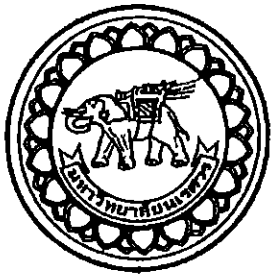


กล่องยาอิเล็กทรอนิกส์

ELECTRONIC PILL DISPENSER

นายกวิตม์ วรางกูร รหัส 51364224

นางสาวธนาภรณ์ การะเกตุ รหัส 51364330

นางสาวธีร์วรา ดวงพระจันทร์ รหัส 51364361

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ - 4 S.A. 2555
เลขทะเบียน 16073979
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร ว323ก 2554

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2554

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน กล่องยาอิเล็กทรอนิกส์

ผู้ดำเนินโครงงาน นายกวิตม์ วรางกูร รหัส 51364224

นางสาวธนาภรณ์ การะเกตุ รหัส 51364330

นางสาวธีร์วรา ควงพระจันทร์ รหัส 51364361

ที่ปรึกษาโครงงาน ดร. พรพิศุทธิ์ วรจิรันตน์

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2554

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน

(ดร. พรพิศุทธิ์ วรจิรันตน์)

.....กรรมการ

(ดร. พิสุทธิ์ อภิชยกุล)

.....กรรมการ

(ดร. พนัส นัถฤทธิ์)

ชื่อหัวข้อโครงงาน	กล่องยาอิเล็กทรอนิกส์
ผู้ดำเนินโครงงาน	นายกวิตม์ วรางกูร รหัส 51364224
	นางสาวธนาภรณ์ การะเกตุ รหัส 51364330
	นางสาวธีร์วรา ดวงพระจันทร์ รหัส 51364361
ที่ปรึกษาโครงงาน	ดร. พรพิศุทธิ์ วรจิรันตน์
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2554

.....

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอโครงงานเกี่ยวกับการนำบอร์ดไมโครคอนโทรลเลอร์มาใช้สำหรับควบคุมการทำงานของกล่องยาเตือนเวลารับประทานยา มีแนวคิดที่จะแก้ปัญหาเรื่องการรับประทานยาที่ขาดประสิทธิภาพ คือ การลืมรับประทานยา การรับประทานยาไม่ตรงตามเวลาที่แพทย์กำหนด จึงนำมาซึ่งการพัฒนาตัวอุปกรณ์ซึ่งมีสองส่วนคือการนำบอร์ดไมโครคอนโทรลเลอร์ควบคุมการทำงานของตัวอุปกรณ์โดยการทำงานร่วมกับการเขียนโปรแกรมด้วยภาษาซี อีกส่วนหนึ่งคือการตั้งเวลาเตือนการรับประทานยาผ่านหน้าจอคอมพิวเตอร์ที่เขียนด้วยโปรแกรมวิชวลเบสิก

โดยในโครงงานนี้ได้สร้างแบบจำลองกล่องยาเพื่อสามารถจะนำไปพัฒนาใช้งานได้จริง โดยใช้บอร์ดไมโครคอนโทรลเลอร์ มาเชื่อมต่อเข้ากับตัวกล่องยาที่มีฐานรองรับ บอร์ดไมโครคอนโทรลเลอร์จะควบคุมการแสดงหลอดไฟแอลอีดีและลำโพง เมื่อมีการเตือนดังที่โปรแกรมกำหนดแล้ว จะทำการบันทึกสถานะการรับประทานยาของผู้ป่วยลงบนไมโครเอสดีการ์ด

Project title	Electronic Pill Dispenser	
Name	Mr. Kawit Warangkun	ID. 51364224
	Miss. Tanaporn Karaket	ID. 51364330
	Miss. Thiwara Duangphrachan	ID. 51364361
Project advisor	Miss. Ponpisut Worrajiran, Ph.D.	
Major	Electrical Engineering	
Department	Electrical and Computer Engineering	
Academic year	2011	

Abstract

Electronic pill dispenser was developed using a microcontroller board and a clear plastic box equip with LEDs and buzzer.It can be used to remind a patient to take his/her medicines, as prescribed.The microcontroller program was written in C language and a graphic user interface(GUI) was developed with Visual Basic(VB).

In this project a prototype was made and tested.The audio and visual alarms can be set by a medicine prescriber from the computer screen via the GUI.The medication status was recorded in a SD card and could be loaded to examine afterward.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาเป็นอย่างยิ่งจาก ดร. พรพิศุทธิ์ วรจิรันตน์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานและให้ความกรุณาในการตรวจทานปริญญานิพนธ์ คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้กับคณะผู้ดำเนินงาน นอกจากนี้ยังต้องขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ให้ยืมอุปกรณ์ และเครื่องมือวัดมาใช้งาน รวมทั้งห้องที่ใช้สำหรับการทดลองโครงงาน จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้

เหนือสิ่งอื่นใด คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณบิดา มารดา ผู้มอบความรัก ความเมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจทำให้ได้รับความสำเร็จอย่างทุกวันนี้ และขอขอบคุณทุก ๆ คนในครอบครัวของคณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวไว้ ณ ที่นี้ด้วย

นายกวิตม์ วรางกูร
นางสาวธนาภรณ์ การะเกตุ
นางสาวธีร์วรา ดวงพระจันทร์

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ฌ
สารบัญรูป ..... ญ

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 2
1.3 ขอบเขตของโครงงาน 2
1.4 ขั้นตอนและแผนการดำเนินงาน 3
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 3
1.6 งบประมาณ 4

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 5
2.1 คุณสมบัติและลักษณะของกล่องบรรจุยา 5
2.2 โครงสร้างบอร์ด ET-BASE MEGA2560 8
2.3 การ์ดหน่วยความจำ SD-Card แบบ Micro-SD 10
2.4 วงจรแหล่งจ่ายไฟ 11
2.5โปรแกรม Visual Basic (VB) 12
2.6 เครื่องโปรแกรม ET-AVRISP mkII 12
2.7 การใช้งาน RTC (Real Time Clock) ด้วย DS 1307 13
2.8 AVR Studio4 16
2.9 WinAVR 16

สารบัญ (ต่อ)

หน้า

2.10 Serial Peripheral Interface, SPI 17

บทที่ 3 ขั้นตอนการเขียนโปรแกรมและการออกแบบชิ้นงาน 18
3.1 การทำงานของอุปกรณ์ที่ใช้ควบคุม 19
3.1.1 การใช้งานลิมิตสวิตช์ 19
3.1.2 การใช้งานวงจร Buzzer 20
3.1.3 การใช้งานวงจรขับ LED แสดงผล 21
3.2 หลักการทำงานของระบบ 22
3.2.1 รูปแบบการส่งข้อมูล 22
3.2.2 รูปแบบการรับส่งข้อมูลแบบ HDLC Protocol 9 byte 23
3.2.3 ส่งค่า Setting จาก PC ไปยังบอร์ดไมโครคอนโทรลเลอร์ AVR 29
3.2.4 บอร์ดไมโครคอนโทรลเลอร์ AVR ตอบกลับไปยัง PC 30
3.2.5 การส่งค่าตั้งเวลาเตือนรับประทานยาและเวลาเตือนรับประทานอาหาร 31
3.2.6 การบันทึกค่าและการอ่านค่าจากหน่วยความจำ SD-Card 32
3.2.7 การบันทึกข้อมูลลง SD-Card เป็นรูปแบบ Text file 33
3.2.8 การนำข้อมูลมาแสดงผล 34
3.2.9 การลบข้อมูลใน SD-Card 36
3.2.10 ฐานข้อมูลผู้ป่วย 39
3.2.10.1 ตารางฐานข้อมูลการตั้งค่าเวลารับประทานยา 40
3.2.10.2 ตารางฐานข้อมูลการตั้งค่าเวลารับประทานอาหาร 41
3.2.10.3 ตารางฐานข้อมูลเวลารับประทานยา 42
3.2.10.4 ตารางฐานข้อมูลการตั้งชื่อผู้ป่วย 42
3.2.10.5 ตารางฐานข้อมูลการตั้งค่าการติดต่อสื่อสาร 43
3.3 การออกแบบ 43
3.3.1 การออกแบบฮาร์ดแวร์อุปกรณ์กล่องยา 43
3.3.2 การติดตั้งอุปกรณ์บอร์ดไมโครคอนโทรลเลอร์ 46
3.3.3 การติดตั้งอุปกรณ์ลิมิตสวิตช์ 47

สารบัญ (ต่อ)

หน้า

3.3.4 การติดตั้ง ET-MINI DS1307 47
3.3.5 การติดตั้งหลอดไฟแอลอีดี 47
3.3.6 การติดตั้งลำโพง 47
3.4 การทำงานของไมโครคอนโทรลเลอร์ 48
3.4.1 การรับส่งข้อมูลอนุกรมของไมโครคอนโทรลเลอร์ AVR 48
3.4.2 การทำงานของไมโครคอนโทรลเลอร์เพื่อส่งสัญญาณเตือนและบันทึกผล 48
3.5 ส่วนควบคุมติดต่อกับผู้ใช้งาน 50
3.5.1 การบันทึก Text file และการแสดงผลบน PC 50
3.5.2 หน้าต่างการตั้งค่าโปรแกรมวิชวลเบสิก 50
3.5.2.1 หน้าต่างตั้งค่าการติดต่อสื่อสาร 51
3.5.2.2 หน้าต่างตั้งค่าวันที่และเวลา 52
3.5.2.3 หน้าต่างตั้งค่าเวลาเตือนรับประทานยาและเวลาเตือนรับประทานอาหาร 52
3.5.2.4 หน้าต่างตั้งค่าข้อมูลผู้ป่วย 56
3.5.2.5 หน้าต่างแสดงผล 57
บทที่ 4 ผลการทดสอบ 60
4.1 ส่วนประกอบและขั้นตอนการทำงานของระบบ 60
4.2 วิธีการใช้งาน 61
4.2.1 ขั้นตอนการตั้งค่าหน้าต่างกำหนดชื่อ นามสกุล และรายละเอียดผู้ป่วย 62
4.2.2 ขั้นตอนการเชื่อมต่อการติดต่อสื่อสารระหว่างพอร์ตกับหน้าต่างวิชวลเบสิก 63
4.2.3 ขั้นตอนการตั้งค่าวันที่และเวลา 64
4.2.4 ขั้นตอนการตั้งค่าเวลาการรับประทานยา 65
4.2.5 ขั้นตอนการแสดงผล 68
4.3 ผลการทดสอบการทำงานสัญญาณเตือนการรับประทานอาหาร 70
4.4 ผลการทดสอบการทำงานสัญญาณเตือนการรับประทานยา 70
4.4.1 ผลการทดสอบกรณีที่มีการหยิบยารับประทานในช่วงเวลาที่สัญญาณเตือน 70
4.4.2 ผลการทดสอบกรณีที่เมื่อหยิบยาแล้วลืมปิดฝากล่องยา 72
4.4.3 ผลการทดสอบกรณีที่ไม่มีการหยิบยาในช่วงเวลาที่กำหนด 72

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลและข้อเสนอแนะ 74
5.1 สรุปผลการดำเนินโครงงาน 74
5.2 ปัญหาและแนวทางแก้ไข 74
5.3 แนวทางในการพัฒนาต่อไป 75

เอกสารอ้างอิง 76

ภาคผนวก ก คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASE MEGA 2560 78
ภาคผนวก ข บอร์ด ET-MINI DS1307 99

ประวัติผู้ดำเนินโครงงาน 100

สารบัญตาราง

ตารางที่ หน้า

3.1 รูปแบบการส่งข้อมูลแบบ HDLC PROTOCOL 9 BYTE 24
3.2 EXCLUSIVE OR (XOR) 25
3.3 คำสั่ง PC ส่งค่าตั้งเวลาให้ AVR 30
3.4 คำสั่ง AVR ตอบกลับให้ PC 31
3.5 คำสั่ง PC อ่าน TEXT FILE จาก AVR 35
3.6 คำสั่ง AVR ส่ง TEXT FILE ให้ PC 35
3.7 คำสั่ง PC ตอบกลับ COMMAND 0X03 36
3.8 คำสั่ง PC ส่งไป DELETE TEXT FILE 37
3.9 คำสั่ง AVR ตอบกลับ DELETE TEXT FILE ให้ PC 37
3.10 คำสั่ง PC ส่งค่า SETTING เวลารับประทานยา 37
3.11 คำสั่ง AVR ตอบกลับให้ PC 38
3.12 คำสั่ง PC ส่งค่า SETTING เวลารับประทานอาหาร 39
3.13 คำสั่ง AVR ตอบกลับให้ PC 40

สารบัญรูป

รูปที่ หน้า

2.1 กล่องบรรจุยา 5
2.2 กล่องบรรจุยาแบบตั้งเวลา 6
2.3 เครื่องจ่ายยาอัตโนมัติ 7
2.4 โครงสร้างของบอร์ด ET-BASE MEGA1280/2560 9
2.5 วงจรการเชื่อมต่อแบบ SPI โดยใช้ขาสัญญาณ PB [0..3] ในการเชื่อมต่อกับการ์ด 11
2.6 แผนภาพวงจรแหล่งจ่ายไฟ 11
2.7 เครื่องโปรแกรม ET-AVRISP MKII 13
2.8 แผนภาพแสดงตำแหน่งขาไอซี RTC DS1307 14
2.9 แผนภาพวงจร ET-MINI DS1307 15
2.10 แผนภาพการเชื่อมต่อ DS1307 เข้ากับไมโครคอนโทรลเลอร์ด้วยระบบบัสแบบ I2C 15
2.11 แผนภาพการเชื่อมต่อแบบ SPI 17
3.1 แผนภาพแสดงการเชื่อมต่ออุปกรณ์ 19
3.2 แผนภาพวงจรการต่อตัวต้านทานแบบ PULL UP 20
3.3 แผนภาพวงจรบัซเซอร์ 21
3.4 แผนภาพวงจรแบบขับกระแส (SOURCE CURRENT) 21
3.5 รูปแบบการคำนวณ CRC 27
3.6 แผนภาพแสดงเฟรมข้อมูลทั้งหมด 27
3.7 รูปแบบการคำนวณ CRC กรณีที่ตรวจไม่พบข้อผิดพลาด 28
3.8 รูปแบบการคำนวณ CRC กรณีที่ตรวจพบข้อผิดพลาด 29
3.9 แผนภาพการรับส่งข้อมูลเรียบร้อย 31
3.10 แผนภาพการส่งค่าเวลาเตือนการรับประทานยาและการรับประทานอาหารเรียบร้อย 32
3.11 แผนภาพรูปแบบ TEXT FILE แสดงสถานะการรับประทานยา 34
3.12 แผนภาพรูปแบบฐานข้อมูลผู้ป่วย 40
3.13 แผนภาพตารางฐานข้อมูลผู้ป่วยของการตั้งค่าเวลารับประทานยา 41
3.14 แผนภาพตารางฐานข้อมูลผู้ป่วยของการตั้งค่าเวลารับประทานอาหาร 41
3.15 แผนภาพตารางฐานข้อมูลผู้ป่วยของการรับประทานยา 42
3.16 แผนภาพตารางฐานข้อมูลผู้ป่วย 42

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.17 แผนภาพตารางฐานข้อมูลผู้ป่วยของการตั้งค่าการติดต่อสื่อสาร 43
3.18 กล่องบรรจุยาด้านนอก 44
3.19 กล่องบรรจุยาเม็ดยาจำนวน 7 กล่อง 44
3.20 การประกอบกล่องบรรจุเม็ดยากับกล่องยาด้านนอก 45
3.21 ฐานสำหรับบอร์ดไมโครคอนโทรลเลอร์และอุปกรณ์ที่ใช้งาน 45
3.22 การติดตั้งอุปกรณ์และบอร์ดไมโครคอนโทรเลอร์ 46
3.23 แผนภาพกรอบแสดงการทำงานของไมโครคอนโทรลเลอร์ 49
3.24 หน้าต่างแอพพลิเคชั่นของโปรแกรมวิชวลเบสิก 51
3.25 หน้าต่างตั้งค่าการติดต่อสื่อสาร 51
3.26 หน้าต่างตั้งค่าวันที่และเวลา 52
3.27 หน้าต่างตั้งค่าเวลาการรับประทานยาและเวลาการรับประทานอาหาร 53
3.28 หน้าต่างตั้งค่าข้อมูลผู้ป่วย 57
3.29 หน้าต่างแสดงผล 58
4.1 หน้าต่างแอพพลิเคชั่นของโปรแกรมวิชวลเบสิก 61
4.2 หน้าต่างตั้งค่าข้อมูลผู้ป่วย 63
4.3 หน้าต่างการตั้งค่าการติดต่อสื่อสาร 63
4.4 รายงานสถานการณ์เชื่อมต่อ 64
4.5 หน้าต่างการตั้งค่าวันที่และเวลา 64
4.6 รายงานการส่งค่าวันที่และเวลา 65
4.7 หน้าต่างการตั้งค่าเวลาการรับประทานยาและเวลาการรับประทานอาหาร 66
4.8 หน้าต่างแสดงการส่งค่าเวลาการเตือนเรียบร้อย 67
4.9 หน้าต่างแสดงการแจ้งเตือนข้อผิดพลาด 67
4.10 หน้าต่างแสดงผลการรับประทานยา 69
4.11 การพิมพ์รายงานผลการรับประทานยา 69
4.12 การแสดงหลอดไฟแอลอีดีรับประทานอาหาร 70
4.13 การติดของหลอดไฟแอลอีดี เมื่อถึงเวลารับประทานยา 71
4.14 หลอดไฟแอลอีดีดับ เมื่อครบกำหนดเวลา 71

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.15 หลอดไฟแอลอีดีติดตลอดช่วงเวลาที่กำหนด 72
4.16 หลอดไฟแอลอีดีติดค้างจนหมดช่วงเวลาเตือน 73
4.17 หลอดไฟแอลอีดีเริ่มทำงานที่ช่วงเวลาใหม่ 73

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องด้วยปัญหาที่พบจากผู้ป่วยสูงอายุ คือ ผู้ป่วยสูงอายุส่วนใหญ่ จะเป็นโรคพร้อมๆกันหลายๆโรค ได้แก่ ความดันเลือดสูง ความจำเสื่อม นอนไม่หลับ กังวล โรคซึมเศร้า โรคเบาหวาน โรคกระดูกเสื่อม เป็นต้น ทำให้ต้องมีการทานยาหลายชนิดและใช้ติดต่อกันนานๆเพื่อควบคุมและรักษาโรคเหล่านี้ ซึ่งการลืมทานยา ทำให้ช่วงระยะเวลาที่ได้รับยาเปลี่ยนแปลงไปอาจส่งผลให้ระดับยาในเลือดผิดปกติและเป็นอันตรายได้ ในเรื่องความร่วมมือในการใช้ยา ถือเป็นความรับผิดชอบร่วมกันทั้งผู้ใช้ยาและผู้สั่งจ่ายยา ที่จะต้องทำร่วมกัน เมื่อมีปัญหาเรื่องยาควรปรึกษาแพทย์หรือเภสัช เพื่อให้ผู้สูงอายุได้ใช้ยาอย่างเหมาะสมและตรงตามเวลาเท่าที่จำเป็นตามคำสั่งของแพทย์ เพื่อให้เกิดผลการรักษาที่ดีที่สุด ซึ่งโครงงานนี้จะมีส่วนช่วยให้ผู้ป่วย ลดอัตราการลืมรับประทานยา ซึ่งการใช้ยาอย่างถูกต้องย่อมเป็นสิ่งที่ทุกคนควรกระทำ แต่ถ้าการใช้ยานั้นเกิดการผิดพลาด ไม่ว่าเป็นเพราะสาเหตุใดก็ตาม อาจทำให้ผู้นั้นได้รับอันตรายหรือบางครั้งรุนแรงถึงแก่ชีวิตได้ โดยเฉพาะอย่างยิ่งการใช้ยาในผู้สูงอายุ ซึ่งพบว่าเป็นผู้ที่มีความเสี่ยง หรือมีโอกาสที่จะได้รับอันตรายจากการใช้ยาได้มากกว่าบุคคลทั่วไป

เพื่อแก้ปัญหาเกี่ยวกับการลืมทานยาทั้งก่อนรับประทานอาหารและหลังรับประทานอาหาร ซึ่งเป็นผลให้เกิดผลเสียตามมา อาทิเช่น การดื้อยา อาการไม่ดีขึ้น และอาจจะเป็นผลทำให้อาการที่เป็นอยู่ทรุดไปจากเดิม ดังนั้น คณะผู้จัดทำและหัวข้อโครงงานจึงมีแนวคิดจัดทำโครงงานในการแก้ปัญหาดังกล่าวที่เกิดขึ้น โดยการนำบอร์ดไมโครคอนโทรลเลอร์ และอุปกรณ์ต่างๆ มาใช้เป็นตัวแจ้งเตือน ตั้งเวลา รวมถึงบันทึกข้อมูลเพื่อเป็นสถิติการรับประทานยาตลอดช่วงเวลาที่ผู้ป่วยต้องรับประทานยา

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อให้ระบบแจ้งเตือนผู้ป่วยเมื่อถึงเวลารับประทานยารวมทั้งแจ้งเตือนเมื่อถึงเวลารับประทานอาหาร

1.2.2 เพื่อป้องกันอันตรายที่อาจเกิดกับผู้ป่วยเมื่อลืมรับประทานยา หรือรับประทานยาไม่ตรงเวลา ให้ได้รับยาอย่างเหมาะสมและตรงตามเวลาที่กำหนด

1.2.3 เพื่อให้ระบบบันทึกข้อมูลการรับประทานยาก่อนอาหารและหลังอาหารในแต่ละมื้ออาหาร เพื่อนำข้อมูลที่ได้มาใช้วิเคราะห์ว่าผู้ป่วยรับประทานยาได้ถูกต้องตามที่กำหนด เพื่อให้เป็นผลดีกับทางผู้ป่วยในการส่งเสริมให้ประสิทธิภาพทางการแพทย์ดียิ่งขึ้น คือ เพื่อการรักษาอย่างต่อเนื่องและได้ผลสูงสุด

1.3 ขอบเขตของโครงงาน

อุปกรณ์ที่พัฒนาขึ้นสามารถบันทึกข้อมูลการใช้ยาในแต่ละครั้งได้ และมีเสียงเตือนเมื่อถึงเวลาทานยา โดยจะมีเสียง รวมทั้งมีไฟแอลอีดีเตือนไปจนกว่าจะหยิบยาไปรับประทานหรือเมื่อถึงเวลาที่กำหนดให้หยุดเตือน รวมทั้งมีการเตือนในการรับประทานอาหารที่แสดงด้วยเสียงเตือนและไฟจากแอลอีดี ซึ่งข้อมูลการใช้ยาในแต่ละครั้งจะแสดงผลออกที่จอภาพของคอมพิวเตอร์ โดยโปรแกรมที่พัฒนาขึ้นสามารถบริหารและจัดการการทำงานของบอร์ดไมโครคอนโทรลเลอร์ในการเตือนเวลารับประทานยาและเวลารับประทานอาหารของผู้ป่วยแต่ละคนได้

1.4 ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	ปี 2554							ปี2555				
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1) ศึกษารายละเอียดการทำโครงงาน												
2) ศึกษาและเลือกอุปกรณ์เพื่อใช้ในโครงงาน												
3) ออกแบบกล่องยา												
4) ศึกษาและเขียนโปรแกรมภาษาซี												
5) ทดสอบและแก้ไขการทำงานของโปรแกรมและชิ้นงาน												
6) สรุปผลการดำเนินโครงงานและจัดทำรูปเล่มปริญญานิพนธ์												

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1.5.1 ผู้สนใจสามารถนำอุปกรณ์ที่พัฒนาขึ้นไปใช้ในการจัดสรรระบบการรับประทานยาได้

1.5.2 ผู้ศึกษาเข้าใจถึงหลักการทำงานของอุปกรณ์ไมโครคอนโทรลเลอร์ และอุปกรณ์สนับสนุนอื่นๆ

1.6 งบประมาณ

บอร์ดไมโครคอนโทรลเลอร์	1,000 บาท
อุปกรณ์สำหรับชิ้นงาน	1,000 บาท
ค่าถ่ายเอกสารและเข้าเล่มปริญญานิพนธ์	700 บาท
อื่นๆ	300 บาท
รวมเป็นเงินทั้งสิ้น (สามพันบาทถ้วน)	3,000 บาท

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

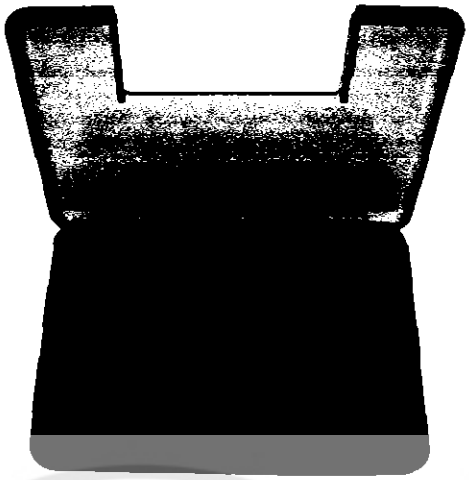
บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 คุณสมบัติและลักษณะของกล่องบรรจุยา

จากการศึกษาเกี่ยวกับลักษณะของกล่องยาที่มีใช้ในปัจจุบัน พบว่ากล่องยามีการใช้งานที่หลากหลาย จึงได้นำตัวอย่างกล่องยาที่ขายตามท้องตลาดมาวิเคราะห์เพื่อช่วยในการออกแบบกล่องยาสำหรับโครงงานนี้

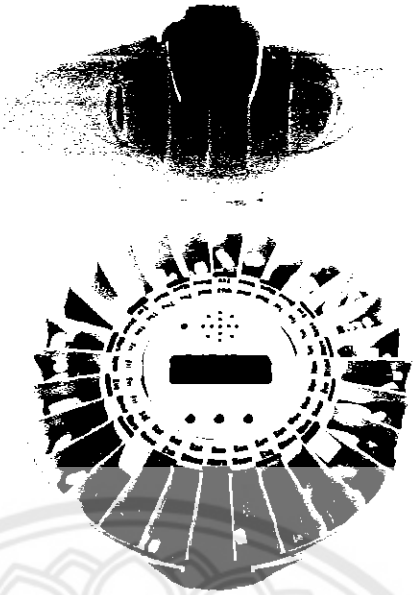
รูปที่ 2.1 กล่องบรรจุยา [1]

กล่องยาชนิดนี้เป็นกล่องยาที่มีช่องสำหรับบรรจุยาได้ 8 ช่อง ข้อดีสำหรับกล่องยานี้คือ มีขนาดเล็กเหมาะสำหรับพกพา เพื่อความสะดวกในการรับประทานยา ข้อเสียสำหรับกล่องยานี้คือ เมื่อผู้ป่วยลืมรับประทานยา อันเป็นเหตุให้มีอาการป่วยเรื้อรัง กล่องยาชนิดนี้จะไม่สามารถเตือนผู้ป่วยให้รับประทานยาตรงตามเวลาได้



รูปที่2.2 กล่องบรรจุยาแบบตั้งเวลา [2]

กล่องยาชนิดนี้เป็นกล่องยาที่ผลิตโดยบริษัท Shenzhen Ruixinlong Electronics Co., Ltd. ที่ตั้งอยู่ในเซินเจิ้น ของจีน เป็นกล่องยาอิเล็กทรอนิกส์ที่มีช่องสำหรับใส่ยาได้ 4 ช่องเพื่อสำหรับแยกประเภทของยา เหมาะสำหรับผู้ที่ต้องการรับประทานยาให้ตรงตามเวลาที่แพทย์สั่ง ตัวกล่องยามีการเตือนด้วยระบบดิจิตอลซึ่งสามารถตั้งเวลาทานยาได้ 4 ครั้งต่อวัน มีขนาด 106 X 73 X 17 มิลลิเมตร การตั้งเวลาจะคล้ายกับการตั้งเวลาของนาฬิกาข้อมือ ข้อดีของกล่องยานี้คือ สามารถตั้งเตือนเวลาได้ ผู้ป่วยจึงสามารถทานยาได้ตามที่กำหนดและมีขนาดเล็กเหมาะสำหรับพกพา ส่วนข้อเสียคือมีช่องใส่ยาที่น้อย ไม่สามารถตั้งเวลาในการรับประทานอาหารได้ จำนวนครั้งในการตั้งเวลาเตือนน้อย ไม่มีไฟแอลอีดี เตือนเมื่อถึงเวลารับประทานยา ไม่มีการบันทึกข้อมูลเพื่อสามารถนำมาเปิดดูในภายหลังได้



รูปที่ 2.3 เครื่องจ่ายยาอัตโนมัติ [3]

กล่องยานี้เป็นกล่องยาแบบอัตโนมัติที่ผลิตโดยบริษัท Pivotell Ltd มีช่องสำหรับใส่ยา 28 ช่องซึ่งทำให้เครื่องจ่ายยานี้สามารถบรรจุยาได้มากถึง 28 วัน ขึ้นอยู่กับการใช้งาน เนื่องจากตัวเครื่องจ่ายยามีการโปรแกรมให้สามารถตั้งเวลาได้ เมื่อถึงเวลาที่ต้องการทานยา จะมีเสียงไฟแสดงว่าถึงเวลารับประทานยา รวมทั้งมีไฟกระพริบในกรณีผู้ใช้เป็นผู้พิการทางการได้ยิน จากนั้นผู้ใช้เพียงแต่เอียงเครื่องจ่ายยาเทยาออกมาเท่านั้น เครื่องจ่ายยานี้มีเสียงเตือนทั้งหมด 4 เสียง ผู้ใช้จึงสามารถเลือกได้ตามความพึงพอใจ ข้อดีคือ บรรจุยาได้มาก ทำให้สะดวกต่อผู้ใช้ที่ไม่ต้องเปลี่ยนยาบ่อยๆ อีกทั้งสามารถตั้งเวลาเตือนพร้อมทั้งมีแสงไฟเมื่อถึงเวลาที่ต้องรับประทานยา ข้อเสียของเครื่องจ่ายยานี้คือ เนื่องจากเครื่องจ่ายยาบรรจุยาได้ปริมาณมากๆ เมื่อต้องบรรจุยาเป็นระยะเวลานาน อาจทำให้ยาเสื่อมคุณภาพ เป็นเหตุให้ประสิทธิภาพในการใช้ยาลดลง และยังไม่สามารถตั้งเวลารับประทานอาหาร เมื่อถึงเวลาที่ต้องรับประทาน

สำหรับกล่องยาที่ใช้สำหรับโครงงาน คือได้มีการปรับเปลี่ยนแก้ไขกล่องบรรจุยา เพื่อให้เหมาะสมและสะดวกต่อการใช้งาน จะมีกล่องใหญ่ 1 กล่องใช้บรรจุกล่องยาเล็กอีก 7 กล่องซึ่งสามารถบรรจุยาได้ 7 ช่วงเวลา ตัวกล่องยาจะอยู่บนฐานกล่องที่เชื่อมต่ออยู่กับบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASE MEGA 2560 ร่วมกับอุปกรณ์ต่างๆ ซึ่งได้แก่ หลอดไฟแอลอีดี ลำโพง และลิมิตสวิตช์ เพื่อช่วยในการแจ้งเตือนตามเวลาที่ผู้ใช้งานกำหนดไว้ที่หน้าต่างการตั้งค่า วิชวลเบสิก โดยการเขียนโปรแกรมภาษาซีที่ใช้สำหรับไมโครคอนโทรลเลอร์ AVR

พร้อมกับโปรแกรมบันทึกค่าลง Micro SD Card และนำมาแสดงผลออกที่จอภาพของคอมพิวเตอร์ ผ่านมาตรฐานการสื่อสารแบบ RS-232 โดยใช้ Format HDLC Protocal 9 byte(ไม่รวม data) ในการส่งผ่านข้อมูลไปยังไมโครคอนโทรลเลอร์ AVR

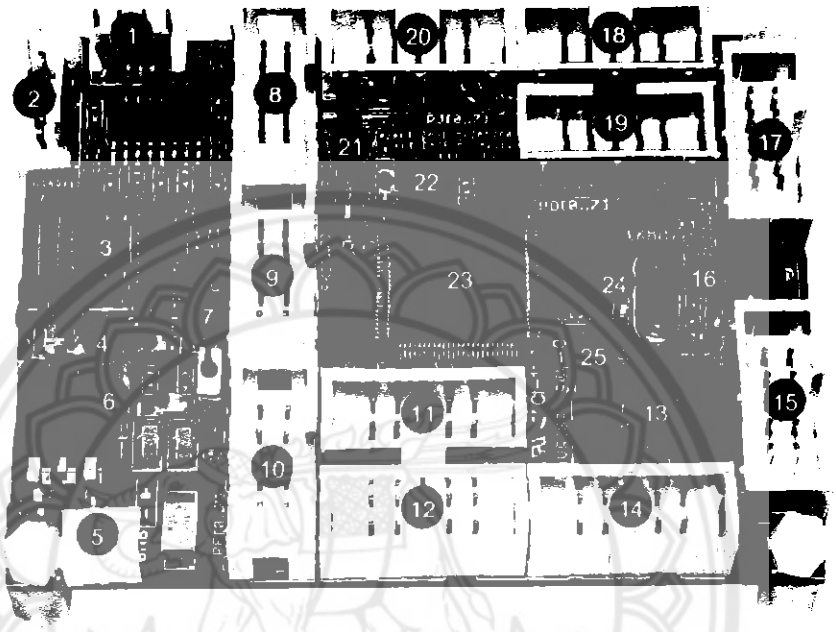
2.2 โครงสร้างบอร์ด ET-BASE MEGA2560

ไมโครคอนโทรลเลอร์ตระกูล AVR ของ ATMEL เป็นไมโครคอนโทรลเลอร์อีกตระกูลหนึ่งซึ่งได้รับความนิยมอย่างแพร่หลายจากผู้ใช้งานทั่วไป ซึ่งทาง ATMEL เองก็ได้มีการปรับปรุงพัฒนาขีดความสามารถของ MCU เพื่อตอบสนองความต้องการใช้งานในลักษณะต่างๆ มีการผลิตชิพ MCU ออกมาจำหน่ายเป็นจำนวนมากมายหลายเบอร์ เพื่อให้ผู้ใช้งานสามารถเลือก MCU ไปประยุกต์ใช้งานให้เหมาะสมกับงานได้ง่ายและสะดวกมากยิ่งขึ้น

ATMEGA ก็เป็น MCU ตระกูล AVR ที่มีความโดดเด่นและมีขีดความสามารถสูงในระดับต้นๆของ MCU 8 บิต ซึ่งใน ATMEGA1280/2560 เองเป็น MCU ตระกูล AVR MEGA ที่มีระบบ Peripheral I/O ต่างๆรวบรวมไว้ภายใน MCU มากมายหลากหลาย สามารถโปรแกรมโหมดการทำงานของ Peripheral I/O ใน ลักษณะต่างๆได้หลากหลาย ทำให้ง่ายและสะดวกในการนำไปดัดแปลงใช้กับงานแบบต่างๆได้โดยง่าย ซึ่งการพัฒนาโปรแกรมของบอร์ดก็สามารถเลือกใช้รูปแบบในการพัฒนาโปรแกรมด้วยโปรแกรมภาษาต่างๆที่สนับสนุนการใช้งานกับ AVR MEGA ได้ทั่วไป ตามความเหมาะสมซึ่งคุณสมบัติของบอร์ด ET-BASE MEGA2560 มีดังนี้

1. ใช้ MCU ตระกูล MEGA AVR เบอร์ ATMEGA1280/2560 ของ ATMEL
2. มีหน่วยความจำ Flash 128KB (ATMEGA 1280,256K(ATMEGA2560)), 8KB Boot loader, Static RAM 8KB และ EEPROM 4KB
3. ใช้ Crystal 16.00 MHz
4. มีวงจร Real Time Counter พร้อม XTAL ค่า 32.768KHz
5. รองรับการโปรแกรมแบบ In-System Programming แบบ ISP
6. มีวงจรเชื่อมต่อกับ AVR-JTAG ขนาด 10 Pin เพื่อทำการ Debug แบบ Real Time ได้
7. Power Supply ใช้แรงดันไฟฟ้า +5VDC พร้อมวงจร Regulate +3V3/3A ภายในบอร์ด พร้อม Jumper เลือกระบบแหล่งจ่ายให้เป็น 3.3V หรือ 5V ได้ตามต้องการ
8. มีวงจรเชื่อมต่อการ์ดหน่วยความจำแบบ SD Card (Micro SD) เชื่อมต่อแบบ SPI จำนวน 1 ช่อง
9. มีวงจรสื่อสาร RS232 โดยใช้ขั้วต่อแบบ 4-PIN มาตรฐาน ETT จำนวน 2 ช่อง

10. มี 83 Bit GPIO อิสระ สำหรับการประยุกต์ใช้งานต่างๆ เช่น A/D, I2C, SPI, USART และ Input/Output แบบต่างๆ โดยมีการจัดสรรใช้งานภายในบอร์ดไว้แล้ว จำนวน 8 เส้นสัญญาณ



