

5

ພຶບຊານຂອງ Raspberry Pi

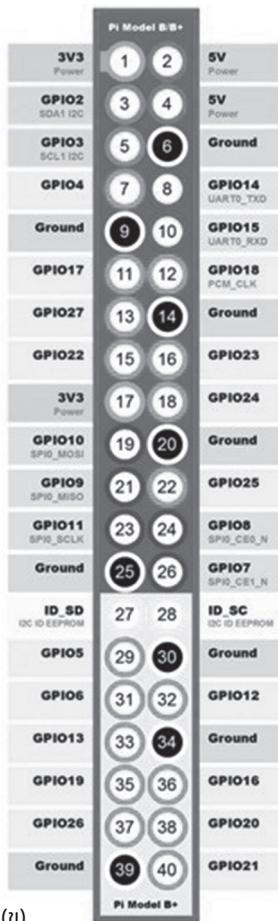
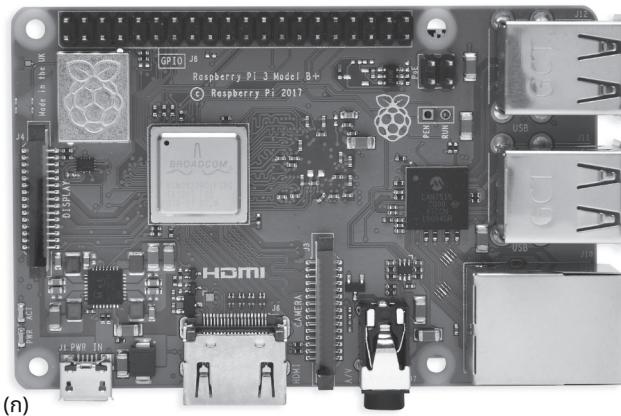
5.1 ຮູ່ເບື້ອງຕົ້ນເກີຍວັກັບ Raspberry Pi

Raspberry Pi ເປັນຄອນພົວເຕັອຮ່ານາດເລື້ອກຄາປະໜັດທ່ານສານຮັບເຊື່ອນຕ່ອກັບຈອກຄອນພົວເຕັອຮ່ານີ້ ເປັນອຸປະນຸມສໍາຫຼັບສໍາຮວຈຄອນພົວເຕັອຮ່ານໂດຍໃຊ້ແປ່ນພົບພື້ນແລະເນາສີສານຮັບເຊື່ອນຕ່ອງຊັ້ນສ່ວນເສີຣີສອງສານຊັ້ນເພື່ອການເຂົ້າເຖິງກໍາຕົ້ງຢືນຂຶ້ນ ສານຮັບເຕັ້ງໂປຣແກຣມໃນກາຫາເຊັ່ນ Scratch ແລະ Python ຄວາມຄືໂປຣເຊີ່ອຮ່ວມມືຕັ້ງແຕ່ 700 MHz ຄົ່ງ 1.4 GHz ລໍາຮັບ Pi 3 ຮູ່ B+ ໃຫ້ວຍຄວາມຈໍາອວນບອດມີຕັ້ງແຕ່ 256 MB ຄົ່ງ 1 GB RAM Raspberry Pi 3+ ມີໂປຣເຊີ່ອຮ່ວມມື ARM cortex A53 ແບບ Quad-core ຄວາມເຮົ້າ 1.4 GHz 64 ພັຕະຫຼຸບກໍາຕົ້ງຢືນຕີ 5.1a ແລະ ດັກ Raspberry Pi

ຮຸນແຮກຂອງ Raspberry Pi 1 ຮູ່ນ "B" ເປັດຕົວໃນເດືອນກຸມພັນນີ້ 2012 ກາຣອອກແບບທີ່
ປັບປຸງແລ້ວ Raspberry Pi 1 ຮູ່ນ "B+" ເປັດຕົວໃນປີ 2014 Raspberry Pi Zero ມີບາດ
ເລື້ອກງານແລະລົດອົນພຸດ/ເອົາຕີພຸດ (I/O) ດ້ວຍຄວາມສານຮັບເອົນພຸດ/ເອົາຕີພຸດເອນກປະສົງ (GPIO)
ແລະເປັດຕົວໃນເດືອນພຸດສົງກາຍນີ້ 2558 Raspberry Pi Zero W ຮູ່ນ Zero W ທີ່ມີຄວາມສານຮັບ
Wi-Fi ແລະ Bluetooth ເປັດຕົວເມື່ອວັນທີ 28 ກຸມພັນນີ້ 2014 Raspberry Pi Zero WH ເປັດ
ຕົວເມື່ອວັນທີ 12 ມັງກອນ 2018; ເປັນເວັບຮັບຂອງ Zero W ທີ່ມີລ່ວນໜັງ GPIO ທີ່ບັດກຽດລ່ວນໜັງຫັນ
Raspberry Pi 3 Model B ເປັດຕົວໃນເດືອນກຸມພັນນີ້ 2559 ພ້ອມ Wi-Fi ອອນບອດ,
Bluetooth 1.2 GHz, ໂປຣເຊີ່ອຮ່ວມມື Quad Core 64 ພັຕະ ແລະຄວາມສານຮັບໃນການບູ້ຕີ USB
Raspberry Pi 3 Model B+ ເປັດຕົວໃນປີ 2018 ດ້ວຍໂປຣເຊີ່ອຮ່ວມມື 1.4 GHz ແລະ 300 Mbit/
s ໂດຍການເຊື່ອນຕ່ອງ USB 2.0 ກາຍໃນຫຼືກໍາຕົ້ງຢືນຕີ 5.1b ແລະ ດັກແພັນກາພັນຂອງຮຸນ Raspberry
Pi 3 B+

5.1.1 ສ່ວນປະກອບ Raspberry Pi

1. **ແໜ່ລັງຈ່າຍໄຟ:** Raspberry Pi ສານຮັບເຄື່ອນດ້ວຍການເຊື່ອນຕ່ອງ micro-USB ທີ່
ສານຮັບຈ່າຍໄຟຍ່າງນ້ອຍ 400 mA ທີ່ 5 V; ເຄື່ອງຫຼັງຈິກໂກຣຄັກພົກນີ້ອື່ນບາຕຣ້ຫຼາບ
ເໝາະສໍາຫຼັບຈຸດປະສົງນີ້ ທາງ Raspberry Pi ໃຊ້ພລັງຈານຈາກພົກພົກຕົວມີຕົວມີຕົວ
ຄອນພົວເຕັອຮ່ານເຄື່ອງອື່ນ ເຄື່ອງຈາກໄມ້ມີກະແສໄຟກໍາຕົ້ງການກຳຕົ້ງຢືນຕີ 5.2 ແລະ ແກ່ລ່າງຈ່າຍ
ໄຟ micro-USB



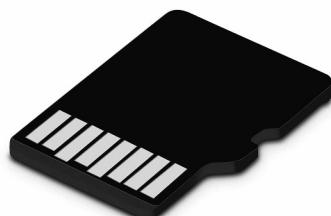
ຮູບທີ 5.1

(ກ) ຮາສເບອຮ໌ Pi. (ບ) ແຜນກາພພິບຂອງຮຸນ Raspberry Pi 3 B+



ຮູປ໌ 5.2
ແຮລ່ງຈ່າຍໄຟແບບໄນໂຄຣຢູເອສນີ

2. **ໜ້າຈອ:** ແນ້າຈອເປັນຈອກພັກຮ້ອໂກຣກັບນີ້ມີອັນພຸດ HDMI ຮັບ DVI ນອກຈາກນີ້ຢັງ
ສາມາດເຊື່ອນຕ່ອກັບອະແດປເຕອർ VGA ອັບອະແດປເຕອർ HDMI ເປັນ VGA
3. **ແປ່ນພິມພົມແລະເມາສີ:** ອຸປະກອນຕ່ອງໆໄຟ່ງ USB ແບບໃຫ້ພລັງຈານຕໍ່າ (ແປ່ນພິມພົມແລະເມາສີ)
ທີ່ມີກຳລັງໄຟບ້ອຍກວ່າ 0.1 A ເໜັນສໍາຮັບການເຊື່ອນຕ່ອກັບ Raspberry Pi ອຸປະກອນທີ່
ສ້ານເປັນເປົ້າພລັງຈານນາກີ່ນອາຈານຕ້ອງໃຫ້ຮັບພລັງຈານແຍກຕ່າງໆ
4. **ກາຮົດ SD:** ກາຮົດ SD ທີ່ໃຫ້ສໍາຮັບ Raspberry Pi ຄວນມື່ອຢ່າງນ້ອຍ Class 4 ແລະນີ້
ພື້ນທີ່ເກີບຂໍອມູລອຢ່າງນ້ອຍ 8 GB [ຮູປ໌ 5.3](#)ແສດງກາຮົດ SD ຕ້ອງຕິດຕັ້ງກາຮົດ SD ບັນ
ຮະບບປົງບັດກາຈາກຄອມພິວເຕອຮີເຄື່ອງອື່ນ ຂັບຕອນການຕິດຕັ້ງນີ້ສາມາດກຳໄດ້ບັນ
Windows, Mac ແລະ Linux ຮະບບປົງບັດກາເຊັ່ນ NOOBS ແລະ Raspbian ເປັນ
ຮະບບປົງບັດກາທີ່ໃຫ້ບ້ອຍທີ່ສຸດໃນການຕິດຕັ້ງບັນກາຮົດ SD



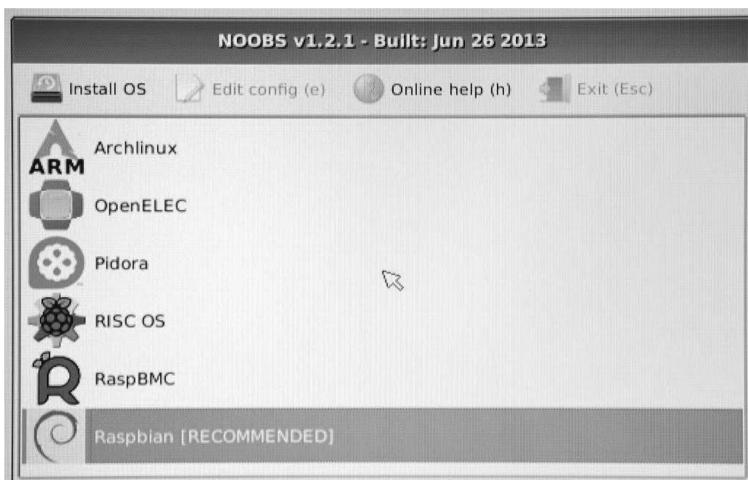
ຮູປ໌ 5.3
ກາຮົດ SD

5.2 ການຕິດຕັ້ງ NOOBS ບນກາຣົດ SD

ຂ້ອງກໍາເໜັດຫຼັກໃນການຕິດຕັ້ງ NOOBS ຄື່ອເດສກທົບປ່ອມ ເນື່ອຈາກໄມ່ສາມາຄຕິດຕັ້ງບັນແລັບກໍາປ່ອມໄດ້
ເນື່ອຕິດຕັ້ງແລ້ວ ກີ່ສາມາຄໃຊ້ກັບແລັບກໍາປ່ອມໄດ້

ບັນຕອນໃນການຕິດຕັ້ງ NOOBS ບນກາຣົດ SD

1. ພອຣແມຕກາຣົດ SD ດ້ວຍຄວາມໜ່ວຍເຫຼືອຂອງພອຣແມຕກາຣົດ SD ສາມາຮັດ
ດາວໂຫຼດໄດ້ຈາກ https://www.sdcard.org/downloads/formatter_4/
2. ດາວໂຫຼດໄຟລ໌ເກີບຄວາວສໍາຫັກ NOOBS ຈາກ <http://www.raspberrypi.org/downloads>.
3. ແຕກໄຟລ໌ເກີບຄວາວຂອງ NOOBS ແລະ ຄັດລອກເນື້ອຫາຂອງໂຟລුເດෝර්ໃນກາຣົດ SD
4. ໄສກາຣົດ SD ໃນ Raspberry Pi ແລະ ເປີດເຄື່ອງ ມັນຕ່າງຈະເປີດຂຶ້ນເມື່ອຢູ່ຕັ້ງທີ່
ແສດງໃນຮູບກໍ່[5.4](#) ຈາກນັບຄຸກທີ່ Raspbian ແລະ ມັນຕ່າງໃໝ່ຈະເປີດຂຶ້ນດັ່ງແສດງ
ໃນຮູບກໍ່[5.5](#). ຄັດໄປ ມັນຕ່າງ Linux ສໍາຫັກ Raspberry Pi ຈະເປີດຂຶ້ນ ແລະ ຕອນ
ນີ້ກີ່ພໍ່ອມໃຊ້ງານແລ້ວ
5. ດາວໂຫຼດໂຫຼດຂອງຟົດແວຣ໌ເຄື່ອງສແກບ IP ຖໍ່ໂກຮຣແລະ ຮັບຖື່ IP ([ຮູບກໍ່ 5.6](#)).
6. ຕອນນີ້ດາວໂຫຼດຂອງຟົດແວຣ໌ "putty" ຢັ້ງ "mobaX term" ແລ້ວ ຄັດລອກ IP ທີ່
ອ່ຍ່ໃນບັນຕອນກ່ອນມັນຫັງແລະ ປ້ອນໃນເກອຮັນນັ້ນ Linux ຂອງ Pi
7. ປ້ອນຮັສຜູ້ໃຊ້ເຮັນຕົ້ນ > pi ແລະ ຮັສຜ່ານ > ຮາສເບອຣີ

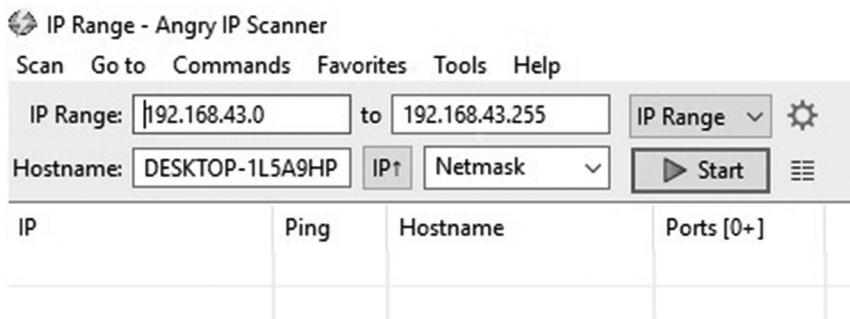


ຮູບກໍ່5.4
ມັນຕ່າງ NOOBS



ຮູບ 5.5

Raspberry Pi ກໍາລັງບຸກປຶ້ນ



ຮູບ 5.6

ສະແກນເບີນອົບ IP ໂກຮຣ

5.3 ການຕິດຕັ້ງ Raspbian ບນກາຣດ് SD

ຂັ້ນຕອນໃນການຕິດຕັ້ງ Raspbian ບນກາຣດ് SD

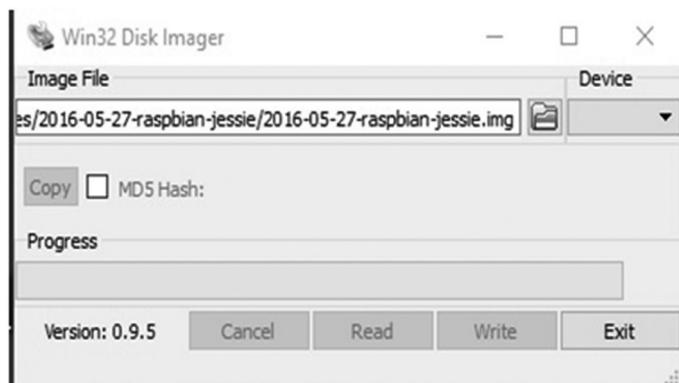
ໃນການດຳເນີນກາຣໄໃຫ້ເສົ້າງສັນ ຈຳເປັນຕ້ອງດາວໂຫລດໂຄໂລດໜອົງທີ່ເວັບສອນຕັ້ງແລະຮະບບປົງບັດຕິການເໜີງຮະບບ (Raspbian)

1. ພົມກາຣດ് SD ດ້ວຍຄວາມຊ່ວຍເຫຼືອຂອງພົມກາຣດ് SD ສາມາດ
ດາວໂຫລດໄດ້ຈາກ https://www.sdcard.org/downloads/formatter_4/ ຮູບກໍ 5.7 ແລະ ດັບອົບຕ່າງຕັ້ງຈັດຮູປແບບ SD



ຮູບທີ 5.7
ຟອົນແມຕ SD

2. ດາວໂຫຼດ Win32 Disk Imager ຈາກ <https://sourceforge.net/projects/win32diskimager/> ເພື່ອເປີ່ນອົມເຈົ້າຂອງ Raspbian ບນກາຣດ SD
3. ດາວໂຫຼດ Raspbian ຈາກ <https://www.raspberrypi.org/downloads/raspbian/> ແລະ ລັບກາພແລະເປັນບັນ Raspberry Pi
4. ໄສ່ກາຣດລອງໃນ Raspberry Pi ແລ້ວຮອປະນານ 2 ບາດ
5. ເຊື່ອມຕ່ອ Raspberry Pi ກັບຄອນພິວເຕອຮັກຜ່ານ LAN
6. ກຳຕາມຂັ້ນຕອນທີ 6, 7 ແລະ 8 ຈາກນາຕາຮາ 5.2.



ຮູບທີ 5.8
ອົມເຈົ້າສົກ Win32

5.4 ຄໍາສັ່ງເກອຮີນັບລັບ

ຮາຍການຄໍາສັ່ງເກອຮີນັບລັບ:

- 1.*sudo apt-get update*: ຄໍາສັ່ງນີ້ອັພເດຕາຮາຍການແພຶກເກຈ
- 2.*sudo apt-get ອັພເກຣດ*: ຄໍາສັ່ງນີ້ຈະດາວໂຫຼດແລະຕິດຕັ້ງແພຶກເກຈທີ່ອັພເດຕ
- 3.*sudo apt-get clean*: ຄໍາສັ່ງນີ້ຈະລ້າງໄຟລ໌ແພຶກເກຈເກົ່າ
- 4.*sudoraspiconfig*: ຄໍາສັ່ງນີ້ກຳທັດຄໍາເຄື່ອງນູ້ສໍາຮັບ Raspberry Pi
- 5.*ls*: ຄໍາສັ່ງນີ້ແສດງຮາຍການເນື້ອຫາໄດ້ເຮັກກອຣີ
- 6.ສັດ: ຄໍາສັ່ງນີ້ປັບໄລຍົບໄດ້ເຮັກກອຣີ
- 7.*mkdir*: ຄໍາສັ່ງນີ້ສ້າງໄດ້ເຮັກກອຣີ
- 8.*rmdir*: ຄໍາສັ່ງນີ້ຈະລົບໄດ້ເຮັກກອຣີ
- 9.*mv*: ຄໍາສັ່ງນີ້ຍ້າຍໄຟລ໌
- 10.ດັ່ງນີ້*-d*: ຄໍາສັ່ງນີ້ແສດງແຜນຜັງຂອງໄດ້ເຮັກກອຣີ
- 11.*pwd*: ຄໍາສັ່ງນີ້ແສດງໄດ້ເຮັກກອຣີປັຈຈຸບັນ
- 12.ແຈ່ນໃສ: ຄໍາສັ່ງນີ້ໃຊ້ສໍາຮັບລ້າງໜ້າຕ່າງເກອຮີນັບລັບ
- 13.*sudo հູດ*: ຄໍາສັ່ງນີ້ກໍາໄລໃຫ້ Raspberry Pi ປິດຕັ້ງລົງ
- 14.*sudo ຮູບຕາ*: ຄໍາສັ່ງນີ້ຄ້ອໃຫ້ສຕາຣກ Raspberry Pi
- 15.*startx*: ຄໍາສັ່ງນີ້ຄ້ອກາຮົບຕົ້ນສະພາພເວດລ້ອມເດສກຖືກ (LXDE)
- 16.*ifconfig*: ຄໍາສັ່ງນີ້ຄ້ອກາຮົບຕົ້ນໜ້າທີ່ອ່າຍ IP ຂອງ Raspberry Pi
- 17.ນາໂນ: ຄໍາສັ່ງນີ້ຄ້ອກາຮົບແກ້ໄຂໄຟລ໌
- 18.ແບວ: ຄໍາສັ່ງນີ້ແສດງເນື້ອຫາຂອງໄຟລ໌
- 19.ອາຣີເອັບ: ຄໍາສັ່ງນີ້ຈະລົບໄຟລ໌
- 20.ສັທິ: ຄໍາສັ່ງນີ້ຄ້ອດລອກໄຟລ໌ເກົ່າໄດ້ເຮັກກອຣີ
- 21.ກັ່ນໜ້າ: ຄໍາສັ່ງນີ້ຄ້ອກາຮົບກັ່ນໜ້າໄຟລ໌
- 22.ໜູດ: ຄໍາສັ່ງນີ້ໃຊ້ໃນບຣກັດຄໍາສັ່ງ Linux; sudo ຢ່ອມຈາກ “SuperUser Do”
- 23.ເດີຟ: ຄໍາສັ່ງນີ້ຄ້ອກາຮົບດູພື້ນກີດສກົກມືອ່ຍໃບແຕ່ລະພາຣຕິຊັນໃນຮະບບ
- 24.ດ: ຄໍາສັ່ງນີ້ແສດງການໃຊ້ດີສກົກຂອງໄຟລ໌ໃນຮະບບ
- 25.ນ້ຳມັນດົນ: “tar -cvf” ໃຊ້ສໍາຮັບສ້າງໄຟລ໌ .tar, -xvf ເພື່ອ untarໄຟລ໌ tar - tf ເພື່ອແສດງຮາຍການເນື້ອຫາຂອງໄຟລ໌ເກົ່າກວາງ ລຸ່ມ
- 26.*zip*, ຄລາຍຊັບ: ຄໍາສັ່ງນີ້ຈະບັບອັດໄຟລ໌ລົງໃນໄຟລ໌ zip ແລະແຕກໄຟລ໌ເພື່ອແຕກໄຟລ໌
- 27.ໄມ້ນີ້ສ່ອງ: ຄໍາສັ່ງ “uname -a” ພິມພັ້ນຂອ້ມູນລ່ວມໃຫຍ່ເກື່ອງກັບຮະບບ

