

8

Python GUI ພຣ້ອມ Tkinter ແລະ Arduino

8.1 Tkinter ສໍາຮັບກາຮອກແບບ GUI

ຄຸນສົນບັດຕີພື້ນຖານສໍາຮັບໄລຍະຮາສ່ວນຕ່ອປະສານກາຟຝກກັບຜູ້ໃຊ້ (GUI) ລວມຄົງຄວາມຈ່າຍໃນກາຮັບຕັດຕັ້ງແລະກາຮັບຕັດຕັ້ງທີ່ສຸດ ກຣອບງານ Tkinter ຕອບສົນອງຄວາມຕ້ອງກາຮັບຕັດຕັ້ງພື້ນຖານບອກຈາກນີ້ຢັງເປັນໄລຍະຮາສ່ວນຕ່ອປະສານກາຟຝກກັບຜູ້ໃຊ້ (GUI) ເຮັດວຽກຕັດຕັ້ງ Python

ອົບເກອຣ໌ເພີ້ນໄລຍະຮາສ່ວນຕ່ອປະສານກາຟຝກ Tkinter ເປັນອົບເກອຣ໌ເພີ້ນໄລຍະຮາສ່ວນຕ່ອປະສານກາຟຝກ Python ຊ້າມແພລຕີໂຟຣົມສໍາຮັບຊຸດເຄື່ອງນີ້ຈະ Tk GUI Tkinter ຈັດເຕີຣຍມເລີຍອົບນີ້ Python ໃນຂະນະທີ່ Tk ຈັດເຕີຣຍມວິດເຈີຕາແບບກາຟຝກ Tkinter ເປັນໄລຍະຮາສ່ວນຕ່ອປະສານກາຟຝກ Python ທີ່ໄດ້ຮັບກາຮອກປັບໃຊ້ເປັນສ່ວນນີ້ຂອງແພື້ນເກຈກາຮັບຕັດຕັ້ງ Python ສໍາຮັບຮະບບປົງປັດຕິກາຮັບຕັດຕັ້ງ

Tkinter ໄດ້ຮັບກາຮອກແບບໂດຍໃຊ້ຄວາມພຍາຍານໃນກາຮັບຕັດຕັ້ງ Python ໃນຂະນະທີ່ Tk ໄດ້ໃຊ້ການກົດສອບເວຼົອຮັບຕັດຕັ້ງ Tk ໃຫ້ໃຊ້ຄໍາສັ່ງຕ່ອງໄປນີ້ ບັນພອມຕີ Python:

```
> > >ນຳເຂົ້າ Tkinter
> > >Tkinter._test()
```

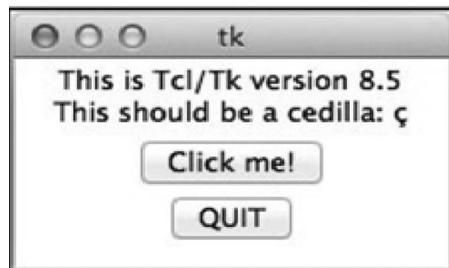
ຮູບພາບທີ່ນີ້ຂອ້ມູນເວຼົອຮັບຕັດຕັ້ງ Tkinter.

ຄ້າໜ້າຕ່າງ (ຮູບທີ່8.1) ໄນປາກຄູໃຫ້ເຫັນ ຈາກນັ້ນຕັດຕັ້ງ Python ໃໝ່

ອົບເກອຣ໌ເພີ້ນໄລຍະຮາສ່ວນຕ່ອປະສານກາຟຝກ Tkinter ອອງຮັບວິດເຈີຕາຕ່າງໆ ເພື່ອພັນນາ GUI^{ກາຮາງ8.1} ອົບເກອຣ໌ເພີ້ນໄລຍະຮາສ່ວນຕ່ອປະສານກາຟຝກ Tkinter

8.2 LED ກະພົບ

ເພື່ອໃຫ້ເຂົ້າໃຈກາຮັບຕັດຕັ້ງ Tkinter GUI ຈຶ່ງໄດ້ອົກແບບວັງຈອຍຢ່າງຈ່າຍ ຮະບບປະກອບດ້ວຍ Raspberry Pi, Arduino, LED ແລະແໜ່ງຈ່າຍໄວ Arduino ເຊື່ອມຕ່ອກກັບ Raspberry Pi ຜ່ານຫັ້ນຕ່ອງ USB ຫັ້ນແອນໂບດຂອງ LED ເຊື່ອມຕ່ອກກັບArduino ຜ່ານຫັ້ນຕ່ອງກັບ Arduino 10-K ແລະຫັ້ນແອນໂບດເຊື່ອມຕ່ອກກັບກາງວັງຈອຍ ຮູບທີ່8.2 ແລະກາງວັງຈອຍເພື່ອເຊື່ອມຕ່ອງ LED

