

คู่มือการใช้งาน

1. ติดตั้งอุปกรณ์

ให้อยู่บริเวณปลายเท้าของเตียงนอน

มอเตอร์ซ้ายจะอยู่บริเวณซ้ายมือแล้วมอเตอร์ขวาจะอยู่บริเวณขวามือของผู้ใช้งาน

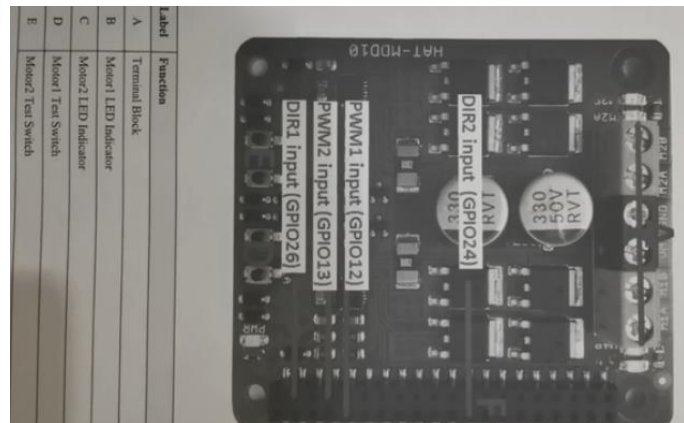
1.1 ต่อมอเตอร์เข้ากับเตียง

(--ต้องถ่ายรูปเตียงมาใส่--)

1.2 ต่อมอเตอร์เข้ากับบอร์ด Raspberry Pi

โดยมีขั้นตอนการทดสอบดังนี้

- ให้ดูที่ภาคเอาพุตของบอร์ด HAT-MDD10 จะมีปุ่มเล็กๆอยู่ 2 คู่ด้วยกัน โดยแต่ละคู่จะมีเลขบอกว่าเป็นมอเตอร์ตัวที่ 1 หรือมอเตอร์ตัวที่ 2



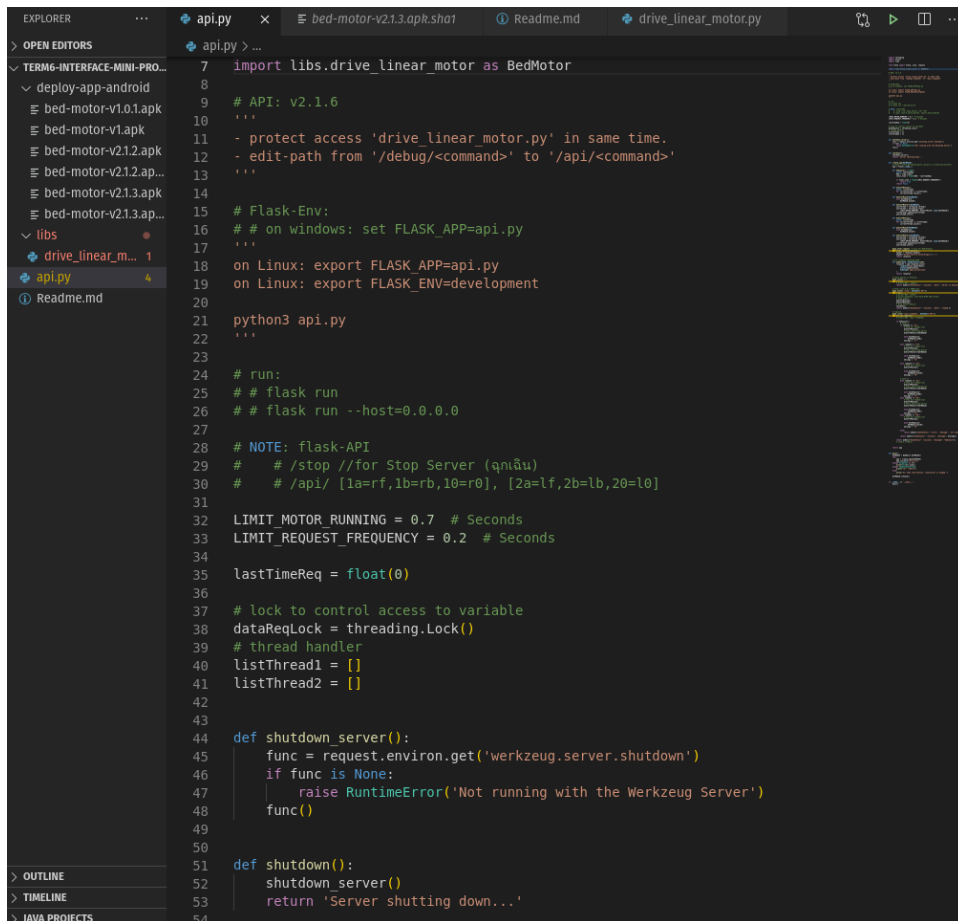
- ให้กดปุ่มจ่ายไฟที่มอเตอร์ตัวที่ 1 ขา a มอเตอร์จะเดินเพียงบริเวณด้านขวาขึ้น
- ให้กดปุ่มจ่ายไฟที่มอเตอร์ตัวที่ 1 ขา b มอเตอร์จะเดินเพียงบริเวณด้านขวาลง
- ให้กดปุ่มจ่ายไฟที่มอเตอร์ตัวที่ 2 ขา a มอเตอร์จะเดินเพียงบริเวณด้านซ้ายขึ้น
- ให้กดปุ่มจ่ายไฟที่มอเตอร์ตัวที่ 2 ขา b มอเตอร์จะเดินเพียงบริเวณด้านซ้ายลง