28. chmod: ຄໍາສັ່ນນີ້ກ່ອນເປົ້າໃຫຍ່ສັດທະນູຂອງໄຟລ໌ຮັບຮູ້ໄດ້ເຮັດວຽກ

29. ping: ຄໍາສັ່ນນີ້ກ່ອນເປົ້າໃຫຍ່ກວ່າມຕົວຢ່າງເປົ້າຂອງຄຸນໄປຢ້າງເຊີຣົບໄວ້

30. ກ່ອນປິດຮັບຮູ້ sudo: ຄໍາສັ່ນນີ້ກ່ອນເປົ້າໃຫຍ່ກວ່າມປິດ Raspberry Pi

5.5 ກ່ອນຕິດຕັ້ງໄລຍະຮັບ Raspberry Pi

```
$ sudo apt-get update  
$ sudo apt-get install libblas-dev  
$ sudo apt-get install liblapack-dev  
$ sudo apt-get install gfortran  
$ sudo apt-get update  
$ sudo easy_install-scipy  
$ ## ກ່ອນນັ້ນອ້າຈາໃຫ້ຈານໄດ້: python-scipy ໂດຍໄມ້ຕ້ອງພື້ນຖານແລ້ວ  
$ sudo apt-get install python-matplotlib
```

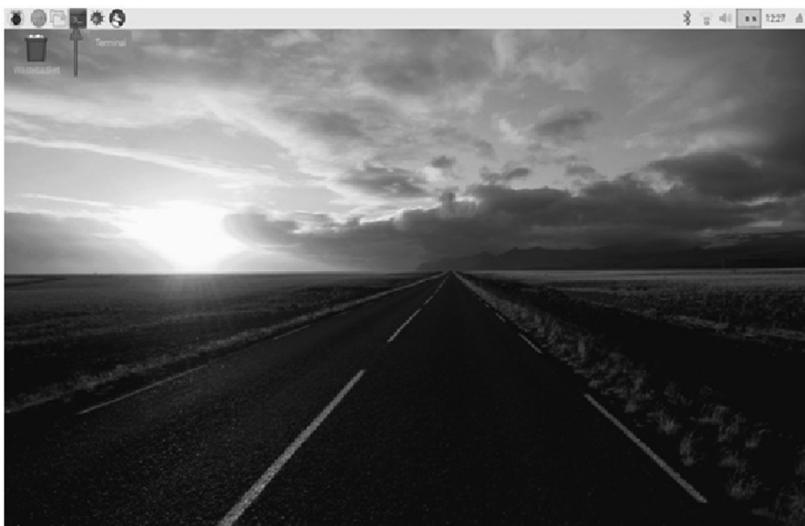
5.6 ກ່ອນຕິດຕັ້ງ IP ແບບຄົງກືຂອງ Raspberry Pi

Raspberry Pi ໄກສາກືອງ IP ວິນ ຖຸກຄັ້ງກີ່ໄດ້ຮັບ "ON" IP ນີ້ຈະເຮັດວຽກກືອງ IP ບັງຈຸບັນ ດັ່ງນັ້ນໃນແຕ່ລະຄັ້ງ ຜູ້ໃຊ້ຈະຕ້ອງຕົວຈຳກັດກືອງ IP ຂອງນັ້ນ ເພື່ອຮັກເລີຍງົມງານນີ້ ຈຳເປັນຕ້ອງສ້າງກືອງ IP ແບບຄົງກື ທີ່ຈະໄປ່ເປົ້າໃຫຍ່ແປລັງເມື່ອ "ປິດ" ອຸປະກອນ

ຂັ້ນຕອນໃນການສ້າງກືອງ IP ແບບຄົງກື:

- ເປີດໜ້າຕ່າງເກອຮົນນັ້ນ (ຮູບ 5.9).
- ກີ່ປັບປຸງຄົດຄໍາສັ່ນ ໃຫ້ເຂົ້າໃນຄໍາສັ່ນ "ifconfig" (ຮູບ 5.10).
ຄໍາສັ່ນນີ້ໃຫ້ຂ້ອງມູນກີ່ເກີຍວ່າຂອງກັບເຄຣືອຂ່າຍ ດີ ຮູບ 5.10 ບໍລິຄວາມກີ່ໄລຕົວແສດຖານກືອງ IP ບັງຈຸບັນ ປີເກີດ 192.168.0.16 ມາຍເຫດຖືກືອງ IP ນີ້ເນື່ອງຈາກຈະຕ້ອງໃຫ້ໃນກາຍ ຫລ້ວ
- ເປີດໄຟລ໌ການກຳນົດຄໍາ
ຕອບນີ້ເປີດໄຟລ໌ການກຳນົດຄໍາກ່ອນສາມາດກຳນົດຄໍາກືອງ IP ແບບຄົງກືໄດ້ ຜູ້ໃຊ້ໄຟລ໌
ຄໍາ `dhcpcd.conf` ແລະ ຖຸກເກີດໄວ້ໃນໂຟລົດເອົ້າຊ່າຍ. ການຕ້ອງການເປົ້າໃຫຍ່ແປລັງ
ໃຫ້ໃຊ້ຕົວແກ້ໄຂ `nano` ໃນໜ້າຕ່າງເກອຮົນນັ້ນໂດຍໃຫ້ຄໍາສັ່ນ: "sudo nano /etc/
dhcpcd.conf" (ຮູບ 5.11). ລັ້ງຈາກຄໍາສັ່ນນີ້ ໜ້າຕ່າງຈະເປີດຂຶ້ນ (ຮູບ 5.12).

ໃນໜ້າຈອງຮູບ 5.12 ໄກສາເລືອນລົງມາຈົນກວ່າຄຸນຈະເຮັດວຽກ: "# Example static IP
configuration" (ຮູບ 5.13).



ຮູບກີ.9
ຫຼັບຕ່າງເກອບນັບລັບ

```
pi@raspberrypi: ~
```

```
File Edit Tabs Help
```

```
pi@raspberrypi: ~ ifconfig
```

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.16 brd 192.168.0.255 broadcast 192.168.0.255
            netmask 255.255.255.0
            broadcast 192.168.0.255
            scopeid 0x0<link>
            inet6 fe80::894b:60fe:1b3:1e7a brd fe80::ff:fe1b:3:1e7a
                  prefixlen 64
                  scopeid 0x20<link>
      linklayer 1000 (Ethernet)
      packets 3456 bytes 399440 (399.4 Kib)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 121 bytes 19887 (19.4 Kib)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
            netmask 255.0.0.0
            broadcast 127.0.0.1
            linklayer 1000 (Loopback)
      linklayer 1000 (Loopback)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.24 brd 192.168.0.255 broadcast 192.168.0.255
            netmask 255.255.255.0
            broadcast 192.168.0.255
            scopeid 0x0<link>
            inet6 fe80::1b2:27ff:fe00:1624 brd fe80::ff:fe00:1624
                  prefixlen 64
                  scopeid 0x20<link>
      linklayer 1000 (Ethernet)
      packets 156 bytes 21187 (20.6 Kib)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 65 bytes 12888 (12.5 Kib)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi: ~
```

ຮູບ 5.10
ຫຼັບຕ່າງຄໍາສັ່ງສໍາເທັບ Raspberry Pi ກັບຈາກຄໍາສັ່ງ “ifconfig”

ຕອນນີ້ລັບ "#" ອອກຈາກແຕ່ລະບຽບກັດຂອງຍ່ອຫັນນີ້ແລະປັບປຸງຕົວເລຂສຸດກໍາຍຂອງ
ກີ່ອ່າງ IP ຈາກຂ່າງ 1 ຄົງ 255 (ຮູບກີ.14). ເນື່ອງການປັບປຸງແປລົງແລ້ວ ໃຫ້ອອກ
ຈາກຫຼັບຕ່າງໂດຍໃຊ້ Ctrl+X ແລະບັນທຶກການປັບປຸງແປລົງໂດຍຄຶກກໍ "ໃໝ່" (ຮູບກີ.
5.15).

4. ຮູ່ມູ Raspberry Pi ດ້ວຍຄໍາສັ່ງ “reboot” (ຮູບກີ.16).
5. ຕຽບສອບກີ່ອ່າງ IP ແບບຄົງກໍຂອງ Raspberry Pi ດ້ວຍຄໍາສັ່ງ “ifconfig” (ຮູບກີ.17).

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi: ~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.10 netmask 255.255.255.0 broadcast 192.168.0.255
              inet6 fe80::2a02:27ff:fe04:63:br linklayer
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
global>
ether b8:27:eb:f3:be:c5 txqueuelen 1000 (Ethernet)
RX packets 345 bytes 39935 (38.9 Kib)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 121 bytes 19887 (19.4 Kib)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 9 bytes 524 (524.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 9 bytes 524 (524.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.4 netmask 255.255.255.0 broadcast 192.168.0.255
              inet6 fe80::1b8c:74b:31b3:9b99 prefixlen 64 scopeid 0x20<link>
global>
ether b8:27:eb:a6:ab:90 txqueuelen 3000 (Ethernet)
RX packets 155 bytes 21197 (20.6 Kib)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 65 bytes 12438 (12.5 Kib)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
pi@raspberrypi: ~ $ sudo nano /etc/dhcpd.conf
```

ຮູບ 5.11

ຕັ້ງແກ່ໃໝ່ໄນ້ສໍາເກຼີບການກຳຫົນດຄ່າກໍອຽ່ IP ແບບຄົກ

```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4          File: /etc/dhcpd.conf
# A sample configuration for dhpcd.
# See dhpcd.conf(5) for details.

# Allow users of this group to interact with dhpcd via the control socket.
#controlgroup wheel

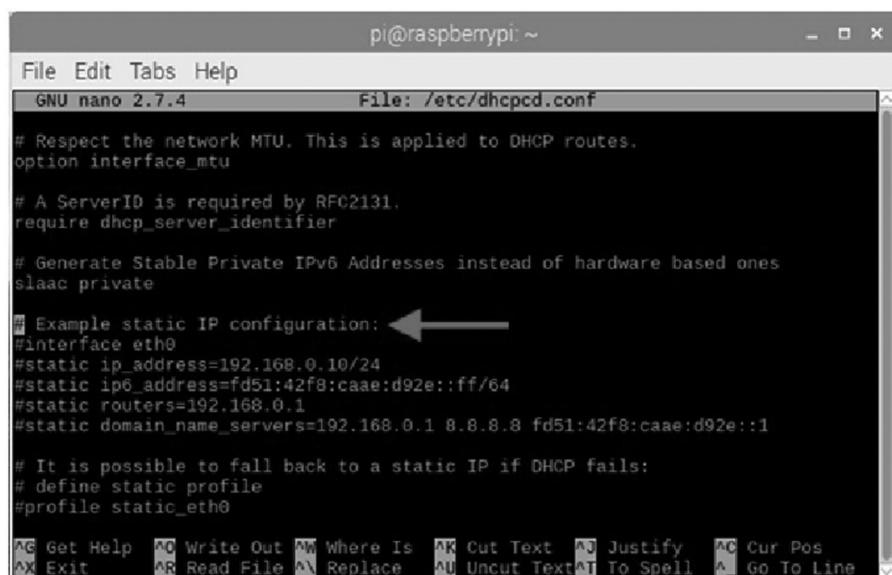
# Inform the DHCP server of our hostname for DDNS.
hostname

# Use the hardware address of the interface for the Client ID.
clientid
# or
# Use the same DUID + IAID as set in DHCPv6 for DHCPv4 ClientID as per RFC4361.
# Some non-RFC compliant DHCP servers do not reply with this set.
# In this case, comment out duid and enable clientid above.
#duid

# Persist interface configuration when dhpcd exits.
persistent
[ Read 74 lines ]
^G Get Help ^C Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell ^L Go To Line
```

ຮູບ 5.12

ໜ້າຕ່າງໆສໍາເກຼີບການກຳຫົນດຄ່າກໍອຽ່ IP



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4          File: /etc/dhcpcd.conf
# Respect the network MTU. This is applied to DHCP routes.
option interface_mtu

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

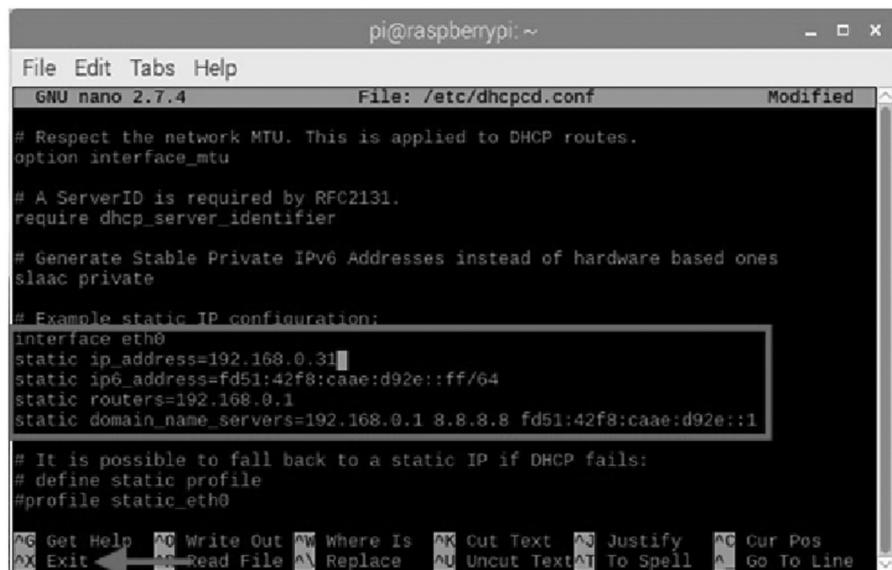
# Example static IP configuration: ←
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0

[G] Get Help [W] Write Out [W] Where Is [C] Cut Text [J] Justify [P] Cur Pos
[E] Exit [R] Read File [R] Replace [U] Uncut Text [T] To Spell [G] Go To Line
```

ຮູບ 5.13

ການກຳຫັນດຳຄ່າ IP ແບບຄົກທີ່



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4          File: /etc/dhcpcd.conf          Modified
# Respect the network MTU. This is applied to DHCP routes.
option interface_mtu

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

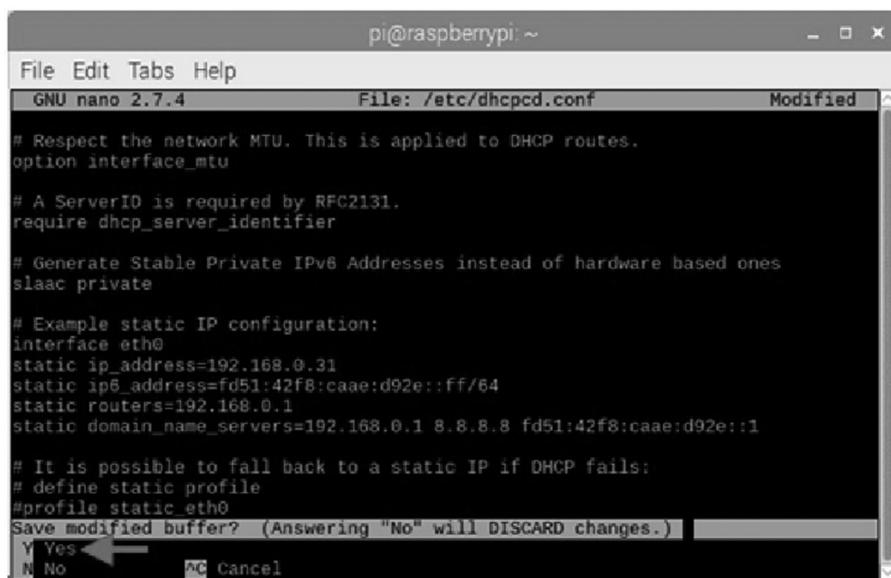
# Example static IP configuration:
interface eth0
static ip_address=192.168.0.31
static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0

[G] Get Help [W] Write Out [W] Where Is [C] Cut Text [J] Justify [P] Cur Pos
[E] Exit [R] Read File [R] Replace [U] Uncut Text [T] To Spell [G] Go To Line
```

ຮູບ 5.14

ກົ່ວ່າງ IP ແບບຄົກທີ່



```
pi@raspberrypi:~
```

File Edit Tabs Help

GNU nano 2.7.4 File: /etc/dhcpcd.conf Modified

```
# Respect the network MTU. This is applied to DHCP routes.
option interface_mtu

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

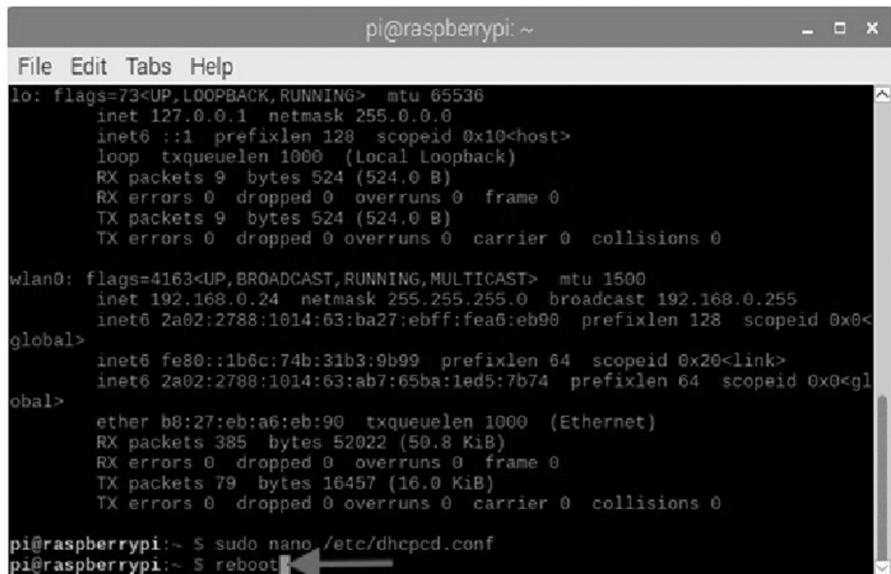
# Example static IP configuration:
interface eth0
static ip_address=192.168.0.31
static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
Save modified buffer? (Answering "No" will DISCARD changes.)
```

Y Yes ← N No AC Cancel

ສົບ 5.15

“ຢ່າງ” ເພື່ອການປ່ອຍແປລ່ອ



```
pi@raspberrypi:~
```

File Edit Tabs Help

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 9 bytes 524 (524.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 9 bytes 524 (524.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.24 netmask 255.255.255.0 broadcast 192.168.0.255
        inet6 2a02:2788:1014:63:ba27:ebff:fea6:eb90 prefixlen 128 scopeid 0x0<global>
            inet6 fe80::1b6c:74b:31b3:9b99 prefixlen 64 scopeid 0x20<link>
            inet6 2a02:2788:1014:63:ab7:65ba:1ed5:7b74 prefixlen 64 scopeid 0x0<global>
        ether b8:27:eb:a6:eb:90 txqueuelen 1000 (Ethernet)
            RX packets 385 bytes 52022 (50.8 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 79 bytes 16457 (16.0 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~ $ sudo nano /etc/dhcpcd.conf
pi@raspberrypi:~ $ reboot
```

ສົບ 5.16

ຮັບຖຸ Raspberry Pi

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi: ~ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.31 brd 192.168.0.255 netmask 255.255.255.0
              broadcast 192.168.0.255
              inet6 fe80::420e:1ff:fe31:89e0 brd fe80::ff:fe31:89e0
                  prefixlen 64
                  scopeid 0x0<global>
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 brd 127.0.0.1 netmask 255.0.0.0
              inet6 ::1 brd ::1 prefixlen 128
                  scopeid 0x10<host>
loop: txqueuelen 1000 (Local Loopback)
        RX packets 9 bytes 524 (524.0 B)
        TX packets 9 bytes 524 (524.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.0.24 brd 192.168.0.255 netmask 255.255.255.0
              broadcast 192.168.0.255
              inet6 fe80::74b:31b3:9b90 brd fe80::ff:31b3:9b90
                  prefixlen 64
                  scopeid 0x20<link>
              inet6 2a02:2788:1014:63:ba27:ebff:fea9:eb90 brd fe80::ff:2a02:2788:1014:63:ba27:ebff:fea9:eb90
                  prefixlen 128
                  scopeid 0x0<glbal>
ether b8:27:eb:90:9b:90 txqueuelen 1000 (Ethernet)
        RX packets 119 bytes 13091 (12.7 Kib)
        TX packets 56 bytes 11052 (10.7 Kib)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
pi@raspberrypi: ~
```

