東北大學 国家示范性軟件学院

SOFTWARE COLLEGE OF NORTHEASTERN UNIVERSITY

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

Identity

Identity

- Who are you?
- What object is sitting on top of the sensor?
- Identification sensors identify people and objects.
- In this chapter, you’ll learn to identify objects with RFID and people with fingerprints and keypads.

Identity —— RFID

- RFID (radio frequency identification) has long been hailed as the next big thing in identification. In reality, it’s slowly been becoming part of everyday life.
- RFID has largely replaced bar codes in warehouses, but bar codes are still used with individual consumer packages of products.
- RFID can be read from a distance, and it can store more data than a bar code.
    - For example, a bar code identifies a product as “1 liter of Foobar, Inc. milk.” An RFID tag can also tell you that this specific carton was carton number 12,209,312, uniquely identifying it.

Identity —— Biometric identification

- Biometric identification is getting commonplace. Many laptops have a fingerprint reader to protect you from bystanders “shoulder-surfing” as you type your password.
- Most governments want to store fingerprints and digitally readable facial images of every citizen with the excuse of using them in passports.
- The digital passports also have a standard for storing an iris image, even though that’s not used yet.
- DNA markers are already used for identifying criminals.

Identity

- The benefit of biometric identification is that it's always with you. The biggest downside is that once it's copied by the adversary, it's not possible to change it. For example, you can't change your fingerprints.
- In practice, many cheap biometric sensors can be easily misled. There have also been claims that even professional, manual fingerprint identification is not as reliable as has been assumed.
- Keypad is likely the most common method for identification. Just think how many times a week you type your PIN into different devices.

Contents

1. Keypad
2. th MX2125
3. Acd Gyro Together
4. Expeii Nunchuk
5. Test Proontrolled by Wii Nunchuk

Keypad

- A keypad is a quick way to type in some numbers.
- If you used a cell phone when they still had keypads, you might remember that keypads work well with one-handed use, and some people could even type numbers (or text!) without looking at the phone.
- You have similar keypads in television remote controls and microwave oven controls.
- Keypad are also used for password entry. You have probably used one with an ATM, numeric door lock, or a burglar alarm.

Keypad

- This experiment uses inexpensive numeric keypad shown in Figure 9-1.
- The wiring used is similar to many other cheap number pads.
- For Raspberry Pi, you could just use a normal USB keyboard or a USB number pad.

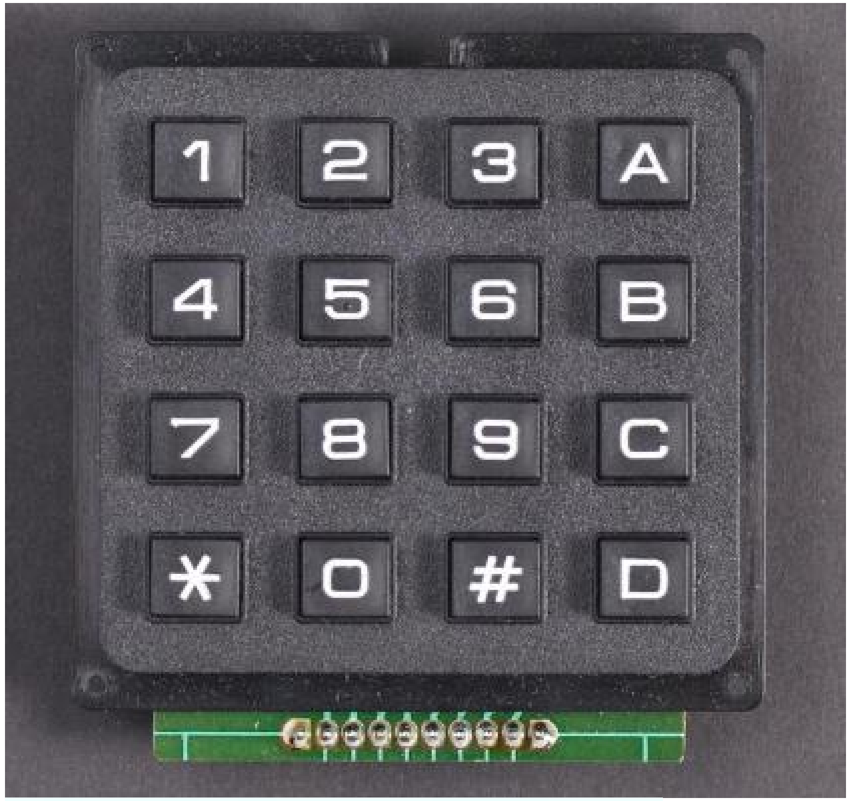
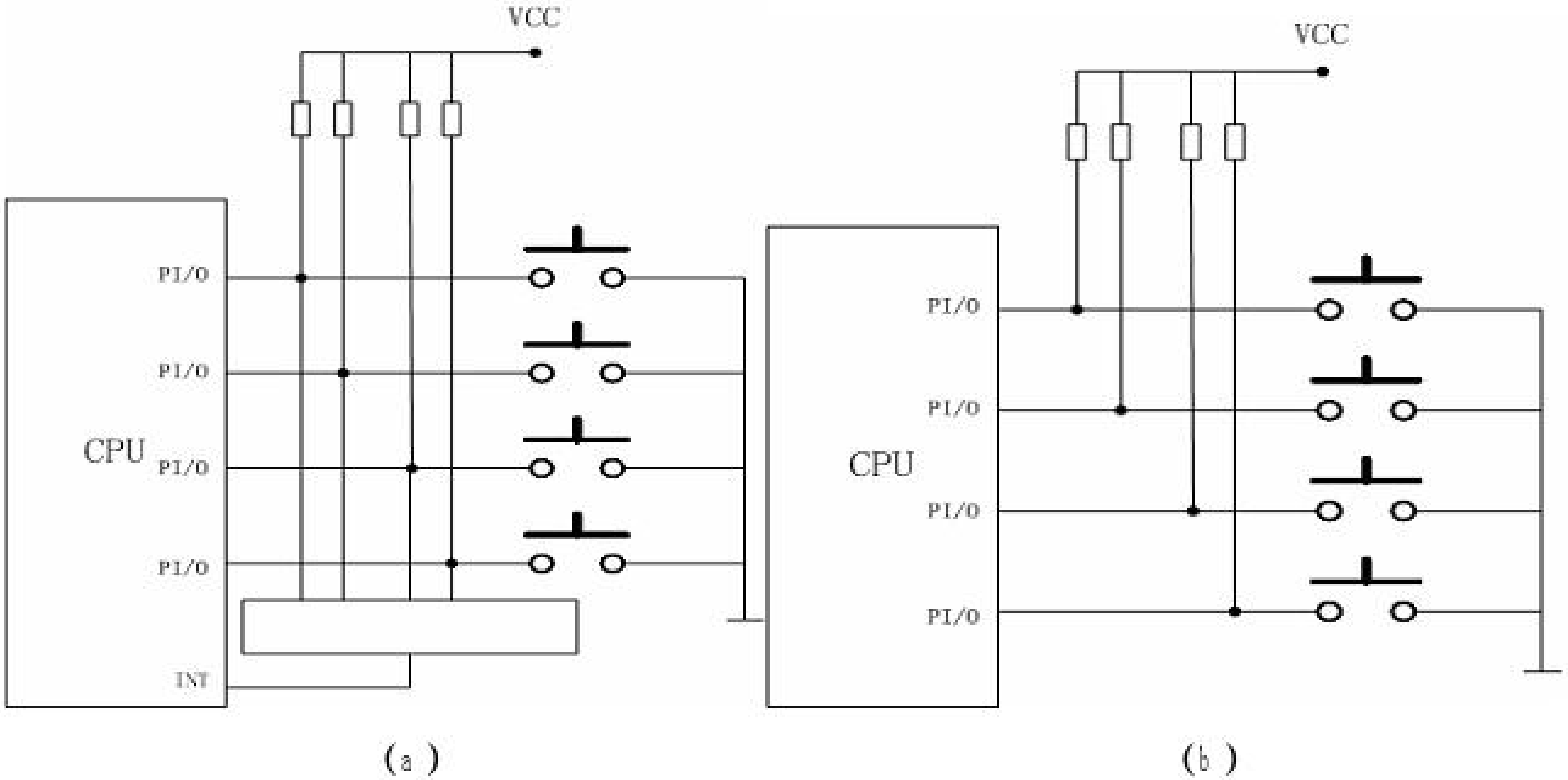


Figure 9-1. 16-key numeric keypad

Keypad

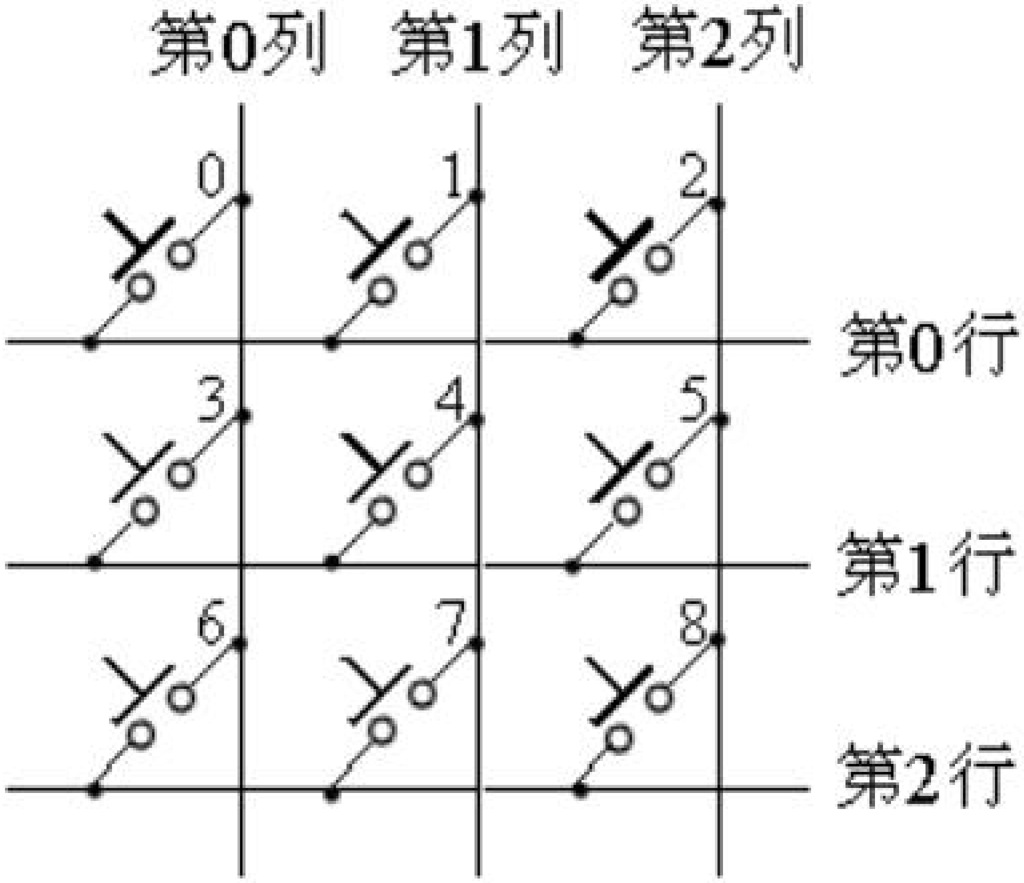
- Under each key, vertical and horizontal wires cross. When you press a key, it connects the wires at each crossing.
- To find out which key is pressed, you need to take one of the vertical columns LOW, but leave all the other columns HIGH. If one of the horizontal lines goes LOW, the key that was pressed is the one where the HIGH lines cross. If there is no match, test the rest of the columns until you find a key that was pressed. The code keeps testing columns, returning to the first column to check for more keypresses.

Keypad



Keypad

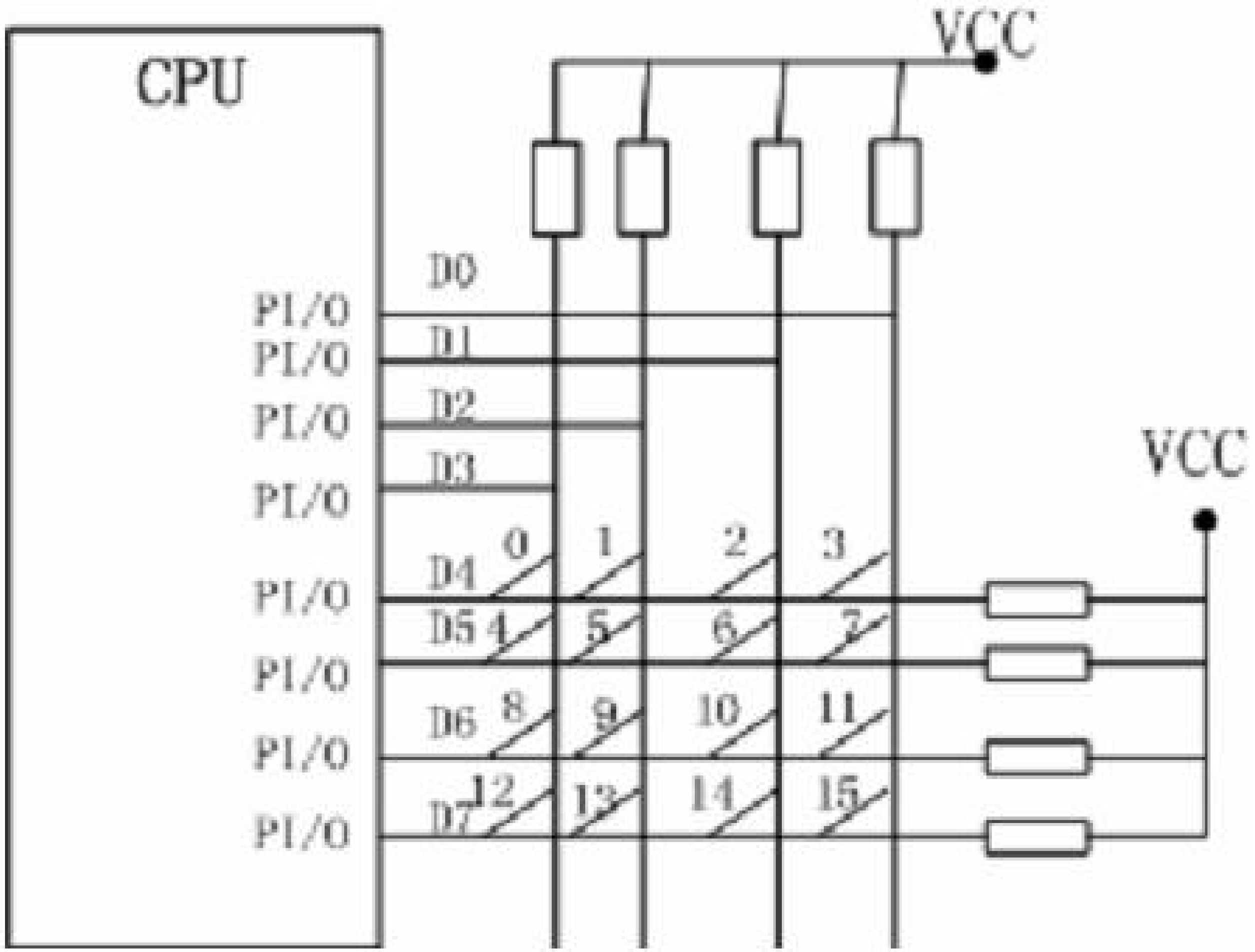
- **按键接口设计**
  - 矩阵式键盘接口设计：矩阵式键盘是指键开关按行列排列,形成二维矩阵的结构；