รูปที่ 2.4 โครงสร้างของบอร์ด ET-BASE MEGA1280/2560 [4]

หมายเลข 1, 2 คือ ขั้วต่อ RS232 สำหรับใช้งาน

หมายเลข 3, 4 คือ ช่องเสียบการ์ดหน่วยความจำ SD Card แบบ Micro-SD และ LED สถานะ

หมายเลข 5, 6 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ดใช้ได้กับไฟ +5VDC และ LED Power

หมายเลข 7 คือ Jumper สำหรับเลือกขนาดแรงดันทำงานของ MCU ระหว่าง 3.3V หรือ 5V

หมายเลข 8,9,10,11 คือ ขั้วต่อ GPIO(PA[0..7]) ,(PK[0..7]),(PF[0..7]) และ GPIO(PE[0..7])

หมายเลข 12 คือ ขั้วต่อ AVR-JTAG สำหรับ Debug แบบ Real Time

หมายเลข 13 คือ ขั้วต่อ AVRISP สำหรับ Download โปรแกรม

หมายเลข 14, 15 คือ LED ขั้วต่อ GPIO (PH[0..7]) และ GPIO(PB[0..7])

หมายเลข 16 คือ SW RESET

หมายเลข 17,18,19,20 คือ ขั้วต่อ GPIO(PL[0..7]), (PC[0..7]),(PD[0..7]) และ GPIO(PJ[0..7])

หมายเลข 21, 22 คือ ขั้วต่อ GPIO (PG[0..2]) และ LED ใช้ทดสอบ Logic Output ของ PG5

หมายเลข 23 คือ MCU เบอร์ ATMEGA1280 หรือ ATMEGA2560 (100Pin TQFP)

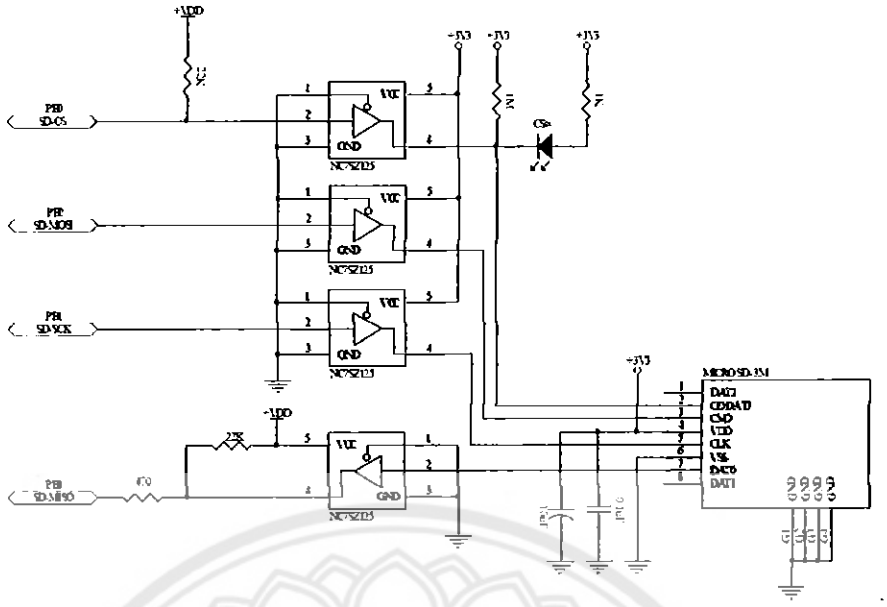
หมายเลข 24 คือ Crystal ค่า 16.00 MHz สำหรับใช้เป็นฐานเวลาระบบให้ MCU

หมายเลข 25 คือ Crystal ค่า 32.768 KHz สำหรับฐานเวลาให้ RTC ภายในตัว MCU

2.3 การ์ดหน่วยความจำ SD Card แบบ Micro-SD

บอร์ด ET-BASE MEGA1280/2560 รองรับการเชื่อมต่อกับการ์ดหน่วยความจำ SD Card แบบ Micro-SD โดยใช้การเชื่อมต่อแบบ SPI โดยใช้ขาสัญญาณ PB[0..3] ในการเชื่อมต่อกับการ์ด ซึ่งในการติดต่อสั่งงาน การ์ดนั้น สามารถโปรแกรม Pin I/O ของ PB[0..3] ให้ทำงานในโหมด SPI

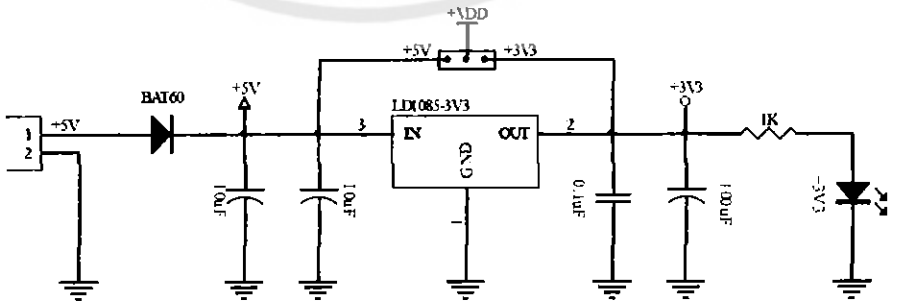
สำหรับ Socket สำหรับติดตั้งการ์ดหน่วยความจำของบอร์ด ET-BASE MEGA1280/2560 นั้น จะใช้ Socket หน่วยความจำแบบใส่การ์ดจากด้านบน แบบเดียวกันกับซิมการ์ดของโทรศัพท์มือถือ โดยเมื่อต้องการจะใส่หรือถอดการ์ดหน่วยความจำจะต้องทำการเปิดฝาครอบ Socket ออกเสียก่อน จากนั้นจึงจะสามารถใส่หรือถอดการ์ดหน่วยความจำได้ โดยการเปิด หรือปิด ฝาครอบ Socket จะใช้การกดเลื่อนฝาครอบเข้าหรือออก ซึ่งถ้าเลื่อนฝาครอบเข้าด้านในจะเป็นการเลื่อนเพื่อเปิดฝา แต่ถ้าเลื่อนออกด้านนอกจะเป็นการเลื่อนเพื่อล็อคฝาครอบ



รูปที่ 2.5 วงจรการเชื่อมต่อแบบ SPI โดยใช้ขาสัญญาณ PB [0..3] ในการเชื่อมต่อกับการ์ด [4]

2.4 วงจรแหล่งจ่ายไฟ

วงจรแหล่งจ่ายไฟของบอร์ดใช้งานได้กับไฟ DC ขนาด +5V/2A โดยใช้ขั้วต่อแบบ 2 Pin Block ป้องกันการเสียบสายกลับขั้ว พร้อมวงจรจ่ายไฟตรงขนาด +3.3V/3A โดยวงจรภาคแหล่งจ่ายไฟในส่วนที่เป็นวงจรจ่ายไฟตรงขนาด 3.3V นั้นจะจ่ายตรงไปให้กับ วงจร SD-Card ส่วนวงจรของ MPU และวงจร I/O ของบอร์ดนั้น จะสามารถเลือกขนาดแรงดันใช้งานได้จาก Jumper ว่าต้องการใช้งานเป็น 3.3V หรือ 5V ดังรูปที่ 2.6



รูปที่ 2.6 แผนภาพวงจรแหล่งจ่ายไฟ [4]

2.5 โปรแกรม Visual Basic (VB)

เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์[6] ที่กำลังเป็นที่นิยมใช้อยู่ในปัจจุบัน โปรแกรม Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล (Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event - Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด ผู้ใช้กดปุ่มเมาส์ เป็นต้น

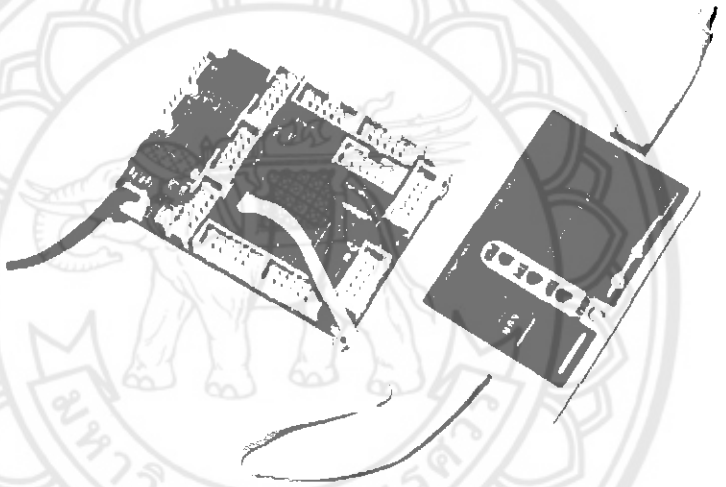
เครื่องมือ หรือ คอนโทรล ต่าง ๆ ที่ Visual Basic ได้เตรียมไว้ให้ ไม่ว่าจะเป็น Form TextBox Label ฯลฯ ถือว่าเป็นวัตถุ (Object ในที่นี้ขอใช้คำว่า ออบเจ็กต์) นั่นหมายความว่า ไม่ว่าจะเป็นเครื่องมือใด ๆ ใน Visual Basic จะเป็นออบเจ็กต์ทั้งสิ้น สามารถที่จะควบคุมการทำงานแก้ไขคุณสมบัติของออบเจ็กต์นั้นได้โดยตรง ในทุกๆ ออบเจ็กต์จะมีคุณสมบัติ (properties) และเมธอด (Methods) ประจำตัว ซึ่งในแต่ละออบเจ็กต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือน หรือต่างกันก็ได้ ขึ้นอยู่กับชนิดของออบเจ็กต์

ในการพัฒนาโปรแกรมประยุกต์ด้วย Visual Basic การเขียนโค้ดจะถูกแบ่งออกเป็นส่วนๆ เรียกว่า โพรซีเดอร์ (procedure) แต่ละโพรซีเดอร์จะประกอบไปด้วย ชุดคำสั่งที่พิมพ์เข้าไปแล้วทำให้คอนโทรลหรือออบเจ็กต์นั้น ๆ ตอบสนองการกระทำของผู้ใช้ ซึ่งเรียกว่าการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming-OOP) แต่ตัวภาษา Visual Basic ยังไม่ถือว่าเป็นการเขียนโปรแกรมแบบ OOP อย่างแท้จริง เนื่องจากข้อจำกัดหลายๆ อย่างที่ Visual Basic ไม่สามารถทำได้

2.6 เครื่องโปรแกรม ET-AVRISP mkII

ET-AVRISP mkII[7] เป็นเครื่องโปรแกรมไมโครคอนโทรลเลอร์ตระกูล AVR โดยใช้ Potocol เดียวกันกับ AVRISP mkII ของ ATMEL โดยเชื่อมต่อกับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB โดยคุณสมบัติของเครื่องโปรแกรม ET-AVRISP mkII จะมีคุณสมบัติการทำงานใกล้เคียงกันกับ AVRISP mkII ของ ATMEL โดยสามารถใช้โปรแกรม MCU ตระกูล AVR ได้ทั้งแบบที่เป็น ISP และ PDI และ TPI โดยใช้งานร่วมกับชุดโปรแกรม AVRStudio ของ ATMEL ซึ่งมีคุณสมบัติดังนี้

1. ใช้งานร่วมกับโปรแกรม AVRStudio
2. รองรับการโปรแกรม MCU ตระกูล AVR แบบ In-Circuit Program ทั้งแบบ ISP, PDI และ TPI
3. สามารถโปรแกรมได้ทั้งหน่วยความจำ Flash และ EEPROM ในตัว MCU
4. สามารถโปรแกรม Fuse Bit และ Lock Bit ของ MCU ได้
5. สามารถใช้ได้กับ Target Board ที่ใช้แหล่งจ่ายตั้งแต่ 1.8V-5.5V
6. สามารถปรับความเร็วของ ISP ในการโปรแกรมได้ตั้งแต่ 50Hz-8MHz
7. เชื่อมต่อกับคอมพิวเตอร์ PC ทางพอร์ต USB 2.0 แบบ Full Speed (12Mbps)
8. ใช้แหล่งจ่ายไฟจากพอร์ต USB



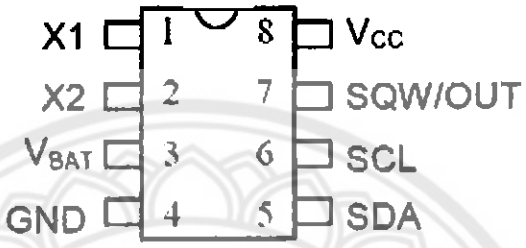
รูปที่ 2.7 เครื่องโปรแกรม ET-AVRISP mkII [4]

2.7 การใช้งาน RTC (Real Time Clock) ด้วย DS1307

ระบบฐานเวลา เป็นสิ่งสำคัญที่สามารถนำไปใช้ในอุปกรณ์อิเล็กโทรนิกส์ได้หลากหลาย ภายในไมโครคอนโทรเลอร์เองก็มีไทเมอร์เพื่อใช้ในการจับเวลา หรือนำไปใช้เป็นฐานเวลาจริงได้เช่นกัน แต่เนื่องจากไมโครคอนโทรเลอร์สามารถทำงานได้ต่อเมื่อมีไฟเลี้ยงเท่านั้น ดังนั้นการใช้ไทเมอร์ของไมโครคอนโทรลเลอร์ สร้างฐานเวลาจริงจึงไม่เหมาะสมในบางแอปพลิเคชัน

DS1307 เป็น IC ฐานเวลาของดัลลัสเซมิคอนดักเตอร์[8] (Dallas Semiconductor) มีบัสรับส่งข้อมูลแบบ I^2C[8] ซึ่งเป็นแบบ 2 wire สามารถสื่อสารได้ 2 ทิศทาง (bi-direction bus) ฐาน

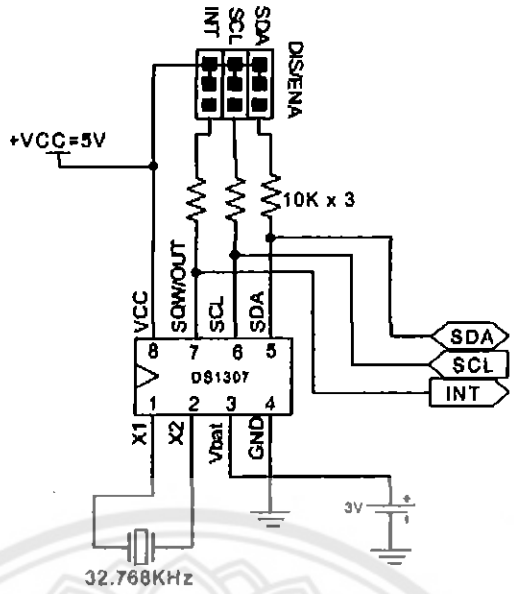
เวลาของ DS1307 นั้นสามารถเก็บข้อมูล วินาที, นาที, ชั่วโมง, วัน, วันที่, เดือน และปี ได้ ระบบเวลาสามารถทำงานโหมดรูปแบบ 24 ชั่วโมง หรือ 12 ชั่วโมง AM/PM ก็ได้ ภายมีระบบตรวจจับแหล่งจ่ายไฟ โดยถ้าแหล่งจ่ายไฟหลักถูกตัดไป DS1307 สามารถสวิตซ์ไปใช้ไฟจากแบตเตอรี่และทำงานต่อไป โดยที่ยังสามารถรักษาข้อมูลไว้ได้ โครงสร้างมีขาทั้งหมด 8 ขาดังแสดงในรูปที่ 2.8 และมีรายละเอียดการทำงานของขาต่าง ๆ ดังนี้



รูปที่ 2.8 แผนภาพแสดงตำแหน่งขาไอซี RTC DS1307 [8]

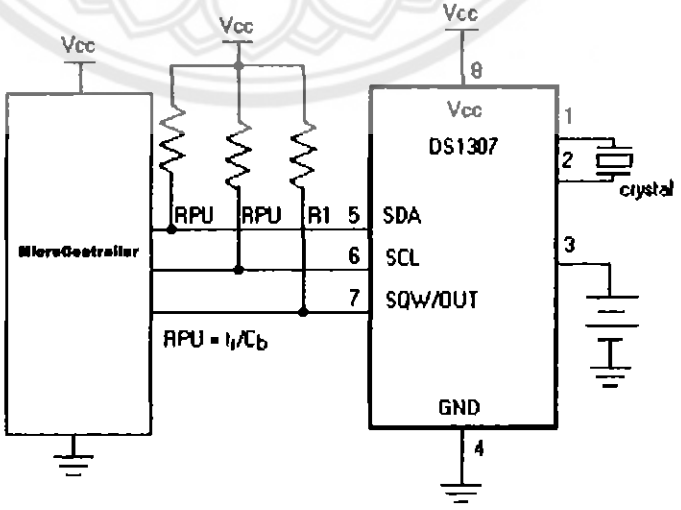
เป็นชุดการต่อวงจรในแบบ I^2C BUS โดยเป็นวงจร RTC ใช้เบอร์ DS1307 ขนาด 8 PIN พร้อมวงจร BATTERY 3V ใช้ BACKUP ข้อมูลของตัว RTC ขั้ว INPUT สัญญาณเข้าแบบ 5 PIN ตัวผู้และตัวเมีย

ก่อนการใช้งานโมดูลนี้จะต้องทำการ Set Jumper SDA, SCL และ INT มาทางด้าน ENA เพื่อเป็นการต่อ R Pull Up ให้กับขาทั้ง 3 ขาของ I^2C ถ้ามีการต่อ R Pull Up ใน Line SDA และ SCL จากภายนอกหรือจาก Module อื่นไว้แล้วให้ Set Jumper ทั้ง 3 มาทางด้าน DIS เพื่อไม่ต้องต่อตัวต้านทาน R Pull Up เข้าไปอีก



รูปที่ 2.9 แผนภาพวงจร ET-MINI DS1307 [4]

การใช้งานกับ ไมโครคอนโทรลเลอร์ แบบอนุกรม ตามมาตรฐานของ I^2C BUS การต่อใช้งานใช้สายสัญญาณเพียง 2 เส้น คือขา serial data (SDA) และขา serial clock (SCL) รายละเอียดของ RTC DS1307 มีการทำงานเป็นแบบ Master, Slave โดยอุปกรณ์ที่ทำหน้าที่เป็น Master (ไมโครคอนโทรลเลอร์) จะควบคุมการรับส่งข้อมูล และควบคุมสัญญาณนาฬิกาบน SCL ส่วนอุปกรณ์ Slave (DS1307) นั้นจะทำงานภายใต้การควบคุมของอุปกรณ์ ไมโครคอนโทรเลอร์



รูปที่ 2.10 แผนภาพการเชื่อมต่อ DS1307 เข้ากับไมโครคอนโทรลเลอร์ด้วยระบบบัสแบบ I2C [4]

ในการทดลองได้ต่อ DS1307 กับไมโครคอนโทรลเลอร์ โดยใช้พอร์ตของไมโครคอนโทรเลอร์เป็นบัส I^2C ต่อกับ SCL และ SDA ของ DS1307 ส่วนขาสัญญาณ SQW/OUT นั้นไม่ได้ใช้สร้างสัญญาณให้ไมโครคอนโทรลเลอร์ แต่ใช้การวนลูปคอยตรวจสอบค่าภายในรีจิสเตอร์ของ DS1307 แทน

ในการควบคุมการทำงานของโปรแกรม และแสดงผล ได้ใช้โปรแกรม HyperTerminal เป็นโปรแกรมติดต่อผ่านพอร์ตอนุกรมด้วยอัตราข้อมูล 9600 bps

2.8 AVR Studio4

ในการพัฒนาโปรแกรมของ AVR เบอร์ ATMEGA1280/2560 จะสามารถเลือกใช้ Compiler ต่างๆที่สนับสนุน MCU เบอร์นี้อยู่ได้ทั้งหมด จึงเลือกใช้โปรแกรม AVR Studio4 ร่วมกับ WinAVR ในการทำงาน

AVR Studio เป็นซอฟต์แวร์สำหรับสร้างโปรแกรมควบคุม[9] โดยภายใน AVR Studio ได้บรรจุเครื่องช่วยอำนวยความสะดวกต่อการพัฒนาโปรแกรมภาษาซีและแอสแซมบลี ทั้งยังสามารถเชื่อมโยงการทำงานในส่วนต่างๆ กับคอมไพเลอร์ที่กำหนดได้ด้วย โดย ซีคอมไพเลอร์ที่นำมาใช้ร่วมกันคือ WinAVR ทั้งยังสามารถติดต่อกับซอฟต์แวร์สำหรับโปแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ AVR ดังนั้นซอฟต์แวร์ AVR Studio จึงเหมาะอย่างยิ่งในการนำมาใช้พัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์ AVR โดย AVR Studio

2.9 WinAVR

WinAVR[10] เป็นซอฟต์แวร์ C คอมไพเลอร์หรือตัวแปลโปรแกรมภาษาซี สำหรับไมโครคอนโทรลเลอร์ AVR โดย WinAVR เป็นซอฟต์แวร์แบบโอเพ่นซอร์ส (open source) พัฒนาต่อจาก GNU GCC คอมไพเลอร์ เมื่อทำการติดตั้ง WinAVR แล้วจะสามารถเชื่อมโยงการทำงานเข้ากับ AVR Studio ได้ ดังนั้นจึงสามารถทำการเขียนโปรแกรมภาษาซี บน AVR Studio แล้วทำการคอมไพล์โปรแกรมด้วย WinAVR ได้อย่างต่อเนื่องโดยผลลัพธ์ของการคอมไพล์จะได้ไฟล์เป็นนามสกุล .hex อันเป็นไฟล์รหัสภาษาเครื่องหรือที่เรียกว่า "แมทชีนโค้ด" โดยเป็นไฟล์ผลลัพธ์ที่ได้จากการพัฒนาสามารถนำดาวน์โหลดลงสู่ไมโครคอนโทรลเลอร์ต่อไปได้ทันที

2.10 Serial Peripheral Interface, SPI

ระบบ SPI หรือ Serial Peripheral Interface[11] เป็นรูปแบบการสื่อสารระบบบัส ในรูปแบบการสื่อสารอนุกรม โดยใช้ตัวนำในการสื่อสาร 3 - 4 ตัวนำ แต่โดยทั่วไป จะใช้ 3 ตัวนำ เป็น Synchronous Serial Data Link ซึ่งจัดการในโหมด Full Duplex มีการติดต่อกับอุปกรณ์ในโหมด Master/Slave สัญญาณสามารถส่งหากันได้ระหว่าง MASTER และ SLAVE ได้อย่างต่อเนื่อง ระบบบัสของ SPI จะใช้สายสัญญาณจำนวน 4 เส้น (ไม่นับสาย GND) คือ SCK (Serial Clock), SSEL (Slave Select), MISO (Master In Slave Output) และ MOSI (Master Out Slave In) โดยสัญญาณ SCK เป็นสัญญาณนาฬิกาที่สร้างให้กับบัสเมื่อทำงานในโหมดมาสเตอร์หรือเป็นขารับ สัญญาณนาฬิกาเมื่อทำงานเป็น Slave ขา MISO เป็นขารับข้อมูลสำหรับมาสเตอร์ ส่วน MOSI เป็นขาส่งข้อมูลจากมาสเตอร์ ซึ่งมาสเตอร์ทำหน้าที่เป็นตัวควบคุมการสื่อสารทั้งหมด โดยควบคุมการสื่อสารตามสัญญาณนาฬิกา ตัวมาสเตอร์จะเป็นตัวที่ตัดสินใจ รับ หรือ ส่งข้อมูล

รูปที่ 2.11 แผนภาพการเชื่อมต่อแบบ SPI [11]

อุปกรณ์ที่ยังคงมีการใช้การสื่อสารแบบ SPI อยู่

1. การแปลงข้อมูลจาก Analog to Digital หรือจาก Digital to Analog
2. การติดต่อกับหน่วยความจำ EEPROM และ FLASH
3. ไอซีประเภท Real Time Clock : RTC
4. เซ็นเซอร์ จำพวก Temperature sensor , Pressure sensor
5. อื่น ๆ เช่น signal mixer , Potentiometer , LCD controller , USART , CAN controller , USB controller , Amplifier

บทที่ 3
ขั้นตอนการเขียนโปรแกรมและการออกแบบชิ้นงาน

โครงงานนี้มีวัตถุประสงค์เพื่อพัฒนาอุปกรณ์กล่องยาอิเล็กทรอนิกส์สำหรับยาชนิดเม็ดของผู้สูงอายุที่มีโรคประจำตัว ที่ต้องใช้ยารักษาโรคเป็นประจำให้ได้รับความสะดวกสบาย และลดโอกาสในการลืมรับประทานยา โดยเฉพาะผู้สูงอายุที่ไม่มีคนดูแล หรือผู้สูงอายุที่ยังสามารถดูแลตนเองได้ ให้มีคุณภาพชีวิตที่ดีขึ้นในเรื่องการรับประทานยา และไม่ขาดโอกาสในการรักษาอย่างต่อเนื่อง ซึ่งมีขั้นตอนการดำเนินงานตามขั้นตอนต่อไปนี้

การออกแบบและการสร้างกล่องบรรจุยา โดยการเริ่มออกแบบส่วนที่เป็นฐานกล่อง ตัวกล่องภายนอก เพื่อใช้สำหรับบรรจุบอร์ดไมโครคอนโทรลเลอร์และอุปกรณ์อื่นๆ ดังนั้นการออกแบบกล่องยาจึงต้องใช้หลักการทางด้านวิศวกรรมและหลักการในการเลือกใช้วัสดุและอุปกรณ์ที่เหมาะสมกับการใช้งานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ รวมถึงต้องคำนึงถึงเรื่องราคาของวัสดุอุปกรณ์ที่เลือกใช้ ดังนั้นจากหลักการและทฤษฎีในบทที่สองสามารถใช้วิเคราะห์การเลือกใช้อุปกรณ์ได้อย่างเหมาะสม เมื่อสร้างกล่องยาและต่อวงจรใช้งานเข้ากับบอร์ดไมโครคอนโทรลเลอร์เรียบร้อยแล้วจึงเริ่มเขียนโปรแกรมควบคุมการทำงานให้กับบอร์ดไมโครคอนโทรเลอร์ AVR ซึ่งจะใช้ภาษาซีในการเขียนโปรแกรม รวมถึงการเขียนโปรแกรมและสร้างซอฟต์แวร์แอพพลิเคชันในรูปแบบต่างๆให้สามารถใช้งานได้ง่ายด้วยโปรแกรมวิชวลเบสิก

หลังจากเข้าใจหลักการทำงานรวมถึงคำสั่งภาษาซีที่ใช้กับ AVR แล้ว เราจึงเริ่มเขียนโปรแกรม โดยต้องกำหนดตำแหน่งอินพุตและเอาท์พุต และอุปกรณ์ภายในให้ตรงตามบอร์ดไมโครคอนโทรเลอร์ในแต่ละรุ่น นำไปควบคุมอินพุตและเอาท์พุต แล้วตรวจสอบความผิดพลาดจากโปรแกรมรวมถึงการทำงานของอุปกรณ์ที่ต่อเข้าอินพุตและเอาท์พุต

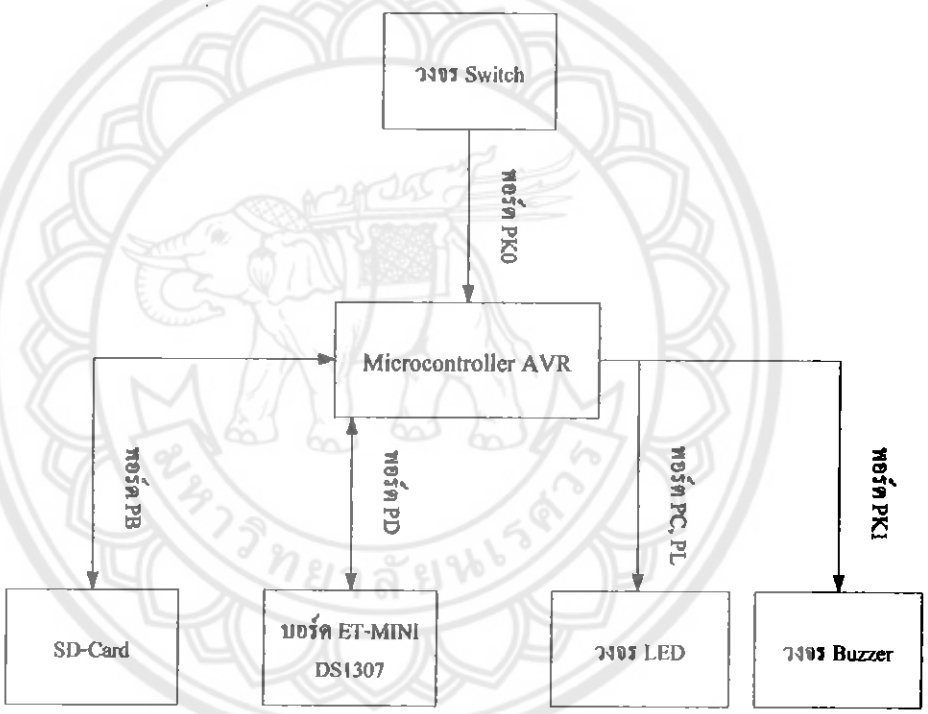
ในการนำไมโครคอนโทรลเลอร์ไปใช้งานนั้น เกี่ยวข้องกับอิเล็กทรอนิกส์โดยตรง ต้องมีความรู้พื้นฐานในเรื่องอิเล็กทรอนิกส์และการเขียนโปรแกรมด้วยภาษาซีกับ AVR ซึ่งรูปแบบใช้งานไมโครคอนโทรลเลอร์นั้น อธิบายได้ดังนี้

1. เขียนโปรแกรมด้วยภาษาซี และบันทึก
2. ในการเขียนโปรแกรมด้วย ภาษาซี นั้น จะใช้ตัวคอมไพเลอร์ AVR studio 4 (แปลภาษา) โปรแกรมที่เขียนขึ้นมาให้เป็นภาษาที่ไมโครคอนโทรลเลอร์รู้จัก
3. โหลดข้อมูลลงไมโครคอนโทรลเลอร์

4. นำไมโครคอนโทรลเลอร์ทดสอบการทำงาน
5. เมื่อทดสอบผ่าน นำไปใช้งานตามที่ออกแบบไว้

3.1 การทำงานของอุปกรณ์ที่ใช้ควบคุม

สำหรับโครงงานนี้ ใช้บอร์ดไมโครคอนโทรลเลอร์ ตระกูล AVR เบอร์ ET-BASE MEGA2560 ทำหน้าที่ในการรับคำสั่งจากเครื่องคอมพิวเตอร์เพื่อไปสั่งงานควบคุมตัวอุปกรณ์ของวงจรใช้งานต่างๆ แสดงผลที่บัซเซอร์ ลิมิตสวิตช์ และหลอดไฟแอลอีดี สามารถอธิบายดังรูปที่ 3.1

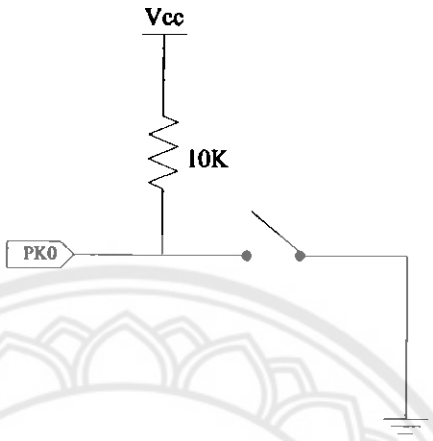


รูปที่ 3.1 แผนภาพแสดงการเชื่อมต่อของอุปกรณ์

3.1.1 การใช้งานลิมิตสวิตช์

การนำลิมิตสวิตช์ต่อกับตัวต้านทานแบบ Pull Up โดยรูปแบบการต่อวงจรเป็นการต่อในลักษณะที่ตัวต้านทานต่อเข้ากับ V_{cc} เพื่อรักษาระดับแรงดันให้คงที่ ทำให้วงจรอยู่ในสถานะลอจิก "1" หรือที่เรียกว่า Active High ตลอดเวลา และเมื่อมีการกดสวิตช์ จะทำให้สถานะเป็นลอจิก "0" หรือที่เรียกว่า Active Low เพราะจะทำงานเมื่อกระแสไฟฟ้าไหลลงกราวด์ วงจรแบบ Pull Up

ได้รับความนิยมมากเพราะกันสัญญาณรบกวนได้ดี ฉะนั้นเมื่อสวิตซ์ off จะได้ "1" และ on จะได้ "0" ซึ่งจะนำลิมิตสวิตช์เชื่อมต่อกับพอร์ต PK0 ดังรูปที่ 3.2

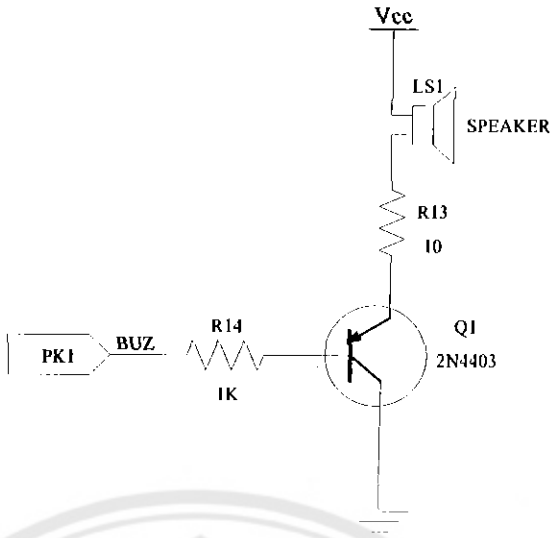


รูปที่ 3.2 แผนภาพวงจรการต่อตัวต้านทานแบบ Pull Up

การทำงานของสวิตช์ เพื่อใช้ตรวจสอบการกดสวิตช์ เช่น เมื่อสัญญาณเตือนเวลาทานยาดังขึ้น ผู้ป่วยจะหยิบยาโดยการเปิดฝากล่องทำให้ฝากล่องยาไปกดที่สวิตช์ เมื่อสวิตช์ถูกกดแล้วจะบันทึกค่าเวลาการรับประทานยาในขณะนั้นเก็บไว้ใน SD-Card แต่ถ้าหากผู้ป่วยไม่ได้หยิบยาไปรับประทาน จนหมดช่วงเวลาการเตือนแล้ว ข้อมูลจะบันทึกลง SD-Card เช่นกัน แต่จะบันทึกเวลาสิ้นสุดการเตือนพร้อมกับบันทึกสถานะว่า "ไม่ทาน"

3.1.2 การใช้งานวงจร Buzzer

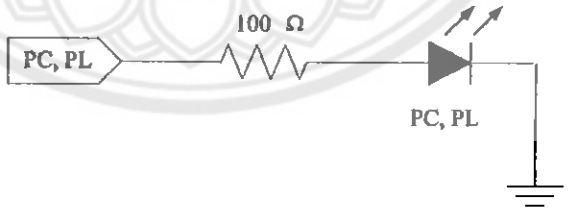
การสร้างสัญญาณเสียงออกทางบัซเซอร์ (Buzzer) ซึ่งจะควบคุมการทำงานโดยพอร์ต PK1 ในการขับบัสเซอร์นั้นจำเป็นต้องต่อผ่านทรานซิสเตอร์ เพราะกระแสที่ได้จากพอร์ตของไมโครคอนโทรลเลอร์นั้นไม่เพียงพอที่จะขับบัสเซอร์ได้โดยตรง โดยนำบัซเซอร์เชื่อมต่อกับพอร์ต PK1 เพื่อแสดงสัญญาณเสียงเมื่อถึงเวลารับประทานยา และรับประทานอาหาร ซึ่งการต่อวงจรบัซเซอร์แสดงดังรูปที่ 3.3



รูปที่ 3.3 แผนภาพวงจรบัซเซอร์

3.1.3 การใช้งานวงจรขับ LED แสดงผล

LED แสดงไฟของบอร์ดไมโครคอนโทรลเลอร์ AVR จะต่อวงจรแบบขับกระแส (Source Current) ทำงานด้วยโลจิก "1" และหยุดทำงานด้วยโลจิก "0" โดยควบคุมการทำงานจากขาสัญญาณ PC และ PL โดยวงจรในส่วนนี้จะใช้สำหรับทดสอบการทำงานของ Output จากขาสัญญาณ PC และ PL



รูปที่ 3.4 แผนภาพวงจรแบบขับกระแส (Source Current)

โดยเมื่อต้องการใช้งานผู้ใช้ต้องกำหนดให้ PC และ PL ทำหน้าที่เป็น GPIO Output Port เสียก่อนแล้วจึงควบคุม Logic ให้กับ PC และ PL โดยที่ (GPIO: General Purpose Input/Output) เป็นการกำหนดให้ขาหรือ PIN ของไมโครคอนโทรลเลอร์ทำงานเป็นอินพุตหรือเอาท์พุต ในที่นี้เลือกการใช้งานเป็นเอาท์พุต

