

## บทที่ 6

### การใช้งาน ET-IR REMOTE KEY

ในบทนี้กล่าวถึงการเขียนโปรแกรมด้วยภาษาไพธอนเพื่อเชื่อมประสานการทำงานของรีโมทรุ่น ET-IR REMOTE KEY กับบอร์ด ET-iLED4x4 RGB เพื่ออ่านค่าหมายเลขปุ่มที่ผู้ใช้กด พร้อมทั้งมีตัวอย่างประยุกต์ใช้การสั่งงานไฟวิ่งแอลอีดี และเล่นเสียงตัวโน้ตดนตรีจากการกดรีโมท

เมื่อจบบทนี้ ผู้อ่านจะเข้าใจถึงขั้นตอนการเขียนโปรแกรม และวิธีการอ่านค่าจากรีโมทรุ่น ET-IR REMOTE KEY เพื่อสั่งงานขา GPIO ที่เชื่อมต่อกับอุปกรณ์ต่อพ่วงอื่นๆ และประยุกต์ใช้กับอุปกรณ์ประเภทอื่นๆ ดังที่จะกล่าวถึงในตัวอย่างในบทที่เกี่ยวข้องต่อไป

#### 1. แนะนำ ET-IR REMOTE KEY

ET-IR REMOTE KEY เป็น KEY IR REMOTE ขนาดเล็ก ความถี่ 38 KHz จำนวนคีย์ 20 คีย์ใช้งาน ขนาด 8.65 x 4.10 x 0.75 โดยใช้แบตเตอรี่แรงดัน 3VDC ดังภาพที่ 6-1-1 ด้านซ้ายและขวาตามลำดับ ซึ่งในการใช้งานจะต้องใช้คู่กับวงจร ET-iLED4x4 RGB หรือต่อวงจรภาครับดังภาพที่ 6-3.1-1



ภาพที่ 6-1-1 (ซ้าย) ET-IR REMOTE KEY (ขวา) แบตเตอรี่ 3VDC สำหรับใช้งาน

#### 2. ทำความเข้าใจเรื่อง IR

การรับส่งข้อมูลด้วยแสงอินฟราเรด หรือ IR (Infra-red) เป็นการสื่อสารข้อมูลแบบไร้สายที่อาศัยย่านความถี่ในช่วงคลื่นอินฟราเรดที่ไม่สามารถมองเห็นด้วยตาเปล่า ซึ่งนิยมนำมาใช้กับระบบควบคุมด้วยรีโมท

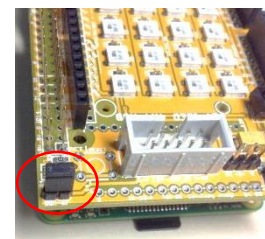
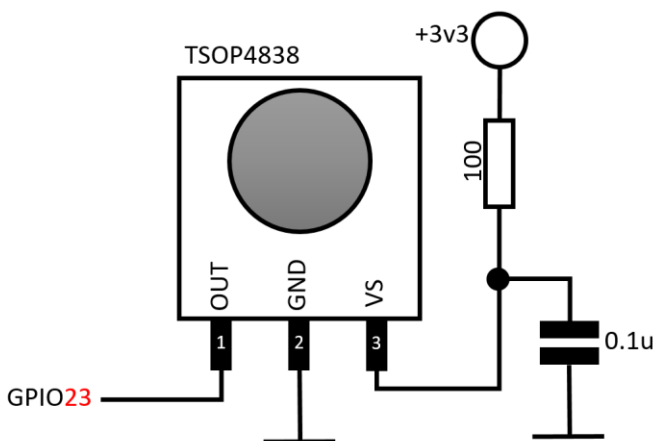
การทำงานของระบบอินฟราเรดประกอบไปด้วยภาครับและภาคส่งที่เป็นแอลอีดีแบบอินฟราเรดที่เป็นภาคส่งซึ่งติดตั้งอยู่ที่ชุด ET-IR REMOTE KEY และแอลอีดีอินฟราเรดภาครับที่ติดตั้งอยู่ที่ ET-iLED4x4 RGB ดังภาพที่ 6-2-1



ภาพที่ 6-2-1 (ซ้าย) แอลอีดีภาคส่งที่ ET-IR REMOTE KEY  
(ขวา) ไอซี TSOP4838 ที่เป็นภาครับที่ติดตั้งอยู่กับ ET-iLED4x4 RGB

