

103年度第二學期 嵌入式系統 Final Project

題目：Lifi 無線資料光傳輸裝置

組別：第三組

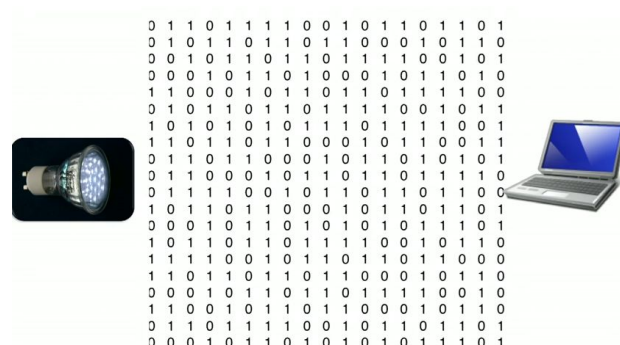
組員：

r03525060 工科碩一 陳加容

r03525056 工科碩一 陳雲濤

b01705008 資管三 張家豪

b01705045 資管三 湯梵平

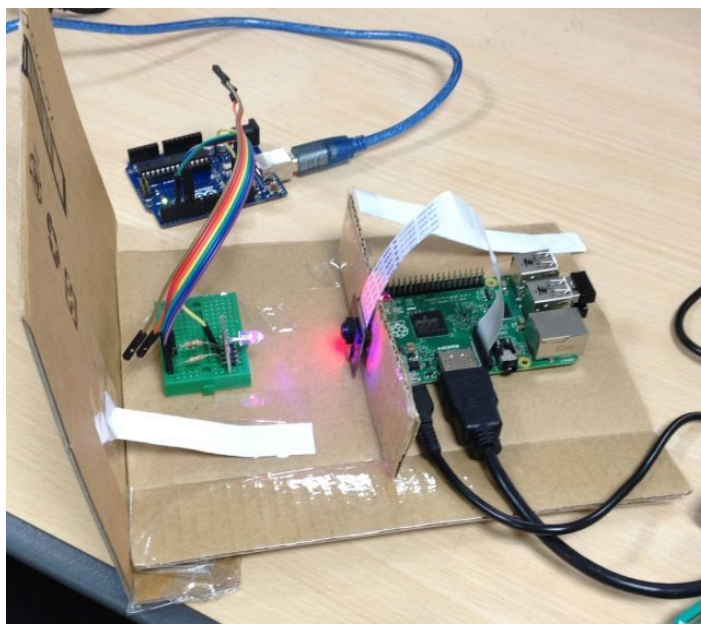


目的與動機：

開車在路上時，最常看到人家的車尾燈，以及車頭燈，如果能夠使用光來做簡單的資訊傳輸，可以很方便地與前後方車輛溝通，例如當我想要超車時可以先告知前方車輛，或是當我方想要停靠到路邊或想要減速時，也可以透過後車燈來告知後方車輛，如此可降低事故發生的可能性。

介紹：

本Project我們使用Arduino搭配三色LED燈泡作為傳輸端，使用者在電腦中輸入想要傳的字串資料，透過Arduino將該字串資料轉為bit code，並且利用LED燈的閃爍將bit呈現出來，接著，再透過Raspberry Pi 接上的相機鏡頭來拍攝該閃爍的光，利用open CV影像辨識來解析接收到的資料，並且連接到外接螢幕上將字串呈現出來。

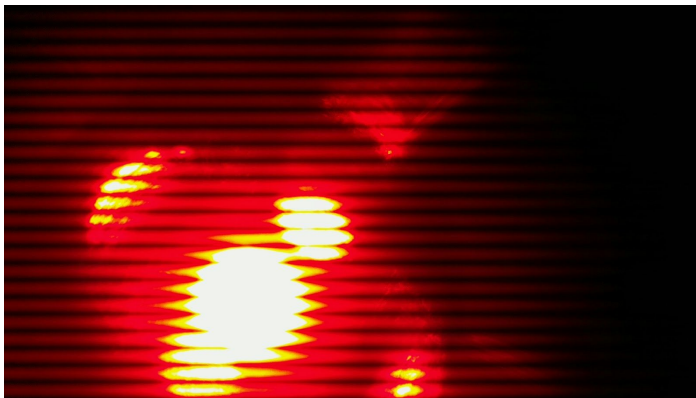
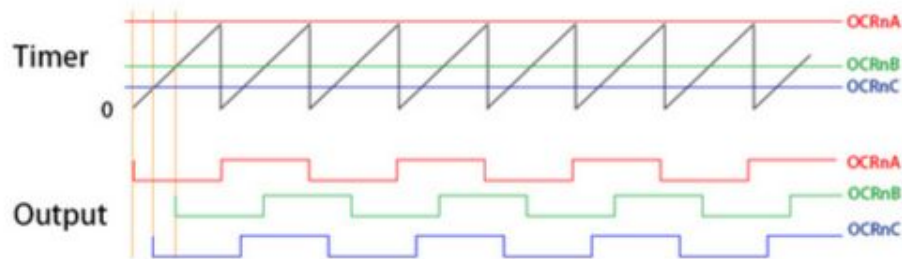


