

PETUNJUK PRAKTIKUM SISTEM EMBEDDED DENGAN MICROPYTHON



Disusun oleh:

Alfian Ma'arif, S. T., M. Eng.

**LABORATORIUM OTOMASI DAN SISTEM EMBEDDED
PRAGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN
2020**

Daftar Isi

Prakt. Sistem Embedded Unit 1 Pengenalan Bahasa Pemrograman Python dan MicroPython.....	3
Prakt. Sistem Embedded Unit 2 Bahasa Pemrograman Python dan Input Output.....	12
Prakt. Sistem Embedded Unit 3 Analog to Digital Converter (ADC)	18
Prakt. Sistem Embedded Unit 4 I2C LCD	23
Prakt. Sistem Embedded Unit 5 Mini Projek.....	28

Praktikum Sistem Embedded Unit 1

Pengenalan Bahasa Pemrograman Python dan MicroPython

oleh Alfian Ma'arif, M. Eng.

Tujuan

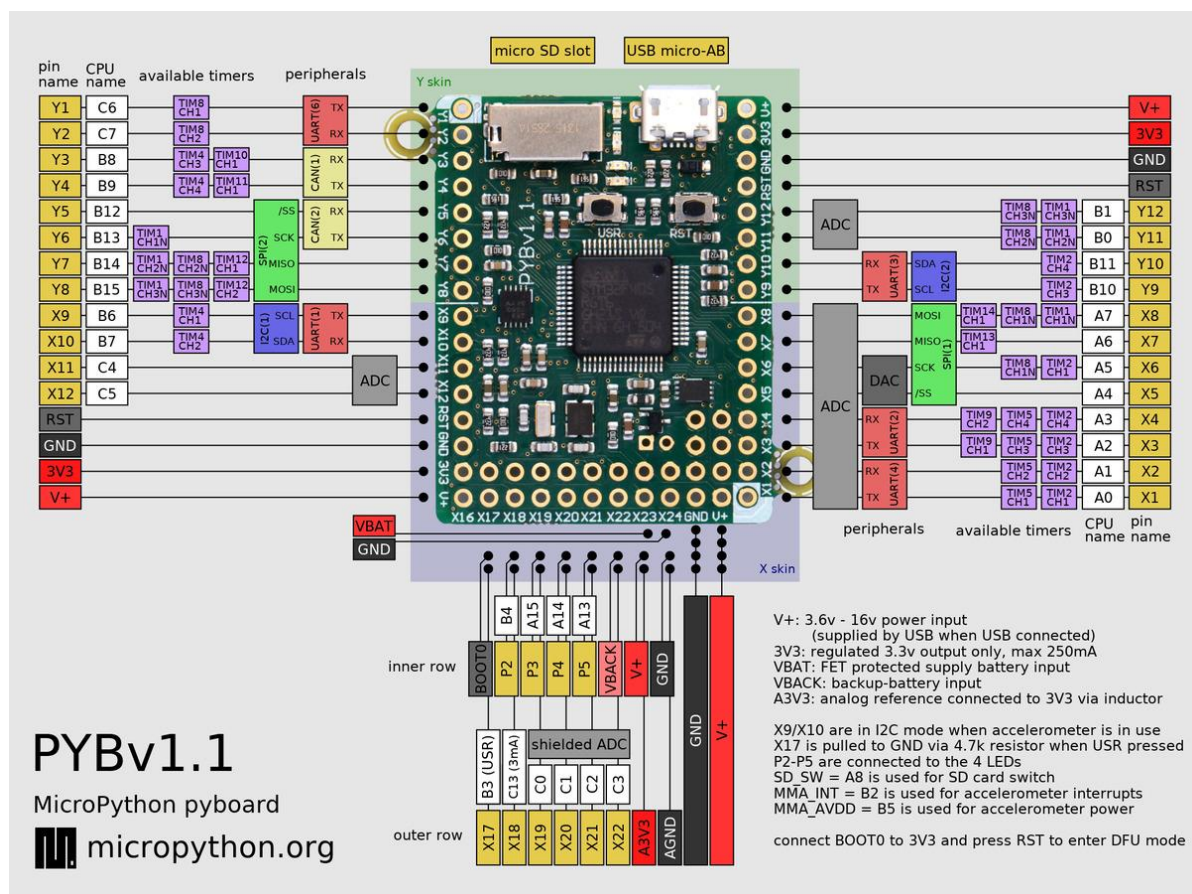
Tujuan Praktikum Sistem Embedded Unit 1 adalah

1. Mempelajari tentang Bahasa pemrograman MicroPython
2. Menjalankan program sederhana pada Board MicroPython PYBv1.1.

MicroPython

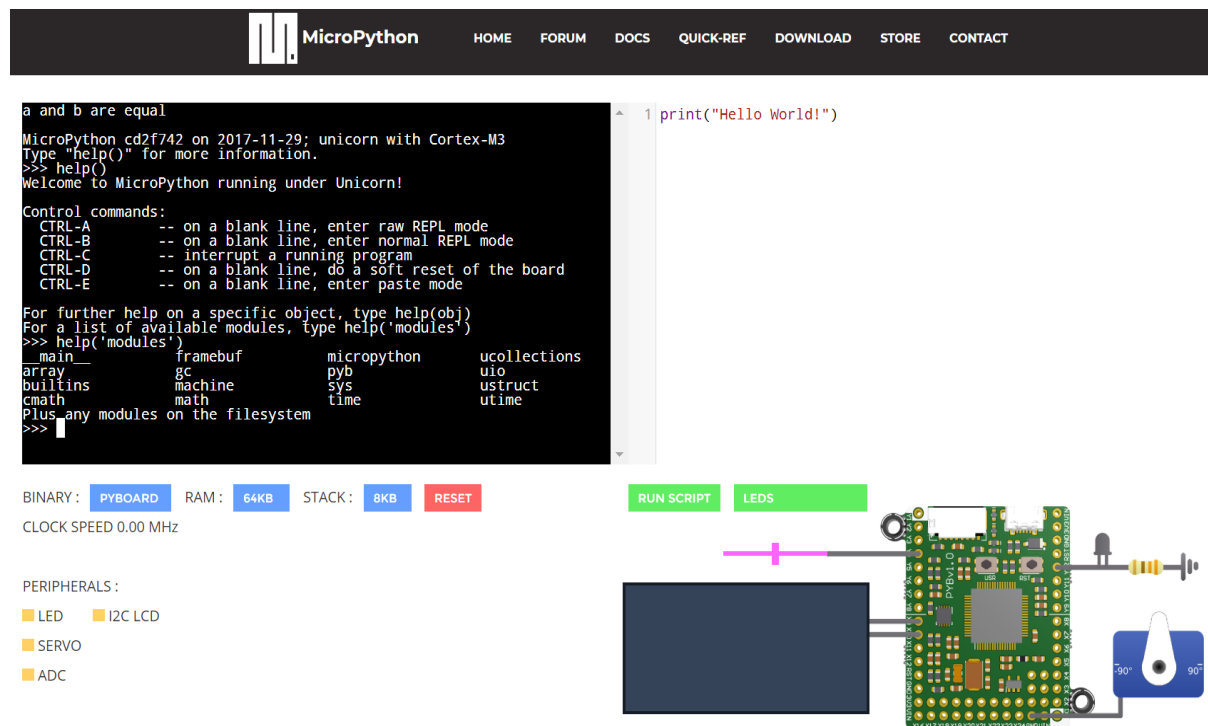
Bahasa python merupakan 3 bahasa yang paling banyak digunakan setelah javascript dan HTML. Bahasa Python memiliki aplikasi yang luas, salah satunya adalah Bahasa Python untuk sistem mikroprosesor dan sistem embedded yaitu MicroPython.

