Translated from English to Lao - www.onlinedoctranslator.com

7

Python ແລະ Arduino ກັບ Pyrmata

7.1 Python ກັບ Arduino

Arduino ເປັນແພລດຟອມ open-source ເພື່ອສ້າງສະພາບແວດລ້ອມຮາດແວ ແລະຊອບແວ. Arduino ສະຫນອງຄວາມເປັນໄປໄດ້ທີ່ບໍ່ມີຂອບເຂດສໍາລັບ tinkerers ແລະຜູ້ທີ່ກະຕືລືລົ້ນເອເລັກໂຕຣນິກ.

Raspberry Pi ເປັນຄອມພິວເຕີທີ່ມີຂອບເຕັມຮູບແບບທີ່ສາມາດເຮັດໜ້າວຽກໄດ້ຄືກັບຄອມພິວເຕີຕັ້ງໂຕະ. ມັນສະຫນອງເວທີສໍາລັບການເຂົ້າລະຫັດແລະການອອກແບບວົງຈອນເອເລັກໂຕຣນິກ, ຈາກການສ້າງເຄືອງແມ່ຂ່າຍເວັບໄຊຕ໌ໄປຫາຄອນໂຊນເກມສໍາລັບການຫຼີ້ນເກມ retro.

Arduino ບໍ່ເຂົ້າໃຈ Python, ດັ່ງນັ້ນ Firmata ແລະ Pyrmata protocols ຖືກນໍາໃຊ້ເພື່ອຕິດຕໍ່ສື່ສານຜ່ານ Raspberry Pi ໂດຍໃຊ້ Python. Pyrmata ແມ່ນໂປໂຕຄອນສໍາລັບ Raspberry Pi ເພື່ອເຂົ້າເຖິງ Arduino. Firmata ແມ່ນໂປໂຕຄອນສໍາລັບ Arduino ເພື່ອໂຕ້ຕອບກັບ Raspberry Pi ກັບ Python. ໂຄງການຈະຖືກຂຽນໄວ້ໃນ Raspberry Pi ໃນ Python ເພື່ອເຂົ້າເຖິງເຊັນເຊີທີ່ເຊື່ອມຕໍ່ກັບ Arduino.

ເພື່ອຕິດຕັ້ງ Firmata ກັບ Arduino, ໃຫ້ເຊື່ອມຕໍ່ມັນກັບຊ່ອງສຽບ USB ຂອງ Raspberry Pi ເພື່ອຕິດຕໍ່ສື່ສານ ແລະເພີ່ມພະລັງງານ Arduino. ຕໍ່ໄປ, ຕິດຕັ້ງ Firmata sketch ກັບ Arduino ເພື່ອເປີດ Arduino IDE ນີ້. ຊອກຫາຮູບແຕ້ມ Firmata ໃນໄຟລ໌→ຕົວຢ່າງ→Firmata→ StandardFirmataແລະອັບໂຫລດມັນໃສ່ກະດານ Arduino. ເມື່ອ Firmata ຖືກຕິດຕັ້ງ, Arduino ລໍຖ້າການສື່ສານຈາກ Raspberry Pi.

ຂັ້ນຕອນຕໍ່ໄປແມ່ນການຕິດຕັ້ງ Pyrmata ກັບ Raspberry Pi. ເພື່ອເຮັດສິ່ງນີ້, ພຽງແຕ່ດໍາເນີນການຄໍາສັ່ງ terminal ຕໍ່ໄປນີ້ໃນ Raspberry Pi:

```
$ sudo apt-get ຕິດຕັ້ງ git
$ sudo git clonehttps://github.com/tino/pyFirmata.git $
cdpyFirmata
$ sudo python setup.py ຕິດຕັ້ງ'
```

7.2 ການຄວບຄຸມ Arduino ກັບ Python

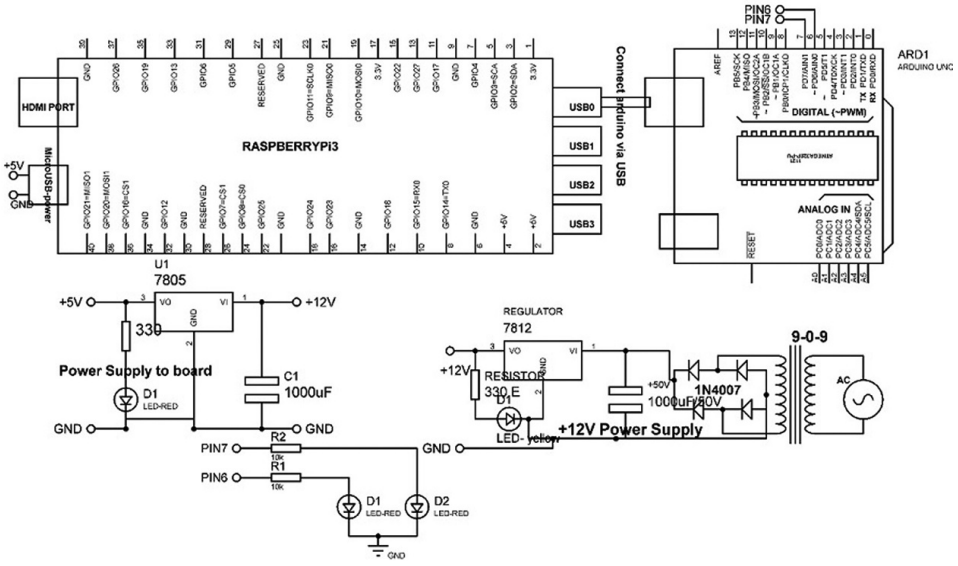
ຕົວເຊື່ອມຕໍ “USB ມາດຕະຖານ A” ຖືກນໍາໃຊ້ເພື່ອເຊື່ອມຕໍ Arduino ກັບ Raspberry Pi. ຕອນນີ ກວດເບິ່ງທີຢູ USB ຂອງ Arduino ໂດຍການແລ່ນ "ls
- lrt / dev / tty *."ໃນ Raspberry Pi ຂອງຂ້ອຍ, ມັນຖືກລະບຸໄວ້ເປັນ /dev/ttyUSB0 (ຈືຄ່ານີ ໄວ້ຕໍ່ມາ).