ຮູບ 5.17

ກ່ອຍ່າ IP ແບບຄອງກີ່ນຂອງ Raspberry Pi

5.7 ເຮັດໃຊ້ໂປຣແກຣມບນ Raspberry Pi

ມີຫລາຍວົງໃນກາຮັນໂປຣແກຣມບນ Raspberry Pi ບາງສ່ວນສາມາຮຄເຮັດໃຊ້ໂປຣແກຣນໄດ້ແລະມີດັງຕ່ອໄປນີ້:

- rc.local
- . bashrc
- ແກັນ init.d
- systemd
- crontab

ກີ່ນເຈົ້າຈຸດຄົງ "rc.local"

5.7.1 ກາຮແກ້ໄຂ rc.local

ບັນ Raspberry Pi ໃກ້ແກ້ໄຂໄຟລ໌ /etc/rc.local ໂດຍໃຊ້ຕັ້ງແກ້ໄຂກີ່ນສຶກທີຣຸກ: "sudo nano /etc/rc.local"

ເພີ່ມຄໍາສັ່ງເພື່ອຮັນໂປຣແກຣນ Python ບັນກີກໄຟລ໌ ແລະອອກ ລາກນິກາຮວນຊ້າແບບໄປສິບສຸດ ໄທ້
ຍຸຕິຄໍາສັ່ງໂດຍເພີ່ມ "&" ຕ່ອກ້າຍ ເຊັ່ນ "sudo python /home/pi/sample.py &"

Pi ເຮັດໃຫ້ໂປຣແກຣມນີ້ເນື່ອບູຕາເຄື່ອງກ່ອນເຮັດບົກການເຊີ້ນ ຖໍ່ Pi ຈະໄມ່ເສົ່າງສັນກະບວນການບູຕາ
ຫາກໄປ່ຽວງານເຄື່ອງໜ້າຍແລະໂປຣແກຣມທຳຈຳນວຍ່າງຕ່ອງເນື່ອງ ເຄື່ອງໜ້າຍແລະອຸນຸມາດໃຫ້ຄໍາສັ່ງຮັນ
ໃນກະບວນການທີ່ແຍກຈາກກັນ ແລະທຳການບູຕາຕ່ອງໂດຍຮັນກະບວນການຫລັກ ຕອນນີ້ ຮູບູຕູ Pi ເພື່ອ
ກົດສອບ: "sudo reboot"

5.8 ການຕິດຕັ້ງເຊື່ອຟິວົວໝາດສົກທີ່ໂປຣະໄກລ

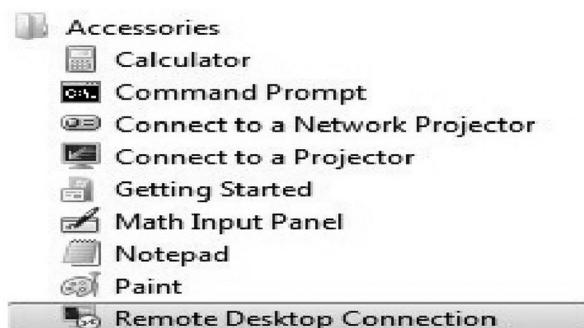
ເຊື່ອຟິວົວໝາດສົກທີ່ໂປຣະໄກລນີ້ປະຢີ້ນສໍາຮັບຜູ້ໃຫ້ໃນການគົບຄຸນຄອນພິວເຕັບຮະຍະໄກລຸ່ານ
ການເຊື່ອນຕ່ອງເຄື່ອງໜ້າຍ ໃນການគົບຄຸນ Raspberry Pi ຈາກຮະຍະໄກລ ຄຸນສາມາດຄືໃຫ້ເຊື່ອຟິວົວໝາດນີ້
ໄດ້ ເປີດເກອຮົມບັບ Raspberry Pi ໂດຍໃຫ້ເມາສີແລະຄີຍບອຮົດ ຮັບໂດຍການເຊື່ອນຕ່ອງໝານ SSH (ໃຊ້
raspi-config) ເພື່ອຕິດຕັ້ງເຊື່ອຟິວົວໝາດສົກທີ່ໂປຣະໄກລ ບັນ Pi ໃຫ້ຕິດຕັ້ງແພັກເຈົ້າ:

"sudo apt-get install xrdp"

ຄັບຫາ "ການເຊື່ອນຕ່ອງເດສົກທີ່ໂປຣະໄກລ" ເພື່ອຕິດຕັ້ງບັບພື້ນ Windows ([ຮູບກີ 5.18](#)).

ເນື້ອຄລິກກີ່ປຸ່ມຕົວເລືອກ ການຕິດຕັ້ງຄ່າຈະເປັນແບບສ່ວນຕົວ ເຊັ່ນ ຄວາມລະເອີຍດහັດຮ້ອກການຕິດຕັ້ງຄ່າ
ຄີຍບອຮົດ/ເສີ່ຍງ

ປັບປຸງໂອສຕ້າຮ້ອງ IP ຂອງ Raspberry Pi (ຮ້ອເຊື່ອຟິວົວໝາດ DNS) ຮັບຊື່ Pi (ຄ່າເຮັບຕົ້ນ: 57
ສະບອຣີຣູ). ທາກພີ້ອຍໆໃນເຄື່ອງໜ້າຍເດືອຍກັບ Pi ໃຫ້ຈົວເຂົ້າສູ່ຮະບບຈະປຣາກກູ້ຂຶ້ນ ຄັນຫາກີ່ອຍູ່ IP
ຂອງ Pi ໃນແນບູຂອງເຮົາເຕັກໄປ່ການໂດຍເປີດ 192.168.0.1 ຮັບອື 192.168.1.1 IP ກາຍໃນ
ຂອງ Pi ຈະອຍູ່ໃນຮູປແບບດັ່ງນີ້: 192.168.x.xxx



ຮູບ 5.18

ການເຊື່ອນຕ່ອງເດສົກທີ່ໂປຣະໄກລ

5.9 ກລ້ອງ Pi

ກລ້ອງ Pi ເປັນໂມຄູລກທີ່ມີອັນເກອຮີເພີ້ມ Python ຕິດຕັ້ງ Picamera ດ້ວຍຄໍາສັ່ງ "sudo apt-get
ຕິດຕັ້ງ python -picamera"

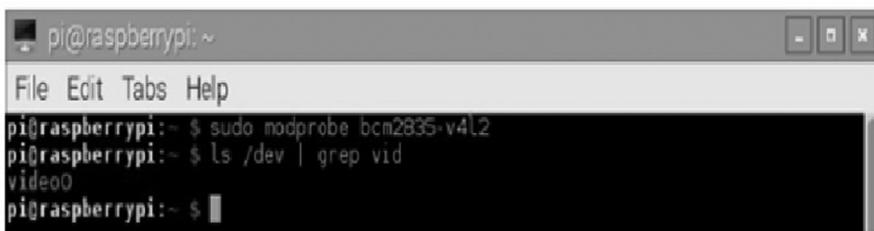
5.9.1 ການກົດສອບກລ້ອງ

ເພື່ອກົດສອບການກຳຈານຂອງກລ້ອງ ໄກສ່າຍ່າພັນທຶນກາພແລະບັນທຶກລົງໃນ /home/pi
ໃນການຈັບກາພໃຫ້ໃຊ້ຄໍາສັ່ງ: "raspistill -o /home/pi/image.jpg" ເນື້ອດຳເນີນການຄໍາສັ່ງ
ກລ້ອງຕົວອ່າງຈະປຣາກງຸ້າບນອບຈອແສດງພລ ການແສດງຕົວອ່າງຈະຢັງເປີດອຸ່ງສອງສາມວິນາກີ
ຈາກນັນຈຶ່ງຄ່າຍກາພ

5.9.2 ກລ້ອງ Raspberry Pi ເປັນອຸປກຮນວິດໄອ USB

ກລ້ອງ Raspberry Pi ເປັນອຸປກຮນວິດໄອ USB ນາຕຣ້ານ (ຕ້ອງໃຊ້ໂດຍ MJPG) ໃນ/ພັດນາ
ໂໂລດໂນດູ "Video for Linux 2" (V4L2) ສໍາຮັບຮັດວຽກ (BCM2835) ໂດຍໃຊ້ຄໍາສັ່ງ "sudo modprobe bcm2835-v4l2" ໂພສ໌ອຸປກຮນ Video0 ກາຍໃນໄດຣັກກອຣີ /dev ຈະ
ປຣາກງຸ້າບນອບ ລັ້ງຈາກຮັບຄໍາສັ່ງ ມາກຕ້ອງການຕຽບສອບ ໄກຮັບຄໍາສັ່ງ

"ls /dev | grep vid" (ຮູບຖື 5.19).



```
pi@raspberrypi:~ $ sudo modprobe bcm2835-v4l2
pi@raspberrypi:~ $ ls /dev | grep vid
video0
pi@raspberrypi:~ $
```

Verifying that USB video device file for Raspberry Pi camera is present inside
/dev directory

5.10 ການຈົດຈຳໃບໜ້າໂດຍໃຊ້ Raspberry Pi

ມີຫລາຍວິທີໃນການຕຽບຈັບໃບໜ້າ ເຊັ່ນ Haar cascades, HOG + Linear SVM ແລະ CNN

ຂັ້ນຕອນການຈົດຈຳໃບໜ້າ

1. ກຳນົດຄໍາ Picamera ໂດຍໃຊ້ຄໍາສັ່ງ “*sudo apt-get* ຕັດຕັ້ງ *python - picamera*”
2. ວັດທະນາການແພັກເກຈຈະບົບໂດຍໃຊ້ຄໍາສັ່ງ “*sudo aptget* ວັດທະນາ”
3. ວັດທະນາແພັກເກຈທີ່ຕັດຕັ້ງເປັນເວັບໄອຮັບຊັ້ນ ຢັກຕັດຕັ້ງ RaspbianOS ໃນ Raspberry ໄດ້ໃຊ້ຄໍາສັ່ງຕ່ອໄປນີ້ເພື່ອຕັດຕັ້ງ openCV: “*sudo apt-get* ຕັດຕັ້ງ *python-opencv*” ແລະ “*sudo pip* ຕັດຕັ້ງ *imutils*”
4. ຕັດຕັ້ງ OpenCV ມີກຳນົດຕັ້ງໄປໃນ Raspberry Pi ໃນ RaspbianOS ໃນ Raspberry ໄດ້ໃຊ້ຄໍາສັ່ງຕ່ອໄປນີ້ເພື່ອຕັດຕັ້ງ OpenCV: “*sudo apt-get* ຕັດຕັ້ງ *python-opencv*” ແລະ “*sudo pip* ຕັດຕັ້ງ *imutils*”
5. ຕັດຕັ້ງຂອງ Python ທີ່ມີ dlib ຂອງ Davis King ໃນສາກຸພແວດລ້ອມເສີມອັນ Python ເດືອນກັບກັບທີ່ຕັດຕັ້ງ OpenCV ໃນ Raspberry Pi Face Recognition ດ້ວຍຄໍາສັ່ງ “\$pip install dlib”
6. ເພີ່ມໃຊ້ pip ເພື່ອຕັດຕັ້ງໂປຣດູລ *face_recognition* ຂອງ Adam Geitgey ໃນ Raspberry Pi Face Recognition ໂດຍໃຊ້ຄໍາສັ່ງ “\$pip install face_recognition”
7. ຕັດຕັ້ງແພັກເກຈ *imutils* ຂອງພັ້ນກັບນຳນັບການສະດວກໃນ Raspberry Pi Face Recognition ດ້ວຍຄໍາສັ່ງ “\$pip install imutils”
8. ຈົກໜ້າຕ່າງເກອຮັນນັບປະກິດ Pi ໃກ້ໄພນົມຄໍາສັ່ງຕ່ອໄປນີ້ເພື່ອດຶງແລະຕັດຕັ້ງໄລຍບຣາີ RPiGPIO:

```
$ sudo apt-get
```

```
    tັດຕັ້ງ python-dev $ sudo
```

```
    apt-get
```

```
    tັດຕັ້ງ python-rpi.gpio
```

5.11 ການຕັດຕັ້ງໄດຣເວັບ I2C ໃນ Raspberry Pi

ສ່ວນນີ້ອອນຍາຍຄໍາແນະນຳກໍລະຂັ້ນຕອນໃນການຕັດຕັ້ງໄດຣເວັບ I2C ສໍາຮັບ Raspberry Pi

1. ກໍາການເສື່ອມຕ່ອງ Raspberry Pi ກັບອັນເກອຮັນເນື້ຕ
2. ໄດ້ຮັບການຕັດຕັ້ງແລ້ວໃນ Raspbian distro ໄກມ ແຕ່ໂດຍຄໍາເຮັມຕັ້ນ ຈະ ຖືກປິດໃຫ້ຈານ ຢັກຕ້ອງການເປີດໃຫ້ຈານ ໃກ້ໄສເຄື່ອງຫຍາຍ #[ຮູບ 5.20](#).

```
# blacklist spi and i2c by default (many users don't need them)
#blacklist spi-bcm2708
#blacklist i2c-bcm2708
```

ຮູບ 5.20

ໜ້າຕ່າງເພື່ອເປີດໃຫ້ຈານໄດຣເວຼອຣ I2C

3. กด "ctrl+x" ແລ້ວກด "y" ເພື່ອບັນທຶກແລະອອກ
4. ແກ້ໄຂໄຟລ໌ຂອງໂນດຸລໂດຍໃຫ້ຄໍາສັ້ງ "rudsonano /etc/modules" ແລະເພີ່ມ i2c-dev ໃນບຽບກັດໃໝ່ (ຮູບ 5.21). ກດ "ctrl+x" ຕາມດ້ວຍ "y" ເພື່ອບັນທຶກແລະອອກ
5. ຕິດຕັ້ງແພັນເກົຍ i2c-tools ດ້ວຍຄວາມຊ່ວຍເຫຼືອຂອງຄໍາສັ້ງ "*sudo apt-get install i2c-tools*"

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
# Parameters can be specified after the module name.

snd-bcm2835
i2c-dev
```

ຮູບ 5.21

i2c-dev

หากເກີດຂ້ອງຜິດພາດ 404 ໄກສາການອັປເດຕໂດຍໃຊ້ຄໍາສັ້ງ "sudo apt-get update"

6. ເຮັດໃຫ້ເພື່ອຕິດຕັ້ງ i2c-tools ອົກຮັງ

ບັນທຶກ :ການຕິດຕັ້ງຈາກໃຫ້ເວລາສັກຄຽງ ຫຶ່ນອຢູ່ກັບວ່າເຊີຣົຟເວຼອຣຢູ່ງແກ່ໄທນ

7. ຕອນນີ້ເພີ່ມຜູ້ໃຊ້ໃໝ່ໃນກຸລຸມ i2c:

sudo adduser pi i2c

8. ຮັບກົດປຶກໂດຍໃຊ້ຄໍາສັ້ງ: *sudo nano -r ຕອນນີ້*

9. ກັບຈາກຮູບຖານ ຖດສອບເພື່ອດູການເຊື່ອມຕ່ອບຂອງອຸປກຮນີໄດ້ ໂດຍ: *sudo*

i2cdetect -y 0

ສໍາຮັບຄໍາສັ້ງບອດ Rev 2: *sudo*

i2cdetect -y 1

ເນັ້ນຕ່າງຈະປາກກູ້ຂຶບ (ຮູບ 5.22).

10. ຕອນນີ້ຕິດຕັ້ງໂນດູລ ພຣອມ Python-smbus

sudo apt-get install python-smbus

11. ຕອນນີ້ Raspberry Pi ວິຊ້ອມກີຈະໃຊ້ i2c ກັບ Python ແລ້ວ

```
pangsk — pi@raspberrypi: ~ — ssh — 80x24
pi@raspberrypi ~ $ i2cdetect -y 0
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -----
10: -----
20: 20 -----
30: -----
40: -----
50: -----
60: -----
70: -----
pi@raspberrypi ~ $
```

ຮູບ 5.22

ເນັ້ນຕ່າງຈະຫັງຈາກຮູບຖານ ດ້ວຍຄວາມຫ່ວຍເຫຼືອຂອງຄໍາສັ້ງ

5.12 Serial Peripheral Interface ພັນຍາກ Raspberry Pi

Adafruit Occidental 0.2 ຮູ້ອີເມວກວ່າໄດ້ຮັບການຄໍາໜັດຄ່າລ່ວງຫຼາດຕ່າງໆຢ່າງສັບສົນຊີເຮັດອຸປະກອນຕ່ອງພ່ວງ (SPI) ສໍາຫັບ Raspbian ຈໍາເປັນຕົ້ນເປົ້າໂປ່ງແລ້ວການຄໍາໜັດຄ່າເລີກນ້ອຍ

ໄດ້ວັນຍາ I2C ໄດ້ຮັບການຕິດຕັ້ງໄວ້ລ່ວງຫຼາດໃນ Raspbian distro ໄກນ ແຕ່ຈະຄຸກປັດໃຫ້ຈານໂດຍ
ຄ່າເຮັມຕົ້ນ ທາງຕ້ອງການເປີດໃຫ້ຈານ ໃຫ້ແສດງຄວາມຄົດເກີບບຣກັດໂດຍໃສ "#" ຫ້າງຫຼາດ ເຊັ່ນ
"sudo nano /etc/modprobe.d/raspi-blacklist.conf" ແລ້ວເພີ່ມ "#" ໃນບຣກັດທີ່ສາມ ([ຮູ້ປ 5.20](#)).

1. ແກ້ໄປໄຟລ໌/ຊລ໌/ໄນດູລໂດຍໃຫ້ຄໍາສັ່ງ "sudo nano /etc/modprobe.d/raspi-blacklist.conf" ແລ້ວເພີ່ມ "#" ລົງໃນ "blacklist spi-bcm2408" ຊຶ່ງຈະກລາຍເປັນ "#blacklist spi-bcm2408"
2. ຄຸນສມບັດ SPI ຂອງ Raspberry Pi ຮອງຮັບໂດຍໄລຍບຣາຣີ Python ຊຶ່ງດຳເນັບການສໍ້ອາສ SPI ກັບໂປຣແກຣມ Python ໃນການຕິດຕັ້ງ ໃຫ້ຕິດຕັ້ງ Git ແລ້ວອອກຄໍາສັ່ງຕ່ອໄປນີ້:

```
$ cd ~  
$ sudo apt-get install python-dev  
$ git clone git://github.com/doceme/py-spidev $  
cdpy-spidev/  
$ sudo python setup.py install
```
3. ຮູ່ມູຕ Pi ແລ້ວພັນຍາກສໍາຫັບ SPI

5.13 ການເຂັ້ມໂປຣແກຣມ Raspberry Pi

ໃນການເຂັ້ມໂປຣແກຣມ Raspberry Pi ໂທມດັບສາມາຮຄໍາໜັດໄດ້ສອງວິຣີກົ້ວ່າ BCM ແລະ BOARD BCM ຢ່ອມຈາກ Broadcom SOC channel ແລະ BCM ມາຍຄົ່ງໜ້າຍເລຂ GPIO (ເຊັ່ນ GPIO21, GPIO22 ເປັນຕົ້ນ) BOARD ມາຍຄົ່ງໜ້າຍເລຂພັນຂອງບອർດ Pi

ຄໍາສັ່ງ:

importRPi.GPIO ເປັນ X

ນີ້ຄ້ອກເຮັດກົ້ງມູດ GPIO ຂອງ Raspberry Pi ຕາມຊື່ "X" ກ່ຽວຂ້ອງຄໍາສັ່ງ

ເວລາຫຼາເຂົ້າ

ລັບເຂົ້າໄລຍບຣາຣີ Time ເພື່ອໃຫ້ເຮົາສາມາຮຄໍາໜັດສົກລົງໄດ້ໃນກາຍຫລັງ

X.setmode(X.BCM)

ພັບເຮີຍກວ່າໜ້າຍເລຂ GPIO ເນື່ອງຈາກເລືອກໂທມດ BCM

X.setwarnings(ເກົ່າ)

ຄໍາສັ່ງນີ້ໄປພັບຜ້ອມຄວາມເຕືອນ GPIO ບນຫຼາຈອ

X.setup(21,X.ອອກ)

ນີ້ແມ່ນກໍາເນົດ GPIO 21 ເປັນພິບເອາະພູຕ

ພິມພົມ ("ສອງ LED ON")

ນີ້ແມ່ນພິມພົມຕົວຢ່າງ

X.output(21, True) ຮັບອຸປະກອດ X.output(21, ສູງ)

ນີ້ແມ່ນກໍາເນົດ GPIO 21 pin "on" ເວລາ

ນອນ(1)

ເພື່ອຫຼຸດໂປຣແກຣມ Python ຫົວຄາວເປັນເວລາ 1

ວິນາກີ ຍັກເວັນແປ່ນພິມພົມຈັງຫວະ:

#ຈົບໂປຣແກຣມດ້ວຍຄີ່ຍບ່ອຮົດ GPIO.

ກາຮສ້າງຂ້ອນມູລ ()

ຮຶນເຊື່ອກາຮຕັ້ງຄ່າ GPIO

5.14 ເລັນກັບ LED ແລະ Raspberry Pi

ກາຮເຊື່ອມຕ່ອງ LED ກັບ Raspberry Pi ເປັນວິທີກໍ່ຈ່າຍກໍ່ສຸດໃນກາຮກໍາຄວາມເຂົ້າໃຈຄໍາສັ້ງພື້ນຫຼານ
ວິຈະຈະປະກອບດ້ວຍໄວ LED ສອງດົວ ຕັ້ງຕ້ານການ 330 ໂອກນສອງຕັ້ງ ແລະສາຍຈັນເປັນໄວ
ເພື່ອໃຫ້ເຂົ້າໃຈກໍາເນົດ LED ໃຫ້ເຊື່ອມຕ່ອງຂັ້ນວົກຂອງ LED1 ກັບ GPIO21 ແລະ LED2 ກັບ
GPIO22 ແລະຂັ້ນລົບກັບພື້ນ ເຊີຍນສູດທີ່ອົງບາຍໄວ້ໃນມາດຕາ 5.14.1 ແລະສັງເກດກາຮກະພົບຂອງ
ໄວ LED

ຮູບຖື 5.23 ແສດງແພນກາພວງຈອງ Raspberry Pi ກີ່ເຊື່ອມຕ່ອງກັບ LED

5.14.1 ສູດຮ່າຍກັບ LED Blink

