

REPORT

Experiment 1: UART protocol

1. UART frame waveform



$h = 8'b01101000$ ，而根據 Serial Frame Format 可以知道 OSC 畫面中由左到右分別會是 stop bit 接著會是要傳輸的內容從 LSB 到 MSB 最後會是 stop bit。從圖中可以清楚的看到由左到又解讀出來的數據為 00010110，與傳輸的字元 h 的 ascii 相符。



$o = 8'b01101111$ ，從圖中也可以清楚的解讀出數據會是 11110110 與 o 的 ascii 相符。

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

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

3. The interval of a bit is 104u (second) which means the Barud rate is equal to 9600 (bps).

```

13 lines (11 sloc) | 301 Bytes

1  void setup() {
2    // initialize serial communication at 9600 bits per second:
3    Serial.begin(9600);
4  }
5
6  void loop() {
7    // put your main code here, to run repeatedly:
8
9    // send a string, which is the first word of your name
10   // truncate this word to 2 characters
11   Serial.print("ho");
12   delay(500);
13 }

```

因為 Arduino 的 source code 中一開始的 setup 將 serial communication 設定為一秒可以傳輸 9600bit，因此可以推算出每個 bit 的區間為 $\frac{1(\text{sec})}{9600} \approx 104\mu\text{s}$ ，可以跟我使用 OSC 的兩條 cursor 夾出來的區間相比，可以發現實驗結果與理論值相符。

Experiment 2: SPI protocol

1. SPI frame waveform



SPI protocol 在判讀時與前面 exp1 的判讀方法較為不同。Channel 1 黃色的訊號為 clk，而 Channel 2 藍色的訊號為 MOSI，而讀取的方法為每當 clk posedge 時去讀取 Channel 2 的訊號為何。而由左至右分別會從 MSB 到 LSB，與前面的 UART protocol 顛倒。c=8'b01100011, h=8'b01101000，在上圖中可以清楚的看到解碼後與 ch 個別的 ASCII 相符。

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

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

3. The frequency of the SCK is equal to 116.2k Hz

```

3 void setup (void) {
4   Serial.begin(115200); //set baud rate to 115200 for usart
5   digitalWrite(SS, HIGH); // disable Slave Select
6   SPI.begin ();
7   SPI.setClockDivider(SPI_CLOCK_DIV128); //main clock 16 MHz divide the clock by 128, clock is approximately equal to 125 KHz
8 }

```

這邊使用的 clk 應該是 main clk 除以 128 所得到，原本的 clk 的頻率為 16MHz，而經過除頻器之後的頻率變為 125kHz，內部實做細節如下：

