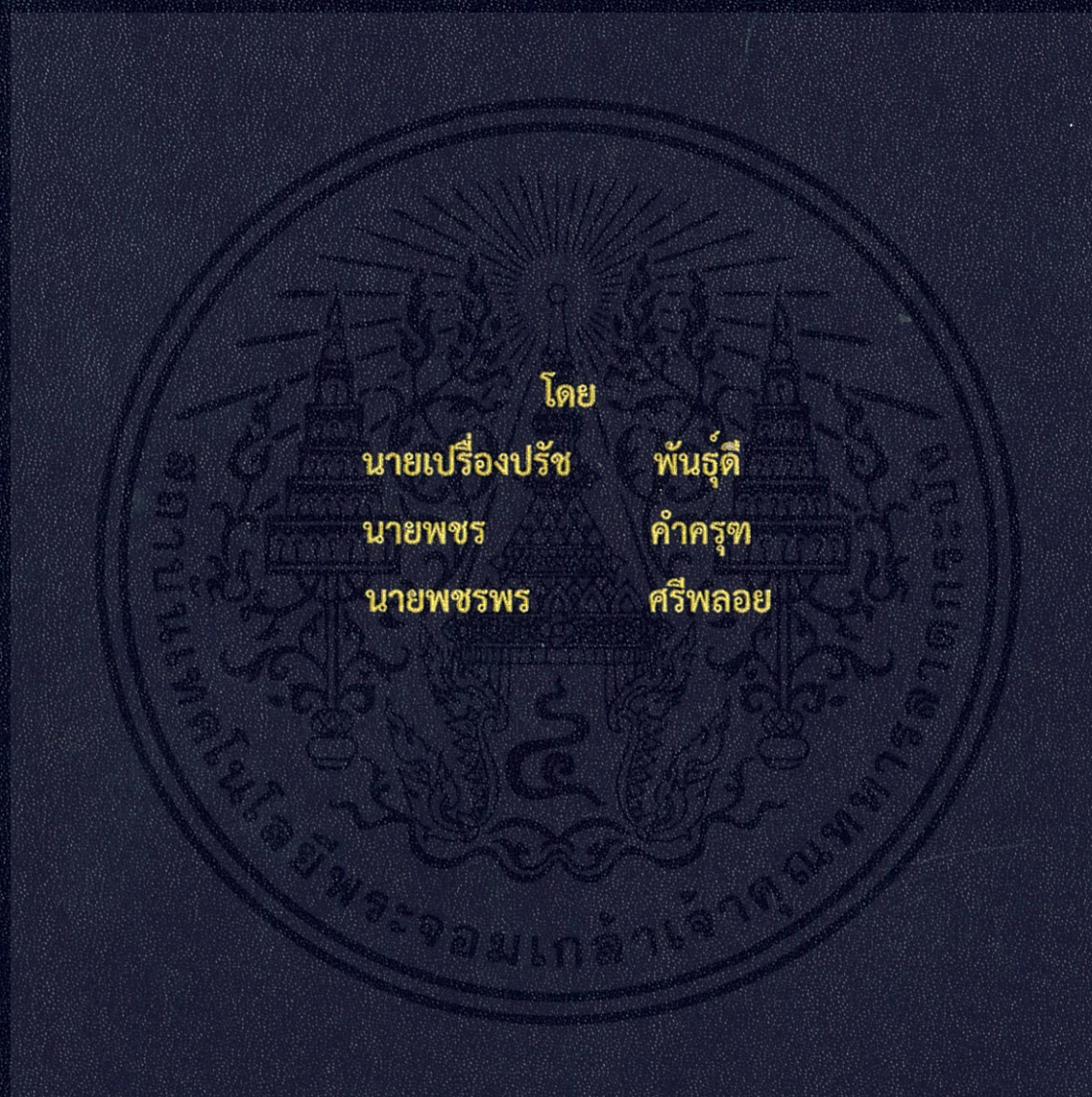


อุปกรณ์สวมใส่เพื่อแสดงสถานะการวิ่ง
WEARABLE DEVICE FOR RUNNING STATUS



โดย

นายเป็ร็องปรัช	พันธ์ุดี
นายพชร	คำครุฑ
นายพชรพร	ศรีพลอย

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

อุปกรณ์สวมใส่เพื่อแสดงสถานะการวิ่ง
WEARABLE DEVICE FOR RUNNING STATUS

โดย

นายเป็ร็องปรัช พันธุ์ดี	54010835
นายเพชร คำครุฑ	54010856
นายเพชรพร ศรีพลอย	54010859

อาจารย์ที่ปรึกษา

ผศ. ดร. ธเนศ พัฒนธาดาพงษ์

ผศ. ดร. นภัทร สระเอี่ยม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

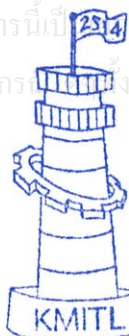
ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

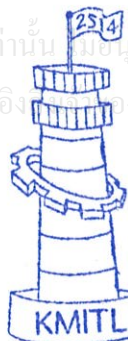
ปีการศึกษา 2557

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยสงขลานครินทร์ วิทยาเขตปัตตานี ซึ่งอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ จะต้องแจ้งให้ทางเอกสารก่อนทุกครั้ง การนำไปใช้



อาจารย์ที่ปรึกษา
14/5/57

วิศวกรรมโทรคมนาคม
Telecommunications Engineering



กรรมการผู้ตรวจชิ้นงาน
14/5/58

วิศวกรรมโทรคมนาคม
Telecommunications Engineering

ปริญญาโทปีการศึกษา 2557

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อุปกรณ์สวมใส่เพื่อแสดงสถานะการวิ่ง

WEARABLE DEVICE FOR RUNNING STATUS

ผู้จัดทำ

- | | |
|----------------------------|----------|
| 1. นายเป็ร็องปรัช พันธุ์ดี | 54010835 |
| 2. นายพชร คำครุฑ | 54010856 |
| 3. นายพชรพร ศรีพลอย | 54010859 |



(ผศ. ดร. ธเนศ พัฒนธาดาพงษ์)

อาจารย์ที่ปรึกษา



(ผศ. ดร. นภัทร สระเอี่ยม)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

กิตติกรรมประกาศ

โครงการฉบับนี้สำเร็จลงได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจาก ผู้ช่วยศาสตราจารย์ ดร. ธเนศ พัฒนธาดาพงษ์ และผู้ช่วยศาสตราจารย์ ดร. นภัทร สระเอี่ยม อาจารย์ที่ปรึกษาโครงการ ที่กรุณาให้คำแนะนำปรึกษาในการทำงาน ตลอดจนปรับปรุงแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่อย่างดียิ่ง คณะผู้จัดทำโครงการตระหนักถึงความตั้งใจจริงและความทุ่มเทของอาจารย์และขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ขอกราบขอบพระคุณบิดา มารดา ผู้เป็นกำลังใจให้แก่คณะผู้จัดทำเสมอมา ตลอดจนผู้มีพระคุณทุกท่านที่มีส่วนร่วมในการช่วยเหลือและขออน้อมบูชาท่านบูรพาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทความรู้ด้วยความรักและเมตตา

อนึ่ง คณะผู้จัดทำหวังว่า งานโครงการฉบับนี้คงจะมีประโยชน์อยู่ไม่น้อย จึงขอมอบส่วนดีทั้งหมดนี้ให้แก่เหล่าคุณอาจารย์ที่ได้ประสิทธิ์ประสาทวิชาจนทำให้โครงการนี้เป็นประโยชน์ต่อผู้ที่เกี่ยวข้อง และขอมอบความกตัญญูกตเวทิตาคุณ แต่บิดา มารดา และผู้มีพระคุณทุกท่าน

สุดท้ายนี้หากมีข้อบกพร่องประการใดที่อาจจะเกิดขึ้นนั้น คณะผู้จัดทำขออน้อมรับผิดมา ณ ที่นี้



เป็รื่องปร้ช พันธ์ดี
พชร คำครุฑ
พรชพร ศรีพลอย
คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์สวมใส่เพื่อแสดงสถานะการวิ่ง

WEARABLE DEVICE FOR RUNNING STATUS

โดย	นายเป็ร็องปร็ช พันธุ์ตี	54010835
	นายพชร คำครุฑ	54010856
	นายพชรพร ศรีพลอย	54010859

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. ธเนศ พัฒนธาดาพงษ์
 ผู้ช่วยศาสตราจารย์ ดร. นภัทร สระเอี่ยม

บทคัดย่อ

ในปัจจุบันการนำเอาเทคโนโลยีมาปรับใช้ในชีวิตประจำวันถือเป็นเรื่องสำคัญและเป็น การตอบสนองความต้องการความสะดวกในการใช้ชีวิต ให้มีความสะดวกรวดเร็วยิ่งขึ้น โครงการนี้ เป็นการพัฒนาระบบแสดงสถานะการวิ่ง โดยการวัดชีพจรขณะวิ่ง ความเร็ว และวัดระยะทางที่วิ่งได้ โดยใช้ Raspberry Pi เขียนโปรแกรมด้วยภาษา Python โดยใช้ Sensor ทำหน้าที่ตรวจชีพจรและ GPS Module จะทำหน้าที่ส่งข้อมูลพิกัดและความเร็วและส่งข้อมูลเพื่อไปประมวลผลโดย Raspberry Pi และจะสั่งให้แสดงผลข้อมูลชีพจรว่ามีอัตราการเต้นเป็นเท่าใด และวิ่งได้ระยะทางเท่าใด

ABSTRACT

Currently, Technology has become an integral part of human daily life for interacting our own life and making the life more simply and easy. This project studies is developing a running sensor of heart rate, speed, and calculate distance using Raspberry Pi with python language. The sensor will monitor heart rate and pass through the Raspberry Pi with its speed and position by GPS Module. The Raspberry Pi will process the data and display its status. นั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V
สารบัญตาราง	VI
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 ราสเบอร์รี่พาย (Raspberry Pi)	2
2.2 ภาษาไพธอน (Python)	20
2.3 ทฤษฎีพื้นฐานเกี่ยวกับเซนเซอร์	21
2.4 อัตราการเต้นของหัวใจ	21
2.5 ทฤษฎีพื้นฐานเกี่ยวกับจีพีเอส	24
บทที่ 3	
การออกแบบและการจัดทำปริญญานิพนธ์	26
3.1 การออกแบบ	26
3.2 เครื่องมือที่ใช้ในการทดลอง	31
3.3 การจัดเก็บผลการทดลอง	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง	36
4.1 การทำให้ Raspberry Pi สามารถติดต่อกับ GPS Module	36
4.2 การเชื่อมระหว่างระบบการวัดอัตราการเต้นของหัวใจเข้าไปในระบบการทำงาน	38
4.3 การทดลองการแสดงผลผ่านจอแสดงผล	39
4.4 เก็บค่าเพื่อแสดงผลผ่านทางเว็บแสดงผล	40
4.5 แสดงวงจรที่ทำการออกแบบแล้วนำไปเตรียมต่ออุปกรณ์ต่างๆ	42
บทที่ 5 สรุปผลและข้อเสนอแนะ	45
5.1 สรุปผล	45
5.2 ข้อเสนอแนะ	45
บรรณานุกรม	46
ภาคผนวก ก โปรแกรมการทำงาน	47
ภาคผนวก ข คู่มือการใช้งาน	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบของ Raspberry Pi B+	3
2.2 ไฟล์ Image ของ Raspbian Wheezy	4
2.3 โปรแกรม Win32 Disk Imager	4
2.4 เลือกไฟล์ Image	5
2.5 การทำงานของโปรแกรม Win32 disk imager (1)	5
2.6 การทำงานของโปรแกรม Win32 disk imager (2)	6
2.7 ใส่ Micro SD Card ลง Raspberry Pi	6
2.8 การเชื่อมต่ออุปกรณ์ต่างๆ กับ Raspberry Pi	6
2.9 การทำงานของ Raspberry Pi	7
2.10 หน้าจอหลังพิมพ์คำสั่ง startx	8
2.11 การต่อสาย LAN กับ Raspberry Pi	8
2.12 คำสั่ง Update ของ Raspberry Pi	9
2.13 การ Upgrade โปรแกรมของ Raspberry Pi	9
2.14 คำสั่งติดตั้ง Chromium Browser	10
2.15 การเปิดโปรแกรม Chromium Browser	10
2.16 ขา GPIO ของ Raspberry Pi	11
2.17 Raspberry Pi B+	11
2.18 บอร์ด Raspberry Pi B+	12
2.19 จำนวนขา GPIO Raspberry Pi B+	13
2.20 ไดอะแกรมพอร์ต GPIO ของ Raspberry Pi B+	13
2.21 จำนวนช่องเสียบ USB	14
2.22 ช่องเสียบ Micro SD Card	14
2.23 ภาควัดจ่ายไฟใหม่	15
2.24 วงจรของ Model B+	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.25 คอนเน็คเตอร์ Audio Out Model B	17
2.26 คอนเน็คเตอร์ Audio Out Model B+	17
2.27 สายแปลง 4 ขั้วเป็น RCA 3 เส้น	18
2.28 Top View บอร์ด Raspberry Pi Model B+	18
2.29 LED แสดงผลบนบอร์ด	19
2.30 สถานะไฟเลี้ยงเข้าบอร์ด และ Activity	19
2.31 เครื่องมือวัดอัตราการเต้นของหัวใจ	21
3.1 Block diagram	26
3.2 แผนผังการทำงานรวมของระบบ	27
3.3 แผนผังการทำงานของการเก็บค่าระยะทางและความเร็ว	28
3.4 แผนผังการทำงานของการวัดอัตราการเต้นของหัวใจ	29
3.5 การออกแบบ Schematic วงจรเพื่อใช้ในการทำงาน	30
3.6 ผลการออกแบบวงจรเพื่อใช้ในการทำงาน	30
3.7 การจัดเรียงขา GPIO	31
3.8 Keyboard & Mouse	32
3.9 สาย LAN	32
3.10 GPS Module	33
3.11 จอแสดงผล	34
4.1 การประกอบอุปกรณ์ต่างๆเข้าไว้ด้วยกัน	36
4.2 ค่าที่รับได้จาก GPS Module	37
4.3 ทดลองรันโปรแกรม แสดงค่าที่รับจาก GPS Module	37
4.4 การทดลองต่ออุปกรณ์ต่างๆเพื่อวัดอัตราการเต้นของหัวใจ	38
4.5 การทดสอบการวัดอัตราการเต้นของหัวใจ	38
4.6 การประมวลผลการเต้นของหัวใจ	39
4.7 การต่อจอแสดงผล	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า	
4.8	จอแสดงผลการทำงาน	40
4.9	ผลการเก็บข้อมูลลงใน MySQL	40
4.10	ผลข้อมูลในเว็บเพื่อแสดงรายละเอียดการวิ่ง	41
4.11	การเข้าหน้าเว็บเพื่อเข้าสู่ข้อมูลการใช้งาน	41
4.12	การตั้งค่าชื่อ อายุ น้ำหนัก ส่วนสูง และอัตราการเต้นหัวใจของผู้ใช้งาน	42
4.13	วงจรที่พร้อมทำการเชื่อมต่อเข้ากับระบบต่างๆ	42
4.14	การต่อวงจรที่ออกแบบเข้ากับบอร์ด Raspberry Pi	43
4.15	อุปกรณ์ที่ประกอบเสร็จสมบูรณ์	43
4.16	การติดอุปกรณ์เข้ากับตัวผู้ใช้งาน	44
4.17	อุปกรณ์ที่พร้อมใช้งาน	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีมีส่วนเข้ามาช่วยให้ชีวิตประจำวันของผู้คนดีขึ้นเป็นอย่างมาก ทำให้สามารถออกกำลังกายได้อย่างตรงจุดมากขึ้น ซึ่งอุปกรณ์ช่วยการออกกำลังกายหลายชนิดตอนนี้สามารถแสดงข้อมูลต่างๆให้กับผู้ใช้งานมากขึ้น ไม่ว่าจะเป็นอัตราการเต้นของหัวใจ ความเร็วระยะทาง ตลอดจนการเผาผลาญพลังงาน อุปกรณ์ช่วยการออกกำลังกายเหล่านี้ในปัจจุบันมีหลากหลายชนิด ทำให้ผู้ใช้มีตัวเลือกที่เยอะขึ้นมากจากในอดีต โครงการนี้จึงได้ทำการศึกษาและพัฒนาอุปกรณ์สำหรับนักวิ่งขึ้นมาเพื่อเป็นอีกทางเลือกสำหรับนักวิ่ง

1.2 วัตถุประสงค์

- 1) เพื่อศึกษาการเขียนโปรแกรมโดยใช้ภาษา Python
- 2) เพื่อศึกษาการทำงานของ Raspberry Pi และระบบปฏิบัติการ Raspbian
- 3) เพื่อสร้างระบบการแสดงผลขณะวิ่ง

1.3 ขอบเขตของปริญญานิพนธ์

โครงการนี้เป็นการพัฒนาระบบวัดชีพจรขณะวิ่ง ระยะทางที่วิ่งได้ ความเร็วและสามารถคำนวณแคลอรีที่ใช้ โดยใช้ Raspberry Pi ซึ่งจะเป็นการเขียนโปรแกรมโดยใช้ภาษา Python โดย Sensor จะทำหน้าที่ตรวจชีพจร และส่งข้อมูลเพื่อไปประมวลผลโดย Raspberry Pi และจะสั่งให้แสดงผลข้อมูลผ่านจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

โครงการเรื่อง "อุปกรณ์สวมใส่เพื่อแสดงสถานะการวิ่ง" การพัฒนาระบบวัดชีพจรขณะวิ่ง วัดระยะทางที่วิ่งได้และสามารถคำนวณแคลอรีที่ใช้ เพื่อใช้ตรวจสอบชีพจร โดยใช้ Raspberry Pi ซึ่งจะเป็นการเขียนโปรแกรมโดยใช้ภาษา Python โดย Sensor จะทำหน้าที่ตรวจชีพจร และส่งข้อมูลเพื่อไปประมวลผลโดย Raspberry Pi และจะสั่งให้แสดงผลข้อมูลชีพจรว่ามีอัตราการเต้นเป็นเท่าใด ทฤษฎีและหลักการทั้งหมดที่เกี่ยวข้องมีดังต่อไปนี้

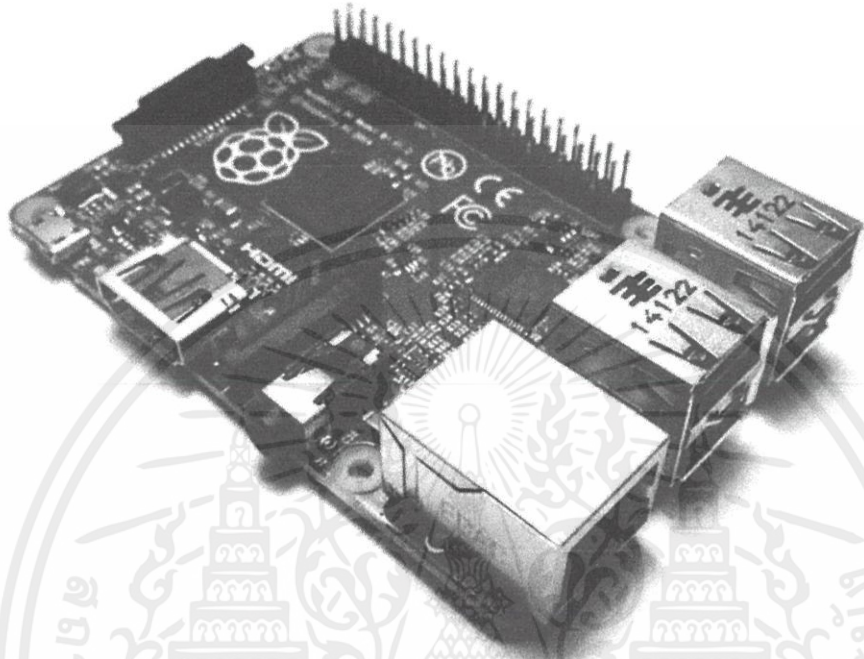
2.1 ราชเบอร์รี่พาย (Raspberry Pi)

