**ถุงมือสำหรับตรวจจับท่าทางมือ**

**SMART GLOVE FOR GESTURE RECOGNITION**

**พัทธวีร์ ชุมภูวร**

**พิทวัส คุณกะมุต**

**รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต  
สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2561**

**บทคัดย่อ**

**ABSTRACT**

**สารบัญ**

**หน้า**

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศIII

สารบัญIV

สารบัญตารางV

สารบัญภาพVI

บทที่ 1 บทนำ1

1.1 ความเป็นมาของปัญหา1

1.2 วัตถุประสงค์ของโครงงาน2

1.3 ประโยชน์ที่คาดว่าจะได้รับ2

1.4 ขอบเขตของโครงงาน3

1.5 ข้อจำกัดของโครงงาน4

1.6 แผนการดำเนินงาน5

**สารบัญตาราง**

**หน้า**

1.1 แผนการดำเนินงาน 5

**สารบัญภาพ**

**หน้า**

1.1 ถุงมือตรวจจับท่าทางมือที่ใช้งานร่วมกับเทคโนโลยี Virtual Reality 1

1.2 ถุงมือตรวจจับท่าทางมือเพื่อใช้แปลภาษามือ 1

1.3 โครงสร้างถุงมือ 3

1.4 โครงสร้างระบบ 4

1.5 ขั้นตอนการทำงาน – การตรวจจับท่าทาง 4

**บทที่ 1**

**บทนำ**

* 1. **ความเป็นมาของปัญหา**

ในปัจจุบันมีการพัฒนาอุปกรณ์ต่าง ๆ โดยมีเป้าหมายเพื่อให้ผู้ใช้สามารถใช้งานอุปกรณ์ทางอิเล็กทรอนิกส์โดยใช้ร่างกายของตนเองเป็น Input ได้ และได้รับความรู้สึกสมจริง หรือสะดวกสบายมากขึ้นหากเทียบกับการใช้งาน Input ปกติอย่างเช่นคีย์บอร์ด, เมาส์ หรือจอสัมผัส

หากกล่าวถึงอุปกรณ์ที่พยายามตรวจจับท่าทางของมือ ในปัจจุบันก็มีการพัฒนาขึ้นมาหลากหลายประเภทเพื่อวัตถุประสงค์ใดอย่างหนึ่งอย่างชัดเจน ตัวอย่างเช่น ถุงมือตรวจจับท่าทางมือที่ใช้งานร่วมกับเทคโนโลยี Virtual Reality เพื่อใช้ในการเล่นเกมส์เพื่อความบันเทิง หรือฝึกฝนทักษะปฏิบัติเสมือนจริง และถุงมือตรวจจับท่าทางมือเพื่อใช้แปลภาษามือเป็นคำหรือตัวอักษรภาษาอังกฤษ เป็นต้น



**รูป 1.1 ถุงมือตรวจจับท่าทางมือที่ใช้งานร่วมกับเทคโนโลยี Virtual Reality**



**รูป 1.2 ถุงมือตรวจจับท่าทางมือเพื่อใช้แปลภาษามือ**

แต่เนื่องจากถุงมือตามตัวอย่างที่กล่าวมาข้างต้นเป็นถุงมือที่ถูกพัฒนาขึ้นมาเพื่อวัตถุประสงค์ใด ๆ อย่างชัดเจน ท่าทางต่าง ๆ ที่ใช้เป็น Input รวมถึง Output จึงจำกัดอยู่ในขอบเขตที่ผู้พัฒนาได้กำหนดไว้เพียงเท่านั้น ผู้ใช้จริงไม่สามารถกำหนดท่าทางต่าง ๆ โดยเฉพาะได้ หรือต้องมีการพัฒนาต่อโดยใช้ชุดพัฒนาซอฟต์แวร์ที่รองรับจากผู้พัฒนา ร่วมกับความรู้เฉพาะด้านเพียงเท่านั้น

ทางผู้พัฒนาจึงสนใจที่จะสร้างถุงมือเพื่อตรวจจับท่าทางมือ โดยใช้เทคโนโลยี Machine Learning ที่ได้รับการพัฒนาประสิทธิภาพขึ้นและมีความนิยมในปัจจุบัน เพื่อลดข้อจำกัดในด้านที่กล่าวมาข้างต้นลง โดยการแยกโครงสร้างการเคลื่อนไหวของท่าทางออกเป็นท่านิ่งหลาย ๆ ท่าที่เชื่อมต่อกัน และมีแอพพลิเคชั่นให้ผู้ใช้สามารถตั้งค่าชุดของท่าทางที่จะใช้เป็น Input และตั้งค่า Output ได้ ดังนั้นถุงมือจะสามารถตรวจจับท่าทางที่เป็นการเคลื่อนไหวได้หลากหลายแบบ หลากหลายท่าติดต่อกันตามที่ต้องการ

**1.2 วัตถุประสงค์ของโครงงาน**

1.) พัฒนาถุงมือเพื่อการตรวจจับท่าทางมือรวมถึงระบบที่เกี่ยวข้อง

2.) ศึกษาการใช้งานไมโครคอนโทรลเลอร์และระบบปฏิบัติการ Linux ภายใน

3.) ศึกษาและพัฒนาเว็บแอพพลิเคชั่นและ Server เพื่อติดต่อกับ Hardware

4.) ศึกษาการประมวลผลข้อมูลของเซนเซอร์ผ่านกระบวนการทาง Machine Learning

**1.3 ประโยชน์ที่คาดว่าจะได้รับ**

1.) ถุงมือสามารถอำนวยความสะดวกให้กับผู้ใช้งานได้

2.) ถุงมือสามารถให้ความบันเทิงกับผู้ใช้งานได้

3.) ถุงมือสามารถช่วยเป็นสื่อกลางในการสื่อสารโดยภาษามือได้บางส่วน

4.) ผู้ใช้สามารถเข้าถึงและตั้งค่าการใช้งานถุงมือได้ง่าย

**1.4 ขอบเขตของโครงงาน**

การจับท่าทางของถุงมือที่เป็นท่านิ่ง หนึ่งท่าจะประกอบไปด้วยการงอนิ้ว กับการวางมือ โดยการงอนิ้ว จะสามารถทำได้อย่างน้อย 30 แบบ และการวางมือ สามารถทำได้อย่างน้อย 10 แบบ