除頻器實做：

```

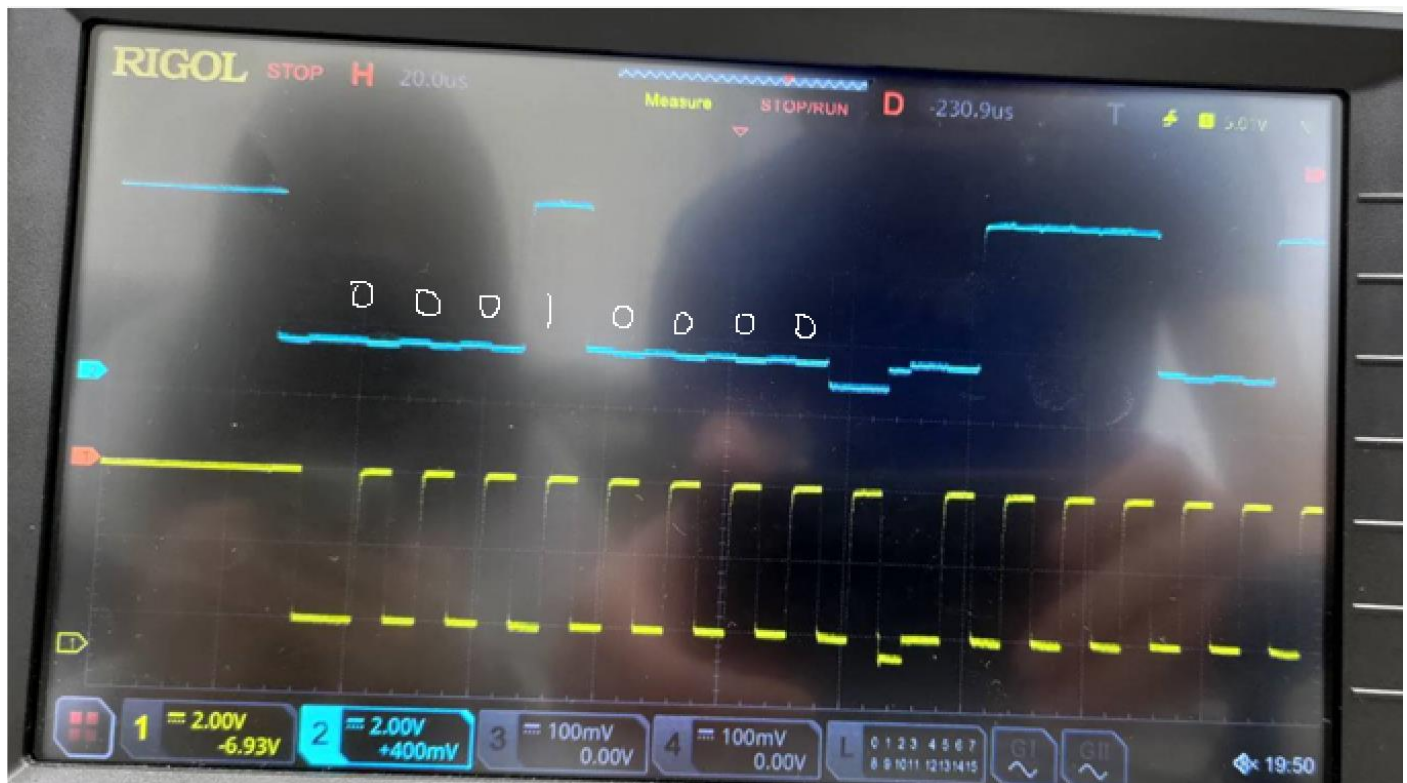
10    //reduce the frequency
11    //original clk run 2500000times get the inverse clk1;
12    //so the clk1 T=2500000*2=5000000 get 2Hz clk1
13    always@(posedge clk or negedge rst)begin
14        if(rst==1'b0)begin
15            count=0;
16            clk1=0;
17        end
18        else if(count==2500000)begin
19            count=0;
20            clk1=~clk1;
21        end
22        else count=count+1;
23    end

```

使用一個計數器，當 main clk 震動一定數量次數之後，再將之後要給其他 module 使用的 clk'反轉，如此一來便達到除頻的效果。

Experiment 3: I2C protocol

1. I2C frame waveform



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

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

3. The frequency of the SCL is equal to 104k Hz

```

3 void setup() {
4   Wire.begin(8);           // join i2c bus with address #8
5   Wire.onReceive(receiveEvent); // register event
6   Serial.begin(9600);      // start serial for output
7 }

```

從 I2C_slave_reciever.ino 的 setup 中可以看到這邊設定了第一個 frame 所要傳送的位址為 8，因此第一個 frame 解出來的位址為 7'b0001000。