Pada praktikum sistem embedded ini akan digunakan Board MicroPython yang ditunjukkan pada Gambar 1.1. Board tersebut menggunakan mikroprosesor STM32F405RGT6.



Gambar 1. 1. Mikrokontroler PYB.v1.1.

Halaman website microPython ditunjukkan pada Gambar 1.1. Terdapat beberapa bagian pada halaman tersebut, keterangan bagian-bagian website tersebut ditunjukkan pada Tabel 1.1.



Gambar 1. 2. Tampilan Website MicroPython

Tabel 1. 1. Bagian-Bagian Website Unicorn MicroPython

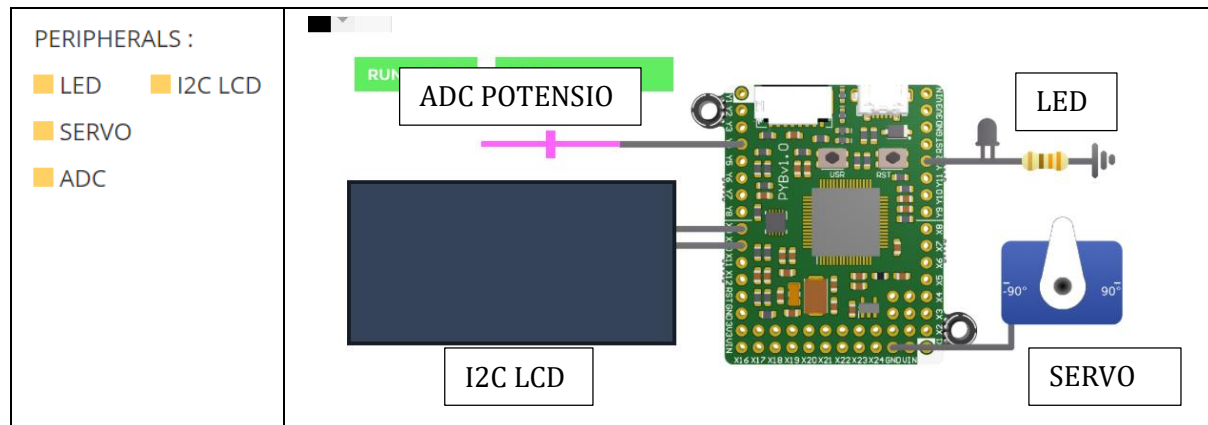
Nama Bagian	Fungsi
Terminal	Bagian untuk menampilkan tulisan dan mengetikkan program secara langsung
Editor	Bagian untuk mengetikkan program secara tidak langsung
Peripherals	Bagian konfigurasi untuk menampilkan LED, Servo, ADC dan I2C pada Board PYBv1.1.
Board PYBv1.1.	Bagian Sistem Minimum Microcontroller dan Simulasi Hardware

Tombol konfigurasi ditunjukkan pada Gambar 1.4. Tombol Reset berfungsi untuk mengulang program dari awal. Tombol Run Script berfungsi untuk menjalankan program yang telah dituliskan pada editor.



Gambar 1. 3. Tombol Konfigurasi

Bagian-Bagian Board PYBv1.1. ditunjukkan pada Gambar. Bagian-Bagian tersebut adalah Lampu LED, Potensiometer, Motor Servo, dan LCD I2C. Tekan pilihan kotak berwarna kuning di samping tulisan perangkat pada bagian Peripherals untuk memunculkan perangkat.



Gambar 1. 4. Sistem Minimum PYB.v1.1.

Library MicroPython

Dalam Bahasa Python, untuk menggunakan suatu perintah tertentu, perlu dilakukan pemanggilan pada awal program. Proses tersebut menggunakan perintah **import**. Beberapa library yang akan digunakan dalam praktikum ini adalah sebagai berikut.

Lampu LED Internal

Library LED internal ditunjukkan pada Program 1.1.

Program 1. 1.	
<code>from pyb import LED</code>	# memanggil fungsi LED pada library pyb
<code>led = LED(1)</code>	# 1=merah, 2=hijau, 3=kuning, 4=biru
<code>led.on()</code>	# menyalakan lampu LED
<code>led.off()</code>	# mematikan lampu LED

Delay/Waktu Tunda

Library Delay/Waktu tunda internal ditunjukkan pada Program 1.1.

Program 1. 2.	
<code>import time</code>	# memanggil library time
<code>time.sleep(1)</code>	# waktu tunda 1 detik
<code>time.sleep_ms(500)</code>	# waktu tunda 500 milidetik
<code>time.sleep_us(10)</code>	# waktu tunda 10 mikrodetik

Tombol Internal

Library Delay/Waktu tunda internal ditunjukkan pada Program 1.1.

Program 1. 3.	
<pre>from pyb import Switch # memanggil fungsi Switch pada library pyb sw = Switch() sw.value() # mengembalikan nilai True(1) atau False(0)</pre>	

Pre Test

Bacalah dasar teori pada sub bab sebelumnya lalu kerjakan soal Pre Test berikut dan tuliskan jawaban pada Laporan.

1. Apa fungsi bagian Terminal, Editor, Peripherals dan Board pada website MicroPython?
2. Sebutkan jenis peripherals (perangkat) pada Board PYBv1.1.!
3. Sebutkan apa saja library python yang akan dipelajari pada praktikum unit 1 ini?

Langkah Praktikum dan Post Test

Silahkan buka website MicroPython Berikut.

<http://micropython.org/unicorn/>

Pemrograman Bahasa Python

a. Membuat Tulisan

Ketikan perintah 1.4 pada bagian Editor MicroPython, lalu tekan tombol Run Script. Amati hasil yang ditampilkan pada Terminal.

Program 1. 4.	
<pre>print("Hello World!")</pre>	

1. Apa fungsi perintah print("Hello World!") pada Program 1.4?

b. Membuat Perintah Kondisi

Ketikan perintah pada Program 1.5 pada Editor, lalu tekan tombol Run Script.

