

D4 TEKNIK KOMPUTER A

Praktikum 10

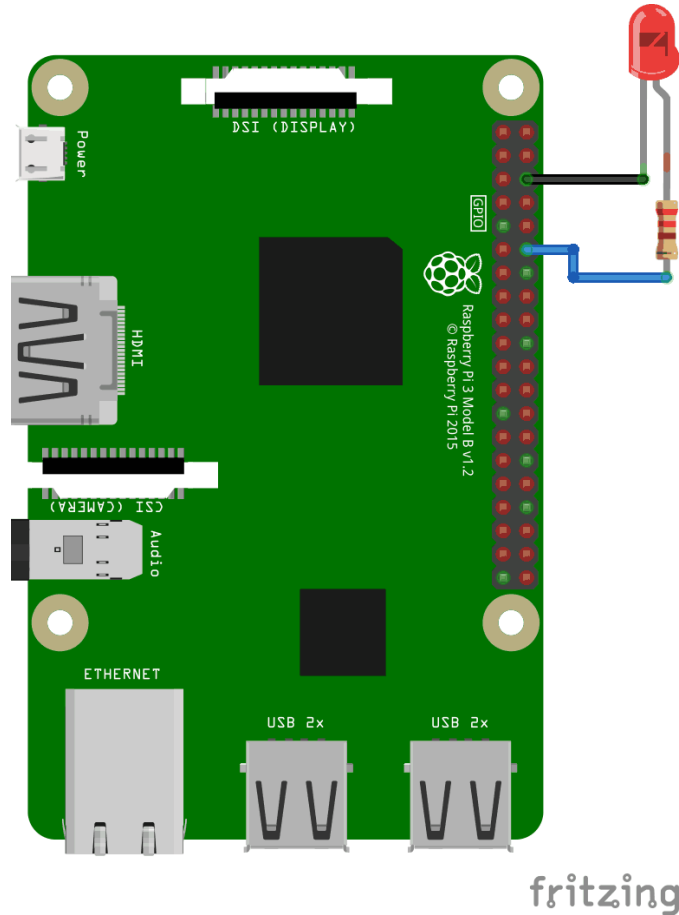
PWM & UART



Nama	:	Febrian Dwi Firmansyah
Kelas	:	2 D4 Teknik Komputer A
NRP	:	3222600028
Dosen	:	Mochamad Mobed Bachtiar
Mata Kuliah	:	Prak. Alat pengembangan Perangkat Lunak
Tanggal	:	12 MEI 2024

I. Raspberry Pi PWM Generation using Python and C

1. Rangkailah led dengan resistor ke dalam board raspi



2. Masukkan kode program ke dalam program

```
import RPi.GPIO as GPIO
from time import sleep

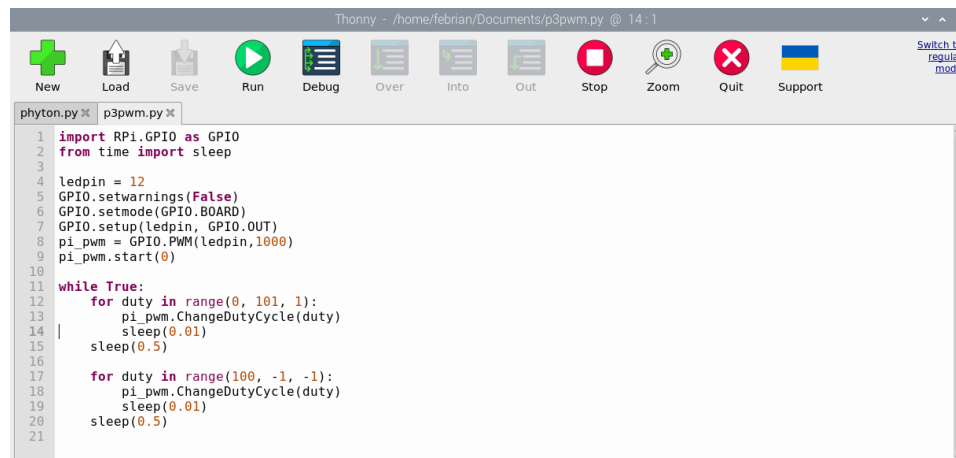
ledpin = 12
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(ledpin, GPIO.OUT)
pi_pwm = GPIO.PWM(ledpin,1000)
pi_pwm.start(0)

while True:
    for duty in range(0, 101, 1):
        pi_pwm.ChangeDutyCycle(duty)
        sleep(0.01)
    sleep(0.5)

    for duty in range(100, -1, -1):
```

```
pi_pwm.ChangeDutyCycle(duty)
sleep(0.01)
sleep(0.5)
```

