ไม้เท้าวัดชีพจร

จัดทำโดย

นรจ.มูฮัมหมัด เหมโคกน้อย

นรจ.สุรพัศ หน่ายสุวรรณ

นรจ.สิรวิชญ์ วงศ์อนุ

นรจ.ธนากร สายมงคล

นรจ.ศราวุฒิ บุญศิริ

นรจ.ณัฐกิตติ์ อุทโท

โครงงานนี้เป็นส่วนหนึ่งของภาคการศึกษาตามหลักสูตรนักเรียนจ่าทหารเรือชั้นปีที่ 2

พรรคพิเศษ เหล่าทหารช่างยุทธโยธา อิเล็กทรอนิกส์ ปีการศึกษา 2564

โรงเรียนอิเล็กทรอนิกส์ กองวิทยาการ กรมอิเล็กทรอนิกส์ทหารเรือ

หัวข้อโครงงาน ไม้เท้าวัดชีพจร (pulse measuring crutch)

ผู้จัดทำ 1. นรจ.มูฮัมหมัด เหมโคกน้อย

2. นรจ.สุรพัศ หน่ายสุวรรณ

3. นรจ.สิรวิชญ์ วงศ์อนุ

4. นรจ.ธนากร สายมงคล

5. นรจ.ศราวุฒิ บุญศิริ

6. นรจ.ณัฐกิตติ์ อุทโท

ครูที่ปรึกษา 1. น.ต.สมเกียรติ สมมิตร

2. พ.จ.อ.พงศกร เชื้อเถาว์

3. พ.จ.อ.ธนากร พละศักดิ์

ปีการศึกษา 2564

บทคัดย่อ

โครงงานนี้สร้างไม้เท้าวัดชีพจรโดยมีวัตถุประสงค์ เพื่อลดความเสี่ยงในการเคลื่อนไหวของผู้สูงอายุ ขณะใช้งานไม้เท้าจะแสดงค่าชีพจรผ่านจอ OLED และจะส่งข้อมูลไปยังแอพพลิเคชั่น Line เมื่อชีพจรเกินค่าที่กำหนดหรือไม้เท้าล้ม

การดำเนินงานการสร้างไม้เท้าวัดชีพจร เริ่มจากการศึกษาข้อมูลเกี่ยวกับเซ็นเซอร์วัดชีพจร และเซ็นเซอร์วัดตามแกน การทดลองโปรแกรมกับไม้เท้าวัดชีพจร สามารถควบคุมการทำงานของเซ็นเซอร์ได้ตามค่าที่กำหนด

ผลการทดลองเมื่อไม้เท้าวัดชีพจรได้นอกย่าน 40 – 100 ครั้งต่อนาที หรือไม้เท้าล้มในมุมที่ต่ำกว่า 40 องศา จะส่งเสียงแจ้งเตือนผ่าน Buzzer ที่ตัวไม้เท้า และแสดงผลการเต้นของชีพจรผ่านจอ OLED พร้อมส่งข้อมูลแจ้งไปที่แอพพลิเคชั่น Line ของญาติ

น.ต. .....

ครูที่ปรึกษาโครงงาน

กิตติกรรมประกาศ

การทำโครงงานเรื่อง “ไม้เท้าวัดชีพจร” ส่งผลให้คณะผู้จัดทำได้รับความรู้และประสบการณ์ในด้านต่างๆ ไม่ว่าจะเป็นความรู้เกี่ยวกับหลักการทำงานของเครื่องวัดชีพจร การใช้งานโปรแกรม เช่น ภาษา c/c โครงงานดังกล่าว พร้อมทั้งรายงานผลความสำเร็จ ซึ่งโครงงานดังกล่าวนี้ได้รับความร่วมมือ คำปรึกษา ข้อแนะนำและการสนับสนุนจากบุคคลดังนี้

1. น.ต.สมเกียรติ สมมิตร 2. พ.จ.อ.พงศกร เชื้อเถาว์ และ 3. พ.จ.อ.ธนากร พละศักดิ์ ซึ่งท่านได้เป็นที่ปรึกษาโครงงานดังกล่าวนี้ ข้าพเจ้าคณะผู้จัดทำโครงงานทุกคน ขอขอบพระคุณ คุณครูที่ปรึกษาเป็นอย่างยิ่งที่ให้ความกรุณาแนะนำข้อมูลที่เป็นประโยชน์ต่อโครงงานนี้ และเป็นที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์ตลอดจนให้การดูแลและให้ความเข้าใจเกี่ยวกับการใช้งานโปรแกรมเป็นอย่างดีมาโดยตลอด ซึ่งข้าพเจ้าขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

คณะผู้จัดทำ

นรจ.มูฮัมหมัด	เหมโคกน้อย
นรจ.สุรพัศ	หน่ายสุวรรณ
นรจ.สิรวิชญ์	วงศ์อนุ
นรจ.ธนากร	สายมงคล
นรจ.ศราวุฒิ	บุญศิริ
นรจ.ณัฐกิตติ์	อุทโท

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
บทที่ 1	1
บทนำ	
1.1 ที่มาและความสำคัญของโครงงาน	
1.2 วัตถุประสงค์ของโครงงาน	
1.3 ขอบเขตการศึกษา	
1.4 สมมติฐาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	3
2.1 อัตราการเต้นของหัวใจในภาวะปกติ ควรเต้นกี่ครั้งต่อนาที	
2.2 บอร์ด ESP8266 NodeMCU	4
2.3 Buzzer	5
2.4 MAX30102	6
2.5 แบตเตอรี่	8
2.6 MMA8452	9
2.7 จอ OLED	10
บทที่ 3	11
3.1 Flowchart การทำงานของวงจร	
3.2 บล็อกไดอะแกรม	12
3.3 หลักการทำงาน	13
3.4 แผนการดำเนินงาน	14
3.5 ทดลองการทำงานของโปรแกรมคำสั่ง และออกแบบลายวงจร	15
3.6 ประกอบวงจรเข้ากล่อง	16
3.7 ประกอบกล่องเข้ากับไม้เท้า	17
บทที่ 4	18
4.1 ตรวจจับวัดชีพจรการเต้นของหัวใจ	
4.2 ตรวจวัดตามองศา	
4.3 การแจ้งเตือนผ่านทางแอพพลิเคชั่น Line	19
บทที่ 5	20
5.1 สรุปผลการทดลอง	
5.2 ปัญหา	

5.3 ข้อเสนอแนะ
ภาคผนวก 21
1. วัสดุอุปกรณ์ 22
2. เขียนโปรแกรมคำสั่งการทำงาน 23
3. โปรแกรม Arduino IDE 24
4. ภาษา C 25
5. รูปแบบของการออกแบบภาษา C
6. โปรแกรมควบคุมการทำงาน 26
บรรณานุกรม 33
ประวัติผู้จัดทำ 34

สารบัญรูป

เรื่อง	หน้า
รูปภาพที่ 2.1. ขาต่างๆของ ESP8266 NodeMCU V3	4
รูปภาพที่ 2.2 Buzzer	5
รูปภาพที่ 2.3 โมดูล MAX30102	6
รูปภาพที่ 2.4 แบตเตอรี่	8
รูปภาพที่ 2.5 โมดูล MMA8452	9
รูปภาพที่ 2.6 ขาต่างๆของ MMA8452	
รูปที่ภาพ 2.7 โมดูลจอแสดงผล OLED	10
รูปภาพที่ 3.1 การทำงาน Flowchart	11
รูปภาพที่ 3.2 บล็อกไดอะแกรม	12
รูปภาพที่ 3.3 หลักการทำงาน	13
รูปภาพที่ 3.4 ทดลองการทำงานของโปรแกรมคำสั่งและออกแบบลายวงจร	15
รูปภาพที่ 3.5 ประกอบวงจรเข้ากล่อง	16
รูปภาพที่ 3.6 ประกอบกล่องเข้ากับไม้เท้า	17
รูปภาพที่ 4.1 การแจ้งเตือนผ่านทางแอพพลิเคชั่น Line	19

สารบัญตาราง

เรื่อง	หน้า
ตารางที่ 3.1 แผนการดำเนินงาน	14
ตารางที่ 4.1 ตรวจจับวัดชีพจรการเต้นของหัวใจ	18
ตารางที่ 4.2 ตรวจวัดตามองศา	

S

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ปัจจุบันสังคมไทยมีจำนวนผู้สูงอายุเพิ่มมากขึ้นและครอบครัวส่วนใหญ่เป็นครอบครัวเดี่ยว บางครั้งได้ปล่อยผู้สูงอายุอยู่บ้านตามลำพัง ทำให้ผู้สูงอายุไม่ได้รับการดูแลเท่าที่ควร ด้วยผู้สูงอายุที่จำนวนเพิ่มมากขึ้น ร่างกายเกิดความเสื่อมโทรมตามอายุทำให้ผู้สูงอายุหกล้มได้ง่ายและขณะที่หกล้ม หรืออุบัติเหตุอาจไม่มีผู้พบเห็นและเข้าช่วยเหลือได้ทันท่วงที จนทำให้ผู้สูงอายุได้รับบาดเจ็บร้ายแรงถึงชีวิต ดังนั้นหากมีเครื่องมือที่คอยเตือนผู้ใช้ เพื่อให้ทราบว่าควรหยุดพักกิจกรรมหรือเตือนเมื่อหกล้ม เช่น หากผู้สูงอายุหรือผู้ใช้งานหกล้มก็จะมีการส่งสัญญาณแจ้งเตือนไปยังบุตรหลานหรือผู้ที่อยู่ใกล้เคียงเพื่อเข้าช่วยเหลือผู้สูงอายุได้ทันท่วงที