矩阵结构

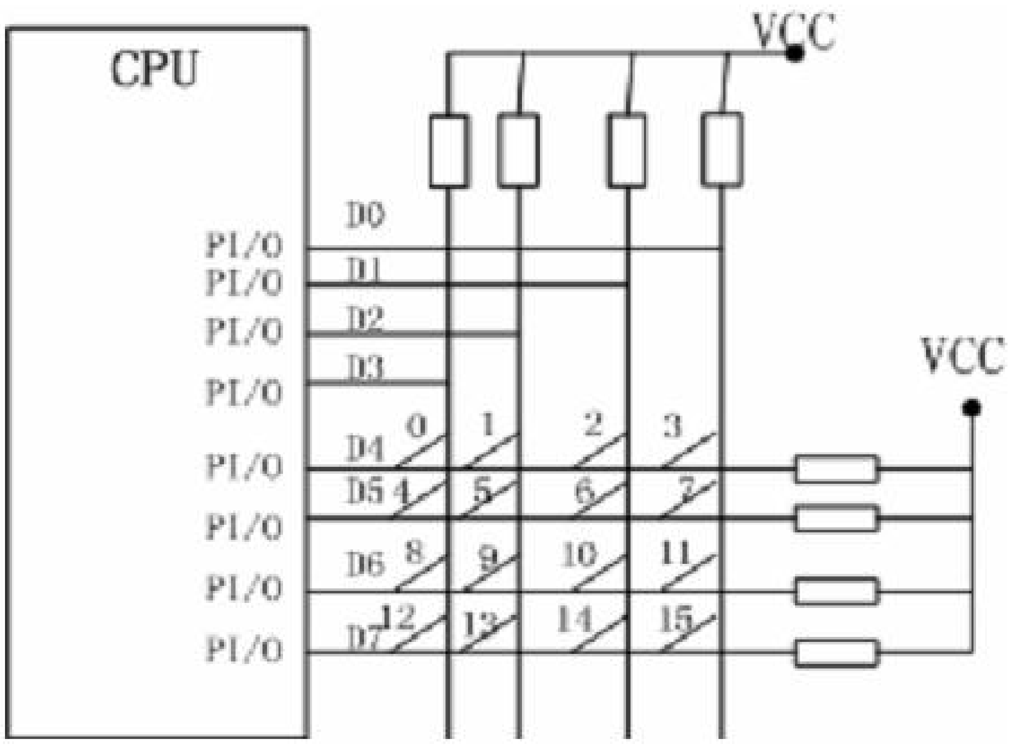
Keypad

- **按键接口设计**
  - 矩阵式键盘接口设计：矩阵式键盘是指键开关按行列排列,形成二维矩阵的结构；



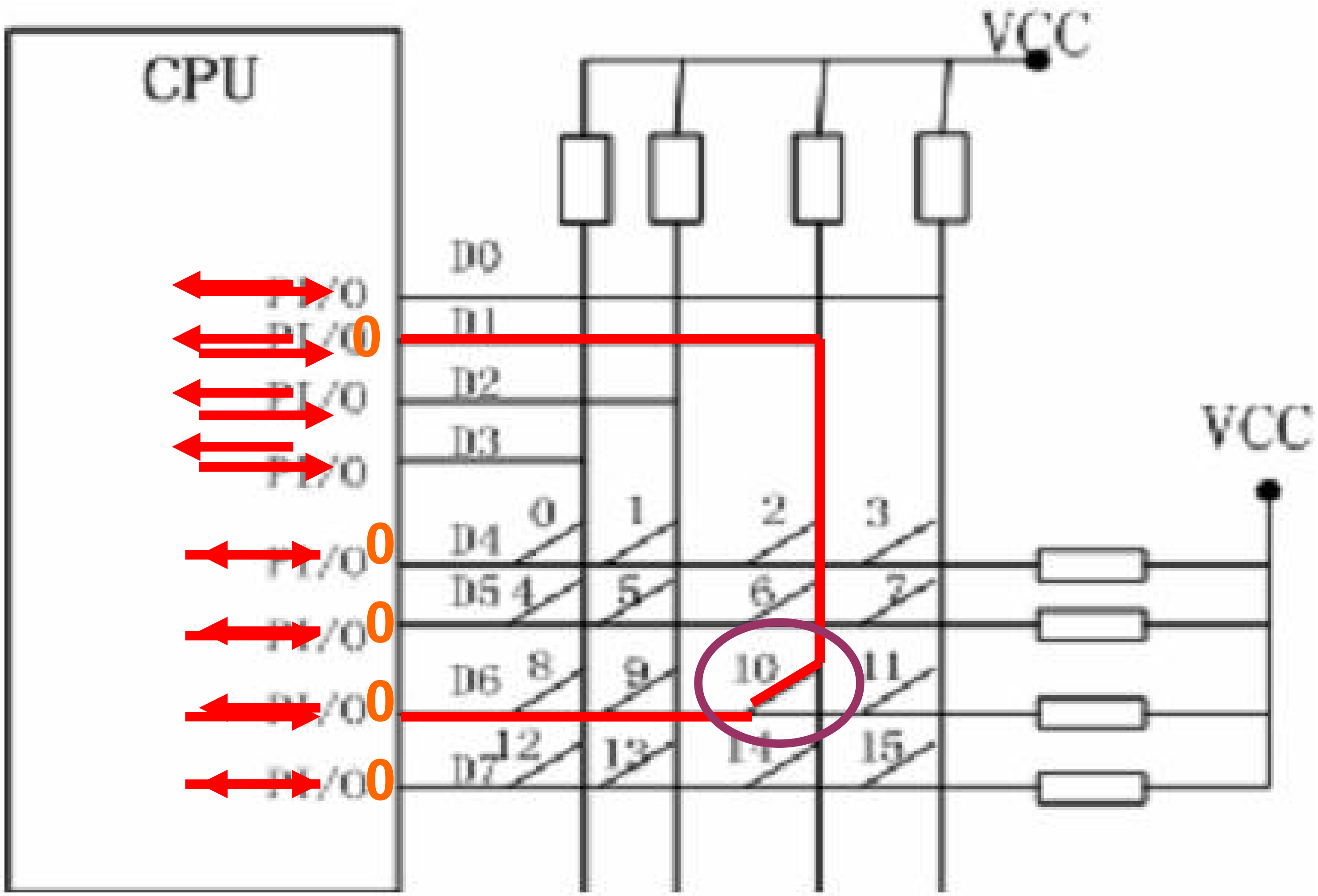
Keypad

- **按键识别（行列扫描法）**
  - □ 通过计数译码，依次将各行输出为0，其余作为1。在扫描每一行时，读列线，若全为1，说明此行无键按下，若某一列为0，说明有键按下，且行号和列号已经确定。然后用同样的办法，依次向列线扫描输出，读行线。如果两次所得的行号和列号分别相同，则确定了闭合键的键码。



Keypad

- 按键识别（行列反转法）



Keypad Code and Connection for Arduino

- Figure 9-2 shows the connections for the keypad and Arduino. Wire it up as shown, and run the sketch shown in Example 9-1.

  - Arduino has internal pull-up resistors.
  - When a digital pin is in INPUT mode, digitalWrite(pin, HIGH) connects it to +5 V through 20 kOhm resistor.

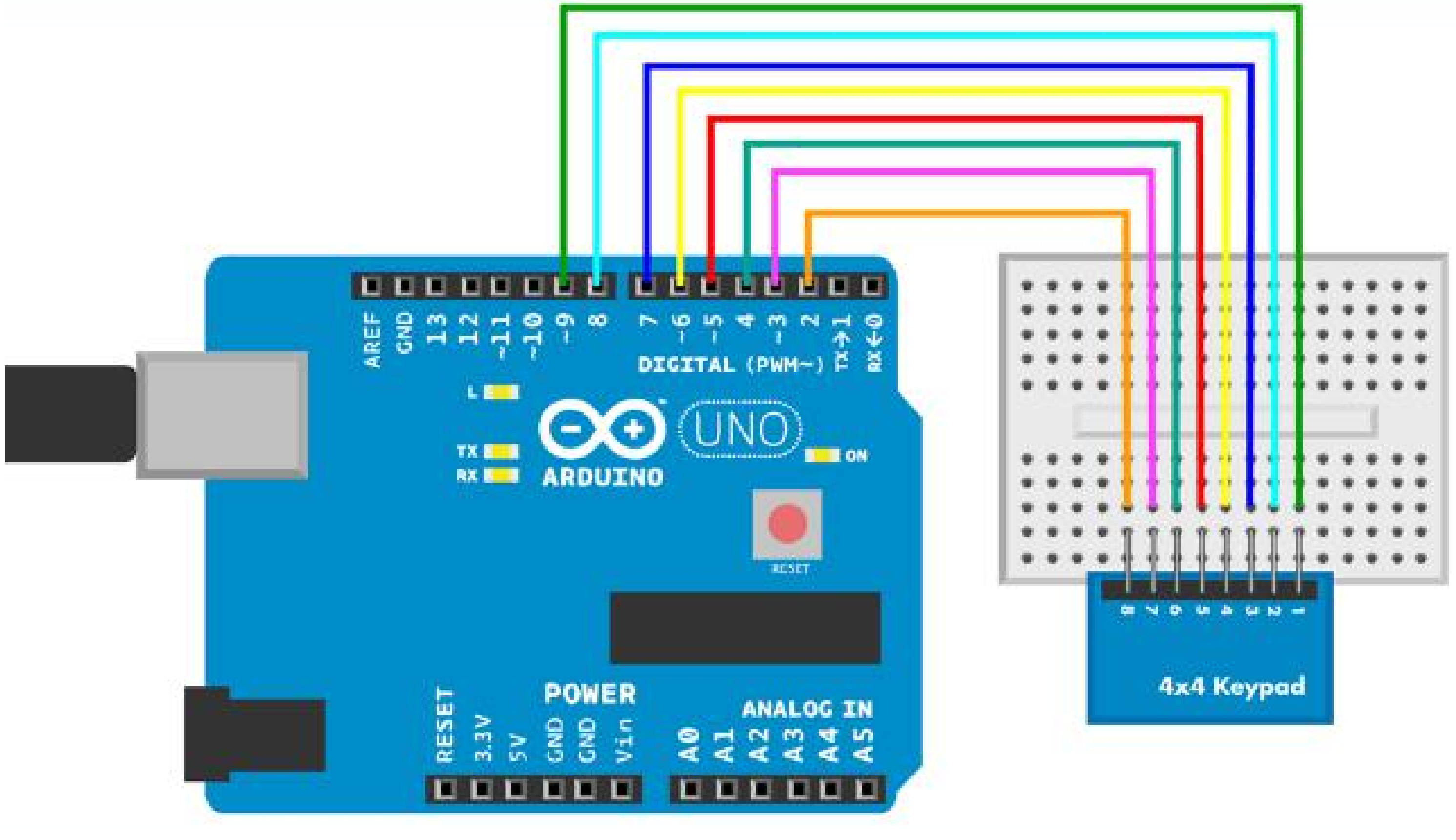


Figure 9-2. Keypad Arduino connections

Example 9-1. keypad.ino

```
// keypad.ino - read 16-key numeric keypad (dx.com sku 149608)
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

const int count = 4;                  //① 垂直和水平方向的按键数量。显然这是一个 4x 4的键盘
char keymap[count][count] = {         //② 变量keymap是二维数组
        {'1', '2', '3', 'A'},         // 将按键的位置(如第0列第1行)映射到按键值(“4”)
        {'4', '5', '6', 'B'},
        {'7', '8', '9', 'C'},
        {'*', '0', '#', 'D'}
};
const char noKey = 'n';
byte columns[count] = {9, 8, 7, 6};        // ③将列映射到Arduino的数字引脚9,8,7,6
byte rows[count] = {5, 4, 3, 2};           //   将行映射到Arduino的数字引脚5,4,3,2
unsigned int lastReadTime;
unsigned int bounceTime = 30;              // ms
```