Experiment 4: 2-digit 7-segment display

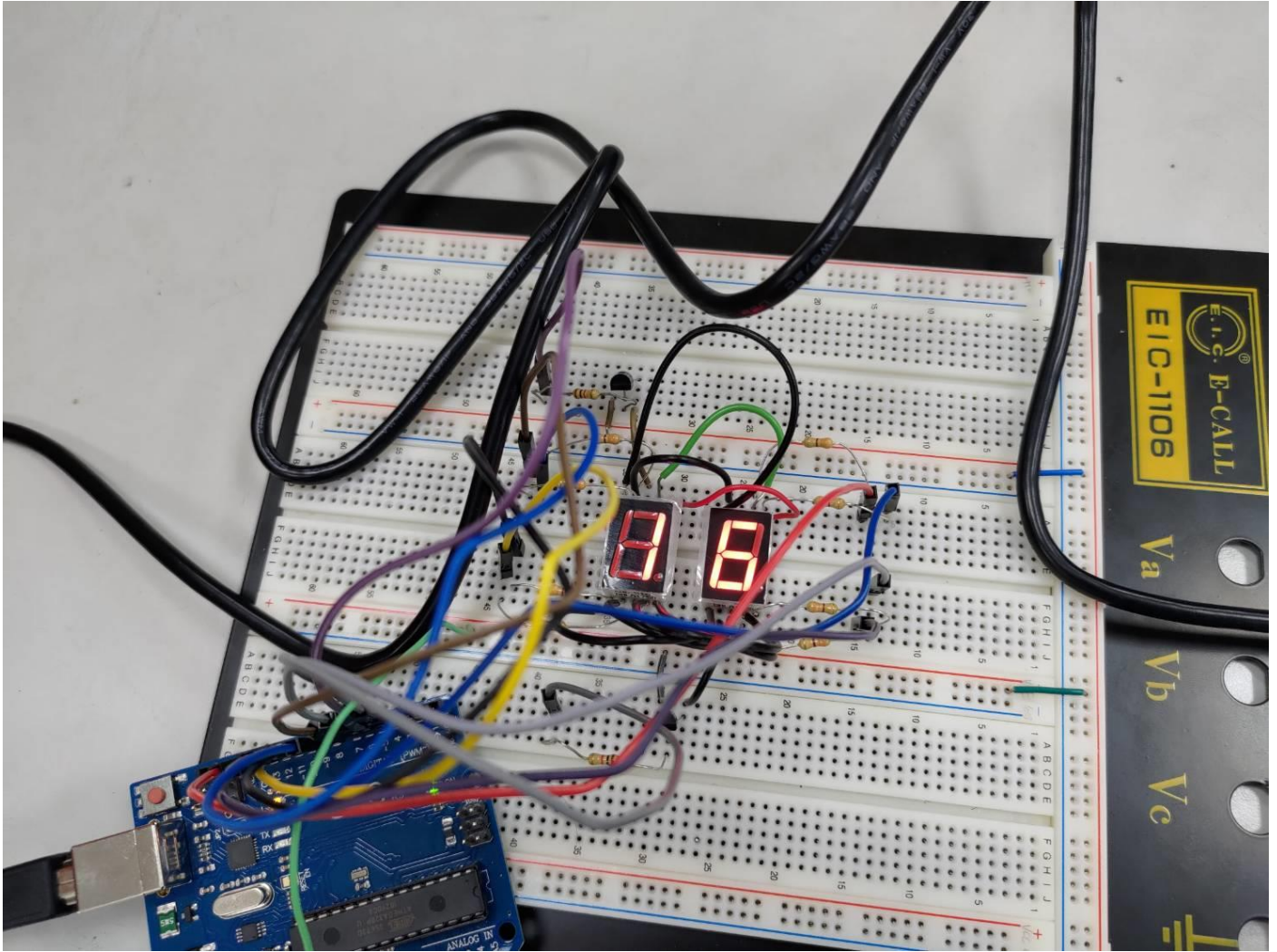
The sketch of your design: (copy from the Arduino IDE window and paste here)

```

1  int pinA = 13, pinB = 12, pinC = 11, pinD = 10, pinE = 9, pinF = 8, pinG = 7, pinP = 6;
2
3  int left = 5, right = 4;
4
5  void setup(){
6    pinMode(pinA, OUTPUT);pinMode(pinB, OUTPUT);pinMode(pinC, OUTPUT);pinMode(pinD, OUTPUT);
7    pinMode(pinE, OUTPUT);pinMode(pinF, OUTPUT);pinMode(pinG, OUTPUT);pinMode(pinP, OUTPUT);
8    pinMode(left, OUTPUT);pinMode(right, OUTPUT);
9  }
10
11 void showSevenSeg(int A, int B, int C, int D, int E, int F, int G, int P){
12   digitalWrite(pinA, A);digitalWrite(pinB, B);digitalWrite(pinC, C);digitalWrite(pinD, D);
13   digitalWrite(pinE, E);digitalWrite(pinF, F);digitalWrite(pinG, G);digitalWrite(pinP, P);
14 }
15
16 void loop(){
17   digitalWrite(right, HIGH);
18   digitalWrite(left, LOW);
19   showSevenSeg(1,0,0,1,1,1,1,1); //1
20   delay(5);
21   digitalWrite(right, LOW);
22   digitalWrite(left, HIGH);
23   showSevenSeg(0,1,0,0,0,0,0,1); //6
24   delay(5);
25 }

```

Take a photo of your display content.



這次的實驗介紹了三種不同的通訊協定，非常有趣。讓我對這些通訊相關的時做有的一些的初步了解。這次的實驗有相較以往比較輕鬆，也沒有遇到甚麼 bug，一次就成功了。希望最後幾次的電子實驗也能如此順利。