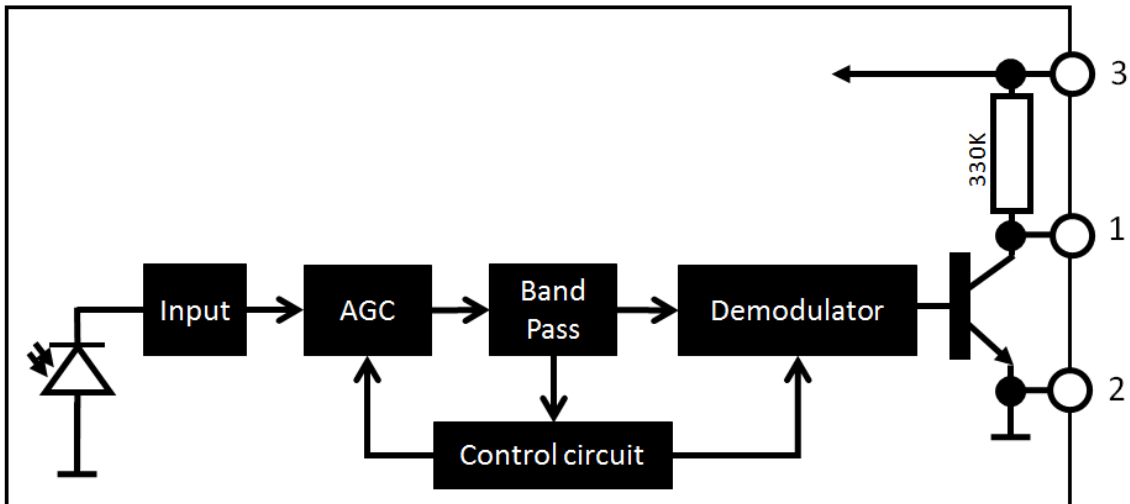
## 2.1 ภาครับ

บอร์ด ET- มีภาครับสัญญาณอินฟราเรดโดยใช้ไอซี TSOP4838 ดังวงจรภาพที่ 6-3.1-1 ซึ่งต่อเข้ากับขา GPIO หมายเลข 23 ซึ่งการทำงานภายในของไอซีตัวนี้แบ่งเป็น 5 ส่วนดังภาพที่ 6-3.1-2 คือ ส่วนนำเข้าข้อมูลจากออปโตไดโอด (Input) ผ่านเข้ากระบวนการ AGC (Automatic Gain Control) ที่ทำหน้าที่ควบคุมสัญญาณรบกวน แล้วส่งข้อมูลที่ตัดสัญญาณรบกวนไปให้ส่วน Band Pass สุดท้ายทำการถอดการเข้ารหัสสัญญาณ (Demodulator) เพื่อส่งข้อมูลออกมาขาที่ 1 ซึ่งการทำงานทั้งส่วน AGC/Band Pass และถอดการเข้ารหัสสัญญาณจะถูกควบคุมการทำงานโดยวงจรควบคุมภายในไอซี (Control Circuit)



Short Jumper JP\_VDD สำหรับไฟเลี้ยง IR 3.3V จาก บอร์ด RPI

ภาพที่ 6-3.1-1 วงจรของ TSOP4838 และการต่อ ไฟเลี้ยงIR(3.3V) สำหรับบอร์ด ET-iLED4x4 RGB



ภาพที่ 6-3.1-2 ผังการทำงานภายในไอซี TSOP4838

## 2.2 การตั้งค่า

วิธีการเข้าถึงอุปกรณ์ IR จะต้องติดตั้งไลบรารี LIRC (Linux Infrared Remote Control) ซึ่งเป็นไลบรารีแบบเปิดเผยโค้ดต้นฉบับที่พัฒนาโดย Aron Szabo (เว็บไซต์ [aron.ws/projects/lirc\\_rpi/](http://aron.ws/projects/lirc_rpi/)) และติดตั้งโดยใช้คำสั่งดังนี้

```
sudo apt-get install lirc liblircclient-dev
```

หลังติดตั้งไลบรารี lirc ต้องทำการเรียกให้โมดูล lirc ทำงานโดยการเพิ่มบรรทัดคำสั่งต่อไปนี้ในไฟล์ /etc/modules ด้วยคำสั่ง `sudo nano /etc/modules` แล้วสั่งบันทึกด้วย `Ctrl+O` และออกจากโปรแกรม nano ด้วย `Ctrl+X`

```
lirc_dev
lirc_rpi gpio_in_pin=23
```