สำหรับโครงงานนี้ LED เป็นพอร์ตเอาต์พุตควบคุมหลอดไฟแอลอีดี เพื่อแสดงสัญญาณไฟเมื่อถึงเวลารับประทานยาและเวลารับประทานอาหาร จะทำงานด้วยโลจิก "1" และ หยุดทำงานด้วยโลจิก "0" โดยควบคุมการทำงานจากพอร์ต PC และ PL โดยเมื่อต้องการใช้งานผู้ใช้ต้องกำหนดให้ PC[0..7] และ PL[0..7] ทำหน้าที่เป็น GPIO Output Port เสียก่อนแล้วจึงควบคุม Logic ให้กับ PC[0..7] และ PL[0..7] ตามต้องการ

3.2 หลักการทำงานของระบบ

3.2.1 รูปแบบการส่งข้อมูล

ในการส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม จะต้องเริ่มต้นด้วยการกำหนดอัตราการรับส่งข้อมูลระหว่างไมโครคอนโทรเลอร์กับคอมพิวเตอร์ก่อน โดยจะต้องมีการกำหนดให้ตรงกัน หากต้องการให้อุปกรณ์ทั้งสองที่ติดต่อกันโดยรับส่งข้อมูลที่เฉพาะเจาะจง ต้องประกอบไปด้วย ลักษณะการส่งข้อมูล รูปแบบการส่งข้อมูล การเริ่มต้นการติดต่อ และการยกเลิกการติดต่อ สิ่งเหล่านี้เรียกว่า โปรโตคอล

โปรโตรคอลเป็นการตกลงกันระหว่างอุปกรณ์ทั้งฝ่ายรับและฝ่ายส่ง หรือการรับข้อมูลระหว่างกันว่าจะใช้การส่งข้อมูลกันในลักษณะไหน เพราะหากไม่มีการกำหนดรูปแบบการรับ-ส่งที่เป็นมาตรฐานเดียวกันแล้ว จะทำให้ไม่สามารถสื่อสารกันได้

ในการออกแบบและพัฒนาซอฟต์แวร์ของบอร์ดไมโครคอนโทรลเลอร์ร่วมกับ Timer และอุปกรณ์สนับสนุนอื่นๆ สามารถเขียนหน้าจอการตั้งค่าการใช้งานด้วยโปรแกรม วิชวลเบสิก เริ่มจากการเขียนโปรแกรมและสร้างซอฟต์แวร์แอพพลิเคชันในรูปแบบต่างๆด้วยภาษาวิชวลเบสิก ที่ใช้กำหนดตั้งค่าการใช้งาน และตั้งค่าบันทึกลงหน่วยความจำ SD Card จากนั้นทำการสั่งค่าลงบอร์ดไมโครคอนโทรลเลอร์โดยใช้ภาษาซีในการออกแบบคำสั่งต่างๆที่จะป้อนลงในไมโครคอนโทรลเลอร์ เมื่อบอร์ดไมโครคอนโทรเลอร์ทำงานตามคำสั่งที่ถูกส่งจากหน้าต่างโปรแกรม วิชวลเบสิก จะมีการแสดงสัญญาณเตือนให้กับผู้ป่วย เพื่อให้ทราบว่าถึงเวลารับประทานยา และหยิบยาไปรับประทาน

3.2.2 รูปแบบการรับส่งข้อมูลแบบ HDLC Protocol 9 byte

โปรโตคอล HDLC (High-Level Data Link Control) เป็นโปรโตคอลมาตรฐานซึ่งกำหนดขึ้นโดยองค์กร ISO[12] เพื่อเพิ่มความเร็วในการส่ง-รับข้อมูลแบบซิงโครนัส สามารถใช้ได้กับการเชื่อมโยงการส่งข้อมูลทั้งแบบจุดต่อจุด และแบบหลายจุด การส่ง-รับข้อมูลเป็นได้ทั้ง Half Duplex และ Full Duplex ข้อมูลที่ส่ง-รับเป็นแบบบิตข้อมูล (Bit-Oriented) ใช้วิธีการส่งผ่านข้อมูลแบบซิงโครนัส

โครงสร้างเฟรมข้อมูล (Frame format)

ประกอบด้วย 6 ฟิลด์ โดยจะมีฟิลด์แฟล็กเริ่มต้น ฟิลด์แอดเดรส ฟิลด์ควบคุม ฟิลด์ควบคุม ฟิลด์ข้อมูล ฟิลด์ตรวจสอบ และฟิลด์แฟล็กสิ้นสุด ถ้ามีเฟรมข้อมูลหลายๆ เฟรมฟิลด์แฟล็กสิ้นสุดของเฟรมหนึ่งสามารถเป็นฟิลด์แฟล็กเริ่มต้นของอีกเฟรมหนึ่งได้

โปรโตคอล HDLC จะกำหนดผู้ส่งและผู้รับเป็นสถานีปฐมภูมิและทุติยภูมิ (Primary/Secondary) โดยสถานีปฐมภูมิจะเป็นผู้ควบคุมการติดต่อเชื่อมโยงและการส่ง-รับข้อมูลภายในเครือข่ายการสื่อสาร ส่วนสถานีทุติยภูมิจะเป็นเพียงผู้รับคอยปฏิบัติตามคำสั่ง และส่งสัญญาณตอบรับเท่านั้น

ลักษณะของข้อมูลที่ส่งออกจากผู้ส่ง หรือสถานีปฐมภูมิจะมีลักษณะเป็นเช่นเดียวกับโปรโตคอล SDLC[13] (Synchronous Data Link Control) คือส่งเป็นเฟรม โดยมีส่วนหัวและส่วนท้ายข่าวสารเช่นเดียวกันตามรูปเฟรมข้อมูลของ SDLC

คุณสมบัติพิเศษอย่างหนึ่งของโปรโตคอล HDLC คือแทนที่จะต้องส่งข้อมูลไปทีละ 1 บล็อก แล้วต้องคอยตอบรับแบบบล็อกต่อบล็อก เช่น ในการสื่อสารในโปรโตคอล SDLC สถานีปฐมภูมิสามารถส่งข้อมูล ได้ทีละหลาย ๆ บล็อกพร้อมกัน แล้วค่อยรอการตอบรับจากสถานีทุติยภูมิว่าผลรับข้อมูลเป็นอย่างไร ซึ่งทำให้ลด การเสียเวลาในการตอบรับบล็อก ซึ่งอาจจะมีการตอบรับเพียงครั้งเดียวสำหรับการส่งข้อมูลทั้งหมดก็ได้ วิธีการ นี้เรียกว่า "Sliding Windows"

โปรโตคอล HDLC ให้ประสิทธิภาพในแง่ที่ประหยัดเวลา สามารถเพิ่มความเร็วในการส่งข้อมูล ได้ดีกว่าโปรโตคอล SDLC ถ้าไม่มีความผิดพลาด หรือมีความผิดพลาดในการส่งข้อมูลน้อย แต่ถ้ามีความผิด พลาดเกิดขึ้นมากจะทำให้เสียเวลาในการส่งข้อมูลย้อนหลังมาให้กับสถานีทุติยภูมิใหม่บ่อย ๆ ทำให้ยิ่งเสียเวลา กว่าการใช้โปรโตคอล SDLC

ตารางที่ 3.1 รูปแบบการส่งข้อมูลแบบ HDLC Protocol 9 byte [14]

			Range Possible	
			Heximal	Decimal
Byte Start	1 byte	0x7E	0x7E	126
Address Destination	1 byte	0x01 (your destine)	0x01-0x1F	0-255
Object	1 byte	0x01 (your destine)	0x01-0x3F	0-255
Data size	2 byte	[byte hi-bit][byte low]	0x000000-0xFFFFFF	0-65535
Data				
CRC	2 byte	[byte hi-bit][byte low](Calculate Data only)	0x000000-0xFFFFFF	0-65535
Byte Stop	1 byte	0x7E	0x7E	126

CRC ย่อมาจาก Cyclic Redundancy Check คือข้อมูล 2 Byte หลังสุดของข้อมูลแต่ละ Message ที่ส่งด้วย Modbus Protocol ประโยชน์เพื่อใช้สำหรับตรวจสอบ Error โดยข้อมูลแต่ละ Byte ใน Message ที่ส่งไปจะถูกนำมาคำนวนหาค่า CRC ในขณะเดียวกันเมื่ออุปกรณ์ที่รับ Message ก็จะคำนวนค่า CRC ด้วยแล้วนำมาเปรียบเทียบกับค่า CRC ที่ส่งมาจากอุปกรณ์ที่ส่ง Message มา ดังนั้นเมื่อข้อมูลไม่ว่าบิตไหนก็ตามของ Message ที่รับมาไม่ถูกต้อง ค่า CRC ที่คำนวนได้ก็จะต่างออกไปจาก CRC ของอุปกรณ์ที่ส่ง Message มา ก็จะทำให้ทราบว่ามี Error เกิดขึ้น

หลักการของ CRC จะใช้รหัสโพลีโนเมียล (Polynomial Codes) ซึ่งต้องมีคุณสมบัติดังนี้

1. บิตซ้ายสุด และบิตขวาสุด จะต้องมีค่าเป็น 1 เสมอ

2. รหัสโพลิโนเมียล จะต้องมีจำนวนบิตที่น้อยกว่าบิตของข้อมูล

3. ลักษณะของโพลิโนเมียลจะเป็นลักษณะแบบ Modulo 2 จะเป็นในลักษณะ Exclusive-OR นั่นเอง โดยบิตที่นำมาบวกหรือลบกัน หากบิตตรงกัน ผลที่ได้จะเท่ากับ 0 ในขณะที่บิตต่างกัน ผลที่ได้จะมีค่าเท่ากับ 1

ตารางที่ 3.2 Exclusive or (XOR) [14]

Data1	Data2	(XOR)output
0	0	0
0	1	1
1	0	1
1	1	0

สูตรคำนวณ

M(x) คือเฟรมข้อมูลที่ต้องการส่ง

G(x) คือโพลีโนเมียลที่ตั้งขึ้นมา

n คือบิตศูนย์ที่เพิ่มเติมด้วยนำไปแปะท้าย M(x) จำนวนบิตที่เติม จะพิจารณาจากดีกรีของ G(x)

R(x) คือผลของเศษที่ได้จากการคำนวณ

T(x) คือเฟรมที่ส่งไป โดย

T(x) = M(x) + R(x)

การคำนวณ CRC

กำหนดให้

M(x) = 1 1 0 1 0 1 1 0 1 1

$G(x) = x^4 + x + 1$

จะได้ว่า

$G(x) = (1*x^4) + (0*x^3) + (0*x^2) + (1*x^1) + (1*x^0)$

$= 1\ 0\ 0\ 1\ 1$

หลังจากที่ได้ค่า G(x) มาแล้วให้นำมาเปรียบเทียบ คุณสมบัติดังนี้

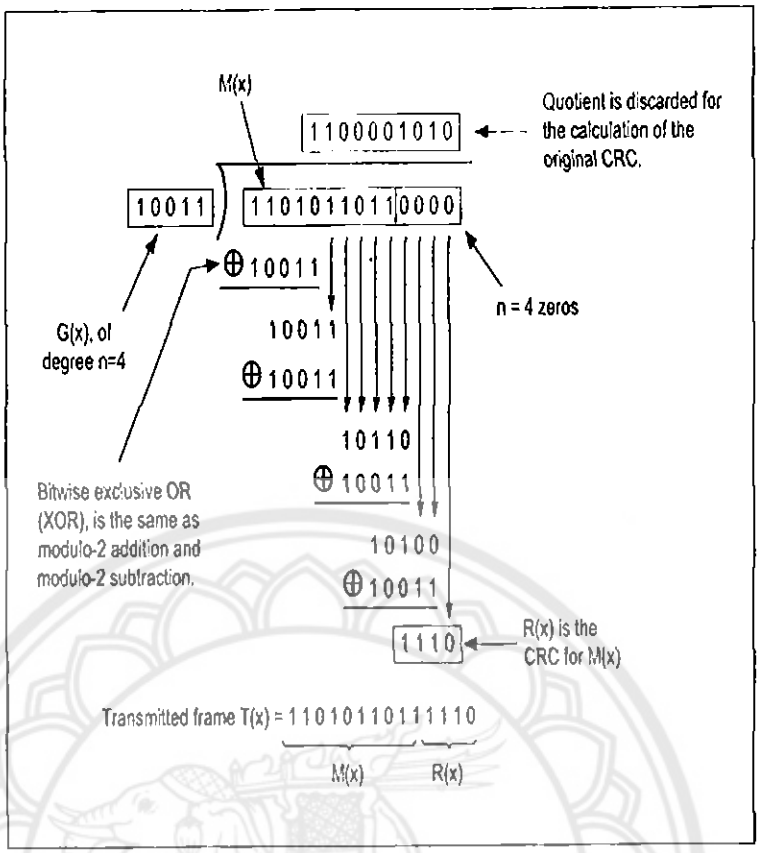
1.บิตทางซ้ายสุดและบิตทางขวาสุดของ G(x) มีค่าเป็น 1

2.รหัสโพลิโนเมียลจะต้องมีจำนวนบิตที่น้อยกว่าบิตของข้อมูล คือ จำนวนบิตของ G(x) มีค่าน้อยกว่าจำนวนบิตของ M(x)

3.ดีกรีของ G(x) เท่ากับ 4 ดังนั้น n จึงมีค่า เท่ากับ 4 โดยให้เพิ่มบิตศูนย์ จำนวน 4 บิต ต่อท้ายเฟรม M(x)

จะได้ 1 1 0 1 0 1 1 0 1 1 0 0 0 0

M(x) n

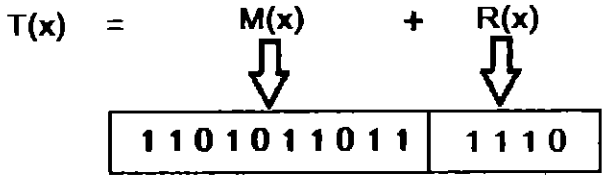


รูปที่ 3.5 รูปแบบการคำนวณ CRC

จากวิธีข้างต้น Frame ที่ถูกส่งไปยังปลายทางก็คือ

$T(x) = M(x) + R(x)$

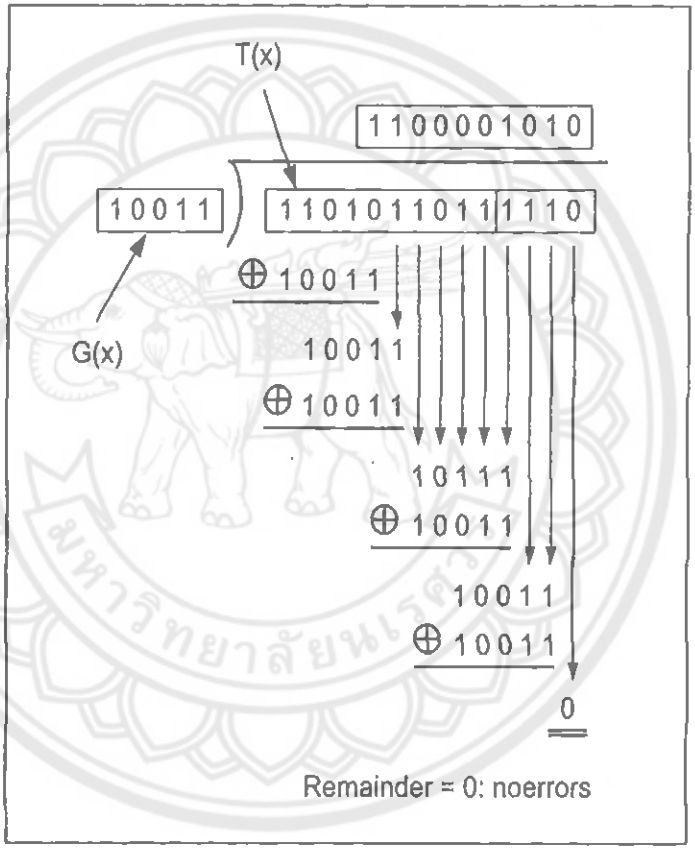
$= 11010110111110$



รูปที่ 3.6 แผนภาพแสดงเฟรมข้อมูลทั้งหมด

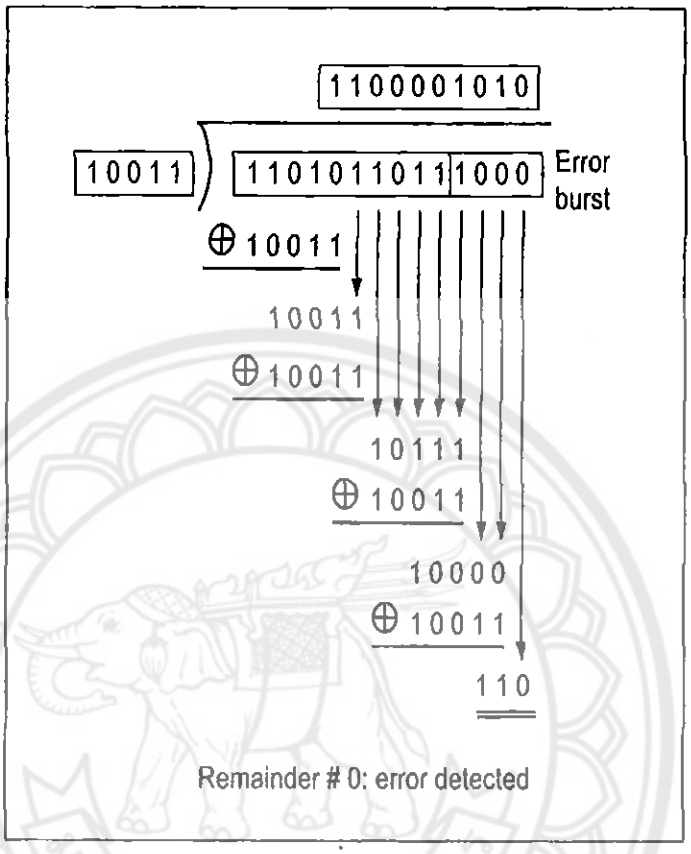
หลังจากที่ได้ทำการคำนวณ CRC เป็นที่เรียบร้อยแล้ว ฝ่ายส่งก็จะส่ง T(x) ไปยังปลายทาง เมื่อปลายทางได้รับ Frame T(x) ก็จะนำ T(x) ไปหารด้วยโพลิโนเมียล หรือ G(x) ที่เป็นรหัสเดียวกันกับฝ่ายผู้ส่ง โดยผลลัพธ์จากการคำนวณ T(x)/G(x) จะต้องหารลงตัว หรือมีเศษเป็นศูนย์เสมอ แสดงว่าตรวจจับไม่พบข้อผิดพลาด แต่ถ้าหากผลหารไม่เป็นศูนย์ แสดงว่า ตรวจพบข้อผิดพลาด

กรณีที่ตรวจไม่พบข้อผิดพลาด เนื่องจากเศษมีค่า เท่ากับ "ศูนย์"



รูปที่ 3.7 รูปแบบการคำนวณ CRC กรณีที่ตรวจไม่พบข้อผิดพลาด

กรณีที่ตรวจพบข้อผิดพลาด เนื่องจากเศษมีค่า ไม่เท่ากับ "ศูนย์"



รูปที่ 3.8 รูปแบบการคำนวณ CRC กรณีที่ตรวจพบข้อผิดพลาด

สำหรับในกรณีที่ฝ่ายรับตรวจจับข้อผิดพลาดพบ เช่น Frame ข้อมูลที่ส่งไปยังปลายทางได้มีการเปลี่ยนแปลงจำนวนบิต 2 บิตด้วยกันคือ 1 1 0 1 0 1 1 0 1 1 1 0 0 0 ผลจากการหารจะไม่ลงตัว นั่นหมายถึงพบ ข้อผิดพลาดแล้ว

3.2.3 ส่งค่า Setting จาก PC ไปยังบอร์ดไมโครคอนโทรลเลอร์ AVR

การที่จะสั่งการให้อุปกรณ์กล่องยามีการแจ้งเตือนเวลารับประทานอาหาร และเวลารับประทานยาเพื่อที่บอร์ดไมโครคอนโทรลเลอร์จะได้ทำงานตามเวลาที่ผู้ใช้กำหนด ต้องทำการตั้งค่าเวลาต่างๆผ่านทางการใช้งานหน้าต่าง VB โดยที่หน้าต่าง VB จะสามารถส่งค่าเวลาดังกล่าวไปยังบอร์ดไมโครคอนโทรลเลอร์ AVR ได้นั้น ทำได้โดยการใช้สาย RS-232 เชื่อมต่อระหว่าง COMPORT 1 ของ PC ต่อเข้ากับ Port UART0 ของบอร์ดไมโครคอนโทรลเลอร์ AVR โดยในการ

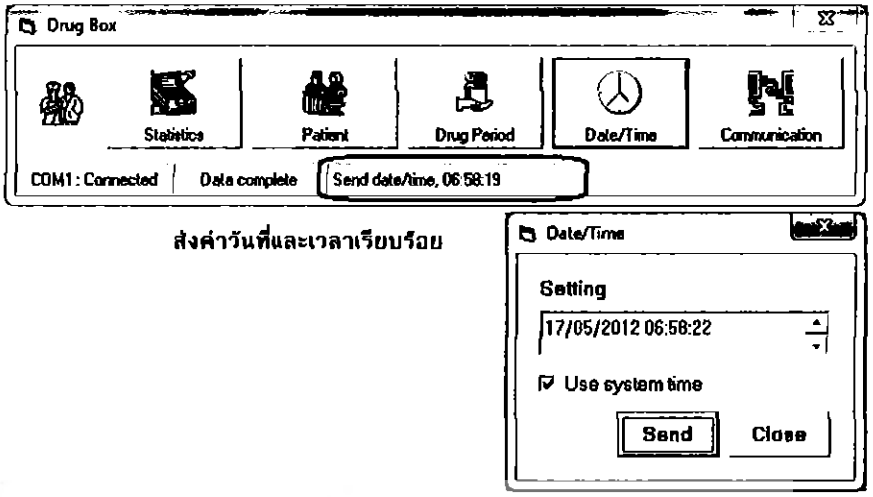
ส่งค่าต่างๆนั้น จะส่งโดยใช้รูปแบบ HDLC Protocol 9 byte (ไม่รวม data) ซึ่งประกอบด้วย byte start 1 byte, address destination 1 byte, command 1 byte, offset 1 byte, data size 2 bytes(byte high, byte low), data n byte, CRC 2 bytes (byte high, byte low) และ byte stop 1 byte ในการสื่อสารระหว่าง 2 อุปกรณ์นี้ และเมื่อถึงเวลาที่ตั้งค่าไว้ บอร์ดไมโครคอนโทรลเลอร์ AVR ก็จะทำการแจ้งเตือนรวมถึงบันทึกเวลาในการหยิบยาไปรับประทาน โดยเวลาที่บอร์ดไมโครคอนโทรลเลอร์ AVR จะทำงานนั้นอ้างอิงจากบอร์ด RTC(real time clock) ซึ่งคำสั่ง command = 0x01 คือคำสั่งตั้งค่าเวลาใน RTC 1307 ดังตารางที่ 3.3

ตารางที่ 3.3 คำสั่ง PC ส่งค่าตั้งเวลาให้ AVR [14]

Command	offset	data size	data		
0x01	0x01	12 byte	[YY 2 byte][MM 2 byte][DD 2 byte][hh 2 byte][nn 2 byte][ss 2 byte]		Ascii
			เช่น วันที่ 15/02/2012 8:00:50 ส่งเป็น '120215080050'		

3.2.4 บอร์ดไมโครคอนโทรลเลอร์ AVR ตอบกลับไปยัง PC

การที่บอร์ดไมโครคอนโทรลเลอร์ AVR จะตอบกลับไปยัง PC ก็ต่อเมื่อได้รับข้อมูล HDLC Protocol ที่ PC ส่งมาถูกต้องแล้ว จึงตอบกลับไปยัง PC เพื่อให้ต้นทางทราบว่าข้อมูล setting ที่ส่งมาได้รับถูกต้องตามที่ส่งค่ามา และถูกบันทึกเพื่อรอการทำงานตามเวลาที่ตั้งไว้แล้ว ดังตารางที่ 3.4 ซึ่งการที่จะทราบว่าค่าที่ส่งไปถูกต้องเรียบร้อยหรือไม่นั้นสามารถตรวจสอบได้จากการเชื่อมต่อระหว่างโปรแกรม HyperTerminal บนตัว PC เข้ากับ Port UART 1 ของบอร์ดไมโครคอนโทรลเลอร์ AVR ซึ่งเมื่อมีการรับ-ส่งค่าระหว่าง 2 อุปกรณ์ ที่หน้าจอ HyperTerminal ก็จะแสดงเป็น text file (แสดงแบบ real time) ให้ทราบว่าการรับ-ส่งข้อมูลเรียบร้อยหรือไม่ หรืออาจดูได้จากที่หน้าต่าง VB โดยเมื่อการรับ-ส่งค่า setting ต่างๆถูกต้องแล้ว ก็จะแสดงเวลา update ที่เวลา ณ ปัจจุบันขึ้นที่หน้าต่าง VB จึงทำให้ทราบว่าการส่งค่า setting ได้ถูกเก็บไว้ตามค่าที่ตั้งค่าไว้แล้ว ดังรูปที่ 3.9



รูปที่ 3.9 แผนภาพการรับส่งข้อมูลเรียบร้อย

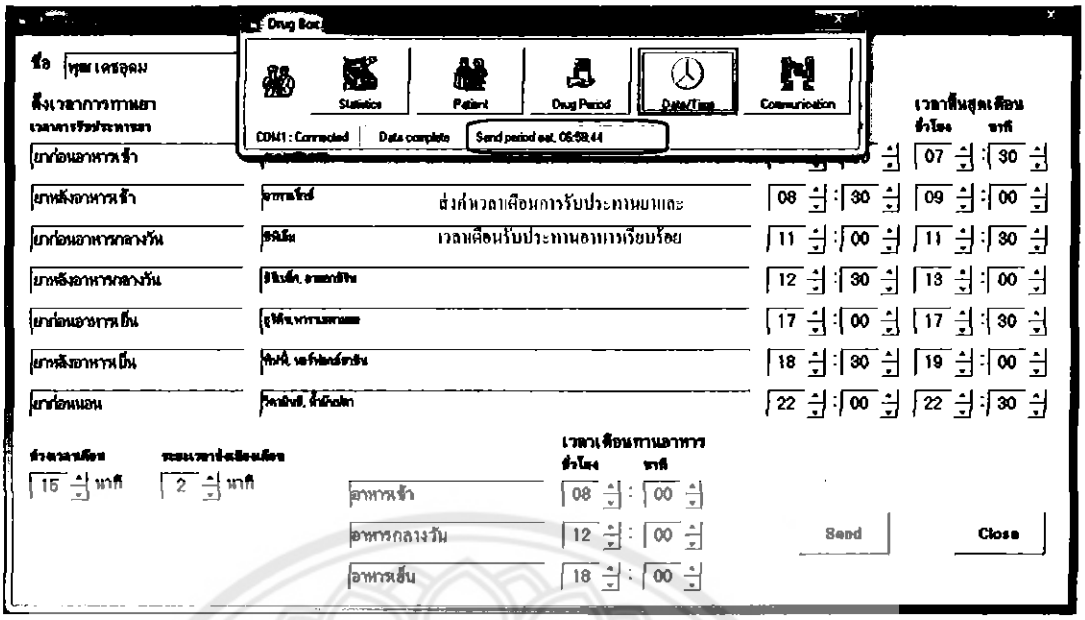
ตารางที่ 3.4 คำสั่ง AVR ตอบกลับให้ PC [14]

Command	offset	datasize	data		
0x01	0x01	2Byte	"OK"		Ascii

3.2.5 การส่งค่าตั้งเวลาเตือนรับประทานยาและเวลาเตือนรับประทานอาหาร

เมื่อตั้งค่าเวลาเตือนรับประทานยาและเวลาเตือนรับประทานอาหารผ่านทางหน้าต่างตั้งค่า VB แล้วส่งค่าเวลาเตือน โดยการกดปุ่ม send ค่าเวลาจะถูกส่งไปยังไมโครคอนโทรลเลอร์ AVR ที่หน้าต่างแอพพลิเคชั่นจะแสดงเวลา update ณ เวลาปัจจุบันที่ส่งค่าไป ดังรูปที่ 3.10 และรูปที่ 3.11 จึงทำให้ทราบว่าการส่งค่า setting ได้ถูกเก็บค่าเวลาไว้ใน SD-Card รอการประมวลผล และทำงานตามเวลาที่ตั้งไว้ เมื่อโปรแกรมทำงานจนครบ ข้อมูลการหยิบยาจะถูกบันทึกใน SD-Card ในรูปแบบวันที่ – เวลา และสถานะ "ทาน - ไม่ทาน"

ในการส่งค่า setting นั้นจะมีคำสั่งให้บันทึกค่าเวลาเริ่มต้นและเวลาสิ้นสุด โดยมีการตอบกลับกันระหว่าง PC กับ AVR โดยรูปแบบการรับ-ส่งค่าแสดงไว้ในตารางที่ 3.10, 3.11, 3.12 และตารางที่ 3.13



รูปที่ 3.10 แผนภาพการส่งค่าเวลาเตือนการรับประทานยาและการรับประทานอาหารเรียบร้อย

3.2.6 การบันทึกค่าและการอ่านค่าจากหน่วยความจำ SD-Card

หากต้องการเก็บข้อมูลไว้ในคอมพิวเตอร์แบบถาวร ก็จะต้องเก็บไว้ในลักษณะของไฟล์ข้อมูล ซึ่งไฟล์ข้อมูล จะมีรูปแบบการจัดเก็บข้อมูล (Format) แตกต่างกันไป แล้วแต่ชนิดของข้อมูลที่จัดเก็บ เช่น ไฟล์เอกสาร Microsoft Office, ไฟล์หนัง, ไฟล์เพลง, ไฟล์รูปภาพ, ไฟล์โปรแกรมต่างๆ, ไฟล์ฐานข้อมูล, ไฟล์ข้อมูลต่างๆ ของระบบปฏิบัติการ เป็นต้น Format ของไฟล์ก็คือการกำหนดว่าแต่ละ byte ข้อมูลในไฟล์แทนเนื้อหาข้อมูลอะไร ซึ่งเราสามารถแยกแยะได้จากนามสกุลของไฟล์ เช่น แต่ละ byte ของไฟล์ข้อความ (text file, .txt) จะแทนตัวอักษร 1 ตัว ทุกๆ 3 bytes ของไฟล์รูปภาพแบบ Bitmap (.bmp) จะแทนค่าสีของแต่ละจุดสีในรูปภาพ ซึ่งจะแทนด้วยรหัสสี 3 สี คือ Red/Green/Blue

disk_initialize(0) เป็นคำสั่งของโปรแกรมใน SD-Card ที่อยู่ในฟังก์ชันการทำงานของการเขียน Text file และฟังก์ชั่นการอ่าน Text file ซึ่งเริ่มแรกจะทำการเช็ค SD-Card ก่อนว่าใส่การ์ดหน่วยความจำหรือยัง ถ้ามีการ์ดหน่วยความจำแล้วก็จะทำการเขียนคำสั่งของโปรแกรมที่รับค่าช่วงวันที่และเวลาเข้ามาเก็บไว้ในรูปแบบ Text จากนั้นจะทำการอ่านค่า Text file และทำการส่ง Text file นี้ไปยังคอมพิวเตอร์ โดยจะอ่านค่าและส่งทีละ 512 byte จนครบไฟล์

3.2.7 การบันทึกข้อมูลลง SD card เป็นรูปแบบ Text file

Text Files เป็นไฟล์ที่เก็บข้อมูลตัวอักษร ซึ่งก็คือไฟล์ข้อความนั่นเอง โดยหากเก็บในรูปแบบของรหัส ASCII (หรือ ANSI) ข้อมูลแต่ละ byte จะแทนตัวอักษร 1 ตัว หากเก็บในรูปแบบของรหัส Unicode ข้อมูลทุกๆ 2 bytes จะแทนตัวอักษร 1 ตัว รูปแบบการเก็บตัวอักษรจะแตกต่างกันไปตามมาตรฐานรหัสที่ใช้เก็บ ซึ่งนอกเหนือจาก ASCII และ Unicode ก็ยังมีรหัสแบบ UTF-8 เป็นต้น

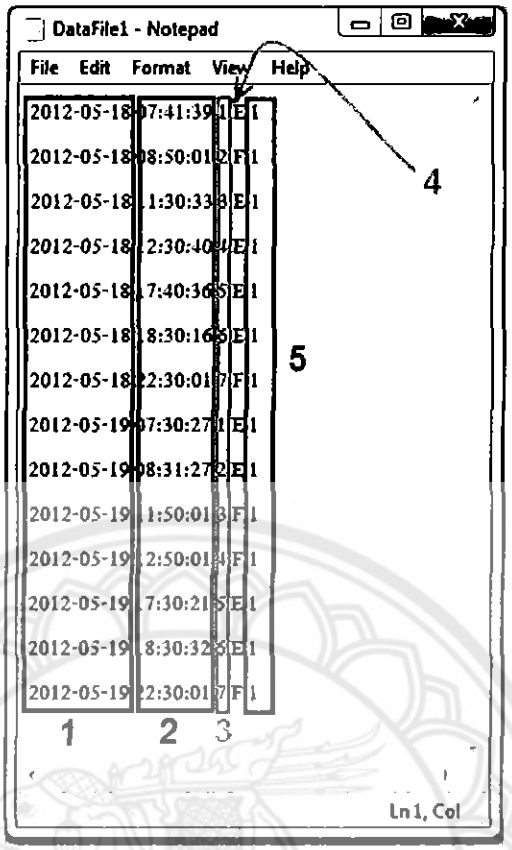
การทำงานกับ Text file มีการดำเนินการหลักๆ อยู่ 2 อย่างคือ

1. การอ่านข้อมูลจากไฟล์
2. การเขียนข้อมูลลงในไฟล์ •ฟังก์ชันที่ใช้เขียนโปรแกรมเพื่อดำเนินการกับไฟล์ จะมีอยู่ใน library ที่ชื่อ stdio.h ปกติเวลาเราจะใช้ฟังก์ชัน printf() หรือ scanf() เราก็จะสั่ง #include library นี้อยู่แล้ว

การจะทำงานกับไฟล์ จะมีขั้นตอนหลักๆ คือ

1. เปิดไฟล์ โดยระบุชื่อหรือ path ของไฟล์
2. ทำการอ่านข้อมูลจากไฟล์ หรือเขียนข้อมูลลงไปในไฟล์
3. ปิดไฟล์

เมื่อมีการใช้งานกล่องยา จะสามารถเก็บข้อมูลและดูสถิติการรับประทานยาของผู้ป่วยได้ โดยการเรียกข้อมูลจาก SD-Card มาเก็บไว้ในโฟลเดอร์ Receive ชื่อไฟล์ Datafile(n).txt เมื่อ n=1,2,3,... ซึ่งข้อมูลที่บันทึกอยู่ใน SD-Card จะอยู่ในรูปแบบ Text file โดยจะแสดงรายละเอียดของวันที่, เวลาที่รับประทานยา, ช่องยา, สถานะ (ทาน – ไม่ทาน) และไอดีผู้ป่วย ตามลำดับ ดังรูปที่ 3.11



รูปที่ 3.11 แผนภาพรูปแบบ Text file แสดงสถานะการรับประทานยา

หมายเลข 1 คือ วัน-เดือน-ปี

หมายเลข 2 คือ เวลาที่ทานยาหรือไม่ได้ทานยา แสดงเป็น ชั่วโมง, นาที, วินาที

หมายเลข 3 คือ ลำดับช่องยา

หมายเลข 4 คือ สถานะการรับประทานยา โดยที่ "E" หมายถึง สถานะการทานยา และ "F" หมายถึง สถานะไม่ทานยา

หมายเลข 5 คือ ไอดีผู้ป่วย (ชื่อผู้ป่วย)

3.2.8 การนำข้อมูลมาแสดงผล

ในการดึงข้อมูลมาแสดงผลที่หน้าต่าง VB นั้น จะมีปุ่มที่ใช้สำหรับเรียกค่า Text file จาก SD-Card ขึ้นมา โดยการกดปุ่ม Getfile จะทำการดึงค่าข้อมูลจาก Text file มาเก็บไว้ในโฟลเดอร์ Receive พร้อมทั้งสามารถกด save ได้ จากนั้นถ้าจะดูรายละเอียดการรับประทานยาทางหน้าต่างแสดงผล (หน้าต่าง Statistic) สามารถค้นหาได้โดยกดเลือกชื่อผู้ป่วย และช่วงวันที่ที่ต้องการ แล้ว

กดปุ่ม Search เพื่อค้นหาข้อมูลต่างๆ ซึ่งข้อมูลนั้นจะถูกเรียกจากใน DataBase ออกมาแสดง หลักการในการรับส่งข้อมูลใน Text file ระหว่าง PC กับ AVR สามารถดูได้จากตารางที่ 3.5, 3.6 และ ตารางที่ 3.7 ซึ่งเป็นการทำงานตามคำสั่ง command นั้นๆ ตามรูปแบบการรับส่งแบบโปรโตคอล HDLC

