

第 4 篇

附录

附录 A Python 程序代码

附录 B HDMI 显示模式



附录 **A**

Python 程序代码

下面的代码清单提供了第 11 章的例 3 和例 4，以及第 12 章联合输入输出程序的解决方案代码。这些代码也可以通过树莓派的用户手册网站下载 <http://www.wiley.com/go/raspberrypiuserguide>。如果你手动输入代码，注意↵符号，它表示上下两行在代码中为一行。如果你在行末遇到这个标记，不要按回车换行，直到某行末没有这个符号。

树莓贪吃蛇（第 11 章，例 3）

```
#!/usr/bin/env python
# Raspberry Snake
# Written by Gareth Halfacree for the Raspberry Pi User Guide

import pygame, sys, time, random
from pygame.locals import *

pygame.init()
fpsClock = pygame.time.Clock()

playSurface = pygame.display.set_mode((640, 480))
pygame.display.set_caption('Raspberry Snake')

redColour = pygame.Color(255, 0, 0)
blackColour = pygame.Color(0, 0, 0)
whiteColour = pygame.Color(255, 255, 255)
greyColour = pygame.Color(150, 150, 150)
snakePosition = [100,100]
snakeSegments = [[100,100],[80,100],[60,100]]
raspberryPosition = [300,300]
raspberrySpawned = 1
direction = 'right'
changeDirection = direction

def gameOver():
    gameOverFont = pygame.font.Font('freesansbold.ttf', 72)
    gameOverSurf = gameOverFont.render(
        ('Game Over', True, greyColour)
    )
    gameOverRect = gameOverSurf.get_rect()
    gameOverRect.midtop = (320, 10)
    playSurface.blit(gameOverSurf, gameOverRect)
    pygame.display.flip()
    time.sleep(5)
    pygame.quit()
    sys.exit()

while True:
```

```

for event in pygame.event.get():
    if event.type == QUIT:
        pygame.quit()
    elif event.type == KEYDOWN:
        if event.key == K_RIGHT or event.key == ord('d'):
            changeDirection = 'right'
        if event.key == K_LEFT or event.key == ord('a'):
            changeDirection = 'left'
        if event.key == K_UP or event.key == ord('w'):
            changeDirection = 'up'
        if event.key == K_DOWN or event.key == ord('s'):
            changeDirection = 'down'
        if event.key == K_ESCAPE:
            pygame.event.post(pygame.event.Event(QUIT))
    if changeDirection == 'right' and not direction == 'left':
        direction = changeDirection
    if changeDirection == 'left' and not direction == 'right':
        direction = changeDirection
    if changeDirection == 'up' and not direction == 'down':
        direction = changeDirection
    if changeDirection == 'down' and not direction == 'up':
        direction = changeDirection
    if direction == 'right':
        snakePosition[0] += 20
    if direction == 'left':
        snakePosition[0] -= 20
    if direction == 'up':
        snakePosition[1] -= 20
    if direction == 'down':
        snakePosition[1] += 20
    snakeSegments.insert(0, list(snakePosition))
    if snakePosition[0] == raspberryPosition[0] and
snakePosition[1] == raspberryPosition[1]:
        raspberrySpawned = 0
    else:
        snakeSegments.pop()
    if raspberrySpawned == 0:
        x = random.randrange(1, 32)
        y = random.randrange(1, 24)
        raspberryPosition = [int(x*20), int(y*20)]
        raspberrySpawned = 1
    playSurface.fill(blackColour)
    for position in snakeSegments:
        pygame.draw.rect(playSurface, whiteColour, Rect
(position[0], position[1], 20, 20))

pygame.draw.rect(playSurface, redColour, Rect
(raspberryPosition[0], raspberryPosition[1], 20, 20))

```

```

pygame.display.flip()
if snakePosition[0] > 620 or snakePosition[0] < 0:
    gameOver()
if snakePosition[1] > 460 or snakePosition[1] < 0:
    for snakeBody in snakeSegments[1:]:
        if snakePosition[0] == snakeBody[0] and
           snakePosition[1] == snakeBody[1]:
            gameOver()

fpsClock.tick(30)