ซึ่งบรรทัดที่ 2 เป็นการกำหนดหมายเลขขาที่เชื่อมต่อกับภาครับ IR ซึ่งบนบอร์ด ET-iLED4x4 RGB ได้เชื่อมเข้ากับขาหมายเลข 23 หลังจากนั้น ใช้คำสั่ง `sudo nano /etc/lirc/hardware.conf` เพื่อกำหนดชื่อไดรเวอร์ที่ต้องใช้ โดยกำหนดค่าของ LIRD\_ARGS, DRIVER, DEVICE และ MODULES จากเดิมที่เขียนให้เป็นตามการตั้งค่าด้านล่างนี้ แล้วสั่งบันทึกด้วย `Ctrl+O` และออกจากโปรแกรม nano ด้วย `Ctrl+X`

```
# /etc/lirc/hardware.conf
#
# Arguments which will be used when launching lircd
LIRCD_ARGS="--uinput"

# Don't start lircmd even if there seems to be a good config file
# START_LIRCMD=false

# Don't start irexec, even if a good config file seems to exist.
# START_IEXEC=false

# Try to load appropriate kernel modules
LOAD_MODULES=true

# Run "lircd --driver=help" for a list of supported drivers.
DRIVER="default"
# usually /dev/lirc0 is the correct setting for systems using udev
DEVICE="/dev/lirc0"
MODULES="lirc_rpi"

# Default configuration files for your hardware if any
LIRCD_CONF=""
LIRCMD_CONF=""
```

สั่งให้ LIRC ทำงานใหม่ โดยสั่งคำสั่งต่อไปนี้

```
sudo /etc/init.d/lirc stop
```

```
sudo /etc/init.d/lirc start
```

แก้ไขไฟล์ /boot/config.txt โดยใช้คำสั่ง `sudo nano /boot/config.txt` แล้วเพิ่มบรรทัดต่อไปนี้ที่ท้ายไฟล์ แล้วกด `Ctrl+O` เพื่อบันทึก และ `Ctrl+X` เพื่อออกจาก nano

```
dtoverlay=lirc-rpi:gpio_in_pin=23
```

ให้รีบูตระบบใหม่ด้วยคำสั่ง

```
sudo reboot
```

วิธีการทดสอบการทำงานของ lirc ทำโดยสั่ง 2 บรรทัดต่อไปนี้ และกดปุ่มที่ ET-IR REMOTE KEY ถ้าการทำงานถูกต้องจะแสดง หมายเลขค่าที่ได้จากการกดแป้นต่างๆ ของ ET-IR REMOTE KEY เป็นดังตารางที่ 6-3.1-1

```
sudo /etc/init.d/lirc stop
mode2 -d /dev/lirc0
```

จะได้ตัวอย่างของผลลัพธ์ดังนี้

```
space 3689839
pulse 9666
space 4506
pulse 629
space 544
pulse 630
space 544
pulse 630
space 544
pulse 605
space 569
pulse 626
```

ขั้นตอนต่อไปคือสร้างไฟล์ชื่อแป้นของ ET-IR REMOTE โดยใช้คำสั่งดังรูปแบบต่อไปนี้

```
irrecord -d /dev/lirc0 ./lircd.conf
```

















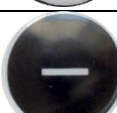



วิธีการตั้งค่าหลังจากกดแป้น Enter ให้ทยอยกดแป้นบนรีโมททีละแป้น ให้สังเกต เครื่องหมายจุด (.) ที่จอของ Raspberry Pi ที่จะแสดงเมื่อกดแป้น ทุกครั้งที่จุดแสดงขึ้นแล้วให้ เปลี่ยนไปกดแป้นถัดไป ทำจนกว่าจะครบ 20 ปุ่ม และทำต่อไปจนครบ 4 รอบ หรือได้เครื่องหมายจุด ครบเต็มบรรทัด แต่ถ้าผิดพลาด ผู้เขียนแนะนำให้กด Ctrl+C เพื่อออกจากโปรแกรม irrecord แล้ว สั่งลบ lircd.conf ด้วยคำสั่ง `rm ./lircd.conf`

