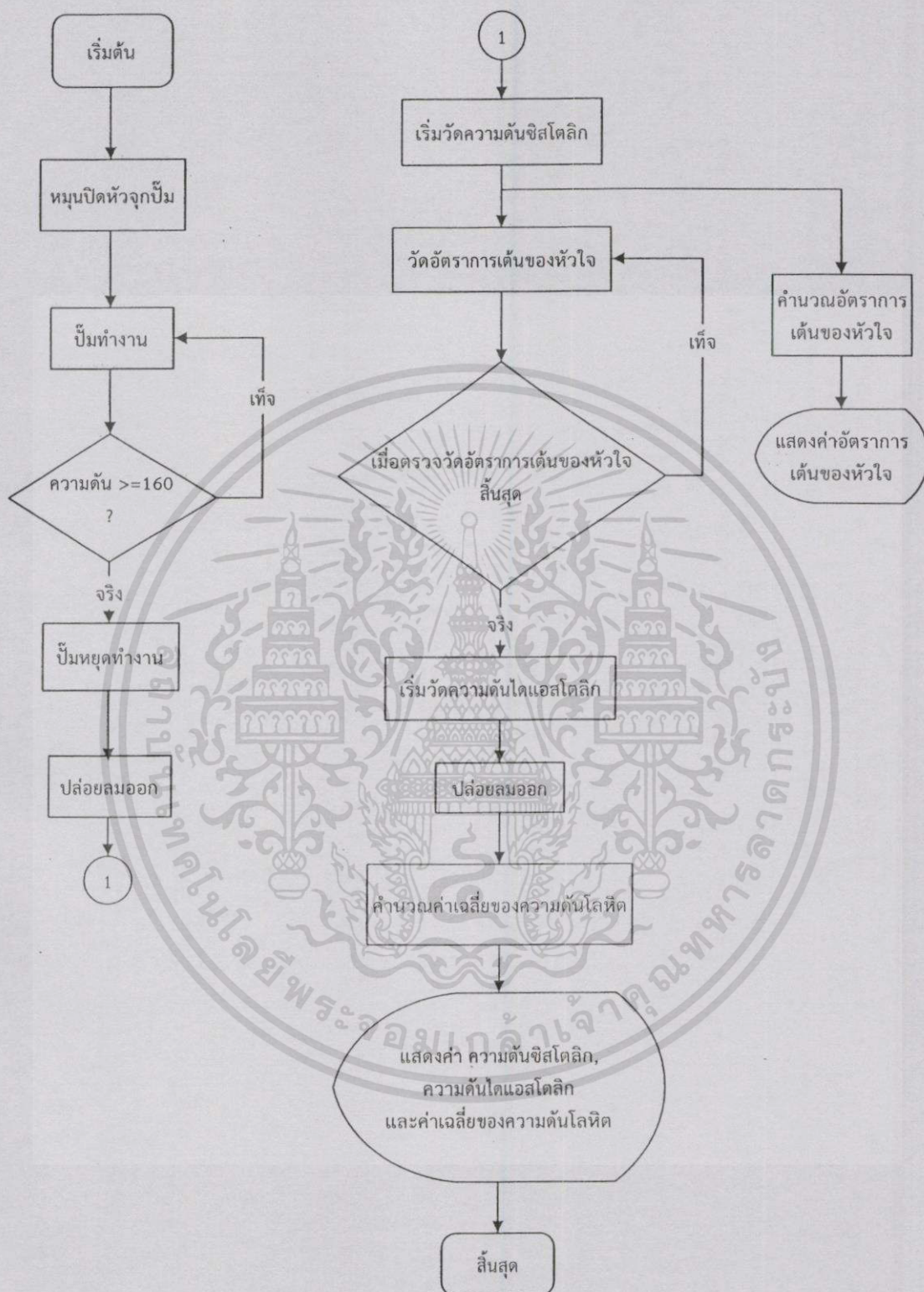
รูปที่ 3.2 แสดงท่อลมเข้ากับอุปกรณ์ต่างๆ



รูปที่ 3.3 เครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO

3.2 ขั้นตอนการทำงานของโปรแกรม

การเขียนโปรแกรมจะใช้ภาษากราฟิก 4DGL เพื่อสั่งให้ไมโครคอนโทรลเลอร์รับค่าจากอุปกรณ์ตรวจวัดความดัน MPX5050 นำมาประมวลผลและส่งค่ามายังจอแสดงผล โดยส่วนของโปรแกรมจะแบ่งออกเป็น 6 ส่วนด้วยกัน คือ



รูปที่ 3.4 แสดงแผนผังการทำงานสำหรับเขียนโปรแกรม

3.2.1 โปรแกรมหลัก

```
func PlotGraph (var BP var DBP var INT)
```

```
var Pulse;
```

เอกสารนี้เป็นเอกสารที่ส่ง BP ไปที่ $(BP*66)/100$; // กำหนดค่าลงบนตัวร้ายที่แสดงผลบนหน้าจอลiquid crystal display ด้านการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (BP >= 66) BP := 66;
BP := 66-BP;

```

if (ABS(DBP) >= 1) DBP := DBP*(10); //ฟังก์ชันคณิตศาสตร์ โดยคำสั่งนี้จะทำให้ค่าที่ได้นั้นเป็นค่าสัมบูรณ์

```

if (DBP >= 60) DBP := 60;
if (DBP <= -60) DBP := -60;
DBP := 60-DBP;

```

```

if (INT >= 60) INT := 60;
if (INT <= -60) INT := -60;
INT := 60-INT;

```

```

XPOS := XPOS + 1;
if (XPOS >= 265) XPOS := 0;

```

```

gfx_RectangleFilled(X_ref+XPOS,Y_ref,X_ref+XPOS+3,Y_ref+130,BLACK);
//gfx_Line(X_ref+XPOS-
1,Y_ref+BP_LST,X_ref+XPOS,Y_ref+BP,YELLOW);
gfx_Line(X_ref+XPOS-
1,Y_ref+DBP_LST,X_ref+XPOS,Y_ref+DBP,WHITE);
//gfx_Line(X_ref+XPOS-
1,Y_ref+IntegralLST,X_ref+XPOS,Y_ref+INT,YELLOW);
BP_LST := BP;
DBP_LST := DBP;
IntegralLST := INT;
endfunc

```

```

func CheckPeriod()

```

```

if (AnalyzeFLG==1)
    Period := 2000 - sys_GetTimer(TIMER0);
    sys_SetTimer(TIMER0,2000);
    Barray[0]:=Barray[1];
    Barray[1]:=Barray[2];
    Barray[2]:=Barray[3];
    Barray[3]:=Barray[4];
    Barray[4]:=Period;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Period := (Barray[0]+Barray[1]+Barray[2]+Barray[3]+Barray[4])/5;
    endif
endfunc

```

```

func CheckPeak()

```

```

    if (PulseCount >= 1 && PS_Dispatch >= 50 && AnalyzeFLG == 1 && PS_Dispatch
< PS_DispatchLST)

```