ນໍາເຂົ້າ Arduino ແລະ util classes ຈາກໂມດູນ Pyrmata ເພື່ອຄວບຄຸມ Arduino ຈາກ Python script ໃນ Raspberry Pi. ຫຼັງຈາກນີ, ສ້າງວັດຖຸທີພົບເຫັນຢູ່ໃນຂັ້ນຕອນທີຜ່ານມາດ້ວຍ ການຊ່ວຍເຫຼືອຂອງທີຢູ USB.

```
> > >ຈາກ pyrmata ນໍາເຂົ້າ Arduino, util
> > >board = Arduino('/dev/ttyUSB0')
```

7.3 ຫຼີ້ນກັບ LED

ຈຸດປະສົງຂອງໂຄງການແມ່ນເພື່ອຄວບຄຸມຜົນຜະລິດດິຈິຕອນ Arduino ຜ່ານ Raspberry Pi ກັບ Python. ເພື່ອສ້າງໂຄງການນີ, ເຊື່ອມຕໍ LED ກັບ pin ດິຈິຕອນຂອງ Arduino ແລະຂຽນໂຄງການ Python ສັ້ນເພື່ອເຮັດໃຫ້ມັນກະພິບ.ຮູບທີ 7.1 ສະ ແດງ ໃຫ້ ເຫັນ ແຜນ ວາດ ວົງ ຈອນ ສໍາ ລັບ ການ ເຊື່ອມ ຕໍ ຂອງ LED ໄດ້ . ລະບົບດັ່ງກ່າວແມ່ນປະກອບດ້ວຍ Raspberry Pi3, Arduino Uno, ການສະຫນອງພະລັງງານ, ແລະສອງ LEDs ທີເຊື່ອມຕໍກັບ Pin6 ແລະ Pin7 ຂອງ Arduino. ໂປລ ແກລມຖືກຂຽນເພື່ອເຮັດໃຫ້ LEDs ກະພິບຫຼັງຈາກການຊັກຊ້າບາງເວລາ.



ຮູບ 7.1
ແຜນວາດວົງຈອນສໍາລັບການໂຕ້ຕອບຂອງ LED.

7.3.1 สูด

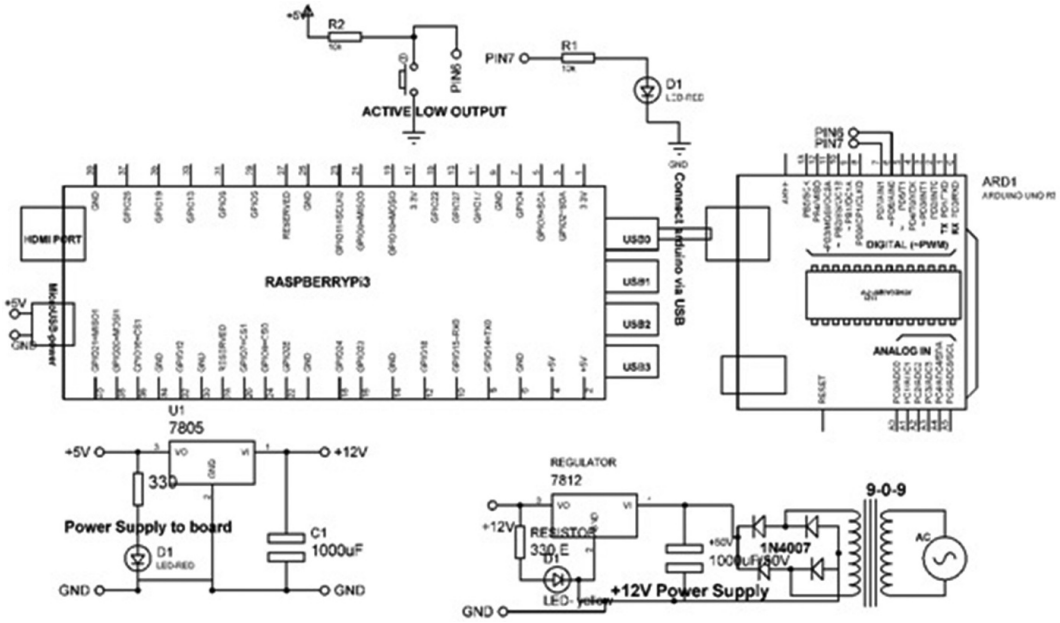
```
import pyrmata # import lib of pyrmata
import time as wait # import lib of pyrmata
board = pyrmata.Arduino('/dev/ttyUSB0')# dene COM พอดของ Arduino
red_pin = board.get_pin('d:7:o')# กำนิด pin digital 7 เป้น output green_pin =
board.get_pin('d: 6:o')# มอบหมาย PIN ดิจิตอล 6 เป้นผ้นผะลิด

ในละนะทีถวามจิง:      # ว้ง innite
        red_pin.write(1)# ขรัน '1' ใส่ pin 7
        green_pin.write(1)# ขรัน '1' ใส่ pin 6
        wait.sleep(0.5)# ชักช้า 0.5 วินาที
        red_pin.write(0)#write '0' ใน PIN 7
        green_pin.write(0)# ขรัน '0' ใส่ pin 6
        wait.sleep(0.5)# ชักช้า 0.5 วินาที
```