เมื่อทำจนครบ 1 บรรทัด โปรแกรมจะให้ทำแบบเดิมอีกครั้งจนได้เครื่องหมายจุดครบบรรทัด ขั้นตอนต่อไปคือใส่ชื่อของแป้นแต่ละอันของรีโมท โดยจะมี 2 ขั้นตอนในแต่ละปุ่มคือ ตั้งชื่อปุ่ม ถ้ารูปแบบถูกต้องจะให้ผู้ใช้กดแป้นนั้นที่รีโมทจนโปรแกรมถามถึงชื่อของแป้นถัดไป ให้ทำจนครบ

ทั้งหมด แล้วกด Enter เพื่อออกจากโปรแกรมและโปรแกรมจะสร้างไฟล์ lircd.conf ให้ดังตัวอย่างต่อไปนี้

```
# Please make this file available to others
# by sending it to <lirc@bartelmus.de>
#
# this config file was automatically generated
# using lirc-0.9.0-pre1(default)on Sat Jan 14 15:47:24 2017
#
# contributed by
#
# brand:          ./lircd.conf
# model no.of remote control:
# devices being controlled by this remote:
#
begin remote
name ./lircd.conf
  bits          16
  flags SPACE_ENC
  eps           30
  aeps          100
  header        9581 4490
  one           638 1646
  zero          638 531
  ptrail        638
  repeat        9590 2205
  pre_data_bits 16
  pre_data      0xFF
  gap           40302
  repeat_gap    96992
  toggle_bit_mask 0x0
begin codes
  KEY_POWER          0xA25D
  KEY_MENU           0xE21D
  KEY_T              0x22DD
  KEY_P              0x02FD
  KEY_B              0xC23D
  KEY_L              0xE01F
  KEY_E              0xA857
  KEY_R              0x906F
  KEY_0              0x6897
  KEY_M              0x9867
  KEY_C              0xB04F
  KEY_1              0x30CF
  KEY_2              0x18E7
  KEY_3              0x7A85
  KEY_4              0x10EF
  KEY_5              0x38C7
  KEY_6              0x5AA5
  KEY_7              0x42BD
  KEY_8              0x4AB5
  KEY_9              0x52AD
end codes
end remote
```

ตารางที่ 6-3.1-1 แสดงค่าของแป้น ET-IR REMOTE KEY

แป้น	ค่าที่อ่านได้	แป้น	ค่าที่อ่านได้	แป้น	ค่าที่อ่านได้
	0x6897 KEY_0		0x30CF KEY_1		0x18E7 KEY_2
	0x7A85 KEY_3		0x10EF KEY_4		0x38C7 KEY_5
	0x5AA5 KEY_6		0x42BD KEY_7		0x4AB5 KEY_8
	0x52AD KEY_9		0xC23D KEY_B		0xB04F KEY_C
	0xE01F KEY_L		0xA857 KEY_E		0x906F KEY_R
	0x02FD KEY_P		0x9867 KEY_M		0x22DD KEY_T
	0xA25D KEY_POWER		0xE21D KEY_MENU		

ขั้นตอนต่อไปคือ ให้คัดลอกไฟล์ ./lircd.conf ไปไว้ที่ /etc/init.d/lirc และสั่งเริ่มต้นการทำงานของ lirc โดยใช้ชุดคำสั่งดังนี้

```
sudo cp lircd.conf /etc/lirc/
sudo /etc/init.d/lirc restart
```

การทดสอบว่าชื่อแป้นตรงหรือไม่ให้เรียกใช้โปรแกรมดังนี้

```
irw
```

เมื่อกดแป้นบนรีโมทจะแสดงชื่อของแป้นดังตัวอย่างต่อไปนี้

```
000000000ffa25d 00 KEY_POWER ./lircd.conf
000000000ffe21d 00 KEY_MENU ./lircd.conf
000000000ff02fd 00 KEY_P ./lircd.conf
000000000ffc23d 00 KEY_B ./lircd.conf
000000000ff30cf 00 KEY_1 ./lircd.conf
000000000ff18e7 00 KEY_2 ./lircd.conf
000000000ff7a85 00 KEY_3 ./lircd.conf
000000000ff10ef 00 KEY_4 ./lircd.conf
```

### 2.3 เขียนโปรแกรมอ่านค่า

การติดตั้งไลบรารี LIRC สำหรับไพธอนสามารถทำได้โดยการสั่งคอมไพล์โปรแกรมและทำการติดตั้ง โดยเข้าไปที่ `~/src/pylirc2-0.1` และสั่งชุดคำสั่งดังต่อไปนี้

```
sudo python3 setup.py install
cd /usr/local/lib/python3/dist-packages
sudo mv pylircmodule.cpython-34m.so pylirc.cpython-34m.so
cd ~/src/pylirc2-0.1
```

