

1-1 簡介

Arduino 是由義大利米蘭互動設計學院 Massimo Banzi、David Cuartielles、Tom Igoe、Gianluca Martino、David Mellis 及 Nicholas Zambetti 等核心開發團隊成員所創造。Arduino 開發板是一塊**開放源碼 (open-source)** 的微控制器 (Micro Control Unit, 簡稱 MCU) 電路板, 軟體原始碼與硬體電路都是開源的。可在 Arduino 官網上購買正版 Arduino 開發板, 或是相關網站購買相容 Arduino 開發板。如圖 1-1 所示 Arduino 註冊商標, 使用一個無限大符號來表示「**實現無限可能的創意**」。Arduino 原始設計目的, 是希望設計師及藝術師不用學習複雜的單晶片結構及指令, 就能夠快速、簡單的設計出與真實世界互動的應用產品。



圖 1-1 Arduino 註冊商標

1-2 Arduino 硬體介紹

Arduino 開發板使用 Microchip/Atmel 公司研發的低價位 ATmega AVR 系列 MCU。從第一代 ATmega8、ATmega168 到現在的 ATmega328, 皆為 8 位元 MCU, 28 腳 DIP 包裝。Arduino 開發板種類很多, 主要差異是內置 MCU 及使用的 USB 介面 IC 不同, 但是程式語法與硬體連接方式大致相同。如表 1-1 所示 ATmega 系列 MCU 內部記憶體容量比較, 以最受歡迎的 Arduino UNO R3 開發板為例, 內置 ATmega328 晶片, 16MHz 時脈, 具有 32kB Flash ROM、2kB SRAM 及 1kB EEPROM。新版 Arduino UNO R4 開發板採用 32 位元 MCU, 48MHz 時脈, 內置 Renesas RA4M1 晶片, 具有 256kB Flash ROM、32kB SRAM 及 8kB EEPROM。

表 1-1 ATmega 系列 MCU 內部記憶體容量比較

記憶體	ATmega8	ATmega168	ATmega328	ATmega1280	ATmega2560
Flash ROM	8kB	16kB	32kB	128kB	256kB
SRAM	1kB	1kB	2kB	8kB	8kB
EEPROM	512bytes	512bytes	1kB	4kB	4kB

1-2-1 UNO R3 板

如圖 1-2 所示 Arduino UNO R3 板，是整個 Arduino 家族中使用最多的開發板，以第三版（Release 3，簡稱 R3）最為人熟知。「UNO」義大利文意思是「一」，用來紀念 Arduino 1.0 的發布。UNO R3 板使用 8 位元 ATmega328P 微控制器，時脈速度 16 MHz，P 代表**低功耗（Pico power）**。板載另一顆微控制器 ATmega16u2，取代 FTDI 公司的 USB 介面 IC，用來處理 USB 傳輸通訊。

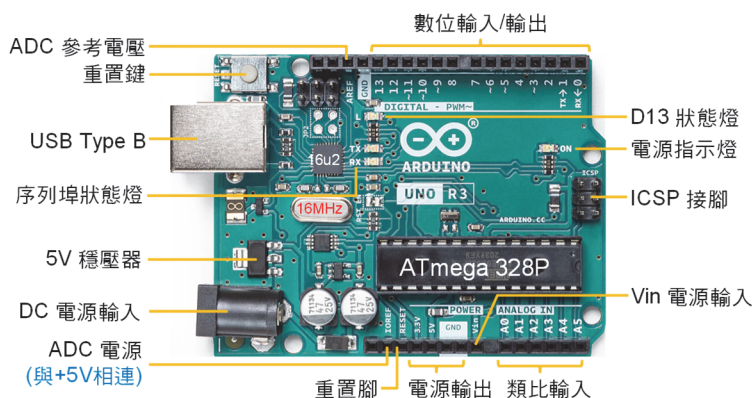


圖 1-2 Arduino UNO R3 板（圖片來源：arduino.cc）

在電源部分，UNO R3 板輸入直流電壓範圍 6~20V，小於 7V 時電壓變得不穩定，大於 12V 時穩壓 IC 過熱將導致損壞，一般**建議值在 7~12V**。UNO R3 板有三種供電方式，第一種方式是直接使用 USB Type-B 供電，第二種方式是外接 7~12V 直流電源到 DC 電源輸入。第三種方式是外接 7~12V 直流電源到 Vin 電源輸入。第二及第三種方式都是使用板載電壓調整器穩壓輸出 5V 及 3.3V，再提供給外部模組使用。5V 電源最大輸出電流 300mA，3.3V 電源最大輸出電流 50mA。