1.3 เชื่อมต่อบอร์ด Raspberry Pi เท่ากับ WiFi และจด IP ของบอร์ดเพื่อไปใช้ใน การตั้งค่าแอปพลิเคชัน

2. การติดตั้งโปรแกรม

2.1 ดาวน์โหลดโปรแกรม Python ได้ที่

2.1.1 (optional) update GPIO-lib to latest-version



```
7 import libs.drive_linear_motor as BedMotor
8
9 # API: v2.1.6
10 '''
11 - protect access 'drive_linear_motor.py' in same time.
12 - edit-path from '/debug/<command>' to '/api/<command>'
13 '''
14
15 # Flask-Env:
16 # # on windows: set FLASK_APP=api.py
17 '''
18 on Linux: export FLASK_APP=api.py
19 on Linux: export FLASK_ENV=development
20
21 python3 api.py
22 '''
23
24 # run:
25 # # flask run
26 # # flask run --host=0.0.0.0
27
28 # NOTE: flask-API
29 # # /stop //for Stop Server (ฉุกเฉิน)
30 # # /api/ [1a=rf,1b=rb,10=r0], [2a=lf,2b=lb,20=l0]
31
32 LIMIT_MOTOR_RUNNING = 0.7 # Seconds
33 LIMIT_REQUEST_FREQUENCY = 0.2 # Seconds
34
35 lastTimeReq = float(0)
36
37 # lock to control access to variable
38 dataReqLock = threading.Lock()
39 # thread handler
40 listThread1 = []
41 listThread2 = []
42
43
44 def shutdown_server():
45     func = request.environ.get('werkzeug.server.shutdown')
46     if func is None:
47         raise RuntimeError('Not running with the Werkzeug Server')
48     func()
49
50
51 def shutdown():
52     shutdown_server()
53     return 'Server shutting down...'
54
```

<https://github.com/ice-deploy/term6-interface-mini-project-python>

2.2 ดาวน์โหลดแอปพลิเคชันสำหรับ android ได้ที่

<https://github.com/ice-deploy/term6-interface-mini-project-python/blob/master/deploy-app-android/bed-motor-v2.1.3.apk>



bed-motor-v2.1.3.apk

04/04/2021 - 7.45 MB

2.3 ตั้งค่า daemon-service เพื่อให้โปรแกรมเริ่มทำงานทุกครั้งที่ตายหรือเปิดเครื่อง

2.3.1 (optional) ถ้า download-file มาไว้คนละตำแหน่งกันให้เปลี่ยนที่อยู่ของไฟล์ใน Service ->WorkingDirectory