บอร์ดคอมพิวเตอร์ขนาดเล็กที่สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ สามารถนำมาประยุกต์ใช้ในการทำโครงการทางด้านอิเล็กทรอนิกส์ การเขียนโปรแกรม หรือเป็นเครื่องคอมพิวเตอร์ตั้งโต๊ะขนาดเล็ก ไม่ว่าจะเป็นการทำงาน Spreadsheet Word Processing ท่องอินเทอร์เน็ต ส่งอีเมลล์ หรือเล่นเกม อีกทั้งยังสามารถเล่นไฟล์วิดีโอความละเอียดสูง (High-Definition) ได้อีกด้วย

บอร์ด Raspberry Pi รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian (Debian) Pidora (Fedora) และ Arch Linux เป็นต้น โดยติดตั้งบน Micro SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆได้อีกด้วย โดยในการทำโครงการชิ้นนี้ได้ใช้บอร์ด Raspberry Pi Model B+

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 ส่วนประกอบของบอร์ด Raspberry Pi Model B+



รูปที่ 2.1 ส่วนประกอบของ Raspberry Pi B+

2.1.2 การเริ่มต้นใช้งานบอร์ด Raspberry Pi

การเริ่มต้นใช้งานบอร์ด Raspberry Pi ต้องเตรียมอุปกรณ์ดังนี้

1. บอร์ด Raspberry Pi
2. Micro SD Card สำหรับเก็บระบบปฏิบัติการ และข้อมูล ควรมีความจุตั้งแต่ 4GB แบบคลาส 4 ขึ้นไป
3. สาย Micro USB ใช้สำหรับต่อกับแหล่งจ่ายไฟ +5V 700mA ที่มีจุดต่อแบบ USB ไม่ควรใช้แหล่งจ่ายไฟจากพอร์ต USB ของคอมพิวเตอร์ เนื่องจากมีความสามารถในการจ่ายกระแสไฟฟ้าไม่เพียงพอ
4. สาย HDMI หรือ RCA (อย่างใดอย่างหนึ่ง) สำหรับต่อกับจอแสดงผล
5. คีย์บอร์ดและเมาส์แบบ USB สำหรับควบคุมการทำงานของบอร์ด Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ตรงตามเงื่อนไข ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ลำโพงที่มีสายต่อเป็นปลั๊กหูฟัง 3.5 มม. เพื่อต่อกับแจ็กเอาต์พุด สัญญาณเสียงของบอร์ด Raspberry Pi

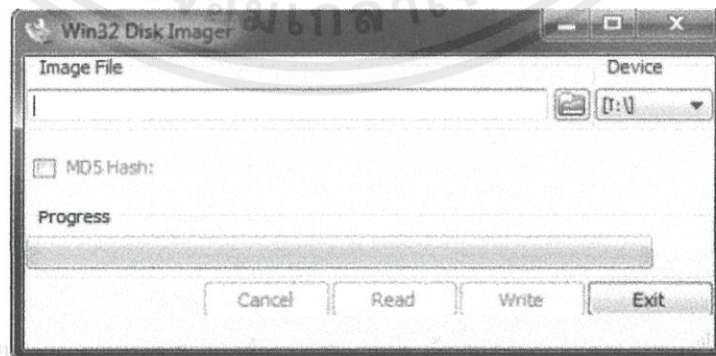
2.1.3 ติดตั้งข้อมูลให้กับ Micro SD Card

- ดาวน์โหลด Win32 Disk Imager จาก <http://www.softpedia.com/get/CD-DVD-Tools/Data-CD-DVD-Burning/Win32-Disk-Imager.html>
- จากนั้นดาวน์โหลดไฟล์ Raspbian “wheezy” อันเป็นไฟล์ระบบปฏิบัติการ จาก <http://www.raspberrypi.org/downloads>
- แดกไฟล์ของ Raspbian “wheezy” ออกมาจะได้ไฟล์ที่มีนามสกุลเป็น .img



รูปที่ 2.2 ไฟล์ Image ของ Raspbian Wheezy

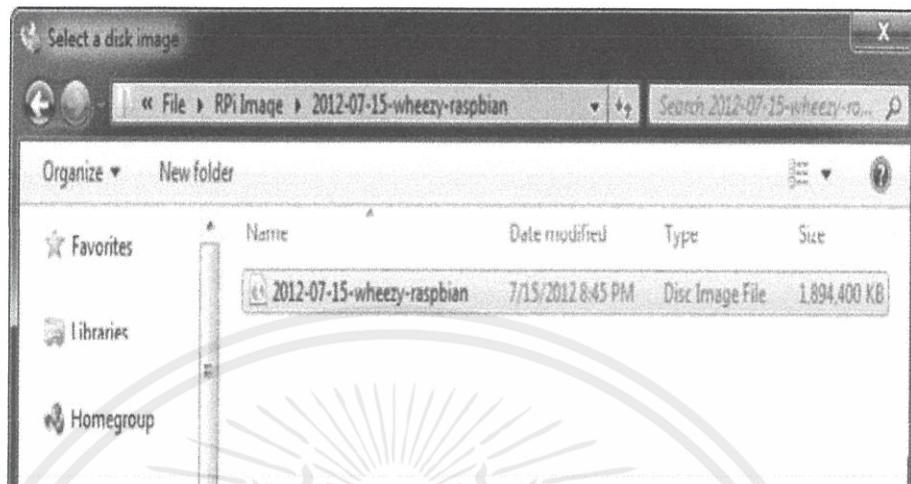
- ต่อ Micro SD Card เข้ากับเครื่องคอมพิวเตอร์ (ผ่านตัวอ่านการ์ด) แล้วเปิดโปรแกรม Win32 Disk Imager ขึ้นมา โปรแกรมจะค้นหา Drive ของ Micro SD Card ให้อัตโนมัติ โดยในตัวอย่างจะเป็น Drive I คลิกปุ่มสัญลักษณ์ รูปแฟ้มสีน้ำเงิน



รูปที่ 2.3 โปรแกรม Win32 Disk Imager

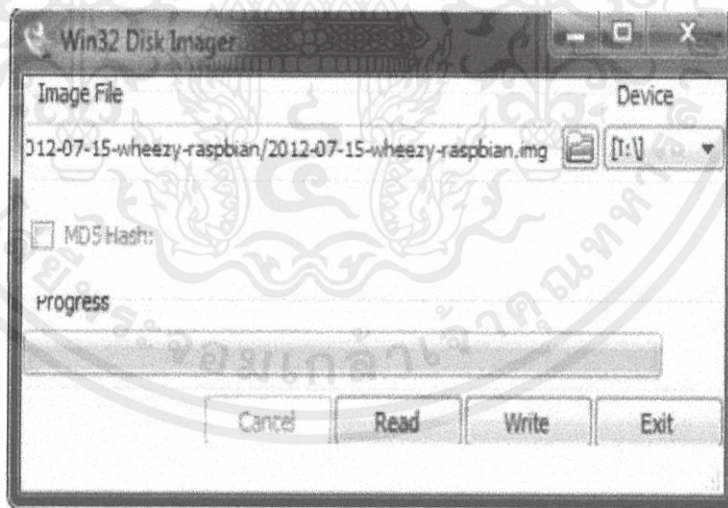
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เฉพาะในวงจำกัดเท่านั้น ขอสงวนสิทธิ์ในเงื่อนไขการใช้งาน
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แล้วเลือกไฟล์ Raspbian “wheezy” ที่ดาวน์โหลดมา



รูปที่ 2.4 เลือกไฟล์ Image

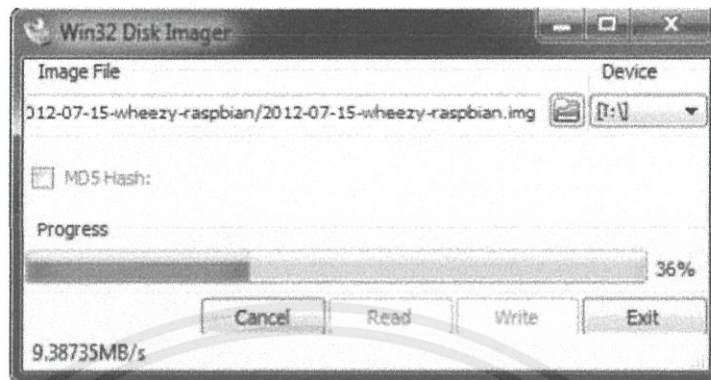
- คลิกที่ปุ่ม Write เพื่อเขียนข้อมูลลงใน Micro SD การ์ด โดยโปรแกรมจะมีหน้าต่างแจ้งเตือนให้คลิก Yes เพื่อเริ่มดำเนินการ



รูปที่ 2.5 การทำงานของโปรแกรม Win32 Disk Imager (1)

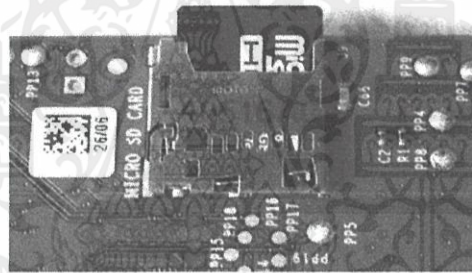
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รอจนกว่าการเขียนข้อมูลลงบน Micro SD Card เสร็จ



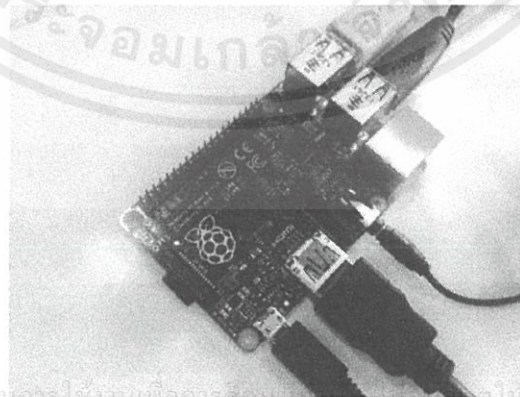
รูปที่ 2.6 การทำงานของโปรแกรม Win32 Disk Imager (2)

- ถอด Micro SD การ์ด จากตัวอ่านการ์ดเพื่อนำมาเสียบเข้ากับบอร์ด Raspberry Pi



รูปที่ 2.7 ใส่ Micro SD Card ลง Raspberry Pi

- ต่อสายเคเบิลบอร์ด, เม้าส์ , จอภาพ และแหล่งจ่ายไฟให้ เรียบร้อย จากนั้นจ่ายไฟให้ กับบอร์ด Raspberry Pi



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ขอเชิญชวนให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.8 การเชื่อมต่ออุปกรณ์ต่างๆ กับ Raspberry Pi
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากนั้นบอร์ด Raspberry Pi จะเริ่มทำงาน รอจนกว่าจะขึ้นหน้าล็อกอินใช้ Username และ Password เริ่มต้นของระบบในการล็อกอิน
Username: pi
Password: raspberry
- หลังจากล็อกอินเรียบร้อยแล้วใช้งานบอร์ด Raspberry Pi ได้ทันที

```

Could not load host key: /etc/ssh/ssh_host_ecdsa_key
. ok
My IP address is 192.168.1.13
Debian GNU/Linux wheezy/sid raspberrypi tty1
raspberrypi login: pi
Password:
Last login: Thu Aug 9 04:26:09 UTC 2012 on tty1
Linux raspberrypi 3.1.9* #168 PREEMPT Sat Jul 14 18:56:31 EST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

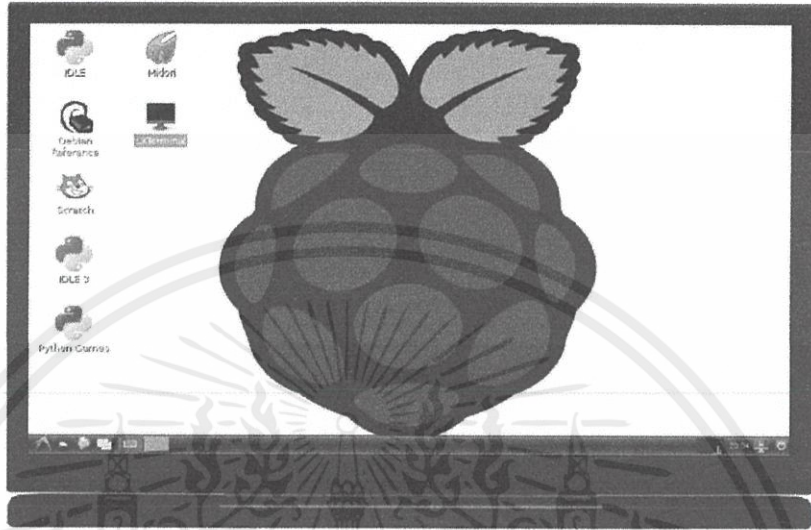
Type 'startx' to launch a graphical session
pi@raspberrypi ~$

```

รูปที่ 2.9 การทำงานของ Raspberry Pi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

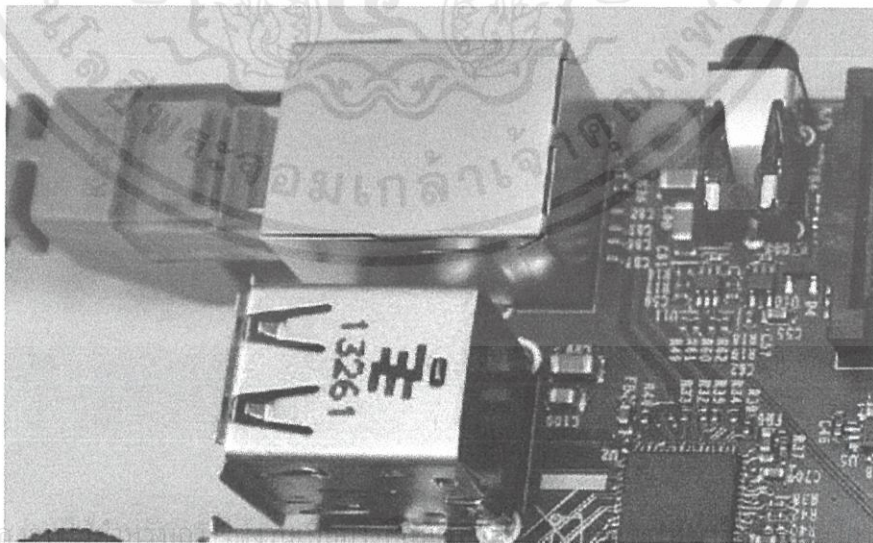
- กรณีที่ต้องการใช้งานแบบกราฟิกให้พิมพ์คำสั่ง startx จากนั้นระบบจะเข้าสู่หน้าต่างที่มีรูปพื้นหลังเป็นราสเบอร์รี่เพื่อเริ่มการใช้งานในโหมดกราฟิก



รูปที่ 2.10 หน้าจอหลังพิมพ์คำสั่ง startx

2.1.4 การใช้งานกับระบบเครือข่ายและการเชื่อมต่ออินเทอร์เน็ต

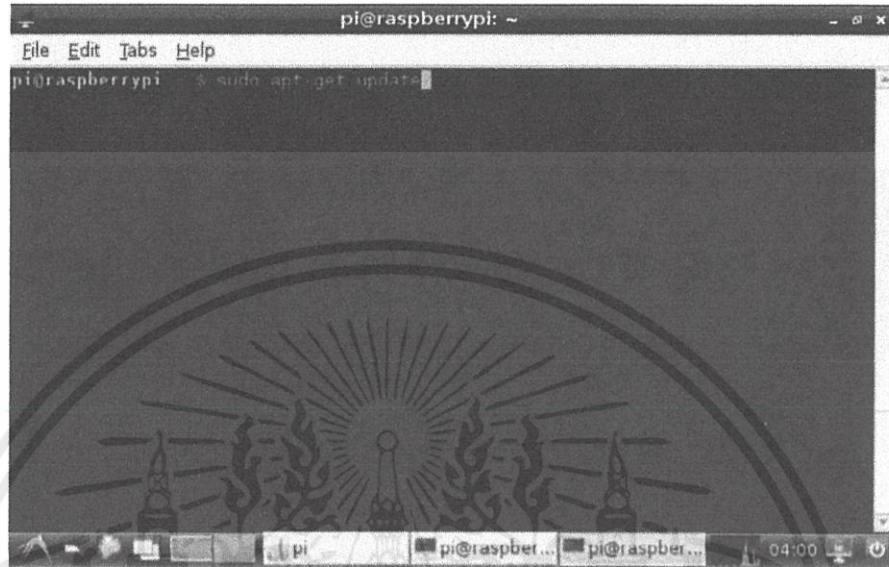
- เมื่อต้องการเชื่อมต่ออินเทอร์เน็ตให้ต่อสาย LAN เข้าที่จุดต่อ LAN ซึ่งเป็นคอนเนคเตอร์ RJ-45 กับเครือข่าย LAN



เอกสารนี้เป็นเอกสารที่... นี้ด้านการค้า

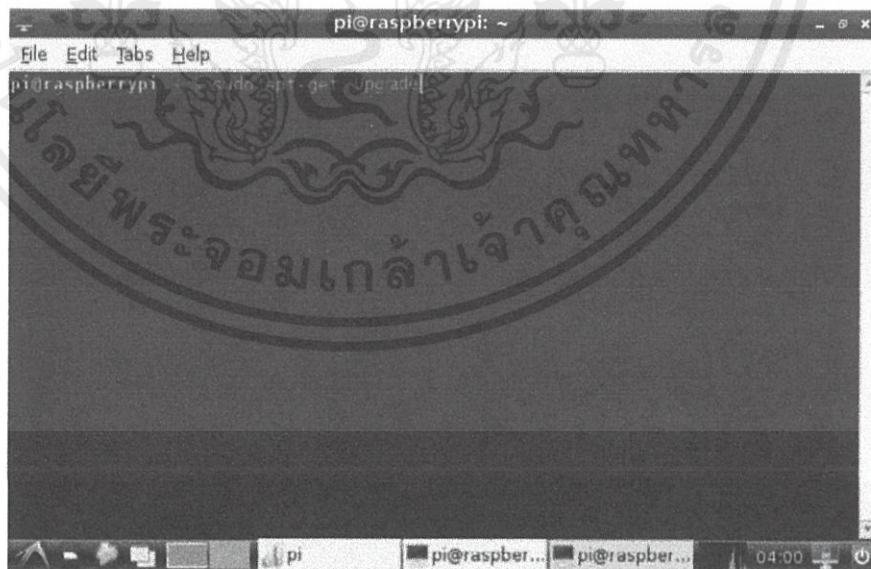
รูปที่ 2.11 การต่อสาย LAN กับ Raspberry Pi
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อเชื่อมต่ออินเทอร์เน็ตแล้ว ให้เปิดโปรแกรม LX Terminal ขึ้นมา แล้วใช้คำสั่ง `sudo apt-get update` เพื่อทำการตรวจสอบโปรแกรมเวอร์ชันล่าสุด



รูปที่ 2.12 คำสั่ง Update ของ Raspberry Pi

- ใช้คำสั่ง `sudo apt-get upgrade` กดปุ่ม Y ตามด้วยกดปุ่ม Enter เพื่อยืนยันแล้ว รอดาวนโหลดไฟล์ จากนั้นระบบจะถามว่าต้องการทำอะไรกับแพ็คเกจกดปุ่ม Y แล้ว Enter เพื่อติดตั้งทับไฟล์เดิมแล้วรอจนกระทั่งการอัปเดตเสร็จสิ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.13 การ Upgrade โปรแกรมของ Raspberry Pi
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 การติดตั้งโปรแกรมอื่นๆ ลงใน Micro SD Card เพื่อใช้งานกับบอร์ด