在記憶體部分，UNO R3 板內含 32kB 快閃（Flash）記憶體（啟動程式 Bootloader 使用 0.5kB），2kB 靜態隨機存取記憶體（Static Random Access Memory，簡稱 SRAM）及 1kB 電子抹除式可覆寫唯讀記憶體（Electrically-Erasable Programmable Read-Only Memory，簡稱 EEPROM）。

UNO R3 板包含 14 支數位輸入 / 輸出腳（Input / Output，簡稱 I/O）D0~D13 及 6 支 10 位元解析度類比輸入腳 A0~A5。每支數位腳具有 20mA 電流驅動能力，其中 D3、D5、D6、D9、D10、D11 等 6 支數位腳，可輸出脈寬調變訊號（Pulse Width Modulation，簡稱 PWM），腳位上標記**正弦波符號「~」**以方便辨識。類比輸入 A0~A5

預設使用+5V 參考電壓（不同參考電壓可以由 AREF 腳輸入），輸出數位值範圍 0~1023。A0~A5 不用時，可以當數位腳 D14~D19 使用。

在通訊部分，UNO R3 板支援非同步收發傳輸器（Universal Asynchronous Receiver / Transmitter，簡稱 **UART**）、積體電路介面匯流排（Inter-Integrated Circuit Bus，簡稱 **I2C**）及串列周邊介面（Serial Peripheral Interface，簡稱 **SPI**）等多種串列雙向資料傳輸。I2C 發音「I-square-C」，於 1982 年由荷蘭飛利浦（Philips）半導體公司開發，主要目的是讓 MCU 或 CPU 以較少接腳連接眾多的低速周邊裝置。UNO R3 開發板內建一組 UART 硬體串列埠（D0:RX，D1:TX）、一組 I2C 介面（A4:SDA，A5:SCL），一組 SPI 介面（D10:SS，D11:MOSI，D12:MISO，D13:SCK）。

UNO R3 板支援兩個外部中斷（interrupt）INT0（D2）及 INT1（D3）。提供五種中斷方式，分別是正緣（RISING）觸發、負緣（FALLING）觸發、訊號改變（CHANGE）觸發、高電位觸發及低電位觸發等。

在 UNO R3 板右邊有一組如圖 1-3 所示線上串列燒錄（In-Circuit Serial Programming，簡稱 ICSP）接頭，功用與 SPI 介面相同。ICSP 用來將 **Bootloader 程式燒錄到 ATmega328P 微控制器中**。Bootloader 程式讓我們可以使用 USB 直接將 PC 端原始程式碼上傳到 ATmega328P 微控制器中執行。板載串列埠狀態指示燈 TX、RX，可用來指示目前串列埠的通訊狀態。



圖 1-3 ICSP 接頭

1-2-2 UNO R4 WiFi 板

如圖 1-4 所示 Arduino UNO R4 WiFi 板，結合日本瑞薩（Renesas）電子 RA4M1 微控制器處理能力及樂鑫（Espressif）科技 ESP32-S3 微控制器的 WiFi、藍牙（Bluetooth）無線連接能力。主核心 RA4M1 採用 32 位元 ARM Cortex®-M4 架構，時脈速度 48MHz，內建 256kB Flash ROM、32kB SRAM 及 8kB EEPROM。ESP32-S3 微控制器採用 32 位元 Xtensa LX7 雙核心處理器，時脈速度最大可達 240MHz，內建 Wi-Fi 及低功耗藍牙（Bluetooth Low Energy，簡稱 BLE）功能。

