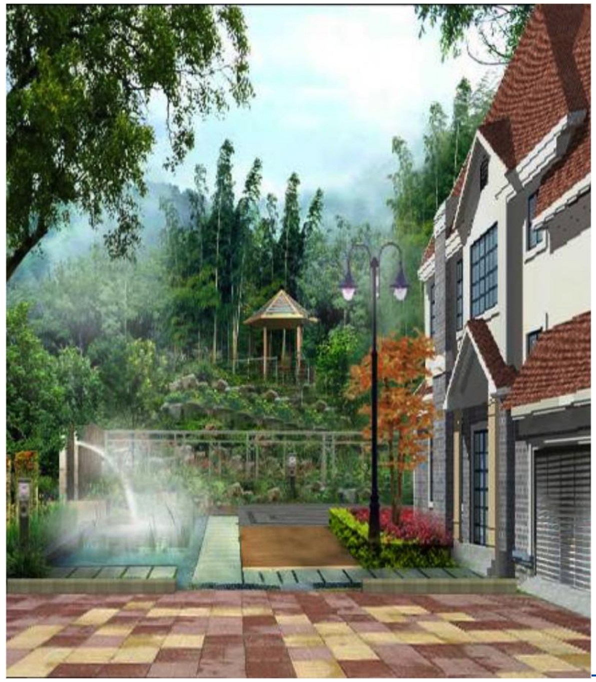
Make:
Sensors
Projects and Experiments to Measure the World with Arduino and Raspberry Pi
Tero Karvinen, Kimmo Karvinen & Ville Valtokari

Movement

Movement

- As you walk to your yard, a light comes on automatically.
- Automatic garden lights detect the moving heat radiated by humans. Joysticks convert motion to a change in resistance. A volume knob is a potentiometer, another kind of variable resistor.



Contents

Experiment: Which Way Is Up? (Tilt Ball Switch)

- If you decide to build a pinball machine, you might want to detect excessive nudging and end the player’s turn with a TILT alarm. Also, a burglar alarm could use a tilt sensor.

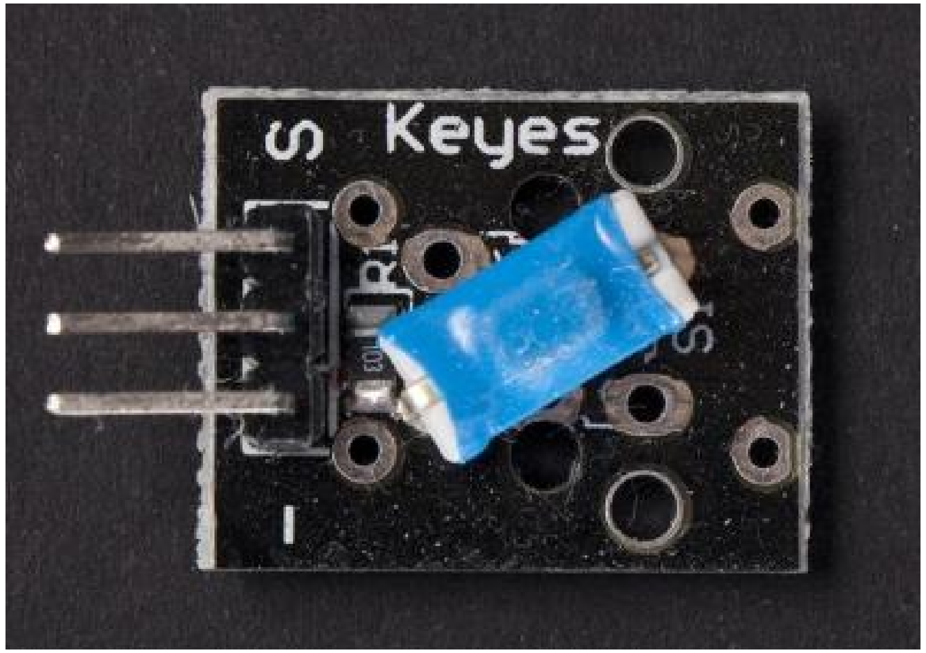


Figure 6-1. Tilt ball switch

Tilt Sensor Code and Connection for Arduino

- Figure 6-2 shows the circuit diagram for an Arduino-based tilt sensor. Build it, and then run the code shown in Example 6-1.

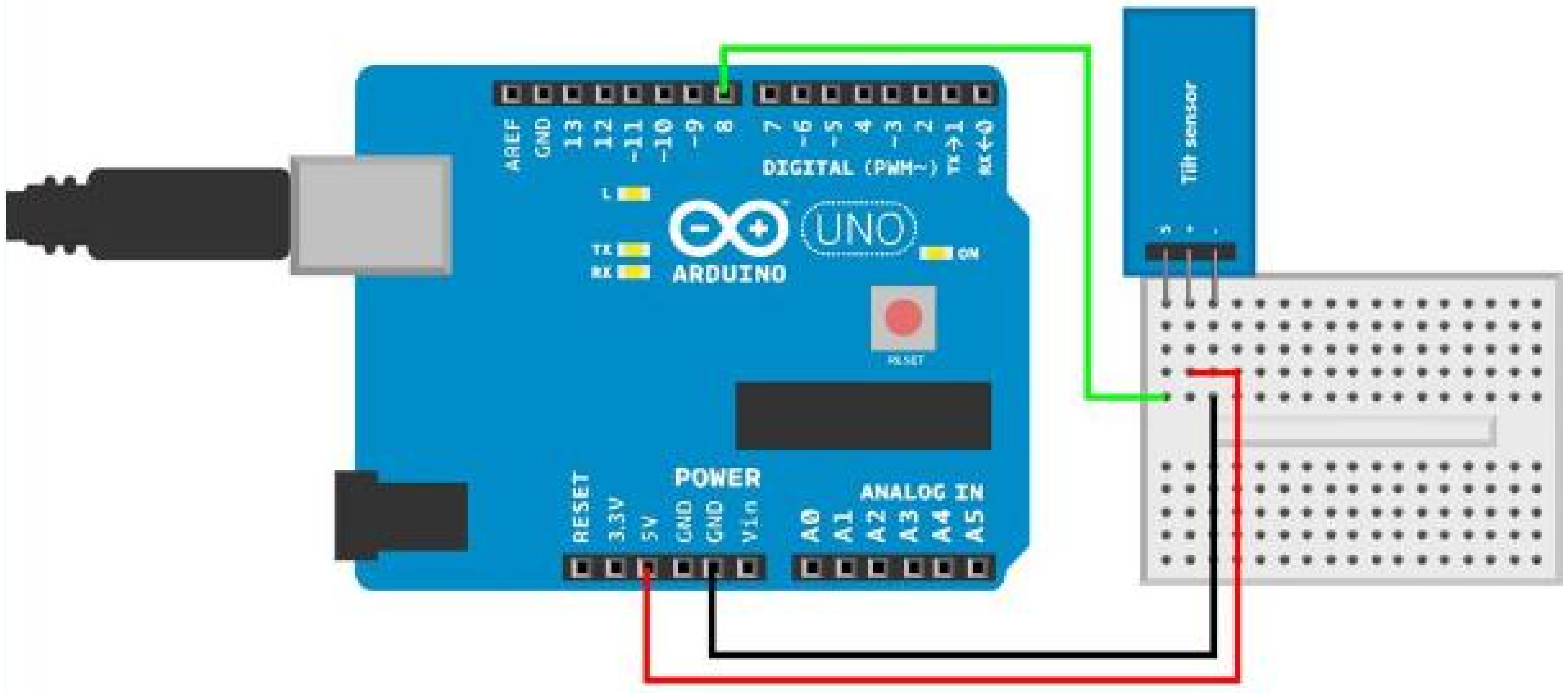


Figure 6-2. Tilt sensor circuit for Arduino

Example 6-1. tilt_sensor.ino

```
// tilt_sensor.ino - detect tilting and print to serial
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

const int tiltPin = 8;
int tilted = -1;

void setup() {
    Serial.begin(115200);
    pinMode(tiltPin, INPUT);
    digitalWrite(tiltPin, HIGH);
}

void loop() {
    tilted = digitalRead(tiltPin);        //❶
    if(tilted == 0) {
        Serial.println("Sensor is tilted");
    } else {
        Serial.println("Sensor is not tilted");
    }
    delay(100);
}
```

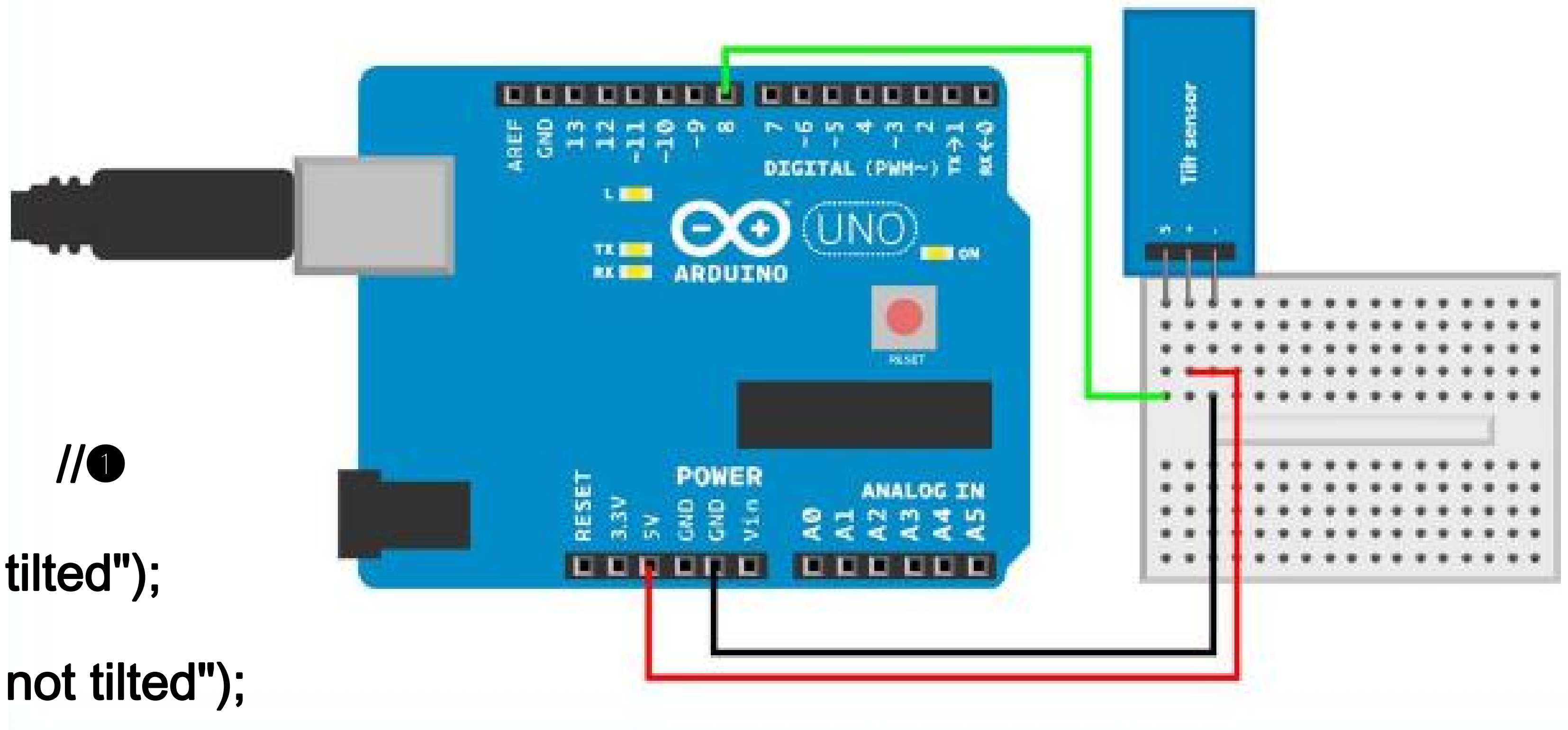
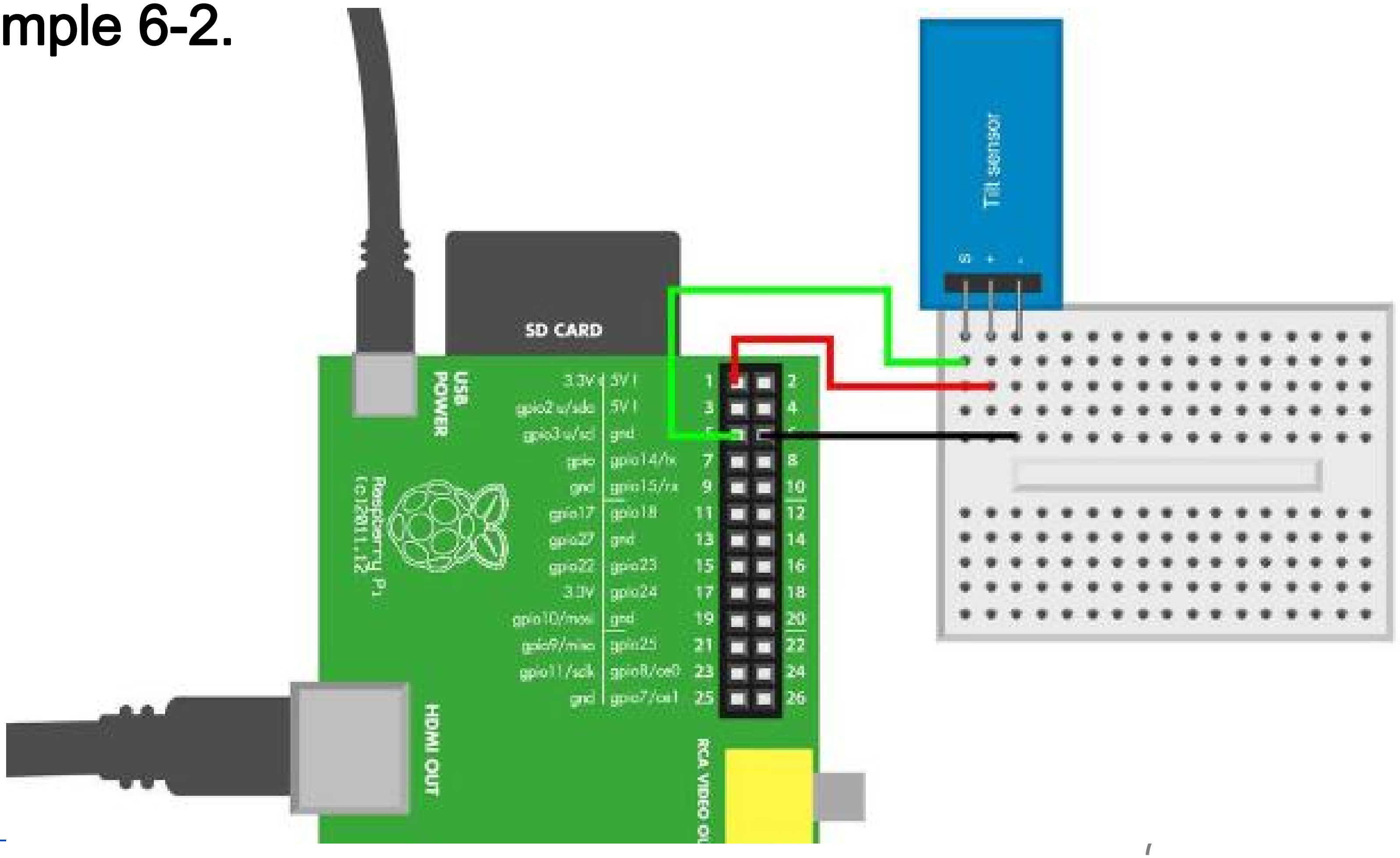


Figure 6-2. Tilt sensor circuit for Arduino

Tilt Sensor Code and Connection for Raspberry Pi

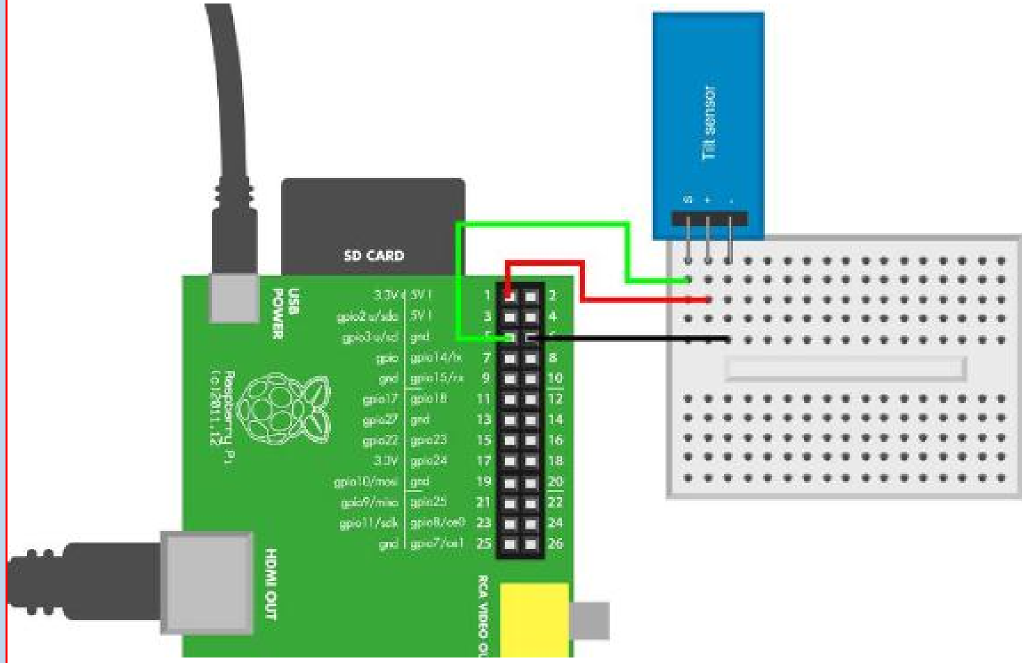
- Figure 6-3 shows the connections for Raspberry Pi. Wire it up and run the program shown in Example 6-2.



