

Embedded Systems Laboratory

- Lap5:
- มีความรู้ความเข้าใจในรายละเอียดของ SPI ของ ESP32
 - การโปรแกรมเพื่อควบคุมการทำงานโดยใช้ SPI
 - การโปรแกรมประยุกต์ในการใช้งาน SPI ของ ESP32

อุปกรณ์ Lab5

1. ชุดอุปกรณ์ Embedded System 1 ชุด
2. Laptop หรือ Notebook 1 เครื่อง (Window/Mac/Linux ในการสอนจะใช้ Window เป็น OS)

5.1 ข้อมูลเบื้องต้น SPI ของ ESP32

ให้นักศึกษาร่วมกันหาข้อมูลเพื่อนำมาตอบคำถามข้างล่างดังนี้

คำถาม	คำตอบ
SPI คือ	เป็นวิธีการ (โปรโตคอล) สื่อสารแบบอนุกรมรูปแบบหนึ่งสำหรับการติดต่อสื่อสารกับอุปกรณ์ภายนอกในระยะใกล้
ESP32 มี SPI จำนวนเท่าใด? และผู้ใช้สามารถใช้ได้จำนวนเท่าใด?	4 ใช้ได้ 3 ชุด
ลักษณะการรับส่งข้อมูล SPI เป็นแบบใด? (Half/Full Duplex)	Full Duplex
ความเร็วโดยทั่วไปของ SPI	20 MHz
Mode ของ SPI มีกี่แบบ? แบบใดเป็นที่นิยม พบเจอได้บ่อยครั้งที่สุด?	4 mode นิยม mode 0 และ 3
ในการสื่อสารสามารถมี Slave ได้ทั้งสิ้นกี่ตัว? ปัจจัยใดเป็นตัวกำหนดจำนวนของ Slave	ประมาณ 20 ตัว โดยขึ้นอยู่กับจำนวน GPIO ที่สามารถใช้งานได้

จงตอบคำถามทั่วไปสำหรับ PIN SPI ของ ESP32

PIN	HSPI	VSPI	รายละเอียดของขา SPI
MISO	GPIO12	GPIO19	Master in Slave out
MOSI	GPIO13	GPIO23	Master out Slave in
SCK	GPIO14	GPIO18	Clock
SS	GPIO15	GPIO5	Slave Select

จงตอบคำถามในเรื่อง Mode การทำงานของ SPI

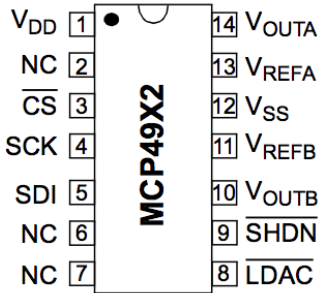
Mode	CPOL logic	CPHA logic	Normal Output (สถานะปกติ 1/0)	Data capture (ขอบขาขึ้น/ลง)
Mode 0	0	0	0	ขอบขาขึ้น
Mode 1	0	1	0	ขอบขาลง
Mode 2	1	0	1	ขอบขาลง
Mode 3	1	1	1	ขอบขาขึ้น

จงอธิบายการทำงาน Function SPI ที่ใช้กับ Lib SPI ใน ArduinoIDE

Function	คำตอบ
SPI.beginTransaction()	เริ่มการใช้งาน SPI
SPI.begin()	เริ่มต้นการทำงานโมดูลสื่อสาร SPI
SPI.transfer()	ส่งข้อมูล
SPI.endTransmission()	หยุดการใช้งานบัส SPI

5.2 IC 49X2

ให้นักศึกษา ค้นหาและหาคำตอบ เพื่อจะนำ ESP32 ติดต่อกับ IC 49X2

 <p>MCP4902: 8-bit dual DAC MCP4912: 10-bit dual DAC MCP4922: 12-bit dual DAC</p>		คำถาม	คำตอบ
		IC เบอร์? (4902, 4912, 4922) ที่ใช้ในการทดลอง	MCP4922
		IC ชนิดนี้ มีหน้าที่? นิยมนำไปใช้ทำอะไร?	แปลงสัญญาณ digital เป็น analog ขนาด 12 bit (DAC12bit)
		ความละเอียดกี่ Bit ในการแปลงข้อมูล	ขนาด 12 bit (DAC12bit)
		มี Output จำนวนกี่ช่อง? มีชื่ออะไรบ้าง?	2 Channel A กับ B
		ความเร็วที่ IC ใช้งาน	20 MHz
		รองรับการใช้งานใน SPI Mode ไตบ้าง?	0,3