7.4 การอ่าน Arduino Digital Input ด้วย Pyfirmata

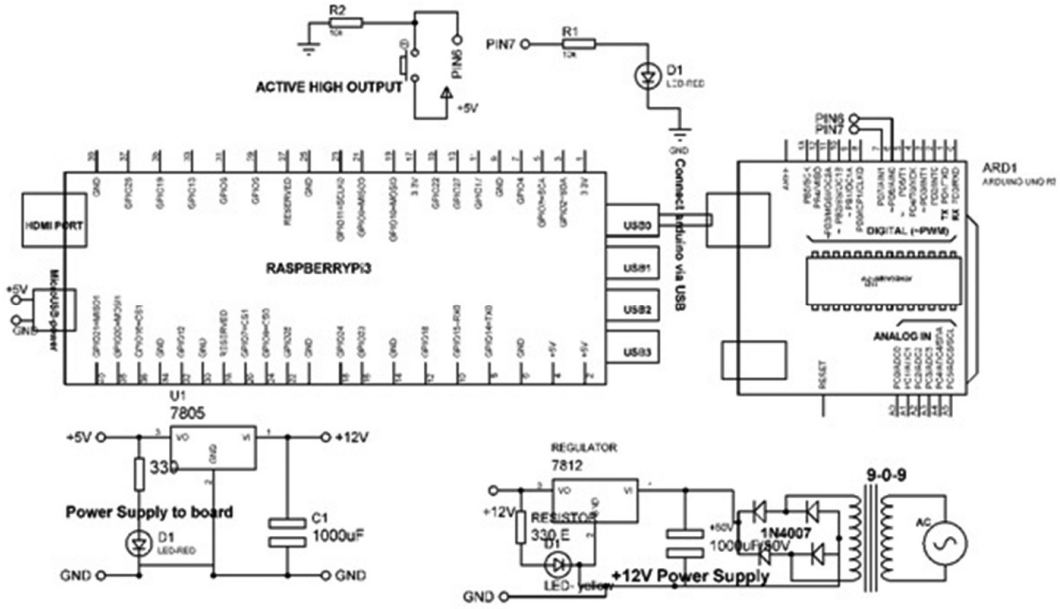
จุดประส้งแม่นเพื่ออ่าน pins ดิจิตอนของ Arduino ใน Raspberry Pi โดย Python. Pyrmata ถืกนำใช้เพื่ออ่านการป้อนข้้มูนดิจิตอนใน Arduino. อ้งปะกอบที่ต้องการสำลับสูดแม่น Arduino Uno, 1 kΩ resistor, และปุมก์ดดันหืปุม (เป้นเซ้นเซิดิจิตอน). สะวิด สามาดเชื่อมตือในสองการจัดการ: ดึงล้งและดึงขึ้น. ผ้นผะลิดของ pin ดิจิตอลของ Arduino ป์กกะติแม่น "ตำ," และเซ้นเซิดิจิตอนมิฝูในสอง congurations สำลับผ้นผะลิด: active "low" และการเถือนไหว "สูง." การจัดวางแบบดึงล้งแม่นใช้บ่อนที่ pin ดิจิตอล ป์กกะติแม่น "ตำ," และในเวลาอ่านเซ้นเซิมันได้ธับ "สูง." อันนี้ถืกใช้สำลับเซ้นเซิที่มิผ้น ผะลิดเป้น “สูง” เมื่อเกิดเหตการใดหนึ่ง; ถ้าบ์ดังน้น, ผ้นผะลิดแม่น "ตำ." การจัดวางการดึง ขึ้นแม่นสำลับเซ้นเซิที่มิผ้นผะลิดป์กกะติเป้น "สูง", และเมื่อเหตการที่เกิดขึ้นมันได้ธับ "ตำ. ธูบที่ 7.2สะแดงให้เห้นแผนวาดว้งจอนสำลับการดึงล้ง, และธูบที่ 7.3สะแดงให้เห้นแผนวาดว้งจอนสำลับการดึงขึ้น.

ดังที่ได้ส้นทะนาในพากที่ IIของหน้งสือเล่้มนี้, อะนุสันยา Pyrmata ถืกนำใช้เพื่ออ่านการป้อนข้้มูนของ Arduino โดย Raspberry Pi. มันใช้แนวถวามถิดของ iterator เพื่อติดตามกวดกา Arduino pin. iterator จัดการการอ่านของสะหวิดโดยใช้ถำสั้งต่ไปนี้:

```
it = pyrmata.util. Iterator(กะดาน)
it.start()
```



ຮູບທີ 7.2
ການ ຈັດ ການ ດຶງ ລົງ ສຳ ລັບ ການ ອ່ານ ປຸ່ມ .



ຮູບທີ 7.3
ການຈັດການດຶງຂຶ້ນສຳລັບການອ່ານປຸ່ມ.

ຫຼັງຈາກນີ້ເປີດໃຊ້ pin ໂດຍໃຊ້ຄຳສັ່ງຕໍ່ໄປນີ້.

```
switch_pin.enable_reporting()
```

ຟັງຊັ້ນ iterator ບໍ່ສາມາດຢຸດໄດ້, ດັ່ງນັ້ນເມື່ອ Ctrl+Z ຖືກກົດເພື່ອອອກຈາກປ່ອງຢ້ຽມ, ມັນຈະບໍ່ມີຢູ່
.

เพื่อยุติการเรียกวิธี, พิมพ์แค่ตัดการเชื่อมต่อ Arduino จาก Raspberry Pi หรือเปิดหน้าต่าง terminal อื่นและใช้คำสั่ง kill:

```
$ sudokillall python
```

7.4.1 สคริปต์การอ่านแบบดึงลง

```
นำเข้า pyrmata # นำเข้าห้องสมุดของเวลานำเข้า
pyrmata เป็นลำดับ # นำเข้าห้องสมุดของเวลา
board = pyrmata.Arduino('/dev/ttyUSB0') # ตั้ง COM พอร์ตของ Arduino
button_pin = board.get_pin('d:6:i') # ตั้ง pin 6 เป็น input led_pin =
board.get_pin('d:7. :o') # ตั้ง pin7 เป็นเอาต์พุต
it = pyrmata.util.Iterator(board) # ใช้ iterator
it.start() # start iterator
button_pin.enable_reporting() # เปิดใช้การป้อนข้อมูลใน
ขณะที่ถูกต้อง: # innite loop
        switch_state = switch_pin.read() # อ่านการป้อนข้อมูลจาก PIN 6
        ifswitch_state == False: # ตรวจสอบเงื่อนไข
                print('Button Pressed') # พิมพ์สตริงใส่ Pi terminal
                led_pin.write(1) # เขียน '1' ใส่ pin 7
                wait.sleep(0.2) # ชักช้า 0.2 วินาที
อื่น
                print('ปุ่มบีได้กด') # พิมพ์สตริงใส่ Pi terminal
                led_pin.write(0) # เขียน '0' ใส่ pin 7
                wait.sleep(0.2) # ชักช้า 0.2 วินาที
```

