การศึกษาและออกแบบสร้างเครื่องมือวัดความดันเลือดแบบอัตโนมัติด้วยระบบฝังตัว

A STUDY AND DESIGN OF AUTOMATIC BP INSTRUMENT BY EMBEDDED SYSTEM

พิพัฒน์ ยิ้มฉลวย
พลท วุฒิพงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

การศึกษาและออกแบบสร้างเครื่องมือวัดความดันเลือดแบบอัตโนมัติด้วยระบบฝังตัว

A STUDY AND DESIGN OF AUTOMATIC BP INSTRUMENT BY EMBEDDED SYSTEM

พิพัฒน์ ยิ้มฉลวย
พลท วุฒิพงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมบัณฑิต
สาขาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A STUDY AND DESIGN OF AUTOMATIC BP INSTRUMENT BY EMBEDDED SYSTEM

PIPAT YIMCHALUAY
PALOT WUTIPONG

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 INSTITUE OF TECHNOLOGY LARDKRABANG
ACADEMIC YEAR 2012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2555
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ การศึกษาและออกแบบสร้างเครื่องมือวัดความดันเลือดแบบอัตโนมัติด้วยระบบฝังตัว
A STUDY AND DESIGN OF AUTOMATIC BP INSTRUMENT BY EMBEDDED SYSTEM

นักศึกษาผู้จัดทำ นายพิพัฒน์ ยิ้มฉลวย **รหัสนักศึกษา** 52010837
นายพลท วุฒิพงศ์ **รหัสนักศึกษา** 52010797

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2555

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์วิริยะ กองรัตน์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การศึกษาและออกแบบสร้างเครื่องมือวัดความดันเลือดแบบอัตโนมัติด้วยระบบฝังตัว A STUDY AND DESIGN OF AUTOMATIC BP INSTRUMENT BY EMBEDDED SYSTEM			
นักศึกษาผู้จัดทำ	นายพิพัฒน์	ยิ้มฉลวย	รหัสนักศึกษา	52010837
	นายพลท	วุฒิพงศ์	รหัสนักศึกษา	52010797
ปีการศึกษา	2555			

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและออกแบบเครื่องมือวัดความดันโลหิตแบบอัตโนมัติด้วยระบบฝังตัวโดยเริ่มทำการศึกษาพฤติกรรมโดยธรรมชาติของความดันโลหิตมนุษย์และใช้โปรแกรม LabVIEW ในการวิเคราะห์สัญญาณที่ได้จากการวัดความดันโลหิตเพื่อที่จะใช้เป็นแนวทางในการเขียนโปรแกรมประมวลผลสัญญาณในไมโครวีจีเอทู(GFX) ที่เขียนด้วยภาษากราฟฟิค 4D เป็นตัวควบคุมการทำงานซึ่งเชื่อมต่อกับจอมอนิเตอร์เพื่อแสดงผลผ่านพอร์ต VGA โดยใช้ I2C อินเตอร์เฟสเป็นตัวเชื่อมต่อการทำงานระหว่างไมโครวีจีเอทู(GFX)ซึ่งเป็นมาสเตอร์กับอุปกรณ์ตัวอื่นๆซึ่งเป็นสเลฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	A STUDY AND DESIGN OF AUTOMATIC BP INSTRUMENT BY EMBEDDED SYSTEM
Authors	Mr. Pipat Yimchaluay
	Mr. Palot Wutipong
Thesis Advisor	Assoc.Prof. Viriya Kongratana
Year	2012

ABSTRACT

This thesis aim to study and design of automatic blood pressure by embedd system.Begin with study for human blood pressure behavior by using LabVIEW program and analyzing behavior of blood pressured while measuring. This analysis guides the way to programming in µVGA-II(GFX) which use 4D Graphic language and connect to monitor by VGA port to show the result. I2C BUS is use to be a connector between µVGA-II that act as master and other devices that act as slave.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีด้วยความเมตตาช่วยเหลืออย่างยิ่งของท่านรองศาสตราจารย์วิริยะ กองรัตน์ที่อนุมัติเห็นชอบ คอยให้คำปรึกษา และสนับสนุนในการจัดทำโครงการชิ้นนี้ ท่านอาจารย์สุธรรม สัทธรรมสกุลที่ให้คำปรึกษาและสนับสนุนการทำรูปเล่มปริญญานิพนธ์ รวมทั้งขอบพระคุณท่านอาจารย์คณะวิศวกรรมศาสตร์สาขาวิชาการวัดคุมทุกท่านที่ให้คำแนะนำและคอยให้กำลังใจคณะผู้จัดทำ และความดีอันเกิดจากการจัดทำโครงการในครั้งนี้ คณะผู้จัดทำขอมอบให้แด่บิดา มารดา ครูอาจารย์ และผู้มีพระคุณทุกท่าน คณะผู้จัดทำมีความซาบซึ้งในความกรุณาอันดียิ่งจากทุกท่านที่ได้กล่าวนามมา และขอกราบขอบพระคุณมา ณ โอกาสนี้

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VII
สารบัญภาพ VIII

บทที่ 1 บทนำ 1
1.1 ความสำคัญของปริญญานิพนธ์ 1
1.2 วัตถุประสงค์ของปริญญานิพนธ์ 1
1.3 ขอบเขตของปริญญานิพนธ์ 1
1.4 ขั้นตอนการศึกษา 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎี 3
2.1 การวัดความดันโลหิต 3
2.1.1 ข้อเปรียบเทียบการวัดความดันโลหิตแบบอินเวซีฟและนันอินเวซีฟ 3
2.1.2 การวัดความดันโลหิตแบบนันอินเวซีฟ 3
2.1.2.1 การวัดความดันโลหิตด้วยวิธีฟังเสียง 3
2.1.2.2 การวัดความดันโลหิตด้วยวิธีตรวจจับแรงดัน 4
2.1.2.3 ข้อเปรียบเทียบและการประยุกต์ใช้ร่วมกันระหว่างการวัดความดันโลหิตด้วยการใช้หูฟังและด้วยการตรวจจับแรงดัน 4
2.2 คุณลักษณะของอุปกรณ์แต่ละตัว 5
2.2.1 อุปกรณ์จัดเก็บข้อมูล (DAQ) 5
2.2.2 แล็ปวิว (LabVIEW) 6
2.2.3 เซนเซอร์ความดัน MPX5050 7
2.2.3.1 คุณลักษณะเฉพาะของอุปกรณ์ 7
2.2.3.2 หลักการของปรากฏการณ์เพียโซรีซิสทีฟ 8
2.2.4 ไมโครวีจีเอทู (GFX) 8
2.2.3.1 PICASO-GFX2 9
2.2.3.2 ขาการเชื่อมต่อกับฮาร์ดแวร์ 9
2.2.4.3 ส่วนของซอฟท์แวร์ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.2.5 อุปกรณ์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล MCP3424.....12
2.2.5.1 คุณสมบัติของอุปกรณ์.....12
2.2.5.2 รีจิสเตอร์ควบคุมของ MCP3424.....13
2.2.6 อุปกรณ์แสดงเวลาและวันที่ DS1307 (Real-Time Clock).....13
2.2.6.1 คุณสมบัติของอุปกรณ์.....14
2.2.6.2 รีจิสเตอร์เก็บเวลา.....14
2.2.6.3 การรับส่งข้อมูล.....15
2.2.7 ปั๊มลมไฟฟ้ากระแสตรง.....16
2.2.8 โซลินอยด์วาล์ว.....16
2.2.9 วาล์วควบคุมอัตราการไหล.....17
2.2.10 รีเลย์.....17
2.3 I2C BUS.....18
2.3.1 การเชื่อมต่ออุปกรณ์แบบ I2C BUS.....18
2.3.2 การเขียน-อ่านข้อมูลบน I2C BUS.....19
2.3.3 การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I2C BUS.....19

บทที่ 3 วิธีการดำเนินงาน.....20
3.1 การเชื่อมต่ออุปกรณ์แต่ละตัวเข้าด้วยกัน.....20
3.2 การเขียนโปรแกรม.....21
3.2.1 ส่วนของโปรแกรมหลัก.....22
3.2.2 ส่วนของโปรแกรมที่ใช้ในการตรวจจับความดันตัวบน.....23
3.2.3 ส่วนของโปรแกรมที่ใช้ในการตรวจจับความดันตัวล่าง.....23
3.3 การทดสอบการทำงานของเครื่องมือวัด.....24

บทที่ 4 ผลการทดลอง.....25
4.1 การทดลองวัดความดันโลหิตเทียบกับเครื่องวัดความดันโลหิตอัตโนมัติ.....25

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....26
5.1 สรุปผล.....26
5.2 ปัญหาและอุปสรรค์.....26
5.2 ข้อเสนอแนะ.....26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บรรณานุกรม.....27

ภาคผนวก.....28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตารางแสดงคุณลักษณะในการทำงานของเซนเซอร์ความดัน MPX5050.....7
2.2 ตารางแสดงรีจิสเตอร์ควบคุม MCP3424.....13
2.3 ตารางแสดงรีจิสเตอร์เก็บเวลา.....15
4.1 ตารางแสดงผลการวัดจากโครงงาน.....25
4.2 ตารางแสดงผลการวัดจากเครื่องวัดความดันอัตโนมัติ.....25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	แสดงอุปกรณ์วัดความดันโลหิตแบบฟังเสียง	4
2.2	แสดงกราฟการกระเพื่อมของความดันเมื่อปล่อยลมออกอย่างช้าๆ	4
2.3	แสดงอุปกรณ์การจัดเก็บข้อมูล	5
2.4	แสดงหน้าตาของโปรแกรมแล็ปวิว	6
2.5	แสดงเซนเซอร์ความดัน MPX5050	7
2.6	แสดงภาพของอุปกรณ์ไมโครวีจีเอทู(GFX)	8
2.7	แสดงภาพของอุปกรณ์ PICASO-GFX2	9
2.8	แสดงขาการเชื่อมต่อของไมโครวีจีเอทู	9
2.9	แผนภาพแสดงการรับส่งข้อมูลที่สัมพันธ์กับอัตราบอด	10
2.10	แสดงการส่งข้อมูลแบบอนุกรมไบต์เดี่ยว	10
2.11	แสดงภาพตัวอย่างการเขียนโปรแกรมภาษา 4DGL	11
2.12	แสดงภาพของอุปกรณ์ MCP3424	12
2.13	แสดงอุปกรณ์แสดงเวลาและวันที่ DS1307	14
2.14	ปั๊มลมไฟฟ้ากระแสตรง	16
2.15	โซลินอยด์วาล์ว	16
2.16	แสดงโครงสร้างของโซลินอยด์วาล์ว	17
2.17	วาล์วควบคุมอัตราการไหล	17
2.18	รีเลย์	18
2.19	แสดงขั้นตอนการส่งและสิ้นสุดการใช้บัส	19
2.20	แสดงแผนภาพเวลาที่กำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I2C BUS	19
3.1	แสดงแผนภาพการเชื่อมต่อของอุปกรณ์แต่ละตัว	20
3.2	แสดงการต่อท่อลมเข้ากับอุปกรณ์ต่างๆ	20
3.3	ผังงานแสดงการทำงานเพื่อใช้เขียนโปรแกรม	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในยุคปัจจุบันแม้ว่าเทคโนโลยีสมัยใหม่ในด้านต่าง ๆ ทำให้มนุษย์มีความสะดวกสบายมากยิ่งขึ้น แต่ความสะดวกสบายมากเกินไปก็อาจจะทำให้เกิดผลเสียตามมาได้เช่นเดียวกัน ยกตัวอย่างเช่นทางด้านร่างกาย ความสะดวกสบายที่มากเกินไปก็จะทำให้ร่างกายไม่ได้ขยับเขยื้อนมากพออย่างที่ควรจะเป็น จึงเป็นสาเหตุให้เกิดโรคภัยไข้เจ็บรุมเร้าเบียดเบียน โครงการชิ้นนี้ได้เน้นแนวคิดเกี่ยวกับสุขภาพของมนุษย์เป็นสำคัญ ถ้ามนุษย์มีสุขภาพดีก็จะทำให้ด้านอื่นๆดีขึ้นไปด้วย ซึ่งค่าความดันโลหิตนั้นสามารถบ่งบอกถึงสภาวะการทำงานของร่างกายได้หลายส่วน เช่น ในส่วนของสมอง ถ้าความดันโลหิตสูงก็จะสามารถทำให้หลอดเลือดในสมองตีบตันหรือแตก ซึ่งเป็นสาเหตุทำให้เป็นอัมพฤกษ์อัมพาตได้ ในส่วนของดวงตา ถ้าความดันโลหิตสูงก็จะทำให้เกิดภาวะตามัวหรืออาจร้ายแรงจนถึงขั้นตาบอดได้ แต่ถ้ามีความดันโลหิตต่ำเกินไปก็จะส่งผลทำให้ประสิทธิภาพการทำงานของร่างกายลดลงเช่นเดียวกัน เป็นต้น ซึ่งในผู้ที่มีความดันโลหิตสูงหรือความดันโลหิตต่ำบางรายอาจจะไม่เกิดอาการของโรค โรคความดันโลหิตสูงหรือความดันโลหิตต่ำนี้จึงได้สมญานามว่าเป็น “ฆาตกรเงียบ” ถ้าเกิดผู้ป่วยทราบถึงความดันโลหิตของตนเองว่าสูงหรือต่ำเกินไปแล้ว ก็จะทำให้สามารถรักษาได้ทันท่วงที

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาความดันเลือดและวิธีการวัดความดันโดย ประยุกต์ใช้อุปกรณ์อิเล็กทรอนิกส์และระบบฝังตัว
2. ศึกษาตัวตรวจวัดที่เหมาะสมกับลูกคลื่นความดันเลือด เพื่อที่จะใช้ในการตรวจวัดที่มีความแม่นยำและถูกต้องมากที่สุด
3. ศึกษาการใช้งานเครื่องมือวัดความดันในการตรวจวัดความดัน และตรวจจับการกระเพื่อมของความดันเมื่อทำการวัดความดันตัวบนและความดันตัวล่างทำให้ทราบได้ว่าความดันโลหิตมีค่าเท่าไร
4. ศึกษาและเขียนโปรแกรมในการตรวจจับความดันกระเพื่อม และตรวจจับความดันเพื่อแสดงผลออกในหน้าจอ LCD หรือ 7 Segment

1.3 ขอบเขตของปริญญานิพนธ์

1. ออกแบบสร้างฟังก์ชันสำหรับระบบฝังตัว เพื่อวิเคราะห์สัญญาณ
2. แสดงผลค่าความดันตัวบน ความดันตัวล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

1. ศึกษาการวัดความดันเลือด
2. ศึกษาการใช้งาน Embedded Board (ADC, DIO, I2C)
3. ศึกษาเซนเซอร์ความดัน
4. ศึกษาและออกแบบสร้างวงจรขายสัญญาณ ตัวตรวจจับความสั่นสะเทือนและความดัน และออกแบบสร้างอุปกรณ์ปรับสภาวะสัญญาณ
5. ศึกษาลักษณะสัญญาณที่ได้จากตัวตรวจจับความดันเลือด (LabVIEW)

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. นำความรู้ที่ได้จากการศึกษาและจัดทำโครงการไปประยุกต์ใช้ในการทำงานจริง
2. สามารถนำเครื่องมือที่ออกแบบไปใช้งานได้จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 การวัดความดันโลหิต

วิธีวัดความดันโลหิตนั้นมีหลักๆ อยู่ 2 วิธี วิธีแรกคือ การวัดความดันโลหิตแบบอินเวซีฟ (Invasive) ซึ่งเป็นวิธีที่สอดเข็มแคนนูล่าเข้าไปในเส้นเลือดแดงซึ่งต่อโดยตรงกับเครื่องวัดความดันโลหิตแบบอินเวซีฟ และวิธีที่สองคือ การวัดความดันโลหิตแบบนันอินเวซีฟ(Non-invasive) ซึ่งจะแบ่งย่อยออกเป็นอีก 5 วิธีคือ การคลำชีพจร การฟังเสียง การตรวจจับแรงดัน การอุดกั้นการไหลของเลือดชั่วคราว และการดูสีผิวเมื่อมีเม็ดเลือด