```

        if (DInput >= DInputLST)

```

```

            DPeak_Plus := DInput;

```

```

        else

```

```

            PeakFLG := 1;

```

```

            if (DInput <= DInputLST)

```

```

                DPeak_Minus := DInput;

```

```

            endif

```

```

        endif

```

```

        if (PeakFLG==1)

```

```

            DInputPeak := ABS(DPeak_Plus) - ABS(DPeak_Minus);

```

```

            Aarray[PeakCount] := DInputPeak;

```

```

            Parray[PeakCount] := PS_Dispatch;

```

```

            Darray[PulseCount] := PS_Dispatch;

```

```

            PeakCount++;

```

```

            PulseCount++;

```

```

            to(COM0);print(DInputPeak,"",PS_Dispatch,"\\n");

```

```

            PeakFLG := 0;

```

```

        endif

```

```

    endif

```

```

    DInputLST := DInput;

```

```

    PS_DispatchLST := PS_Dispatch;

```

```

endfunc

```

```

func CheckPulse()

```

```

    if (DInput != 0)

```

```

        if (DInput > 1) PrsState := 1;

```

```

        if (DInput < 1) PrsState := 0;

```

```

        State := (PrvState ^ PrsState) & (! PrvState);

```

```

        if (State == 1)

```

```

            gfx_CircleFilled (20,20,6,RED);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (AnalyzeFLG == 1)
            CheckPeriod();
            PulseCount++;
        endif
    else
        gfx_CircleFilled (20,20,6,BLACK);
    endif
    PrvState := PrsState;
endif
endfunc

func CheckTouch()
var TouchState,X,Y,exit;
    TouchState := touch_Get(0);
    X := touch_Get(1);
    Y := touch_Get(2);
    if (TouchState == 1)
        if ((X > 275 && X < 420) && (Y > 200 && Y < 266)) exit := 1;
        snd_Volume(127);
        file_PlayWAV("CHIMES.wav");
        return exit;
    else
        return 0;
    endif
endfunc

```

3.2.2 โปรแกรมตรวจวัดความดันซิสโตลิก

```

var i,j,k,P;
    P := 0;
    i := 0;
    repeat
        if (Aarray[i] > P) //ทำการเปรียบเทียบค่าใน Aarray[i] กับค่า P
            P := Aarray[i]; //กำหนดให้ค่า P เท่ากับค่าใน Aarray[i] นั้นๆ
            j := i;
        endif
        i++; //ทำการวนลูปเก็บค่าใน Aarray ไปเรื่อยๆ
    repeat

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

until (Aarray[i]==0 | i == j);
//จนกระทั่ง Aarray[i] มีค่าเท่ากับ 0 โดยที่ค่า i เท่ากับ j
DisplaySYS(Parray[0]); //แสดงค่าที่ได้ลงบนจอแสดงผล

```

3.2.3 โปรแกรมตรวจวัดความดันไดแอสโตลิก

```

DisplayDIA(Darray[PulseCount]); //แสดงค่าที่ได้ลงบนจอแสดงผล

```

3.2.4 โปรแกรมตรวจวัดอัตราการเต้นของหัวใจ

```

if (AnalyzeFLG == 1)
    if (PulseCount==5) Timer := 200; //ทำการหาค่าเฉลี่ยอัตราการเต้นของหัวใจ
        if (Timer > 0)
            Timer--;
        endif
        BeatRate := (6000/(Period/10)); //หาค่าเฉลี่ยอัตราการเต้นของหัวใจ
        DisplayHR(BeatRate); //แสดงผลอัตราการเต้นของหัวใจ
    endif
endif

```

3.2.5 โปรแกรมตรวจวัดค่าเฉลี่ยความดันโลหิต

```

if (PS_Disb <= 50 && AnalyzeFLG == 1)
    AnalyzeFLG := 0;
    Timer := 0;
    SearchSYSDIA(); //โปรแกรมการหาค่าความดันซิสโตลิกและความดันไดแอสโตลิก
    MAP := ((2*Darray[PulseCount])+Parray[0])/3; //การคำนวณหาค่า MAP
    DisplayMAP(MAP); แสดงผลค่า MAP
endif

```

3.2.6 โปรแกรมแสดงค่าบนจอแสดงผล

```

//การกำหนดรูปแบบหน้าจอและตัวอักษรในจอแสดงผล
txt_MoveCursor(1,9);
txt_Set(TEXT_HEIGHT,2);
txt_Set(TEXT_WIDTH,2);
txt_FGcolour(0xDBBE);
print("Blood Pressure Monitor");

txt_MoveCursor(2,1);
txt_Set(TEXT_COLOUR,YELLOW);
txt_Set(TEXT_HEIGHT,1);

```

```
txt_Width(1);
print("SYS :");
```

```
txt_MoveCursor(9,17);
txt_Set(TEXT_COLOUR,YELLOW);
txt_Set(TEXT_HEIGHT,1);
print("mmHg");
txt_Height(1);
txt_Width(1);
```

```
txt_MoveCursor(11,1);
txt_Set(TEXT_COLOUR,YELLOW);
txt_Set(TEXT_HEIGHT,1);
print("DIA :");
txt_Height(1);
txt_Width(1);
txt_MoveCursor(15,17);
txt_Set(TEXT_COLOUR,YELLOW);
txt_Set(TEXT_HEIGHT,1);
print("mmHg");
txt_Height(1);
txt_Width(1);
```

```
txt_MoveCursor(17,12);
txt_Set(TEXT_COLOUR,YELLOW);
txt_Set(TEXT_HEIGHT,1);
print("MAP :");
txt_Height(1);
txt_Width(1);
```

```
txt_MoveCursor(17,1);
txt_Set(TEXT_COLOUR,YELLOW);
txt_Set(TEXT_HEIGHT,1);
print("PULSE :");
txt_Height(1);
txt_Width(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//ทำปุ่มสัมผัสบนจอแสดงผล
I2C_Initialize();
setbaud(BAUD_115200);
touch_DetectRegion(275,200,420,266);
touch_Set(0);
WriteConfigADC(CHANNEL2);
Timer := 0;
AnalyzeFLG := 0;
DIinputPeak := 0;

//กำหนดรูปแบบการแสดงผลค่าความดันซิสโตลิก
txt_MoveCursor(5,5);
txt_Height(5);
txt_Width(4);
txt_Set(TEXT_COLOUR,WHITE);
if (HiBP == 0)
    print ("---");
else
    if (HiBP < 10)
        print (" ",HiBP);
    else
        if (HiBP < 100)
            print (" ",HiBP);
        else
            print(HiBP);
        endif
    endif
endif
txt_Height(1);
txt_Width(1);
endfunc

//กำหนดรูปแบบการแสดงผลตัวเลขความดันลมที่เอาปล่อยเข้ามายังผ้าพันแขน
txt_MoveCursor(9,23);
txt_Height(2);
txt_Set(TEXT_COLOUR,WHITE);
if (PS < 10)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    print (" ",PS);
else
    if (PS < 100)
        print (" ",PS);
    else
        print(PS);
    endif
endif
txt_Height(1);
txt_Width(1);
endfunc

//กำหนดรูปแบบการแสดงผลค่าความดันไดแอสโตลิก
txt_MoveCursor(12,4);
txt_Height(5);
txt_Width(4);
txt_Set(TEXT_COLOUR,0xFC38);
if (LoBP == 0)
    print("—");
else
    if (LoBP < 10)
        print("%",LoBP);
    else
        if (LoBP < 100)
            print (" ",LoBP);
        else
            if (LoBP < 1000) print(LoBP);
        endif
    endif
endif
txt_Height(1);
txt_Width(1);
endfunc

//กำหนดรูปแบบการแสดงผลค่าอัตราการเต้นของหัวใจ
txt_MoveCursor(18,1);
txt_Height(4);
txt_Width(3);
txt_Set(TEXT_COLOUR,0x4FC8);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HR := ABS(HR);
if (HR == 0)
    print (" --");
else
    if (HR < 10)
        print (" ",HR);
    else
        if (HR < 100)
            print (" ",HR);
        else
            if (HR < 1000) print(HR);
        endif
    endif
endif
txt_Height(1);
txt_Width(1);
endfunc
//กำหนดรูปแบบการแสดงผลค่าเฉลี่ยความดัน
txt_MoveCursor(18,11);
txt_Height(4);
txt_Width(3);
txt_Set(TEXT_COLOUR,0x4FC8);
if (MAP == 0)
    print (" --");
else
    if (MAP < 10)
        print (" ",MAP);
    else
        if (MAP < 100)
            print (" ",MAP);
        else
            if (MAP < 1000) print(MAP);
        endif
    endif
endif
txt_Height(1);
txt_Width(1);