在電源部分，UNO R4 WiFi 板提供三種供電方式，第一種方式是直接使用 USB Type-C 供電，第二種方式是外接 6~24V 直流電源到 DC 電源輸入，第三種方式是外接 6~24V 直流電源到板上的 Vin 電源輸入，一般電源輸入**建議值為 7~12V**。

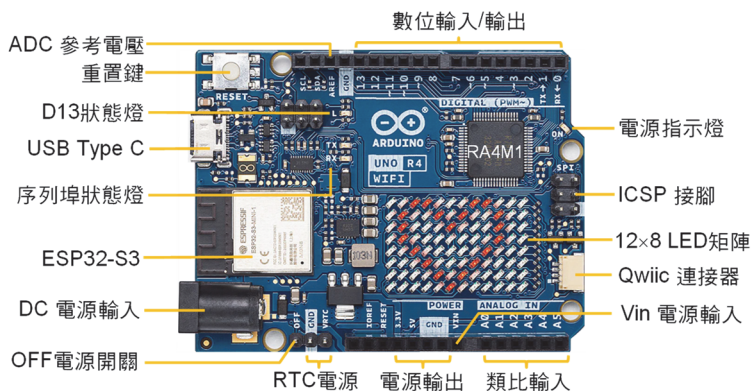


圖 1-4 Arduino UNO R4 WiFi 板（圖片來源：arduino.cc）

UNO R4 WiFi 板與 UNO R3 板接腳排列相同，包含 14 支數位腳 D0~D13（最大輸出電流 8mA），6 支類比輸入腳 A0~A5（最大 14 位元解析度），6 支 PWM 輸出（數位腳 D3、D5、D6、D9、D10、D11），一組 UART 硬體串列埠（D0:RX，D1:TX），一組 I2C 介面（A4:SDA，A5:SCL），以及一組 SPI 介面（D10:SS，D11:MOSI，D12:MISO，D13:SCK）。UNO R4 WiFi 板新增多種周邊，包含網域控制器（Controller Area Network，簡稱 CAN）通訊介面、12 位元數位類比轉換器（Digital to analog converter，簡稱 DAC）、運算放大器（Operational Amplifier，簡稱 OP AMP）、快速型 I2C（Qwick I2C，簡稱 Qwiic）連接器，以及人機介面裝置（Human Interface Device，簡稱 HID）。另外，板載提供即時時鐘（real-time clock，簡稱 RTC）及 12×8 LED 矩陣，可以讓創客實現無限可能的創意。將 OFF 腳接地可以關閉 Arduino 主電源。

1-2-3 UNO R4 Minima 板

如圖 1-5 所示 Arduino UNO R4 Minima 板，與 UNO R4 WiFi 板比較，主要少了 **ESP32-S3 微控制器及板載 12×8 LED 矩陣**。

