

Pertemuan 10

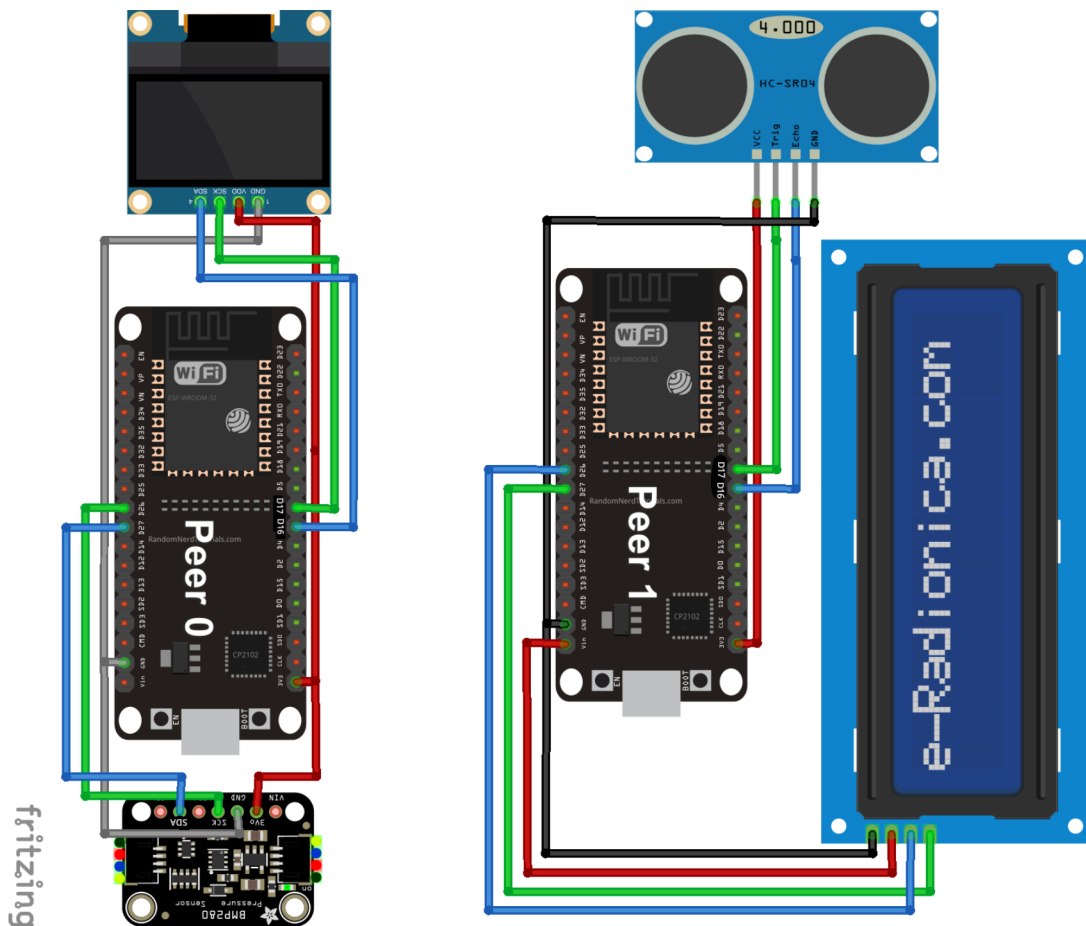
Protokol Komunikasi Nirkabel ESP-NOW

Dasar Teori ESP Now
<p>ESP-NOW adalah protokol komunikasi yang dikembangkan oleh Espressif Systems. Espressif Systems adalah produsen perangkat mikrokontroler keluarga ESP32 dan ESP8266. ESP-NOW memungkinkan kita untuk mengkonfigurasi ESP32/ESP8266 agar dapat berkomunikasi antara satu dengan lainnya (<i>peer-to-peer</i>) melalui layer 2 OSI (MAC Address) dan menggunakan jaringan nirkabel/wireless tanpa menggunakan teknologi WiFi. ESP-NOW hanya dapat digunakan pada perangkat dengan mikrokontroler ESP32/ESP8266</p> <p>Mode jaringan yang didukung oleh protokol ESP-NOW antara lain :</p> <ol style="list-style-type: none">1. One-to-Many : dimana satu perangkat dapat mengirim informasi ke banyak perangkat2. Many-to-One : dimana banyak perangkat dapat mengirim informasi ke satu perangkat3. Many-to-Many : dimana banyak perangkat dapat mengirim informasi ke banyak perangkat <p>Komunikasi yang dilakukan oleh protokol ESP NOW dapat bersifat half-duplex (satu perangkat menjadi transmitter dan lainnya menjadi receiver) atau full-duplex (semua perangkat yang terhubung dapat menjadi transmitter dan receiver sekaligus)</p>

Persiapan Praktik
<p>Untuk dapat menggunakan protokol komunikasi ESP NOW maka dibutuhkan firmware MicroPython versi 1.21.0, untuk itu lakukan instalasi MicroPython versi tersebut ke ESP32 atau ESP8266. Untuk tata cara instalasi dapat dilihat pada modul/materi Pertemuan 4</p>

Praktik 1 – Bertukar data antara 2 perangkat

1. Untuk praktik 1 ini, dibutuhkan dua perangkat ESP32 yang akan dihubungkan secara wireless menggunakan protokol ESP-NOW. Kedua buah perangkat akan saling bertukar data dan menampilkan data yang akan diperoleh ke perangkat display masing-masing perangkat. Untuk kode programnya sebagai berikut :