Program 1. 5.	
<pre>a=10 b=5 if a>5: print("variabel a lebih besar dari b")</pre>	

2. Tuliskan hasil yang ditampilkan pada Terminal. Mengapa bisa ditampilkan tulisan tersebut? Apa hubungannya dengan nilai variabel a dan b?
3. Buatlah flowchart Program 1.5!

c. Membuat Perintah Perulangan

While

Ketikan perintah pada Program 1.6 untuk membuat deret angka kemudian jalankan program.

Program 1. 6.
<pre>i=1 while i<=10: print(i) i=i+1</pre>

4. Berdasarkan Program 1.6, berapa nilai awal deret angka? Berapa nilai akhir deret angka? Tuliskan hasil running program dan jelaskan program tersebut.

For

Ketikan program 1.7 pada editor, kemudian jalankan program tersebut.

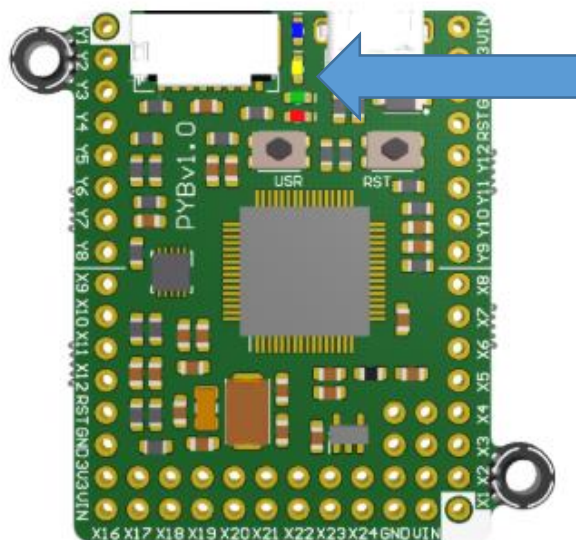
Program 1. 7.
<pre>for i in range(0,10): print(i)</pre>

5. Berdasarkan pada Program 1.7, berapa nilai awal deret angka? Berapa nilai akhir deret angka? Tuliskan hasil running program dan jelaskan program tersebut.

6. Bandingkan hasil deret angka dengan perintah while dan for. Apa perbedaan antara kedua program tersebut? Mengapa hasilnya bisa berbeda?

Menyalakan Lampu LED Internal

Lampu LED internal pada Board PYBv1.0 ditunjukkan pada Gambar 1.5. Terdapat 4 buah LED pada board tersebut yang berwarna merah, hijau, kuning dan Biru.



Gambar 1. 5. LED Internal Board PYBv1.0

Ketikkan Program 1.8 pada Terminal (bukan Editor). Tekan tombol enter setelah mengetikkan satu baris program

Program 1. 8.
<pre>from pyb import LED LED(1).on()</pre>

7. Amati pada board, lampu LED warna apa yang menyala?

8. Apa fungsi perintah LED(1).on()?

Catatan: Perintah from pyb import LED berfungsi untuk memanggil fungsi LED pada library/pustaka pyb.

Tambahkan Program 1.9 pada Terminal.

Program 1. 9.
<pre>LED(1).off()</pre>

9. Amati lampu LED pada Board, apa fungsi perintah LED(1).off()?

Tekan tombol Reset untuk mengulang program.

Menyalakan Beberapa Lampu LED

Ketikkan Program 1.10 pada Editor lalu tekan tombol Run Script.

Program 1. 10.
<pre>from pyb import LED LED(1).on() LED(2).on()</pre>

10. Amati lampu LED, warna lampu LED apa saja yang menyala?

11. Tuliskan program untuk menyalakan keempat lampu LED dan berikan keterangan fungsi setiap baris program di bagian kanan setiap perintah.

12. Gambarkan Tabel 1.2 pada laporan dan lengkapi bagian yang kosong.

Tabel 1. 2. Hasil Pengamatan Lampu LED

Nomor Lampu LED	Warna Lampu LED
LED(1)	
LED(2)	
LED(3)	
LED(4)	

13. Tuliskan program untuk mematikan semua lampu LED.

Menyalakan dan Mematikan Lampu LED dengan waktu tunda

Ketikan Program 1.11 pada Editor lalu tekan tombol Run Script.

Program 1. 11.

```
from pyb import LED
import time
ledM=LED(1)
while True:
    ledM.on()
    time.sleep(1)
    ledM.off()
    time.sleep(1)
```

Perintah while True: berfungsi untuk membuat program berjalan secara terus menerus. Fungsinya sama dengan while(1) pada Bahasa C/C++.

14. Amati lampu LED merah. Bagaimana kondisi Lampu LED merah? Mengapa lampu LED merah bisa berada dalam kondisi tersebut?

15. Tuliskan Program 1.11 dan jelaskan cara kerja program tersebut.

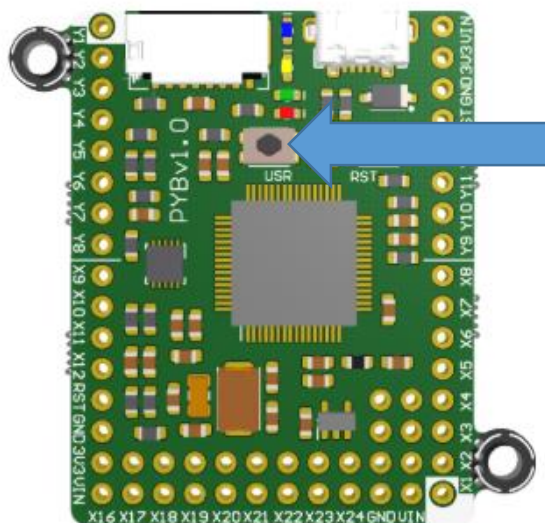
16. Buatlah flowchart untuk program di atas.

Catatan: Perintah import time berfungsi untuk memanggil library time sehingga dapat digunakan fungsi atau perintah time.sleep(1).

17. Buatlah program untuk menyalakan dan mematikan keempat lampu LED dengan waktu tunda 1 detik.

Menyalakan Lampu LED dengan Tombol

Tombol Internal Board PYBv1.1 ditunjukkan pada Gambar 1.6.



Gambar 1. 6. Tombol Internal Board PYBv1.1

Ketikkan program 1.12 pada Editor, lalu tekan tombol Run Script.

Program 1. 12.
<pre> import time import pyb while True: if pyb.Switch().value(): pyb.LED(1).on() time.sleep_ms(50) else: pyb.LED(1).off() time.sleep_ms(50) </pre>

Tekan tombol internal pada Board dan amati lampu LED merah.