```

IRC 用户列表（第 11 章，例 4）

```

#!/usr/bin/env python
# IRC User List
# Written by Tom Hudson for the Raspberry Pi User Guide
# http://tomhudson.co.uk/

import sys, socket, time

RPL_NAMREPLY = '353'
RPL_ENDOFNAMES = '366'

irc = {
    'host': 'chat.freenode.net',
    'port': 6667,
    'channel': '#raspiuserguide',
    'namesinterval': 5
}

user = {
    'nick': 'botnick',
    'username': 'botuser',
    'hostname': 'localhost',
    'servername': 'localhost',
    'realname': 'Raspberry Pi Names Bot'
}

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print 'Connecting to %(host)s:%(port)s... ' % irc
s.connect((irc['host'], irc['port']))
except socket.error:
    print 'Error connecting to IRC server'
    %(host)s:%(port)s' % irc
    sys.exit(1)

s.send('NICK %(nick)s\r\n' % user)

```

```

s.send('USER %(username)s %(hostname)s %(servername)s : \n'
      %(realname)s\r\n' % user)
s.send('JOIN %(channel)s\r\n' % irc)
s.send('NAMES %(channel)s\r\n' % irc)

read_buffer = ''
names = []

while True:
    read_buffer += s.recv(1024)
    lines = read_buffer.split('\r\n')
    read_buffer = lines.pop();
    for line in lines:
        response = line.rstrip().split(' ', 3)
        response_code = response[1]
        if response_code == RPL_NAMREPLY:
            names_list = response[3].split(': ')[1]
            names += names_list.split(' ')
        if response_code == RPL_ENDOFNAMES:
            print '\r\nUsers in %(channel)s: ' % irc
            for name in names:
                print name
            names = []
            time.sleep(irc['namesinterval'])
            s.send('NAMES %(channel)s\r\n' % irc)

```

GPIO 输入输出（第 12 章）

```

#!/usr/bin/env python
# Raspberry Pi GPIO Input/Output example
# Written by Gareth Halfacree for the Raspberry Pi User Guide

import RPi.GPIO as GPIO

GPIO.setup(11, GPIO.OUT)
GPIO.setup(12, GPIO.IN)
GPIO.output(11, False)

while True:
    input_value = GPIO.input(12)
    if input_value == False:
        print "The button has been pressed. Lighting LED. "
        GPIO.output(11, True)
        while input_value == False:
            input_value = GPIO.input(12)
        print "The button has been released. Extinguishing LED. "
    if input_value == True:
        GPIO.output(11, False)

```

附录 B

HDMI 显示模式

你可以参考表 B-1 和表 B-2 的数据来配置 HDMI 视频程序 config.txt 中的 hdmi_mode 参数。详情请参考本书第 6 章。

表 B-1 HDMI 组 1 (CEA)

Value	Description
1	VGA (640x480)
2	480p 60Hz
3	480p 60Hz(16:9 aspectratio)
4	720p 60Hz
5	1080i 60Hz
6	480i 60Hz
7	480i 60Hz(16:9aspect ratio)
8	240p 60Hz
9	240p 60Hz(16:9 aspect ratio)
10	480i 60Hz(Pixel quadrupling enabled)
11	480i 60Hz(Pixel quadrupling enabled)(16:9 aspect ratio)
12	240p 60Hz(Pixel quadrupling enabled)
13	240p 60Hz (Pixel quadrupling enabled) (16:9 aspect ratio)
14	480p 60Hz(Pixel doubling enabled)
15	480p 60Hz(Pixel doubling enabled)(16:9 aspect ratio)
16	1080p 60Hz
17	576p 50Hz
18	576p 50Hz(16:9 aspect ratio)
19	720p 50Hz
20	1080i 50Hz
21	576i 50Hz
22	576i 50Hz(16:9 aspect ratio)
23	288p 50Hz
24	288p 50Hz(16:9 aspect ratio)
25	576i 50Hz(Pixel quadrupling enabled)
26	576i 50Hz(Pixel quadrupling enabled)(16:9 aspect ratio)
27	288p 50Hz(Pixel quadrupling enabled)
28	288p 50Hz(Pixel quadrupling enabled)(16:9 aspect ratio)