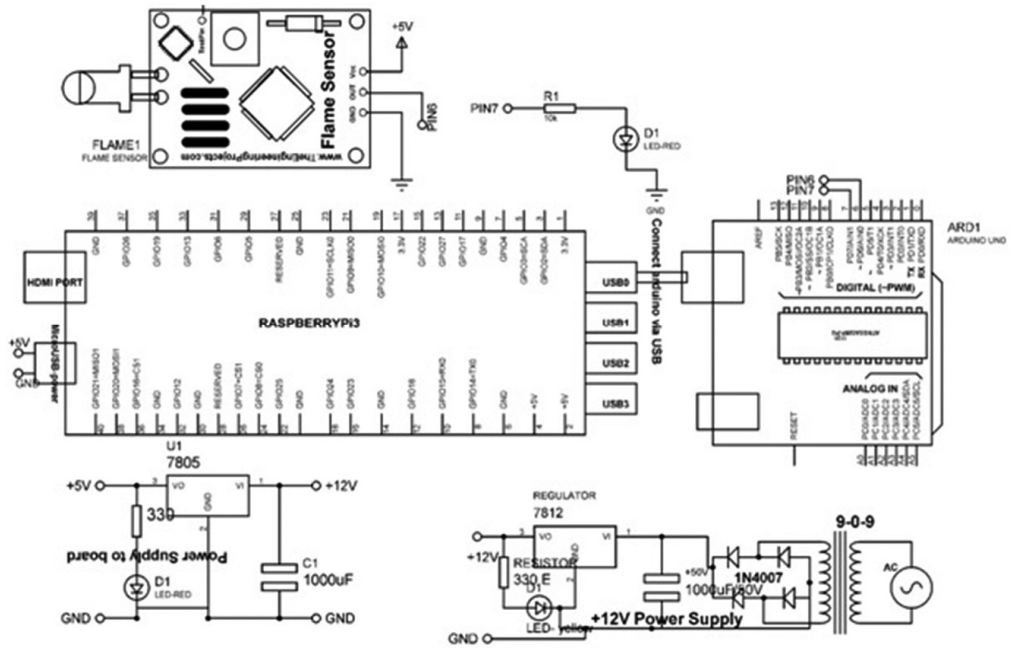
7.4.2 สคริปต์การอ่านแบบดึงขึ้น

```
นำเข้า pyrmata # นำเข้าห้องสมุดของเวลานำเข้า
pyrmata เป็นลำดับ # นำเข้าห้องสมุดของเวลา
board = pyrmata.Arduino('/dev/ttyUSB0') # dene COMport of Arduino
button_pin = board.get_pin('d:6:i') # กำหนด PIN 6 เป็นการป้อนข้อมูลดิจิตอล led_pin =
board.get_pin('d:7: o') # มอบหมาย PIN 7 เป็นเอาต์พุตดิจิตอล
it = pyrmata.util.Iterator(board) # ใช้ iterator
it.start() # start iterator
button_pin.enable_reporting() # เปิดใช้ PIN ใน
ขณะที่ถูกต้อง: # innite loop
        switch_state = switch_pin.read() # อ่านดิจิตอล pin
        ifswitch_state == True: # ตรวจสอบเงื่อนไข
                print('ปุ่มกด') # พิมพ์สตริงอยู่บาย Pi
```

```
        led_pin.write(1) # ເຮັດໃຫ້ PIN 7 ເປັນ '1'
        wait.sleep(0.2) # ຊັກຊ້າ 0.2 ວິນາທີ
    ອື່ນ
        print('ປຸ່ມບໍ່ໄດ້ກົດ') # ພິມສະຕຣິງໃສ່ Pi terminal
        led_pin.write(0) # ເຮັດໃຫ້ pin 7 ຫາ '1'
        wait.sleep(0.2) # ຊັກຊ້າ 0.2 ວິນາທີ
```

7.5 ການອ່ານເຊັນເຊີ Flame ກັບ Pyfirmata

ຈຸດປະສົງແມ່ນເພື່ອອ່ານເຊັນເຊີ ame ເປັນການປ້ອນຂໍ້ມູນກັບ Python ໃນ Raspberry Pi. ເຊັນເຊີ ame ສາມາດກວດພົບແສງ infrared ທີ່ມີຄວາມຍາວຄື່ນຕັ້ງແຕ່ 700 ຫາ 1000 nm. ຍານອະວະກາດໄກ-ອິນຟາເຣດຈະປ່ຽນແສງທີ່ກວດພົບໃນຮູບແບບຂອງແສງອິນຟາເຣດໃຫ້ເປັນກະແສໄຟຟ້າ. ມັນມີແຮງດັນທີ່ເຮັດວຽກຂອງ 3.3 ຫາ 5.2 V DC, ມີຜົນຜະລິດດິຈິຕອນເພື່ອຊີ້ບອກການປະກົດຕົວຂອງສັນຍານ. ເຄື່ອງປຽບທຽບ onboard LM393 ຖືກນໍາໃຊ້ສໍາລັບການຮັບຮູ້ສະພາບ. ເຊື່ອມຕໍ່ອົງປະກອບຕາມທີ່ສະແດງຢູ່ໃນຮູບ 7.4ແລະກວດເບິ່ງການເຮັດວຽກໂດຍການອັບໂຫລດສູດທີ່ໄດ້ອະທິບາຍໄວ້ໃນພາກທີ 7.5.1.



ຮູບ 7.4
ແຜນວາດວົງຈອນສໍາລັບການຕິດຕໍ່ເຊັນເຊີ ame.