Raspberry Pi

- ในกรณีที่ต้องการติดตั้งโปรแกรมอื่นๆเพิ่มเติมให้ใช้คำสั่ง `sudo apt-get install` (ชื่อโปรแกรม) ผู้ใช้งานจะต้องตรวจสอบโปรแกรมหรือซอฟต์แวร์ที่นำมาติดตั้งก่อนว่าโปรแกรมนั้นๆ สามารถทำงานบนระบบปฏิบัติการที่ติดตั้งให้กับบอร์ด Raspberry Pi ได้หรือไม่
- ตัวอย่างการติดตั้งโปรแกรม Chromium Browser ให้กับบอร์ด Raspberry Pi ใช้คำสั่ง `sudo apt-get install chromium-browser` จากนั้นกดปุ่ม Y ตามด้วย Enter เพื่อยืนยันการติดตั้ง



รูปที่ 2.14 คำสั่งติดตั้ง Chromium Browser

- ติดตั้งเสร็จแล้วเมื่อต้องการเรียกใช้งานให้ไปที่ Start > Internet > Chromium Web Browser



รูปที่ 2.15 การเปิดโปรแกรม Chromium Browser

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 การจัดขาของ GPIO

บอร์ด Raspberry Pi Model B+ มีจำนวน GPIO pins 40 pins ซึ่งมีการจัดเรียงขาตามรูป

Raspberry Pi B+ J8 Header			
Pin#	NAME		NAME Pin#
01	3.3v DC Power		DC Power 5v 02
03	GPIO:2 (SDA1 , I2C)		DC Power 5v 04
05	GPIO:3 (SCL1 , I2C)		Ground 06
07	GPIO:4 (GPIO_GCLK)		(TXD0) GPIO14 08
09	Ground		(RXD0) GPIO15 10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18 12
13	GPIO27 (GPIO_GEN2)		Ground 14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23 16
17	3.3v DC Power		(GPIO_GEN5) GPIO24 18
19	GPIO10 (SPI_MOSI)		Ground 20
21	GPIO:9 (SPI_MISO)		(GPIO_GEN6) GPIO25 22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO:8 24
25	Ground		(SPI_CE1_N) GPIO:7 26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC 28
29	GPIO:5		Ground 30
31	GPIO:6		GPIO12 32
33	GPIO13		Ground 34
35	GPIO19		GPIO16 36
37	GPIO26		GPIO20 38
39	Ground		GPIO21 40

Rev. 1.1
18/07/2014 <http://www.element14.com>

รูปที่ 2.16 ขา GPIO ของ Raspberry Pi

2.1.7 รายละเอียดเพิ่มเติมเกี่ยวกับ Raspberry Pi Model B+

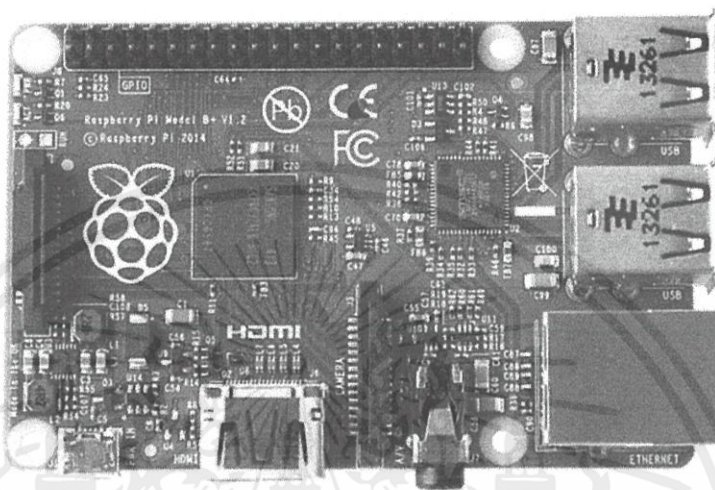
ข้อมูลรายละเอียดเกี่ยวกับบอร์ด Raspberry Pi Model B+ ให้เห็นความแตกต่างเทียบกับ Model B รุ่นก่อนหน้า



ภาพที่ 2.17 Raspberry Pi Model B+

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูงาน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวบอร์ด Raspberry Pi ได้มีการอัปเดตจาก Raspberry Pi Model B เป็นบอร์ด Raspberry Pi Model B+ ดังนั้นบอร์ด Raspberry Pi ในปัจจุบันทั้งหมดที่มีได้แก่ Model A Model B และ Model B+



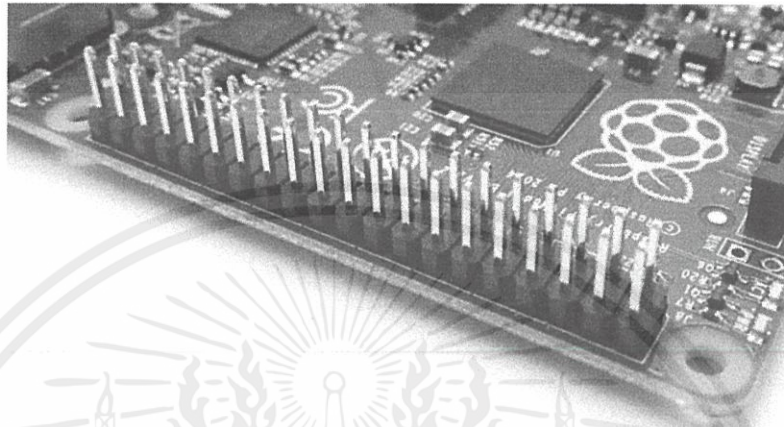
ภาพที่ 2.18 บอร์ด Raspberry Pi Model B+

Raspberry Pi Model B+ ใช้ชิพ BCM2835 ของ Broadcom ในตระกูล Application Processor ของ ARM บนสถาปัตยกรรม ARM11 มีความเร็วในการทำงานที่ 700MHz ใช้ชิพช่วยประมวลผลกราฟิกของ VideoCore IV มีหน่วยความจำหลัก (Primary Memory: RAM) ขนาด 512MB (บอร์ดลง RAM ครอบ CPU ไว้) และไม่มีหน่วยความจำสำรอง (Secondary Storage Memory) ขนาดความกว้างและความยาวบอร์ดเท่าเดิม ใช้แหล่งจ่ายไฟผ่านคอนเน็คเตอร์ microUSB สามารถรันระบบปฏิบัติการ Raspbian หรือระบบอื่นๆ ขาสัญญาณบนพอร์ต GPIO ในส่วน 26 ขาแรกจัดเรียงเหมือนเดิม เมื่อเทียบกับ Model B ขาใช้พอร์ต HDMI แบบ Full-Size พอร์ตสำหรับส่วนของ Audio Out ผ่าน Audio Jack สามารถทำหน้าที่เป็นคอนเน็คเตอร์สำหรับต่อกล้อง Raspberry Pi Camera และ DSI Display

Model B + ได้รับการพัฒนาและปรับปรุงเปลี่ยนแปลงจาก Model B ในหลายๆ ส่วนให้ดีขึ้นดังนี้

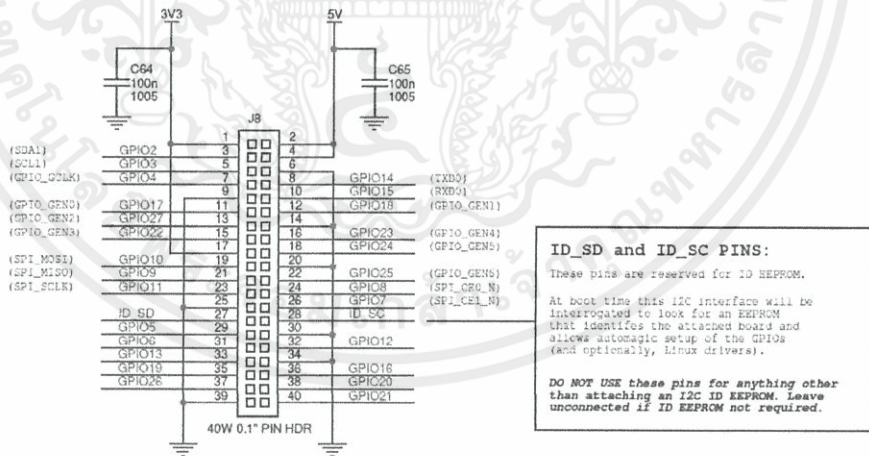
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพิ่มจำนวนขาสัญญาณของ GPIO มากขึ้น จากเดิม 26 ขา เป็น 40 ขา โดยยังคงออกแบบให้ขาสัญญาณ 26 ขาแรกจัดเรียงไว้อยู่ในรูปแบบเดิมเพื่อให้ยังพอที่จะสามารถต่อใช้งานกับอุปกรณ์เชื่อมต่อเดิมได้หรือหากต้องแก้ไขก็ทำเพียงเล็กน้อย



ภาพที่ 2.19 จำนวนขา GPIO Raspberry Pi B+

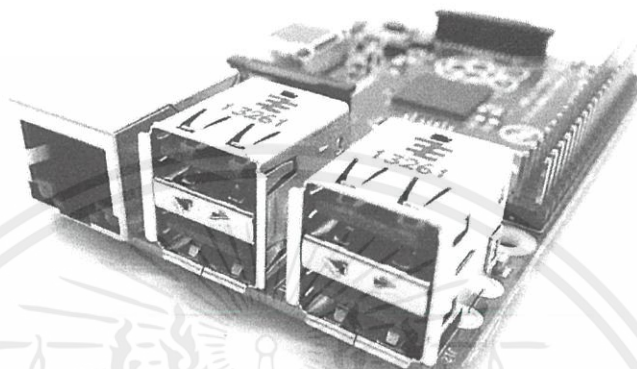
ขา GPIO ที่เพิ่มขึ้นคือขา ID_SD และ ID_SC สำหรับเชื่อมต่อกับ EEPROM แบบ I2C ใช้เก็บค่าคอนฟิกต่างๆ ที่ต้องการเพื่อกำหนดค่า GPIO อัตโนมัติในตอนเริ่มระบบนอกจากนี้เป็นขา GND เพิ่มขึ้นมา 3 ขาและ GPIO เพิ่มขึ้นมา 9 ขา



ภาพที่ 2.20 ไดอะแกรมพอร์ต GPIO ของ Raspberry Pi B+

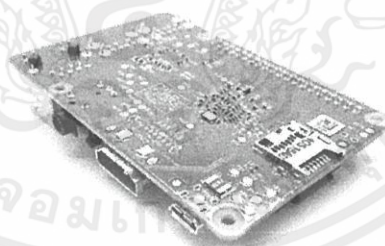
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เพิ่มจำนวนช่องเสียบ USB จากเดิมที่มี 2 ช่องใน Model B ได้เพิ่มเป็น 4 ช่องใน Model B+ โดยเปลี่ยนชิพที่ใช้เป็นเบอร์ที่มีช่องสัญญาณมากขึ้นจาก LAN9512 เป็น LAN9514 แต่ยังคงเป็นมาตรฐาน USB 2.0 เช่นเดิม



ภาพที่ 2.21 จำนวนช่องเสียบ USB

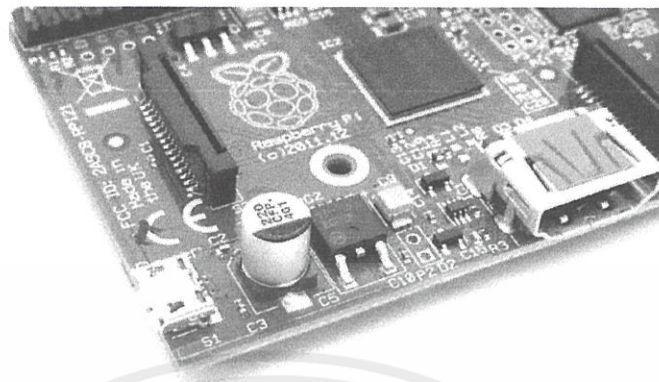
- เปลี่ยนมาใช้ช่องเสียบ Micro SD Card แทน Standard Full-Size SD Card เพื่อลดพื้นที่แผ่นวงจร เพราะช่องเสียบของรุ่นเก่ามีขนาดใหญ่และการ์ดที่ยื่นออกมาจากตัวบอร์ดค่อนข้างมากทำให้อาจเกิดการค้ำจนตัวการ์ดหักได้ นอกจากนี้การ์ดที่ขายในท้องตลาดปัจจุบันส่วนใหญ่เป็น Micro SD Card เมื่อจะใช้กับ Model B ก็ต้องใช้ Adapter ซึ่งเมื่อใช้งานบ่อยครั้ง อาจจะทำให้เกิดการหลวม หลุดหรือหน้าสัมผัสไม่สนิทได้ง่าย และช่องเสียบรุ่นใหม่เปลี่ยนไปใช้แบบ Push-Push ที่มีสปริงภายในทำให้เวลาถอดหรือเสียบหน้าสัมผัสเข้าตำแหน่งได้ดีขึ้น



ภาพที่ 2.22 ช่องเสียบ Micro SD Card

- ปรับปรุงภาคจ่ายไฟใหม่ ส่งผลต่อเสถียรภาพโดยรวมของบอร์ด ทำให้ยืดอายุการใช้งานของอุปกรณ์ต่างๆ เนื่องจากการใช้ไฟที่เสถียรขึ้น และยังช่วยลดโอกาสที่จะเกิดระบบจะล่มได้จากการดึงไฟเวลาถอด-เสียบ USB

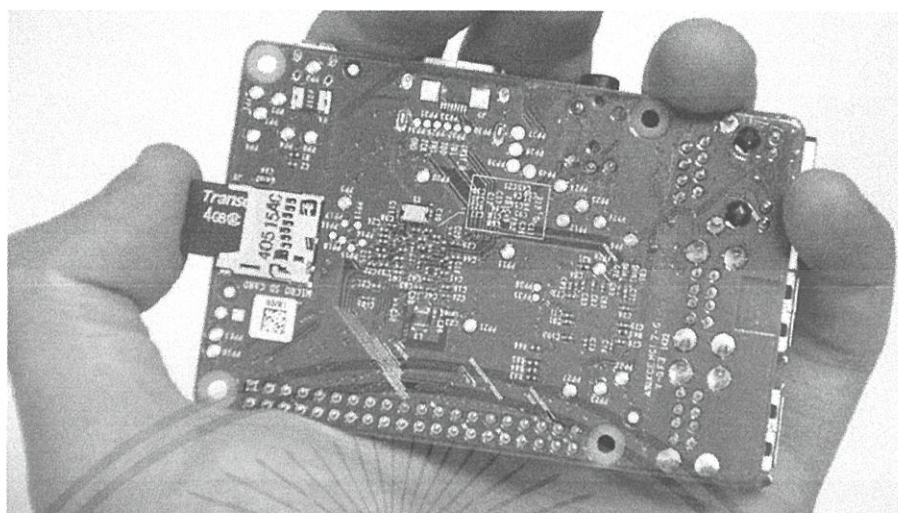
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.23 ภาคจ่ายไฟใหม่

บอร์ด Model A และ Model B รับไฟ 5 โวลต์จาก USB แล้วใช้วงจรแหล่งจ่ายบนบอร์ดไฟสร้างแรงดัน 3.3, 2.5 และ 1.8 โวลต์เพื่อใช้งาน ภาคจ่ายไฟของรุ่นเก่าจะอาศัยวงจรที่ประกอบด้วยส่วนประกอบที่น้อยลงเพื่อลดราคาตัวบอร์ด ซึ่งสามารถทำงานได้ปกติแต่อาจเกิดเหตุการณ์ที่ไม่คาดคิด เช่น แรงดันที่เข้ามาทางพอร์ต USB ต่ำกว่า 5 โวลต์ เป็นต้น ก็จะส่งผลกระทบต่อแรงดันทั้งระบบ รวมถึงแรงดันที่ไปเลี้ยงซีพียูและวงจรทางด้านเน็ตเวิร์คด้วย ปัญหาต่อมาคือการออกแบบเดิมเรียงลำดับของชุดวงจรแปลงแรงดัน (Regulator) แปลงจาก 5 ไปเป็น 3.3 โวลต์ก่อน แล้วเอา 3.3 ไปเป็น 2.5 โวลต์อีกที แล้วสุดท้ายเอา 2.5 มาแปลงเหลือ 1.8 โวลต์ตามลำดับ ซึ่งแรงดันที่ตกคร่อมชุดแปลงแรงดันจาก 5 เป็น 3.3 โวลต์ นั้นสูงถึง 1.7 โวลต์ทำให้วงจรแปลงแรงดันชุดนี้มีความร้อนสูงกระทบกับต่อประสิทธิภาพในการแปลงแรงดันโดยรวม และฟิวส์ที่ใช้ป้องกันกระแสไฟฟ้าเกินทั้งระบบนั้นรับได้เพียง 1 แอมป์ ซึ่งน้อยเกินไปสำหรับการใช้งานในบางสถานการณ์และไม่มีการป้องกันการถอดเสียบแบบ Hot-Swap พอร์ต USB การเสียบอุปกรณ์ที่กินกระแสสูงอย่างพวก Wi-Fi Dongle อาจส่งผลให้อุปกรณ์ USB ทั้งหมดที่เชื่อมต่ออยู่หลุดจากระบบหรือในกรณีร้ายแรงคือทำให้ซีพียูและบอร์ดรีบูตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

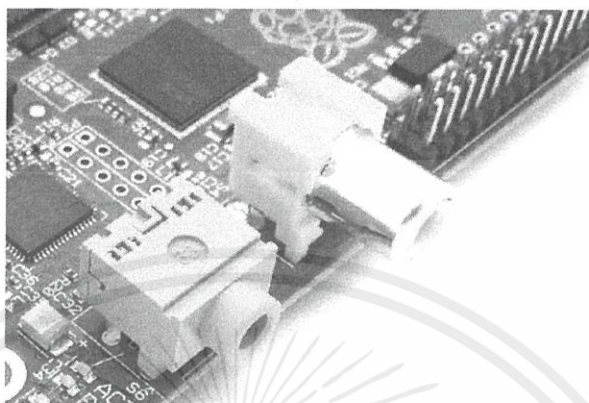


ภาพที่ 2.24 วงจรของ Model B+

ภาคจ่ายไฟชุดใหม่ของ Model B+ มีความเสถียรภาพมากขึ้นและยังลดการสูญเสียภายในวงจร โดยอัพเกรดให้ฟิวส์มีขนาด 2 แอมป์ และเปลี่ยนมาใช้ MOSFET แทน Diode เป็นตัวป้องกันในชั้นถัดมาช่วยลดการสูญเสียแรงดันของวงจรแหล่งจ่ายจาก 0.5 โวลต์เหลือ 0.1 โวลต์ ในส่วนของวงจรแปลงแรงดันใช้ Buck Converter แบบ Dual แทนของเดิม ลดการสูญเสียพลังงานจากความร้อนเมื่อแปลงเป็น 3.3 และ 1.8 โวลต์ และเพิ่มตัว Step-Down Converter สำหรับชุดของแรงดัน 3.3 โวลต์ และ Hot-Swap Protector เพื่อป้องกันในส่วนของการรักษาระดับแรงดัน 5V ทำให้สามารถถอดและเสียบอุปกรณ์ USB ในแบบ Hot-Swap ได้โดยไม่กระทบกับระบบ และผลรวมของการเลือกใช้ชุดจ่ายไฟใหม่นี้ทำให้ลดการสิ้นเปลืองพลังงานของบอร์ดลงได้อีก 0.5 - 1 วัตต์