จงอธิบายรายละเอียดของแต่ละ PIN

PIN	Symbol	Function
1	VDD	Supply Voltage Input (2.7V to 5.5V)
2	NC	No Connection
3	CS	Chip Select Input
4	SCK	Serial Clock Input
5	SDI	Serial Data Input
6	NC	No Connection
7	NC	No Connection
8	LDAC	Synchronization Input.
9	SHDN	Hardware Shutdown Input
10	VOUTB	DACB Output
11	VREFB	DACB Reference Voltage Input (VSS to VDD)
12	VSS	Ground reference point for all circuitry on the device
13	VREFA	DACA Reference Voltage Input (VSS to VDD)
14	VOUTA	DACA Output

รายละเอียดของ Command Register

REGISTER 5-1: WRITE COMMAND REGISTER

Upper Half:							
W-x	W-x	W-x	W-0	W-x	W-x	W-x	W-x
$\overline{A/B}$	BUF	\overline{GA}	\overline{SHDN}	D11	D10	D9	D8
bit 15				bit 8			

Lower Half:							
W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
D7	D6	D5	D4	D3	D2	D1	D0
bit 7				bit 0			

- bit 15 $\overline{A/B}$: DAC_A or DAC_B Select bit
 1 = Write to DAC_B
 0 = Write to DAC_A
- bit 14 **BUF**: V_{REF} Input Buffer Control bit
 1 = Buffered
 0 = Unbuffered
- bit 13 \overline{GA} : Output Gain Select bit
 1 = 1x (V_{OUT} = V_{REF} * D/4096)
 0 = 2x (V_{OUT} = 2 * V_{REF} * D/4096)
- bit 12 **SHDN**: Output Power Down Control bit
 1 = Output Power Down Control bit
 0 = Output buffer disabled, Output is high impedance
- bit 11-0 **D11:D0**: DAC Data bits
 12 bit number "D" which sets the output value. Contains a value between 0 and 4095.

ต้องการสั่งงานดังนี้

- เลือกช่อง DACA
- Unbuffer
- Gain 1x
- Output Control bit 11-0
- Vref 3.3V
- ต้องการให้ VoutA มีแรงดันออก 1.8V

แสดงวิธีการคำนวณการหา Data 12Bit ที่ทำให้มีแรงดันออกตามที่สั่งงาน

Vref = 3.3 V
 3.3 V => 4095
 1.8 V => 4095 * 1.8 / 3.3 = 2234
 8 B A
 1000 1011 1010