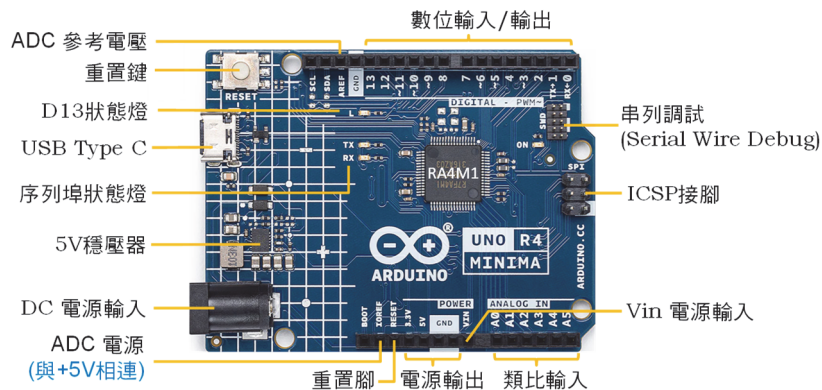


圖 1-5 Arduino UNO R4 Minima 板（圖片來源：arduino.cc）

如表 1-2 所示 Arduino 家族 UNO 開發板特性比較，包含 UNO R3 板、UNO R4 WiFi 板及 UNO R4 Minima 板。

表 1-2 Arduino 家族 UNO 開發板特性比較

主要特性	UNO R3	UNO R4 WiFi	UNO R4 Minima
微控制器	8 位元 / ATmega328P	32 位元 / RA4M1	32 位元 / RA4M1
時脈速度	16MHz	48MHz	48MHz
VIN 輸入電壓	6~20V	6~24V	6~24V
每個 I/O 電流	20mA	8mA	8mA
數位 I/O	14	14	14
類比輸入	6	6	6
PWM 輸出	6	6	6
Flash ROM	32kB	256kB	256kB
SRAM	2kB	32kB	32kB
EEPROM	1kB	8kB	8kB
UART	1	1	1
SPI	1	2	2
I2C	1	2	2
CAN BUS	—	1	1
DAC（12 位元）	—	1	1
OP AMP	—	1	1
RTC	—	1	1
Qwiic I2C 連接器	—	1	—

5-1 認識開關

如圖 5-1 所示常用機械開關，如**搖動開關**、**滑動開關**、**指撥開關**及**按鍵開關**等，主要用途是接通（ON）或斷開（OFF）電路。機械開關是利用金屬片與接點接觸而產生接通狀態，一般會在接點上電鍍抗腐蝕金屬，以避免因氧化物所產生的接點接觸不良現象。有時也會改用導電塑膠等非金屬接觸面材料，來提高接點接觸可靠性。



圖 5-1 常用機械開關

5-1-1 指撥開關

如圖 5-2 所示指撥開關（DIP switch）屬於單刀單投開關，依其開關組合數量可分為 1P、2P、4P、8P 等多種包裝。如圖 5-2(a)所示為 4P 元件，開關向上撥為 ON，向下撥為 OFF。如圖 5-2(b)所示為 4P 符號。



圖 5-2 指撥開關

如圖 5-3 所示指撥開關電路，有兩種接線方式。圖 5-3(a) 所示高電位產生電路，開關 OFF 時，輸出電壓 $V_o = 0$ ，開關 ON 時，輸出電壓 $V_o = +5V$ 。圖 5-3(b) 所示低電位產生電路，開關 OFF 時，輸出電壓 $V_o = +5V$ ，開關 ON 時，輸出電壓 $V_o = 0$ 。



圖 5-3 指撥開關電路

5-1-2 按鍵開關

如圖 5-4 所示按鍵開關，常應用在電話按鍵、手機按鍵及電腦鍵盤等。平時按鍵為 OFF 狀態，A、B 兩點不通，按下按鍵時為 ON 狀態，A、B 兩點導通。

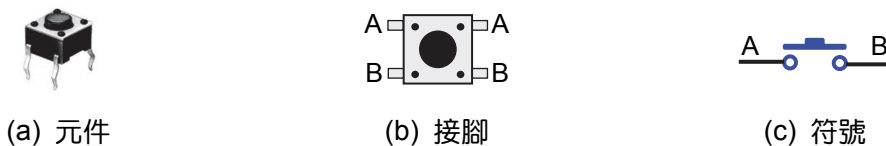


圖 5-4 按鍵開關

如圖 5-5 所示按鍵開關電路，有兩種接線方式，如圖 5-5(a) 所示正脈波產生電路，平時開關為 OFF 狀態，輸出電壓 $V_o = 0$ ，按住開關時，輸出電壓 $V_o = +5V$ 。如圖 5-5(b) 負脈波產生電路，平時開關為 OFF 狀態，輸出電壓 $V_o = +5V$ ，按住開關時，輸出電壓 $V_o = 0$ 。



圖 5-5 按鍵開關電路

理想上每按一下按鍵開關，輸出只會產生一個脈波輸出。實際上開關有**機械彈跳** (bounce) 的問題存在。所謂彈跳是指每按一下按鍵，可能產生不固定次數的脈波輸出。如圖 5-6 所示脈波產生電路的機械彈跳現象，彈跳時間約為 **10ms~20ms**。本章使用**延遲除彈跳程序**來解決彈跳問題，程序簡單，原理是當檢測有按鍵產生時，先延遲一段時間以避開不穩定狀態，等待放開按鍵再處理指定動作。延遲時間過短，無法有效消除彈跳，延遲時間過長則按鍵反應不敏靈。