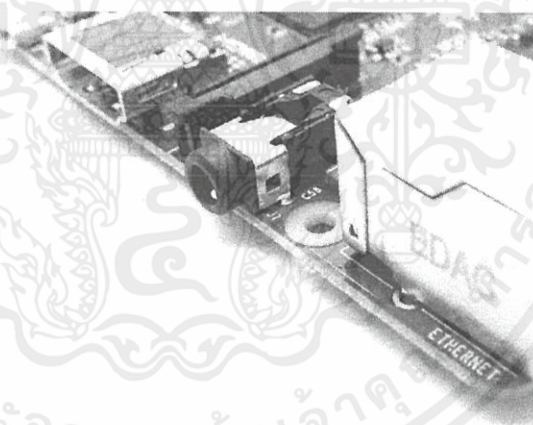
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วงจรเสียง Model B+ ได้รับการปรับปรุงด้านการป้องกันสัญญาณรบกวนให้ดีขึ้น
อีกเล็กน้อยจาก Model B ซึ่งเป็นผลพวงจากการปรับปรุงภาคจ่ายไฟของบอร์ด



ภาพที่ 2.25 คอนเน็คเตอร์ Audio Out Model B

ในส่วนของคอนเน็คเตอร์ Audio Out ได้รวมสัญญาณ Video Out จากคอนเน็คเตอร์
แบบ RCA เดิมมาไว้ด้วยกัน ทำให้คอนเน็คเตอร์ Audio Out ตัวใหม่เปลี่ยนหัวภายในจาก 3 แกน
เป็น 4



ภาพที่ 2.26 คอนเน็คเตอร์ Audio Out Model B+

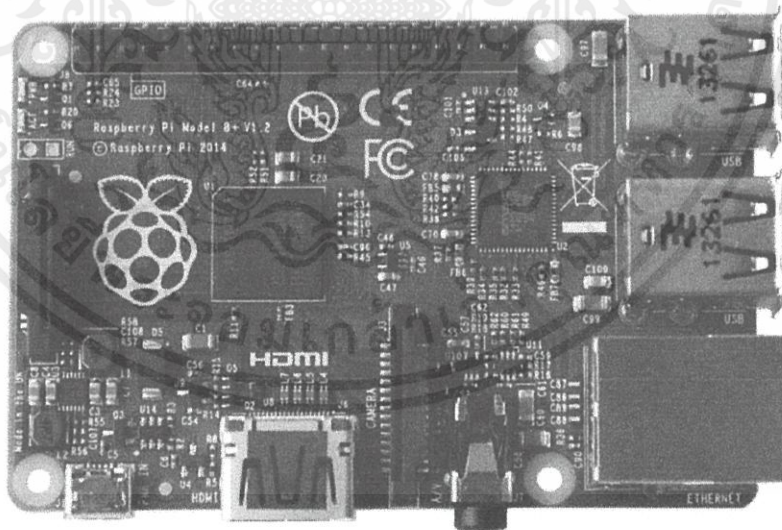
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การใช้งาน Audio Out กับ Video Out ต้องพึ่งพาอุปกรณ์เสริมเป็นสายแปลงหัว จาก 4 ขั้วให้เป็น Audio Out แบบ Stereo (สีขาวกับสีแดง) และ Video Out (สีเหลือง) แต่ถ้าใช้งาน HDMI Audio ก็ต่อผ่านพอร์ต HDMI ได้เช่นเดิม



ภาพที่ 2.27 สายแปลง 4 ขั้วเป็น RCA 3 เส้น

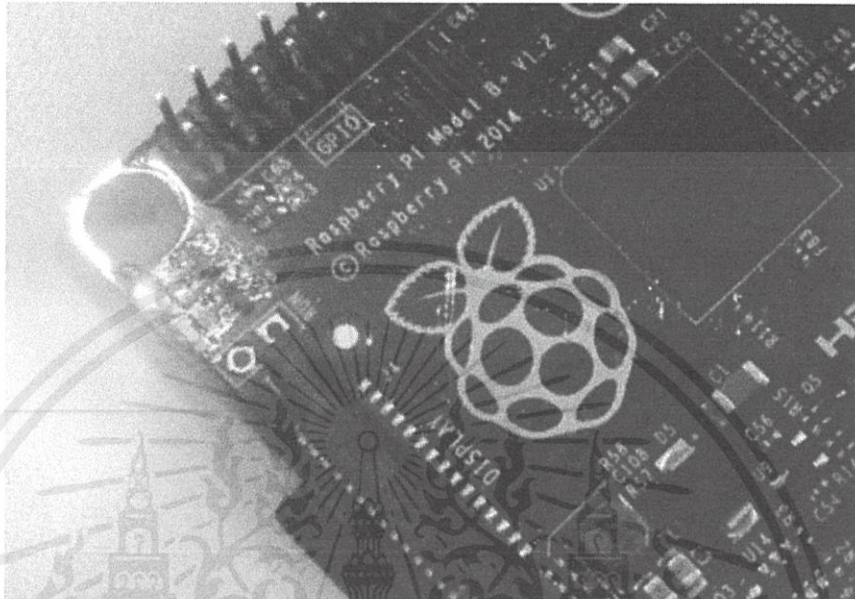
- ตำแหน่งของจุดยึดน็อตบนแผงวงจรและตำแหน่งคอนเน็คเตอร์ต่างๆ เปลี่ยนไป



ภาพที่ 2.28 Top View บอร์ด Raspberry Pi Model B+

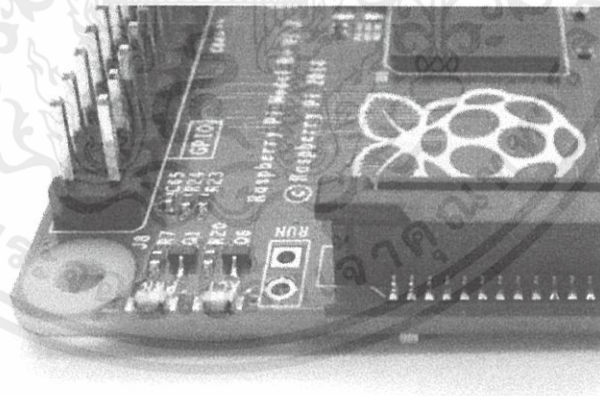
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- LED แสดงผลบนบอร์ดก็เปลี่ยนตำแหน่งไปและลดจำนวนลงจากเดิมจะมี LED แสดงสถานะต่างๆ ได้แก่ Power Activity Full-Duplex Link 100Mbps



ภาพที่ 2.29 LED แสดงผลบนบอร์ด

- ลดเหลือเพียง LED Power สำหรับแสดงสถานะไฟเลี้ยงเข้าบอร์ด และ Activity สำหรับแสดงสถานะเมื่อ CPU ทำงานเท่านั้น



ภาพที่ 2.30 สถานะไฟเลี้ยงเข้าบอร์ด และ Activity

- ส่วนไฟแสดงสถานะของ LAN ย้ายไปอยู่บนพอร์ต RJ-45 แบบใหม่ที่มีไฟ Link กับ Activity ในตัว เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ภาษาไพธอน

ภาษาไพธอน พัฒนาโดยชาวเนเธอร์แลนด์ชื่อ Guido van Rossum ซึ่งได้ออกแบบมาเพื่อให้ทำงานได้กับ Web Application ที่ลักษณะคล้ายกับภาษา Perl, PHP และ JAVA ซึ่งภาษาไพธอน มีคุณสมบัติที่ดีดังต่อไปนี้

2.2.1 สามารถใช้ได้ทุกแพลตฟอร์ม กล่าวคือ สามารถทำงานได้ทุก ๆ CPU หลาย ๆ ระบบปฏิบัติการ เพียงแต่ผู้เขียนโปรแกรมเขียนจากแพลตฟอร์มใดๆ แล้วนำโปรแกรมที่ได้ไปให้ทำงานต่างแพลตฟอร์มกันได้

2.2.2 ไม่ต้องเสียค่าใช้จ่ายในการจัดซื้อโปรแกรมต้นฉบับ โดยปกติแล้วโปรแกรมภาษาทั่ว ๆ ไปจะต้องจัดซื้อโปรแกรมต้นฉบับเพื่อนำมาติดตั้งในราคาแพงมาก แต่โปรแกรมภาษาไพธอนสามารถดาวน์โหลดจาก www.python.org ได้โดยตรง แล้วนำมาติดตั้งและศึกษาการใช้ด้วยตนเอง เพราะเป็นโปรแกรมประเภท Open Source

2.2.3 ภาษาไพธอนได้นำเอาข้อดีของโปรแกรมในอดีตเข้ามาไว้ด้วยกัน เช่น ภาษา C, C++, Java และ Perl เป็นต้น

2.2.4 มีความปลอดภัยสูง เนื่องจากภาษาไพธอนทำงานอยู่ด้าน Server เป็นหลัก เมื่อมีการร้องขอจากเครื่อง Client จะประมวลผลที่เครื่อง Server ทำให้ผู้ใช้ทั่วไปไม่สามารถเข้าถึงเครื่อง Server ได้โดยตรงจึงมีความปลอดภัยสูงกว่า

2.2.5 ใช้ในการพัฒนา Web Service ซึ่งในปัจจุบันการพัฒนาซอฟต์แวร์ได้เน้นที่มีการแลกเปลี่ยนข้อมูลซึ่งกันและกันทั้งในองค์กรเดียวกันหรือแม้แต่ต่างองค์กรกัน ทำให้เกิดความสะดวกสบาย ไม่ต้องใช้ซอฟต์แวร์อื่นๆ มาแปลงข้อมูลเพื่อให้เข้ากันได้อีกต่อไป เรียนรู้ได้เร็วกว่าโปรแกรมภาษาอื่น ๆ เพราะมีโครงสร้างภาษาที่ไม่ซับซ้อน

เนื่องจากโปรแกรมภาษาไพธอน เป็นซอฟต์แวร์ประเภท open source จึงมีแหล่งรวบรวมให้ผู้ใช้ทั่วไปดาวน์โหลดมาติดตั้งได้ด้วยตนเองตามที่อยู่ของเว็บไซต์ <http://python.org/download>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ทฤษฎีพื้นฐานเกี่ยวกับ Sensor

Sensor เป็นอุปกรณ์ที่ใช้ตรวจจับสถานะใดๆ เช่น อุณหภูมิ สี แสง หรือวัตถุต่างๆ โดยอาศัยหลักการที่แตกต่างกันไปตามแต่ชนิด เช่น Sensor สีขาวดำ โดยอาศัยหลักการสะท้อนแสงของสีขาวและดำ ทางฟิสิกส์แล้วจะเห็นว่าสีขาวมีอัตราการสะท้อนแสงมากกว่าสีดำ จึงสามารถนำแสงสะท้อนมาเปรียบเทียบได้ โดยใช้ตัว Sensor คือ อุปกรณ์จำพวกโฟโต้ เช่น โฟโต้ไดโอด โฟโต้ทรานซิสเตอร์ LDR เป็นต้น ซึ่งมีความไวต่อแสงมาก โดยส่วนใหญ่จะแสดงผลเอาต์พุตในรูปของความต้านทานที่เปลี่ยนแปลงไปตามสถานะของตัว Sensor นั้นๆ ซึ่งในโครงการนี้เลือกใช้ Sensor Heart Rate Sensor เป็น Sensor ประเภทโฟโต้เพื่อใช้ในการวัดอัตราการเต้นของชีพจรขณะวิ่ง



รูปที่ 2.31 เครื่องมือวัดอัตราการเต้นของหัวใจ

2.4 อัตราการเต้นของหัวใจ

อัตราการเต้นของหัวใจ (Heart Rate) หมายถึงความเร็วของการบีบตัวของหัวใจในช่วงระยะเวลาหนึ่งๆ โดยทั่วไปนิยมใช้หน่วย "ครั้งต่อนาที" อัตราการเต้นของหัวใจสามารถเปลี่ยนแปลงได้ขึ้นกับสรีรวิทยาของร่างกาย เช่นความต้องการออกซิเจนและการขับออกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คาร์บอนไดออกไซด์ของร่างกาย สิ่งที่มีผลกับอัตราการเต้นของหัวใจได้แก่กิจกรรมของร่างกาย เช่น การออกกำลังกาย การนอนหลับ ความเจ็บป่วย การย่อยอาหาร และยา

อัตราการเต้นของหัวใจ สามารถแบ่งออกได้ตามช่วงเวลาได้ดังนี้

(1) Resting Heart Rate (RHR) - เป็นอัตราการเต้นของหัวใจขณะที่พัก ไม่ว่าจะป็นตอนที่ตื่นขึ้นมาตอนเช้าหรือในช่วงระหว่างวัน อัตราการเต้นขณะพักของผู้ชายควรอยู่ประมาณ 60-80 ครั้งต่อนาที ขณะที่ผู้หญิงอยู่ที่ 70-90 ครั้งต่อนาที ในกรณีที่ไม่ดูแลสุขภาพ อัตราการเต้นของหัวใจอาจจะเกิน 100 ครั้งต่อนาที

อัตราการเต้นของหัวใจขณะพักจะสามารถบ่งบอกถึงระดับความฟิตของร่างกายได้ นักกีฬาที่ออกกำลังกายสม่ำเสมออาจจะมีอัตราการเต้นของหัวใจขณะพักที่ 40-50 ครั้งต่อนาที นอกจากนี้ หากมีอัตราการเต้นของหัวใจค่อนข้างสูง อาจหมายถึงสัญญาณเตือนของการออกกำลังกายหนักเกินไปจนเกิดปัญหา Over Training หมายถึงเวลาที่ควรจะพักในการออกกำลังกายในวันนั้น หรือจะออกกำลังกายเพียงเล็กน้อย จนกว่าอัตราการเต้นจะกลับเข้ามาที่ระดับปกติ

(2) Maximum Heart Rate (MHR) - คืออัตราการเต้นสูงสุดที่ร่างกายจะรับได้ ซึ่งสามารถคำนวณโดยนำอายุหักออกจาก 220 เช่น หากอายุประมาณ 40 ปี อัตราการเต้นสูงสุดจะอยู่ที่ 220-40 คือ 180 ครั้งต่อนาที

(3) Training Heart Rate (THR) - ตามหลักแล้วควรจะทำให้ออกกำลังกายในระดับความเข้มข้นที่สูงเพียงพอให้เกิดผลจากการฝึกที่เพียงพอ (a sufficient training effect) แต่จะต้องไม่หนักเกินกว่าที่ร่างกายจะรับได้ ดังนั้นจึงควรจะทำให้ออกกำลังกายในระดับที่ทำให้อัตราการเต้นอยู่ในช่วง THR

American College of Sports Medicine (ACSM) กำหนดว่า ระดับต่ำสุดของ THR น่าจะอยู่ที่ 60% ของ MHR แต่ควรจะทำให้ออกกำลังกายให้อัตราการเต้นของหัวใจอยู่ที่ประมาณ 70-80% ของ MHR หากออกกำลังกายเกินค่านี้ จะเป็นช่วงเปลี่ยนการออกกำลังกายจาก Aerobic Exercise เป็น Anaerobic Exercise ซึ่งหัวใจไม่สามารถนำออกซิเจนเข้าสู่ร่างกายตามความต้องการได้ทัน ซึ่งจะออกกำลังกายได้ไม่นาน และจะเกิดความเสี่ยงต่อการเกิดอาการบาดเจ็บสูง และไม่เกิดผลดีต่อความฟิตของร่างกาย

อย่างไรก็ตาม คนที่เพิ่งเริ่มการออกกำลังกาย การกำหนดอัตราการเต้นที่ 70-80% ของ MHR อาจจะรวดเร็วเกินไป ดังนั้น เพื่อความปลอดภัยควรจะทำให้ออกกำลังกายให้อัตราการเต้นของหัวใจอยู่ในช่วง 60-80% ของ MHR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(4) Recovery Heart Rate เมื่อมีการออกกำลังกายแล้ว จะต้องมีการ cool-down เหมือนเป็นการผ่อนเครื่อง ให้ร่างกายค่อยๆลดระดับอัตราการเต้นของหัวใจลง ซึ่งควรจะลดลงมาอยู่ในระดับที่สูงกว่าอัตราการเต้นของหัวใจก่อนพัก ประมาณ 20 ครั้งต่อนาที หาก Recovery Heart Rate ลดลงช้า อาจเกิดจากการออกกำลังกายหนักเกินไปหรือ cool - down ร่างกายด้วยวิธีที่ไม่ถูกต้อง ทำให้อัตราการเต้นของหัวใจไม่เป็นไปตามเป้าหมาย

อัตราการเต้นของหัวใจตามเป้าหมายสำหรับผู้ออกกำลังกาย

คนอายุ 20 ปี อัตราการเต้นสูงสุด= 200ครั้ง อัตราการเต้นที่40%=80ครั้ง อัตราการเต้นที่ 60%=120ครั้ง อัตราการเต้นที่ 80%= 160ครั้ง อัตราการเต้นที่90%= 180ครั้ง

คนอายุ 25 ปี อัตราการเต้นสูงสุด= 195ครั้ง อัตราการเต้นที่40%=78ครั้ง อัตราการเต้นที่ 60%=117ครั้ง อัตราการเต้นที่ 80%= 156ครั้ง อัตราการเต้นที่90%= 175ครั้ง

คนอายุ 30 ปี อัตราการเต้นสูงสุด= 190ครั้ง อัตราการเต้นที่40%=76ครั้ง อัตราการเต้นที่ 60%=114ครั้ง อัตราการเต้นที่ 80%= 152ครั้ง อัตราการเต้นที่90%= 171ครั้ง

คนอายุ 35 ปี อัตราการเต้นสูงสุด= 185ครั้ง อัตราการเต้นที่40%=74ครั้ง อัตราการเต้นที่ 60%=111ครั้ง อัตราการเต้นที่ 80%= 148ครั้ง อัตราการเต้นที่90%= 166ครั้ง

คนอายุ 40 ปี อัตราการเต้นสูงสุด= 180ครั้ง อัตราการเต้นที่40%=72ครั้ง อัตราการเต้นที่ 60%=108ครั้ง อัตราการเต้นที่ 80%= 144ครั้ง อัตราการเต้นที่90%= 162ครั้ง

คนอายุ 45 ปี อัตราการเต้นสูงสุด= 175ครั้ง อัตราการเต้นที่40%=70ครั้ง อัตราการเต้นที่ 60%=105ครั้ง อัตราการเต้นที่ 80%= 140ครั้ง อัตราการเต้นที่90%= 157ครั้ง

คนอายุ 50 ปี อัตราการเต้นสูงสุด= 170ครั้ง อัตราการเต้นที่40%=68ครั้ง อัตราการเต้นที่ 60%=102ครั้ง อัตราการเต้นที่ 80%= 136ครั้ง อัตราการเต้นที่90%= 153ครั้ง

คนอายุ 55 ปี อัตราการเต้นสูงสุด= 165ครั้ง อัตราการเต้นที่40%=66ครั้ง อัตราการเต้นที่ 60%=99ครั้ง อัตราการเต้นที่ 80%= 132ครั้ง อัตราการเต้นที่90%= 149ครั้ง

คนอายุ 60 ปี อัตราการเต้นสูงสุด= 160ครั้ง อัตราการเต้นที่40%=64ครั้ง อัตราการเต้นที่ 60%=96ครั้ง อัตราการเต้นที่ 80%= 128ครั้ง อัตราการเต้นที่90%= 144ครั้ง

คนอายุ 65 ปี อัตราการเต้นสูงสุด= 155ครั้ง อัตราการเต้นที่40%=62ครั้ง อัตราการเต้นที่ 60%=93ครั้ง อัตราการเต้นที่ 80%= 124ครั้ง อัตราการเต้นที่90%= 140ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ทฤษฎีพื้นฐานเกี่ยวกับจีพีเอส (GPS)

ระบบกำหนดตำแหน่งบนโลก (Global Positioning System: GPS) คือระบบบอกตำแหน่งบนพื้นผิวโลก โดยอาศัยการคำนวณจากความถี่สัญญาณนาฬิกาที่ส่งมาจากดาวเทียมที่โคจรรอบโลกซึ่งทราบตำแหน่ง ทำให้ระบบนี้สามารถบอกตำแหน่ง ณ จุดที่สามารถรับสัญญาณได้ทั่วโลก โดยเครื่องรับสัญญาณ GPS สามารถคำนวณความเร็วและทิศทางนำมาใช้ร่วมกับโปรแกรมแผนที่ เพื่อใช้ในการนำทางได้

