7. Smoke and Gas

Chapter 4. Smoke and Gas

- BEEP BEEP BEEP! The loud noise of a fire alarm has saved many lives, waking up residents before carbon monoxide lulls them into a permanent sleep.

- Another gas sensor, an alcometer, has kept many drunk drivers off the road and avoided lethal consequences.

Chapter 4. Smoke and Gas

- When you start yawning at work or in an otherwise interesting class, the culprit could be carbon dioxide (CO_2). It's the gas all animals (including humans) exhale.

- Sensors in a building's ventilation/air conditioning system could notice an elevated CO_2 level and send you some needed fresh air.

Chapter 4. Smoke and Gas

- The fire department can measure if there is hydrocarbon vapor in the air to avoid explosive surprises.

- There is also a gas sensor inside your car engine, which measures the fuel-air ratio. A correct fuel-air ratio ensures that all the gasoline burns in the cylinder.

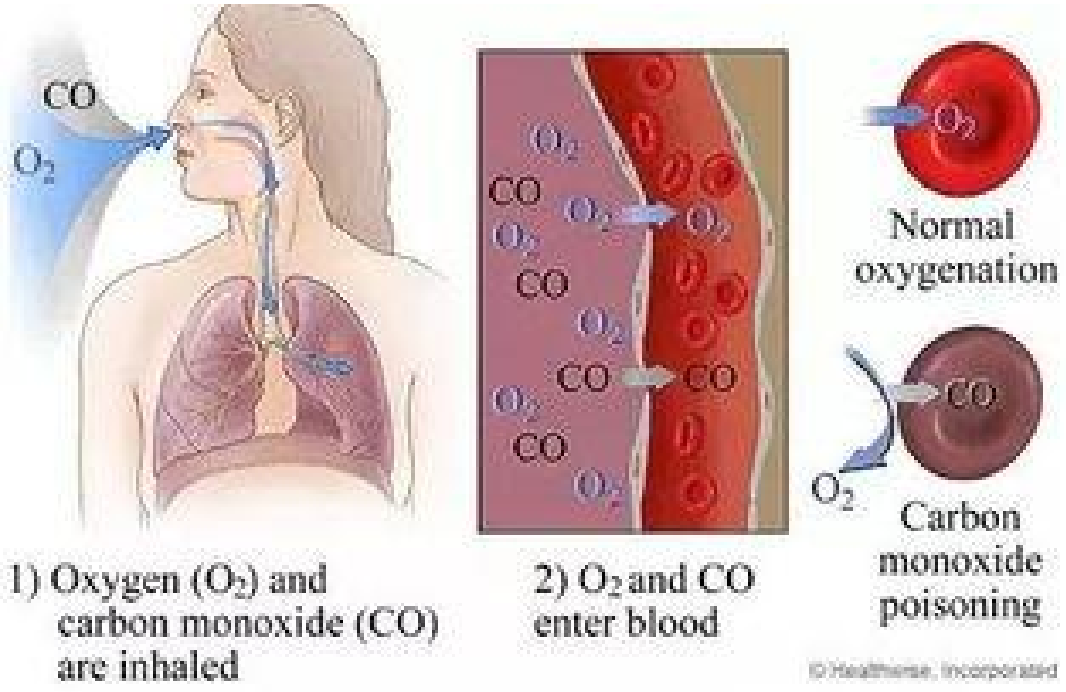
Chapter 4. Smoke and Gas

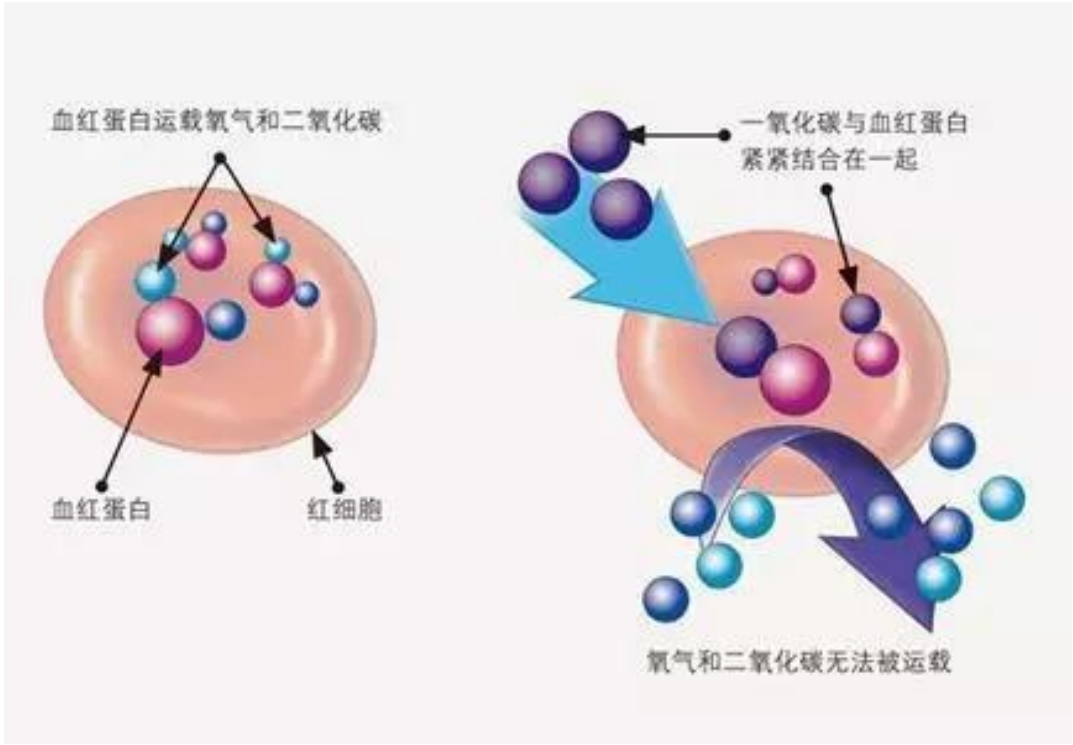
- MQ is a series of inexpensive gas sensors. There are sensors for many gases, some of which are listed in Table 4-1.