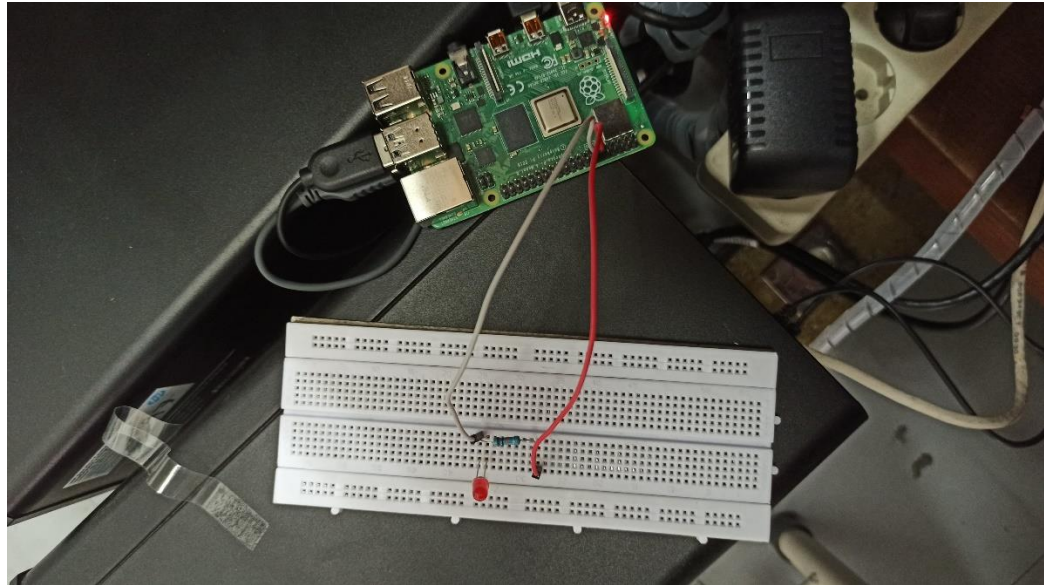
Generate PWM menggunakan python yang dimana menggunakan Thonny untuk compiler python



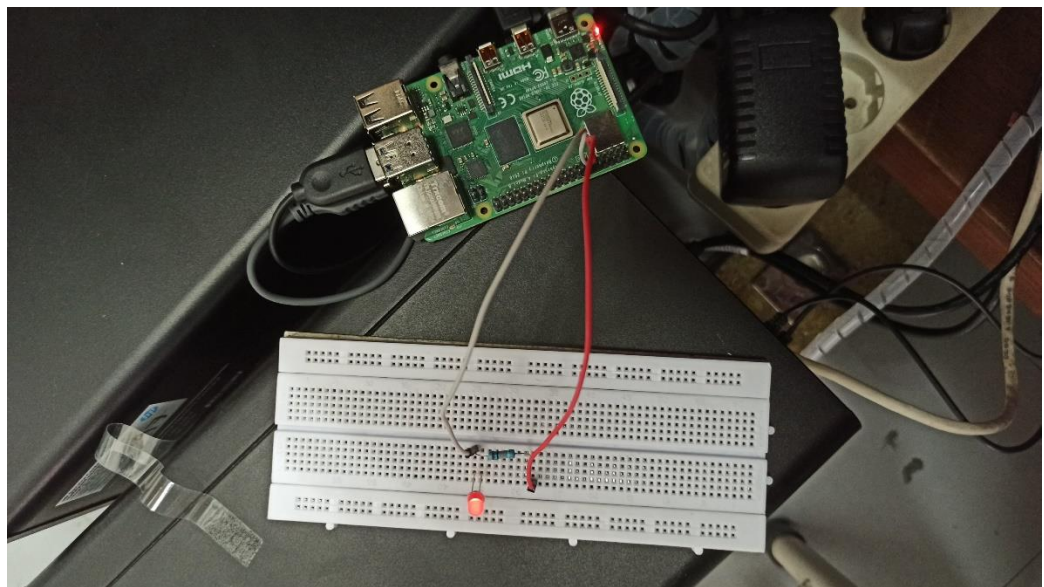
The screenshot shows the Thonny IDE interface. The title bar indicates the file path is `/home/febrian/Documents/p3pwm.py` at line 14. The toolbar includes icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, Quit, and Support. The editor window displays the following Python code:

```
1 import RPi.GPIO as GPIO
2 from time import sleep
3
4 ledpin = 12
5 GPIO.setwarnings(False)
6 GPIO.setmode(GPIO.BOARD)
7 GPIO.setup(ledpin, GPIO.OUT)
8 pi_pwm = GPIO.PWM(ledpin,1000)
9 pi_pwm.start(0)
10
11 while True:
12     for duty in range(0, 101, 1):
13         pi_pwm.ChangeDutyCycle(duty)
14         sleep(0.01)
15     sleep(0.5)
16
17     for duty in range(100, -1, -1):
18         pi_pwm.ChangeDutyCycle(duty)
19         sleep(0.01)
20     sleep(0.5)
21
```

Kode tersebut adalah penggunaan modul `RPi.GPIO` yang dimana memanfaatkan penomoran pin fisik (BOARD), fungsi `sleep` dari modul `time` untuk menunda eksekusi kode. Inisialisasi objek PWM menggunakan dua parameter yang dimana parameter pertama digunakan untuk nomor pin LED sedangkan parameter kedua digunakan untuk frekuensi PWM yang digunakan pada kode menggunakan 1000Hz. Objek PWM dimulai dengan tingkat kecerahan 0 menggunakan `start(0)`. Pada loop LED secara bertahap diatur dari 0 hingga 100 dengan kenaikan 1 setiap iterasi. Setelah kecerahan maksimum program menunda selama 0,5 detik sebelum memulai penurunan kecerahan. Setelah penundaan, kecerahan LED kemudian dikurangi dari 100 ke 0 dengan langkah 1 setiap iterasi.



Led dengan tingkat kecerahaan 0



Led dengan tingkat kecerahaan 100

3. Generate PWM menggunakan bahasa c

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>

const int LED = 1;

int main(void){
    int intensity;
```

```

if(wiringPiSetup () == -1)
    exit(1);

pinMode (LED, PWM_OUTPUT);

while (1){
    for (intensity = 0; intensity <1024; ++ intensity)
    {
        pwmWrite (LED, intensity);
        delay(1);
    }
    for (intensity = 1023; intensity >= 0; --intensity)
    {
        pwmWrite (LED, intensity);
        delay(1);
    }
    delay(1);
}
}

```

Pada pemrograman ini menggunakan library WiringPi, yang dimana untuk mengakses pin GPIO pada raspi, LED didefinisikan untuk menentukan nomor pin GPIO yang akan digunakan untuk mengendalikan LED. Untuk konfigurasi pada pin menggunakan pinMode(LED, PWM_OUTPUT) untuk mengatur pin yang ditentukan dalam konstanta LED sebagai output PWM. Seperti halnya PWM LED diatur dari nilai kecerahan 0 hingga 1023 (nilai maksimum untuk PWM pada WiringPi). Setelah mencapai kecerahan maksimum (1023), terdapat loop kedua yang bertugas untuk mengurangi kecerahan secara bertahap dari 1023 hingga 0. Dan untuk output yang dihasilkan sama seperti menggunakan python.