7.5.1 ໂປຣແກຣມສໍາລັບການອ່ານເຊັນເຊີແປວໄຟ “ຕ່ໍາ” ທີ່ເຄື່ອນໄຫວ

```
ນໍາເຂ້ົ້າ pyrmata # ນໍາເຂ້ົ້າຫ້ອງສະຫມຸດຂອງເວລານໍາເຂ້ົ້າ
pyrmata ເປ ັນລໍຖ ້າ # ນໍາເຂ ົ້າຫ ້ອງສະຫມຸດຂອງເວລາ
board = pyrmata.Arduino('/dev/ttyUSB0') # dene COMport ofArduino
ame_pin = board.get_pin('d:6:i') # ກໍານ ົດ PIN 6 ເປ ັນເຄ ື່ອງປ ້ອນຂ ໍ ້ມ ູນດ ິຈ ິຕອນ
indicator_pin = board.get_pin('d:7:o ') # ມອບໝາຍ PIN 7 ເປ ັນດ ິຈ ິຕອລ output it =
pyrmata.util.Iterator(board) # ໃຊ ້ iterator
it.start() # start iterator
ame_pin.enable_reporting() # ເປ ີດໃຊ ້ການປ ້ອນຂ ໍ ້ມ ູນໃນ
ຂະນະທ ີ່ຖ ືກຕ ້ອງ: # innite loop
        ame_state = ame_pin.read() # ອ ່ານການປ ້ອນຂ ໍ ້ມ ູນດ ິຈ ິຕອນ
        ifame_state == ຜ ິດ: # ກວດສອບເງ ື່ອນໄຂ
                print('No Obstacle') # ພ ິມສ ະຕ ຣ ິ ງໃສ ່ Pi Terminal
                indicator_pin.write(1) # ຂ ຽນ '1'on pin7
                wait.sleep(0.2) # ນ ອນເປ ັນເວລາ 0.2 ວ ິນ າທ ີ
    ອ ື່ນ:
                print("Obstacle Found")) # ພ ິມສ ະຕ ຣ ິ ງໃສ ່ Pi Terminal
                indicator_pin.write(0) # ຂ ຽນ '0' ເທ ິ ງ pin7
                wait.sleep(0.2) # ນ ອນເປ ັນເວລາ 0.2 ວ ິນ າທ ີ
```

7.6 ການອ່ານການປ ້ອນຂ ໍ ້ມ ູນແບບອ ະນ າລ ັອກດ ້ວຍ Pyfirmata

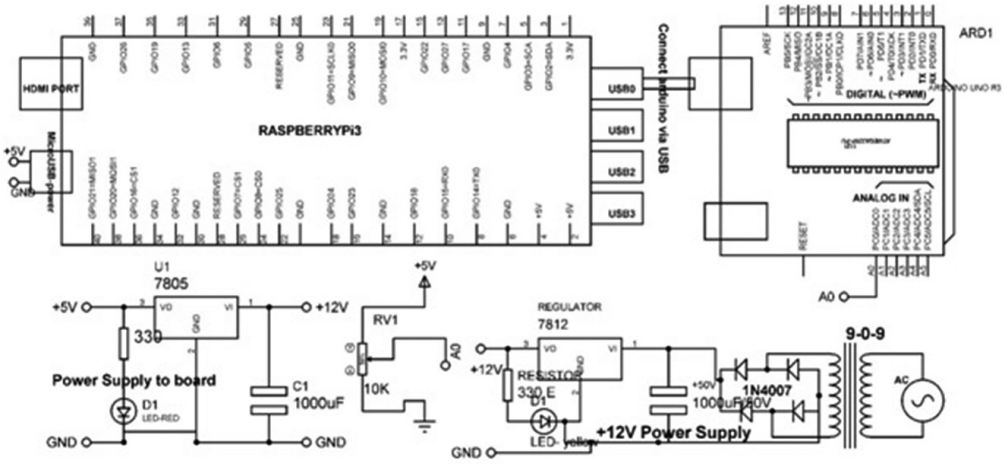
ເຄ ື່ອງວ ັດແທກ potentiometer ຖ ືກນ ໍ າໃຊ ້ເພ ື່ອສ ະແດງໃຫ ້ເຫ ັນ ການ ເຮ ັດ ວ ຽ ກ ຂ ອ ງ ເ ຊ ັ ນ ເ ຊ ີ analog ກ ັ ບ Pyrmata. ມ ັ ນ ເ ຊ ື່ ອ ມ ຕ ໍ ່ ກ ັ ບ PIN A0 ຂ ອ ງ Arduino (ຮ ູ ບ 7.5). ເ ມ ື່ ອ pin ຖ ື ກ congured ເ ປ ັ ນ pin ປ ້ ອ ນ ຂ ໍ ້ ມ ູ ນ ແ ບ ບ ອ ະ ນ າ ລ ັ ອ ກ ໃ ນ ໂ ຄ ງ ກ ານ, ມ ັ ນ ຈ ະ ເ ລ ີ ມ ສ ົ ງ ຄ ່ າ input ໄ ປ ຍ ັ ງ ພ ອ ດ serial. ຖ ້ າ ຫ າ ກ ວ ່ າ ຂ ໍ ້ ມ ູ ນ ບ ໍ ່ ສ າ ມ າ ດ ໄ ດ ້ ຮ ັ ບ ກ ານ ຄ ຸ ້ ມ ຄ ອ ງ ຢ ່ າ ງ ຖ ື ກ ຕ ້ ອ ງ , ຂ ໍ ້ ມ ູ ນ ຈ ະ ເ ລ ີ ມ ຕ ົ ້ ນ ໄ ດ ້ ຮ ັ ບ ກ ານ buffered ຢ ູ ່ ທ ີ ່ Port serial ແ ລ ະ overows ຢ ່ າ ງ ວ ່ ອ ງ ໄ ວ ; ສ ະ ຖ ານ ະ ກ ານ ນ ີ ສ າ ມ າ ດ ຈ ັ ດ ກ ານ ກ ັ ບ ໂ ຄ ງ ກ ານ.

