



VTI1E2 – APLIKASI MIKROKONTROLER dan ANTARMUKA[©] SEMESTER GANJIL – KURIKULUM 2020

Denny Darlis S.Si., M.T. - 13770026

Program Studi D3 Teknologi Telekomunikasi
Fakultas Ilmu Terapan - Universitas Telkom

VTI2D3
Aplikasi Mikrokontroler dan Antarmuka
Materi ke-3: Pemrograman Dasar Mikrokontroler I (Micropython)

Denny Darlis, S.Si., M.T.

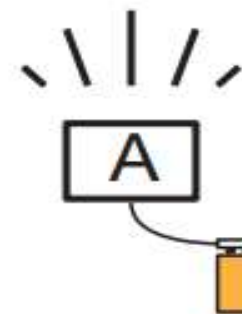
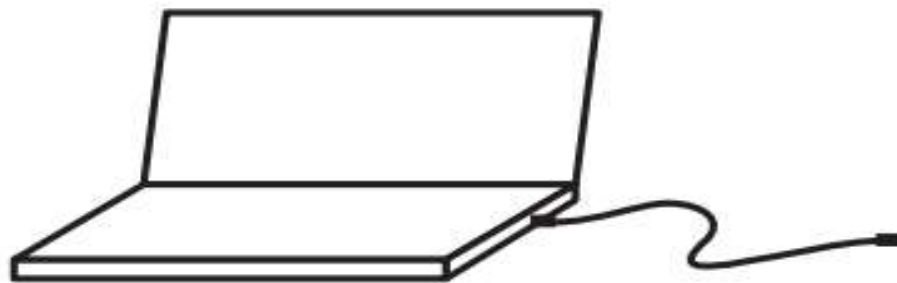
13770026

Bagaimana kita membuat Aplikasi Mikrokontroler?

1. Mengetahui type/jenis mikrokontrolernya -> ATmega328? ESP8266? ESP32? dll -> cek datasheet
2. Mengetahui antarmuka yang akan digunakan -> Digital? Analog? Serial? Paralel? SPI? I2C? Dll -> cek datasheet
3. Mengetahui sistem minimum mikrokontrolernya (untuk prototyping) -> Arduino Uno? Modul ESP8266? Modul ESP32? Dll -> cek manual guide
4. Mengetahui komponen elektronika utama dan tambahan (untuk implementasi) -> pengukuran (sensor)? Kendali (aktuator)? AI? dll -> cek kebutuhan sistem
5. Mengetahui bahasa pemrograman yang bisa digunakan -> Assembly? C/C++? C Arduino? Micropython? Lua? Processing? dll
6. Mengetahui tools dan/atau IDE yang sesuai -> Compiler? Linker? Visual Studio? Arduino IDE? Thonny IDE? uPyCraft? dll

Running Code Stand-Alone

Run Arduino in stand alone mode



Arduino interacts with its environment and runs on battery power

- ▶ **Apa itu MicroPython?**
- ▶ MicroPython adalah implementasi ulang Python 3 yang ditargetkan untuk mikrokontroler dan sistem tertanam. MicroPython sangat mirip dengan Python biasa. Jadi, jika Anda sudah tahu cara memprogram dengan Python, Anda juga tahu cara memprogram di MicroPython.
- ▶ **Python vs MicroPython**
- ▶ Terlepas dari beberapa pengecualian, fitur bahasa Python juga tersedia di MicroPython. Perbedaan terbesar antara Python dan MicroPython adalah bahwa MicroPython dirancang untuk bekerja dalam kondisi terbatas.