```
led.c x  phytonled.c x  p3pwm.c x  p3pwm2.c x
1  #include <wiringPi.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  const int LED = 1;
6
7  int main(void){
8      int intensity;
9      if(wiringPiSetup () == -1)
10         exit(1);
11
12         pinMode (LED, PWM_OUTPUT);
13
14     while (1){
15         for (intensity = 0; intensity <1024; ++ intensity)
16         {
17             pwmWrite (LED, intensity);
18             delay(1);
19         }
20         for (intensity = 1023; intensity >= 0; --intensity)
21         {
22             pwmWrite (LED, intensity);
23             delay(1);
24         }
25         delay(1);
26     }
27 }
28
```

4. Untuk program kedua menggunakan bahasa c dan juga library library PWM Software dari wiringPi.

```
#include <wiringPi.h>
#include <stdio.h>
#include <softPwm.h>

int main(void){
    int LED = 1;
    int intensity;
    wiringPiSetup();
    pinMode(LED,OUTPUT);
    softPwmCreate(LED, 1, 100);

    while (1){
        for (intensity = 0; intensity <101; intensity++)
        {
            softPwmWrite (LED, intensity);
            delay(1);
        }
        delay(1);

        for (intensity = 100; intensity >= 0; intensity--)
        {
            softPwmWrite (LED, intensity);
            delay(10);
        }
    }
}
```