ดังนั้นไม้เท้าวัดชีพจรจึงเป็นอีกหนึ่งทางเลือกที่สามารถช่วยเหลือผู้สูงอายุที่ประสบอุบัติเหตุหรือหกล้มได้ ซึ่งไม้เท้านอกจากจากทำหน้าที่พยุงไม่ให้ล้มแล้ว หากเราทำการติดตั้ง MMA8452Q ไว้ที่ไม้เท้า ถ้าหากไม้เท้าเกิดการล้มก็จะส่งเสียงเรียกผู้ที่อยู่ใกล้เคียงให้ได้รับทราบได้ทันเวลา พร้อมส่งสัญญาณแจ้งเตือนไปยัง Smart phone ของบุตรหลานในคราวเดียวกัน สมมติฐานนี้หากสามารถทำได้สำเร็จจะเป็นประโยชน์กับสังคมผู้สูงอายุอย่างยิ่ง ซึ่งอุบัติเหตุที่สร้างการบาดเจ็บรุนแรงให้กับผู้สูงอายุ มีผลมาจากการที่ผู้สูงอายุไม่ทราบว่าตนเองมีอาการผิดปกติหรือไม่ ซึ่งอาจเกิดจากการเดินหรือเคลื่อนไหวร่างกายโดยที่ไม่ได้ระวังว่าจะเกิดผลที่เป็นภาระแก่ร่างกาย

1.2 วัตถุประสงค์ของโครงงาน

เพื่อสร้างไม้เท้าแจ้งเตือนผ่านระบบไลน์เมื่อชีพจรช้าหรือเร็วกว่ากำหนด หรือผู้ใช้ไม้เท้าหกล้ม

1.3 ขอบเขตโครงงาน

1.3.1 ตรวจวัดชีพจรโดยการสัมผัสด้วยมือ

1.3.2 แจ้งเตือนการเต้นของหัวใจที่เกินค่ากำหนดด้วยเสียง

1.3.3 ส่งการแจ้งเตือนการล้มไปที่ Smart phone

1.3.4 ติดตั้ง sensor ตรวจสอบการล้มของไม้เท้า

1.3.5 แสดงผลการวัดชีพจรไปที่จอ OLED

1.4 สมมติฐาน

1.4.1 จากปัญหาข้างต้นควรมีระบบแจ้งเตือนเมื่อชีพจรผิดปกติ และเมื่อไม้เท้าเกิดการล้ม โดยติดตั้งระบบเข้ากับไม้เท้า เพื่อที่จะทำให้สะดวกต่อการใช้งาน ไม้เท้าที่สร้างขึ้นนี้จะตรวจจับการเต้นของชีพจรและการล้มของไม้เท้าเพื่อให้ทราบว่าชีพจรปกติหรือผิดปกติ และถ้าหากไม้เท้ามีการล้มก็จะทำการแจ้งเตือนไปที่ผู้อยู่ใกล้เคียงโดยเสียงหรือแจ้งเตือนไปยังลูกหลานผ่านโทรศัพท์มือถือ

1.4.2 ถ้าจังหวะการเต้นของหัวใจผิดปกติจะแจ้งเตือนผู้สูงอายุ ให้หยุดพักการเคลื่อนไหวร่างกายเพื่อให้ชีพจรกลับสู่ภาวะปกติ ซึ่งจะช่วยลดการสูญเสียและการเจ็บป่วยที่รุนแรงได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้ไม้เท้าที่สามารถวัดสัญญาณชีพจร

1.5.2 ได้เรียนรู้การต่อวงจรอุปกรณ์ sensor ต่างๆ

1.5.3 ช่วยลดโอกาสเกิดอุบัติเหตุรุนแรงให้กับผู้สูงอายุ

1.5.4 ได้เรียนรู้หลักการทำงานเกี่ยวกับตัวอุปกรณ์อิเล็กทรอนิกส์

1.5.5 เป็นเครื่องต้นแบบที่หน่วยงานอื่นสามารถนำไปใช้งานและต่อยอดได้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

อัตราการเต้นของหัวใจ (Heart Rate) ของคนปกติ จะมีอัตราเปลี่ยนแปลงไปตามการกระตุ้นของระบบประสาทอัตโนมัติ (Autonomic Nervous System) เช่น เต้นเร็วขึ้นขณะที่มีอาการตกใจ ดีใจ และเต้นช้าลงขณะนอนหลับหรือได้รับยาลดความดันโลหิตบางตัวมีผลข้างเคียงที่เป็นสาเหตุที่ทำให้หัวใจเต้นช้าลงอัตราการเต้นของหัวใจจะเกิดจากการบีบตัวของหัวใจ

ซึ่งจำเป็นต้องอาศัยกระแสไฟฟ้าในการกระตุ้นกล้ามเนื้อหัวใจแต่ละครั้ง โดยกระแสไฟฟ้านี้จะมีจุดกำเนิดมาจากเนื้อเยื่อพิเศษของหัวใจห้องบนขวา (Right Atrium) สร้างกระแสไฟฟ้าแล้วปล่อยลงมาตามทางเนื้อเยื่อพิเศษที่สามารถนำไฟฟ้าได้ และลงมายังสถานีที่อยู่ระหว่างหัวใจห้องบนกับห้องล่างเรียกว่า AV node ก่อนที่จะส่งต่อลงไปยังหัวใจห้องล่างแล้วแยกออกไปยังแขนด้านซ้ายและขวา เมื่อกระแสไฟฟ้าเดินทางมาถึงจุดหมายจะกระตุ้นให้กล้ามเนื้อหัวใจบีบตัวและกระแสไฟฟ้านี้จะสามารถตรวจจับได้ด้วยเครื่องตรวจคลื่นไฟฟ้าหัวใจ หรือที่เรียกว่า การตรวจ EKG

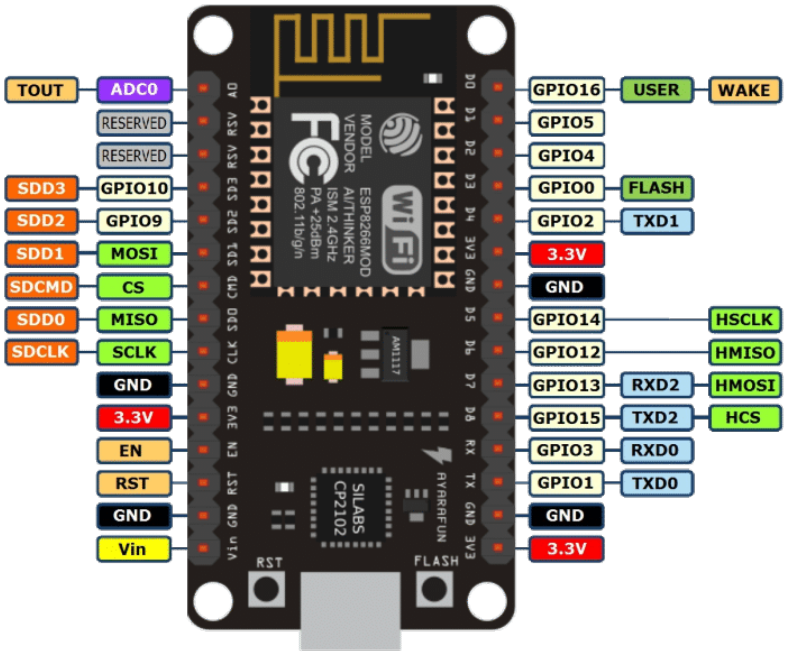
2.1 อัตราการเต้นของหัวใจในภาวะปกติ ควรเต้นกี่ครั้งต่อนาที

ในภาวะร่างกายปกติจะมีอัตราการเต้นของหัวใจอยู่ในช่วง 60-100 ครั้ง/นาที ซึ่งอัตราการเต้นของหัวใจจะช้าหรือเร็วจะขึ้นกับปัจจัยอื่นร่วมด้วยอาทิ กิจกรรมและความต้องการออกซิเจนของร่างกาย เช่น ขณะนอนหลับอัตราการเต้นของหัวใจจะอยู่ที่ประมาณ 40-60 ครั้ง/นาที ขณะพักอัตราการเต้นของหัวใจจะอยู่ที่ประมาณ 60-80ครั้ง/นาที ขณะเดินอัตราการเต้นของหัวใจจะอยู่ที่ประมาณ 60-80 ครั้ง/นาที และขณะวิ่งอัตราการเต้นของหัวใจจะอยู่ที่ประมาณ 100 ครั้ง/นาที