2.1.1 ข้อเปรียบเทียบวิธีวัดความดันโลหิตแบบอินเวซีฟและนันอินเวซีฟ

ปัจจุบันการวัดความดันโลหิตแบบอินเวซีฟไม่ค่อยได้รับความนิยมแล้วอันเนื่องมาจาก

2.1.1.1 ความปลอดภัย

วิธีการวัดความดันโลหิตแบบอินเวซีฟเป็นวิธีที่ต้องเจาะเข็มเข้าไปในหลอดเลือดแดง ซึ่งอาจจะทำให้เกิดอันตรายกับผู้ถูกวัดได้ถ้าไม่มีผู้ดูแลอย่างใกล้ชิด แต่วิธีวัดความดันโลหิตแบบนันอินเวซีฟไม่ทำให้ผู้ถูกวัดเกิดอันตรายแต่อย่างใด แม้ไม่มีผู้ดูแล

2.1.1.2 ระยะเวลา

วิธีการวัดความดันโลหิตแบบอินเวซีฟต้องใช้อุปกรณ์และจัดเตรียมเครื่องมือเยอะกว่าวิธีการวัดแบบนันอินเวซีฟ ซึ่งวิธีวัดแบบอินเวซีฟนั้นยังสามารถกระทำได้ที่บ้านหรือที่พักอาศัย โดยไม่จำเป็นต้องไปสถานพยาบาลหรือโรงพยาบาล ทำให้ประหยัดเวลามากขึ้นไปอีก

2.1.1.3 ความแม่นยำ

แม้ว่าวิธีการวัดความดันแบบอินเวซีฟจะมีความแม่นยำมากกว่าวิธีการวัดแบบนันอินเวซีฟ แต่ความปลอดภัยต้องมาเป็นอันดับหนึ่งสำหรับผู้ถูกวัด

2.1.2 การวัดความดันโลหิตแบบนันอินเวซีฟ

วิธีการวัดความดันโลหิตแบบนันอินเวซีฟแบ่งย่อยออกเป็น 5 วิธีคือ การคลำชีพจร การฟังเสียง การตรวจจับแรงดัน การอุดกั้นการไหลของเลือดชั่วคราว และการดูสีผิวเมื่อมีเม็ดเลือด แต่จะยกการวัดความดันโลหิตมาอธิบายด้วยกัน 2 วิธีคือ การฟังเสียง และการตรวจจับแรงดัน

2.1.2.1 การวัดความดันโลหิตด้วยวิธีฟังเสียง

เป็นการวัดความดันโลหิตโดยการฟังเสียงโครอทคอฟ (Korotkoff sound)ด้วยหูฟังสเตโธสโคป เมื่อปั๊มลมเข้าไปในสายรัดแขนจนเลือดไม่สามารถไหลผ่านได้ ก็จะเริ่มทำการปล่อยลมออกจากสายรัดแขนช้าๆ เมื่อแรงดันในสายรัดแขนลดลงจนเท่ากับความดันตัวบนในหลอดเลือดแดงใหญ่ที่ถูกบีบเลือดก็จะพร้อมที่จะไหลผ่านได้ เมื่อลดความดันในสายรัดแขนลงอีกนิดหนึ่ง เลือดก็จะสามารถไหลผ่านหลอดเลือดแดงที่เป็นรูแคบๆ ทำให้เกิดเสียงแรกขึ้นถ้าเอาหูฟังไปจับเสียง

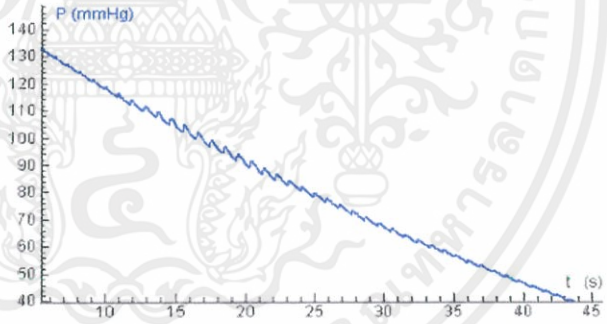
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

บริเวณข้อศอก นั่นคือความดันตัวบน และถ้าค่อยๆปล่อยลมออกไปอีกเรื่อยๆเสียงก็จะดังต่อเนื่องจนกระทั่งแรงดันในสายรัดแขนมีค่าเท่ากับความดันตัวล่างเสียงก็จะเงียบลงไป

รูปที่ 2.1 แสดงอุปกรณ์วัดความดันโลหิตแบบฟังเสียง

2.1.2.2 การวัดความดันโลหิตด้วยวิธีการตรวจจับแรงดัน

เป็นการตรวจวัดความดันโลหิตโดยการตรวจวัดความดันที่กระเพื่อมแทนการฟังเสียง เมื่อปั๊มลมเข้าไปในสายรัดแขนจนถึงค่าๆหนึ่งที่เลือดไม่สามารถไหลผ่านหลอดเลือดได้ก็จะปล่อยลมออกช้าๆ แรงดันในสายรัดแขนจะลดลงเรื่อยๆ พอถึงจุดที่เลือดสามารถไหลผ่านหลอดได้ แรงดันลมของสายรัดแขนจะกระเพื่อมและเซนเซอร์ความดันก็จะสามารถตรวจจับการกระเพื่อมของแรงดันลมที่กำลังลดลงได้ จุดที่แรงดันลมกระเพื่อมจุดแรกจะเป็นค่าความดันตัวบน และจุดสุดท้ายที่แรงดันลมกระเพื่อมจะเป็นค่าความดันตัวล่าง



รูปที่ 2.2 แสดงกราฟการกระเพื่อมของความดันเมื่อปล่อยลมออกอย่างช้าๆ

2.1.2.3 ข้อเปรียบเทียบและการประยุกต์ใช้ร่วมกันระหว่างการวัดความดันโลหิตแบบใช้หูฟังและแบบการตรวจจับแรงดัน

โดยปกติแล้วการวัดโดยใช้วิธีการตรวจจับแรงดันนั้น จะได้รับค่าความดันที่สูงกว่าการวัดในแบบใช้หูฟัง อันเนื่องมาจากการวัดแบบใช้หูฟัง จะเป็นการฟังเสียงเลือดที่สูบฉีดชนผนังเส้นเลือด ดังนั้นการสั่นกระเพื่อมของความดันจะถูกตรวจจับก่อนที่จะได้ยินเสียงการสูบฉีดเลือดชนผนังเส้นเลือด เพราะหูของคนปกติจะได้ยินแต่เสียง ไม่ใช่การสั่นสะเทือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

วิธีการที่ดีที่สุดในการยืนยันความถูกต้องของความแม่นยำของเครื่องมือวัดแบบการตรวจจับแรงดันก็คือ ให้แพทย์ฟังเสียงด้วยหูฟังสเตโธสโคปร่วมกับเครื่องมือวัดแบบการตรวจจับแรงดัน โดยทำการทดลองที่คนไข้คนเดียวกัน และทำการเปรียบเทียบค่าที่ได้จากเครื่องกับแพทย์ผู้ใช้หูฟัง

2.2 คุณลักษณะของอุปกรณ์แต่ละตัว

2.2.1 อุปกรณ์จัดเก็บข้อมูล (DAQ)

งานทางวิศวกรรมหลายอย่างใช้อุปกรณ์จัดเก็บข้อมูลหรือDAQ นี้ต่อเชื่อมเข้ากับคอมพิวเตอร์เพื่อเก็บข้อมูลและส่งต่อไปเก็บไว้ที่หน่วยความจำของคอมพิวเตอร์ หรือเก็บอยู่ในรูปของเอกสารข้อมูลเพื่อนำมาประมวลผลอย่างละเอียดอีกครั้งหนึ่ง นอกจากนี้ยังมีอุปกรณ์บางส่วนสามารถส่งผ่านข้อมูลไปยังคอมพิวเตอร์โดยผ่านทางพอร์ทอนุกรม พอร์ทขนาน หรือพอร์ทUSB เป็นต้น การเก็บข้อมูลจากอุปกรณ์วัดด้วยคอมพิวเตอร์ จะต้องมีอุปกรณ์หลักๆดังต่อไปนี้

2.2.1.1 เครื่องคอมพิวเตอร์

2.2.1.2 เซนเซอร์

เป็นอุปกรณ์ที่ทำหน้าที่ตรวจรู้การเปลี่ยนแปลงทางกายภาพและเปลี่ยนแปลงให้อยู่ในรูปสัญญาณทางไฟฟ้า เพื่อทำให้อุปกรณ์จัดเก็บข้อมูลสามารถเก็บข้อมูลและแสดงผลได้

2.2.1.3 อุปกรณ์ปรับสัญญาณ

เป็นอุปกรณ์ที่ใช้ปรับสัญญาณทางไฟฟ้า สัญญาณที่ตรวจรู้จากเซนเซอร์บางทีก็อาจจะมีระดับสัญญาณที่ต่ำหรือสูงเกินไป อุปกรณ์ปรับสัญญาณนี้จะมีอยู่หลายประเภท เช่น ขยายสัญญาณ กรองสัญญาณ หรือลดทอนสัญญาณ เป็นต้น

2.2.1.4 อุปกรณ์จัดเก็บข้อมูล

เป็นอุปกรณ์ที่เชื่อมต่อกับคอมพิวเตอร์ มีหน้าที่แปลงสัญญาณจากสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล หรือจากสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก

2.2.1.5 ซอฟท์แวร์

ใช้สำหรับเขียนคำสั่งต่างๆเพื่อใช้ในการควบคุมการทำงานของระบบการจัดเก็บข้อมูล

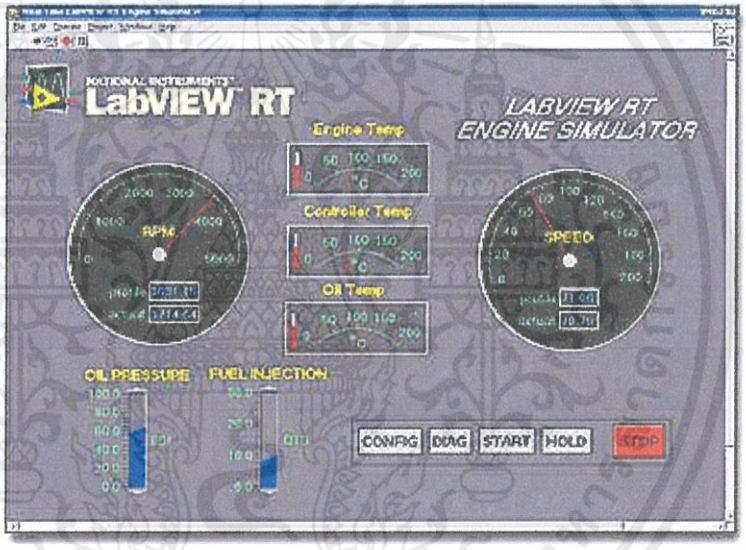
รูปที่ 2.3 แสดงอุปกรณ์การจัดเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 แล็ปวิว (LabVIEW)

โปรแกรมคอมพิวเตอร์ที่สร้างเพื่อนำมาใช้ในด้านการวัดและเครื่องมือวัดสำหรับงานทางวิศวกรรม LabVIEW ย่อมาจาก Laboratory Virtual Instrument Engineering Workbench ซึ่งหมายความว่าเป็นโปรแกรมที่สร้าง เครื่องมือวัดเสมือนจริงในห้องปฏิบัติการทางวิศวกรรม ดังนั้นจุดประสงค์หลักของการทำงานของโปรแกรมนี้ก็คือการจัดการในด้านการวัดและเครื่องมือวัด อย่างมีประสิทธิภาพ และในตัวของโปรแกรมจะประกอบไปด้วยฟังก์ชันที่ใช้ช่วยในการวัดมากมายและแน่นอนที่สุด โปรแกรมนี้จะมีประโยชน์อย่างสูงเมื่อใช้ร่วมกับเครื่องมือวัดทางวิศวกรรมต่างๆ

สิ่งที่แล็ปวิวแตกต่างจากโปรแกรมอื่นอย่างเห็นได้ชัดที่สุดก็คือ แล็ปวิวเป็นโปรแกรมประเภท จียูไอ (Graphic User Interface) โดยสมบูรณ์ นั่นคือเราไม่จำเป็นต้องเขียนโค๊ด หรือคำสั่งใดๆ ทั้งสิ้น และที่สำคัญลักษณะภาษาที่ใช้ในโปรแกรมนี้เราจะเรียกว่าเป็น ภาษารูปภาพ หรือเรียกอีกอย่างว่าภาษา จี (Graphical Language) ซึ่งจะแทนการเขียนโปรแกรมเป็นบรรทัดอย่างที่เราคุ้นเคยกับภาษาพื้นฐาน เช่น ซี, เบสิก หรือ ฟอร์ทราน

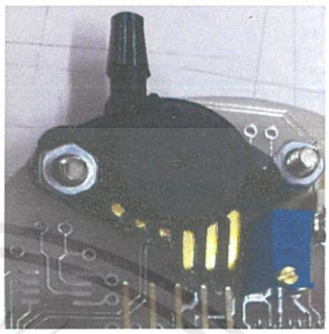


รูปที่ 2.4 แสดงหน้าตาของโปรแกรมแล็ปวิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 เซนเซอร์ความดัน MPX5050

MPX5050 เป็นเซนเซอร์ความดัน ที่ใช้หลักการเพียโซรีซิสทีฟซึ่งออกแบบมาให้ใช้ได้อย่างหลากหลายในการประยุกต์ใช้ แต่จะเจาะจงไปยังการใช้ร่วมกับไมโครคอนโทรลเลอร์ หรือไมโครโปรเซสเซอร์ ร่วมกับอุปกรณ์แปลงสัญญาณอนาล็อกเป็นดิจิตอล



รูปที่ 2.5 แสดงเซนเซอร์ความดัน MPX5050

2.2.3.1 คุณลักษณะเฉพาะของอุปกรณ์

1) ค่าความคลาดเคลื่อนสูงสุด 2.5% ที่อุณหภูมิใช้งาน 0 ถึง 85 องศาเซลเซียส
2) มีการชดเชยอุณหภูมิที่อุณหภูมิใช้งาน -40 ถึง 125 องศาเซลเซียส
3) เหมาะสำหรับระบบที่ใช้ไมโครโพรเซสเซอร์ หรือไมโครคอนโทรลเลอร์ในการควบคุม
4) ชิ้นงานทำจากอีพ็อกซีซึ่งมีความทนทาน
5) มีส่วนเสริมที่ง่ายต่อการใช้งาน

ตารางที่ 2.1 แสดงคุณลักษณะในการทำงานของเซนเซอร์ความดัน MPX5050

คุณสมบัติ	ต่ำสุด	ใช้งาน	สูงสุด	หน่วย
พิสัยของความดัน	0	-	50	kPa
แหล่งจ่ายแรงดันไฟฟ้า	4.75	5	5.25	Vdc
แหล่งจ่ายกระแส	-	7.0	10.0	mAdc
ออฟเซตแรงดันต่ำสุด	0.088	0.2	0.313	Vdc
เอาท์พุตเต็มสเกล (0ถึง85°C)	4.587	4.7	4.813	Vdc
สแปนเต็มสเกล (0ถึง85 °C)	-	4.5	-	Vdc
ความแม่นยำ (0ถึง85 °C)	-	-	± 2.5	%Vfss
ความไว	-	90	-	mV/kPa
เวลาตอบสนอง	-	1.0	-	ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.2 หลักการของปรากฏการณ์เพียโซรีซิสทีฟ

หลักการเชิงกลของเพียโซรีซิสทีฟ การเปลี่ยนแปลงในช่องว่างของอะตอมที่มีผลมาจากความเครียดส่งผลกระทบต่อช่องว่างระหว่างแถบ(Bandgaps) ทำให้อิเล็กตรอนนั้นสามารถเพิ่มขึ้นในช่องว่างนำไฟฟ้า(Conduction band) สิ่งนี้เกิดผลลัพธ์คือ การเปลี่ยนแปลงสภาพต้านทานของเซมิคอนดักเตอร์ ซึ่งเพียโซรีซิสทิวิตี้อธิบายได้โดย

$$\rho_\sigma = \frac{\partial_\rho}{\rho} \qquad (1.1)$$

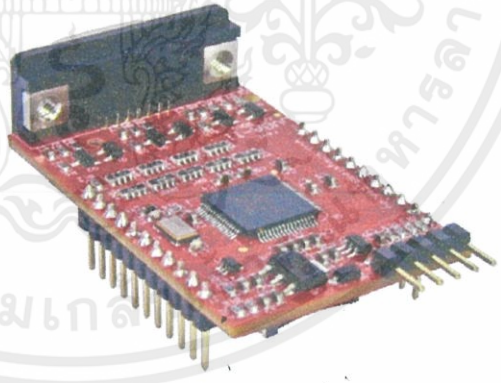
∂_ρ = การเปลี่ยนแปลงสภาพการต้านทาน

ρ = สภาพการต้านทานเดิม

ε = ความเครียด

ปรากฏการณ์เพียโซรีซิสทีฟของเซมิคอนดักเตอร์นั้นได้ถูกใช้เพื่อเป็นเซนเซอร์ โดยใช้วัสดุทุกๆชนิดของเซมิคอนดักเตอร์ เช่น เจอมาเนียม โพลีคริสตัลไลน์ซิลิคอน อะมอฟอสซิลิคอน และ ซิงเกิลคริสตัลซิลิคอน ตั้งแต่ซิลิคอนนั้นเป็นวัสดุที่ใช้สำหรับวงจรแบบอินทริเกตและอนาล็อก การใช้อุปกรณ์เพียโซลิซิสทีฟของซิลิคอนได้ถูกสนใจเป็นอย่างมากเนื่องจากง่ายต่อการตรวจรู้ความเค้น