Example 9-1. keypad.ino

```
// keypad.ino - read 16-key numeric keypad (dx.com sku 149608)
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

void setup(){
        Serial.begin(115200);
        lastReadTime = millis();
        for(int i = 0; i < count; i++)  {
                pinMode(rows[i], INPUT);
                digitalWrite(rows[i], HIGH); //为每一行设置上拉电阻，即使引脚悬空也依然为高电平
                pinMode(columns[i], INPUT);
        }
}
```

Example 9-1. keypad.ino

```
// keypad.ino - read 16-key numeric keypad (dx.com sku 149608)
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

void loop(){
        char key = getKeyPress();
        if(key != noKey) {
                Serial.print(key);
        }
        delay(100);
}
```

Example 9-1. keypad.ino

```
// keypad.ino - read 16-key numeric keypad (dx.com sku 149608)
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

//函数不支持检测多个按键按下，只返回第一个被按下的按键
char getKeyPress(){
    char foundKey = noKey;
    if((millis() - lastReadTime) > bounceTime) {  //在一定时间间隔之后才识别按键
        //向列发送脉冲，然后检测行
        for(int c = 0; c < count; c++) {
            //发送脉冲
            pinMode(columns[c], OUTPUT);
            // 设置当前(垂直)列为低电平，其他列为高电平
            digitalWrite(columns[c], LOW);
```

Example 9-1. keypad.ino

```
// keypad.ino - read 16-key numeric keypad (dx.com sku 149608)
// (c) BotBook.com - Karvinen, Karvinen, Valtokari

//读取行
for(int r = 0; r < count; r++) {
    if(digitalRead(rows[r]) == LOW) {      //如果本行是低电平
    // 那么低电平的列和低电 平的行所在的交点就是按键按下的位置
        foundKey = keymap[r][c];
    }
}
digitalWrite(columns[c], HIGH);
```

Example 9-1. keypad.ino

```
// keypad.ino - read 16-key numeric keypad (dx.com sku 149608)
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
```

```
            digitalWrite(columns[c], HIGH);
            pinMode(columns[c], INPUT);
            //通过测试发现，当列处于output模式时会干扰读数，所以将其设置为高阻抗的input模式
            if(foundKey != noKey){
                break;
            }
        }
        lastReadTime = millis();
    }
    return foundKey;
}
```

Keypad Code and Connection for Raspberry Pi

- Figure 9-3 shows the wiring diagram for Raspberry Pi.
- Hook everything up as shown, and run the program in Example 9-2.

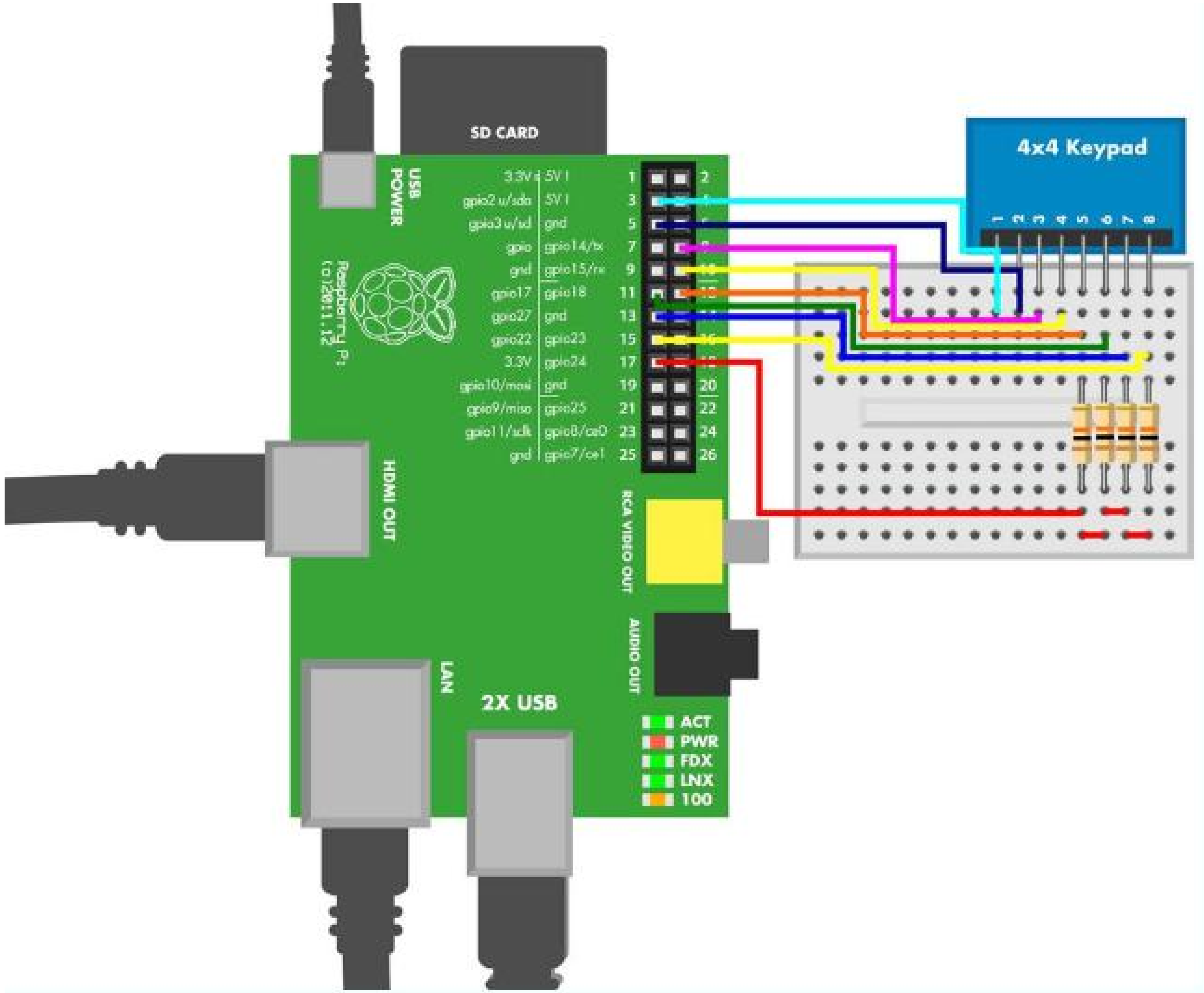


Figure 9-3. Keypad connected to Raspberry Pi

Example 9-2. keypad.py

```
# keypad.py - read 16-key numeric keypad (dx.com sku 149608)
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
import time
import botbook_gpio as gpio     # 确保库文件botbook gpio.py 与你的程序位于同一个目录中。
keymap = []
keymap.append(['1', '2', '3', 'A'])     # 将按键的物理位置(如第1行第0列)映射到按键值(“4”)。
keymap.append(['4', '5', '6', 'B'])
keymap.append(['7', '8', '9', 'C'])
keymap.append(['*', '0', '#', 'D'])

columns = [2, 3, 14, 15]        # 将列号(如2)映射到GPIO引脚号( 14)。
rows = [18, 17, 27, 22]
lastReadTime = None
bounceTime = 0.03 # s

def initializeKeyPad():
        for x in range(len(rows)):
                gpio.mode(rows[x], 'in')
                gpio.mode(columns[x], 'in')
```

Example 9-2. keypad.py

```
# keypad.py - read 16-key numeric keypad (dx.com sku 149608)
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
def getKeyPress():
    global lastReadTime
    foundKey = None
    if((time.time() - lastReadTime) > bounceTime):  #检查键盘在30ms内的读取次数不超过一
次。
        #pulse columns and read pins
        for c in range(len(columns)):
            gpio.mode(columns[c], 'out')
            gpio.write(columns[c], gpio.LOW)       # 设置某一列(垂直)为LOW,其他列为
HIGH.
            for r in range(len(rows)):
                #检测行是否为LOW.其他行被上拉电阻上拉为HIGH,如图9-3所示。
                if gpio.read(rows[r]) == gpio.LOW:
                    foundKey = keymap[r][c] # 按下的按键位于低电平列和低电平行的交点。
            gpio.write(columns[c], gpio.HIGH)
            gpio.mode(columns[c], 'in')     #
            if not foundKey == None:
                break   #跳出for循环(因为已经找到了按下的按键)
        lastReadTime = time.time()
    return foundKey
```