เมื่อติดตั้งไลบรารีสำเร็จ ขั้นตอนต่อไปคือสร้างไฟล์ตั้งค่าสำหรับโปรแกรม โดยในที่นี้ผู้เขียนใช้ชื่อไฟล์ว่า `ett_ir_conf` โดยตั้งชื่อของโปรแกรมเป็น `ett` วิธีการสร้างไฟล์ทำได้โดยการสั่ง `nano ett_ir_conf` และพิมพ์ข้อมูลต่อไปนี้

```
begin
    remote = *
    button=KEY_POWER
    prog = ett
    config = Power
end
begin
    remote = *
    button= KEY_MENU
    prog = ett
    config = menu
end
begin
    remote = *
    button = KEY_T
    prog = ett
    config = test
end
begin
    remote = *
    button = KEY_P
    prog = ett
    config = +
end
begin
    remote = *
    button = KEY_M
    prog = ett
    config = -
end
begin
    remote = *
    button = KEY_E
    prog = ett
    config = play
end
begin
    remote = *
    button = KEY_B
    prog = ett
    config = back
end
begin
    remote = *
    button = KEY_C
    prog = ett
    config = C
end
begin
    remote = *
    button = KEY_0
    prog = ett
    config = 0
end
begin
    remote = *
    button = KEY_1
    prog = ett
    config = 1
end
begin
    remote = *
    button = KEY_4
    prog = ett
    config = 4
end
begin
    remote = *
    button = KEY_5
    prog = ett
    config = 5
end
begin
    remote = *
    button = KEY_6
    prog = ett
    config = 6
end
begin
    remote = *
    button = KEY_7
    prog = ett
    config = 7
end
begin
    remote = *
    button = KEY_8
    prog = ett
    config = 8
end
```



```

begin
  remote = *
  button = KEY_L
  prog = ett
  config = left
end
begin
  remote = *
  button = KEY_R
  prog = ett
  config = right
end
begin
  remote = *
  button = KEY_2
  prog = ett
  config = 2
end
begin
  remote = *
  button = KEY_3
  prog = ett
  config = 3
end
begin
  remote = *
  button = KEY_9
  prog = ett
  config = 9
end

```

ไลบรารี pylirc มีชุดคำสั่งสำหรับใช้งาน 4 คำสั่ง คือ `init()`, `nextcode()`, `exit()` และ `blocking()` โดยการเรียกใช้งานไลบรารีจะต้องใช้ชุดคำสั่งนำเข้าดังนี้ และแต่ละคำสั่งมีหน้าที่ดังหัวข้อ 2.3.1, 2.3.2, 2.3.3 และ 2.3.4

```
import pylirc
```

### 2.3.1 คำสั่งเริ่มต้นทำงาน

คำสั่งสำหรับตั้งค่าเพื่อเริ่มต้นทำงานมีรูปแบบการใช้งานดังต่อไปนี้

```
ตัวแปรจำนวนเต็ม = pylirc.init( string name
                                [, string configuration
                                [, integer blocking]])
```

โดยที่ ตัวแปรจำนวนเต็ม คือ ตัวแปรเก็บผลลัพธ์จากการสั่งตั้งค่าเพื่อเริ่มต้นทำงาน และ

คืนค่ากลับมา 2 ลักษณะได้แก่

- 1) 0 เมื่อการตั้งค่าล้มเหลว
- 2) ตัวเลขจำนวนเต็มที่เป็นค่าหมายเลขช่องทางสื่อสาร (socket)

`name` คือ ชื่อของโปรแกรมที่ได้ระบุในไฟล์ `configuration`

`configuration` คือ ชื่อไฟล์การตั้งค่าของ LIRC ซึ่งโดยปกติคือไฟล์ `~/.lircrc` หรือกำหนดขึ้นเองได้

`blocking` คือ ค่ากำหนดสถานะการทำงานแบบการบล็อก เพื่อรองรับการทำงานในลักษณะของมัลติทาสก์ (multi-tasking) หรือมัลติเธรดิง (multi-threading) ค่าที่ใช้กำหนดคือ