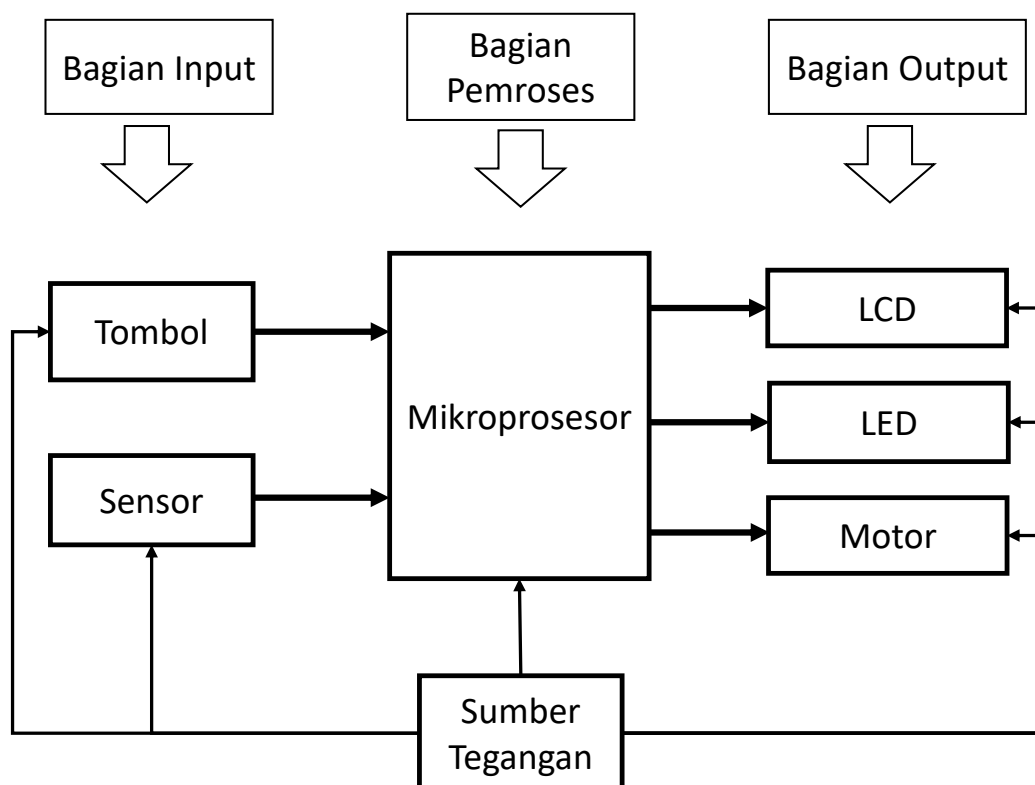
18. Tuliskan kode program di atas dan berikan analisis pada bagian kanan program.

19. Gambarkan Flowchart Program 1.12.

20. Gambarkan diagram blok sistem menyalakan lampu LED dengan tombol tersebut. Lihatlah contoh diagram blok sistem pada Lampiran.

Lampiran 1: Diagram Blok Sistem

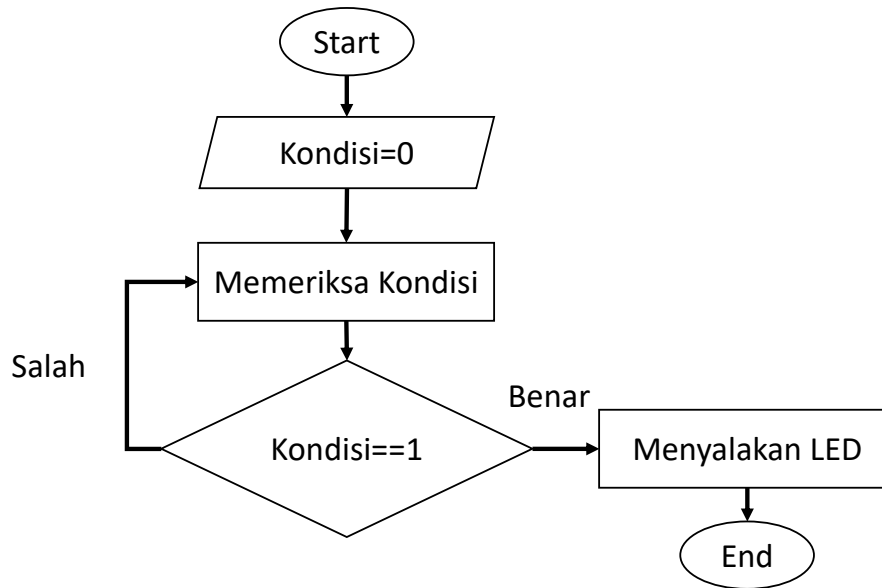
Contoh diagram blok ditunjukkan pada Gambar 1.7.



Gambar 1. 7. Contoh Diagram Blok Sistem

Lampiran 2: Diagram Alir/Flowchart

Contoh diagram alir ditunjukkan pada Gambar 1.8.



Gambar 1. 8. Contoh Diagram Alir

Lampiran 3: Laporan

Laporan dikerjakan dengan ditulis tangan menggunakan bolpoin tinta biru atau hijau. Laporan dikirimkan ke elearning dengan subjek Praktikum Sistem Embedded. Laporan tidak memerlukan halaman muka atau halaman cover. Cantumkan jenis praktikum, judul unit praktikum, nama dan NIM pada bagian atas dilanjutkan dengan tujuan praktikum, alat dan bahan praktikum dan jawaban Pre-Test dan Post-Test. Laporan dibuat 2 kolom dengan garis tepi sebesar 1cm pada bagian atas, bawah, kanan, dan kiri. Contoh laporan ditunjukkan pada ilustrasi berikut.

Contoh Laporan Praktikum Embedded

Praktikum Sistem Embedded Unit 1 Pengenalan Sistem Embedded Nama (NIM)	
<u>Tujuan Praktikum</u> 1. <u>Alat dan Bahan Praktikum</u> 1. Komputer/ Laptop <u>Jawaban Pre-Test</u> 1.	<u>Jawaban Post-Test</u> 1. 2. 3. 4. 5. 6. 7.