Tilt Sensor Code and Connection for Raspberry Pi

Example 6-2. tilt_sensor.py

```
# tilt_sensor.py - print if sensor was tilted
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

import time
import botbook_gpio as gpio
def main():
        tiltpin = 3          # has internal pull-up     # ❶
        gpio.mode(tiltpin, "in")
        while (True):
                isNotTilted = gpio.read(tiltpin)      # ❷
                if(isNotTilted == gpio.LOW):
                        print "Sensor is tilted"
                else:
                        print "Sensor is not tilted“
                time.sleep(0.3) # seconds
if __name__ == "__main__":
        main()
```



Contents

Experiment: Good Vibes with Interrupt (Digital Vibration Sensor)

- A vibration sensor can detect tiny vibrations, like the ground shaking (Figure 6-4).

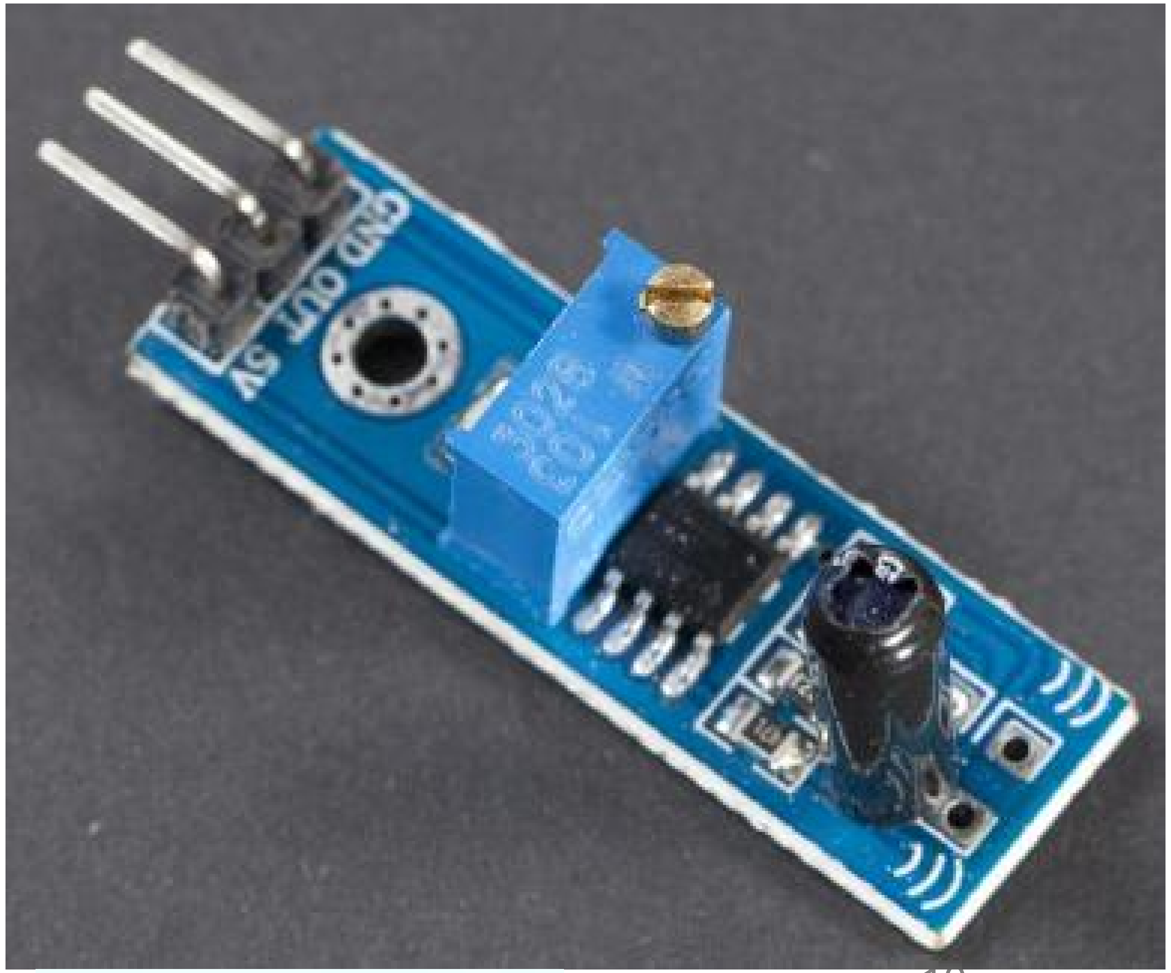


Figure 6-4. Vibration sensor

Vibration Code and Connection for Arduino

- The signal sent by a vibration sensor is very short. You must use an interrupt to catch it.
- For most sensors, you could just poll them. Polling means that you check the state of the digital pin, wait a while, and then check again.
- With an interrupt, you can tell Arduino to run a function whenever some event happens. Interrupts make it harder to follow the flow of the code you write, but they allow you to catch very short-lived events, like a pin going up and then down quickly.

Vibration Code and Connection for Arduino

➢ Figure 6-5 shows the circuit diagram for the vibration sensor. Wire it up as shown, and then run the code shown in Example 6-3.

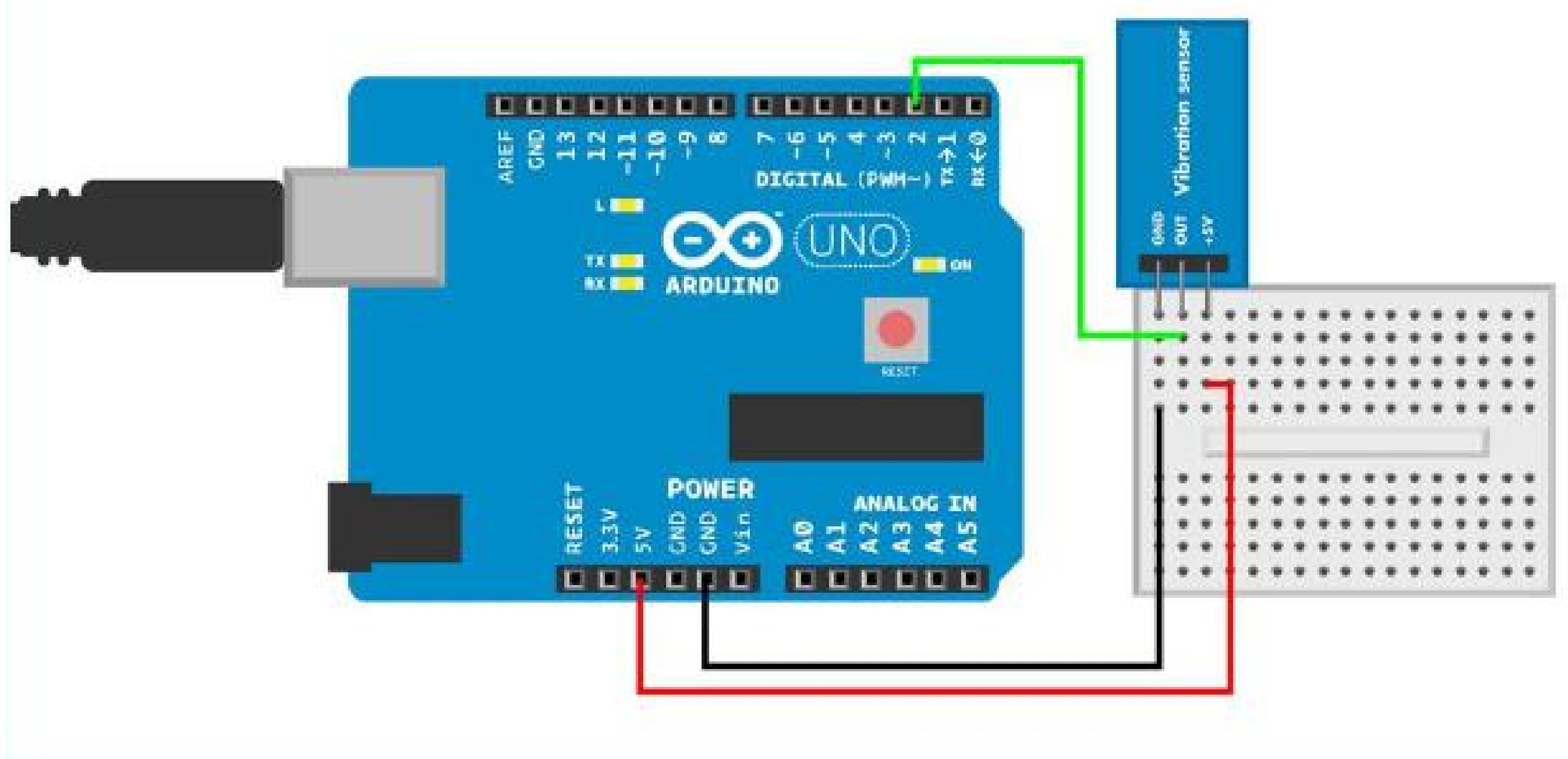


Figure 6-5. Vibration sensor circuit for Arduino

Example 6-3. vibration_sensor.ino

```
// vibration_sensor.ino - detect vibration using interrupt
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

const int sensorPin = 0;          //UNO,Mega pin 2, Leonardo pin 3
volatile int sensorState = -1;
void setup() {
    Serial.begin(115200);
    attachInterrupt(sensorPin, sensorTriggered, RISING);// ❶
}
void loop() {
    if(sensorState == 1) {                              //  ❷
        Serial.println("Vibrations detected");
        delay(1);   // ms
        sensorState = 0;
    }
    delay(10);
}
void sensorTriggered() {                                // ❸
    sensorState = 1;                                    // ❹
}
```

Vibration Code and Connection for Raspberry Pi

- Figure 6-6 shows the wiring diagram for the Raspberry Pi version of this sensor circuit.
- Wire it up as shown and then run the program shown in Example 6-4.

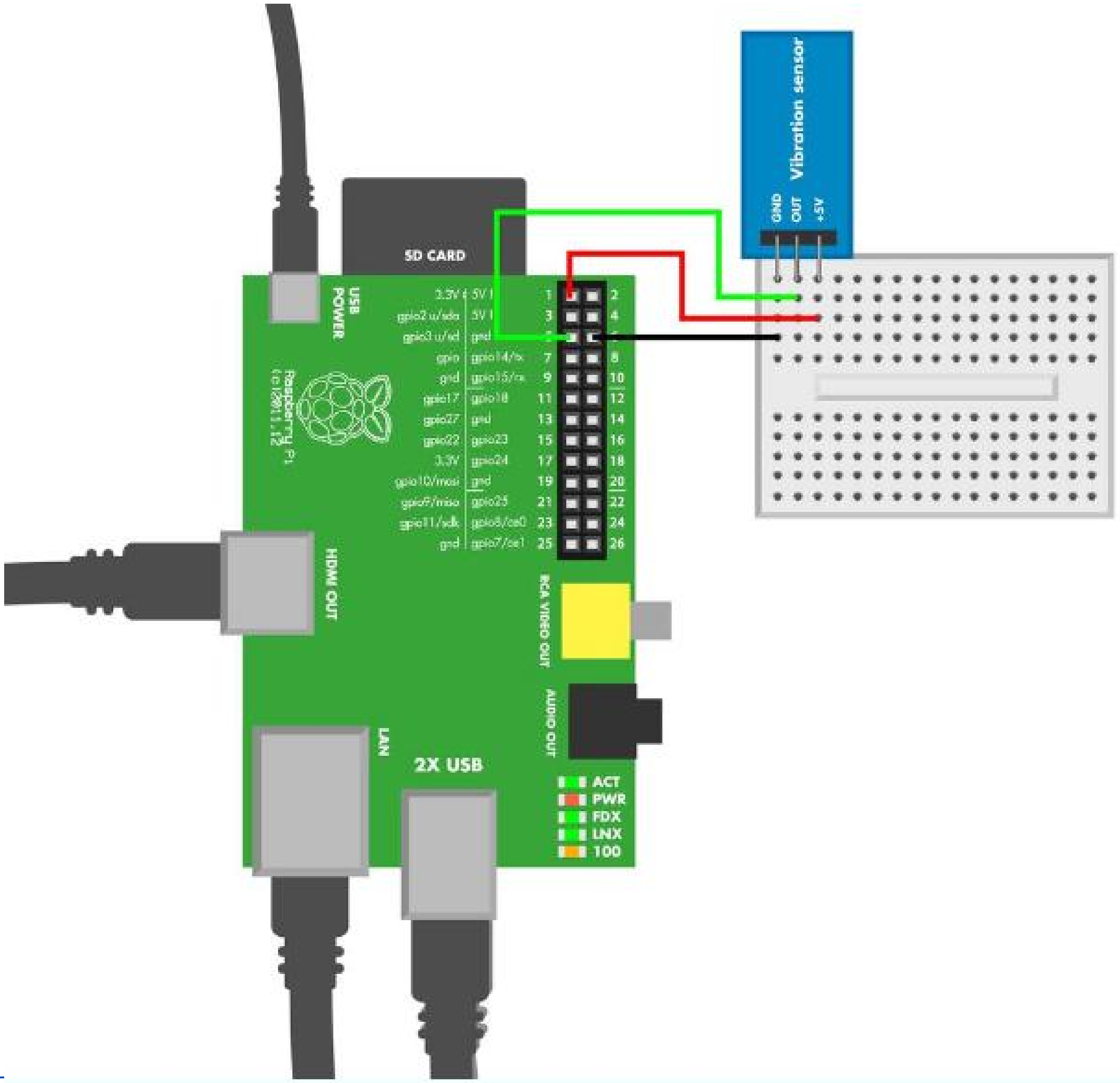


Figure 6-6. Vibration sensor circuit for Raspberry Pi

Example 6-4. vibration_sensor.py

```
# vibration_sensor.py - detect vibration
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

import time
import botbook_gpio as gpio      #❶

def main():
    vibPin = 3
    gpio.mode(vibPin, "in")
    while (True):
        vibrationInput = gpio.read(vibPin)  #❷
        if(vibrationInput == gpio.LOW):
            print "Vibration"
            time.sleep(5)              #❸
        time.sleep(0.01) # seconds     #❹

if __name__ == "__main__":
    main()
```

- Raspberry Pi code can't detect as short vibrations as Arduino can.
- The code doesn't use an interrupt, because it would complicate the code, would still need fast looping, and still couldn't match Arduino.
- If you need very precise vibration detection in Raspberry Pi, you can connect Arduino to Raspberry Pi (Talking to Arduino from Raspberry Pi).