<https://github.com/ice-deploy/term6-interface-mini-project-python#readme>

3. การตั้งค่าแอปพลิเคชัน

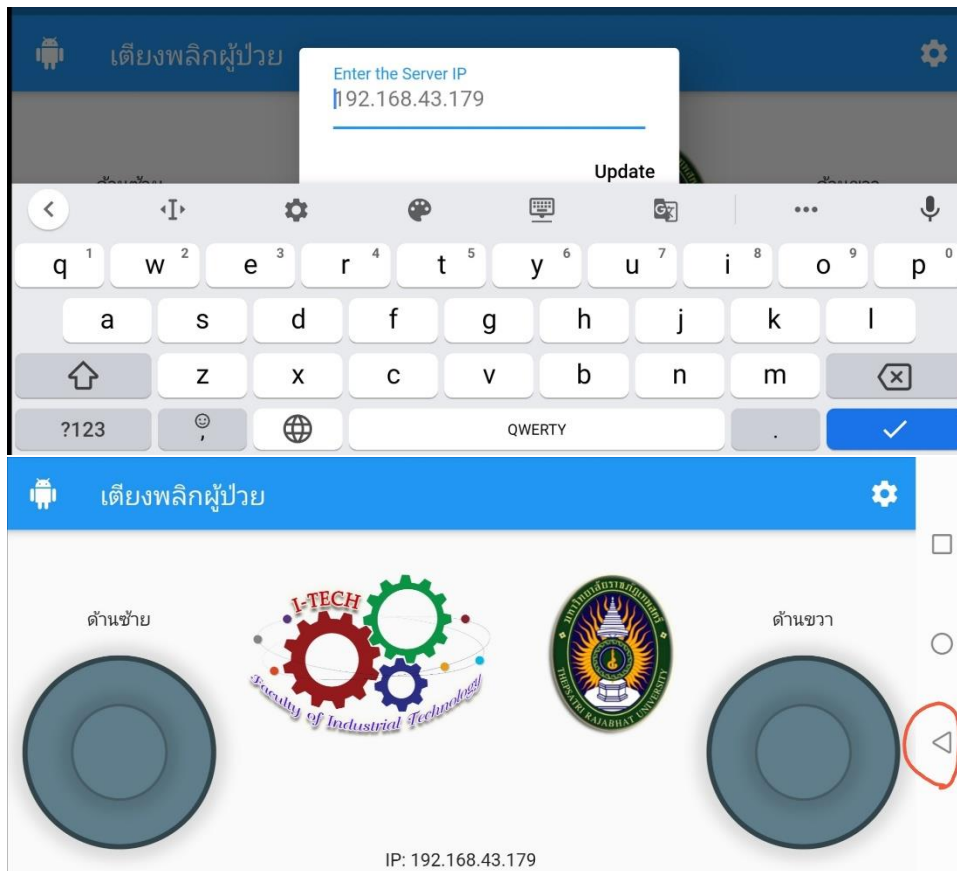
3.1 กดไอคอนรูปเฟืองบริเวณมุมขวาบน



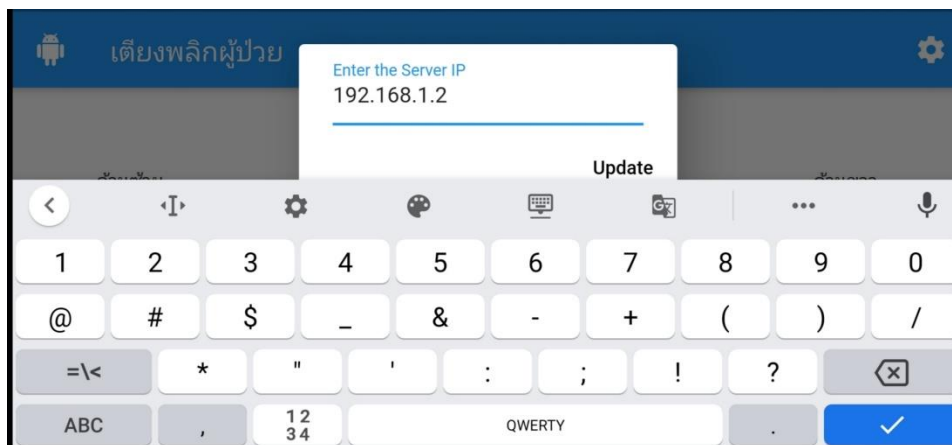
(--ของเดิม-

ใส่เพื่อเทียบสามารถลบออกเมื่อใช้--)

3.2 จะมีหน้าต่างป๊อปอัพขึ้นมา ตัวเลขสีจางๆคืค่าเริ่มต้นหรือค่าล่าสุด ของ IP ของบอร์ด Raspberry Pi หาก IP ไม่เปลี่ยนแปลงสามารถกดยกเลิกได้



3.3 ใส่ IP ใหม่ของบอร์ด Raspberry Pi ลงไป แล้วกด Update เพื่ออัปเดต IP



3.4 หลังจากกดอัปเดตจะมี snackbar สีเขียวแจ้งเตือนมาแสดงว่าได้ทำการตั้งค่า IP เสร็จเรียบร้อยแล้ว หากต้องการปิด snack Bar อย่างรวดเร็วให้กดที่ snackbar แล้วลากลงเพื่อปิด snackbar ทันที



เตียงพลิกผู้ป่วย



ด้านซ้าย



ด้านขวา



Updated IP: 192.168.1.2

4. การใช้งานแอปพลิเคชัน

4.1 เลื่อนจอยแพดด้านซ้าย ขึ้น เพื่อพลิกเตียง ด้านซ้าย ขึ้นไปทางแกนกลางจนได้มุมที่ต้องการ

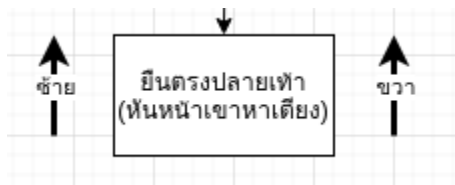
4.2 เลื่อนจอยแพดด้านซ้าย ลง เพื่อพลิกเตียง ด้านซ้าย กลับลงมาที่มุม 0 องศา