สำหรับท่าที่เป็นเคลื่อนไหว สามารถเลือกได้สองแบบ คือแบบที่เกิดจากท่านิ่งหลายท่าต่อกัน และแบบท่าหนึ่งเปลี่ยนไปอีกท่าหนึ่งโดยมีการตรวจจับการเปลี่ยนผ่านของท่า ซึ่งแบบนี้จะสามารถทำได้อย่างน้อย 10 ท่า

และโดยรวมแล้ว จะสามารถตรวจจับท่าทางภาษามือที่เป็นตัวอักษรภาษาไทยได้อย่างน้อย 30 ตัว

**1.5 ข้อจำกัดของโครงงาน**

รูปแบบท่าทางของของมือผู้ใช้ที่ไม่เหมือนกันอาจจะส่งผลให้ความแม่นยำแตกต่างกัน เกิดจากค่าที่ได้จากเซนเซอร์ ซึ่งอาจจะมีค่าผิดพลาดหรือแตกต่างมากพอที่จะทำให้ตรวจจับได้ท่าที่ผิดพลาดได้ โดยเฉพาะเซนเซอร์ที่ใช้วัดความงอของนิ้วมือ ซึ่งวัดการงอได้เพียงแค่ทิศทางที่กำมือเท่านั้น สำหรับผู้ใช้ที่สามารถกางและงอนิ้วมือไปในทิศตรงข้ามได้มาก อาจจะทำให้เกิดการตรวจจับที่ผิดพลาดได้

จากที่กล่าวมาข้างต้น เซนเซอร์วัดความงอนิ้วมือนั้นสามารถวัดได้เพียงทิศทางเดียว ดังนั้นการงอนิ้วในทิศทางด้านข้าง เช่นการไขว้นิ้ว จะไม่สามารถตรวจจับได้ ซึ่งท่าดังกล่าวเป็นท่าที่ไม่ได้อยู่ในขอบเขตการพัฒนา เนื่องจากท่าในปัจจุบัน มีเพียงพอสำหรับการใช้งานเบื้องต้นแล้ว

**1.6 แผนการดำเนินงาน**

**ตารางที่ 1.1** แผนการดำเนินงาน

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **หัวข้อกิจกรรม** | **เดือน** | | | | | | | | | |
| **ส.ค.** | **ก.ย.** | **ต.ค.** | **พ.ย.** | **ธ.ค.** | **ม.ค.** | **ก.พ.** | **มี.ค.** | **เม.ย.** | **พ.ค.** |
| 1. ค้นหาหัวข้อที่สนใจ และปรึกษาหัวข้อดังกล่าวกับอาจารย์ |  |  |  |  |  |  |  |  |  |  |
| 2. ศึกษาเทคโนโลยีที่ใช้พัฒนา | | | | | | | | | | |
| 2.1 ศึกษาวิธีการใช้งานระบบปฏิบัติการ |  |  |  |  |  |  |  |  |  |  |
| 2.2 ทดสอบการติดตั้งส่วนประกอบต่าง ๆ ที่ใช้ในการพัฒนา |  |  |  |  |  |  |  |  |  |  |
| 3. ออกแบบ | | | | | | | | | | |
| 3.1 ออกแบบโครงสร้างถุงมือ |  |  |  |  |  |  |  |  |  |  |
| 3.2 ออกแบบโครงสร้างระบบ |  |  |  |  |  |  |  |  |  |  |
| 3.3 ออกแบบโครงสร้างและวิธีประมวลผล Dataset |  |  |  |  |  |  |  |  |  |  |
| 3.4 ออกแบบเว็บแอพพลิเคชั่น |  |  |  |  |  |  |  |  |  |  |
| 4. พัฒนา | | | | | | | | | | |
| 4.1 ประกอบถุงมือเพื่อให้สามารถใช้งานได้เบื้องต้น |  |  |  |  |  |  |  |  |  |  |
| **หัวข้อกิจกรรม** | **เดือน** | | | | | | | | | |
| **ส.ค.** | **ก.ย.** | **ต.ค.** | **พ.ย.** | **ธ.ค.** | **ม.ค.** | **ก.พ.** | **มี.ค.** | **เม.ย.** | **พ.ค.** |
| 4.2 พัฒนาเว็บแอพพลิเคชั่นเพื่อใช้แสดงผลและสร้าง Dataset |  |  |  |  |  |  |  |  |  |  |
| 4.5 พัฒนาเว็บแอพพลิเคชั่นเพื่อใช้ตั้งค่า Input/output |  |  |  |  |  |  |  |  |  |  |
| 5. ทดสอบ ปรับปรุง และแก้ไข | | | | | | | | | | |
| 5.1 ทดสอบและแก้ไขการทำงานโดยรวมของระบบ |  |  |  |  |  |  |  |  |  |  |
| 5.2 พัฒนาส่วนจ่ายไฟให้ถุงมือ |  |  |  |  |  |  |  |  |  |  |
| 5.3 ปรับปรุงรูปลักษณ์ถุงมือ |  |  |  |  |  |  |  |  |  |  |
| 5.4 ปรับปรุงเว็บแอพพลิเคชั่น |  |  |  |  |  |  |  |  |  |  |
| 5.5 ปรับปรุงโมเดล Machine Learning |  |  |  |  |  |  |  |  |  |  |