```

    delay(1);
  }
}

```

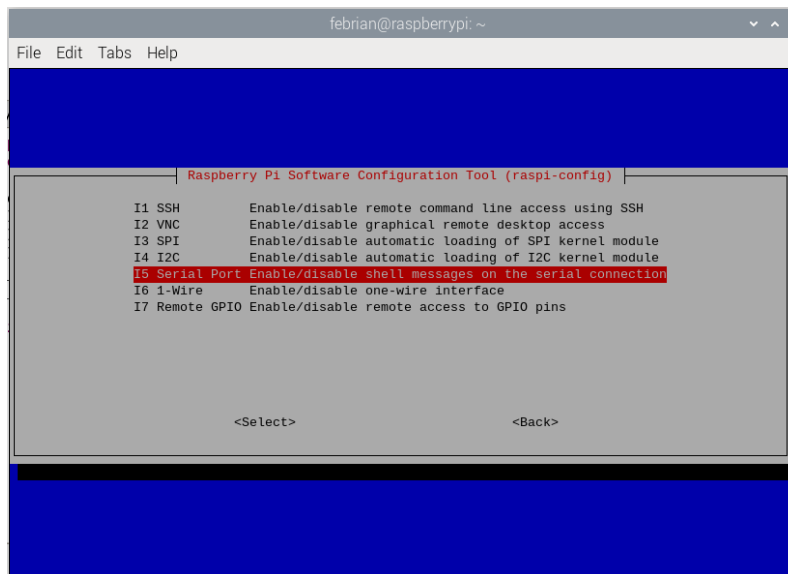
Seperti kode sebelumnya menggunakan bahasa c dengan penggunaan WiringPi namun sebagai soft PWM, Soft PWM digunakan ketika tingkat presisi PWM yang tinggi tidak diperlukan, atau ketika pin GPIO yang tidak mendukung PWM perangkat keras ingin digunakan untuk mengendalikan kecerahan LED. Perbedaan terletak pada inisialisasi fiberukan fungsi `softPwmCreate(LED, 1, 100)` digunakan untuk membuat saluran Soft PWM pada pin yang ditentukan dalam variabel LED. Diartikan bahwa argumen kedua adalah nilai awal kecerahan dan argumen ketiga adalah nilai maksimum. penundaan dalam loop juga disesuaikan untuk menghasilkan efek yang diinginkan dengan nilai kecerahan yang lebih rendah. Output yang diberikan LED berkedip lebih cepat dibanding dengan kode pertama.

II. Raspberry Pi UART Communication using Python and C

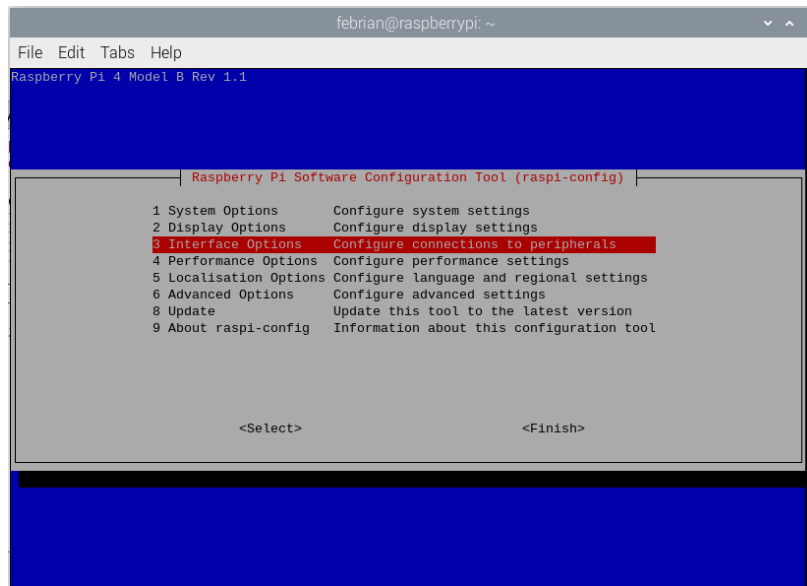
1. Konfigurasi UART pada Raspberry

`sudo raspi-config`

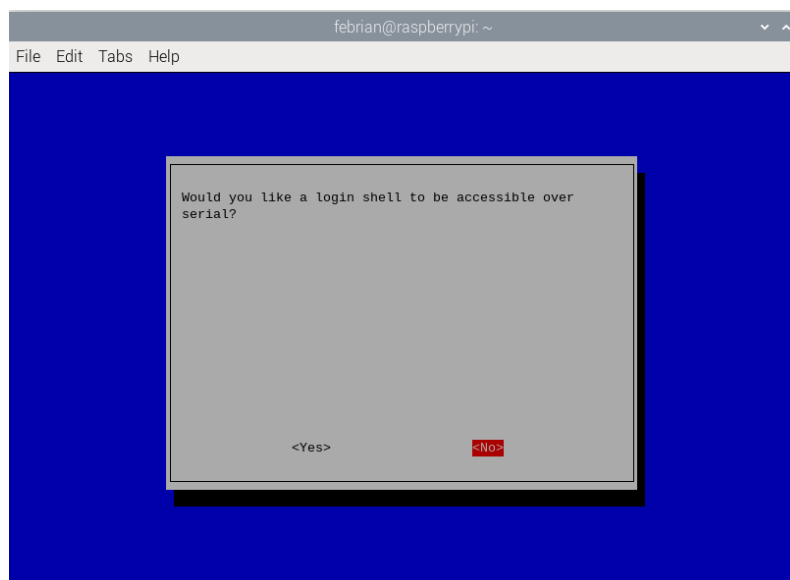
Masukkan kode ke dalam terminal



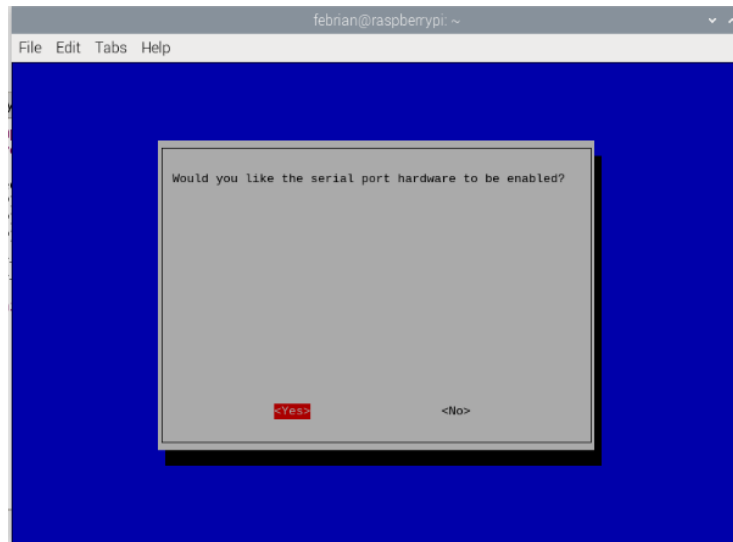
Pilih serial port



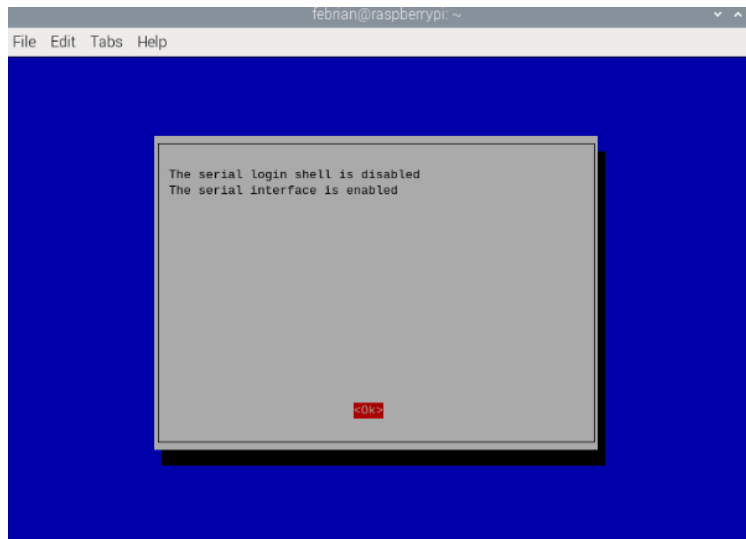
Setelah memilih interface options, lalu pilih interface options



Kemudian akan meminta shell login agar mendapatkan akses melalui serial, pilih NO