Contents

Experiment: Turn the Knob

- A rotary encoder measures turning (see Figure 6-7).
- An absolute rotary encoder tells the position (position is 83 deg),
- and relative rotary encoders tell the change (turned 23 deg from previous position).

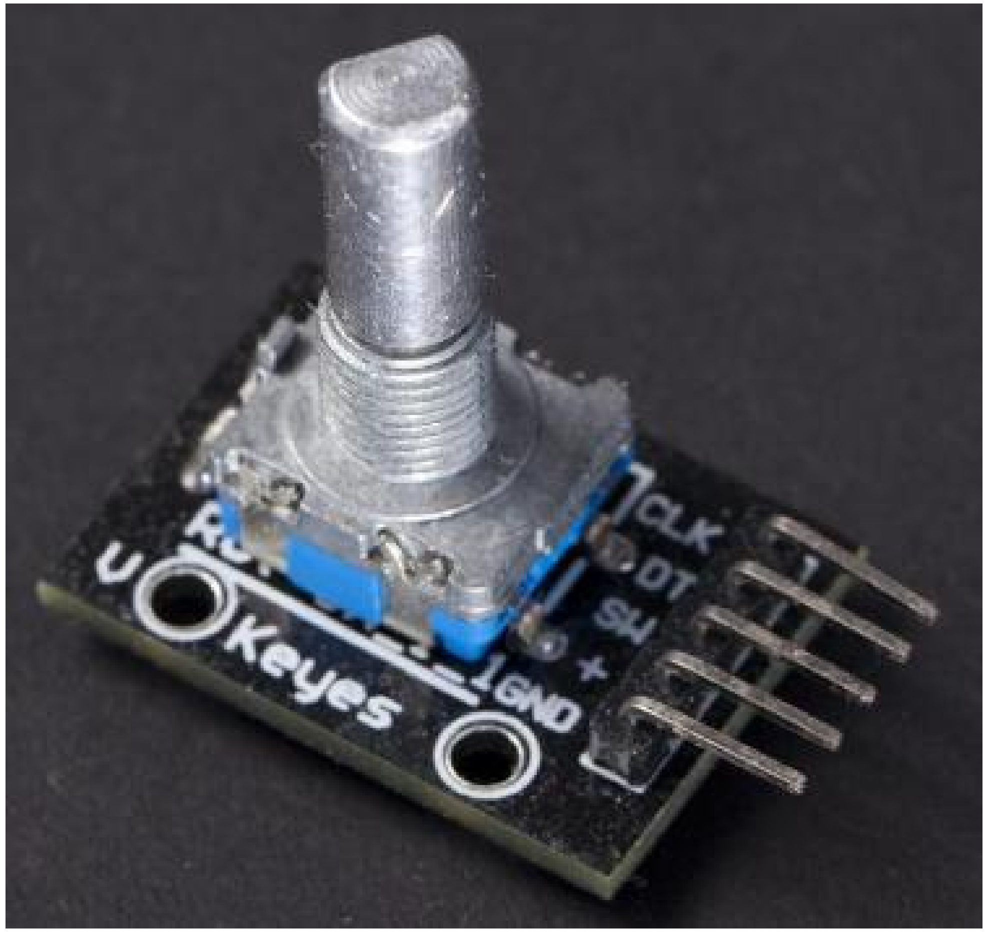
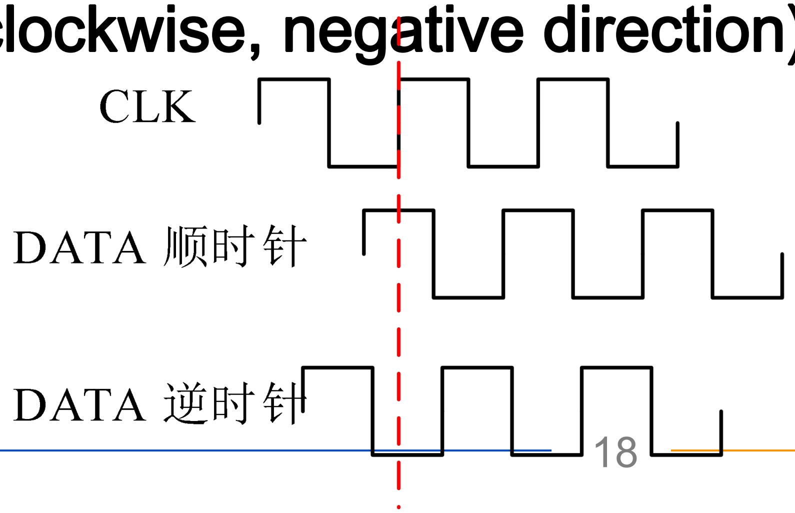


Figure 6-7. Rotary encoder

Experiment: Turn the Knob

- **The rotary encoder we use here is a relative one.**
- **When you turn the knob, the encoder sends clock pulses on the clock pin.**
  - On the rising edge (the clock goes from LOW to HIGH), one data pulse comes on the data pin，another comes on the falling edge (HIGH to LOW transition).
- **If the data pulse is HIGH, you’re turning right (clockwise, negative direction).**
- **If the data pulse is LOW, you’re turning left.**



Rotary Encoder Code and Connection for Arduino

- This code uses interrupts, so it might look different from the Arduino codes you've worked with before. Wire up the circuit as shown in Figure 6-8, and run the sketch shown in Example 6-5.

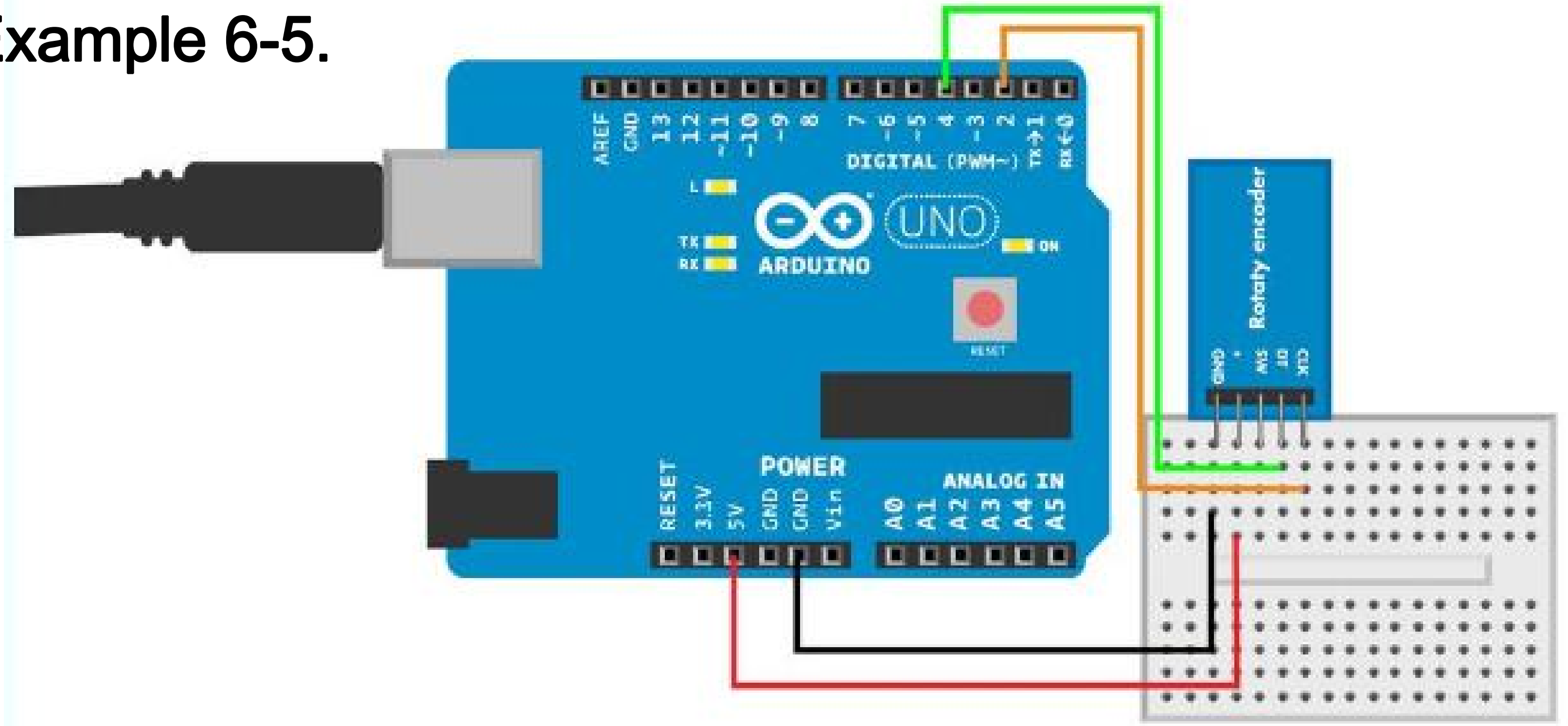
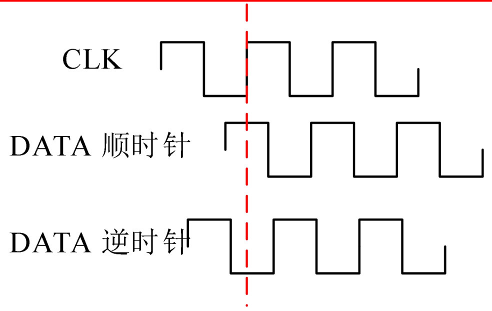


Figure 6-8. Rotary encoder circuit for Arduino

Example 6-5. rotary_encoder.ino

```
// rotary_encoder.ino - print encoder position
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

const int clkPin = 2;
const int dtPin = 4;
volatile unsigned int encoderPos = 0;     // ❶

void setup(){
    Serial.begin(115200);
    pinMode(clkPin, INPUT);
    digitalWrite(clkPin, HIGH); // 设置上拉电阻     ❷
    pinMode(dtPin, INPUT);
    digitalWrite(dtPin, HIGH); // pull up
    attachInterrupt(0, processEncoder, CHANGE);❸
}

void loop(){
        Serial.println(encoderPos);
        delay(100);
}
```

```
void processEncoder()   // {  ❹
    if(digitalRead(clkPin) == digitalRead(dtPin))
    {  ❺
        encoderPos++;      // turning right
    }
    else {
        encoderPos--;
        }
}
```



Rotary Encoder Code and Connection for Raspberry Pi

- Figure 6-9 shows the wiring diagram
  Raspberry Pi.
- Wire it up as shown and then run the

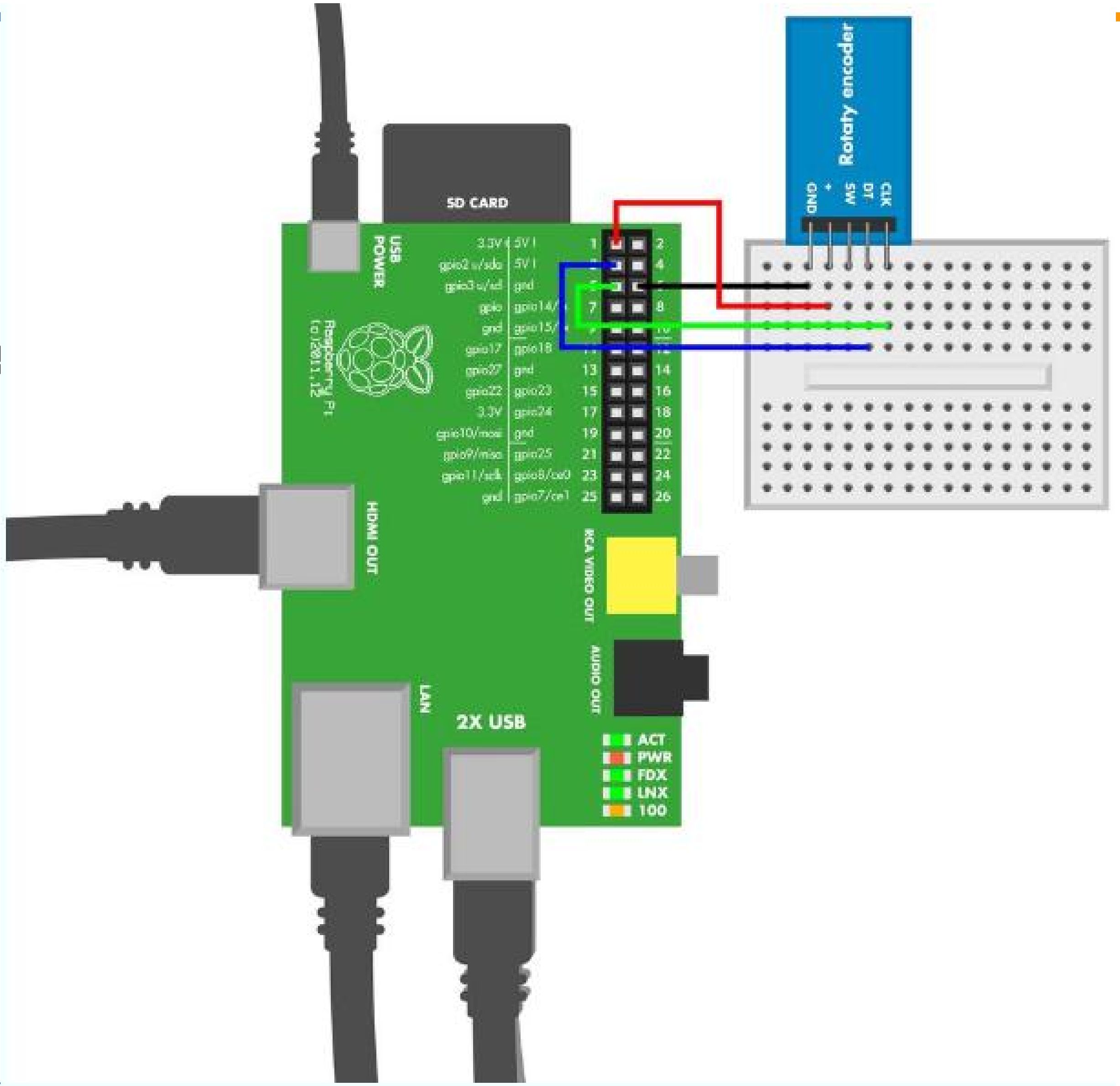
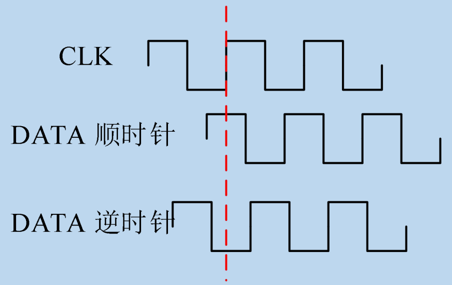


Figure 6-9. Rotary encoder circuit for Raspberry Pi

Example 6-6. rotary_encoder.py

```
# rotary_encoder.py - read rotary encoder
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

import time
import botbook_gpio as gpio     # ❶

def main():
    encoderClk = 3
    encoderDt = 2
    gpio.mode(encoderClk, "in")
    gpio.mode(encoderDt, "in")
    encoderLast = gpio.LOW
    encoderPos = 0
    lastEncoderPos = 0
```

```
while True:
    clk = gpio.read(encoderClk)
    if encoderLast == gpio.LOW and clk == gpio.HIGH:    #❷
        if(gpio.read(encoderDt) == gpio.LOW):           #❸
            encoderPos += 1
        else:
            encoderPos -= 1
    encoderLast = clk
    if encoderPos != lastEncoderPos:
        print("Encoder position %d" % encoderPos)
    lastEncoderPos = encoderPos
    time.sleep(0.001)          # s                      #❹
if __name__ == "__main__":
    main()
```



Contents

Experiment: Thumb Joystick(Analog Two-Axis Thumb Joystick)

- If you've played on any videogame consoles, you have used a joystick. In the old days, you grabbed a big joystick with your strong hand.
- Modern gaming consoles like Xbox, PlayStation, or Ouya use multiple thumb joysticks.