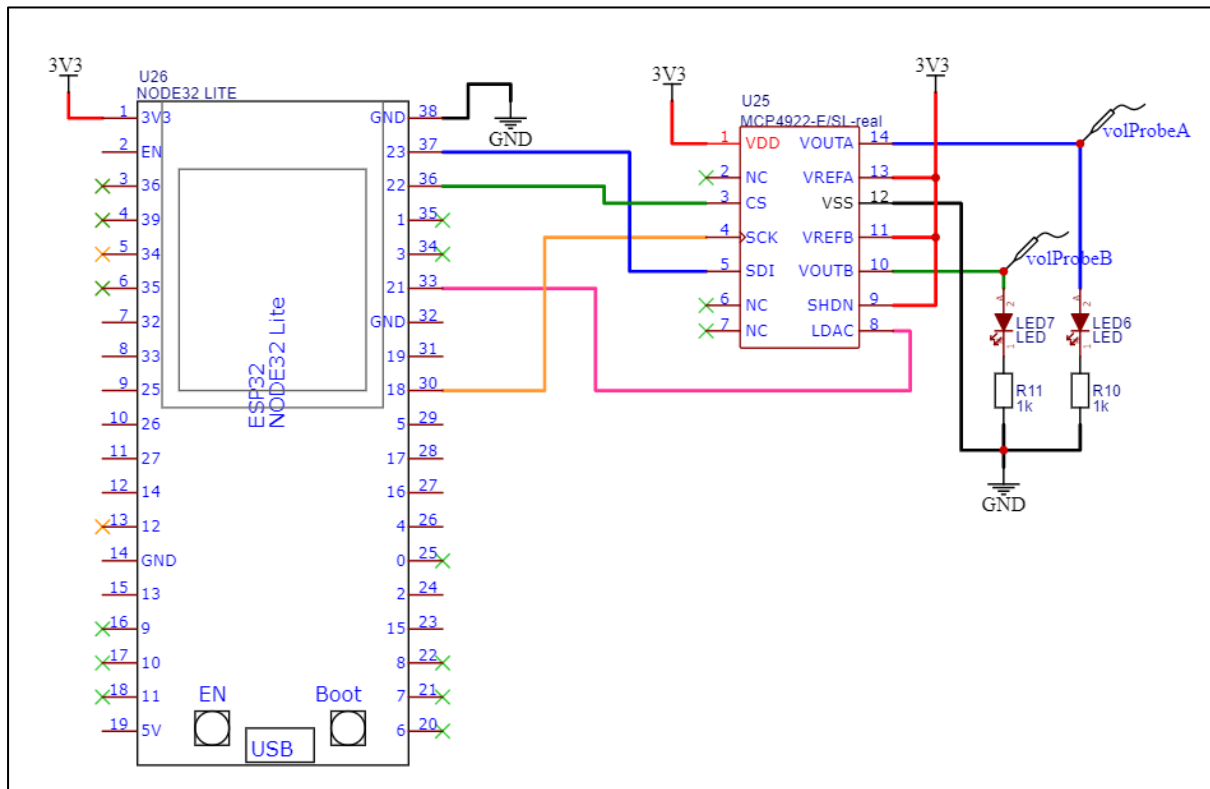
เขียน Command Register ที่สำหรับใช้สั่งงานด้านบน

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	0	0	0	1	0	1	1	1	0	1	0

5.3 Basic Interface MCP4922 by SPI

ให้นักเขียนโปรแกรมควบคุมให้ MCP4922 ตามวงจรทดสอบด้านล่าง

แผนผังวงจรไฟฟ้าสำหรับการทดลอง



การทำงานของ Firmware Lab 5.3 ดังนี้

1. เมื่อเริ่มทำงาน หลอด LED6 และ LED7 จะกะพริบติดดับสลับกันทุกๆ 1000ms
2. การติดของ LED7 (VOUTB) ให้คำนวณแรงดันออกอยู่ที่ 3.3V ส่วนการดับให้แรงดันอยู่ที่ 0.0V
3. การติดของ LED6 (VOUTA) ให้คำนวณแรงดันออกอยู่ที่ 2.0V ส่วนการดับให้แรงดันอยู่ที่ 0.0V
4. การทำงานของ Firmware ให้นำเอา Timer Interrupt มาใช้งาน

ตัวอย่าง Function การควบคุมการทำงานของ IC 4922

```
void writeDac1V()
{
    //1. คำสั่ง Start SPI
    //2. สั่งให้ขา CS เป็น Low
    //3. คำนวณหา MCP4922 Write Command Register ex int16_t dataDAC = 0x34d9;
    //4. นำเอาข้อมูลในข้อ 3.) ส่งข้อมูลออกโดยใช้ SPI Protocol
    //5. หน่วยเวลาเล็กน้อย ประมาณ 5ms
    //6. สั่งให้ขา LDAC เป็น Low
    //7. หน่วยเวลาเล็กน้อย ประมาณ 5ms
    //8. สั่งให้ขา LDAC เป็น High
    //9. สั่งให้ขา CS เป็น High
    //10. คำสั่ง Stop SPI
}
```

เขียนโปรแกรมลงในกล่องคำตอบด้านล่าง และถ่ายวิดีโอผลลัพธ์ของโจทย์นี้ Upload ไฟล์ตามหมู่เรียน