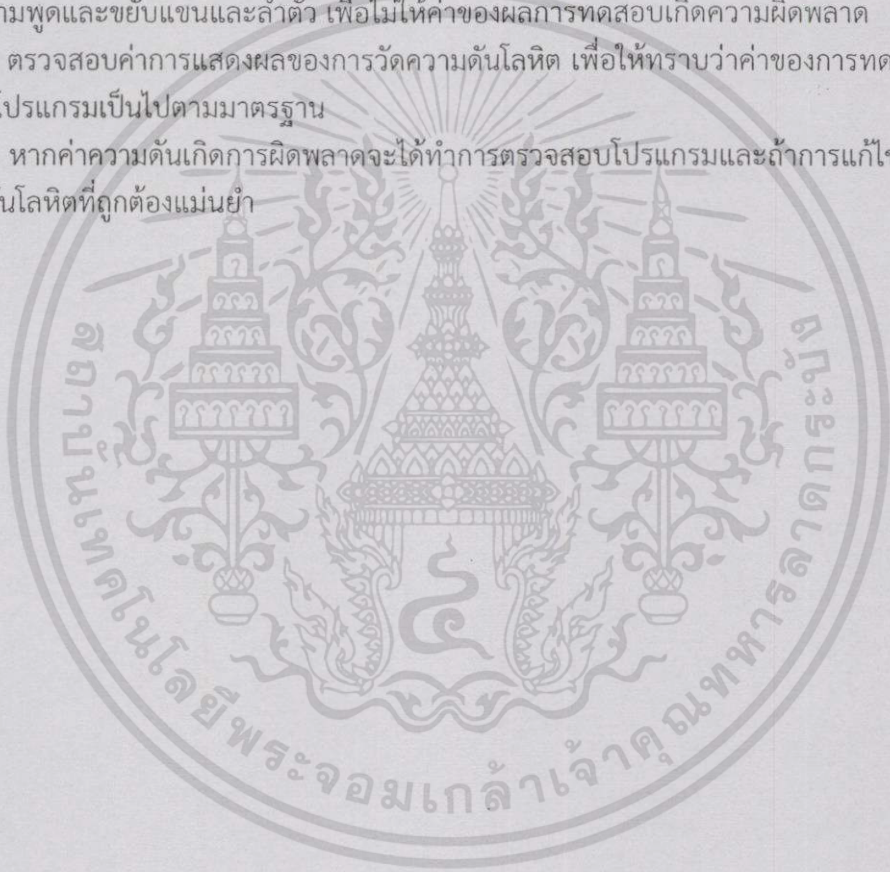
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

endfunc

3.3 วิธีการทดสอบการทำงานของเครื่องวัดความดันโลหิต

1. เริ่มโดยการต่อสาย USB เข้ากับพอร์ตของคอมพิวเตอร์ และต่อไปยังพอร์ตของเครื่องวัดความดันเพื่อจ่ายกระแสไฟฟ้าไปยังอุปกรณ์ ทำการประมวลผลเพื่อตรวจสอบความถูกต้องของโปรแกรม จากนั้นทำการดาวน์โหลดคำสั่งในโปรแกรมไปยังเครื่องวัดความดัน
2. ทำการพันผ้ารัดแขนให้กับผู้ถูกทดสอบ โดยทำการพันผ้าที่แขนข้างใดข้างหนึ่งให้อยู่เหนือข้อพับตรงเส้นเลือดแดงประมาณ 2 นิ้ว ซึ่งผู้ถูกทดสอบต้องนั่งอยู่บนเก้าอี้ ทำตัวตามสบาย วางแขนบนโต๊ะในลักษณะหงายมือและให้อยู่ในระดับเดียวกับหัวใจ ในขณะที่ทำการวัดความดันโลหิต ผู้ถูกทดสอบห้ามพูดและขยับแขนและลำตัว เพื่อไม่ให้ค่าของผลการทดสอบเกิดความผิดพลาด
3. ตรวจสอบค่าการแสดงผลของการวัดความดันโลหิต เพื่อให้ทราบว่าค่าของการทดสอบและการเขียนโปรแกรมเป็นไปตามมาตรฐาน
4. หากค่าความดันเกิดการผิดพลาดจะได้ทำการตรวจสอบโปรแกรมและถ้าการแก้ไขเพื่อให้ได้ค่าความดันโลหิตที่ถูกต้องแม่นยำ



บทที่ 4

ผลการทดลอง

4.1 ผลการทดสอบเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO

เมื่อทำการทดสอบการวัดความดันโลหิตแบบอัตโนมัติโดย PICASO โดยทำการวัดความดันโลหิตของผู้ทดสอบจำนวน 2 คน ซึ่งจะทำการวัดความดันโลหิตคนละ 5 ครั้ง จากนั้นนำค่าที่ได้ทั้ง 5 ครั้งมาหาค่าเฉลี่ย เมื่อนำค่าเฉลี่ยไปเปรียบเทียบกับค่าการวัดในแต่ละครั้ง จะมีค่าที่คาดเคลื่อนไม่เกิน ± 10

ตารางที่ 4.1 ตารางแสดงผลการทดสอบการวัดค่าความดันโลหิตแบบอัตโนมัติโดย PICASO

ผู้ทำการทดสอบลำดับที่	จำนวนครั้งที่ทำการทดสอบ					ค่าเฉลี่ย
	1	2	3	4	5	
1						
SYS (mmHg.)	116	105	115	119	114	114
DIA (mmHg.)	83	84	72	71	70	76
PR (BPM.)	81	87	72	76	77	78.6
MAP	94	91	86	87	84	88.4
ผู้ทำการทดสอบลำดับที่	จำนวนครั้งที่ทำการทดสอบ					ค่าเฉลี่ย
2	1	2	3	4	5	
SYS (mmHg.)	95	102	109	118	111	107
DIA (mmHg.)	75	84	84	60	91	79
PR (BPM.)	70	78	76	68	62	71
MAP	81	90	92	79	77	84
ผู้ทำการทดสอบลำดับที่	จำนวนครั้งที่ทำการทดสอบ					ค่าเฉลี่ย
3	1	2	3	4	5	
SYS (mmHg.)	147	141	154	147	148	147.4
DIA (mmHg.)	63	69	88	73	59	70.4
PR (BPM.)	70	61	64	66	68	65.8
MAP	91	93	110	97	88	95.8

ผู้ทำการทดสอบลำดับที่ 4	จำนวนครั้งที่ทำการทดสอบ					ค่าเฉลี่ย
	1	2	3	4	5	
SYS (mmHg.)	128	140	138	133	124	132.6
DIA (mmHg.)	76	65	77	77	93	77.6
PR (BPM.)	75	78	62	59	90	72.8
MAP	93	90	95	95	103	95.2
ผู้ทำการทดสอบลำดับที่ 5	จำนวนครั้งที่ทำการทดสอบ					ค่าเฉลี่ย
	1	2	3	4	5	
SYS (mmHg.)	113	114	110	112	104	110.6
DIA (mmHg.)	89	92	90	79	91	88.2
PR (BPM.)	70	77	80	75	74	75.2
MAP	97	99	96	90	95	95.4

4.2 ผลการทดสอบและเปรียบเทียบการวัดความดันโลหิตแบบอัตโนมัติหือ ROSSMAX กับเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO

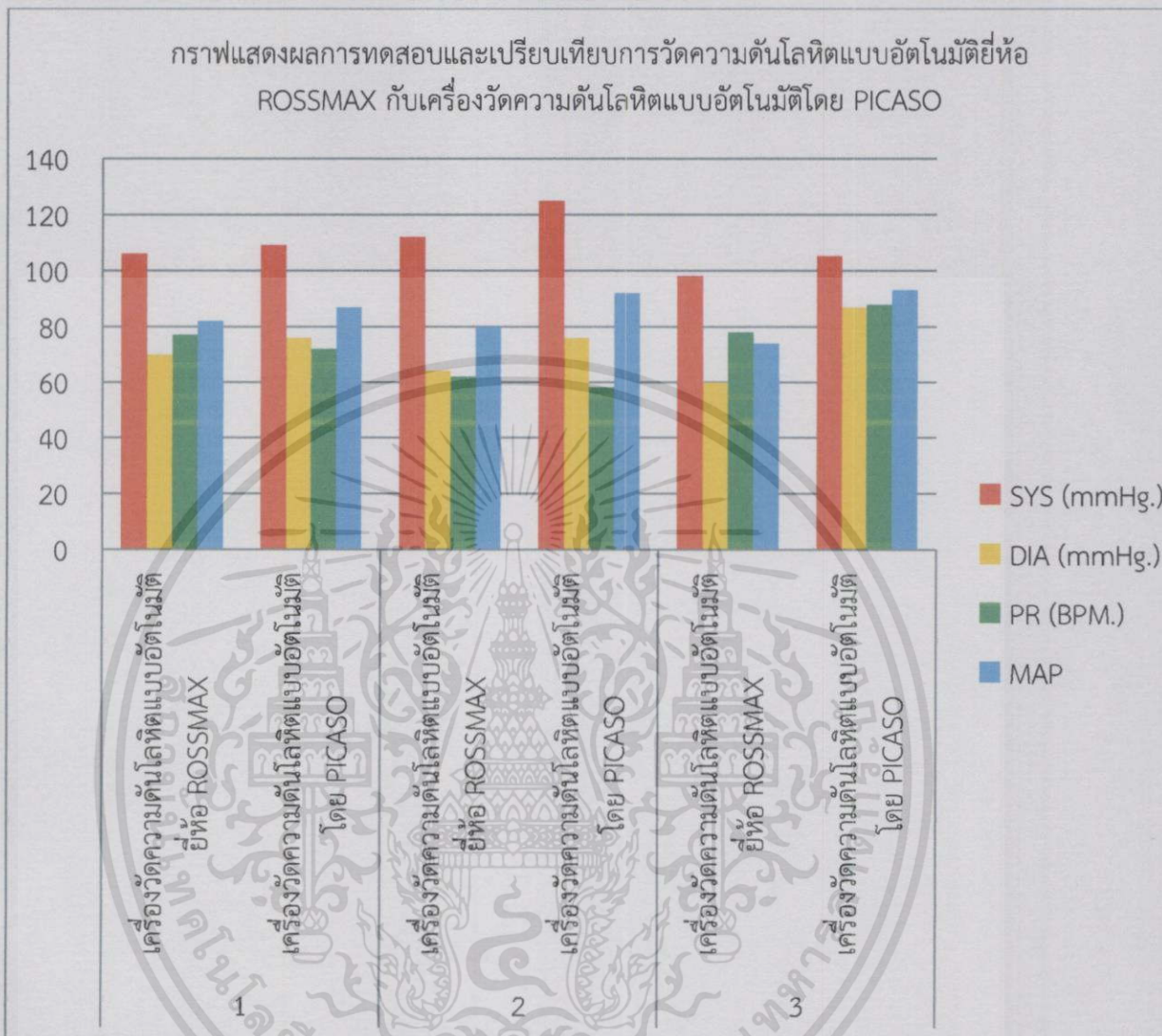
หลังจากทำการทดสอบการทำงานของเครื่องมือวัดความดันโลหิตแบบอัตโนมัติโดย PICASO ผลการทดสอบที่ได้รับนั้นอยู่ในเกณฑ์ที่น่าพอใจ จากนั้นทำการทดสอบและเปรียบเทียบการวัดความดันโลหิตแบบอัตโนมัติหือ ROSSMAX กับเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO โดยทำการวัดความดันโลหิตของผู้ทดสอบจำนวน 3 คน

ตารางที่ 4.2 ตารางแสดงผลการทดสอบและเปรียบเทียบการวัดความดันโลหิตแบบอัตโนมัติยี่ห้อ ROSSMAX กับเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO

คนที่	1		2		3	
ค่าที่วัด ออกมา ได้	เครื่องวัด ความดัน โลหิตแบบ อัตโนมัติ ยี่ห้อ ROSSMAX	เครื่องวัด ความดัน โลหิตแบบ อัตโนมัติ โดย PICASO	เครื่องวัด ความดัน โลหิตแบบ อัตโนมัติ ยี่ห้อ ROSSMAX	เครื่องวัด ความดัน โลหิตแบบ อัตโนมัติ โดย PICASO	เครื่องวัด ความดัน โลหิตแบบ อัตโนมัติ ยี่ห้อ ROSSMAX	เครื่องวัด ความดัน โลหิตแบบ อัตโนมัติ โดย PICASO
SYS (mmHg.)	106	109	112	125	98	105
DIA (mmHg.)	70	76	64	76	60	87
PR (BPM.)	77	72	62	58	78	88
MAP	82	87	80	92	74	93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 กราฟแสดงผลการทดสอบและเปรียบเทียบการวัดความดันโลหิตแบบอัตโนมัติยี่ห้อ ROSSMAX กับเครื่องวัดความดันโลหิตแบบอัตโนมัติโดย PICASO



บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากผลการทดสอบการวัดความดันโลหิตแบบอัตโนมัติโดย PICASO โดยทำการวัดความดันโลหิตของผู้ทดสอบจำนวน 2 คน ซึ่งจะทำการวัดความดันโลหิตคนละ 5 ครั้ง ทำให้ทราบว่า ค่าที่ได้จากเครื่องมือวัดเมื่อนำมาคิดค่าเฉลี่ยและเปรียบเทียบระหว่างกันแล้วเห็นได้ว่าค่าเฉลี่ยแต่ละค่ามีความคลาดเคลื่อนไม่เกิน ± 10 ซึ่งอยู่ในช่วงที่ยอมรับได้ แต่การคลาดเคลื่อนที่เกิดขึ้นอาจมีสาเหตุมาจากการที่เคลื่อนมือวัดได้รับการกระทบกระเทือนจากตัวผู้ทำการวัด และผู้ถูกทดสอบขณะที่มีการวัด ซึ่งจะมีผลกับค่ากราฟ และผลของการวัดความดันที่แสดงออกมา นอกจากนี้อาจมีสาเหตุมาจากการปล่อยลมเนื่องจากหัวจุกที่ใช้ในการปล่อยลมมีขนาดแข็ง รวมทั้งการปล่อยลมของผู้ที่ทำการวัดแต่ละบุคคลจะมีการปล่อยที่ไม่เท่ากัน ทำให้จังหวะในการปล่อยลมมีความคลาดเคลื่อนได้ อย่างไรก็ตาม การทำโครงการครั้งนี้ ได้ทราบถึงลักษณะของความดันโลหิตในหลายรูปแบบ รวมทั้งวิธีการวัดความดันโลหิตแบบออสซิลโลเมตริก ได้รับความรู้ที่เกี่ยวอุปกรณ์วัดความดันเพิ่มขึ้น ได้รับความรู้เกี่ยวกับหลักการทำงานและคุณสมบัติของอุปกรณ์อิเล็กทรอนิกส์ตัวอื่นๆ และได้เรียนรู้การเขียนโปรแกรมภาษากราฟิกเพื่อให้อุปกรณ์เครื่องมือวัดทำงาน