Selanjutnya akan diminta untuk mengaktifkan port serial port hardware, pilih YES



Setelah UART telah diaktifkan, reboot raspi

Mini UART dan PL011 UART adalah dua jenis UART yang ada pada Raspberry Pi. Mini UART dipetakan ke pin UART secara default pada Raspberry Pi 3, sedangkan PL011 UART terhubung ke modul Bluetooth on-board pada Raspberry Pi 3. Pada model-model Raspberry Pi sebelumnya, PL011 UART digunakan untuk output console Linux dan tidak ada modul Bluetooth on-board. Port UART adalah antarmuka komunikasi serial yang digunakan untuk mentransfer data antara Raspberry Pi dan perangkat eksternal. Pada Raspberry Pi 3, port UART GPIO14 (TXD) dan GPIO15 (RXD) dikenal sebagai serial0, sedangkan port UART lain

yang terhubung ke modul Bluetooth dikenal sebagai serial1. Nama-nama ini diciptakan sebagai alias serial untuk memudahkan portabilitas antara versi Raspberry Pi. Pemetaan UART merujuk pada pengaturan yang menentukan apakah mini UART (ttyS0) atau PL011 UART (ttyAMA0) yang dipetakan ke pin UART GPIO14 dan GPIO15. Untuk memeriksa pemetaan UART, perintah tertentu dapat digunakan pada terminal Raspberry Pi. Untuk memeriksa pemetaan UART menggunakan

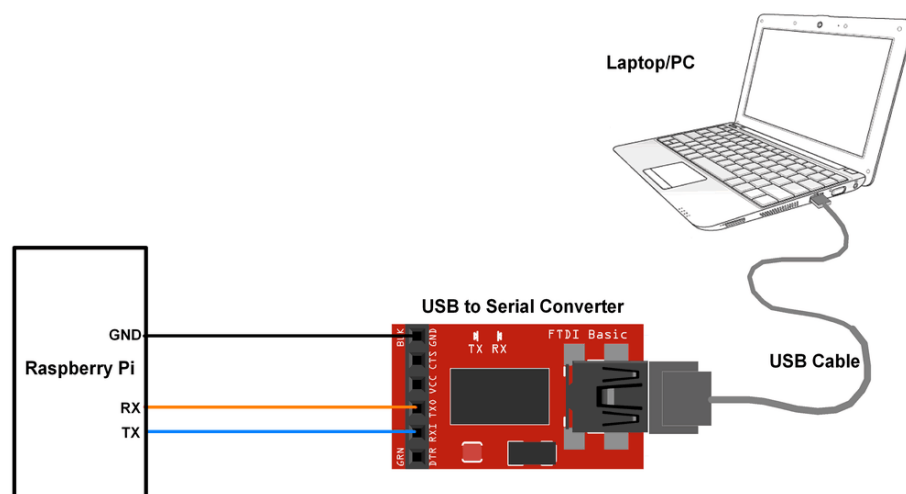
```
ls -l /dev
```