ระบบ GPS จะทำงานได้นั้น ต้องประกอบไปด้วย 3 ส่วนหลักๆ คือ

1. สถานีฐาน : มีหน้าที่ควบคุมการทำงาน รวมถึงวงโคจรของดาวเทียม และให้คำสั่งสัญญาณนาฬิกาที่ถูกต้อง กับดาวเทียม GPS สถานีใหญ่อยู่ที่ Falcon Air Force Base ประเทศอเมริกา และศูนย์ควบคุมย่อยอีก 5 จุด กระจายไปยังภูมิภาคต่าง ๆ ทั่วโลก
2. ดาวเทียม GPS : ส่วนของดาวเทียม GPS ในปัจจุบันจะมาจาก 4 ค่ายหลักๆ คือ อเมริกา รัสเซีย ยุโรป และจีน

- NAVSTAR (Navigation Satellite Timing and Ranging GPS)

NAVSTAR เป็นดาวเทียมของประเทศสหรัฐอเมริกา มีทั้งหมด 28 ดวงใช้งานจริง 24 ดวงอีก 4 ดวงเป็นตัวสำรอง บริหารงานโดย Department of Defense มีรัศมีวงโคจรจากพื้นโลก 20,162.81 กม.หรือ 12,600 ไมล์ ดาวเทียมแต่ละดวงใช้เวลาโคจรรอบโลก 12 ชั่วโมงต่อ 1 รอบ

- Galileo (Galileo Positioning System)

บริหารงานโดย ESA หรือ European Satellite Agency ถูกพัฒนาโดยสหภาพยุโรป ร่วมกับประเทศจีน อิสราเอล อินเดีย โมร็อกโก ซาอุดีอาระเบีย เกาหลีใต้ และยูเครน ปัจจุบันมีทั้งหมดรวมจำนวน 30 ดวง

- GLONASS (Global Navigation Satellite System)

พัฒนาโดย Russia VKS (Russia Military Space Force) ปัจจุบันมี 30 ดวง

- Beidou (BDS : Beidou Satellite System)

เป็นดาวเทียม GPS ที่กำลังพัฒนาโดยประเทศจีน ปัจจุบันมีการส่งขึ้นไปแล้วจำนวน 16 ดวง เพื่อใช้ในระบบดาวเทียมบอกพิกัดนำทาง กับการใช้งานทางด้านการบริการโทรคมนาคมที่เรียกว่า Beidou navigation system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เครื่องรับสัญญาณ GPS : ผู้ใช้งานต้องมีเครื่องรับสัญญาณที่สามารถรับคลื่นและแปลงรหัสจากดาวเทียมเพื่อนำมาประมวลผลให้เหมาะสมกับการใช้งานในรูปแบบต่าง ๆ สามารถรับสัญญาณ GPS ได้จากอุปกรณ์หลายๆอย่าง เช่น โทรศัพท์เคลื่อนที่ที่รับสัญญาณ GPS ได้, GPS Receiver (ต่อกับ computer, โทรศัพท์เคลื่อนที่) หรือ GPS Module

ความแม่นยำของตัวรับสัญญาณ GPS ขึ้นอยู่กับจำนวนดาวเทียมที่สามารถรับสัญญาณได้ในขณะนั้น การระบุพิกัดขั้นต่ำ ต้องการดาวเทียมอย่างน้อย 3 ดวง เพื่อระบุตำแหน่งในแกน X, Y, Z โดยที่ X คือข้อมูลว่าดาวเทียมนี้คือตัวไหน, Y คือตำแหน่งของดาวเทียม และ Z คือเวลาที่ส่งข้อมูลในขณะนี้ ให้ได้ตำแหน่งที่ถูกต้อง โดยตัวเครื่อง GPS จะนำสัญญาณนาฬิกา ที่ได้รับจากดาวเทียมทั้ง 3 ดวง มาคำนวณระยะห่างของเครื่องรับสัญญาณ GPS กับดาวเทียมเพื่อแจ้งพิกัด ปัจจุบันถ้าระยะห่างระหว่างดาวเทียมที่ใช้งานอยู่ห่างกัน จะให้ค่าที่แม่นยำกว่าที่อยู่ใกล้กัน และยังมีจำนวนดาวเทียมที่รับสัญญาณได้มากก็ยิ่งให้ความแม่นยำมากขึ้น นอกจากนี้ยังมีความแปรปรวนของชั้นบรรยากาศซึ่งชั้นบรรยากาศประกอบด้วยประจุไฟฟ้า ความชื้น อุณหภูมิ และความหนาแน่นที่แปรปรวนตลอดเวลา คลื่นเมื่อตกกระทบกับวัตถุต่างๆ จะเกิดการหักเหทำให้สัญญาณที่ได้อ่อนลงและสิ่งแวดล้อมในบริเวณรับสัญญาณเช่น มีการบดบังจากกระจก ละอองน้ำ ใบไม้ จะมีผลต่อค่าความถูกต้องของความแม่นยำ เนื่องจากถ้าสัญญาณจากดาวเทียมมีการหักเหก็จะทำให้ค่าที่คำนวณได้จากเครื่องรับสัญญาณผิดเพี้ยนไป และสุดท้ายก็คือประสิทธิภาพของเครื่องรับสัญญาณว่ามีความไวในการรับสัญญาณแค่ไหนตลอดจนความเร็วในการประมวลผลด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

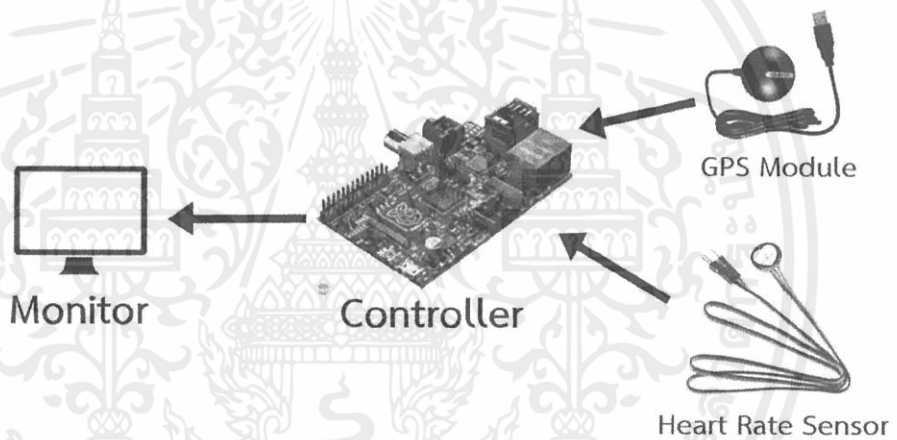
บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 Block diagram

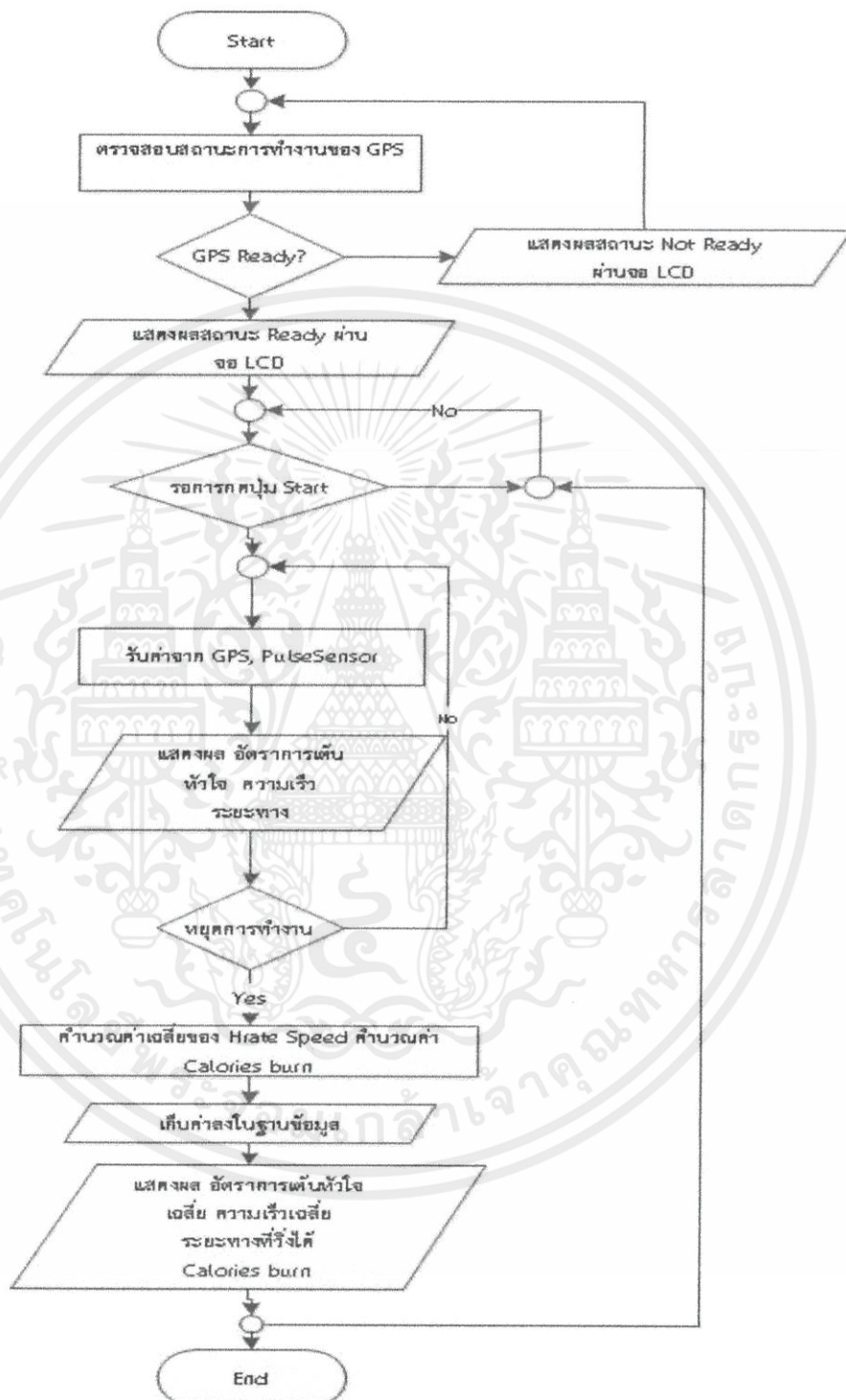
การออกแบบของระบบจะแบ่งออกเป็น 4 ส่วน คือ GPS Module จะรับค่าพิกัดและความเร็วจากดาวเทียม Heart Rate Sensor จะทำการวัดค่าอัตราการเต้นของหัวใจ และในส่วนประมวลผลจะใช้เป็นตัวบอร์ด Raspberry Pi เป็นไปตามรูป



รูปที่ 3.1 Block diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

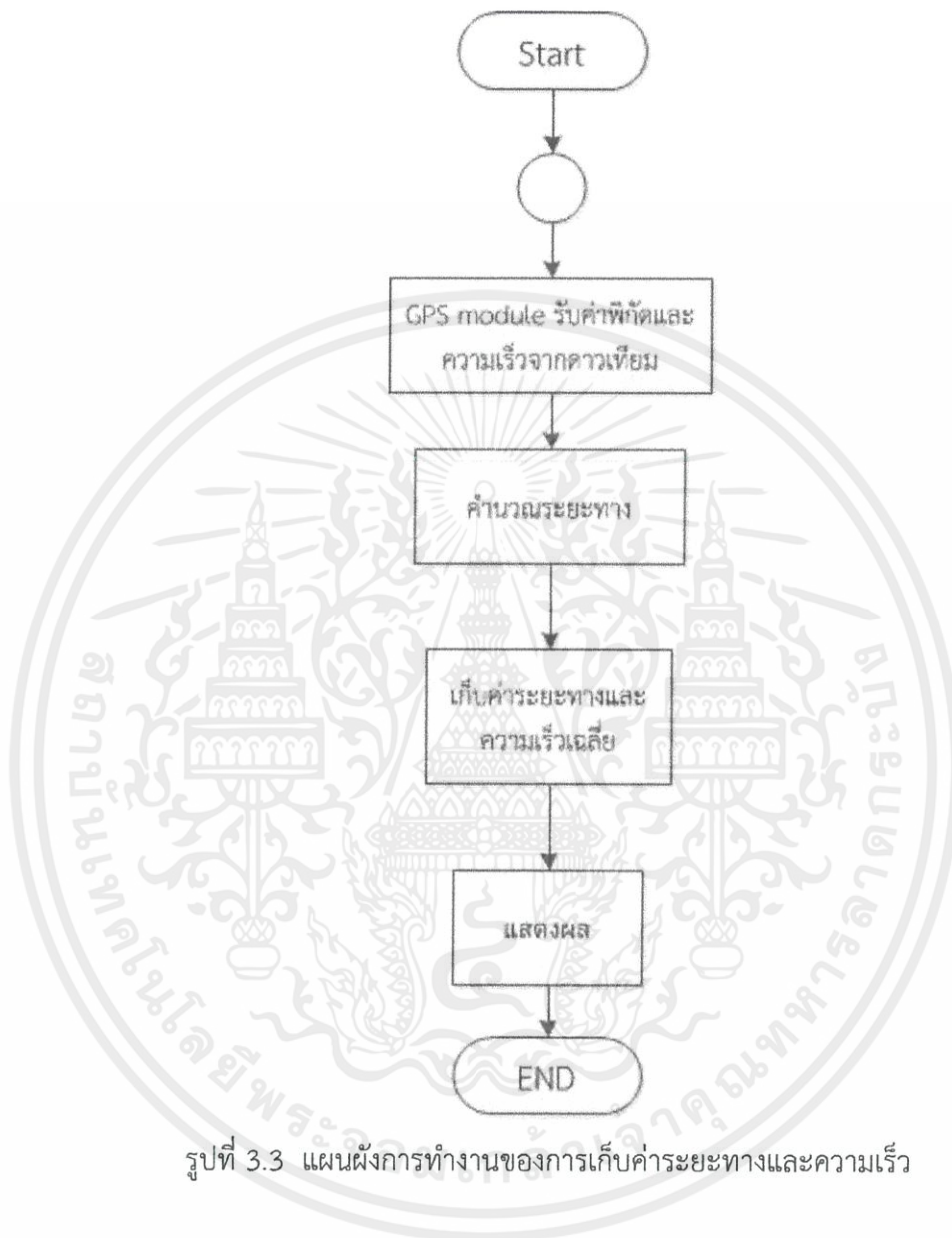
3.1.2 ระบบการทำงานรวมของระบบ



รูป 3.2 แผนผังการทำงานรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

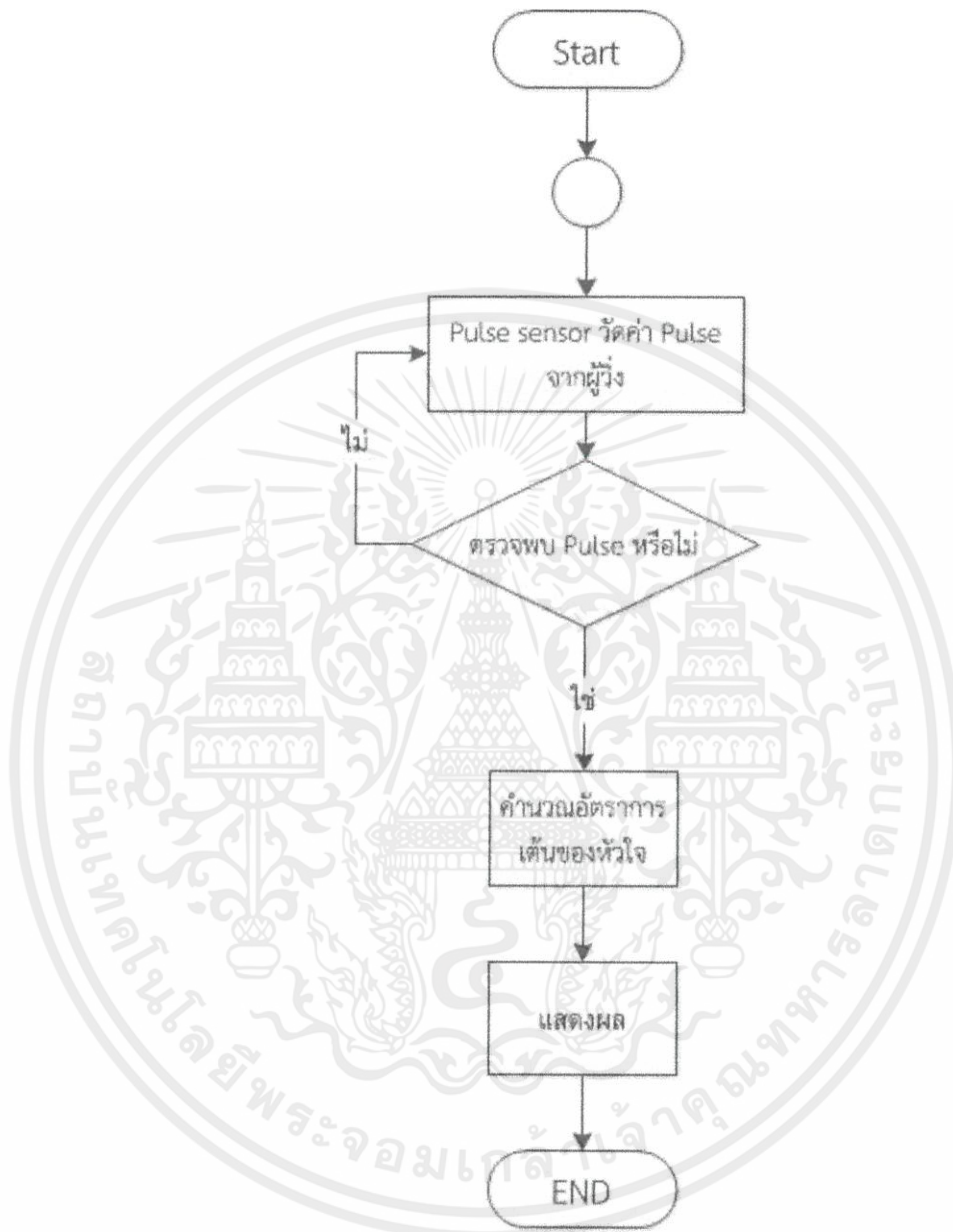
3.1.3 การทำงานของการเก็บค่าระยะทางและความเร็ว



รูปที่ 3.3 แผนผังการทำงานของกรเก็บค่าระยะทางและความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การทำงานของการวัดอัตราการเต้นของหัวใจ