ตารางที่ 3.5 คำสั่ง PC อ่าน Text file จาก AVR [14]

Command	ofset	data size	Data		
0x03	0x01	9 byte	"READ TEXT"		Ascii

ตารางที่ 3.6 คำสั่ง AVR ส่ง Text file ให้ PC [14]

Command	offset	data size	Data			
0x03	0x01	524 byte	[ลำดับ Packet 4 bytes][จำนวน Packet ทั้งหมด 4 bytes][DataSize 4 byte][data ของ Text file 512 byte]	ลำดับ Packet	"0"-"9999"	Ascii
				จำนวน Packet ทั้งหมด	"0"-"9999"	Ascii
			ตัวอย่างเช่น Text file ขนาด 1200 bytes จะแบ่งเป็น Packet ละ 512 byte จะได้จำนวน packet ทั้งหมดดังนี้	DataSize ของแต่ละ Packet	"0"-"9999"	Ascii
			1200 / 512 = 3 packet โดยรวม Packet ที่เป็น เศษด้วย			
			โดย ลำดับ Packetที่ 1 คือ 512 byte แรก	data ของ Text file 512 byte		Ascii
			ลำดับ Packetที่ 2 คือ 512 byte ต่อจาก Packetที่ 1			
			ลำดับ Packetที่ 3 คือ 176 byte ต่อจาก Packetที่ 2			

ตารางที่ 3.6 คำสั่ง AVR ส่ง Text file ให้ PC (ต่อ) [14]

			ดังนั้น Data แต่ละการส่งจะส่งทีละ Packet ดังนี้			
			Packet ที่ 1			
			["0000"]["0003"]["0512"][data จำนวน 512 bytes]			
			Packet ที่ 2			
			["0001"]["0003"]["0512"][data จำนวน 512 bytes]			
			Packet ที่ 3			
			["0002"]["0003"]["0176"][data จำนวน 176 bytes]			

ในการส่งแต่ละ Packet PC จะต้องตอบ Command ดังนี้

ตารางที่ 3.7 คำสั่ง PC ตอบกลับ Command 0x03 [14]

Command	offset	data size	data	ลำดับ Packet	"0"-"9999"	Ascii
0x03	0x02	8 byte	[ลำดับ Packet 4 bytes][จำนวน Packet ทั้งหมด 4 bytes]	จำนวน Packet ทั้งหมด	"0"-"9999"	Ascii

3.2.9 การลบข้อมูลใน SD-Card

ในการลบข้อมูลใน SD-Card นั้นโดยการกดปุ่ม Delete file จะทำการลบข้อมูล Text file ที่อยู่ใน SD-Card ทั้งหมด ทำให้ไม่สามารถเรียกข้อมูลมาเก็บไว้ใน DataBase ได้ แต่ข้อมูลเก่าที่อยู่ใน DataBase ยังคงมีอยู่จนกว่าจะมีข้อมูลใหม่มาแทนที่ ซึ่งคำสั่งการส่งค่าและตอบกลับระหว่าง PC กับ AVR ของการ Delete Text file แสดงได้ดังตารางที่ 3.8 และตารางที่ 3.9

ตารางที่ 3.8 คำสั่ง PC ส่งไป Delete Text file [14]

Command	offset	data size	data		
0x04	0x01	11 byte	"DELETE TEXT"		Ascii

ตารางที่ 3.9 คำสั่ง AVR ตอบกลับ Delete Text file ให้ PC [14]

Command	offset	data size	data		
0x04	0x01	15 byte	"DELETE COMPLETE"		Ascii

คำสั่ง Command = 0x05 คือคำสั่งให้บันทึกค่าเวลาเริ่มต้นและเวลาสิ้นสุด ทั้ง 7 ช่วง interval และระยะเวลาที่เสียงดังที่รับจาก computer ดังตารางที่ 3.10

ตารางที่ 3.10 คำสั่ง PC ส่งค่า setting เวลารับประทานยา [14]

Command	offset	data size	data		
0x05	0x01	88 byte	1.(เวลาเริ่ม 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte](วันที่เวลาหยุด 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii
			2.(เวลาเริ่ม 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte](วันที่เวลาหยุด 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii
			3.(เวลาเริ่ม 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte](วันที่เวลาหยุด 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii
			4.(เวลาเริ่ม 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte](วันที่เวลาหยุด 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii

ตารางที่ 3.10 คำสั่ง PC ส่งค่า setting เวลารับประทานยา (ต่อ) [14]

			5.(เวลาเริ่ม 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte](วันที่เวลาหยุด 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii
			6.(เวลาเริ่ม 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte](วันที่เวลาหยุด 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii
			7.(เวลาเริ่ม 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte](วันที่เวลาหยุด 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii
			[ช่วงเวลาการเตือน 2 bytes]	ช่วงเวลาการเตือน = "00"-"59"	Ascii
			[ระยะเวลาส่งเสียงเตือน 2 bytes]	ระยะเวลาส่งเสียงเตือน = "00"-"59"	Ascii

ตารางที่ 3.11 คำสั่ง AVR ตอบกลับให้ PC [14]

(Command)	offset	datasize	data		
0x05	0x00	2byte	"01"		Ascii

คำสั่ง command = 0x06 คือคำสั่งให้บันทึกค่าเวลาในการรับประทานอาหารทั้ง 3 เวลา ที่รับจาก computer ดังตารางที่ 3.12

ตารางที่ 3.12 คำสั่ง PC ส่งค่า setting เวลารับประทานอาหาร[14]

Command	offset	data size	data		Ascii
0x06	0x01	18 byte	(เวลาเช้า 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii
			(เวลากลางวัน 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii
			(เวลาเย็น 6 byte) = [hh 2 byte][nn 2 byte][ss 2 byte]		Ascii

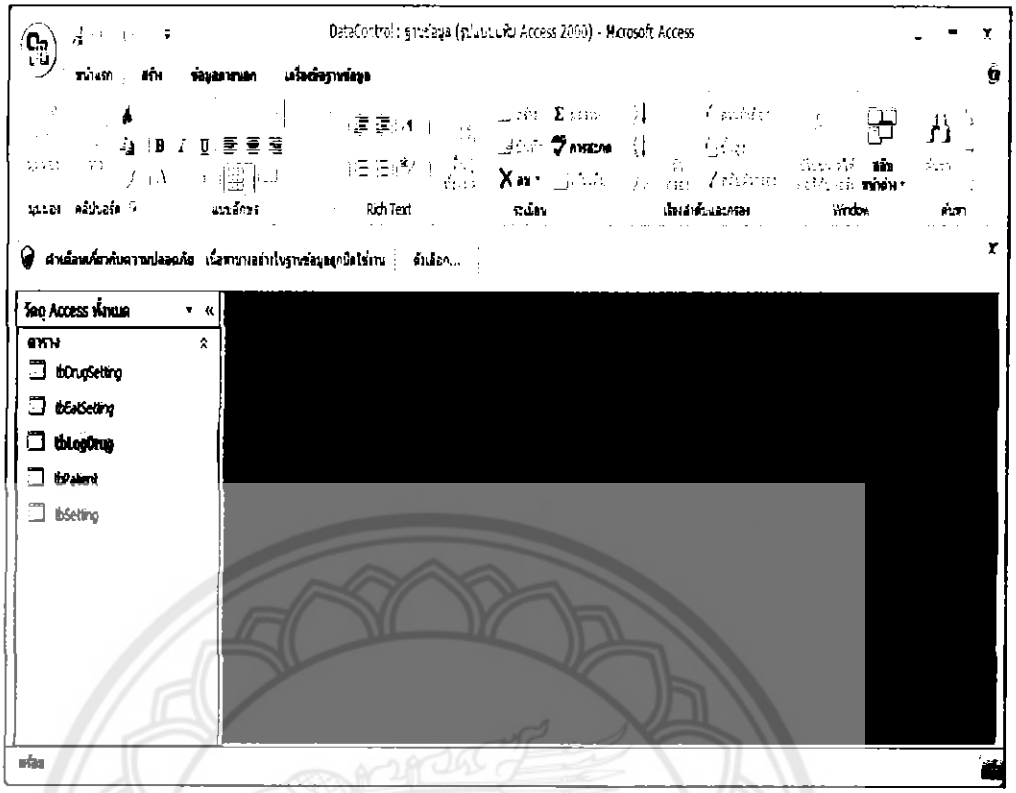
ตารางที่ 3.13 คำสั่ง AVR ตอบกลับให้ PC [14]

Command	offset	datasize	data
0x06	0x01	28byte	"OK"

3.2.10 ฐานข้อมูลผู้ป่วย

ฐานข้อมูลผู้ป่วยนั้น จะถูกเก็บไว้ที่ database บนอุปกรณ์ PC เพื่อใช้เก็บเวลาที่ต้องรับประทานยาในมื้อต่างๆ ชื่อ-นามสกุลผู้ป่วย เวลาที่ผู้ป่วยรับประทานยา รวมถึงชนิดยาที่ผู้ป่วยรับประทานด้วย ซึ่งข้อมูลที่ถูกเก็บอยู่ที่ database นั้นรับค่ามาจากหน่วยความจำ SD-Card ผ่านการเชื่อมต่อสาย RS-232 ระหว่าง PC กับ บอร์ดไมโครคอนโทรลเลอร์ โดยเมื่อเราต้องการใช้หน้าต่าง VB เพื่อปริ้นหรือหาสถิติการรับประทานยาของผู้ป่วย โปรแกรมจะค้นหาช่วงเวลาที่เราต้องการจากฐานข้อมูลผู้ป่วยซึ่งถูกเก็บไว้ที่ database แล้วนำมาแสดงที่หน้าต่างแสดงผล VB

สามารถดูเข้าดูข้อมูลได้ในโฟลเดอร์ VB -> DataBase -> DataControl -> รหัสผ่านฐานข้อมูล (software007) -> ตกลง จากนั้นจะปรากฏรูปแบบฐานข้อมูลดังรูปที่ 3.12

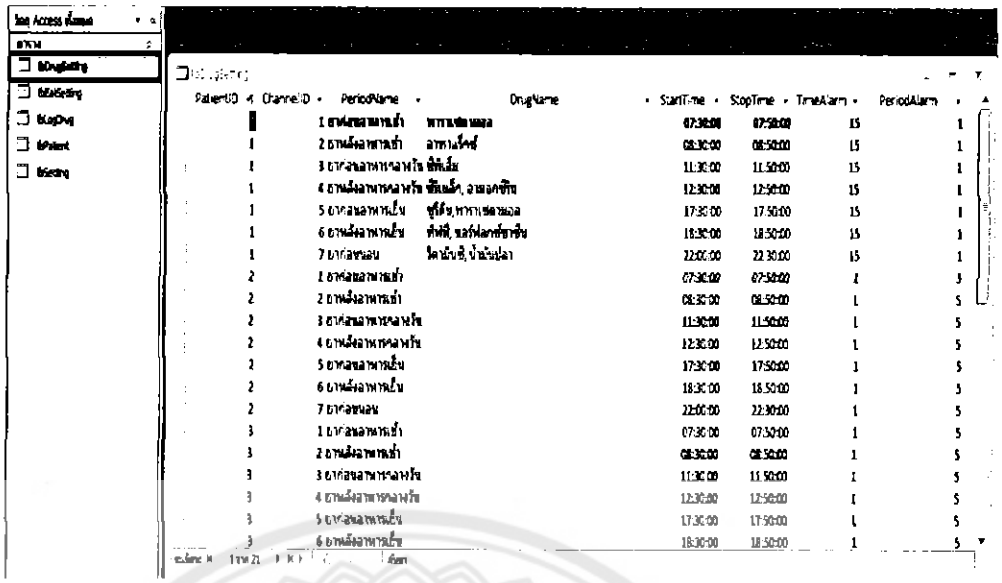


รูปที่ 3.12 แผนภาพรูปแบบฐานข้อมูลผู้ป่วย

ในฐานข้อมูลนี้จะประกอบไปด้วยตารางต่างๆ ดังต่อไปนี้

3.2.10.1 ตารางฐานข้อมูลการตั้งค่าเวลาการรับประทานยา

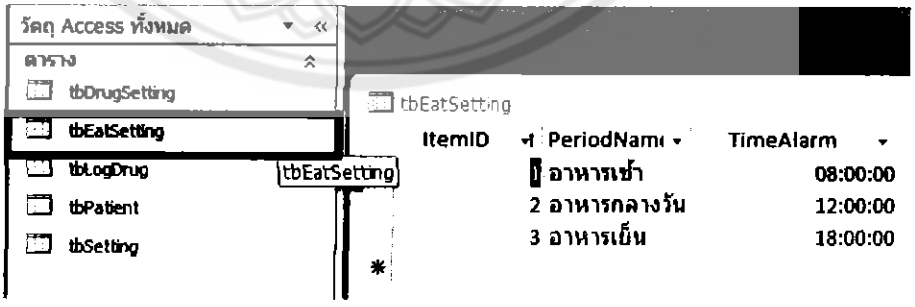
ในส่วนของตาราง tbDrugsetting ดังรูปที่ 3.13 จะแสดงรายละเอียด PatientID (ไอดีผู้ป่วย), ChannelID (ช่องยา), PeriodName (ชื่อมื้อยา), DrugName (ชื่อยา), StartTime (เวลาเริ่มเตือน), StopTime (เวลาสิ้นสุดเตือน), TimeAlam (ช่วงเวลาเตือน) ข้อมูลดังกล่าวถูกส่งมาจากหน้าต่างตั้งค่าเวลารับประทานยา เก็บไว้ใน DataBase และส่งค่าเวลาเตือนไปยังบอร์ดไมโครคอนโทรลเลอร์



รูปที่ 3.13 แผนภาพตารางฐานข้อมูลผู้ป่วยของการตั้งค่าเวลารับประทานยา

3.2.10.2 ตารางฐานข้อมูลการตั้งค่าเวลาการรับประทานอาหาร

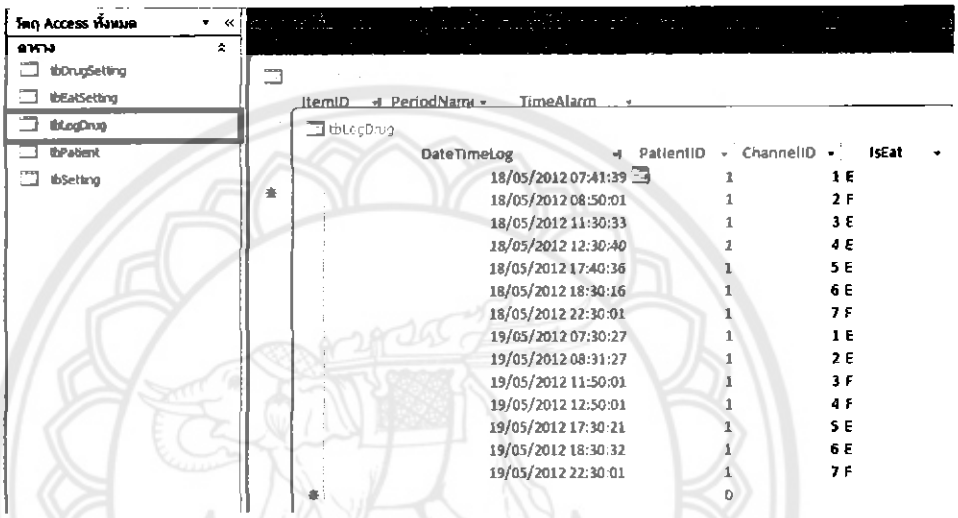
ในส่วนของตาราง tbEatsetting ดังรูปที่ 3.14 จะแสดงรายละเอียด ItemID (ช่องรับประทานอาหาร), PeriodName (ชื่อมื้ออาหาร), TimeAlam (เวลาเตือน) ข้อมูลดังกล่าวถูกส่งมาจากหน้าต่างตั้งค่าเวลารับประทานอาหารมาเก็บไว้ใน DataBase และส่งค่าเวลาเตือนไปยังบอร์ดไมโครคอนโทรลเลอร์



รูปที่ 3.14 แผนภาพตารางฐานข้อมูลผู้ป่วยของการตั้งค่าเวลารับประทานอาหาร

3.2.10.3 ตารางฐานข้อมูลเวลาการรับประทานยา

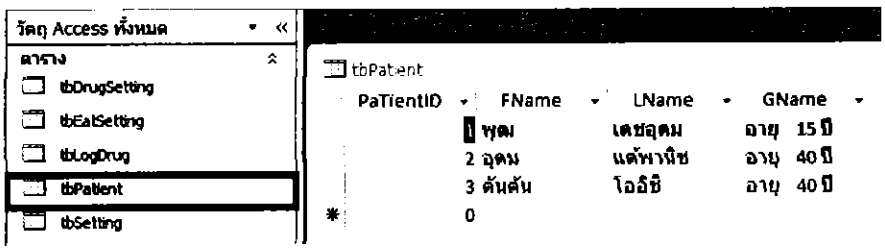
ในส่วนของตาราง tbLogDrug ดังรูปที่ 3.15 จะแสดงรายละเอียด DateTimeLog (เวลารับประทานยา), PatientID (ไอดีผู้ป่วย), ChannelID (ช่องยา), IsEat (สถานะ ทาน-ไม่ทาน) ในส่วนนี้จะเป็นการนำข้อมูลที่อยู่ใน SD-Card ในรูปแบบ Text file มาเก็บไว้ใน DataBase เพื่อรอให้หน้าต่างแสดงผลเรียกดูข้อมูล



รูปที่ 3.15 แผนภาพตารางฐานข้อมูลผู้ป่วยการรับประทานยา

3.2.10.4 ตารางฐานข้อมูลการตั้งชื่อผู้ป่วย

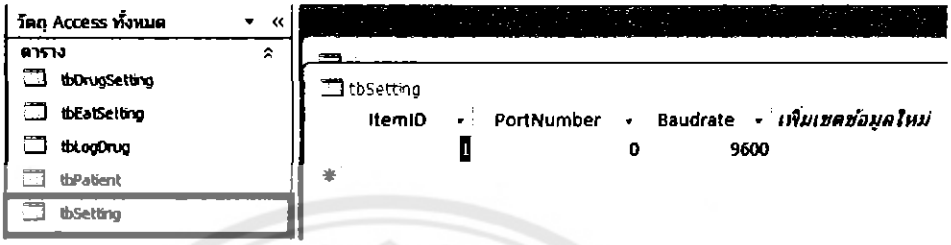
ในส่วนของตาราง tbPatient ดังรูปที่ 3.16 จะแสดงรายละเอียด PatientID (ลำดับชื่อ), FName (ชื่อ), LName (นามสกุล), GName (รายละเอียด) ข้อมูลดังกล่าวถูกส่งมาจากหน้าต่างตั้งชื่อมาเก็บไว้ใน DataBase เพื่อบันทึกประวัติผู้ป่วยเก็บไว้



รูปที่ 3.16 แผนภาพตารางฐานข้อมูลผู้ป่วย

3.2.10.5 ตารางฐานข้อมูลการตั้งค่าการติดต่อสื่อสาร

ในส่วนของตาราง tbSetting ดังรูปที่ 3.17 เป็นส่วนตั้งค่าการติดต่อสื่อสาร จะแสดงรายละเอียด Baudrate และ Port ที่ตั้งค่าผ่านหน้าต่างการติดต่อสื่อสารมาเก็บไว้



รูปที่ 3.17 แผนภาพตารางฐานข้อมูลผู้ป่วยของการตั้งค่าการติดต่อสื่อสาร

3.3 การออกแบบ

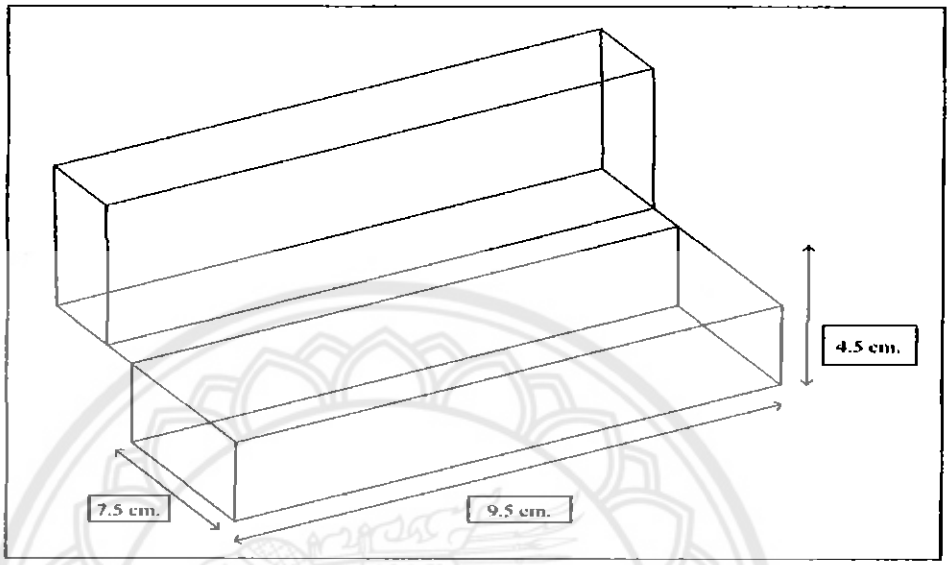
จากการที่ได้ศึกษาหลักการต่างๆเกี่ยวกับการออกแบบอุปกรณ์ การทำงานของบอร์ดไมโครคอนโทรลเลอร์ และการใช้โปรแกรม วิชวลเบสิก สามารถแยกการทำงานออกเป็นสองส่วนคือ ฮาร์ดแวร์ และซอฟแวร์ คือ

3.3.1 การออกแบบฮาร์ดแวร์อุปกรณ์กล่องยา

ในการออกแบบและพัฒนาฮาร์ดแวร์ของกล่องยานี้ได้มีแนวคิดมาจากงานวิจัยเรื่องการศึกษาและพัฒนากล่องยาอิเล็กทรอนิกส์สำหรับยาชนิดเม็ดของผู้สูงอายุที่มีโรคประจำตัว[14] คือ กล่องยานั้นควรมีสีสันสะอาดตา มีขนาดที่เหมาะสม กล่องยาควรใส่ยาได้หลาย ๆ ช่อง และมีการแบ่งช่องใส่ยาที่ชัดเจนในแต่ละมื้ออาหาร และบนตัวกล่องยาควรมีแสงไฟแสดง จากการศึกษาการจ่ายยาทางการแพทย์แล้ว พบว่าส่วนใหญ่จะมีการรับประทานยา 4 เวลา คือ เช้า กลางวัน เย็น และก่อนนอน จึงได้ออกแบบกล่องยาที่สามารถบรรจุยาในแต่ละวันได้วันละ 4 มื้อ ซึ่งในแต่ละมื้ออาหารจะมีช่วงเวลาก่อนรับประทานอาหาร และหลังรับประทานอาหาร โดยมีรายละเอียดการออกแบบกล่องยาและหลักการออกแบบกล่องยาดังต่อไปนี้

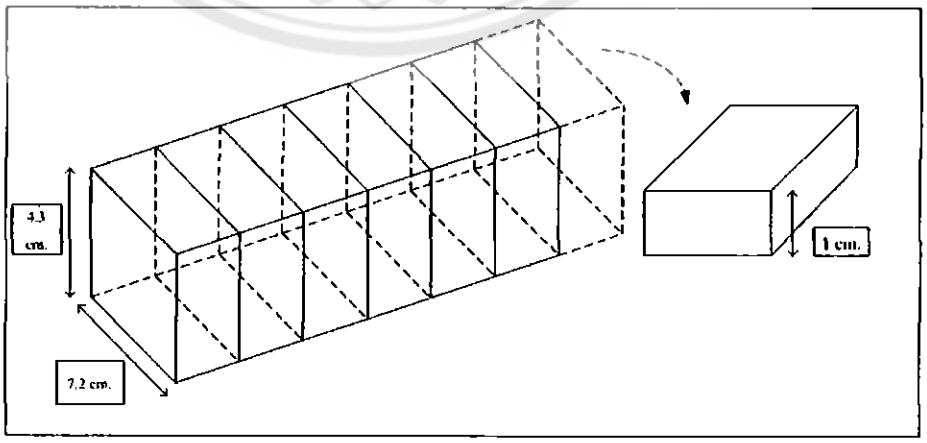
ในการออกแบบตัวกล่องยา และฐานกล่องบรรจุบอร์ดไมโครคอลโทรลเลอร์และอุปกรณ์อื่นๆ วัสดุที่เลือกใช้คือแผ่นอะครีลิคใส ความหนาขนาด 1.5 mm ส่วนประกอบหลักของกล่องยา คือ ตัวฐานของกล่อง และกล่องบรรจุยาด้านนอก

ตัวกล่องบรรจุยาด้านนอกมีลักษณะที่ปิด-เปิด เป็นบานพับ (Hinge) แบบบานประตู สามารถบรรจุกล่องยาย่อยได้อีก ดังรูปที่ 3.18



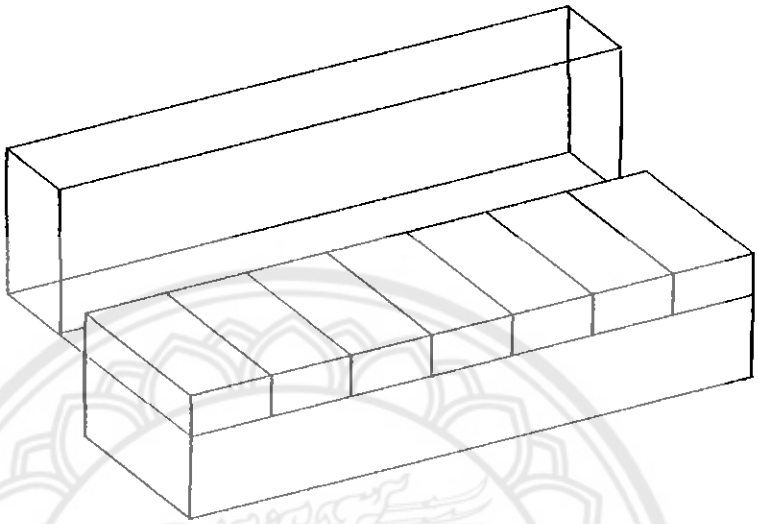
รูปที่ 3.18 กล่องบรรจุยาด้านนอก

สำหรับตัวกล่องยาจะมีจำนวนกล่องใส่ยาทั้งหมด 7 กล่อง เพื่อให้เพียงพอสำหรับรับประทานยาตามจำนวนมื้อในแต่ละวัน สามารถดึงแยกออกมาได้ และกล่องยานี้จะบรรจุยาได้ในหนึ่งวัน ดังรูปที่ 3.19

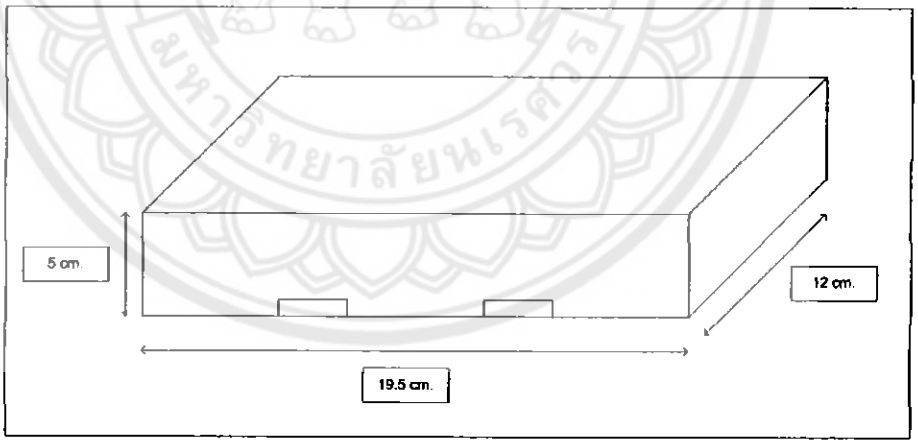


รูปที่ 3.19 กล่องบรรจุเม็ดยาจำนวน 7 กล่อง

นำกล่องบรรจุเม็ดยาจำนวน 7 กล่องใส่ในตัวกล่องยาด้านนนอกเพื่อให้สะดวกในการปรับเปลี่ยนยาในแต่ละวัน โดยสามารถจัดยาไว้เป็นชุดๆของวันต่อวันได้ ดังรูปที่ 3.20



รูปที่ 3.20 การประกอบกล่องบรรจุเม็ดยากับกล่องยาด้านนอก



รูปที่ 3.21 ฐานสำหรับบอร์ดไมโครคอนโทรลเลอร์และอุปกรณ์ที่ใช้งาน

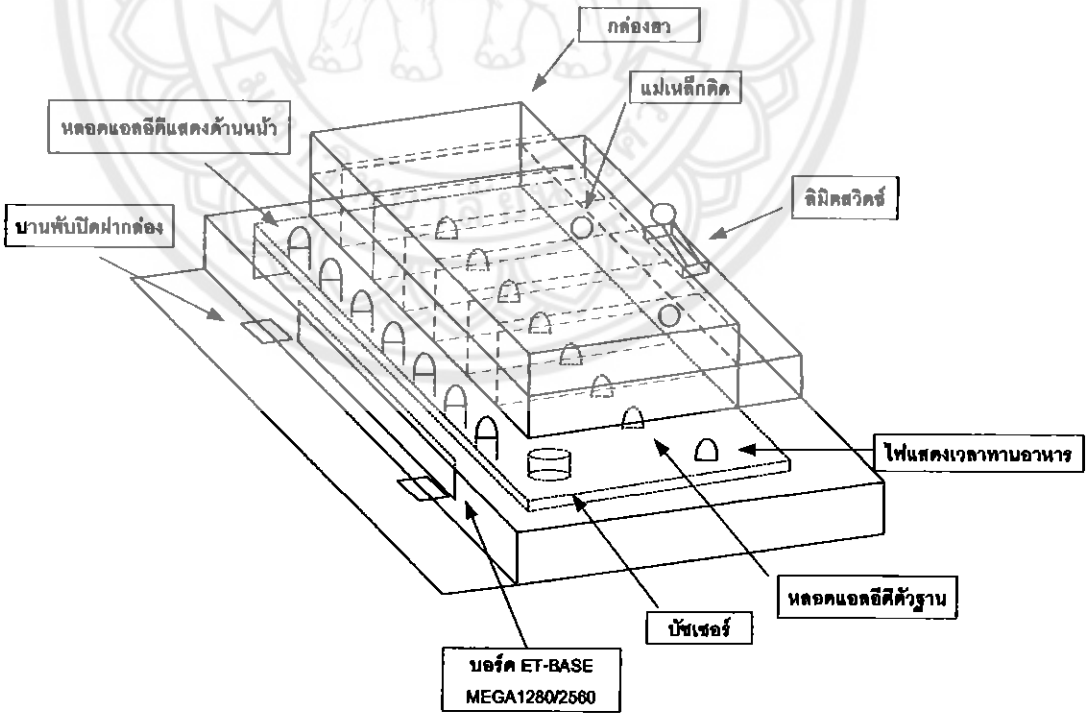
เพื่อให้ไม่สับสนในการหยิบยารับประทานในแต่ละมื้อ ที่ฐานกล่องยาจะมีหลอดไฟแอลอีดีเป็นสีที่มองเห็นได้ชัดเจนกระพริบที่ด้านหน้าช่องยาและใต้ช่องยาที่รับประทาน ดังรูปที่ 3.22 พร้อมกับเสียงเตือน ซึ่งสามารถตั้งเวลาและมีเสียงเตือนเมื่อถึงเวลารับประทานยา เสียงเตือนนี้จะดังต่อเนื่องเป็นระยะเวลานานตามที่กำหนด จนกว่าจะเปิดกล่องยา แล้วเสียงจึงหยุด ถ้าหากไม่

มีการเปิดกล่องยาในช่วงระยะเวลานั้นๆ ก็จะมีเสียงเตือนอีกครั้งในทุกๆ ช่วงเวลาที่กำหนด เช่น ทุกๆ 10 นาที, 15 นาที, 30 นาที จนหมดช่วงเวลารับประทานยาในมื้อนั้น แล้วจะมีการเตือนใหม่อีกครั้งในมื้อการรับประทานยาถัดไป

นอกจากนี้ยังมีสัญญาณเตือนเวลาในการรับประทานอาหารในแต่ละมื้ออีกด้วย คือ เมื่อถึงเวลาทานอาหารในมื้อเช้า กลางวัน และเย็น จะต้องมีสัญญาณเตือนให้ทราบว่า ถึงเวลารับประทานอาหารในแต่ละมื้อนั้นๆแล้ว ซึ่งสามารถกำหนดตำแหน่งการติดตั้งอุปกรณ์ได้ดังนี้

3.3.2 การติดตั้งอุปกรณ์บอร์ดไมโครคอนโทรลเลอร์

ในการติดตั้งบอร์ดไมโครคอนโทรลเลอร์ จะนำเอาตัวบอร์ดไปติดตั้งเข้ากับฐานที่ออกแบบมา ซึ่งฐานของกล่องยามีขนาดใหญ่กว่าขนาดของกล่องบรรจุเม็ดยาเล็กน้อย (เพื่อติดหลอดไฟแอลอีดี) หลังจากนั้นนำฐานที่ติดตั้งบอร์ดเรียบร้อยแล้ว ไปประกอบเข้ากับกล่องยาในลักษณะที่ตัวกล่องบรรจุยาอยู่ด้านบนและฐานของบอร์ดอยู่ด้านล่าง ซึ่งบอร์ดไมโครคอนโทรลเลอร์จะรับคำสั่งจากการตั้งค่าผ่านหน้าต่าง วิชวลเบสิก เพื่อควบคุมการทำงานของอุปกรณ์ตัวอื่นๆในตัวชิ้นงาน



รูปที่ 3.22 การติดตั้งอุปกรณ์และบอร์ดไมโครคอนโทรเลอร์

3.3.3 การติดตั้งอุปกรณ์ ลิมิตสวิตช์

ในการติดตั้งอุปกรณ์ ลิมิตสวิตซ์ จะติดตั้งไว้ด้านหลัง โดยเมื่อทำการเปิดกล่องยา ฝากล่องจะไปกดสวิตช์ แล้วทำให้สัญญาณเตือนหยุด เป็นการรับค่าจากการกดไปแสดงที่คอมพิวเตอร์ จากนั้นโปรแกรมจะส่งผลการเปิดฝากล่องหยิบยาไปทาน บันทึกลงใน Micro SD card

3.3.4 การติดตั้ง ET-MINI DS1307

ในการติดตั้งบอร์ด ET-MINI DS1307 จะนำไปเชื่อมต่อกับพอร์ตของบอร์ดไมโครคอนโทรเลอร์ จะมีขั้วต่อ SDA และ SCL สำหรับต่อเข้ากับขา SDA และ SCL ของ MCU ตามลำดับ ส่วนขั้วต่อ INT จะต่อเข้ากับ MCU ก็ต่อเมื่อต้องการใช้งาน Interrupt โดยให้ต่อเข้ากับขาของ MCU ที่ผู้ใช้ได้กำหนดให้มีการรับสัญญาณ INT. จากภายนอกไว้ ส่วนไฟเลี้ยงโมดูลจะอยู่ที่ VDC 3V-5V I^2C เบอร์นี้จะมี Control Byte อยู่ที่ "1101000x " นอกจากนี้ก็จะมีในส่วนของที่สำหรับใส่แบตเตอรี่เพื่อใช้ในการ Back Up ฐานเวลา ให้นาฬิกายังคงเดินได้อย่างถูกต้อง เมื่อไม่มีการจ่ายไฟเลี้ยงให้กับ Module

3.3.5 การติดตั้งหลอดไฟแอลอีดี

ในการติดตั้งหลอดไฟแอลอีดีนั้น เราจะติดตั้งไว้ด้านหน้ากล่องบรรจุยาและใต้กล่องบรรจุยา และเมื่อบอร์ดไมโครคอนโทรลเลอร์รับคำสั่งจากหน้าต่างวิชวลเบสิก ในการแจ้งเตือนเวลารับประทานยา หลอดไฟแอลอีดีก็จะส่องสว่างขึ้นบริเวณด้านช่องยาที่เราต้องรับประทาน และเพื่อความชัดเจนในการมองเห็น รวมถึงป้องกันความผิดพลาดในการหยิบยาที่จะรับประทานผิดที่ เราจึงติดตั้งหลอดไฟแอลอีดีไว้ที่ใต้ช่องยาด้วย

3.3.6 การติดตั้งลำโพง

ในการติดตั้งลำโพงเข้ากับชิ้นงาน จะติดตั้งไว้บริเวณข้างในฐานของบอร์ด ซึ่งคือบริเวณเดียวกับที่ติดตั้งบอร์ดไมโครคอนโทรลเลอร์และบอร์ด Timer การที่ติดตั้งลำโพงให้ชิ้นงานก็เพื่อที่จะส่งเสียงเตือนเมื่อถึงเวลาที่ต้องรับประทานยา ซึ่งความต่อเนื่องกับระยะเวลาในการส่งเสียงเตือนจะควบคุมโดยบอร์ดไมโครคอนโทรลเลอร์