Example 9-2. keypad.py

```
# keypad.py - read 16-key numeric keypad (dx.com sku 149608)
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
def main():
    global lastReadTime
    initializeKeyPad()
    lastReadTime = time.time()
    while True:
        key = getKeyPress()
        if not key == None:
            print(key)
        time.sleep(0.1) # s

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

Environment Experiment: Revealing Fingerprints

- How can you bypass a keypad, short of breaking it?
- You could try all the combinations, but that' s a lot of trial and error. For example, there are a lot of three-character permutations in a 16-key pad.
  - □ 16 * 16 * 16 = 16**3 = 4096
- Can you see which keys are pressed most? In a keypad that' s been used a lot, the paint or sticker on frequently used keys could be worn.
- You probably have a new keypad. Press the same code a couple of times. Try, if you can, to see grease on well-used keys by using bright light (Figure 9-4).

Environment Experiment: Revealing Fingerprints

➢ How can you bypass a keypad, short of breaking it?

- What if you pulverize graphite from a pencil or take some beauty powder and spread that on keys? What if you clean the keys before typing the combination? Does it matter if your fingers are greasy?

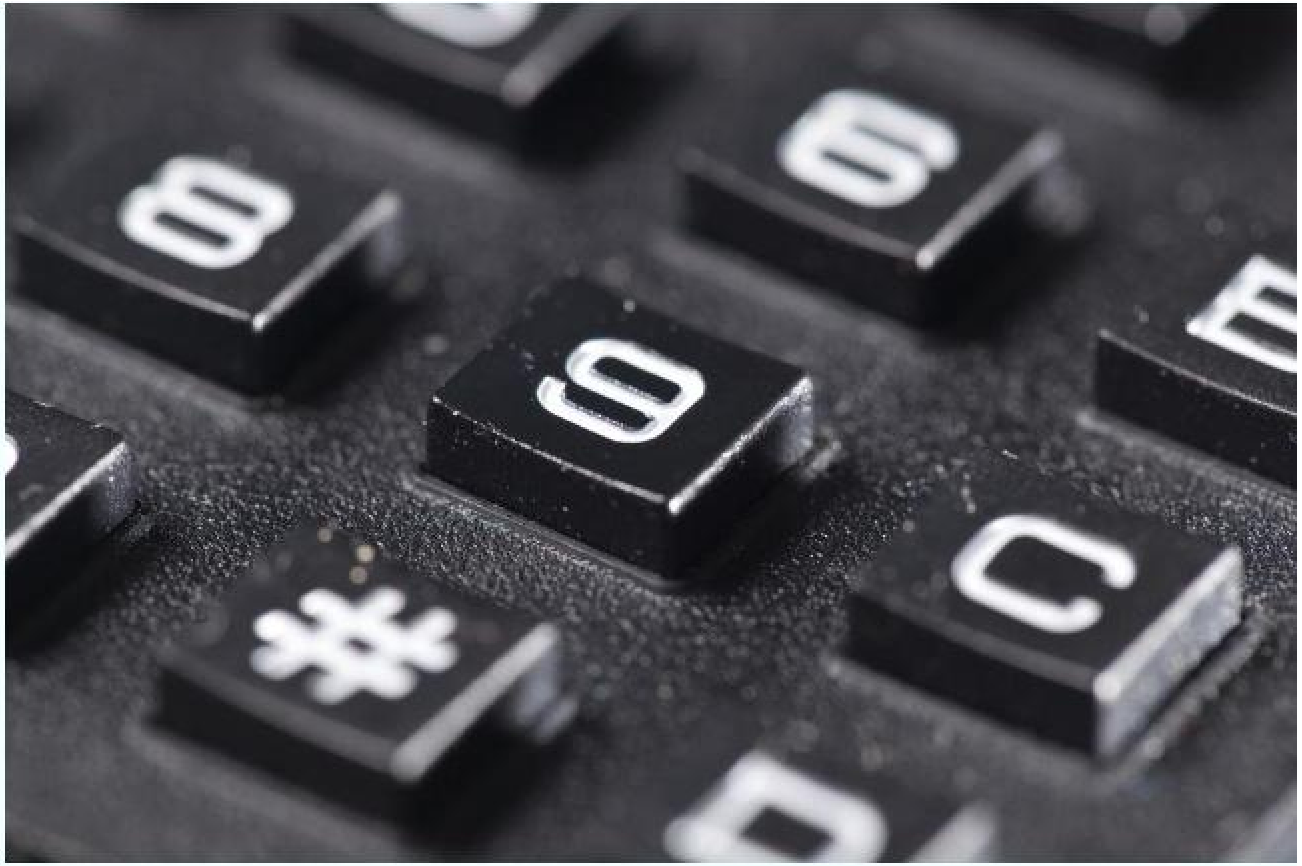


Figure 9-4. Try to see grease on well-used keys by using bright light

Environment Experiment: Revealing Fingerprints

- Latent (hidden) fingerprints can be captured with cyanoacrylate superglue vapor. Even though this is a very efficient technique for glass and plastic, cyanoacrylate fumes are unhealthy and must be handled in a fume hood. Also, superglue annoyingly sticks to everything.
- Did you find the three most used keys? Now there are fewer to try, as you just have to find the correct order of these three keys.
  - 3 * 3 * 3 = 3**3 = 27

Fingerprint Scanner GT-511C3

- A fingerprint scanner is a more sophisticated identification device than the keypad. It is unique and won't get lost or forg worry about someone shoulder surfing as yo
- The GT-511C3 fingerprint scanner (Figure 9 communicate with Arduino or Raspberry Pi. data sheet, available by searching the Web d from the distributor's (SparkFun) catalog pag

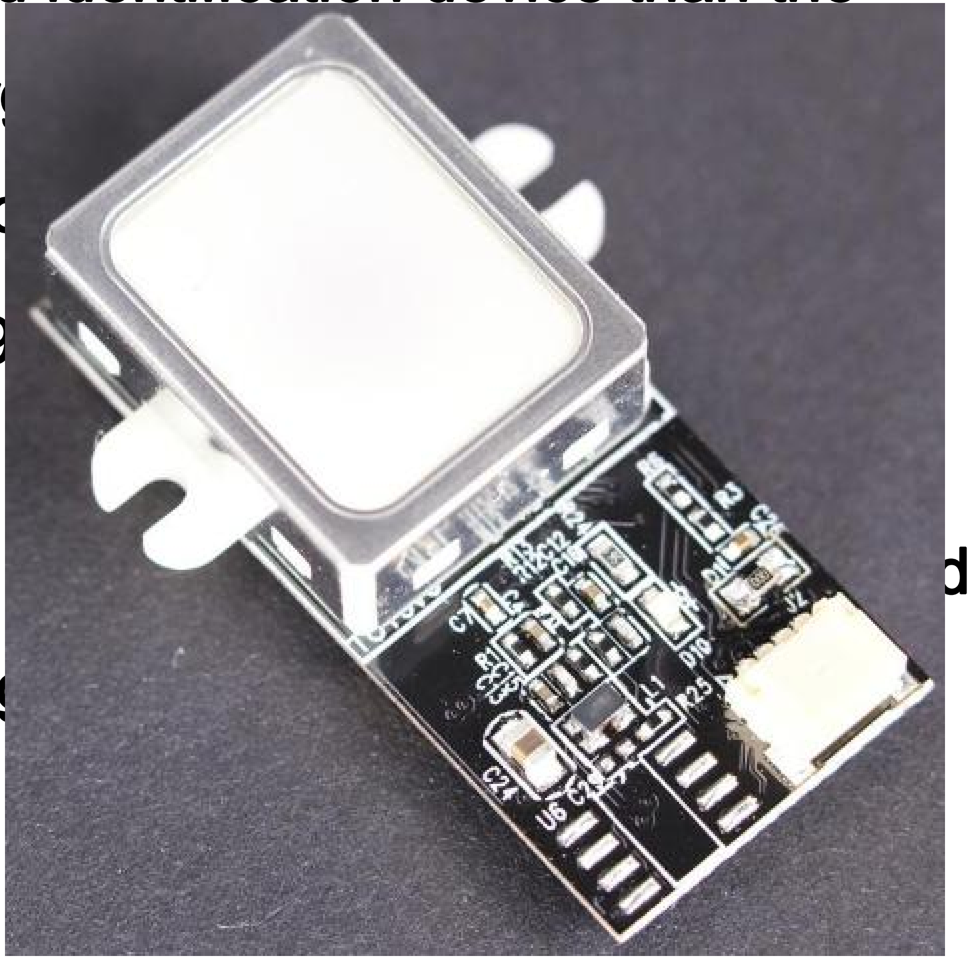


Figure 9-5. Fingerprint scanner GT-511C3

Fingerprint Scanner GT-511C3

- The scanner can store the fingerprints into its own memory, making it easier to use from Arduino.
- The GT-511C3 fingerprint scanner communicates at 9600 bit/s over serial port. The scanner is sometimes slow to respond, so you should configure the serial port for an extra-long timeout with setTimeout(), as shown in the next example program.

Fingerprint Scanner GT-511C3

- The protocol consists of packages. This code stores fingerprint images in the scanner, so only command packages are used (see Table 9-1). The header and device are always the same, and checksum is always calculated the same way. The part that changes is your command and parameter.

表9-1 GT-511C3 指纹传感器的命令包

作用	案例	说明
报头，header	0x55,0xAA	所有的命令数据包都是相同的
设备ID，device ID	1	在同一个设备上始终是相同的
参数，parameter	0	关闭
命令，command	0x12	控制LED（CMD_LED）
校验和，cheecksum		字节的和

Fingerprint Scanner GT-511C3

- The microcontroller (master) sends a command package to the scanner (slave). Commands include the following:
- Light up an LED.
- Delete all fingerprints from scanner memory.
- Enroll 1 (scan and store a new fingerprint to slot one).
- Identify (scan a fingerprint and tell if it’ s one of the stored fingerprints).

Fingerprint Scanner GT-511C3

- The scanner responds with a command package.
- The response command is either ACK (acknowledge, OK, success) or NACK (negative acknowledge, failure, error).
- The parameter in the answer contains additional information, such as the following:
    - ID of recognized fingerprint
    - The fact that fingerprint is unknown
    - Error code
- The most common error codes are 100F (connection already opened) and 100A (memory is empty).

Fingerprint Scanner GT-511C3

- You can get a lot done with the code in this chapter.
- But if you need more details about the fingerprint sensor, search the Web for "GT-511C3 datasheet."
- Fingerprint scanners are vulnerable to fake fingers. Gelatin, like that found in gummy bears, is an easy material to get started with. Other materials include Scotch tape, Play-Doh, and printed transparency with Blu-Tack. The choice of material for fake prints depends on the technique used in the target fingerprint scanner. Chaos Computer Club has published information on extracting fingerprints from objects and using them for creating fake fingerprints.

Fingerprint Sensor Code and Connection for Arduino Mega

- Use the Arduino Mega for easier debugging. Mega has multiple serial ports, so you can connect the fingerprint scanner to one serial and use another for USB code upload and debugging.

- If you want a cheaper option, try the Arduino Leonardo. Although it looks a lot like an Arduino Uno, the Leonardo has the advantage of having two serial ports. Unlike the Uno, the RX/TX pins (pins 0 and 1) and the USB/Serial port are not tied together. With the Leonardo, Serial refers to the USB/Serial port, and Serial1 refers to pins 0 (RX) and 1 (TX). If you use the Leonardo instead of the Mega, you will need to change the example code and the circuit wiring to use Serial1 (pins 0 and 1) for the fingerprint scanner instead of Serial3 (pins 15 and 14).

Fingerprint Sensor Code and Connection for Arduino Mega

- Figure 9-6 shows the connection diagram for Arduino. Wire it up as shown, and then run the sketch shown in Example 9-3.

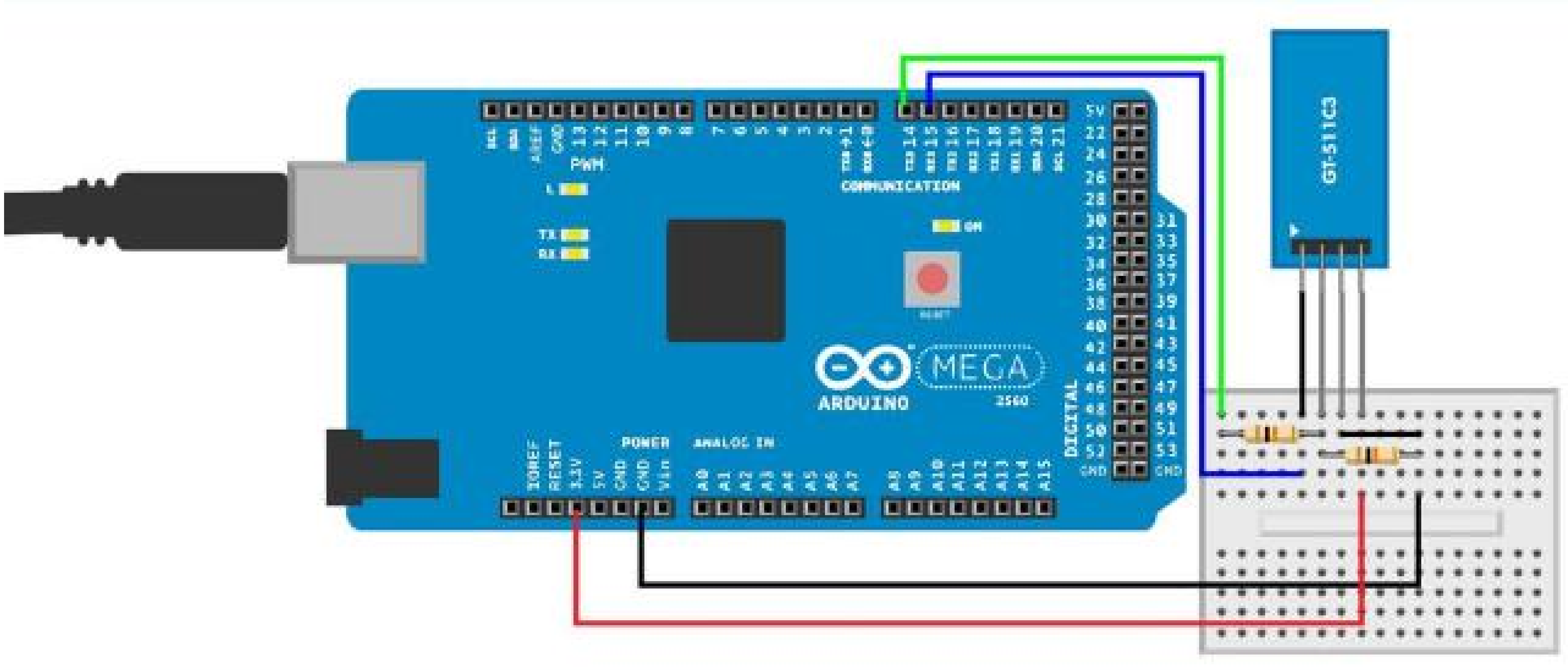


Figure 9-6. Connections for the fingerprint scanner with Arduino

Fingerprint Sensor Code and Connection for Arduino Mega

- Arduino uses 5 V, but fingerprint sensor uses 3.3 V. Excessive voltage could destroy the sensor, reduce its usable age, or cause incorrect readings.
- In this project, you'll use a voltage divider to reduce the voltage given by the Arduino TX pin. All you need are two resistors.
- You don't need a voltage divider on the sensor's TX pin, because it has a maximum of 3.3 V—much less than Arduino's RX pin. Arduino will still recognize 3.3 V as HIGH, because it's more than half of 5 V

Fingerprint Sensor Code and Connection for Arduino Mega

- Difficult code! This code is more difficult than some other code examples in this book: it uses pointers to convert between a struct and a byte buffer. You don’t have to completely understand it to use it. You could just build the circuit, upload the code, and enjoy your fingerprint reader. But if you want to understand it, read on.

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
const byte STX1 = 0x55;                    //命令数据包的报头。
const byte STX2 = 0xAA;
const word CMD_OPEN = 0x01;                //datasheet中的命令
const word CMD_CLOSE = 0x02;
const word CMD_LED = 0x12;
const word CMD_GET_ENROLL_COUNT = 0x20;
const word CMD_ENROLL_START = 0x22;
const word CMD_ENROLL_1 = 0x23;
const word CMD_ENROLL_2 = 0x24;
const word CMD_ENROLL_3 = 0x25;
const word CMD_IS_FINGER_PRESSED = 0x26;
const word CMD_DELETE_ALL = 0x41;
const word CMD_IDENTIFY = 0x51;
const word CMD_CAPTURE_FINGER = 0x60;

const word ACK = 0x30;                     // 返回值
const word NACK = 0x31;                    //Error
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
struct package {		// 命令数据包结构
		byte header1;
		byte header2;
		word deviceID;
		unsigned long param;
		word cmd;
		word checksum;
};
const int SIZE_OF_PACKAGE = 12;

word calcChecksum(struct package *pkg) {		//将pdu中的所有字节求和后得到校验和.

		word checksum = 0;
		byte *buffer = (byte*)pkg;	//
		for(int i=0; i < (sizeof(struct package) - sizeof(word)); i++)  {
				checksum += buffer[i];
		}
		return checksum;
}
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
```

```
int sendCmd(word cmd, int param) {      // 发送一条命令
        struct package pkg;
        pkg.header1 = STX1;
        pkg.header2 = STX2;
        pkg.deviceID = 1;                  //设备端的设备ID固定为1
        pkg.param = param;
        pkg.cmd = cmd;
        pkg.checksum = calcChecksum(&pkg);
        Serial.println("Sending command");
        byte *buffer = (byte*)&pkg;        //在发送命令之前，将结构体转换为字节指针

        int bytesSent = Serial3.write(buffer, sizeof(struct package));

        if(bytesSent != sizeof(struct package)) {
                Serial.println("Error communicating");
                return -1;
        }
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
```

```
	int bytesReceived = 0;
	char recvBuffer[SIZE_OF_PACKAGE];                    // 定义接收缓冲区
	struct package *recvPkg = (struct package*) recvBuffer;   // 接收缓冲区转变成结构体指针

	bytesReceived = Serial3.readBytes(recvBuffer, sizeof(struct package));    //填充接收缓冲
区
	if(bytesReceived != SIZE_OF_PACKAGE) {
		Serial.println("Error communicating");
		return -1;
	}

	if( recvPkg->header1 != STX1 || recvPkg->header2 != STX2) {   //检查报头
		Serial.println("Header error!");
		return -1;
	}
	if(recvPkg->checksum != calcChecksum(recvPkg)) {
		Serial.println("Checksum mismatch error!");
		return -1;
	}
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
```

```
        if(recvPkg->cmd == NACK) {
                Serial.println("NACK - Cmd error!");
                Serial.print("Error: ");
                Serial.println(recvPkg->param,HEX);
                return -1;
        }
        return recvPkg->param;
}

//使用变量定义的命令代码在数据手册中均有定义.
void setup() {
        Serial.begin(115200); //
        Serial3.begin(9600);  //
        Serial3.setTimeout(10*1000); // ms    //
}
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
void flashLed(int time) {
        sendCmd(CMD_LED, 1);
        delay(time);
        sendCmd(CMD_LED, 0);
}

void loop() {
        Serial.println("Sending open command");
        sendCmd(CMD_OPEN, 0);
        if(sendCmd(CMD_DELETE_ALL, 0) >= 0) {      //为了便于测试，一开始删除所有的指纹.
        //LED闪烁3次.
                flashLed(500);   delay(500);
                flashLed(500);   delay(500);
                flashLed(500);
        }

        Serial.println("Starting capture");            //表示准备登记指纹
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
```

```
int id = 0;
id = sendCmd(CMD_GET_ENROLL_COUNT, 0);        //寻找第一个尚未使用的插槽
sendCmd(CMD_LED, 1);
sendCmd(CMD_ENROLL_START, id);
Serial.println("Press finger to start enroll");
int ret = 0;
WaitForFinger(false);     //等待手指按下
Serial.println("Capturing finger");
ret = sendCmd(CMD_CAPTURE_FINGER, 1);
                    //参数1表示最佳画质，但扫描速度慢，0画质一般，但扫描速度快。
if(ret < 0) {
          EnrollFail();
          return;
}
Serial.println("Remove finger");

sendCmd(CMD_ENROLL_1, 0);
WaitForFinger(true);                //等待指纹离开
Serial.println("Press finger again");
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
```

```
WaitForFinger(false);     //等待手指按下
ret = sendCmd(CMD_CAPTURE_FINGER, 1);
if(ret < 0) {
        EnrollFail();
        return;
}
Serial.println("Remove finger");

sendCmd(CMD_ENROLL_2, 0);
WaitForFinger(true);      //等待手指离开
Serial.println("Press finger again");

WaitForFinger(false);     //等待手指按下
ret = sendCmd(CMD_CAPTURE_FINGER, 1);
if(ret < 0) {
        EnrollFail();
        return;
}
Serial.println("Remove finger");
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
```

```
ret = sendCmd(CMD_ENROLL_3, 0);
if(ret != 0) {
        EnrollFail();
        return;
}
WaitForFinger(true);      //等待手指离开
flashLed(500);   delay(500);
flashLed(500);   delay(500);
Serial.println("Enroll completed");
Serial.println("Press finger for identify");
sendCmd(CMD_LED, 1);

// Identify
WaitForFinger(false);     //等待手指按下
ret = sendCmd(CMD_CAPTURE_FINGER, 1); //
if(ret < 0) {
        IdentFail();
        return;
}
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
```

```
        ret = sendCmd(CMD_IDENTIFY, 0);        //识别扫描的指纹
        if(ret >= 0 && ret < 200) {
                Serial.print("ID found: ");
                Serial.println(ret);
                flashLed(500);   delay(500);
                flashLed(500);   delay(500);
                flashLed(500);   delay(500);
                flashLed(500);   delay(500);
                flashLed(500);   delay(500);
                flashLed(500);   delay(500);
        } else {
                Serial.println("ID not found");
        }
        sendCmd(CMD_CLOSE,0);
        delay(100000);
}
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
```

```
void WaitForFinger(bool bePressed) { //true：等待直到手指离开，false：等待直到手指按下
		delay(500);
		if(!bePressed) {
				while(sendCmd(CMD_IS_FINGER_PRESSED, 0) > 0) {//finger is not pressed
						delay(200);
				}
		} else {
				while(sendCmd(CMD_IS_FINGER_PRESSED, 0) == 0) {//finger is pressed
						delay(200);
				}
		}
}
```