Praktikum Sistem Embedded Unit 2

Bahasa Pemrograman Python dan Input Output

oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 2 adalah

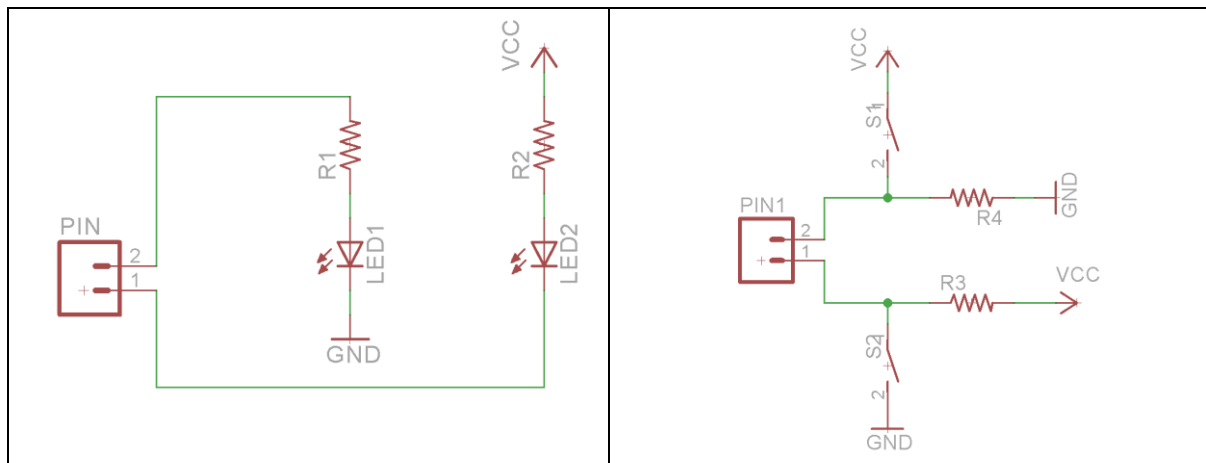
1. Mempelajari tentang Bahasa Pemrograman Python Bagian 2
2. Mempelajari tentang Pin Input dan Output pada MicroPython
3. Mempelajari tentang mengendalikan Servo pada MicroPython

Pin

Sebuah Pin adalah objek dasar untuk mengendalikan input atau output. Pin memiliki metode untuk mengatur mode Pin (input, output dan yang lainnya) dan metode untuk mendapatkan dan mengatur level logika digital (0 atau 1). Untuk dapat menggunakan sebuah Pin sebagai input atau output, dapat dilakukan dengan prosedur sebagai berikut.

1. Memilih pin sebagai input atau output dan mendeklarasikan sebagai input atau output
2. Memberikan logika awal 0 atau 1

Terdapat dua buah rangkaian konfigurasi Pin yaitu konfigurasi Pin aktif High dan Konfigurasi Pin Aktif Low. Rangkaian konfigurasi tersebut lebih jelas dapat dilihat pada Gambar 2.1.



Gambar 2.1. Konfigurasi Output dan Input Aktif High dan Low

Konfigurasi output aktif high adalah konfigurasi untuk menyalakan LED dengan cara memberi logika High atau 1 (VCC). Sementara Konfigurasi output aktif Low adalah konfigurasi untuk menyalakan LED dengan cara memberi logika Low atau 0 (GND).

Hal yang sama berlaku untuk konfigurasi pada input. Konfigurasi input aktif High berarti perangkat input akan memberikan sinyal berupa logika 1. Sementara konfigurasi input aktif Low berarti perangkat input memberikan sinyal berupa logika 0.

Library Pins

Library Pins ditunjukkan pada Program 2.2.

Program 2.1.	
<code>from machine import Pin</code>	<code># Memanggil fungsi Pin pada Library pyb</code>
<code>p_out = Pin('X1', Pin.OUT_PP)</code>	<code># Mendefinisikan bahwa Pin X1 adalah output</code>
<code>p_out.value(1)</code>	<code># Memberikan logika high atau 1</code>
<code>p_out.value(0)</code>	<code># Memberikan logika low atau 0</code>
<code>p_in = Pin('X2', Pin.IN, Pin.PULL_UP)</code>	<code># Mendefinisikan bahwa Pin X2 adalah input</code>
<code>p_in.value()</code>	<code># Mendapatkan nilai 0 atau 1</code>

Servo

Motor servo adalah jenis motor yang dapat digerakkan berdasarkan pada sudut tertentu atau berdasarkan pada kecepatan tertentu. Kelebihan motor servo adalah dapat diatur pergerakan berdasarkan pada posisi sudut dengan akurat. Motor Servo dikendalikan dengan menggunakan 3 buah Pin yaitu Pin ground, Pin power (vcc) dan Pin sinyal.

Library Math

Library Math adalah pustaka fungsi matematika yang didefinisikan dengan standar C. Contoh perintah pada library math ditunjukkan pada Program 2.1.

Program 2.2.	
<code>import math</code>	<code># Perintah untuk memanggil library math</code>
<code>print(math.pow(2,8))</code>	<code># Menampilkan hasil 2 pangkat 8</code>
<code>print(math.sqrt(4))</code>	<code># Menampilkan hasil akar kuadrat 5</code>
<code>print(math.log10(100))</code>	<code># Menampilkan hasil 10 log 100</code>
<code>print(math.degrees(math.pi/2))</code>	<code># Menampilkan konversi radian ke derajat</code>
<code>print(math.radians(90))</code>	<code># Menampilkan konversi derajat ke radian</code>
<code>print(math.sin(math.pi/2))</code>	<code># Menampilkan hasil sin pi/2 radian</code>
<code>print(math.cos(math.pi/2))</code>	<code># Menampilkan hasil cos pi/2 radian</code>
<code>print(math.asin(1))</code>	<code># Menampilkan hasil arcsin 1 dalam radian</code>
<code>print(math.acos(1))</code>	<code># Menampilkan hasil arccos 1 dalam radian</code>

Library Servo

Library Servo ditunjukkan pada Program 2.3.

Program 2.3.

```
from pyb import Servo # memanggil fungsi Servo pada library pyb

s1 = Servo(1)          # Menyimpan motor Servo 1 pada variabel s1 (X1, VIN, GND)
s1.angle(45)           # Menggerakkan Servo ke Sudut 45
s1.angle(-60, 1500)    # Menggerakkan Servo ke Sudut -60 selama 1500ms
```

Pre Test

Bacalah dasar teori pada sub bab sebelumnya lalu kerjakan soal Pre Test berikut dan tuliskan jawaban pada Laporan.

1. Jelaskan tentang konfigurasi output aktif High dan output aktif Low beserta dengan gambar rangkaiannya.
2. Apa kelebihan Motor Servo? Secara umum apa saja Pin pada Motor Servo?

Langkah Praktikum dan Post Test

Silahkan buka website MicroPython Berikut.

<http://micropython.org/unicorn/>

Pemrograman Bahasa Python Library Math

Ketikkan perintah program 2.4 pada editor kemudian jalankan. Amati hasilnya pada Terminal dan tuliskan hasilnya pada laporan.

Program 2.4.

<pre>import math print(math.pow(2,10)) print(math.sqrt(100)) print(math.log10(1000)) print(math.pi) print(math.degrees(math.pi/4)) print(math.radians(45)) print(math.sin(math.pi/4)) print(math.cos(math.pi/4)) print(math.acos(1)) print(math.asin(1))</pre>

1. Apa fungsi perintah `math.pow(2,10)`?
2. Apa fungsi perintah `math.sqrt(100)`?
3. Apa fungsi perintah `math.log10(1000)`?
4. Apa fungsi perintah `math.pi`? Dengan menggunakan program hitunglah nilai $\pi/2$, $\pi/3$, $\pi/4$ dan $\pi/6$ lalu tuliskan dan lengkapi Tabel 2.1 pada laporan.