```

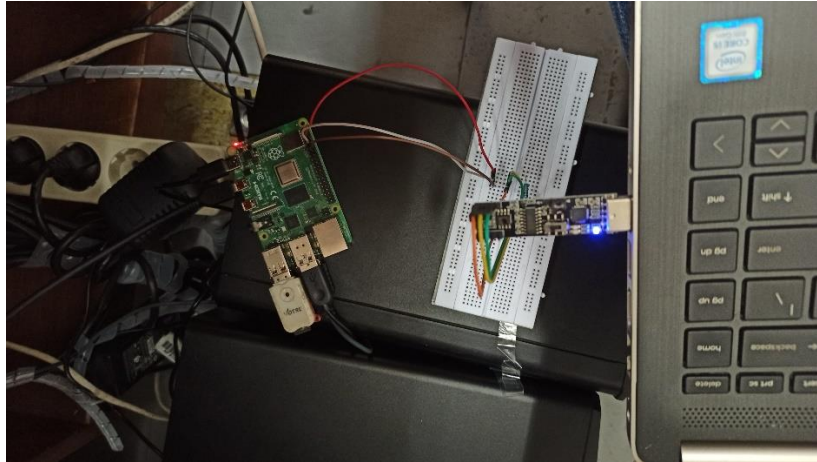
febian@raspberrypi: ~
File Edit Tabs Help
brw-rw---- 1 root disk 1, 6 May 6 09:18 ram6
brw-rw---- 1 root disk 1, 7 May 6 09:18 ram7
brw-rw---- 1 root disk 1, 8 May 6 09:18 ram8
brw-rw---- 1 root disk 1, 9 May 6 09:18 ram9
crw-rw-rw- 1 root root 1, 8 May 6 09:18 random
crw-rw-rw- 1 root netdev 10, 242 May 6 09:18 rfkill
brw-rw---- 1 root disk 8, 0 May 6 09:18 sda
brw-rw---- 1 root disk 8, 1 May 6 09:18 sda1
lrwxrwxrwx 1 root root 5 May 6 09:18 serial0 -> ttyS0
crw-rw---- 1 root disk 22, 0 May 6 09:18 sgl
drwxrwxrwt 2 root root 80 May 6 09:18 tmp
drwxr-xr-x 3 root root 220 May 6 09:18 snd
lrwxrwxrwx 1 root root 15 Jan 1 1970 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root 15 Jan 1 1970 stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root 15 Jan 1 1970 stdout -> /proc/self/fd/1
crw-rw-rw- 1 root tty 5, 0 May 6 09:18 tty
crw--w---- 1 root tty 4, 0 May 6 09:18 tty0
crw----- 1 febian tty 4, 1 May 6 09:18 tty1
crw--w---- 1 root tty 4, 10 May 6 09:18 tty10
crw--w---- 1 root tty 4, 11 May 6 09:18 tty11
crw--w---- 1 root tty 4, 12 May 6 09:18 tty12
crw--w---- 1 root tty 4, 13 May 6 09:18 tty13
crw--w---- 1 root tty 4, 14 May 6 09:18 tty14
crw--w---- 1 root tty 4, 15 May 6 09:18 tty15

```

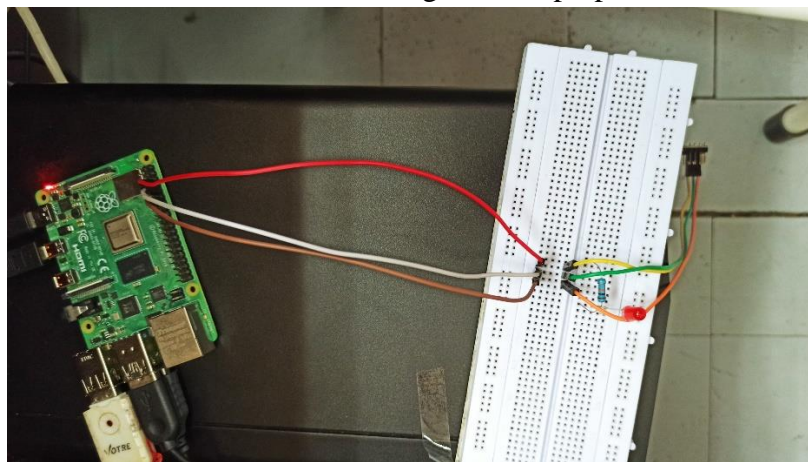
2. Test serial communication antara Raspberry Pi dengan PC



Untuk pemasangan serial harus terbalik TX > RX, RX > TX



Contoh dihubungkan ke laptop



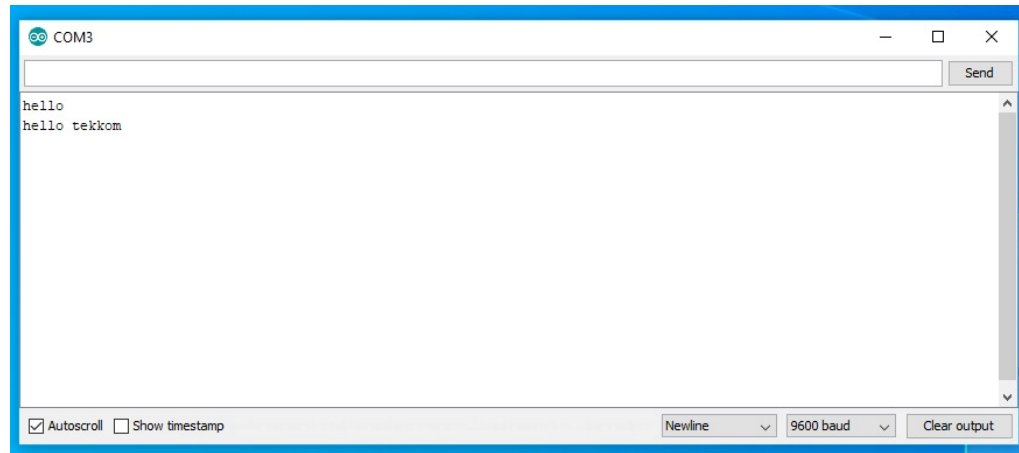
Contoh dihubungkan ke PC

3. Percobaan untuk menerima data yang akan dikirim Raspberry maupun sebaliknya

Menggunakan perintah yaitu `echo "Hello" > /dev/ttyS0`