```
importRPi.GPIO ເປັນ RAj
ນໍາເຂົ້າວລາຮອ
LED_PIN1=20 # ກໍາເນົດ LED_PIN1 ໃຫ້ຂາ 20 LED_PIN2=21 #
ກໍາເນົດ LED_PIN1 ໃຫ້ຂາ 21 RAj.setmode(RAJ.BCM) # ໃຫ້ພິບ pi
ໃນໂທນດ BCM RAj.setwarnings(False) # ລົບຄໍາເຕືອນ
RAj.setup(LED_PIN1,RAJ .OUT) # ກໍາພິບ 21 ເພື່ອສ່າງອອກ
RAj.setup (LED_PIN2,RAJ.OUT) # ກໍາພິບ 20 ເພື່ອເອາກພູດ ລອງ:
```

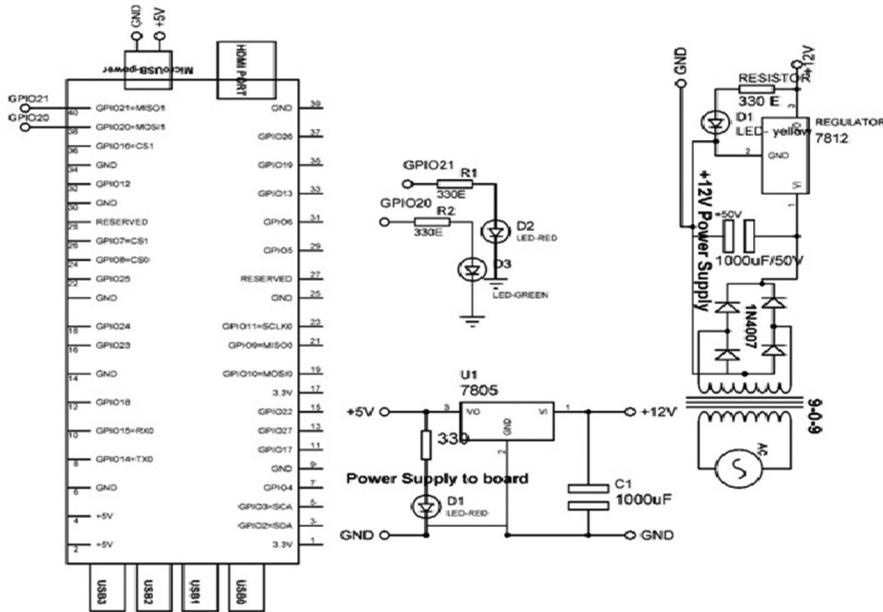
ໃນຂນະກິຈົງ:

RAj.output(LED_PIN1, True) # ກໍາພິບ 20 ເປັນ HIGH

RAj.output(LED_PIN2, True) # ກໍາພິບ 21 ເປັນ HIGH

ພິບຊານຂອງ Raspberry Pi

65



ຮັບ 5.23
 ແຜນກາພວງຈະຂອງ Raspberry Pi ທີ່ເຊື່ອມຕ່ອກັບ LED

```
print("ສອງ LED ON") # ພິມພົດຕົງບະນເກອຮີມີນັດ
wait.sleep(1)
RAJ.output(LED_PIN1, False) # ກໍາພິນ 20 ຄົງ LOW
RAJ.output (LED_PIN2, ເກົ່າ) # ກໍາພິນ 20 ຄົງ LOW ພິມວ ("ສອງ LED OFF") # ພິມພົດຕົງບະນເກອຮີມີນັດ wait.sleep (1)
```

ຍກເວັນແປ່ນພິມພົດຈັດຈະວະ:

```
RAJ.cleanup()
```

5.14.1.1 ສູດສໍາຫັນ LED Blink ໄດ້ໃຫ້ Function

```
importRPi.GPIO เป็น RAJ # RPi.GPIO ສາມາຄຣເຢກເປັນ GPIO
import time เป็น wait # import time library
ledPin1 = 21 # ເຊື່ອມຕ່ອງ led1 ບນຫາ 21
ledPin2 = 20 # ເຊື່ອມຕ່ອງ led2 ບນກາຮຕັ້ງຄ່າ
def 20 ຂາ ():
    RAJ.setmode(RAJ.BOARD)      # GPIO ກາຮກໍາຫຼັດໝາຍເລຂພິນ
    RAJ.setup (ledPin1, RAJ.OUT)  # ຕັ້ງ ledPin1 ເປັນເອາຕົ່ມ
```

RAJ.setup (ledPin2, RAJ.OUT) # ຕັ້ງຄ່າ ledPin2 ເປັນເອາຕີພຸດ

def ຖຸກ ():

ໃນຂະນະກີຈົງ:

ພິມພົມ ("ໄວ LED ກັ້ງສອງຕົດ") # ພິມພົມຕົດຕັດ

RAJ.output (ledPin1, RAJ.HIGH) #ເປີດ LED1

RAJ.output(ledPin2, RAJ.HIGH) #ເປີດ LED2

wait.sleep(1.0) # ລ່າຍ້າ 1 ວິນາຖື

ພິມພົມ ("ໄວ LED ກັ້ງຄູ່ດັບ") # ພິມພົມຕົດຕັດ

RAJ.output (ledPin1, RAJ.LOW) #ປັດ LED1

RAJ.output(ledPin2, RAJ.LOW) #ປັດ LED2

ຮອສສັບ(1.0) #ຮອ1ວິນາຖື

ໂປຣແກຣມປ້ອງກັນ ():

RAJ.output(ledPin1, RAJ.LOW) #ປັດ LED1

RAJ.output(ledPin2, RAJ.LOW) #ປັດ LED2

RAJ.cleanup() # ທຳມະນຸຍາດ GPIO

ຄ້າ __name__ == '__main__': # ໂປຣແກຣມຫຼັກເພື່ອເຮັດໃຊ້ການຕັ້ງຄ່າ

ຟັງກົບ ()

ລອງ:

ວນ ()

ຍກເວັນKeyboardInterrupt: # ແປ່ນພິມພົມຈັດຈັງຫວະເພື່ອຫຼຸດໂປຣແກຣມ

endprogram()

5.15 ການອ່ານອັນພຸດດີຈິຕອລ

ອັນພຸດດີຈິຕອລ ທີ່ Raspberry Pi ສາມາຮັດອ່ານໄດ້ສອງວິທີທີ່ເຮັດວຽກວ່າດີ່ງລົງແລະດີ່ງຂັ້ນ ໂດຍການຕັ້ງ
ຄ່າພິບ GPIO ເປັນເອາຕີພຸດ LOW ທີ່ໃຊ້ໜານອູ່ຢູ່ຮັບເອາຕີພຸດ HIGH

importRPi.GPIO ເປັນ RAJ

RAJ.setup(23, RAJ.IN, pull_up_down=RAJ.PUD_DOWN)

RAJ.setup(24, RAJ.IN, pull_up_down=RAJ.PUD_UP)

ຄໍາສັ້ງເຫຼຳນີ້ເປີດໃຊ້ໜານຕັ້ງຕ້ານການແບບດີ່ງລົງທີ່ພິບ 23 ແລະຕັ້ງຕ້ານການແບບດີ່ງຂັ້ນທີ່ພິບ 24 Pi
ກຳລັງນອງໜາແຮງດັນສູງທີ່ພິບ 23 ແລະແຮງດັນຕໍ່າກີ່ພິບ 24 ສິ່ງເຫຼຳນີ້ຈະເປັນສໍາຮັບການກຳໜົດໃນ
ສູ່ປົກ ຕັ້ງນັ້ນ ກໍ່ສາມາຮັດວິສອນແຮງດັນພິບໄດ້ຍ່າງຕ່ອງເນື່ອງ ເພື່ອໃກ້ເຂົ້າໃຈແວວິດນີ້ ໄກສົງຈາກນາ
ໂປຣແກຣມຂາດເລັກສໍາຮັບສວິຕ່າງ

5.15.1 ສູຕ

```
importRPi.GPIO เป็น RAJ
RAJ.setmode (GPIO.BCM) # ໃຊ້ pi ໃນໂທນດ BCM
RAJ.setup (23, RAJ.IN, pull_up_down = RAJ.PUD_DOWN) # ດັ່ງພົບເປັນ
          ບ້ອນຂອ່ມຍຸລ
RAJ.setup(24, RAJ.IN, pull_up_down = RAJ.PUD_UP) # set pin as input
while True:
    ຄ້າ(RAJ.input(23) ==1):
        print("ກົດປຸນ 1") # ພິມພົຕຣີງບົນເກອຮົມບັດ
    if(RAJ.input(24) == 0):
        ພິມພົ ("ກົດປຸນ 2") # ພິມພົຕຣີງບົນເກອຮົມບັດ RAJ.cleanup
    () # ລ້າງ GPIO ກັ້ງໜດ
```

5.16 ການອ່ານອັນພຸດກີ່ກຣິກເກອຮົດໄດ້ຂອບ

ໂປຣແກຣມໃນນາຕາ 5.15 ຄັ້ອການຕຽບສອບວ່າມີການກົດປຸນໃດປຸນໜຶ່ງຫຼືອໄມ໌ ກາກປຸນເປັນກຣິກເກອຮົດການຮະກຳຫຼືອຄໍາສັ້ງເພີ່ງຄັ້ງເດືຍວ່າ ລໍາຮັບສິນນີ້ ໂຄບຮາຣ໌ GPIO ມີຟັງກໍຂັບໃນຕັ້ງສໍາຮັບການຂຶ້ນແລະລົງ ຂອບຖ້າທີ່ເພີ່ມຂຶ້ນຄູກກໍາເໜັດໂດຍກັນຖ້າທີ່ເນື້ອສັນຍານເປົ້າຢືນຈາກຕໍ່ໄປສູງ ແຕ່ສາມາຮັດຕຽບການເປົ້າຢືນແປລ່ອໄດ້ເກົ່ານັ້ນ ແນວດີ່ງຈະເກີດຂຶ້ນກັນຖ້າທີ່ເນື້ອສັນຍານຈາກສູງໄປຕໍ່າ

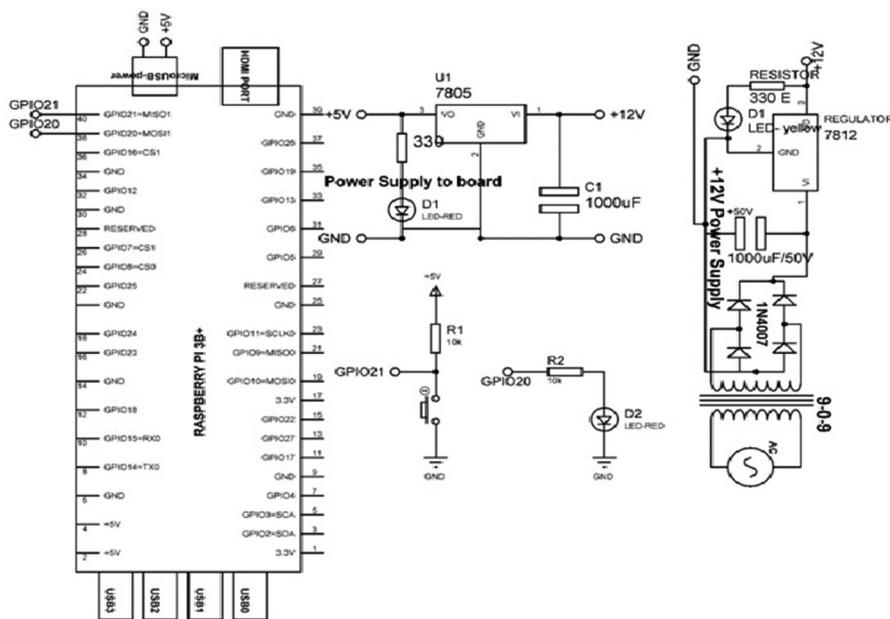
ວັງຈອນເຊື່ອມຕ່ອລົວຕີ່ກັບ Raspberry Pi ປະກອບດ້ວຍ LED ຕັ້ງຕ້ານການ ສົວຕີ່ ແລະສາຍຈັບເປື່ອ ເພື່ອໃຫ້ເຂົ້າໃຈການກຳໜານຂອງສົວຕີ່ ໃຫ້ເຊື່ອມຕ່ອລົວປະກອບຕ່າງໆ ດັ່ງແສດງໃນຮູບ 5.24. ເພີ່ມສູຕຮົດກີ່ກຣິກໄວ້ໃນສ່ວນ 5.16.1 ຜ່ານ 5.16.3 ເພື່ອສັງເກດການກຳໜານຂອງສົວຕີ່ໃນຮູບແບບຕ່າງໆ

5.16.1 ສົວຕີ່ກຳໜານໃນການກຳໜາດຄ່າແບບດີ່ງລົງ

ການເຊື່ອມຕ່ອ:

- ເຊື່ອມຕ່ອຂັ້ງໜຶ່ງຂອງສົວຕີ່ກັບກຣາວດໍາລະເກອຮົມບັດເລື່ອນ ທີ່ ກັບ GPIO21 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອຕັ້ງຕ້ານການໜຶ່ງຕັ້ງທີ່ 10 K ຮະຫວ່າງ +5 V ແລະ GPIO21 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອຂັ້ງບົກຂອງ LED ກັບ GPIO20 ຂອງ Raspberry Pi ຜ່ານຕັ້ງຕ້ານການ ແລະຂັ້ງລົບກັບກຣາວດໍາ

ຮູບ 5.24ແສດງແບບກາພວງຈຽບຂອງ Raspberry Pi ທີ່ເຊື່ອມຕ່ອກັບສົວຕີ່



ຮັບ 5.24
 ແຜນກາພວງຈະຂອງ Raspberry Pi ທີ່ເຊື່ອນຕ່ອກັບສວົດໃນການກຳເຫັນດໍາ "ດຶງລອງ"

5.16.1.1 ສູດຮັບການກຳເຫັນດໍາແບບດຶງລອງ

```

importRPi.GPIO เป็น RAj
นำเข้าເວລາທີ່ຮອງ # ເພີ່ມເວລາ lib button_pin = 21 # ກຳເຫັນ
ສ້ອ button_pin ໃຫ້ກັບ GPIO21 led_pin = 20 # ກຳເຫັນສ້ອ
led_pin ໃຫ້ກັບ GPIO20 RAj.setmode(GPIO.BCM) # ໃຊ້
Pi pins เป็น BCM
RAj.setup (button_pin, RAj.IN, pull_up_down = RAj.PUD_UP) # GPIO
ເປັນອົບພຸຕ
RAj.setup (led_pin, RAj.OUT) # ກໍາໃຫ້ led_pin ເປັນເອົາຕົວພຸຕ
ລອງ:
ໃນຂນະກິຈົງ:
button_var = RAj.input(button_pin)# ອ່ານພັນ 21
ifbutton_var== ເກົ່າ:
    RAj.output(led_pin, True) # make led_pin to HIGH
    print("Switch Pressed..") # print string on terminal
    wait.sleep(0.2) # delay of 200mSec
ອັນ:

```

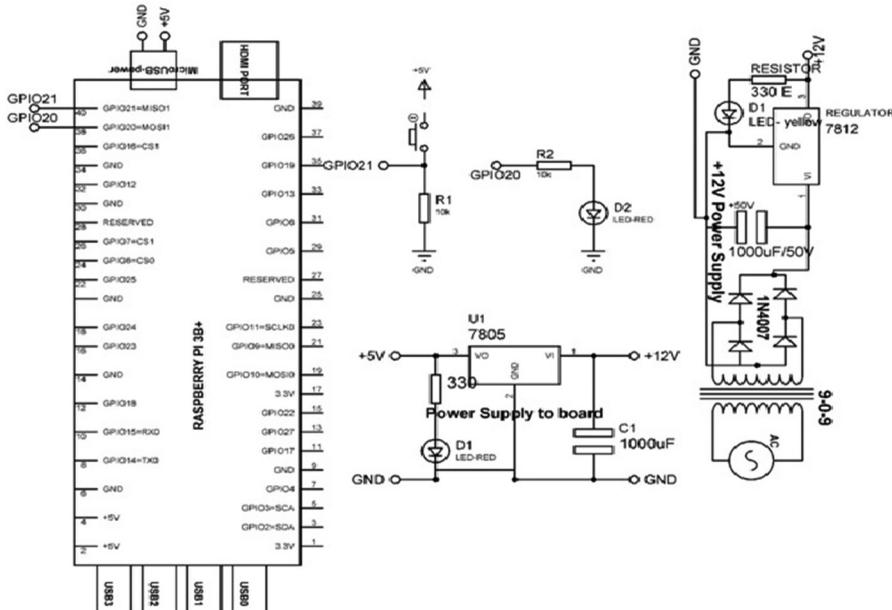
```
RAJ.output (led_pin, False) # make led_pin to LOW
print ("ໄປໄດ້ຄົດສົຕື່ຈ ... ") # ພິມພົດຕົງບັນເກອຮົມບັນດຸ
wait.sleep (0.2) # ລ້າໜ້າ 200mSec
ຍກເວັນແປ່ນພິມພັດຈັງຫວະ: ພິມພົດຕົງບັນເກອຮົມບັນດຸ
ການຫັດຈັງຫວະແປ່ນພິມພົດ") RAJ.cleanup () //
ລ້າງ GPIO ກັ່ງໜູນດ
```

5.16.2 ສົຕື່ການອ່ານໃນການກໍາໜັດຄໍາແບບດົງຂຶນ

ການເຊື່ອນຕ່ອ:

- ເຊື່ອນຕ່ອຫັນໜີ້ຂອງສົຕື່ໄປກີ +5 V ແລະ ອັກຫັນໜີ້ກັບ GPIO21 ຂອງ Raspberry Pi
- ເຊື່ອນຕ່ອຕັ້ງຕ້ານການໜີ້ຕັ້ງກີ 10 K ຮະຫວ່າງກຽວດີກັບ GPIO21 ຂອງ Raspberry Pi
- ເຊື່ອນຕ່ອຫັນວິທະຍາກຂອງ LED ກັບ GPIO20 ຂອງ Raspberry Pi ຜ່ານຕັ້ງຕ້ານການ ແລະ ຊັ້ນລົບກັບກຽວດີ

ຮູບກີ 5.25 ແສດງແຜນກາພວງຈອນຂອງ Raspberry Pi ທີ່ເຊື່ອນຕ່ອກັບສົຕື່



ຮູບກີ 5.25
 ແຜນກາພວງຈອນຂອງ Raspberry Pi ທີ່ເຊື່ອນຕ່ອກັບສົຕື່ໃນການກໍາໜັດຄໍາ "Pull-up"

5.16.2.1 ສູດຮ່າຍການກຳໜັດຄໍາແບບດົ່ງນີ້

```
importRPi.GPIO เป็น RAj
```

```
เวลาบໍາເຫຼົາເປັນຮອ button_pin = 21 # ກຳໜັດຊື່ໃກ້ກັບພິນ 21
```

```
led_pin = 23 # ກຳໜັດຊື່ໃກ້ກັບພິນ 23 RAj.setmode
```

```
(RAJ.BCM) // ໃຊ້ພິນຂອງ pi ໃນໂທມດ BCM
```

```
RAJ.setup (button_pin, RAJ.IN, pull_up_down = RAJ.PUD_DOWN)
```

```
# ຕັ້ງພິນ 21 ເປັນອົບພຸດ
```

```
RAJ.setup(led_pin,RAJ.OUT) # set pin 23 as output ລອງ
```

```
:
```

ໃນຂນະກິຈີ:

```
button_VAR = RAJ.input(button_pin) # read switch
```

```
ifbutton_VAR == True:
```

```
RAJ.output(led_pin, True) # make pin 23 to HIGH
```

```
print("button pressed...") # print string on terminal
```

```
wait.sleep(0.2)
```

ອືບ:

```
RAJ.output(led_pin,False) # make pin 23 to LOW
```

```
print("button not pressed...") # print string on terminal
```

```
wait.sleep(0.2)
```

ຢັກເວັນແປ່ນພິມພັດຈັງຫວະ ພິມພົມ 'ຕຽວຈັບ

ກາຮັດຈັງຫວະຂອງແປ່ນພິມພົມ' RAj.cleanup

```
()
```

5.17 ການເຊື່ອນຕ່ອງເລີຍກັບ Raspberry Pi

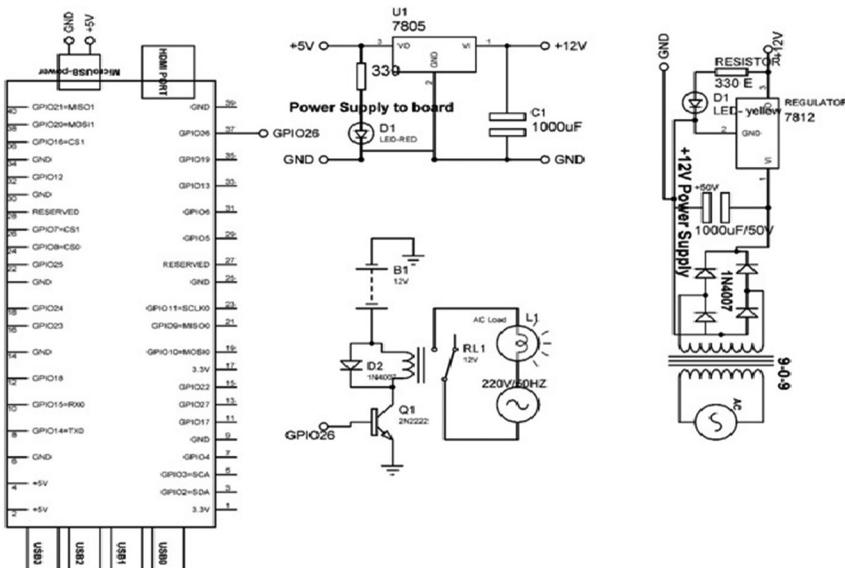
ຮັບເລີຍເປັນອຸປະກອນທີ່ໃຊ້ໃນການປັບປຸງສຳຄັນຂອງລະບົບໄຟຟ້າ ສາມາດໃຊ້ເພື່ອປັບປຸງສຳຄັນຂອງ
ເຄື່ອງໃຊ້ກາຍໃນບ້ານທີ່ກຳໜັດດ້ວຍແຮງດັບໄຟຟ້າກະແສສລັບ 220 ໂວລັດ ເພື່ອໃກ້ເຂົ້າໃຈຄົງການເຊື່ອ^{5.26}
ຕ່ອງເລີຍກັບ Raspberry Pi ຄວາມຄວາມນາງຈອງຢ່າຍໆ ສໍາຫັບການກຳໜັດໄຟ "ON" ແລະ
"OFF" ກໍາການເຊື່ອນຕ່ອງສ່ວນປະກອບຕາມທີ່ແສດງໃນຮູບກໍ 5.26. ກຣາບເຊີສເຕອຣ 2N2222 ກໍາ
ບ້ານທີ່ເປັນສົວິຕົກທີ່ນີ້ ເນື້ອໄປຄົງ "1" ທີ່ສ້າງ ບັນຈະກຳບ້ານທີ່ເປັນສົວິຕົກປັດ ແລະ ເນື້ອຄົງ "0" ທີ່ສ້າງ ມັນ
ຈະກຳບ້ານທີ່ເປັນສົວິຕົກຕ່າງໆ

ການເຊື່ອນຕ່ອງ:

- ເຊື່ອນຕ່ອງ "ສ້າງ" ຂອງກຣານເຊີສເຕອຣ 2N2222 ກັບ GPIO26 ຂອງ Raspberry Pi
- ຕ່ອງ "ອົບແອລ" ຂອງກຣານເຊີສເຕອຣກັບກຣາວົດ

ພື້ນຖານຂອງ Raspberry Pi

71



ຮູບ 5.26

ແຜນກາພວງຈອສໍາຫັກເຊື່ອນຕ່ອຮີເລຍີກັບ Raspberry Pi

- ເຊື່ອນຕ່ອ “collector” ຂອງກຣານຊເສເຕອຣກັບ “L2” ຂອງຮີເລຍີ
- ຕ່ອບັນຍາກຂອງແບຕເຕອຣ +12 V ເຂົາກັບຮີເລຍີ “L1”
- ເຊື່ອນຕ່ອໄດໂອດ 1N4004 ກັບ “L1” ແລະ “L2”
- ເຊື່ອນຕ່ອບັນຍາໄຟຟ້າກຮະແສສລັບໜຶ່ງບັນຍາກັບ “ກົ້ວໄປ” ຂອງຮີເລຍີ ແລະ ອັກບັນຍາກັບບັນຍາຂອງໄລດ້ໄຟຟ້າກຮະແສສລັບ (ຫລອດໄຟ)
- ເຊື່ອນຕ່ອບັນຍາອັນຂອງໄລດ້ກຮະແສສລັບກັບບັນຍາ “NO” ຂອງຮີເລຍີ

5.17.1 ສູດຮ

```
importRPi.GPIO as MAHE # ບໍາເຂົາ pi GPIOs lib ບໍາ  

    ເຂົາຕາມເວລາຮອ # ຮອເວລາ lib  

    MAHE.setmode(MAHE.BCM) # ໃຊ້ພົນ pi ໃນໂຄນດ BCM  

    MAHE.setup(26, MAHE.OUT) # ດັ່ງຄ່າພົນ 26 ເປັນພົນເອົາດົກໃນ  

    ຂະນະກີ (ຈົງຈົງ):  

    GPIO.output(26, MAHE.HIGH) // ຖໍາ pin 26 pin ເພື່ອພົນພົນ HIGH  

    ('relay ON') # ພົນພົຕົງບນເກອຮີມັນນັດ  

    wait.sleep(5) # ລໍາສ້າງ 5 ວິນາກີ GPIO.output(26,  

    GPIO.LOW) ພົນພົນ ('relay OFF') # ພົນພົຕົງບນ  

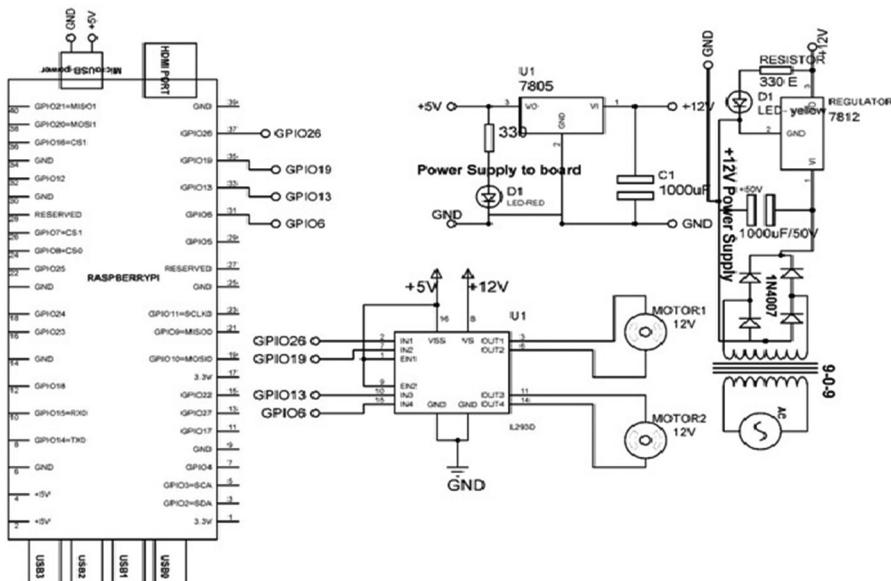
    ເກອຮີມັນນັດ wait.sleep(5) # ລໍາສ້າງ 5 ວິນາກີ
```

5.18 ການເຊື່ອມຕ່ອ DC Motor ກັບ Raspberry Pi

ມອເຕອຣກະແສຕຽນແປລັງພລັງຈານໄຟຟ້າເປັນພລັງຈານກລ ເນື້ອກະແສໄຟຟ້າໄຫດຜ່ານຂດລວດໃນສານແມ່ເໜັກ ແຮງແມ່ເໜັກຈະຖຸກສ້າງຂຶ້ນເພື່ອສ້າງແຮງບົດໃນນອເຕອຣກະແສຕຽນ ເພື່ອໃຫ້ເຂົ້າໃຈກົ່າການເຊື່ອມຕ່ອຂອງນອເຕອຣກະແສຕຽນກັບ Raspberry Pi ຄວາມພັງຄານນາງຈອງຍ່າງໆ ເພື່ອໃຫ້ມອເຕອຣເຄລື່ອນທີ່ໃນກົດການ "ຕາມເບີນນາພິກາ" ແລະ "ກວບເບີນນາພິກາ" ວັຈຣປະກອບດ້ວຍ Raspberry Pi, ໄດ້ເວັບນອເຕອຣ L293D, ມອເຕອຣ DC ສອງຕົ້ງ ແລະ ແກ່ລ່າງຈ່າຍໄຟ ກໍາການເຊື່ອມຕ່ອສ່ວນປະກອບຕາມທີ່ແສດງໃນ **ຮູບກໍ 5.27**.

ການເຊື່ອມຕ່ອ:

- ເຊື່ອມຕ່ອ pin2 (IN1) ຂອງ L293D ກັບ GPIO26 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອ pin7 (IN2) ຂອງ L293D ກັບ GPIO19 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອ pin10 (IN3) ຂອງ L293D ກັບ GPIO13 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອ pin15 (IN4) ຂອງ L293D ກັບ GPIO6 ຂອງ Raspberry Pi
- ຕ່ອ pin3 (OUT1) ຂອງ L293D ເຂົ້າກັບບັບບັງນັ້ນຂອງ motor1
- ຕ່ອ pin6 (OUT2) ຂອງ L293D ເຂົ້າກັບບັບບັງອັນຂອງ motor1



ຮູບ 5.27

ແຜນກາວງຈຈໍາຫັກເຊື່ອມຕ່ອນອເຕອຣກະແສຕຽນກັບ Raspberry Pi

- ຕ່ອ pin11 (OUT3) ຂອງ L293D ເຊົາກັບຫັ້ວໜີ້ງຂອງ motor2
- ຕ່ອ pin14 (OUT4) ຂອງ L293D ເຊົາກັບຫັ້ວອືບຂອງ motor2.
- ເຊື່ອນຕ່ອ pin8 ຂອງ L293D ກັບ +12 V DC
- ເຊື່ອນຕ່ອພິບ 1, 9 ແລະ 16 ຂອງ L293D ກັບ +5 V DC
- ເຊື່ອນຕ່ອມູດ "GND" ຂອງ L293D ກັບກຣາວດ

5.18.1 ສູດຮ

```
importRPi.GPIO เป็น MAHI #import GPIO เป็น MAHI
นำເຂົາຕາມເວລາຮອ #include time library M1_PIN1=26 #
ກຳທັດຕົວແປຣໃຫ້ກັບ GPIO26 M1_PIN2=19 # ກຳທັດ
ຕົວແປຣໃຫ້ກັບ GPIO19
M2_PIN1=13 # ກຳທັດຕົວແປຣໃຫ້ກັບ GPIO13
M2_PIN2=6 # ກຳທັດຕົວແປຣໃຫ້ກັບ GPIO6
```

```
MAHI.setmode(MAHI.BCM) # set pi เป็น BCM MAHI.setup
(M1_PIN1,MAHI.OUT) #setting GPIO26 เป็น OUTPUT MAHI.setup
(M1_PIN2,MAHI.OUT) #setting GPIO19 เป็น OUTPUT MAHI.setup
(M2_PIN1, MAHI .OUT) #setting GPIO13 เป็น OUTPUT
MAHI.setup(M2_PIN2,MAHI.OUT) #setting GPIO6 เป็น OUTPUT
```

ລອງ:

ໃນຂນະກິຈົງ:

```
MAHI.output(M1_PIN1, True) # ກຳພິບ 26 ຄົງ HIGH
MAHI.output(M1_PIN2, ເກົ່າ) # ກຳພິບ 19 ຄົງ LOW
MAHI.output(M2_PIN1, ຈົງ) # ກຳພິບ 13 ເປັນ HIGH
MAHI.output (M2_PIN2, ເກົ່າ) #make pin 6 to LOW
print("Toth Motor Forward") #print string on terminal
wait.sleep(10) # delay 10 ວິນາກີ
MAHI.output(M1_PIN1, False) #ກຳພິບ 26 ຄົງ LOW
MAHI.output(M1_PIN2, ຈົງ) #ກຳພິບ 19 ຄົງ HIGH
MAHI.output(M2_PIN1, ເກົ່າ) #ກຳພິບ 13 ຄົງ LOW
MAHI.output(M2_PIN2, ຈົງ) #make pin 6 to HIGH
print("ກັ້ງສອງນອເຕອຣຍ້ອນກລັບ") #print string uu terminal
wait.sleep(10) # ລໍາຊ້າ 10 ວິນາກີ
```

ຍກເວັນແປ້ນພິບພັດຈັງຫວະ:

```
MAHI.cleanup() # ລ້າງ GPIO ກັ້ງທັດ
```

5.18.2 ສູຕຣຄວບຄຸມນອເຕອຣ໌ທີ່ນຶ່ງຕັ້ງໄດ້ໃຫ້ຝັກໜັນ

```
importRPI.GPIO เป็น MAHI
ນໍາເຂົາເວລາຮອ
# ພັນສໍາຮັບອັນພຸດໄດ້ຮວ່ອນນອເຕອຣ໌ M11 = 26
#ກຳນົດຕັ້ງແປຣໃຫ້ກັບ GPIO26 M12= 19 #
ກຳນົດຕັ້ງແປຣໃຫ້ກັບການຕັ້ງຄ່າ def ຂອງ GPIO19
():
    MAHI.setmode (MAHI.BCM) # ເລືອກໂໜ່ດ BCM
    MAHI.setup (M11, MAHI.OUT) # ກໍາໃຫ້ M11 ເປັນເອາະພຸດ
    MAHI.setup (M12, MAHI.OUT) # ກໍາໃຫ້ M12 ເປັນເອາະພຸດ
    MAHI.setup (M1E, MAHI. OUT) # ກໍາໃຫ້ M1E ເປັນເອາະພຸດ

def ຮູ່ປຸ ():
    #ການເຄລືອນກໍໄປໜ້າງໜ້າ
    MAHI.output(M11,MAHI.HIGH) # set M11 to HIGH
    GPIO.output(M12,MAHI.LOW) # set M12 to LOW
    GPIO.output(M1E,MAHI.HIGH) # set M1E to HIGH
    wait.sleep(5 ) // ນັ່ວງເວລາ 5 ວິນາຖື
        #ສໍາຮັບການເຄລືອນໄວ້ຍ້ອນກລັບ
    MAHI.output(M11,MAHI.LOW) # set M11 to LOW
    MAHI.output(M12,MAHI.HIGH) # set M12 to HIGH
    MAHI.output(M1E,MAHI.HIGH) # ຕັ້ງຄ່າ M1E ເປັນ HIGH
    wait.sleep(5 ) // ນັ່ວງເວລາ 5 ວິນາຖື
        #ສໍາຮັບຫຍຸດການເຄລືອນໄວ້
    MAHI.output(M1E,MAHI.LOW) # ຕັ້ງຄ່າ M1E ເປັນ LOW

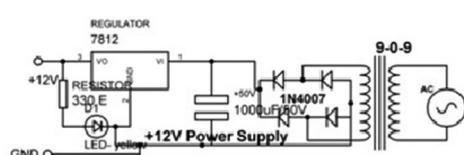
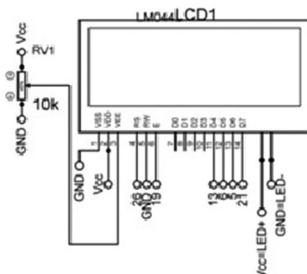
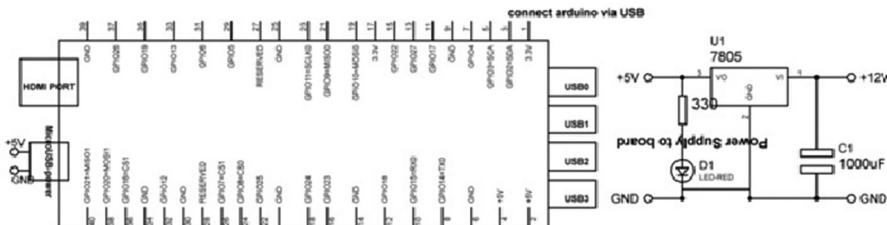
def ກໍາລາຍ ():
    MAHI.cleanup() # ລ້າງ GPIOs
    ຄ້າ __name__ == '__main__': # ໂປຣແກຣມເຮັມເຮັນຕັ້ນຈາກການຕັ້ງ
    ຄ່າກີ່ນ ()
    ລອງ:
    ວນ ()
    ຍກເວັບKeyboardInterrupt:
    destroy() # stop the operation
```

5.19 ການເຊື່ອມຕ່ອ LCD ກັບ Raspberry Pi

ຈຳແສດງພລຄຣີສຕ້ລ໌ເໜວເປັນອປກຮນີແສດງພລ ມັນໄມ່ເປົ່າງແສງ ແຕ່ມີແສງສີດຳເພື່ອສ້າງກາພ
ແກນ ສາມາຄໃຊ້ເພື່ອແສດງຂ້ອງຄວາມຮ້ອງຂ້ອງມູລກາງປະສາກສັນພັສໃນຮບບ ວຈປປກອບດ້ວຍ
Raspberry Pi, LCD ແລະ ແກ່ລ່າງຈ່າຍໄວ ກໍາການເຊື່ອມຕ່ອສ່ວນປະກອບຕາມກີແສດງໃນຮູບ 5.28.

ການເຊື່ອມຕ່ອ:

- ເຊື່ອມຕ່ອ pin1 (Vss), pin5, (RW) ແລະ pin16 (LED-) ຂອງ LCD ກັບກາວົດ
- ເຊື່ອມຕ່ອ pin2 (Vcc) ແລະ pin15 (LED+) ຂອງ LCD ເຂົ້າກັບ +5 V DC
- ເຊື່ອມຕ່ອ pin3 (V_{EE}) ຂອງ LCD ເພື່ອເອາກົກຸຽໂພເກນຂອມືເຕອຣ 10 K ແລະ ຕ່ອຂັ້ງໂພ
ເກນຂອມືເຕອຣອັກສອງຂັ້ງເຂົ້າກັບ +5 V DC ແລະ ກາວົດ
- ເຊື່ອມຕ່ອ pin4 (RS) ຂອງ LCD ກັບ GPIO26 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອ pin6 (E) ຂອງ LCD ກັບ GPIO19 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອ pin11 (D4) ຂອງ LCD ກັບ GPIO13 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອ pin12 (D5) ຂອງ LCD ກັບ GPIO6 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອ Pin13 (D6) ຂອງ LCD ກັບ GPIO5 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອ Pin14 (D7) ຂອງ LCD ກັບ GPIO21 ຂອງ Raspberry Pi



ຮູບ 5.28

ແພນກາພວງຈະສໍາຮັບເຊື່ອມຕ່ອ LCD ກັບ Raspberry Pi

5.19.1 ມີຫຼັງສຸດ Adafruit ສໍາເຫັນ LCD

ດໍາເນີນການຄໍາສັ້ນໃນເຖິງນິບັດ Raspberry Pi ເພື່ອຕິດຕັ້ງແລະອັປເດຕໄລຍບຣາຣີ

1. sudo apt-get update
2. sudo apt-get tິດຕັ້ງ build-essential python-dev python-smbus
python-pip git
3. sudo pip tິດຕັ້ງ RPi GPIO

ລະເວັບຄໍາເຕືອນເກີຍກັບການພຶ້ງພາໄດ້ ຖໍ່ Adafruit ມີຫຼັງສຸດເພື່ອໃຫ້ຈານ LCD ໃນໂຄມດ 4 ບັດ

ຂັ້ນຕອນໃນການຕິດຕັ້ງໄລຍບຣາຣີ Raspberry Pi:

ຂັ້ນຕອນທີ 1:ຕິດຕັ້ງ Git ແລ້ວ Raspberry Pi ໂດຍໃຊ້ຄໍາສັ້ນ: sudo apt-get
tິດຕັ້ງ git

Git ຂ່າຍໃນການໂຄລນໄຟລ໌ໂຄຮງກາຣບນ github ແລະໃຊ້ກັບ Raspberry Pi

ຂັ້ນຕອນທີ 2:ໃນການສ້າງໂຄລນຂອງໄຟລ໌ໂຄຮງກາຣບນ Pi ໄກສະກຳຄໍາສັ້ນ:

sudo git clone https://github.com/adafruit/Adafruit_Python_CharLCD

ຂັ້ນຕອນທີ 3:ເປີເລີຍໄດ້ເຮັດວຽກກັດດ້ວຍຄໍາສັ້ນ: cd Adafruit_
Python_CharLCD

ຂັ້ນຕອນທີ 4:ເປີດໄດ້ເຮັດວຽກ ຈະນີໄຟລ໌ຊ່ອ setup.py. ຕິດຕັ້ງ
ໂດຍໃຊ້ຄໍາສັ້ນ: sudo python setup.py tິດຕັ້ງ

5.19.2 ສູດຮກັບ Adafruit Library