5. Apa fungsi perintah `math.degrees(math.pi/4)`? Dengan perintah yang sama tuliskan nilai konversi π , $\pi/2$, $\pi/3$, $\pi/6$ radian ke derajat. Tuliskan hasil pada Tabel 2.1 pada kolom nilai derajat

Tabel 2.1. Nilai Konversi Radian ke Derajat

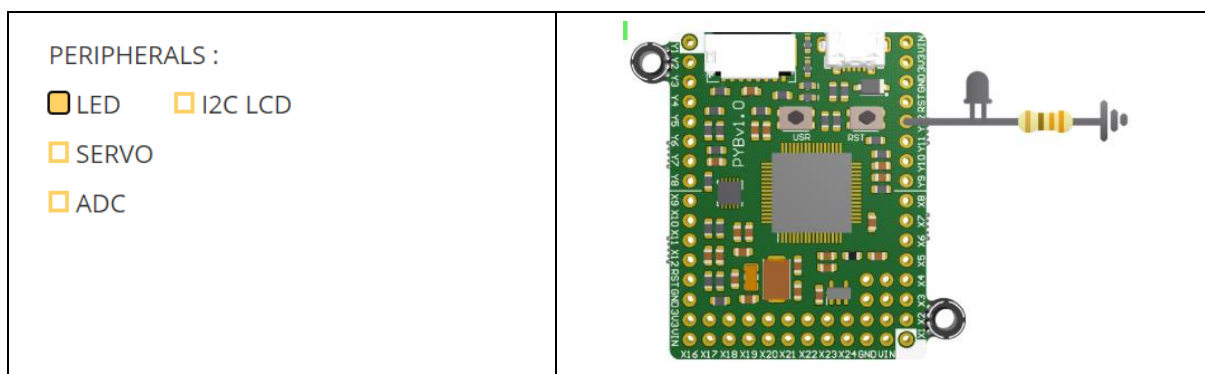
Nilai Desimal	Nilai Radian	Nilai Derajat
π	π	
$\pi/2$	$\pi/2$	
$\pi/3$	$\pi/3$	
$\pi/4$	$\pi/4$	
$\pi/6$	$\pi/6$	

6. Apa fungsi perintah `math.radians(45)`? Hitunglah nilai radian dari 180, 90, 60, 30 derajat. Bandingkan dengan dengan nilai decimal pada Tabel 2.1. Berikan kesimpulan dari pengamatan.
7. Apa fungsi perintah `math.asin(1)` dan `math.acos(1)`? Berapa nilai sudutnya dalam derajat?

Menyalakan Lampu LED

Buatlah konfigurasi sesuai Gambar 2.1. Pilih pilihan LED pada bagian Peripherals. Ketikkan Program 2.3 untuk menyalakan lampu LED kemudian amati apakah lampu LED menyala. Untuk mematikan lampu LED ubah perintah pada bagian `p12.value(0)`.

Program 2.5.
<pre>from machine import Pin p12 = Pin('Y12', Pin.OUT) p12.value(1)</pre>



Gambar 2.2. Konfigurasi Output LED

8. Mengapa logika 1 pada `p12.value(1)` dapat menyalakan lampu LED dan logika 0 pada `p12.value(0)` dapat mematikan lampu LED?

9. Gambarkan rangkaian pada Gambar 2.2 pada laporan kemudian berikan penjelasan mengapa logika 1 dapat menyalakan lampu LED.
10. Apakah konfigurasi pada Gambar 2.2 merupakan konfigurasi aktif low atau aktif High? Berikan alasan dan penjelasan.

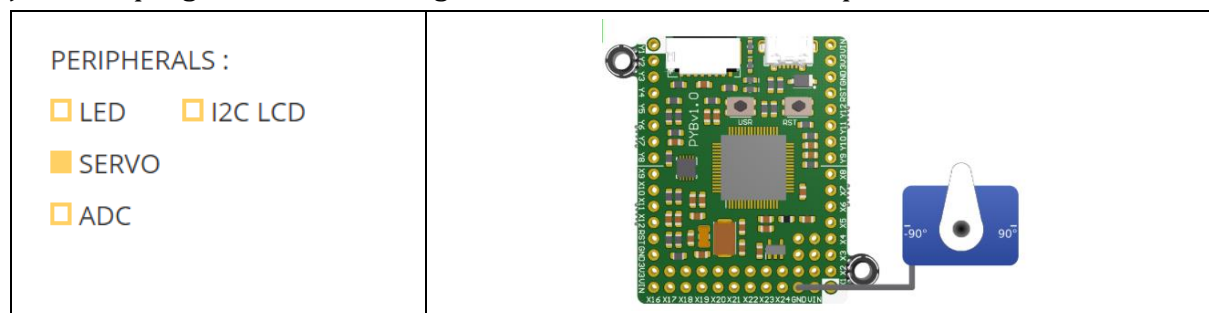
Ketikkan Program 2.6 untuk menyalakan lampu dengan waktu tunda sehingga lampu dapat berkedip dengan waktu tunda 100 milisekon.

Program 2.6.
<pre> from machine import Pin import time p12 = Pin('Y12', Pin.OUT) while True: p12.value(1) time.sleep_ms(100) p12.value(0) time.sleep_ms(100) </pre>

11. Modifikasilah Program 2.6 untuk membuat lampu LED berkedip dengan waktu tunda 1 detik (Buka kembali materi unit 1 untuk melihat library time).

Menggerakkan Motor Servo

Buatlah konfigurasi sesuai Gambar 2.3 lalu ketikkan Program 2.6 pada Editor, kemudian jalankan program tersebut dengan menekan tombol Run Script.



Gambar 2.3. Konfigurasi Motor Servo

Program 2.7.
<pre> import machine import pyb servo = pyb.Servo(1) servo.angle(90, 1000) </pre>

12. Gambarkan posisi sudut awal sebelum dan setelah program 2.6 dijalankan. Untuk mengembalikan motor servo ke posisi awal ketikkan perintah berikut pada terminal
`>>> servo.angle(0, 1000)`
13. Ketikkan perintah berikut pada Terminal kemudian gambarkan posisi sudut motor servo.
`>>> servo.angle(30, 1000)`
`>>> servo.angle(60, 1000)`
`>>> servo.angle(-60, 1000)`
14. Ketikkan perintah berikut pada terminal lalu amati perbedaan antara kedua perintah berikut.
Perintah 1:
`>>> servo.angle(0, 5000)`
`>>> servo.angle(90, 5000)`
Perintah 2:
`>>> servo.angle(0, 1000)`
`>>> servo.angle(90, 1000)`
Apa yang menyebabkan kedua perintah tersebut memberikan hasil yang berbeda pada perpindahan sudut motor servo?