**บทที่ 2**

**ทฤษฎีและงานวิจัยที่เกี่ยวข้อง**

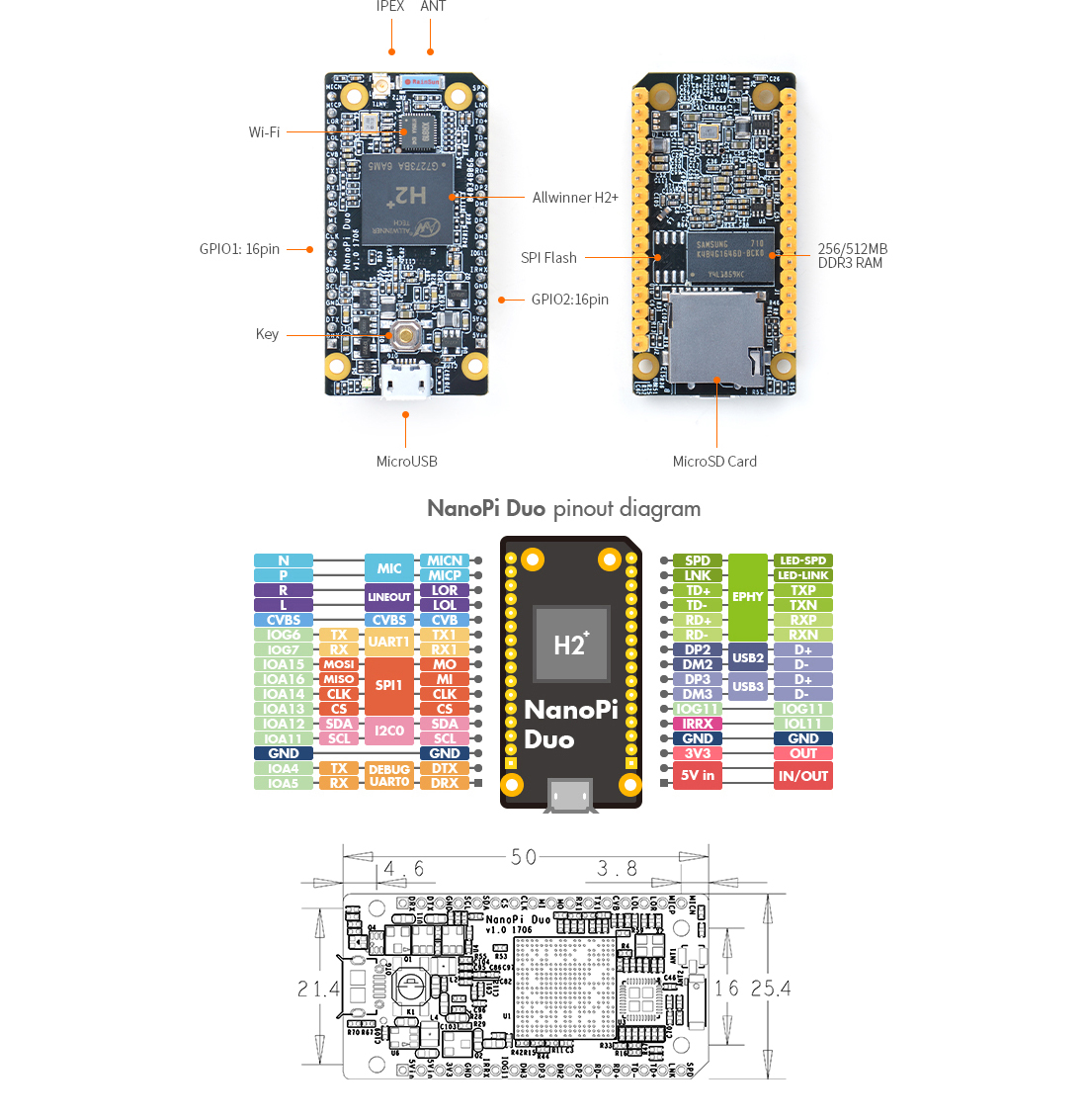
**2.1 เทคโนโลยีด้านฮาร์ดแวร์**

**2.1.1 ไมโครคอนโทรลเลอร์ (Microcontroller)**

Nanopi-DUO หน่วยประมวลผลขนาดเล็กที่สามารถวางไว้บนข้อมือได้อย่างสะดวก มีคอร์ในการประมวลผลถึง 4 คอร์ จึงสามารถทำงานหลายงานพร้อมกันได้สะดวก เหมาะสมกับโครงการที่จำเป็นจะต้องประมวลผล เปิดเซิร์ฟเวอร์ และติดต่อกับฮาร์ดแวร์ในเวลาเดียวกัน อีกทั้งยังรันในระบบปฏิบัติการ Linux (Ubuntu) ที่ยอดนิยม สามารถศึกษาเพิ่มเติมได้ง่าย และสามารถติดตั้งแพคเกจที่จำเป็นต้องใช้ในการดำเนินโปรเจคได้