MQ Sensor	Gases detected
MQ-2	Flammable gas and smoke （可燃气体和烟雾）
MQ-3， MQ-303A	Alcohol (ethanol) （酒精——乙醇）
MQ-4	Methane (CH4) （甲烷）
MQ-7	Carbon monoxide （一氧化碳）
MQ-8	Hydrogen （氢）
MQ-9	Carbon monoxide, methane, LPG (propane or butane) （一氧化碳，甲烷，丙烷或丁烷）

Contents

Experiment: Detect Smoke (Analog Gas Sensor)





Experiment: Detect Smoke (Analog Gas Sensor)

- An MQ-2 smoke sensor reports smoke by the voltage level it puts out. The more smoke there is, the higher the voltage（模拟传感器）. The MQ-2 we used has a built-in potentiometer for adjusting sensitivity (see Figure)
- As the MQ-2 has three leads, it takes ground (black, 0 V) , +5 V (red) and S pin.
- It measures smoke and sets its S pin voltage higher (nearer to +5 V) when it detects smoke.

Figure 4-1. Analog gas sensor

MQ-2 Code and Connection for Arduino

- Arduino has a built-in analog-to-digital converter, so you can read the MQ-2 with a call to analogRead(). Use the potentiometer on the breakout board to adjust its sensitivity.