3.4 การทำงานของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ AVR มีสถาปัตยกรรมแบบ RISC คือหนึ่งคำสั่งทำงานใช้สัญญาณนาฬิกาเพียง 1 ลูก โปรแกรมข้อมูลได้แบบ ISP (In-System Programming) เขียนโปรแกรมด้วยภาษา C ด้วย AVR Studio 4 / WinAVR เพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ AVR โดยหลักการทำงานคือ รับการโปรแกรม Code ภาษา C จาก PC ผ่านเครื่องโปรแกรม ET-AVRISP mkII ซึ่งเป็นการโปรแกรมข้อมูลแบบ ISP

3.4.1 การรับส่งข้อมูลอนุกรมของไมโครคอนโทรลเลอร์ AVR

การสื่อสารข้อมูลโดยใช้ไมโครคอนโทลเลอร์ AVR สามารถทำการเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์ได้ ในแบบการสื่อสารแบบ UART (Universal Asynchronous Receiver Transimtter) ดังที่กล่าวมา

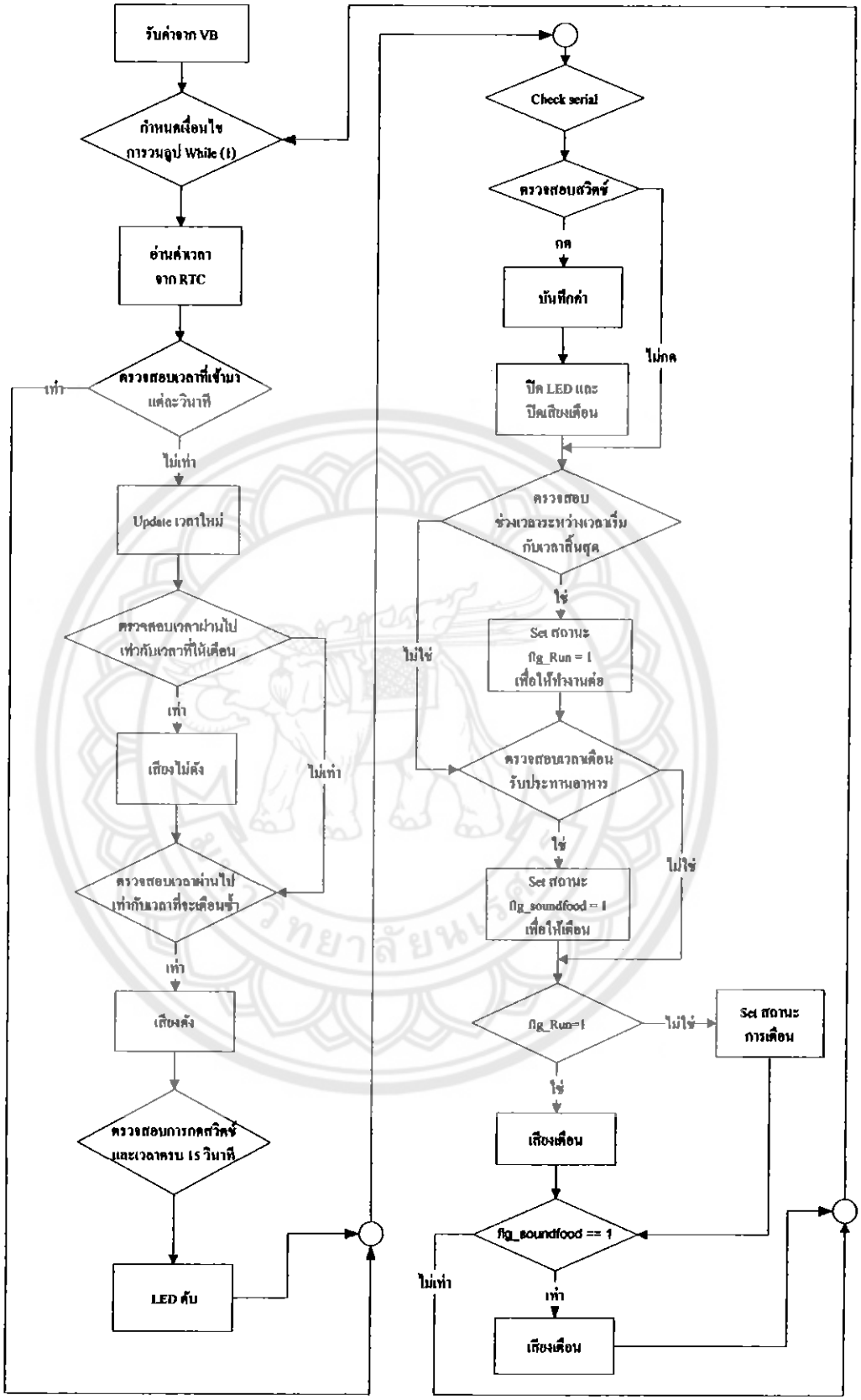
ในการเขียนโปรแกรมในการรับส่งข้อมูลมีหลักการดังนี้

1. วงจรเชื่อมต่อภายนอกเพื่อแปลงระดับแรงดัน RS232 ให้เป็นระดับสัญญาณที่เป็นไปตามข้อกำหนดลักษณะสัญญาณลอจิก (Time To Live, TTL) ซึ่งเป็นระดับสัญญาณที่มีแค่ +5 กับ 0 โดยที่ค่า +5 โวลต์ มีค่าเป็นลอจิก 1 และค่า 0 โวลต์ มีค่าเป็นลอจิก 0

2. เตรียม Setting ในการติดต่อ เช่น อัตราการสื่อสารข้อมูล (Baud rate) แล้วกี่ Parameter ต่างๆ ที่เกี่ยวข้อง เช่น ต้องการสื่อสารกับพอร์ตอนุกรมโดยใช้อัตราการสื่อสารข้อมูล เท่ากับ 9600 Bit/Sec และส่งข้อมูลทีละ 8 Bit และต้องการ 1 Bit ในการหยุด และไม่มีการทดสอบความผิดพลาดของข้อมูลในรูปแบบภาวะเลขคี่ หรือ ภาวะเลขคู่ Setting = 9600,8,N,1 ดังนั้นถ้าเราเขียนโปรแกรมในการส่งข้อมูลให้ PC เราจะต้องทำการ Set parameter ต่างๆ ทางฝั่งของ PC ให้ตรงกันด้วย สำหรับการตั้งค่าหน้าต่าง วิชวลเบสิก จะเชื่อมต่อการสื่อสารกับ COM1 โดยใช้อัตราการสื่อสารข้อมูลจาก Baud rate = 9600 Bit/Sec

3.4.2 การทำงานของไมโครคอนโทรลเลอร์เพื่อส่งสัญญาณเตือนและบันทึกผล

การทำงานของไมโครคอนโทรลเลอร์ในส่วนนี้ จะเป็นการควบคุมการทำงานของอุปกรณ์ต่างๆ คือ การติดดับของไฟแอลอีดี การดังของลำโพง การอ่านค่าเวลาจาก RTC ลิมิตสวิสช์และ Micro SD card โดยโปรแกรมในส่วนนนี้ จะมีฟังก์ชันในการตรวจสอบซีเรียลเพื่อให้ตรงกับการใช้งานในแต่ละอุปกรณ์ จากนั้นตัวอุปกรณ์จะทำงานตามที่โปรแกรมสั่งคือ จะมีการเตือนเมื่อถึงเวลา จากนั้นจะมีการบันทึกค่าเป็น E เมื่อมีผู้ป่วยมาหยิบยาไปรับประทาน และจะบันทึกเป็น F เมื่อหมดช่วงเวลาแล้วไม่มีผู้ป่วยมาหยิบยาไปรับประทาน



รูปที่ 3.23 แผนภาพกรอบแสดงการทำงานของไมโครคอนโทรลเลอร์

3.5 ส่วนควบคุมติดต่อกับผู้ใช้งาน

ในการประยุกต์การใช้งานพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ AVR ติดต่อเครื่องคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม โดยกำหนดเป็นรูปแบบการรับส่งข้อมูลที่เรียกว่าโปรโตคอล (Protocol) โดยทางฝั่งคอมพิวเตอร์จะเขียนโปรแกรมควบคุมพอร์ตอนุกรมด้วย วิชวลลเบสิก (Visual Basic) เพื่อแสดงถึงการรับส่งข้อมูลระหว่างไมโครคอลโทรเลอร์กับคอมพิวเตอร์

3.5.1 การบันทึก Text file และการแสดงผลบน PC

เมื่อเราต้องการที่จะดึงข้อมูลการรับประทานยาของผู้ป่วยมาดู สามารถทำได้โดยการเปิดหน้าต่าง VB -> Statistic -> Get file หลังจากนั้นก็ save text file ไว้ในตำแหน่งที่ต้องการบันทึก โดยการที่ text file จะถูกบันทึกลงบน PC นั้น ข้อมูลจะถูกส่งผ่านสาย RS-232 ที่เชื่อมต่อระหว่าง COMPORT 1 ของ PC ต่อเข้ากับ Port UART 0 ของบอร์ดไมโครคอนโทรลเลอร์ AVR หลังจากกด Get file ที่หน้าต่าง VB แล้ว ข้อมูลผู้ป่วยและสถิติการรับประทานยาจะถูกเก็บไว้ที่ database เพื่อให้เราสามารถ search ดูสถิติการรับประทานยาตามวันที่เราต้องการดูได้ โดยสามารถเลือกช่วงเวลาที่ต้องการดูสถิติการรับประทานยาได้ผ่านทางหน้าต่าง VB -> Statistic -> เลือกช่วงเวลา 2 ช่วงเวลา รวมถึงสามารถ print ชื่อผู้ป่วยและข้อมูลการรับประทานยาออกมาได้ผ่านการใช้งานที่หน้าต่าง VB อีกด้วย

3.5.2 หน้าต่างการตั้งค่าโปรแกรมวิชวลเบสิก

สำหรับหน้าต่างแอพพลิเคชั่นของโปรแกรมวิชวลเบสิก ดังรูปที่ 3.24 ประกอบด้วย

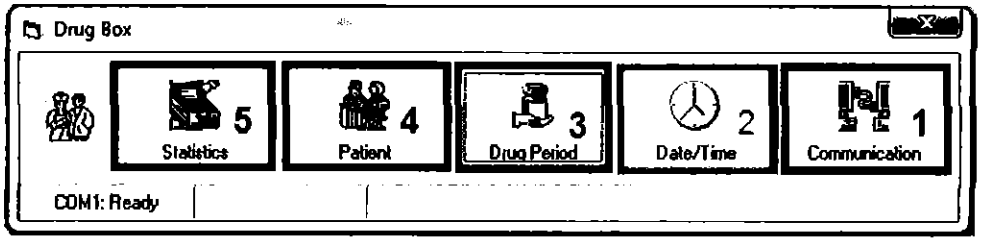
หมายเลข 1 คือ หน้าต่างตั้งค่าการติดต่อสื่อสาร

หมายเลข 2 คือ หน้าต่างตั้งค่าวันที่และเวลา

หมายเลข 3 คือ หน้าต่างตั้งค่าเวลารับประทานยาและเวลารับประทานอาหาร

หมายเลข 4 คือ หน้าต่างตั้งชื่อผู้ป่วย

หมายเลข 5 คือ หน้าต่างแสดงผล



รูปที่ 3.24 หน้าต่างแอพพลิเคชั่นของโปรแกรมวิชวลเบสิก

3.5.2.1 หน้าต่างตั้งค่าการติดต่อสื่อสาร

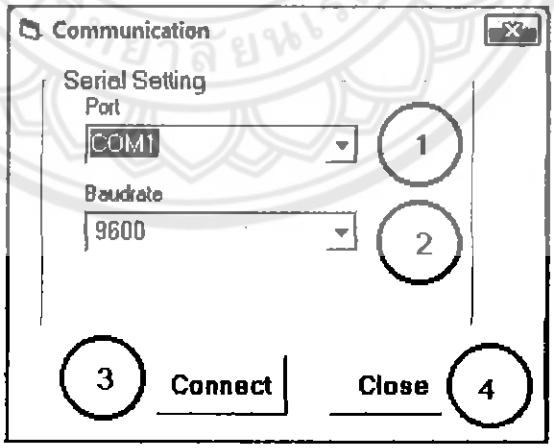
สำหรับหน้าต่างตั้งค่าการติดต่อสื่อสาร เพื่อเชื่อมต่อระหว่าง PC กับ AVR **ดังรูปที่ 3.25 มี** รายละเอียดดังนี้

หมายเลข 1 คือ ปุ่มเลือก Port สำหรับติดต่อกับ AVR

หมายเลข 2 คือ ปุ่มเลือก อัตราการสื่อสาร (Baudrate) เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม

หมายเลข 3 คือ ปุ่ม Connect สำหรับส่งค่าการติดต่อสื่อสารไปยัง AVR

หมายเลข 4 คือ ปุ่ม Close สำหรับปิดหน้าต่าง Communication



รูปที่ 3.25 หน้าต่างตั้งค่าการติดต่อสื่อสาร

3.5.2.2 หน้าต่างตั้งค่าวันที่และเวลา

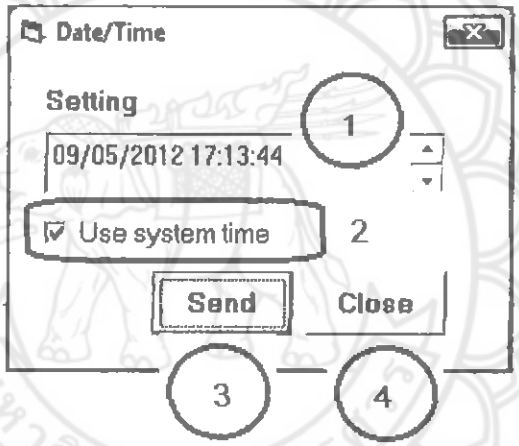
สำหรับหน้าต่างตั้งค่าวันที่ และเวลาให้ตรงกับระบบคอมพิวเตอร์ ดังรูปที่ 3.26 แสดงรายละเอียดการทำงานของปุ่มกดต่างๆ ดังนี้

หมายเลข 1 คือ ปุ่มเลือกวันที่ และเวลา

หมายเลข 2 คือ ปุ่มเลือกใช้เวลาของระบบ

หมายเลข 3 คือ ปุ่ม Send สำหรับส่งค่าวันที่และเวลาไปยังบอร์ดไมโครคอนโทรลเลอร์

หมายเลข 4 คือ ปุ่ม Close สำหรับปิดหน้าต่าง Date/Time



รูปที่ 3.26 หน้าต่างตั้งค่าวันที่และเวลา

3.5.2.3 หน้าต่างตั้งค่าเวลาเตือนรับประทานยาและเวลาเตือนรับประทานอาหาร

สำหรับหน้าต่างตั้งค่าเวลาเตือนรับประทานยาและเวลาเตือนรับประทานอาหาร ดังรูปที่ 3.27 แสดงรายละเอียดของช่องข้อความและปุ่มกดต่างๆ ดังนี้

หมายเลข 1 คือ ช่องเลือกรายชื่อผู้ป่วย

หมายเลข 2 คือ ช่องเลือกรายการมื้อยา

หมายเลข 3 คือ ช่องกรอกรายละเอียด ชนิดยา

หมายเลข 4 คือ ปุ่มเลือกเวลาเริ่มต้นการเตือนเป็น ชั่วโมง นาที

หมายเลข 5 คือ ปุ่มเลือกเวลาสิ้นสุดการเตือนเป็น ชั่วโมง นาที

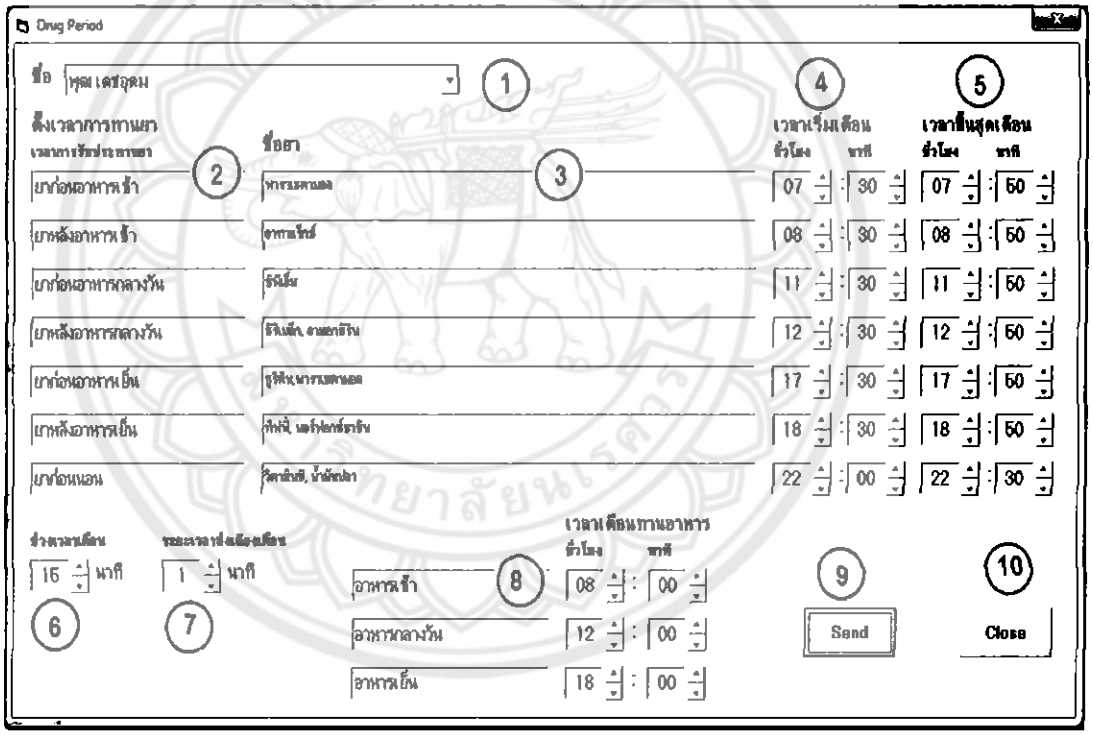
หมายเลข 6 คือ ปุ่มเลือกช่วงเวลาการเตือน หรือระยะห่างการเตือน

หมายเลข 7 คือ ปุ่มเลือกระยะเวลาส่งเสียงเตือนแต่ละครั้งว่าจะให้สัญญาณเสียงดังนานเท่าไร

หมายเลข 8 คือ ช่องมื้ออาหาร เช่น อาหารเช้า อาหารกลางวัน อาหารเย็น

หมายเลข 9 คือ ปุ่ม Send สำหรับส่งค่าเวลาเตือนไปยังบอร์ดไมโครคอนโทรลเลอร์

หมายเลข 10 คือ ปุ่ม Close สำหรับปิดหน้าต่าง Drug Period



รูปที่ 3.27 หน้าต่างตั้งค่าเวลาการรับประทานยาและเวลาการรับประทานอาหาร

ตัวอย่างโค้ดโปรแกรมในการส่งค่าของปุ่ม Send

Private Sub cmdSend_Click()	- ส่งการตั้งเวลารับประทานยา และเวลารับประทานอาหารไปที่ Microcontroller
Dim PatientId As String	ประกาศตัวแปร
Dim ItemID As String	
Dim PeriodName As String	
Dim DrugName As String	
Dim StartTime As String	
Dim StopTime As String	
Dim TimeAlarm As String	
Dim PeriodAlarm As String	
Dim AlarmTime As Integer	
Dim AlarmPeriod As Integer	
Dim i As Byte	
EnableButton (False)	- ปิดการทำงานของปุ่มต่างๆ
On Error Resume Next	- ถ้า Error จะไม่ทำงานด้านล่าง
MainForm.StatusBar1.Panels(3).Text = ""	- ส่วนล่างสุดของหน้าต่าง
If CheckDrugTime Then	- ตรวจสอบเวลารับประทานยา
If CheckEatTime Then	- ตรวจสอบเวลารับประทานอาหาร
If Not MainForm.MSComm.PortOpen Then	- ตรวจสอบการเชื่อมต่อกับพอร์ตอนุกรม
MainForm.MSComm.PortOpen = True	- เปิดการเชื่อมต่อกับพอร์ตอนุกรม
MainForm.MSComm.InputMode= comInputModeBinary	- รับข้อมูลเข้า
MainForm.StatusBar1.Panels(1).Text= frmCommunication.cbbPortName.Text & ": Connected"	- แสดงสถานะของการเชื่อมต่อ (Connected) ในส่วนของ StatusBar
End If	- จบการทำงาน
AlarmTime = CInt(txtAlarmTime.Text)	- รับค่าช่วงเวลาการเตือน " CInt ใช้แปลงชนิดตัวแปรให้เป็นตัวแปรชนิด Integer
AlarmPeriod=CInt(txtAlarmPeriod.Text)	- ระยะเวลาส่งเสียงเตือน

DataSend = ""	- ส่งข้อมูล
For i = 0 To 6	- เริ่มตั้งแต่ 0 ถึง 6
DataSend=DataSend & Set2Pos(txtStartHour(i).Text)&Set2Pos(txtStartMin(i).Text)&"00"&_	- รูปแบบข้อมูลเวลาเริ่มต้น เป็น ชั่วโมง กับ นาที
Set2Pos(txtStopHour(i).Text)&Set2Pos(txtStopMin(i).Text)& "00"	- รูปแบบข้อมูลเวลาสิ้นสุด เป็น ชั่วโมง กับ นาที
Next i	- ถัดไป
DataSend=DataSend&Format$(AlarmTime, "00")&Format$(AlarmPeriod,"00")&Format$(cbbPatient.ListIndex + 1, "000")	- ส่งข้อมูล ช่วงเวลาเตือน และระยะเวลาส่งเสียงเตือน
MainForm.InitialDataReceive	- ตั้งค่าเริ่มต้นของการส่งข้อมูล
CallModule.CommPortSendData(&H1,&H5,&H1,DataSend)	- ส่งข้อมูลไปที่ Microcontroller ในรูปแบบ HDLC (Address, Command 0x05, Offset, Data)
ConnDB.Open	- เปิดการติดต่อฐานข้อมูล
DeleteSetting	- ลบข้อมูล
For i = 0 To 6	- เริ่มตั้งแต่ 0 ถึง 6
PatientId = "" & (cbbPatient.ListIndex + 1)	- หมายเลขผู้ป่วย
ItemID = "" & i + 1	- ลำดับช่องทั้ง 7 ช่อง
PeriodName = txtPeriod(i).Text	- ช่องกรอกมื้อยาทั้ง 7 ช่อง
DrugName = txtDrug(i).Text	- ช่องกรอกชนิดยาทั้ง 7 ช่อง
StartTime=Set2Pos(txtStartHour(i).Text) & ":" & Set2Pos(txtStartMin(i).Text) & ":" & "00"	- รูปแบบเวลาเริ่มต้นเป็น ชั่วโมง กับ นาที
StopTime=Set2Pos(txtStopHour(i).Text) & ":" & Set2Pos(txtStopMin(i).Text) & ":" & "00"	- รูปแบบเวลาสิ้นสุดเป็น ชั่วโมง กับ นาที
TimeAlarm = "" & UpDown1.Value	- ช่วงเวลาการเตือน
PeriodAlarm = "" & UpDown2.Value	- ระยะเวลาส่งเสียงเตือน
NamePeriod(i)=txtPeriod(i).Text	
CallSaveSetting(PatientId,ItemID,PeriodName, DrugName, StartTime, StopTime, TimeAlarm, PeriodAlarm)	- บันทึกข้อมูล (PatientId,ItemID,PeriodName,DrugName,StartTime,StopTime,TimeAlarm,PeriodAlarm)

Next i	- ถัดไป
ConnDB.Close	- ปิดการติดต่อฐานข้อมูล
Delay (2)	- ดีเลย์ 2 วินาที
Call cmdSendEat_Click	- ส่งการตั้งค่าเวลาการรับประทานยา
Else	
MsgBox "ตั้งค่าเวลาทานยาผิดพลาด", vbOKOnly, "แจ้งข้อผิดพลาด"	- หน้าต่างแจ้งข้อผิดพลาด แสดงข้อความ " ตั้งค่าเวลาทานยาผิด "
End If	- จบการทำงาน
Else	
MsgBox "ตั้งค่าเวลาทานยาผิดพลาด", vbOKOnly, "แจ้งข้อผิดพลาด"	- หน้าต่างแจ้งข้อผิดพลาด แสดงข้อความ " ตั้งค่าเวลาทานยาผิด "
End If	
EnableButton (True)	- เปิดการทำงานของปุ่มต่างๆ
End Sub	- จบการทำงาน

3.5.2.4 หน้าต่างตั้งค่าข้อมูลผู้ป่วย

สำหรับหน้าต่างกรอกประวัติผู้ป่วย ดังรูปที่ 3.28 แสดงรายละเอียดของปุ่มกดต่างๆ ดังนี้

หมายเลข 1 คือ ช่องกรอกรายชื่อผู้ป่วย

หมายเลข 2 คือ ช่องกรอกนามสกุลผู้ป่วย

หมายเลข 3 คือ ช่องกรอกรายละเอียดผู้ป่วย เช่น อายุ ที่อยู่

หมายเลข 4 คือ ช่องแสดงลำดับรายการ เช่น ชื่อ นามสกุล รายละเอียด หลังจากการเพิ่มหรือเปลี่ยนแปลงประวัติผู้ป่วย

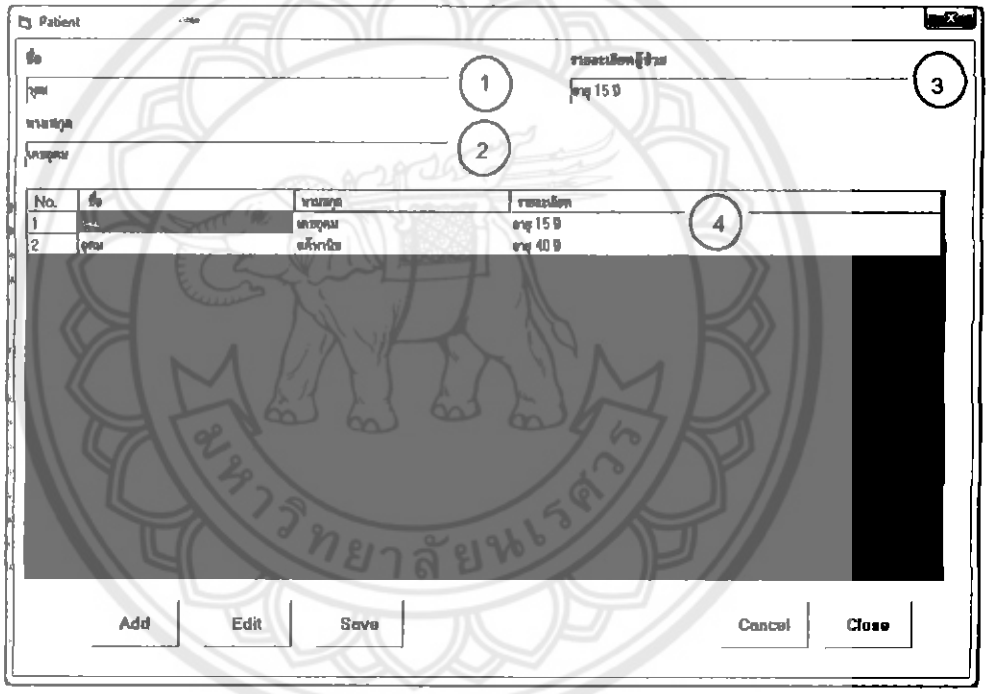
หมายเลข 5 คือ ปุ่ม Add สำหรับเพิ่มช่องกรอกประวัติผู้ป่วยรายใหม่

หมายเลข 6 คือ ปุ่ม Edit สำหรับบันทึกข้อมูลที่ทำการเปลี่ยนแปลงหลังจากกดปุ่ม Edit สำหรับเปลี่ยนชื่อ – นามสกุล และรายละเอียดผู้ป่วยในช่องกรอกข้อความ ข้อมูลดังกล่าวที่กรอกไว้ จะถูกบันทึกอยู่ในส่วนแสดงผล

หมายเลข 7 คือ ปุ่ม Save สำหรับบันทึกข้อมูลไปเก็บไว้ใน DataBase

หมายเลข 8 คือ ปุ่ม Cancel สำหรับยกเลิกรายการข้อมูลออกจากส่วนแสดงผล โดยการกดปุ่ม Cancle รายการดังกล่าวที่อยู่ในส่วนแสดงผลก็จะถูกลบออกไป

หมายเลข 9 คือ ปุ่ม Close สำหรับปิดหน้าต่าง Patient



รูปที่ 3.28 หน้าต่างตั้งค่าข้อมูลผู้ป่วย

3.5.2.5 หน้าต่างแสดงผล

สำหรับหน้าต่างแสดงผลข้อมูลการรับประทานยา ดังรูปที่ 3.29 แสดงรายละเอียดของปุ่มกดต่างๆ ดังนี้

หมายเลข 1 ปุ่มกดเลือกรายชื่อผู้ป่วยจากรายชื่อที่มีอยู่

หมายเลข 2 ปุ่มกดเลือกช่วงเวลาที่เราต้องการทราบข้อมูล

หมายเลข 3 ปุ่มกดเลือกช่องมื้อยา เช่น ยาก่อนอาหารเช้า ยาหลังอาหารเช้า ยาก่อนอาหารกลางวัน ยาหลังอาหารกลางวัน ยาก่อนอาหารเย็น ยาหลังอาหารเย็น ยาก่อนนอน อื่นๆ หรือทุกช่อง

หมายเลข 4 ปุ่มกดเลือกสถานะ (ทาน-ไม่ทาน)

หมายเลข 5 ปุ่มกด Search สำหรับค้นหาข้อมูลผู้ป่วยแต่ละราย

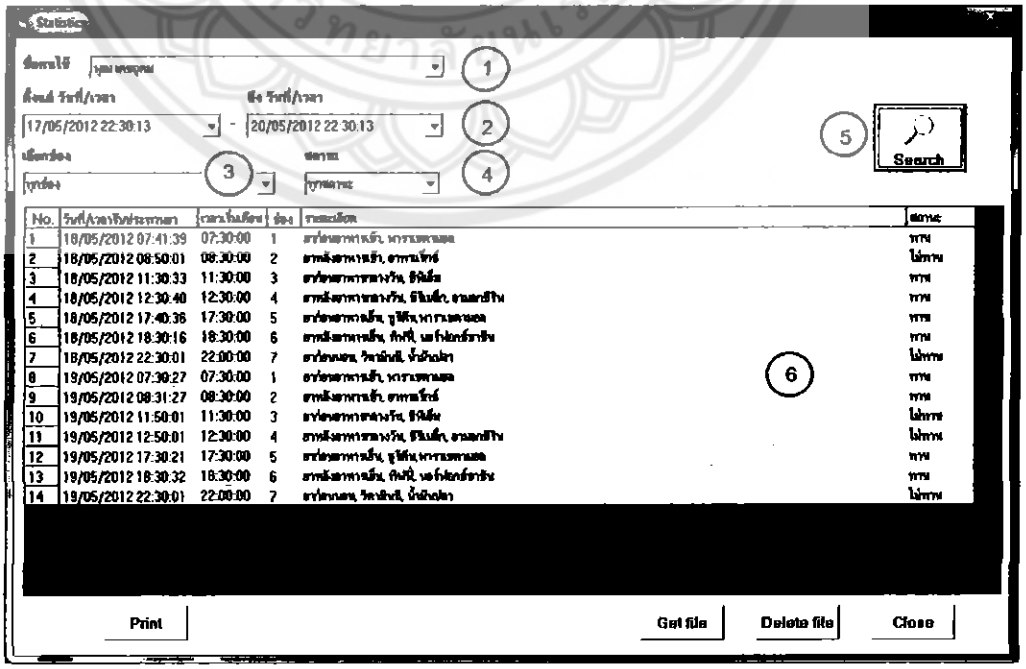
หมายเลข 6 ส่วนแสดงรายละเอียดข้อมูล ลำดับ เวลาเริ่มต้นการเตือน วันที่/เวลารับประทานยา ช่อง รายละเอียด (ชื่อยา) และสถานะ (ทาน-ไม่ทาน)

หมายเลข 7 ปุ่มกด Print เพื่อแสดงรายละเอียดต่างๆในการรับประทานยาออกมา สามารถพิมพ์เพื่อนำมาดูสถิติการรับประทานยาของผู้ป่วยแต่ละราย

หมายเลข 8 ปุ่มกด Get file ดึงไฟล์มาจาก Text File ใน SD-Card เพื่ออ่านค่าเก็บไว้ใน DataBase

หมายเลข 9 ปุ่มกด Delete file สำหรับลบข้อมูลในMemomy (SD card) แต่ยังคงข้อมูลล่าสุดที่มีการ Get file ไว้อยู่

หมายเลข 10 กดปุ่ม Close เพื่อปิดหน้าต่างแสดงผล



รูปที่ 3.29 หน้าต่างแสดงผล

ตัวอย่างโค้ดโปรแกรมปุ่ม Get file

Private Sub cmdGetData_Click()	- ดึงข้อมูล text file จาก Microcontroller
EnableButton (False)	- ปิดการทำงานของปุ่มต่างๆ
cmdPrint.Enabled = False	- ปิดการทำงานของปุ่ม cmdPrint
Screen.MousePointer = vbHourglass	ใ - ช้ Mouse cursor แบบ Hourglass
On Error Resume Next	- ถ้า Error จะไม่ทำงานด้านล่าง
MainForm.StatusBar1.Panels(3).Text = ""	- ส่วนล่างสุดของหน้าต่าง (StatusBar)
If Not MainForm.MSComm.PortOpen Then	- ตรวจสอบการเชื่อมต่อกับ Serial port
MainForm.MSComm.PortOpen = True	- เปิดการเชื่อมต่อพอร์ตอนุกรมของเครื่องคอมพิวเตอร์
MainForm.MSComm.InputMode = comInputModeBinary	- รับข้อมูลเข้า
MainForm.StatusBar1.Panels(1).Text = frmCommunication.cbbPortName.Text & ": Connected"	- แสดงสถานะการเชื่อมต่อ (Connected)
End If	- จบการทำงาน
PatientId = cbbPatient.ListIndex + 1	- หมายเลขผู้ป่วยจาก List Box
DataSend = "READ TEXT"	- ส่งข้อมูลที่อ่านได้จาก Text
MainForm.InitialDataReceive	- ตั้งค่าเริ่มต้นของการส่งข้อมูล
Call Module.CommPortSendData(&H1, &H3, &H1, DataSend)	- ส่งข้อมูลไปที่ Microcontroller ในรูปแบบ HDLC (Address, Command 0x03, Offset, Data)
Screen.MousePointer = vbDefault	- เปิด Mouse cursor เป็นลูกศรปกติ
EnableButton (True)	- เปิดการทำงานของปุ่มต่างๆ
End Sub	- จบการทำงาน

บทที่ 4
ผลการทดสอบ

หลังจากออกแบบและสร้างตัวกล่องบรรจุยา รวมทั้งออกแบบหน้าต่างตั้งค่าช่วงเวลาสำหรับการรับประทานยาและรับประทานอาหารแล้ว ผู้ดำเนินโครงงานได้ดำเนินการทดสอบการทำงานของบอร์ดไมโครคอนโทรลเลอร์ AVR โดยการตั้งค่าผ่านทางหน้าต่างแอพพลิเคชั่น วิชวลเบสิก ดังจะอธิบายในหัวข้อถัดไป

4.1 ส่วนประกอบและขั้นตอนการทำงานของระบบ

เมื่อเริ่มโปรแกรม ทำการตั้งค่าเวลารับประทานยาและเวลารับประทานอาหารผ่านหน้าต่าง VB ในการส่งข้อมูลจาก PC ไปบอร์ดไมโครคอนโทรลเลอร์ AVR ข้อมูลจะถูกส่งผ่านพอร์ตอนุกรม RS-232 ในรูปแบบ HDLC Protocol 9 byte โดยจะทำตามลำดับขั้นตอนในการส่งแต่ละ byte ดังอธิบายในตารางที่ 3.1 จากนั้น ไมโครคอนโรลเลอร์ AVR จะรับค่าคำสั่งจาก PC มาเก็บไว้ในหน่วยความจำ SD-Card และทำการประมวลผลโปรแกรมตอบกลับไปยัง PC เมื่อไมโครคอนโทรลเลอร์ AVR ทำงานตามคำสั่งโปรแกรมแล้วจะส่งสัญญาณเตือนออกมาเมื่อถึงเวลาที่ตั้งไว้ให้ผู้ป่วยหยิบยาไปรับประทาน