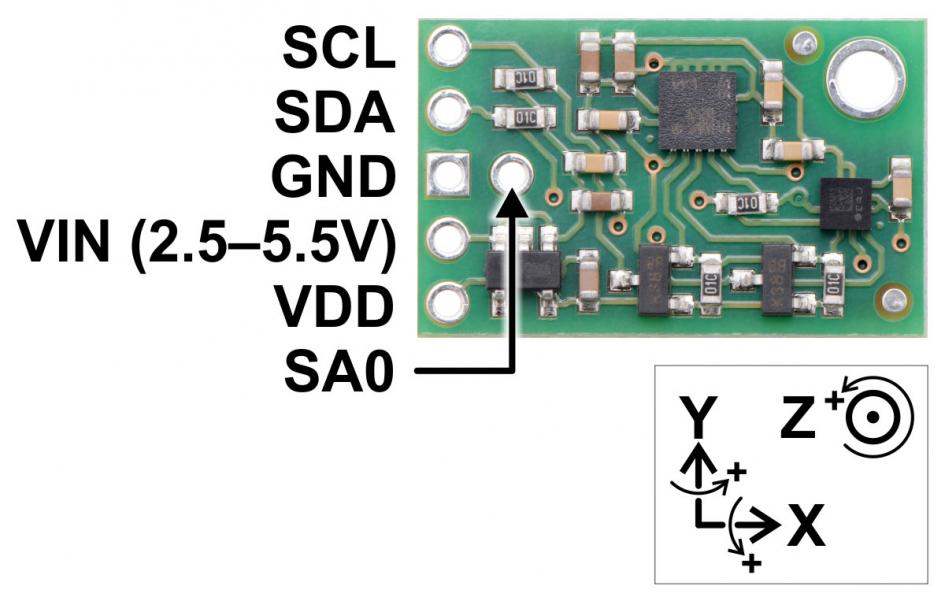
ข้อมูลจำเพาะ

* CPU: Allwinner H2+, Quad-core Cortex-A7
* DDR3 RAM: 256MB/512MB
* Connectivity: 10/100M Ethernet
* Wifi：XR819
* USB Host: 2.54mm pin x2, exposed in 2.54mm pitch pin header
* MicroSD Slot x 1
* MicroUSB: OTG and power input
* Debug Serial Interface: exposed in 2.54mm pitch pin header
* Audio input/output Interface: exposed in 2.54mm pitch pin header
* GPIO1: 2.54mm spacing 16pin. It includes UART, SPI, I2C, Audio etc
* GPIO2: 2.54mm spacing 16pin. It includes USB,10/100M Ethernet, IO etc
* PCB Dimension: 25.4 x 50mm
* Power Supply: DC 5V/2A
* Temperature measuring range: -40 C to 80 C
* OS/Software: U-boot，Linux Kernel 4.11.2 (mainline) , Ubuntu 16.04.2 LTS (Xenial)



**รูป 2.1 แผนภาพโครงสร้าง, ขาเชื่อมต่อของ NanoPI-DUO**

**2.1.2 เซนเซอร์ (Sensors)** เซนเซอร์ที่ใช้ตรวจจับการเคลื่อนไว้ของมือจะประกอบไปด้วย Flex Sensor เพื่อวัดความงอของแต่ละนิ้วมือ และ IMU (Inertial Measurement Unit: Accelerometer+Gyroscope+Magnetometer หรือเซนเซอร์หน่วยวัดความเร่งและสนามแม่เหล็ก 9 แกน) เพื่อใช้ตรวจจับการเคลื่อนไหวและวัดมุมปัจจุบันของมือ



**รูป 2.2 Polulu MinIMU9-v5**

Polulu MinIMU9-v5 เป็น IMU ตัวที่ใช้ในโครงงานนี้ ซึ่งมีขนาดเล็ก วางบนมือและปลายนิ้วได้ สามารถส่งข้อมูลผ่านบัส I2C ได้

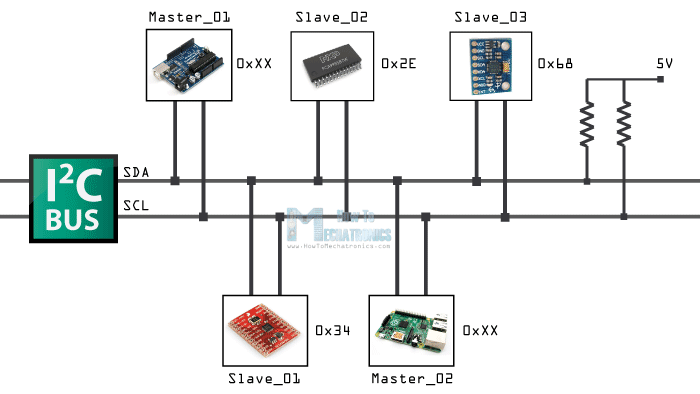
**2.1.3 จอแสดงผล (Display)**



**รูป 2.3 SSD1306**

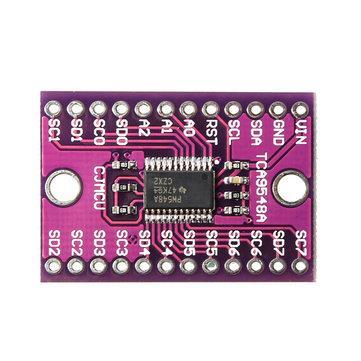
SSD1306 Bi-Color OLED Display เป็นจอแสดงผลสองสี ขนาด 128x64 ซึ่งมีขนาดพอดี สามารถวางไว้บนมือได้ นอกจากนั้นยังใช้การสื่อสารด้วย I2C เช่นเดียวกันกับเซนเซอร์อื่น ๆ

**2.1.4 การติดต่อระหว่างฮาร์ดแวร์**



**รูป 2.4 ตัวอย่างเครือข่ายการเชื่อมต่อของบัส I2C**

Inter Integrate Circuit (I2C) เป็นการสื่อสารอนุกรมแบบเข้าสัญญาณ ที่ทั้งสองฝั่งสามารถเป็นได้ทั้งผู้รับและผู้ส่ง เซนเซอร์จะต่อเข้ากับไมโครคอนโทรลเลอร์ผ่านบัส I2C แล้วไมโครคอนโทรลเลอร์สามารถเขียนและอ่านเซนเซอร์ได้จากการเข้าถึงเลขตำแหน่ง (Address) จำเพาะนั้น ๆ ของเซนเซอร์



**รูป 2.5 TCA9548A**

โมดูลขยายช่องสัญญาณ I2C 8 ช่อง (TCA9548A) เนื่องจาก IMU ที่ใช้ในการดำเนินโครงงานอาจมีหลายตัว แต่ทุกตัวที่ใช้นั้นมีเลขตำแหน่งเดียวกัน เพื่อแก้ไขปัญหาที่เลขตำแหน่งซ้ำ จึงต้องใช้โมดูลตัวนี้ โดยหลักการการทำงานเหมือน Multiplexer กล่าวคือสามารถเขียนช่องที่ต้องการไปยังตำแหน่งของ TCA9548A เพื่ออ่านค่าจากช่องนั้น ๆ ได้

สำหรับการอ่านข้อมูลจาก I2C จะใช้ไลบรารี่ SMBus ของ Python ที่เขียนขึ้นมาเพื่อรับส่งข้อมูลด้วย I2C โดยเฉพาะ

**2.2 เทคโนโลยีด้านการประมวลผลท่าทางมือ**

การตรวจจับและประมวลผลท่าทางของมือ จะใช้ภาษา Python เวอร์ชั่น 3.6 เนื่องมาจากการที่เป็นภาษายอดนิยมในการพัฒนา Machine Learning ซึ่งส่งผลให้มีเฟรมเวิร์คหรือไลบรารี่ต่าง ๆ ที่สนับสนุนมากมาย

TensorFlow เป็นเฟรมเวิร์คที่ใช้ในการพัฒนาโมเดล Machine Learning เพราะสามารถใช้....