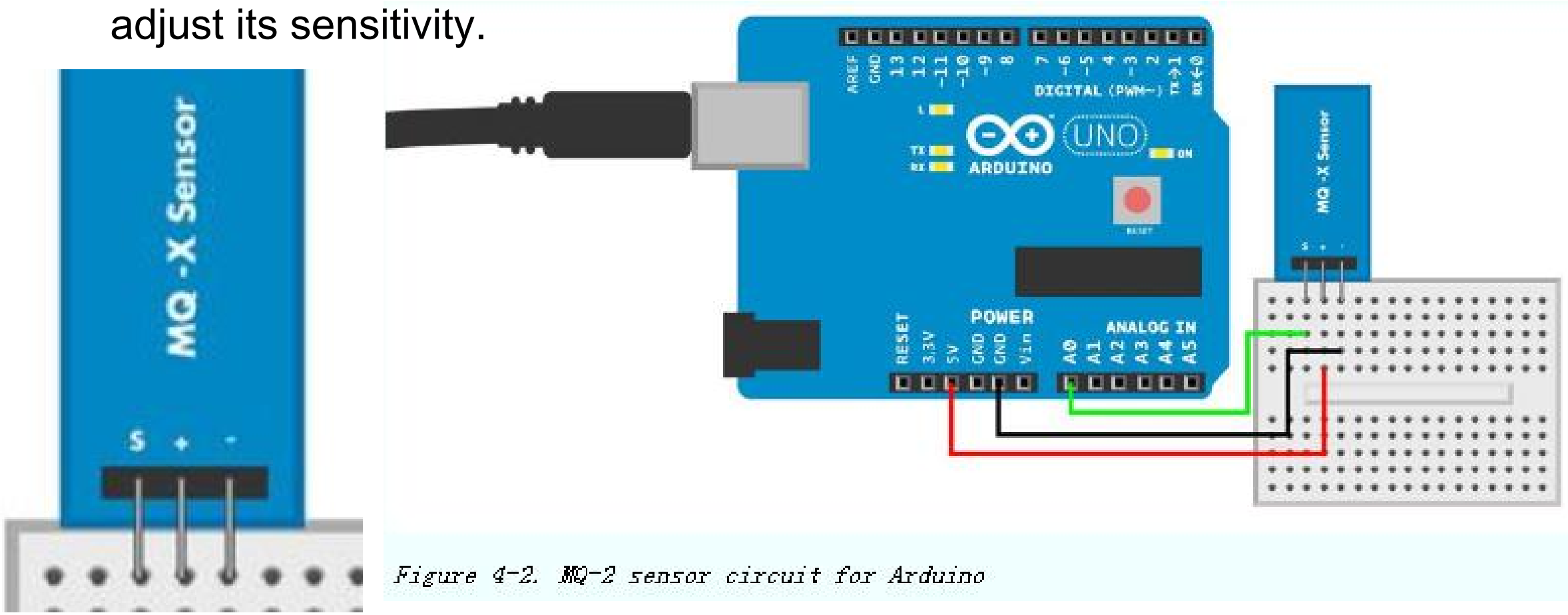


Figure 4-2. MQ-2 sensor circuit for Arduino

MQ-2 Code and Connection for Arduino

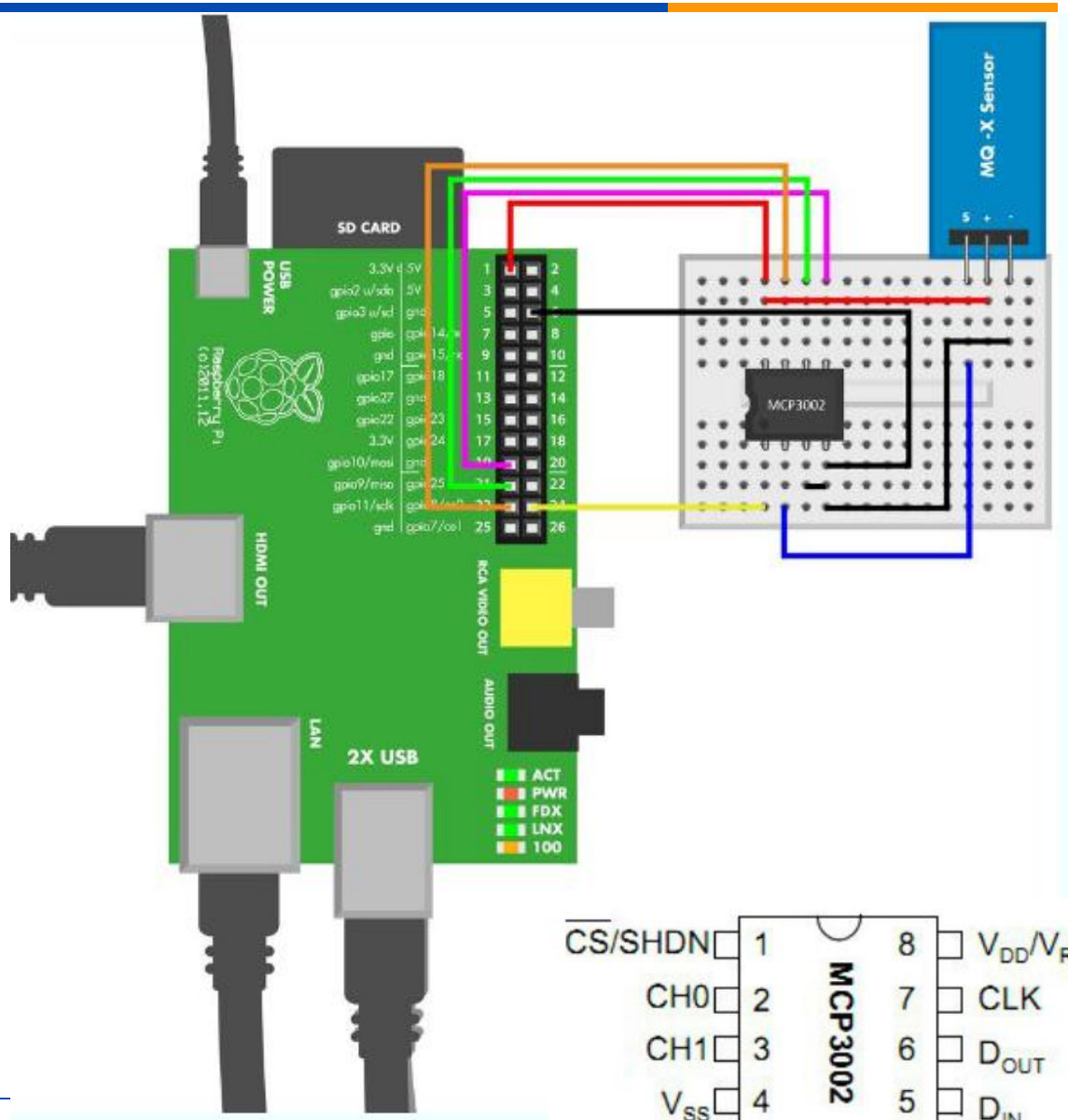
Example 4-1. mq_x_smoke_sensor.ino

```
// mq_x_smoke_sensor.ino - print smoke level to serial
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

const int sensorPin = A0;
int smoke_level = -1;  // 将烟雾浓度初始化一个传感器不可能获得的数值，如果输出-1，说明未读取值
void setup() {
        Serial.begin(115200); // bit/s
        pinMode(sensorPin, INPUT);
}
void loop() {
        smoke_level = analogRead(sensorPin);  //  MQ-2是模拟电阻传感器，读出值在0~1023(0-5V)
        Serial.println(smoke_level);
        if(smoke_level > 120) {     // 在代码中设置临界值，也可以通过传感器上的电位器调整灵敏度
                Serial.println("Smoke detected");
        }
        delay(100); // ms
}
```

MQ-2 Code and Connection for Raspberry Pi

- The Raspberry Pi needs an external analog-to-digital converter (ADC) to read the MQ-2.
- Similar to other analog resistance sensors (see Compound Eye Code and Connections for Raspberry Pi), you can use a cheap MCP3002 chip for this conversion.



ure 4-4. MQ-2 sensor circuit for Raspberry Pi

MQ-2 Code and Connection for Raspberry Pi

Example 4-2. mq_x_smoke_sensor.py

```
# mq_x_smoke_sensor.py - print smoke level
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

import time
import botbook_mcp3002 as mcp   # ❶
smokeLevel = 0
def readSmokeLevel():
        global smokeLevel
        smokeLevel = mcp.readAnalog()   # ❷

def main():
        while True:   # ❸
                readSmokeLevel()    # ❹
                print("Current smoke level is %i " % smokeLevel)    # ❺
                if smokeLevel > 120:
                        print("Smoke detected")
                time.sleep(0.5)      # s
if __name__ == "__main__":
        main()
```