ในกรณีตัวหน้ากระดาษไม่พอให้เพิ่มหน้าแทรกในไฟล์แทน

```
#include <SPI.h>
SPISettings MCP4922(20000000,MSBFIRST,SPI_MODE0);
#define SS 22
#define LDAC 21
#define LED 2
bool stateToggle = true;
hw_timer_t *timer0 = NULL;
void outA2V2()
{ SPI.beginTransaction(MCP4922);
  digitalWrite(SS,LOW);
  SPI.transfer16(0x39B2);
  digitalWrite(SS,HIGH);
  delay(1);
  digitalWrite(LDAC,LOW);
  delay(1);
  digitalWrite(LDAC,HIGH);
  SPI.endTransaction();
}
void outA0V0()
{ SPI.beginTransaction(MCP4922);
  digitalWrite(SS,LOW);
  SPI.transfer16(0x3000);
  digitalWrite(SS,HIGH);
  delay(1);
  digitalWrite(LDAC,LOW);
  delay(1);
  digitalWrite(LDAC,HIGH);
  SPI.endTransaction();
}
void outB3V3()
{ SPI.beginTransaction(MCP4922);
  digitalWrite(SS,LOW);
  SPI.transfer16(0xBFFF);
  digitalWrite(SS,HIGH);
  delay(1);
  digitalWrite(LDAC,LOW);
  delay(1);
  digitalWrite(LDAC,HIGH);
  SPI.endTransaction();
}

void outB0V0()
{ SPI.beginTransaction(MCP4922);
  digitalWrite(SS,LOW);
  SPI.transfer16(0xB000);
  digitalWrite(SS,HIGH);
  delay(1);
  digitalWrite(LDAC,LOW);
  delay(1);
  digitalWrite(LDAC,HIGH);
  SPI.endTransaction();
}

void IRAM_ATTR timer_isr()
{ stateToggle=!stateToggle;
  digitalWrite(LED,stateToggle);
  if(stateToggle==true)
  { outB3V3();
    20outA0V0();
  }else
  { outB0V0();
    outA2V2();
  }
}

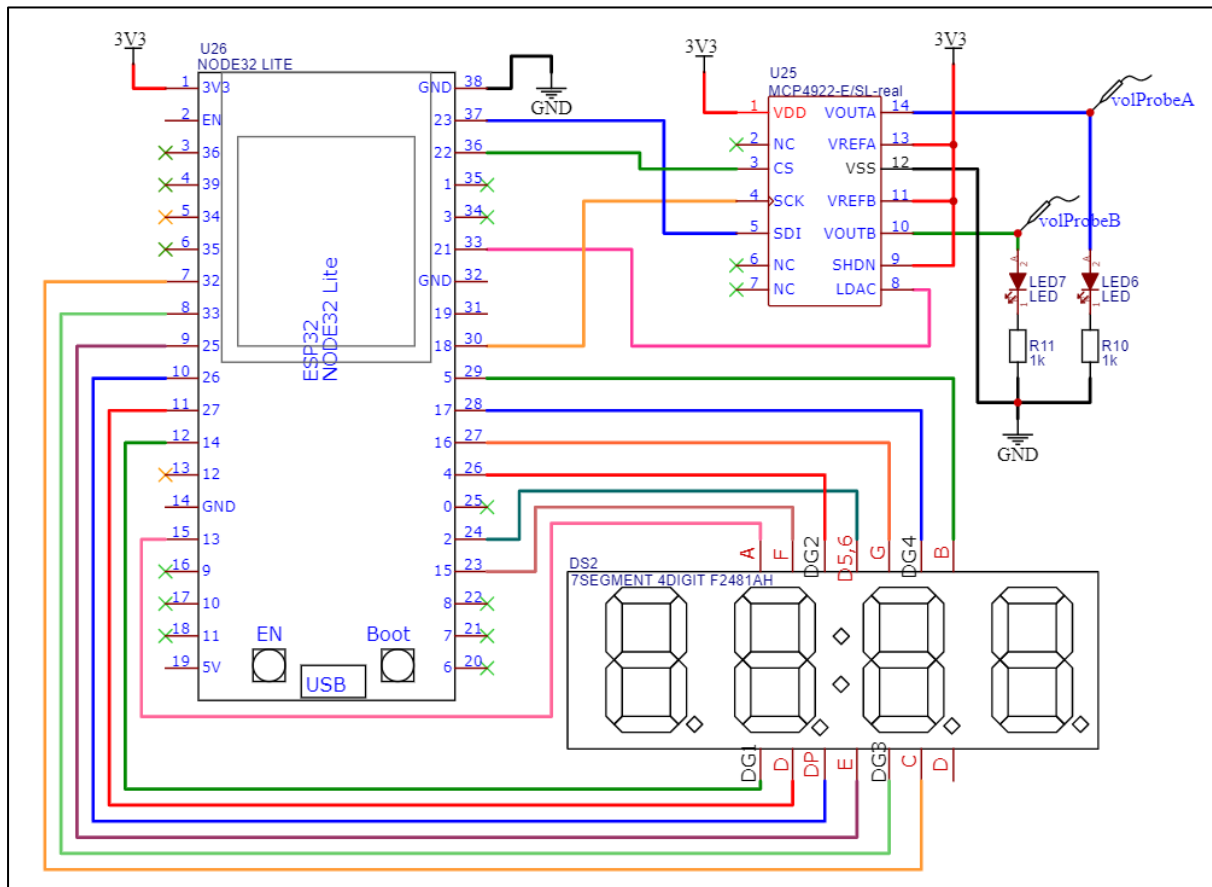
void setup() {
  pinMode(SS,OUTPUT);
  pinMode(LDAC,OUTPUT);
  pinMode(LED,OUTPUT);
  SPI.begin();
  digitalWrite(LDAC,HIGH);
  timer0 = timerBegin(0,80,true);
  timerAlarmWrite(timer0,1000000,true);
  timerAttachInterrupt(timer0,&timer_isr,true);
  timerAlarmEnable(timer0);
}

void loop() {
}
```