หลักการทำงานทั้งหมดของระบบเชื่อมต่อเครื่องโปรแกรม ET-AVRISP mkII กับบอร์ดไมโครคอนโทรลเลอร์ AVR เปิดโปรแกรม AVR Stuido 4 และโปรแกรม code ลงไป จากนั้นเชื่อมต่อพอร์ต RS-232 ระหว่าง PC กับบอร์ดไมโครคอนโทรลเลอร์ AVR ตั้งค่าเวลาการรับประทานยาผ่านหน้าต่างตั้งค่า วิชวลเบสิก จากนั้นเวลาที่ตั้งค่าจะถูกส่งจาก PC ผ่านสาย RS-232 เพื่อที่จะแปลงข้อมูลขนาน ที่จะส่งจากคอมพิวเตอร์โดยการส่งผ่านสาย RS 232 ซึ่งจะเป็นการส่งข้อมูลแบบอนุกรม และเมื่อบอร์ดไมโครคอนโทรลเลอร์ได้รับข้อมูลอนุกรมดังกล่าว จะทำการแปลงข้อมูลอนุกรมนั้นไปเป็นข้อมูลแบบขนานเพื่อนำข้อมูลเหล่านั้นไปใช้งานต่อไป คำสั่งการตั้งค่าจะถูกเก็บไว้ที่ SD-Card เพื่อรอการประมวลผลคำสั่งการทำงาน และส่งตอบกลับไปยัง PC เมื่อไมโครคอนโทรลเลอร์ทำงานตามคำสั่งโปรแกรมแล้วจะแสดงสัญญาณการเตือนตามช่วงเวลาที่ตั้งค่าออกทางหลอดไฟแอลอีดี และบัซเซอร์ ให้ผู้ป่วยหยิบยาไปทาน จากนั้นจะมีการเก็บบันทึกค่าช่วงเวลาการทาน-ไม่ทานยา ไว้ในหน่วยความจำ SD-Card ข้อมูลจาก SD-Card จะถูกเก็บบันทึกในรูปแบบ Text file ซึ่งข้อมูลดังกล่าวจะเก็บไว้ใน Database ของโปรแกรม VB เราสามารถดูสถิติ

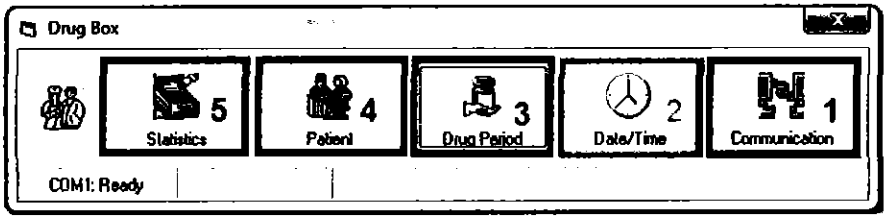
การรับประทานยาของผู้ป่วยได้ โดยการแสดงผลทางหน้าต่าง VB ซึ่งจะดึงค่าข้อมูลจาก Databse มาแสดงผลออกทางจอภาพ ขั้นตอนการทำงานทั้งหมดมีดังต่อไปนี้

1. เริ่มต้น โปรแกรมโค้ด AVR ลงบอร์ดไมโครคอนโทรลเลอร์
2. เชื่อมต่อสาย RS-232 ระหว่างบอร์ดไมโครคอนโทรลเลอร์ AVR กับ คอมพิวเตอร์
3. เปิดหน้าต่างโปรแกรมวิชวลเบสิกเพื่อทำการตั้งค่าเวลาการเตือน
4. เวลาที่ตั้งค่าถูกส่งไปบันทึกลง SD-Card ในบอร์ดไมโครคอนโทรลเลอร์ AVR
5. ไมโครคอนโทรลเลอร์ AVR ทำงานตามคำสั่งการตั้งค่าเวลา
6. บันทึกข้อมูลเวลาการรับประทานยาลง SD-Card และเก็บข้อมูลในรูปแบบ Text file
7. ใช้หน้าต่างแสดงผลของโปรแกรมวิชวลเบสิกดึงข้อมูล Text file ใน SD-Card มาแสดงผล
8. จบการทำงาน

4.2 วิธีการใช้งาน

นอกจากนี้ ได้ออกแบบหน้าต่างโปรแกรม วิชวลเบสิก ที่ใช้สำหรับการตั้งค่าและแสดงผลข้อมูลที่จัดเก็บไว้ในหน่วยความจำของไมโครคอนโทรลเลอร์ ซึ่งจะแสดงข้อมูล ชื่อผู้ป่วย วันที่และเวลา ชนิดยา การควบคุมอุปกรณ์โดยผ่านบอร์ดไมโครคอนโทรลเลอร์ จะต้องทำการตั้งค่าต่างๆผ่านทางหน้าต่างโปรแกรมวิชวลเบสิก

ทำการติดตั้งอุปกรณ์กล่องยาและเชื่อมต่ออุปกรณ์ต่างๆร่วมกันระหว่างคอมพิวเตอร์กับบอร์ดไมโครคอนโทรลเลอร์ AVR จากนั้นเปิดแอพพลิเคชั่นของโปรแกรมวิชวลเบสิกขึ้นมา จะปรากฏหน้าต่างการตั้งค่าใช้งานดังรูปที่ 4.1



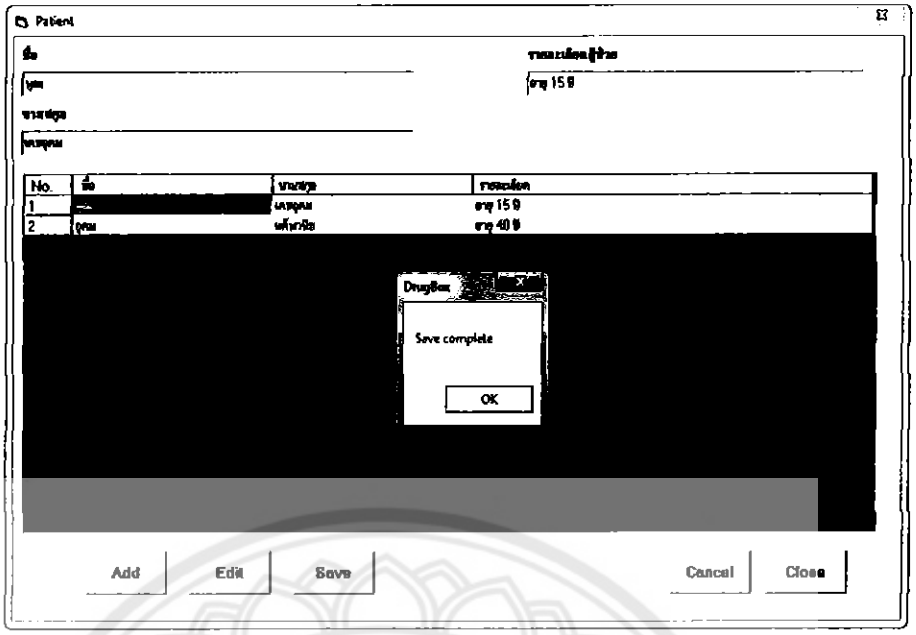
รูปที่ 4.1 หน้าต่างแอพพลิเคชั่นของโปรแกรมวิชวลเบสิก

ในการใช้งานหน้าต่างตั้งค่า วิชวลเบสิก โดยการคลิกเลือกส่วนที่เราจะตั้งค่า มีลำดับขั้นตอนดังนี้

4.2.1 ขั้นตอนการตั้งค่าหน้าต่างกำหนดชื่อ-นามสกุล และรายละเอียดผู้ป่วย

ในการกรอกประวัติผู้ป่วยผ่านทางหน้าต่างตั้งค่า มีขั้นตอนการตั้งค่าดังนี้

1. กดปุ่ม Add สำหรับเพิ่มช่องรายชื่อผู้ป่วยรายใหม่
2. กรอกรายชื่อผู้ป่วย
3. กรอกนามสกุลผู้ป่วย
4. กรอกรายละเอียดผู้ป่วย
5. กดปุ่ม Edit สำหรับเปลี่ยนชื่อ – นามสกุลและรายละเอียด ในส่วนที่กดเลือกรายการนั้น จะกรอกชื่อ – นามสกุล รายละเอียดผู้ป่วย แล้วกดปุ่ม Edit ประวัติผู้ป่วยดังกล่าวที่กรอกไว้ จะถูกบันทึกอยู่ในส่วนแสดงผล
6. ถ้ากดปุ่ม Cancel จะยกเลิกข้อมูล ออกจากส่วนแสดงรายชื่อ รายชื่อดังกล่าวที่อยู่ในส่วนแสดงรายชื่อก็จะถูกลบออกไป
7. กดปุ่ม Save สำหรับบันทึกข้อมูลเมื่อกรอกรายชื่อ-รายละเอียดเรียบร้อยแล้วจะปรากฏหน้าต่างแสดงข้อความ "Save complete" แสดงว่าการบันทึกข้อมูลเรียบร้อย แล้วกดปุ่ม OK ดังรูปที่ 4.2
8. ดับเบิ้ลคลิกที่ชื่อเพื่อลบรายชื่อนั้นออก
9. กดปุ่ม Close เพื่อปิดหน้าต่าง Patient

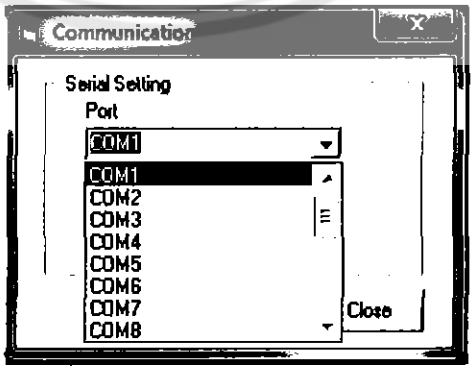


รูปที่ 4.2 หน้าต่างตั้งค่าข้อมูลผู้ป่วย

4.2.2 ขั้นตอนการเชื่อมต่อการติดต่อสื่อสารระหว่างพอร์ตกับหน้าต่างวิชวลเบสิก

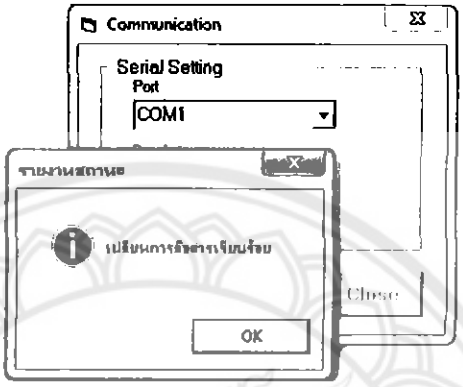
การเชื่อมต่อการสื่อสารมีขั้นตอนการตั้งค่าดังนี้

1. เลือกข้อมูลจากรายการข้อมูลที่มีอยู่ โดยกดเลือก COM 1
2. เลือกการสื่อสารกับพอร์ตอนุกรมโดยใช้อัตราการสื่อสารข้อมูลจาก Baud rate = 9600 Bit/Sec



รูปที่ 4.3 หน้าต่างการตั้งค่าการติดต่อสื่อสาร

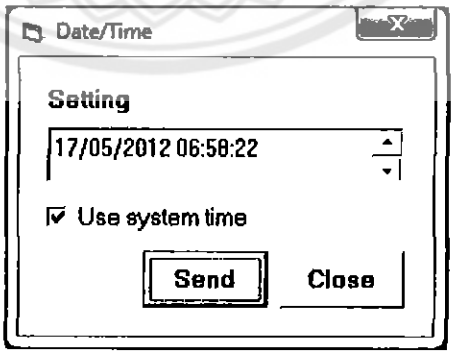
1. กดปุ่ม Connect เพื่อติดต่อการสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ AVR เมื่อกดปุ่ม Connect จะปรากฏหน้าต่างรายงานสถานะ "เปลี่ยนการสื่อสารเรียบร้อย" แล้วกดปุ่ม OK ดังรูปที่ 4.4
2. ปิดหน้าต่าง Communication โดยกดปุ่ม Close



รูปที่ 4.4 รายงานสถานการณ์เชื่อมต่อ

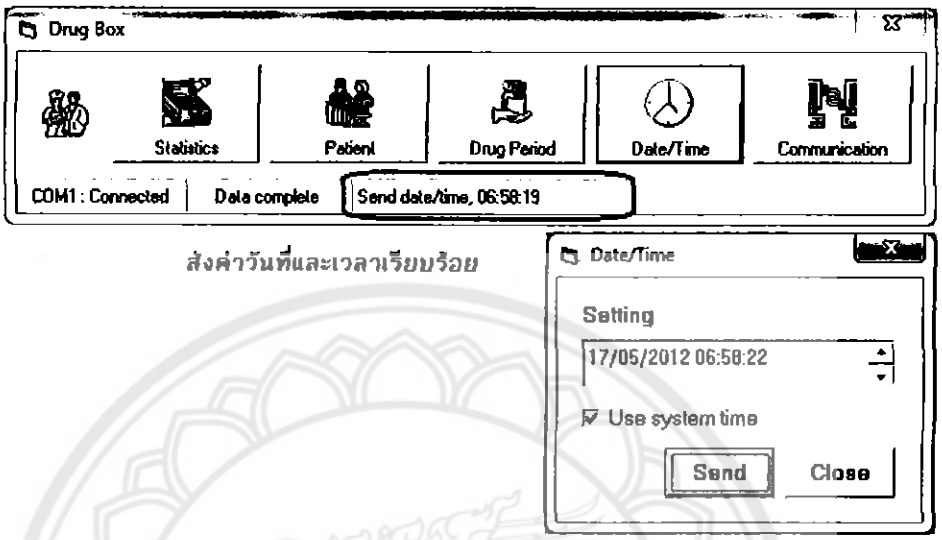
4.2.3 ขั้นตอนการตั้งค่าวันที่และเวลา

ซึ่งเมื่อเปิดแอพพลิเคชันขึ้นมา ระบบจะเซตเวลาตามเครื่องคอมพิวเตอร์อยู่แล้ว ซึ่งหน้าต่างการตั้งค่าวันที่และเวลาแสดงดังรูปที่ 4.5



รูปที่ 4.5 หน้าต่างการตั้งค่าวันที่และเวลา

เมื่อตั้งค่าวันที่และเวลาเรียบร้อยแล้ว กด Send เพื่อส่งค่าวันที่และเวลาไปให้ไมโครคอนโทรลเลอร์ จะปรากฏเวลาที่ถูกส่งไป คือ "Send date/time 06:58:19" ดังรูปที่ 4.6



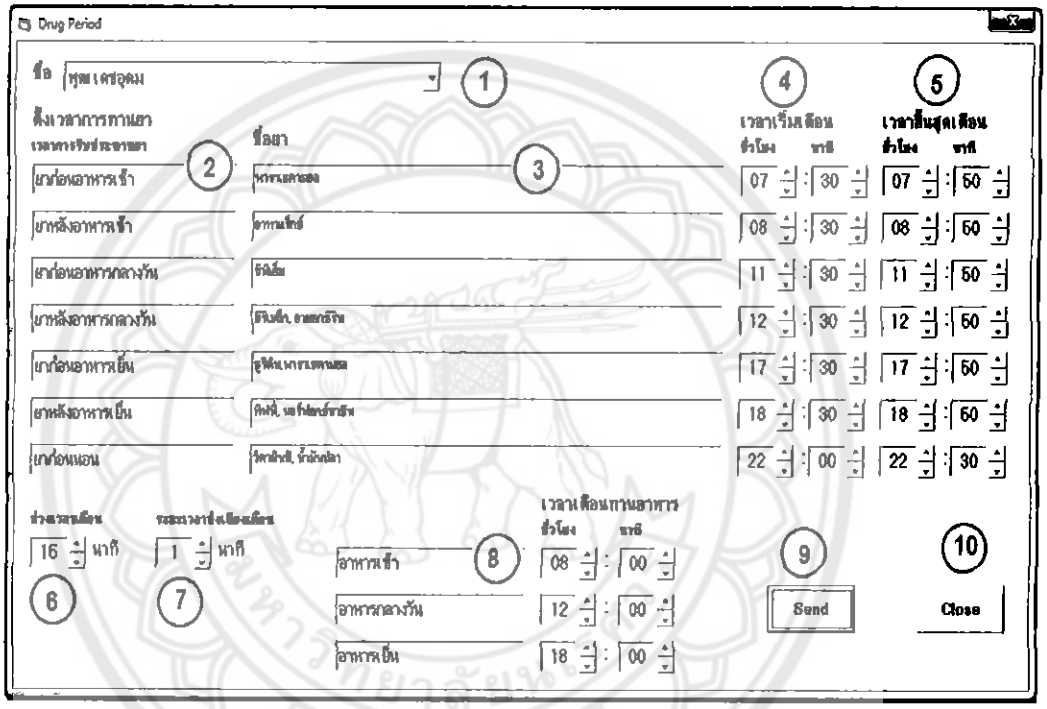
รูปที่ 4.6 รายงานการส่งค่าวันที่และเวลา

4.2.4 ขั้นตอนการตั้งค่าเวลาการรับประทานยาและเวลารับประทานอาหาร

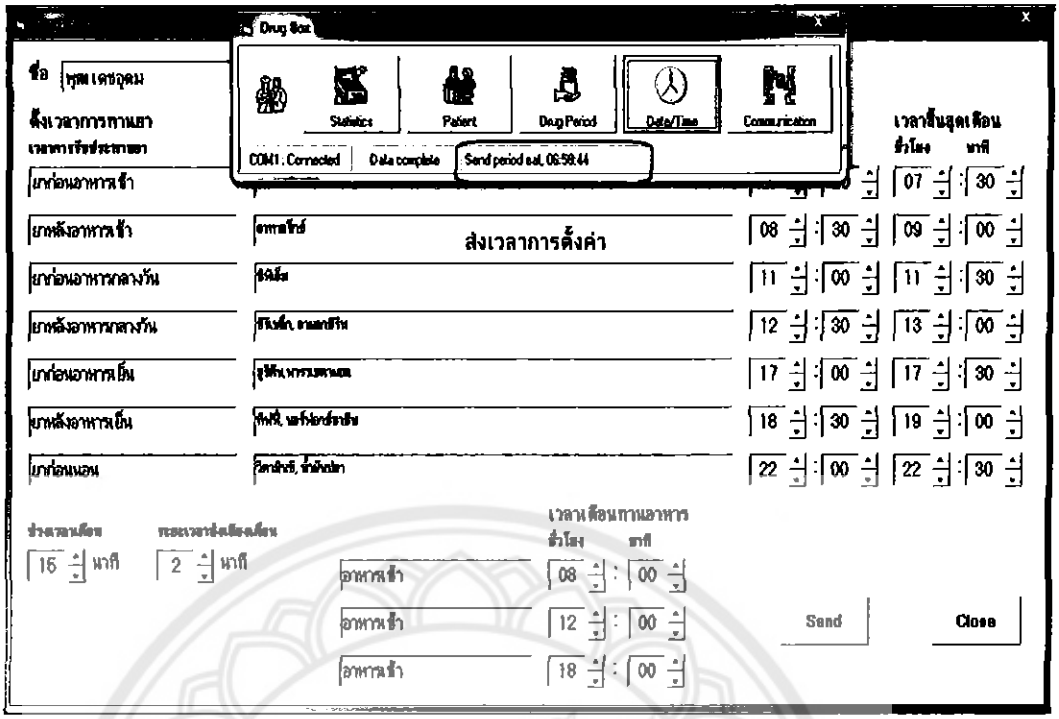
การตั้งค่าเวลาเตือนการรับประทานยามีหลักการดังนี้

1. หมายเลข 1 กดเลือกรายชื่อผู้ป่วยจากรายการข้อมูลที่มีอยู่
2. หมายเลข 2 ช่องสำหรับกรอกรายละเอียดมื้อยาเช่น ยาก่อนอาหารเช้า ยาก่อนอาหารกลางวัน ยาก่อนอาหารเย็น ยาหลังอาหารเช้า ยาหลังอาหารกลางวัน ยาหลังอาหารเย็น ยาก่อนนอน หรือ อื่นๆ
3. หมายเลข 3 ช่องสำหรับกรอกรายละเอียดชนิดยา (ชื่อยา)
4. หมายเลข 4 กดเลือกเวลาเริ่มต้นการเตือนการรับประทานยาที่ต้องการให้ทำงานตามช่วงเวลาเป็น ชั่วโมง นาที
5. หมายเลข 5 กดเลือกเวลาสิ้นสุดการเตือนการรับประทานยาที่ต้องการให้ทำงานตามช่วงเวลาเป็น ชั่วโมง นาที
6. หมายเลข 6 กดเลือกช่วงเวลาการเตือน (ระยะห่างของช่วงเวลาถัดไปหลังจากไม่มีการหยิบยาไปรับประทาน)
7. หมายเลข 7 เลือกระยะเวลาส่งเสียงเตือนในแต่ละครั้งที่เกิดสัญญาณเตือน

8. หมายเลข 8 สำหรับกรอกรายละเอียดมื้ออาหาร เช่น อาหารเช้า อาหารกลางวัน อาหารเย็น
9. หมายเลข 9 กดปุ่ม Send เพื่อส่งข้อมูลจากหน้าต่างการตั้งค่าผ่าน Serial port ไปยังบอร์ดไมโครคอนโทรลเลอร์ โดยเมื่อส่งค่าเวลาการเตือนไปเรียบร้อยแล้ว จะปรากฏเวลาปัจจุบันที่ส่งไป ดังรูปที่ 4.8
10. หมายเลข 10 กดปุ่ม Close เพื่อปิดหน้าต่างตั้งเวลาการรับประทานอาหาร

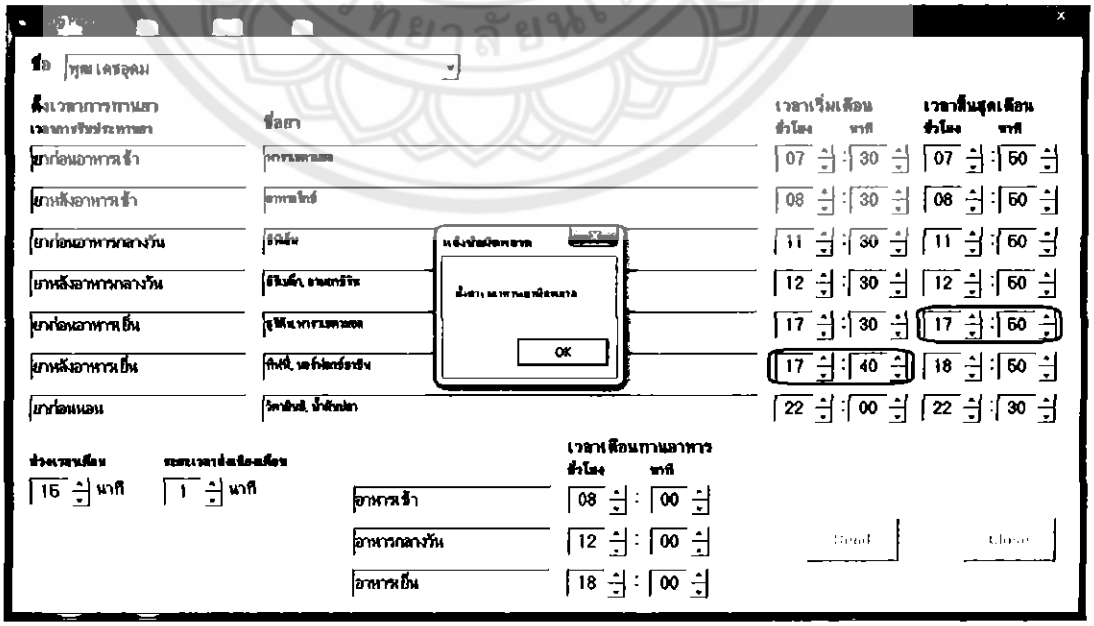


รูปที่ 4.7 หน้าต่างการตั้งค่าเวลาการรับประทานยาและเวลาการรับประทานอาหาร



รูปที่ 4.8 หน้าต่างแสดงการส่งค่าเวลาการเตือนเรียบร้อย

ในกรณีที่มีการตั้งค่าเวลาการเตือนซ้ำกันจะมีหน้าต่างแจ้งข้อผิดพลาด ดังรูปที่ 4.9

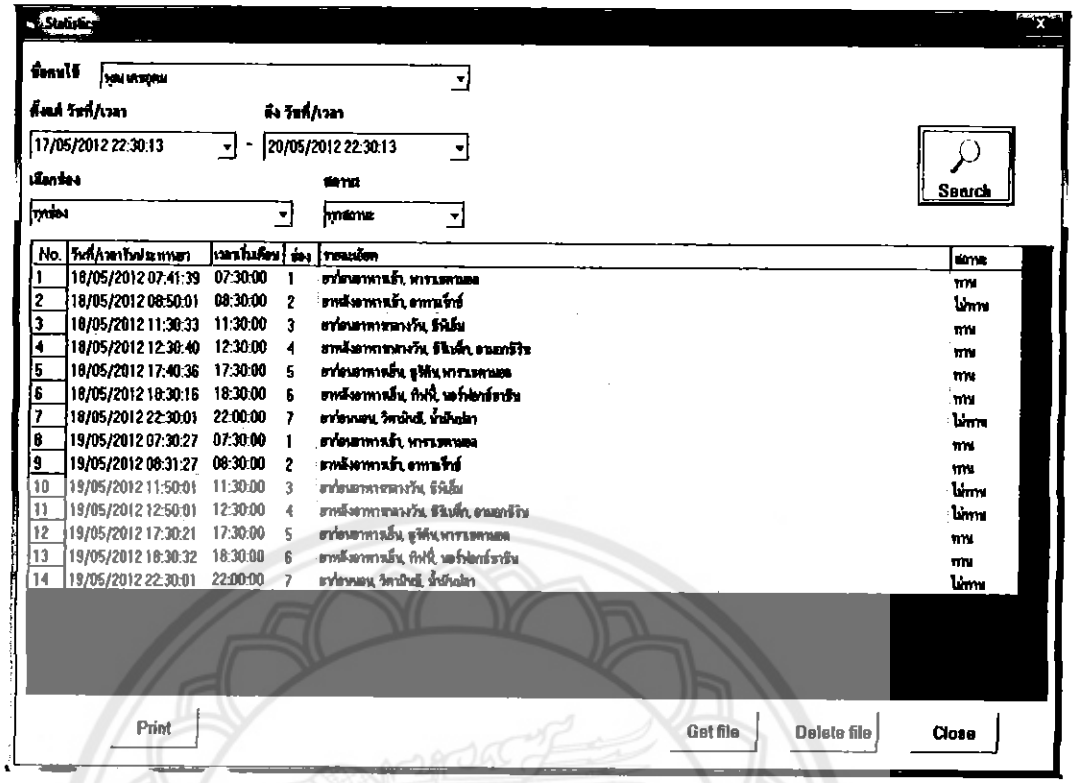


รูปที่ 4.9 หน้าต่างแสดงการแจ้งเตือนข้อผิดพลาด

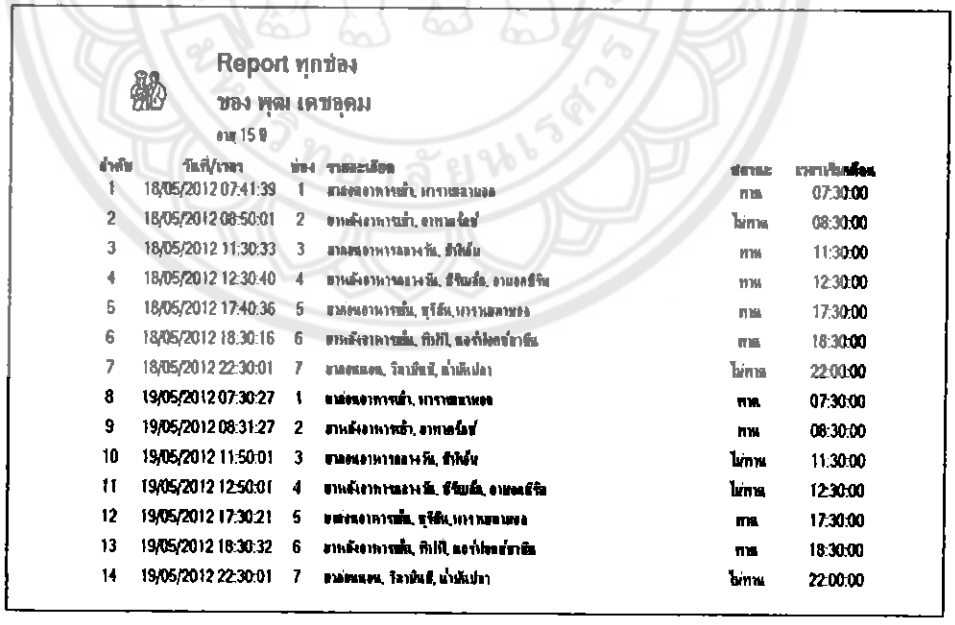
4.2.5 ขั้นตอนการแสดงผล

เมื่อทำการตั้งค่าผ่านทางหน้าต่างโปรแกรมวิชวลเบสิกแล้ว ข้อมูลดังกล่าวจะถูกส่งผ่าน Serial port ไปยังบอร์ดไมโครคอนโทรลเลอร์ AVR จากนั้นไมโครคอนโทรลเลอร์ AVR จะทำตามคำสั่งที่ได้รับมา เมื่อไมโครคอนโทรเลอร์ทำงาน โดยแสดงออกมาเป็นสัญญาณเตือน ถ้ามีการกดสวิตช์ ไมโครคอนโทรเลอร์จะส่งค่ากลับไปยัง คอมพิวเตอร์ จากนั้นคอมพิวเตอร์จะอ่านค่า Text File จาก Micro SD Card จะแสดงรายละเอียดในส่วนต่างๆดังนี้

1. กดเลือกรายชื่อผู้ป่วยจากรายชื่อที่มีอยู่
2. กดเลือกช่วงเวลาที่เราต้องการทราบข้อมูล
3. กดเลือกช่องมื้อยา (ยาก่อนอาหารเช้า-กลางวัน-เย็น, ยาหลังอาหารเช้า-กลางวัน-เย็น, ยาก่อนนอน อื่นๆ หรือทุกช่อง)
4. กดเลือกสถานะ (ทาน-ไม่ทาน)
5. กดปุ่ม Get file ดึงไฟล์มาจาก Text File เพื่อบันทึกค่าเก็บไว้ในโฟลเดอร์ Receive
6. กดปุ่ม Search สำหรับค้นหาข้อมูลผู้ป่วยแต่ละราย
7. ตารางแสดงรายละเอียดข้อมูล ลำดับ, เวลาเริ่มเตือน, วันที่/เวลารับประทานยา, ช่อง, รายละเอียด (ชื่อยา), และสถานะ (ทาน-ไม่ทาน)
8. กดปุ่ม Print เพื่อแสดงรายละเอียดต่างๆในการรับประทานยาออกมา สามารถพิมพ์เพื่อนำมาดูสถิติการรับประทานยาของผู้ป่วยแต่ละราย ดังรูปที่ 4.11
9. กดปุ่ม Delete file สำหรับลบข้อมูลในMemomy (SD card) แต่ยังคงข้อมูลล่าสุดที่มีการ Get file ไว้อยู่
10. กดปุ่ม Close เพื่อปิดหน้าต่างแสดงผล



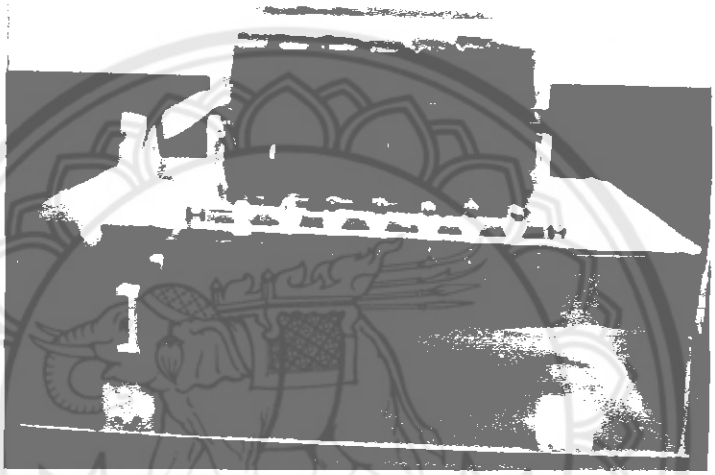
รูปที่ 4.10 หน้าต่างแสดงผลการรับประทานยา



รูปที่ 4.11 การพิมพ์รายงานผลการรับประทานยา

4.3 ผลการทดสอบการทำงานสัญญาณเตือนการรับประทานอาหาร

ตั้งค่าเวลารับประทานอาหารผ่านทางหน้าต่าง วิชวลเบสิก เมื่อถึงเวลารับประทานอาหาร จะมีเสียงเตือนพร้อมไฟแอลอีดีสีน้ำเงินเตือนจะเห็นได้ดังรูปที่ 4.12 แต่ถ้าหากไม่มีการส่งค่ามายังบอร์ดไมโครคอนโทรลเลอร์ในส่วนการรับประทานอาหาร หลอดแอลอีดีสีน้ำเงินจะไม่มีการแสดงค่าใดๆ



รูปที่ 4.12 การแสดงหลอดไฟแอลอีดีรับประทานอาหาร

4.4 ผลการทดสอบการทำงานของสัญญาณเตือนการรับประทานยา

สัญญาณเตือนการรับประทานยา ทำการทดสอบร่วมกับกล่องยาที่มีการติดหลอดแอลอีดีไว้ที่ฐาน โดยสามารถจำแนกได้ดังนี้

4.4.1 ผลการทดสอบกรณีที่มีการหยิบยารับประทานในช่วงเวลาที่สัญญาณเตือน

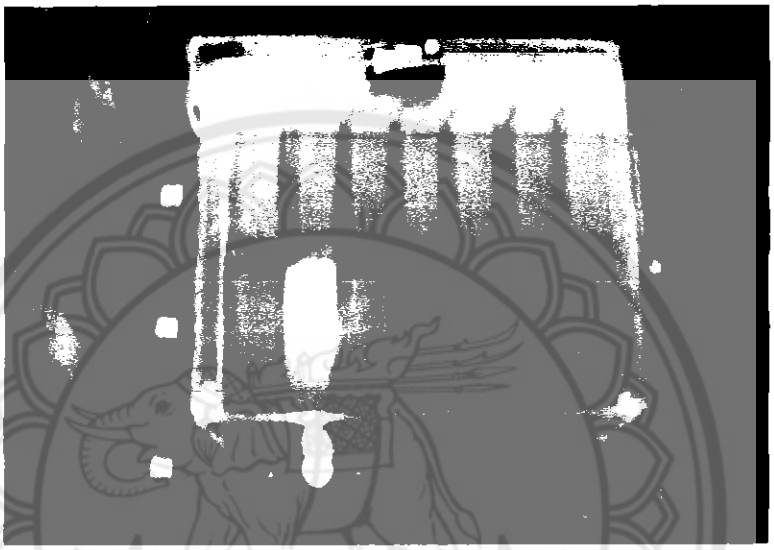
เมื่อตั้งค่าเวลาการรับประทานยาในแต่ละมื้ออาหารผ่านทางหน้าต่างโปรแกรมวิชวลเบสิก เมื่อถึงเวลาดังกล่าวสัญญาณไฟจากหลอดไฟแอลอีดี จะติดเป็นสีส้มด้านหน้าช่องยา และสีเหลืองใต้ฐานของช่องยานั้น พร้อมกับมีเสียงสัญญาณเตือนจากบัซเซอร์ดังไปพร้อมๆกัน เป็นระยะเวลาเท่าที่กำหนดตามการตั้งค่า จนเมื่อมีการเปิดกล่องยา ฝากล่องจะกดกับลิมิตสวิตช์ทำให้บันทึกค่าเวลา ณ เวลาที่กดสวิตช์ลงใน SD-Card เมื่อฝากล่องยาถูกเปิดค้างไว้ เสียงเตือนยังดัง จนกระทั่งปิดฝากล่องกลับคืนเหมือนเดิม สัญญาณเตือนดังกล่าวจึงหยุดการทำงาน แต่สำหรับหลอดไฟแอลอีดี

จะติดอยู่ 15 วินาที เพื่อป้องกันการหยิบยาผิดช่อง โดยการติดของหลอดไฟดูได้จากรูปที่ 4.13 เมื่อครบเวลาที่กำหนดไฟแอลอีดีจะดับแสดงดังรูปที่ 4.14 การควบคุมจะเป็นลักษณะนี้สำหรับทุกๆ มื้ออาหารเรียงตามลำดับกันไป

รูปที่ 4.13 การติดของหลอดไฟแอลอีดี เมื่อถึงเวลารับประทานยา

รูปที่ 4.14 หลอดไฟแอลอีดีดับ เมื่อครบกำหนดเวลา

และสำหรับกรณีที่เมื่อมีสัญญาณเตือนเวลาการรับประทานยาจนครบเวลาที่ให้เตือนแล้วนั้น แต่ยังไม่มีผู้หยิบยาไปรับประทาน สัญญาณเตือนจะหยุดเตือนไปสักพัก แล้วส่งสัญญาณเตือนใหม่อีกครั้งตามระยะเวลาเตือนที่ได้ตั้งค่าไว้ และจะเป็นเช่นนี้ไปเรื่อยๆ จนกว่าจะมีผู้ใช้หยิบยาไปรับประทาน แต่หลอดแอลอีดีจะติดอยู่เสมอไปจนกว่าจะหมดช่วงเวลาการเตือนหรือเวลาสิ้นสุดการเตือนนั่นเอง