```
febrian@raspberrypi: ~  
File Edit Tabs Help  
febrian@raspberrypi:~ $ echo "hello" > /dev/ttyS0  
febrian@raspberrypi:~ $ echo "hello" > /dev/ttyS0  
febrian@raspberrypi:~ $ echo "hello tekkom" > /dev/ttyS0  
febrian@raspberrypi:~ $
```

Perintah ini akan menampilkan Hello pada port UART yaitu pin TX dan akan dimunculkan kedalam aplikasi terminal, kami menggunakan terminal pada arduino ide, pastikan baud rate ke 9600.

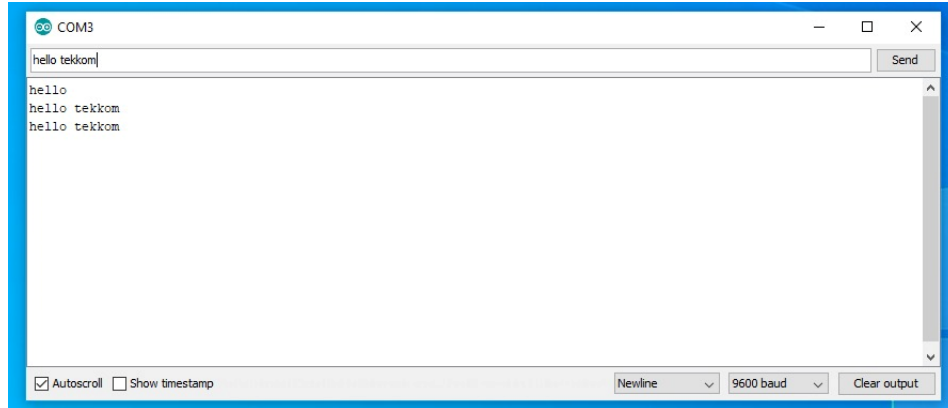


4. Pemrograman UART dengan Raspi menggunakan python

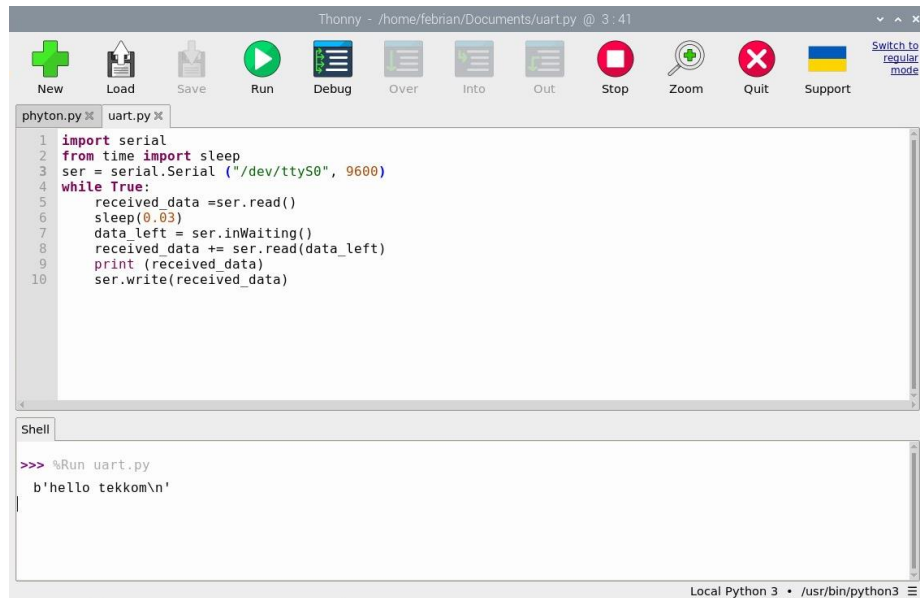
```
import serial
from time import sleep
ser = serial.Serial ("/dev/ttyS0", 9600)
while True:
    received_data = ser.read()
    sleep(0.03)
    data_left = ser.inWaiting()
    received_data += ser.read(data_left)
    print (received_data)
    ser.write(received_data)
```

Kode ini adalah penggunaan modul serial dalam bahasa python untuk berkomunikasi melalui port UART, dalam konfigurasi, menentukan objek serial. `Serial("/dev/ttyS0", 9600)`. Ini mengatur komunikasi serial dengan menggunakan port `/dev/ttyS0` pada kecepatan baud 9600 bit per detik. Di dalam loop, `ser.read()` digunakan untuk membaca data dari port serial. Data yang diterima disimpan dalam variabel `received_data`. Kemudian, `sleep(0.03)` digunakan untuk memberikan jeda sejenak sebelum memeriksa apakah ada data yang tersisa di buffer serial. Untuk `ser.inWaiting()` digunakan untuk memeriksa jumlah data yang masih tersimpan di buffer serial. Jika ada data yang tersisa (`data_left > 0`), `ser.read(data_left)` digunakan untuk membaca data yang tersisa dan menambahkannya ke `received_data`. Setelah membaca dan menggabungkan semua data yang tersedia,

received_data dicetak ke konsol menggunakan `print(received_data)`. Selanjutnya, data yang diterima juga dikirim kembali ke perangkat yang mengirimnya menggunakan `ser.write(received_data)`.



Contoh mengirim data menggunakan Arduino IDE



5. Komunikasi UART pada Raspberry Pi menggunakan C

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <wiringPi.h>
#include <wiringSerial.h>
```

```

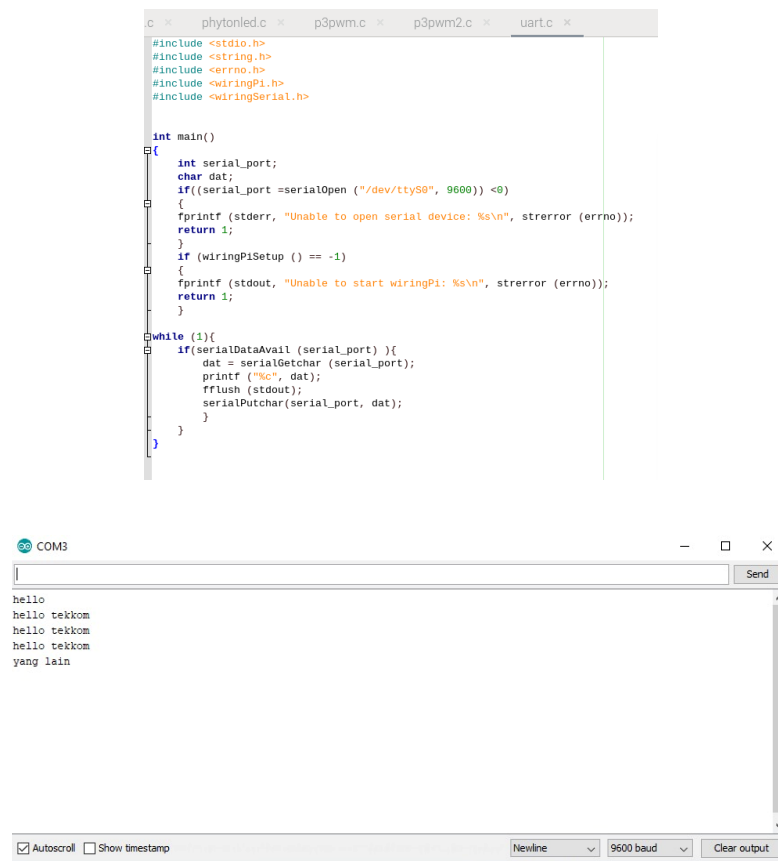
int main()
{
    int serial_port;
    char dat;
    if((serial_port = serialOpen ("/dev/ttyS0", 9600)) < 0)
    {
        fprintf (stderr, "Unable to open serial device: %s\n",
strerror (errno));
        return 1;
    }
    if (wiringPiSetup () == -1)
    {
        fprintf (stdout, "Unable to start wiringPi: %s\n", strerror
(errno));
        return 1;
    }

    while (1){
        if(serialDataAvail (serial_port) ){
            dat = serialGetchar (serial_port);
            printf ("%c", dat);
            fflush (stdout);
            serialPutchar(serial_port, dat);
        }
    }
}