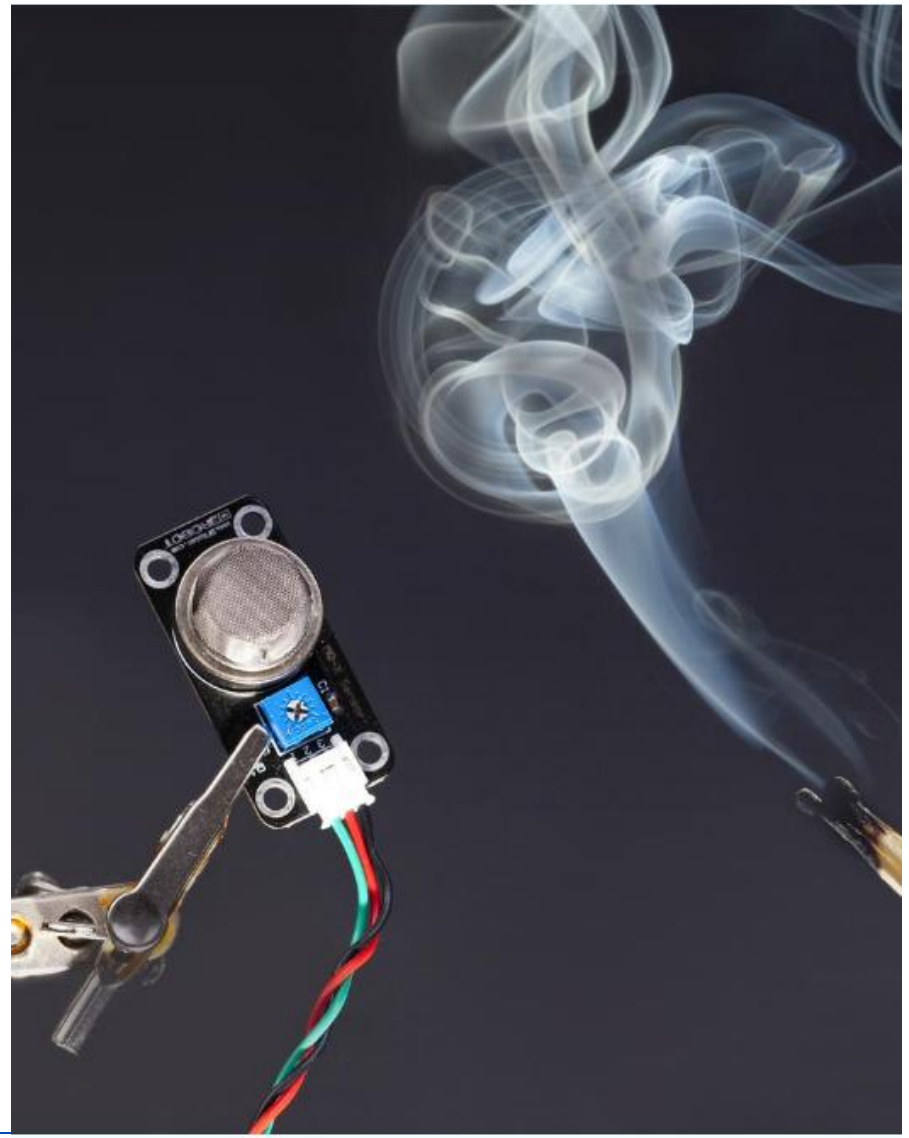


Figure 4-5. Smoke goes up

Contents

Experiment: Breathalyzer (Alcohol Sensor MQ-303A)

- A Breathalyzer is used for checking whether a person has alcohol in his blood. More specifically, ethanol, the alcohol found in wine, beer, and liquor.
- The MQ-3 alcohol sensor looks a lot like the other MQ series gas sensors. (see Figure).
- Just as the gas exchange in lungs brings in oxygen and gets rid of carbon dioxide, some blood alcohol is released in the air you exhale. This is the alcohol measured by an alcometer.
- The more ethanol in your blood, the more there is in the air you exhale. Blood alcohol content gives a good indication how drunk a person is.

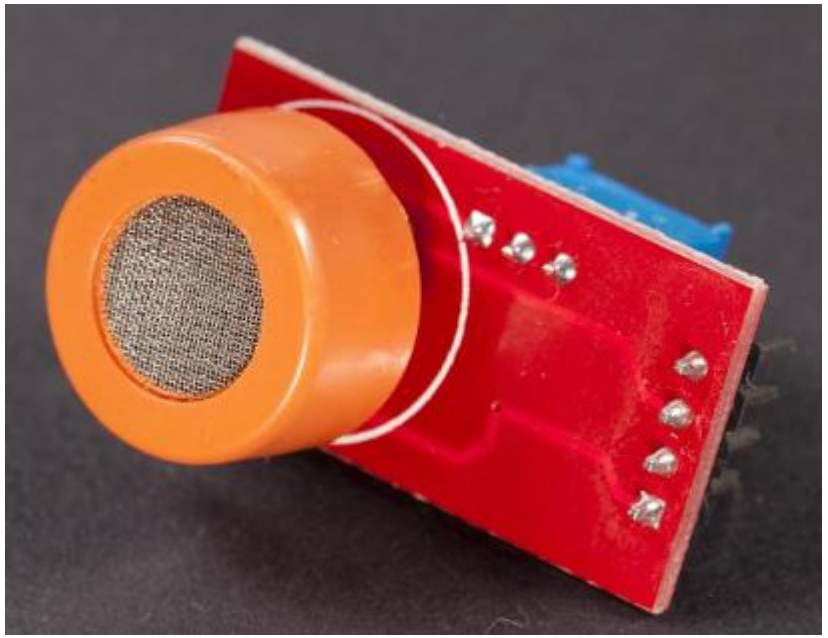


Figure 4-6. MQ-3 alcohol sensor

Experiment: Breathalyzer (Alcohol Sensor MQ-303A)

- Even though more alcohol in blood makes the same person more drunk, the drunkenness differs from person to person.
- For making laws, a typical value is chosen as the limit. For example, the limit for DUI charges in Finland is 0.5%.
- Officially accepted alcometers are calibrated periodically to get reliable readings.
- Alcometers use a built-in formula to estimate blood alcohol content from exhaled air alcohol content.

Experiment: Breathalyzer (Alcohol Sensor MQ-303A)