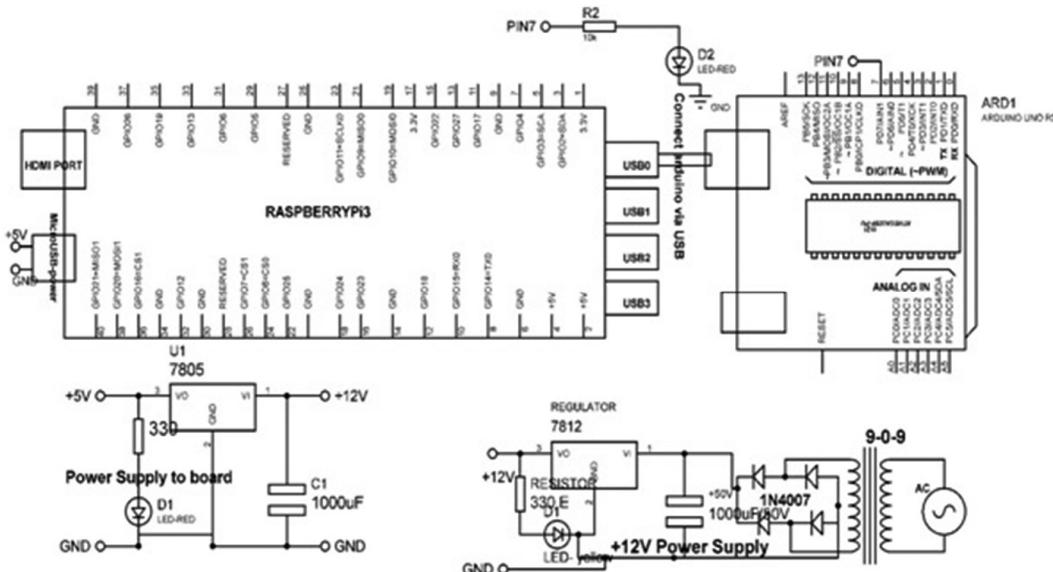


ຮູບຖ່າ 8.1
 ຮຸນກີເຄ

ຕາரັງ 8.1

ວິດເຈີຕ Tkinter ເພື່ອພັນນາ GUI

ວິດເຈີຕ	ຄໍາອອິບາຍ
ກີເຄ()	ວິດເຈີຕຮູກທີ່ຈໍາເປັນໃນບຸກໂປຣແກຣມ ມັນແສດງ
ຈລາກ()	ຮູປາພຫວູ້ຂ້ອຂ້ວຄວາມ
ປຸບ()	ດໍາເນັບການຮະກຳ
ຮາຍກາຣ()	ຂ້ອຄວານ eld ເພື່ອໃກ້ອັນພຸດ
ມາດຮ່າວນ()	ມັນໃກ້ຄ່າຕົວເລຂນໂດຍກາລາກ ຕັ້ງເລືອນ
ຂ່ອງກຳເຄົາຮ່າຍ ()	ຂ່ວຍໃກ້ສລັບໄປມາຮ່າງວ່າງສອງຄ່າ



ຮູບຖ່າ 8.2
 ແຜນການວົງຈອເພື່ອເຊັນຕ່ອງ LED

8.2.1 ສູດຮ່າຍ LED ກະພຣີບພຣ້ອມກາຮນ່ວງເວລາຄົກກື

```
ນໍາເຂົ້າ Tkinter
ນໍາເຂົ້າ pyrmata
ນໍາເຂົ້າເວລາຮອ
# ເຊື່ອມໂຍງພອຣຕແລະບອຣດກັບບອຣດ pyrmata =
pyrmata.Arduino('/dev/ttyUSB0')
wait.sleep(5) # ນ່ວງເວລາ 5 ວິນາກີ
led_Pin = board.get_pin('d:7:o') # ເຊື່ອມຕ່ອ led ກັບພິບ 7 ແລະໃຊ້ເປັນເອາະພູດ def
call_LED_BLINK():
    button.cong(state = Tkinter.DISABLED)
    led_Pin.write(1) # ມາດ led_Pin ເປັນ HIGH
    print('LED at pin7 is ON') # ພິບທີ່ໃຫຍ້
    wait.sleep(5) # delay 5 ວິນາກີ
    ພິມພົມ('LED ກັບພິບ 7 ປັດຍຸ່ງ') # ພິມພົບບເກອ່ນນັບສັດ
    led_Pin.write (0) # ດຳໃກ້ led_Pin ເປັນປຸນ
    LOW.cong (ສຄານະ = Tkinter.ACTIVE)
# ເຮັດວຽກກັບຫຼັກສິນທີ່ມີການປັບປຸງທີ່ມີການປັບປຸງ
    TOP = Tkinter.Tk()
    TOP.title("ໄວ LED ກະພຣີບໂດຍໃຊ້ປຸນ")
    TOP.minsize(300,30)
# ສ່າງປຸນບັນຫຼາຕ່າງໆແລະເຊື່ອມໂຍງກັບວິທີກາຮນ່ວງເວລາ
    ປຸນ = Tkinter.Button (ບນສຸດ text="ກົດເຮັດວຽກ" ຄຳສັ່ງ =
        Tkinter.ACTIVE)
    ປຸນ.pack()
    TOP.mainloop()
```

8.2.1.1 Tkinter GUI ສໍາຮັບ LED ກະພຣີບພຣ້ອມກາຮນ່ວງເວລາຄົກກື

ເຮັດວຽກໃຊ້ໂປຣແກຣມກີ່ອຣີບາຍໄວ້ໃນມາຕາ 8.2.1 ແລະ GUI ຈະປາກກູ້ຂຶ້ນ (ຮູບກີ່ອຣີບາຍ 8.3).

ໜ້າລັດຈາກກົດປຸນ “Press start to Blink” ໃນ GUI ມັນຈະພິມພົມ “LED at pin7 is ON”
ເປັນເວລາ 5 ວິນາກີ ແລ້ວພິມພົມ “LED at pin7 is OFF” (ຮູບກີ່ອຣີບາຍ 8.4).