4.3 เลื่อนจอยแพดด้านขวา ขึ้น เพื่อพลิกเตียง ด้านขวา ขึ้นไปทางแกนกลางจนได้มุมที่ต้องการ

4.4 เลื่อนจอยแพดด้านขวา ลง เพื่อพลิกเตียง ด้านขวา กลับลงมาที่มุม 0 องศา

(--ใส่รูปหน้าจอตอนโยก แล้วได้มุมพลิกผู้ป่วยแบบต่างๆ--)

(--ทิศทางการย่น--)



อธิบายหลักการทำงาน

Diagram การใช้งาน

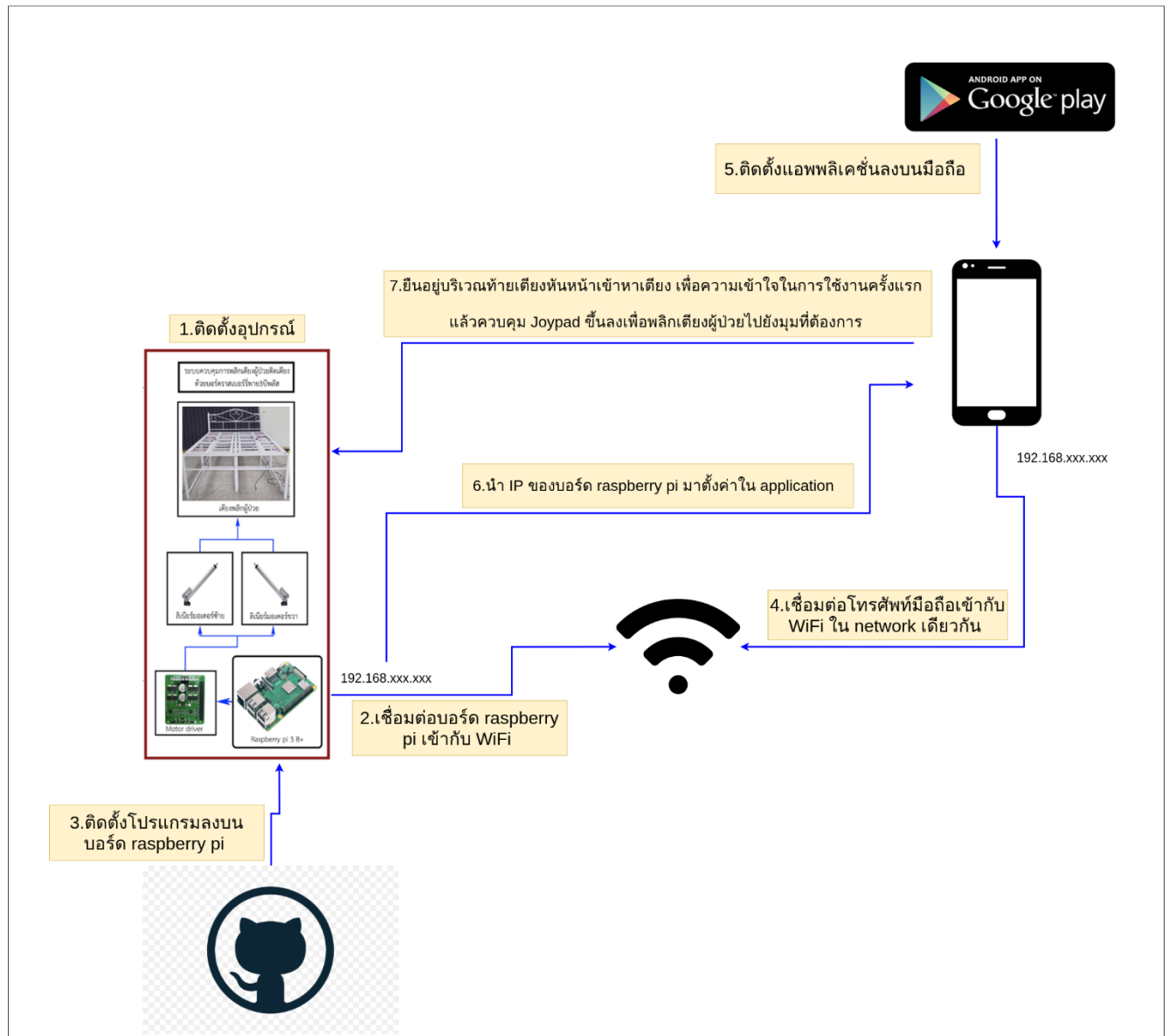