Praktikum Sistem Embedded Unit 3
Analog to Digital Converter (ADC)
oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 3 adalah

1. Mempelajari tentang Analog to Digital Converter (ADC)

Analog to Digital Converter

Analog to Digital Converter atau yang biasa disebut dengan ADC adalah salah satu fitur pada mikrokontroler yang berfungsi untuk mengkonversi data analog menjadi data digital. Data analog yang dimaksud adalah data tegangan atau arus, kebanyakan adalah data tegangan. Sebuah mikrokontroler tidak bisa memproses data analog. Data analog tersebut harus diubah menjadi data digital sehingga mikrokontroler dapat memprosesnya. Jadi hanya data digital yang dapat diproses oleh mikrokontroler. Proses yang dimaksud adalah dapat dilakukan suatu operasi pemrograman, salah satunya operasi matematika seperti tambah, bagi, kurang, kali dan sebagainya.

Ukuran Bit ADC

Terdapat beberapa ukuran bit dalam ADC diantaranya 8 bit, 10 bit, 12 bit, 16 bit. Ukuran bit ADC berhubungan dengan ukuran data ADC. Ukuran data ADC dapat dihitung dengan menggunakan persamaan sebagai

$$2^n \quad (1)$$

Dengan n adalah ukuran bit. Semakin besar ukuran bit ADC, semakin besar ukuran datanya. Ukuran bit ADC dan jangkauan datanya dapat dilihat pada Tabel 3.1.

Tabel 3. 1. Ukuran Bit dan Jangkauan Datanya

Ukuran Bit	Jangkauan Data	Nilai Awal	Nilai Akhir
8 bit	$2^8 = 256$	0	255
10 bit	$2^{10} = 1024$	0	1023
12 bit	$2^{12} = 4096$	0	4095
16 bit	$2^{16} = 65536$	0	65535

Proses Konversi ADC

Proses konversi data analog menjadi data digital melibatkan beberapa variabel yaitu nilai tegangan input, tegangan referensi dan ukuran bit. Persamaan untuk menghitung nilai konversi ADC dapat dituliskan sebagai

$$Data\ ADC = \frac{Tegangan\ Input}{Tegangan\ Referensi} * Ukuran\ bit \quad (2)$$

Contoh: Diketahui sebuah mikroprosesor memiliki ukuran bit ADC 8 bit, dengan tegangan referensi 5 volt. Hitunglah berapa nilai konversi ADC nya jika nilai tegangan input adalah 2 volt.

Jawab: Nilai konversi ADC dapat dihitung dengan Persamaan (2) sebagai

$$\begin{aligned} \text{Data ADC} &= \frac{\text{Tegangan Input}}{\text{Tegangan Referensi}} * \text{Ukuran bit} \\ &= \frac{2}{5} * 255 = 102 \end{aligned}$$

Jadi nilai data ADC untuk tegangan input 2 volt adalah 102.

Mengkonversi data ADC ke Tegangan Asal

Pada contoh soal sebelumnya diketahui bahwa nilai ADC untuk tegangan 2 volt adalah 102. Lalu bagaimana caranya agar mikrokontroler dapat mengetahui bahwa nilai asal ADC tersebut adalah 2 volt. Proses konversi ke tegangan asli (asal) dapat diketahui dengan menggunakan Persamaan (2) sebagai

$$\text{Data ADC} = \frac{\text{Tegangan Input}}{\text{Tegangan Referensi}} * \text{Ukuran bit} \quad (3)$$

Dengan menggunakan pindah silang untuk mencari nilai tegangan input pada Persamaan (3) dapat diperoleh

$$\text{Tegangan Input} = \frac{\text{Data ADC}}{\text{Ukuran bit}} * \text{Tegangan Referensi} \quad (4)$$

Oleh karena itu dapat diperoleh tegangan asli nilai ADC 102 adalah

$$\begin{aligned} \text{Tegangan Input} &= \frac{\text{Data ADC}}{\text{Ukuran bit}} * \text{Tegangan Referensi} \\ &= \frac{102}{255} * 5 = 2\text{volt} \end{aligned}$$

Library ADC

Library ADC ditunjukkan pada Program 3.1.

Program 3. 1.	
import machine	# Memasukkan library/ Pustaka machine untuk Pin
import pyb	# Memasukkan library/ Pustaka pyb untuk ADC
adc = ADC(Pin('X19'))	# Menyimpan pembacaan ADC Pin X19 pada variabel adc
adc.read()	# Membaca nilai ADC

Pre Test

Bacalah dasar teori pada sub bab sebelumnya lalu kerjakan soal Pre Test berikut dan tuliskan jawaban pada Laporan.

1. Apa itu fitur ADC pada mikrokontroller?
2. Diketahui suatu mikroprosesor memiliki ukuran bit ADC sebesar 4 bit, 6bit dan 14 bit. Berapa jangkauan data ADC pada mikroprosesor tersebut?
3. Diketahui suatu mikroprosesor memiliki ukuran bit ADC sebesar 10bit dan tegangan referensi ADC adalah 5 volt. Hitunglah nilai ADC untuk tegangan input 3 volt dan 4 volt.
4. Diketahui suatu mikroprosesor memiliki ukuran bit ADC sebesar 10bit dan tegangan referensi ADC adalah 5 volt. Hitunglah berapa nilai tegangan input jika diketahui nilai ADC adalah 255 dan 750.

Langkah Praktikum dan Post Test

Silahkan buka website MicroPython Berikut.

<http://micropython.org/unicorn/>

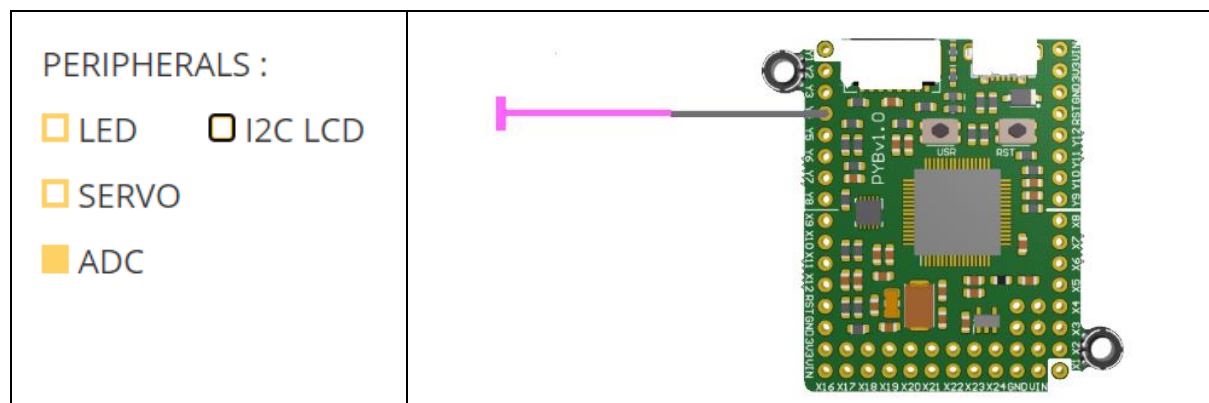
Membaca ADC

Buatlah konfigurasi sesuai Gambar 3.1. Pilih pilihan ADC pada bagian Peripherals. Ketikkan Program 3.2 untuk membaca data ADC pada resistor geser (slider). Geser slider tersebut ke kanan lalu tekan tombol Run Script untuk mengetahui nilai ADC yang tertampil pada terminal.