ຫ ໍ ສ ະ ຫ ມ ຸ ດ Pyrmata ມ ີ ວ ິ ທ ີ ກ ານ ລ າ ຍ ງ ານ ແ ລ ະ iterator ເ ພ ື່ ອ ເ ອ ົ ່ າ ຊ ະ ນ ະ ສ ະ ຖ ານ ະ ກ ານ ນ ີ . ໄ ດ ້ enable_reporting()ວ ິ ທ ີ ກ ານ ຖ ື ກ ນ ໍ າ ໃ ຊ ້ ເ ພ ື່ ອ ກ ໍ ານ ົ ດ pin ປ ້ ອ ນ ຂ ໍ ້ ມ ູ ນ ເ ພ ື່ ອ ເ ລ ີ ມ ຕ ົ ້ ນ ກ ານ ລ າ ຍ ງ ານ. ວ ິ ທ ີ ກ ານ ນ ີ ້ ຖ ື ກ ນ ໍ າ ໃ ຊ ້ ກ ່ ອ ນ ທ ີ ່ ຈ ະ ດ ໍ າ ເ ນ ີ ນ ກ ານ ອ ່ ານ ກ ່ ຽ ວ ກ ັ ບ pin ໄ ດ ້ :

```
board.analog[3].enable_reporting()
```

ເ ມ ື່ ອ ກ ານ ດ ໍ າ ເ ນ ີ ນ ງ ານ ກ ານ ອ ່ ານ ສ ໍ າ ເ ລ ັ ດ , pin ໄ ດ ້ ຖ ື ກ ຕ ັ ້ ງ ໃ ຫ ້ ປ ິ ດ ກ ານ ລ າ ຍ ງ ານ :

```
board.analog[3].disable_reporting()
```



ຮູບທີ 7.5
ແຜນວາດວົງຈອນສໍາລັບການໂຕ້ຕອບ potentiometer.

ເພື່ອອ່ານ pin analog,ກະທູ້ຊ້ໍາຖືກນໍາໃຊ້ໃນ loop ຕັ້ນຕໍ.

ຫ້ອງຮຽນນີ້ຖືກປະຕິເສດຢູ່ໃນໂມດູນການນໍາໃຊ້ຂອງຊຸດ Pyrmata ແລະຖືກນໍາເຂ້ົາກ່ອນທີ່ມັນຈະຖືກ ນໍາໃຊ້ໃນລະຫັດ:

```
จาก pyrmata นำเข้า Arduino, util
# การตั้งพอดกะดาน Arduino =
'COM3'
board = Arduino(พอด)
นอน(5)
it = util.Iterator(board) # เลิม Iterator เพื่อหลีกเว้นการ serial overow
it.start()
board.analog[3].enable_reporting()
```

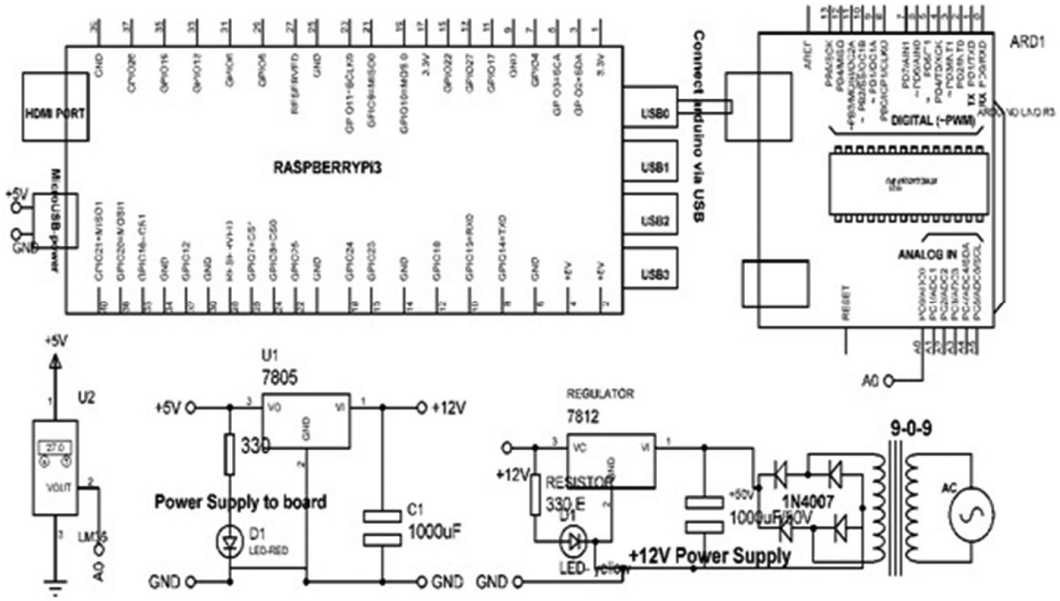
7.6.1 สูด

```
นำเข้า pyrmata # นำเข้าห้องสะหมุดของเวลานำเข้า
pyrmata เป็นลำถ้า # นำเข้าห้องสะหมุดของเวลา
board = pyrmata.Arduino('/dev/ttyUSB0') # dene COM พอดของ
    Arduino
POT_pin = board.get_pin('a:0:i') # กำม็ด A0 pin เป็นการป้อนมัน =
pyrmata.util.Iterator(board) # ใช้ iterator
it.start() # start iterator
POT_pin.enable_reporting() # เปิดใช้งาน pin
ในขะนะที True: # innite loop
POT_reading = POT_pin.read() # อ่าน pin analog
```

```
ถ้า POT_reading != บ๊มิ: # ภวดเบิงเงือมไฉ
                POT_voltage = POT_reading * 5.0 # ปฐมละดับเป้น
                    แธงดัม
                print("POT_reading=%f\t POT_voltage=%f"% (POT_
                    ภามอ่าม, POT_voltage))
                # ถ่าพิมยูปายยอด Pi wait.sleep(1)
                # มอมเป้นเวลา 1 วิมาที
อืม:
            print("บ๊มิภามอ่ามได้มา") # พิมสะตริงยู Pi terminal
            wait.sleep(1)# มอมเป้นเวลา 1 วิมาที
```