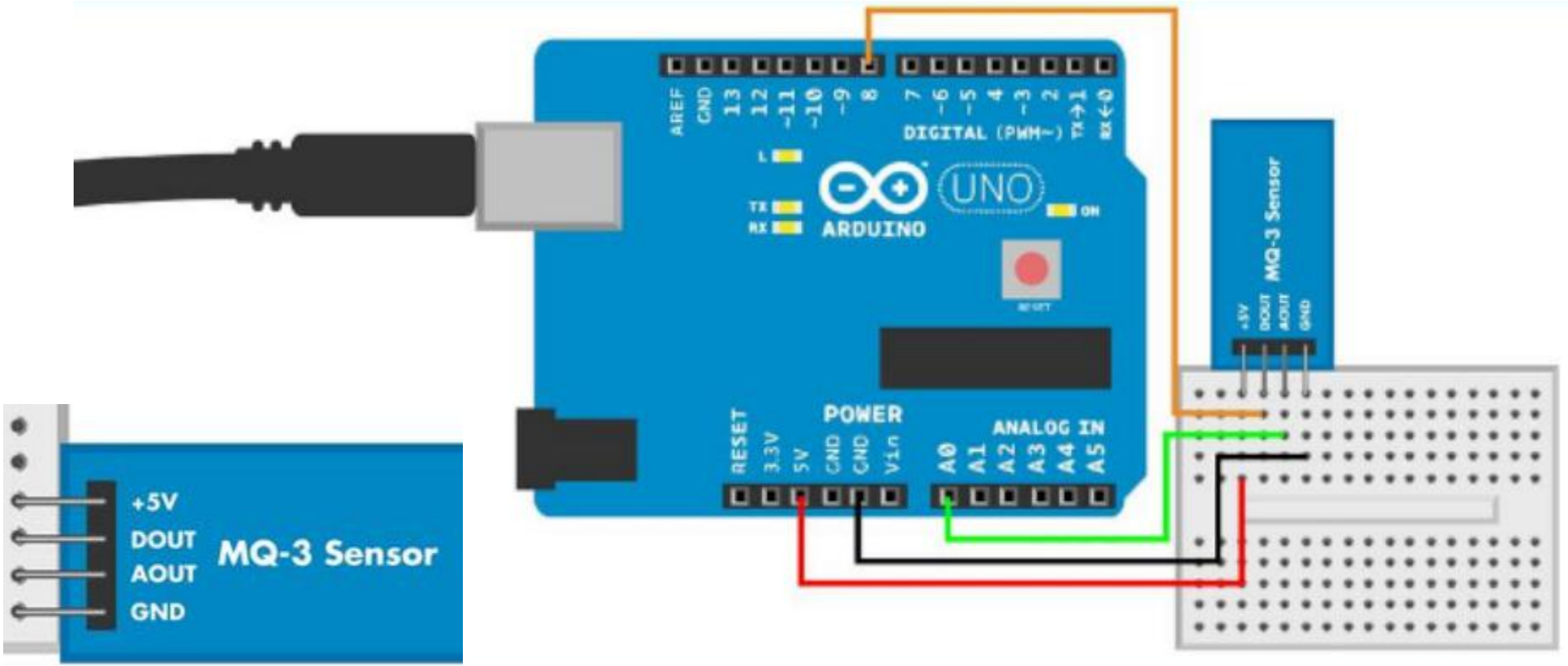


Figure 4-7. MQ-3 sensor circuit for Arduino

Experiment: Breathalyzer (Alcohol Sensor MQ-303A)

Example 4-3. mq_3_alcohol_sensor.ino

```
// mq_3_alcohol_sensor.ino - print alcohol value and limit digital info.
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

const int analogPin = A0;
const int digitalPin = 8;
Int limit = -1;
int value = 0;
void setup() {
        Serial.begin(115200);
        pinMode(digitalPin,INPUT);
}
void loop(){
        value = analogRead(analogPin);          //Read analog value
        limit = digitalRead(digitalPin);        //Check if alcohol limit is breached
        Serial.print("Alcohol value: ");
        Serial.print(value);