2.2.4 ไมโครวีจีเอทู(GFX)



รูปที่ 2.6 แสดงภาพของอุปกรณ์ไมโครวีจีเอทู(GFX)

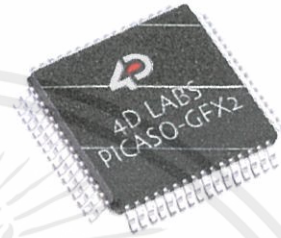
เป็นอุปกรณ์ที่ออกแบบมาได้กะทัดรัดและมีราคาเหมาะสมกับประสิทธิภาพการทำงานที่สามารถแสดงผลกราฟฟิคในรูปแบบ VGA ได้โดยอาศัย PICASO-GFX2 เป็นตัวควบคุมกราฟฟิค และอุปกรณ์นี้ยังสามารถแสดงผลของกราฟฟิคในความละเอียดระดับ QVGA/VGA/WVGA ให้กับโครงการที่ใช้ระบบฝังตัวเนื่องด้วยความสามารถที่สูงในการแสดงผลทางกราฟฟิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4.1 PICASO-GFX2

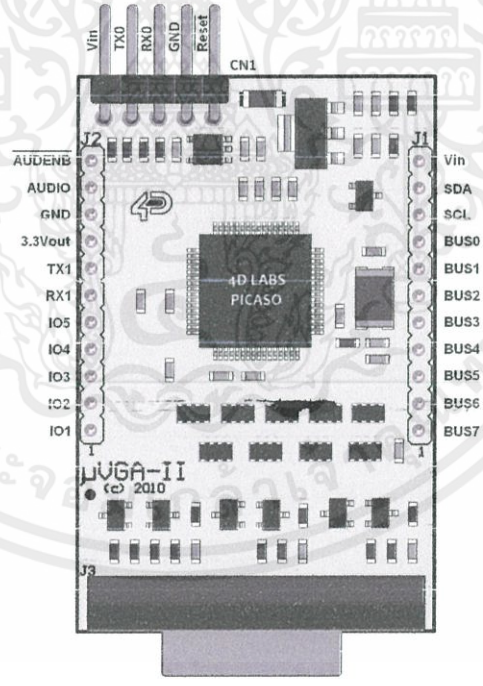
ในส่วนของ PICASO-GFX2 นั้นทำงานในแบบสแตนด์อโลน ซึ่งจะอนุญาตให้ผู้ใช้สามารถควบคุมฮาร์ดแวร์ทั้งหมด ด้วยความสามารถอันนี้ทำให้ไม่ต้องการตัวควบคุมภายนอกสำหรับควบคุมฮาร์ดแวร์เหล่านั้น

การเขียนโปรแกรมสำหรับอุปกรณ์ตัวนี้นั้นจะใช้ภาษากราฟฟิค 4D ซึ่งจะเป็นภาษาเฉพาะสำหรับไมโครคอนโทรลเลอร์ชนิดนี้ โดยการป้อนโปรแกรมจะทำการป้อนผ่านสายเคเบิลโปรแกรมมิ่ง 4D



รูปที่ 2.7 แสดงภาพของอุปกรณ์ PICASO-GFX2

2.2.4.2 ขาการเชื่อมต่อกับฮาร์ดแวร์



รูปที่ 2.8 แสดงขาการเชื่อมต่อของไมโครวีจีเอทู

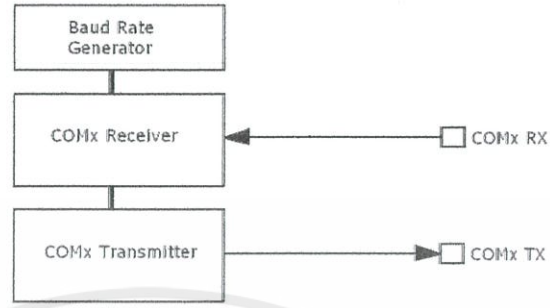
พอร์ตอนุกรม – COM0, COM1, UARTS ไมโครคอนโทรลเลอร์ PICASO-GFX2 มีสองพอร์ทอนุกรมที่สามารถเชื่อมต่อกับอุปกรณ์อนุกรมอื่นๆได้ นั่นคือ COM0 และ COM1 คุณสมบัติหลักคือ

1) การส่งข้อมูลแบบ ฟูล-ดูเพล็ก 8 บิต และการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ ไม่ว่ากรณีใดๆ

2) ฟอแมตข้อมูลคือ 8 บิต ไม่มีพาริตี้บิต และมี 1 บิตหยุด

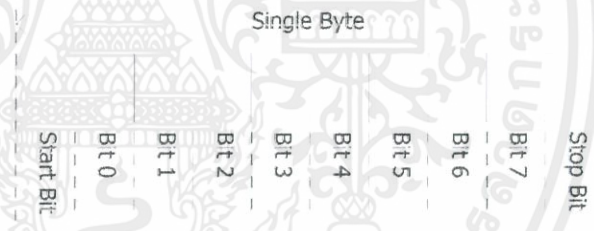
3) อัตราบอดที่อิสระ เริ่มต้นจาก 300 จนถึง 256k บอด



รูปที่ 2.9 แผนภาพแสดงการรับส่งข้อมูลที่สัมพันธ์กับอัตราบอด

4) การส่งและรับข้อมูลแบบไบต์เดี่ยว หรือ การบริการบัฟเฟอร์แบบเต็ม

การรับส่งข้อมูลแบบไบต์เดี่ยวนั้นประกอบไปด้วย บิตเริ่มต้น ข้อมูล 8 บิต และตามด้วยบิตหยุด โดยบิตเริ่มต้นจะเป็นศูนย์เสมอ ในขณะที่บิตหยุดนั้นจะเป็นหนึ่งเสมอ LSB จะถูกส่งออกหลังเป็นอันดับแรก หลังจากบิตเริ่มต้น รูปต่อไปนี้แสดงการส่งข้อมูล



รูปที่ 2.10 แสดงการส่งข้อมูลแบบอนุกรมไบต์เดี่ยว

COM0 ยังเป็นอินเตอร์เฟสหลักสำหรับผู้ใช้ภาษา 4DGL ในการดาวน์โหลดโปรแกรมและการตั้งค่าจาก PmmC อีกด้วย เมื่อใดที่โปรแกรมที่ถูกคอมไพล์แล้วได้ถูกดาวน์โหลดเข้าสู่ไมโครคอนโทรลเลอร์ โค๊ดของผู้ใช้งานจะถูกดำเนินการ หลังจากนั้นพอร์ทอนุกรมจะว่างสำหรับแอพพลิเคชั่นของผู้ใช้งาน

ขา TX0 (ส่งข้อมูลแบบอนุกรม COM0) คือพอร์ทส่งข้อมูลอนุกรมแบบไม่เป็นจังหวะ ควรการเชื่อมต่อขานี้เข้ากับอุปกรณ์อนุกรมอื่นๆในสัญญาณรับข้อมูล ขาการเชื่อมต่อนี้ทนแรงดันไฟฟ้าได้ 5 โวลต์

ขา RX0 (รับข้อมูลแบบอนุกรม COM0) คือพอร์ทรับข้อมูลอนุกรมแบบไม่เป็นจังหวะ ควรการเชื่อมต่อขานี้เข้ากับอุปกรณ์อนุกรมอื่นๆในสัญญาณส่งข้อมูล ขาการเชื่อมต่อนี้ทนแรงดันไฟฟ้าได้ 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GPIO (อินเตอร์เฟสอินพุตเอาท์พุตทั่วไป) ภายในอุปกรณ์นี้จะมีขาการเชื่อมต่อ อินพุตเอาท์พุตทั่วไปให้ผู้ใช้งาน 13 ขา โดยจะถูกแบ่งกลุ่มเป็น IO1..IO5 และ BUS0..BUS7

โดยขาอินพุตเอาท์พุตทั้งห้า จะให้ความยืดหยุ่นของบิตที่อิสระในการทำงาน ในขณะที่ขาทั้ง 8 (BUS0..BUS7) ทำงานในรูปแบบไบต์ IO4 และ IO5 ยังเป็นรับสัญญาณเพื่อควบคุม GPIO บัส GPIO บัสสามารถเขียนหรืออ่านได้โดยการรับพัลส์ต่ำ (50 นาโนวินาทีหรือมากกว่า) ส่วน IO4/BUS_RD หรือ IO5/BUS_WR จะสำหรับการอ่านหรือเขียน

ขาระบบ (System Pins) ได้แก่ ขา Vin (แรงดันไฟฟ้าอินพุต) ซึ่งทำหน้าที่รับไฟจากแหล่งจ่ายโดยมีพิสัย 4.0 โวลต์ ถึง 5.5 โวลต์ กระแสตรง โดยปกติจะทำงานอยู่ที่ 5.0 โวลต์ ขา 3.3Vout (เอาท์พุต3.3 โวลต์) วงจรภายนอกที่ต้องการแหล่งจ่าย 3.3 โวลต์สามารถรับได้จากขานี้โดยจะมีข้อจำกัดคือสามารถจ่ายกระแสได้มากที่สุด 400 มิลลิแอมป์ ขา GND (กราวด์) ขานี้จำเป็นต้องเชื่อมต่อกับกราวด์ ขา RESET (ขารีเซต) มาสเตอร์รีเซตสัญญาณ ภายในจะถูกดึงแรงดันไฟฟ้าขึ้นเป็น 3.3โวลต์ผ่านตัวต้านทาน 4.7k โอห์ม การที่พัลส์ต่ำเกิดขึ้นเดิน 2 ไมโครวินาที จะทำให้เกิดการรีเซต ถ้าโมดูลต้องการจะรีเซตจากภายนอก สามารถทำได้โดยใช้วงจรคอลเล็คเตอร์แบบเปิด

2.2.4.3 ส่วนของซอฟท์แวร์

4DGL ภายใน PICASO-GFX2 เป็นหน่วยประมวลผลที่อยู่ในตระกูลที่ประมวลผลด้วยเอนจิ้น E.V.E (Extensible Virtual Engine) EVE เป็นหน่วยประมวลผลที่มีประสิทธิภาพสูง ด้วยคำสั่งที่เป็นแบบไบต์โค้ดที่กว้างขวาง เพื่อที่จะดำเนินการโปรแกรมที่ถูกเขียนขึ้น 4DGL (4D Graphics Language) เป็นภาษาที่ถูกสร้างขึ้นสำหรับ EVE เอนจิ้นโดยเฉพาะ มันเป็นภาษาที่มีระดับสูง ซึ่งง่ายต่อการเรียนรู้และเข้าใจ และยังมีประสิทธิภาพพอที่จะทำงานให้กับระบบฝังตัวที่ใช้แอพพลิเคชั่นแบบกราฟฟิค

4DGL คือภาษาที่อนุญาตให้การเขียนและพัฒนาโปรแกรมนั้นเป็นไปได้อย่างรวดเร็วอย่างยิ่งและคำสั่งที่เขียนนั้นยังมีพื้นฐานการออกแบบมาจากภาษาที่เป็นที่ยอดนิยมต่างๆ อย่างเช่น ภาษาซี เบสิค พาสคาล และอื่นๆ ผู้เขียนโปรแกรมที่คุ้นเคยกับภาษาเหล่านี้จะรู้สึกคุ้นเคยเช่นเดียวกันกับ 4DGL ซึ่งจะประกอบไปด้วยคำสั่งที่คุ้นเคยอย่างเช่น IF..ELSE..ENDIF, WHILE..WEND, REPEAT..UNTIL, GOSUB..ENDSUB, GOTO, PRINT หรือจะเป็นคำสั่งที่พิเศษอย่างเช่น SERIN, SEROUT, GFS_LINE, GFX_CIRCLE และอื่นๆอีกมากมาย

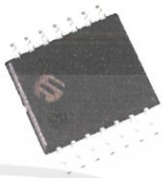
```
var buffer[100];    // 200 character buffer for a source string
var n, *p;

func main()
    to(buffer); print("Testing 1 2 3");

    p := str_Ptr(buffer);          // get a byte pointer from a word region

    n := 0;

    while (n <= strlen(buffer))

        print( [HEX2] str_GetByte(p + n++)," ");          // print all the chars hex values


    wend

    repeat
    forever
endfunc
```

รูปที่ 2.11 แสดงภาพตัวอย่างการเขียนโปรแกรมภาษา 4DGL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

2.2.5 อุปกรณ์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล MCP3424

เป็นอุปกรณ์แปลงสัญญาณจากสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอลที่มีสัญญาณรบกวนต่ำและมีความแม่นยำสูง อยู่ในตระกูลของ MCP342X จากบริษัท Microchip Technology ซึ่งสามารถแปลงสัญญาณได้ละเอียดมากถึง 18 บิต



รูปที่ 2.12 แสดงภาพของอุปกรณ์ MCP3424

2.2.5.1 คุณสมบัติของอุปกรณ์

1) แรงดันไฟฟ้าอ้างอิงมีพิสัยคือ -2.048V ถึง +2.048V
2) มีช่องการเชื่อมต่อ(Channel) 4 ช่องด้วยกัน
3) สามารถให้เอาท์พุตที่มีอัตราการเปลี่ยนแปลงของสัญญาณที่ 3.75, 15, 60 หรือ 240 ตัวอย่างต่อหนึ่งวินาที (โดยปรับตั้งค่าได้จากการตั้งค่าบิตผ่านทาง I2C อินเตอร์เฟส)
4) สามารถปรับอัตราขยายก่อนที่จะรับการแปลงสัญญาณได้ที่ x1, x2, x4 หรือ x8 ซึ่งทำให้สามารถแปลงสัญญาณที่อ่อนมากๆให้เป็นสัญญาณที่มีความละเอียดสูง
5) MCP3424 นั้นมีสองโหมดการทำงาน โหมดแรกคือ โหมดแปลงสัญญาณหนึ่งครั้ง (One-Shot Conversion Mode) ซึ่งอุปกรณ์จะทำการแปลงสัญญาณเพียงครั้งเดียวแล้วเข้าสู่โหมดแสตนบาย เพื่อรอคำสั่งการแปลงสัญญาณอีกครั้งหนึ่ง สิ่งนี้ทำให้ลดการสิ้นเปลืองกระแสไฟฟ้าได้อย่างดี และในโหมดการทำงานแบบแปลงสัญญาณต่อเนื่อง (Continuous Conversion Mode) การแปลงสัญญาณจะทำโดยต่อเนื่องตามความเร็วการแปลงข้อมูลที่ได้ตั้งไว้ ดังนั้นอุปกรณ์ที่รับข้อมูลไปก็จะได้ข้อมูลที่ปัจจุบันมากที่สุด
6) การรับส่งข้อมูลผ่าน I2C อินเตอร์เฟส ซึ่งสามารถทำงานได้ตั้งแต่โหมดปกติ (100kHz) โหมดเร็ว (400kHz) และโหมดความเร็วสูง (3.4MHz)
7) I2C แอดเดรสสำหรับ MCP3424 สามารถกำหนดได้โดยขา Adr0 และ Adr1 ผู้ใช้สามารถตั้งค่าได้อุปกรณ์นี้ให้เป็นหนึ่งในแปดแอดเดรสโดยต่อขาแอดแดรสด้วย Vdd, Vss, หรือปล่อยลอยก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5.2 รีจิสเตอร์ควบคุมของ MCP3424

อุปกรณ์นี้จะมีรีจิสเตอร์ควบคุมในลักษณะ 8 บิต เพื่อใช้ในการเลือกว่าจะใช้ช่องการเชื่อมต่อ โหมดการแปลงสัญญาณ อัตราการแปลงสัญญาณ และอัตราขยาย PGA รีจิสเตอร์นี้อนุญาตให้ผู้ใช้งานเปลี่ยนแปลงข้อกำหนดการทำงานของอุปกรณ์และตรวจสอบการทำงานของอุปกรณ์ได้ ผู้ใช้งานสามารถเขียนรีจิสเตอร์ควบคุมนี้ได้ตลอดเวลาการทำงานของอุปกรณ์ โดยรีจิสเตอร์จะแสดงในตารางต่อไปนี้