2.2 บอร์ด ESP8266 NodeMCU V3

ผู้สร้างชิพ ESP คือTeo Swee Ann ชาวสิงคโปร์ โดยในโมดูลประกอบด้วย chip Microcontroller + WiFi Module ตัวบอร์ด ESP8266 NodeMCU V3 สามารถเขียนโปรแกรม ใช้งานแทนไมโครคอนโทรลเลอร์ได้ โดยพื้นที่สำหรับเขียนโปรแกรม 4 MB

ESP8266 ทำงานที่แรงดันไฟฟ้า 3.3V - 3.6V การนำไปใช้งานร่วมกับเซ็นเซอร์อื่นๆ ที่ใช้แรงดัน 5V ต้องใช้วงจรแบ่งแรงดันมาช่วยเพื่อไม่ให้โมดูลพังเสียหาย กระแสที่โมดูลใช้งานสูงสุดคือ 200mA ความถี่คริสตอล 40MHz เมื่อนำไปใช้งานอุปกรณ์ที่ทำงานรวดเร็วตามความถี่ เช่น LCD ทำให้การแสดงผลข้อมูลรวดเร็วกว่าไมโครคอนโทรลเลอร์ยอดนิยม Arduino



รูปภาพที่ 2.1 ขาต่างๆของ ESP8266 NodeMCU V3

http://pichadiy.lnwshop.com/product/7/esp8266-v3

2.3 Buzzer

Buzzer คือ ลำโพงแบบแม่เหล็กหรือแบบเปียโซที่มีวงจรกำเนิดความถี่ (oscillator) อยู่ภายในตัว ซึ่งใช้ไฟเลี้ยง 3.3 - 5V สามารถสร้างเสียงเตือนหรือส่งสัญญาณที่เป็นรูปแบบต่างๆ Buzzer เป็นอุปกรณ์ที่ให้กำเนิดเสียงทำหน้าที่เปลี่ยนสัญญาณไฟฟ้าให้อยู่ในรูปสัญญาณเสียง
Buzzer มีอยู่ 2 ประเภท ได้แก่

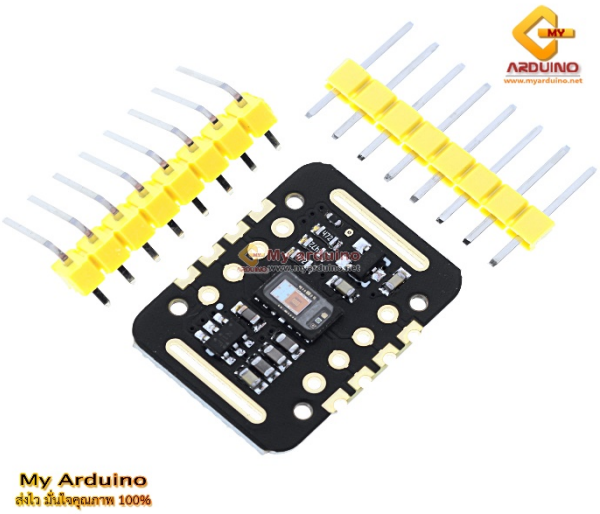
2.2.1 แบบแอคทีฟ (Active) ลำโพงชนิดนี้มีวงจรกำเนิดความถี่อยู่ภายใน สามารถสร้างสัญญาณเสียงเตือนได้ทันทีเพียงแค่จ่ายแรงดันไฟฟ้าเข้าไปใน Buzzer

2.2.2 แบบพาสซีฟ (Passive) ลำโพงชนิดนี้ทำงานเหมือนลำโพงขนาดเล็ก คือ ถ้าป้อนแรงดันไฟฟ้ากระแสตรงเข้าไปจะไม่มีเสียงถ้าต้องการให้มีสัญญาณเสียงต้องทำการป้อนสัญญาณความถี่ ลำโพงชนิดนี้สามารถกำเนิดเสียงที่มีความแตกต่างกันตามความถี่ที่ป้อนเข้ามาเราอาจจะเคยได้ยินเสียง Buzzer อยู่บ่อยๆ เช่น เสียงเตือนที่อยู่ในคอมพิวเตอร์ก็ใช้ Buzzer ในการส่งสัญญาณให้ทราบสถานะของคอมพิวเตอร์ให้ทราบว่ามีปัญหา

รูปภาพที่ 2.2 Buzzer

https://www.modulemore.com/product/474

2.4 MAX30102 เซ็นเซอร์วัดออกซิเจนในเลือดและอัตราการเต้นของหัวใจ



รูปภาพที่ 2.3 โมดูล MAX30102

https://www.cybertice.com/product/2745/max30102

คุณสมบัติ

MAX30102 เป็นเครื่องวัดออกซิเจนในเลือดแบบบูรณาการและโมดูลไบโอเซนเซอร์ตรวจวัดอัตราการเต้นของหัวใจ รวม LED สีแดงและ LED อินฟราเรด เครื่องตรวจจับแสง อุปกรณ์ออปติคัล และวงจรอิเล็กทรอนิกส์สัญญาณรบกวนต่ำพร้อมการลดแสงโดยรอบ MAX30102 ส่วนอุปกรณ์วัดอัตราการเต้นของหัวใจและเก็บออกซิเจนในเลือด อินเทอร์เฟซการสื่อสารที่เข้ากันได้กับ I2C มาตรฐานสามารถส่งค่าที่รวบรวมไว้ไปยัง Arduino, KL25Z และไมโครคอนโทรลเลอร์อื่นๆ สำหรับการคำนวณอัตราการเต้นของหัวใจและออกซิเจนในเลือด

นอกจากนี้ชิพยังสามารถปิดโมดูลผ่านซอฟต์แวร์ กระแสไฟขณะสแตนด์บายใกล้ศูนย์และแหล่งจ่ายไฟจะยังคงอยู่ เนื่องจากประสิทธิภาพที่ยอดเยี่ยม ชิพนี้จึงถูกใช้อย่างแพร่หลายในโทรศัพท์มือถือ Samsung Galaxy S series ชิพรวมฝาครอบแก้วเพื่อกำจัดการรบกวนของแสงภายนอกและภายในอย่างมีประสิทธิภาพและให้ประสิทธิภาพที่เชื่อถือได้ดีที่สุด

พารามิเตอร์ผลิตภัณฑ์

- ความยาวคลื่นสูงสุดของ LED : 660nm/880nm
- แรงดันไฟ LED : 3.3 ~ 5V
- ประเภทสัญญาณตรวจจับ : สัญญาณสะท้อนแสง (PPG)
- อินเทอร์เฟซสัญญาณเอาต์พุต : อินเทอร์เฟซ I2C
- แรงดันไฟฟ้าอินเทอร์เฟซการสื่อสาร : 1.8 ~ 3.3V ~ 5V (อุปกรณ์เสริม
- ขนาดรูประกอบบอร์ดสำรอง : 0.5X8.5mm

หลักการทำงาน

- วิธีการละลายด้วยแสง : วัดชีพจรและความอิ่มตัวของออกซิเจนในเลือดโดยใช้เนื้อเยื่อของมนุษย์ทำให้เกิดการส่องผ่านของแสงที่แตกต่างกัน เมื่อหลอดเลือดมีการเต้น

- แหล่งกำเนิดแสง : หลอดขั้วเรืองแสงที่มีความยาวคลื่นเฉพาะที่เลือกไว้สำหรับออกซีเฮโมโกลบิน (HbO2 และฮีโมโกลบิน (Hb) ในเลือดแดง

- การส่งผ่านจะถูกแปลงเป็นสัญญาณไฟฟ้า : ปริมาตรของการเต้นของหลอดเลือดแดงจะเปลี่ยนเป็นการส่งผ่านแสงในเวลาที่แสงสะท้อนจากเนื้อเยื่อของร่างกายมนุษย์โดยการแปลงตามแนว เป็นสัญญาณไฟฟ้าขยายและส่งออก

2.5 แบตเตอรี่

แบตเตอรี่เป็นอุปกรณ์เปลี่ยนพลังงานเคมีเป็นพลังงานไฟฟ้าที่มีพัฒนาการมาตั้งแต่ศตวรรษที่18 จนถึงปัจจุบัน เราสามารถแบ่งถ่านไฟฉายและแบตเตอรี่ได้เป็น 2 ประเภทคือ

2.5.1 เซลล์ปฐมภูมิ (primary cell) ซึ่งเป็นถ่านไฟฉายหรือแบตเตอรี่ชนิดใช้แล้วทิ้งไม่สามารถประจุไฟซ้ำได้ เช่น ถ่านไฟฉายธรรมดา ถ่านแอลคาไลน์ ถ่านนาฬิกา เป็นต้น

2.5.2 เซลล์ทุติยภูมิ (secondary cell) ซึ่งเป็นถ่านหรือแบตเตอรี่ที่สามารถนำมาอัดประจุไฟซ้ำได้ เช่น แบตเตอรี่รถยนต์ แบตเตอรี่มือถือ ถ่านไฟฉายแบบประจุไฟใหม่ได้ (rechargeable battery) เป็นต้น