        Serial.print(" Limit: ");
        Serial.println(limit);
        delay(100);
}
```

Experiment: Breathalyzer (Alcohol Sensor MQ-303A)

Example 4-4. mq_3_alcohol_sensor.py

```
# mq_3_alcohol_sensor.py - read digital output from alcohol sensor
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

import time
import botbook_gpio as gpio
def readLimit():
        limitPin = 23
        gpio.setMode(limitPin,"in")
        return gpio.read(limitPin)
def main():
        while True:
                if readLimit() == gpio.HIGH:
                print("Limit breached!")
                time.sleep(0.5)
if __name__ == "__main__":
        main()
```

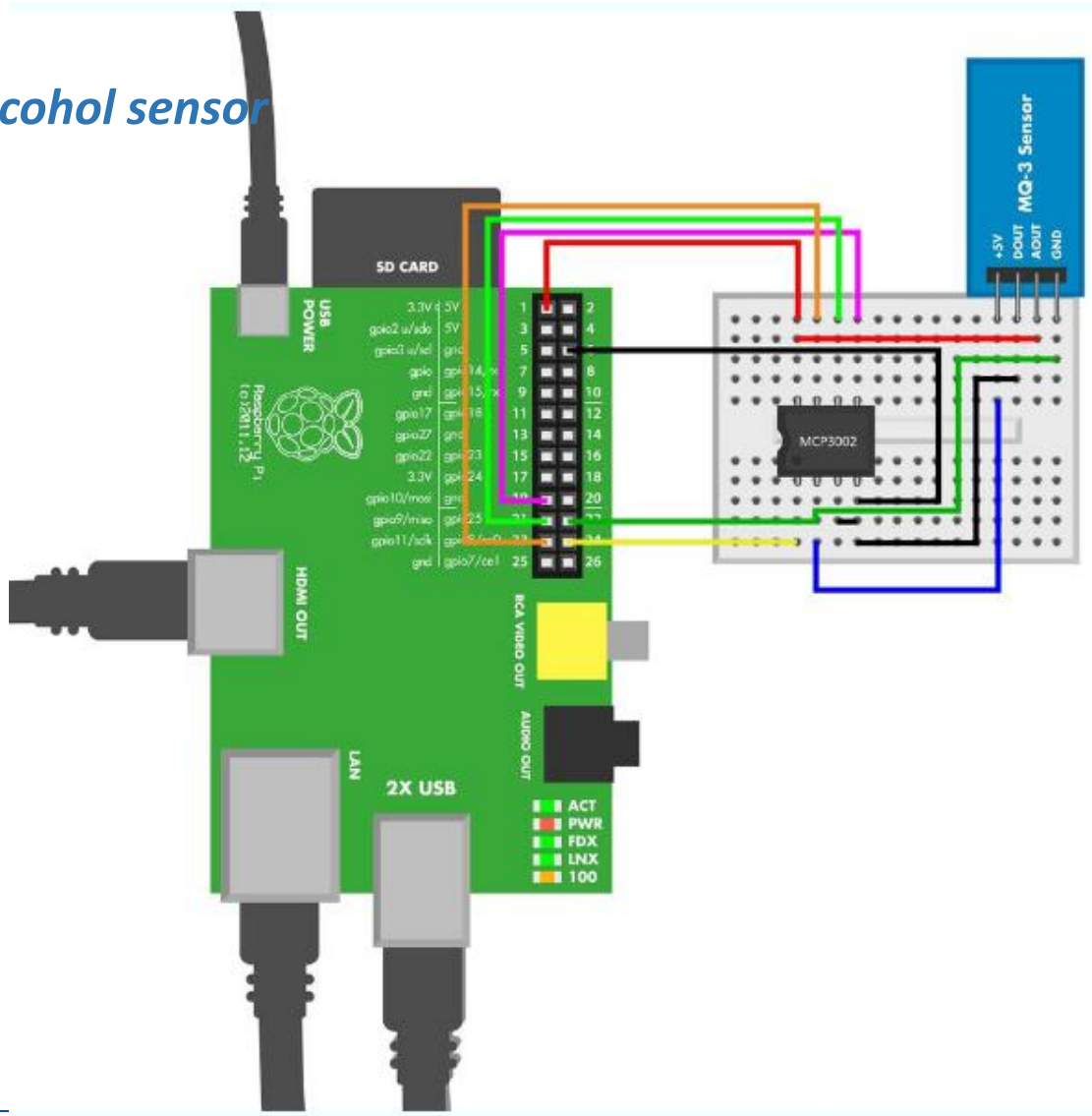
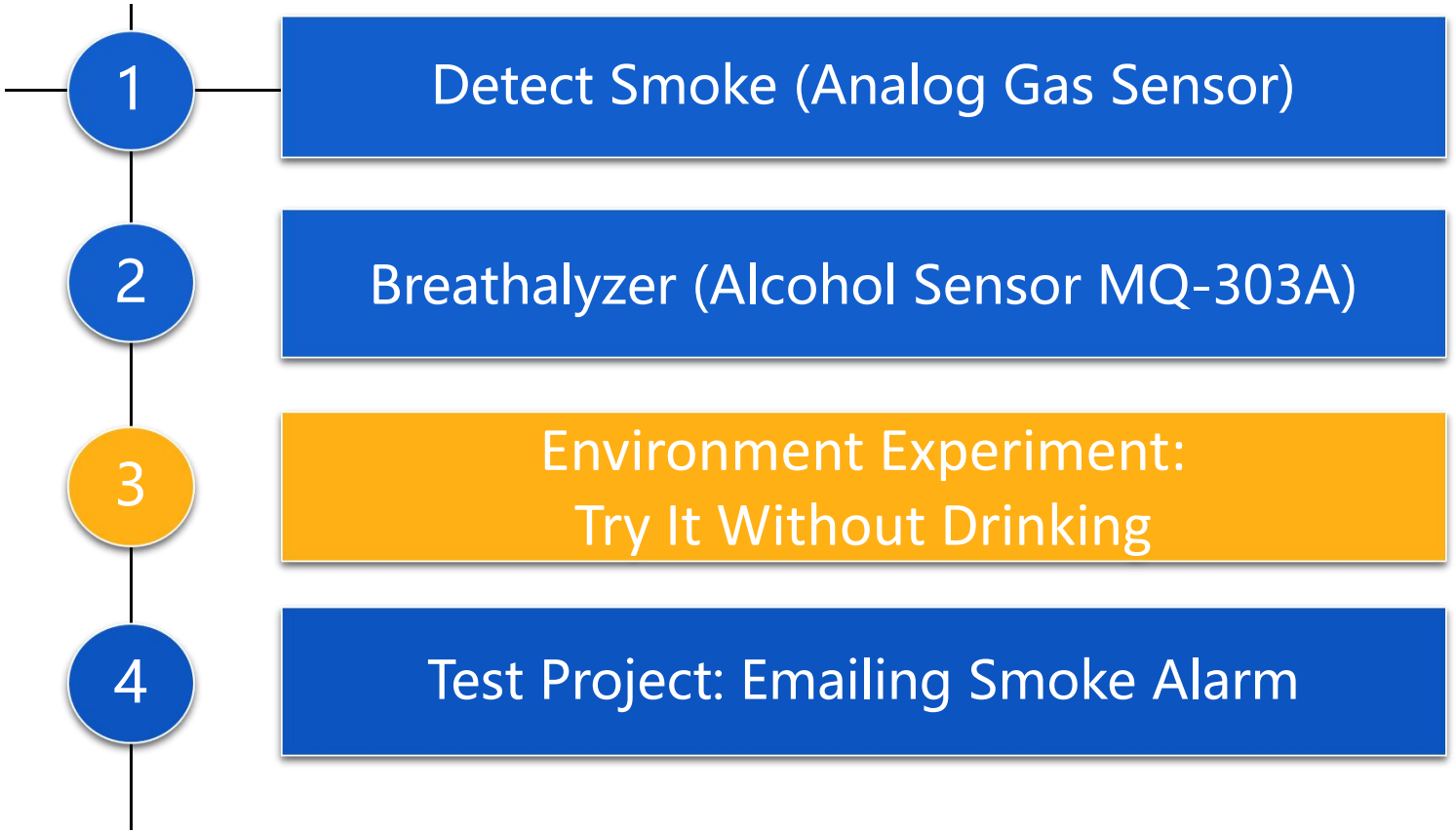