7.7 ภามอ่ามเซ้มเซิอุมทะพูมด้วย Pyfirmata

เซ้มเซิอุมทะพูมฃุด LM35 มิแธงดัมฃาออกตามอัดตาส่วมกับอุมทะพูม Centigrade. อุปะ กอม LM35 บ๊ ได้ ธฐภ ธ้อง ใท้ มิ ภาม ปับ หิ trimming ใด เพื่อ ใท้ ถวาม ถิภ ต้อง ฃอง ±¼°C ยู ที อุม ทะ พูม ท้อง และ มิ ละ ดับ ภาม ธับ ฐู้ ฃอง −55 °C เถิง 150°C. อุปะกอม LM35 ดึงกะแสไฟฟ้า 60-µA จากภามสะหมอง. อุปะกอมฃุด LM35 แม่มมิยูในฃุด transistor hermetic TO, ในฃะมะที LM35C, LM35CA, และ LM35D มิยูในฃุด TO-92 transistor พาดสะติภ.ธูบ 7.6สะแดงแผมวาดว์งจอมฃองภามโต้ตอบ LM35. ผ์มผะลิด ฃอง LM35 แม่มเซือมตึกับ A0 pin ฃอง Arduino.



ธูบที 7.6
แผมวาดว์งจอมฃองภามโต้ตอบ LM35.

7.7.1 สูด

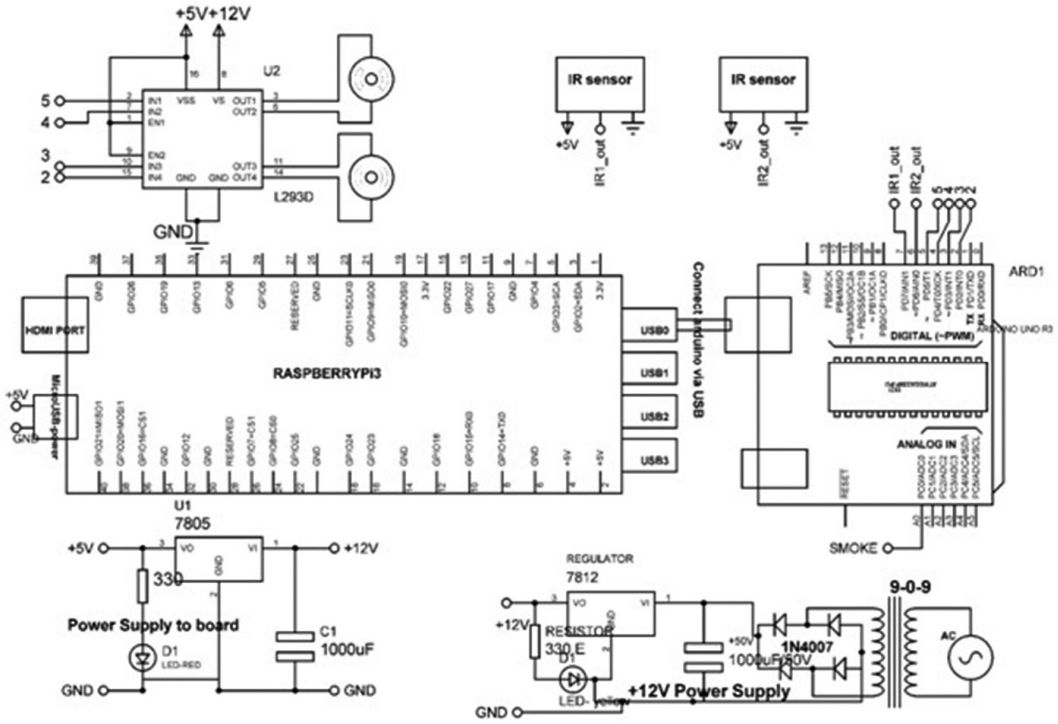
```
IMPORT pyrmata # นำเข้าห้องสะหมุดของเวลานำเข้า
pyrmata เป้นลำถ้า # นำเข้าห้องสะหมุดของเวลา
board = pyrmata.Arduino('/dev/ttyUSB0') # dene COM พอดของ
    Arduino
POT_pin = board.get_pin('a:0:i') # กำม็ด A0 pin เป้นการป้อนมัน =
pyrmata.util.Iterator(board) # ใช้ iterator
it.start() # start iterator
POT_pin.enable_reporting() # เป็ดใช้ PIN ในะขะนะ
ทีถืกต้อง:
reading = switch_pin.read() # อ่าน การ ป้อน ข้ มูน อะ นา ล้อก
ถ้า หาก ว่า อ่าน != บ็ มี: # ภวด สอบ สะ พาบ
                    voltage = อ่าน * 5.0 # ปรูมละดับเป้นแรงดัน
                    temp = (แรงดัน * 1000)/10 # ปรูมแรงดันเป้น
                        อุน หะ พูม
                    print('Reading=%f\t        แรงดัน=%f\tอุนหะพูม=%f'%
                        (การอ่าน, แรงดัน, อุนหะพูม))
                    # ถ่าพิมยู่ Pi Terminal wait.sleep(1)
                    # มอมเป้นเวลา 1 วิมาที
อืน:
                    print("No readingObtained") # พิมสะตริงยู่ Pi Terminal
                    wait.sleep(1)# มอมเป้นเวลา 1 วิมาที
```

7.8 Line-Following Robot กับ Pyfirmata

ทุ่นย์นผู้ติดตามสายติดตามพาบยู่ทางเทิง ทื่เพดาน. โดยป์กะติแล้ว, เส้นสายตาแม่นสีดำยู่เทิงพื้นผิวสีขาว, เถิงแม่นว่าเส้นสีขาวยู่ด้านสีดำภืเป้นไปได้. ทุ่นย์นติดตามสายแม่นใช้ในอุดสาหะกำการผะลิดสำลับะบวนการอัดตะโนมัด. มันเป้นหนึ่งในทุ่นย์นพื้นถานทีสุดสำลับผู้เลิ่มต้ัน. เพื่อเข้าใจการออกแบบของทุ่นย์นทีมี Raspberry Pi และ Arduino Uno, ละบ็บดังก่าวประกอบด้วยไดเว้มໍเติ L293D, สองมໍเติ DC, ล้ฟริ (เพื่อเชื่อมต็ยู่ด้านหน้าของทุ่นย์น), สองเซ้นเซิ IR, และพะลังงาน. การสะหนอง.