รูปภาพที่ 2.4 แบตเตอรี่

https://www.officemate.co.th/th

สำหรับคำว่า 9V นั้น ย่อมาจาก 9 Volts นั่นก็คือ แรงดันไฟฟ้าขนาด 9 โวลต์ แต่สำหรับความหมายของ 9V ในที่นี้นั้น ถูกใช้ในการเรียกแทนขนาดของแบตเตอรี่ เช่นเดียวกับการเรียกขนาดของถ่านอื่นๆ เช่น AA, AAA, C, SC, D, 18650, 14500

ถ่าน 9V เป็นแบตเตอรี่ทรงสี่เหลี่ยมฝืนผ้า ขนาด 48.5 × 26.5 × 17.5 mm หรือขนาด PP3 ตามมาตรฐานของ The American National Standards Institute (ANSI) แต่ในความเป็นจริงนั้น ถ่าน 9V มีอยู่หลายขนาดตั้งแต่ PP1 ถึงขนาด PP11 แต่โดยทั่วไปที่ใช้งานกันอยู่คือ ขนาด PP3 จึงเป็นที่เข้าใจกันว่า ถ่าน 9V นั้นหมายถึงขนาด PP3 นั่นเอง

เนื่องจากถ่าน 9V นั้นเรียกตามขนาดของแบตเตอรี่ แรงดันไฟของแบตเตอรี่ประเภทนี้จึงไม่ได้แปลว่ามีแรงดัน 9V เสมอ ทั้งนี้ขึ้นอยู่กับชนิดของแบตเตอรี่ที่นำมาประกอบเป็นถ่าน 9V

2.6 MMA8452Q

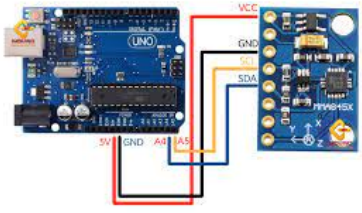
MMA8452Q เป็นเครื่องวัดความเร่งแบบไมโครเมคานิคัลแบบสามแกนที่ใช้พลังงานต่ำอย่างชาญฉลาด พร้อมความละเอียด 12 บิต ปริมาตรความเร่งนี้มีชุดคุณสมบัติที่ฝังไว้มากมายพร้อมตัวเลือกที่ผู้ใช้สามารถตั้งโปรแกรมได้ยืดหยุ่นสำหรับพินอินเทอร์รัปต์สูงสุดสองพิน ฟังก์ชันขัดจังหวะที่ฝังไว้ช่วยประหยัดพลังงานโดยรวมและบรรเทาโปรเซสเซอร์โฮสต์จากการสำรวจข้อมูลอย่างต่อเนื่อง

คุณสมบัติ MMA8452Qช่วงที่ผู้ใช้เลือกได้ ±2g/±4g/±8g สำหรับเอาต์พุตแบบเรียลไทม์ของข้อมูลที่กรองผ่านความถี่สูงและข้อมูลที่ไม่มีการกรอง อุปกรณ์สามารถกำหนดค่าให้สร้างสัญญาณขัดจังหวะการปลุกด้วยการผสมผสานของฟังก์ชันฝังตัวที่กำหนดค่าได้ ซึ่งช่วยให้ MMA8452Q สามารถคงโหมดพลังงานต่ำไว้ได้ในขณะหยุดนิ่งขณะตรวจสอบเหตุการณ์

รูปภาพที่ 2.5 โมดูล MMA8452

http://th.sz-kuongshun.com/uno/uno-sensor/gy-45-mma8452-3-axis-accelerator.html

- แอปพลิเคชั่นเข็มทิศอิเล็กทรอนิกส์ MMA8452
- การตรวจจับทิศทางนิ่ง (แนวนอน/แนวตั้ง, ขึ้น/ลง, ซ้าย/ขวา, การจดจำตำแหน่งด้านหน้า/ด้านหลัง
- การตรวจจับทิศทางแบบเรียลไทม์ (ความจริงเสมือนและการตอบสนองตำแหน่งผู้ใช้ 3D ของคอนโซลเกม



รูปภาพที่ 2.6 ขาต่างๆของ MMA8452

https://www.cybertice.com/product/3021/gy-45-mma8452

2.7 จอ OLED

รูปภาพที่ 2.7 โมดูลจอแสดงผล OLED V2

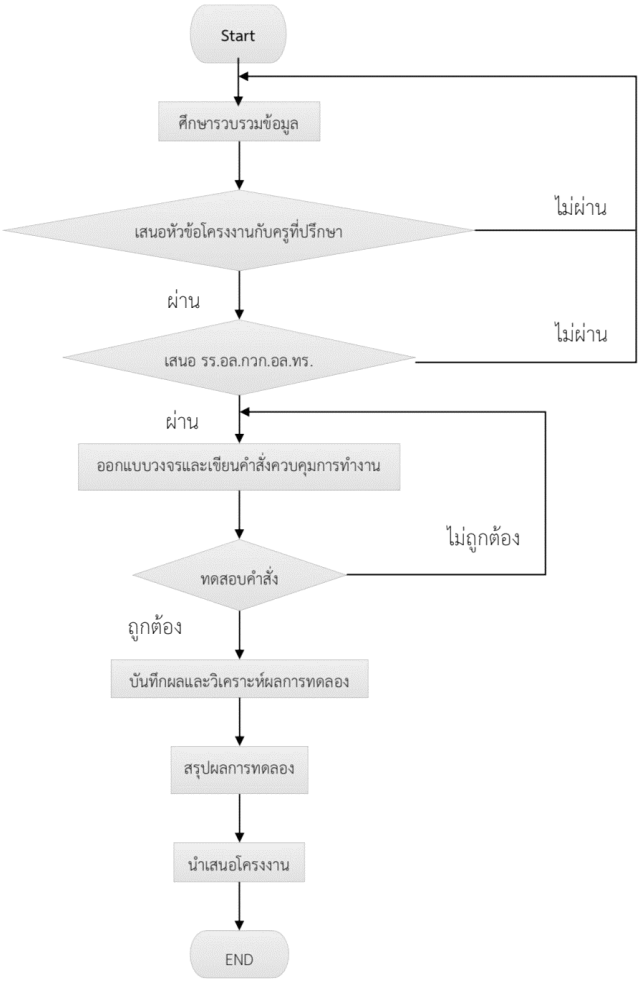
https://www.arduitronics.com/product/3113/จอแสดงผล-oled

- ไม่จำเป็นต้องมีไฟBlacklight หน่วยแสดงผลสามารถส่องสว่างได้เอง
- ความละเอียดสูง: 128*64
- มุมมอง:> 160 °
- รองรับchipควบคุมมากมาย: เข้ากันได้กับ Arduino, 51 Series, MSP430 Series, STM32 / 2, CSR IC เป็นต้น
- ใช้พลังงานต่ำมาก: สว่างเต็มหน้าจอ 0.08W
- แรงดันไฟฟ้า: 3V ~ 5V DC
- อุณหภูมิในการทำงาน: -30 ~ 70
- ขนาดโมดูล: 27.0MM * 27.0MM * 4.1MM
- ไดรเวอร์ IC: SSD1306
- สีฟ้า
- ที่อยู่ I2C 0x3C