ตารางที่ 2.2 รีจิสเตอร์ควบคุม MCP3424

R/W-1	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
$\overline{\text{RDY}}$	C1	C0	$\overline{\text{O}}$/C	S1	S0	G1	G0
1	0	0	1	0	0	0	0

Bit 7 $\overline{\text{RDY}}$ คือบิตพร้อม บิตนี้จะเป็นข้อมูลที่ชี้ให้เห็นว่ารีจิสเตอร์เอาท์พุตได้ถูกอัพเดตด้วยการแปลงสัญญาณที่เป็นปัจจุบัน และในโหมดแปลงสัญญาณหนึ่งครั้ง (One-Shot Conversion Mode) การเขียนที่บิตนี้ด้วย 1 จะทำให้อุปกรณ์แปลงสัญญาณใหม่

Bit 6-5 C1-C0 คือบิตเลือกช่องการเชื่อมต่อ ถ้าผู้ใช้งานตั้งค่าบิตเป็น 00 จะเลือกช่องการเชื่อมต่อที่1 เมื่อตั้งค่าเป็น01 จะเลือกช่องการเชื่อมต่อที่2 เมื่อตั้งค่าเป็น10 จะเลือกช่องการเชื่อมต่อที่3และเมื่อตั้งค่าเป็น11จะเลือกช่องการเชื่อมต่อที่4

Bit 4 $\overline{\text{O}}$/C คือบิตเลือกโหมดการทำงาน โดยหากตั้งค่าเป็น1 จะเลือกโหมดการแปลงสัญญาณต่อเนื่อง หากตั้งค่าเป็น0 จะเลือกโหมดแปลงสัญญาณหนึ่งครั้ง

Bit 3-2 S1-S0 คือบิตเลือกอัตราการแปลงสัญญาณ หากตั้งค่าเป็น00 จะเลือก 240 SPS (12 บิต) (ตั้งค่าจากโรงงาน) หากตั้งค่าเป็น01 จะเลือก 60 SPS (14 บิต) หากตั้งค่าเป็น10 จะเลือก15 SPS (16 บิต) ตั้งค่าเป็น11 จะเลือก3.75 SPS (18 บิต)

Bit 1-0 G1-G0 คือบิตเลือกอัตราขยายสัญญาณ โดยตั้งค่าเป็น00 จะขยายสัญญาณ 1 เท่า ตั้งค่าเป็น01 จะขยายสัญญาณ 2 เท่า ตั้งค่าเป็น10 จะขยายสัญญาณ 4 เท่า ตั้งค่าเป็น11 จะขยายสัญญาณ 8 เท่า

2.2.6 อุปกรณ์แสดงเวลาและวันที่ DS1307 (Real-Time Clock)

DS1307 คือนาฬิกาและปฏิทิน ที่ใช้พลังงานต่ำ และเป็น BCD เป็นสมบูรณ์ โดยข้อมูลและแอดเดรสสามารถส่งผ่าน I2C ได้ในรูปแบบอนุกรม โดยเวลาและวันที่นั้นจะสามารถแสดงได้ตั้งแต่ วินาที นาที ชั่วโมง วัน วันที่ เดือน และ ปี หากจบเดือนแต่ล่ะเดือนจะสามารถปรับให้เข้าสู่เดือนที่มีวันน้อยกว่า 31 วันได้อย่างอัตโนมัติ รวมไปถึงการทำให้ถูกต้องของปีที่มีจำนวนวันมากกว่าปกติด้วย นาฬิกาสามารถทำงานได้ทั้งในรูปแบบ 24 ชั่วโมง หรือแบบ 12 ชั่วโมงพร้อมทั้งตัวบ่งบอกถึง AM/PM DS1037 มีการวงจรตรวจจับพลังงานไฟฟ้า ที่จะตรวจจับเมื่อเกิดกรณีพลังงานไฟฟ้าล้มเหลว และจะสับเปลี่ยนไปใช้พลังงานสำรองอย่างอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.13 แสดงอุปกรณ์แสดงเวลาและวันที่ DS1307

2.2.6.1 คุณสมบัติของอุปกรณ์

1) แสดงเวลาจริงนับตั้งแต่ วินาที นาที ชั่วโมง วันที่ของเดือน เดือน และวันในแต่ละอาทิตย์ รวมไปถึงปีที่มีวันมากกว่าปกติ ซึ่งถูกปรับให้ถูกต้องไปจนถึง ปี 2100
2) 56-ไบต์ แบตเตอร์รี่สำรอง และหน่วยความจำแบบไม่ลบเลือน
3) I2C อินเตอร์เฟส แบบอนุกรม
4) มีสัญญาณเอาท์พุตแบบเหลี่ยมที่โปรแกรมได้
5) สามารถตรวจจับพลังงานไฟฟ้าล้มเหลว และสลับเปลี่ยนไปใช้แบตเตอร์รี่สำรองได้
6) กินพลังงานน้อยกว่า 500 นาโนแอมป์
7) สามารถใช้งานได้ตั้งแต่อุณหภูมิ –40 องศาเซลเซียส ถึง 85 องศาเซลเซียส
8) มีตัวถังแบบ 8 ขา

2.2.6.2 รีจิสเตอร์เก็บเวลา

เวลาและปฏิทิน ข้อมูลของเวลาและปฏิทินจะได้มาโดยการอ่านรีจิสเตอร์ไบต์ที่เหมาะสม ดังที่แสดงในตาราง ซึ่งแสดงถึง รีจิสเตอร์ RTC เวลาและปฏิทินจะตั้งค่าหรือเริ่มต้นโดยเขียนรีจิสเตอร์ที่ถูกต้อง ข้อมูลรีจิสเตอร์ของเวลาและปฏิทิน จะอยู่ในรีจิสเตอร์ที่มีฟอแมตเป็น BCD แต่ละวันใน 1 สัปดาห์จะเพิ่มขึ้นในเวลาเที่ยงคืน ค่าจะสัมพันธ์กับแต่ละวันใน 1 สัปดาห์ตามที่ผู้ใช้กำหนด แต่จะต้องเรียงตามลำดับ (เช่น ถ้า 1 เท่ากับวันอาทิตย์ ดังนั้น 2 เท่ากับวันจันทร์ และเป็นแบบนี้ไปเรื่อยๆ) ความไม่สมเหตุสมผลของเวลาและวันที่นั้น จะเกิดในการควบคุมที่ไม่ได้กำหนด บิต 7 ของรีจิสเตอร์ 0 คือบิตที่เวลาหยุด เมื่อบิตนี้ถูกตั้งค่าเป็น 1 เครื่องกำเนิดสัญญาณ จะถูกปิด และเมื่อเปลี่ยนค่าเป็น 0 เครื่องกำเนิดสัญญาณจะทำงาน

ใน DS1307 จะสามารถทำงานได้ทั้งโหมด 12 ชั่วโมง หรือ 24 ชั่วโมง บิตที่ 6 ของรีจิสเตอร์ชั่วโมงจะบิตที่จะเลือกว่าจะใช้โหมดเวลาแบบ 12 ชั่วโมง หรือ 24 ชั่วโมง เมื่อบิตนั้นอยู่ในสถานะสูง โหมด 12 ชั่วโมงจะถูกเลือกใช้ และในโหมด 12 ชั่วโมง บิตที่ 5 คือบิตแสดง AM/PM โดยที่ถ้าเป็นสถานะสูง แสดงถึง PM ส่วนในโหมด 24 ชั่วโมงนั้น บิตที่ 5 คือ บิตแสดงวินาที สิบชั่วโมง (20 ถึง 23ชั่วโมง) ค่าชั่วโมงจะต้องถูกใส่ค่าใหม่อีกครั้ง เมื่อมีการเปลี่ยนโหมด

เมื่อมีการอ่านหรือเขียน รีจิสเตอร์เวลาและวันที่ บัฟเฟอร์ที่สอง จะถูกใช้เพื่อป้องกันการผิดพลาดเมื่อรีจิสเตอร์ภายในถูกอัพเดต เมื่อมีการอ่านเวลาและวันที่ บัฟเฟอร์ของผู้ใช้จะดำเนินการพร้อมกันกับ รีจิสเตอร์ภายใน บนสัญญาณเริ่มต้นของ I2C ข้อมูลของเวลาจะอ่านจากรีจิสเตอร์ที่สอง ในขณะที่เวลายังเดินอยู่ สิ่งนี้ทำให้ไม่ต้องการการอ่านซ้ำในกรณีที่ รีจิสเตอร์ภายในถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัพเดตในขณะที่ถูกอ่านอยู่ ห่วงโซ่การหารนั้นจะถูกรีเซตเมื่อรีจิสเตอร์ที่สองนั้นได้ถูกเขียน การส่งข้อมูลการเขียนนั้นจะเกิดขึ้นเมื่อ I2C ทำการตอบรับ (Acknowledge) จาก DS1307 เมื่อใดที่โซ่การหารนั้นได้ถูกรีเซตเพื่อหลีกเลี่ยงการโรลโอเวอร์ รีจิสเตอร์เวลาและวันที่ที่เหลืออยู่จะต้องถูกเขียนภายในหนึ่งวินาที

ตารางที่ 2.3 แสดงรีจิสเตอร์เก็บเวลา

แอดเดรส	บิตที่ 7	บิตที่ 6	บิตที่ 5	บิตที่ 4	บิตที่ 3	บิตที่ 2	บิตที่ 1	บิตที่ 0	ฟังก์ชั่น	พิสัย
00H	CH	10 วินาที			วินาที				วินาที	00-59
01H	0	10 นาที			นาที				นาที	00-59
02H	0	12	10 ชั่วโมง	10 ชั่วโมง	ชั่วโมง				ชั่วโมง	1-12 +AM/PM
		24	PM/AM							00-23
03H	0	0	0	0	0	วัน			วัน	01-07
04H	0	0	10 วันที่		วันที่				วันที่	01-31
05H	0	0	0	10 เดือน	เดือน				เดือน	01-12
06H	10 ปี				10 ปี				ปี	00-99
07H	OUT	0	0	SQWE	0	0	RS1	RS0	ควบคุม	-
08H-3FH									แรม 56 x 8	00H-FFH

2.2.6.3 การรับส่งข้อมูล

การรับส่งข้อมูลภายใน DS1307 จะแบ่งโหมดการทำงานเป็นสองโหมด

1) โหมดรับข้อมูลแบบสเลฟ (โหมดเขียน) ข้อมูลและสัญญาณนาฬิกาจะรับผ่าน SDA และ SCL หลังจากที่แต่ละไบต์ได้รับข้อมูลแล้ว บิตตอบรับจะถูกส่งออกไป เงื่อนไขเริ่มต้นและหยุดจะถูกจำหลังจากการจองแอดเดรสของเสลฟและบิตทิศทาง (Direction bit) แอดเดรสไบต์ของสเลฟคือไบต์แรกที่มาสเตอร์จะได้รับหลังจากส่งเงื่อนไขเริ่มต้นออกไป สเลฟแอดเดรสจะประกอบไปด้วย 7 บิต แอดเดรส ของ DS1307 คือ 1101000 ตามด้วยบิตทิศทาง (R/$\overline{W}$) ซึ่งสำหรับการเขียนนั้นบิตนี้จะเป็น 0 หลังจากที่รับและถอดรหัสข้อมูลของแอดเดรสไบต์ DS1307 จะส่งเอาท์พุตตอบรับออกไปที่ SDA หลังจากที่ DS1308 ตอบรับแอดเดรสกับบิตเขียน มาสเตอร์จะส่งเวิร์ดแอดเดรสให้แก่ DS1307 สิ่งนี้จะตั้งค่าพอยท์เตอร์ของรีจิสเตอร์ใน DS1307 และด้วยการตอบรับของ DS1307 มาสเตอร์จะสามารถส่งข้อมูลได้ พอยท์เตอร์จะเพิ่มขึ้นโดยอัตโนมัติหลังจากข้อมูลได้ถูกเขียน และมาสเตอร์จะสามารถหยุดการเขียนได้โดยส่งเงื่อนไขหยุด

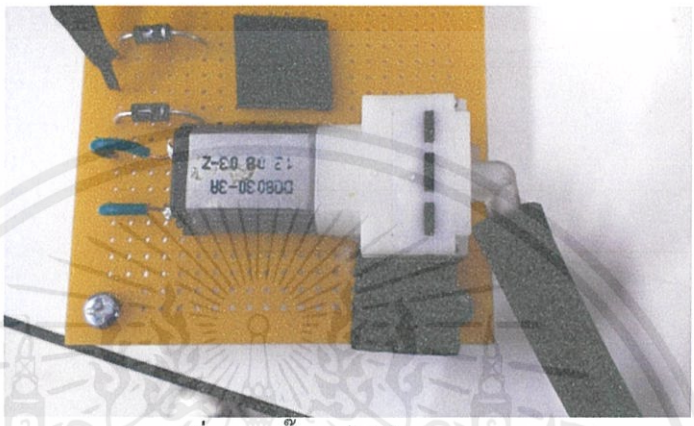
2) โหมดส่งข้อมูลแบบสเลฟ (โหมดอ่าน) แอดเดรสของสเลฟได้รับหลังจากเงื่อนไขเริ่มต้นเกิดขึ้นโดยมาสเตอร์ ออกไป สเลฟแอดเดรสจะประกอบไปด้วย 7 บิต แอดเดรส ของ DS1307 คือ 1101000 ตามด้วยบิตทิศทาง (R/$\overline{W}$) ซึ่งสำหรับการเขียนนั้นบิตนี้จะเป็น 1 สำหรับการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ
ไม่ว่ากรณีใดๆ

เขียน หลังจากได้รับและถอดรหัส DS1307 จะส่งการตอบรับ หลังจากนั้นจะเริ่มการส่งข้อมูลตามที่รีจิสเตอร์พอยท์เตอร์ได้ชี้ไว้ DS1307 จะต้องได้รับ Not Acknowledge เพื่อจบการอ่าน

2.2.7 ปั๊มลมไฟฟ้ากระแสตรง

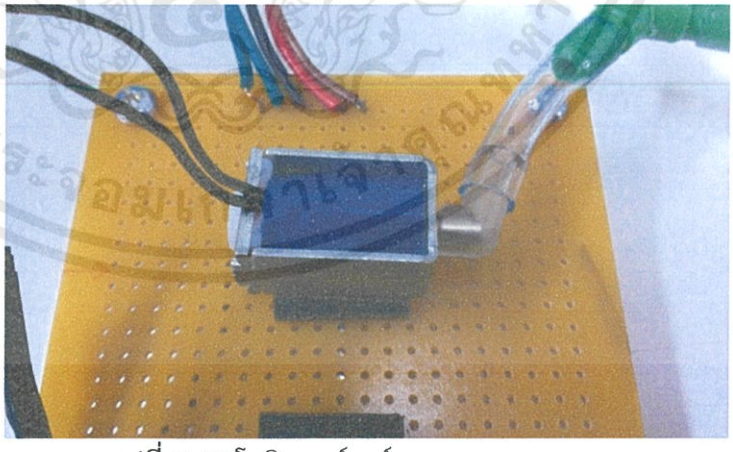
ใช้สำหรับปั๊มลมเข้าไปในสายรัดแขน โดยใช้แรงดันไฟฟ้า 5 โวลต์กระแสตรง โดยจะใช้มอเตอร์กระแสตรงเพื่อหมุนให้กลไกภายในสร้างแรงดันลมและส่งให้กับสายรัดแขน