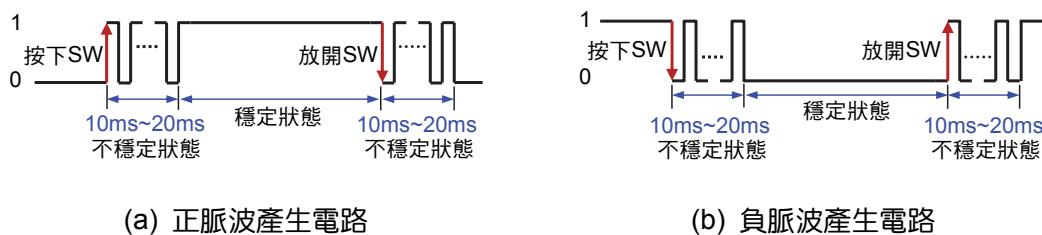


圖 5-6 脈波產生電路的機械彈跳現象

5-3-3 按鍵開關控制一個 LED 亮與暗實習

一 功能說明

如圖 5-11 所示電路接線圖，使用按鍵開關控制一個 LED 亮與暗。每按一下按鍵開關，LED 改變狀態，依序為 ON→OFF→ON。

二 電路圖及麵包板接線圖

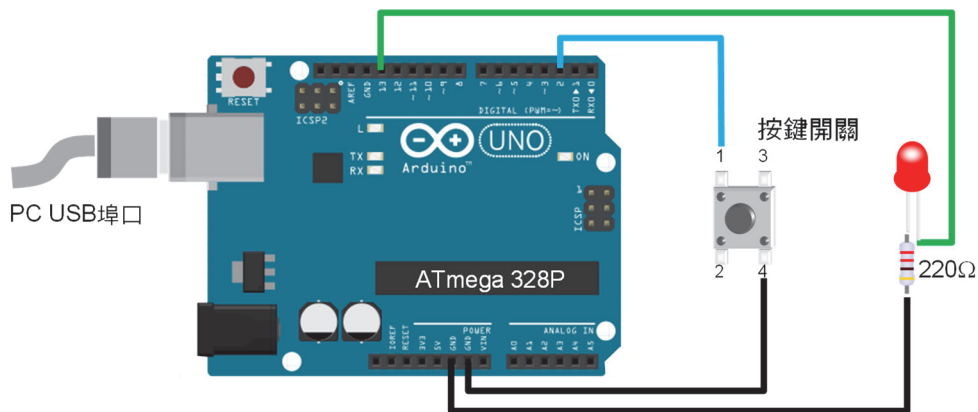


圖 5-11 按鍵開關控制一個 LED 亮與暗實習電路接線圖

三 程式：ch5_3.ino

```
const int sw=2;                //按鍵開關連接至數位腳 D2。
const int led=13;              //LED 連接至數位腳 D13。
const int debounceDelay=20;    //消除機械彈跳延遲時間。
int ledStatus=LOW;              //LED 初始狀態為 LOW。
int val;                        //按鍵開關狀態。
//初值設定
void setup()
{
    pinMode(sw, INPUT_PULLUP);    //設定 D2 為輸入埠，使用內建提升電阻。
    pinMode(led, OUTPUT);         //設定 D13 為輸出模式。
}
//主迴圈
void loop()
{
    val=digitalRead(sw);          //讀取按鍵開關狀態。
    if(val==LOW)                  //按下按鍵開關？
    {
```


<code>delay(debounceDelay);</code>	<code>//消除按鍵開關的機械彈跳。</code>
<code>while(digitalRead(sw)==LOW)</code>	<code>//已放開按鍵開關？</code>
<code>;</code>	<code>//等待放開按鍵開關。</code>
<code>ledStatus=!ledStatus;</code>	<code>//改變 LED 狀態。</code>
<code>digitalWrite(led,ledStatus);</code>	<code>//設定 LED 狀態。</code>
<code>}</code>	
<code>}</code>	