```

Kode ini berfungsi untuk membaca dan menuliskan data melalui port UART menggunakan library wiringPi, untuk mendapatkan fungsi-fungsi dasar wiringPi menggunakan library wiringPi.h dan untuk fungsi serial menggunakan wiringSerial.h, Variabel serial_port dideklarasikan untuk menyimpan file descriptor untuk port serial. Variabel dat digunakan untuk menyimpan data yang diterima dari port serial. Fungsi serialOpen("/dev/ttyS0", 9600) digunakan untuk membuka port serial /dev/ttyS0 pada kecepatan baud 9600 bit per detik. Jika pembukaan port gagal, pesan kesalahan akan dicetak ke stderr. Di dalam loop utama, fungsi serialDataAvail(serial_port) digunakan untuk memeriksa apakah ada data yang tersedia untuk dibaca dari port serial. Jika ada, fungsi serialGetchar(serial_port) digunakan untuk membaca satu karakter dari port serial dan disimpan dalam variabel dat. kemudian dicetak ke konsol menggunakan

printf("%c", dat) dan di-flush menggunakan fflush(stdout) untuk memastikan keluaran segera ditampilkan.



The image shows a C program in a text editor and its output in a serial monitor. The C program, named 'uart.c', includes headers for stdio, string, errno, wiringPi, and wiringSerial. It defines a main function that opens a serial port at '/dev/ttyS0' with a baud rate of 9600. It checks for errors in opening the port and starting wiringPi. A while loop reads data from the serial port, prints it to the console, flushes the output, and then writes it back to the serial port. The serial monitor window, titled 'COM3', shows the output: 'hello', 'hello tekkom', 'hello tekkom', 'hello tekkom', and 'yang lain'. The monitor settings are set to 9600 baud and 'Newline'.

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <wiringPi.h>
#include <wiringSerial.h>

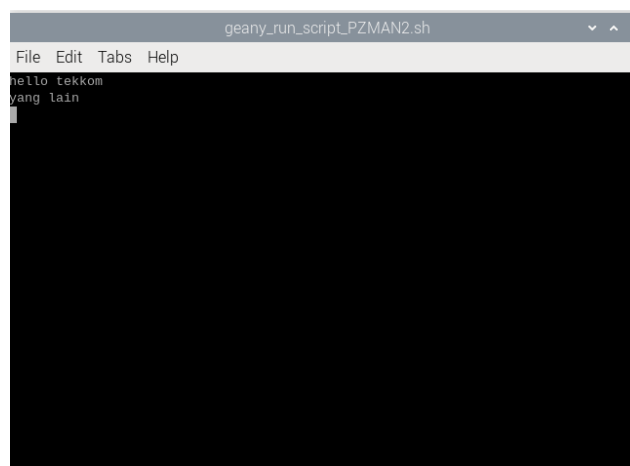
int main()
{
    int serial_port;
    char dat;
    if((serial_port = serialOpen ("/dev/ttyS0", 9600)) < 0)
    {
        fprintf (stderr, "Unable to open serial device: %s\n", strerror (errno));
        return 1;
    }
    if (wiringPiSetup () == -1)
    {
        fprintf (stdout, "Unable to start wiringPi: %s\n", strerror (errno));
        return 1;
    }
    while (1){
        if(serialDataAvail (serial_port) ){
            dat = serialGetchar (serial_port);
            printf ("%c", dat);
            fflush (stdout);
            serialPutchar(serial_port, dat);
        }
    }
}
```

COM3

hello
hello tekkom
hello tekkom
hello tekkom
yang lain

Autoscroll Show timestamp Newline 9600 baud Clear output

Dengan memasukkan kata “yang lain”



The image shows a terminal window titled 'geany_run_script_PZMAN2.sh'. The terminal displays the output of a script: 'hello tekkom' and 'yang lain'. The rest of the terminal is black, indicating that the output has been scrolled out of view.

```
geany_run_script_PZMAN2.sh
File Edit Tabs Help
hello tekkom
yang lain
```

Output pada terminal Raspi

III. Tugas

1. Apa yang dimaksud dengan TTL?

TTL adalah singkatan dari "Transistor-Transistor Logic". Ini adalah suatu metode atau standar untuk menerjemahkan sinyal digital dalam elektronika. TTL merujuk pada jenis logika digital yang menggunakan transistor sebagai komponen utama dan memiliki dua tingkat tegangan untuk merepresentasikan nilai logika digital.

2. Apa yang dibutuhkan untuk bisa koneksi serial?

Yang dibutuhkan untuk bisa konek serial adalah Hardware, port serial, perangkat lunak (terminal serial, atau kode program), konfigurasi port.

3. Apa yang dimaksud serial RS232?

Serial RS232 adalah standar komunikasi serial yang dikembangkan oleh Electronic Industries Association (EIA) pada tahun 1960-an. RS dalam RS232 singkatan dari "Recommended Standard", dan angka 232 mengidentifikasi nomor standar tersebut. RS232 adalah metode komunikasi serial yang digunakan untuk mentransmisikan data antara dua perangkat elektronik melalui kabel yang menghubungkan keduanya.

4. Apa yang dimaksud dengan komunikasi sinkron dan komunikasi asinkron?

- Komunikasi sinkron, pengiriman data dilakukan dalam waktu yang teratur dan disesuaikan dengan sinyal clock yang terpisah. Contoh sistem yang menggunakan komunikasi sinkron termasuk Ethernet dan protokol komunikasi sinkron pada bus data dalam komputer.
- Komunikasi asinkron, pengiriman data tidak disesuaikan dengan sinyal clock terpisah. Sebaliknya, setiap karakter atau byte data dikirimkan secara individual dengan bit awal dan bit akhir yang menandai awal dan akhir setiap data. Contoh sistem yang menggunakan komunikasi asinkron termasuk port serial RS232, USB (Universal Serial Bus), dan komunikasi melalui jaringan.

5. Apa yang dimaksud dengan baudrate?

Baudrate adalah ukuran yang digunakan untuk menentukan kecepatan transfer data dalam komunikasi serial. Ini mengukur jumlah simbol data yang dapat ditransmisikan per detik melalui kanal komunikasi.