รูปที่ 2.14 ปั๊มลมไฟฟ้ากระแสตรง

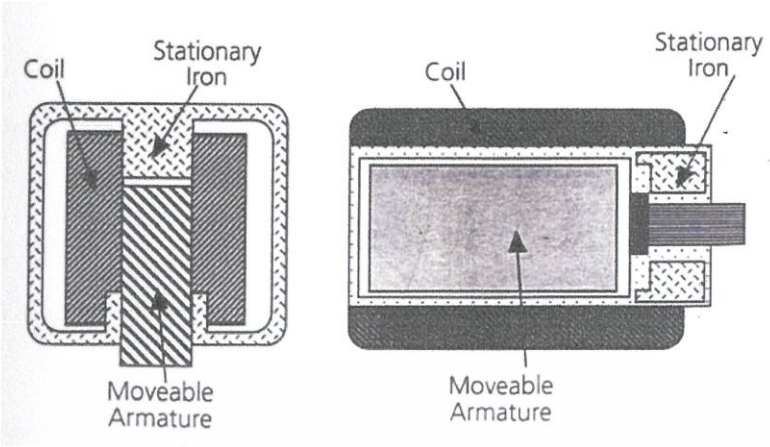
2.2.8 โซลินอยด์วาล์ว

ใช้สำหรับเปิด/ปิดในขั้นตอนการปั๊มลมเข้าและปล่อยลมออก โดยใช้แรงดันไฟฟ้า 5 โวลต์กระแสตรง หลักการทำงานคือเมื่อจ่ายกระแสไฟฟ้าให้กับขดลวด จะเกิดสนามแม่เหล็กดึงแกนเหล็กเพื่อเปิดวาล์ว หากไม่มีการจ่ายกระแสไฟฟ้าแกนเหล็กก็จะถูกสปริงดันกลับมาเปิดวาล์วเหมือนเดิม



รูปที่ 2.15 โซลินอยด์วาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงโครงสร้างของโซลินอยด์วาล์ว

2.2.9 วาล์วควบคุมอัตราการไหล

ใช้สำหรับปล่อยลมออกทีละน้อยๆด้วยค่าคงที่ เพื่อให้เซนเซอร์ความดันสามารถตรวจจับความดันที่กระเพื่อมได้ และนำไปวิเคราะห์หาความดันตัวบนและความดันตัวล่างได้

รูปที่ 2.17 วาล์วควบคุมอัตราการไหล

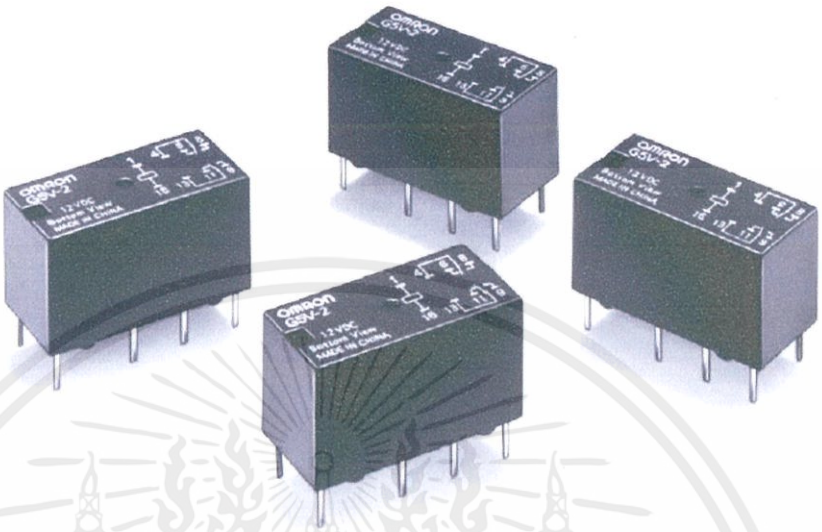
2.2.10 รีเลย์

รีเลย์คืออุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่ตัดต่อวงจร คล้ายกับสวิตซ์โดยใช้หลักการหน้าสัมผัส และการที่จะทำให้มันทำงานก็ต้องจ่ายไฟให้มันตามที่กำหนดเพราะเมื่อจ่ายไฟให้กับตัวรีเลย์มันจะทำให้หน้าสัมผัสติดกัน กลายเป็นวงจรปิดและตรงข้ามทันทีที่ไม่ได้จ่ายไฟ มันก็จะกลายเป็นวงจรเปิด

รีเลย์มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (solenoid) รีเลย์ใช้ในการควบคุมวงจร ไฟฟ้าได้อย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ รีเลย์กำลัง (power relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magneticcontactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีเลย์ควบคุม (control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุม บางทีเรียกกันง่าย ๆ ว่า "รีเลย์"



รูปที่ 2.18 รีเลย์

2.3 I2C BUS

เป็นการสื่อสารอนุกรมแบบซิงโครนัส เพื่อใช้ในการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอกอื่นๆ สายสัญญาณจะใช้เพียงแค่ 2 เส้น คือ สายข้อมูลอนุกรม (SDA) และสายสัญญาณนาฬิกาอนุกรม (SCL) ซึ่งทำให้สามารถเชื่อมต่ออุปกรณ์จำนวนหลายๆเข้าตัวเข้ากัน โดยจะใช้พอร์ทของไมโครคอนโทรลเลอร์เพียง 2 พอร์ทเท่านั้น

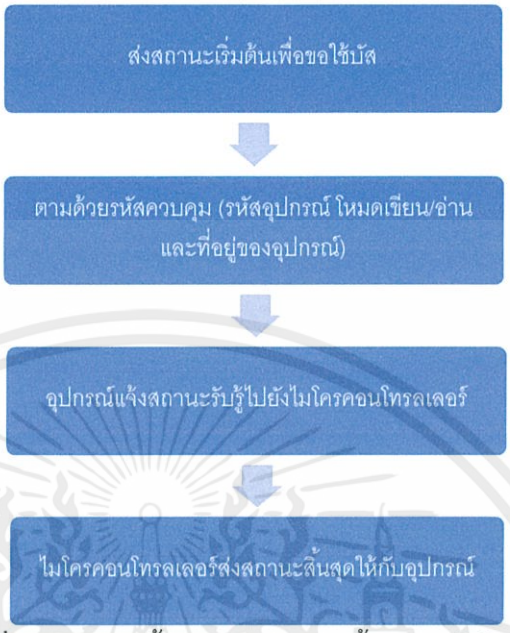
2.3.1 การเชื่อมต่ออุปกรณ์แบบ I2C BUS

อย่างที่กล่าวไว้ข้างต้น สายสัญญาณจะใช้เพียง 2 เส้น คือ SDA และ SDL ใช้สำหรับติดต่ออุปกรณ์แบบ 2 ทิศทาง โดยขาสัญญาณทั้ง 2 จะต้องต่อตัวต้านทานแบบพูลอัพขนาด10กิโลโอห์ม เพื่อที่จะทำให้เอาท์พุตเชื่อมต่อกันได้หลายๆ ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การเขียน-อ่านข้อมูลบน I2C BUS

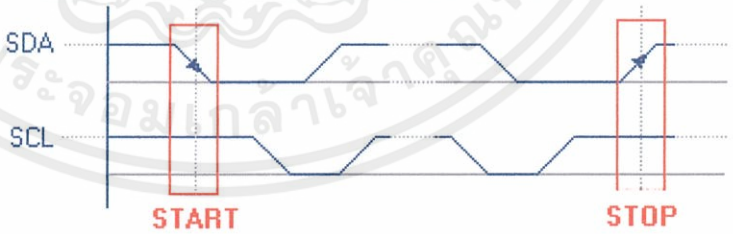
จะเริ่มต้นการส่งข้อมูลดังนี้



รูปที่ 2.19 แสดงขั้นตอนการส่งและสิ้นสุดการใช้บัส

การกำหนดโหมดเขียน/อ่านสามารถทำได้โดยกำหนดจากบิตควบคุมการเขียนอ่าน เมื่อไมโครคอนโทรลเลอร์ต้องการเขียนข้อมูลไปยังอุปกรณ์ ให้กำหนดบิตนี้เป็น 0 และถ้าต้องการให้ไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากอุปกรณ์ ให้กำหนดบิตนี้เป็น 1

2.3.3 การกำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I2C BUS



รูปที่ 2.20 แสดงแผนภาพเวลาที่กำหนดสถานะเริ่มต้นและสถานะสิ้นสุดของ I2C BUS

จากแผนภาพข้างต้นอธิบายสถานะเริ่มต้นและสิ้นสุดได้ดังนี้

1) ถ้าต้องการส่งข้อมูล ไมโครคอนโทรลเลอร์ต้องส่งสถานะเริ่มต้น คือ SDA เปลี่ยนจากลอจิก 1 เป็นลอจิก 0 ในขณะที่ SCL เป็นลอจิก 1

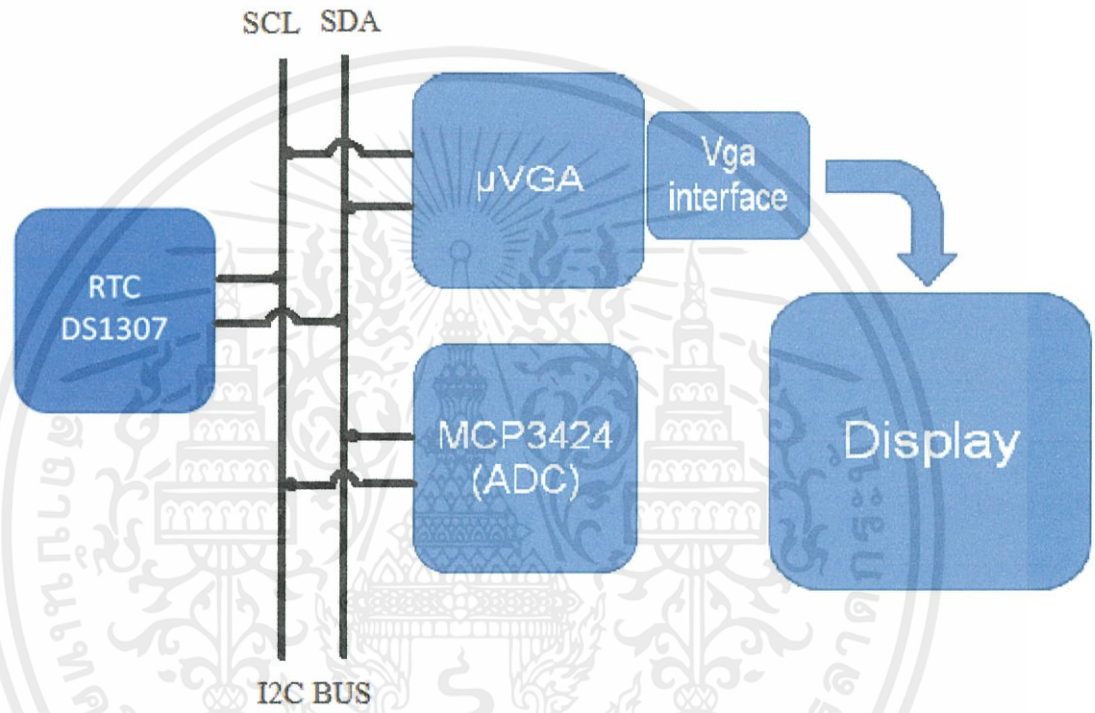
2) ถ้าต้องการสิ้นสุดการส่งข้อมูล ไมโครคอนโทรลเลอร์ต้องส่งสถานะสิ้นสุด คือ SDA เปลี่ยนจากลอจิก 0 เป็นลอจิก 1 ในขณะที่ SCL เป็นลอจิก 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

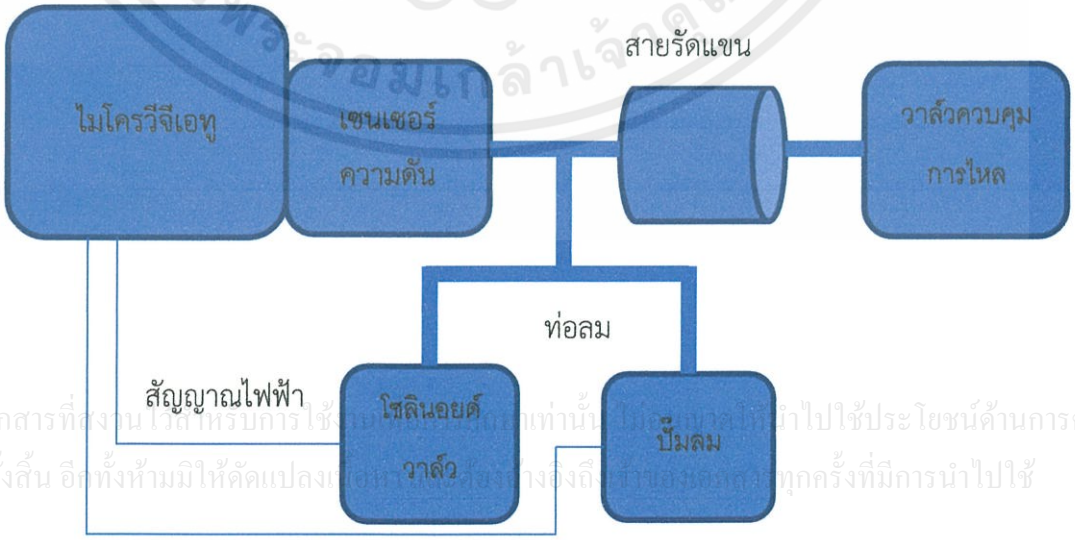
บทที่ 3
วิธีการดำเนินงาน

3.1 การเชื่อมต่ออุปกรณ์แต่ละตัวเข้าด้วยกัน

ทำการเชื่อมต่ออุปกรณ์แต่ละตัวลงบนแผงวงจรอิเล็กทรอนิกส์และเชื่อมต่ออุปกรณ์อื่นๆที่เกี่ยวกับแรงดันลมโดยมีแผนภาพดังนี้



รูปที่ 3.1 แสดงแผนภาพการเชื่อมต่อของอุปกรณ์แต่ละตัว

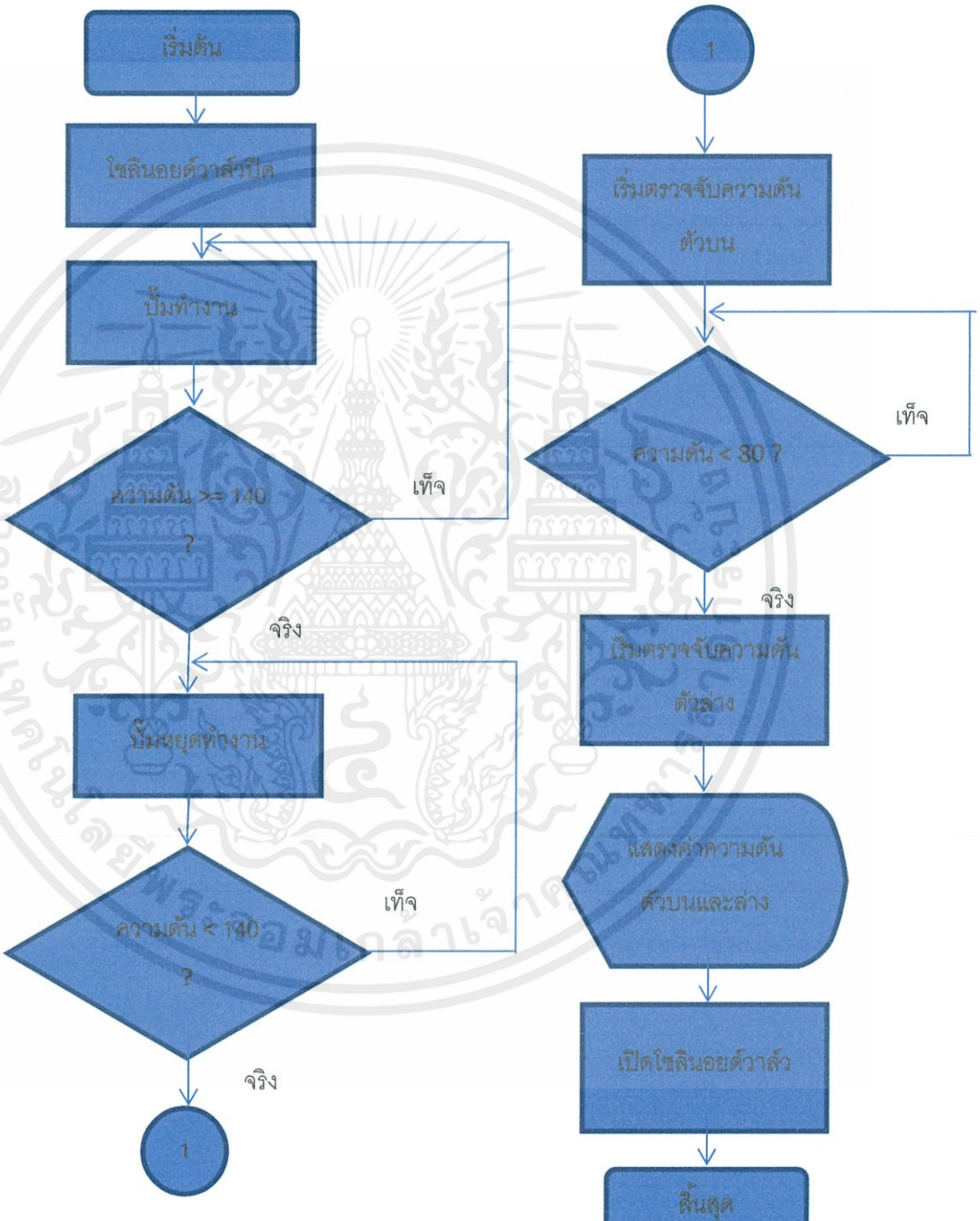


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

รูปที่ 3.2 แสดงการต่อท่อลมเข้ากับอุปกรณ์ต่างๆ

3.2 การเขียนโปรแกรม

การเขียนโปรแกรมจะใช้ภาษากราฟฟิค 4D ให้กับอุปกรณ์ไมโครวีจีเอทู(GFX) เพื่อใช้ในการควบคุมอุปกรณ์ภายนอก โดยจะทำการสื่อสารกันผ่านทาง I2C BUS โดยส่วนของโปรแกรมจะแบ่งออกเป็น 3 ส่วนคือ ส่วนโปรแกรมหลัก ส่วนตรวจจับความดันตัวบน ส่วนตรวจจับความดันตัวล่าง โดยหลักการเขียนโปรแกรม จะแสดงในผังงานต่อไปนี้