5.2 ปัญหาและอุปสรรค

1. เมื่อทำการวัดความดันหากเครื่องวัดความดันได้รับการกระทบกระเทือนจากตัวผู้ทำการวัดและผู้ถูกทดสอบ ทำให้ค่าความดันโลหิตมีความคลาดเคลื่อน และการแสดงผลของเครื่องวัดความดันมีความช้าส่งผลให้ผู้ถูกวัดความดันมีอาการเจ็บบริเวณต้นแขนเนื่องจากแรงบีบตัวและแรงคลายตัว
2. การเขียนโปรแกรมให้แสดงค่า MAP ช่วงแรกเกิดปัญหาเนื่องจากค่าไม่แสดงบนหน้าจอและค่า DIA ในช่วงแรกมีความคลาดเคลื่อน เนื่องจากการเขียนโปรแกรมที่ผิดทำให้เกิดการอ่านค่าที่ผิดพลาด
3. การปล่อยลมออก เนื่องจากหัวจุกที่ใช้ในการปล่อยลมออกมีลักษณะแข็ง รวมทั้งการปล่อยลมของผู้ที่ทำการวัดแต่ละบุคคลจะมีการปล่อยที่ไม่เท่ากัน ทำให้จังหวะในการปล่อยลมมีผลต่อค่าความดันบน และความดันล่าง
4. ช่วงจังหวะในการปล่อยความดันมิเตอร์จะแสดงผลการลดลง เมื่อปล่อยความดันได้ระยะหนึ่งเข็มมิเตอร์จะมีการตีกลับไปมาเนื่องจากอัตราการปล่อยความดันไม่คงที่จากหัวจุก ซึ่งจะมีผลทำให้ความดันที่วัดไม่มีประสิทธิภาพ
5. การปล่อยความดันที่รวดเร็วเกินไปทำให้ค่าความดันโลหิต และค่าจังหวะการเต้นของหัวใจมีค่าผิดพลาด

5.3 ข้อเสนอแนะ

จากการวัดความดันโลหิตด้วยเครื่องวัดความดันจะให้ค่าความดันโลหิตที่มีค่าใกล้เคียงกับเครื่องวัดความดันโลหิตแบบอัตโนมัติที่มีจำหน่ายทั่วไป แต่เนื่องจากการออกแบบเครื่องวัดความดันนั้นต้องใช้ทักษะในการเขียนโปรแกรม และความรู้เกี่ยวกับอุปกรณ์อิเล็กทรอนิกส์ เพื่อที่จะให้ได้

อุปกรณ์ที่มีความแม่นยำและโครงการนี้สามารถนำไปพัฒนาฟังก์ชันการใช้งาน เช่น สามารถพิมพ์ผล
ค่าความดันโลหิตที่วัดได้และอาจส่งค่าความดันโลหิตผ่านระบบไร้สาย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ณัฐพงษ์ บุขบา, ปารินทร์ ศรีเพชร และ วรณวิณี โตชัยภูมิ, การศึกษาและสร้างเครื่องมือวัดความดันโลหิตโดยใช้ตัววัดความดันร่วมอุปกรณ์ GODELOX – GFX2. ปรินญาณิพนธ์สำหรับปรินญาณิวิศวรรณมบัณฑิต สาขาวิศวรรณมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2555
- [2] พิพัฒน์ ยิ้มฉลวย และ พลท วุฒิพงค์, การศึกษาและออกแบบสร้างเครื่องมือวัดความดันเลือดแบบอัตโนมัติด้วยระบบฝังตัว. ปรินญาณิพนธ์สำหรับปรินญาณิวิศวรรณมบัณฑิต สาขาวิศวรรณมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2555
- [3] ภัทรพิชญ์ เพิ่มสินธุ์, ภูธิป ดลเฉลิมเกียรติ และ มนัส พันธุ์แดง, การศึกษาอุปกรณ์วัดเสียงร่วมกับอุปกรณ์วัดความดันเพื่อวัดความดันเลือดด้วยโปรแกรม LABVIEW. ปรินญาณิพนธ์สำหรับปรินญาณิวิศวรรณมบัณฑิต สาขาวิศวรรณมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2555
- [4] สมศรี ดาวฉาย. ความดันเลือด. ชมรมอุปกรณ์การแพทย์. 5, 14. (พฤษภาคม – ตุลาคม 2537) : 17 – 28
- [5] สิทธิชัย ใจตุม. ออสซิลโลเมตริก ทฤษฎีสำหรับความดันซิลโตลิกและไดแอสโตลิก. ชมรมอุปกรณ์การแพทย์. 5, 14. (พฤษภาคม – ตุลาคม 2537) : 14 – 16
- [6] Bloggang.com, ตารางแสดงระดับความดันโลหิต [Online].
Available : <http://www.bloggang.com/mainblog.php?id=kitpooh22&month=21-11-2012&group=8&gblog=69>
- [7] cache.freescale.com, คู่มืออุปกรณ์ตรวจวัดความดัน MPX 5050 [Online].
Available : http://cache.freescale.com/files/sensors/doc/data_sheet/MPX5050.pdf
- [8] 4DGL Programmers, คู่มืออธิบายรายละเอียดเกี่ยวกับการใช้โปรแกรม 4DGL [Online].
Available : 4DGL Programmers Reference Manual and Language Specifications
- [9] 4dsystems.com, อธิบายโครงสร้างและการทำงานของสายเคเบิล Programming [Online].
Available : http://www.4dsystems.com.au/product/17/114/Accessories/4D_Programming_Cable/
- [10] μ LCD43(GFX), คู่มืออธิบายรายละเอียดโครงสร้างและการทำงานของจอแสดงผล [Online].
Available : μ LCD43(GFX) 4.3" LCD Display Module Data Sheet
- [11] MCP3422/3/4 18-Bit, คู่มือแสดงรายละเอียดโครงสร้างและการทำงานการแปลงสัญญาณจากอนาลอกเป็นสัญญาณดิจิตอลของชิป MCP3424 [Online].
Available : MCP3422/3/4 18-Bit, Multi-Channel $\Delta\Sigma$ Analog-to-Digital Converter with I2C™ Interface and On-Board Reference
- [12] medi.moph.go.th, ความหมายของความดันโลหิตและวิธีการวัดแบบต่างๆ [Online].
Available : <http://medi.moph.go.th/education/Tpum.pdf>

[13] microchip.com, คู่มืออุปกรณ์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล MCP3424 [Online].

Available : <http://ww1.microchip.com/downloads/en/DeviceDoc/22088b.pdf>

[14] PICASO-GFX2, คู่มืออธิบายโครงสร้างและรายละเอียดการทำงานของไมโครคอนโทรลเลอร์ PICSO [Online].

Available : PICASO-GFX2 Internal 4DGL Functions

[15] somkiet.com, ข้อเสนอแนะทั่วไปในการใช้เครื่องวัดความดันโลหิต [Online].

Available : <http://www.somkiet.com/Miscellaneous/BloodPressure.htm>

[16] thaiheartclinic.com, เทคนิคการวัดความดันที่ถูกต้อง [Online].