8.2.2 ສູດຮ່າຍໄວ LED ກະພຣີບພຣ້ອມຕົວແປຮນ່ວງເວລາ

```
ນໍາເຂົ້າ Tkinter
ນໍາເຂົ້າ pyrmata
```



ຮູບ 8.3

Tkinter GUI ເພື່ອຄວບຄຸນ LED ພ້ອມການໜ່ວຍເວລາ xed

```
pi@raspberrypi:~ $ nano pyfir_led_tkinter_BOOK.py
pi@raspberrypi:~ $ python pyfir_led_tkinter_BOOK.py
LED at pin7 is ON
LED at pin 7 is OFF
```

ຮູບ 8.4

ໄວ LED ເປີດ/ປັດ

ບໍາເຫຼົາເວລາຮອ

```
board = pyrmata.Arduino('/dev/ttyUSB0')
wait.sleep(5) # ລ່າໜ້າ 5 ວິນາກ
led_Pin = board.get_pin('d:7:o') # ເຊື່ອມຕ່ອ led ກັບພິບ 7 ແລະໃຊ້ເປັນເອາດົບ
def led_blink_variable_delay():
    # ຄ່າຂອງຄວາມລ່າໜ້າໄດ້ມາຈາກອັນພຸຕົວດເຈີຕາຍການ time_Period =
    time_Period_Entry.get()
    time_Period = oat (time_Period) button.cong (state =
Tkinter.DISABLED) led_Pin.write (1) # ທຳໃຫ້ led_Pin
    ເປັນການພົມພຸງ ('pin7 ເຊື່ອມຕ່ອ led ເປີດຢູ່') # ພົມພົບນເກອຮົມ
    ບັນ wait.sleep (time_Period) # ລ່າໜ້າຂອງ 5 ວິນາກ
```

```
ພົມພ ('pin7 ຖືເຊື່ອມຕ່ອ led ເປີດຢູ່') # ພົມພົບນເກອຮົມບັນດັບ
led_Pin.write (0) # ທຳໃຫ້ led_Pin ຕໍ່າ
button.cong(state=Tkinter.ACTIVE)
```

```
TOP = Tkinter.Tk()
TOP.title("ປ້ອນເວລາຕັ້ງແປຣ")
time_Period_Entry = Tkinter.Entry(TOP, bd=6, width=28)
time_Period_Entry.pack()
time_Period_Entry.focus_set()
ປຸມ = Tkinter.Button (ບັນສຸດ, text = "ເຮັ່ນກະພຣິບຕາ", ຄໍາສັ້ນ = led_
    blink_variable_delay)
ປຸມ.pack()
TOP.mainloop()
```



ຮູບ 8.5

Tkinter GUI ເພື່ອគົບຄຸມ LED ພັນຍາແປຣດີເລີຍ

```
pi@raspberrypi:~ $ nano pyfir_led_tkinter_variable_time_B00K.py
pi@raspberrypi:~ $ python pyfir_led_tkinter_variable_time_B00K.py
pin7 connected led is ON
pin7 connected led is off
```

ຮູບ 8.6

ສະບັບສິດສໍາຫັບເປີດ/ປັດ LED ໂດຍມີການເນັ້ນເວລາ

8.2.2.1 Tkinter GUI ສໍາຫັບ LED ກະພຽບພັນຍາແປຣດີເນັ້ນເວລາ

ເຮັດວຽກໃຊ້ໂປຣແກຣມທີ່ອີງຕາມໄວ້ໃນມາຕາ 8.2.2 ແລະ GUI ຈະປາກຄູຂຶ້ນ (ຮູບ 8.5).

ຈົດຄ່າດີເລີຍທີ່ຕ້ອງການລອງໃນໜ້ອງວ່າງ ເຊັ່ນ ໃຊ້ຄ່າ 5 ລັບຈາກກົດປຸ່ມ “start to Blink” ແລະ ໄນວ່າງເວລາ 5 ວິນາກີບນ GUI ບັນຈະພົບພ “Pin7connected is ON” ເປັນເວລາ 5 ວິນາກີ ແລ້ວ ພົບພ “ເຊື່ອນຕ່ອ Pin7 ປັດຍູ້” (ຮູບ 8.6).

8.3 ການគົບຄຸມຄວາມສ່ວ່າງ LED

ຮູບ 8.3 ແສດງແພນກາພວງຈຮສໍາຫັບການເຊື່ອນຕ່ອ LED ໂດຍທີ່ LED ເຊື່ອນຕ່ອກັບພິບ (7) ຂອງ Arduino ພິບ (7) ຍັງເປັນພິບ PWM ໃນ Arduino Uno ດ້ວຍຕັ້ງບັນດັ່ງຍົງຈະເຕີຍວັດທີ່ໃນຮູບ 8.3, ຄວາມສ່ວ່າງ LED ສາມາດគົບຄຸມໄດ້ດ້ວຍໂປຣແກຣມອື່ນ

8.3.1 ສູດຕ

```
ນໍາເຂົາ Tkinter # ເພີ່ມໄລບຣາຣ Tkinter ນໍາເຂົາ
pyrmata # ເພີ່ມເວລານໍາເຂົາໄລບຣາຣ pyrmata ຂອນ
ຮອ # ເພີ່ມບອຮດໄລບຣາຣເວລາ =
pyrmata.Arduino('/dev/ttyUSB0')
wait.sleep(5) # ລ່າຍ້າ 5 ວິນາກີ
led_Pin = board.get_pin('d:7:o') # ເຊື່ອນຕ່ອ led ກັບພິບ 7 ແລະໃຊ້ເປັນເອາດພູຕ def
call_led_blink_pwm():
    time_Period = time_Period_Entry.get()
```

```
time_Period = ຂ້າວໂຈີຕ (time_Period)
led_Brightness = ຄວາມສວ່າງ_Scale.get ()
led_Brightness = ຂ້າວໂຈີ (led_Brightness)
button.config (state= Tkinter.DISABLED)
led_Pin.write (led_Brightness/100.0)
ພິມພົມ 'ກາຮຄວບຄຸນຄວາມສວ່າງ LED' # ພິມພົບນເທັອຣມິນັບ
wait.sleep (time_Period)
led_Pin.write(0) # ທຳໃກ້ led_Pin ໄປກໍ LOW
button.config (state=Tkinter.ACTIVE)

TOP = Tkinter.Tk()
time_Period_Entry = Tkinter.Entry(TOP, bd=7, width=30)
time_Period_Entry.pack()
time_Period_Entry.focus_set()
bright_Scale = Tkinter.Scale(ບນຊຸດ, ຈາກ_=0, ຄົງ=100, orient=Tkinter.
    ແນວຕັ້ງ)
ຄວາມສວ່າງ_Scale.pack()
ປຸມ = Tkinter.Button (ບນຊຸດ, text="Start", ຄຳສັ້ງ =
    call_led_blink_pwm)
ປຸມ.pack()
TOP.mainloop()
```