Diagram หลักการทำงานโดยรวม

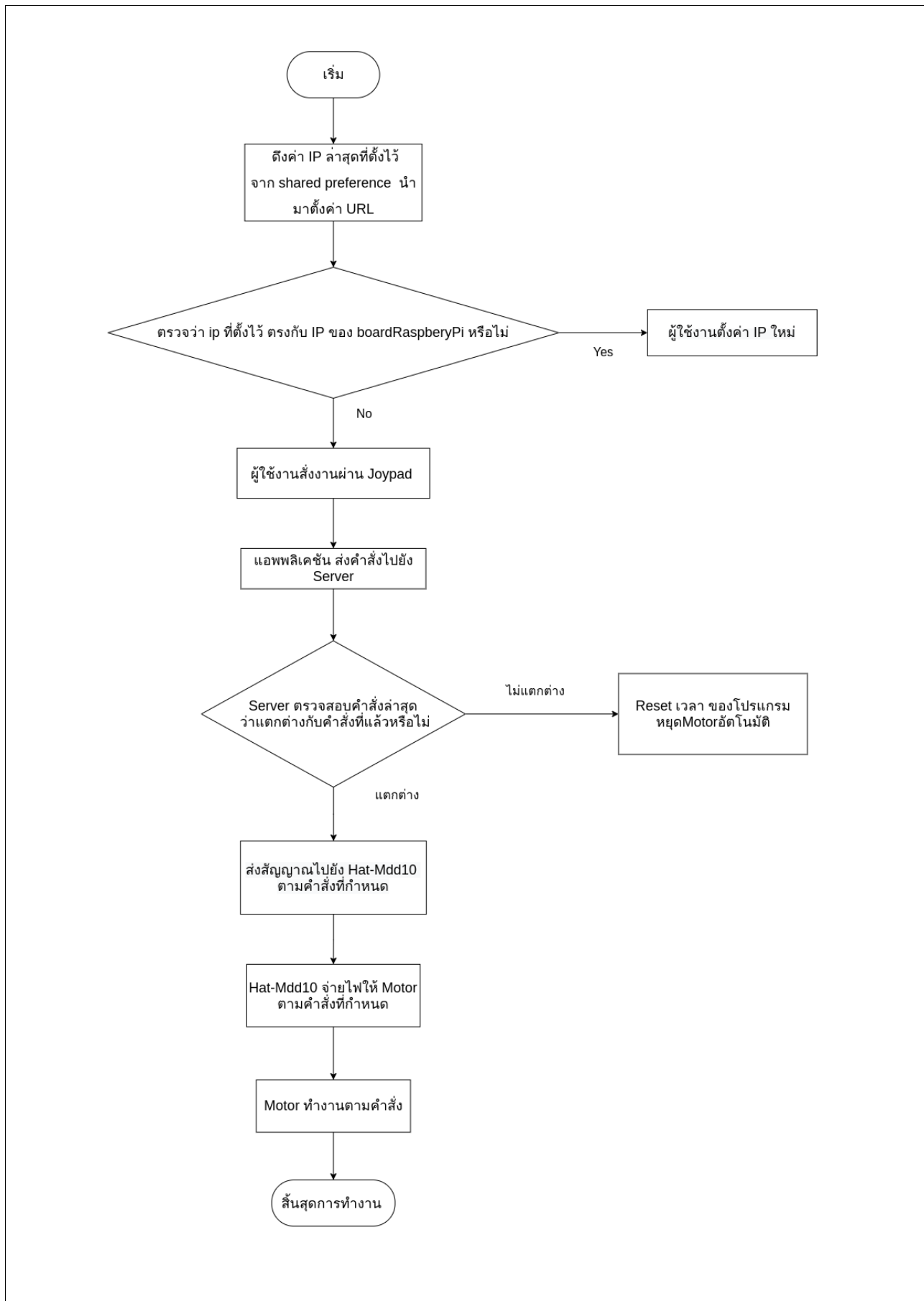


Diagram หลักการทำงานแต่ละส่วน

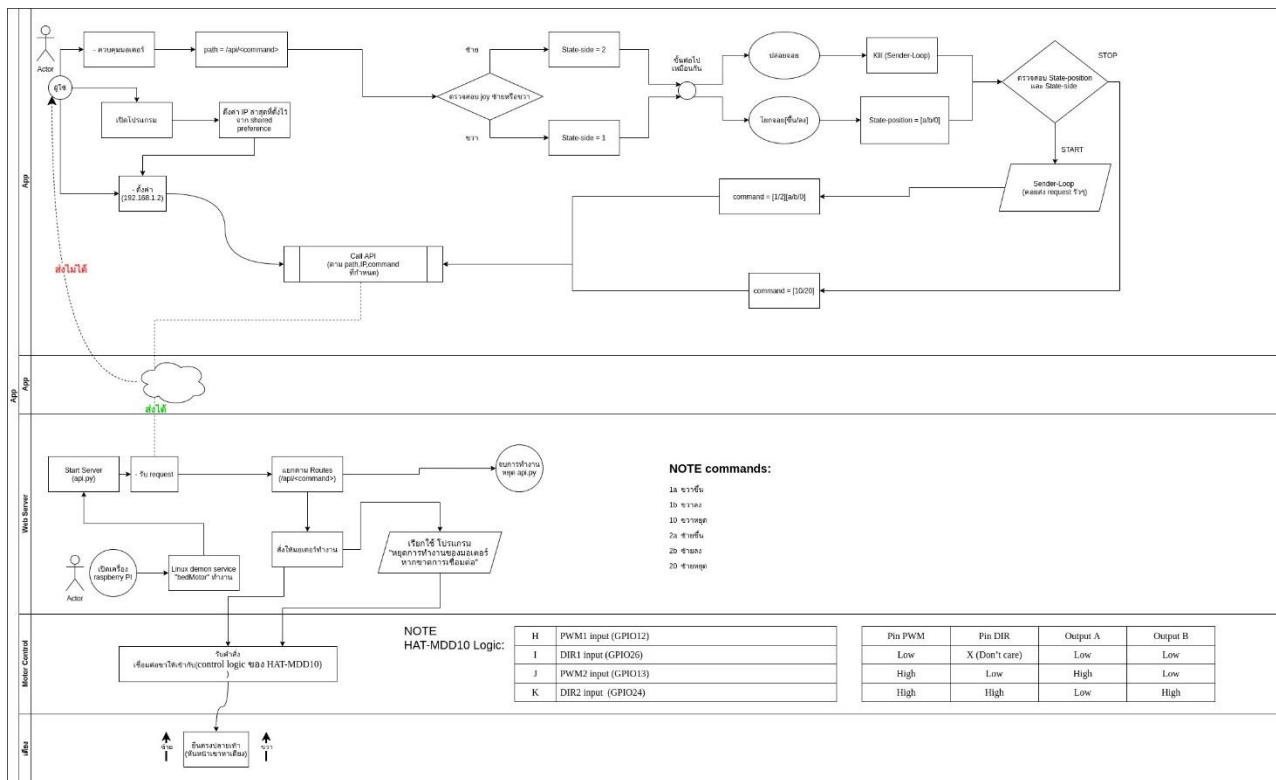


Diagram การทำงานของ App