**2.3 เทคโนโลยีด้านฐานข้อมูล**



**รูป 2.6 โลโก้ MongoDB**

MongoDB เป็นระบบจัดการฐานข้อมูลแบบไม่เป็นตาราง (Non-relational Database) ที่ยอดนิยม ซึ่งมีประโยชน์อย่างมากในการดำเนินโครงงาน เนื่องจากมีข้อมูลบางชนิดที่ไม่ทราบขนาด การใช้ฐานข้อมูลแบบนี้และเก็บข้อมูลเป็นวัตถุแทนจะทำให้ดำเนินการเก็บและเรียกค้นได้ง่ายขึ้น

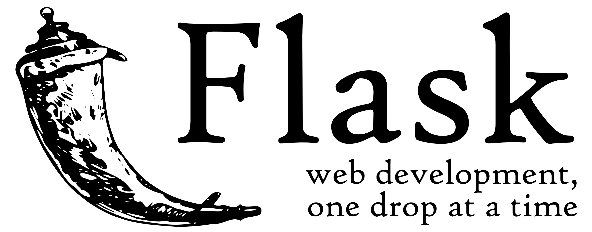
**2.4 เทคโนโลยีด้านแอพพลิเคชั่น**

ในการพัฒนาแอพพลิเคชั่นโต้ตอบกับผู้ใช้ ผู้พัฒนาเลือกที่จะทำเป็นเว็บแอพพลิเคชั่น เนื่องจากเมื่อผู้ใช้เชื่อมต่อกับถุงมือ ผู้ใช้ก็จะสามารถเข้าถึงได้ทันทีผ่านอุปกรณ์ทั่วไป เช่น คอมพิวเตอร์, โทรศัพท์ และแท็บเล็ต โดยไม่ต้องดาวน์โหลดแอพพลิเคชั่นใด ๆ ล่วงหน้า



**รูป 2.7 โลโก้ VueJS**

VueJS เป็นเฟรมเวิร์คที่ใช้ในการพัฒนาเว็บแอพพลิเคชั่นฝั่งผู้ใช้ เขียนโดยใช้ภาษา Javascript โดยเป็นเฟรมเวิร์คสามารถพัฒนาเว็บแอพพลิเคชั่นที่สามารถเปลี่ยนแปลงข้อมูลได้แบบเรียลไทม์ จึงเหมาะสมสำหรับการพัฒนาที่ต้องมีการป้อนและอัพเดทข้อมูลจากเซนเซอร์ตลอดเวลา



**รูป 2.8 โลโก้ Flask**

Flask เป็นเฟรมเวิร์คใช้ในการพัฒนาเว็บแอพพลิเคชั่นฝั่งเซิร์ฟเวอร์ เขียนโดยใช้ภาษา Python สามารถติดต่อฐานข้อมูล MongoDB ได้ง่าย และด้วยเนื่องจากการประมวลผลข้อมูลทำโดยใช้ภาษา Python จึงควรใช้เฟรมเวิร์คสำหรับพัฒนา Server เป็นภาษาเดียวกันด้วย ถึงจะสามารถทำงานร่วมกันได้

Socket.IO เป็นโมดูลที่ทำให้เว็บแอพพลิเคชั่นฝั่งผู้ใช้สามารถเชื่อมต่อกับเซิร์ฟเวอร์แบบเรียลไทม์ได้ เมื่อมีการเชื่อมต่อตั้งแต่เริ่มเข้าเว็บแอพพลิเคชั่น ทั้งสองฝั่งสามารถส่งข้อมูลหากันได้ตลอดเวลา และใช้เวลาน้อย เพราะไม่จำเป็นจะต้องเปิดการเชื่อมต่อทุกครั้งที่ส่งข้อมูลดังเช่นการส่งข้อมูลแบบ HTTP Request ที่นิยมทั่วไป

Eventlet เป็นไลบรารี่สำหรับช่วยในการทำงานแบบ Multithread ช่วยให้สามารถทำการเปิดเซิร์ฟเวอร์โดยใช้ Flask และรอข้อความจาก Socket.IO พร้อมกับการประมวลผลข้อมูลด้วย Machine Learning ไปพร้อม ๆ กันได้

**2.5 งานวิจัยที่เกี่ยวข้อง**

**2.5.1 Sign2Glove**

**2.5.2 A GESTURE DETECTION GLOVE FOR HUMAN-COMPUTER INTERACTION**

งานวิจัยนี้มีเป้าหมายเพื่อตรวจจับท่าทางต่าง ๆ ในการควบคุมเมาส์และคีย์บอร์ด โดยใช้ MCU เป็น FPGA และใช้เซนเซอร์ IMU ที่กลางมือและปลายนิ้วทั้ง 5 รวม 6 ตัว