- 1) 0 หมายถึงไม่ใช้การบล็อก
- 2) 1 หมายถึงใช้การบล็อก

### 2.3.2 การอ่านค่า

การทำงานของการทำงานอ่านค่าเป็นแบบโพลลิง (Polling) คือ ผู้เขียนโปรแกรมต้องเขียนโปรแกรมเพื่อไปอ่านค่าจาก LIRC เนื่องจากไลบรารีไม่มีการสร้างสัญญาณขัดจังหวะหรือการแจ้งเตือนเมื่อมีข้อมูล โดยรูปของคำสั่งอ่านค่าเป็นดังนี้

```
ตัวแปรลิสต์ = pylirc.nextcode( )
```

ตัวอย่างการอ่านข้อมูลจาก ET-IR REMOTE และแสดงผล สามารถเขียนได้ดังนี้

```
ir_code = pylirc.nextcode( )  
print(ir_code)
```

### 2.3.3 ออกจากการเชื่อมต่อ

คำสั่งสำหรับการออกจากการเชื่อมต่อกับ LIRC มีรูปแบบดังนี้

```
pylirc.exit()
```

### 2.3.4 เปลี่ยนโหมด

การเปลี่ยนโหมดคือการเปลี่ยนการทำงานของการทำงานการบล็อกโดยรูปแบบของคำสั่งสำหรับการเปลี่ยนโหมดเป็นดังนี้

```
ผลลัพธ์ = pylirc.blocking( สถานะ )
```

โดย สถานะ คือ 0 ถ้าต้องการปิดการบล็อก และ 1 คือเปิดการบล็อก

ผลลัพธ์ คือ 0 ถ้าการเปลี่ยนโหมดล้มเหลว และ 1 คือการเปลี่ยนโหมดสำเร็จ

### 2.3.5 ตัวอย่างโปรแกรม

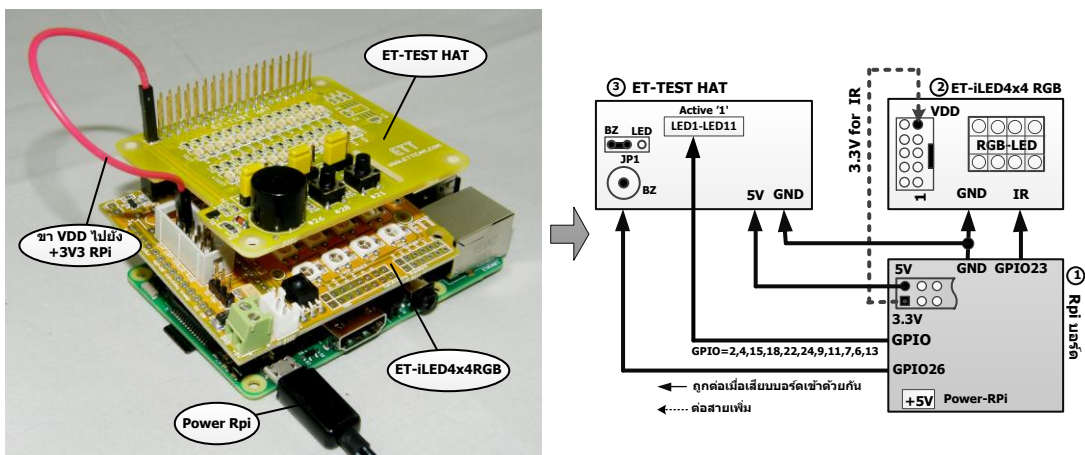
ตัวอย่างโปรแกรมที่ 6-2.3.5-1 เป็นโปรแกรมที่ทำหน้าที่รอข้อมูลจากการกดแป้นบน ET-IR REMOTE KEY เพื่อแสดงชื่อแป้นที่กดทางจอภาพ และจะออกจากโปรแกรมเมื่อแป้น KEY\_POWER ถูกกด

โปรแกรมที่ 6-2.3.5-1 อ่านเป็นจาก ET-IR REMOTE KEY

บรรทัด	โค้ด
1	import time
2	import sys
3	import pylirc
4	<b>def setup():</b>
5	pylirc.init("ett", "./ett_ir_conf", 0)
6	<b>def clean():</b>
7	pylirc.exit()
8	<b>def loop():</b>
9	ir_codes = pylirc.nextcode()
10	if ir_codes != None:
11	for i in ir_codes:
12	print(ir_codes[0])
13	if (ir_codes[0] == 'Power'):
14	clean()
15	sys.exit()
16	time.sleep(0.1)
17	print("ET-IE REMOTE KEY (6-2.3.5-1)")
18	print("กดแป้น Ctrl-C เพื่อออกจากโปรแกรม หรือ กด [Power]")
19	setup()
20	try:
21	while True:
22	loop()
23	except KeyboardInterrupt:
24	clean()