การเชื่อมต็:

- เชื่อมต็ pins (IN1, IN2, IN3, IN4) ของ L293D กับ pins (5, 4, 3, 2) ของ Arduino Uno, ตามลำดับ.
- เชื่อมต็มໍเติ DC (M1) ละหว่าง pins (OUT1 และ OUT2) ของ L293D.



ຮູບທີ 7.7
ແຜນວາດວົງຈອນສຳລັບຫຸ່ນຍົນຕາມເສັ້ນ.

•ເຊື່ອມຕໍ່ມໍເຕີ DC ອື່ນໆ (M2) ລະຫວ່າງ pins (OUT3 ແລະ OUT4) ຂອງ L293D.

•ເຊື່ອມຕໍ່ pins (Vcc ແລະດິນ) ຂອງ IR1 ແລະ IR2 ກັບ +5 VDC ແລະດິນ, ຕາມລຳດັບ.

• ເຊື່ອມຕໍ່ pin (OUT) ຂອງ IR1 ກັບ pin (7) ຂອງ Arduino Uno.
• ເຊື່ອມຕໍ່ pin (OUT) ຂອງ IR2 ກັບ pin (6) ຂອງ Arduino Uno.
• ເຊື່ອມຕໍ່ Arduino Uni ກັບ Raspberry Pi ຜ່ານ USB.

ຮູບທີ 7.7ສະແດງໃຫ້ເຫັນແຜນວາດວົງຈອນສຳລັບຫຸ່ນຍົນຕາມເສັ້ນ.

7.8.1 ສູດ

```
ນຳເຂົ້າ pyrmata
ນຳເຂົ້າເວລາລໍຖ້າ
ກະດານ = pyrmata.Arduino('/dev/ttyUSB10')
ir1_pin = board.get_pin('d:7:i') # ເຊື່ອມຕໍ່ IR sensor1 ກັບ pin 7 ແລະໃຊ້
    ເປັນການປ້ອນຂໍ້ມູນ
ir2_pin = board.get_pin('d:6:i') # ເຊື່ອມຕໍ່ IR sensor2 ກັບ pin 6 ແລະໃຊ້
    ເປັນການປ້ອນຂໍ້ມູນ
```

```
M11_pin = board.get_pin('d:5:o') # ເຊື່ອມຕໍ rst motor pin ກັບ 5 ແລະໃຊ້
    ເປັນຜົນຜະລິດ
M12_pin = board.get_pin('d:4:o') # ເຊື່ອມຕໍ rst motor pin ກັບ 4 ແລະໃຊ້
    ເປັນຜົນຜະລິດ
M21_pin = board.get_pin('d:3:o') # ເຊື່ອມຕໍ່ມໍເຕີທີສອງກັບ 3 ແລະ
    ໃຊ້ເປັນຜົນຜະລິດ
M22_pin = board.get_pin('d:2:o') # ເຊື່ອມຕໍ motor pin ທີສອງກັບ 2 ແລະ
    ໃຊ້ເປັນຜົນຜະລິດ
it = pyrmata.util.Iterator(board) # ໃຊ້ iterator
it.start() # start iterator
ir1_pin.enable_reporting() # ເປີດໃຊ້ການລາຍງານຂອງ IR sensor1
ir2_pin.enable_reporting() # ເປີດໃຊ້ການລາຍງານຂອງ IR sensor2 ໃນຂະນະ
ທີຖືກຕ້ອງ:
                    ir1_state = ir1_pin.read() # ອ່ານ IR sensor 1
                    ir2_state = ir2_pin.read() # ອ່ານ IR sensor 2 ຖ້າ
                    ir1_state == False ແລະ ir2_state == False:
                            M11_pin.write(1) # ເຮັດໃຫ້ pin5 ເປັນ
                                ສູງ
                            M12_pin.write(0) # ເຮັດໃຫ້ pin4 ເປັນ
                                ຕ່ຳ
                            M21_pin.write(1) # ເຮັດໃຫ້ pin3 ເປັນ
                                ສູງ
                        M22_pin.write(0) # ເຮັດໃຫ້ pin2 ເປັນການ
                        ພິມໜ້ອຍ('forward') # ພິມໃສ່ເຄືອງ
                                wait.sleep(0.5) # ຄວາມລ່າຊ້າຂອງ 500mSec
                    elif ir1_state == ບໍຖືກຕ້ອງ ແລະ ir2_state == ຖືກ:
                            M11_pin.write(1) # ເຮັດໃຫ້ pin5 ເປັນ
                                ສູງ
                            M12_pin.write(0) # ເຮັດໃຫ້ pin4 ເປັນ
                                ຕ່ຳ
                            M21_pin.write(0) # ເຮັດໃຫ້ pin3 ເປັນ
                                ຕ່ຳ
                            M22_pin.write(0) # ເຮັດໃຫ້ pin2 ເປັນ
                                ຕ່ຳ
                        print('ຊ້າຍ') # ພິມຢູ່ terminal
                        time.sleep(0.5) # ຊັກຊ້າ 500mSec
```

```
elif ir1_state == ถึก และ ir2_state == ผิด:
        M11_pin.write(0) # เธัดใท้ pin5 เป้น
          ต๋า
        M12_pin.write(0) # เธัดใท้ pin4 เป้น
          ต๋า
        M21_pin.write(1) # เธัดใท้ pin3 เป้น
          สูง
        M22_pin.write(0) # เธัดใท้ pin2 เป้น
          ต๋า
        print('Right') # พิมยู terminal
        time.sleep(0.5)# ชักช้า 500mSec
elif ir1_state == True และ ir2_state == True:
        M11_pin.write(0) # เธัดใท้ pin5 เป้น
          ต๋า
        M12_pin.write(0) # เธัดใท้ pin4 เป้น
          ต๋า
        M21_pin.write(0) # เธัดใท้ pin3 เป้น
          ต๋า
        M22_pin.write(0) # เธัดใท้ pin2 เป้น
          ต๋า
    print('Stop') # พิมยู terminal
    time.sleep(0.5) # ชักช้า 500mSec
```