Example 9-3. fingerprint_scanner.ino

```
// fingerprint_scanner.ino - learn and recognize fingerprints with GT-511C3
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
```

```
// Flash LED 3 times for failure and close device.
void IdentFail() {
		Serial.println("Ident failed!");
		flashLed(500);	delay(500);
		flashLed(500);	delay(500);
		flashLed(500);	delay(500);
		sendCmd(CMD_CLOSE, 0);
}

// Flash LED 4 times for failure and close device.
void EnrollFail() {
		Serial.println("Enroll failed!");
		flashLed(500);	delay(500);
		flashLed(500);	delay(500);
		flashLed(500);	delay(500);
		flashLed(500);	sendCmd(CMD_CLOSE, 0);
}
```

Fingerprint Sensor Code and Connection for Raspberry Pi

- To use the serial port in Raspberry Pi, you must first release it from use as a login terminal. See Enabling the Serial Port in Raspberry Pi.
- To access the serial port from Python, you need to install the PySerial library by running the command:
  - □ sudo apt-get update && sudo apt-get install python-serial
- The connection for Raspberry Pi is simple—even simpler than the connection for Arduino. Raspberry Pi uses 3.3 V for HIGH, the same as the fingerprint sensor. As the voltage is already correct, you don't need resistors to build a voltage divider with Raspberry Pi.

Fingerprint Sensor Code and Connection for Raspberry Pi

- Wire up the scanner as shown in Figure 9-7,
- and then run the code in Example 9-4.

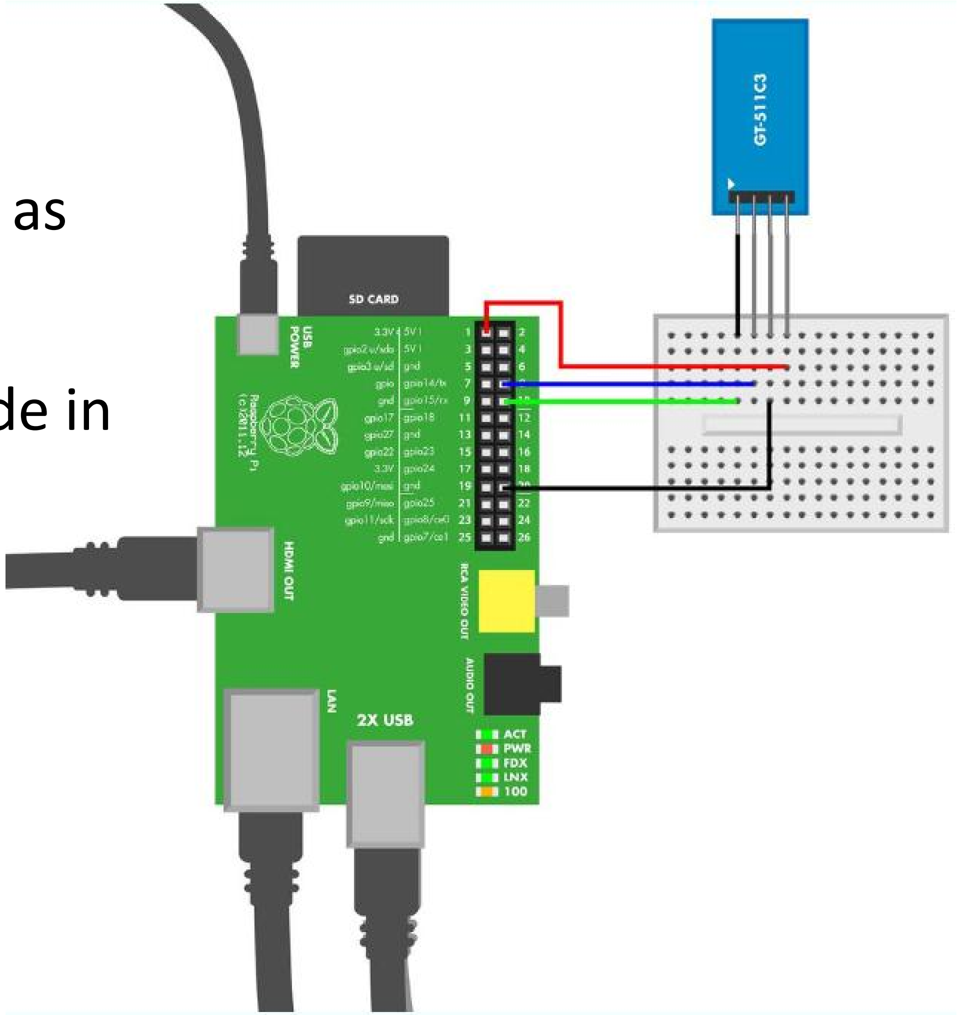


Figure 9-7. Fingerprint scanner connection diagram for Raspberry Pi

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
import time
import serial
import struct
STX1 = 0x55     # 命令数据包的报头
STX2 = 0xAA
CMD_OPEN = 0x01    # 数据手册中说明的命令
CMD_CLOSE = 0x02
CMD_LED = 0x12
CMD_GET_ENROLL_COUNT =  0x20
CMD_ENROLL_START = 0x22
CMD_ENROLL_1 = 0x23
CMD_ENROLL_2 = 0x24
CMD_ENROLL_3 = 0x25
CMD_IS_FINGER_PRESSED = 0x26
CMD_DELETE_ALL = 0x41
CMD_IDENTIFY =  0x51
CMD_CAPTURE_FINGER = 0x60
```

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
ACK = 0x30      # 响应为ACK，表示成功
NACK = 0x31    #响应为NACK，表示不成功

port = None

def calcChecksum(package):      # 每个命令数据包都有校验和，即对所有字节求和
        checksum = 0
        for byte in package:
                checksum += ord(byte)
                //ord()函数以一个字符（长度为1的字符串）作为参数，返回对应的ASCII数值
        return int(checksum)
```

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
def sendCmd(cmd, param = 0):    # 发送一个命令并读取响应信息。返回响应中的参数，错误返回负数
        package = chr(STX1)+chr(STX2)+struct.pack('<hih', 1, param, cmd)
        # '<'小端字节序，'h'有符号两字节短整型，'i'普通四字节整数
        checksum = calcChecksum(package)
        package += struct.pack('<h',checksum)   # 将计算出的校验和添加到原始数据后面
        sent = port.write(package)
        if(sent != len(package)):
                print "Error communicating"
                return -1
        recv = port.read(sent)    #读取响应信息
        recvPkg = struct.unpack('cchihh',recv)  # 将原始数据 拆包到元组
        if recvPkg[4] == NACK:
                print("error: %s" % recvPkg[3])
                return -2
        time.sleep(1)
        return recvPkg[3]
```

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
def startScanner():
        print("Open scanner communications")
        sendCmd(CMD_OPEN)

def stopScanner():
        print("Close scanner communications")
        sendCmd(CMD_CLOSE)

def led(status = True):
        if status:
                sendCmd(CMD_LED,1)
        else:
                sendCmd(CMD_LED,0)
def enrollFail():
        print("Enroll failed")
        led(False)
        stopScanner()
```

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
def identFail():
    print("Ident failed")
    led(False)
    stopScanner()

def startEnroll(ident):
    sendCmd(CMD_ENROLL_START,ident)

def waitForFinger(state):
    if(state):
        while(sendCmd(CMD_IS_FINGER_PRESSED) == 0):
            time.sleep(0.1)
    else:
        while(sendCmd(CMD_IS_FINGER_PRESSED) > 0):
            time.sleep(0.1)
def captureFinger():
    return sendCmd(CMD_CAPTURE_FINGER)
```

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
def enroll(state):
    if state == 1:
        return sendCmd(CMD_ENROLL_1)
    if state == 2:
        return sendCmd(CMD_ENROLL_2)
    if state == 3:
        return sendCmd(CMD_ENROLL_3)

def identifyUser():
    return sendCmd(CMD_IDENTIFY)

def getEnrollCount():
    return sendCmd(CMD_GET_ENROLL_COUNT)

def removeAll():
    return sendCmd(CMD_DELETE_ALL)
```

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
def main():
    print("Remove all identities from scanner")
    startScanner()
    removeAll()
    led()
    print("Start enroll")
    newID = getEnrollCount()
    print(newID)
    startEnroll(newID)
    print("Press finger to start enroll")
    waitForFinger(False)
    if captureFinger() < 0:
        enrollFail()
        return
    enroll(1)
    print("Remove finger")
    waitForFinger(True)
```

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
    print("Press finger again")
    waitForFinger(False)
    if captureFinger() < 0:
        enrollFail()
        return
    enroll(2)
    print("Remove finger")
    waitForFinger(True)
    print("Press finger again")
    waitForFinger(False)
    if captureFinger() < 0:
        enrollFail()
        return
    if enroll(3) != 0:
        enrollFail()
        return
```

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
    print("Remove finger")
    waitForFinger(True)

    print("Press finger again to identify")
    waitForFinger(False)
    if captureFinger() < 0: # 向扫描仪发送命令，但还没有进行识别指纹的任务。
        identFail()
        return
    ident = identifyUser()
    if(ident >= 0 and ident < 200): #   返回识别出的指纹ID
        print("Identity found: %d" % ident)
    else:
        print("User not found")
    led(False)
    stopScanner()
```

Example 9-4. fingerprint_scanner.py