รูปที่ 3.3 ผังงานแสดงการทำงานเพื่อใช้เขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 ส่วนของโปรแกรมหลัก

```
repeat
start:
PlotGraph1 (HBP); // พลอตกราฟ
DisplayClock(); // แสดงเวลา
DisplayDate(); //แสดงวันที่
txt_MoveCursor(27,60); //เลื่อนเคอร์เซอร์
ReadADC(); //เรียกฟังชั่นอ่านค่าจาก ADC
HBP := (AnalogVal*18)/10;//แปลงค่าเป็นมิลลิเมตรปรอท
print(HBP);
print("\n");
if(HBP<20) //  ถ้าความดันต่ำกว่า 20 เข้าโหมดปั๊มลมเข้า
inflatemode(HBP);// โหมดที่จ่ายไฟให้มอเตอร์และโซลินอยด์วาล์ว
endif
if(HBP>=140) //ถ้าความดันไปถึง140มิลลิเมตรปรอทให้หยุดมอเตอร์
        deflatemode(HBP);
        state:=1;
        endif
if(state==1)
        if(HBP<140) // ถ้าความดันเริ่มลดลงให้เริ่มนำค่าไปตรวจจับ
        calculatemode(HBP);
                goto cal;
                endif
endif
forever
cal:
        calsys(); เรียกฟังชั่นตรวจจับความดันตัวบน
        sys:=calsys();
        callcaldias();เรียกฟังชั่นตรวจจับความดันตัวล่าง
        dias:=caldias();
repeat
ReadADC();
HBP := (AnalogVal*18)/10;
txt_MoveCursor(27,40);
print("systolic\n");
print (sys); // แสดงค่าที่วัดได้
print ("\n");
print("diastolic\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
print(dias-10); // เนื่องจากความดันตัวล่างผิดไปจากเครื่องวัดทั่วไปประมาณ 10 อย่างคงที่ ดังนั้นจึงลบ 10
pin_LO(IO3_PIN);
pause(10000);
SystemReset();
forever
endfunc
```

3.2.2 ส่วนของโปรแกรมที่ใช้ในการตรวจจับความดันตัวบน

```
func calsys()
pause(1000); // ตัวอย่าง1
ReadADC(); // อ่านค่า
x1:=(AnalogVal*18)/10; // ให้ค่าที่อ่านเป็น x1
ReadADC();
sys1:=(AnalogVal*18)/10; // ให้ความดันที่ x1 เป็น sys1
pause(1000); // เว้นช่วงการทำงานของโปรแกรม 1 วินาที
ReadADC(); // อ่านค่าหลังจากนั้น 1 วินาที
y1:=(AnalogVal*18)/10; // ให้เป็นy1
z1:=x1-y1; //  นำx1-y1
if(z1<=3) // ถ้า น้อยกว่า 3 แสดงว่ามีการกระเพื่อม
sys:=sys1; // ให้ความดันที่ x1 เป็นค่าความดันตัวบน
Else // ถ้าไม่น้อยกว่า 3 แสดงว่า ยังไม่เกิดการกระเพื่อม ให้สุ่มตัวอย่างถัดไป
```