8.3.2 Tkinter GUI ສໍາຮັບກາຮຄວບຄຸນຄວາມສວ່າງ LED

ເຮັດໃຫ້ໂປຣແກຣມທີ່ອີຍບາຍໄວ້ໃນ [ມາຕາ 8.3.1](#). ຄວາມສວ່າງສາມາຮຄວບຄຸນໄດ້ດ້ວຍວິດເຈີຕສເກລ () ເນື້ອໄພ LED ດັບລົງຫລັງຈາກກາຮ່ານ່ວງເວລາ ຕັ້ງເລືອນສາມາຮຄົງເຊື້ອໄປຢັ້ງຕໍາແໜ່ງອື່ນເພື່ອ ເປັນຍິນຄ່າຄວາມສວ່າງແບບໄດ້ນາມືກ ຕັ້ງເລືອນສາມາຮຄວາມໃນແນວບອນແກນທີ່ຈະເປັນແນວຕັ້ງ ເນື້ອ ຄລິກກີ່ "ເຮັບ" ຈະແສດງຂໍອຄວາມ "ກາຮຄວບຄຸນຄວາມສວ່າງ LED" ([ຕັ້ງເລີກ 8.7 ແລະ 8.8](#)).



ສູງ 8.7

Tkinter GUI ສໍາຮັບກາຮຄວບຄຸນຄວາມສວ່າງ LED

```
pi@raspberrypi:~ $ nano pyfir_led_pwm_tkinter_B00K.py
pi@raspberrypi:~ $ python pyfir_led_pwm_tkinter_B00K.py
LED brightness control
```

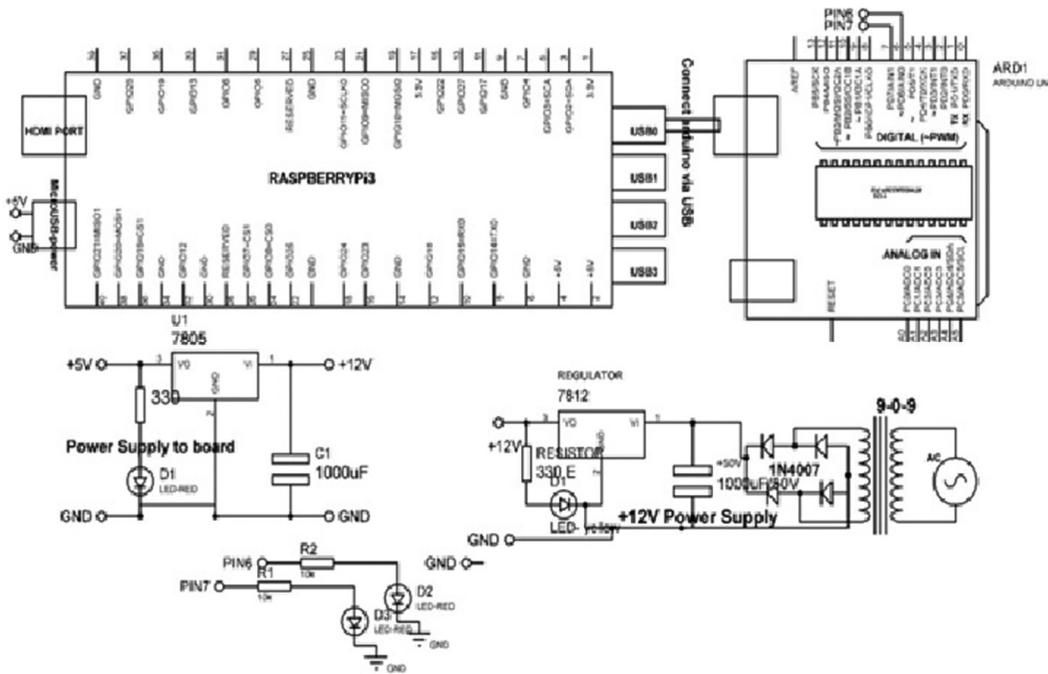
ຮູບ 8.8

ສກርນຂຶ້ອຕໍ່ສໍາຫັກການគົບຄຸມຄວາມສ່ວ່າງ LED

8.4 ການເລືອກຈາກຫາຍຕັວເລືອກ

ເນື່ອຜູ້ໃຊ້ຕ້ອງການເລືອກຈາກຫາຍຕັວເລືອກຈຸດຄ່າທີ່ກຳນົດ ຄວາມສັບສົນຂອງໂຄຮງກາຈະ
 ເພີ່ມຂຶ້ນ ຕັ້ງອ່າງເຫັນ ເນື່ອ LED ຈຳນວນຫາຍຕັວເຊື່ອນຕ່ອກັບບ່ອນດ Arduinio ຜູ້ໃຊ້ຈໍາເປັນຕ້ອງ
 ເລືອກ LED ຮັບ LED ເພື່ອເປີດໃຫ້ຈາກ ໄລບຣາຣී Tkinter ຈັດເຕີຣຍບອນເກອຣ໌ເຟຟສໍາຫັກວິດເຈິ້ຕ
 ທີ່ເຮັດວຽກກ່າວກຳກົດກຳນົດກຳນົດ ແລະ ສາມາດເຂົ້າໃຈໄດ້ວ່າງຈົງຢ່າຍໆ ຂອງການເຊື່ອນຕ່ອງ LED ສອງດວງກັບ Arduinio ແລະ
 Raspberry Pi ໂດຍໃຊ້ Pyrmata

ຮບປະປອບດ້ວຍ Raspberry Pi, Arduinio Uno, ໄວ LED ສອງດວງແລະແແລ່ງຈ່າຍໄວ
 Arduinio ເຊື່ອນຕ່ອກັບ Raspberry Pi ພ່ານ USB ຫັ້ນເອໂນດຂອງ LED1 ແລະ LED2 ເຊື່ອນ
 ຕ່ອກັບພິບ (6) ແລະ ພິບ (7) ຕາມລຳດັບ ພ່ານຕັວຕ້ານການ 10-K ແຕ່ລະຕັວ ແລະຫັ້ວແຄໂກດຂອງ
 LED ກັ້ງສອງເຊື່ອນຕ່ອກັບກຣາວດ (ຮູບກໍ8.9).