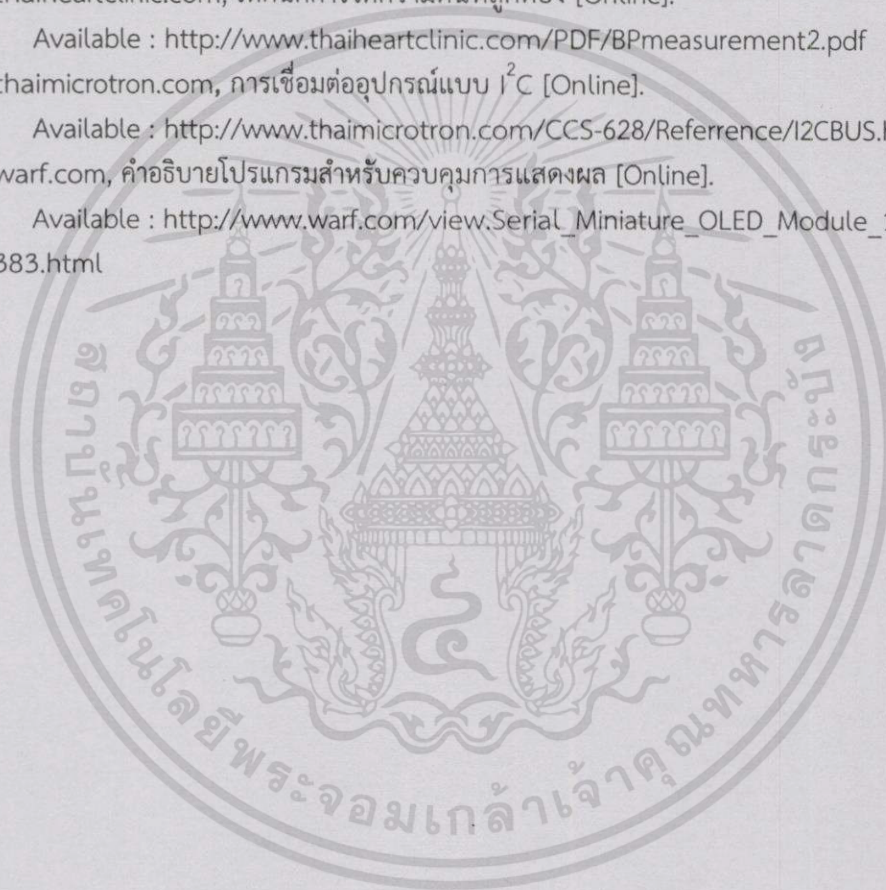
Available : <http://www.thaiheartclinic.com/PDF/BPmeasurement2.pdf>

[17] thaimicrotron.com, การเชื่อมต่ออุปกรณ์แบบ I²C [Online].

Available : <http://www.thaimicrotron.com/CCS-628/Reference/I2CBUS.htm>

[18] warf.com, คำอธิบายโปรแกรมสำหรับควบคุมการแสดงผล [Online].

Available : http://www.warf.com/view.Serial_Minature_OLED_Module_1.5_-383.html



ภาคผนวก ก
การเขียนโปรแกรม

```
#platform "uLCD-43PCT"
```

```
// Program Skeleton 1.0 generated 14/1/2014 21:35:16
```

```
#inherit "4DGL_16bitColours.fnc"
```

```
#inherit "VisualConst.inc"
```

```
#inherit "BPMConst.inc"
```

```
#constant X_ref 212
```

```
#constant Y_ref 54
```

```
#constant SECTORSIZE 512
```

```
#constant CHANNEL1 0x00
```

```
#constant CHANNEL2 0x20
```

```
#constant CHANNEL3 0x40
```

```
#constant CHANNEL4 0x60
```

```
#constant MPC3424 0xDC
```

```
#constant DS1307 0xD0
```

```
#constant WR 1
```

```
var Aarray[1000];
```

```
var Parray[1000];
```

```
var Darray[100];
```

```
var Barray[10];
```

```
var sectorbuf[SECTORSIZE/2];
```

```
var LoBP;
```

```
var HiBP;
```

```
var XPOS;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

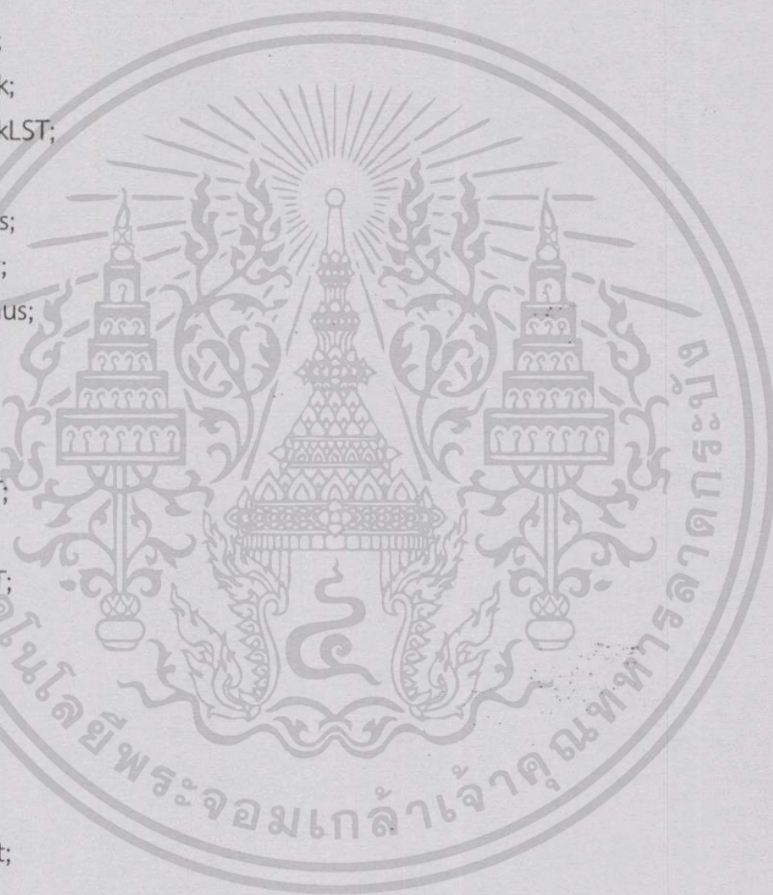
var AnalogValue;
var BP;
var BP_LST;
var DBP_LST;
var PS;
var PS_LST;
var Input;
var InputLST;
var DInput;
var DInputLST;
var DInputPeak;
var DInputPeakLST;

var DPeak_Plus;
var PeakCount;
var DPeak_Minus;
var PeakFLG;

var Integral;
var IntegrallST;
var PS_Dis;
var PS_DisLST;
var PrsState;
var PrvState;
var State;
var Timer;
var Period;
var PulseCount;

var BeatRate;
var AnalyzeFLG;
var DialDispCount;
var seconds;
var address;
var r;
var MAP;
var ADIA;
var ASYS;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

func createframe()
    gfx_Rectangle(1,1,479,271,WHITE);
    gfx_RectangleFilled(1,1,479,2,WHITE);
    gfx_RectangleFilled(1,40,479,42,WHITE);
endfunc

func createschedule()
    gfx_Rectangle(180,42,180,271,WHITE);
    gfx_Rectangle(1,122,180,122,WHITE);
    gfx_Rectangle(1,202,180,202,WHITE);
    gfx_Rectangle(1,271,180,271,WHITE);
    gfx_Rectangle(90,202,90,271,WHITE);
endfunc

func button()
var state;
    // Winbutton1 1.0 generated 15/1/2014 14:03:12
    img_SetWord(hndl, iWinbutton1, IMAGE_FLAGS, (img_GetWord(hndl, iWinbutton1,
IMAGE_FLAGS) | I_STAYONTOP) & ~I_TOUCH_DISABLE); // set to enable touch, only
need to do this once
    img_Show(hndl, iWinbutton1); // show button, only do this once
    img_SetWord(hndl, iWinbutton1, IMAGE_INDEX, state); // where state is 0 for up
and 1 for down
    img_Show(hndl,iWinbutton1);
endfunc

func creategraph()
    gfx_Line(210,50,210,182,WHITE); //Y Axis
    gfx_Line(201,50,210,50,WHITE); //X Axis +1
    gfx_Line(206,83,210,83,WHITE); //X Axis +0.5
    gfx_Line(201,116,210,116,WHITE); //X Axis 0
    gfx_Line(206,149,210,149,WHITE); //X Axis -0.5
    gfx_Line(201,182,210,182,WHITE); //X Axis -1
endfunc

/* .....ANALOG TO DIGITAL.....
*/