Figure 4-8. MQ-3 sensor circuit for Raspberry Pi

Contents


1
Detect Smoke (Analog Gas Sensor)
2
Breathalyzer (Alcohol Sensor MQ-303A)
3
Environment Experiment:
Try It Without Drinking
4
Test Project: Emailing Smoke Alarm

Environment Experiment: Try It Without Drinking

Contents

Test Project: Emailing Smoke Alarm

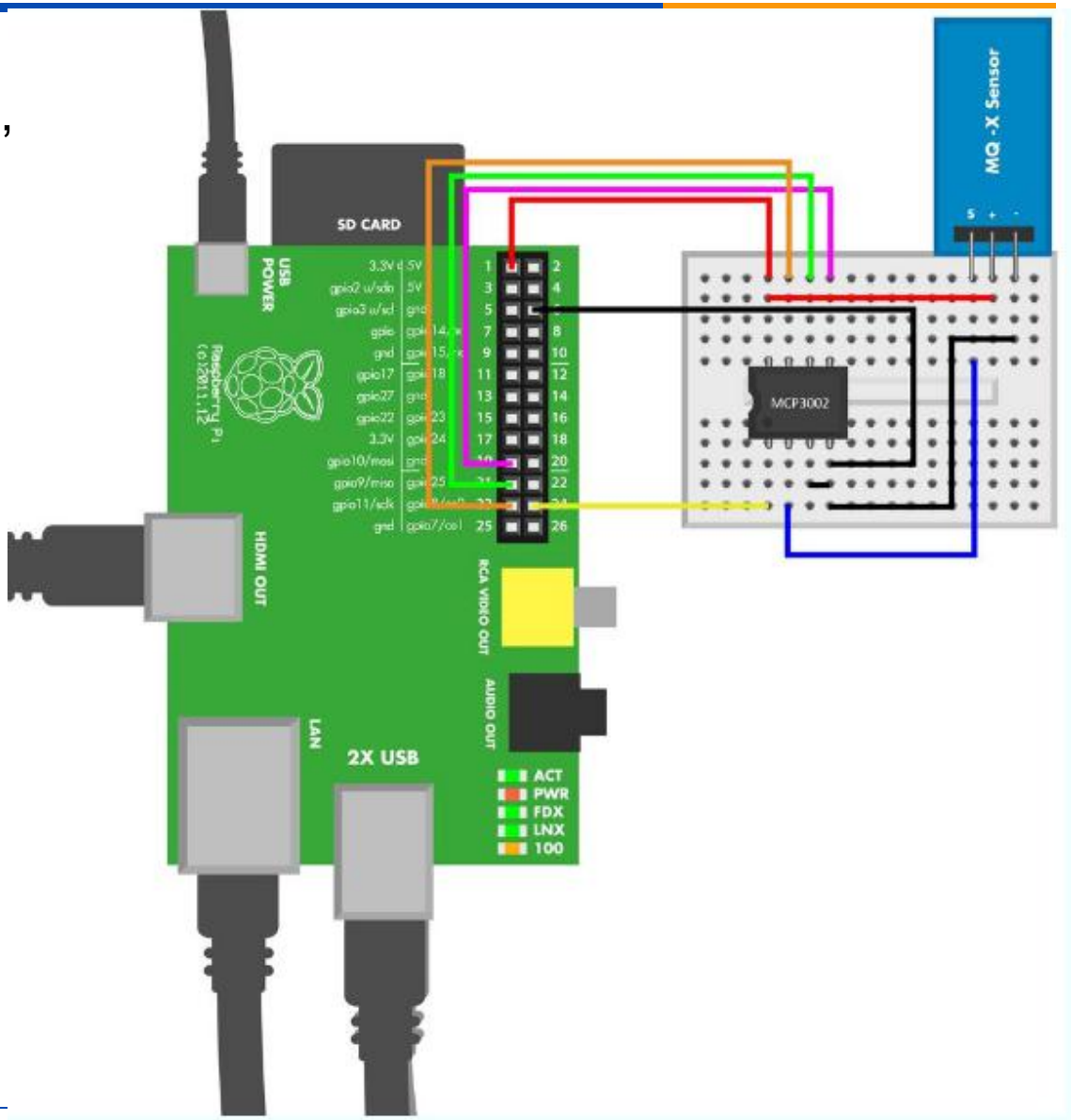
- **In this project, you'll learn:**
  - How to react to environment changes with actions: if there's smoke, send an email warning.
  - How to automatically send email from Raspberry Pi.
  - The basics of sending email.

Python for Email and Social Media

- This is an example of a project that's easy with Raspberry Pi. Sending email automatically from Raspberry Pi is no different from automatically emailing on any Linux system.
- Python is known for having "batteries included," in that there is a library for everything. Email sending is trivial using existing libraries. Similarly, there are libraries for sending and receiving data over the Web using the HTTP protocol.
- What about social media? You could adapt the program to send messages to Twitter, Facebook, or similar "social media" services. But all of these use protocols that though they are built on open technologies, are proprietary and create lock-in to their services. Those giants have rules on how they run their service—rules that could unilaterally change any day.

Building It

- Connect the MQ-2 smoke sensor to Raspberry Pi, as shown in MQ-2 Code and Connection for Raspberry Pi. It’s a good idea to test the smoke sensor with just the sensor code (Example 4-2) first.
- Once you’re sure it works, you can advance to sending emails in your code.
- Test your email credentials normally before using them in this program.（用户名、密码）



ure 4-4. MQ-2 sensor circuit for Raspberry Pi

pay attention

- Don’t use your personal email account for this kind of testing.
- If you make a mistake in your code and send out too many email messages, your email server may interpret this as an attempt at sending spam.
- It’s best to create an email account just for these purposes.

How Does Email Work?

- It’s essential to remember that sending and receiving email involves two different servers (the sender’s and the recipient’s). It’s common that the sending and receiving server are in different networks, even on different continents.
- When you click Send in your Mail User Agent, your computer contacts your SMTP server. The SMTP can be in your local network, run by your ISP (such as smtp.verizon.net) and may accept all connections originating within its network without password or login. Or it could be run by your email provider (such as smtp.gmail.com), and require TLS/SSL encryption and login. When your email is accepted, the SMTP server will deliver your message to the recipient.

How Does Email Work?

- When your recipient checks her mail, her Mail User Agent contacts an IMAP server. This is run by her email provider (such as Gmail or GMX) or her ISP.
- Reading private email of course requires a login and password, and common sense dictates that one should use encryption (SSL/TLS) when connecting (most email providers require encryption). From the IMAP server, a mail user agent can download message headers and full copies of messages.
- And when someone sends you an email, the reverse happens: her mail user agent contacts her SMTP server, delivers it to your mail server, and when you check your mail, you retrieve your copy of the email from your IMAP server.
- Messages are typically left on IMAP server, which is why you can check your mail on your computer and mobile phone and see the same list of messages on both.

Could Arduino Send Email? Not Easily

- A typical way to build a similar project with Arduino would use an external computer.
- But even with an Ethernet or WiFi shield, it would be difficult for you to send mail from Arduino.
- Although the email protocol is, on its surface, fairly simple, there are enough possible surprises when sending an email that it would be hard to write a reliable SMTP library for Arduino (also, many servers require SSL, and the Arduino Uno and similar models don't support this).
- Obviously, it is easier to do this project with Raspberry Pi, or a similarly capable microcontroller board such as the BeagleBone (or the Arduino/Linux hybrid Arduino Yún).

Code for Raspberry Pi

Example 4-5. smoke_alarm.py

```
# smoke_alarm.py - send email every 5 minutes when smoke is detected
# (c) BotBook.com - Karvinen, Karvinen, Valtokari

import time
import botbook_mcp3002 as mcp
import smtplib                                  # ❶ “内置电池”： Python内置了SMTP通信协议库。你不需要手工构建底层的套接字。

from email.mime.text import MIMEText            # ❷ 电子邮件最初是纯文本的。每一行一个标头，然后跟一个空白行，最后是文本。
                                                # 但是现在我们都希望文本内容本地化，即支持非ASCII字符。这就是为什么邮件需要MIME编码，而邮件的内容就像附件一样。

# Email addresses
email_to = 'example@gmail.com'                  # ❸ 收件人地址，即接收烟雾警报的电子邮件地址。这些都是全局变量，所有函数都可以访问。
email_from = 'example@gmail.com'                # ❹ 发件入地址，理论上可以是任何地址。但若随意填写就可能被垃圾邮件过滤器视为拉圾部件。
```