**บทที่ 3**

**การออกแบบและพัฒนา**

**3.1 การออกแบบถุงมือ**

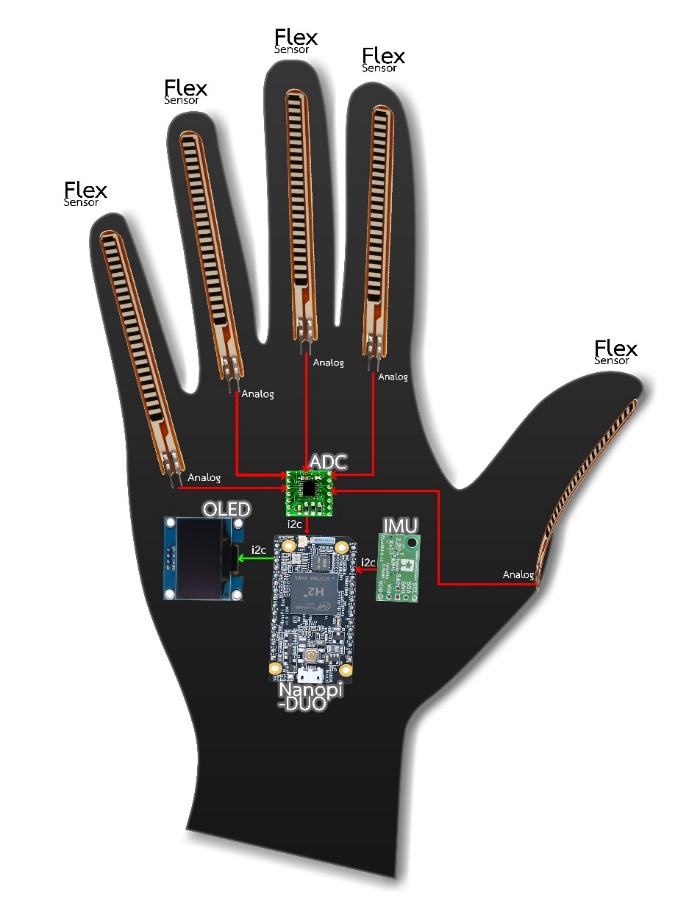
สำหรับถุงมือ ผู้พัฒนาได้ออกแบบไว้ 2 แบบ คือแบบที่ใช้เซนเซอร์วัดความงอสำหรับนิ้ว และเซนเซอร์ IMU สำหรับวัดมุมของมือ และอีกแบบคือแบบที่ใช้ IMU วัดทั้งความงอและมุมของมือ ถุงมือทั้ง 2 แบบ จะมีรูปแบบข้อมูลที่ได้รับแตกต่างกัน และส่งผลให้การประมวลผลในโมเดล Machine Learning ต่างกันด้วย

3.1.1 ถุงมือแบบที่ 1:

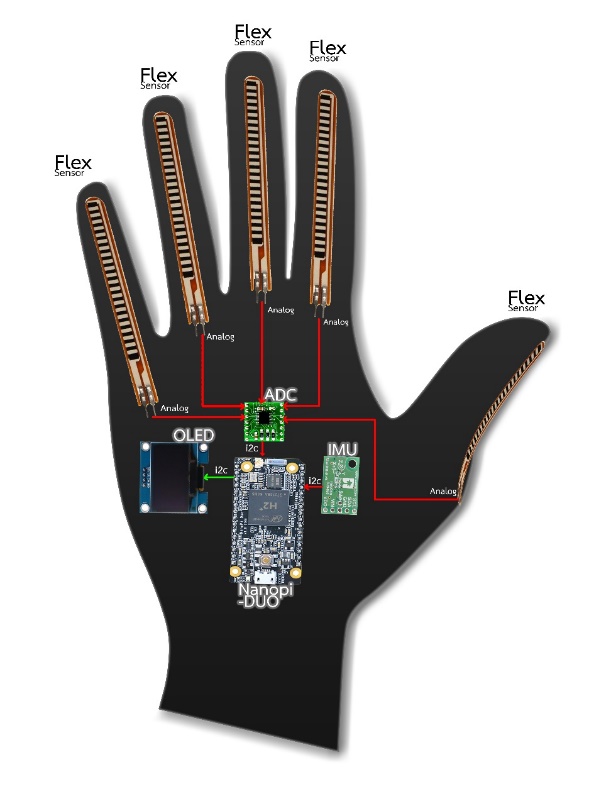
เป้าหมายของถุงมือรูปแบบนี้ คือเพื่อความแม่นยำของการงอนิ้ว 1 แกนที่เป็นแกนหลัก นั่นคือแกนในการกำมือ และเพื่อความง่ายในการพัฒนา เซนเซอร์ที่ใช้จะมีทั้งหมด 6 ตัว ประกอบไปด้วยเซนเซอร์ตรวจจับการงอ (Flex Sensor) ของนิ้ว 5 ตัว วางตามนิ้วแต่ละนิ้ว และเซนเซอร์ IMU ตรวจสอบการเคลื่อนไหวและการวางมือ วางไว้กลางหลังมือ 1 ตัว

3.1.2 ถุงมือแบบที่ 2:

เป้าหมายของถุงมือรูปแบบนี้ คือเพื่อความแม่นยำของตำแหน่งนิ้วในทุกแกน เซนเซอร์ที่ใช้จะมีทั้งหมด 6 ตัว ประกอบไปด้วยเซนเซอร์ IMU ของนิ้ว 5 ตัว วางตามปลายนิ้วบริเวณเล็บของแต่ละนิ้ว และเซนเซอร์ IMU ตรวจสอบการเคลื่อนไหวและการวางมือ วางไว้กลางหลังมือ 1 ตัว



**รูป 3.1 โครงสร้างถุงมือแบบที่ 1**



**รูป 3.2 โครงสร้างถุงมือแบบที่ 2**

**3.2 การออกแบบระบบโดยรวม**

เพื่อให้เห็นได้ชัดเจน ระบบสามารถแบ่งออกได้เป็น 3 ส่วน คือฮาร์ดแวร์, เซิร์ฟเวอร์ และเว็บแอพพลิเคชั่น

3.2.1 ฮาร์ดแวร์ (Hardware)

คือส่วนของถุงมือทั้งหมด รวมถึงเซนเซอร์และ MCU มีหน้าที่เก็บตัวอย่างข้อมูล (Sample) จากการทำท่าทางต่าง ๆ ของผู้ใช้ และแสดงผลข้อมูลต่าง ๆ ตามที่ได้คำสั่งจากเซิร์ฟเวอร์

3.2.2 เซิร์ฟเวอร์ (Server)

คือส่วนซอฟต์แวร์ที่รันบน MCU ตอดเวลา เขียนด้วยภาษา Python และมีหน้าที่ดังต่อไปนี้