```

```

func WriteConfigADC(var Channel)
    I2C_Start();           // Generate Start condition
    I2C_Write(MPC3424);    // send slave address
    I2C_Write(Channel|0x94); // Send register address
    I2C_Stop();
endfunc

```

// read a Channel byte from the required register

```

func ReadADC()
var nd2,rd3;
    I2C_Start();           // Generate Start condition
    I2C_Write(MPC3424+1); // send slave address
    nd2 := I2C_Read();    // read the 2nd byte
    I2C_Ack();
    rd3 := I2C_Read();    // read the 3rd byte
    I2C_Nack();           // finished reading, send Not Ack
    I2C_Stop();           // Send Stop Condition
    AnalogValue := (nd2*256)+rd3;
    return AnalogValue;  // return the register value
endfunc

```

```

func DisplaySYS(var HiBP)

```

```

    txt_MoveCursor(5,5);
    txt_Height(5);
    txt_Width(4);
    txt_Set(TEXT_COLOUR,WHITE);
    if (HiBP == 0)
        print ("---");
    else
        if (HiBP < 10)
            print (" ",HiBP);
        else
            if (HiBP < 100)
                print (" ",HiBP);
            else
                print(HiBP);
            endif

```

```

    endif
endif
txt_Height(1);
txt_Width(1);
endfunc

```

```

func DisplayPS(var PS)
    txt_MoveCursor(9,23);
    txt_Height(2);
    txt_Set(TEXT_COLOUR,WHITE);
    if (PS < 10)
        print (" ",PS);
    else
        if (PS < 100)
            print (" ",PS);
        else
            print(PS);
        endif
    endif
    txt_Height(1);
    txt_Width(1);
endfunc

```

```

func DisplayDIA(var LoBP)
    txt_MoveCursor(12,4);
    txt_Height(5);
    txt_Width(4);
    txt_Set(TEXT_COLOUR,0xFC38);
    if (LoBP == 0)
        print(" --");
    else
        if (LoBP < 10)
            print (" ",LoBP);
        else
            if (LoBP < 100)
                print (" ",LoBP);
            else
                if (LoBP < 1000) print(LoBP);
            endif
        endif
    endif
endfunc

```

```

        endif
    endif
endif
txt_Height(1);
txt_Width(1);
endfunc

```

```

func DisplayHR(var HR)
    txt_MoveCursor(18,1);
    txt_Height(4);
    txt_Width(3);
    txt_Set(TEXT_COLOUR,0x4FC8);
    HR := ABS(HR);
    if (HR == 0)
        print (" --");
    else
        if (HR < 10)
            print (" ",HR);
        else
            if (HR < 100)
                print (" ",HR);
            else
                if (HR < 1000) print(HR);
            endif
        endif
    endif
    txt_Height(1);
    txt_Width(1);
endfunc

```

```

func DisplayMAP(var MAP)
    txt_MoveCursor(18,11);
    txt_Height(4);
    txt_Width(3);
    txt_Set(TEXT_COLOUR,0x4FC8);
    if (MAP == 0)
        print(" --");
    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (MAP < 10)
    print (" ",MAP);
else
    if (MAP < 100)
        print (" ",MAP);
    else
        if (MAP < 1000) print(MAP);
    endif
endif
endif
txt_Height(1);
txt_Width(1);
endfunc

func GetPressure()
    ReadADC();
    PS := PS_LST + (((AnalogValue - PS_LST)*10)/65); //37 for beat rate Measurement
    filtered a= 0.3
    PS_LST := PS;
    Input := PS;
endfunc

func ClearArray()
var i;
for (i:=0; i<100; i++)
    Aarray[i] := 0;
    Parray[i] := 0;
    Darray[PulseCount] := 0;
next
for (i:=0; i<10; i++)
    Barray[i] := 0;
next
endfunc

func Differential()
    DInput := Input - InputLST; //Delta Input
    PS_Displacement := ((PS/39)-8);
    if (PS_Displacement >= 160) PS_Displacement := 160;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (PS_Disp <= 0) PS_Disp := 0;
InputLST := Input;
endfunc

```

```

func PlotGraph (var BP var DBP var INT)

```

```

var Pulse;

```

```

BP := (BP*66)/100;

```

```

if (BP >= 66) BP := 66;

```

```

BP := 66-BP;

```

```

if (ABS(DBP) >= 1) DBP := DBP*(10);

```

```

if (DBP >= 60) DBP := 60;

```

```

if (DBP <= -60) DBP := -60;

```

```

DBP := 60-DBP;

```

```

if (INT >= 60) INT := 60;

```

```

if (INT <= -60) INT := -60;

```

```

INT := 60-INT;

```

```

XPOS := XPOS + 1;

```

```

if (XPOS >= 265) XPOS := 0;

```

```

gfx_RectangleFilled(X_ref+XPOS,Y_ref,X_ref+XPOS+3,Y_ref+130,BLACK);

```

```

gfx_Line(X_ref+XPOS-1,Y_ref+DBP_LST,X_ref+XPOS,Y_ref+DBP,WHITE);

```

```

BP_LST := BP;

```

```

DBP_LST := DBP;

```

```

IntegralLST := INT;

```

```

endfunc

```

```

func CheckPeriod()

```

```

if (AnalyzeFLG==1)

```

```

Period := 2000 - sys_GetTimer(TIMERO);

```

```

sys_SetTimer(TIMERO,2000);

```

```

Barray[0]:=Barray[1];

```

```

Barray[1]:=Barray[2];

```

```

Barray[2]:=Barray[3];

```

```

Barray[3]:=Barray[4];
Barray[4]:=Period;

Period := (Barray[0]+Barray[1]+Barray[2]+Barray[3]+Barray[4])/5;
endif
endfunc

```

```

func CheckPeak()
  if (PulseCount >= 1 && PS_Displ >= 50 && AnalyzeFLG == 1 && PS_Displ <
PS_DisplST)
    if (DInput >= DInputLST)
      DPeak_Plus := DInput;
    else
      PeakFLG := 1;
      if (DInput <= DInputLST)
        DPeak_Minus := DInput;
      endif
    endif
    if (PeakFLG==1)
      DInputPeak := ABS(DPeak_Plus) - ABS(DPeak_Minus);
      Aarray[PeakCount] := DInputPeak;
      Parray[PeakCount] := PS_Displ;
      Darray[PulseCount]:= PS_Displ;
      PeakCount++;
      PulseCount++;
      to(COM0);print(DInputPeak,",",PS_Displ,"\n");
      PeakFLG := 0;
    endif
  endif
  DInputLST := DInput;
  PS_DisplST := PS_Displ;
endfunc

```

```

func CheckPulse()
  if (DInput != 0)
    if (DInput > 1) PrsState := 1;
    if (DInput < 1) PrsState := 0;
    State := (PrvState ^ PrsState) & (! PrvState);

```

```

if (State == 1)
    gfx_CircleFilled (20,20,6,RED);
    if (AnalyzeFLG == 1)
        CheckPeriod();
        PulseCount++;
    endif
else
    gfx_CircleFilled (20,20,6,BLACK);
endif
PrvState := PrsState;
endif
endfunc

func CheckTouch()
var TouchState,X,Y,exit;
    TouchState := touch_Get(0);
    X := touch_Get(1);
    Y := touch_Get(2);
    if (TouchState == 1)
        if ((X > 275 && X < 420) && (Y > 200 && Y < 266)) exit := 1;

        snd_Volume(127);
        file_PlayWAV("CHIMES.wav");

        return exit;
    else
        return 0;
    endif
endfunc

func SearchSYSDIA()
// Search SYS
var i,j,k,P;
    P := 0;
    i := 0;
    repeat
        if (Aarray[i] > P)
            P := Aarray[i];