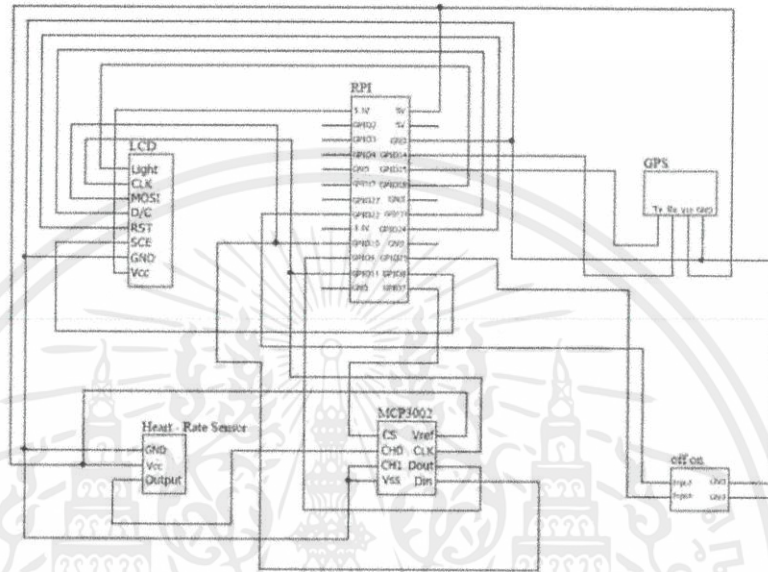


รูปที่ 3.4 แผนผังการทำงานของ การวัดอัตราการเต้นของหัวใจ

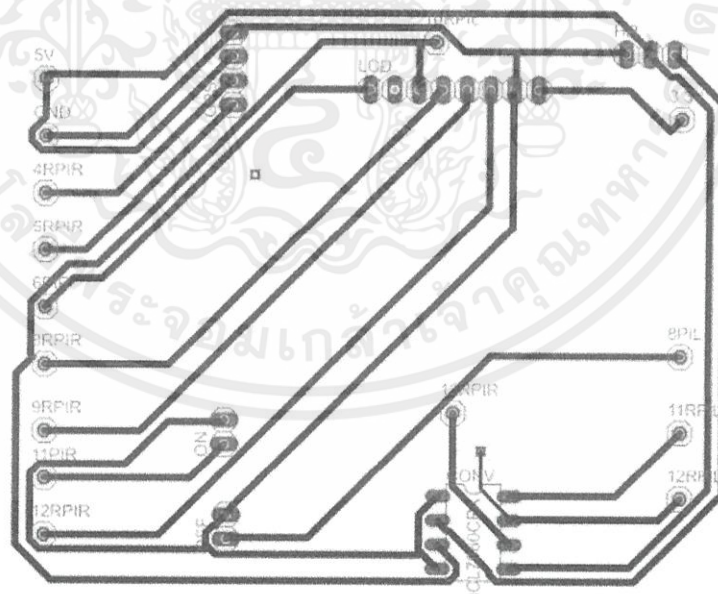
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 การออกแบบวงจรเพื่อใช้เชื่อมต่อกับบอร์ด Raspberry Pi

การออกแบบวงจรเพื่อใช้ในการเชื่อมต่อบอร์ด Raspberry Pi เข้ากับ GPS Module และ Sensor การวัดอัตราการเต้นของหัวใจ โดยใช้โปรแกรม Altium วาดวงจร



รูปที่ 3.5 การออกแบบ schematic วงจรเพื่อใช้ในการทำงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.6 ผลการออกแบบวงจรเพื่อใช้ในการทำงานไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

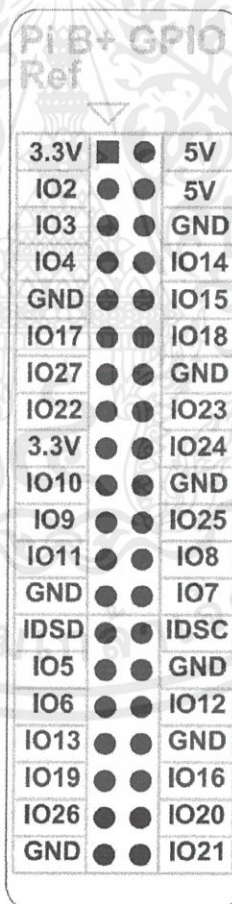
3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 Raspberry Pi Model B+ (บอร์ดคอมพิวเตอร์ 32 บิตขนาดเล็ก)

บอร์ดไมโครคอมพิวเตอร์แบบแผ่นเดี่ยวที่บรรจุความสามารถไว้เพื่อรองรับระบบปฏิบัติการ Raspbian บรรจุลงใน Micro SD Card การ์ดสำหรับการพัฒนา พร้อมจุดเชื่อมต่ออุปกรณ์อินพุตเอาต์พุตทั้งผ่านพอร์ต USB, LAN, HDMI, ช่องสัญญาณภาพและ GPIO สำหรับต่อกับวงจรหรืออุปกรณ์อิเล็กทรอนิกส์

3.2.2 GPIO

บอร์ด Raspberry Pi มีขา Input, Output ดิจิตอลหรือ GPIO สำหรับติดต่อกับอุปกรณ์ภายนอก รวม 40 ขา โดยมีการจัดเรียงขามาตรฐานตามรูป

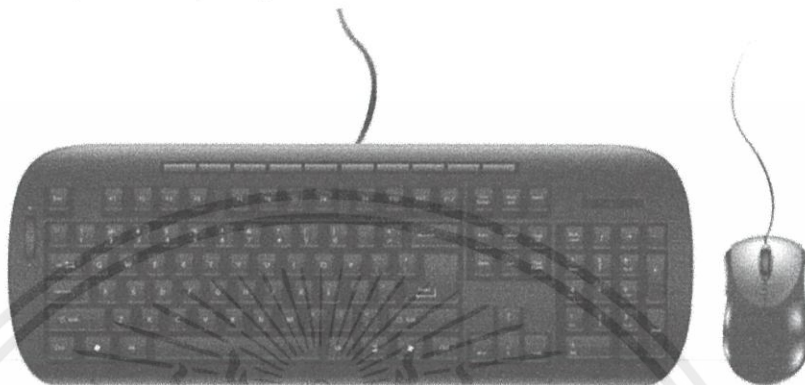


Pi B+ GPIO		Ref	
3.3V	5V	3.3V	5V
IO2	5V	IO2	5V
IO3	GND	IO3	GND
IO4	IO14	IO4	IO14
GND	IO15	GND	IO15
IO17	IO18	IO17	IO18
IO27	GND	IO27	GND
IO22	IO23	IO22	IO23
3.3V	IO24	3.3V	IO24
IO10	GND	IO10	GND
IO9	IO25	IO9	IO25
IO11	IO8	IO11	IO8
GND	IO7	GND	IO7
IDSD	IDSC	IDSD	IDSC
IO5	GND	IO5	GND
IO6	IO12	IO6	IO12
IO13	GND	IO13	GND
IO19	IO16	IO19	IO16
IO26	IO20	IO26	IO20
GND	IO21	GND	IO21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ส่วนบุคคลเท่านั้น กรุณาให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 Keyboard & Mouse

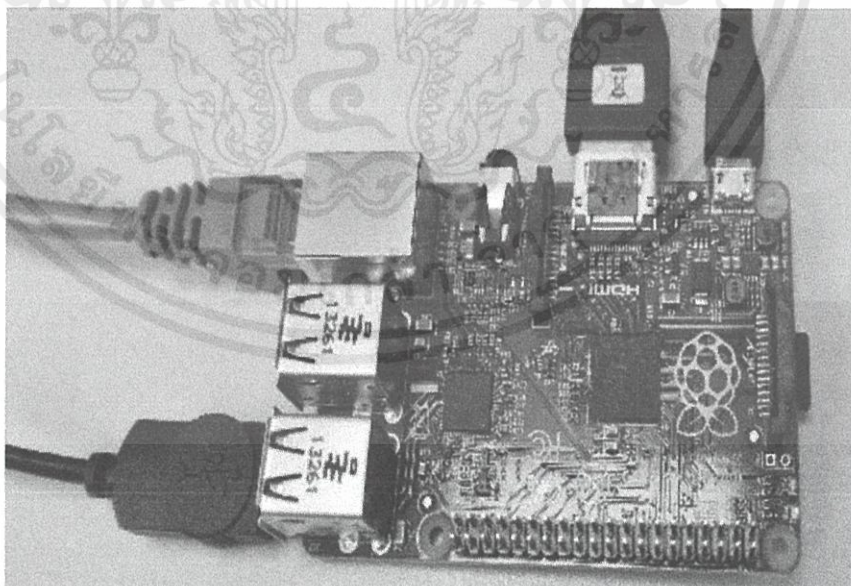
เป็นอุปกรณ์ที่ใช้ในการทำงานร่วมกับ Raspberry Pi เพื่อใช้ในการป้อนคำสั่งของโปรแกรมส่วนต่างๆให้กับ Raspberry Pi



รูปที่ 3.8 Key-board & Mouse

3.2.4 สาย LAN

เป็นอุปกรณ์ที่ใช้ในการเชื่อมต่อ Raspberry Pi เพื่อต่อเข้ากับคอมพิวเตอร์ในการดูข้อมูลประวัติการวิ่งย้อนหลัง

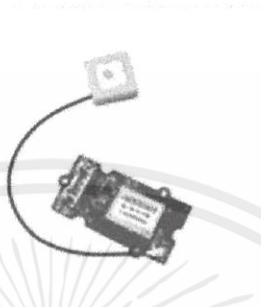


รูปที่ 3.9 สาย LAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 จีพีเอส โมดูล (GPS Module)

อุปกรณ์เพื่อใช้รับพิกัดตำแหน่งของผู้วิ่งขณะวิ่ง โดยจะสามารถรับค่าได้ทั้งความเร็ว และระยะทางที่ใช้ในการวิ่งในแต่ละครั้ง



รูปที่ 3.10 GPS Module

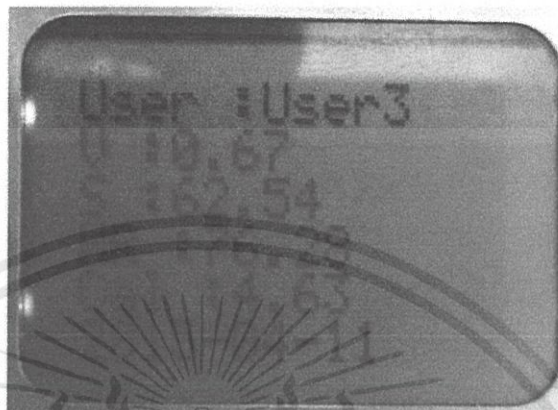
Features

- Grove compatible interface
- Ultra high sensitivity : -160dBm
- Low power consumption : Max 40mA@3.0V
- Configurable from 4800 Baud to 57600 Baud rates
- Supports NMEA and U-Blox 6 protocols

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 จอแสดงผล LCD รุ่น PCD 8554

อุปกรณ์เพื่อใช้แสดงผลข้อมูลที่แสดงอัตราการวิ่งในแต่ละครั้ง



รูปที่ 3.11 จอแสดงผล

Features

- Single chip LCD controller/driver
- 48 row, 84 column outputs
- Display data RAM 48 × 84 bits
- On-chip:
 - Generation of LCD supply voltage (external supply also possible)
 - Generation of intermediate LCD bias voltages
 - Oscillator requires no external components (external clock also possible).
- External RES (reset) input pin
- Serial interface maximum 4.0 Mbits/s
- CMOS compatible inputs
- Mux rate: 48
- Logic supply voltage range VDD to VSS: 2.7 to 3.3 V
- Display supply voltage range VLCD to VSS
 - 6.0 to 8.5 V with LCD voltage internally generated (Voltage generator enabled)
 - 6.0 to 9.0 V with LCD voltage externally supplied (Voltage generator switched-off)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตเห็นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การจัดเก็บผลการทดลอง

3.3.1 การทำให้ Raspberry Pi สามารถติดต่อกับ GPS Module ได้เพื่อใช้หาตำแหน่งของตัวผู้วิ่งและนำไปใช้ประมวลผลเพื่อหาระยะทางในการวิ่ง

3.3.2 การเขียนชุดคำสั่งเพื่อรับค่าจาก GPS Module เพื่อใช้ในการเริ่มและหยุดการรับข้อมูลและดึงค่าที่ได้รับมาจาก GPS Module ออกมา

3.3.3 การเขียนชุดคำสั่งเพื่อรับค่าจาก Heart - Rate Sensor

3.3.4 เก็บข้อมูลการรับพิกัดระยะทางและการคำนวณความเร็ว

3.3.5 เก็บข้อมูลการวัดอัตราการเต้นของหัวใจโดยตัว Sensor กับ Analog to Digital Converter (ADC) และ Raspberry Pi เพื่อรับข้อมูลจาก Heart - Rate Sensor เพื่อใช้ในการวัดอัตราการเต้นของหัวใจ

3.3.6 การทดลองการแสดงผลผ่านจอแสดงผลโดยทดลองต่อกับเข้าบอร์ด Raspberry Pi เพื่อประมวลผลข้อมูลก่อนจะแสดงค่าออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

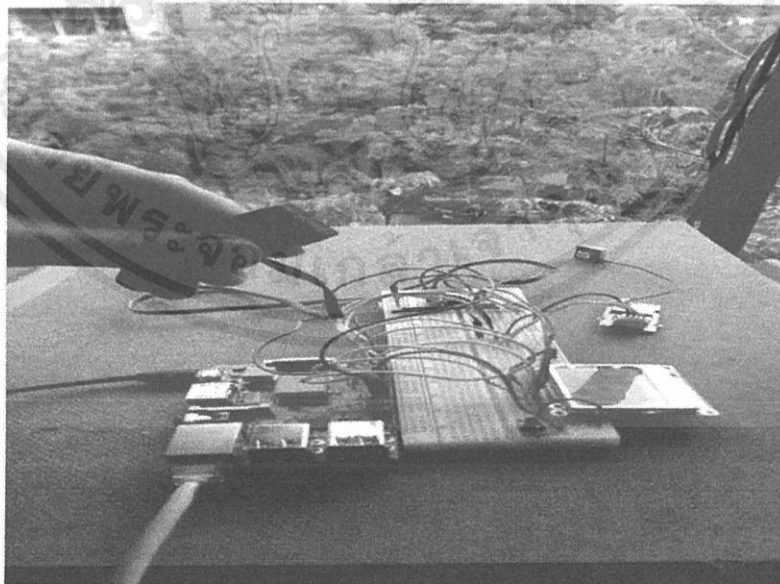
โครงการนี้เป็นการพัฒนาาระบบวัดชีพจรขณะวิ่ง วัดระยะทางที่วิ่งได้และสามารถคำนวณแคลอรีที่ใช้ และเพื่อใช้ในการตรวจสอบชีพจรขณะผู้ใช้งานกำลังวิ่ง

โดยการเริ่มจากที่ทำให้ Sensor ทั้งหมดให้สามารถทำงานได้และโปรแกรมไมโครคอนโทรลเลอร์รับข้อมูลจากอุปกรณ์ได้โดยใช้ Raspberry Pi ซึ่งจะเป็นการเขียนโปรแกรมโดยใช้ภาษา Python โดย Sensor จะทำหน้าที่ตรวจชีพจร และส่งข้อมูลเพื่อไปประมวลผลโดย Raspberry Pi และ Raspberry Pi จะสั่งให้แสดงผลข้อมูลชีพจรว่ามีอัตราการเต้นเป็นเท่าใด

4.1 การต่อ GPS Module เข้ากับระบบ

ขั้นตอนแรกเป็นการทำให้ Raspberry Pi สามารถติดต่อกับ GPS Module ได้เพื่อใช้หาตำแหน่งของตัวผู้วิ่งและนำข้อมูลไปประมวลผลหาระยะทางในการวิ่ง และความเร็วที่วิ่งในขณะนั้น

4.1.1 ทดลองต่อ GPS Module กับ Raspberry Pi เพื่อรับข้อมูลพิกัดจาก GPS Module



เอกสารนี้เป็นเอกสารที่สงวนไว้เฉพาะรูปที่ 4.1 การประกอบอุปกรณ์ต่างๆเข้าไว้ด้วยกัน เป็นประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ทดลองทำการรับค่าจาก GPS Module โดยใช้ คำสั่ง `sudo cgps -s` เพื่อทดสอบการทำงานของ GPS Module ว่าอยู่ในสภาพพร้อมใช้งานในการรับค่าหรือไม่ โดยการเชื่อมต่อกับ Raspberry Pi ผ่านโปรแกรม GPS Daemon ซึ่งเป็นโปรแกรมที่ทดสอบการรับค่า GPS

```

pi@raspberrypi: ~
File Edit Tabs Help

Time: 2014-10-22T18:19:58.000Z
Latitude: 13.726823 N
Longitude: 100.769888 E
Altitude: 22.2 m
Speed: 0.0 kph
Heading: 306.9 deg (true)
Climb: n/a
Status: 3D FIX (1 secs)
Longitude Err: +/- 9 m
Latitude Err: +/- 9 m
Altitude Err: +/- 33 m
Course Err: n/a
Speed Err: +/- 66 kph
Time offset: 0.779
Grid Square: OK03jr

FRN  Elev  Azim  SNR  Used:
 8    17    291   17    Y
14    24    119   30    Y
16    75    312   34    Y
19    17    186   27    Y
20    31    287   30    Y
23    16    325   19    Y
27    39    161   24    Y
31    24    044   32    Y
32    56    233   43    Y
 4    02    231   00    N
22    02    154   00    N
  
```

รูปที่ 4.2 ค่าที่รับได้จาก GPS

4.1.3 เขียนโปรแกรม เพื่อใช้ในการเริ่มและหยุดการเชื่อมต่อกับ GPS Module และดึงค่าที่ได้รับมาจาก GPS Module ออกมา โดยเลือกข้อมูลที่จะใช้งานคือ ความเร็ว ละติจูด ลองติจูด เพื่อคำนวณระยะทาง และแสดงผล ความเร็ว (V) และระยะทาง (S) โดยให้มีการอัปเดตค่าใหม่ทุก 1 วินาที

```

pi@raspberrypi: ~
File Edit Shell Debug Options Windows Help

S : 0.82
V : 0.6

S : 1.74
V : 1.4

S : 3.12
V : 2.0

S : 5.31
V : 3.3

S : 8.32
V : 5.5

S : 14.81
V : 9.8

S : 16.31
V : 2.4

S : 19.42
V : 2.2

S : 22.31
V : 1.8

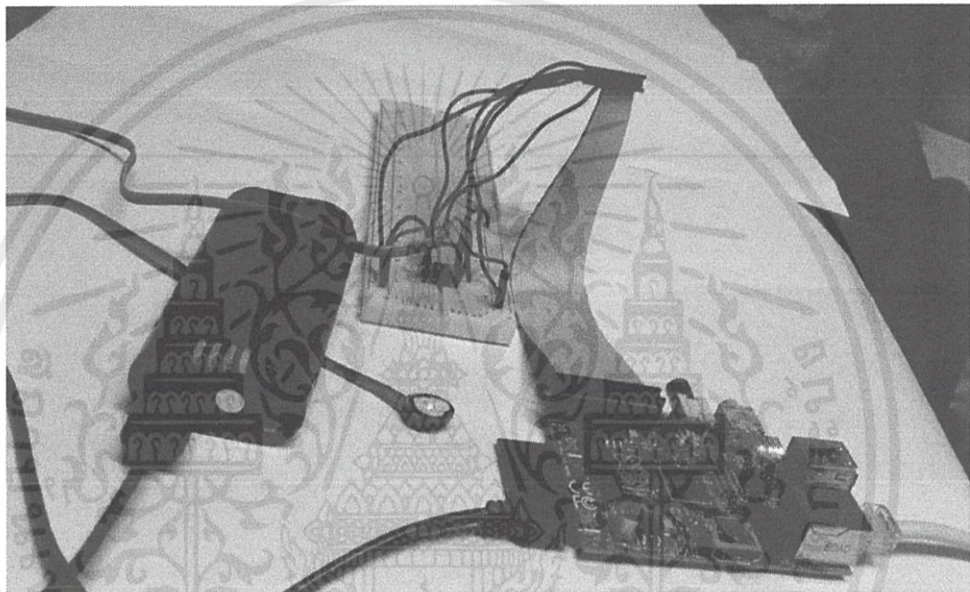
S : 25.45
V : 1.3
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 4.3 ทดลองรันโปรแกรม แสดงค่าที่รับจาก GPS Module
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การต่อ Heart-Rate Sensor เข้ากับระบบ

การเชื่อมระหว่างระบบการวัดอัตราการเต้นของหัวใจเข้าไปในระบบการทำงาน เพื่อที่จะสามารถทำงานร่วมกับการหาตำแหน่งของผู้วิ่ง และระยะทางในการวิ่ง

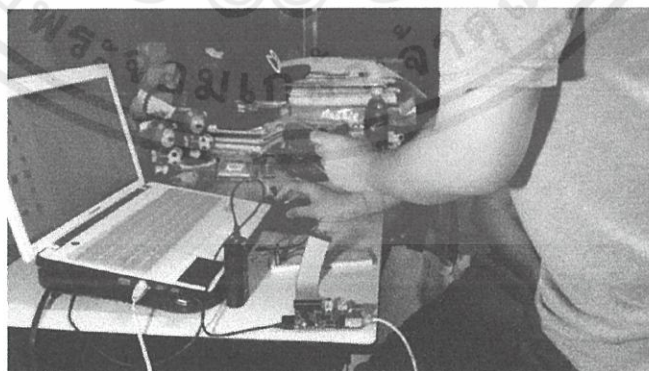
4.2.1 เริ่มต่อตัว Heart - Rate Sensor กับ Analog to Digital Converter (ADC) และ Raspberry Pi เพื่อรับข้อมูลจาก Heart - Rate Sensor เพื่อใช้ในการวัดอัตราการเต้นของหัวใจ



รูปที่ 4.4 การทดลองต่ออุปกรณ์ต่างๆเพื่อวัดอัตราการเต้นของหัวใจ

4.2.2 ทำการทดลองวัดค่าโดยวัดจากนิ้วมือของผู้ทดลองเพื่อหาอัตราการเต้นของ

หัวใจ



รูปที่ 4.5 การทดสอบการวัดอัตราการเต้นของหัวใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.5 การทดสอบการวัดอัตราการเต้นของหัวใจ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 หลังจากนั้นผลประมวลผลอัตราการเต้นของหัวใจที่วัดจากนิ้วมือของผู้ทดลอง จะทำให้ได้ค่าอัตราการเต้นของหัวใจออกมา โดยจะแสดงค่าออกมาในโปรแกรมเป็นค่าอัตราการเต้นของหัวใจเป็นจำนวนครั้งต่อนาที