* อ่านข้อมูลจากฮาร์ดแวร์ ทำการคำนวณเพื่อแปลงข้อมูลหรือเพื่อความแม่นยำหากจำเป็น
* ตรวจสอบข้อมูลจากฮาร์ดแวร์เพื่อหาช่วงของชุดข้อมูลที่ต้องนำไปคำนวณในโมเดล Machine Learning
* นำชุดข้อมูลไปคำนวณในโมเดล Machine Learning และรอรับคำตอบ
* ส่งเว็บแอพพลิเคชั่นให้ผู้ใช้หากมีการเรียกใช้งาน
* คอยรับและตอบคำร้องขอข้อมูลผ่าน HTTP (HTTP Request) จากเว็บแอพพลิเคชั่นฝั่งผู้ใช้
* คอยรับและส่งข้อมูลผ่าน WebSocket จากเว็บแอพพลิเคชั่นฝั่งผู้ใช้

3.2.3 เว็บแอพพลิเคชั่น (Web Application)

คือส่วนที่จะทำงานก็ต่อเมื่อผู้ใช้ส่งคำขอ HTTP (HTTP Request) ให้แก่เซิร์ฟเวอร์ แล้วเซิร์ฟเวอร์ก็จะส่งคำตอบรับ (HTTP Response) เป็นเว็บแอพพลิเคชั่น และทำงานอยู่บนบราวเซอร์ของผู้ใช้เอง ส่วนนี้มีหน้าที่เบื้องต้นดังนี้

* เชื่อมต่อ รับ และส่งข้อมูลผ่าน WebSocket จากเซิร์ฟเวอร์
* แสดงผลข้อมูล/ผลการตรวจจับท่าทางที่ส่งมาจาก Server
* เป็นส่วนที่ติดต่อกับผู้ใช้ (UI) สำหรับการตั้งค่าต่าง ๆ ที่เกี่ยวกับระบบโยยรวม

**รูป 3.3 โครงสร้างระบบ**

API

Send

MCU

Web App

Sensors

Server

Socket.IO

Run

(Threads)

Config

ML

Hardware

Web App

Sensors sample data

Server

- Visualize data

- Display result

Read data from Sensor

Process raw data

Classify gesture

**รูป 3.4 ขั้นตอนการทำงาน – การตรวจจับท่าทาง**

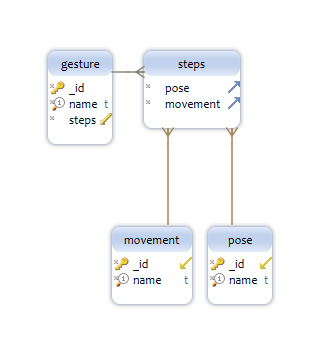
**3.3 การออกแบบฐานข้อมูล**

สำหรับการเก็บข้อมูลท่า จะใช้ Collection ทั้งหมด 4 Collection ประกอบไปด้วย

Gesture: ชุดท่าทาง 1 ชุด ประกอบไปด้วย steps หลายท่าที่ทำต่อเนื่องกัน

Steps: ขั้นตอน 1 ขั้นตอนของการทำท่า ประกอบไปด้วย pose และ movement อย่างละแบบ

Pose: การวางมือ และ Movement: การขยับมือ



**รูป 3.4 แผนภาพฐานข้อมูล**

**3.2 การออกแบบชุดข้อมูล**

ชุดข้อมูลที่จะส่งไปใช้ในการฝึกโมเดลและการตรวจจับท่าทาง จะเป็นชุดของค่าที่จับได้จากเซนเซอร์ติดต่อกัน 20 ค่า ที่แสดงการเปลี่ยนแปลงของค่าเซนเซอร์ติดต่อกันภายในเวลา 20 วินาที จึงทำให้ชุดข้อมูลหนึ่งจะมี 20 ชุดย่อย ซึ่งชุดย่อยจะประกอบไปด้วยค่ามุมจากเซนเซอร์ตัวละ 3 แกน รวมเป็น 18 ค่า ชุดย่อยสามารถแสดงเป็นอาร์เรย์ได้ดังนี้

[ มือ\_x, มือ\_y, มือ\_z , โป้ง\_x, โป้ง\_y, โป้ง\_z, ชี้\_x, ชี้\_y, ชี้\_z, กลาง\_x, กลาง\_y, กลาง\_z, นาง\_x, นาง\_y, นาง\_z, ก้อย\_x, ก้อย\_y, ก้อย\_z ]

ดังนั้นหากนำชุดย่อยมาต่อกัน 20 ตัว จะกลายเป็นชุดข้อมูล 1 ชุด และในการฝึกโมเดล จะใช้ชุดข้อมูลทั้งหมดอย่างน้อย 50 ตัว

**3.4 การออกแบบโมเดล Machine Learning**

3.4.2 ใช้โครงข่ายประสาทเทียม (Neural Network) แบบ สำหรับ

3.4.2 …..

3.4.2 …..

**บทที่ 4**

**การทดลอง**

การทดลองแบ่งได้เป็น 3 ส่วนหลัก ๆ คือ

**4.1 การทดลองเกี่ยวกับฮาร์ดแวร์**

คือการทดลองเกี่ยวกับถุงมือที่ได้ออกแบบมาทั้งสองแบบ ว่าสามารถจับค่าได้แม่นยำและมีปัญหาในการจับค่าหรือไม่

**4.1.1 การทดลองใช้ถุงมือแบบที่ 1**

**4.1.1.1 วิธีการทดลอง**

ประดิษฐ์ถุงมือแบบที่ 1 ขึ้นตามที่ออกแบบไว้ จากนั้นลองอ่านข้อมูลที่ได้จากเซนเซอร์ทุกตัว (Flex Sensor 5 ตัว และ IMU 1 ตัว) สำหรับ Flex Sensor จะมีค่าตั้งแต่ 0-1023 และ IMU จะอ่านเพียงค่าดิบของ Accelerometer 3 แกนและ Gyroscope 3 แกน ทั้งสองมีค่าตั้งแต่ -2000 ถึง 2000 จากนั้นลองทำท่านิ่งเบื้องต้น ได้แก่ท่ากำมือ แบมือ ชูนิ้วแต่ละนิ้ว ชูนิ้วชี้กับนิ้วกลางพร้อมกัน และไขว้นิ้ว