練習

1. 接續範例，每按一下按鍵開關，LED 的狀態改變。若 LED 原來狀態為暗，按一下按鍵開關，LED 閃爍，若 LED 原來狀態為閃爍，按一下按鍵開關，LED 暗。
2. 設計五段調光燈，使用按鍵開關控制連接於 D10 的 LED 亮度。每按一下按鍵，LED 亮度增加一級，亮度依序為：暗→1 級→2 級→3 級→4 級→5 級→暗。

5-3-4 按鍵開關控制四個 LED 單燈移位實習

一 功能說明

如圖 5-12 所示電路接線圖，使用按鍵開關控制四個 LED 移位方向。每按一下按鍵，LED 改變移位方向，若 LED 原為單燈右移，按一下按鍵變為單燈左移；若 LED 原為單燈左移，按一下按鍵變為單燈右移。

二 電路接線圖

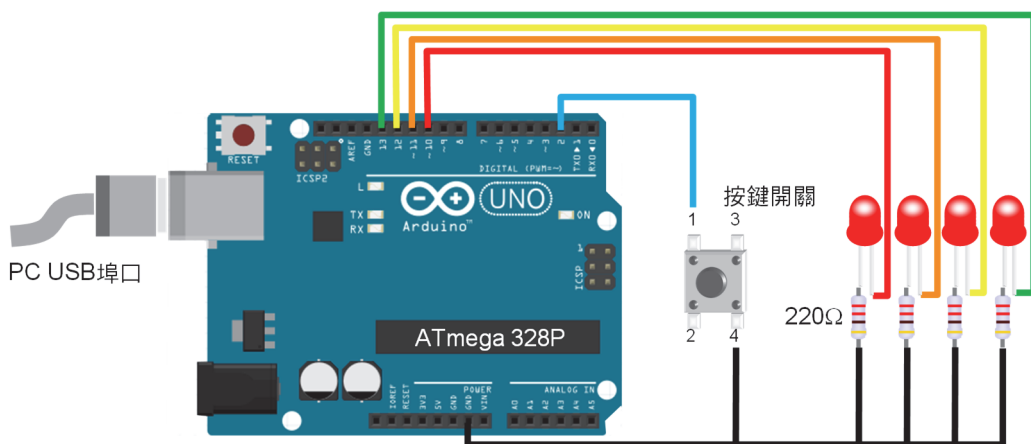


圖 5-12 按鍵開關控制四個 LED 單燈移位實習電路接線圖

11-1 認識聲音

聲音是一種**波動**，聲音的振動會引起空氣分子有節奏的振動，使周圍的空氣產生疏密變化，形成疏密相間的縱波，因而產生了聲波。人耳可以聽到的聲音頻率範圍在 **20Hz~20kHz** 之間。如圖 11-1 所示蜂鳴器及喇叭，常用來將電能轉換成聲能。蜂鳴器依其驅動方式可以分成**有源**及**無源**兩種，如圖 11-1(a) 所示有源蜂鳴器，內含振盪電路且底部被密裝起來，加上直流電壓可以產生固定頻率的聲音輸出。如圖 11-1(b) 所示無源蜂鳴器內部不含振盪電路，底部明顯可以看到電路板，依所加的交流訊號頻率不同，所發出的音調也不同。如果要用來產生音樂輸出，可以使用無源蜂鳴器或是使用如圖 11-1(c) 所示喇叭。



圖 11-1 蜂鳴器及喇叭

如圖 11-2(a) 所示正弦波是組成聲音的基本波形，音量與波形振幅成正比，而音調與波形週期成反比。Arduino 利用 **tone()函式** 產生如圖 11-2(b) 所示方波訊號，來模擬真實聲音，一次只能輸出一種音調。



