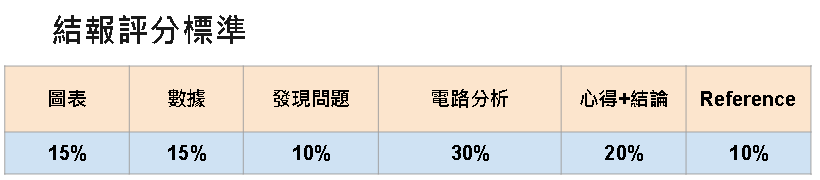
****

\*\*請假後補交結報的規定\*\*

1.請假需依規定提出假單申請，並安排時間補做實驗並將核準過的假單截圖貼於下方，助教才會進行結報的批改。

2.以請假日計算；需在一星期內完成補做驗，二個星期內補交結報(將結報交至Delay區)。逾時不進行結報批改。例如:3/1請假，需在3/8前完成補做實驗，3/15前補交結報。

-------------------------------------------------------------------------------------

**REPORT**

|  |
| --- |
| **Experiment 1: UART protocol** |

1. UART frame waveform

一張含有 文字, 螢幕擷取畫面, 多媒體軟體 的圖片

AI 產生的內容可能不正確。

2. Frame content (Fill the blank with 0 or 1)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | START | Bit0  (LSB) | Bit1 | Bit2 | Bit3 | Bit4 | Bit5 | Bit6 | Bit7  (MSB) | STOP |
| 1st frame | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 2nd frame | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

以上的bit是由y:1111001、a:1100001所構成

3. The interval of a bit is 105u (second) which means the Barud rate is equal to 9523.81 (bps).

* 鮑率推算:

一張含有 筆跡, 文字, 字型, 書法 的圖片

AI 產生的內容可能不正確。

* 那什麼是UART通訊協定?

**UART（Universal Asynchronous Receiver/Transmitter）**通常是嵌入式系統中最常見的通訊接口之一。**UART**是一種串列通訊協議，用於在**MCU**和其他設備之間傳輸數據。**UART**通訊不同於並行通訊，它通常使用少量的引腳，僅需一條傳輸線（**TX**）和一條接收線（**RX**）。

**基本原理**

**UART**通訊是一種點對點通訊方式，其中一個**MCU**充當發送器，另一個**MCU**或設備充當接收器。通的基本原理如下：

1. 發送端將要傳送的數據（字節或字元）傳輸到UART的傳輸線（**TX**）。
2. 接收端使用**UART**的接收線（**RX**）來接收數據。
3. 數據以串列方式傳輸，通常每個字節的起始和結束都有特定的起始位和停止位，以確保數據的正確傳輸。
4. 通的兩端必須配置相同的波特率（9600）以確保正確的數據接收。

* **電路分析**

從實驗一我們可以發現兩個MCU分別是Arduino UNO版和示波器，透過兩條導線讓這兩者連接在一起，由Arduino傳輸訊號然後在示波器上顯示Binary bit的波型(0、1)。

* **問題**

1. 為什麼UART的訊號輸出訊號bit的順序與I2C、SPI顛倒 ?

這主要的差距我覺得是在於因為UART他是非同步訊號(不被CLK控制)，這是與I2C、SPI相比最大的不同。正因為它是非同步訊號，所以他會先從LSB開始先傳，最後才是MSB。

* **結論**

實驗一主要是實做UART通訊協定在Arduino上的應用，透過兩條導線間傳送訊號讓我們理解這通訊協定的原理同時認識到Ardunio對於一個字元適用什麼方式在傳輸的。

|  |
| --- |
| **Experiment 2: SPI protocol** |

1. SPI frame waveform

一張含有 螢幕擷取畫面, 文字, 電子產品 的圖片

AI 產生的內容可能不正確。

2. Frame content (Fill the blank with 0 or 1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Bit7  (MSB) | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0  (LSB) |
| 1st frame | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 2nd frame | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

以上的bit是由w:1110111、e:1111000所構成

3. The frequency of the SCK is equal to 128205.18 Hz

* SCK(serial clock)推算:

一張含有 筆跡, 文字, 字型, 書法 的圖片

AI 產生的內容可能不正確。

* **電路分析**

由Arduino UNO版的SPI 我們可以知道CPOL=0、CPHA=0，所以我們可以根據這數據知道SCK會由低電位到高電位(也就是除了CLK脈衝之外都是低電位)，然後從第一個edge取樣(rising edge)判斷binary number，根據上述的線索我們可以鎖定SCK就是CH1的波型(黃色)，相對的我們可以知道CH2為MOSI的訊號，接下來我們可以透過rising edge取得區段電壓的值作為binary number，最後因為SPI是同步協議，所以會從MSB一路讀到LSB。