Code for Raspberry Pi

```
# SMTP email server settings
server = 'smtp.gmail.com'   # ❺ SMTP是发送邮件的服务器。本程序仅发送邮件而不读取邮件列表，因此仅使用SMTP服务器。
mail_port = 587
user = 'example@gmail.com'          # ❻ 登录SMTP的用户名。它应该是你的电子邮件地址（exampled@gmail．com）或是一个简单的登录名
password = 'password'               # ❼ 你的SMTP密码应该与登录网页邮件客户端的密码或连接邮件客户端的密码相同。
gracePeriod = 5 * 60 # seconds      # ❽ 避免过分频繁发送邮件，设置缓冲期

def sendEmail(subject, msg):        # ❾ 发送电子部件的函数。函放的参数分别是标题（ subject，邮件的主题）和消息（msg，件的内容）。
    msg = MIMEText(msg)             # ❿ 邮件的文本采用MIME编码，它可以有效地处理非ASCII字符。
    msg['Subject'] = subject        # ⓫ 为了修改电子邮件的标头，你需要使用字典数据类型的方法修改msg这个MIMETEXT对象。
    msg['To'] = email_to            # ⓬ 收件人和发送人信息来自全局变量。
    msg['From'] = email_from
    smtp = smtplib.SMTP(server,mail_port)    # ⓭创建SMTP类的对象，用来连接SMTP服务器。
    smtp.starttls()
    smtp.login(user, password)               # ⓮使用全局变显 username和 password连接SMTP服务器。
    smtp.sendmail(email_from, email_to, msg.as_string())        # ⓯发送邮件。
    smtp.quit ()                                                # ⓰ 关闭连接并清理资源。
```

Code for Raspberry Pi

```
def main():
    while True:
        smokeLevel = mcp.readAnalog()
        print("Current smoke level is %i " % smokeLevel)
        if smokeLevel > 120:
            print("Smoke detected")
            sendEmail("Smoke","Smoke level was %i" % smokeLevel) #
            time.sleep(gracePeriod) #
        time.sleep(0.5) # s     #

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

Packaging

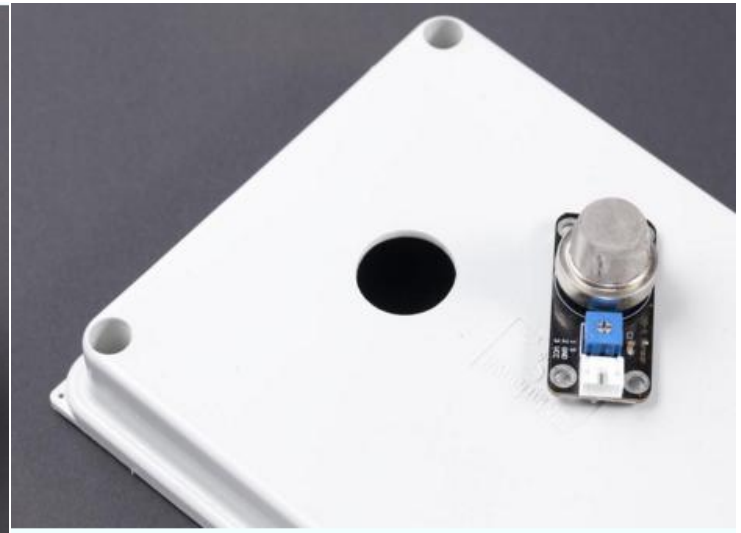


Figure 4-11. Hole for the smoke sensor

Figure 4-12. Sensor in place

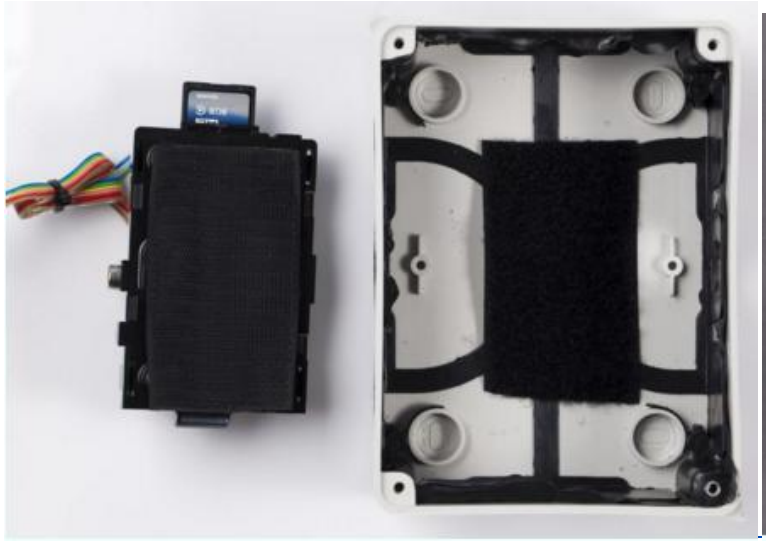


Figure 4-13. Raspberry Pi's Velcro attachement

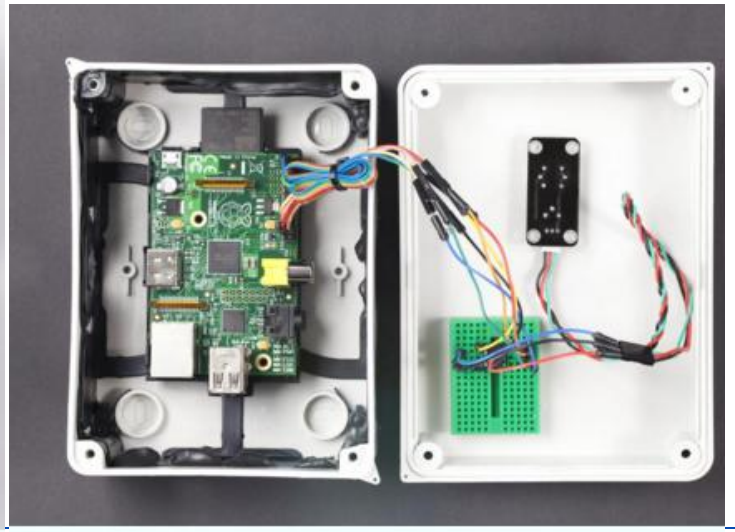


Figure 4-14. Mini breadboard with zip-tied wires

Figure 4-16. Cables attached