圖 11-2 聲音訊號

11-1-1 音符

如圖 11-3 所示鋼琴鍵盤排列，共有 88 個鍵，分成 52 個白鍵及 36 個黑鍵，最低音符 A0（頻率 28Hz），最高音符 C8（頻率 4186Hz）。**任何一個八音度（octave）都有 12 個音符，由對比強烈的 7 個白鍵及 5 個黑鍵組成。**白鍵依序表示全音 Do、

Re、Mi、Fa、So、La、Si，簡符為 C、D、E、F、G、A、B。黑鍵依序表示半音 Do#、Re#、Fa#、So#、La#，簡符為 C#、D#、F#、G#、A#。半音是介於兩個全音之間的音符，以井字升音記號「#」依附在全音後面來表示。例如 C#比 C 多半音，而比 D 少半音。位於鋼琴鍵盤中間位置的中央（middle）C 為 C4，其頻率為 262Hz。在中央 C 之上的 A4 為標準音，頻率為 440Hz，用來調校樂器。

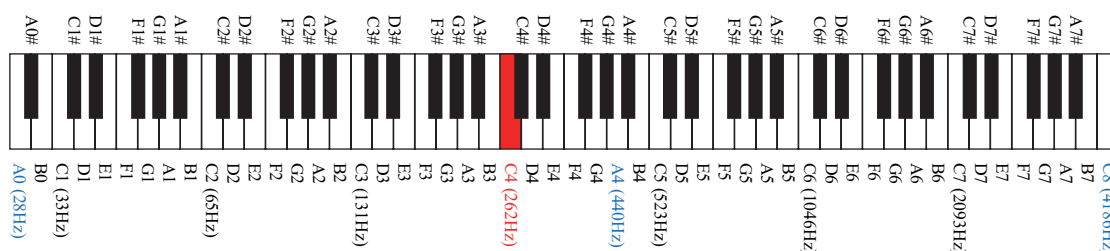


圖 11-3 鍵鋼琴鍵盤排列

音樂也是一種語言，使用如圖 11-4 所示五線譜來記錄樂譜中的音符。符號 treble clef 是高音譜記號，用來定義後面音符的音高，由五線譜下方橫線開始，依序為 C4、D4、E4、F4、G4、A4、B4、C5、D5、E5 等音符。

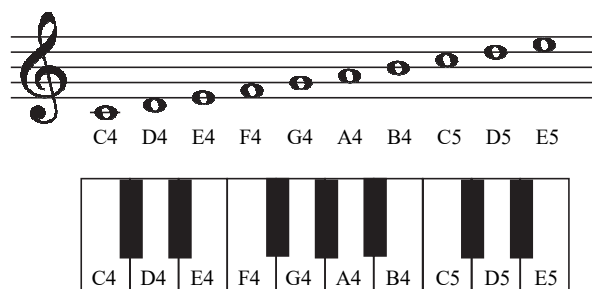


圖 11-4 五線譜

在樂譜中，使用不同種類的音符（note）來表示不同音長。如圖 11-5(a) 所示全音符（whole note），使用空心音符表示，是時間最長的音符。如圖 11-5(b) 所示二分音符（half note），使用加直線空心音符表示，是全音符時間長度的一半。如圖 11-5(c) 所示四分音符（quarter note），使用加直線實心音符表示，是全音符時間長度的四分之一。如圖 11-5(d) 所示八分音符（eighth note），使用帶一條尾巴直線實心音符表示，是全音符時間長度的八分之一。如圖 11-5(e) 所示十六分音符（sixteenth note），使用帶兩條尾巴直線實心音符表示，是全音符時間長度的十六分之一。

練習

1. 接續範例，新增連接於 D7~D13 共七個 LED，用來指示目前播放的音符 C、D、E、F、G、A、B 七個音符。
2. 接續範例，新增兩首樂曲，開機時預設為靜音。每按一下按鍵，依序變換音樂：小蜜蜂→小星星→樂曲 1→樂曲 2→靜音。

