

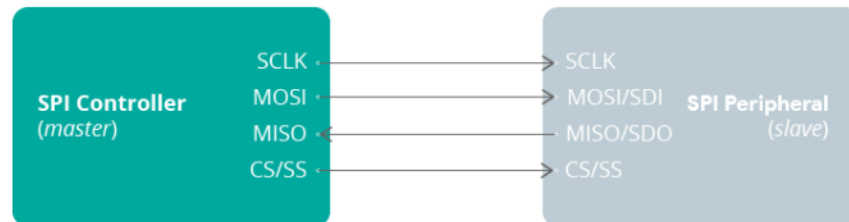
## Pertemuan 06

### Serial Peripheral Interface (SPI)

#### Dasar Teori SPI

SPI adalah singkatan dari Serial Peripheral Interface, dan merupakan protokol data serial sinkron yang digunakan oleh mikrokontroler untuk berkomunikasi dengan satu atau lebih periferal. Misalnya, papan ESP32 Anda berkomunikasi dengan sensor yang mendukung SPI atau dengan mikrokontroler lain.

Dalam komunikasi SPI, selalu ada pengontrol (disebut juga master) yang mengontrol perangkat periferal (disebut juga slave). Data dapat dikirim dan diterima secara bersamaan. Artinya master dapat mengirimkan data ke slave, dan slave dapat mengirimkan data ke master pada saat yang bersamaan.



Anda hanya dapat memiliki satu master, yang akan menjadi mikrokontroler (ESP32), namun Anda dapat memiliki banyak slave. Slave dapat berupa sensor, layar, kartu microSD, dll., atau mikrokontroler lainnya. Ini berarti Anda dapat menghubungkan ESP32 ke beberapa sensor, tetapi sensor yang sama tidak dapat dihubungkan ke beberapa papan ESP32 secara bersamaan.

#### Antarmuka SPI

Untuk komunikasi SPI Anda memerlukan empat jalur:

- MISO : Master In, Slave Out
- MOSI : Master Out, Slave In
- SCK : Serial Clock
- CS/SS: Chip Select (digunakan untuk memilih perangkat ketika beberapa perangkat digunakan pada bus SPI yang sama)

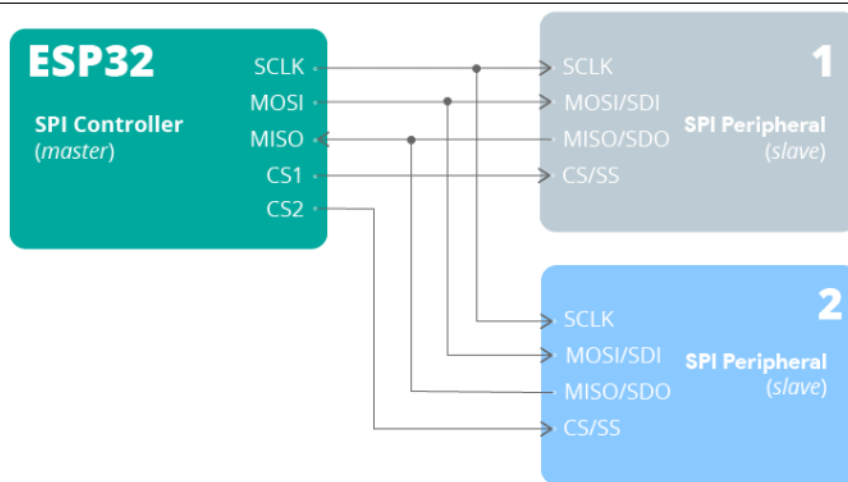
Empat jalur SPI tersebut mungkin memiliki nama-nama yang berbeda pada perangkat yang berbeda. Tabel berikut menunjukkan Pin SPI pada ESP32

LABEL	GPIO
MISO/SDO	19 / 12
MOSI/SDA	23 / 13
SCK/SCL	18 / 14
CS/SS/SDA/CSB	5 / 15

Seperti yang telah kita lihat sebelumnya, Anda dapat menggunakan dua bus SPI berbeda pada ESP32 dan setiap bus dapat menghubungkan hingga tiga periferal berbeda. Artinya kita dapat menghubungkan hingga enam perangkat SPI ke ESP32. Jika Anda perlu menggunakan lebih banyak, Anda dapat menggunakan multiplexer SPI

#### Multi Perangkat SPI (bus sama, berbeda pada pin CS)

Untuk menghubungkan beberapa perangkat SPI, Anda dapat menggunakan bus SPI yang sama selama setiap periferal menggunakan pin CS yang berbeda.