```

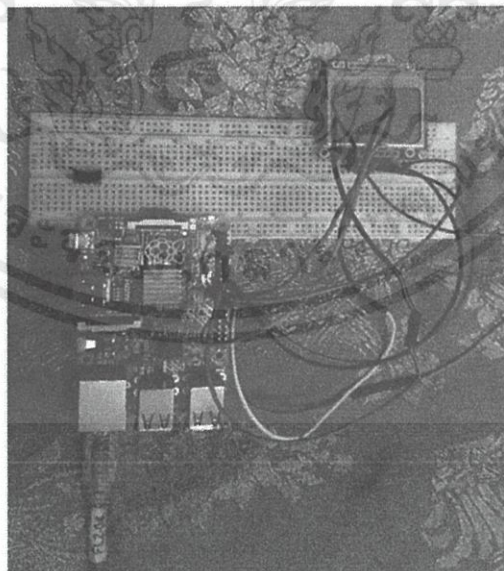
heart rate (bpm) : 105.47
heart rate (bpm) : 107.08
heart rate (bpm) : 103.74
heart rate (bpm) : 102.95
heart rate (bpm) : 102.32
heart rate (bpm) : 99.95
heart rate (bpm) : 99.5
heart rate (bpm) : 97.6
heart rate (bpm) : 91.86
heart rate (bpm) : 87.11
heart rate (bpm) : 87.19
heart rate (bpm) : 83.47
heart rate (bpm) : 85.58
heart rate (bpm) : 86.76
heart rate (bpm) : 86.96
heart rate (bpm) : 86.27
heart rate (bpm) : 86.34
heart rate (bpm) : 86.63
heart rate (bpm) : 86.76
heart rate (bpm) : 86.16

```

รูปที่ 4.6 การประมวลผลอัตราการเต้นของหัวใจ

4.3 การต่อหน้าจอแสดงผล

การทดลองการแสดงผลผ่านจอแสดงผล โดยทดลองต่อกับบอร์ด Raspberry Pi เพื่อประมวลผลข้อมูลก่อนจะแสดงค่าออกมาทางจอ LCD

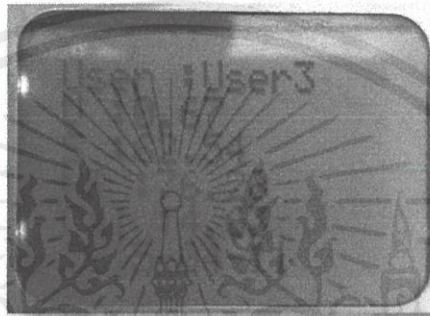


รูปที่ 4.7 การต่อจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายใน การศึกษา การวิจัย ไม่ควรนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำอุปกรณ์ส่วนการแสดงผลเพื่อใช้แสดงผลโดยมีการแบ่งเป็นส่วนๆ ดังต่อไปนี้

- User คือ ชื่อผู้ใช้งาน
- V คือ ความเร็ว หน่วยเมตรต่อวินาที
- S คือ ระยะทาง หน่วยเมตร
- HR คือ อัตราการเต้นของหัวใจ หน่วยครั้งต่อนาที
- วันที่ใช้งานขณะทำการวิ่ง



รูปที่ 4.8 จอแสดงผลการทำงาน

4.4 ส่วนแสดงผลทางเว็บเบราว์เซอร์

เมื่อแสดงผลผ่านจอ LCD ขณะวิ่งแล้ว ข้อมูลการวิ่งจะถูกบันทึกค่าเก็บไว้เพื่อแสดงผลผ่านทางเว็บแสดงผลโดยการเชื่อมข้อมูลใช้โปรแกรม MySQL ในการบันทึกข้อมูล และนำมาแสดงผลผ่านทางเบราว์เซอร์

←T→	Date	Speed	Distance	Heartrate	Calories	Duration
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-19 17:29:40	0.78	45.16	71.86	1.86	0:40
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-19 17:30:39	0.81	42.47	72.63	1.75	3:10
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-20 18:38:15	2.67	927.65	81.53	38.18	5:44
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-20 18:50:22	3.56	449.38	79.09	18.5	2:8
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-20 18:52:58	1.21	484.52	91.24	19.94	6:14
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-22 16:14:16	0.93	93.78	93.43	3.7	1:44
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-22 18:13:10	3.28	423.25	92.27	17.42	2:17
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-22 18:17:22	2.82	406.22	84.46	16.72	2:38
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-22 18:31:20	2.51	71.77	89.41	2.95	0:55
<input type="checkbox"/> Edit <input type="checkbox"/> Inline Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2015-04-22 18:33:48	2.59	99.1	93.41	4.08	0:41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปที่ 4.9 ผลการเก็บข้อมูลลงใน MySQL นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Date	Speed(m/s)	Distance(m)	Heartrate(bpm)	Calories(Kcal)	Duration(min)
2015-04-19 17:29:40	0.78	45.16	71.86	1.86	0:40
2015-04-19 17:30:39	0.81	42.47	72.63	1.75	3:10
2015-04-20 18:38:15	2.67	927.65	81.53	38.18	5:44
2015-04-20 18:50:22	3.56	449.38	79.09	18.5	2:8
2015-04-20 18:52:58	1.21	484.52	91.24	19.94	6:14
2015-04-22 16:14:16	0.93	93.78	93.43	3.7	1:44
2015-04-22 18:13:10	3.28	423.25	92.27	17.42	2:17
2015-04-22 18:17:22	2.82	406.22	84.46	16.72	2:38
2015-04-22 18:31:20	2.51	71.77	89.41	2.95	0:55
2015-04-22 18:33:48	2.59	99.1	93.41	4.08	0:41

รูปที่ 4.10 ผลข้อมูลในเว็บเพื่อแสดงรายละเอียดผลการวิ่ง

Running Status

Personaldata

Name : user
Age : 22
Weight : 72
Height : 179
AverageHeartrate : 76

Edt

Date	Speed(m/s)	Distance(m)	Heartrate(bpm)	Calories(Kcal)	Duration(min)
2015-04-19 17:29:40	0.78	45.16	71.86	1.86	0:40
2015-04-19 17:30:39	0.81	42.47	72.63	1.75	3:10
2015-04-20 18:38:15	2.67	927.65	81.53	38.18	5:44
2015-04-20 18:50:22	3.56	449.38	79.09	18.5	2:8
2015-04-20 18:52:58	1.21	484.52	91.24	19.94	6:14
2015-04-22 16:14:16	0.93	93.78	93.43	3.7	1:44
2015-04-22 18:13:10	3.28	423.25	92.27	17.42	2:17
2015-04-22 18:17:22	2.82	406.22	84.46	16.72	2:38
2015-04-22 18:31:20	2.51	71.77	89.41	2.95	0:55
2015-04-22 18:33:48	2.59	99.1	93.41	4.08	0:41

Clear Data
Back

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครู ใช้งานเพื่อการศึกษานานาชาติ ไปอนภาคนี้ไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.11 การเข้าหน้าเว็บเพื่อเข้าสู่ข้อมูลการใช้งาน
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Name: Update

Age: Update

Weight: Update

Height: Update

AverageHeartRate: Update

Back

รูปที่ 4.12 การตั้งค่าชื่อ อายุ น้ำหนัก ส่วนสูง และอัตราการเต้นหัวใจของผู้ใช้งาน

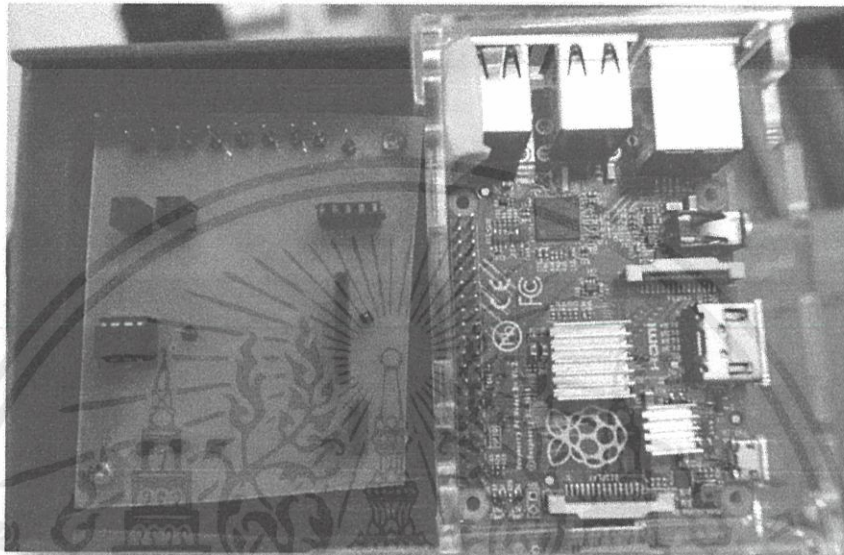
4.5 การประกอบอุปกรณ์

จัดทำวงจรเพื่อประกอบอุปกรณ์ต่างๆ โดยจะมีจุดที่เป็นเทอร์มินอลเพื่อเชื่อมต่อเข้ากับอุปกรณ์ต่างๆ ทั้ง Heart - Rate Sensor กับ GPS Module เพื่อใช้ตรวจสอบความเร็วของผู้วิ่ง ระยะทางที่วิ่ง และอัตราการเต้นของหัวใจ

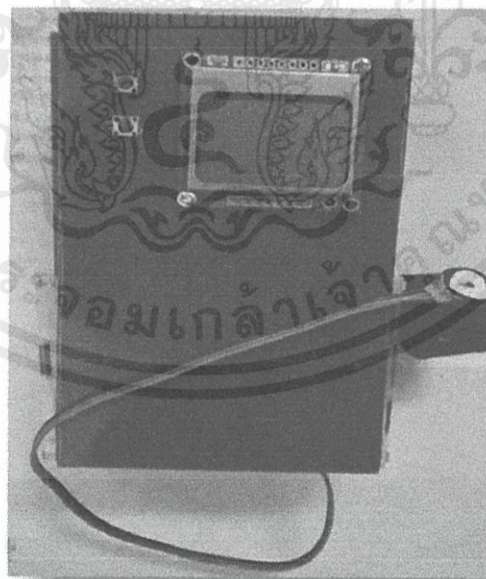


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 4.13 วงจรที่พร้อมทำการเชื่อมต่อเข้ากับระบบต่างๆ ซึ่งประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

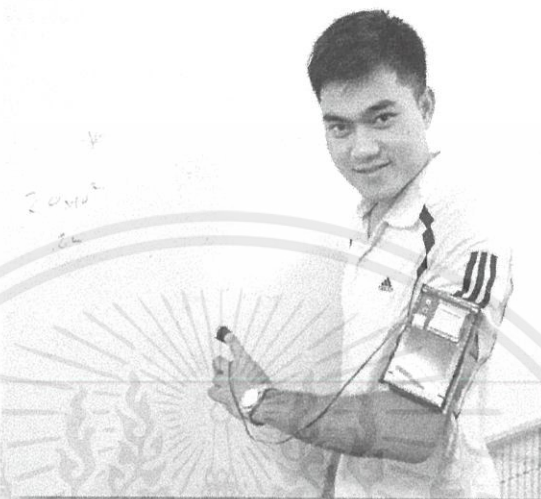
ส่วนการประกอบแผ่นวงจรเข้ากับบอร์ด Raspberry Pi เพื่อพร้อมใช้งานในขั้นตอนสุดท้าย พร้อมทั้งจะนำไปใช้งาน



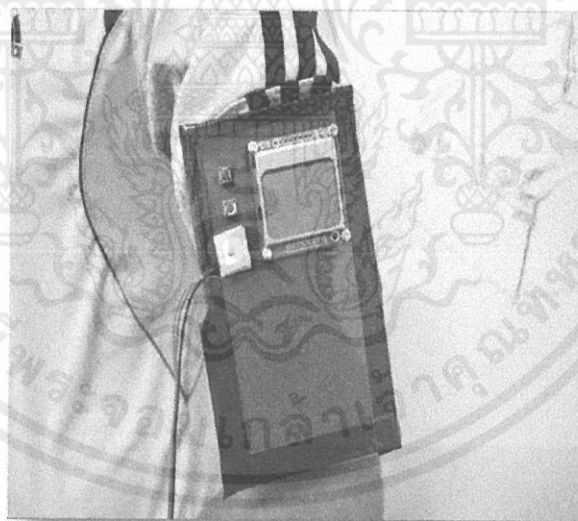
รูปที่ 4.14 การต่อวงจรที่ออกแบบเข้ากับบอร์ด Raspberry Pi



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.15 อุปกรณ์ที่ประกอบเสร็จสมบูรณ์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 การติดอุปกรณ์เข้ากับตัวผู้ใช้งาน



รูปที่ 4.17 อุปกรณ์ที่พร้อมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การพัฒนาระบบตรวจสอบสถานะการวิ่งเพื่อใช้ตรวจสอบสถานะของผู้วิ่งว่ามีความเร็วในการวิ่ง ระยะทางในการวิ่ง และอัตราการเต้นของหัวใจเป็นเท่าไร โดยใช้ Heart Rate Sensor และ GPS Module เป็นตัววัดค่า และ Raspberry Pi ประมวลผล เป็นระยะทาง ความเร็ว และอัตราการเต้นของหัวใจ โดยใช้ชุดคำสั่งภาษา Python สามารถทำการวัดค่าอัตราการเต้นของหัวใจ ระยะทางและความเร็ว จาก Heart Rate Sensor และ GPS Module จากนั้นได้นำข้อมูลที่ได้จากการวัดได้มาประมวลผลข้อมูลและแสดงผลผ่านทางจอแสดงผลได้ ซึ่งอุปกรณ์นี้จะช่วยอำนวยความสะดวกให้กับบุคคลที่ทำการวิ่งแล้วต้องการดูข้อมูลในขณะที่วิ่ง

5.2 ข้อเสนอแนะ

1. การนำอุปกรณ์ไปใช้งานต่อเนื่องอาจมีผลต่อตัวอุปกรณ์ อุปกรณ์ต้องมีการพัฒนาในด้านการจัดรูปแบบอุปกรณ์การทำงานเพื่อให้มีความทนทานมากขึ้น และใช้งานได้โดยมีความสะดวกมากขึ้น
2. การคำนวณระยะทางจาก GPS Module มีความคลาดเคลื่อนซึ่งอาจจะเกิดจากปัจจัยรบกวนต่างๆ เช่น สภาพอากาศ สถานที่ในการวิ่ง เป็นต้น เพื่อเพิ่มความแม่นยำควรใช้ GPS Module ที่มีความละเอียดสูง
3. ในการวัดอัตราการเต้นของหัวใจ อุปกรณ์ที่ใช้เป็น Heart Rate Sensor ที่ใช้การตรวจจับด้วยแสง ซึ่งมีความคลาดเคลื่อนที่เกิดจากการขยับของอุปกรณ์ เหนือของผู้วิ่ง ดังนั้นเพื่อเพิ่มความแม่นยำควรปรับให้อุปกรณ์มีความกระชับกับผู้ใช้งานมากขึ้น หรือเปลี่ยนไปใช้สายรัดหน้าอกวัดชีพจรแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] โชติพันธ์ หล่อเลิศสุนทร . “คู่มือเรียน เขียนโปรแกรม Python (ภาคปฏิบัติ)” : ซีเอ็ด ยูเคชั่นจำกัด, 2554
- [2] พร้อมเลิศ หล่อวิจิตร . “คู่มือเรียน PHP และ MySQL สำหรับผู้เริ่มต้น” : สำนักพิมพ์ โปรวีชั่น, 2550
- [3] <http://www.mindphp.com>
- [4] <http://www.pulsesensor.com>
- [5] <http://www.shapesense.com/fitness-exercise>
- [6] <http://www.raspberrypi.org>
- [7] Tanaka H., Monhan, Monahan KD., Seals, D.G., “Age-predicted maximal heart rate revisited”. J Am Coll Cardiol 2001 . ;
doi : 10.1016/S0735-1097(00)01054-8
- [8] บรรพต ดลวิทยากุล . “เก่ง Apache ให้ครบสูตร” : วิตต์กรู๊ป, 2549
- [9] จักรกฤษณ์ แสงแก้ว . “การเขียนโปรแกรมภาษาไพธอนด้วยตนเอง” : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2550
- [10] เจริญศักดิ์ รัตนวรราช, ฐิสันต์ ทิพย์ศุภธนนท์ . “PHP & MySQL for Web Programming” : ซีเอ็ดยูเคชั่นจำกัด, 2552

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก
โปรแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งการทำงานเบื้องต้นของโปรแกรม

```

from __future__ import division
import spidev, time, threading, os, sys, Queue, MySQLdb
import RPi.GPIO as GPIO
import pcd8544.lcd as lcd
from gps import *
from math import *
conn = spidev.SpiDev(0, 1)
conn.max_speed_hz = 1200000
adc_channel = 0
queue = Queue.Queue()
hqueue = Queue.Queue()
vqueue = Queue.Queue()
squeue = Queue.Queue()
ON, OFF = [1, 0]
db = MySQLdb.connect("localhost", "root", "raspberry", "project")
cur = db.cursor()
cur.execute("SELECT * FROM userdata WHERE Priority=1")
result = cur.fetchall()
for row in result:
    name = row[0]
    age = row[1]
    weight = row[3]
    nhr = row[5]
    surface_factor = row[6]
tf = 0.84
mhr = 208 - (0.7*age)
rhr = nhr*3
vo2max = 15.3*(mhr/rhr)
if vo2max >= 56:
    cff = 1
elif vo2max < 56 and vo2max >= 54:
    cff = 1.01
elif vo2max < 54 and vo2max >= 52:
    cff = 1.02
elif vo2max < 52 and vo2max >= 50:
    cff = 1.03
elif vo2max < 50 and vo2max >= 48:
    cff = 1.04
elif vo2max < 48 and vo2max >= 46:
    cff = 1.05
elif vo2max < 46 and vo2max >= 44:
    cff = 1.06
elif vo2max < 44:
    cff = 1.07
def bitstring(n):
    s = bin(n)[2:]
    return '0'*(8-len(s)) + s
def read():
    cmd = 128
    if adc_channel:
        cmd += 32
    reply_bytes = conn.xfer2([cmd, 0])