11-3-6 倒車警示器實習

一 功能說明

如圖 11-15 所示電路接線圖，使用 Arduino UNO 開發板及超音波感測器，完成倒車警示器。當車尾與障礙物距離 30 公分以上，蜂鳴器不發聲且 LED 燈不亮。車尾與障礙物距離愈近，嗶聲及 LED 閃爍速度愈快。

二 電路接線圖

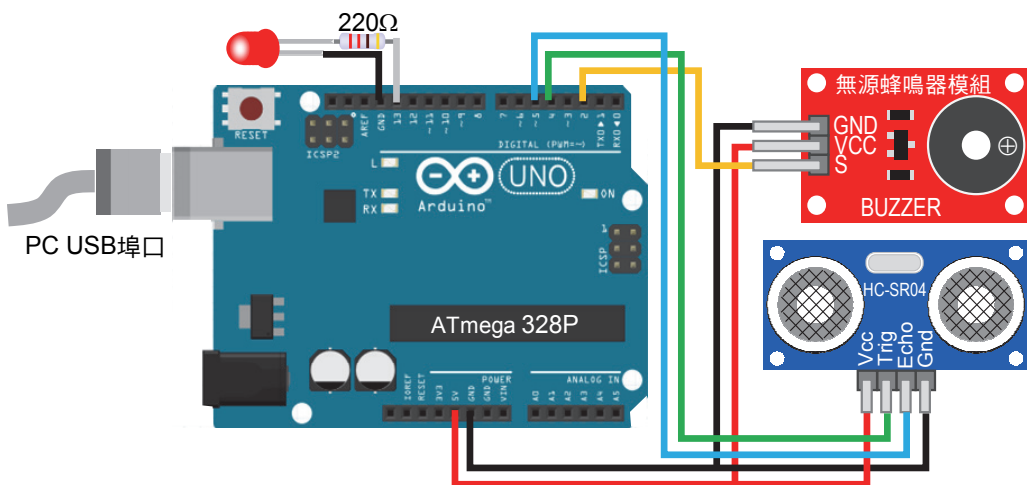


圖 11-15 倒車警示器實習電路接線圖

三 程式：ch11_6.ino

```
const int buzzer=2;           //D2 連接蜂鳴器輸出 S。
const int Trig=4;             //D4 連接超音波模組 Trig 腳。
const int Echo=5;             //D5 連接超音波模組 Echo 腳。
const int led=13;             //D13 連接 LED 燈。
unsigned int distance;         //車尾障礙物距離。
```

//初值設定

```

void setup() {
    pinMode(buzzer, OUTPUT); //設定 D2 為輸出埠。
    pinMode(Trig, OUTPUT); //設定 D4 為輸出埠。
    pinMode(Echo, INPUT); //設定 D5 為輸入埠。
    pinMode(led, OUTPUT); //設定 D13 為輸出埠。
}
//主迴圈
void loop() {
    distance=measure(); //讀取車尾障礙物距離，單位 cm。
    if(distance>30) //障礙物距離超過 30 公分？
    {
        noTone(buzzer); //蜂鳴器不發聲。
        digitalWrite(led, LOW); //LED 不亮。
    }
    else //障礙物距離小於 30 公分。
    {
        digitalWrite(led, HIGH); //LED 點亮。
        tone(buzzer, 1000); //蜂鳴器發出 1kHz 嗶聲。
        delay(distance*10); //障礙物距離愈近，嗶聲及 LED 閃爍速度愈快。
        digitalWrite(led, LOW); //LED 不亮。
        noTone(buzzer); //蜂鳴器不發聲。
        delay(distance*10); //障礙物距離愈近，嗶聲及 LED 閃爍速度愈快。
    }
}
//距離測量函式
unsigned int measure(void) {
    unsigned long duration; //車尾障礙物距離與脈寬成正比。
    unsigned int cm; //車尾障礙物距離，單位 cm。
    digitalWrite(Trig, LOW); //輸出負脈波觸發訊號，啟動超音波感測器。
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
    duration=pulseIn(Echo, HIGH); //讀取與障礙物距離成正比的脈寬。
    cm=duration/29/2; //計算車尾障礙物距離。
    return cm; //回傳距離值給主函式。
}

```