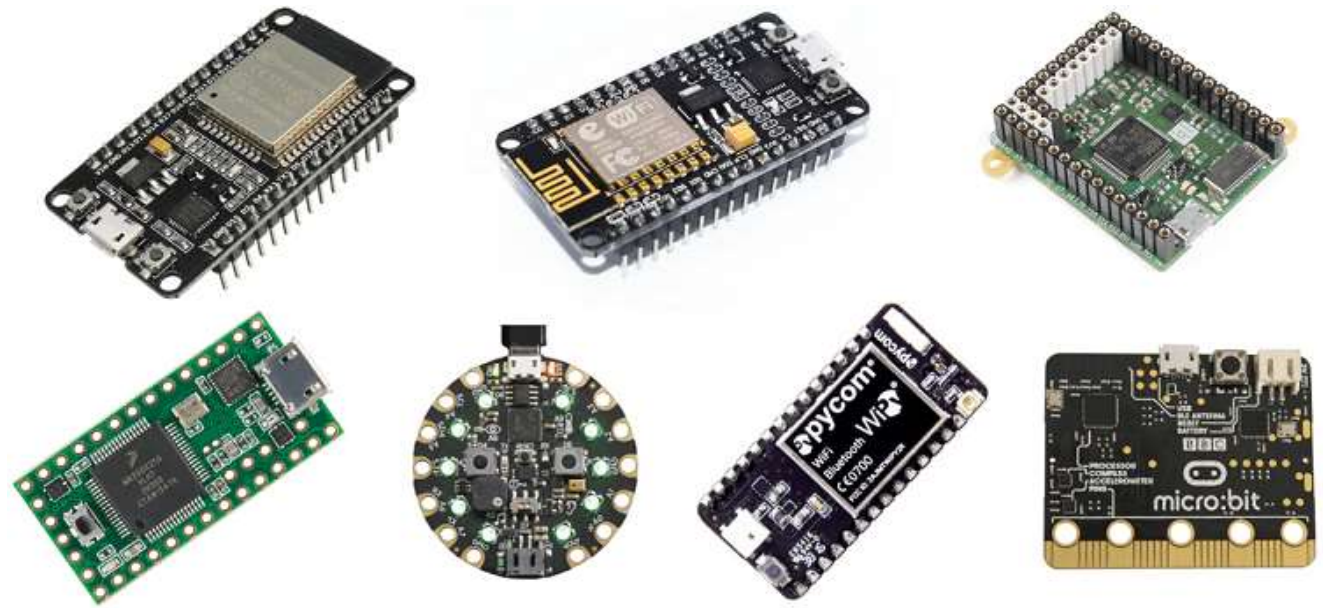
- ▶ Karena itu, MicroPython tidak dilengkapi dengan pustaka standar lengkap. Ini hanya mencakup subset kecil dari pustaka standar Python. Namun, itu termasuk modul untuk mengakses perangkat keras tingkat rendah - ini berarti bahwa ada perpustakaan untuk dengan mudah mengakses dan berinteraksi dengan GPIO.
- ▶ Selain itu, perangkat dengan kemampuan Wi-Fi seperti ESP8266 dan ESP32 menyertakan modul untuk mendukung koneksi jaringan

Mengapa MicroPython?

- ▶ Python adalah salah satu bahasa pemrograman yang paling banyak digunakan, sederhana dan mudah dipelajari. Jadi, munculnya MicroPython membuatnya sangat mudah dan sederhana untuk memprogram elektronika digital. Jika Anda belum pernah memprogram elektronik digital sebelumnya, MicroPython adalah titik awal yang baik.
- ▶ Tujuan MicroPython adalah membuat pemrograman elektronik digital sesederhana mungkin, sehingga dapat digunakan oleh siapa saja. Saat ini, MicroPython digunakan oleh penghobi, peneliti, guru, pendidik, dan bahkan dalam produk komersial.

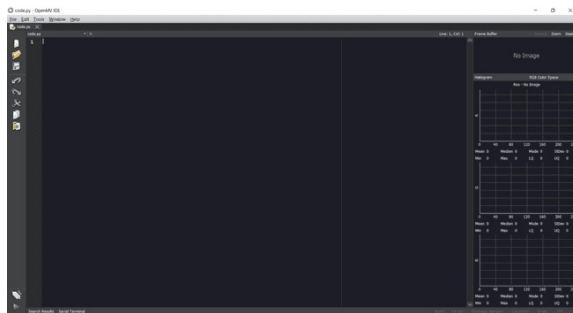
Papan yang Kompatibel dengan MicroPython

- [ESP32](#)
- [ESP8266](#)
- PyBoard
- Micro:Bit
- Teensy 3.X
- WiPy – Pycom
- Adafruit Circuit Playground Express
- Other ESP32/ESP8266 based boards



Beberapa Editor MicroPython

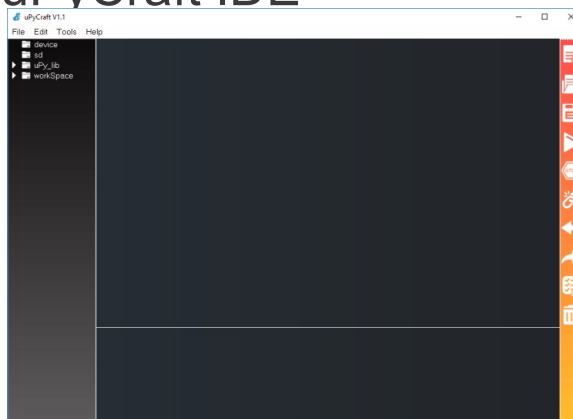
▶ OpenMV



▶ Thonny IDE



▶ uPyCraft IDE



Contoh program micropython

Kode untuk mengedipkan LED pada ESP32 atau ESP8266 sesederhana berikut:

```
# Complete project details at
https://RandomNerdTutorials.com
from machine import Pin
from time import sleep

led = Pin(2, Pin.OUT)

while True:
    led.value(not led.value())
    sleep(0.5)
```

Skrip sederhana yang akan mencetak "Hello world!" setiap detik.

```
import time

content = "Hello world!"

while True:
    print(content)
    time.sleep(1)
```

Skrip ini dicetak "Hello world!" setiap detik. Sebagai tambahan, `counter_function()` juga

```
import time

content = "Hello world!"
count = 0

def counter_function():
    global count
    count = count + 1

while True:
    counter_function()
    print(content, count)
    time.sleep(1)
```