Untuk memilih peripheral yang ingin Anda ajak berkomunikasi, Anda harus menyetel pin CS-nya ke LOW. Misalnya, bayangkan Anda memiliki peripheral 1 dan peripheral 2. Untuk membaca dari peripheral 1, pastikan pin CS-nya disetel ke LOW. Kemudian, apabila anda ingin membaca dari peripheral 2. Anda harus menonaktifkan pin CS peripheral 1 dengan menyetelnya ke HIGH, dan mengaktifkan pin CS peripheral 2 dengan menyetelnya ke LOW

### Praktik 1 (BMP280, Sensor Suhu dan Tekanan)

Download terlebih dulu modul untuk membaca sensor BMP280 pada link yang berada di halaman terakhir :

Kemudian download file-file berikut ini dan upload ke ESP32

1. bmp280\_spi.py
2. bmp280.py
3. bmp280\_configuration.py

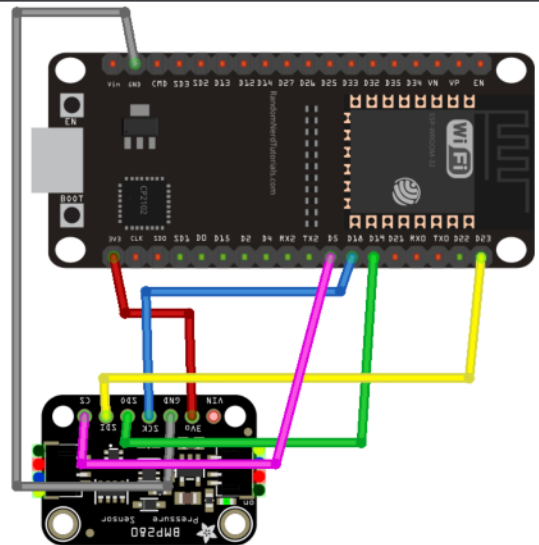
Sedangkan untuk file **main.py** kode programnya sebagai berikut :

```
from machine import Pin, SPI
from utime import sleep

from bmp280_spi import BMP280SPI

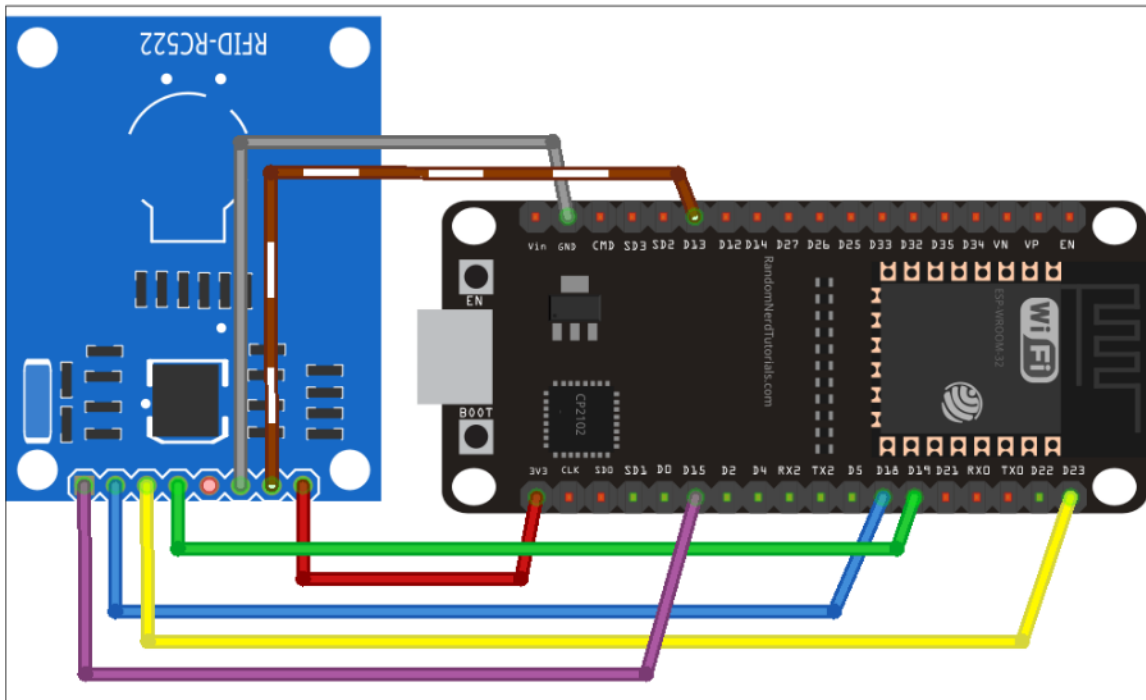
spi_cl = Pin(18)
spi_mosi = Pin(23)
spi_miso = Pin(19)
spi_cs = Pin(5, Pin.OUT, value=1)
spi1 = SPI(1, sck=spi_cl, mosi=spi_mosi,
miso=spi_miso)
bmp280_spi = BMP280SPI(spi1, spi_cs)

while True:
    data = bmp280_spi.measurements
    print("Suhu: {} °C, tekanan: {}
hPa".format(data['t'],data['p']))
    sleep(5)
```



fritzing

## Praktik 2 (MFRC522, RFID Reader dan Writer) Mode Read



Download terlebih dulu modul untuk membaca sensor MFRC522 pada link yang berada di halaman terakhir :