บทที่ 3

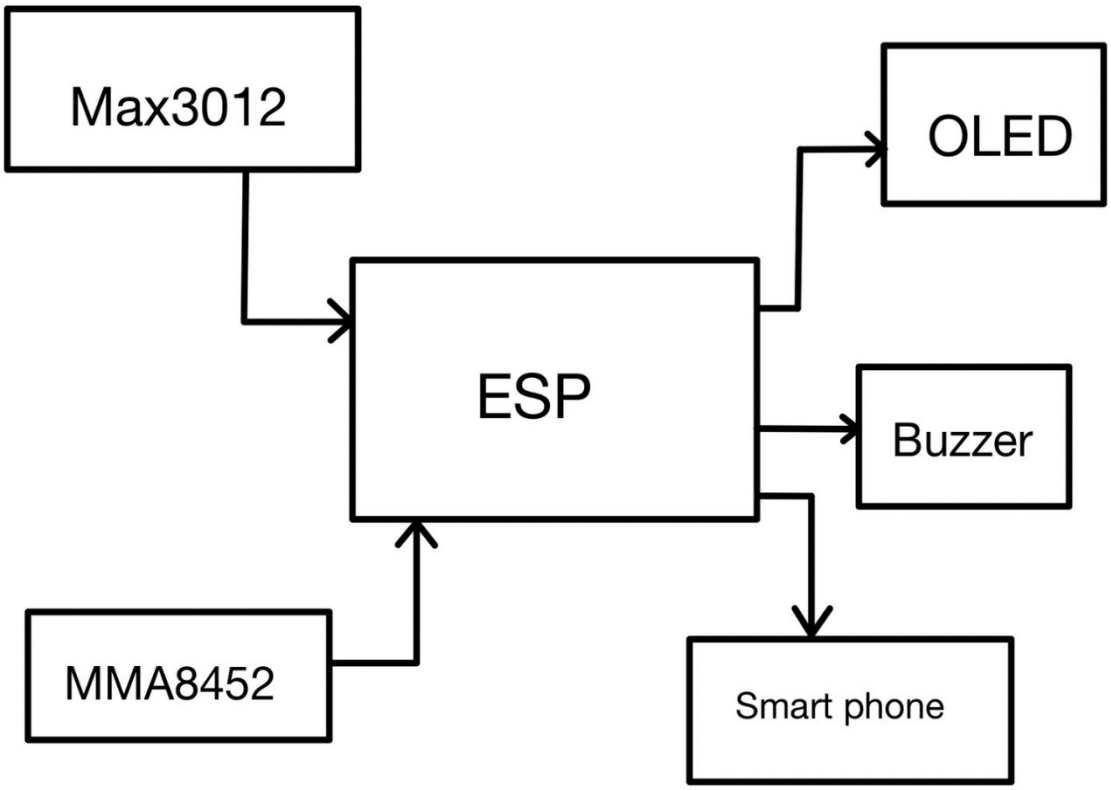
วิธีการดำเนินงาน

3.1 การทำงาน Flowchart



รูปภาพที่ 3.1 การทำงาน Flowchart

3.2 บล็อกไดอะแกรม

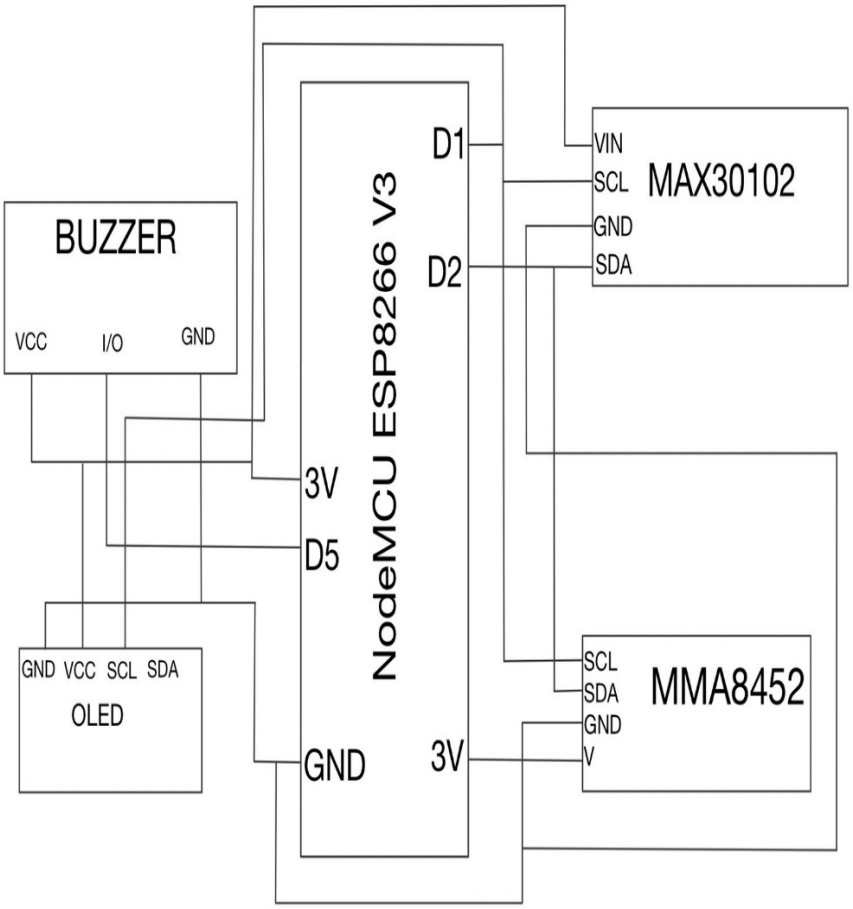


รูปภาพที่ 3.2 บล็อกไดอะแกรม

บล็อกไดอะแกรมของระบบ เมื่อเปิดสวิตช์แล้วจะเริ่มการทำงานของ ESP8266 โดยที่เมื่อเริ่มการทำงาน max30102 จะมีไฟสีแดงขึ้น หลังจากนั้นถ้าเรานำมือไปสัมผัสที่ max30102 จะทำการวัดอัตราการเต้นของหัวใจโดยส่งผลการวัดที่ได้ไปเก็บไว้ที่ ESP8266 และส่งออกไปทางจอ OLED ถ้าหากว่าเกินค่าที่กำหนดไว้ ก็จะเกิดเสียงดังขึ้นออกทาง BUZZER

ส่วน MMA8452 เมื่อไม้เท้าทำมุมตั้งแต่ 40องศาขึ้นไปก็จะไม่มีเสียงออกทาง BUZZER แต่ถ้าเกิดไม้เท้าทำมุมตั้งแต่ 40 องศาลงไป ก็จะเกิดเสียงออกจากตัว BUZZER แล้วเมื่อเราทำการเชื่อมต่อ WI-FI เข้าไปที่ ESP8266 การแจ้งเตือนก็จะถูกส่งไปยังแอพพลิเคชั่น Line ที่เราได้กำหนดไว้ทั้งการวัดชีพจรและการล้มของไม้เท้า

3.3 หลักการทำงาน



รูปภาพที่ 3.3 หลักการทำงาน

จากบอร์ดจะใช้ Pin D1และD2 เป็นขากำหนดการส่งข้อมูลเข้าไปที่ตัวบอร์ด เมื่อบอร์ดได้รับข้อมูลจาก MAX30102 โดยสัญญาณที่ได้รับจะเป็นสัญญาณ Digital บอร์ดจะทำการประมวลผลแล้วส่งผลไปที่จอ OLED ส่วนค่าข้อมูลของ MMA8452 ก็จะถูกประมวลผลและเก็บข้อมูลไว้ในบอร์ด และเมื่อค่าของ MAX30102 มากกว่า 100 ครั้ง/นาที หรือไม้เท้าทำมุมน้อยกว่า 40 องศา Buzzer จะมีการแจ้งเตือน

3.4 แผนการดำเนินงาน

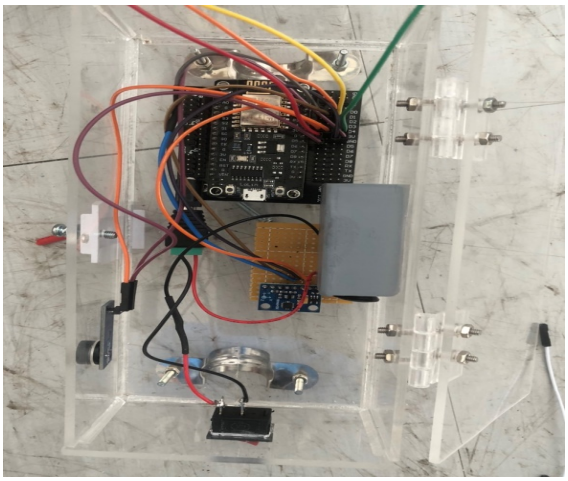
ขั้นตอนการดำเนินงาน	ระยะเวลาการดำเนินงาน				
	2564		2565		
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. เสนอหัวข้อโครงงาน	◄	►			
2. ศึกษาระบบ		◄	►		
3. ศึกษาการใช้งานโปรมแกรม		◄	►		
4. ออกแบบระบบและจัดซื้อ			◄	►	
5. ทดลองระบบ			◄	►	
6. แก้ไขปรับปรุงโครงงาน				◄	►
7. จัดทำเอกสาร	◄				►
8. ส่งรูปเล่มโครงงาน				◄	►
9. นำเสนอโครงงาน					◄►