器材：Arduino UNO, RGB三色LED燈, 麵包板, Raspberry Pi 2, Pi Camera NoIR Board, 外接螢幕等

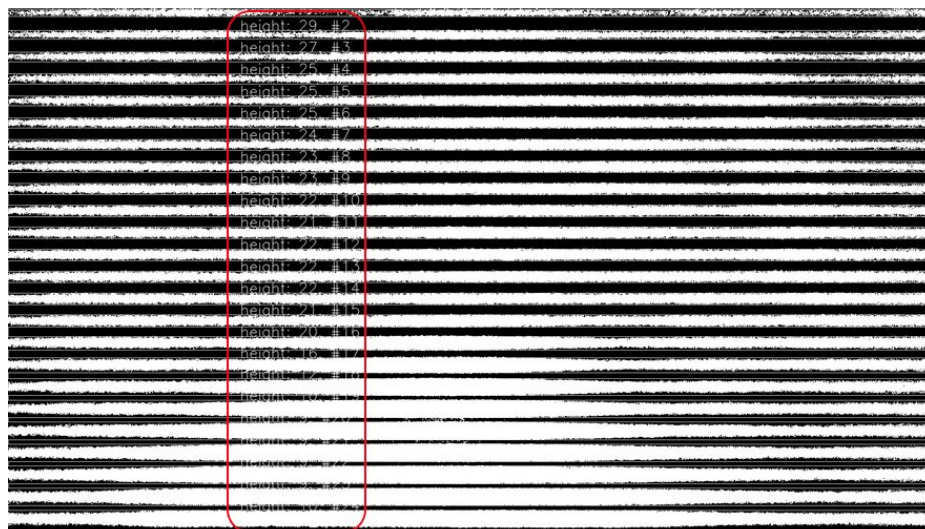
實作過程：

起先我們想要利用產生與解析光頻率(水波紋 Moire Pattern)來傳送與接收資料，我們使用Arduino來控制LED燈閃爍頻率30Hz，並且Raspberry Pi接收相機鏡頭設定shutter speed = 0.001來接收

Generate accurate square waves (CTC Mode or FSK)



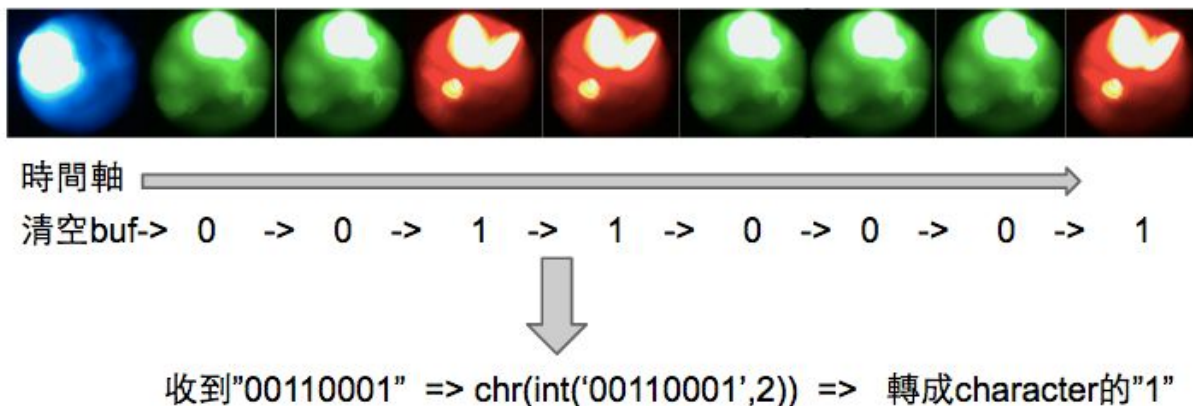
拍攝到影像後再利用open CV來解析波紋之間的寬度



如上圖，我們所解析出來的寬度理當是全部等寬的，但是因為下方亮度較高，導致解析出來的寬度比較小，如此無法做到精準地傳遞資料。

於是我們改採用以光的三原色RGB (Red, Green, Blue)來傳輸資料
例如我們要傳輸 '1' 如下圖所示

傳輸character"1"