### 3. ตัวอย่างเลือกรูปแบบไฟรั้งด้วยรีโมท

ตัวอย่างเลือกรูปแบบไฟรั้งด้วยรีโมทอาศัยการเชื่อมต่อดังภาพที่ 6-3-1 คือ ใช้บอร์ด ET-iLED 4x4 RGB เป็นบอร์ดสำหรับรับข้อมูลจาก ET-IR REMOTE KEY โดยต่อสายขาที่ 1 ซึ่งเป็น +3V3 บนบอร์ด ET-TEST HAT ไปยังขาที่ 10 ของขั้วต่อ 10 ขาของบอร์ด ET-iLED 4x4 RGB หรือ ใช้การ Jump จุด JP-VDD ของบอร์ด ET-iLED 4x4 RGB ดังรูป 6-3.1-1 ก็ได้เพื่อใช้เป็นแหล่งจ่ายไฟสำหรับวงจรภาครับ IR และใช้ขา GPIO ที่ 2, 4, 15, 18, 22, 24, 9, 11, 7, 6, 13 และ 16 เป็นขาเชื่อมต่อกับกับแอลอีดีเพื่อใช้เป็นไฟรั้ง



ภาพที่ 6-3-1 การเชื่อมต่อบอร์ด ET-iLED4x4 RGB กับ ET-TEST HAT

ตัวอย่างโปรแกรมที่ 6-3-1 เป็นโปรแกรมที่ทำหน้าที่รอรับข้อมูลจาก ET-IR REMOTE KEY โดยมีเงื่อนไขของการกดแป้นดังนี้

1. KEY\_POWER ออกจากโปรแกรม
2. KEY\_1 ไฟรั้งจากซ้ายไปขวา
3. KEY\_2 ไฟรั้งจากขวาไปซ้าย

#### หมายเหตุ

ต้องระวังขา GPIO หมายเลข 23 เนื่องจากเป็นขาที่เชื่อมต่อกับภาครับ IR ถ้าไปเปิดให้ GPIO ขา 23 ทำงานกับแอลอีดีจะทำให้ภาครับ IR ไม่ทำงาน

โปรแกรมที่ 6-3-1 เลือกรูปแบบไฟรั้งด้วย ET-IR REMOTE KEY

บรรทัด	โค้ด
1	import time, sys, pylirc
2	import RPi.GPIO as GPIO
3	leds_top = [2,4,15,18,22,24,9,11,7,6,13,16]
4	<b>def setup():</b>
5	pylirc.init("ett", "./ett_ir_conf", 0)
6	GPIO.setmode(GPIO.BCM)
7	GPIO.setwarnings(False)
8	for i in leds_top:
9	GPIO.setup( i, GPIO.OUT )
10	GPIO.output( i, False )
11	<b>def clean():</b>
12	pylirc.exit()
13	GPIO.cleanup()
14	<b>def move_left():</b>
15	for i in range(0, len(leds_top)):
16	GPIO.output( leds_top[i], True )
17	time.sleep(0.5)
18	GPIO.output( leds_top[i], False )
19	time.sleep(0.1)
20	<b>def move_right():</b>
21	for i in range(-1, -1*(len(leds_top)+1), -1):
22	GPIO.output( leds_top[i], True )
23	time.sleep(0.5)
24	GPIO.output( leds_top[i], False )
25	time.sleep(0.1)
26	<b>def loop():</b>
27	ir_codes = pylirc.nextcode()
28	if ir_codes != None:
29	for i in ir_codes:
30	print(ir_codes[0])
31	if (ir_codes[0] == 'Power'):
32	clean()
33	sys.exit()
34	if (ir_codes[0] == '1'):
35	move_left()

```

36         if (ir_codes[0] == '2'):
37             move_right()
38         time.sleep(0.1)
39     print("ET-IE REMOTE KEY + LEDs (6-3-1)")
40     print("กดแป้น Ctrl-C เพื่อออกจากโปรแกรม หรือ กด [Power]")
41     setup()
42     try:
43         while True:
44             loop()
45     except KeyboardInterrupt:
46         clean()