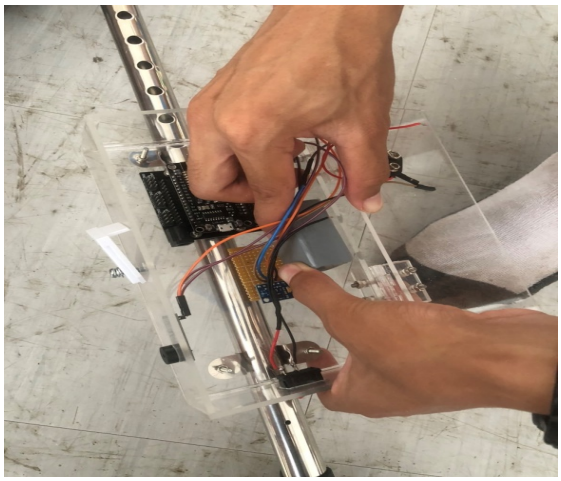
ตารางที่ 3.1 แผนการดำเนินงาน

3.5 ทดลองการทำงานของโปรแกรมคำสั่งและออกแบบลายวงจร

รูปภาพที่ 3.4 ทดลองการทำงานของโปรแกรมคำสั่งและออกแบบลายวงจร

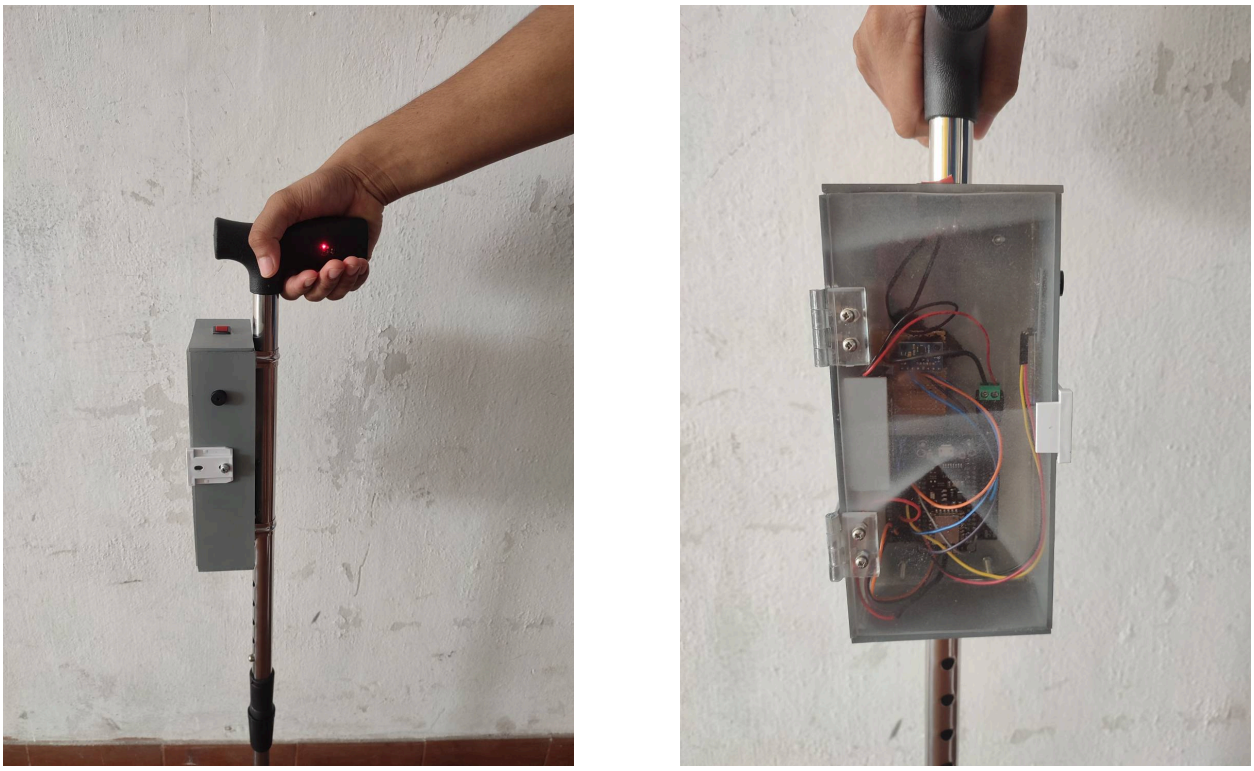
3.6 ประกอบอุปกรณ์เข้ากล่อง





รูปภาพที่ 3.5 ประกอบอุปกรณ์เข้ากล่อง

3.7 ประกอบกล่องเข้ากับไม้เท้า



รูปภาพที่ 3.6 ประกอบกล่องเข้ากับไม้เท้า

บทที่ 4

ผลการดำเนินงาน

ตาราง 4.1 ตรวจวัดอัตราการเต้นของชีพจร

ชีพจร ครั้งต่อนาที	การแจ้งเตือน Line	Buzzer
น้อยกว่า 40	แสดงการแจ้งเตือน	on
40 - 60	ไม่แสดงการแจ้งเตือน	off
61 - 80	ไม่แสดงการแจ้งเตือน	off
81 - 100	ไม่แสดงการแจ้งเตือน	off
มากกว่า 100	แสดงการแจ้งเตือน	on

เมื่อตรวจวัดชีพจร หากอัตราการเต้นของชีพจรแสดง 40 – 100 ครั้งต่อนาที จะไม่มีการแจ้งเตือนผ่านทาง แอพพลิเคชั่น Line ถ้าหากค่าอัตราการเต้นของชีพจรมากกว่า 100 ครั้งต่อนาที หรือค่าอัตราการเต้นของชีพจรน้อยกว่า 40 ครั้งต่อนาที จะมีการแจ้งเตือนผ่านทางแอพพลิเคชั่น Line และมีแจ้งเตือนเสียงดังขึ้นในระบบ Buzzer การเต้นของชีพจรจะแสดผลผ่านจอ OLED

ตาราง 4.2 ตรวจวัดตามองศา

องศา	การแจ้งเตือน Line	Buzzer
น้อยกว่า 40 องศา	แสดงการแจ้งเตือน	on
มากกว่า 40 องศา	ไม่แสดงการแจ้งเตือน	off

กรณีเมื่อไม้เท้าวัดชีพจรทำมุมตั้งแต่ 40 องศา ลงไปจะมีเสียงแจ้งเตือนออกทาง Buzzer และจะมีการแจ้งเตือนผ่านทางแอพพลิเคชั่น Line หากไม้เท้าวัดชีพจรทำมุมตั้งแต่ 40 องศา ขึ้นไปจะไม่มีเสียงแจ้งเตือนออกทาง Buzzer และจะไม่มีการแจ้งเตือนผ่านทางแอพพลิเคชั่น Line

4.3. การแจ้งเตือนผ่านทางแอพพลิเคชั่น Line

รูปภาพที่ 4.1 การแจ้งเตือนผ่านทางแอพพลิเคชั่น Line

เมื่อตรวจวัดอัตราการเต้นของชีพจรมากกว่า 100 ครั้งต่อนาที หรือ น้อยกว่า 40 ครั้งต่อนาที จะมีการแจ้งเตือนผ่านทาง แอพพลิเคชั่น Line หรือไม้เท้าวัดชีพจรทำมุมตั้งแต่ 40 องศาลงไป จะมีการแจ้งเตือนผ่านทางแอพพลิเคชั่น Line

บทที่ 5

สรุป ปัญหาและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากผลการทดลองจะสรุปได้ว่า ชีพจรของคนเราจะมีอัตราการเต้นที่แตกต่างกันไปตามการกระทำแต่ละชนิดถ้าเรายิ่งขยับร่างกายมากขึ้นชีพจรก็จะเต้นเร็วขึ้น ส่วนการล้มของไม้เท้าก็จะมีเสียงดังขึ้นถ้าหากมีการล้มหรือการเอียงที่น้อยกว่า 40 องศา

เมื่อ ESP8266 มีการเชื่อมต่อไวไฟ ก็จะมีการแจ้งเตือนไปยังแอพพลิเคชั่น Line หากไม้เท้าทำมุมน้อยกว่า 40 องศา แต่ถ้ามากกว่า 40 องศา จะไม่มีการแจ้งเตือนผ่านทางไปยังแอพพลิเคชั่น Line

5.2 ปัญหา

5.2.1 Sensor วัดชีพจรที่จัดหาได้ เป็นอุปกรณ์เพื่อการศึกษาจึงยังให้ผลการวัดที่ไม่เสถียรดีเท่าเทียมเครื่องมือแพทย์ที่ได้มาตรฐาน

5.2.2 วัสดุอุปกรณ์ที่นำมาประกอบกันเป็นระบบเครื่องอิเล็กทรอนิกส์แบบแสวงเครื่อง มิได้ผลิตมาเป็นการเฉพาะงานสำหรับติดตั้งบนไม้เท้า ทำให้ขนาดและความแข็งแรง อาจยังไม่เหมาะสมเท่าที่ควร

5.3 ข้อเสนอแนะ

5.3.1 ควรพัฒนาวิธีใช้แบตเตอรี่แบบใหม่ที่มีความจุมากกว่านี้

5.3.2 ควรเปลี่ยนรูปแบบกล่องให้เหมาะสมและแข็งแรงทนทานมากขึ้น

5.3.3 ควรติดตั้ง GPS เพื่อให้ญาติสามารถทราบตำแหน่งของผู้ใช้งานไม้เท้า

5.3.4 ปรับองศาการล้มของไม้เท้าให้คอบคุมการใช้งานในสถานที่และสถานการณ์ต่างๆ

ภาคผนวก

ภาคผนวก

1. วัสดุอุปกรณ์

ลำดับ	รายการ	จำนวน	ราคา (บาท)	ราคา (รวม)	รูปประกอบ
1	ESP8266 NodeMCU V3	1	350	350	
2	Buzzer	1	19	19	
3	MAX30102	1	90	90	
4	ถ่านไฟฉาย	2	50	100	
5	ไม้เท้า	1	189	189	
6	กล่อง	1	70	70	
7	MMA8452	1	70	70	

8	OLED	1	140	140	
9	NodeMCU V3 Base	1	150	150	
ลำดับ	รายการ	จำนวน	ราคา (บาท)	ราคา (รวม)	รูปประกอบ
10	สายไฟจั้มเปอร์	1	75	75	
11	สวิตซ์ไฟ	1	25	25	

งบประมาณที่ใช้ในการทำโครงงานประดิษฐ์
เป็นเงินทั้งสิ้น .....1,278..... (หนึ่งพันสองร้อยเจ็ดสิบแปดบาทถ้วน)