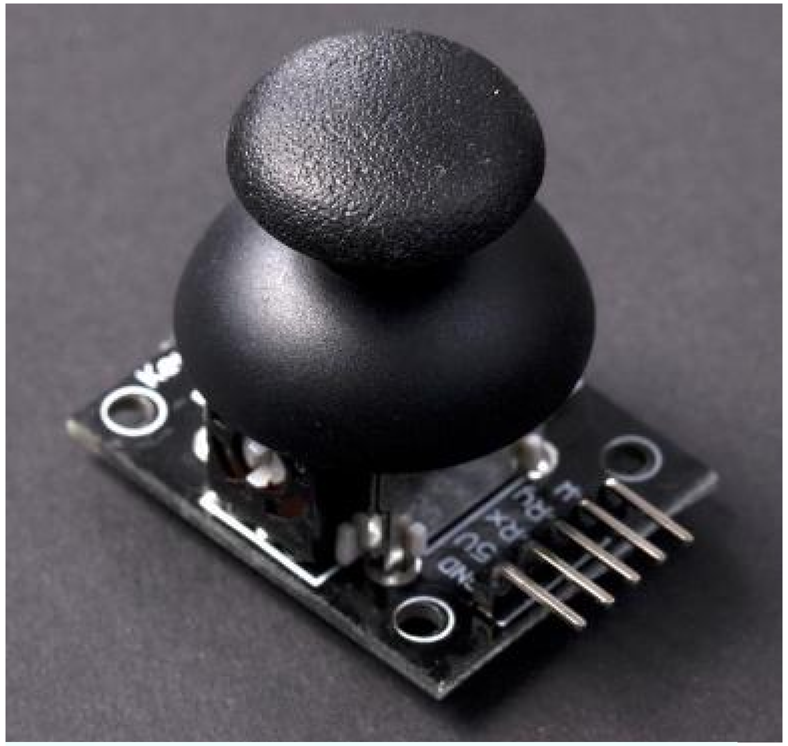
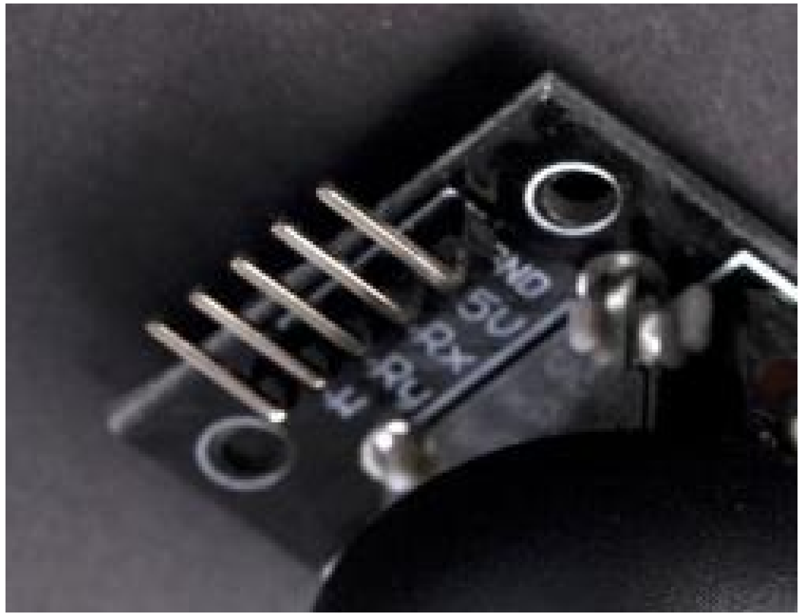


Figure 6-10. Two-axis thumb joystick

Experiment: Thumb Joystick(Analog Two-Axis Thumb Joystick)

- Typically, a joystick has two potentiometers (pots), which are variable resistors.
  - Tilting the joystick along the vertical y-axis changes the resistance of one pot.
  - Tilting along the horizontal x-axis changes the resistance of another pot.
- In many joysticks, the potentiometers are in three-lead configuration.
  - They have one lead on the ground,
  - another on +5 V, and one in the middle, giving a varying voltage between 0 V and +5 V.
  - In this three-lead configuration, no pull-up or pull-down resistors are needed.



Experiment: Thumb Joystick(Analog Two-Axis Thumb Joystick)

- Mobile phones and Wii consoles can use accelerometers instead of joysticks.
- The device attitude is measured against gravity and is used for controlling games. Gravity has the same effect as acceleration. To get precise attitude measurements, see Chapter 8.

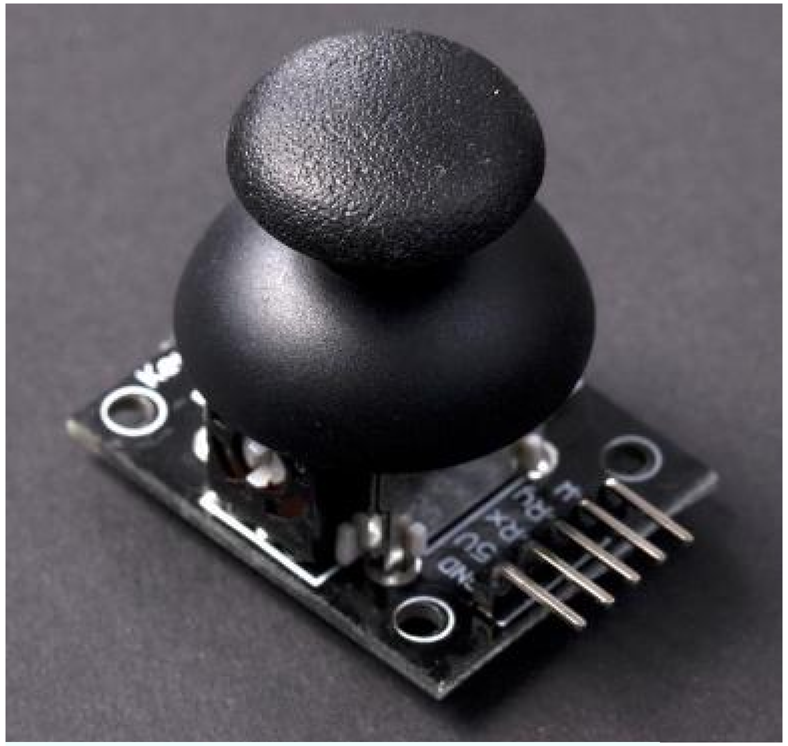


Figure 6-10. Two-axis thumb joystick

Joystick Code and Connection for Arduino

➢ Figure 6-11 shows the circuit design for using Arduino with a joystick. Wire it up and run the sketch shown in Example 6-7.

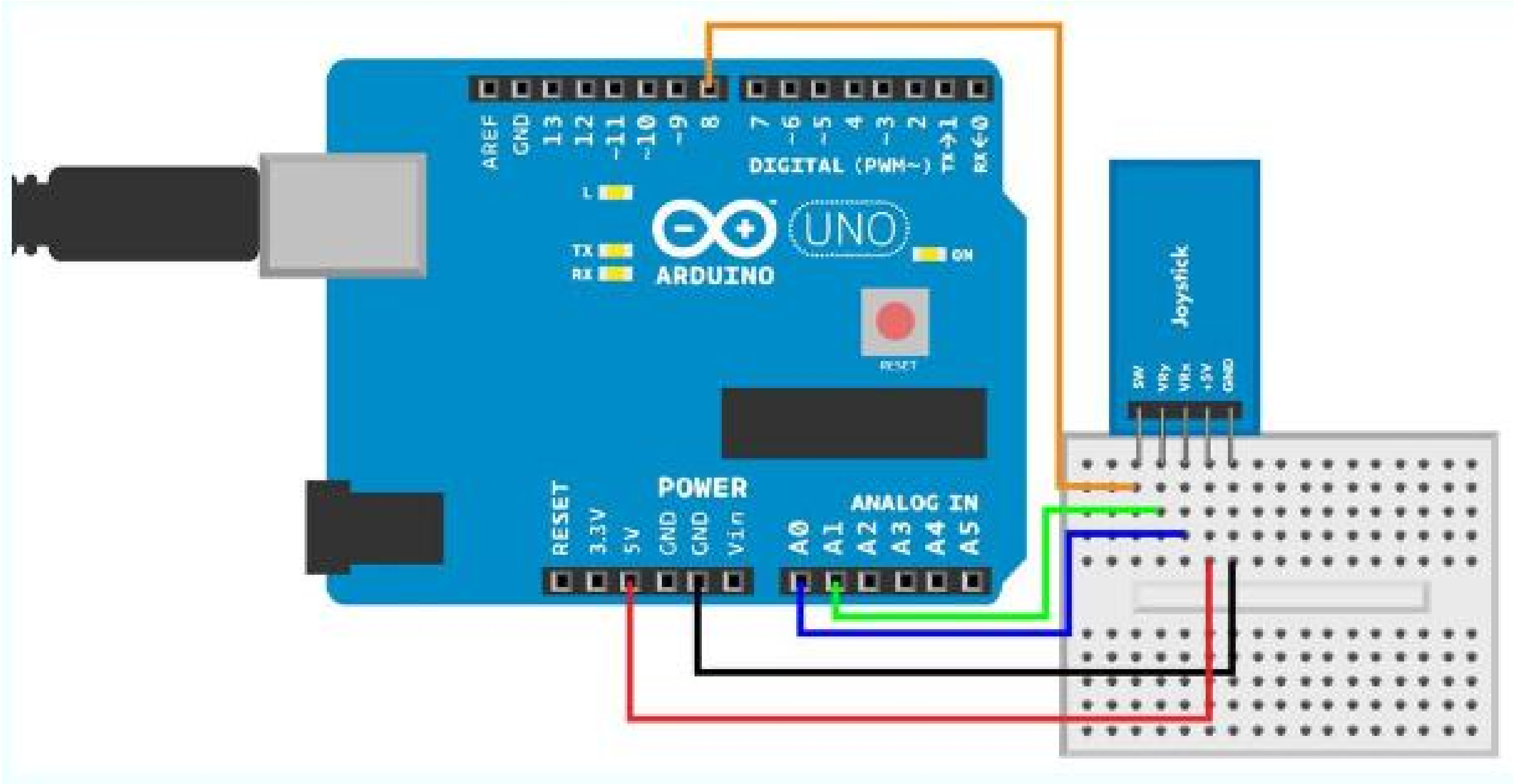


Figure 6-11. Joystick circuit for Arduino

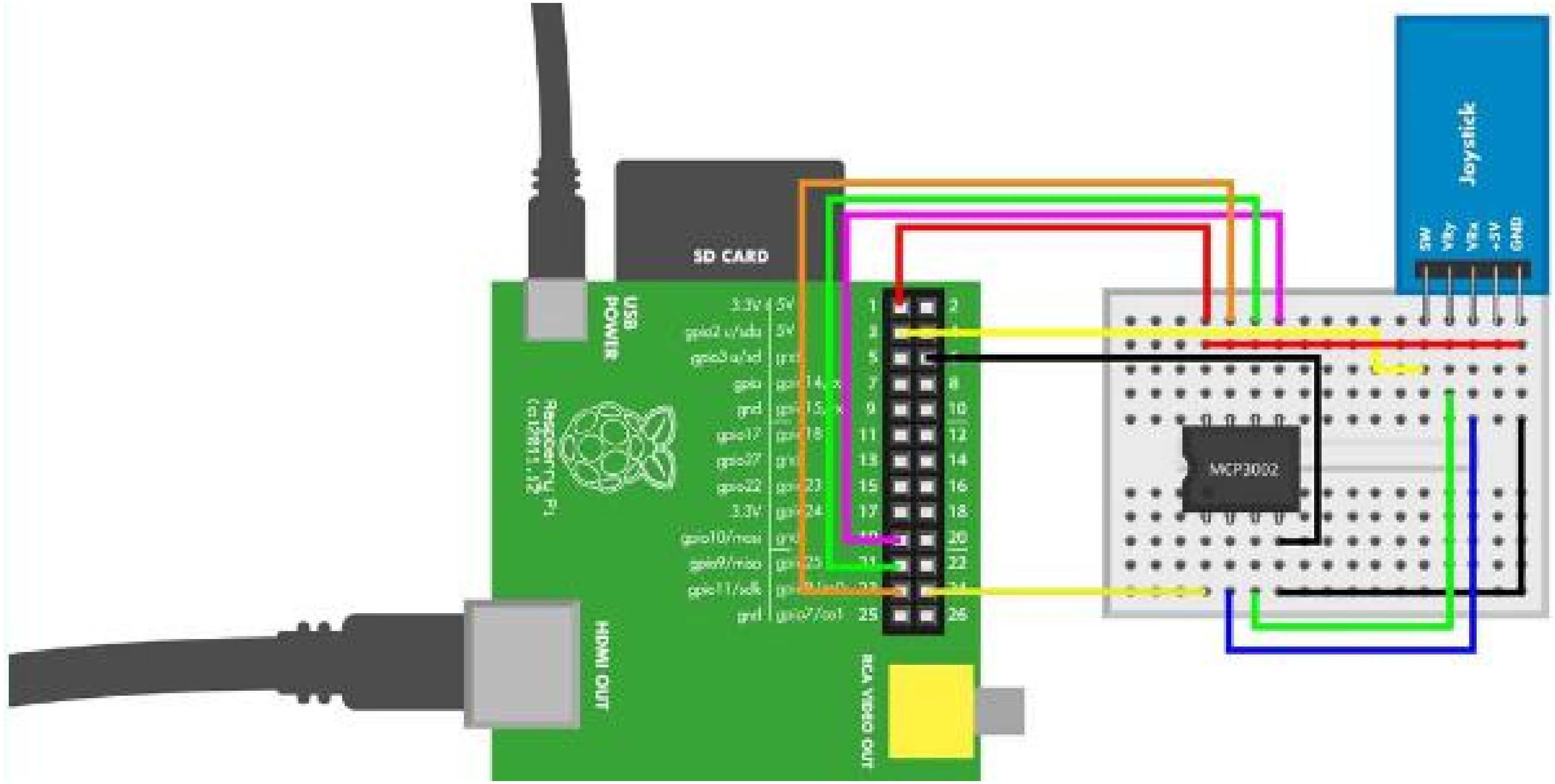
Example 6-7. ky_023_xyjoystick.ino

```
// ky_023_xyjoystick.ino - print joystick position to serial
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

const int VRxPin = 0;                  // ❶
const int VRyPin = 1;
const int SwButtonPin = 8;

int button = -1;                       // ❷ LOW or HIGH
int x = -1; // 0..1023
int y = -1; // 0..1023

void readJoystick() {                  // ❸
        button = digitalRead(SwButtonPin);     // ❹
        x = analogRead(VRxPin);
        y = analogRead(VRyPin);
}
void setup()  {
        pinMode(SwButtonPin, INPUT);
        digitalWrite(SwButtonPin, HIGH);       // ❺ pull-up resistor
        Serial.begin(115200);
}
```

```
void loop()  {
   readJoystick();        // ❻
   Serial.print("X: ");
   Serial.print(x);
   Serial.print(" Y: ");
   Serial.print(y);
   Serial.print(" Button: ");
   Serial.println(button);
   delay(10);
}
```

Joystick Code and Connection for Raspberry Pi

- Figure 6-12 shows the circuit layout for Raspberry Pi and a joystick. Hook everything up as shown, and then run the program shown in Example 6-8.



Example 6-8. xy_joystick.py

# xy_joystick.py - print KY 023 joystick tilt and button status

# (c) BotBook.com - Karvinen, Karvinen, Valtokari

```
import time
import botbook_mcp3002  as mcp            # ❶
import botbook_gpio as gpio               # ❷

def readX():                              # ❸
    return mcp.readAnalog(0, 0)

def readY():
    return mcp.readAnalog(0, 1)     #❹

def readButton():
    buttonPin = 25
    gpio.mode(buttonPin, "in")
    return gpio.read(buttonPin)
```

```
def main():
    while True:                   #❺
        xAxel = readX()           #❻
        yAxel = readY()
        button = readButton()
        print("X: %i, Y: %i, Button: %r" % (xAxel, yAxel, button)) #❼
        time.sleep(0.5)

if __name__ == "__main__":
    main()
```

Environment Experiment: Salvage Parts from an Xbox Controller

- If you have an old game console (Xbox, PlayStation, etc.) control pad lying around, you can salvage two sensors from it and use them with your Arduino or Raspberry Pi (Figure 6-13). Opening the controller is quite easy, but you'll need to use a soldering iron to detach the components from the circuit board.