我們設定藍色做為清空buffer, initial與結束的訊號, 綠色代表 0, 紅色代表 1, 來傳輸bit資訊, 並且設定傳輸的頻率為 1000/33 Hz, 接收端則是動態地利用拍照的方式, 每秒的接收frame rate設定成每秒30張frame, 並且利用open CV辨識顏色來接收訊息, 再decode成原本的文字資訊呈現給使用者。

問題與討論

水波紋：

在Raspberry影像辨識端, 拍攝燈泡的照片中, 會有一區塊亮度極高, 造成辨識出來的波紋區段不一致。解決辦法是改用均勻的LED燈管作為傳輸端, 如此接收到的波紋才是均勻的, 或是直接使用color sensor來作為接收端。

傳輸亂碼問題：

以傳輸字串ieso為例, 一開始可以正常地依序地循環傳輸 ieosieos..., 但過一陣子開始會出現 iosieos 或是 eoieos 這種漏掉一個字母的情況, 甚至是開始出現單一個亂碼

造成問題的原因在於, LED傳輸頻率計算與接收端camera的接收frame rate會有些許差距, 我們的LED傳輸頻率設定為30Hz, 但是接收端我們買到的camera的frame rate為 1000/33 與30Hz有 0.001的差距, 造成每接收累計frame超過一定程度的數量時, 就會造成資訊遺失或出現亂碼。

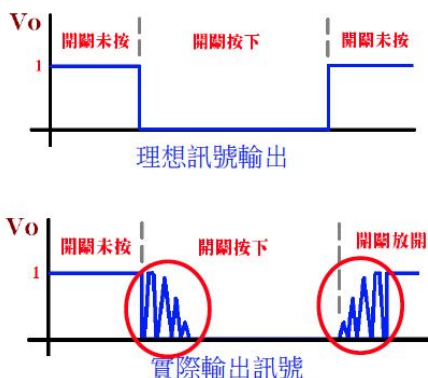
傳輸速度問題：

使用Raspberry Pi迴圈來控制閃爍頻率的極限為 4.2MHz (參考資料：

<https://www.liukang.com/2014/04/shell-script-raspberry-pi-gpio-benchmark/>), 國內也沒辦法買到適合的Lifi接收器可以接收如此高頻率的閃爍光。

心得

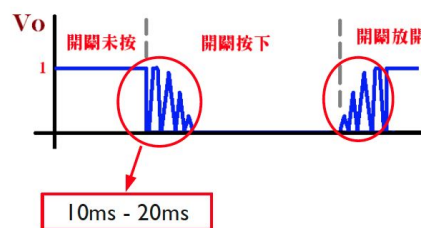
還沒有修這門課之前，我們還沒有想過硬體上的時脈會影響程式碼要如何實作，在本Project實作過程當中，我們接觸了camera以及open CV以及Arduino，麵包板等，在利用讓LED閃爍時，即使我們根據document與說明書提供的規格，計算出確切的參數，使得Raspberry Pi可以算出控制每分鐘亮暗32次(32Hz)，但是實際測試計時的結果發現，有時候只會達到30次或29次，而在改用Arduino來作為傳輸端後，Raspberry Pi接上Camera Board使用PiCamera library來控制接收的frame rate，也是會有些許的參數誤差，在嘗試製作實體電路按鈕的過程中，也要考慮加上消除bounce的RC電路，這些問題都是我們平常寫的軟體程式當中比較少遇到的，經過這門課的學習，使得我們對於開發Embedded System有了初步的經驗與研究，希望以後能夠做出一個完善的軟硬整合服務。



<http://120.101.72.1/Onechip/PPT/實習單元三.ppt>

解決彈跳問題 (de-bounce)

- 硬體方法：以 RC 電路或正回授的比較器電路解決
- 軟體方法：調整觸發的延遲時間
- 不同的按鍵會有不同的延遲時間



134

Source Code

project github:

https://github.com/how2945ard/ieos_data_parsing.git

傳輸端 Arduino:

https://github.com/how2945ard/ieos_data_parsing/blob/master/ieos_final_proj_v2/ieos_final_proj_v2.ino

接收端 Raspberry Pi:

https://github.com/how2945ard/ieos_data_parsing/blob/master/video.py

參考資料

Camera framerate 1% faster than requested

<https://www.raspberrypi.org/forums/viewtopic.php?p=575968>

<https://www.liukang.com/2014/04/shell-script-raspberry-pi-gpio-benchmark/>

QTI sensor

<http://learn.parallax.com/KickStart/555-27401>

Pi-camera focus:

<http://www.raspi.today/how-to-focus-the-raspberry-pi-camera-lens/>

<https://learn.adafruit.com/adafruit-arduino-lesson-3-rgb-leds/arduino-sketch>