```
# fingerprint_scanner.py - learn and recognize fingerprints with GT-511C3
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
    if __name__ == "__main__":
        try:
            if port == None:
                port = serial.Serial(
                    "/dev/ttyAMA0",
                    baudrate=9600,
                    timeout=None)   # 指纹扫描仪在9600波特率的串口上通
信
            main()
        except Exception, e:
            print e
            port.close()
        finally:
            port.close()
```

RFID with ELB149C5M Electronic Brick

- Radio frequency identification (RFID) offers cheap, unique object identification from a distance. It's already used a lot in warehouses, and as prices go down, it's making its way to consumer packages.
- Some pets have microchips in the neck so that they can be identified even if they lose their collar tags. In Finland, some libraries use RFID instead of EAN bar codes.
- There are multiple RFID standards that differ on price, reading distance, security, amount of stored data, and popularity.
- In this project, you'll play with the ELB149C5M Electronic Brick. It reads uem4100 standard tags at 125 kHz.

RFID with ELB149C5M Electronic Brick

- Once you've built the circuit and powered the sensor, hold the card to the sensor. A green light blinks when the sensor reads the card.
  - The ELB149C5M sensor can use one of two protocols, serial port or Wiegand protocol. As serial port is much more common and familiar, the experiment here uses serial port. You can find the description of the protocol by searching for “ELB149C5M data sheet.”

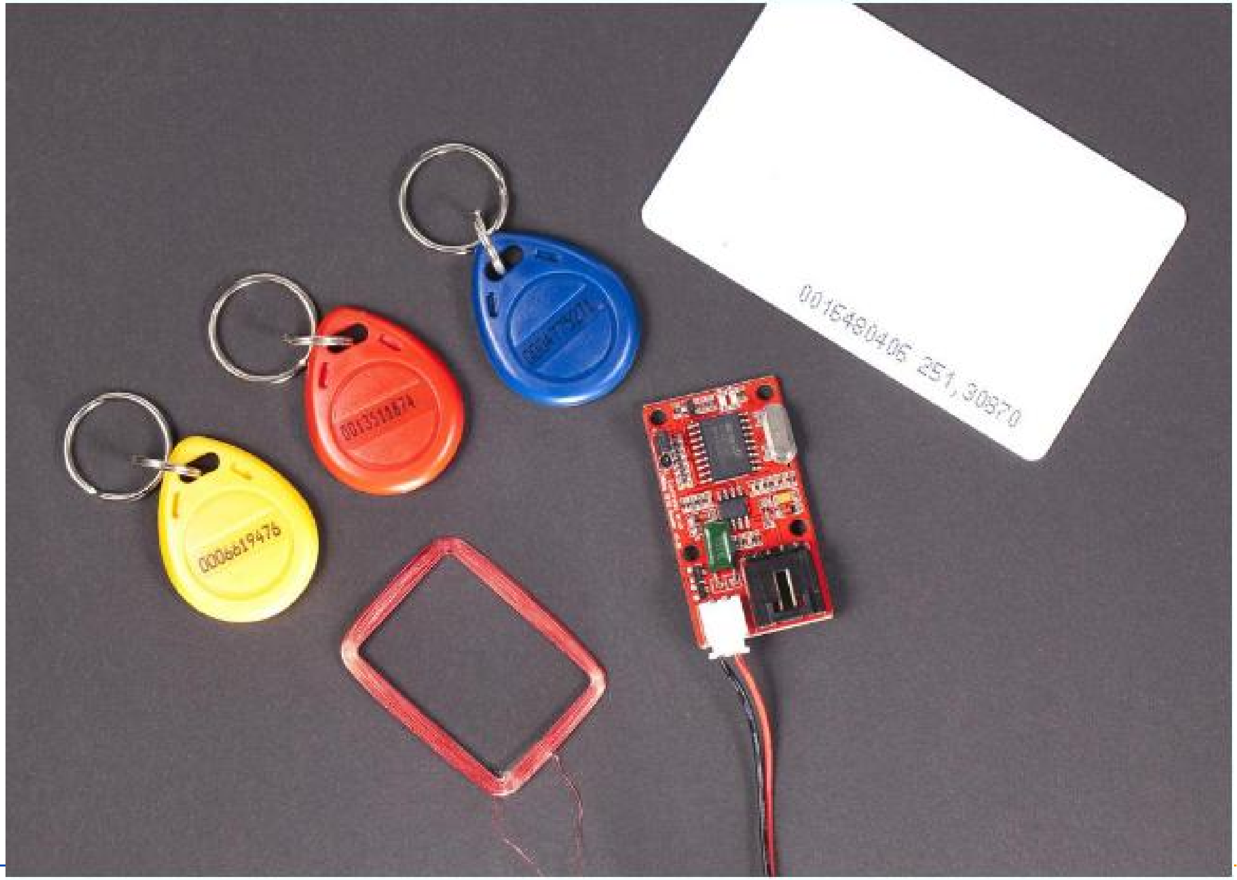


Figure 9-8. The ELB149C5M brick

RFID with ELB149C5M Electronic Brick

- To select serial mode, the UW jumper on the sensor is set to U for UART serial port.
- The serial port uses 9600 N81 TTL: 9600 bit/s speed with typical settings of no verify bit (N), 8 data bits (8), and one stop bit (1). The signal levels are TTL/UART, so you can just connect it to the Arduino pins with jumper wire.

RFID with ELB149C5M Electronic Brick

- You can usually connect TTL directly to Arduino or Raspberry PI.
- RS232 connects to serial port in older computers and should not be connected directly to Raspberry Pi or Arduino.
- TTL (transistor-transistor logic) uses LOW for 0 and HIGH for 1 .
- For TTL, LOW is 0 V (GND) and HIGH is typically 3.3 V or 5 V.
- The old-fashioned RS232 serial port uses voltages between -25 V and +25 V.
- To connect RS232 to Arduino or Raspberry Pi, you would need a conversion chip like the MAX 232.

RFID with ELB149C5M Electronic Brick

- The RFID reader initiates communication by sending a packet. The packet contains the static identifying number of the RFID tag (see Table 9-2).

作用	ASCII字符长度	字节长度	说明
开始（Start）	1字符	未解码	0x02 - STX, ASCII 正文开始
卡号（Card number）	10字符	5字节	1字节为制造商，4字节为id
校验和（Checksum）	2字符	1字节	字节之间的XOR位运算
结束（End）	1字符	未解码	0x03 - ETX, ASCII正文结束

- The RFID reader sends ASCII characters in an odd way, such that every pair of characters represents a hexadecimal number. For example, the two byte string “3E” represents one byte hex code 0x3E (62).

RFID with ELB149C5M Electronic Brick

- After the microcontroller verifies the packet checksum, you know you have a valid card number that you can use. In this experiment, the program prints the number on the serial monitor.
- For easier debugging, this project uses Arduino Mega. Mega has multiple serial ports, so you can use the serial monitor (Tools→Serial Monitor in the Arduino IDE) at the same time the RFID reader is attached. With the Uno, you would have only one serial port, which would make prototyping extremely slow and frustrating. Our code example uses two serial ports and will not work on Uno without modifications.

RFID Code and Connection for Arduino Mega

- Wire up the Arduino as shown in Figure 9-9, then run the sketch listed in Example 9-5.

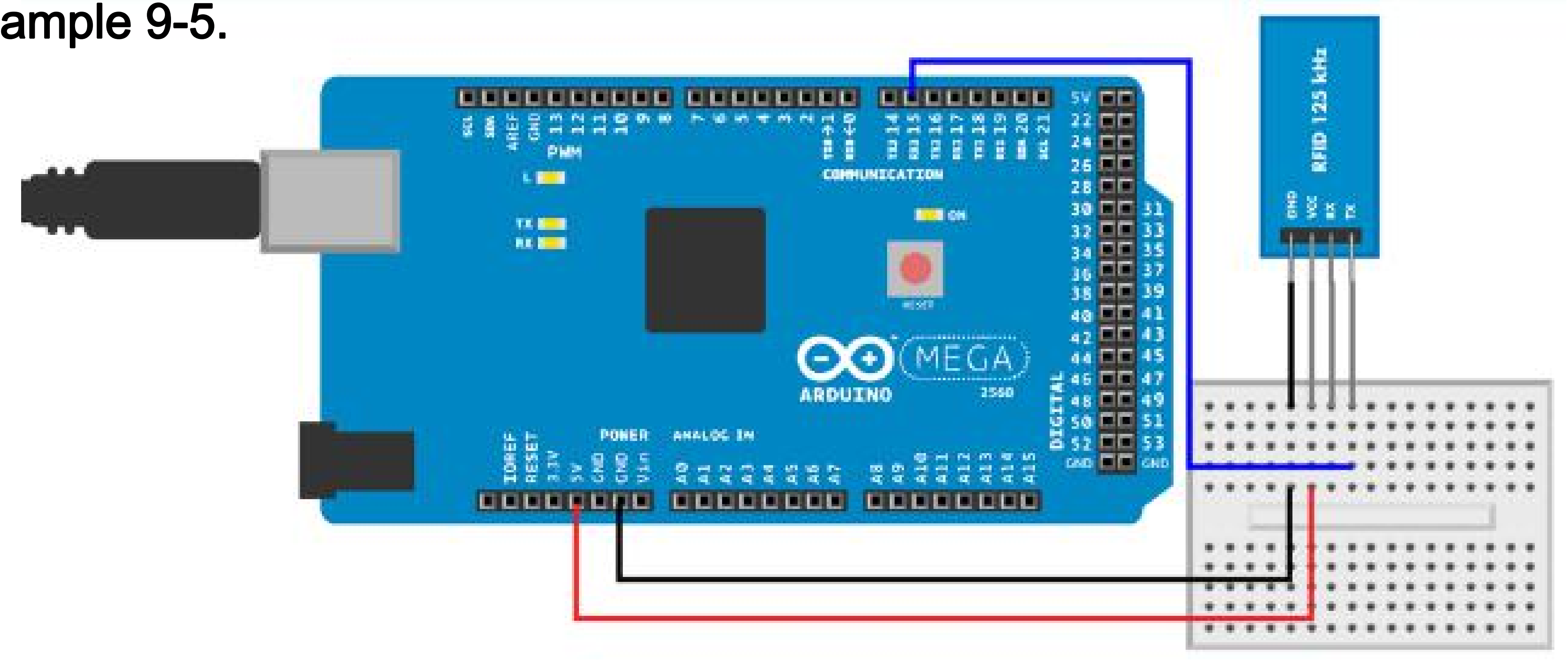


Figure 9-9. Connection diagram for Arduino and RFID reader

Example 9-5. rfid_reader.ino

```
// rfid_reader.ino - read 125 kHz RFID tags with ELB149C5M electronic brick
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
int bytesRead = 0;          // 当前正在处理的数据包的字节数量
char buffer[13];            //初始化14个字节的缓冲区，存储从读卡器中获取的数据包

void setup() {
        Serial.begin(115200);               //连接PC节串口，作为程序输出监视串口
        computer  Serial3.begin(9600);  //连接RFID的串口，这个额外的串口让调试更方便
        }

long toLong(String data) {                      //将2个字节的字符串16进制数变成1个字节的真正的16进
制数
        char buf[20];
        data = "0x"+data;
        data.toCharArray(buf, 19);
        return strtol(buf, NULL, 0);
}
```

Example 9-5. rfid_reader.ino

```
// rfid_reader.ino - read 125 kHz RFID tags with ELB149C5M electronic brick
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
void loop() {
    char recv;
    if(Serial3.available() > 0) {                    // 如果串口有数据可读，则读取
        recv = Serial3.read();
        if(recv == 0x02) {  //表示正文开始， 程序忽略缓冲区其他内容，从数据包起始内容开始
            bytesRead = 0;
            Serial.println("Start reading tag");
        }
        else if(bytesRead == 12 && recv == 0x03) {//已经读取12个字节，且接收到正文结束标记
            Serial.println();
            String data = buffer;
            byte checksum = 0;
            byte chk = toLong(data.substring(10, 12));
            long id = toLong(data.substring(4, 10));
```

Example 9-5. rfid_reader.ino

```
// rfid_reader.ino - read 125 kHz RFID tags with ELB149C5M electronic brick
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
// Requires Arduino Mega for extra serial port
            for(int i = 0; i < 10; i=i+2) {
                checksum ^= toLong(data.substring(i, i+2));    //计算校验和
            }
            Serial.print(id);  //输出ID
            if(checksum == chk) {    // 检验校验和是否与数据包中的校验和一致
                Serial.println(" Card ok");
            } else {
                Serial.println(" Checksum error!");
            }
        } else {
            buffer[bytesRead] = recv;
            bytesRead++;
            Serial.print(recv);
        }
    }
    delay(10);
}
```

RFID Code and Connection for Raspberry Pi

- To use the serial port in Raspberry P s a login terminal. See Enabling the Seri

- Figure 9-10 shows the connection di berry Pi. Wire it up as shown, and then rur

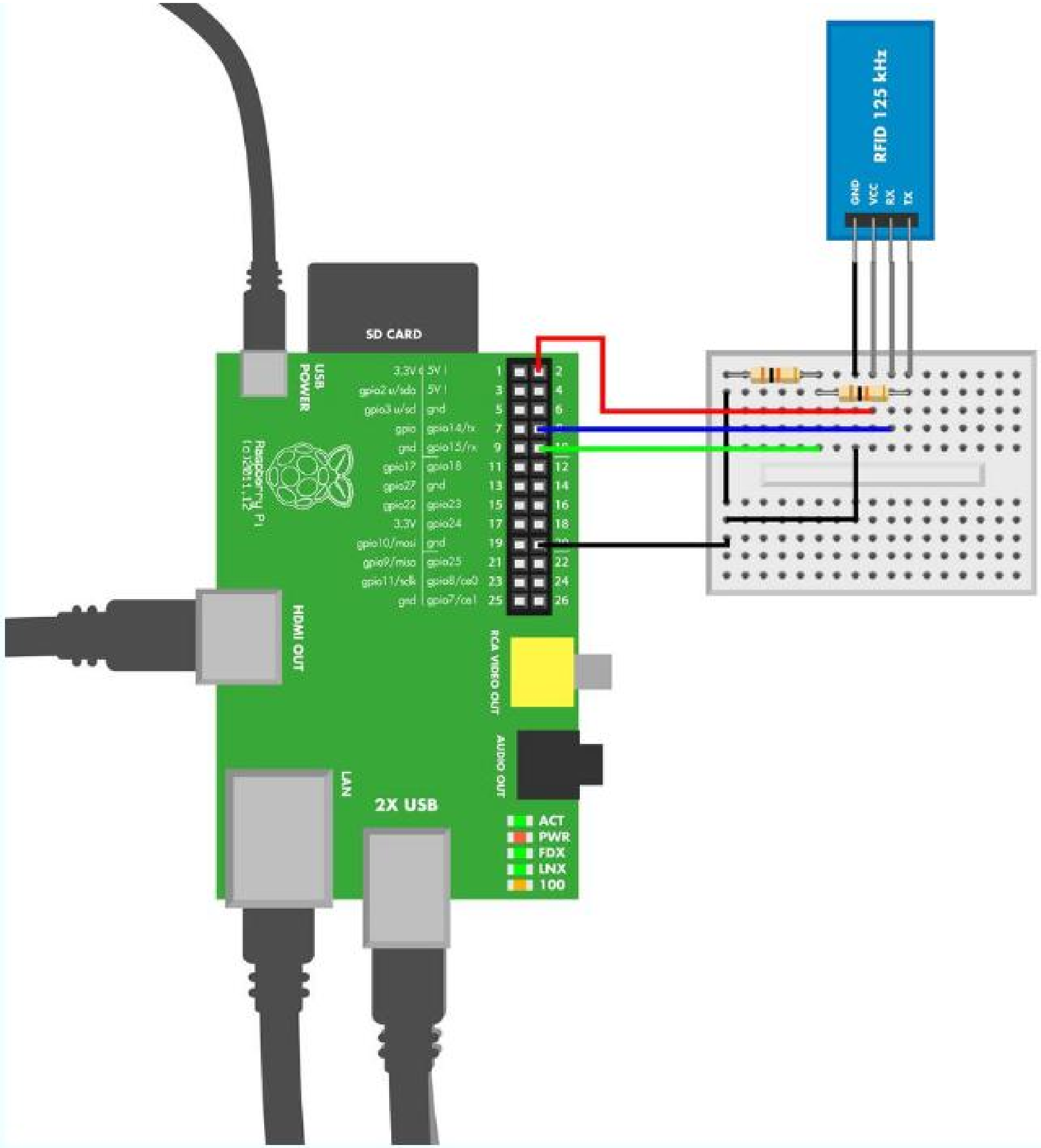


Figure 9-10. Raspberry Pi connections for the RFID reader

Example 9-6. rfid_reader.py