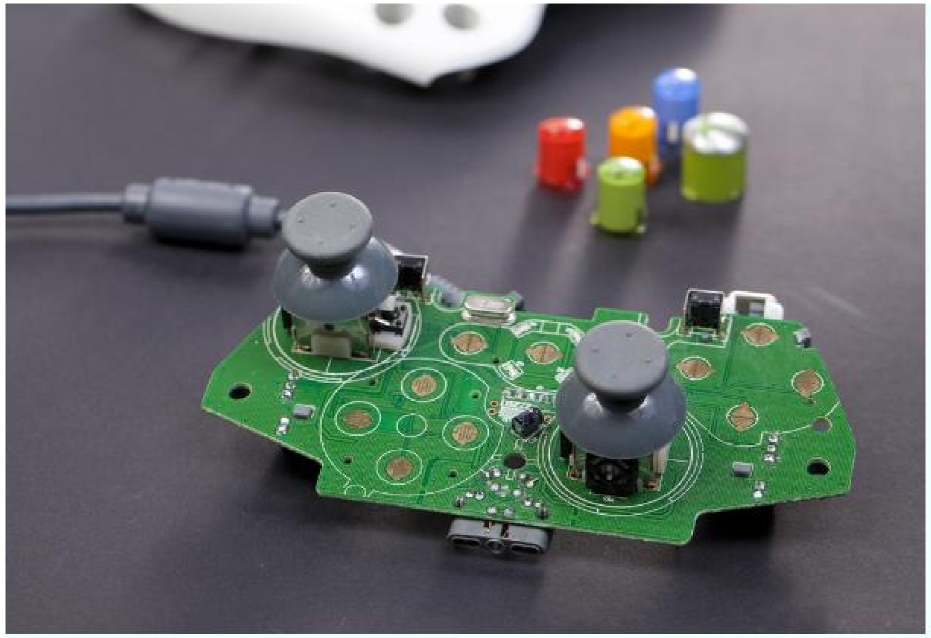


Figure 6-13. Thumb joysticks inside Xbox controller

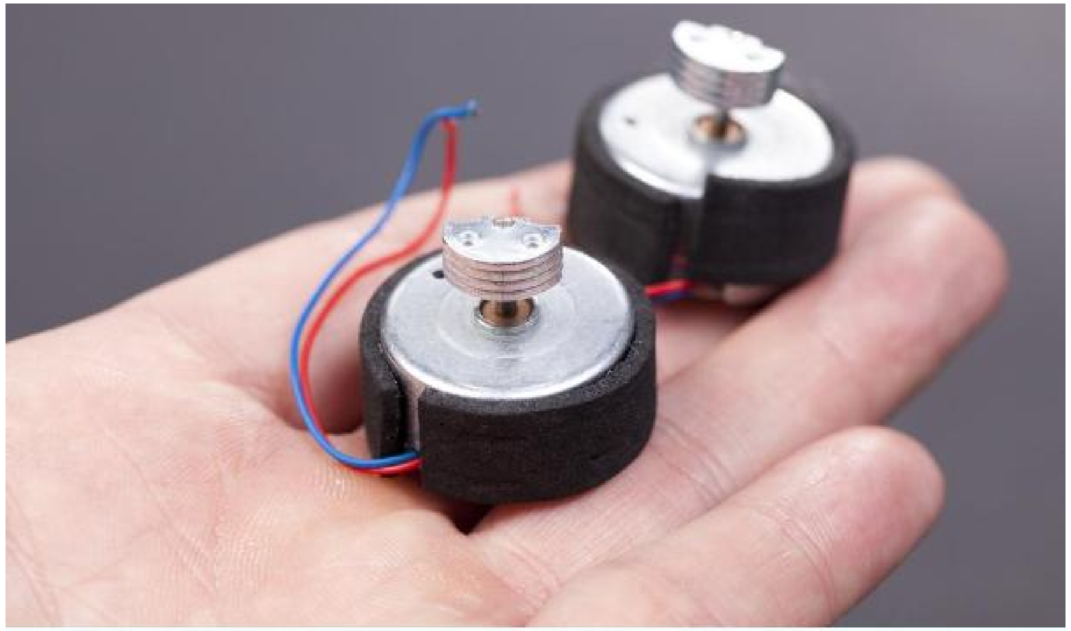
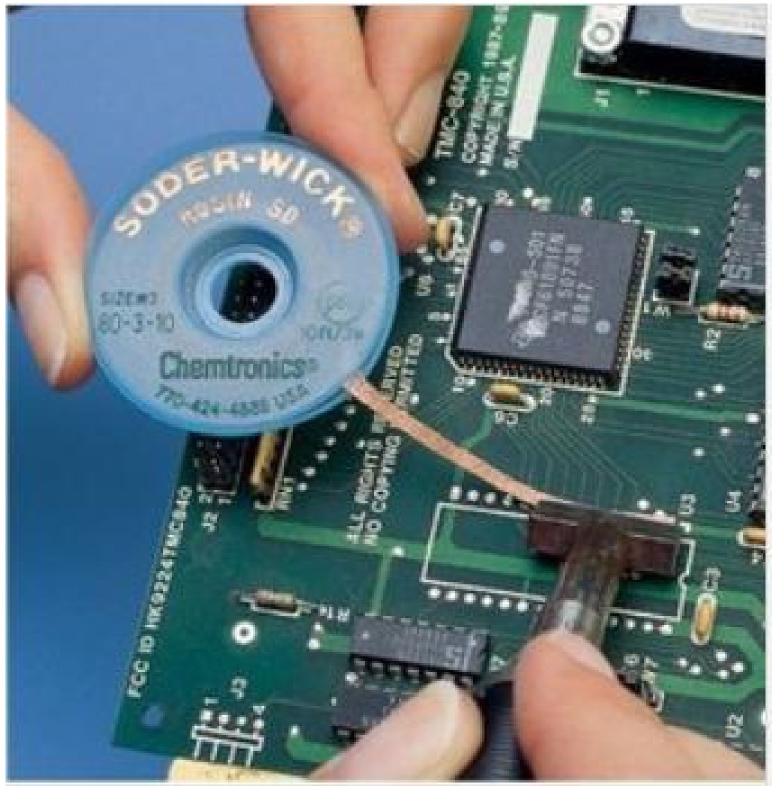


Figure 6-14. Salvaged vibration motors

Environment Experiment: Salvage Parts from an Xbox Controller

- For detaching components, you may also find these tools/supplies helpful: a desoldering pump, solder wick (also known as a braid), or even a flux pen.



吸锡线

Environment Experiment: Salvage Parts from an Xbox Controller

- Many controllers also have a simple force feedback system. For example, if a player takes damage in a game, force feedback makes the controller shake. This is done with vibration motors—eccentric DC motors that spin and shake when powered. Take these out, too, and give them a new home in your own force feedback gadget (Figure 6-14).

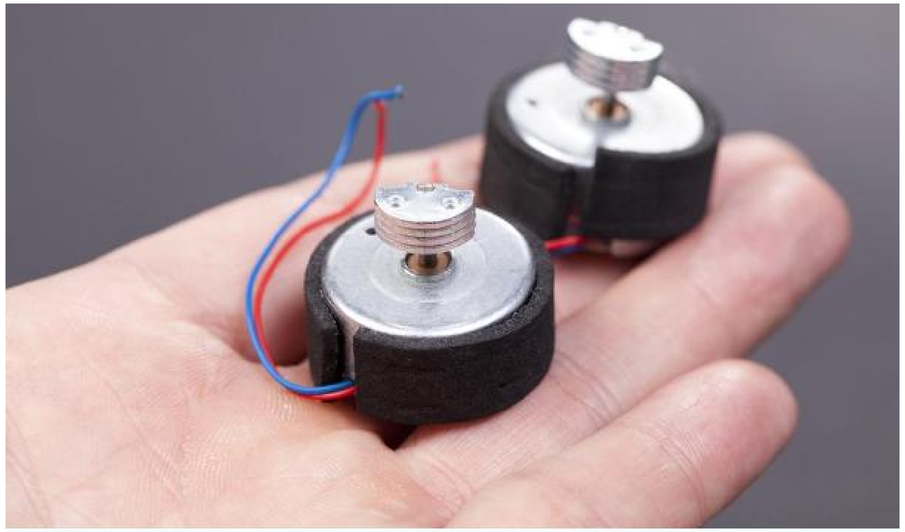


Figure 6-14. Salvaged vibration motors

Contents

Experiment: Burglar Alarm! (Passive Infrared Sensor 被动式红外传感器 PIR)

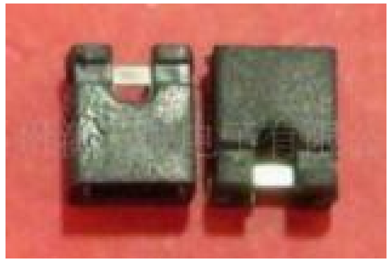
- A passive infrared (PIR) sensor is probably the most common burglar alarm.
- All warm objects radiate invisible infrared light. A PIR sensor reacts to changing infrared light. This change is typically caused by a warm human moving.
- Be sure to let your PIR sensor adapt to its environment first. Because a PIR only detects change, it has to first know the heat pattern in the room when there is no burglar.
- After you turn on the power, the PIR sensor needs 30 seconds to adapt to the environment. There must be no movement or people in the watched area during the adaptation period.

Experiment: Burglar Alarm! (Passive Infrared Sensor 被动式红外传感器 PIR)

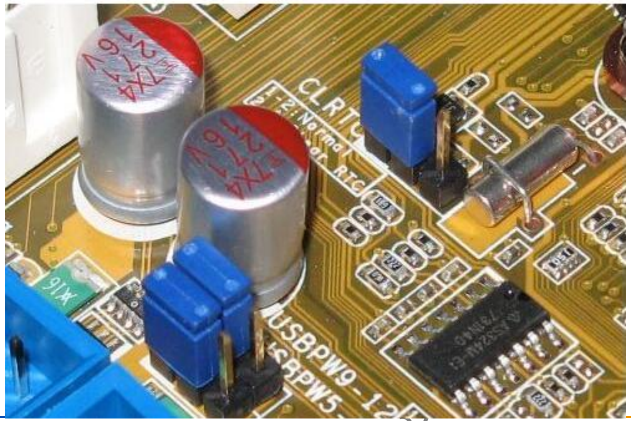
- You can use a box to limit the area watched by a PIR sensor.
- When you want to quickly test a PIR sensor, put it in an upside-down box, so that it can only see upward.
- While the PIR sensor learns what the environment looks like, you can keep writing code without needing to be absolutely still.
- When you want to send an alarm to test your code, wave your hand above the box.

Experiment: Burglar Alarm! (Passive Infrared Sensor 被动式红外传感器 PIR)

- The Parallax PIR has a tiny jumper wire to choose the mode of operation.
- The Parallax PIR has two modes of operation:
  - H for stay High as long as there is movement
  - L for return to Low after an alarm



- So in L mode, the PIR just sends one pulse even if movement continues after it's first detected.



- In this project, set the jumper to H (stay High) mode.

Burglar Alarm Code and Connection for Arduino

- Figure 6-15 shows the circuit diagram for Arduino. Wire it up and run the sketch shown in Example 6-9.

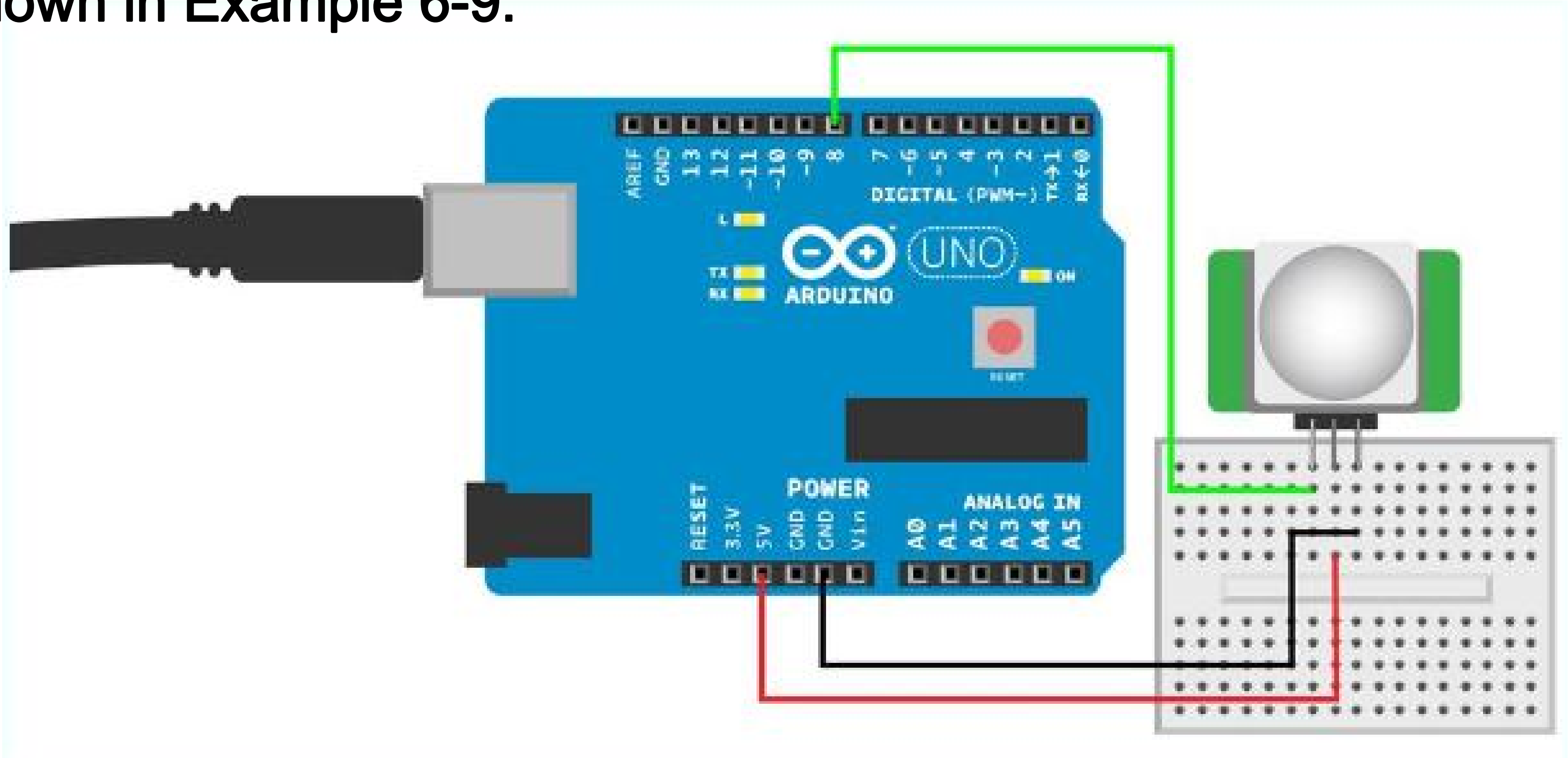


Figure 6-15. Parallax PIR sensor Rev A circuit for Arduino

Example 6-9. parallax_pir_reva.ino

```
// parallax_pir_reva.ino - print movement detection
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

const int sensorPin = 8;
const int ledPin = 13;
const int learningPeriod = 30*1000;       // ms  ❶

int learningCompleted = 0;

void setup() {
    Serial.begin(115200);
    pinMode(sensorPin, INPUT);
    Serial.println("Starting to learn
                     unmoving environment...");  //❷
    pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
  if(millis() < learningPeriod) {              // ❸
        delay(20);    //ms                      // ❹
        return;                                 // ❺
  }
  if(learningCompleted == 0) {                 // ❻
        learningCompleted = 1;
        Serial.println("Learning completed.");
  }
  if(digitalRead(sensorPin) == HIGH) {  // ❼

        Serial.println("Movement detected");
        digitalWrite(ledPin,HIGH);
  } else {
        Serial.println("No movement detected");
        digitalWrite(ledPin,LOW);
  }
  delay(100);
}
```

Burglar Alarm Code and Connection for Raspberry Pi

- Figure 6-16 shows the circuit for Raspberry Pi. Wire it up as shown, and then run the code in Example 6-