```

```

        j := i;
    endif
    i++;
until (Aarray[i]==0 | i == j);
DisplaySYS(Parray[0]);

// Search DIA
DisplayDIA(Darray[PulseCount]);

snd_Volume(127);
file_PlayWAV("CHIMES.wav");

to(COM0);print (Parray[0]," ",Darray[PulseCount]," ",MAP," ",BeatRate,"");
endfunc

func main()
    var hstrings ; // Handle to access uSD strings, uncomment if required
    var hFontx ; // Handle to access uSD fonts, uncomment if required and change n
to font number
// Uncomment the following if uSD images, fonts or strings used.

//putstr("Mounting...\n");
if (!(disk:=file_Mount()))
    while(!(disk :=file_Mount()))
        putstr("Drive not mounted...");
        pause(200);
        gfx_Cls();
        pause(200);
    wend
endif
gfx_TransparentColour(0x0020);
gfx_Transparency(ON);

// hFontn := file_LoadImageControl("BPM.dan", "BPM.gcn", 1); // Open handle to
access uSD fonts, uncomment if required and change n to font number dropping a
and c if > 9

```

```

// hstrings := file_Open("BPM.txf", 'r') ; // Open handle to access uSD strings,
uncomment if required
    hndl := file_LoadImageControl("BPM.dat", "BPM.gci", 1);

    gfx_Set(SCREEN_MODE, LANDSCAPE);

//Total text
    txt_MoveCursor(1,9);
    txt_Set(TEXT_HEIGHT,2);
    txt_Set(TEXT_WIDTH,2);
    txt_FGcolour(0xDBBE);
    print("Blood Pressure Monitor");

    txt_MoveCursor(2,1);
    txt_Set(TEXT_COLOUR,YELLOW);
    txt_Set(TEXT_HEIGHT,1);
    txt_Width(1);
    print("SYS:");

    txt_MoveCursor(9,17);
    txt_Set(TEXT_COLOUR,YELLOW);
    txt_Set(TEXT_HEIGHT,1);
    print("mmHg");
    txt_Height(1);
    txt_Width(1);

    txt_MoveCursor(11,1);
    txt_Set(TEXT_COLOUR,YELLOW);
    txt_Set(TEXT_HEIGHT,1);
    print("DIA:");
    txt_Height(1);
    txt_Width(1);

    txt_MoveCursor(15,17);
    txt_Set(TEXT_COLOUR,YELLOW);
    txt_Set(TEXT_HEIGHT,1);
    print("mmHg");
    txt_Height(1);

```

```

txt_Width(1);

txt_MoveCursor(17,12);
txt_Set(TEXT_COLOUR,YELLOW);
txt_Set(TEXT_HEIGHT,1);
print("MAP :");
txt_Height(1);
txt_Width(1);

txt_MoveCursor(17,1);
txt_Set(TEXT_COLOUR,YELLOW);
txt_Set(TEXT_HEIGHT,1);
print("PULSE :");
txt_Height(1);
txt_Width(1);

setbaud(BAUD_115200);
touch_DetectRegion(275,200,420,266);
touch_Set(0);
WriteConfigADC(CHANNEL2);
Timer := 0;
AnalyzeFLG := 0;
DInputPeak := 0;

repeat
createframe();
createschedule();
button();
creategraph();
GetPressure();
Differential();
CheckPulse();
CheckPeak();           //Search Peak of Pulse Amplitude
DisplayPS(PS_Disg);
PlotGraph (PS_Disg,DInput,Integral);
    if (CheckTouch() == 1)
        PulseCount := 0;
        BeatRate := 0;

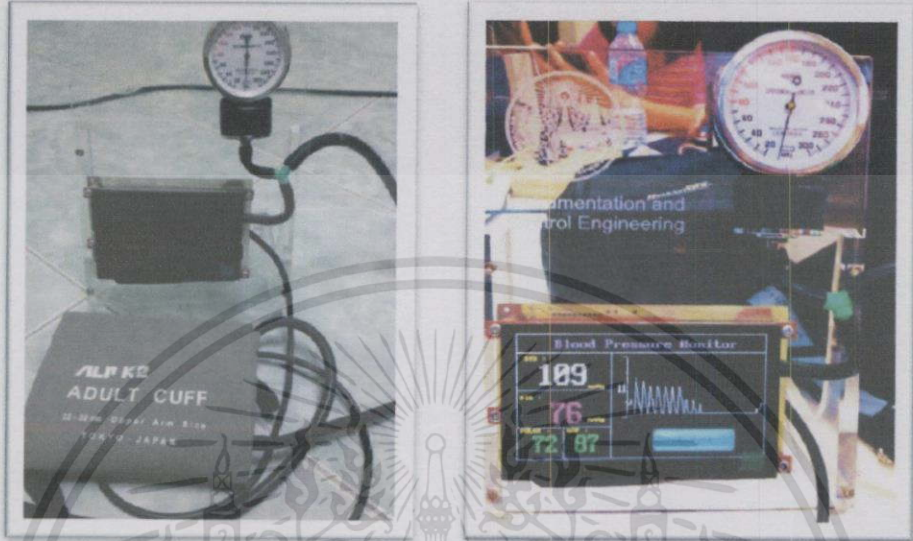
```

```

PeakCount := 0;
DPeak_Minus := 0;
DPeak_Plus := 0;
DInputLST := 0;
AnalyzeFLG := 1;
DisplayDIA(0);
DisplaySYS(0);
DisplayHR(0);
DisplayMAP(0);
ClearArray();
endif
if (AnalyzeFLG == 1)
  if (PulseCount==5) Timer := 200;
  if (Timer > 0)
    Timer--;
  endif
  if (Timer == 1)
    BeatRate := (6000/(Period/10));
    DisplayHR(BeatRate);
  endif
endif
if (PS_Displacement <= 50 && AnalyzeFLG == 1)
  AnalyzeFLG := 0;
  Timer := 0;
  SearchSYSDIA();
  MAP := ((2*Darray[PulseCount]+Parray[0])/3);
  DisplayMAP(MAP);
endif
forever
endfunc

```

ภาคผนวก ข
รูปภาพอุปกรณ์ที่ใช้ในปริญญานิพนธ์เล่มนี้

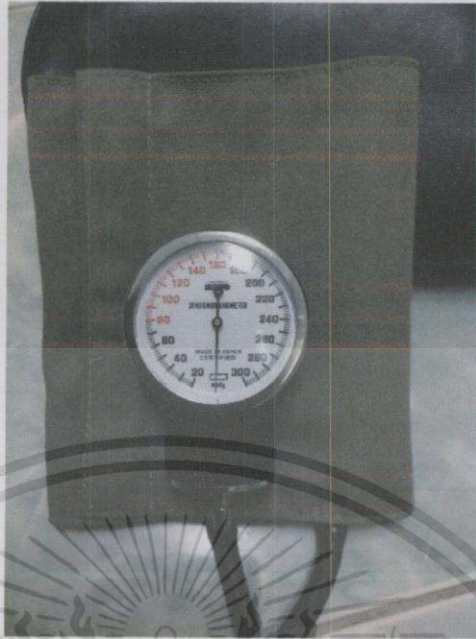


ข.1 แสดงอุปกรณ์ทั้งหมดของเครื่องวัดความดันโลหิตแบบอัตโนมัติ



ข.2 แสดงขนาดของผ้าพันแขน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ข.3 แสดงเกจวัดความดันขนาด 300 มิลลิเมตรปรอท



ข.4 แสดงฮาร์ดแวร์รวมของวงจรการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้