3.2.3 ส่วนของโปรแกรมที่ใช้ในการตรวจจับความดันตัวล่าง

```
caldias()
pause(1000); // ตัวอย่าง1
ReadADC();
x1:=(AnalogVal*18)/10;
ReadADC();
dias1:=(AnalogVal*18)/10;
pause(1000);
ReadADC();
y1:=(AnalogVal*18)/10;
z1:=x1-y1;
if(z1>=2) // ถ้าความดันมีการกระเพื่อมลดลงให้เป็นค่าความดันตัวล่าง
dias:=dias1;
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การทดสอบการทำงานของเครื่องมือวัด

มีวิธีการทดลองดังนี้

3.3.1 ให้ผู้ทดลองสวมสายรัดแขนให้อยู่เหนือข้อพับขึ้นไปประมาณ 2.5 เซนติเมตร โดยจัดท่านั่งโดยนั่งบนเก้าอี้ เท้าทั้งสองติดพื้น และวางแขนลงบนโต๊ะให้อยู่ในระดับเดียวกับหัวใจ โดยตลอดการวัดห้ามมิให้ผู้ทดลองขยับแขนเด็ดขาดเพราะอาจจะทำให้ผลการทดลองเกิดความคลาดเคลื่อนได้

3.3.2 เริ่มทำการวัดโดยการเสียบสายUSB เข้ากันระหว่างอุปกรณ์กับคอมพิวเตอร์เพื่อทำการจ่ายกระแสไฟฟ้าให้กับอุปกรณ์ และทำการเชื่อมต่อระหว่างอุปกรณ์กับจอแสดงผลด้วยสาย VGA ผ่านทางพอร์ทVGA

3.3.3 ตรวจสอบการทำงานของอุปกรณ์และการแสดงผล ว่าอุปกรณ์แต่ล่ะชนิดทำงานได้ตามที่ได้เขียนโปรแกรมไว้หรือไม่

3.3.4 ทดสอบการตั้งค่าเพดานการกระเพื่อมของความดัน เพื่อตรวจจับความดันโลหิตตัวบนและล่าง ทำการปรับเปลี่ยนให้เหมาะสมเพื่อให้ได้การวัดความดันโลหิตที่แม่นยำที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

4.1 การทดลองวัดความดันโลหิตเทียบกับเครื่องวัดความดันโลหิตอัตโนมัติ

หลังจากทดสอบการทำงานของเครื่องมือวัดที่ทำขึ้นและได้ผลทดสอบอันเป็นที่น่าพอใจแล้ว ก็จึงนำเครื่องมือวัดมาทำการวัดความดันของผู้ทดลองจำนวน 2 คน โดยทำการวัดคนละ 5 ครั้ง จากนั้นนำผลการทดลองมาทำการเปรียบเทียบค่าความดันที่วัดได้จากเครื่องมือวัดอัตโนมัติในเวลาที่ไล่เลี่ยกันเพื่อหาเปอร์เซนต์ความคลาดเคลื่อนเฉลี่ยของเครื่องมือวัดที่ทำขึ้น

ตารางที่ 4.1 ผลการวัดจากโครงงาน

ครั้งที่	คนที่1		คนที่ 2	
1	100	58	111	58
2	102	60	102	67
3	109	60	120	60
4	102	62	120	58
5	100	60	120	62
ค่าเฉลี่ย	102.6	60	118.6	61

ตารางที่ 4.2 ผลการวัดจากเครื่องวัดความดันอัตโนมัติ

ครั้งที่	คนที่1		คนที่ 2	
1	121	60	122	50
2	110	58	117	52
3	113	58	113	55
4	111	57	115	50
5	121	61	113	48
ค่าเฉลี่ย	115.2	58.8	116	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผล

จากการทดลองวัดความดันโลหิตพบว่า เมื่อลองทำการวัดความดันโลหิตของผู้ทดลองจำนวน 2 คน โดยทำการวัดคนละ 5 ครั้งทำให้พบว่า ค่าที่วัดได้จากเครื่องมือวัดที่ทำขึ้นนั้นยังมีความคลาดเคลื่อนอยู่ และยังใช้เวลาในการประมวลผลนานกว่าเครื่องมือวัดความดันอัตโนมัติทั่วไป ซึ่งอาจจะเป็นสาเหตุมาจากอุปกรณ์บางตัวอย่างเช่น มอเตอร์หรือวาล์วระบายที่ไม่ได้ออกแบบมาใช้เฉพาะ ทำให้อุปกรณ์ใช้งานได้อย่างไม่เต็มประสิทธิภาพ รวมทั้งอาจจะเกิดจากในส่วนของโปรแกรมที่ใช้สำหรับวัดอาจจะทำได้ยังไม่ดีพอที่จะส่งผลให้ค่าที่ได้จากการวัดมีประสิทธิภาพสูงสุด อย่างไรก็ตาม การทำโครงการในครั้งนี้ ได้รับความรู้เกี่ยวกับการวัดความดันโลหิตในแบบต่างๆ การทำงานของอุปกรณ์อิเล็กทรอนิกส์แต่ละตัว และการทำงานของอุปกรณ์ทางกลที่ควบคุมโดยไมโครวีจีเอ รวมทั้งการเขียนโปรแกรมเพื่อควบคุมการทำงานของอุปกรณ์ต่างๆ ด้วย

5.2 ปัญหาและอุปสรรค์

1. วิธีการตรวจจับหาความดันตัวบนและตัวล่างมีไม่มีความเร็วมากพอ ทำให้ใช้เวลาในการตรวจจับนาน ส่งผลให้ผู้ถูกวัดเกิดความไม่สบายตัว และหากปล่อยลมออกจากสายรัดแขนเร็วเกินไปจะทำให้การวัดนั้นแกว่งมากขึ้น
2. การปล่อยลมของวาล์วควบคุมการไหล มีผลอย่างมากในการตรวจจับหาความดันตัวบนและล่างเพราะว่า ในการเขียนโปรแกรมจะใส่อัตราการไหลไว้คงที่ หากมีการหมุนหรือปรับเปลี่ยนอัตราการไหลจากวาล์ว จะทำให้การวัดผิดพลาดอย่างมาก
3. การรัดแขนจากสายรัดแขนมีผลต่อค่าความดันตัวบนและตัวล่างที่ได้ จากที่ได้ทดลอง เมื่อรัดสายรัดแขนแน่น จะทำให้ค่าที่วัดได้ไม่ตรงและผิดพลาดอย่างมาก ดังนั้นจึงต้องรัดแขนให้มีช่องว่างพอที่จะสอดนิ้วเข้าไปได้ การวัดถึงจะแสดงผลได้ดี
4. อัตราการลดลงของความดันไม่คงที่ สังเกตจากเมื่อปั๊มลมไปถึง 140 mmHg อัตราการลดลงจะเป็นไปตามที่ปรับไว้ แต่เมื่อความดันลดต่ำลง อัตราการลดลงของความดันก็ต่ำลงด้วย ทำให้การเขียนโปรแกรมต้องปรับเปลี่ยน
5. ความดันตัวล่างที่มีค่าสูงผิดปกติ จึงต้องทำการลดค่าลงด้วยโปรแกรมเองถึงจะแสดงผลได้ใกล้เคียงเครื่องวัดอัตโนมัติอื่นๆ

5.3 ข้อเสนอแนะ

การออกแบบเครื่องมือวัดความดันโลหิตอัตโนมัติโดยใช้วิธีตรวจจับแรงดัน นั้นจะต้องใช้การเขียนโปรแกรม อุปกรณ์ช่วยในการดำเนินขั้นตอนการวัดที่เหมาะสม ถึงจะได้ค่าความดันที่ถูกต้อง แต่อย่างไรก็ตาม การวัดความดันในแบบตรวจจับแรงดันนั้นก็จะตรวจจับความดันได้สูงกว่าวิธีฟังเสียงเพราะการสั่นสะเทือนอันเนื่องมาจากการไหลของเลือด เกิดขึ้นก่อนเสียงที่มนุษย์จะได้ยิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Freescale semiconductor. 2010. **MPX5050 pressure sensor datasheet.** [Online]. Available : http://www.freescale.com/webapp/.../prod_summary.jsp?code=MPXx5050.

University of Ljubljana. 2006. **OSCILLOMETRIC VIRTUAL INSTRUMENT FOR BLOOD PRESSURE MEASUREMENT.** [Online]. Available : sim-bp.googlecode.com/.../PWC-2006-TC13-020u.pdf.

Wikipedia The Free Encyclopedia. **Blood pressure article.** [Online]. Available : http://en.wikipedia.org/wiki/Blood_pressure.

Smiths Medical PM, Inc. 2005. **Oscillometric, Linear Bleed Non-Invasive Blood Pressure Technology.** [Online] Available : www.smiths-medical.com/.../Smiths%20Oscillometric%...

Thaimicrotron.com. **การเชื่อมต่ออุปกรณ์แบบ I2C.** [Online]. Available : http://www.thaimicrotron.com/CCS-628/Referrence/I2CBUS.htm.

Wikipedia The Free Encyclopedia. **Piezoresistive effect.** [Online]. Available : http://en.wikipedia.org/wiki/Piezoresistive_effect.

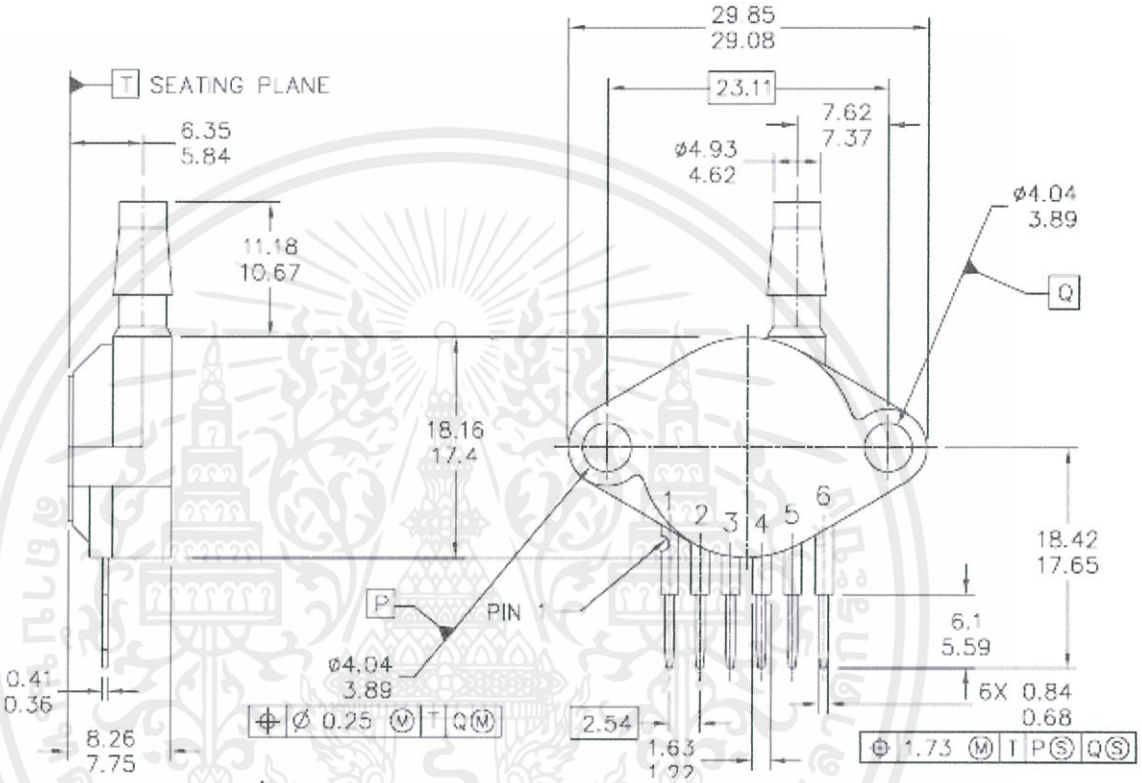
4D Systems company. **4DGL Codebase Global Bytes Exchange.** [Online] Available : http://www.4dsystems.com.au/code/index.php?node=78&id=88.

บริษัทซีเอ็ดยูเคชั่น จำกัด (มหาชน). **หลักการทำงานของโซลินอยด์.** [Online] Available : http://electronics.se-ed.com/contents/046s061/046s061_p02.asp.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
รายละเอียดเพิ่มเติมเกี่ยวกับอุปกรณ์

1. เซนเซอร์ความดัน MPX5050



รูปที่ ก.1 แสดงโครงสร้างของเซนเซอร์ความดัน PMX5050

1.1 ขาการเชื่อมต่อของเซนเซอร์ความดัน MPX5050

ขาที่ 1(PIN1) : V_{out}

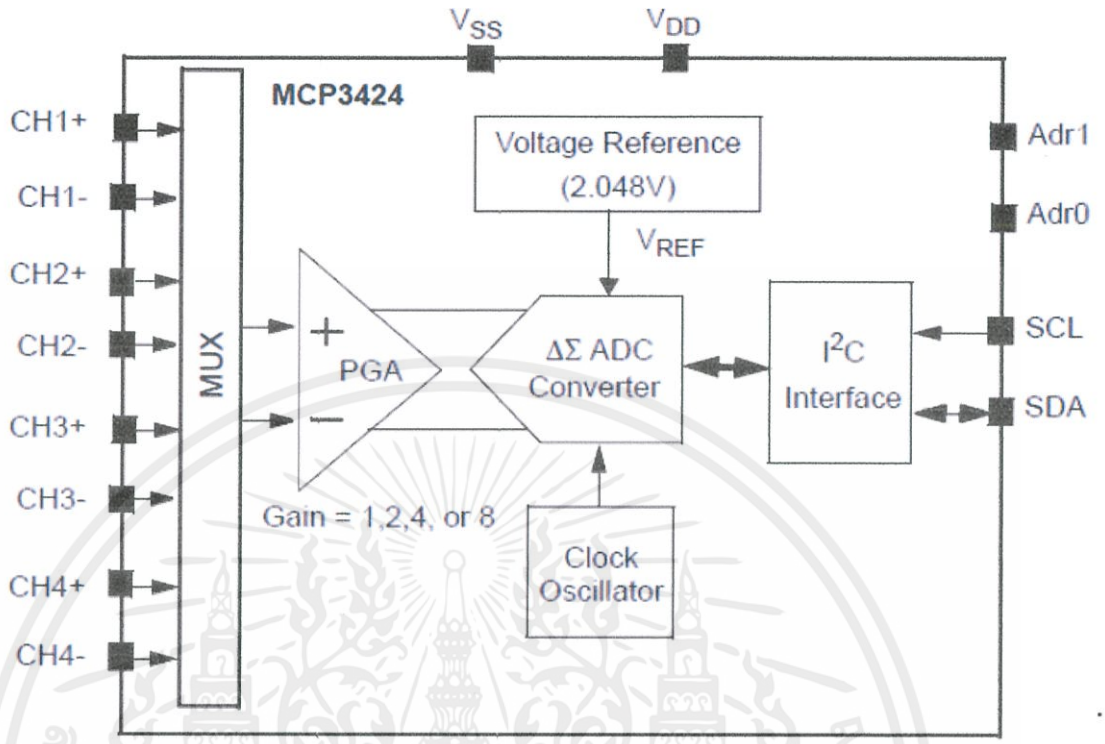
ขาที่ 2(PIN2) : ต่อลงดิน

ขาที่ 3(PIN3) : V_{cc}

ขาที่ 4-6(PIN4-6) : ไม่ต้องเชื่อมต่อ(สำหรับunibody)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. อุปกรณ์แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล MCP3424



รูปที่ ก.2 แผนภาพบล็อกการทำงานของ MCP3424

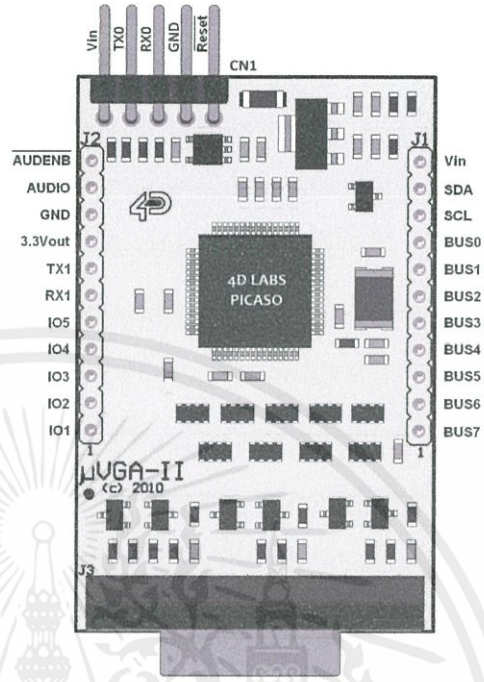
3. รีจิตเตอร์เก็บข้อมูลเวลาของ DS1307

ตารางที่ ก.1 แสดงรีจิสเตอร์เก็บข้อมูลเวลาของ DS1307 เพื่อเรียกใช้ในการแสดงวันและเวลา

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00H	CH	10 Seconds			Seconds				Seconds	00–59
01H	0	10 Minutes			Minutes				Minutes	00–59
02H	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03H	0	0	0	0	0	DAY			Day	01–07
04H	0	0	10 Date		Date				Date	01–31
05H	0	0	0	10 Month	Month				Month	01–12
06H	10 Year				Year				Year	00–99
07H	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08H-3FH									RAM 56 x 8	00H–FFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ไมโครวีจีเอทู



รูปที่ ก.3 แสดงขาการเชื่อมต่อของอุปกรณ์ไมโครวีจีเอทู

ตารางที่ ก.2 แสดงขาการเชื่อมต่อไมโครยูเอสบี

ขาที่	สัญลักษณ์	อินพุต/เอาท์พุต	คำอธิบาย
1	V_{in}	P	แหล่งจ่ายพลังงานไฟฟ้าหลัก มีการป้องการกลับของขั้ว พิสัยของแรงดันไฟฟ้าอยู่ที่ 4.0 ถึง 5.5 โวลต์ โดยปกติที่ 5.0 โวลต์
2	TX0	O	ขาส่งข้อมูลอนุกรมแบบไม่เป็นจังหวะ เชื่อมต่อขานี้เข้ากับคอนโทรลเลอร์ภายนอก คอนโทรลเลอร์จะได้รับข้อมูลจากไมโครวีจีเอทูผ่านขาเชื่อมต่อนี้ ขาการเชื่อมต่อนี้สามารถทนแรงดันไฟฟ้าได้ถึง 5.0 โวลต์
3	RX0	I	ขารับข้อมูลอนุกรมแบบไม่เป็นจังหวะ เชื่อมต่อขานี้เข้ากับคอนโทรลเลอร์ภายนอก คอนโทรลเลอร์ภายนอกจะสามารถส่งข้อมูลสู่ไมโครวีจีเอทูได้ผ่านขาการเชื่อมต่อนี้ ขาการเชื่อมต่อนี้สามารถทนแรงดันไฟฟ้าได้ถึง 5.0 โวลต์
4	GND	P	ขากราวด์ของแหล่งจ่ายไฟ
5	Reset	I	มาสเตอร์รีเซตสัญญาณ ภายในจะถูกดึงแรงดันไฟฟ้าขึ้นเป็น3.3โวลต์ผ่านตัวต้านทาน 4.7k โอห์ม การที่พัลส์ต่ำเกิดขึ้นเดิน 2 ไมโครวินาที จะทำให้เกิดการรีเซต ถ้าโมดูลต้องการจะรีเซตจากภายนอก สามารถทำได้โดยใช้วงจรคอลเล็คเตอร์แบบเปิด ขาการเชื่อมต่อนี้ไม่ได้ถูกขับให้ต่ำจากภายในดังนั้นโฮสต์ควรควบคุมขาการเชื่อมต่อนี้ผ่านพอร์ตของตัวมันเองโดยใช้การจัดการ คอลเล็คเตอร์เปิดหรือเดรน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ก.3 แสดงขาเชื่อมต่อเอาท์พุตด้าน J1 (ขาเชื่อมต่อเสริม)

ขาที่	สัญลักษณ์	อินพุต/เอาท์พุต	คำอธิบาย
1	IO1	I/O	ขาอินพุตหรือเอาท์พุตที่ 1 ขาการเชื่อมต่อนี้สามารถทนแรงดันได้ถึง 5.0 โวลต์
2	IO2	I/O	ขาอินพุตหรือเอาท์พุตที่ 2 ขาการเชื่อมต่อนี้สามารถทนแรงดันได้ถึง 5.0 โวลต์
3	IO3	I/O	ขาอินพุตหรือเอาท์พุตที่ 3 ขาการเชื่อมต่อนี้สามารถทนแรงดันได้ถึง 5.0 โวลต์
4	IO4/BUS_RD	I/O	ขาอินพุตเอาท์พุตที่ 4 หรือบัสการอ่าน ขาการเชื่อมต่อนี้สามารถทนแรงดันได้ถึง 5.0 โวลต์
5	IO5/BUS_WR	I/O	ขาอินพุตเอาท์พุตที่ 5 หรือบัสการเขียน ขาการเชื่อมต่อนี้สามารถทนแรงดันได้ถึง 5.0 โวลต์
6	RX1	I	พอร์ตอนุกรมรับข้อมูลแบบไม่เป็นจังหวะที่ 1 COM1
7	TX1	O	พอร์ตอนุกรมส่งข้อมูลแบบไม่เป็นจังหวะที่ 1 COM1
8	3.3Vout	P	เอาท์พุต 3.3 โวลต์ จ่ายกระแสไฟฟ้า 400mA
9	GND	P	กราวด์แหล่งจ่าย
10	AUDIO	O	PWM เอาท์พุตเสียง
11	AUDENB	O	ชาเอาท์พุตเสียง

ตารางที่ ก.4 แสดงขาเชื่อมต่อเอาท์พุตด้าน J2 (ขาเชื่อมต่อเสริม)

ขาที่	สัญลักษณ์	อินพุต/เอาท์พุต	คำอธิบาย
1	BUS7	I/O	อินพุตเอาท์พุตบัส (บัส0..7) บิตที่ 7
2	BUS6	I/O	อินพุตเอาท์พุตบัส (บัส0..7) บิตที่ 6
3	BUS5	I/O	อินพุตเอาท์พุตบัส (บัส0..7) บิตที่ 5
4	BUS4	I/O	อินพุตเอาท์พุตบัส (บัส0..7) บิตที่ 4
5	BUS3	I/O	อินพุตเอาท์พุตบัส (บัส0..7) บิตที่ 3
6	BUS2	I/O	อินพุตเอาท์พุตบัส (บัส0..7) บิตที่ 2
7	BUS1	I/O	อินพุตเอาท์พุตบัส (บัส0..7) บิตที่ 1
8	BUS0	I/O	อินพุตเอาท์พุตบัส (บัส0..7) บิตที่ 0
9	SCL	O	I2C ขาคล็อกเอาท์พุต
10	SDA	I/O	I2C ขาข้อมูลแบบสองทิศทาง
11	Vin	P	ขาเชื่อมต่อแหล่งจ่ายหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. อุปกรณ์แสดงวันและเวลา DS1307

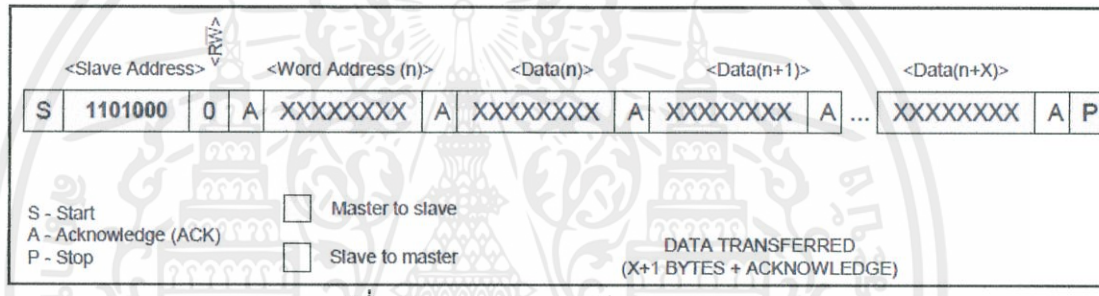
ตารางที่ ก.5 แสดงขาการเชื่อมต่อและฟังก์ชันการทำงาน

ขาเชื่อมต่อ	ชื่อ	ฟังก์ชันการทำงาน
1	X1	เชื่อมต่อกับผลึกคริสตัล 32.768กิโลเฮิร์ทซ์ วงจรการสั่นภายในออกแบบมาให้ทำงานกับคริสตัลที่มีภาระตัวเก็บประจุ 12.5 พิโคฟารัด X1เป็นอินพุตต่อกับตัวแกว่ง และสามารถที่จะเลือกต่อกับตัวแกว่งภายนอกขาด 32.768 กิโลเฮิร์ทซ์ได้ ไม่ต้องเชื่อมต่อเอาท์พุตของตัวแกว่งภายในX2 ถ้าตัวแกว่งภายนอกได้เชื่อมต่อกับX1
2	X2	
3	V_{BAT}	อินพุตของแหล่งจ่ายไฟฟ้าสำรองสำหรับเซลล์ลิเทียมหรือแหล่งจ่ายพลังงานอื่นๆขนาด 3 โวลต์ ความต่างศักย์ไฟฟ้าของแบตเตอรี่จะต้องอยู่ที่ระหว่างขอบเขตค่าต่ำสุดและสูงสุดเพื่อการทำงานที่เหมาะสม นำไดโอดเชื่อมต่ออนุกรมระหว่างแบตเตอรี่และขาเชื่อมต่อ V_{BAT} อาจจะทำให้การทำงานไม่ถูกต้อง ถ้าไม่ต้องการแหล่งจ่ายไฟฟ้าสำรอง ขาเชื่อมต่อ V_{BAT} จะต้องต่อลงดิน แบตเตอรี่ลิเทียมขนาด 48 มิลลิแอมป์ชั่วโมงหรือสูงกว่าจะสำรองการทำงานของDS1307ได้มากกว่า 10 ปี โดยที่พลังงานไม่ขาดหายที่ +25 องศาเซลเซียส
4	GND	ต่อลงดิน
5	SDA	อินพุตและเอาท์พุตข้อมูลอนุกรม SDAเป็นอินพุตและเอาท์พุตข้อมูลสำหรับอินเตอร์เฟสอนุกรมกับI2C ขาเชื่อมต่อSDAเป็นแบบปล่อยลอยและต้องต่อกับตัวต้านทานภายนอก
6	SCL	อินพุตสัญญาณนาฬิกาอนุกรม SCLเป็นอินพุตสัญญาณนาฬิกาสำหรับอินเตอร์เฟสI2C และใช้สำหรับกำหนดจังหวะการเคลื่อนย้ายข้อมูลบนอินเตอร์เฟสอนุกรม
7	SWQ/OUT	ตัวขับสัญญาณคลื่นสี่เหลี่ยมและเอาท์พุต เมื่อต้องการเปิดใช้งานให้ตั้งค่าบิทSQWEเป็น1 เอาท์พุตของขาเชื่อมต่อSWQ/OUTจะให้ความถี่คลื่นสี่เหลี่ยมได้1ค่าใน4ค่า (1เฮิร์ทซ์, 4กิโลเฮิร์ทซ์, 8เฮิร์ทซ์, 32กิโลเฮิร์ทซ์) ขาเชื่อมต่อSWQ/OUTเป็นแบบปล่อยลอยและต้องต่อกับตัวต้านทานภายนอก SWQ/OUTยังสามารถประยุกต์ใช้งานได้กับทั้ง V_{CC} หรือ V_{BAT}
8	V_{CC}	แหล่งจ่ายพลังงานอันดับแรก เมื่อใช้ศักย์ไฟฟ้าภายในขอบเขตปกติอุปกรณ์จะสามารถเข้าถึงได้ทั้งหมดสามารถเขียนและอ่านข้อมูลได้ เมื่อเชื่อมต่อแหล่งจ่ายไฟฟ้าสำรองเข้ากับอุปกรณ์และ V_{CC} ต่ำกว่า V_{TP} จะไม่สามารถอ่านและเขียนได้ อย่างไรก็ตามแม้ว่าศักย์ไฟฟ้าอินพุตลดลงก็คงยังไม่มีผลต่อฟังก์ชันการเก็บเวลา

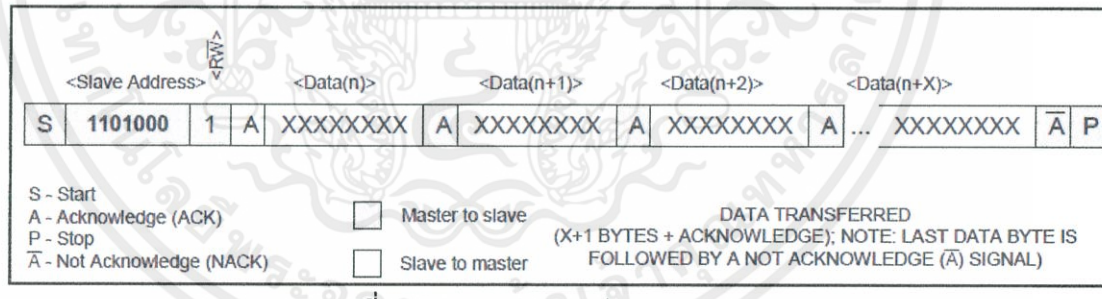
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 รายละเอียดคำอธิบาย

DS1307 เป็นนาฬิกาและปฏิทินที่ใช้พลังงานต่ำพร้อมด้วยแบตเตอร์รี่ต่อSRAMขนาด 56 ไบต์ นาฬิกาและปฏิทินประกอบด้วยข้อมูลวินาที นาที ชั่วโมง วัน วันที่ เดือน และปี เมื่อวันสุดท้ายของเดือนจะถูกปรับเปลี่ยนอัตโนมัติสำหรับเดือนที่มีน้อยกว่า 31 วัน รวมไปถึงมีการตรวจแก้สำหรับปีที่มีจำนวนวันมากกว่าปกติ DS1307ทำงานเป็นอุปกรณ์สเลฟบนบัสI2c สามารถเข้าถึงได้ด้วยเงื่อนไข STARTและโดยมีเงื่อนไขว่ารหัสการระบุอุปกรณ์จะต้องมาจากที่อยู่รีจิสเตอร์ หลังจากที่รีจิสเตอร์สามารถเข้าได้ถึงตามลำดับจนกระทั่งทำเงื่อนไขSTOP เมื่อ V_{CC} น้อยกว่า1.25xV_{BAT} อุปกรณ์จะยุติการเข้าถึงกระบวนการและตัวนับที่อยู่ของอุปกรณ์จะเริ่มทำงานใหม่ อินพุตที่ต่อไปยังอุปกรณ์จะไม่รับรู้ได้ ณเวลานี้เพื่อป้อนกันข้อมูลที่ผิดพลาดจากระบบout-of-tolerance ที่ยังคงเขียนไปยังอุปกรณ์อยู่ เมื่อ V_{CC} ต่ำกว่า V_{BAT} อุปกรณ์จะสลับไปยังสถานะแบตเตอรี่สำรองกระแสต่ำ เมื่อพลังงานเพิ่มขึ้นอุปกรณ์จะสลับจากแบตเตอรี่ไปยัง V_{CC} เมื่อ V_{CC} มีค่ามากกว่า V_{BAT} +0.2โวลต์ และอินพุตจะรับรู้เมื่อ V_{CC} มีค่ามากกว่า1.25xV_{BAT} แผนภาพบล็อกในรูป แสดงส่วนหลักของ RTC อนุกรม



รูปที่ ก.4 ภาพแสดงการเขียนของ DS1307



รูปที่ ก.5 ภาพแสดงการอ่านของ DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
การเขียนโปรแกรม