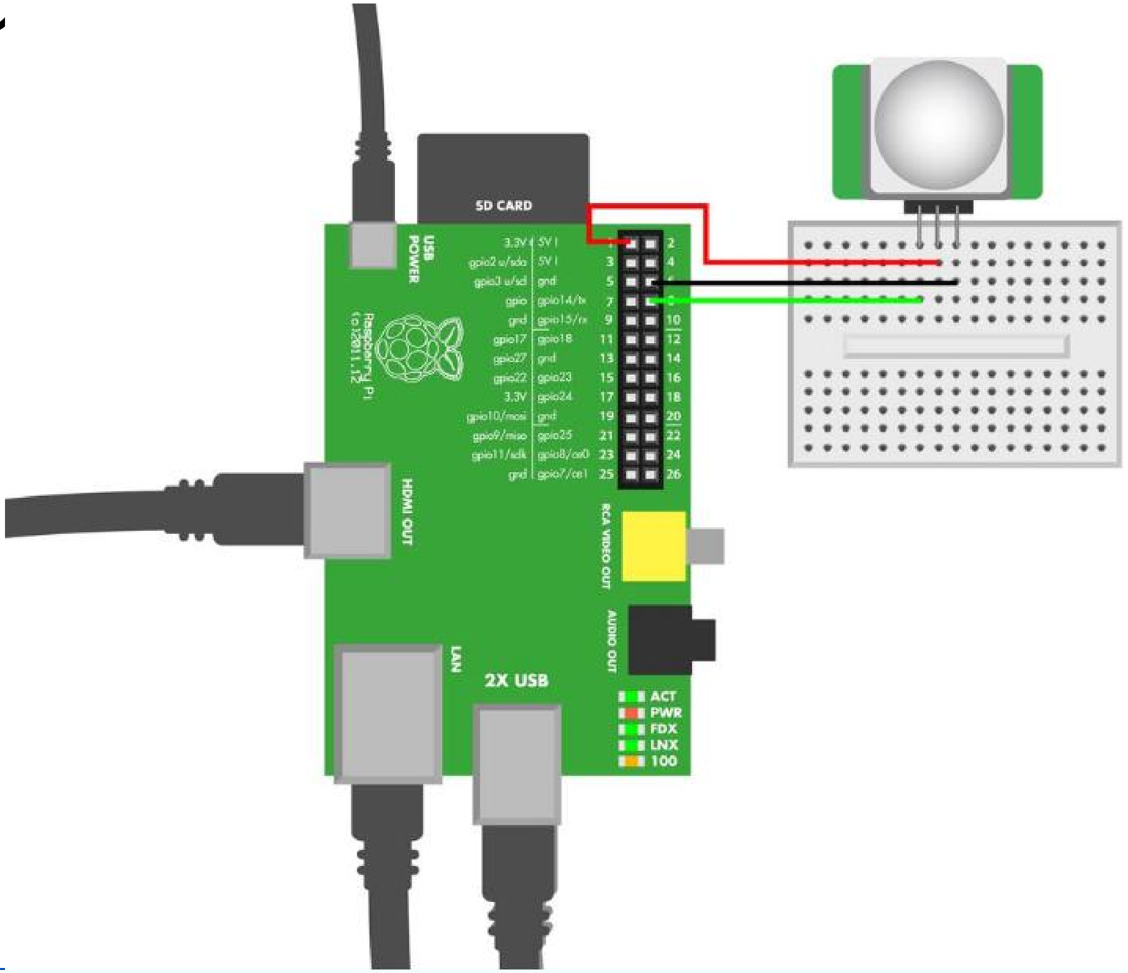


Figure 6-16. Parallax PIR sensor Rev A circuit for Raspberry Pi

Example 6-10. parallax_pir_reva.py

```
# parallax_pir_reva.py - write to screen when movement detected
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

import time
import botbook_gpio as gpio
learningPeriod = 60                #s 传感器需要一段时间处于静置状态，这里设定60s，也可以尝试其他时长

def main():
    pirPin = 14
    gpio.mode(pirPin, "in")
    time.sleep(learningPeriod)         #Learning period
    while (True):
        movement = gpio.read(pirPin)   #当检测到移动时针脚变为High
        if(movement == gpio.HIGH):
            print "Movement detected"
        else:
            print "No movement detected"
        time.sleep(0.3)

if __name__ == "__main__":
    main()
```

Environment Experiment: Cheating an Alarm

- Sometimes a penetration tester (a security expert who probes for vulnerabilities) must move by alarms unnoticed.

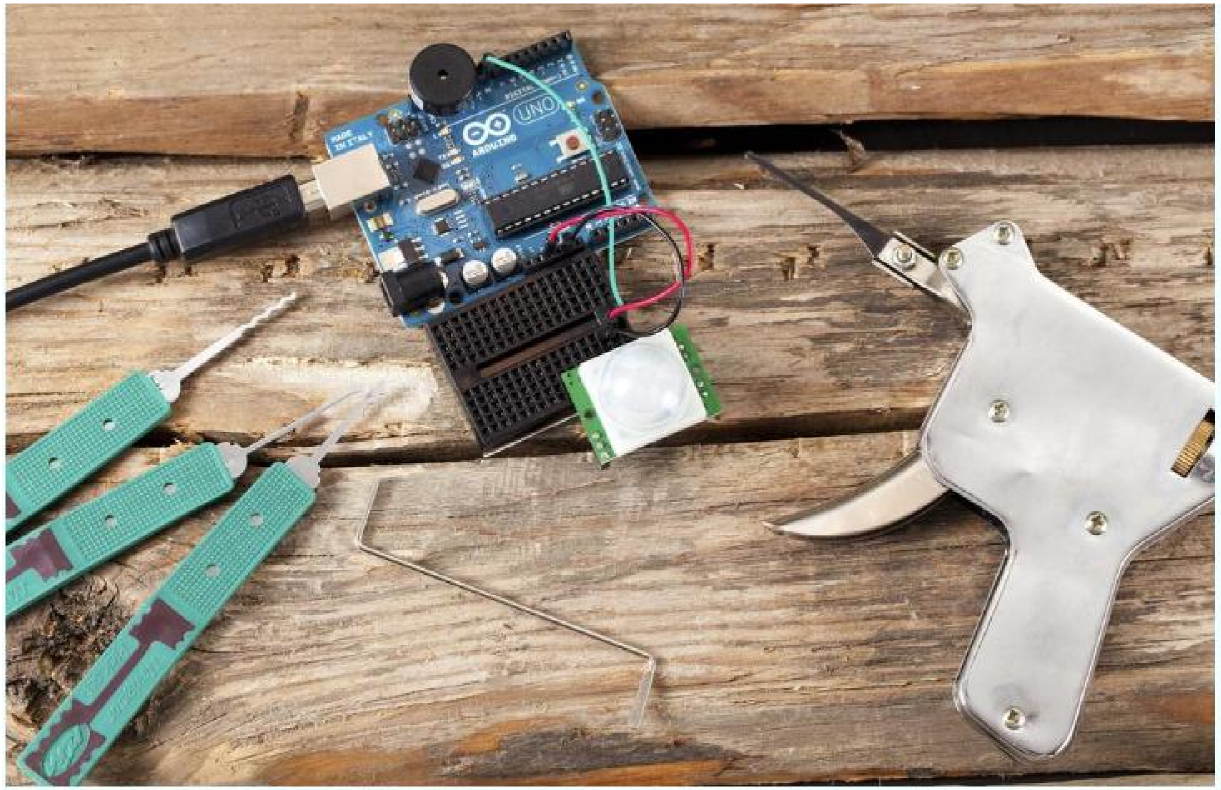


Figure 6-17. Sometimes a penetration tester must bypass alarms

Environment Experiment: Cheating an Alarm

- You can try cheating your PIR sensor, but as you will soon notice it's not quite as easy as in action films. We'll use the same code as before but add a piezo beeper. This way, it's easier to know when movement is detected. Connect the piezo according to the circuit diagram in Figure 6-18 and upload the code in Example 6-11.

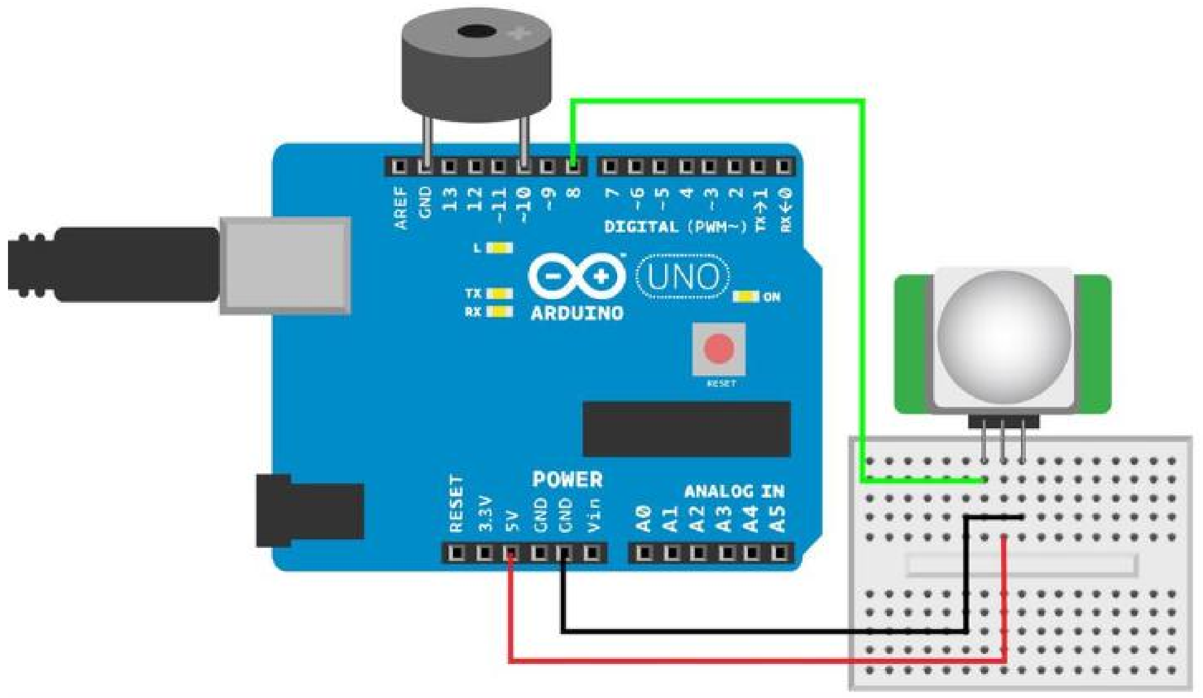


Figure 6-18. Parallax PIR sensor Rev A circuit for Arduino with LED and speaker

Example 6-11. parallax_PIR_revA_cheating_pir.ino

```
// parallax_PIR_revA_cheating_pir.ino - light an LED when movement detected
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

const int sensorPin = 8;
const int ledPin = 13;
int speakerPin = 10;
const int learningPeriod = 30*1000;        // 30s的红外传感器学习适应期.
int learningCompleted = 0;

void setup() {
	Serial.begin(115200);
	pinMode(speakerPin, OUTPUT);
	pinMode(sensorPin, INPUT);
	Serial.println("Start learning for next 30 seconds.");
	pinMode(ledPin, OUTPUT);
}
```

Example 6-11. parallax_PIR_revA_cheating_pir.ino

```
// parallax_PIR_revA_cheating_pir.ino - light an LED when movement detected
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

void alarm(){
	wave(speakerPin, 440, 40);
	delay(25);
	wave(speakerPin, 300, 20);
	wave(speakerPin, 540, 40);
	delay(25);
}

void wave(int pin, float frequency, int duration){
	float period=1/frequency*1000*1000;	// microseconds (us)
	long int startTime=millis();
	while(millis()-startTime < duration) {
		digitalWrite(pin, HIGH);
		delayMicroseconds(period/2);
		digitalWrite(pin, LOW);
		delayMicroseconds(period/2);
	}
}
```

Example 6-11. parallax_PIR_revA_cheating_pir.ino

```
// parallax_PIR_revA_cheating_pir.ino - light an LED when movement detected
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

void loop() {
	if(millis() < learningPeriod) {
		return;		// Sensor has not yet learned its environment.
	}
	if(learningCompleted == 0) {
		learningCompleted = 1;
		Serial.println("Learning completed.");
	}
	if(digitalRead(sensorPin) == HIGH) {
		Serial.println("Movement detected");
		alarm();
		digitalWrite(ledPin, HIGH);
	} else {
		Serial.println("No movement detected");
		digitalWrite(ledPin, LOW);
	}
	delay(100);
}
```

Contents

Test Project: Pong

- Detecting movement is much more interesting when you can actually present that information to the user. In this project you'll learn how to use sensor data to move objects on the screen.
- To keep things simple, we use a joystick as an input for a Pong game in this example.
- What you learn here can be easily adapted to other projects. Any sensor could be the input device, and only your imagination limits what is shown on the screen.

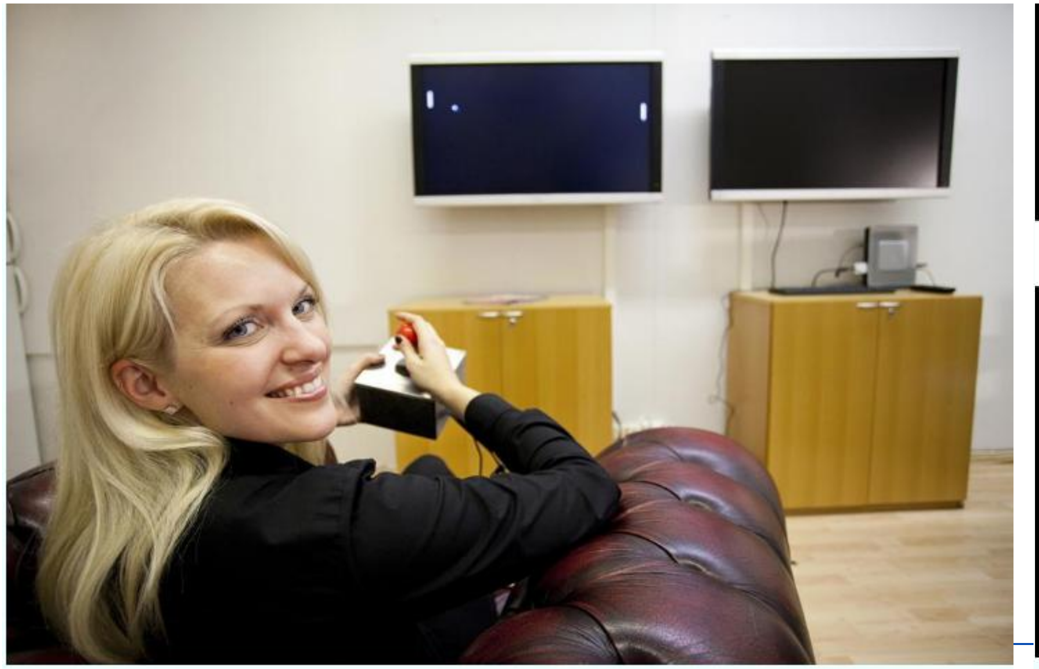


Figure 6-19. Game on!