2. เขียนโปรแกรมคำสั่งการทำงาน

3. โปรแกรม Arduino IDE

โครงสร้างของการเขียนโปรแกรมแบ่งออกเป็นส่วนย่อย หลายๆส่วน โดยเรียกแต่ละส่วนว่าฟังก์ชั่น และเมื่อนำฟังก์ชั่นมารวมเข้าด้วยกันก็จะเรียกว่าโปรแกรม โดยโครงสร้างการเขียนโปรแกรมของ Arduino ทุกโปรแกรมจะต้องประกอบไปด้วยฟังก์ชั่นจำนวนเท่าใดก็ได้แต่อย่างน้อยที่สุด ต้องมีฟังก์ชั่นจำนวน 2 ฟังก์ชั่น คือ setup() และ loop() โครงสร้างพื้นฐานของภาษาซีที่ใช้กับ Arduino นั้นจะประกอบไปด้วย 3 ส่วนคือ

4.1 Header ในส่วนนี้จะมีหรือไม่มีก็ได้ ถ้ามีต้องกำหนดไว้ในส่วนเริ่มต้นของโปรแกรม

4.2 setup() ในส่วนนี้เป็นฟังก์ชั่นบังคับที่ต้องกำหนดให้มีในโปรแกรม ถึงแม้ว่าในบางโปรแกรมจะไม่ต้องการใช้งาน ก็ยังจำเป็นต้องประกาศไว้ด้วยเสมอ เพียงแต่ไม่ต้องเขียนคำสั่งใดๆ ไว้ในระหว่างวงเล็บปีกกา {} วงเล็บปีกกาที่ใช้เป็นตัวกำหนดขอบเขตของฟังก์ชั่น โดยฟังก์ชั่นนี้จะใช้สำหรับบรรจุคำสั่งในส่วนที่ต้องการให้โปรแกรมทำงานเพียงรอบเดียว และตอนเริ่มต้นการทำงานของโปรแกรมครั้งแรกเท่านั้น ซึ่งได้แก่คำสั่งเกี่ยวกับการ Setup ค่าการทำงานต่างๆ เช่น การกำหนดหน้าที่การใช้งานของ Pin Mode และการกำหนดค่า Baud rat สำหรับใช้งานพอร์ตสื่อสารอนุกรม เป็นต้น

4.3 loop() เป็นส่วนฟังก์ชันบังคับที่ต้องกำหนดให้มีในทุกๆโปรแกรมเช่นเดียวกันกับฟังก์ชั่น setup() โดยฟังก์ชัน loop()นี้ จะใช้บรรจุคำสั่งที่ต้องการให้โปรแกรมทำงาน เป็นวงรอบซ้ำๆ กันไปไม่รู้จบ ซึ่งถ้าเปรียบเทียบกับรูปแบบของ ANSI-C ส่วนนี้ก็คือ ฟังก์ชัน main() นั่นเอง

หน้าที่ของฟังก์ชันของ Arduino

#include <header.h>

เมื่อพบคำสั่ง #include ตัวแปลภาษาของ Arduino จะไปค้นหาไฟล์ที่ระบุไว้ในเครื่องหมาย <> หลังคำสั่ง #include จากตำแหน่ง Directory ที่เก็บไฟล์ Library ของโปรแกรม Arduino ไว้ซึ่งแน่นอน ส่วนของ Header จะนับรวมไปถึงคำสั่งส่วนที่ใช้ประกาศสร้าง ตัวแปร (Variable Declaration) และค่าคงที่ (Constant Declaration) รวมทั้ง ฟังก์ชันต่างๆ (Function Declaration) ด้วย ซึ่งจากตัวอย่างได้แก่ส่วนที่เป็นคำสั่ง สำหรับส่วนที่สำคัญที่สุดและขาดไม่ได้ คือ ฟังก์ชัน setup() และ ฟังก์ชัน Loop) ซึ่งฟังก์ชันทั้ง 2 ส่วนนี้ มีรูปแบบโครงสร้างที่เหมือนกัน แต่ถูกกำหนดด้วยชื่อ ฟังก์ชันเป็นการเฉพาะ คือ setup() และ loop() โดย setup() จะเขียนไว้ก่อน loop() ซึ่งทั้ง 2 ฟังก์ชัน นี้ มีขอบเขต เริ่มต้นและสิ้นสุดอยู่ภายใต้เครื่องหมาย{} Void setup()

{

คำสั่งต่างๆ ที่ต้องการเขียนไว้ภายใต้ฟังก์ชัน setup()

}

หน้าที่ของฟังก์ชัน setup() ใน Arduino คือ ใช้ทำหน้าที่เป็นส่วนของโปรแกรมย่อยสำหรับใช้บรรจุคำสั่งต่างๆที่ใช้สำหรับกำหนดการทำงานของระบบหรือกำหนดคุณสมบัติการทำงานให้กับอุปกรณ์ต่างๆ ซึ่งคำสั่งทั้งหมดที่บรรจุไว้ภายใต้ฟังก์ชันของ Setup() นี้ จะถูกเรียกขึ้นมาทำงานเพียงรอบเดียวคือตอนเริ่มต้นการทำงานของโปรแกรม โดยคำสั่งที่นิยมบรรจุไว้ในฟังก์ชันส่วนนี้ ได้แก่ คำสั่งสำหรับกำหนดโหมดการทำงานของ Digital Pin หรือ คำสั่งสำหรับกำหนดคุณสมบัติของพอร์ตสื่อสารอนุกรม เป็นต้น

void loop()

{

คำสั่งต่างๆ ที่ต้องการให้ทำงานภายใต้ฟังก์ชั่น loop()

}

หน้าที่ของฟังก์ชั่น loop() ใน Arduino คือใช้ทำหน้าที่เป็นส่วนของโปรแกรมหลัก สำหรับใช้บรรจุคำสั่งควบคุมการทำงานต่างๆของโปรแกรม ที่ต้องการใช้โปรแกรมการทำงาน โดยคำสั่งที่บรรจุไว้ในฟังก์ชันนี้จะถูกเรียกขึ้นมาทำงานซ้ำๆ กันตามลำดับและเงื่อนไขที่กำหนดไว้

4. ภาษา C

ภาษา C++ เป็นภาษาโปรแกรมคอมพิวเตอร์อเนกประสงค์มีโครงสร้างภาษาที่มีการจัดชนิดข้อมูลแบบสแตติก (Statically typed) และสนับสนุนรูปแบบการเขียนโปรแกรมที่หลากหลาย (Multi paradigm language) ได้แก่ การโปรแกรมเชิงกระบวนคำสั่ง การนิยามข้อมูล การโปรแกรมเชิงวัตถุ และการโปรแกรมแบบ (generic programming) ภาษา C++ เป็นภาษาโปรแกรมเชิงพาณิชย์ที่นิยมมากภาษาหนึ่งนับตั้งแต่ช่วงทศวรรษ 1990 (Bjarne Stroustrap) จาก (BelLabs) เป็นผู้พัฒนาภาษา C++ (เดิมใช้ชื่อ "C with classes") ในปี ค.ศ. 1983 เพื่อพัฒนาภาษาซีดั้งเดิมสิ่งที่พัฒนาขึ้นเพิ่มเติมนั้นเริ่มจากการเพิ่มเติมการสร้างคลาสจากนั้นก็เพิ่ม

คุณสมบัติต่างๆตามมา ได้แก่ เวอร์ชวลไลเซชั่น การโอเวอร์โหลดโอเปอเรเตอร์ การสืบทอดหลายสายเทมเพลตและการจัดการเอกเซสชั่น มาตรฐานของภาษา C++ ได้รับการรับรองในปี ค.ศ. 1998 เป็น มาตรฐาน ISO/IEC 14882:1998 เวอร์ชั่นล่าสุดคือ เวอร์ชั่นในปี ค.ศ. 2003 ซึ่งเป็นมาตรฐาน ISO/IEC 14882:2003 ในปัจจุบันมาตรฐานของภาษาในเวอร์ชั่นใหม่ (รู้จักกันในชื่อ C++Ox) กำลังอยู่ในขั้นพัฒนา

5. รูปแบบของการออกแบบภาษา C

ภาษา C++ ได้ถูกออกแบบมาเพื่อเป็นภาษาสำหรับการเขียนโปรแกรมทั่วไป สามารถรองรับการเขียนโปรแกรมในระดับภาษาเครื่องได้เช่นเดียวกับภาษาซี ในทางทฤษฎีภาษา C++ ควรจะมีความเร็วเทียบเท่าภาษา C แต่ในการเขียนโปรแกรมจริงนั้น ภาษา C เป็นภาษาที่มีการเปิดกว้าง ให้โปรแกรมเมอร์เลือกรูปแบบการเขียนโปรแกรม ซึ่งทำให้มีแนวโน้มที่โปรแกรมเมอร์อาจใช้รูปแบบที่ไม่เหมาะสม ซึ่งทำให้โปรแกรมที่เขียนมีประสิทธิภาพต่ำกว่าที่ควรจะเป็น และภาษา C++ นั้นเป็นภาษาที่มีความซับซ้อนมากกว่าภาษาซี จึงทำให้มีโอกาสเกิดอาการไม่เสถียรขณะคอมไพล์มากกว่า ภาษา C++ ได้รับการออกแบบเพื่อเข้ากันได้กับภาษา C ในเกือบทุกกรณี