**4.1.1.2 ผลการทดลอง**

ถุงมือไม่มีปัญหาด้านการจับค่า สามารถทำการจับค่าได้ตามปกติ ยกเว้นในบางกรณีดังนี้ 1.) กรณีแบมือจนสุด ทำให้ Flex Sensor เกิดการงอ ถุงมือจะจับค่าได้เกือบเท่าการงอนิ้ว 2.) กรณีที่มีการไขว้นิ้ว ที่มีค่าอยู่นอกแกนที่ Flex Sensor สามารถจับได้ และ 3.) งอนิ้วโป้งไปหลายทิศทาง ซึ่งอยู่นอกแกนที่ Flex Sensor สามารถจับได้เช่นกัน ซึ่งทำให้มีปัญหาในการแยกท่าบางท่า

**4.1.2 การทดลองใช้ถุงมือแบบที่ 2**

**4.1.2.1 วิธีการทดลอง**

ประดิษฐ์ถุงมือแบบที่ 2 ขึ้นตามที่ออกแบบไว้ จากนั้นลองอ่านข้อมูลที่ได้จากเซนเซอร์ทุกตัว (IMU 6 ตัว) สำหรับ IMU ทุกตัว จะอ่านเพียงแค่ค่าดิบของ Accelerometer 3 แกนและ Gyroscope 3 แกน ทั้งสองมีค่าตั้งแต่ -2000 ถึง 2000 แล้วนำมาคำนวณตามสูตรคำนวณหามุมจาก Accelerometer และ Gyroscope ให้กลายเป็นค่ามุม 3 ค่า โดย IMU ที่อยู่หลังมือจะมีค่ามุมระหว่าง 0 ถึง 360 และ IMU ที่อยู่ปลายนิ้วจะนำไปหาความแตกต่างจาก IMU ที่อยู่หลังมือก่อน กลายเป็นค่าความแตกต่างระหว่างช่วง -180 ถึง 180 จากนั้นลองทำท่านิ่งเบื้องต้น ได้แก่ท่ากำมือ แบมือ ชูนิ้วแต่ละนิ้ว ชูนิ้วชี้กับนิ้วกลางพร้อมกัน และไขว้นิ้ว

**4.1.1.2 ผลการทดลอง**

ถุงมือไม่มีปัญหาด้านการจับค่า ท่าที่เคยมีปัญหาในถุงมือแบบที่ 1 มีค่าแตกต่างกันอย่างชัดเจน ทำให้สามารถแยกท่าที่เคยมีปัญหาออกจากกันได้

**4.2 การทดลองเกี่ยวกับ Machine Learning**

คือการทดลองเกี่ยวกับโมเดล Machine Learning ต่าง ๆ ที่ใช้ในการวิเคราะห์ท่าทางมือ โดยการทดลองอัลกอรึธึมการสร้างโมเดลที่แตกต่างกัน

**4.3 การทดลองเกี่ยวกับเว็บแอพพลิเคชั่น**

คือการทดลองพัฒนาเว็บแอพพลิเคชั่นในส่วนต่าง ๆ รวมถึงการเชื่อมต่อ แลกเปลี่ยนข้อมูลกับส่วนอื่น ๆ เพื่อประยุกต์ใช้งานในขั้นตอนต่าง ๆ ของโครงงาน

**4.3.1 การทดลองการส่งค่าผ่าน Socket.IO**

**4.3.1.1 วิธีการทดลอง**

เชื่อมต่อเว็บแอพพลิเคชั่นเข้ากับเซิร์ฟเวอร์ผ่านทาง Socket.IO เมื่อเริ่มเข้าเว็บไซต์ ส่งข้อมูลจำนวน 20 ชุดต่อวินาทีจากเซิร์ฟเวอร์ไปยังเว็บแอพพลิเคชั่น จากนั้นลองกดปุ่มบนเว็บแอพพลิเคชั่นเพื่อส่งข้อมูลล่าสุดทั้ง 20 ชุดกลับไปวิเคราะห์ที่เซิร์ฟเวอร์

**4.3.1.2 ผลการทดลอง**

ทั้งสองฝั่งสามารถส่งข้อมูลหากันได้ทุกครั้งตลอดกระบวนการ

**บรรณานุกรม**

S.S. Fels, G.E. Hinton. “Glove-Talk: a neural network interface between a data-glove and a speech

synthesizer”. [Online] Available: https://ieeexplore.ieee.org/abstract/document/182690/

Rung-Huei Liang, Ming Ouhyoung. “A real-time continuous gesture recognition system for sign \

Language”. [Online] Available: https://ieeexplore.ieee.org/abstract/document/671007/

Ji-Hwan Kim, Nguyen Duc Thang, Tae-Seong Kim. “3-D hand motion tracking and gesture

recognition using a data glove”. [Online] Available: https://ieeexplore.ieee.org/ abstract/document/5221998/

J. Weissmann, R. Salomon. “Gesture recognition for virtual reality applications using data gloves and

neural networks”. https://ieeexplore.ieee.org/abstract/document/832699

Lidia Santos, Nicola Carbonaro, Alessandro Tognetti, José Luis González, Eusebio de la Fuente, Juan

Carlos Fraile, Javier Pérez-Turiel. “Dynamic Gesture Recognition Using a Smart Glove in

Hand-Assisted Laparoscopic Surgery”. [Online] Available: http://www.mdpi.com/2227-7080/6/1/8

Yunhao Ge, Bin Li, Weixin Yan, Yanzheng Zhao. “A real-time gesture prediction system using

neural networks and multimodal fusion based on data glove”. [Online] Available: https://ieeexplore.ieee.org/abstract/document/8377532/

Felipe A. Quirino, Marcelo Romanssini, Rafael R. Dorneles, Enzo H. Weber, and Alessandro Girardi.

“A Gesture Detection Glove for Human-Computer Interaction”. [Online] Available:

http://ieee-cas.org/sites/ieee-cas.org/files/2017-2018-final-report\_r9\_a-gesture-detection-glove.pdf

NanoPI-DUO Hardware spec. [Online] Available: http://wiki.friendlyarm.com/ wiki/index.php/NanoPi\_Duo#Hardware\_Spec