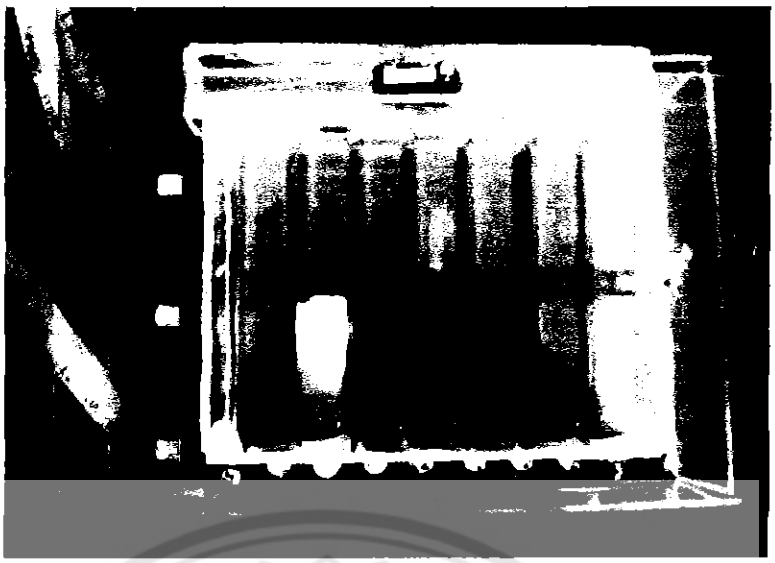
รูปที่ 4.15 หลอดไฟแอลอีดีติดตลอดช่วงเวลาที่กำหนด

4.4.2 ผลการทดสอบกรณีที่เมื่อหยิบยาแล้วฉีมปิดฝากล่องยา

เป็นกรณีที่เมื่อมีสัญญาณเสียงเตือนและผู้ป่วยหยิบยาไปรับประทาน แต่ลืมปิดฝากล่องยา กรณีนี้จะมีสัญญาณเสียงเตือนดังไปจนกว่าจะมีการปิดฝากล่องยา เมื่อปิดฝากล่องยาแล้วสัญญาณเสียงจะหยุดลง แต่หลอดไฟแอลอีดียังคงติดอยู่ 15 วินาที แล้วจึงดับ พร้อมกับรอการเตือนในมื้อยาถัดไป

4.4.3 ผลการทดสอบกรณีที่ไม่มีการหยิบยาในช่วงเวลาที่กำหนด

กรณีที่เมื่อมีสัญญาณเตือนเวลาการรับประทานยาจนครบเวลาที่ให้เตือนแล้วนั้น แต่ยังไม่มีผู้หยิบยาไปรับประทาน จนกระทั่งหมดช่วงเวลาการรับประทานยาในมื้อยานั้น ระบบควบคุมจะเริ่มสัญญาณเตือนในมื้อยาถัดไป และจะเป็นลักษณะนี้ในแต่ละกรณี จนหมดจำนวนมื้อยาที่ตั้งค่าไว้ในแต่ละวัน แสดงดังรูปที่ 4.16 และรูปที่ 4.17



รูปที่ 4.16 หลอดไฟแอลอีดีติดค้างจนหมดช่วงเวลาเตือน

รูปที่ 4.17 หลอดไฟแอลอีดีเริ่มทำงานที่ช่วงเวลาใหม่

บทที่ 5
สรุปผลและข้อเสนอแนะ

จากการดำเนินโครงงานสามารถสรุปผล ชี้แจงปัญหาที่เกิดขึ้นในระหว่างการดำเนินงาน รวมทั้งเสนอแนวทางแก้ปัญหา พร้อมให้ข้อเสนอแนะในการนำโครงงานไปพัฒนาต่อไป

5.1 สรุปผลการดำเนินโครงงาน

ในโครงงานนี้ได้นำบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASEMEGA1280/2560 มาใช้เพื่อควบคุมการทำงานของกล่องยาเตือน ใช้การเขียนโปรแกรมภาษาด้วยภาษาซี ควบคุมการทำงานของหลอดไฟและเสียงไฟเตือน เมื่อถึงเวลารับประทานยาหรือรับประทานอาหาร รวมถึงทำการบันทึกข้อมูลการรับประทานยาของผู้ป่วยรายนั้นๆ เพื่อสามารถนำมาวิเคราะห์ต่อผลการรักษา โดยมีการทำแบบจำลองกล่องยาเพื่อใช้ในการทดสอบ

รูปแบบการทำงานคือ จะต้องมีการตั้งค่าช่วงเวลาในการรับประทานยาหรือรับประทานอาหาร โดยตั้งค่าจากหน้าจอคอมพิวเตอร์ด้วยหน้าต่าง Visual Basic ที่ผู้จัดทำได้จัดสรรไว้ เมื่อใส่ค่าแล้วจึงทำการส่งค่าลงไปยังตัวบอร์ดไมโครคอนโทรลเลอร์ เมื่อถึงเวลากำหนดรับประทานยาจะมีเสียงเตือนและไฟแสดงช่องยาจะติด เมื่อมีการเปิดฝากล่องยาจะทำการบันทึกค่าว่าผู้ป่วยได้หยิบยาไปใช้แล้ว ในกรณีที่ไม่มีผู้ป่วยมาหยิบยาไปรับประทาน จะมีการเตือนซ้ำโดยดูจากเวลาที่ตั้งไว้ในหน้าต่าง Visual Basic ในคราวแรก หากไม่มีการหยิบยาไปรับประทาน ก็จะถูกบันทึกค่าว่าผู้ป่วยไม่ได้หยิบยาไปทาน โดยข้อมูลเหล่านั้นสามารถนำมาวิเคราะห์ลักษณะการทานยาของผู้ป่วยได้

5.2 ปัญหาและแนวทางแก้ไข

1. ในการใช้งาน ผู้ตั้งค่าการใช้งานด้วยหน้าจอ Visual Basic จะต้องมีความเข้าใจในส่วนนี้ด้วย มิฉะนั้นอาจจะทำให้ตั้งค่าผิด ส่งผลต่อการทานยาของผู้ป่วย ดังนั้นจึงต้องมีการจัดทำคู่มือในการตั้งค่าหน้าจอ Visual Basic เพื่อให้ผู้ตั้งค่า ไม่เกิดการผิดพลาดในการใช้งาน

2. ควรมีข้อมูลของผู้ป่วย 1 ราย ต่อการใช้งานบอร์ดไมโครคอนโทรลเลอร์ 1 บอร์ด เนื่องจากในการโปรแกรมลงบอร์ดแต่ละบอร์ด หากโปรแกรมทำงานซ้ำกัน อาจส่งผลให้เกิดความผิดพลาดในการรับประทานยาได้

5.3 แนวทางในการพัฒนาต่อไป

1. กรณีที่มีการมีบันทึกค่าการใช้ยาในผู้ป่วยหลายราย การค้นหารายชื่อจากปุ่มเลือกจึงทำให้เกิดความยุ่งยาก ควรมีการพัฒนาโปรแกรมให้สามารถเสิร์ชค้นหารายชื่อผู้ป่วย เพื่อให้ค้นหารายชื่อได้รวดเร็วขึ้น

2. การใช้หน้าต่างวิชวลเบสิกในส่วนการตั้งค่าเวลารับประทานยา อาจเพิ่มปุ่มกด Add สำหรับเลือกช่องยาที่ต้องการใช้งานได้ ในกรณีผู้ใช้งานไม่ต้องการใช้งานช่องยาทั้งหมด

เอกสารอ้างอิง

[1] แสงเดือน. (29 ตุลาคม 2551). ประกาศขาย.สืบค้นเมื่อ 12 กรกฎาคม 2554,จาก http://www.one2mall.com/detail/7A0002866

[2] Shenzhen Ruixinlong Electronics Co., Ltd. ,**Alarm Pill Box Timer (JT351).** Retrieved July 12 2011, from http://cnjinruitai2010zd.en.made-in-china.com/product/tqYEaKAvITiH/China-Alarm-Pill-Box-Timer-JT351-.html

[3] @buyourflowers.com. (December 01 2010). **Seniors in Florida: Electronic Pill Reminder_Dispenser: Medication Alert: Pill Organizer.** Retrieved July 12 2011, from http://www.prlog.org/11117602-seniors-in-florida-electronic-pill-reminder-dispenser-medication-alert-pill-organizer.html

[4] บริษัท อีทีที จำกัด. ET-BASE MEGA2560. กรุงเทพฯ: สำนักพิมพ์อีทีที.

[5] ETT CO.,LTD. **MODULE ET-MINI DS1307**.Retireved January 25, 2012, from http://www.vk5dj.com/MINIDS1307.pdf

[6] บริษัทไมโครซอฟท์. (25 มิถุนายน 2541). **Visual Basic Version 6.0**. สืบค้นเมื่อ 20 สิงหาคม2554,จาก http://th.wikipedia.org/wiki/%E0%B8%A7%E0%B8%B4%E0%B8%8A%E0%B8%A7%E0%B8%A5%E0%B9%80%E0%B8%9A%E0%B8%AA%E0%B8%B4%E0%B8%81

[7] บริษัท อีทีที จำกัด. ET-AVRISP mkII. สืบค้นเมื่อ 15 สิงหาคม 2554, จาก http://www.etteam.com/prod2010/ET-AVR/ET-AVR%20ISP%20mkII%20.html

[8]webmaster@wind-tek.net. **การใช้งาน RTC (Real Time Clock) ด้วย DS1307.** สืบค้นเมื่อ 16 กันยายน 2554.จาก http://www.mind-tek.net/ds1307.php

[9] ประจิน พลังสันติกุล. **การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ AVR ด้วยภาษา C กับ WinAVR (C Compiler) (C Programming for AVR Microcontroller and WinAVR (C Compiler)).** กรุงเทพฯ: สำนักพิมพ์ บริษัท แอพซอฟต์เทค จำกัด. 2551. บริษัทที่ผลิต ATMEL

[10] ประจิน พลังสันติกุล. **การประยุกต์ใช้งานภาษา C สำหรับไมโครคอนโทรลเลอร์ AVR เล่ม 2 (Apply C Programming for AVR Microcontroller Book 2).** กรุงเทพฯ: สำนักพิมพ์ บริษัท แอพซอฟต์เทค จำกัด. 2551.บริษัทที่ผลิต ATMEL เมื่อ 27 พฤศจิกายน 2545.

[11] Wisit P. (1 เม.ย. 2554). **การติดต่อสื่อสารด้วย SPI : Serial Peripheral Interface.** สืบค้นเมื่อ 16 กันยายน 2554. จาก http://www.123microcontroller.com/Hardware-Interfacing/SPI-Serial-Peripheral-Interface-communication

[12] Miss Vichulada Kum-ay. **บทที่ 1 OSI Model (Open System Interconnection 7 – layer Reference Model).** สืบค้นเมื่อ 16 กันยายน 2554. จาก http://mymint.tripod.com/report5.html

[13] Webmaster. **โปรโตคอล SDLC (Synchronous Data Link Control).** สืบค้นเมื่อ 25 พฤศจิกายน 2554.จาก http://cptd.chandra.ac.th/selfstud/datacom/CAI/part3-7.htm#framesdlc

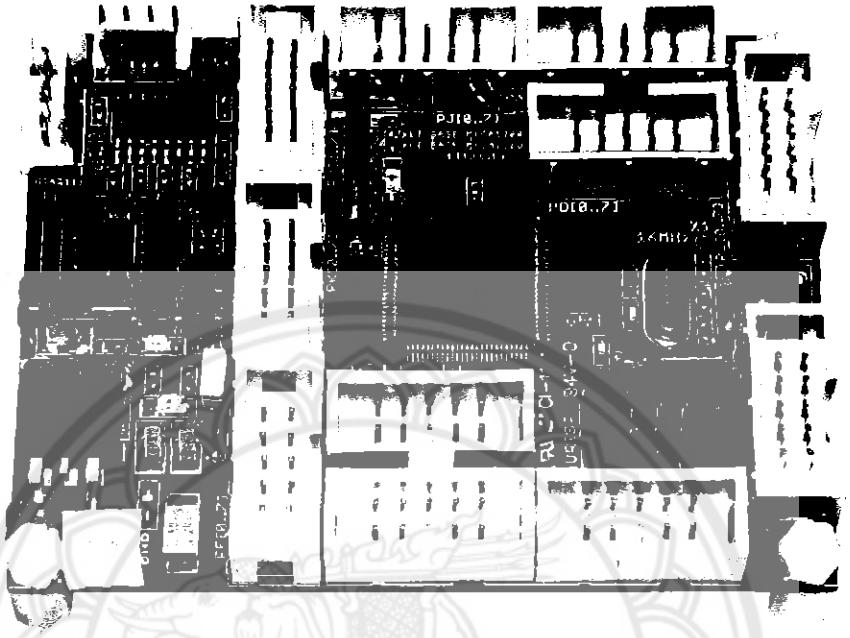
[14] webmaster.**Datalink leyer Cont.**http://ladysmilepasta.blogspot.com/2009_08_02_archive.html

[15] พรพิมล พจนาพิมล. (2551). **การศึกษาและพัฒนากล่องยาอิเล็กทรอนอกส์สำหรับยาชนิดเม็ดของผู้สูงอายุที่มีโรคประจำตัว.** วิทยานิพนธ์ ศป.ม., มหาวิทยาลัยศรีนครินทรวิโรฒ

ภาคผนวก ก

คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASE MEGA 2560

ET-BASE MEGA1280/2560



ไมโครคอนโทรลเลอร์ตระกูล AVR ของ ATMEL เป็นไมโครคอนโทรลเลอร์อีกตระกูลหนึ่งซึ่งได้รับความนิยมอย่างแพร่หลายจากผู้ใช้งานทั่วไป ซึ่งทาง ATMEL เองก็ได้มีการปรับปรุง พัฒนาขีดความสามารถของ MCU เพื่อตอบสนองความต้องการใช้งานในลักษณะต่างๆ มีการผลิตชิพ MCU ออกมาจำหน่ายเป็นจำนวนมากมายหลายเบอร์ เพื่อให้ผู้ใช้งานสามารถเลือก MCU ไปประยุกต์ใช้งานให้เหมาะสมกับงานได้ง่ายและสะดวกมากยิ่งขึ้น

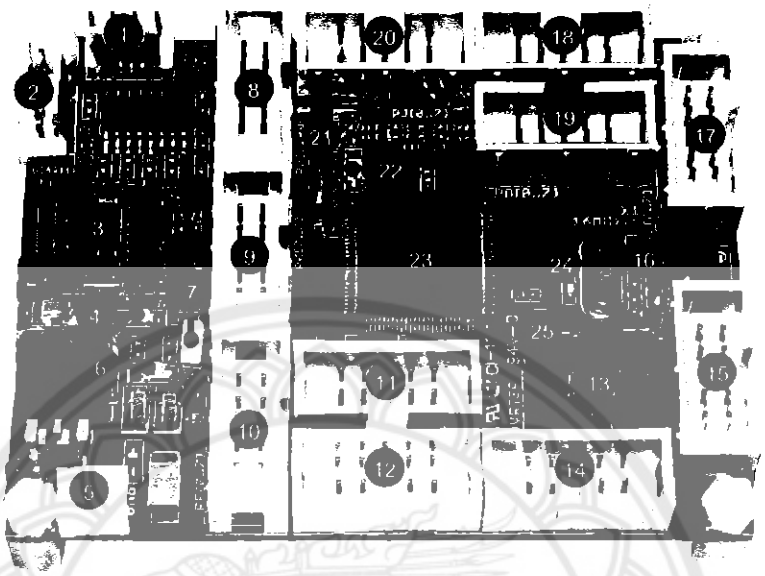
ATMEGA ก็เป็น MCU ตระกูล AVR ที่มีความโดดเด่นและมีขีดความสามารถสูงในระดับต้นๆของ MCU 8 บิต ซึ่งใน ATMEGA1280/2560 เองเป็น MCU ตระกูล AVR MEGA ที่มีระบบ Peripheral I/O ต่างๆรวบรวมไว้ภายใน MCU มากมายหลากหลาย สามารถโปรแกรมโหมดการทำงานของ Peripheral I/O ใน ลักษณะต่างๆได้หลากหลาย ทำให้ง่ายและสะดวกในการนำไปดัดแปลงใช้กับงานแบบต่างๆได้โดยง่าย ซึ่งการพัฒนาโปรแกรมของบอร์ดก็สามารถเลือกใช้รูปแบบในการพัฒนาโปรแกรมด้วยโปรแกรมภาษาต่างๆที่สนับสนุนการใช้งานกับ AVR MEGA ได้ทั่วไป ตามความเหมาะสม

โดยโครงสร้างของบอร์ดได้ออกแบบให้มีความสะดวกในการพัฒนาโปรแกรม และ ประยุกต์ใช้งานได้โดยสะดวก โดยมีพอร์ตสื่อสาร RS232 และ Micro-SD Card เป็นอุปกรณ์พื้นฐานภายในบอร์ด ส่วน GPIO ต่างๆจะออกแบบเป็น IDE Connector ไว้ให้เพื่อให้เกิดความสะดวกในการเชื่อมต่อออกไปใช้งาน โดยได้เพิ่มช่องทางในการพัฒนาโปรแกรมได้ทั้งการโปรแกรมผ่าน Bootloader หรือ ISP Programmer หรือ JTAG Interface สำหรับโปรแกรมและ Debug การทำงานได้อีกด้วย

คุณสมบัติของบอร์ด ET-BASE MEGA1280/2560

- ใช้ MCU ตระกูล MEGA AVR เบอร์ ATMEGA1280/2560 ของ ATMEL
- มีหน่วยความจำ Flash 128KB(ATMEGA1280,256K(ATMEGA2560), 8KB Boot loader, Static RAM 8KB และ EEPROM 4KB
- ใช้ Crystal 16.00 MHz
- มีวงจร Real Time Counter พร้อม XTAL ค่า 32.768KHz
- รองรับการโปรแกรมแบบ In-System Programming แบบ ISP
- มีวงจรเชื่อมต่อกับ AVR-JTAG ขนาด 10 Pin เพื่อทำการ Debug แบบ Real Time ได้
- Power Supply ใช้แรงดันไฟฟ้า +5VDC พร้อมวงจร Regulate +3V3/3A ภายในบอร์ด พร้อม Jumper เลือกระบบแหล่งจ่ายให้เป็น 3.3V หรือ 5V ได้ตามต้องการ
- มีวงจรเชื่อมต่อการ์ดหน่วยความจำแบบ SD Card(Micro SD) เชื่อมต่อแบบ SPI จำนวน 1 ช่อง
- มีวงจรสื่อสาร RS232 โดยใช้ขั้วต่อแบบ 4-PIN มาตรฐาน ETT จำนวน 2 ช่อง
- มีวงจร LED แสดงสถานะเพื่อทดลอง Output(PG5) จำนวน 1 ชุด
- มี 83 Bit GPIO อิสระ สำหรับประยุกต์ใช้งานต่างๆ เช่น A/D, I2C, SPI, USART และ Input / Output แบบต่างๆ โดยมีการจัดสรรใช้งานภายในบอร์ดไว้แล้ว จำนวน 8 เส้นสัญญาณ คือ PB[0..3] สำหรับ micro SD Card, PD[2..3] และ PE[0..1] สำหรับ RS232 แต่สัญญาณทั้ง 8 เส้น ดังกล่าวยังมีการเชื่อมต่อสัญญาณออกมาไว้ที่ขั้วต่อ 10PIN IDE ของพอร์ตด้วย โดยใช้ขั้วต่อสัญญาณแบบ 10PIN IDE จำนวน 10 ชุด มีดังนี้
  - Header 10Pin IDE (PA[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PB[0..7]) สำหรับ GPIOและ PB[0..3] ถูกใช้สำหรับ SPI ในการเชื่อมต่อกับ micro SD Card
  - Header 10Pin IDE (PC[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PD[0..7]) สำหรับ GPIO(PD[2..3] ถูกใช้สำหรับ USART0)
  - Header 10Pin IDE (PE[0..7]) สำหรับ GPIO(PE[0..1] ถูกใช้สำหรับ USART1)
  - Header 10Pin IDE (PF[0..7]) สำหรับ GPIO(PF[4..7])ถูกใช้สำหรับ JTAG)
  - Header 10Pin IDE (PH[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PJ[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PK[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PL[0..7]) สำหรับ GPIO
  - Header 1x5Pin (PG[0..2]) สำหรับ GPIO

โครงสร้างบอร์ด ET-BASE MEGA1280/2560

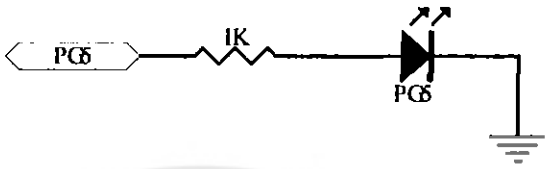


รูปแสดง โครงสร้างของบอร์ด ET-BASE MEGA1280/2560

- หมายเลข 1,2 คือ ขั้วต่อ USART0(RS232) และ USART1(RS232) สำหรับใช้งาน
- หมายเลข 3,4 คือ ช่องเสียบการ์ดหน่วยความจำ SD Card แบบ Micro-SD และ LED สถานะ
- หมายเลข 5,6 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ดใช้ได้กับไฟ +5VDC และ LED Power
- หมายเลข 7 คือ Jumper สำหรับเลือกขนาดแรงดันทำงานของ MCU ระหว่าง 3.3V หรือ 5V
- หมายเลข 8,9,10,11 คือ ขั้วต่อ GPIO(PA[0..7]) ,(PK[0..7]),(PF[0..7]) และ GPIO(PE[0..7])
- หมายเลข 12 คือ ขั้วต่อ AVR-JTAG สำหรับ Debug แบบ Real Time
- หมายเลข 13 คือ ขั้วต่อ AVRISP สำหรับ Download โปรแกรม
- หมายเลข 14,15 คือ LED ขั้วต่อ GPIO(PH[0..7]) และ GPIO(PB[0..7])
- หมายเลข 16 คือ SW RESET
- หมายเลข 17,18,19,20 คือ ขั้วต่อ GPIO(PL[0..7]), (PC[0..7]),(PD[0..7]) และ GPIO(PJ[0..7])
- หมายเลข 21,22 คือ ขั้วต่อ GPIO(PG[0..2]) และ LED ใช้ทดสอบ Logic Oulpul ของ PG5
- หมายเลข 23 คือ MCU เบอร์ ATMEGA1280 หรือ ATMEGA2560 (100Pin TQFP)
- หมายเลข 24 คือ Crystal ค่า 16.00 MHz สำหรับใช้เป็นฐานเวลาระบบให้ MCU
- หมายเลข 25 คือ Crystal ค่า 32.768KHz สำหรับฐานเวลาให้ RTC ภายในตัว MCU

การใช้งานวงจรขับ LED แสดงผล

LED แสดงผลของบอร์ด จะต่อวงจรแบบขับกระแส (Source Current) ทำงานด้วยโลจิก "1" และหยุดทำงานด้วยโลจิก "0" โดยควบคุมการทำงานจากขาสัญญาณ PG5 โดยวงจรในส่วนนี้จะใช้สำหรับทดสอบการทำงานของ Output จากขาสัญญาณ PG5



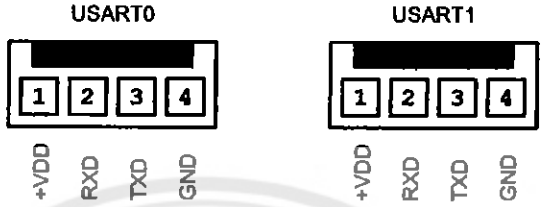
โดยเมื่อต้องการใช้งานผู้ใช้ต้องกำหนดให้ PG[5] ทำหน้าที่เป็น GPIO Output Port เสียก่อนแล้วจึงควบคุม Logic ให้กับ PG[5] ตามต้องการ ดังตัวอย่าง

```
#define PORT_LED PORTG          // Port Drive LED = PG
#define DIR_LED  DDRG           // Port Direction
#define LED 5                   // Pin Drive LED = PG5
.
.
.
int main(void)
{
  DIR_LED |= (1<<LED);          // Pin Drive LED = Out
  .
  .
  .
  PORT_LED &= ~(1<<LED);        // Pin LED = 0(OFF LED)
  .
  .
  .
  PORT_LED |=  (1<<LED);        // Pin LED = 1(ON LED)
  .
  .
  .
  PORT_LED ^=  (1<<LED);        // Pin LED = Toggle
  .
  .
  .
}
```

ตัวอย่าง การกำหนดค่าการใช้งาน PG5 เป็น Output LED

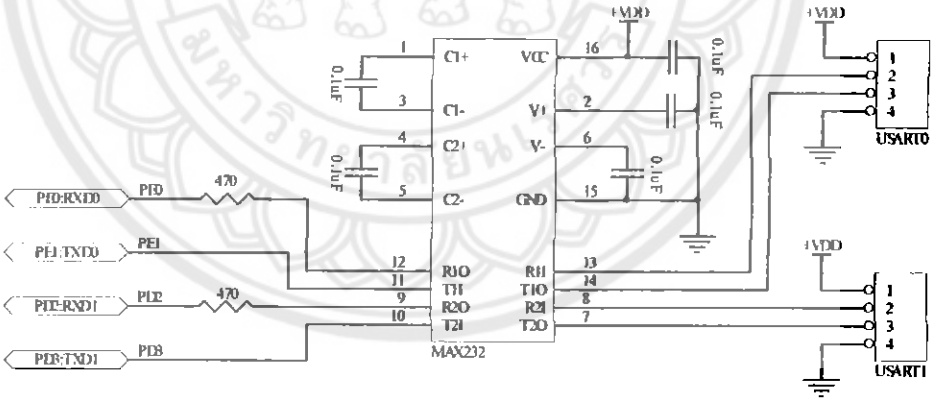
พอร์ต RS232

เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณ MAX3232 เรียบร้อยแล้ว โดยมีจำนวน 2 ช่อง ด้วยกันคือ USART0 และ USART1 โดยทั้ง 2 ช่องสามารถใช้เชื่อมต่อกับสัญญาณ RS232 เพื่อรับส่งข้อมูลได้



- USART0 ใช้ขาสัญญาณจาก PE0(RXD) และ PE1(TXD)
- USART1 ใช้ขาสัญญาณจาก PD2(RXD) และ PD3(TXD)

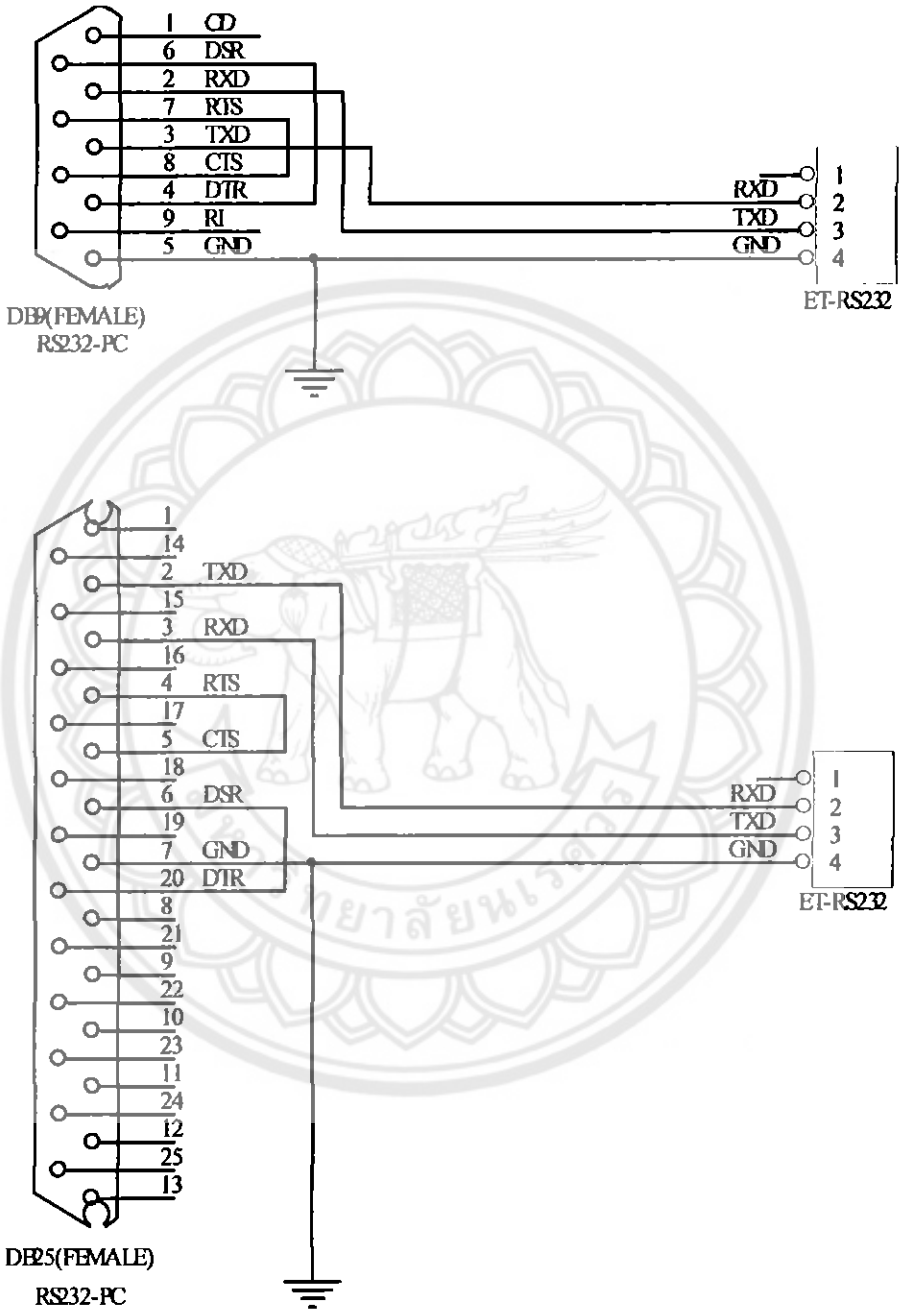
เนื่องจากระบบ Hardware USART ของ ATMEGA1280/2560 นั้นจะมี USART ไว้ใช้งานมากถึง 4 ชุด คือ PE0,PE1 และ PD2,PD3 และ PH0,PH1 และ PJ0,PJ1 โดยในกรณีของบอร์ด ET-BASE MEGA1280/2560 จะจัดวงจร USART ร่วมกับวงจร Line Driver ของ RS232 ไว้ให้จำนวน 2 ช่อง คือ PE0,PE1 และ PD2,PD3 ดังวงจร



```
#include <avr/io.h>
#include <stdio.h>
#define F_CPU 16000000UL                    //16.0 MHz
#define BAUD 9600                           //9600 BPS
#define MYUBRR F_CPU/16/BAUD-1

void init_serial(unsigned int ubrr);        //Initil UART
int   my_putchar(char c, FILE *stream);
int   my_getchar(FILE *stream);

/* Retarget STDIO(putchar,getchar of printf) to My Function */
FILE uart_str = FDEV_SETUP_STREAM(my_putchar, my_getchar, _FDEV_SETUP_RW);

int main(void)
{
  init_serial(MYUBRR);                      //Initilial UART0 = 9600,N,8,1
  stdout = stdin = &uart_str;

  printf("Hello World\n\r");
  while(1)
  {
    putchar(getchar());                     //Echo Receive Char
  }
}

/****************/
/* Initial UART */
/****************/
void init_serial(unsigned int ubrr)
{
  UBRR0H = (unsigned char)(ubrr>>8);        //Set baud rate
  UBRR0L = (unsigned char)ubrr;
  UCSR0B = (1<<RXEN0)|(1<<TXEN0);           //Enable RX and TX
  UCSR0C = (1<<USBS0)|(3<<UCSZ00);          //8data, 2stop bit
}

/*****************************/
/* Write 1 Character To UART */
/*****************************/
int my_putchar(char c, FILE *stream)
{
  while (!(UCSR0A & (1<<UDRE0)));           // Wait TXD Buffer Empty
  UDR0 = c;
  return 0;
}

/*****************************/
/* Get 1 Character From UART */
/*****************************/
int my_getchar(FILE *stream)
{
  while(!(UCSR0A & (1<<RXC0)));             // Wait RXD Receive Data Ready
  return (UDR0);                            // Get Receice Data & Return
}
```

ตัวอย่างการใช้งาน USART0 รับส่งข้อมูล

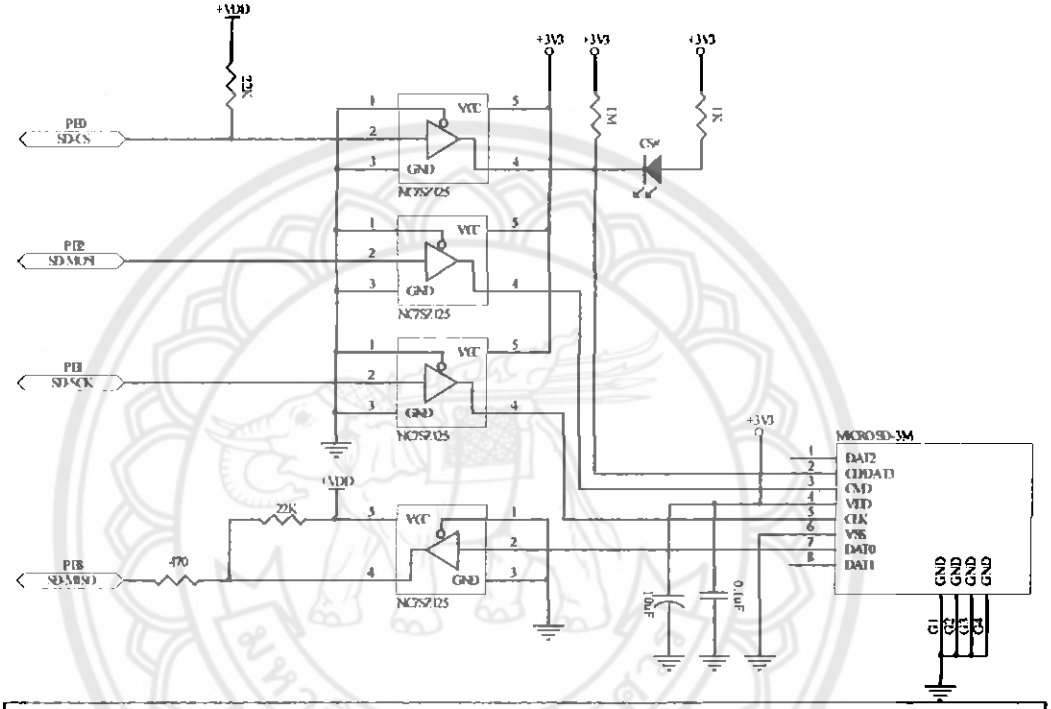
สำหรับ Cable ที่จะใช้ในการเชื่อมต่อ RS232 ระหว่าง Comport ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้วต่อ USART0 และ USART1 ของบอร์ด ET-BASE MEGA1280/2560 นั้น เป็นดังนี้



รูป แสดงวงจรสาย Cable สำหรับ RS232

การ์ดหน่วยความจำ SD Card แบบ Micro-SD

บอร์ด ET-BASE MEGA1280/2560 รองรับการเชื่อมต่อกับการ์ดหน่วยความจำ SD Card แบบ Micro-SD โดยใช้การเชื่อมต่อแบบ SPI โดยใช้ขาสัญญาณ PB[0..3] ในการเชื่อมต่อกับการ์ด ซึ่งในการติดต่อสั่งงาน การ์ดนั้น สามารถโปรแกรม Pin I/O ของ PB[0..3] ให้ทำงานในโหมด SPI โดยต้องกำหนดหน้าที่ของขาสัญญาณ PB[0..3] ของ MCU เป็นดังนี้

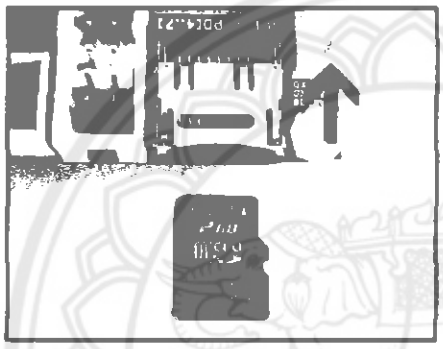



```
#define DDR_SPI DDRB
#define SD_CS   DDB0
#define SD_SCK  DDB1
#define SD_MOSI DDB2
.
.
// Set as master, clock and chip select output
DDR_SPI |= (1<<SD_MOSI)|(1<<SD_SCK)|(1<<SD_CS);  //MOSI,SCK,CS = Out
SPCR     = (1<<SPE)|(1<<MSTR);                   //SPI, Master

// SPI = Low Speed(Max. 400 kBit used in Card Initialization)
// SPI2X:SPR1:SPR0 = 0:1:1 (SCK = Fosc/128)
// SCK = Fosc  / 128
//     = 16MHz / 128
//     = 126KHz
SPSR &= ~(1 << SPI2X);                           // SPI2X = 0
SPCR |=  ((1 << SPR1) | (1 << SPR0));            // SPR1:SPR0 = 1:1
```