```
#ກຳບັດວັດຖຸແລະເຂື່ອນຕ່ອ່າ LCD RS=26, E=19, D4=13, D5=6, D6=5 ແລະ  
D7=21
```

```
RAJ_LCD = Adafruit_CharLCD(rs=26, en=19, d4=13, d5=6, d6=5, d7=21,  
cols=16, spi=2)
```

```
RAJ_LCD.clear()# clear the contents of LCD
```

```
RAJ_LCD.message('Welcome') # print string on LCD
```

```
wait.sleep(3) # delay of 3 ວິນາກີ
```

```
RAJ_LCD.clear() # clear the contents of LCD
```

```
RAJ_LCD.message(' ໄສ່ຂ່ອງຄວາມໃດ ທີ່ນີ້ຈາກແປ່ນພິກ')
```

```
RAJ_LCD.message('ຂ້ອງຄວາມ')
```

```
wait.sleep(5) # ລ້າໜ້າ 5 ວິນາກີ RAJ_LCD.clear() #
```

```
ລ້າງເນື້ອຫາຂອງ LCD
```

5.19.3 ສູຕຣ໌ໃນມີຫ້ອງສຸມຸດ

```
importRPi.GPIO เป็น RAJ
```

```
เวลาบໍາເຫຼັກຄະຮອ RAJ_LCD_RS = 26 # RS ພິບຂອງ LCD RAJ_LCD_E = 19  
# E ພິບຂອງ LCD RAJ_LCD_D4 = 13 # D4 ພິບຂອງ LCD RAJ_LCD_D5 = 6  
# D5 ພິບຂອງ LCD RAJ_LCD_D6 = 5 # D6 ພິບຂອງ LCD RAJ_LCD_D7 =  
21 # D6 ພິບຂອງ LCD RAJ_LCD_D7 = 21 ຂອງ LCD RAJ_LCD_WIDTH =  
16 # ດຳນັກຕົວກັບ 16 RAJ_LCD_CHR = True # ກຳເຫັນດີເລີນ  
RAJ_LCD_CMD = False # ກຳເຫັນດີເລີນ RAJ_LCD_LINE_1 = 0x80 # ທີ່ອຍ່າ  
RAM ຂອງ LCD ສໍາຫັບບຣກັດແຮກ RAJ_LCD_LINE_2 = 0xC0 # ທີ່ອຍ່າ RAM  
ຂອງ LCD ສໍາຫັບບຣກັດກໍ່ສອງ RAJ_E_PULSE = 0.0005 # ໄກສະເໜີ
```

```
RAJ_E_DELAY = 0.0005 # ໄກສະເໜີ
```

```
def ກັກ ():
```

```
RAJ.setwarnings (False) # ລົບຄໍາເຕືອນ RAJ.setmode (RAJ.BCM) # ໃຊ້  
ໝາຍເລຂ BCM GPIO RAJ.setup (RAJ_LCD_E, RAJ.OUT) # ຕັ້ງຄໍາໃຫ້ເປີດ  
ໃຊ້ງານພິບເປັນເອົາຕົວ RAJ.setup (RAJ_LCD_RS, RAJ.OUT) # ຕັ້ງຄໍາພິບ  
RS ເປັນເອົາຕົວ RAJ.setup (RAJ_LCD_D4, RAJ.OUT) # ຕັ້ງຄໍາ D4 ພິບເປັນ  
ເອົາຕົວ RAJ.setup (RAJ_LCD_D5, RAJ.OUT) # ຕັ້ງຄໍາ D5 ພິບເປັນ  
ເອົາຕົວ RAJ.setup (RAJ_LCD_D6, RAJ.OUT) # ຕັ້ງຄໍາ D6 ພິບເປັນ  
ເອົາຕົວ RAJ.setup (RAJ_LCD_D7, RAJ.OUT) # ຕັ້ງຄໍາ D7 ພິບເປັນ  
ເອົາຕົວ RAJ_lcd_init () # ເຮັດວຽກ LCD
```

ໃນຂະໜາດ:

```
RAJ_lcd_string("R Singh ,LCD_LINE_1) # ພິມພື້ນຕົວໄປກໍ ROW 1  
ຂອງ LCD
```

```
RAJ_lcd_string(" Presents ",LCD_LINE_2) # ພິມພື້ນຕົວໄປກໍ ROW2  
ຂອງ LCD
```

```
wait.sleep(3) # ລ່າຍ້າ 3 ວິນາກີ
```

```
RAJ_lcd_string("123456789012345",LCD_LINE_1)
```

```
RAJ_lcd_string("ABCDEFGHIJK",LCD_LINE_2)
```

```
wait.sleep(3) # ລ່າຍ້າ 3 ວິນາກີ
```

```
defRAJ_lcd_init():
    RAJ_lcd_display(0x28,RAJ_LCD_CMD) # ເລືອກໂທນັດ 4 ພົດແລະແຄວ
    RAJ_lcd_display(0x0C,RAJ_LCD_CMD) # ຄົ່ງບນຈອແສດງຜລໄນ໌ມີເຄອຮ່ອງ
    ກະພຽບຕາ
    RAJ_lcd_display(0x01,RAJ_LCD_CMD) # ລ້າງເນື້ອຫາຂອງ LCD Wait.sleep
    (RAJ_E_DELAY) # ລ້າໜ້າ

    defRAJ_lcd_display(bits, mode): RAJ.output(RAJ_LCD_RS,
    mode) # set Mode RAJ.output(RAJ_LCD_D4, False) # ທຳ D4 pin
    ຂອງ LCD LOW RAJ.output(RAJ_LCD_D5, False) # ທຳ D5 pin ຂອງ
    LCD LOW RAJ เອາຕີພູດ (RAJ_LCD_D6, ເຖິງ) # ທຳພັນ D6 ຂອງ LCD
    LOW RAJ.output (RAJ_LCD_D7, ເຖິງ) # ທຳພັນ D7 ຂອງ LCD LOW ຄໍາ
    ພົດ & 0x10 == 0x10:

    RAJ.output(RAJ_LCD_D4, True) # ທຳ D4 pin ຂອງ LCD HIGH ຄໍາ
    bits&0x20==0x20:
    RAJ.output(RAJ_LCD_D5, True) # ທຳ D5 pin ຂອງ LCD HIGH ຄໍາ
    bits&0x40==0x40:
    RAJ.output(RAJ_LCD_D6, True) # ທຳ D6 pin ຂອງ LCD HIGH ຄໍາ
    bits&0x80==0x80:
    RAJ.output(RAJ_LCD_D7, True) # ທຳ D7 pin ຂອງ LCD HIGH

    RAJ_lcd_toggle_enable() defRAJ_lcd_toggle_enable():
        wait.sleep(RAJ_E_DELAY) # ລ້າໜ້າ RAJ.output(RAJ_LCD_E,
        True) # ທຳໃກ້ E ປັກຫຸດໄວ້ກໍ HIGH wait.sleep
        (RAJ_E_PULSE) # ລ້າໜ້າ

    RAJ.output(RAJ_LCD_E, False) # ທຳໃກ້ E ຕຽ້ງໄວ້ກໍ LOW
    wait.sleep(RAJ_E_DELAY) # ລ້າໜ້າ

    defRAJ_lcd_string(message,line): message =
    message.ljust(RAJ_LCD_WIDTH," ")
    RAJ_lcd_display(line, RAJ_LCD_CMD)
    ສໍາຮັບຜນອຍໃນໜ່ວງ (RAJ_LCD_WIDTH):
    RAJ_lcd_display(ord(message[i]),RAJ_LCD_CHR)

    ຄໍາ __name__ == '__main__':
    ລອງ:
    ແລັກ()
```

ຍັກເວັນແປ່ນພິມພັດຈັງຫວະ: ຜ່ານ

ໃນກີ່ສຸດ:

```
RAJ_lcd_display(0x01, RAJ_LCD_CMD)
RAJ.cleanup() # ລ້າງ GPIO ກັ້ນເບີດ
```

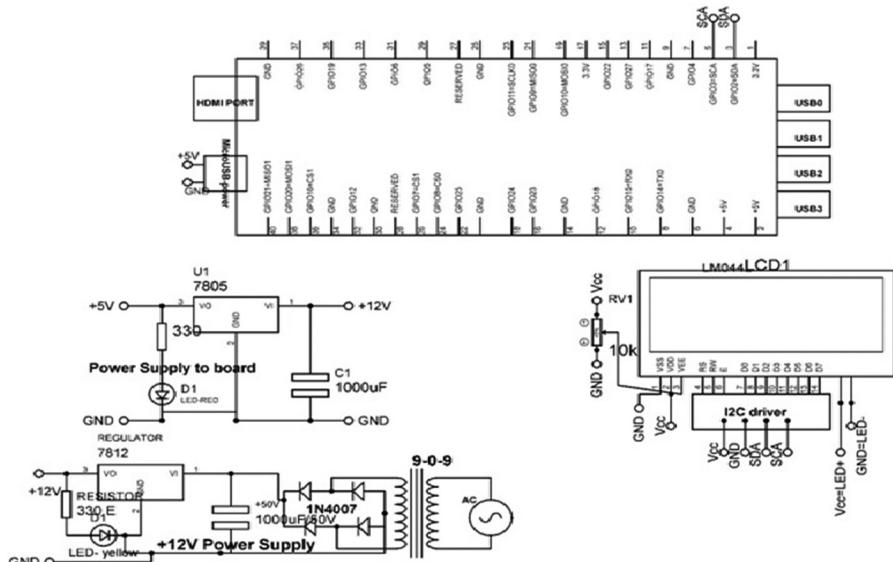
5.20 ການເຊື່ອນຕ່ອ LCD ກັບ Raspberry Pi ໃນໂໜ້ນ I2C

ວັຈຈະປະກອບດ້ວຍ Raspberry Pi, LCD ແລະແກ່ລ່າງຈ່າຍໄຟ ໄດ້ເວັບ I2C ເຊື່ອນຕ່ອກັບ LCD ກໍາການເຊື່ອນຕ່ອລ່ວນປະກອບຕາມກີ່ແສດງໃນຮູບ 5.29.

ຂັ້ນຕອນທີ 1:ເຊື່ອນຕ່ອ LCD ກັບ Raspberry Pi

ການເຊື່ອນຕ່ອ:

- ເຊື່ອນຕ່ອ pin1 (Vss) ແລະ pin16 (LED-) ຂອງ LCD ແລະ pin (GND) ຂອງໄດ້ເວັບ I2C ກັບກາວົດ
- ເຊື່ອນຕ່ອ pin2 (Vcc) ແລະ pin15 (LED+) ຂອງ LCD ແລະ ພິມ (Vcc) ຂອງໄດ້ເວັບ I2C ກັບ +5 V DC



ຮູບ 5.29
ແຜນການງານຈຳສໍາຮັບການເຊື່ອນຕ່ອ LCD ໃນໂໜ້ນ I2C

- ເຊື່ອມຕ່ອງ R_in3 (V_EE) ຂອງ LCD ກັບເຄົາດົກຕູຂອງໂພເກນຊ້ອມົມເຕອຣ 10 K ແລະ ເຊື່ອມຕ່ອງ
ຂັ້ວັດສອງຂັ້ວຂອງໂພເກນຊ້ອມົມເຕອຣກັບ +5 V DC ແລະ ກຣາວົ່ງ
- ເຊື່ອບຕ່ອງພິນ (SDA) ຂອງໄໂດຣເວຼອຣ LCD I2C ກັບ GPIO3 ຂອງ Raspberry Pi
- ເຊື່ອບຕ່ອງພິນ (SCA) ຂອງໄໂດຣເວຼອຣ LCD I2C ກັບ GPIO5 ຂອງ Raspberry Pi **ຫັນ**

ຕອນທີ 2:ດາວນໄໂລດສຄຣິປ່ຕໍ Python ໂດຍໃຊ້:

`wget https://bitbucket.org/MattHawkinsUK/rpi-spy-misc/raw/master/python/lcd_i2c.py`

ຫັນຕອນທີ 3:ເປີດໃຈ້ຈານອົນເກອຮີເຟີ້ I2C ແລະ Raspberry Pi ໂດຍກຳຕານ

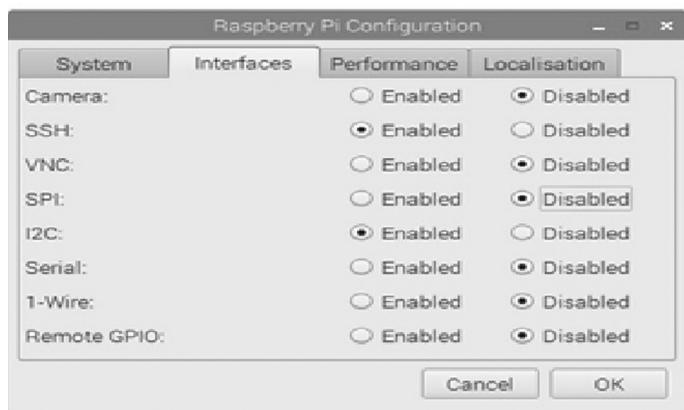
ເສັບກາງ ແມ່ນ>ການຕັ້ງຄ່າ>ການກຳຫຸດຄ່າ Raspberry Pi ([ຮູບທີ 5.30](#)). ຈາກບັນເລືອກ
ແກ້ບ "ອົນເກອຮີເຟີ້" ແລະຕັ້ງຄ່າ I2C ເປັນ "ເປີດໃຈ້ຈານ" ([ຮູບ 5.31](#)).

ຄລືກປຸມ "ຕກລອງ" ກ່າວໄດ້ຮັບແຈ້ງໄກ້ຮູບຕູ ໄກສີເລືອກ "ໃໝ່" ເພື່ອໃກ້
ການປັບປຸງແປ່ງສາມາດກຳໄດ້ອ່າຍ່າງມີປະສິກີກາວ ([ຮູບທີ 5.32](#)). Raspberry Pi ຈະ
ຮູບຕູແລະເປັດໃຈ້ຈານອົນເກອຮີເຟີ້ I2C ແລ້ວ

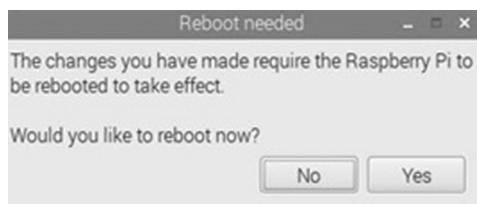
ຫັນຕອນທີ 4:ເປີດສຄຣິປ່ຕໍ LCD ດ້ວຍຄໍາສັ່ງ:`sudo nano lcd_i2c.py` **ຫັນຕອນທີ 5:**ຮັບ
ສຄຣິປ່ຕໍດ້ວຍຄໍາສັ່ງ:`sudo python lcd_i2c.py`



ຮູບທີ 5.30
ການຕັ້ງຄ່າບະເກອຮີບັນລັບ



ຮູບ 5.31
ເປີດໃຫ້ຈານ I2C



ຮູບ 5.32
ຮັບຕຸດ

5.20.1 ສຸຕຣສໍາຫຼັບອົນເກອຮີເຟ්‍යුඩ LCD ໃນໄໂນມດ I2C

```
import smbus
# ບຳເຂົາເວລາຮອງ
RAJ_I2C_ADDR = 0x24 # ຖື່ອຍ່າງອຸປະກອນ I2C RAJ_LCD_WIDTH = 16
# ຈຳນວນອັກຂະຫົວທີ່ຕ່ອງຮັບກັດ RAJ_LCD_CHR = 1 # ໄໂນມດ - ກໍາລັງສ່າງຂ້ອບູລ
RAJ_LCD_CMD = 0 # ໄໂນມດ - ກໍາລັງສ່າງຄຳສັ່ງ RAJ_LCD_LINE_1 = 0x80 #
LCD_RACLINE ຖື່ອຍ່າງ RAM ສໍາຫຼັບບຽບກັດກີ່ 1 ຖື່ອຍ່າສໍາຫຼັບບຽບກັດກີ່ 2
RAJ_LCD_LINE_3 = 0x94 # ຖື່ອຍ່າ LCD RAM ສໍາຫຼັບບຽບກັດກີ່ 3
RAJ_LCD_LINE_4 = 0xD4 # ຖື່ອຍ່າ LCD RAM ສໍາຫຼັບບຽບກັດກີ່ 4
RAJ_LCD_BACKLIGHT = 0x08
```

```
RAJ_ENABLE = 0b00000100 # LCD ເປີດໃຫ້ຈານ (E) ປົດ
RAJ_E_PULSE = 0.0005
RAJ_E_DELAY = 0.0005
```

```
bus = smbus.SMBus(1) # Rev 2 Pi ໃຊ້ 1
defRAJ_lcd_init():
    RAJ_lcd_byte(0x33,RAJ_LCD_CMD) # 110011 Initialise
    RAJ_lcd_byte(0x32,RAJ_LCD_CMD) # 110010 Initialise
    RAJ_lcd_byte(0x06,RAJ_LCD_CMD) # 000110 Cursor move direction
    RAJ_lcd_byte(0x0C,RAJ_LCD_CMD) # 001100 ເປີດການແສດງພລ, ເປີດເຄວົຮ, ປຶກຄະພິບຕາ
    RAJ_lcd_byte(0x28,RAJ_LCD_CMD) # 101000 ຄວາມຍາວຂ້ອມູລ ຈຳນວນ
        ເສັ້ນ ຂບາດຕັ້ງອັກຂອງ
    RAJ_lcd_byte(0x01,RAJ_LCD_CMD) # 000001 ລ້າງຫຼ້າຈອ
    wait.sleep(RAJ_E_DELAY)
defRAJ_lcd_byte(ຝຶຕ, ໂກມດ):
    bits_high = ໂກມດ | (ຝຶຕ & 0xF0) | LCD_BACKLIGHT
    bits_low = ໂກມດ | ((ຝຶຕ
        <<4) & 0xF0) | LCD_BACKLIGHT
    RAJ_lcd_byte(0x33,LCD_CMD) # 110011 Initialise
    RAJ_lcd_byte(0x32,LCD_CMD) # 110010 Initialise
    RAJ_lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
    RAJ_lcd_byte(0x100D Off)

RAJ_lcd_byte(0x28,LCD_CMD) # 101000 ຄວາມຍາວຂ້ອມູລ ຈຳນວນບຣກັດ
    ຂບາດຕັ້ງອັກຂອງ
    RAJ_lcd_byte(0x01,LCD_CMD) # 000001 ລ້າງຫຼ້າຈອ
    wait.sleep(RAJ_E_DELAY)
    bus.write_byte(I2C_ADDR, bits_high)
    RAJ_lcd_toggle_enable(bits_high)
    bus.write_byte(RAJ_I2C_ADDR, bits_low)
    RAJ_lcd_toggle_enable(bits_low)
defRAJ_lcd_toggle_enable(ຝຶຕ):
    wait.sleep(RAJ_E_DELAY) # ຮອ bus.write_byte
    (I2C_ADDR, (ຝຶຕ | RAJ_ENABLE)) wait.sleep
    (RAJ_E_PULSE) # ຮອ
    bus.write_byte(I2C_ADDR,(ຝຶຕ & ~RAJ_ENABLE))
    wait.sleep(RAJ_E_DELAY) # ຮອ
deflcd_string(ຂ້ອຄວາມ,ບຣກັດ):
    ຂ້ອຄວາມ = message.ljust(RAJ_LCD_WIDTH," ")
    RAJ_lcd_byte(line, RAJ_LCD_CMD)
    ສໍາຮັບອັນອຸ່ນໃນຊ່ວງ (RAJ_LCD_WIDTH):
```