```

#### 4. ตัวอย่างเล่นเสียงดนตรีด้วยรีโมท

ตัวอย่างเล่นเสียงดนตรีด้วยรีโมทอาศัยการเชื่อมต่อดังภาพที่ 6-4-1 คือ ใช้บอร์ด ET-iLED 4x4 RGB เป็นบอร์ดสำหรับรับข้อมูลจาก ET-IR REMOTE KEY โดยต่อสายขาที่ 1 ซึ่งเป็น +3V3 บนบอร์ด ET-TEST HAT ไปยังขาที่ 10 ของขั้วต่อ 10 ขาของบอร์ด ET-iLED 4x4 RGB หรือ ใช้การ Jump จุด JP-VDD ของบอร์ด ET-iLED 4x4 RGB ดังรูป 6-3.1-1 ก็ได้ เพื่อใช้เป็นแหล่งจ่ายไฟสำหรับวงจรภาครับ IR และใช้ขา GPIO ที่ 26 เป็นขาเชื่อมต่อกับลำโพง (ตรวจสอบขั้วต่อบนบอร์ด ET-TEST HAT ให้ถูกต้องก่อนทดลอง)

ตัวอย่างโปรแกรมที่ 6-4-1 เป็นโปรแกรมที่ทำหน้าที่รอรับข้อมูลจาก ET-IR REMOTE KEY โดยมีเงื่อนไขของการกดแป้นดังนี้

1. KEY\_POWER ออกจากโปรแกรม
2. KEY\_1,KEY\_2, ..., KEY\_7 เล่นตัวโน้ต โด, เร, มี, ... และ ที

โปรแกรมที่ 6-4-1 เล่นเสียงดนตรีด้วย ET-IR REMOTE KEY

บรรทัด	โค้ด
1	import time
2	import sys
3	import pylirc
4	import RPi.GPIO as GPIO
5	notes = [0.0001908, 0.00017,0.0001515,0.0001432,0.0001275,0.0001136,0.0001012]
6	freq = [524,588,660,698,784,880,988]
7	<b>def setup():</b>
8	pylirc.init("ett", "./ett_ir_conf",0)
9	GPIO.setmode(GPIO.BCM)
10	GPIO.setwarnings(False)
11	GPIO.setup( 26, GPIO.OUT )
12	<b>def clean():</b>
13	pylirc.exit()
14	GPIO.cleanup()
15	<b>def play(k):</b>
16	for i in range(0, freq[k]):
17	GPIO.output( 26, True )
18	time.sleep( notes[k] )
19	GPIO.output( 26, False )
20	time.sleep( notes[k] )

```

21  def loop():
22      ir_codes = pylirc.nextcode()
23      if ir_codes != None:
24          for i in ir_codes:
25              print(ir_codes[0])
26              if (ir_codes[0] == 'Power'):
27                  clean()
28                  sys.exit()
29              if ((ir_codes[0] >= '1') and (ir_codes[0] <= '7')):
30                  k = int(ir_codes[0])-1
31                  play(k)
32          time.sleep(0.1)
33  print("ET-IE REMOTE KEY + Play Note (6-4-1)")
34  print("กดแป้น Ctrl-C เพื่อออกจากโปรแกรม หรือ กด [Power]")
35  setup()
36  try:
37      while True:
38          loop()
39  except KeyboardInterrupt:
40      clean()

```

## 5. สรุป

การใช้ ET-IR REMOTE KEY จะต้องตั้งค่าการทำงานเพื่อสร้างแฟ้มเก็บรหัสแป้นก่อนใช้งานเสมอ ซึ่งต้องอาศัยความพยายามในการตั้งค่าค่อนข้างสูง แต่อย่างไรก็ดี ทางอีทีที ได้เตรียมไฟล์สำหรับใช้งานไว้ 2 ไฟล์ คือ ett\_ir.conf และ lircd.conf ดังนั้น ทุกครั้งที่พัฒนาโปรแกรมต้องมีไฟล์ทั้งสองอยู่ที่เดียวกับโปรแกรมที่เขียนด้วยเสมอ

นอกจากนี้การใช้งานบอร์ด ET-iLED4x4 RGB ในส่วนของภาครับสัญญาณอินฟราเรดจาก ET-IR REMOTE KEY จะต้องเชื่อมต่อสายไฟจากขา +3V3 ไปที่ขา VDD ของขั้วต่อ 10 ขา ด้วยเสมอ หรือใช้การ Short Jumper JP-VDD ก็ได้