Kemudian download file-file berikut ini dan upload ke ESP32

1. mfr522.py

Sedangkan untuk file **main.py** kode programnya sebagai berikut :

```
import mfr522

from machine import Pin, SPI
from utime import sleep

rdr = mfr522.MFRC522(18, 23, 19, 16, 15)
print("Tempatkan kartu pada reader untuk membaca blok 0x08")

def read():
    try:
        (stat, tag_type) = rdr.request(rdr.REQIDL)
        if stat == rdr.OK:
            (stat, raw_uid) = rdr.anticoll()
            if stat == rdr.OK:
                print("\n\nKartu terdeteksi")
                print(" - tipe tag: 0x%02x" % tag_type)
                print(" - uid      : 0x%02x%02x%02x%02x" % \
                    (raw_uid[0], raw_uid[1], raw_uid[2], raw_uid[3]))
                print("")
                if rdr.select_tag(raw_uid) == rdr.OK:
                    key = [0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF]

                    #cek autentikasi pada blok ke-8
                    if rdr.auth(rdr.AUTHENT1A, 8, key, raw_uid) == rdr.OK:
                        #membaca pada blok 8
```

```

        text=[chr(x) for x in rdr.read(8)]
        print("Data pada blok 0x08 : %s" % "".join(text))
        rdr.stop_crypto1()
    else:
        print("Otentikasi gagal")
else:
    print("Tidak dapat mengenali tag")

except KeyboardInterrupt:
    print("Bye")

def write(data):
    try:
        (stat, tag_type) = rdr.request(rdr.REQIDL)
        if stat == rdr.OK:
            (stat, raw_uid) = rdr.anticoll()
            if stat == rdr.OK:
                print("Kartu terdeteksi")
                print(" - tag type: 0x%02x" % tag_type)
                print(" - uid : 0x%02x%02x%02x%02x" % \
                    (raw_uid[0], raw_uid[1], raw_uid[2], raw_uid[3]))
                print("")
                if rdr.select_tag(raw_uid) == rdr.OK:
                    key = [0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF]

#angka 8 adalah nomor blok. ada 64 blok, yang bisa dipakai blok 1-63
#dengan catatan : blok ke-4 tiap sektor (3,7,11,15,19, ..., 63) tidak bisa dipakai
#karena untuk autentikasi
                if rdr.auth(rdr.AUTHENT1A, 8, key, raw_uid) == rdr.OK:
                    stat = rdr.write(8, bytes(data,"ascii"))
                    rdr.stop_crypto1()
                    if stat == rdr.OK:
                        print("Berhasil menulis data ke kartu")
                    else:
                        print("Gagal menulis data ke kartu")
                else:
                    print("Otentikasi gagal")
            else:
                print("Tidak dapat mengenali tag")
        sleep(1)
    except KeyboardInterrupt:
        print("Bye")

while True:
    read()
    #write("benni agung ngrh")
    sleep(2)

```

### Praktik 3 (MFRC522, RFID Reader dan Writer) Mode Write

Ganti pada bagian while True menjadi seperti berikut :

```
while True:
    #read()
    write("benni agung ngrh")
    sleep(2)
```

### Praktik 4 (Tugas)

Buatlah aplikasi IoT untuk pengaman pintu menggunakan RFID reader, Servo Motor, dan LCD dengan skenario sebagai berikut :

1. Tuliskan terlebih dulu identitas pada kartu RFID anda, semisal, nomor NIM anda
2. Secara default servo motor berada pada posisi 90°
3. Buat aplikasi **smart door lock** dengan ketentuan :
  - Jika RFID reader membaca bahwa NIM yang terdapat pada kartu RFID sesuai dengan yang tersimpan di aplikasi, maka pintu akan membuka (disimulasikan dengan cara servo motor bergerak ke sudut 0°) dan pada LCD menampilkan NIM yang masuk. Pintu akan menutup secara otomatis setelah 5 detik.
  - Apabila NIM tidak sesuai maka pintu akan menutup (disimulasikan dengan cara servo motor bergerak ke sudut 90° atau tetap menutup) dan pada LCD menampilkan tulisan : "DILARANG MASUK"

Link download modul/library :

[https://drive.google.com/drive/folders/16Et72bcJpY4nhdBNOPH-pEcU6jDbreE1?usp=drive\\_link](https://drive.google.com/drive/folders/16Et72bcJpY4nhdBNOPH-pEcU6jDbreE1?usp=drive_link)