```
RAJ_lcd_byte(ord(message[i]),RAJ_LCD_CHR)
def main():
    RAJ_lcd_init()
    ในຂນະກິດຈົບ:
    RAJ_lcd_string("I2C LCD<",RAJ_LCD_LINE_1) # ພຶບພິສຕົງໃນບຣກັດກີ 1
        uu LCD
    RAJ_lcd_string("Connected<",RAJ_LCD_LINE_2) # ພຶບພິສຕົງບັນ
        line2 uu LCD
    wait.sleep(3) # ລ່າໜ້າ 3 ວິນາກີ
    RAJ_lcd_string(">      I2C LCD", RAJ_LCD_LINE_2) # ພຶບພິສຕົງບັນ
        line2 uu LCD
    wait.sleep(3) # ລ່າໜ້າ 3 ວິນາກີ ຄໍາ
    __name__ == '__main__': ລອງ:
        ແລ້ກ()
        ຍກເວັນແປ້ນພິມພັດຈັດຫວະ: ຜ່ານ
    ໃນກີສຸດ:
        RAJ_lcd_byte(0x01, RAJ_LCD_CMD)
```

5.21 ການເຊື່ອມຕ່ວຂອງເຊັບເຊົ່ວ DHT11 ກັບ Raspberry Pi

ເຊັບເຊົ່ວ DHT11 ກໍາງານຮ່ວມກັບຮະບບສາຍໄຟເດືອຍ ດ້ວຍຄວາມຊັບວັດໂດຍເຊັບເຊົ່ວ
ຈາກນັ້ນຂໍ້ມູນນີ້ຈະໄດ້ຮັບການສ້ອສາງເປັນລໍາດັບຜ່ານຫາອຸກ ເປັນອຸປະກອນສີພິບ
ຕ້ອງວາງຕັ້ງຕ້ານການ ກີມຄ່າ 4.4-10 K ຮະຫວ່າງພິບ 1 (3.3 V) ແລະພິບ 2 (ຂໍ້ມູນ) ອຸປະກອນສາມພິບນີ້ຕ້ົວຕ້ານການໃນຕັ້ງ
(ຮູບ 5.33).

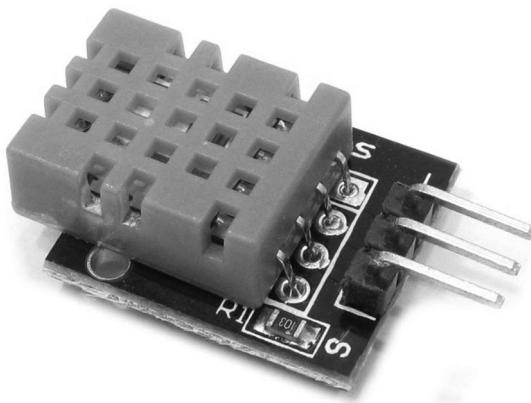
ໄລບຣາຣີ Adafruit DHT11 ສາມາຮັກໃຊ້ກັບ DHT11, DHT22 ຮັ້ອເຊັບເຊົ່ວອຸນຫຼຸມແບບ
ສາຍເດືອຍຈົນໆ ຂັ້ນຕອນການຕິດຕັ້ງໄລບຣາຣີ DHT11 ຈະຄລ້າຍກັບຂັ້ນຕອນໃນການຕິດຕັ້ງໄລບຣາຣີ
LCD

ຂັ້ນຕອນໃນການເຊື່ອມຕ່ວ DHT11

1. ອັບເປດຕາຍການແພັດເກຈແລະຕິດຕັ້ງໄລບຣາຣີ Python ດ້ວຍຄວາມຊ່ວຍເຫຼືອຂອງຄໍາສັ່ງ:

```
sudo apt-get update
```

```
sudo apt-get install build-essential python-dev
```



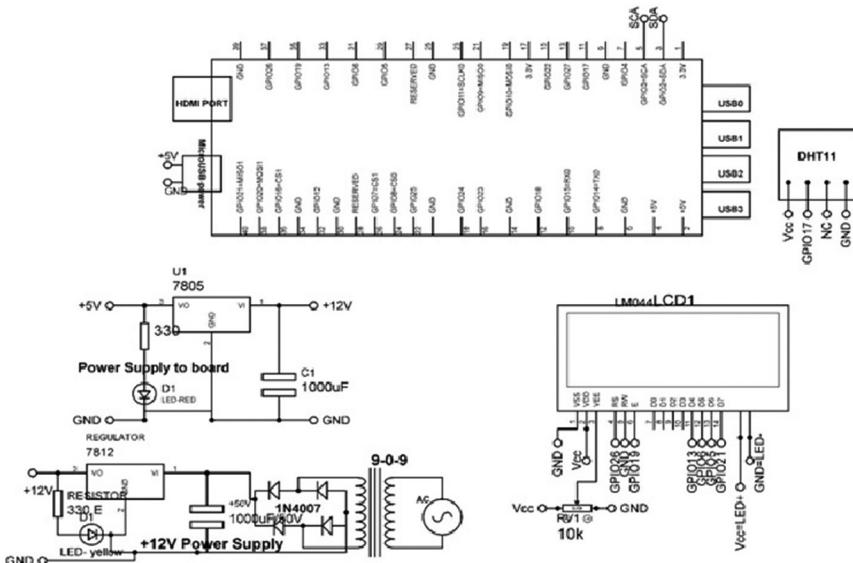
ຮູບ 5.33

ມູນຄວາມຂອງເຂັ້ມເຊີຣີ DHT11

2. ດາວໂຫຼດໄລຍບຣາີ Adafruit ໂດຍໃຊ້ຄໍາສັ່ງ: *ໄດ້ລັງ git https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT*
3. ຕິດຕັ້ງໄລຍບຣາີສໍາຮັບ Python 2 ແລ້ວ Python 3 ໂດຍໃຊ້ຄໍາສັ່ງ: *sudo python setup.py* ຕິດຕັ້ງ *sudo python3 setup.py* ຕິດຕັ້ງ
4. ເຊື່ອມຕ່ອສ່ວນປະກອບດັ່ງຕ່ອໄປນີ້:

ການເຊື່ອມຕ່ວ່າ:

- ເຊື່ອມຕ່ວ່າ pin1 (Vss), pin5 (RW) ແລ້ວ pin16 (LED-) ຂອງ LCD ກັບກາວດີ
- ເຊື່ອມຕ່ວ່າ pin2 (Vcc) ແລ້ວ pin15 (LED+) ຂອງ LCD ກັບ +5 V DC
- ເຊື່ອມຕ່ວ່າ pin3 (V_{EE}) ຂອງ LCD ເພື່ອເອາກົກພຸດໂພເກນຊອມເຕັກ 10 K ແລ້ວຕ່ອງໜັງໂພເກນຊອມເຕັກ ທີ່ຕ່ອງໜັງ +5 V DC ແລ້ວກາວດີ
- ເຊື່ອມຕ່ວ່າ pin4 (RS) ຂອງ LCD ກັບ GPIO26 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ວ່າ pin6 (E) ຂອງ LCD ກັບ GPIO19 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ວ່າ pin11 (D4) ຂອງ LCD ກັບ GPIO13 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ວ່າ pin12 (D5) ຂອງ LCD ກັບ GPIO6 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ວ່າ pin13 (D6) ຂອງ LCD ກັບ GPIO5 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ວ່າ pin14 (D7) ຂອງ LCD ກັບ GPIO21 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ວ່າພັບ (Vcc) ແລ້ວພັບ (GND) ຂອງ DHT11 ຄື່ງ +5 V ແລ້ວກາວດີຕາມລຳດັບ
- ເຊື່ອມຕ່ວ່າຂາອົກຂອງ DHT11 ກັບ GPIO17 ຂອງ Raspberry Pi



ຮູບ 5.34

ແຜນກາພວງຈອສໍາຮັບການເຊື່ອມຕ່ອ DHT11S

ຮູບ 5.34 ແລະ ແຜນກາພວງຈອສໍາຮັບການເຊື່ອມຕ່ອ LCD ແລະ DHT11 ກັບ Raspberry Pi

5.21.1 ສູດການອ່ານເຊັນເຊົ່ວ DHT11

```
importAdafruit_DHT เป็นเวลาบាເຂົ້າ
RAJ_DHT เป็น wait
RAJ_sensor = RAJ_DHT.DHT11 RAJ_gpio =
17 # ຄຳຫັດຕັ້ງແປໄກກັບພິບ 17
HUM, TEMP = RAJ_DHT.read_retry(RAJ_sensor, RAJ_gpio) ຄ້າ
ຄວາມຊັ້ນໄມ້ໃໝ່ ໄນມີ ແລະ ອຸນຫຼຸມໄມ້ໃໝ່ ໄນມີ: ພິມົງ ('TEMP={0:0.1f}*C
HUM={1:0.1f}%' .format(TEMP , HUM)) ຮອ. ສລັບ(0.2)
```

ອີບ:

```
print('ລອງອັກຄັ້ງເພື່ອອ່ານຈາກ DHT11')
wait.sleep(0.2)
```

5.21.2 ສູດກາສໍາຮັບອ່ານເຊັນເຊົ່ວ DHT11 ແລະ ແຜນກາພວງຈອສໍາຮັບ LCD

```
ນໍາເຂົ້າເວລາຮອ #ນໍາເຂົ້າກ້ອງສມຸດເວລາ importAdafruit_CharLCD ເປັນ
RAJ_LCD #ນໍາເຂົ້າກ້ອງສມຸດ LCD importAdafruit_DHTas RAJ_DHT #ນໍາ
ເຂົ້າກ້ອງສມຸດ DHT
```

```
RAJ_sensor_name = RAJ_DHT.DHT11 #ເຮົາກໍາລັງໃຊ້ເຊື້ນເຊົ່ວ DHT11
RAJ_sensor_pin = 17 # ກໍານົດຊື້ໃຫ້ພັບ 17 ກີ່ເຊື້ນເຊົ່ວເຊື່ອມຕ່ອງຍຸ່ງ
RAJ_lcd_rs = 26 # ກໍານົດຊື້ໃຫ້ກັບພັບ 26 ໂດຍກີ່ພັບ RS ຂອງ LCD ຕົວ
ເຊື່ອມຕ່ວ
RAJ_lcd_en = 19 #ກໍານົດຊື້ໃຫ້ກັບພັບ 26 ໂດຍກີ່ E ຂາຂອງ LCD ເຊື່ອມຕ່ອງຍຸ່ງ
RAJ_lcd_d4 = 13 #ກໍານົດຊື້ໃຫ້ກັບພັບ 26 ໂດຍກີ່ D4 ພັບຂອງ LCD ຕົວ
ເຊື່ອມຕ່ວ
RAJ_lcd_d5 = 6 #ກໍານົດຊື້ໃຫ້ກັບພັບ 26 ໂດຍກີ່ D5 ພັບຂອງ LCD ຕົວ
ເຊື່ອມຕ່ວ
RAJ_lcd_d6 = 5 #ກໍານົດຊື້ໃຫ້ກັບພັບ 26 ໂດຍກີ່ D6 ພັບຂອງ LCD ຕົວ
ເຊື່ອມຕ່ວ
RAJ_lcd_d4 = 21 ກໍານົດຊື້ໃຫ້ກັບພັບ 26 ໂດຍກີ່ D7 ພັບຂອງ LCD ຕົວ
ເຊື່ອມຕ່ວPI
RAJ_lcd_backlight= 0
RAJ_lcd_columns = 20 # LCD 20x4
RAJ_lcd_rows = 4 # LCD 20x4
RAJ_lcd = RAJ_LCD_CharLCD(RAJ_lcd_rs, RAJ_lcd_en, RAJ_lcd_d4,
RAJ_lcd_d5, RAJ_lcd_d6, RAJ_lcd_d4, RAJ_lcd_columns, RAJ_lcd_ແຄວ,
RAJ_lcd_backlight)
# ຮະບຸຮາຍລະເອີ້ນພັບໄປຢັ້ງໄລບຣາຣີ LCD RAJ_lcd.message("VALUE FROM
DHT11") #print message on LCD wait.sleep(2) # ບັນວັນເວລາ 2 ວິນາກີ
```

ໃນຂະນະກີ 1:

```
HUM, TEMP = Adafruit_DHT.read_retry (RAJ_sensor_name, RAJ_)
ເຊື້ນເຊົ່ວ_ພັບ)
RAJ_lcd.clear() #clear ເນື້ອຫາຂອງ LCD
RAJ_lcd.message ('Temp= %.1f C' % TEMP) # ພິມພົກຄ້າຫຼັກຮາບນ LCD
RAJ_lcd.message ('\nHum = %.1f %%' % HUM) ພິມພົກຄ້າ hum ແລ້ວ LCD ພິມພົກ
('TEMP={0 :0.1f}*C HUM={1:0.1f}%.format(TEMP, HUM))
wait.sleep(2) #ລໍາສຳ 2 ວິນາກີ
```

5.22 ການເຊື່ອມຕ່ວ Ultrasonic Sensor ກັບ Raspberry Pi

ເຊື້ນເຊົ່ວອັລຕຣາໂຈນິກໃຊ້ເພື່ອວັດຮະຍະໜ່າງຂອງວັດຖຸຈາກເຊື້ນເຊົ່ວ ປະກອບດ້ວຍເຄື່ອງສົ່ງອັລຕຣາໂຈນິກ ວັດຮະຍະຄຸນ ແລະ ເຄື່ອງຮັບອັລຕຣາໂຈນິກ ມີ 4 ພັບ ໄດ້ແກ່ VCC, TRIG (Trigger), ECHO (Echo),



ຮູບ 5.35
ເຊັນເຂອຮ້ອລ໌ຕາໂຈນີກ

ແລະ GND ເຄື່ອງສ່ງອັດຕາໂຈນີກໃນເຊັນເຂອຮ້ອສ້າງອັດຕາຫາວັດ 40 KHz ຊຶ່ງຈະແພ່ຮ່າຍໃນ
ອາການແລະສະກ້ອນກຳລັບທາກນີ້ສິ່ງກົດຂວາງໃດ ທຸກໆ ເກີດຂຶ້ນໃນເສັ້ນກາງ ຮະຍະກາງຄຳນວນຕານຫົວໜວຍ
ຮ່າງສັນຍານທີ່ສ່ົງແລະຮັບກັບເພື່ອສະກ້ອນພິບຂອງເຊັນເຂອຮ້ອ ໃນການສ້າງກົກເກອຮ້ສັນຍານ
ອັດຕາໂຈນີກ ພິບຕ້ອງກີ່ວິດ "ສູງ" ເປັນຮະຍະເວລາຫັນຕໍ່າ 10ໄນໂຄຣເອສຮູບ 5.35ແສດງເຊັນເຂອຮ້ອ
ອັດຕາໂຈນີກ

ຕັວອ່າງເຫັນ ກໍາເວລາກີ່ ECHO ສູງກີ່ 588ໄນໂຄຣS ແລ້ວຮະຍະກາງສາບາຣຄຄຳນວນໄດ້ໂດຍໃຊ້
ຄວາມເຮົວຂອງເສີຍຊື່ງເກົ່າກັບ 340 m/s

$$\text{distance} = \text{ຄວາມເຮົວຂອງເສີຍຊື່ງ}/(\text{ເວລາ}/2)$$

$$=340 \text{ m./ວິນາກີ່ } / (588 \text{ໄນໂຄຣS}/2) = 10 \text{ ຊມ.}$$

ຕ້ອງໃຊ້ສຸຕຽນນີ້ໃນໂປຣແກຣມຄຳນວນຮະຍະກາງ
ເພື່ອໃຫ້ເຂົ້າໃຈການກຳຈານຂອງເຊັນເຂອຮ້ອອັດຕາໂຈນີກສາບາຣຄອອກແບບວັງຈອຍຢ່າງໜ່າຍໄດ້
ປະກອບດ້ວຍ Raspberry Pi ແລ້ວຈ່າຍໄຟ ເຊັນເຂອຮ້ອອັດຕາໂຈນີກ ແລະ LCD ເຊັນຕ່ອງສ່ວນ
ປະກອບດັ່ງຕ່ອງໄປນີ້:

ການເຊື່ອມຕ່ວ:

- ເຊື່ອມຕ່ວ pin1 (Vss), pin5 (RW) ແລະ pin16 (LED-) ຂອງ LCD ກັບກາວົດ
- ເຊື່ອມຕ່ວ pin2 (Vcc) ແລະ pin15 (LED+) ຂອງ LCD ເຂົ້າກັບ +5 V DC
- ເຊື່ອມຕ່ວ pin3 (V_{EE}) ຂອງ LCD ກັບເອົາດົກຫຼັງໂພເກນຊອວົມເຕັກ 10 K ແລະ
ເຊື່ອມຕ່ວບັງອົກສອງບັງຂອງໂພເກນຊອວົມເຕັກ +5 V DC ແລະກາວົດ
- ເຊື່ອມຕ່ວ pin4 (RS) ຂອງ LCD ກັບ GPIO26 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ວ pin6 (E) ຂອງ LCD ກັບ GPIO19 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ວ pin11 (D4) ຂອງ LCD ກັບ GPIO13 ຂອງ Raspberry Pi

- ເຊື່ອມຕ່ອງ pin12 (D5) ຂອງ LCD ກັບ GPIO6 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອງ pin13 (D6) ຂອງ LCD ກັບ GPIO5 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອງ pin14 (D7) ຂອງ LCD ກັບ GPIO21 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອງພິບ (Vcc) ແລະພິບ (GND) ຂອງອັລຕຣາໂຈັກກັບ +5 V ແລະກຣາວດຕາມສຳດັບ
- ເຊື່ອມຕ່ອງພິບ (TRIG) ຂອງເຊັນເຊົ່ວຮັບອັລຕຣາໂຈັກກັບ GPIO20 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອງພິບ (ECHO) ຂອງເຊັນເຊົ່ວຮັບອັລຕຣາໂຈັກກັບ GPIO21 ຂອງ Raspberry Pi

ຮູບກໍ 5.36 ແສດງແຜນກາພວງຈະສໍາຫັບການເຊື່ອມຕ່ອງ LCD ແລະເຊັນເຊົ່ວຮັບອັລຕຣາໂຈັກກັບ Raspberry Pi

5.22.1 ສູດກາຮ່ານ Ultrasonic Sensor ແລະແສດງຂ້ອມຸລບນ LCD

importRPi.GPIO ເປັນເວລານໍາ

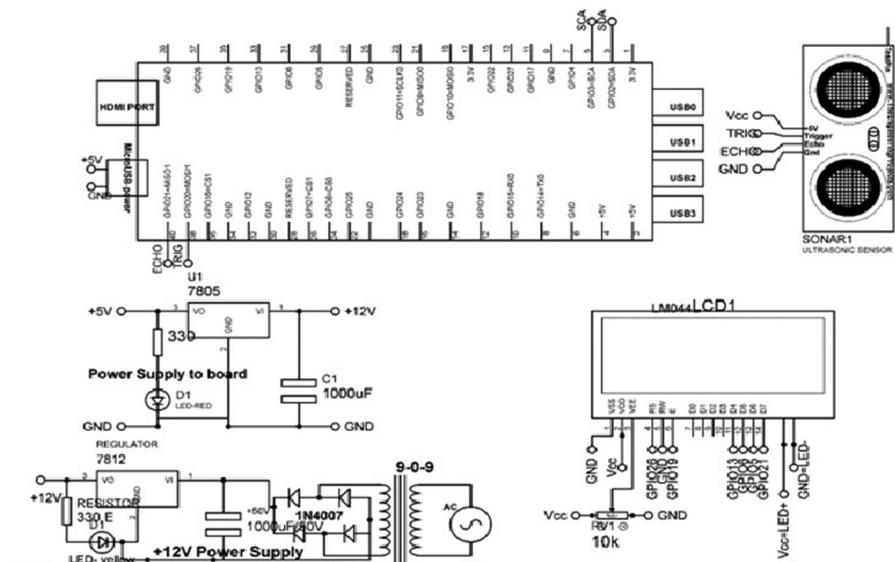
ເຂົ້າຂອງ ANITA ເປັນຮອ

importAdafruit_CharLCD ເປັນ LCD #ນຳເຂົ້າຫ້ອງສຸດ LCD

RAJ_lcd_rs = 26 #ກຳທັດຊ່ອໃກ້ກັບພິບ 26 ໂດຍຖີ່ພິບ RS ຂອງ LCD ຄູ້
ເຊື່ອມຕ່ອງ

RAJ_lcd_en = 19 #ກຳທັດຊ່ອໃກ້ກັບພິບ 26 ໂດຍຖີ່ E ຂອງ LCD ເຊື່ອມຕ່ອງຢູ່

RAJ_lcd_d4 = 13 #ກຳທັດຊ່ອໃກ້ກັບພິບ 26 ໂດຍຖີ່ D4 ພິບຂອງ LCD ຄູ້
ເຊື່ອມຕ່ອງ



ຮູບ 5.36

ແຜນກາພວງຈະສໍາຫັບເຊື່ອມຕ່ອງເຊັນເຊົ່ວຮັບອັລຕຣາໂຈັກກັບ Raspberry Pi

```
RAJ_Lcd_d5 = 6 #ກຳນົດຊື່ອໃຫ້ກັບພັນ 26 ໂດຍຖີ່ D5 ພິນຂອງ LCD ຄູ້  
ເຊື່ອມຕ່ອ  
RAJ_Lcd_d6 = 5 #ກຳນົດຊື່ອໃຫ້ກັບພັນ 26 ໂດຍຖີ່ D6 ພິນຂອງ LCD ຄູ້  
ເຊື່ອມຕ່ອ  
RAJ_Lcd_d7 = 21 #ກຳນົດຊື່ອໃຫ້ກັບພັນ 26 ໂດຍຖີ່ D7 ພິນຂອງ LCD ຄູ້  
ເຊື່ອມຕ່ອ  
RAJ_Lcd_backlight = 0 #ກຳນົດ 0 ອົບພູຕໃຫ້ກັບແບັບໄລກ  
RAJ_Lcd_columns = 20 #ເລືອກ LCD 20x4  
RAJ_Lcd_rows = 4 #fochoose LCD 20x4  
RAJ_Lcd = LCD.Adafruit_CharLCD(RAJ_Lcd_rs, RAJ_Lcd_en,  
    RAJ_Lcd_d4, RAJ_Lcd_d5, RAJ_Lcd_d6, RAJ_Lcd_d4, lcd_columns,  
    RAJ_Lcd_rows, RAJ_Lcd_backlight) #ຈັດຫາຮາຍລະເວີດໃຫ້ກັບຫ້ອງສຸດ LCD  
  