5.4 Assignment SPI

ให้นักเรียนเขียนโปรแกรมควบคุมให้ MCP4922 ตามวงจรทดสอบด้านล่าง

แผนผังวงจรไฟฟ้าสำหรับการทดลอง



การทำงานของ Firmware Lab 5.4 ดังนี้

1. เมื่อเริ่มทำงาน หลอด LED6 และ LED7 จะกระพริบติดดับสลับกันทุกๆ 1000ms
2. การติดของ LED7 (VOUTB) ให้คำนวณแรงดันออกอยู่ที่ 3.3V ส่วนการดับให้แรงดันอยู่ที่ 0.0V
3. การติดของ LED6 (VOUTA) ให้คำนวณแรงดันออกอยู่ที่ 2.0V ส่วนการดับให้แรงดันอยู่ที่ 0.0V
4. แรงดันจะถูกนำไปแสดงบน 7Segment 4Digit โดยแบ่งเป็นซ้ายและขวา
 - ซ้าย แสดงค่าแรงดันของ VOUTA
 - ขวา แสดงค่าแรงดันของ VOUTB
5. สามารถปรับแรงดันของ VOUTA และ VOUTB โดยผ่าน Serial Monitor กำหนดดังนี้
 - Input String จำนวน 3 ตัวประกอบด้วย ABC โดย Aคือช่องVOUT(A/B) BCคือแรงดันที่ต้องการ
 - เช่น A10 หมายถึง VOUTA แรงดัน 1.0V เมื่อไฟติด, B33 หมายถึง VOUTB แรงดัน 3.3V
6. แรงดันที่ออกกำหนดได้ 4ช่วงคือ (2.0V, 2.2, 3.0V, 3.3V) หรือสามารถกำหนดได้ทุกช่วงแรงดัน แต่ขั้นต่ำคือ4ช่วง
7. การทำงานของ Firmware ให้นำเอา Timer Interrupt มาใช้งาน ใน Function loop สามารถเขียน Code ได้

เขียนโปรแกรมลงในกล่องคำตอบด้านล่าง และถ่ายวิดีโอผลลัพธ์ของโจทย์นี้ Upload ไฟล์ตามหมู่เรียน
ในกรณีตัวหน้ากระดาษไม่พอให้เพิ่มหน้าแทรกในไฟล์แทน