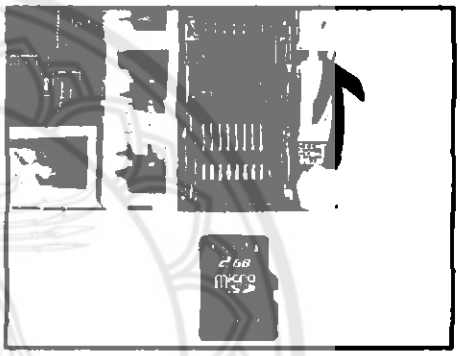
ตัวอย่าง การกำหนดค่า Pin สำหรับใช้งาน SD Card

การใส่การ์ดหน่วยความจำใน Socket

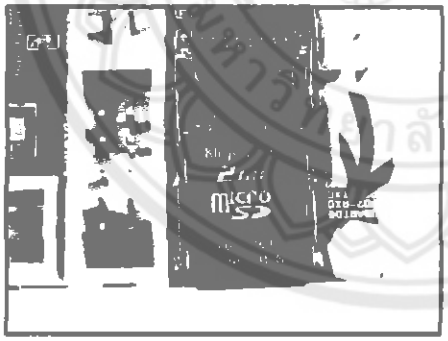
สำหรับ Socket สำหรับติดตั้งการ์ดหน่วยความจำของบอร์ด ET-BASE MEGA1280/2560 นั้น จะใช้ Socket หน่วยความจำแบบใส่การ์ดจากด้านบน แบบเดียวกันกับซิมการ์ดของโทรศัพท์มือถือ โดยเมื่อต้องการจะใส่หรือถอดการ์ดหน่วยความจำจะต้องทำการเปิดฝาครอบ Socket ออกเสียก่อน จากนั้นจึงจะสามารถใส่หรือถอดการ์ดหน่วยความจำได้ โดยการเปิด หรือ ปิด ฝาครอบ Socket จะใช้การกดเลื่อนฝาครอบเข้าหรือออก ซึ่งถ้าเลื่อนฝาครอบเข้าด้านในจะเป็นการเลื่อนเพื่อเปิดฝา แต่ถ้าเลื่อนออกด้านนอกจะเป็นการเลื่อนเพื่อล๊อคฝาครอบดังรูป



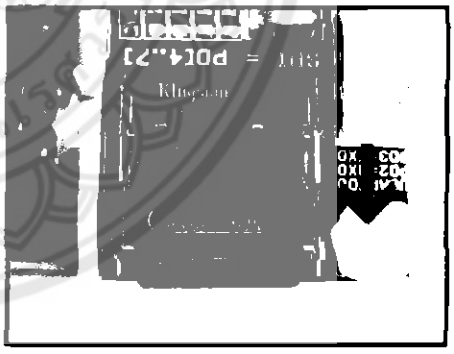
1. กดฝาครอบและเลื่อนเข้าเพื่อเปิดล๊อค



2. เปิดฝาครอบ Socket ออก



3. ใสการ์ดหน่วยความจำใน Socket

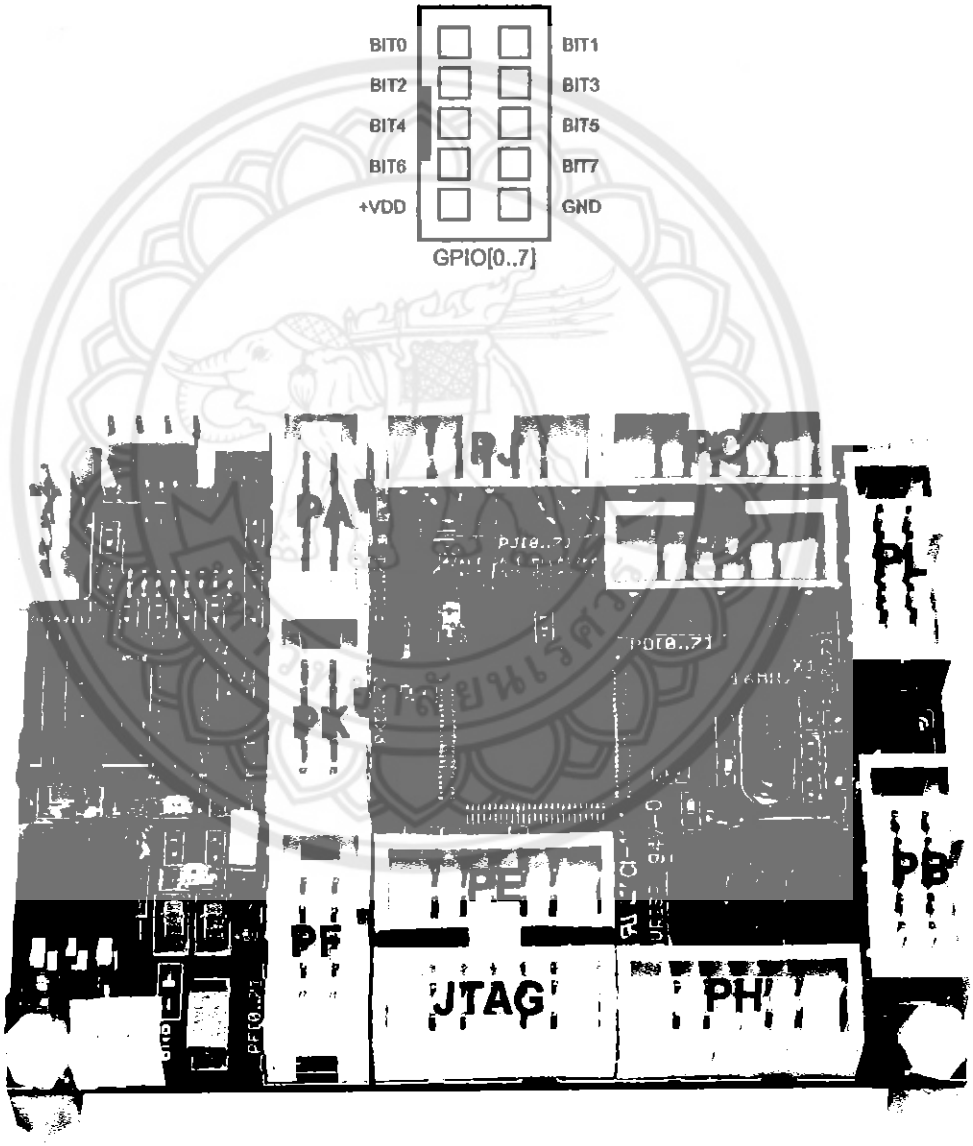


4. ปิดฝาครอบและกดเลื่อนออกเพื่อล๊อค

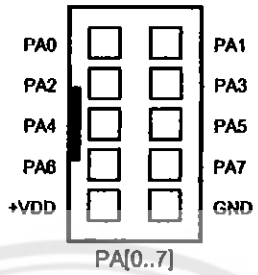
แสดงลำดับขั้นตอนการใส่การ์ดหน่วยความจำ

ขั้วต่อ Port I/O ต่างๆของบอร์ด

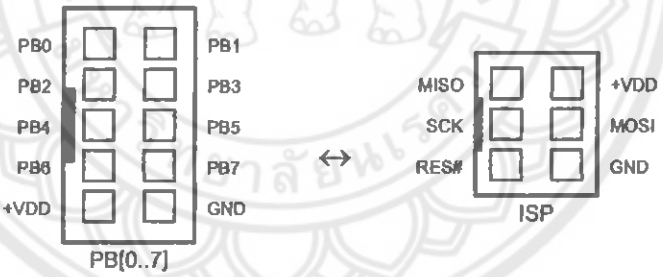
สำหรับขั้วต่อ Port I/O ของ MCU ของบอร์ด ET-BASE MEGA1280/2560 นั้น จะจัดเรียงออกมารอไว้ยังขั้วต่อแบบ IDE 10 Pin จำนวน 10 ชุดๆละ 8บิต สำหรับให้ผู้ใช้เลือกต่อออกไปใช้งานตามต้องการ โดยรูปแบบการจัดเรียงสัญญาณของแต่ละพอร์ต จะเรียงลำดับตำแหน่งบิต I/O เหมือนกัน โดยมีรายละเอียดของสัญญาณแต่ละชุดดังนี้



- ขั้วต่อ IDE 10 Pin ของ PA[0..7] โดย Port PA ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้

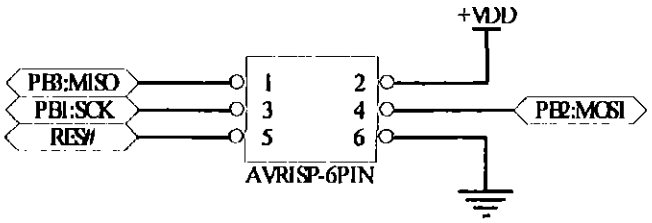


- ขั้วต่อ IDE 10 Pin ของ PB[0..7] โดย Port PB ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 นั้น จะมีการเชื่อมต่อ PB[1..3] ไปยังขั้วต่อ AVRISP และ PB[0..3] ก็จะถูกเชื่อมต่อเข้ากับวงจรของ Socket Micro SD-Card ด้วย ส่วน PB[4...7] จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ ซึ่งถ้ามีการต่อ PB[1..3] กับอุปกรณ์ภายนอกไว้ เมื่อต้องการโปรแกรมผ่าน AVRISP จะต้องปลดขาสัญญาณชุดนี้ให้เป็นอิสระจากอุปกรณ์ภายนอกด้วย ไม่เช่นนั้นอาจไม่สามารถโปรแกรมผ่าน AVRISP ได้ โดย PB[0..7] มีการจัดเรียงสัญญาณไว้ดังนี้

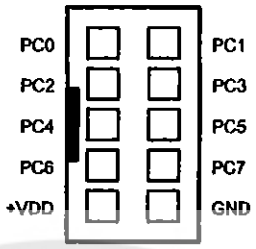


หมายเหตุ PB1...PB3 จะซ้อนทับกับ ISP Function ด้วย ซึ่งขาสัญญาณ PB1...PB3 ของบอร์ดจะถูกเชื่อมต่อไปยังขั้วต่อ AVRISP และ Micro SD-Card ด้วย โดยขาสัญญาณ PB1...PB3 เมื่อทำหน้าที่เป็น AVRISP จะมีหน้าที่ดังนี้

- PB1 = SCK
- PB2 = MOSI
- PB3 = MISO

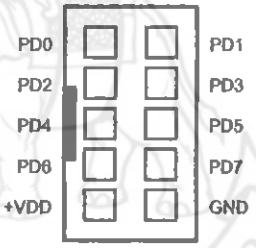


- ขั้วต่อ IDE 10 Pin ของ PC[0..7] โดย Port PC ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้



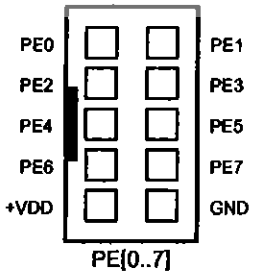
PC[0..7]

- ขั้วต่อ IDE 10 Pin ของ PD[0..7] โดย Port PD ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 นั้น จะมีการเชื่อมต่อ PD2...PD3 ไปยังวงจร USART1 ด้วย ส่วนสัญญาณ PD0...PD1 และ PD[4..7] จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้



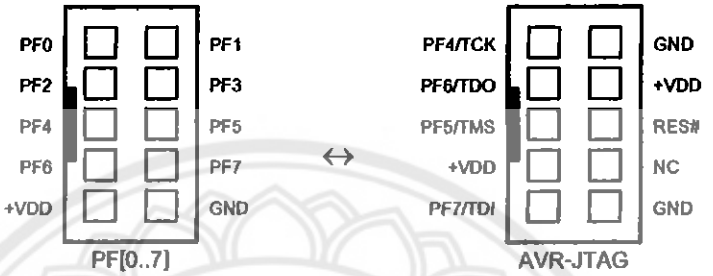
PD[0..7]

- ขั้วต่อ IDE 10 Pin ของ PE[0..7] โดย Port PE ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 นั้น จะมีการเชื่อมต่อ PE0...PE1 ไปยังวงจร USART0ด้วย ส่วน PE[2...7] จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้



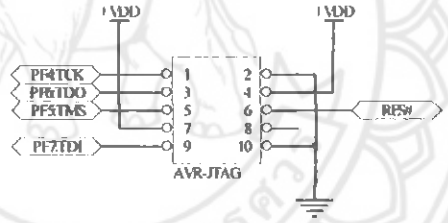
PE[0..7]

- ขั้วต่อ IDE 10 Pin ของ PF[0..7] โดย Port PF ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 นั้น จะมีการเชื่อมต่อ PF[4..7] ไปยังขั้วต่อสำหรับ AVR-JTAG ด้วย ส่วน PF[0...3] จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เอง ตามความต้องการ ซึ่งถ้าต้องการใช้งาน PF[4..7] ต้องไปสั่ง Disable Fuse Bit ของ JTAGEN ออกเสียก่อนด้วยเครื่องโปรแกรมทาง ISP Port โดย PF[0..7] มีการจัดเรียงสัญญาณไว้ดังนี้

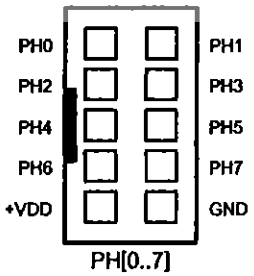


หมายเหตุ PF4...PF7 จะซ้อนทับกับ JTAG Function ด้วย ซึ่งขาสัญญาณ PF4...PF7 ของบอร์ดจะถูกเชื่อมต่อไปยังขั้วต่อ AVR-JTAG ด้วย ซึ่งถ้าผู้ใช้ได้เชื่อมต่อ AVR-JTAG ไว้ด้วย จะไม่สามารถใช้งานขาสัญญาณ PF4...PF7 ได้ โดยขาสัญญาณ PF4...PF7 เมื่อทำหน้าที่เป็น JTAG จะมีหน้าที่ดังนี้

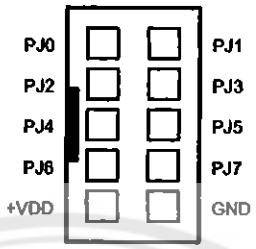
- PF4 = TCK JTAG
- PF5 = TMS JTAG
- PF6 = TDO JTAG
- PF7 = TDI JTAG



- ขั้วต่อ IDE 10 Pin ของ PH[0..7] โดย Port PH ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้

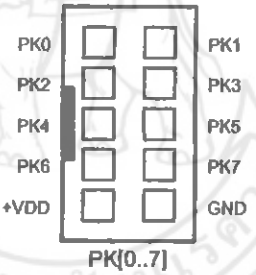


- ขั้วต่อ IDE 10 Pin ของ PJ[0..7] โดย Port PJ ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่าง ไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้



PJ[0..7]

- ขั้วต่อ IDE 10 Pin ของ PK[0..7] โดย Port PK ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่าง ไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้

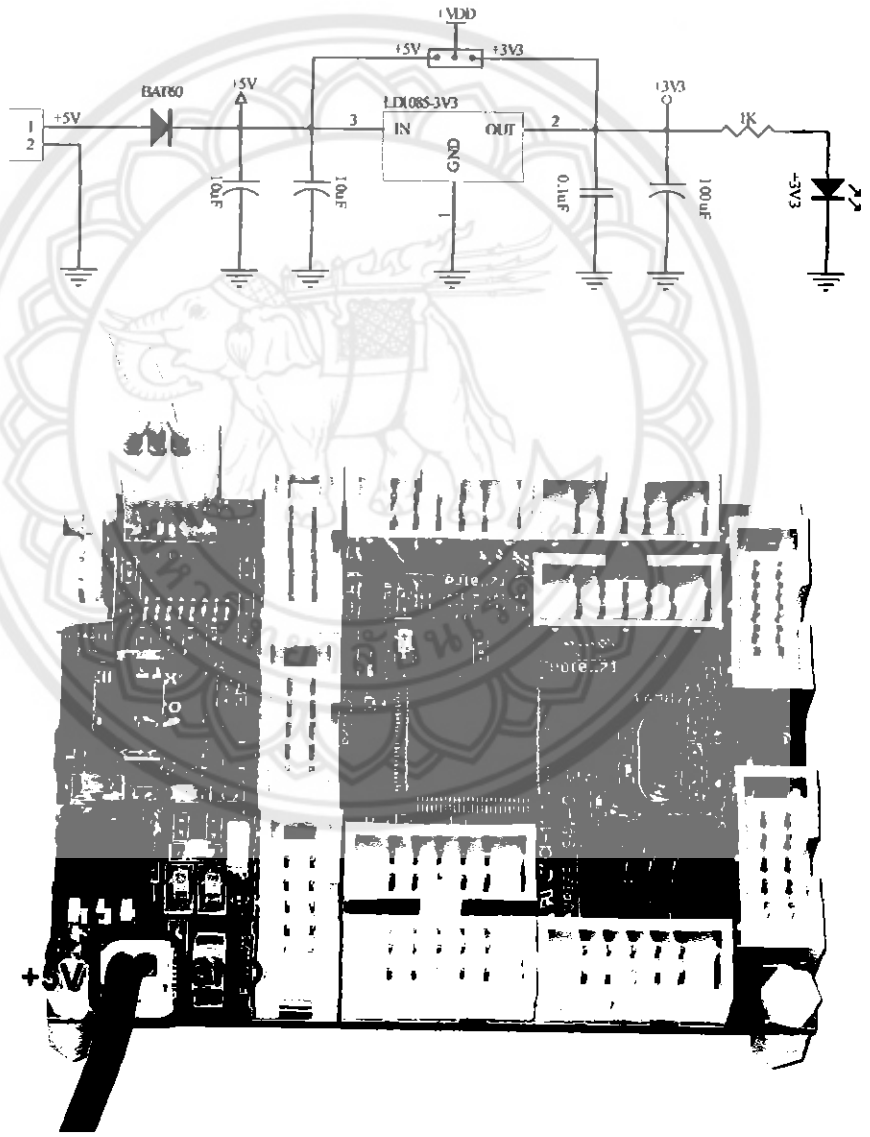


PK[0..7]

วงจรแหล่งจ่ายไฟ

วงจรแหล่งจ่ายไฟของบอร์ดใช้งานได้กับไฟ DC ขนาด +5V โดยใช้ขั้วต่อแบบ 2 Pin Block ป้องกันการเสียบสายกลับขั้ว พร้อมวงจร Regulate ขนาด +3V3/3A

โดยวงจรภาคแหล่งจ่ายไฟในส่วนที่เป็นวงจร Regulate ขนาด 3.3V นั้นจะจ่ายตรงไปให้กับ วงจร SD-Card ส่วนวงจรของ MPU และวงจร I/O ของบอร์ดนั้น จะสามารถเลือกขนาดแรงดันใช้งานได้จาก Jumper ว่าต้องการใช้งานเป็น 3V3 หรือ 5V ดังรูป



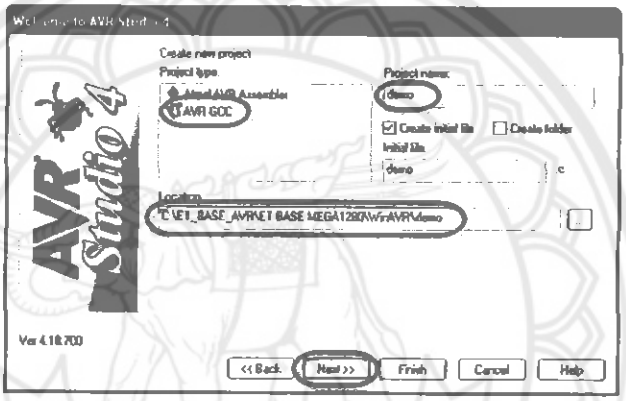
ตัวอย่างการพัฒนาโปรแกรมด้วย WinAVR ร่วมกับ AVR Studio4

ในการพัฒนาโปรแกรมของ AVR เบอร์ ATMEGA1280/2560 นั้น ตามปรกติแล้วจะสามารถเลือกใช้ Compiler ต่างๆที่สนับสนุน MCU เบอร์นี้อยู่ได้ทั้งหมด แต่ในที่นี้จะขอแนะนำให้ใช้โปรแกรม AVR Studio4 ร่วมกับ WinAVR ซึ่งเป็นชุดโปรแกรมที่แจกจ่ายให้ใช้ได้ฟรีๆไม่เสียค่าใช้จ่าย มีการพัฒนาปรับปรุงความสามารถของโปรแกรมกันอย่างต่อเนื่องและมีผู้ใช้งานกันอย่างแพร่หลายทั่วโลก สามารถค้นหาตัวอย่างโปรแกรม และ Library ต่างๆที่ผู้ใช้ต่างๆจำนวนมากได้สร้างและเผยแพร่ไว้มาเป็นแนวทางในการศึกษาได้มากมายซึ่งปัจจุบัน (สิงหาคม 2553) ทาง ATMEL ได้ทำการปรับปรุงโปรแกรม AVR Studion4 เป็นรุ่น V4.18 แล้ว ส่วนโปรแกรม WinAVR สามารถ Download ได้จาก http://winavr.sourceforge.net/ จะปรับปรุงเป็น WINAVR-20100110 แล้ว ซึ่งผู้ใช้สามารถไป Download มาติดตั้งใช้งานได้ฟรี โดยในที่นี้จะขอแนะนำแนวทางการพัฒนาโปรแกรมแบบพอสังเขป เพื่อเป็นแนวทางสำหรับผู้เริ่มต้น ซึ่งรายละเอียดต่างๆสามารถศึกษาได้จากคู่มือของโปรแกรมได้ โดยแนวทางการพัฒนาโปรแกรมของ ATMEGA1280 และ ATMEGA2560 โดยใช้โปรแกรม AVR Studio4 ร่วมกับ WinAVR มีลำดับขั้นตอนดังนี้

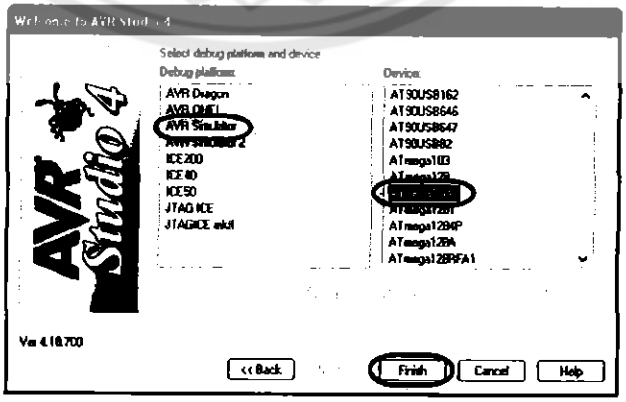
1. สั่ง Run Program AVR Studio4 ซึ่งจะได้ผลดังรูป

2. สั่งสร้าง project ใหม่ โดยเลือกที่ project → New project จากนั้นเลือกกำหนดตัวเลือกต่างๆ ให้กับโปรแกรมดังนี้

   - Project type เลือกกำหนดเป็น AVR GCC
   - Location สำหรับบันทึก project ให้ระบุตำแหน่ง Folder ที่ต้องการใช้บันทึกไฟล์ และ Codeต่างของ project
   - Project name ให้กำหนดชื่อ project ตามต้องการในตัวอย่างกำหนดเป็น "demo" และให้เลือก Create initial file ไว้ด้วย ซึ่งเมื่อเราทำการกำหนดชื่อ project name เสร็จแล้ว โปรแกรมจะสร้างไฟล์ ที่มีชื่อเดียวกันกับ project name ให้เองโดยอัตโนมัติ ซึ่งถ้าต้องการกำหนดชื่อไฟล์เป็นชื่ออื่น ก็ไม่ต้องเลือก Create initial file



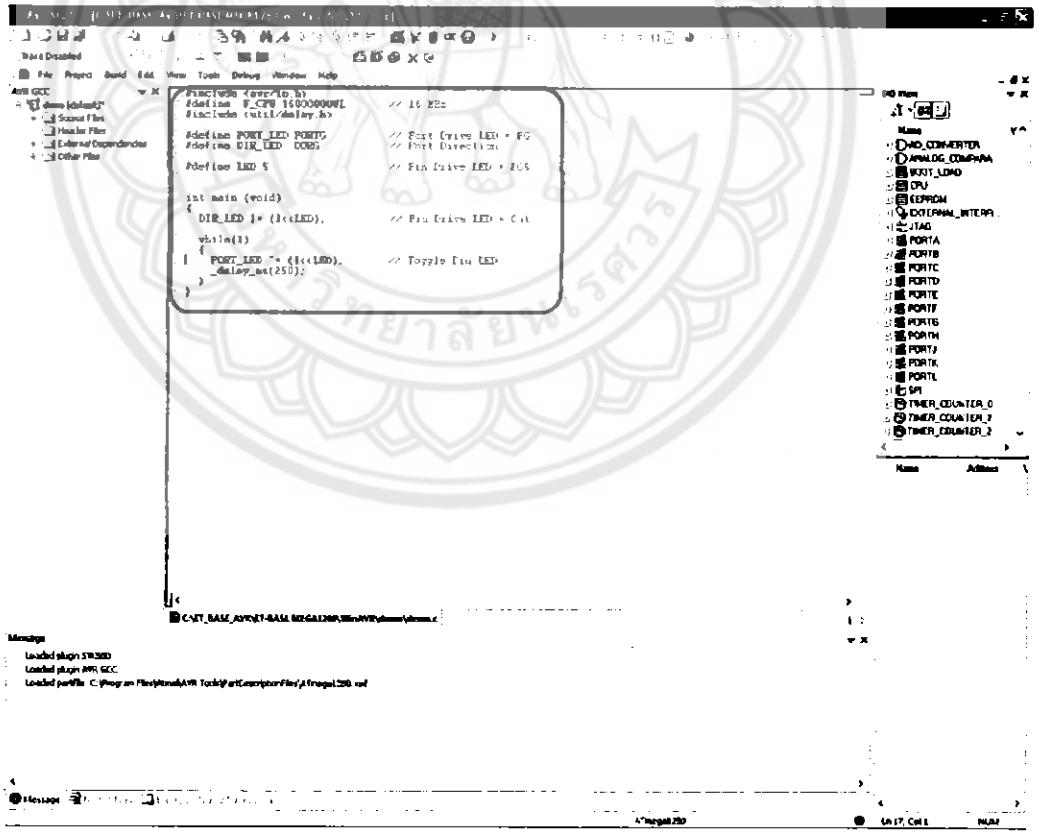
3. เมื่อกำหนดค่าตัวเลือกต่างๆ ให้กับโปรแกรมเรียบร้อยแล้ว ให้เลือกที่ Next แล้วกำหนดค่าใน Debug platform เป็น AVR Simulator และเลือก Device เป็น ATmega1280 ซึ่งเมื่อสร้าง project เสร็จโปรแกรมจะสร้างไฟล์ภาษาซีให้ โดยมีชื่อเดียวกับ project ไฟล์ ซึ่งในที่นี้จะเป็นไฟล์ชื่อ demo.c ให้เองโดยอัตโนมัติ เพียงแต่ไฟล์ดังกล่าวจะยังไม่มี code ใดๆบรรจุไว้ให้ เป็นเพียงหน้ากระดาษเปล่าๆ ซึ่งต้องรอให้เราเขียน code เพิ่มเข้าไปเอง ดังรูป



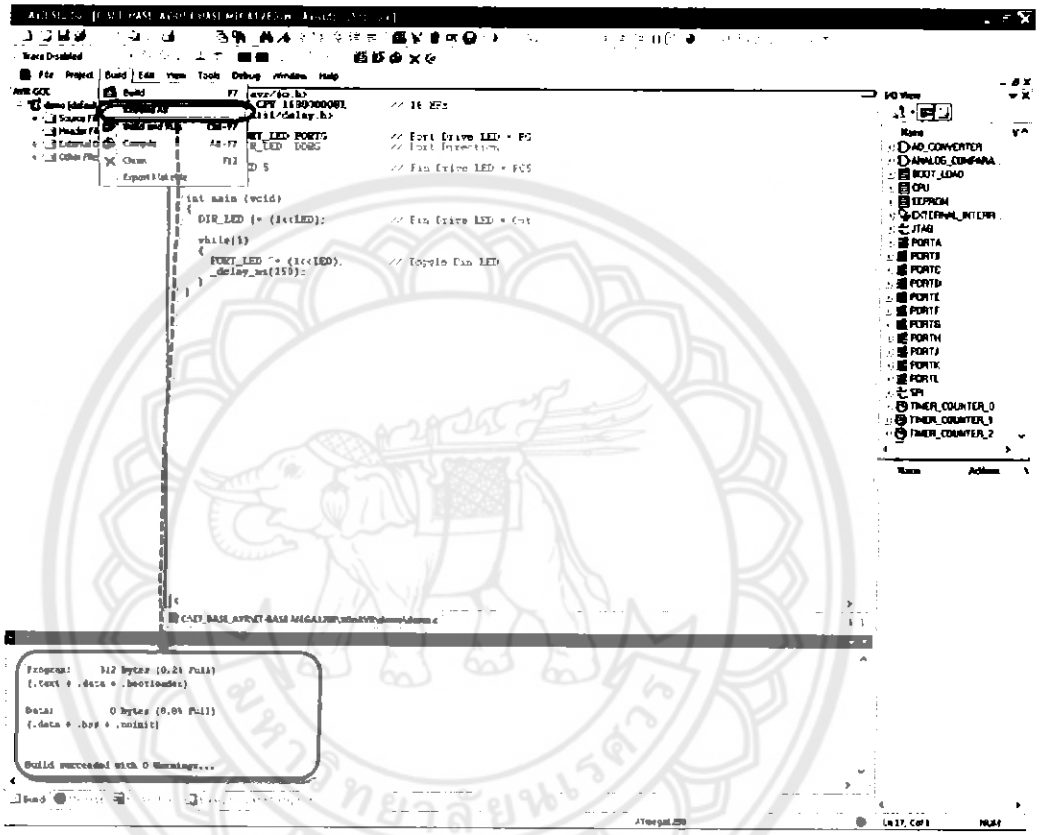
4. ให้พิมพ์คำสั่งของโปรแกรมสำหรับทดสอบการทำงาน ในหน้าต่าง Text Editor ของโปรแกรม โดยในที่นี้จะทดสอบด้วย Codeโปรแกรม สำหรับทำหน้าที่ทดสอบการทำงานของบอร์ดในเบื้องต้น โดยทำหน้าที่ ON/OFF หลอดแสดงผล LED ซึ่งต่อควบคุมจากขา PG5 ดังตัวอย่าง

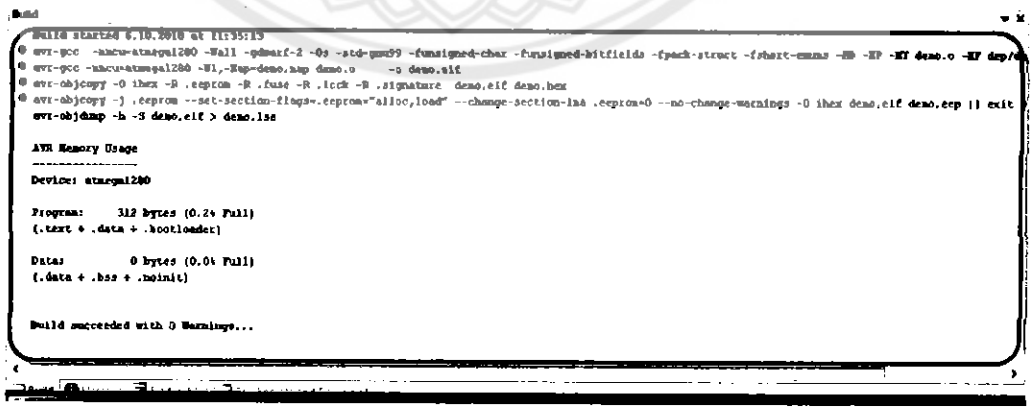
```
#include <avr/io.h>
#define  F_CPU 16000000UL                  // 16 MHz
#include <util/delay.h>

#define PORT_LED PORTG                     // Port Drive LED = PG
#define DIR_LED  DDRG                      // Port Direction
#define LED 5                              // Pin Drive LED = PG5

int main (void)
{
  DIR_LED |= (1<<LED);                     // Pin Drive LED = Out

  while(1)
  {
    PORT_LED ^= (1<<LED);                  // Toggle Pin LED
    _delay_ms(250);
  }
}
```

↓



5. หลังจากพิมพ์ Code โปรแกรมเสร็จแล้วให้สั่งแปลโปรแกรม โดยเลือกที่ build →rebuild all ซึ่งถ้าทุกอย่างถูกต้อง ผลการแปลคำสั่งจะได้ผลลัพธ์เป็น "Build succeeded with 0 Warnings... " และจะรายงานผลการแปลพร้อมขนาดหน่วยความจำที่ใช้ไป และจะได้ Output เป็น HEX File ที่มีชื่อเดียวกันกับ project ที่สร้างไว้ โดยจะบรรจุอยู่ในDirectory ย่อยชื่อ default ดังรูป



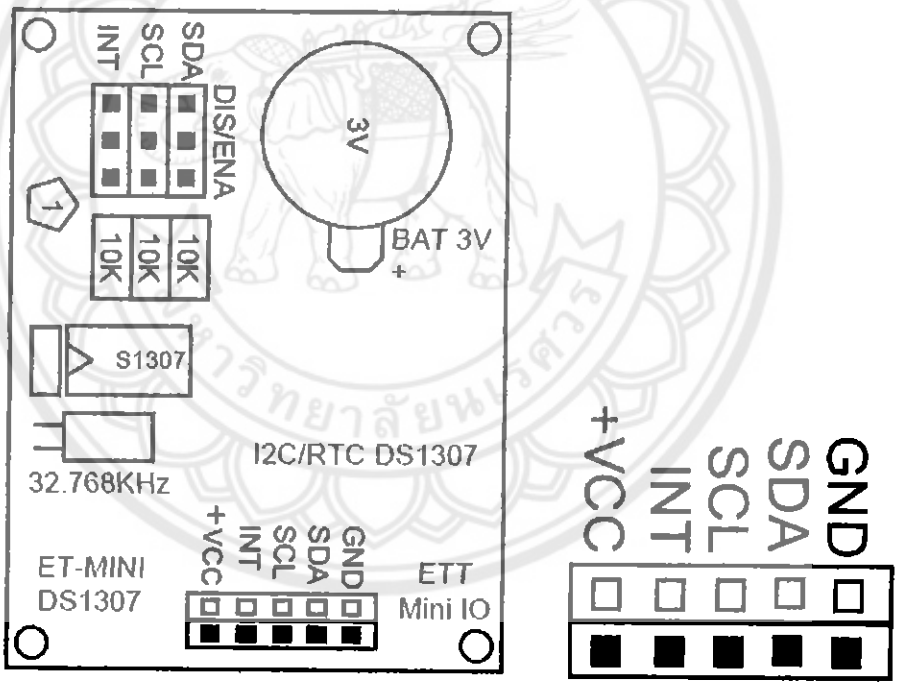


ภาคผนวก ข

บอร์ด ET-MINI DS1307

บอร์ด ET-MINI DS1307

ในการติดตั้งบอร์ด ET-MINI DS1307 จะนำไปเชื่อมต่อกับพอร์ตของบอร์ดไมโครคอนโทรเลอร์ ซึ่ง Module ET-MINI DS1307 เป็นชุด I^2C ประเภท Real Time Clock (RTC) ซึ่งทำหน้าที่เกี่ยวกับฐานเวลาในลักษณะของ นาฬิกา เวลา และปฏิทิน ในโมดูลนี้ จะมีขั้วต่อ SDA และ SCL ดังรูปที่ 3.7 สำหรับต่อเข้ากับขา SDA และ SCL ของ MCU ตามลำดับ ส่วนขั้วต่อ INT จะต่อเข้ากับ MCU ก็ต่อเมื่อต้องการใช้งาน Interrupt โดยให้ต่อเข้ากับขาของ MCU ที่ผู้ใช้ได้กำหนดให้มีการรับสัญญาณ INT. จากภายนอกไว้ ส่วนไฟเลี้ยงโมดูลจะอยู่ที่ VDC 3V-5V I^2C เบอร์นี้จะมี Control Byte อยู่ที่ "1101000x " นอกจากนี้ก็จะมีในส่วนของที่สำหรับใส่แบตเตอรี่เพื่อใช้ในการ Back Up ฐานเวลา ให้นาฬิกายังคงเดินได้อย่างถูกต้อง เมื่อไม่มีการจ่ายไฟเลี้ยงให้กับ Module



โครงสร้าง ET-MINI DS1307 และตำแหน่งขาพอร์ตใช้งาน