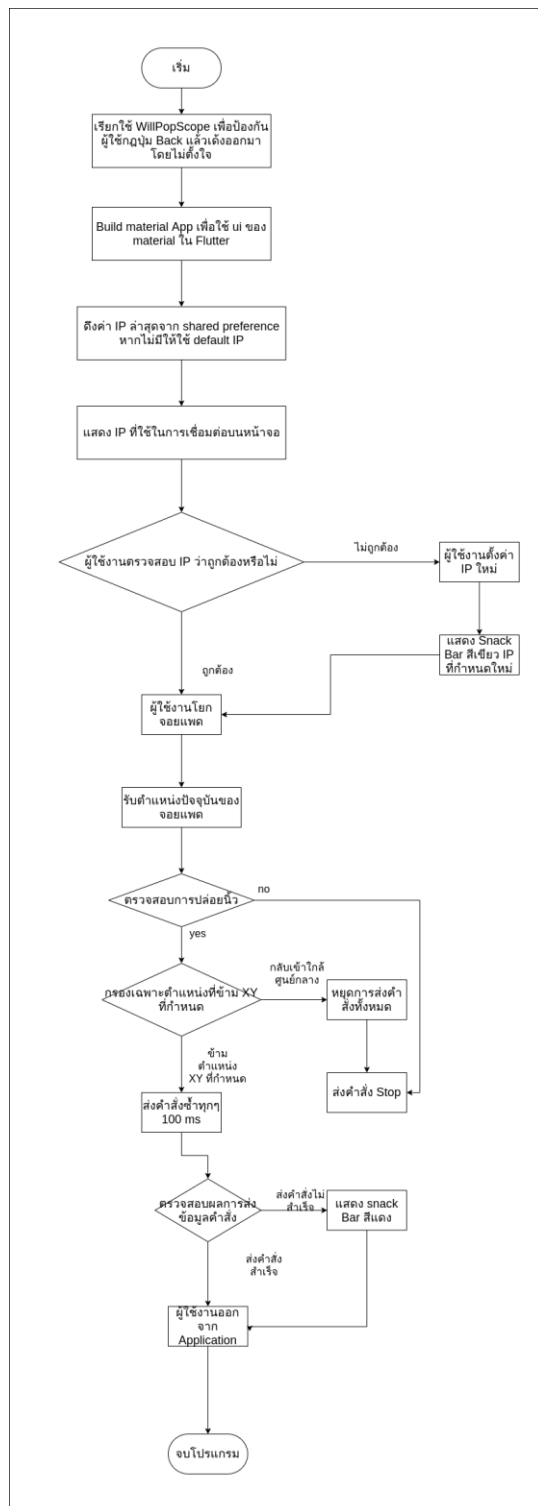


Diagram การทำงานของ Server

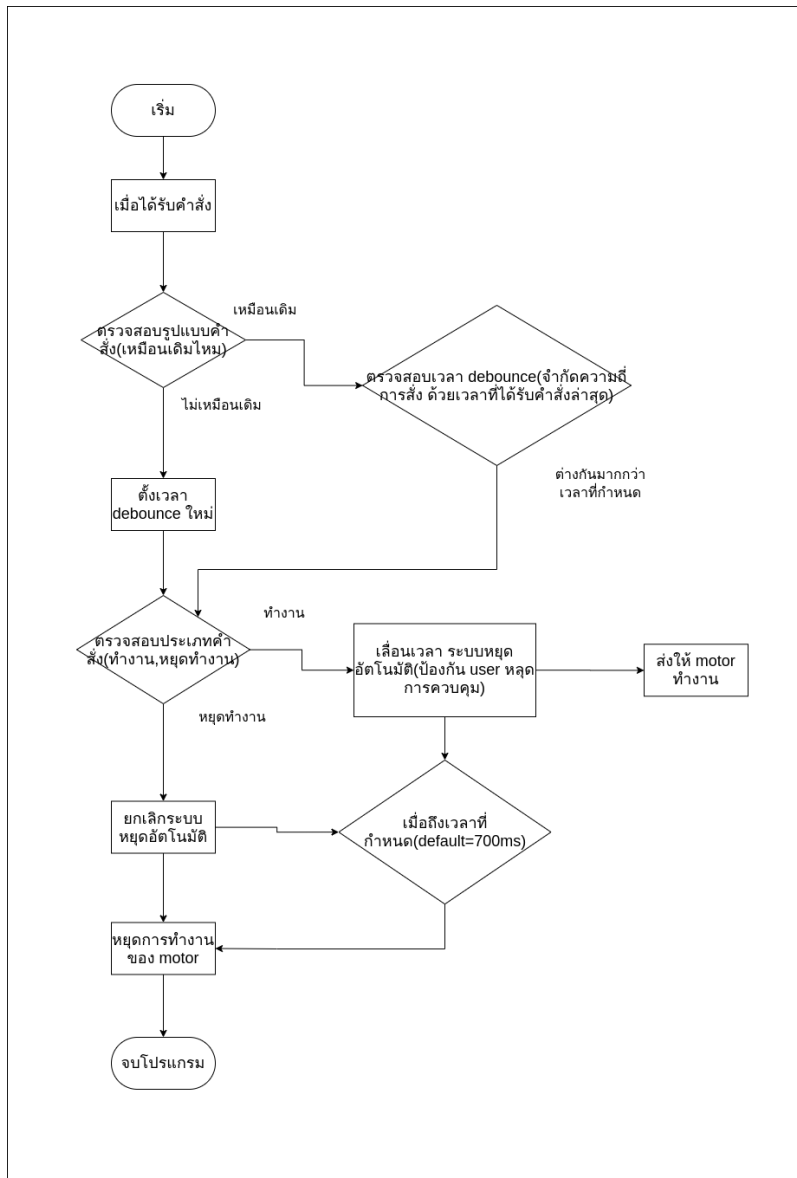
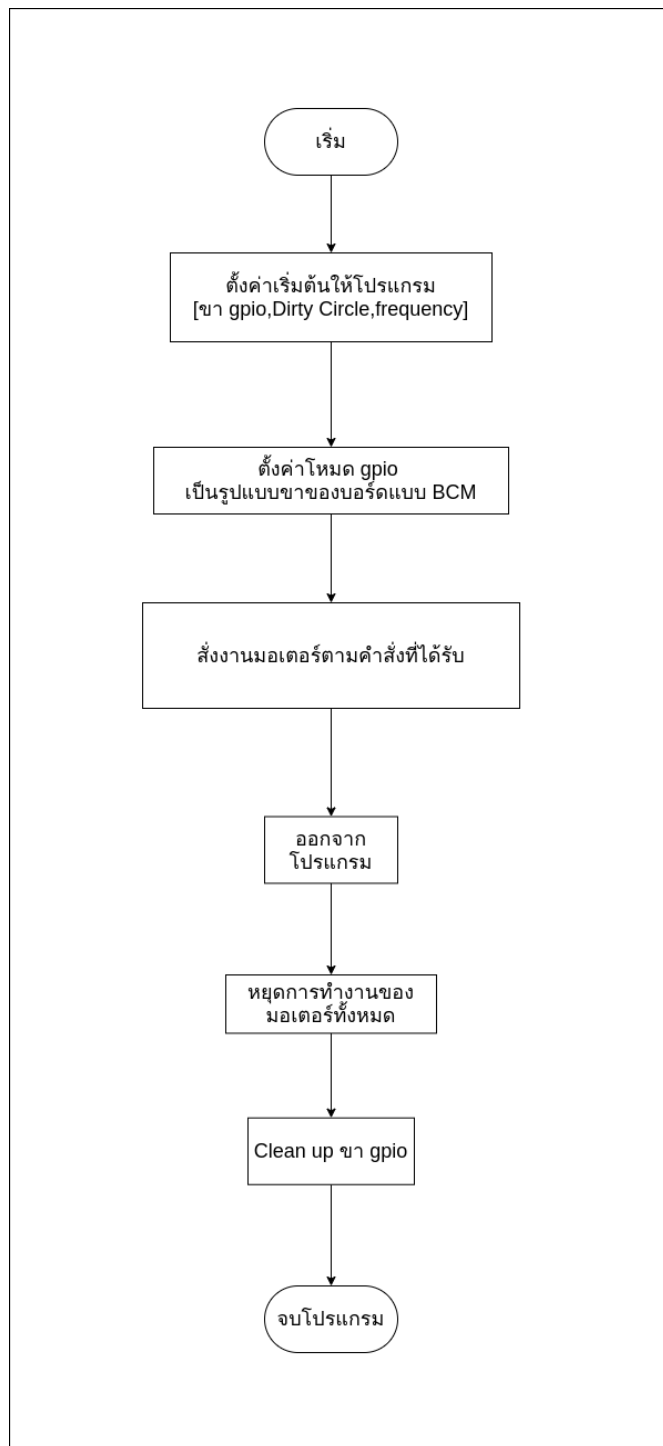