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นกรณีที่มีเหตุผลอันสมควรและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

reply_bitstring = ''.join(bitstring(n) for n in reply_bytes)
reply = reply_bitstring[5:15]
return int(reply, 2)

```

คำสั่งการควบคุมการทำงานของระบบ GPS

```

class gpscontrol(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
        self.gpsd = gps(mode=WATCH_ENABLE)
        self.running = False
    def run(self):
        self.running = True
        while self.running:
            self.gpsd.next()
    def stopgpscontrol(self):
        self.running = False
@property
def fix(self):
    return self.gpsd.fix
@property
def utc(self):
    return self.gpsd.utc

```

คำสั่งในส่วนในการคำนวณระยะทาง

```

def distance(lon1, lat1, lon2, lat2):
    lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * atan2(sqrt(a), sqrt(1-a))
    dis = 6371000 * c
    return dis

```

คำสั่งในส่วนการคำนวณ Sensor วัดอัตราการเต้นของชีพจร

```

class heartsensor(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
        self.running = False
    def run(self):
        self.running = True
        BPM = 0
        Signal = 0
        IBI = 600
        Pulse = False
        rate = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        sampleCounter = 0
        lastBeatTime = 0

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tresh = 512
amp = 100
firstBeat = True
secondBeat = True
N = 0
c1 = 0
c2 = 0
fingerFlag = True
while self.running:
    Signal = read()
    sampleCounter += 2
    N = sampleCounter - lastBeatTime
    if Signal < tresh and N > ((IBI/5)*3):
        if Signal < T:
            T = Signal

    if Signal > tresh and Signal > P:
        P = Signal
    if N > 250:
        if Signal > tresh*1.2 and Pulse == False and N >
(IBI/5)*3 and fingerFlag == True:
            Pulse = True
            c1 = 0
            c2 = 0
            IBI = sampleCounter - lastBeatTime
            lastBeatTime = sampleCounter
            if firstBeat:
                firstBeat = False
                continue
            if secondBeat:
                secondBeat = False
                for i in range(10):
                    rate[i] = IBI
                runningTotal = 0

            for i in range(9):
                rate[i] = rate[i+1]
                runningTotal += rate[i]
            rate[9] = IBI
            runningTotal += rate[9]
            runningTotal /= 10
            BPM = 60000/runningTotal
            queue.put(round(BPM, 2))

    if Signal > 850:
        c1 += 1
        if c1 > 250:
            fingerFlag = False
            tresh = 512
            P = 512
            T = 512
            rate = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
            lastBeatTime = sampleCounter

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        firstBeat = True
        secondBeat = True
        c1 = 0

    if fingerFlag == False:
        queue.put(0)
    if Signal < 100:
        c2 += 1
        if c2 > 250:
            fingerFlag = True
            c2 = 0
    if queue.qsize() > 5:
        with queue.mutex:
            queue.queue.clear()
    if Signal < tresh and Pulse == True:
        Pulse = False
        amp = P - T
        tresh = amp/2 + T
        P = tresh
        T = tresh
    if N > 2500:
        tresh = 512
        P = 512
        T = 512
        rate = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
        lastBeatTime = sampleCounter
        firstBeat = True
        secondBeat = True
        time.sleep(0.002)
    def stopsensor(self):
        self.running = False
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(25, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    GPIO.setup(22, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    startFlag = False
    gpsc = gpscontrol()
    heart = heartsensor()
    timestamp = 'none'
    s = 0
    gpsc.start()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งการกำหนดการทำงานของระบบ

```

def stopmain(channel):
    global startFlag
    if startFlag == True:
        startFlag = False
        totalh = 0
        totalv = 0
        n1 = hqueue.qsize()
        n2 = vqueue.qsize()
        while not hqueue.empty():
            ghr = hqueue.get()
            totalh += ghr
        while not vqueue.empty():
            gv = vqueue.get()
            totalv += gv
        if totalh == 0:
            avhr = 0
        else:
            avhr = round(totalh/n1,2)
        if totalv == 0:
            avv = 0
        else:
            avv = round(totalv/n2,2)
        if surface_factor == 0:
            calories_burn = round((0.95*weight +
tf)*(s/1000)*cff,2)
        elif surface_factor == -1:
            calories_burn = round((((-0.02*-10) + 0.35) * weight +
tf)*(s/1000)*cff,2)
        elif surface_factor == -2:
            calories_burn = round((((-0.01*-20) + 0.50) * weight +
tf)*(s/1000)*cff,2)
        elif surface_factor == 1:
            calories_burn = round((((0.05*10) + 0.95) * weight +
tf)*(s/1000)*cff,2)
        elif surface_factor == 2:
            calories_burn = round((((0.07*15) + 0.7) * weight +
tf)*(s/1000)*cff,2)
        cur.execute("UPDATE runstatus SET Speed=%s, Distance=%s,
Heartrate=%s, Calories=%s WHERE Date=%s",
(avv,s,avhr,calories_burn,timestamp))
        db.commit()
        lcd.cls()
        lcd.mtext(0,"User :" + str(name))
        lcd.mtext(1,"V :" + str(avv))
        lcd.mtext(2,"S :" + str(s))
        lcd.mtext(3,"HR :" + str(avhr))
        lcd.mtext(4,"Cal :" + str(calories_burn))
        lcd.mtext(5,str(timestamp[0:10]))
GPIO.wait_for_edge(25, GPIO.FALLING)
restart()
def restart():

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

python = sys.executable
os.execl(python, python, * sys.argv)
def main():
    global startFlag, timestamp, s
    time.sleep(2)
    lcd.init()
    lcd.backlight(ON)
    poll = gpsc.fix.altitude
    while str(poll) == 'nan' and startFlag == False:
        poll = gpsc.fix.altitude
        lcd.cls()
        lcd.mtext(2, "GPS not ready")
        time.sleep(0.5)
    lcd.cls()
    lcd.mtext(2, "GPS standby")
    lcd.mtext(3, "Press Start")
    GPIO.wait_for_edge(25, GPIO.FALLING)
    heart.start()
    startFlag = True
    GPIO.add_event_detect(22, GPIO.FALLING,
        callback=stopmain, bouncetime=500)
    try:
        utc = gpsc.utc
        gmt = int(utc[11:13]) + 7
        if gmt >= 24:
            gmt = gmt%24
        timestamp = utc[0:10] + " " + str(gmt) + utc[13:19]
        cur.execute("INSERT INTO runstatus
(Date, Speed, Distance, Heartrate, Calories)
VALUES (%s, 0, 0, 0, 0)", timestamp)
        db.commit()
        rlon1 = 0
        rlat1 = 0
        rlon2 = 0
        rlat2 = 0
        palt = 0
        ohrate = 0
        s = 0
        while startFlag == True:
            rlon2 = round(gpsc.fix.longitude, 5)
            rlat2 = round(gpsc.fix.latitude, 5)
            nalt = gpsc.fix.altitude
            lon1 = rlon1
            lat1 = rlat1
            lon2 = rlon2
            lat2 = rlat2
            dalt = nalt - palt
            if lon1 != 0:
                d = round(distance(lon1, lat1, lon2, lat2), 2)
                rd = round(sqrt(d**2 + dalt**2), 2)
                s += rd
            v = gpsc.fix.speed
            if queue.empty():

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        hrate = ohrate
    else:
        hrate = queue.get(
    if v != 0:
        vqueue.put(v)
    if hrate != 0:
        hqueue.put(hrate)

```

คำสั่งในการแสดงผลออกจอ LCD

```

        lcd.cls()
        lcd.mtext(0,"User :"+ str(name))
        lcd.mtext(1,"V :"+ str(v))
        lcd.mtext(2,"S :"+ str(round(s,2)))
        lcd.mtext(3,"HR :"+ str(hrate))
        lcd.mtext(5,str(timestamp[0:10]))
        ohrate = hrate
        rlon1 = rlon2
        rlat1 = rlat2
        palt = nalt
        time.sleep(1)
    except KeyboardInterrupt:
        print "Force Cancele by User"
    finally:
        pass
if __name__ == '__main__':
    main()

```

Code ส่วนหน้า login

```

<html>
<div align = "center">
<img src ="head.jpg">
<link rel = "stylesheet" type ="text/css" href="test.css">
<h1>
<form action="login.php" method="post">
    Username : <input type="text"
name="username"><p>
    Password : <input type = "password"
name="password"><p>
    <input type = "submit" value="Log in">
<div class="clear"></div>
    </form>
</h1></font></html>

```

Code ส่วนการแสดงผลการวิ่ง

```

<html><TITLE>PROJECT2</TITLE>
<div align="center">
<img src ="head.jpg">
<Head><CENTER><font size = "10">Personaldata</font></CENTER></Head>

```

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<body background="bg.jpg">
<?php
$connection = mysql_connect('localhost', 'root', 'raspberry');
mysql_select_db('project');

$query = "SELECT * FROM userdata";
$result = mysql_query($query);
?>
<table>
<?php
while($row = mysql_fetch_array($result)){
echo "<tr><td>Name : </td><td>" . $row['Name'] .
"</td></tr><tr><td>Age : </td><td>" . $row['Age'] .
"</td></tr><tr><td>Weight : </td><td>" . $row['Weight'] .
"</td></tr><tr><td>Height : </td><td>" . $row['Height'] .
"</td></tr><tr><td>AverageHeartrate : </td><td>" . $row['Heartrate']
. "</td></tr>";
}
?>
</table>
<?php
mysql_close();
?>
<form action = "Edit.html" method = "get">
    <button type = "submit">Edit</button>
<br><br>
<?php
$connection = mysql_connect('localhost', 'root', 'raspberry');
mysql_select_db('project');
?>
<table border="4" bordercolor="grey" cellpadding="2"
cellspacing="2">
    <tr>
<th>Date</th><th>Speed</th><th>Distance</th><th>Heartrate</th><th>Ca
lories</th> </tr>

<?php
    $result = mysql_query("SELECT * FROM runstatus;");
    $row = mysql_num_rows($result);
    for($i=0; $i<$row; $i++){
        $Date = mysql_result($result, $i, 'Date');
        $Speed = mysql_result($result, $i, 'Speed');
        $Distance = mysql_result($result, $i, 'Distance');
        $Heartrate = mysql_result($result, $i,
'Heartrate');
        $Calories = mysql_result($result, $i, 'Calories');

        echo "<tr> <td>$Date</td><td><p align='center'>
$Speed</p></td><td>$Distance</td><td>$Heartrate</td><td>$Calories</t
d> </tr>";
    }
echo"</table>";
mysql_close();

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
?>
<br><br>
<a href="index.html"><font size="4">Back</font></a>
</html>
```

Code ส่วนการแก้ไขข้อมูล

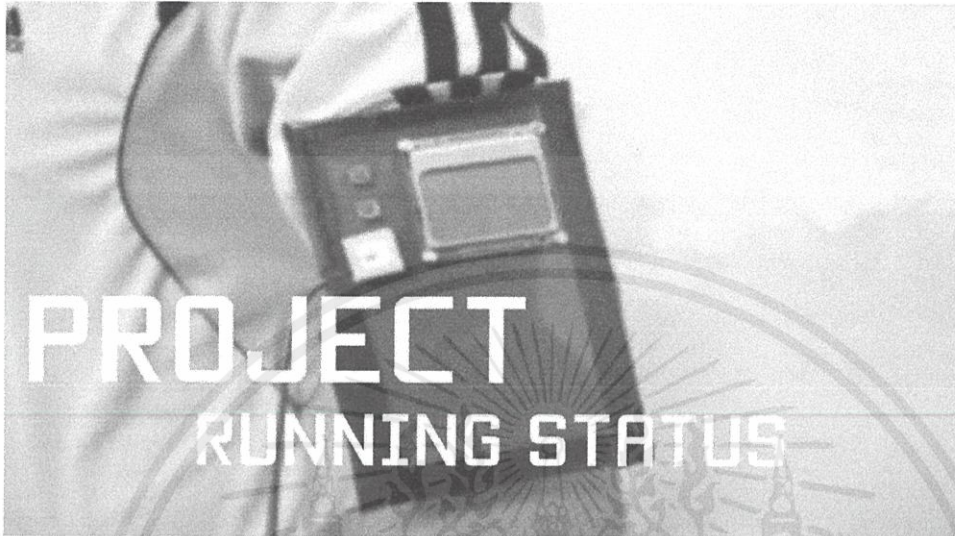
```
<html>
<link rel="stylesheet" type="text/css" href="test2.css">
<form action="editname.php" method="post">
    <font size="5">Name</font> : <input type="text"
name="Name">
    <input type = "submit" value="Update">
</form>
<form action="editage.php" method="post">
<font size="5">Age</font> : <input type="text" name="Age">
    <input type = "submit" value="Update">
</form>
<form action="editweight.php" method="post">
<font size="5">Weight</font> : <input type="text" name="Weight">
<input type = "submit" value="Update">
</form>
<form action="edithight.php" method="post">
<font size="5">Height</font> : <input type="text" name="Height">
<input type = "submit" value="Update">
</form>
<form action="editahr.php" method="post">
<font size="5">AverageHeartRate</font> : <input type="text"
name="ahr">
    <input type = "submit" value="Update">
</form>
    <form action="echo.php" method="post">
<input type = "submit" value="Back">
</form>
<br>
</html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข
คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



User Manual

Features

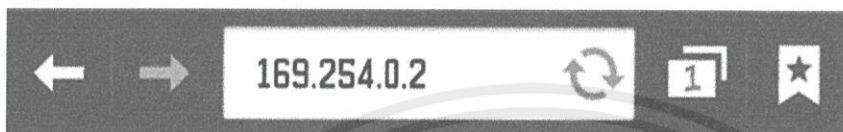
- วัดอัตราการเต้นของหัวใจ
- ตรวจสอบสถานะการวิ่ง
 - ความเร็วที่ใช้ในการวิ่ง
 - ระยะทางที่วิ่งได้
 - จำนวนแคลอรีที่เผาผลาญ
- เก็บข้อมูลสถานะการวิ่งในดาต้าเบส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

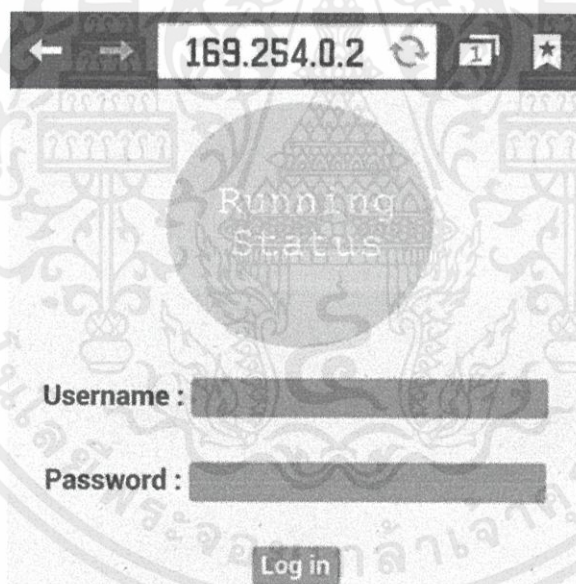
Set up

1.เมื่อเริ่มต้นใช้งานผู้ใช้จะต้องทำการกรอกข้อมูลส่วนตัวลงในระบบโดยการ ต่อสายแลน เข้ากับคอมพิวเตอร์หรือเราเตอร์ และต่อสายไฟและจ่ายไฟให้กับอุปกรณ์

2.เปิดเบราว์เซอร์ และพิมพ์ 169.254.0.2 บนช่อง URL กด Enter



3.กรอกชื่อผู้ใช้งาน และรหัสผ่าน จากนั้นกด Log in



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.หลังจาก Login เข้ามาแล้วให้เข้าไปกรอกข้อมูลส่วนตัว โดยกดปุ่ม Edit

Running Status

Personaldata

Name : User3
 Age : 22
 Weight : 72
 Height : 179
 AverageHeartRate : 76

Edit

Date	Speed	Distance	Heartrate	Calories
------	-------	----------	-----------	----------

5.กรอกข้อมูลชื่อ อายุ น้ำหนัก ส่วนสูง อัตราการเต้นหัวใจเฉลี่ย

Name : Update

Age : Update

Weight : Update

Height : Update

AverageHeartRate : Update

Back

6.ถอดสายแลน และอุปกรณ์จะพร้อมใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Get Start

1. สวมใส่อุปกรณ์

2. ต่อแหล่งจ่ายไฟเข้ากับอุปกรณ์ และเตรียมตัววิ่ง รอจนกว่า หน้าจอจะแสดงคำว่า “GPS standby Press Start”



3. กดปุ่ม On เพื่อเริ่มวิ่ง

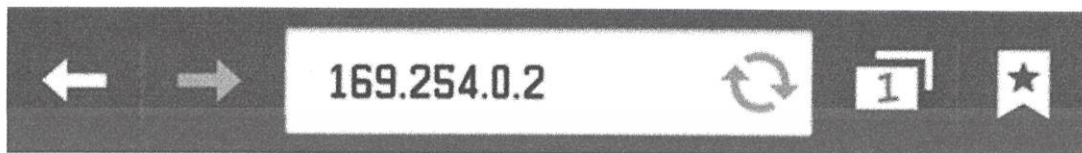
4. เมื่อวิ่งเสร็จกดปุ่ม OFF อุปกรณ์จะแสดง ข้อมูล ความเร็วเฉลี่ย ระยะทางที่วิ่งได้ อัตราการเต้นของหัวใจเฉลี่ย และแคลอรีที่เผาผลาญไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. หลังจากกดปุ่ม OFF อุปกรณ์จะทำการบันทึกข้อมูลสถานะการวิ่ง โดยสามารถดูข้อมูลย้อนหลังได้โดยต่อสายแลน เข้ากับคอมพิวเตอร์หรือเราเตอร์ และต่อสายไฟและจ่ายไฟให้กับอุปกรณ์

6. เปิดเบราว์เซอร์ และพิมพ์ 169.254.0.2 บนช่อง URL กด Enter



7. กรอกชื่อผู้ใช้งาน และรหัสผ่าน จากนั้นกด Log in

8. หน้าจอแสดงประวัติสถานะการวิ่ง จะแสดง วันเวลา ความเร็วเฉลี่ย อัตราการเต้นของหัวใจเฉลี่ย และแคลอรีที่เผาผลาญ

 A screenshot of a web application interface titled "Running Status". The interface features a large circular watermark in the background with Thai text and a central emblem. Below the title, there is a section for "Personaldata" with the following details: Name: user, Age: 22, Weight: 72, Height: 179, and Average Heart rate: 76. Below this is a table with 6 columns: Date, Speed(m/s), Distance(m), Heart rate(bpm), Calories(Kcal), and Duration(min). The table contains 8 rows of data. At the bottom of the interface, there are two buttons: "Clear Data" and "Back".

Date	Speed(m/s)	Distance(m)	Heart rate(bpm)	Calories(Kcal)	Duration(min)
2015-04-19 16:58:54	0.58	15.67	76.46	1.16	2:0
2015-04-19 17:19:26	0.67	58.84	78.44	2.42	1:4
2015-04-19 17:20:59	0.81	30.45	77.88	1.25	0:34
2015-04-19 17:28:40	0.78	45.16	71.86	1.80	0:40
2015-04-19 17:30:39	0.81	42.47	72.63	1.75	3:10
2015-04-20 18:38:15	2.67	927.65	81.53	38.18	5:44
2015-04-20 18:50:22	3.56	449.38	79.09	18.5	2:8
2015-04-20 18:52:58	1.21	484.52	91.24	19.94	6:14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้