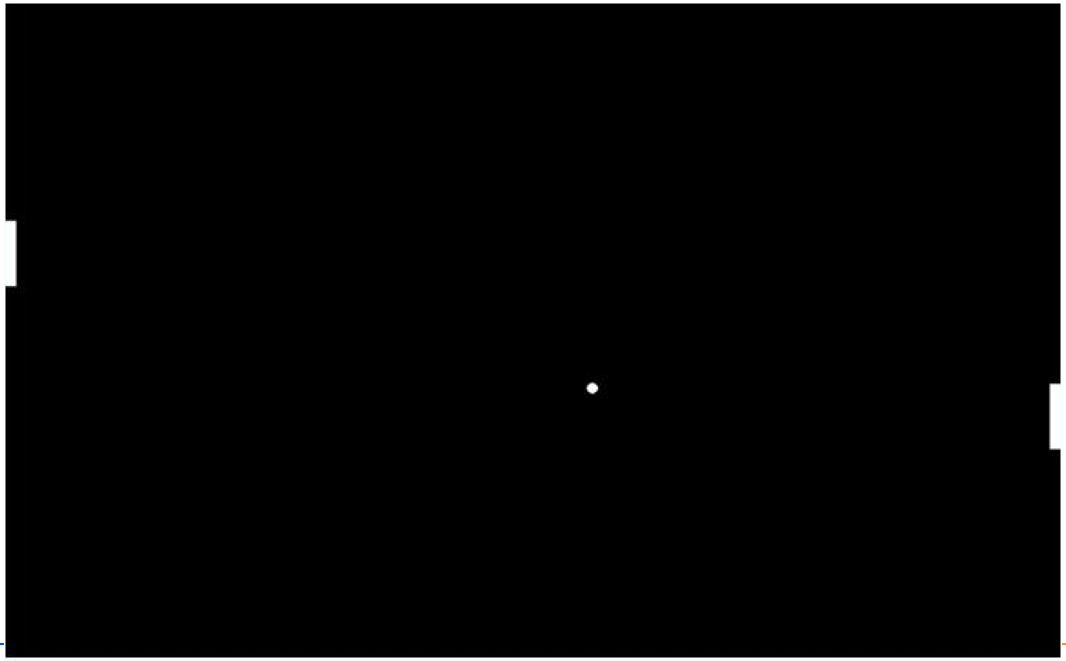


Figure 6-20. Pong playfield

Test Project: Pong

- This project introduces you to pyGame, one of the easiest libraries for programming games.
- You'll build your own game console and learn to use sensor input for moving things in the big screen. For added effect, use a video projector for output.
- This project is easy to do with Raspberry Pi, as you can connect your normal television or video projector to Raspberry Pi's HDMI connector. Doing the same in Arduino would not be as straightforward.

Test Project: Pong

- **What You'll Learn，In the Pong project:**
  - Use data from a sensor to move objects on the screen.
  - Display full high-def graphics with Raspberry Pi.
  - Make Raspberry Pi react faster by drawing directly to screen, without going through the desktop environment or the X Window System.
  - Program a simple game with Python's pyGame.
  - Automatically start your program when Raspberry Pi boots.

Test Project: Pong

- Figure 6-21 shows the wiring diagram for the Pong project. Wire it up as shown, and then run the program shown in Example 6-12.
- Be sure that the botbook_gpio.py library is in the same directory as the pong.py program.

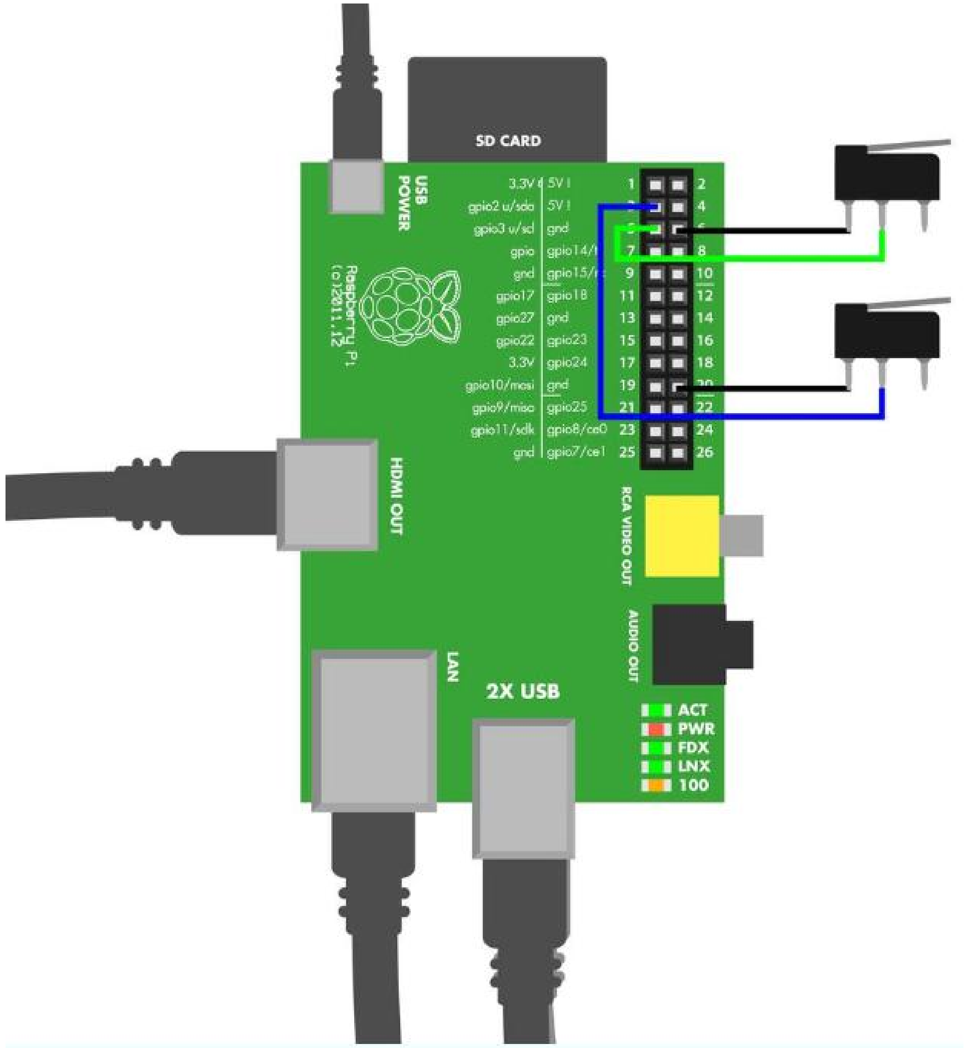


Figure 6-21. Pong connection for Raspberry Pi

Example 6-12. pong.py

```
# pong.py - play ball game classic with joystick and big screen
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

import time
import sys
import pygame
import botbook_gpio as gpio
from pygame.locals import *

print "Loading BotBook.com Pong...“
pygame.init()                                    #❶ 使用pyGame之前要进行初始化

width = pygame.display.Info().current_w          #❷ pyGame预定义画布尺寸
height = pygame.display.Info().current_h
size = width, height                             #❸ 把画布尺寸存储在size元组中
background = 0, 0, 0                             #❹ pyGame使用RGB表示颜色值，（0,0,0）代表黑色
screen = pygame.display.set_mode(size,pygame.FULLSCREEN)      #❺ screen是进行绘图的对象
```

Example 6-12. pong.py

# pong.py - play ball game classic with joystick and big screen
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

```
ballrect = Rect(width/2, height/2, 16, 16)             #❻ 将球置于屏幕中央，碰撞范围是16*16像素
computerrect = Rect(width-20, 0, 20, 120)              #❼ 将计算机玩家的球拍置于屏幕右下角，碰撞范围是20*120
像素
playerrect = Rect(0, 0, 20, 120)                       #❽ 将玩家的球拍置于屏幕左上角，碰撞范围是20*120像素

normalSpeed = 512
speed = [normalSpeed, normalSpeed]                     #❾ 乒乓球的速度向量，X轴上移动64px/s，Y轴上移动64px/s
                                                          如果X，Y方向的运动速度不同，乒乓球只能向45°方向移动
#❿ 创建Clock对象并存储到变量中，它的作用是保持速度的一致性，即使树莓派处于超频状态
clock = pygame.time.Clock()
pygame.mouse.set_visible(False)
```

Example 6-12. pong.py

```
# pong.py - play ball game classic with joystick and big screen
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

uppin = 2
downpin = 3
gpio.mode(uppin, "in")
gpio.mode(downpin, "in")
mainloop = True

while mainloop:                              #⑪ pyGame采用游戏中常见的主循环编程方式
    seconds = clock.tick(30) / 1000.0        #⑫ 从上一帧开始所经历的秒数
    #处理用户输入
    for event in pygame.event.get():         #⑬输入由pygame.event 进行处理，.get返回一个可迭代的LIST
        if event.type == pygame.QUIT: mainloop = False        #⑭
        if (event.type == KEYUP) or (event.type == KEYDOWN):  #有按键按下或抬起
            if event.key == K_ESCAPE: mainloop = False
```

Example 6-12. pong.py

# pong.py - play ball game classic with joystick and big screen
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

```
# 动作和碰撞
playerspeed = 0
if gpio.read(uppin) == gpio.LOW:
    playerspeed = -normalSpeed          #如果up键按下，则减掉一定速度的高度，越往上y值越小
if gpio.read(downpin) == gpio.LOW:
    playerspeed = normalSpeed           #如果down键按下，则加一定速度的高度，越往下y值越大
ballrect.x += speed[0] * seconds        #⑮ 球匀速运动
ballrect.y += speed[1] * seconds
if ballrect.left < 0 or ballrect.right > width:    #⑯如果乒乓球撞击到了场地的边界，让它反弹
    ballrect.x = width/2;                          #撞到左右边界，弹回到水平中部
if ballrect.top < 0 or ballrect.bottom > height:
    speed[1] = -speed[1]                           #撞到上下边界，y运行方向取反
```

Example 6-12. pong.py

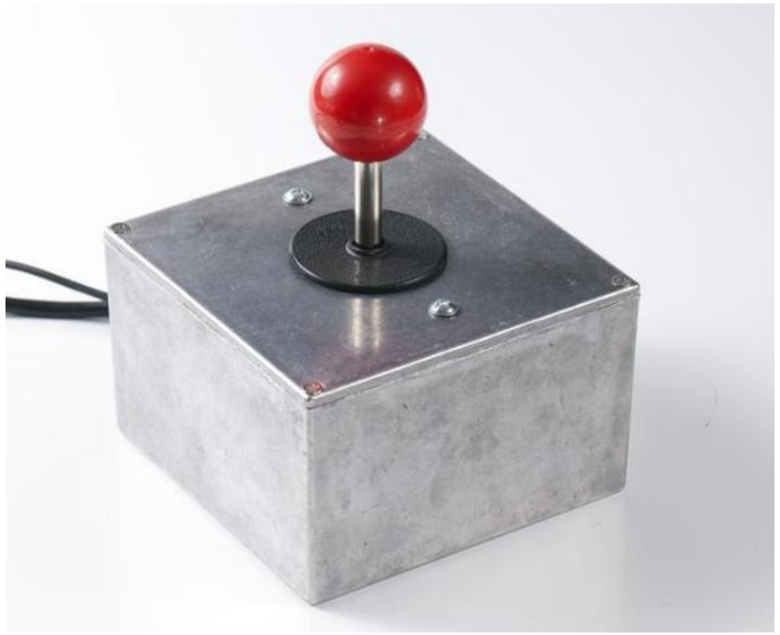
```
# pong.py - play ball game classic with joystick and big screen
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

computerrect.y = round(ballrect.y)              #⑰把机器球拍移到与球同一个高度
playerrect.y += playerspeed * seconds           #⑱根据输入的加速度，垂直地移动玩家的球拍
#如果玩家球拍移动超出画布边界，就停留在边界
if playerrect.top < 0:                          #⑲
    playerrect.top = 0
if playerrect.bottom > height:                  #⑳
    playerrect.bottom = height

if computerrect.colliderect(ballrect):   #检测球是否碰撞到球拍，如果是则设置乒乓球方向相反
    speed[0] = -normalSpeed                #机器球拍在右侧，所以速度方向由正变负
if playerrect.colliderect(ballrect):       #玩家球拍在左侧，所以速度方向由负变正
    speed[0] = normalSpeed
```

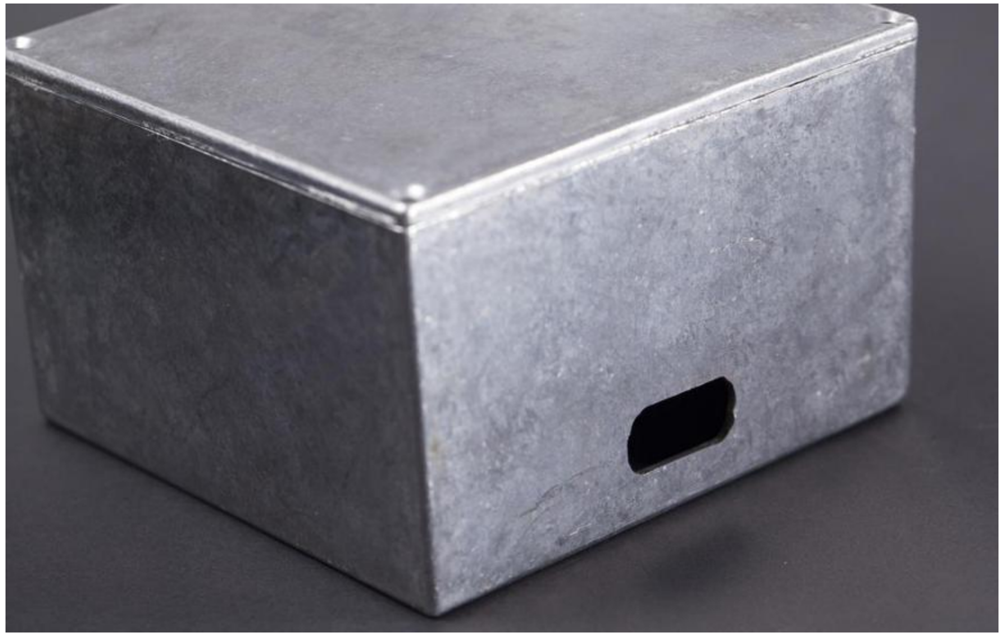
Example 6-12. pong.py

```
# pong.py - play ball game classic with joystick and big screen
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

# 绘制帧
screen.fill(background)
#将乒乓球绘制在计算的位置上
pygame.draw.circle(screen, (255, 255, 255), ( int(round(ballrect.x+8)),
                                              int(round(ballrect.y+8))), 10)
#将计算机的球拍绘制在之前计算的位置。因为它的形状为矩形，所以其碰撞边界与球拍自身完全相同
pygame.draw.rect(screen, (255, 255, 255), computerrect)
pygame.draw.rect(screen, (255, 255, 255), playerrect)
pygame.display.update()          # 绘制本帧中的所有物体
```