Kode program perangkat **Peer 0** (tambahkan library yang dibutuhkan)

```
import network
import espnow
from utime import sleep
from machine import Pin, SoftI2C
from ssd1306 import SSD1306_I2C
from bmp280_i2c import BMP280I2C

#INI ADALAH DEVICE PEER 0

#pilih WLAN STANDARD_INTERFACE
wlan = network.WLAN(network.STA_IF)

def activateWlan():
    print("Mengaktifkan WLAN ...")
    global wlan
    wlan.active(True)
    if wlan.active():
        return True
    else:
        return False

if activateWlan():
    #mac address peer1
    peer1 = b'\xa8B\xe3\xca\xa48' #Ganti dengan MAC Addr peer1

    #mengaktifkan protokol ESP Now
    esp = espnow.ESPNow()
    esp.active(True)
    esp.add_peer(peer1) #menambahkan peer1 ke dalam daftar peer yg diajak komunikasi
```

```

#mendapatkan MAC address device ini
macAddr = wlan.config("mac")
#print MAC in byte format
print(macAddr)
#print MAC in string format
print("MAC ADDR : ", ":".join(["{:02X}".format(byte) for byte in macAddr]))

#mengaktifkan oled
i2c_oled = SoftI2C(scl=Pin(17), sda=Pin(16))
oled_width = 128
oled_height = 32
oled = SSD1306_I2C(oled_width, oled_height, i2c_oled)
oled.fill(False)
oled.text("I2C Comm",0,0)
oled.show()

#mengaktifkan BMP280
i2c_bmp280 = SoftI2C(scl=Pin(26), sda=Pin(27))
bmp280 = BMP280I2C(i2c_bmp280,0x76)

flip = True

while True:
    flip = not flip
    if flip:
        rto="---"
    else:
        rto="|||"

    data = bmp280.measurements

    oled.fill(False)
    oled.text("T = {:.2f} C".format(data['t']),0,0)
    oled.text("P = {:.2f} hPa".format(data['p']),0,8)

    #jika menerima message
    mac, msg = esp.recv(100)
    if msg:
        oled.text("D = {:.2f} cm".format(float(msg)),0,16)
    else:
        oled.text("D = {} cm".format(rto),0,16)
    oled.show()

    #mengirim pesan ke peer1
    esp.send(peer1,str(data['t'])+";"+str(data['p']),False)
    sleep(0.9)
else:
    prin("WiFi tidak aktif, ESP Now tidak aktif")
    activateWlan()

```

Kode program perangkat **Peer 1** (tambahkan library yang dibutuhkan)

```

import network
import espnow
from mp_i2c_lcd1602 import LCD1602
from machine import Pin, SoftI2C
from utime import sleep_ms, sleep_us, ticks_us, ticks_diff

#INI ADALAH DEVICE PEER 1
#pilih WLAN STANDARD_INTERFACE
wlan = network.WLAN(network.STA_IF)

```

```

#mengaktifkan LCD
i2c = SoftI2C(scl=Pin(27), sda=Pin(26))
lcd = LCD1602(i2c)

#inisialisasi ultrasonik
trig = Pin(17, Pin.OUT)
echo = Pin(16, Pin.IN)

def activateWlan():
    print("Mengaktifkan WLAN ...")
    global wlan
    wlan.active(True)
    if wlan.active():
        return True
    else:
        return False

if activateWlan():
    #MAC Address peer0
    peer0 = b'\xa8B\xe3L-$' #Ganti dengan MAC Address peer0

    #mengaktifkan protokol esp dan menambah peer
    esp = espnow.ESPNow()
    esp.active(True)
    esp.add_peer(peer0) #menambahkan peer0 kedalam daftar peer yg diajak komunikasi

    #menampilkan MAC Address device ini
    macAddr = wlan.config("mac") #get MAC
    print(macAddr) #print MAC in byte format
    print("MAC ADDR : ", ":".join("{:02X}".format(byte) for byte in macAddr))

    flip = True

    while True:
        flip = not flip
        if flip:
            rto = "---"
        else:
            rto = "|||"
        lcd.clear()

        #menerima pesan dari peer1
        mac, msg = esp.recv(100)
        if msg:
            tokens = (msg.decode('utf-8')).split(";")
            lcd.puts('T = {:.2f} C'.format(float(tokens[0])), 0, 0)
            lcd.puts('P = {:.2f} hPa'.format(float(tokens[1])), 0, 1)
        else:
            lcd.puts('T = {} C'.format(rto), 0, 0)
            lcd.puts('P = {} hPa'.format(rto), 0, 1)

        #mendeteksi jarak
        sleep_us(5)
        trig.value(1)
        sleep_us(10)
        trig.value(0)

        while echo.value() == False:
            pass
        timerStart = ticks_us()

```

```
while echo.value() == True:
    pass
timerEnd = ticks_us()

duration = ticks_diff(timerEnd,timerStart)
dist = (str(duration*0.017))[:5]

#mengirim data jarak
esp.send(peer0,dist,False)
sleep_ms(900)
else:
    prin("WiFi not active, ESP Now not active")
    activateWlan()
```

Praktik 2 – Bertukar data antara 3 perangkat

1. Untuk praktik 2 ini, tambahkanlah perangkat Peer 2 seperti gambar dibawah ini. Tambahkan kode program agar perangkat Peer 2 dapat mengirimkan data hasil pembacaan sensor cahaya (LDR) ke perangkat Peer 0. Kemudian pada Peer 0 tambahkan kode program untuk menampilkan nilai intensitas cahaya yang didapat ke OLED. Peer 2 hanya mengirim data, tidak menerima data dari manapun