* **結論**

實驗二主要是在實做SPI的通訊協議，透過連接Arduino模組的SCK、output接腳觀察波型並記錄clock與被取樣的值之間的關係。

|  |
| --- |
| **Experiment 3: I2C protocol** |

1. I2C frame waveform

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

AI 產生的內容可能不正確。

2. Frame content (Fill the blank with 0 or 1)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Address6 | Address5 | Address4 | Address3 | Address2 | Address1 | Address0 | Read  /Write | ACK |
| 1st frame | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
|  | Bit7  (MSB) | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0  (LSB) | ACK |
| 2nd frame | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 3rd frame | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |

以上的bit是由l:0110 1100、a:0110 0001所構成

3. The frequency of the SCL is equal to 100k Hz

* SCL頻率計算:

一張含有 筆跡, 字型, 書法, 文字 的圖片

AI 產生的內容可能不正確。

* **電路分析**

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

AI 產生的內容可能不正確。I2C protocol 與前兩個協議最大的不同在於支援多對多裝置的連線，今天的I2C通訊協議實驗就是依照這性質而設定的，透過master輸出訊號然後slave讀入訊號的方式通訊，開始讀取於SCL為高電位然後SDA訊號從高電位轉低電位時(圖一)，一路讀到SCL也是高電位，但SDA訊號從低電位轉到高電位結束(圖二)



(圖一) (圖二)一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

AI 產生的內容可能不正確。

* **結論**

實驗三實做I2C通訊協議，透過其多對多的特性讓Arduino之間傳輸姓名訊號，然後再透過SCL的rising clock取值得出正確的binary number。

|  |
| --- |
| **Experiment 4: 2-digit 7-segment display** |

The sketch of your design: (copy from the Arduino IDE window and paste here)

前兩段先讓對應的電晶體打開

隨後設計圖案(1為暗，0則亮)

讓圖案延續一段時間，但不能超過眼睛辨別的頻率，使其在視覺暫留作用下能連續看清兩個數字。

**一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

AI 產生的內容可能不正確。**

Take a photo of your display content.

一張含有 電子產品, 電子工程, 電氣線路, 纜線 的圖片

AI 產生的內容可能不正確。

* **電路分析**

七段顯示器實驗主要分為兩部分。第一部分涉及電晶體控制的原理，通過調節基極（Base）的電壓來控制電晶體的行為。當基極電壓達到高電位時，電晶體進入飽和區，此時可視為一個閉合的開關。然而，若基極電壓過高，電路反而無法正常導通，導致七段顯示器的供電電壓為零，使其無法點亮；反之，當基極電壓適當時，顯示器則能正常發亮。第二部分則是 Arduino 控制的應用，通過設定 Arduino 輸出腳位的電壓，控制七段顯示器特定 LED 的亮滅。當輸出腳位為低電位時，對應的 LED 會導通而點亮；反之，設為高電位時，LED 則不會亮起。

* **結論**

在七顯實驗中我們透過電晶體作為開關、LED控制七顯發光的方式控制七顯顯示我們的桌號。

* **心得**

今天的主題很有吸引力！一開始看到要做「通訊協議」實驗時，我還一頭霧水；直到發現是操作 Arduino 才恍然大悟。沒想到之前網路安全程式設計課學的通訊協定，在這裡也能派上用場。上這堂課前，我一直以為通訊協定只是軟體層面的東西，現在才大開眼界——原來 Arduino 內部還藏了這麼多不為人知的細節。

這四個實驗中，最讓我挫敗的是實驗四。因為好幾週沒用七段顯示器，連 LED 接腳都搞反了。我一開始誤以為電壓設成 HIGH 就能讓 LED 亮，對此深信不疑，甚至後來出現 bug 時也沒懷疑過這點……最後的結論真的讓我學到一課。不過連累夥伴等了超久，真的欠他一個大大的道歉……

* **引用**
* UART

<https://www.eettaiwan.com/20210713ta71-uart-a-hardware-communication-protoco/>

* SPI

<https://wiki.csie.ncku.edu.tw/embedded/SPI>

* I2C

<https://zh.wikipedia.org/zh-tw/I%C2%B2C>