Pong Packaging Tips

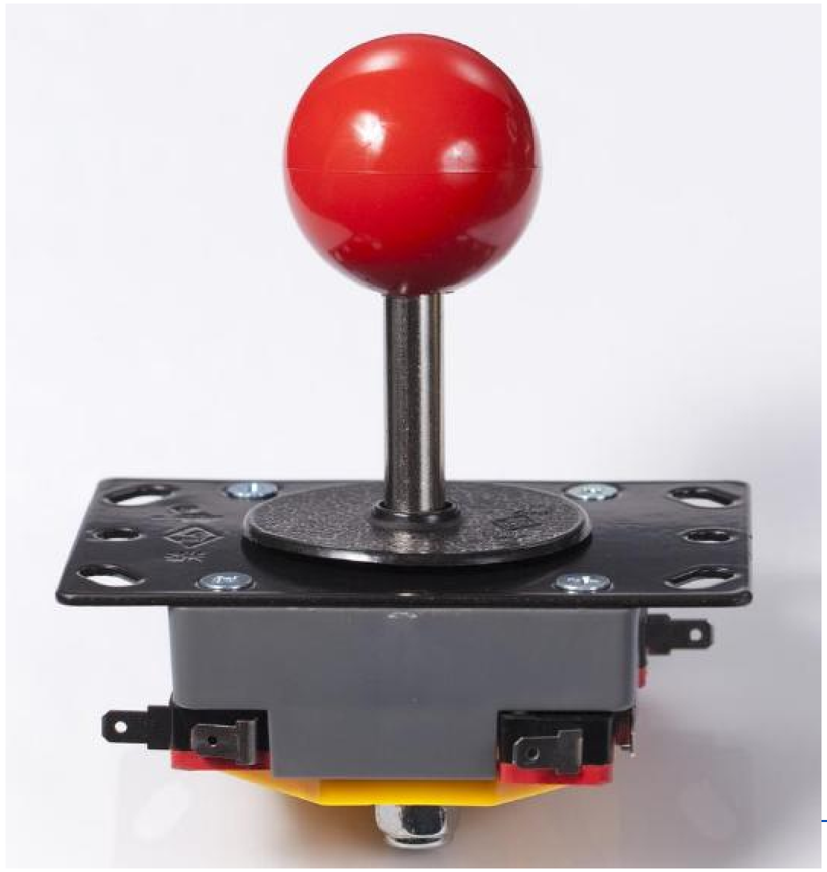
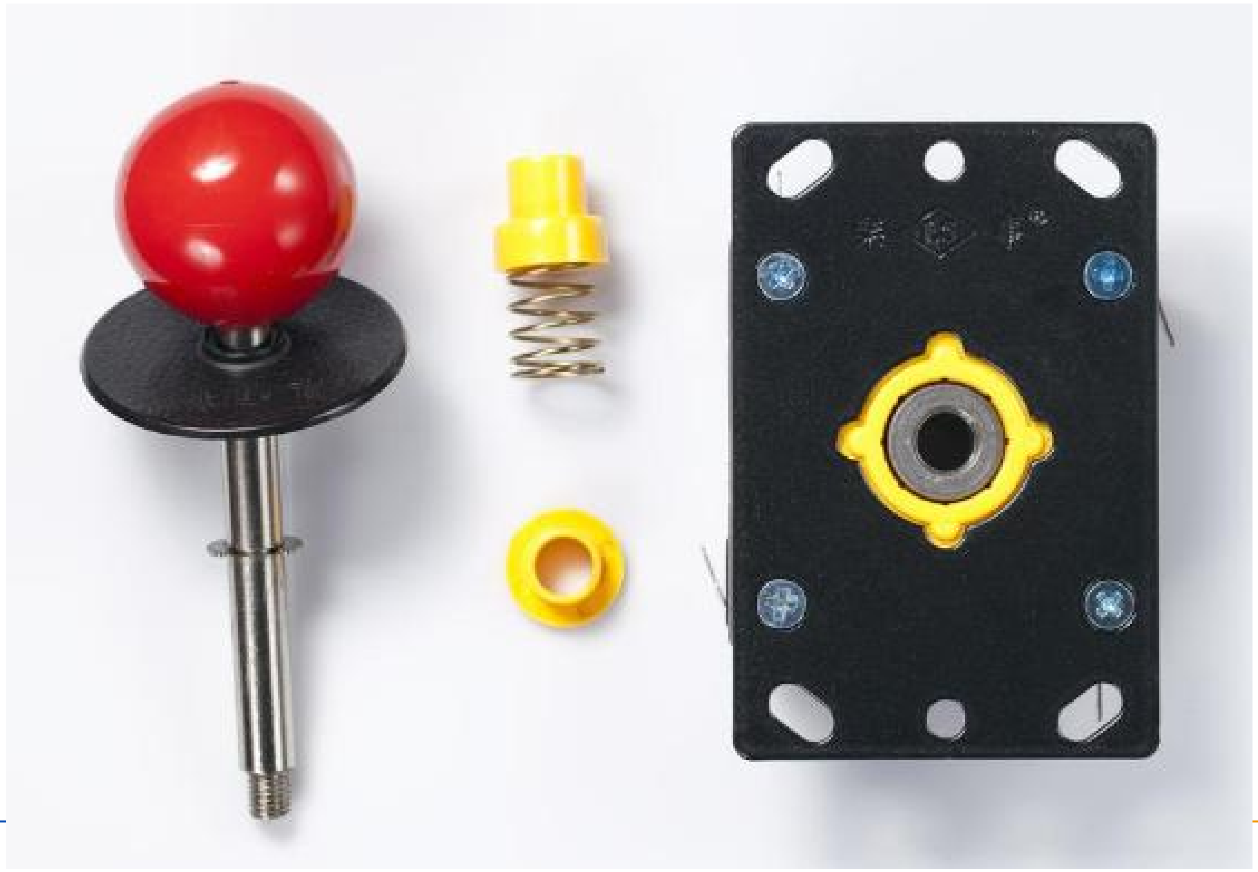
- We used a diecast aluminum box to make a super robust casing with street credibility for our game console (Figure 6-22). This construction makes an attractive alternative to the flimsy plastic gadgets we usually see and use.
- In the back of the box, we made a hole for power and HDMI cables (Figure 6-23). To make a wide hole like ours, drill two large holes separately and then remove the metal in between with a jigsaw blade.





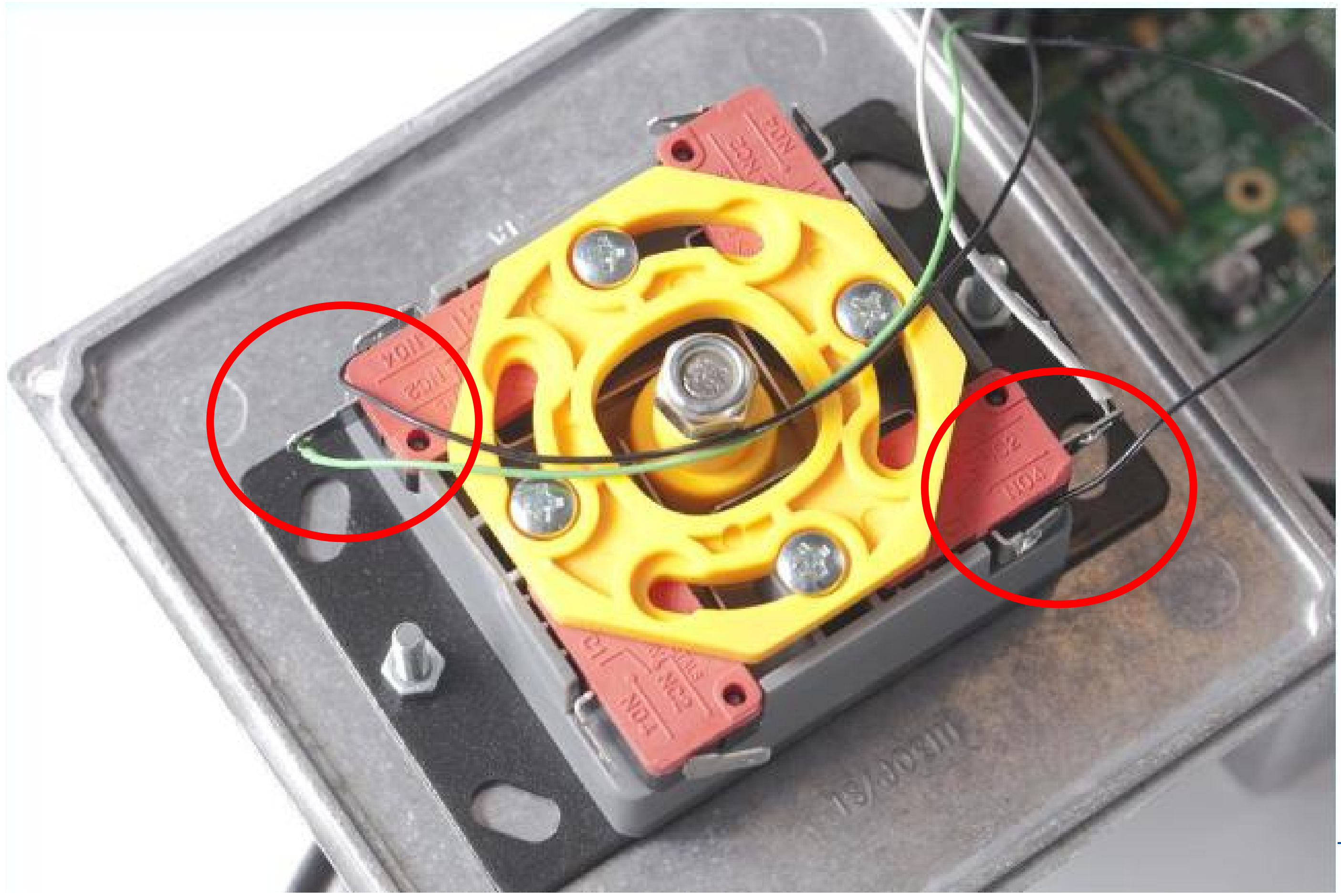
Pong Packaging Tips

- A traditional arcade joystick makes a perfect match for the indestructible box (Figure 6-24). To attach the joystick, you need three holes: two 5 mm to mount the frame and one 10 mm to get the actual stick through the cover (Figure 6-25).

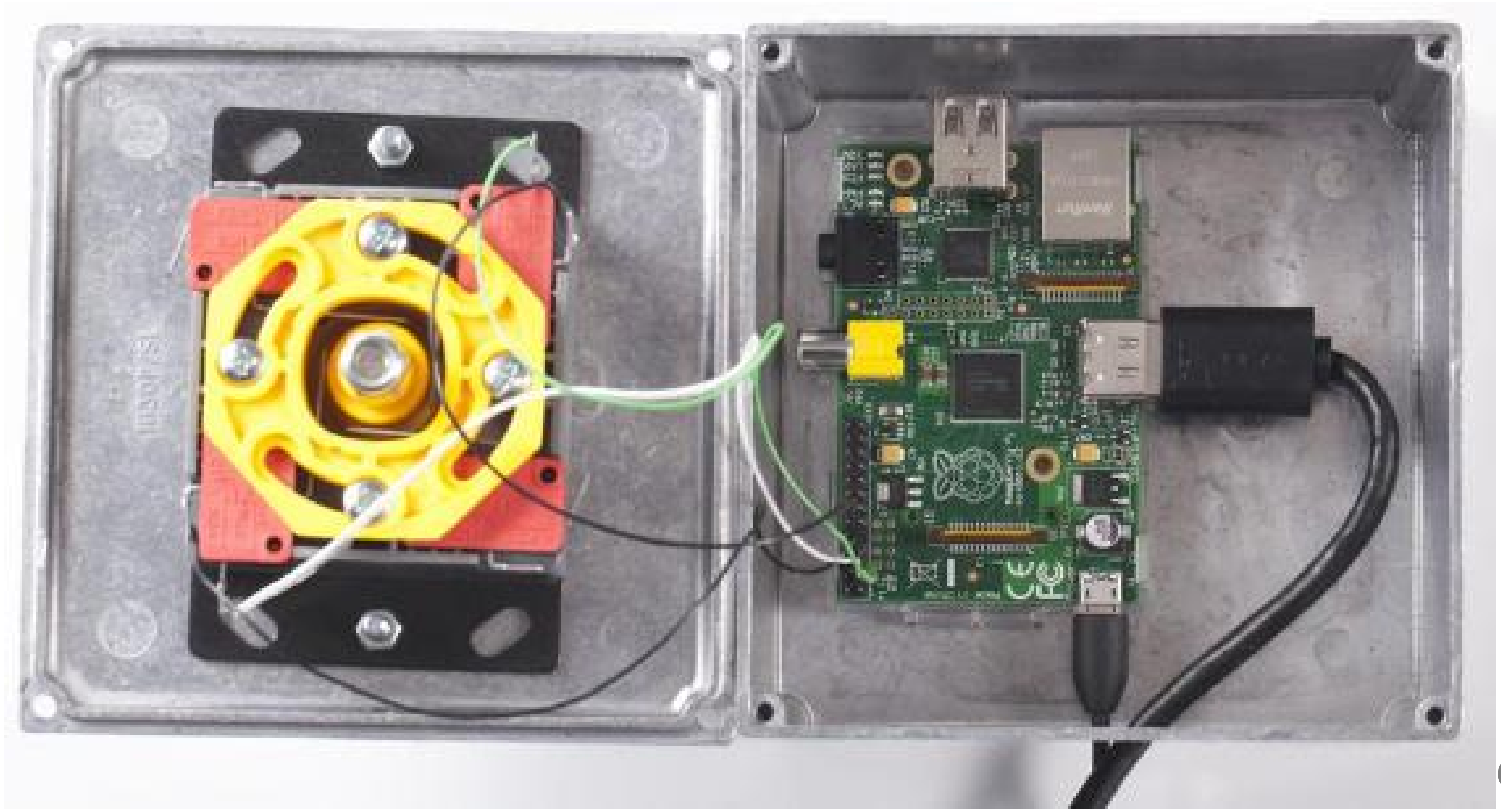
Pong Packaging Tips

- As we are going to control only up and down movement, we need to solder wires to two of the joystick's microswitches, as shown in Figure 6-26.



Pong Packaging Tips

- Raspberry Pi was attached simply by hot gluing the Raspberry Pi's cover box bottom to the bottom of our aluminum box (see Figure 6-27).



Automatically Start Your Game When Raspberry Pi Boots

- Wouldn't it be nice if the game started on boot?
  - Without a keyboard, it's difficult to type *python pong.py* each time you want to start the game.
- To start the game as a normal user, you'll set the system to log in as the user pi automatically. Then you'll configure things to start the game in the user's login script. This way, the game starts immediately when you boot Raspberry Pi.

Run Game on Login

- When you log in, bash opens. Bash is your shell: it interprets the commands that you type at the command prompt ($).
- If your Raspberry Pi automatically boots into the graphical desktop (the default), you should change it to start up in a text mode shell, because you need to run it from text mode in order for it to run automatically on login.
- Open LXTerminal and run *sudo raspi-config*. There you can choose Enable Boot To Desktop and choose Console Text console.

Run Game on Login

- The first thing that bash does is to run scripts, such as
  - *.profile, .login, .bash_profile, and .bash_login.*
- Just like all per-user configuration files, they are hidden files in the user home directory. To see them, you must use *ls -a*.
- If you're not already in your home directory, you can quickly change to it (/home/pi/) with cd without arguments. Here's a set of commands that change to your home directory and list all the files there:

```
$ cd
$ ls -a
```

Run Game on Login

- The .bash_login file is a shell script: it has commands to run, one after another.
- Before you add this command to the file, try it out on the command line:

```
$ python /home/pi/makesensors/pong/pong.py
```

- If everything worked OK, you can now open the .bash_login file with the nano editor:

```
$ nano .bash_login
```

- Add the line *python /home/pi/makesensors/pong/pong.py* to .bash_login. If there are any other lines already in the file, delete them.

Run Game on Login

- Example 6-13 shows what the .bash_login file should now look like. Save the file by typing Control-X, then press y when prompted to Save, and finally type Enter/Return to confirm the filename.

Example 6-13. bash_login

```
# /home/pi/.bash_login - automatically start pong game on login

python /home/pi/makesensors/pong/pong.py
```

- Log out:

```
$ exit
```

- Next, log back in. The game starts automatically.

Automatic Login

- Now it’s time to make your user “pi” log in automatically on boot. Because the game runs immediately after “pi” logs in, this will start the game after you power up the Pi. If you’re still in the game, press Esc to escape from it.
- The init program controls system boot, so you will need to modify its settings. All system-wide settings are in /etc/, and the configuration file usually starts with the name of the thing: /etc/init* (in this case, /etc/inittab). Because logging users in is a process that requires full privileges, you need to edit the file as root with the sudoedit command:

```
$ sudoedit /etc/inittab
```

Automatic Login

- Modify the line that governs the first “virtual terminal” so that it automatically logs you in (don’t add this line, modify the one that starts with 1:2345:respawn:/sbin/getty):

```
1:2345:respawn:/sbin/getty --noclear 38400 tty1 --autologin pi
```

- Save with Control-X, answer y when asked about saving, and then press Enter or Return to confirm the filename. You can see a complete, modified /etc/inittab in Example 6-14.

Example 6-14. inittab

# /etc/inittab - automatically log in user pi on boot (also disable serial)

```
id:2:initdefault:
si::sysinit:/etc/init.d/rcS
~~:S:wait:/sbin/sulogin
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6

z6:6:respawn:/sbin/sulogin
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop
```

```
# 已修改:
1:2345:respawn:/sbin/getty --noclear 38400 tty1 --autologin pi
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
# 移除串行控制台，让串口和传感器一起使用:
# T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

Automatic Login

- Shut down the Raspberry Pi:
  - $ sudo shutdown -P now
- Then disconnect and reconnect USB power.
- Relax as you are automatically logged in. The game starts, and your very own game console is ready. Time to play Pong!

summary

- You have now tested movement in many ways, starting from basic buttons and potentiometers. These sensors are the archetypes of resistance sensors. The button is the simplest digital (on/off or zero/infinite resistance) resistance sensor, and the potentiometer is the simplest analog resistance sensor.
- For more specialized sensors, you’ve detected touch—even through wood. And you’ve measured pressure, which could be useful in your projects to see if a bed or a seat is occupied, or just to have a finger strength competition.
- Next, you’ll measure a less tangible form of energy: light.