ຮູບ 8.9

ແພນກາພວງຈຣເພື່ອເຊື່ອນຕ່ອງ LED ຫາຍດວງ

8.4.1 ສູຕ

```
ນໍາເຂົ້າ Tkinter # ເພີ່ມໄລບຣາຣී Tkinter ນໍາເຂົ້າ
pyrmata # ເພີ່ມເວລານໍາເຂົ້າໄລບຣາຣී pyrmata ຂະໜະ
ຮອ # ເພີ່ມບອർດໄລບຣາຣීເວລາ =
pyrmata.Arduino('/dev/ttyUSB0')
wait.sleep(5) # ລ່າຍ້າ 5 ວິທາກີ
red_led_pin = board.get_pin('d:7:o') # ເຊື່ອມຕ່ອ led ກັບ pin 7 ແລະໃຊ້ເປັນ
ພລພັດ
green_led_pin = board.get_pin('d:6:o') # ເຊື່ອມຕ່ອ led ກັບພັນ 6 ແລະໃຊ້ແລ້ວ
ເປັນພລພັດ
def start_button_press():
    red_led_pin.write(red_led_Var.get())
    green_led_pin.write(green_led_Var.get())
    ພິມພີ 'start...' # ພິມພົບນເກອຮີນັ້ນ
def stop_button_press():
    red_led_pin.write(0) # make pin 7 to HIGH
    green_led_pin.write(0) # make pin 6 to HIGH
    print 'stop....' # ພິມພົບນເກອຮີນັ້ນ
TOP = Tkinter.Tk()
red_led_Var = Tkinter.IntVar()
red_CheckBox = Tkinter.Checkbutton (ບັນຫຼາດ text="Red_LED_status",
    ຕັ້ງແປຣ=red_led_Var)
red_CheckBox.grid (ຄອລັບນີ້ = 1 ແລ້ວ = 1)
green_led_Var = Tkinter.IntVar ()
green_CheckBox = Tkinter.Checkbutton (ບັນຫຼາດ, text="Green_LED_status",
    ຕັ້ງແປຣ=green_led_Var)
green_CheckBox.grid(ຄອລັບນີ້=2 ແລ້ວ=1)
start_Button = Tkinter.Button(TOP, text="Start_button", ຄຳສັ່ງ =
    start_button_press)
start_Button.grid(ຄອລັບນີ້=1,ແລ້ວ=2)
stop_Button = Tkinter.Button (TOP, text="Stop_button", ຄຳສັ່ງ =
    stop_button_press)
stop_Button.grid(ຄອລັບນີ້=2 ແລ້ວ=2)
exit_Button = Tkinter.Button (ບັນຫຼາດ text="Exit_button" ຄຳສັ່ງ = TOP
    ເລີກ)
exit_Button.grid(ຄອລັບນີ້=3,ແລ້ວ=2)
TOP.mainloop()
```



ຮູບກໍ 8.10
Tkinter GUI ສໍາຮັບ LED ລາຍດວງ

```
pi@raspberrypi:~ $ nano pyfir_two_led_tkinter_B00K.py
pi@raspberrypi:~ $ python pyfir_two_led_tkinter_B00K.py
start...
stop....
```

ຮູບກໍ 8.11
ສກເຮັບຂຶ້ອຕໍ່ສໍາຮັບເຫັນດີ່ນທີ່ສໍານະ

8.4.2 Tkinter GUI

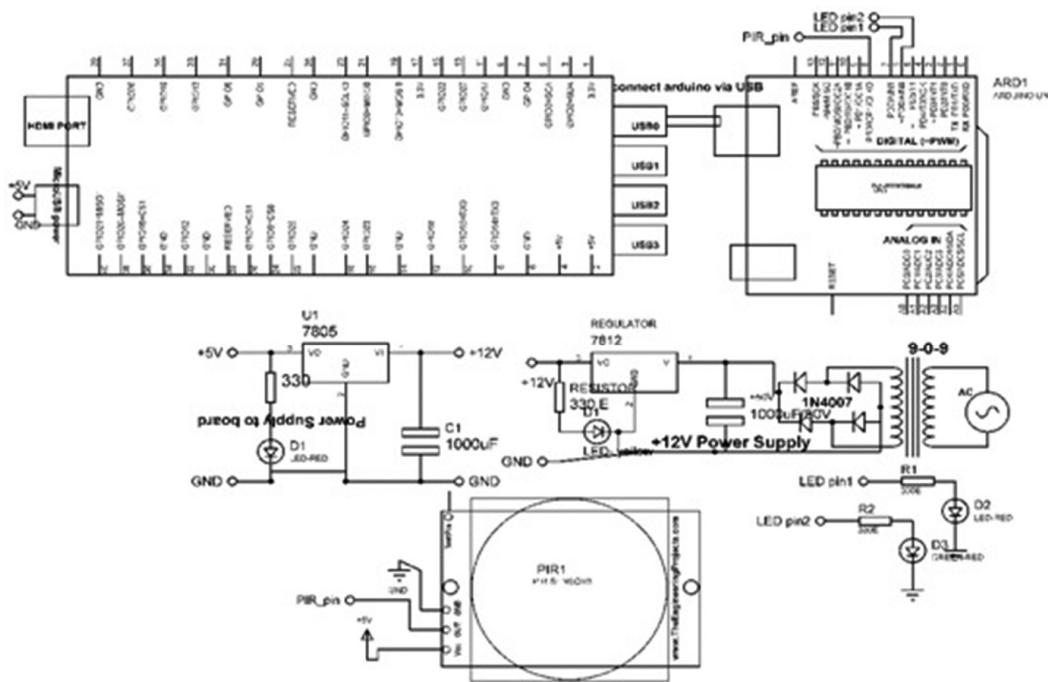
ໃນການສ້າງ Tkinter GUI ສໍາຮັບ LED ລາຍດວງ ໃຫ້ຮັບໂປຣແກຣມທີ່ອີເບາຍໄວ້ໃນ [ບາຕາຮາ](#) **8.4.1** ແລະ ໜ້າມຕ່າງຈະປາກຸງຂຶ້ນ ([ຮູບກໍ 8.10](#)). GUI ມີວິດເຈື້ຕ Checkbutton() ສອງອັນສໍາຮັບ ແຕ່ລະ LED ສີແດງແລະສີເຊີຍວ່າ ຜູ້ໃຊ້ສາມາຄເລືອກ LED ກີລະດວງຫຼືອຽນກັນເພື່ອເຮັ່ນຕົ້ນຫຼືອ ພູດ ([ຮູບກໍ 8.11](#)).