GPIO.setmode(GPIO.BCM) #set mode ຂອງ GPIO ເປັນ BCM  
RAJ_GPIO_TRIGGER = 20 # ກຳນົດ 20 pin ໃຫ້ກັບ trigger pin ຂອງເຂັ້ມເຊອງ  
RAJ_GPIO_ECHO = 21 # ກຳນົດ 21 ພັນໃຫ້ກັບ ECHO pin ຂອງເຂັ້ມເຊອງ  
ANITA.setup(RAJ_GPIO_TRIGGER, ANITA.OUT) # ກຳນົດຖືກຄາກຂອງ  
    ປັກບຸດເປັນ #OUTPUT  
ANITA.setup(RAJ_GPIO_ECHO, ANITA.IN) # ກຳນົດຖືກຄາກຂອງພົບເປັນ  
    ພລົມຄົດ  
ຮະຢະ def ():  
    ANITA.output(RAJ_GPIO_TRIGGER, True) # ທຳໃຫ້ 20 pin ເປັນ HIGH  
    wait.sleep(0.00001) # ລໍາໜ້າ 10 uSec  
    ANITA.output(RAJ_GPIO_TRIGGER, False) # ທຳ 20 ພົບເປັນ LOW  
    Start_Time = wait.time()  
    Stop_Time = wait.time()  
    while ANITA.input(RAJ_GPIO_ECHO) == 0:  
        Start_Time = ສອເວລາ ()  
    ໃນຂະໜາດ ANITA.input(RAJ_GPIO_ECHO) == 1:  
        Stop_Time = ສອເວລາ ()  
    Time_Elapsed = Stop_Time - Start_Time # ບັນທຶກເວລາທີ່ຜ່ານໄປ  
    RAJ_distance = (Time_Elapsed * 34300) / 2  
    ກລັບRAJ_distance  
    ຄ້າ __name__ == '__main__':  
        ລອງ:  
        lcd.message('DISTANCE:') #print string uu LCD  
        wait.sleep(2) #ຮອ 2 ວິນາຖື
```

ໃນຂນະກິຈົງ:

```
RAJ_dist = ຮະຍະກາງ ()  
print ("DISTANCE = %.1f cm" % RAJ_dist)  
RAJ_lcd.clear() #clear the contents of LCD  
RAJ_lcd.message ('DIST = %.1f cm' % RAJ_dist)  
wait.sleep(1)  
ຢກເວັນແປ່ນພິມພັດຈັງຫວະ: ພິມພົມ  
("ຫຍຸດກາວວັດ")  
ANITA.cleanup()
```

5.23 ການເຊື່ອມຕ່ອກລ້ອງກັບ Raspberry Pi

ກລ້ອງໃຊ້ສໍາຮັບຄ່າຍກາພື້ນຖານຂອງ Raspberry Pi ກັບໂນໂລກລ້ອງສາມາດຄຳໄດ້ດ້ວຍ Python ແລະ Picamera ປັດ Raspberry Pi ຈາກນັ້ນເຊື່ອມຕ່ອກລ້ອງເຂົ້າກັບພອຣູຕ ກລ້ອງແລະເປີດ Raspberry Pi

5.23.1 ການກຳເໜດຄ່າກລ້ອງດ້ວຍ Raspberry Pi

1. ເຊື່ອມຕ່ອກລ້ອງແລະເຮັ່ມຕົ້ນ Raspberry Pi
2. ເປີດເຄື່ອງນັ້ນກຳເໜດຄ່າຈາກເມຸ່ງຫຼັກ ([ຮູບກີ 5.37](#)).
3. ເປີດໃໝ່ຈານຊອຟັບແວຣກລ້ອງແລະຮູບຕູກ Raspberry Pi ([ຮູບ 5.38](#)).

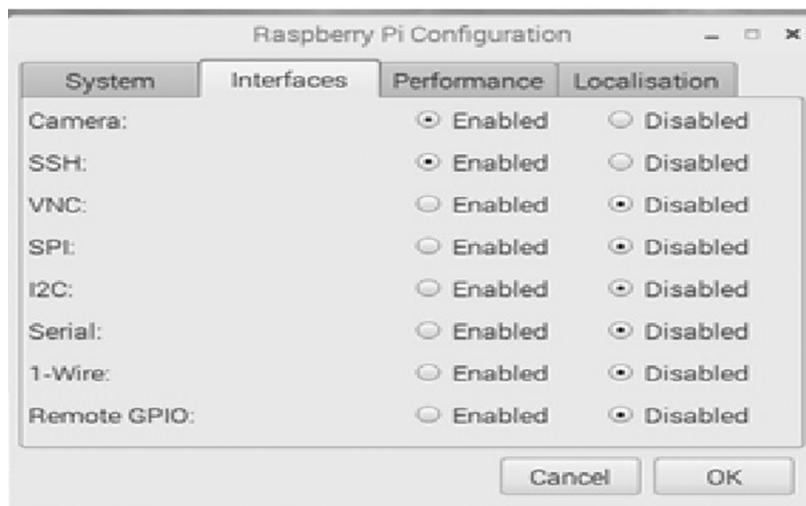
5.23.2 ການຈັບກາພດ້ວຍກລ້ອງ Pi

1. ສ້າງໄຟລ໌ກໍສາມາດແກ່ໄປໄດ້ໃໝ່ໂດຍໃຊ້ຄໍາສັ່ງ `sudo nano image_RAJ.py` ໃນ Pi terminal

```
from picamera import PiCamera  
RAJ_camera = PiCamera()  
RAJ_camera.start_preview()  
sleep(10)  
RAJ_camera.stop_preview()
```



ຮູບ 5.37
ການກໍານົດຕໍ່າ Raspberry Pi



ຮູບ 5.38
ເປີດໃຫ້ຈານກລ້ອງ

2. ບັນທຶກດ້ວຍ `ctrl+o` ແລ້ວກວດ “Enter” ເພື່ອດຳເນັນການດ້ວຍຄໍາສັ່ງ `sudo python image_RAJ.py`. ກາພຕັວອ່າງກລ້ອງຈະແສດງເປັນເວລາ 10 ວິນາກີ ແລ້ວປັດ ເລື່ອນ ກລ້ອງໂປຣບໍາ ເພື່ອດັວວ່າຢ່າງ
3. ກາກກາພຕັວອ່າງກລັບດ້ານ ໃຫ້ຮຸມບຸກພາວ 90, 180 ຮູ່ອ 240 ອົງຄາໂດຍໃຊ້ຄໍາສັ່ງຕ່ອໄປນີ້ :

```
RAJ_camera.rotation = 180
RAJ_camera.start_preview()
so.ສສປ(10)
RAJ_camera.stop_preview()
```

4. ເປົ້າຍແຄວານໂປ່ງໃສຂອງການແສດງຕັວອ່າງກລ້ອງໂດຍການຕັ້ງຄ່າຮຸມບຸກພາ (ຈາກຄ່າ 0-255):

```
from picamera ນຳເຂົ້າ ເວລານຳເຂົ້າ
PiCamera ເປົ້າຮອ
RAJ_camera = PiCamera()
RAJ_camera.start_preview(alpha=200)
ສສປ(10)
ກລ້ອງ.stop_preview()
```

5. ກາກຕ້ອງການຄ່າຍກາພນີ້ ໃຫ້ແກ້ໄຂໂຄົດເພື່ອລົດການສລືປ່ອຍ່າງນ້ອຍ 2 ວິນາກີ ແລະເພີ່ມ `camera.capture()` ແລ້ວກັດ:

```
ກລ້ອງ.start_preview()
ນອນ(5)
RAJ_camera.capture('/home/pi/Desktop/image.jpg')
RAJ_camera.stop_preview()
```

6. ຮັບໂຄຓດແລະດູກາພຕັວອ່າງກລ້ອງທີ່ເປີດໄວ້ 5 ວິນາກີກ່ອນກ່ຈະຄ່າຍກາພນີ້

7. ຮູປກາພອ່ຽນບັນດეສກທຶນ ດັບເບີລຄລິກກໍໄອຄອນໄຟລ໌ເພື່ອເປີດ

8. ກ່ອນນີ້ລອງເພີບກາຣວນຫ້າເພື່ອຄ່າຍກາພໜ້າກາພ:

```
RAJ_camera.start_preview() ສໍາຮັບວັນໃນໜົວງ
(5):
so.ນອນ(5)
```

```
RAJ_camera.capture('/home/pi/Desktop/image%s.jpg' % i)
RAJ_camera.stop_preview()
```

ຕັ້ງແປຣ “I” ມີໜ່າຍເລກກາຣວນຫ້າປັ້ງຈຸບັນຕັ້ງແຕ່ 0 ຕີ້ງ 4 ດັ່ງນັ້ນຮູປກາພຈະຄຸກບັນກຶກ ເປັນ `image0.jpg`, `image1.jpg` ແລະອັນນີ້

9. ຮູປກາພສານາຮັບປັບຂາດໄດ້ໂດຍໃຊ້ໄລບຣາຣີເຊັນ PIL ແລະ OpenCV ສາມາດກຳໄດ້ ດ້ວຍພາມີເຕອຮີປັບປັບຂາດຂອງວິຊີການ `capture()` :

ນຳເຂົ້າເວລາຮອ

ນໍາມາເຊົາກລ້ອງ

ດ້ວຍກລ້ອງຄ່າຍການ *PiCamera()* ເປີນ *RAJ_camera:*

RAJ_camera.resolution = (1024, 468)

RAJ_camera.start_preview()

ສອ.ນອນ(2)

RAJ_camera.capture('image.jpg', ປັບປາດ=(320, 240))

5.23.3 ກາຣຈັບກາພົດໄວດ້ວຍກລ້ອງ Pi

1. ເປີຍບຣັසສໍາຮຽບຮູປກາພເພື່ອແກນທີກາຣຈັບກາພ () ດ້ວຍ *start_recording()* ແລະ *stop_recording()*:

```
RAJ_camera.start_preview()  
RAJ_camera.start_recording('/home/pi/video.h264')  
wait.sleep(10)  
RAJ_camera.stop_recording()  
RAJ_camera.stop_preview()
```

2. ເຮັດໃຊ້ຮັສ ມັບຈະບັນທຶກວິດໄອເປັນເວລາ 10 ວິນາຖືແລ້ວປິດກາຣແສດງຕັວວ່າຢ່າງ ລາກ
ຕ້ອງກາຣເລີນວິດໄອ ໄກສະໜັບເຖິງເກຣມີນັລດໂດຍຄລິກໄວຄອນຈອກພີ່ຕຳໃນແກບຈານ (

ຮູປກ 5.39).

3. ເປີດເກຣມີນັລດແລ້ວພິມພົມຄໍາສັ້ງ *omxplayer video.h264* ແລະ ກົດ "Enter" ເພື່ອເລີນ
ວິດໄອ

4. ຄວາມລະເອີຍດຂອງກລ້ອງສາມາຄຄໍານັດດ່າໄດ້ດ້ວຍຄໍາສັ້ງ: *RAJ_camera.*

resolution = (2592, 1944) RAJ_camera.framerate = 15

```
RAJ_camera.start_preview()  
ສອ.ສສບ(5)  
RAJ_camera.capture('/home/pi/Desktop/max.jpg')  
RAJ_camera.stop_preview()  
“ຄວາມລະເອີຍດຂັບຕໍ່າກ່ອນມາດຕູກົດ 64x64”
```

5. ເພີ່ມຂ້ອຄວາມລອງໃນຮູປກພັດວະນາ ອັນນອຕັ້ງ *annotate_text* ໂດຍຄໍາສັ້ງ:

*RAJ_camera.start_preview() RAJ_camera.annotate_text = “
ສວັນດີໜາວໂລກ!”*



ສອ.ນອນ(5)

```
RAJ_camera.capture('/home/pi/Desktop/text.jpg')  
RAJ_camera.stop_preview()
```

6. ຄວາມສ່ວ່າງເຮັນຕົ້ນຄ່ອ "50"; ມັນສາມາຮັດເປົ້າຢັບແປລົງໄດ້ໂດຍຄໍາສັ້ງ:

```
RAJ_camera.start_preview() RAJ_camera.brightness = 40 ສອ.ສສປ(5)
```

```
RAJ_camera.capture('/home/pi/Desktop/bright.jpg')  
RAJ_camera.stop_preview()
```

7. ຄວາມສ່ວ່າງສາມາຮັດເປົ້າຢັບແປລົງໄດ້ໂດຍໃຊ້ລູປ:

```
RAJ_camera.start_preview() ສໍາຮຽນວັນໃນໜັງ (100):
```

```
RAJ_camera.annotate_text = "ຄວາມສ່ວ່າງ.%ສ%ພມ  
RAJ_camera.brightness = i  
ສອ. ສສປ(0.1)  
RAJ_camera.stop_preview()
```

8. ສ້າງຄູ່ສາມາຮັດເປົ້າຢັບແປລົງໄດ້ໂດຍໃຊ້ຄໍາສັ້ງ:

```
RAJ_camera.start_preview() ສໍາຮຽນວັນໃນໜັງ (100):
```

```
RAJ_camera.annotate_text = "ຄວາມຄົນຫັດ.%ສ%ພມ  
RAJ_camera.contrast = i  
ສອ. ສສປ(0.1)  
RAJ_camera.stop_preview()
```

9. ຂັບແນວຂ້ອງຄວາມຄົນຫັດ (ຄ່າຕັ້ງແຕ່ 6 ຄື 160) ໂດຍຄ່າເຮັນຕົ້ນ 32 ສາມາຮັດ
ເພີ່ມໄດ້ດ້ວຍຄໍາສັ້ງ:

```
RAJ_camera.annotate_text_size = 50
```

10. ສ້າງຄູ່ສາມາຮັດເປົ້າຢັບແປລົງໄດ້ໂດຍໃຊ້ຄໍາສັ້ງ:

```
from picamera import PiCamera, Color
```

11. ຈາກນັບສືອັນໆ ສາມາຮັດແກ້ໄປໄດ້ໂດຍໃຊ້ຄໍາສັ້ງ: *RAJ_camera.start_preview()*
RAJ_camera.annotate_background = Color('blue')
RAJ_camera.annotate_foreground = Color('yellow')
RAJ_camera.annotate_text = "ສວັນດີຈາວໂລກ " wait.sleep(5)

```
RAJ_camera.stop_preview()
```

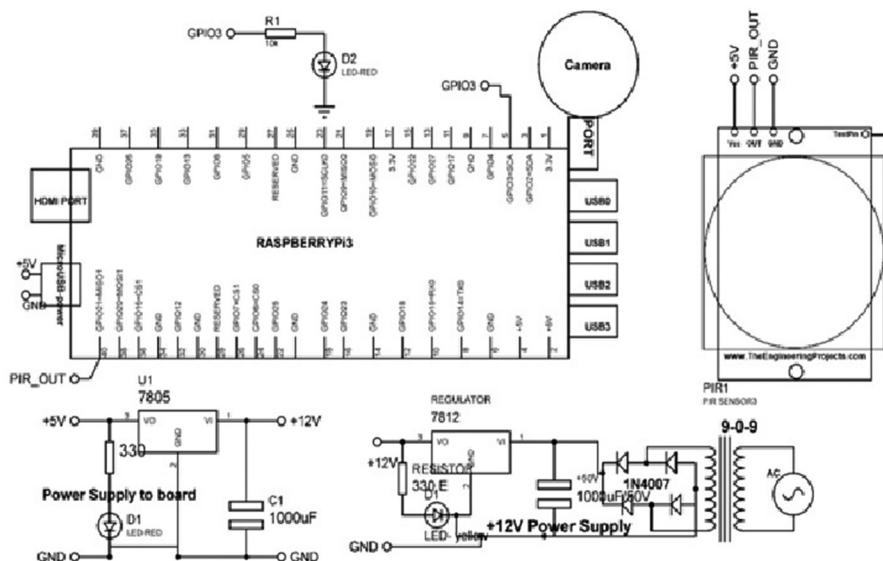
5.23.4 ຕັ້ງຕຽບຈັບຄວາມເຄລືອນໄຫວແລະການຄ່າຍກາວ

ເພື່ອໃຊ້ເຂົ້າໃຈຄົງການທຳມານຂອງກລ້ອງ ເຮົາໄດ້ກ່າວຄືງຈອງຈ່າຍໆ ໃນການຈັບກາພເນືດຕຽບຈັບການ
ເຄລືອນໄຫວໃຫ້ກີ່ໄວ LED ຈະຕິດສ່ວ່າງເນືດຕຽບການເຄລືອນໄຫວ ຮັບໃສ່ວຸດແກນ LED ໄດ້
ຮະບົບປະກອບດ້ວຍ Raspberry Pi, ເຊັນເຊື່ອຮັບຈັບຄວາມເຄລືອນໄຫວ PIR, ກລ້ອງ Pi, LED
ແລະແහລັງຈ່າຍໄວ

ການເຊື່ອມຕ່ອ:

- ເຊື່ອມຕ່ອກລ້ອງກັບ Raspberry Pi ຕາມກີ່ອີເບາຍໄວໃນມາດຮາ 5.23.
- ເຊື່ອມຕ່ອພັນ (Vcc) ແລະພັນ (GND) ຂອງເຊັນເຊື່ອຮັບຈັບຄວາມເຄລືອນໄຫວ ເຂົ້າກັບ +5 V ແລະກາວົດ
ຕາມລຳດັບ
- ຕ່ອຂາ (OUT) ຂອງເຊັນເຊື່ອຮັບຈັບຄວາມເຄລືອນໄຫວ ເຂົ້າກັບ GPIO21 ຂອງ Raspberry Pi
- ເຊື່ອມຕ່ອຂັ້ນບົກຂອງ LED ກັບ GPIO3 ຂອງ Raspberry Pi ຜ່ານຕັ້ງຕໍານາມ 10
K ແລະຂັ້ນບົກຂອງ LED ກັບກາວົດ

ຮູບ 5.40 ແຜນກາພວງຈອງສໍາຮັບເຊື່ອມຕ່ອເຊັນເຊື່ອຮັບຈັບຄວາມເຄລືອນໄຫວ ແລະກລ້ອງກັບ Raspberry Pi



ຮູບ 5.40

ແຜນກາພວງຈອງສໍາຮັບເຊື່ອມຕ່ອເຊັນເຊື່ອຮັບຈັບຄວາມເຄລືອນໄຫວ ແລະກລ້ອງກັບ Raspberry Pi

5.23.4.1 ສູດໃນການຈັບກາພໃນການຕຽວຈັບກາເຄລືອນໄຫວ

```
importRPi.GPIO เป็นเวลาหน้า  
เข้าของ ANITA เป็นรือ1  
ANITA.setwarnings (ເກົ່າ) ANITA.setmode (ANITA.BCM) ANITA.setup (3,  
ANITA.OUT) # กำหนด pin3 เป็นขาตัวพุต ANITA.setup (21, ANITA.IN) # เสื่อม  
ต่อเซ็นเซอร์ຕຽວຈັບຄວາມເຄລືອນໄຫວบน GPIO21 importpicamera
```

```
withpicamera.PiCamera() เป็น RAJ_camera:
```

```
RAJ_camera.resolution = (1024, 468)
```

```
ลอง:
```

```
ໃນຂະໜາດ 1:
```

```
VAL=GPIO.ອັບພຸດ(21)
```

```
ສໍາ(VAL==1):
```

```
ພິມພົມ ("ຕຽວຈັບວັດຖຸ") # ພິມພົບນາໂທອົບນັບລົດ ANITA.output  
(3, ຈົງ) # ກຳໃຫ້ພັນ 3 ເປັນ HIGH
```

```
RAJ_camera.start_preview () # ເຮັດແສດງຕັວອຍ່າງກລ້ອງ
```

```
RAJ_camera.start_recording ('/home/pi/video.h264')
```

```
so.sleep(10)# ລ່າໜ້າ 1 ວິນາກີ
```

```
RAJ_camera.stop_recording () # ຂູ່ດັກລ້ອງ
```

```
RAJ_camera.stop_preview () # ຂູ່ດັກແສດງ
```

```
ຕັວອຍ່າງ ຮອ ສລືບ (1) # ລ່າໜ້າ 1 ວິນາກີ
```

```
ອັນ:
```

```
ພິມພົມ ("ໄມ່ມີວັດຖຸ") # ພິມພົບນາໂທອົບນັບລົດ
```

```
ANITA.output (3, ເກົ່າ) # ກຳໃຫ້ພັນ 3 ເປັນ LOW
```

```
wait.sleep (1) # ລ່າໜ້າ 1 ວິນາກີ
```

```
ຢັກເວັ້ນແປ່ນພິມພົມພັດຈັງຫວະ:
```

```
ພິມພົມ ("ຂູ່ດັກຮະບວນການ")
```

```
ແອນນິຕ້າ.ຄລືນອັພ()
```