6. โปรแกรมควบคุมการทำงาน

```
#include <Wire.h>
#include <MMA8453_n0m1.h>
#include <MAX30105.h>
#include "heartRate.h"
#include <TridentTD_LineNotify.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 16
#define D5 14
#define BUZZER_PIN D5
#define SSID        "B"      // บรรทัดที่ 11 ให้ใส่ ชื่อ Wifi ที่จะเชื่อมต่อ
#define PASSWORD    "boatolo28"     // บรรทัดที่ 12 ใส่ รหัส Wifi
#define LINE_TOKEN  "umIcCeLkC80sJonhsT6PO2C3YxIpHacHCbaideNUENj"  // บรรทัดที่ 13 ใส่ รหัส TOKEN ที่ได้มาจากข้างบน
MAX30105 particleSensor;
```

```
MMA8453_n0m1 accel;
Adafruit_SSD1306 display(OLED_RESET);
const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
float beatsPerMinute;
int beatAvg;
int BUZZER = D5;
void setup()
{

Serial.begin(115200);
Serial.println("Initializing...");
Serial.println();
Serial.println(LINE.getVersion());
if (WiFi.begin(SSID, PASSWORD)){
Serial.printf("WiFi connecting to %s\n",  SSID);
WiFi.status() != WL_CONNECTED ;
Serial.print(".");
delay(400);
}
else{
void NOT();
void loop();
  }
Serial.printf("\nWiFi connected\nIP : ");
Serial.println(WiFi.localIP());
LINE.setToken(LINE_TOKEN);
accel.setI2CAddr(0x1C);
accel.dataMode(true, 2);
if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz
speed {
Serial.println("MAX30105 was not found. Please check wiring/power. ");
while (1);
```

```
}
Serial.println("Place your index finger on the sensor with steady pressure.");
particleSensor.setup(); //Configure sensor with default settings
particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is running
particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
display.begin(SSD1306_SWITCHCAPVCC, 0x3c); //Start the OLED display
display.display();
delay(1000);
pinMode(BUZZER_PIN, OUTPUT);
}
void loop()
{
long irValue = particleSensor.getIR();
if (checkForBeat(irValue) == true)

{
 //We sensed a beat!
long delta = millis() - lastBeat;
lastBeat = millis();
beatsPerMinute = 60 / (delta / 1000.0);
if (beatsPerMinute < 255 && beatsPerMinute > 20)
{
rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array
rateSpot %= RATE_SIZE; //Wrap variable
//Take average of readings
beatAvg = 0;
for (byte x = 0 ; x < RATE_SIZE ; x++)
beatAvg += rates[x];
beatAvg /= RATE_SIZE;
}

}
Serial.print("IR=");
Serial.print(irValue);
```

```
Serial.print(", BPM=");
Serial.print(beatsPerMinute);
Serial.print(", Avg BPM=");
Serial.println(beatAvg);
accel.update();
int accX = accel.x();
int accY = accel.y();
int accZ = accel.z();
Serial.print("x:");
Serial.print(accX);
Serial.print("\ty:");
Serial.print(accY);
Serial.print("\tz:");
Serial.println(accZ);
if ((abs(accX) > 150) && (abs(accX) < 260 ))
{
digitalWrite(BUZZER_PIN, LOW);
delay(500);
digitalWrite(BUZZER_PIN, HIGH);
delay(250);
digitalWrite(BUZZER_PIN, LOW);
delay(500);
digitalWrite(BUZZER_PIN, HIGH);
delay(250);
digitalWrite(BUZZER_PIN, LOW);
delay(500);
digitalWrite(BUZZER_PIN, HIGH);
delay(250);
digitalWrite(BUZZER_PIN, LOW);
delay(500);
digitalWrite(BUZZER_PIN, HIGH);
delay(250);
LINE.notify("ไม้เท้าล้ม" ;
}
else if ((abs(accZ) > 150 ) || (abs(accZ) < 260 ))
```

```
{
digitalWrite(BUZZER_PIN, LOW);
delay(500);
digitalWrite(BUZZER_PIN, HIGH);
delay(250);
digitalWrite(BUZZER_PIN, LOW);
delay(500);
digitalWrite(BUZZER_PIN, HIGH);
delay(250);
digitalWrite(BUZZER_PIN, LOW);
delay(500);
digitalWrite(BUZZER_PIN, HIGH);
delay(250);
digitalWrite(BUZZER_PIN, LOW);
delay(500);
digitalWrite(BUZZER_PIN, HIGH);
delay(250);
LINE.notify("ไม้เท้าล้ม" ;
}
else {
digitalWrite(BUZZER_PIN, HIGH);
}
if(irValue > 20000){
BUZZ();
}else if(irValue < 20000){
display.clearDisplay();
Max();
}
{
Serial.println();
}
}
void BUZZ(){
int beatAvg;
//Take average of readings
```

```
beatAvg = 0;
for (byte x = 0 ; x < RATE_SIZE ; x++)
beatAvg += rates[x];
beatAvg /= RATE_SIZE;
display.clearDisplay();
display.setTextSize(2);
display.setTextColor(WHITE);
display.setCursor(0,25);
display.print("AvgBPM");
display.setTextSize(2);
display.setTextColor(WHITE);
display.setCursor(100,25);
display.print(beatAvg);
display.display();
if (  beatAvg > 100){
digitalWrite(BUZZER,LOW);
delay(250);
digitalWrite(BUZZER,HIGH);
delay(250);
digitalWrite(BUZZER,LOW);
delay(250);
digitalWrite(BUZZER,HIGH);
delay(250);
digitalWrite(BUZZER,LOW);
delay(250);
digitalWrite(BUZZER,HIGH);
delay(250);
}else{
digitalWrite(BUZZER ,HIGH);
}
}
void Max(){
int beatAvg;
//Take average of readings
beatAvg = 0;
```

```
for (byte x = 0 ; x < RATE_SIZE ; x++)
beatAvg += rates[x];
beatAvg /= RATE_SIZE;
if ((beatAvg > 100) || (beatAvg < 40)) {
digitalWrite(BUZZER, HIGH);
LINE.notify("ชีพจรผิดปกติ" ;
LINE.notify(beatAvg);
delay(2000);
ESP.restart();
}else {
}
}
void NOT(){
if (WiFi.begin(SSID, PASSWORD)){
Serial.printf("WiFi connecting to %s\n",  SSID);
WiFi.status() != WL_CONNECTED ;
Serial.print(".");
delay(400);
}
else{
void loop();
}
}
```

บรรณานุกรม

CyberTice.//(2560)// บอร์ด ESP8266 V3//สืบค้นเมื่อ 22 กุมภาพันธ์ 2565./

จาก/https://www.cybertice.com/article/79/สอนใช้งาน-nodemcu-esp8266-เริ่มต้นติดตั้ง-nodemcu-esp8266-ลงบน-arduino-ide

CyberTice.//(2558)// max30102//สืบค้นเมื่อ 25 กุมภาพันธ์ 2565./

จาก/https://www.cybertice.com/product/2745/max30102-เซ็นเซอร์วัดออกซิเจนในเลือดและอัตราการเต้นของหัวใจ-ชีพจร-นิ้วมือ-blood-oxygen-concentration

CyberTice.//(2559)// buzzer//สืบค้นเมื่อ 20 กุมภาพันธ์ 2565./

จาก/https://www.cybertice.com/category/71/electronic-component/ลำโพง-buzzer-ไฟสัญญาณเตือน

Arduino4.//(2560)// GY-45 3-axis Accelerometer Module
//สืบค้นเมื่อ 27 กุมภาพันธ์ 2565./

https://www.arduino4.com/product/662/gy-45-3-axis-accelerometer-module-mma845x-โมดูลไจโร-gyro-accelerometer-3-แกน

ประวัติผู้จัดทำ

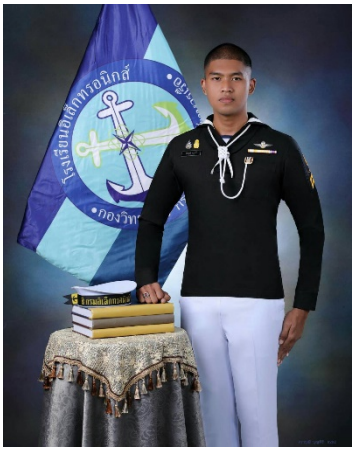
ชื่อ-นาม นรจ.มูฮัมหมัด เหมโคกน้อย
พรรค - เหล่า อิเล็กทรอนิกส์

ชื่อ-นาม นรจ.สุรพัศ หน่ายสุวรรณ
พรรค - เหล่า อิเล็กทรอนิกส์

ชื่อ-นาม นรจ.ณัฐกิตติ์ อุทโท
พรรค พิเศษ เหล่า อิเล็กทรอนิกส์ - ตำรวจน้ำ

ชื่อ-นาม นรจ.ธนากร สายมงคล
พรรค พิเศษ เหล่า อิเล็กทรอนิกส์

ชื่อ-นาม นรจ.สิรวิชญ์ วงศ์อนุ
พรรค พิเศษ เหล่า อิเล็กทรอนิกส์



ชื่อ-นาม นรจ.ศราวุฒิ บุญศิริ
พรรค นาวิน เหล่า อุทกศาสตร์