8.5 ການອ່ານເຊື້ນເຊື່ອຣ PIR

ເຊື້ນເຊື່ອຣອັນຟຣາເຣດແບບໄຟໂຣອັເລີກທົກ (PIR) ໃຊ້ສໍາຮັບຕຽບຈັບການເຄລື່ອນໄວ້ ຮະບບາໄດ້ຮັບ ການອອກແບບດ້ວຍເຊື້ນເຊື່ອຣ PIR ໂດຍກໍ່ LED ສອງດວງແສດງສໍານະການເຄລື່ອນໄວ້ ກາກຕຽບ ໄນພົບການເຄລື່ອນໄວ້ໄຟ LED ສີເຊີຍວ່າຈະຕິດສ່ວ່າງ ແລະ ກາກຕຽບພົບການເຄລື່ອນໄວ້ ຮະບບະຈະ ສ້າງການແຈ້ງເຕືອນດ້ວຍການກໍາໃໝ່ LED ສີແດງເປັນ "ເປີດ" ຮະບບເປີດ/ປິດດ້ວຍ Tkinter GUI ສໍານະຂອງເຊື້ນເຊື່ອຣຈະພິມພົບນພຣອມຕົ້ນຂອງ Python ຮະບບປະກອບດ້ວຍ Raspberry Pi, Arduino, LED ສອງດວງ, ເຊື້ນເຊື່ອຣ PIR ແລະ ແກ່ລ່າງຈ່າຍໄຟ Arduino ເຊື້ອນຕ່ອກັບ Raspberry Pi ຜ່ານ USB ຫັ້ນແອໂນໂດຂອງ LED1 ແລະ LED2 ເຊື້ອນຕ່ອກັບພິນ (7) ແລະພິນ (6) ຂອງ Arduino ຕາມລຳດັບ ພິນ (OUT) ຂອງເຊື້ນເຊື່ອຣ PIR ເຊື້ອນຕ່ອກັບພິນ (8) ຂອງ Arduino ແລະພິນ (Vcc) ແລະພິນ (GND) ເຊື້ອນຕ່ອກັບ +5 V DC ແລະ ກຣາວດຕາມລຳດັບ ([ຮູບກໍ 8.12](#)).

8.5.1 ສູຕຣການອ່ານເຊື້ນເຊື່ອຣ PIR

```
# ປັບປຸງກໍ່ສໍາຮັບຕຽບຈັບການເຄລື່ອນໄວ້
# ເພີ້ມກຳນົດເອງເພື່ອດຳເນັບການ Blink def
blink_LED (ພິນ, ຊົ້ວຄວາມ):
```



ຮູບກໍ8.12
 ແຜນກາພວງຈර້າຫັກບກາຣເຊື່ອມຕ່ອເຊັນເຊອຣ PIR

Motion_Label.cong(ຂ້ອຄວາມ=ຂ້ອຄວາມ)

Motion_Label.update_idletasks()

TOP.update()

pin.write(1) # ປັກໜຸດໄວ້ກໍ HIGH

wait.sleep(1) # ລ້າໜ້າ 1 ວິບາກໍ

pin.write(0) # ປັກໜຸດໄວ້ກໍ HIGH

wait.sleep(1) # ລ້າໜ້າ 1 ວິບາກໍ

ປົງເສດກະກຳທີ່ເກີຍຂ້ອງກັບປຸນ Start ກດປຸນ def

press_start_button():

ໃນຂນະກິຈ:

ຄ້າ FLAG.get():

ຄ້າ PIR_pin.read() ເປັນຈິງ:

blink_LED(red_led_pin, "motion_status:Y")

ພິມພໍ 'ຕຽວຈັບກາຣເຄລືອນໄຫວ' # ພິມພົບນເກອຮນັບັດ

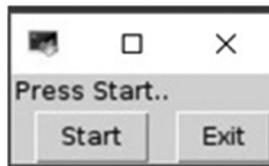
ອືບ:

blink_LED (green_led_pin, "motion_status:N") ພິມພໍ

'ໄມ້ມີກາຣເຄລືອນໄຫວ' # ພິມພົບນເກອຮນັບັດ

ວິນ:

```
    ထູດພັກ
    board.exit()
    TOP.ກໍາລາຍ()
def press_exit_button():
    FLAG.set(ເຖິງ)
    ບໍາເຫຼົາ Tkinter # ບໍາເຫຼົາຫ້ອງສມຸດ Tkinter ບໍາເຫຼົາ
    pyrmata # ບໍາເຫຼົາຫ້ອງສມຸດ pyrmata ບໍາເຫຼົາຕາມ
    ເວລາຮອ # ບໍາເຫຼົາກະຕາມຫ້ອງສມຸດເວລາ =
    pyrmata.Arduino('/dev/ttyUSB0')
    wait.sleep(5) # ສອ 5 ວິນາທີ
    PIR_pin = board.get_pin('d:8:i') # ເຊື່ອມຕ່ອງເຊື້ນເຊື່ອຣ PIR ກັບ pin8 ແລະເປີນ
        ບັບຂ້ອມມູນ
    red_led_pin = board.get_pin('d:7:o') # ເຊື່ອມຕ່ອງ LED ສີແດງກັບ pin7 ແລະເປີນ
        ພຸລົມ
    green_led_pin = board.get_pin('d:6:o') # ເຊື່ອມຕ່ອງ Green ບຳໄປສູ່ pin6 ແລະ
        ເປີນພຸລົມ
    # ໃໃໝ່ເຮັດ iterator ເພື່ອເລືອກເລື່ອງບັຟຟຝ່ອຣ overow =
    pyrmata.util.Iterator (ບອຮົດ)
    it.start()                      # ເຮັດຕັ້ງ
    PIR_pin.read() # ອ່ານເຊື້ນເຊື່ອຣ PIR
    # ເຮັດຕັ້ນຫຼັບຕ່າງໆ ດ້ວຍຊື່ແລະຂາດ TOP =
    Tkinter.Tk()
    TOP.title("PIR_sensor_for_motion")
    # ສ້າງບ້າຍກຳກັບສໍາຮຽບການຕຽບຈັບການເຄລືອນໄຫວ Motion_Label =
    Tkinter.Label(TOP, text="Press Start..")
    Motion_Label.grid(column=1, row=1)
    # ສ້າງ ag ເພື່ອກຳນົດກັບ indenite while loop
    FLAG = Tkinter.BooleanVar(TOP)
    FLAG.set(ັກສູ່)
    # ສ້າງປຸມ Start ແລະເຊື່ອໂຢັງກັບ onStartButtonPress method
    Start_Button = Tkinter.Button (TOP , text="Start", ຄຳສັ່ງ = ກົດ _start_
        ປຸມ)
    Start_Button.grid(ຄອລັນບົດ=1,ແຄວ=2)
    # ສ້າງປຸມ Stop ແລະເຊື່ອໂຢັງກັບເມຣອດ onStopButtonPress
```



ຮູບກໍ8.13

Tkinter GUI ສໍາຫຼັບການຕຽບຈັບຄວາມເຄລືອນໄວ

```
pi@raspberrypi:~ $ nano pyfir_digital_in.tkinter_B00K.py
pi@raspberrypi:~ $ python pyfir_digital_in.tkinter_B00K.py
No motion
No motion
No motion
No motion
```

ຮູບກໍ8.14

ໜ້າຕ່າງສຄາບສໍາຫຼັບເຊືນເຊືອຮ່ວງຈັບຄວາມເຄລືອນໄວ

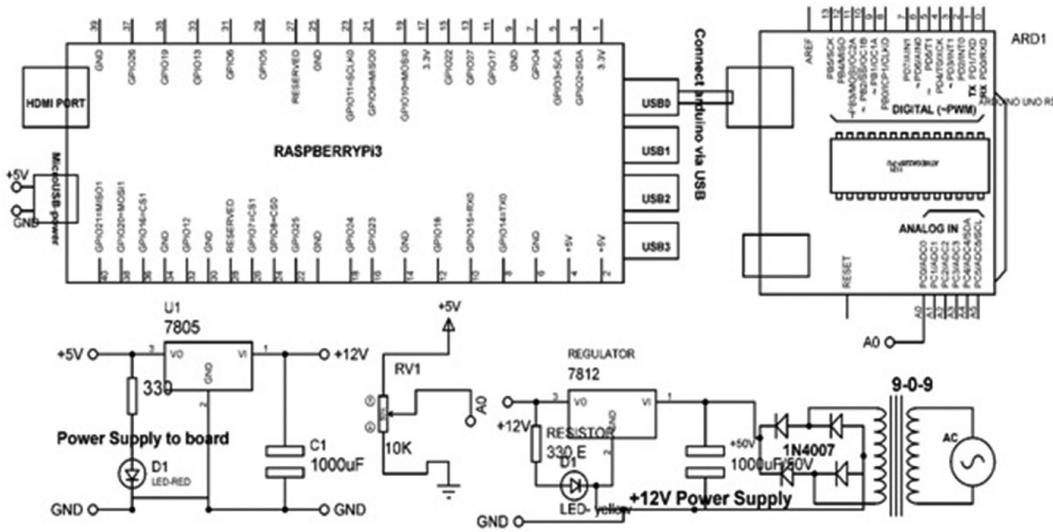
```
Stop_Button = Tkinter.Button (ບນສຸດ, text="Exit", command=press_exit_
    ປຸມ)
Stop_Button.grid(ຄອລັນບົດ=2,ແຄວ=2)
# ເຮັດແລະເປີດໜ້າຕ່າງ
TOP.mainloop()
```

8.5.2 Tkinter GUI

ໃນການສ້າງ Tkinter GUI ສໍາຫຼັບສຄາບການເຄລືອນໄວ ໃຫ້ຮັນໂປຣແກຣມທີ່ອົບຍາໄວ້ໃນ [ມາຕຣາ 8.5.1](#) ແລະ ໜ້າຕ່າງຈະປາກຸງຂຶ້ນ ([ຮູບກໍ8.13](#)). ເນື້ອກດຸນ "ເຮັດ" ຈະກຳໃຫ້ສຄາບຂອງເຊືນເຊືອຮ່ວງ ຕຽບຈັບຄວາມເຄລືອນໄວເປັນ "ໄມ້ມີການເຄລືອນໄວ" ທີ່ຈະເປັນ "ຕຽບພົບການເຄລືອນໄວ" ([ຮູບກໍ8.14](#)).