Contoh program micropython

For loop

Simple use of a for loop and functions. This script counts to 10, and then back to 0.

```
import time

content = "Hello world!"
count = 0

def function_increase():
    global count
    count = count + 1
    print(count)

def function_decrease():
    global count
    count = count - 1
    print(count)

while True:
    for x in range(10):
        function_increase()
        time.sleep(1)

    for x in range(10):
        function_decrease()
        time.sleep(1)
```

Delay

To use a simple delay, we can use the `time` module. If we want to write in seconds, we can

```
USE time.sleep(seconds), and for
milliseconds time.sleep_ms(milliseconds).

import time

while True:
    time.sleep(0.5) #or time.sleep_ms(500)
    print("Hello world!")

    # ...
```

PWM

Write a specific duty to a specific pin.

```
from machine import Pin, PWM, ADC

pwm = PWM(Pin(15))
duty = 30000 #between 0-65000

pwm.freq(1000)

while True:
    pwm.duty_u16(duty)
```

Interrupt

Below is an example of a simple interrupt that uses a pull up button and an LED.

The program blinks an LED, until the button is pressed. The button is attached to an interrupt, which turns off an LED for 3 seconds.

```
import machine
from machine import Pin
import time

interrupt = False

def callback(pin):
    global interrupt
    interrupt = True

led = Pin(6, Pin.OUT)

button = machine.Pin(25, machine.Pin.IN, machine.Pin.PULL_UP)

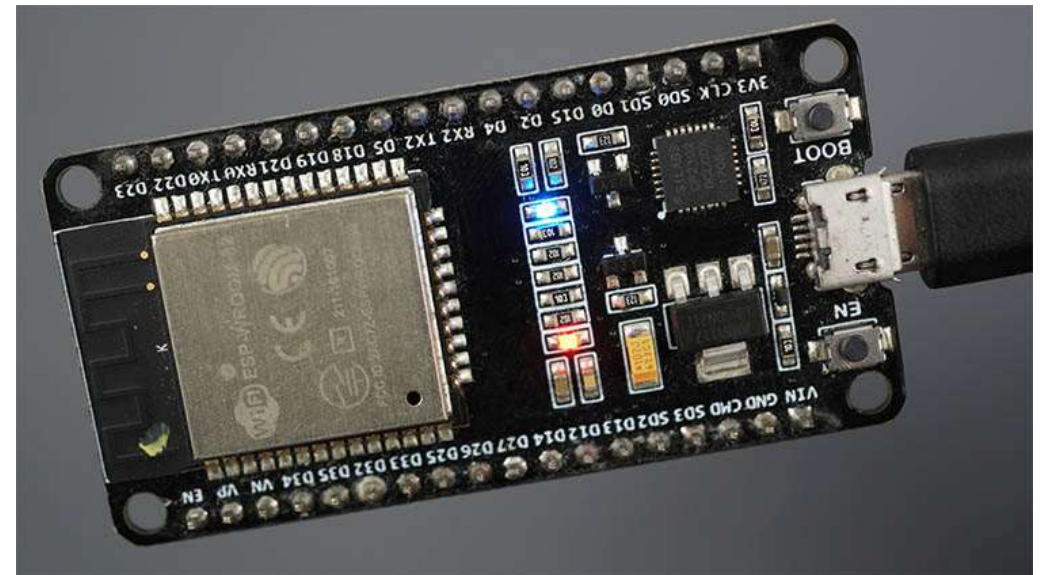
button.irq(trigger=machine.Pin.IRQ_FALLING, handler=callback)

while True:
    led.on()
    time.sleep(0.5)
    led.off()
    time.sleep(0.5)

    if interrupt:
        state = machine.disable_irq()
        machine.enable_irq(state)
        led.off()
        print("Interrupt: LED off for 3 seconds!")
        time.sleep(3)
        interrupt = False
```

Contoh Program Sederhana

```
Thonny - C:\Users\Sara\Desktop\untitled.py @ 1:1
File Edit View Run Device Tools Help
untitled.py x
1
Shell x
MicroPython v1.9.4-779-g5064df207 on 2019-01-13; ESP32 module with ESP
32
Type "help()" for more information. [backend=ESP32]
>>> from machine import Pin
>>> Pin(2, Pin.OUT).value(1)
>>> |
```



1. Steven F. Barrett and Daniel J. Pack, "Microcontrollers Fundamentals for Engineers and Scientists", 2006, Morgan & Claypool
2. <https://docs.arduino.cc/learn/programming/arduino-and-python>
3. <https://randomnerdtutorials.com/getting-started-micropython-esp32-esp8266/>
4. <https://randomnerdtutorials.com/getting-started-micropython-esp32-esp8266/>
5. <https://docs.micropython.org/en/latest/esp32/quickref.html>
6. <https://docs.micropython.org/en/latest/esp8266/quickref.html>
7. <https://docs.micropython.org/en/latest/rp2/quickref.html>

Ada pertanyaan?

**Semoga Bermanfaat dan
Terima Kasih atas Perhatiannya**