```
# rfid_reader.py - read 125 kHz RFID tags with ELB149C5M electronic brick
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
import time
import serial   # 导入Python的PySerial库
import struct
port = None
def main():
        global port
        bytesRead = -1  #  初始化bytesRead为一个正常情况不可能获取的数值，目的是便于调试
        buff = [0x00]*12  #  初始化buff为全0的列表。正文开始字符和结束字符(共3个）不在此缓冲区中
        print("Ready to receive tag")
        while True:     #  死循环
                recv = port.read()      #  从串口读取一个字节的数据
                if(ord(recv) == 0x02):  #  0x02表示正文开始
                        bytesRead = 0
                        print("Start reading tag")
                elif(bytesRead == 12 and ord(recv) == 0x03):    # 正文结束（0x03）且长度等于12
                        print("Checking tag")
                        data = ""       # 变量data将存储原始字节
```

Example 9-6. rfid_reader.py

```
# rfid_reader.py - read 125 kHz RFID tags with ELB149C5M electronic brick
# (c) BotBook.com - Karvinen, Karvinen, Valtokari
                        checksum  = 0x00
                        for x in 0, 2, 4, 6, 8, 10:
                                hexString = ''.join( buff[ x : x+2 ] )  #  每两个字节存储一个十六进制数
                                translatedByte = int(hexString, 16)
                                data += chr(translatedByte)     #   将字节数据添加到 data尾部
                                checksum = checksum ^ translatedByte     #  检验校验和
                        cardData = struct.unpack(">cic", data)  # 将数据拆包到元组
                        if checksum != 0:
                                print "Checksum calculation failed"
                        print cardData[1]       #    输出元组的第二个元素，也就是10个字符的卡号
                else:   # 如果不是数据包的起始或结束字符，则说明是数据包的中间部分所以
                        buff[bytesRead] = recv  #  接收一个字节存储到buff中
                        bytesRead += 1