8.6 ການອ່ານເຊືນເຊືອຮ່ວງອາລືອກ

ໄຟເທັນຊີອອົມືເຕັອຣ (POT) ນັນຄ້າຍຄລິ້ງກັບເຊືນເຊືອຮ່ວງອາລືອກ ການປັບປຸງແປລົງຂອງແຮງດັບໄຟຂາອຸກສາມຮັບສັງເກດໄດ້ຈາກການປັບປຸງຄ່າຄວາມຕ້າມການໂດຍການເລືອນປຸນ ໃນການສ້າງ Tkinter GUI ສໍາຫຼັບ POT ຮະບບປະກອບດ້ວຍ Raspberry Pi, Arduino, 10-K POT ແລະ ແກ່ລົງຈ່າຍໄວ Arduino ເຊື່ອມຕ່ອກກັບ Raspberry Pi ພ່ານ USB ມີມີສາມຂັ້ງ: ບົດ



ຮູບຖ້ວທ 8.15
 ແນບກາພວງຈරສໍາຫຼັບເຊື່ອນຕ່ວ່າ POT

ເກອຮົບນັດລັບເຊື່ອນຕ່ວ່າກັບ +5 V ແລະ ສ່ວນອັນໆ ຕ່ອກຮາວດ ກີບດ້ານັ້າຝນຂອງ POT ເຊື່ອນຕ່ວ່າກັບພິບ (A0) ຂອງ Arduino ([ສູກຖ້ວທ 8.15](#)).

8.6.1 ສູຕຣ

```
# ປັບປຸງການກະທຳກີບປຸມ Start ໃຫ້ກົດ def
start_button_press():
```

ໃນຂະນະກ່ຽວຂ້ອງ:

ຄ້າ FLAG.get():

```
analog_Read_Label.config(text=str(a0.read()))
analog_Read_Label.update_idletasks()
TOP.update()
ພິມພ 'ຄ່າແອນະລືອຂອງ pot' ອຢ່າງ
```

ອັບ:

留守ພັກ

board.exit()

TOP.ກໍາລາຍ()

```
# ປັບປຸງການກະທຳກີບປຸມອອກ ກົດ def
exit_button_press():
    FLAG.set(ເທົ່າງ)
```

```
ນໍາເຂົາ Tkinter # ນໍາເຂົາຫ້ອງສມຸດ Tkinter ນໍາເຂົາ
pyrmata# ນໍາເຂົາຫ້ອງສມຸດ pyrmata ນໍາເຂົາເວລາຮອ
# ນໍາເຂົາບອົດຫ້ອງສມຸດເວລາ =
pyrmata.Arduino('/dev/ttyUSB0')
wait.sleep(5)
ມັນ = pyrmata.util.Iterator(ບອົດ)
it.start() # start iterator
# ກໍາທັດບທກທາງແລະຕັ້ງແປຣໃຫ້ກັບຂາວະນາລືອກ 0 a0 =
board.get_pin('a:0:i') # ເຊື່ອນຕ່ອຂາເຊັນເຊອ່ນ A0
# ເຮັດຕັ້ນຫຼັບຕ່າງໜ້າຕ່າງໜ້າດ້ວຍຊື່ແລະຂາດ TOP =
Tkinter.Tk()
TOP.title("ການອ່ານໜຸດ POT")
# ສ້າງບ້າຍກຳກັບເພື່ອອ່ານອົນພຸດແບບອະນາລືອກ
description_Label = Tkinter.Label(TOP, text="POT_input:- ")
description_Label.grid(column=1, row=1)
# ສ້າງບ້າຍກຳກັບເພື່ອອ່ານອົນພຸດແບບອະນາລືອກ
analog_Read_Label = Tkinter.Label(TOP, text="Press_Start_process")
# ການຕັ້ງຄ່າ ag ເພື່ອສລັບການອ່ານຕັ້ງເລື່ອກ
FLAG = Tkinter.BooleanVar (TOP)
FLAG.set (ຈົງ)
# ສ້າງປຸມເຮັດແລະເຊື່ອນໂຢັງກັບວິທີ onStartButtonPress
start_Button = Tkinter.Button (ບບສຸດ text="Start_reading",
    command=start_button_press)
start_Button.grid(ຄອລັນບົດ=1,ແຄວ=2)
# ສ້າງປຸມ Stop ແລະເຊື່ອນໂຢັງກັບວິທີ onStopButtonPress
exit_Button = Tkinter.Button (ບບສຸດ, text="Exit_reading",
    ຄໍາສັ່ງ=exit_button_press)
exit_Button.grid(ຄອລັນບົດ=2,ແຄວ=2)
# ເຮັດແລະເປີດຫຼັບຕ່າງ
TOP.mainloop()
```

8.6.2 Tkinter GUI

ເຮັດໃຊ້ໂປຣແກຣມທີ່ອີເມວຢ້າງໄວ້ໃນມາຕາ 8.6.1 ເພື່ອສ້າງ Tkinter GUI ສໍາຮັບອ່ານ POT ແລະ
ຫຼັກຕ່າງຈະປາກໝູ້ນ (ຮູບຖ້ວນ 8.16). ເນື້ອກດຸ່ມ “Start_reading” ຈະມີການອ່ານຄ່າ POT (ຮູບຖ້ວນ
8.17).



ຮູບຖ້ວນ 8.16
Tkinter GUI ສໍາຮັບອ່ານ POT

```
pi@raspberrypi:~ $ nano analog_in_tkinter.py
pi@raspberrypi:~ $ python analog_in_tkinter.py
analog values of pot
analog values of pot
analog values of pot
```

ຮູບຖ້ວນ 8.17
ຫຼັກຕ່າງແສດງຄ່າແອນະລຶກຂອງ POT