```
#platform "uVGA-II_GFX2"
#inherit "4DGL_16bitColours.fnc"
#constant X_ref          170
#constant Y_ref          65
#constant X_chart        20
#constant Y_chart        65
#constant SECTORSIZE     512
#constant CHANNEL1       0x00
#constant CHANNEL2       0x20
#constant CHANNEL3       0x40
#constant CHANNEL4       0x60
#constant MPC3424        0xD8

#constant SECONDS        0
#constant MINUTES        1
#constant HOURS          2
#constant DAYS           3
#constant DATES          4
#constant MONTHS         5
#constant YEARS          6
#constant DS1307         0xD0
#constant WR             1

var Yarray1[400];
var Yarray2[400];
var img;
var sectorbuf[SECTORSIZE/2];
var Graph_pres;
var Graph_lst;
var Graph_X;
var Graph_Y;

var buf[100]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var n;
var AnalogVal;
var seconds;
var address;
var r;
var HBP;
var sys;
var dias;

func CreateFrame()
    gfx_Rectangle(1,1,620,450,WHITE);
    gfx_RectangleFilled(1,1,620,15,WHITE);
    gfx_Rectangle(1,50,620,50,WHITE);
    txt_MoveCursor(2,1);
    txt_Set(TEXT_COLOUR,ORANGE);
    txt_Set(TEXT_HEIGHT,2);
    print("              ABPM Instrumentation Engineering KMITL");
endfunc

func CreateFrontPage()
    if(!media_Init())                    // initialize uSD card
      print("media_Init failed\n") ;
      repeat forever
    endif

    print("media_Init sucessful") ;
    media_SetSector(0, 0);               // set stream sector address
    if(!media_RdSector(sectorbuf))
       print("media_RdSector failed\n");
       repeat forever
    endif
    gfx_RectangleFilled(1,1,620,450,WHITE);
    while(!media_Init()); media_SetSector(0x0000, 0x0E79);
    media_Image(130,40); //KMITL LOGO
    pause(2000);
    gfx_Cls();
endfunc
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
func CreateGraphFrame()
var X_offset, Y_offset, l;
    gfx_RectangleFilled(X_ref,Y_ref,X_ref+400,Y_ref+3,GREEN);
    gfx_Rectangle(X_ref,Y_ref,X_ref+400,Y_ref+200,GREEN);
        for (l:= 1; l<10; l++)
            X_offset := l*40;
            gfx_Vline(X_ref+X_offset,Y_ref,Y_ref+200,GREEN);
        next
        for (l := 1; l<5; l++)
            Y_offset := l*40;
            gfx_Hline(Y_ref+Y_offset,X_ref,X_ref+400,GREEN);
        next
        txt_Set(TEXT_HEIGHT,1);
        txt_MoveCursor(6,19);
        txt_Set(TEXT_COLOUR,WHITE);
        txt_Set(TEXT_HEIGHT,1);
        print("250");
        txt_MoveCursor(9,19);
        txt_Set(TEXT_COLOUR,WHITE);
        txt_Set(TEXT_HEIGHT,1);
        print("200");
        txt_MoveCursor(12,19);
        txt_Set(TEXT_COLOUR,WHITE);
        txt_Set(TEXT_HEIGHT,1);
        print("150");
        txt_MoveCursor(15,19);
        txt_Set(TEXT_COLOUR,WHITE);
        txt_Set(TEXT_HEIGHT,1);
        print("100");
        txt_MoveCursor(18,19);
        txt_Set(TEXT_COLOUR,WHITE);
        txt_Set(TEXT_HEIGHT,1);
        print("50");
        txt_MoveCursor(22,19);
        txt_Set(TEXT_COLOUR,WHITE);
        txt_Set(TEXT_HEIGHT,1);
        print("0");
        txt_Set(TEXT_HEIGHT,2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
endfunc

func ReadRTC(var addr)
    I2C_Start();                    // Generate Start condition
    I2C_Write(DS1307);              // send slave address
    I2C_Write(LObyte(addr));        // Send register address
    I2C_Restart();                  // Generate Restart
    I2C_Write(DS1307+WR);           // send control byte for Read
    r := I2C_Read();                // read the byte
    I2C_Nack();                     // finished reading, send Not Ack
    I2C_Stop();                     // Send Stop Condition
    return r;                       // return the register value
endfunc

// write a single byte to the required register
func WriteRTC(var register, var value)
    I2C_Start();                    // Generate Start condition
    I2C_Write(DS1307);              // send slave address
    I2C_Write(register);            // select the register
    I2C_Write(value);               // save the value in selected register
    I2C_Stop();                     // finished with bus
endfunc

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

func WriteConfigADC(var Channel)
    I2C_Start();                    // Generate Start condition
    I2C_Write(MPC3424);             // send slave address
    I2C_Write(Channel|0x90);        // 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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    nd2 := I2C_Read();                  // read the 2nd byte
    I2C_Ack();
    rd3 := I2C_Read();                  // read the 3nd byte
    I2C_Nack();                         // finished reading, send Not Ack
    I2C_Stop();                         // Send Stop Condition
    AnalogVal := (nd2*256)+rd3;
    return AnalogVal;                   // return the register value
endfunc

func I2C_Initialize()
var c;
  I2C_Open(I2C_SLOW);                   // 100khz
    c:=ReadRTC(0) & 0x7F;               // ensure CH bit is clear else clock wont run
    WriteRTC(0, c);
    WriteRTC(7, 0x10);                  // enable square wave output
    pause(1000);
endfunc

func DisplayClock()
      txt_MoveCursor(2,2);
      txt_Set(TEXT_COLOUR,WHITE);
      txt_Set(TEXT_HEIGHT,2);
      print(
          [HEX2] ReadRTC(HOURS),":",
          [HEX2] ReadRTC(MINUTES),":",
          [HEX2] ReadRTC(SECONDS)
         );
      txt_Height(1);
      txt_Width(1);
endfunc

func DisplayDate()
var day,month;
      txt_MoveCursor(2,64);
      txt_Set(TEXT_COLOUR,WHITE);
      txt_Set(TEXT_HEIGHT,2);
      day := ReadRTC(DAYS);
      if (day == 1) print ("SUN ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (day == 2) print ("MON ");
        if (day == 3) print ("TUE ");
        if (day == 4) print ("WED ");
        if (day == 5) print ("THU ");
        if (day == 6) print ("FRI ");
        if (day == 7) print ("SAT ");
        month := ReadRTC(MONTHS);
        if (month == 1) print ("JAN ");
        if (month == 2) print ("FEB ");
        if (month == 3) print ("MAR ");
        if (month == 4) print ("APR ");
        if (month == 5) print ("MAY ");
        if (month == 6) print ("JUN ");
        if (month == 7) print ("JUL ");
        if (month == 8) print ("AUG ");
        if (month == 9) print ("SEP ");
        if (month == 16) print ("OCT ");
        if (month == 17) print ("NOV ");
        if (month == 18) print ("DEC ");
        print([HEX2] ReadRTC(DATES),",",
            [HEX2] ReadRTC(YEARS)
            );
        txt_Height(1);
        txt_Width(1);
endfunc

func DisplaySYS(var HiBP)
    txt_MoveCursor(27,2);
    txt_Height(10);
    txt_Width(7);
    txt_Set(TEXT_COLOUR,WHITE);
    HiBP := ABS(HiBP);
    if (HiBP < 10)
        print ("  ",HiBP);
    else
        if (HiBP < 100)
            print (" ",HiBP);
        else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            print(HiBP);
        endif
    endif
    txt_Height(1);
    txt_Width(1);
endfunc

func PlotGraph1 (var HBP)
    HBP := 195-HBP;
    Graph_pres := HBP;
    if (HBP > 199) HBP := 195;
    Graph_X := Graph_X + 1;
    if (Graph_X > 395) Graph_X := 5;
    gfx_Vline(X_ref+Graph_X,Y_ref+5,Y_ref+198,BLACK);
    gfx_CircleFilled(170+Graph_X,0+1+HBP,1,YELLOW);
endfunc

func main()
    var HBP, LBP, HR, N;
    var state;
    CreateFrontPage();
    CreateFrame();
    CreateGraphFrame();
    I2C_Initialize();
    WriteConfigADC(CHANNEL1);
    Graph_X := 5; //Start
    txt_MoveCursor(12,1);
    print ("---------status----------Result----------pressure-----");
  repeat
  start:
 PlotGraph1 (HBP);
 DisplayClock();
  DisplayDate();
 txt_MoveCursor(27,60);
  ReadADC();
  HBP := (AnalogVal*18)/10;

   pin_Set(INPUT,IO1_PIN);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pin_Set(OUTPUT,IO2_PIN);
pin_Set(OUTPUT,IO3_PIN);
print(HBP);
print("\n");
if(HBP<20)
inflatemode(HBP);

endif
 if(HBP>=140)
      deflatemode(HBP);
      state:=1;
      endif
if(state==1)
          if(HBP<140)
          calculatemode(HBP);
         goto cal;
         endif
endif

forever
cal:
  calsys();
   sys:=calsys();
  callcaldias();
  dias:=caldias();
repeat
ReadADC();
HBP := (AnalogVal*18)/10;
txt_MoveCursor(27,40);// can stop value from adc
print("systolic\n");
print (sys);
print ("\n");

print("diastolic\n");
print(dias-10);
pin_LO(IO3_PIN);
pause(10000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    SystemReset();
    forever

endfunc

func inflatemode(var HBP)
txt_MoveCursor(27,10);
pin_HI(IO2_PIN);
pin_HI(IO3_PIN);
print("inflating");
endfunc


func deflatemode(var HBP)
txt_MoveCursor(27,10);
pin_HI(IO3_PIN);
pin_LO(IO2_PIN);
print("\n");
print("deflating");
endfunc

func calculatemode(var HBP)
txt_MoveCursor(29,10);
print("calculating");
endfunc

func calsys()
var x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5;
var sys1,sys2,sys3,sys4,sys5;
var private sys;
var state;
pause(1000);
ReadADC();
x1:=(AnalogVal*18)/10;
ReadADC();
sys1:=(AnalogVal*18)/10;
pause(1000);
ReadADC();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
y1:=(AnalogVal*18)/10;
z1:=x1-y1;
if(z1<=3)
sys:=sys1;
else

    pause(1000);
    ReadADC();
    x2:=(AnalogVal*18)/10;
    ReadADC();
    sys2:=(AnalogVal*18)/10;
    pause(1000);
    ReadADC();
    y2:=(AnalogVal*18)/10;
    z2:=x2-y2;
    if(z2<=3)
        sys:=sys2;

        else

        pause(1000);
        ReadADC();
        x3:=(AnalogVal*18)/10;
        ReadADC();
        sys3:=(AnalogVal*18)/10;
        pause(1000);
        ReadADC();
        y3:=(AnalogVal*18)/10;
        z3:=x3-y3;
        if(z3<=3)
            sys:=sys3;
            else

            pause(1000);
            ReadADC();
            x4:=(AnalogVal*18)/10;
            ReadADC();
            sys4:=(AnalogVal*18)/10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            pause(1000);
            ReadADC();
            y4:=(AnalogVal*18)/10;
            z4:=x4-y4;
               if(z4<=3)
               sys:=sys4;
               else

               pause(1000);
               ReadADC();
               x5:=(AnalogVal*18)/10;
               ReadADC();
               sys5:=(AnalogVal*18)/10;
               pause(1000);
               ReadADC();
               y5:=(AnalogVal*18)/10;
               z5:=x5-y5;
                  if(z5<=3)
                  sys:=sys5;
                  endif

               endif
         endif
      endif
endif
return sys;
endfunc

func caldias()
var x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,x5,y5,z5;
var dias1,dias2,dias3,dias4,dias5;
var private dias;
pause(1000);
ReadADC();
x1:=(AnalogVal*18)/10;
ReadADC();
dias1:=(AnalogVal*18)/10;
pause(1000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ReadADC();
y1:=(AnalogVal*18)/10;
z1:=x1-y1;
if(z1>=2)
dias:=dias1;
else
    pause(1000);
   ReadADC();
   x2:=(AnalogVal*18)/10;
   ReadADC();
   dias2:=(AnalogVal*18)/10;
   pause(1000);
   ReadADC();
   y2:=(AnalogVal*18)/10;
   z2:=x2-y2;
   if(z2>=2)
   dias:=dias2;
   else
        pause(1000);
        ReadADC();
        x3:=(AnalogVal*18)/10;
        ReadADC();
        dias3:=(AnalogVal*18)/10;
        pause(1000);
        ReadADC();
        y3:=(AnalogVal*18)/10;
        z3:=x3-y3;
        if(z3>=2)
        dias:=dias3;
        else

             pause(1000);
             ReadADC();
             x4:=(AnalogVal*18)/10;
             ReadADC();
             dias4:=(AnalogVal*18)/10;
             pause(1000);
             ReadADC();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                y4:=(AnalogVal*18)/10;
                z4:=x4-y4;
                if(z4>=2)
                dias:=dias4;
                else
                    pause(1000);
                    ReadADC();
                    x5:=(AnalogVal*18)/10;
                    ReadADC();
                    dias5:=(AnalogVal*18)/10;
                    pause(1000);
                    ReadADC();
                    y5:=(AnalogVal*18)/10;
                    z5:=x5-y5;
                    if(z5>=2)
                    dias:=dias5;
                    else
                    endif
                endif
            endif
        endif
    endif

return dias;
endfunc

func displaysys(var sys)
txt_MoveCursor(27,20);
print(sys);
endfunc

func callcaldias()
var H;
    repeat
    ReadADC();
    H := (AnalogVal*18)/10;
    pause(500);
    until(H<=80);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    caldias();
endfunc
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้