Diagram การทำงานของ Motor



การพัฒนาหรือแก้ไขปรับปรุงโปรแกรม

คู่มือการตั้งค่าขา เพื่อเชื่อมกับบอร์ด HAT-MDD10

<https://github.com/ice-deploy/term6-interface-mini-project--for-extend/blob/master/docs/datasheets/2x10a-motor-driver-hat-for-raspberry-pi-Users-Manual.pdf>

NOTE HAT-MDD10 Logic:

PWM1 input (GPIO12)
DIR1 input (GPIO26)
PWM2 input (GPIO13)
DIR2 input (GPIO24)

Pin PWM	Pin DIR	Output A	Output B
Low	X (Don't care)	Low	Low
High	Low	High	Low
High	High	Low	High

ปัจจุบันโปรแกรมเป็นเวอร์ชัน 2.1.x ที่เน้นความเรียบง่าย

ความหมายของ Commands:

1a ขวาคืบ

1b ขวาลง

10 ขวาทหยุด

2a ซ้ายคืบ

2b ช้ายลง

20 ช้ายหยุด

สามารถดูข้อมูลเพิ่มเติมได้ที่

<https://github.com/ice-deploy/term6-interface-mini-project-python>

ยังมีโปรแกรมเวอร์ชัน 2.2.x ที่มีการปรับปรุงเพิ่ม เช่น

- เปลี่ยน Routes ใหม่
- แสดงค่า Ping
- รองรับ version-web
- เก็บสถานะล่าสุดของ Motor ช้ายและขวา

R = right

L = left

U = up

D = down

0 = stop

00 = stop[left,right]

ru

rd

r0

lu

ld

l0

00 #add stop-all

สามารถดูข้อมูลเพิ่มเติมได้ที่:

<https://github.com/ice-deploy/term6-interface-mini-project--for-extend/tree/master/src/python/api-v2.2>

<https://github.com/ice-deploy/term6-interface-mini-project--for-extend/tree/master/src/flutterApp/with-http>

<https://github.com/ice-deploy/term6-interface-mini-project--for-extend/tree/master/src/web-src/vue-src-v2.2>

และโปรแกรมเวอร์ชัน 3.0.x ที่เปลี่ยนไปใช้ WebSocket สามารถดูข้อมูลเพิ่มเติมได้ที่:

<https://github.com/ice-deploy/term6-interface-mini-project--for-extend/tree/master/src/python/api-v3>

<https://github.com/ice-deploy/term6-interface-mini-project--for-extend/tree/master/src/flutterApp/with-socket>

<https://github.com/ice-deploy/term6-interface-mini-project--for-extend/tree/master/src/web-src/vue-src-v3>