续表

Value	Description
29	576p 50Hz(Pixel doubling enabled)
30	576p 50Hz(Pixel doubling enabled)(16:9 aspect ratio)
31	1080p 50Hz
32	1080p 24Hz
33	1080p 25Hz
34	1080p 30Hz
35	480p 60Hz(Pixel quadrupling enabled)
36	480p 60Hz(Pixel quadrupling enabled)(16:9 aspect ratio)
37	576p 50Hz(Pixel quadrupling enabled)
38	576p 50Hz(Pixel quadrupling enabled)(16:9 aspect ratio)
39	1080i 50Hz(Reduced blanking)
40	1080i 100Hz
41	720p 100Hz
42	576p 100Hz
43	576p 100Hz(16:9 aspect ratio)
44	576i 100Hz
45	576i 100Hz(16:9 aspect ratio)
46	1080i 120Hz
47	720p 120Hz
48	480p 120Hz
49	480p 120Hz(16:9 aspect ratio)
50	480i 120Hz
51	480i 120Hz(16:9 aspect ratio)
52	576p 200Hz
53	576p 200Hz(16:9 aspect ratio)
54	576i 200Hz
55	576i 200Hz(16:9 aspect ratio)
56	480p 24Hz0
57	480p 24Hz0(16:9 aspect ratio)
58	480i 240Hz
59	480i 240Hz(16:9 aspect ratio)

表 B-2 HDMI 组 2 (DMT)

Value	Description
1	640×350 85Hz
2	640×400 85Hz
3	720×400 85Hz
4	640×480 60Hz
5	640×480 72Hz
6	640×480 75Hz
7	640×480 85Hz
8	800×600 56Hz
9	800×600 60Hz
10	800×600 72Hz
11	800×600 75Hz
12	800×600 85Hz
13	800×600 120Hz
14	848×480 60Hz
15	1024×768 43Hz, incompatible with the Raspberry Pi
16	1024×768 60Hz
17	1024×768 70Hz
18	1024×768 75Hz
19	1024×768 85Hz
20	1024×768 120Hz
21	1152×864 75Hz
22	1280×768(Reduced blanking)
23	1280×768 60Hz
24	1280×768 75Hz
25	1280×768 85Hz
26	1280×768 120Hz(Reduced blanking)
27	1280×800(Reduced blanking)
28	1280×800 60Hz

续表

Value	Description
29	1280×800 75Hz
30	1280×800 85Hz
31	1280×800 120Hz(Reduced blanking)
32	1280×960 60Hz
33	1280×960 85Hz
34	1280×960 120Hz(Reduced blanking)
35	1280×1024 60Hz
36	1280×1024 75Hz
37	1280×1024 85Hz
38	1280×1024 120Hz(Reduced blanking)
39	1360×768 60Hz
40	1360×768 120Hz(Reduced blanking)
41	1400×1050(Reduced blanking)
42	1400×1050 60Hz
43	1400×1050 75Hz
44	1400×1050 85Hz
45	1400×1050 120Hz(Reduced blanking)
46	1440×900(Reduced blanking)
47	1440×900 60Hz
48	1440×900 75Hz
49	1440×900 85Hz
50	1440×900 120Hz(Reduced blanking)
51	1600×1200 60Hz
52	1600×1200 65Hz
53	1600×1200 70Hz
54	1600×1200 75Hz
55	1600×1200 85Hz
56	1600×1200 120Hz(Reduced blanking)

续表

Value	Description
57	1680×1050(Reduced blanking)
58	1680×1050 60Hz
59	1680×1050 75Hz
60	1680×1050 85Hz
61	1680×1050 120Hz(Reduced blanking)
62	1792×1344 60Hz
63	1792×1344 75Hz
64	1792×1344 120Hz(Reduced blanking)
65	1856×1392 60Hz
66	1856×1392 75Hz
67	1856×1392 120Hz(Reduced blanking)
68	1920×1200(Reduced blanking)
69	1920×1200 60Hz
70	1920×1200 75Hz
71	1920×1200 85Hz
72	1920×1200 120Hz(Reduced blanking)
73	1920×1440 60Hz
74	1920×1440 75Hz
75	1920×1440 120Hz(Reduced blanking)
76	2560×1600(Reduced blanking)
77	2560×1600 60Hz
78	2560×1600 75Hz
79	2560×1600 85Hz
80	2560×1600 120Hz(Reduced blanking)
81	1366×768 60Hz
82	1920×1080(1080p) 60Hz
83	1600×900(Reduced blanking)
84	2048×1152(Reduced blanking)
85	1280×720(720p) 60Hz
86	1366×768(Reduced blanking)