

Arduino實作五：可見光通訊裝置

ASCII

ASCII code是基於英文字母的一套編碼系統，在ASCII中總共可以顯示95種字元，包括英文字母A~Z的大小寫、數字及半形標點符號。一個ASCII編碼的字元大小為8 bits，由於組合數量少，無法顯示英文以外的多國文字，現代編碼多採用能夠和ASCII兼容的unicode。

在Arduino上轉換ASCII

在Arduino程式碼中，若將正整數指定給char變數，會自動使用ASCII編碼進行轉換。比如說字元「a」的ASCII碼為0110 0001，換成十進位制等於97，若在Arduino程式中輸入：

```
char word = 97;
```

會將word這個char變數指定成字元「a」。而在ASCII編碼中，英文字母是連續排列的。因此也可以將char變數進行加減來得到有相對排序的字元。比如輸入：

```
word += 5;
```

則此時word就從「a」變成了「f」。反過來也可以將char指定給int變數，一樣可以透過ASCII編碼轉換。

練習一：轉換ASCII編碼

在序列埠監控視窗輸入任意字元，程式必須轉換並顯示出該字元的ASCII code和十進位制的形式。

EX: 輸入 a

輸出 letter: a, ASCII code: 0110 0001, ASCII in decimal: 97.

提示: 使用Serial.read()來讀取序列埠監控視窗輸入的字元。

RLL編碼

RLL(Run Length Limited，游程長度限制)編碼是一種將較少的二進位碼改編成較長的編碼的方法。雖然會使傳輸速度下降(因為單位時間內能傳的位元固定)，但有以下兩點好處：

1. 限制連續的0或1出現的次數，即為「游程長度限制」的由來。
2. 由於第1點的限制，可以將改編後不會出現的位元組拿來做傳資料以外的用途。

RLL編碼有許多種類，一般表示為RLL(a,b)，代表會連續出現的0的數量最少為a個最多為b個。

RLL(0,1)

原始位元組	改編後位元組
0	10
1	11

RLL(0,2)

原始位元組	編碼後位元組	原始位元組	編碼後位元組
0000	11001	1000	11010
0001	11011	1001	01001
0010	10010	1010	01010
0011	10011	1011	01011
0100	11101	1100	11110
0101	10101	1101	01101
0110	10110	1110	01110
0111	10111	1111	01111

封包

在各式各樣的通訊形式中，都需要一個裝置間約定好的「通訊協定」，以分辨接收到的訊號是否來自正確的發送源，以及判斷要如何解碼這個訊號。傳輸資料時依據各種通訊協定將欲傳輸的內容分裝成一個個的「封包」，一個封包包含以下兩個部份：

1. 資料表頭(header): 根據通訊協定定義的固定形式位元組，當接收端讀到header代表一個封包的開始。
2. 酬載(payload): 接續在header之後，記錄發送端欲傳輸的資料本身。

設計自己的封包?

結合前面所學的RLL編碼和封包的概念，是否能夠設計一個通訊協定，使payload以加入RLL(0,2)的ASCII code編碼傳輸，並包含了獨特的header可以辨別封包的開頭，同時又不會和payload搞混呢?請同學互相討論並提出自己設計的封包結構。

練習二：可見光通訊傳輸

如何將數字轉換輸出???

Ex. 111→0110 1111 我們存入bits string 但是方向剛好相反 1111 0110

之後1111 轉換RLL(0,2) →01111

0110轉換RLL(0,2) →10110

存成bitsRLL→0111110110

灌入Header 10101000

輸出→101010000111110110

Practice 8-2 flow:

輸入數字→按A確認→將輸入數字改成二進位的string(bits)→利用function改成RLL(0,2)，並命名為bitsRLL→傳輸發光亮暗(搭配header)

練習二：可見光通訊傳輸

1. 先將int v 轉換成二進位，並加入bits string 中
 - 利用迴圈判斷二進位 ex. 25得到000011001則bits='10011000'
2. 完成transmit function
 - 此function需要先傳出header，接著依照bitsRLL[k]共十碼，1要亮 0要暗開關LED

***建議header 開關開關開關關關