if __name__ == "__main__":
        if port == None:
                port = serial.Serial("/dev/ttyAMA0", baudrate=9600, timeout=None)
                port.flushInput()
        main()
```

Test Project: Ancient Chest from the Future

- Marry a chest and fingerprint sensor. You'll get a box that opens only with your finger. You can even authorize your friends' fingers, as well.
- What You'll Learn in the Ancient Chest project how to:
  - Use the fingerprint sensor to control a lock.
  - Split your code into multiple files.
  - Build a simple lock with a servo.
  - Package your project with ancient material for style.
- Figure 9-16 shows the wiring diagram for the chest.

Test Project: Ancient Chest from the Future

- Are you authorized to open the box? The fingerprint scanner glows blue. Press your finger on it, and—beep, whir—the lock opens.
- Enjoy the contents of the Ancient C

Test Project: Ancient Chest from the Future

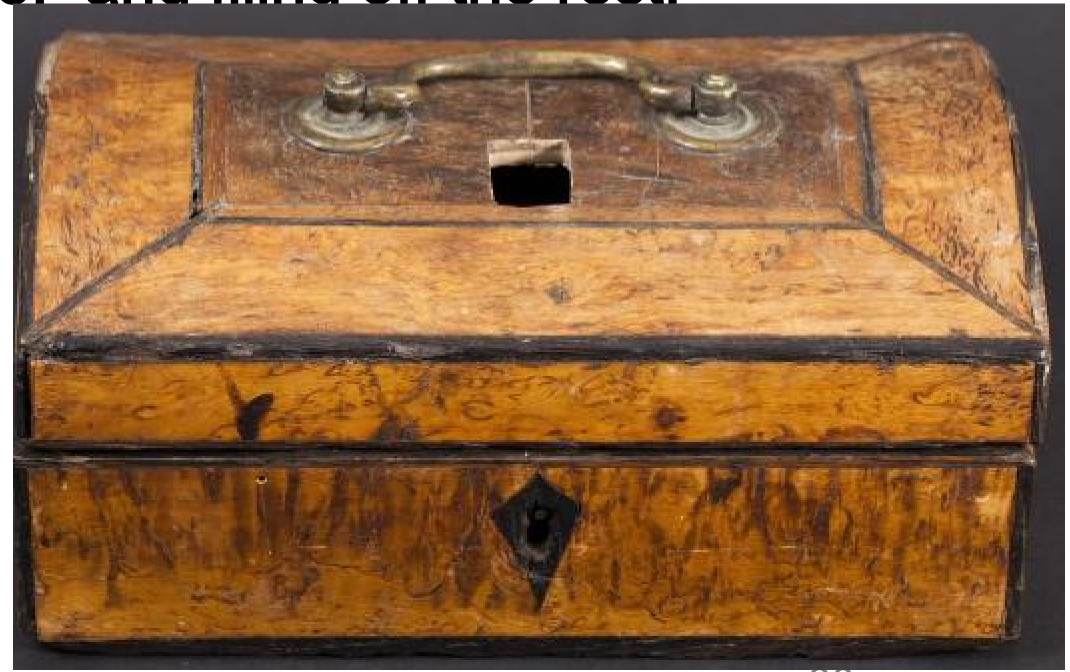
- Inside the chest, there are two buttons: add and reset. These buttons are needed only to change who is authorized to open the box.
- While authorizing fingers, the chest communicates with beeps.
- Click the chest reset button to erase all trusted fingerprints. (The chest reset button is not the same as the Arduino on-board reset button.)
- Click add to authorize a finger. Your finger is scanned three times. If scanning fails, five beeps tell you to press the add button and try again. The first successful scan makes one beep, the second makes two beeps, and the third makes three beeps. Now that you've authorized your finger, you can authorize more fingers or just go ahead and play with the box. To authorize another

Test Project: Ancient Chest from the Future

- An authorized finger can lock the chest.
- Close the lid and press your finger on the blue fingerprint scanner.
- Beep, whir—the chest locks.
- Press your finger over the blue reader again, and—beep, whir—the chest opens.

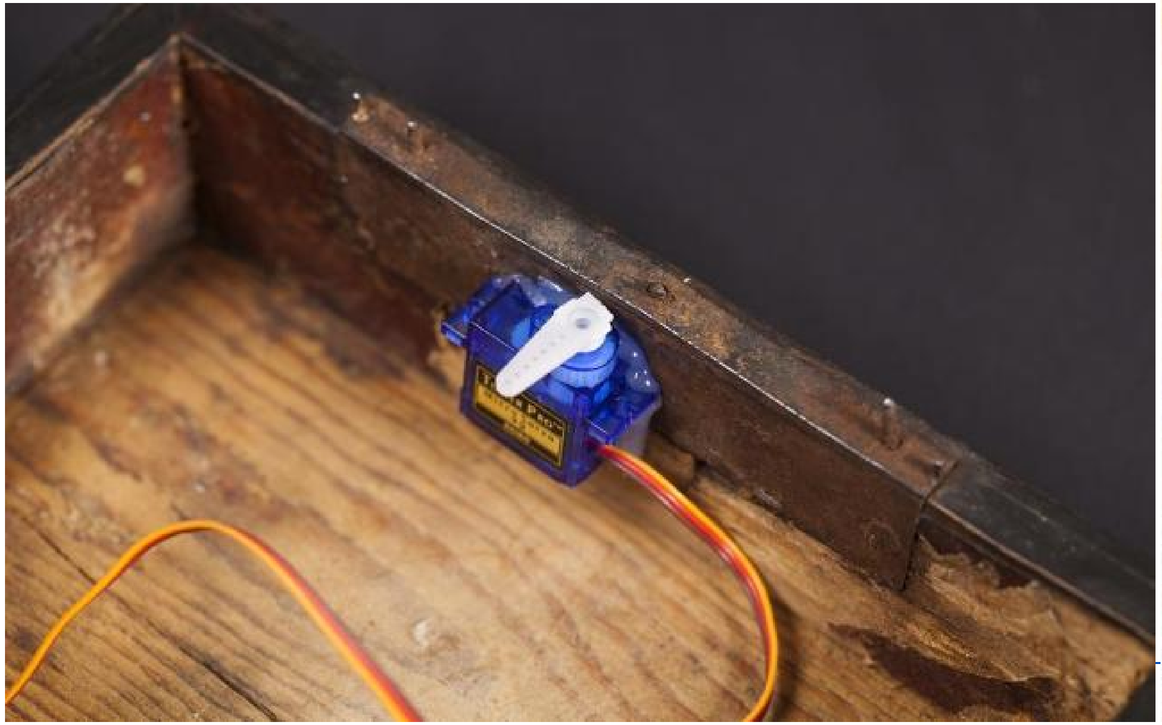
The Box

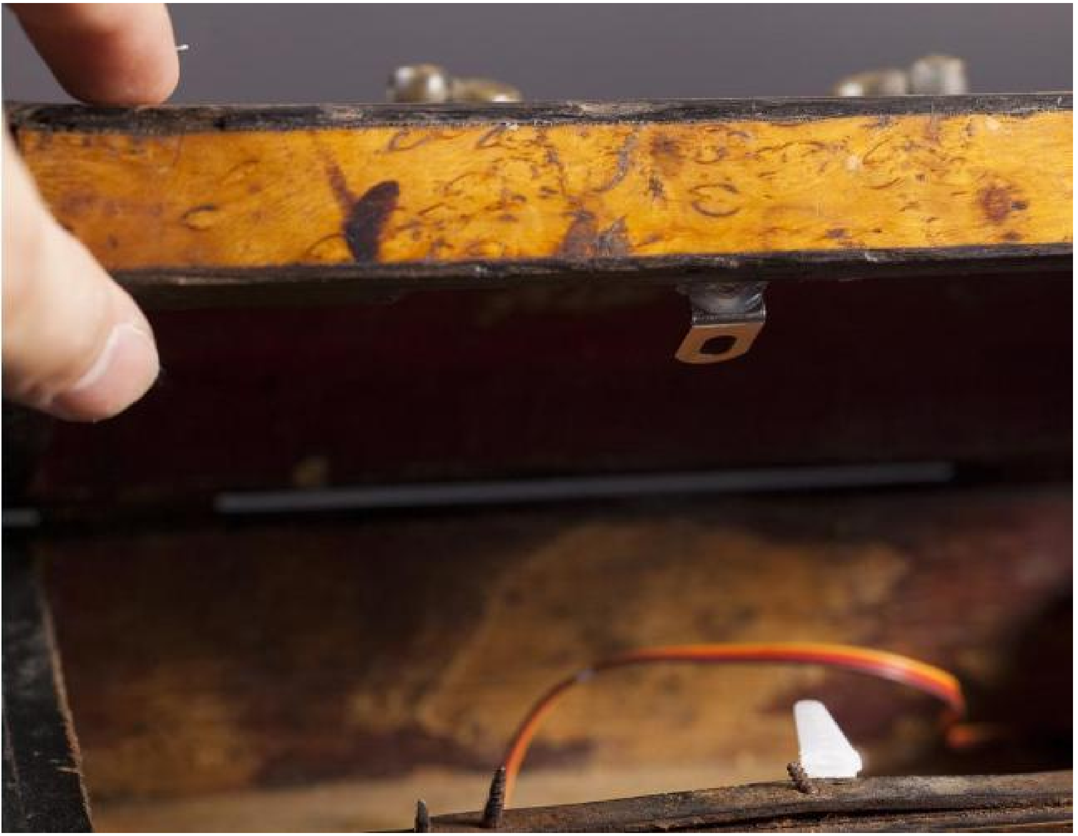
- If you don't happen to have a 19th century box to modify like we did, you'll need to adjust these instructions a little bit. First make a hole for the fingerprint sensor (see Figure 9-12). We did this by drawing the sensor outline on the box, drilling holes in the corners of the outline, and filing off the rest.



The Box

- We used a very simple locking mechanism. A servo arm in the bottom part locks the box by going over a top bracket (see Figure 9-13). For the bracket, we used a part from a Meccano toy, but any small L-bracket from the hardware store will do (see Figure 9-14).





The Box

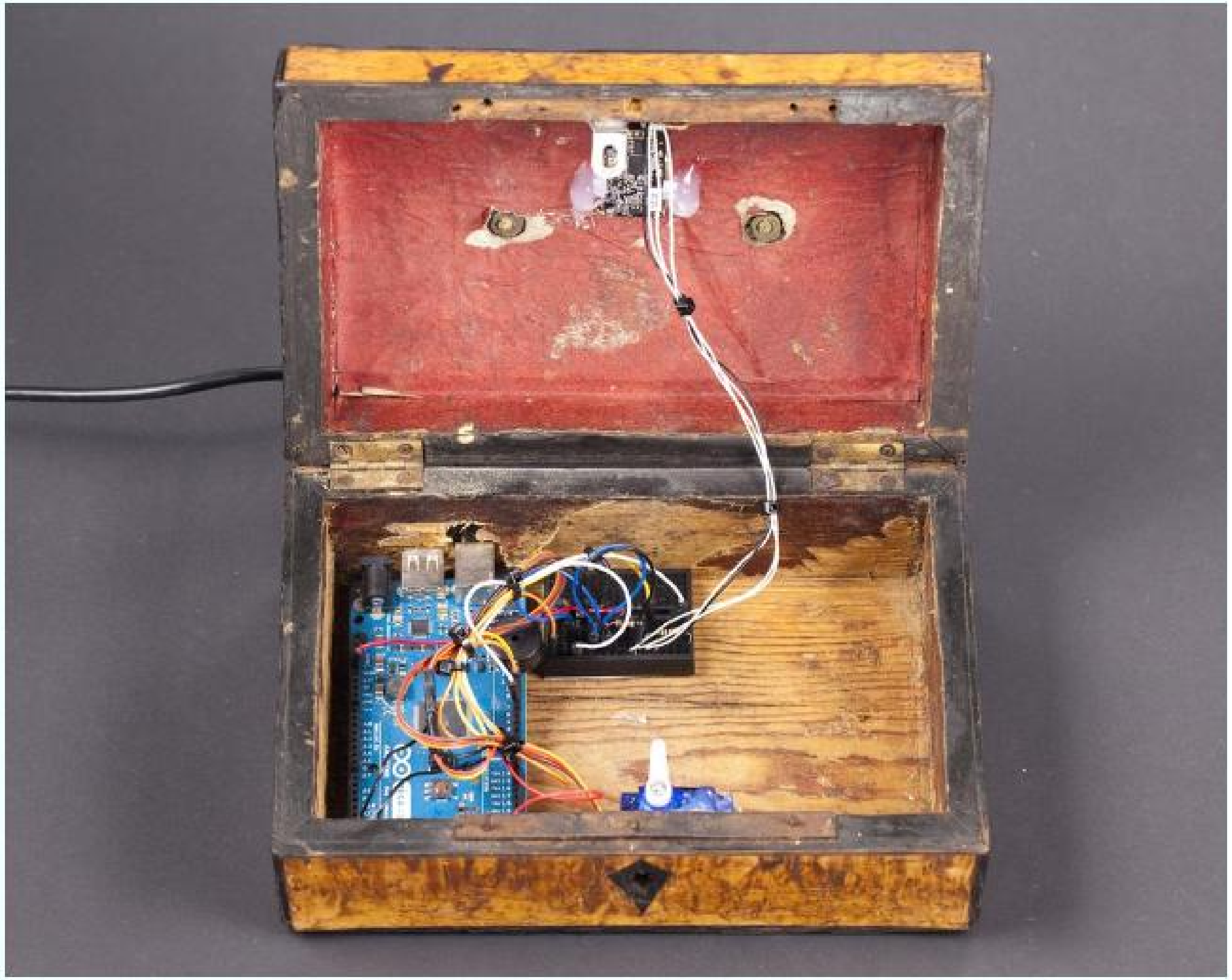


Figure 9-15. Everything packed inside the chest

Ancient Chest Code and Connection for Arduino

- Figure 9-16 shows the wiring diagram for the Ancient Chest. Wire it up as shown, and then run the sketch shown in Example 9-7.

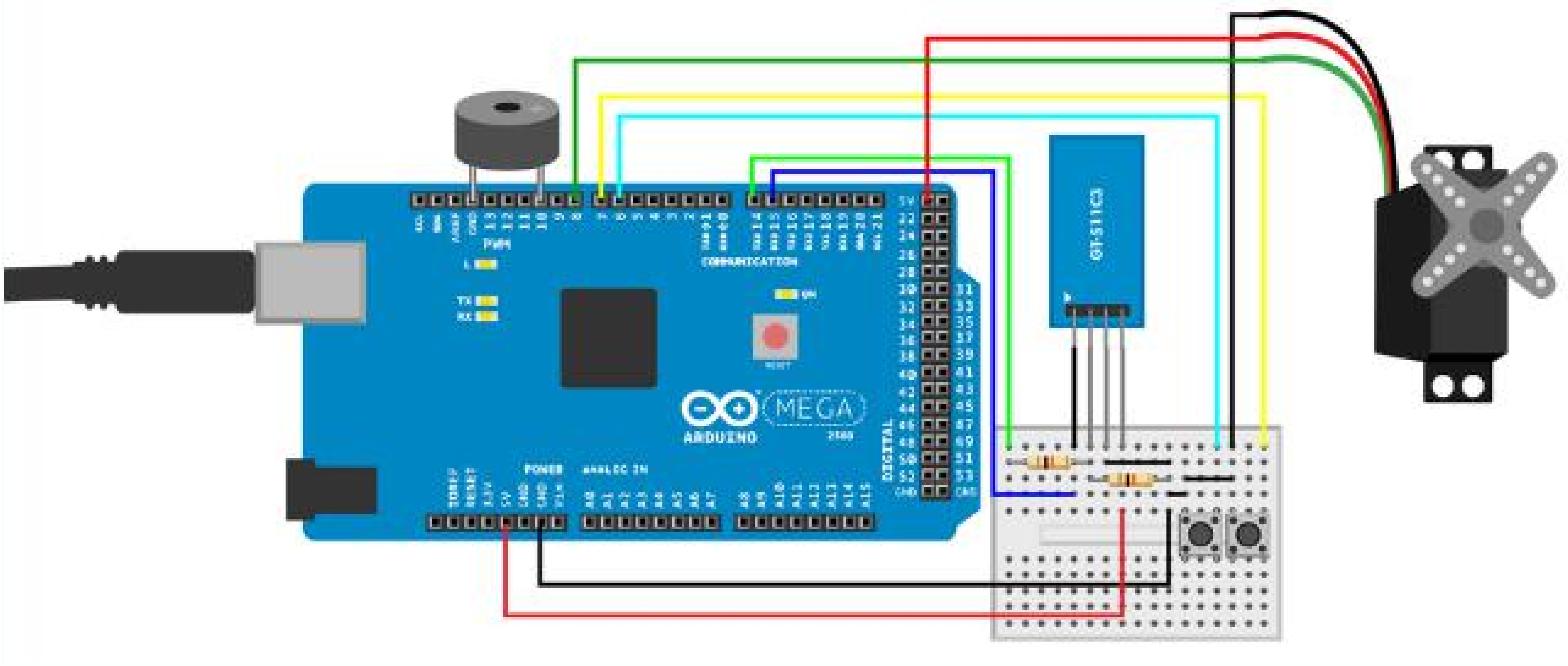


Figure 9-16. Wiring up the Ancient Chest

Ancient Chest Code and Connection for Arduino

- The fingerprint scanner is explained in Fingerprint Scanner GT-511C3, and its Arduino code is in Fingerprint Sensor Code and Connection for Arduino Mega. Servo motors are explained in Servo Motors. The comments after Example 9-7 just explain the new concepts introduced in this example.

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
const byte STX1 = 0x55;
const byte STX2 = 0xAA;
const word CMD_OPEN = 0x01;
const word CMD_CLOSE = 0x02;
const word CMD_LED = 0x12;
const word CMD_GET_ENROLL_COUNT = 0x20;
const word CMD_ENROLL_START = 0x22;
const word CMD_ENROLL_1 = 0x23;
const word CMD_ENROLL_2 = 0x24;
const word CMD_ENROLL_3 = 0x25;
const word CMD_IS_FINGER_PRESSED = 0x26;
const word CMD_DELETE_ALL = 0x41;
const word CMD_IDENTIFY = 0x51;
const word CMD_CAPTURE_FINGER = 0x60;

const word ACK = 0x30;
const word NACK = 0x31;
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
struct package {
        byte header1;
        byte header2;
        word deviceID;
        unsigned long param;
        word cmd;
        word checksum;
};

const int SIZE_OF_PACKAGE = 12;

const int lockPin = 8;
const int resetButtonPin = 7;
const int addButtonPin = 6;
const int speakerPin = 10;
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
float lowPeep = 220;
float highPeep = 440;
int closed = 2000;
int opened = 1000;
int state = 0;

word calcChecksum(struct package *pkg) {
        word checksum = 0;
        byte *buffer = (byte*)pkg;
        for(int i=0; i < (sizeof(struct package) - sizeof(word)); i++)  {
                checksum += buffer[i];
        }
        return checksum;
}
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
int sendCmd(word cmd, int param) {
    struct package pkg;
    pkg.header1 = STX1;  pkg.header2 = STX2;
    pkg.deviceID = 1;
    pkg.param = param;
    pkg.cmd = cmd;  pkg.checksum = calcChecksum(&pkg);
    byte *buffer = (byte*)&pkg;
    int bytesSent = Serial3.write(buffer, sizeof(struct package));
    if(bytesSent != sizeof(struct package)) {
        Serial.println("Error communicating");
        return -1;
    }

    int bytesReceived = 0;
    char recvBuffer[SIZE_OF_PACKAGE];
    struct package *recvPkg = (struct package*) recvBuffer;
    bytesReceived = Serial3.readBytes(recvBuffer, sizeof(struct package));
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
        if(bytesReceived != SIZE_OF_PACKAGE) {
                Serial.println("Error communicating");     return -1;
        }
        if( recvPkg->header1 != STX1 || recvPkg->header2 != STX2) {
                Serial.println("Header error!");           return -1;
        }
        if(recvPkg->checksum != calcChecksum(recvPkg)) {
                Serial.println("Checksum mismatch error!"); return -1;
        }
        if(recvPkg->cmd == NACK) {
                Serial.println("NACK - Cmd error!");
                Serial.print("Error: ");
                Serial.println(recvPkg->param,HEX);
                return -1;
        }
        return recvPkg->param;
}
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
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);
        }
}

void pulseServo(int servoPin, int pulseLenUs){
        digitalWrite(servoPin, HIGH);
        delayMicroseconds(pulseLenUs);
        digitalWrite(servoPin, LOW);
        delay(15);
}
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
void peep(int count, float frequency){
        for(int i = 0; i < count; i++) {
                wave(speakerPin, frequency, 400);
                delay(400);
        }
}

void WaitForFinger(bool bePressed) {
        delay(500);
        if(!bePressed) {
                while(sendCmd(CMD_IS_FINGER_PRESSED, 0) > 0) {
                        delay(200);    }
        } else {
                while(sendCmd(CMD_IS_FINGER_PRESSED, 0) == 0) {
                        delay(200);    }
        }
}
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
void enrollFinger() {
	int id = 0;
	int ret = 0;
	id = sendCmd(CMD_GET_ENROLL_COUNT, 0);
	sendCmd(CMD_ENROLL_START, id);
	peep(1,lowPeep);
	WaitForFinger(false);
	ret = sendCmd(CMD_CAPTURE_FINGER, 1);
	if(ret < 0) {
		peep(5,highPeep);
		return;
	}
	sendCmd(CMD_ENROLL_1, 0);
	peep(1,highPeep);
	WaitForFinger(true);
	WaitForFinger(false);
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
        ret = sendCmd(CMD_CAPTURE_FINGER, 1);
        if(ret < 0) {
                peep(5,highPeep);
                return;
        }
        sendCmd(CMD_ENROLL_2, 0);
        peep(2,highPeep);
        WaitForFinger(true);
        WaitForFinger(false);
        ret = sendCmd(CMD_CAPTURE_FINGER, 1);  if(ret < 0) {
                peep(5,highPeep);
                return;
        }
        sendCmd(CMD_ENROLL_3, 0);
        peep(3,highPeep);
        WaitForFinger(true);
}
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
void setup() {
	Serial.begin(115200);
	Serial3.begin(9600);
	Serial3.setTimeout(10*1000);
	delay(100);
	sendCmd(CMD_OPEN, 0);
	sendCmd(CMD_LED, 1);
	pinMode(resetButtonPin, INPUT);
	pinMode(addButtonPin, INPUT);
	pinMode(lockPin, OUTPUT);
	pinMode(speakerPin, OUTPUT);
	digitalWrite(resetButtonPin, HIGH);
	digitalWrite(addButtonPin, HIGH);
	for(int i = 0; i < 20; i++) {
		pulseServo(lockPin, closed);
	}
}
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
void loop() {
    if (digitalRead(resetButtonPin) == LOW) {    // 盒子的重置按钮会删除所有已存储的指纹
        if(sendCmd(CMD_DELETE_ALL, 0) >= 0) {
            peep(5,lowPeep);
        } else {
            peep(2,lowPeep); // 已经清空
        }
    }
    if (digitalRead(addButtonPin) == LOW) {      // 添加按钮存储欲授权打开盒子的指纹
        enrollFinger();
    }
    if(sendCmd(CMD_GET_ENROLL_COUNT, 0) == 0) {  // 如果没有授权的指纹，也就没有必要扫描和检查指纹了
        delay(100);
        return;
    }
    if(sendCmd(CMD_IS_FINGER_PRESSED, 0) == 0) {
        sendCmd(CMD_CAPTURE_FINGER, 1);
        int ret = sendCmd(CMD_IDENTIFY, 0);
```

Example 9-7. ancient_chest.ino

```
// ancient_chest.ino - fingerprint unlocks chest
// (c) BotBook.com - Karvinen, Karvinen, Valtokari
        if(ret >= 0 && ret < 200) {      // 找到了一枚已授权打开盒子的指纹
            if(state == 0) {
                peep(1,highPeep);
                for(int i = 0; i < 20; i++) {
                    pulseServo(lockPin, opened);        //打开锁具。
                    //变量opened知识存储了一个1ms的脉冲常数，使舵机转到最小的角度上。
                }
                state = 1;          Serial.println("Open");
            } else {
                Serial.println("Close");
                peep(1,lowPeep);
                for(int i = 0; i < 20; i++) {
                    pulseServo(lockPin, closed);
                }
                state = 0;
            }
        } else {
            peep(5,lowPeep);
        }
        WaitForFinger(true);
    }
}
```