Program 3. 2.

```
import machine
import pyb

y4 = machine.Pin('Y4')
adc = pyb.ADC(y4)
print(adc.read())
```

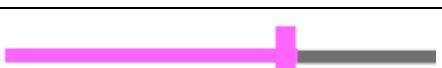


Gambar 3. 1. Konfigurasi Penggunaan Resistor Geser untuk ADC

1. Tuliskan Program 3.2, lalu berikan arti perintah pada setiap baris tersebut.

2. Geser slider pada posisi tengah, berapa nilai ADC yang tertampil pada terminal?
3. Geser slider pada posisi ujung kanan dan ujung kiri. Berapa nilai ADC pada masing-masing ujung slider tersebut? Berdasarkan data tersebut, berapa bit ukuran ADC pada mikroprosesor tersebut?
4. Masukkan Program 3.3 pada editor, lalu geser slider sesuai dengan gambar pada kolom pertama Tabel 3.2. Tekan tombol Run Script dan amati hasilnya pada Terminal.
 Pada Tabel 3.2, Lengkapi berapa nilai data ADC yang ditampilkan pada terminal. Lengkapi juga Nilai tegangan dengan perhitungan menggunakan program pada terminal dan nilai tegangan perhitungan manual, jika diketahui nilai tegangan referensi adalah 3 volt dan ukuran bit ADC adalah 8 bit. Tuliskan Tabel pada laporan.

Tabel 3. 2.

Posisi Slider	Nilai ADC	Nilai Tegangan perhitungan program	Nilai Tegangan perhitungan manual
			
			
			
			
			
			

Program 3. 3.

```

import machine, pyb

y4 = machine.Pin('Y4')
adc = pyb.ADC(y4)

tegangan_referensi=3
ukuran_bit=255

print("Nilai ADC=", adc.read())
print("Nilai Tegangan=", adc.read()*tegangan_referensi/ukuran_bit, "volt")

```

5. Pada Program 3.3 pada baris ke 5 dan 6, berasal dari mana nilai 3 dan 255 tersebut?
6. Apakah hasil antara perhitungan tegangan menggunakan program dan perhitungan manual memberikan hasil yang sama? Jika terdapat perbedaan, berikan alasan mengapa terdapat perbedaan nilai tersebut.
7. Lakukan modifikasi pada Program 3.3, untuk nilai tegangan referensi bernilai 5 volt. Tuliskan program tersebut pada laporan lalu lengkapi Tabel 3.3. Tuliskan tabel tersebut pada laporan.

Tabel 3. 3.

Posisi Slider	Nilai ADC	Nilai Tegangan perhitungan program	Nilai Tegangan perhitungan manual
			
			
			
			
			
			

8. Buatlah diagram blok dan flowchart/ diagram alir Program 3.3 dan Program pada nomor 7.

Praktikum Sistem Embedded Unit 4 I2C LCD

oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 4 adalah

1. Mempelajari tentang komunikasi I2C dan penerapan pada LCD I2C.

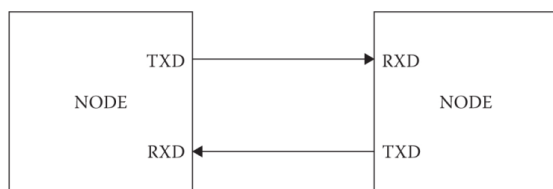
Protokol Komunikasi Sistem Embedded

Pada sistem embedded terdapat beberapa protokol komunikasi di antaranya adalah Komunikasi Serial/ UART, SPI dan I2C. Perbedaan antara ketiga protokol komunikasi tersebut ditunjukkan pada Gambar 4.1, 4.2 dan 4.3. Terdapat perbedaan pada jenis jumlah jalur komunikasi dan metode komunikasinya.

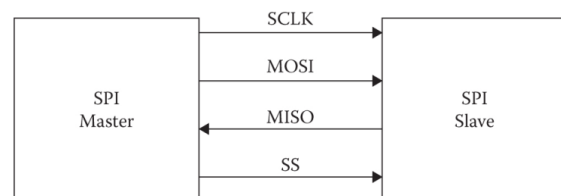
Inter-Integrated Circuit (I2C)

Pada bagian ini akan mempelajari tentang Protokol Komunikasi I2C. Inter-Integrated Circuit atau disingkat sebagai I2C memiliki terjemahan sirkuit antar-terintegrasi. Protokol Komunikasi I2C adalah sebuah protokol dua jalur untuk berkomunikasi antar perangkat. Pada tingkat hardware terdapat dua buah jalur komunikasi I2C yaitu SCL dan SDA. Jalur SCL untuk jalur Clock dan jalur SDA untuk jalur data. Contoh komunikasi I2C adalah komunikasi antara dua buah mikroprosesor, komunikasi antara mikroprosesor dan sensor dan komunikasi antara mikroprosesor dan LCD.

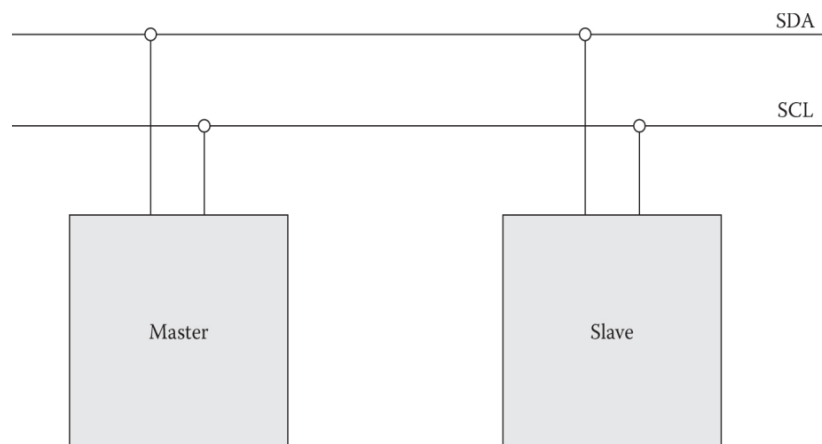
Pada mikroprosesor terdapat Pin tertentu yang memiliki fungsi sebagai I2C. Namun protokol ini juga dapat menggunakan Pin biasa yang difungsikan sebagai I2C. Diagram blok komunikasi I2C ditunjukkan pada Gambar 4.3.



Gambar 4. 1. Komunikai UART



Gambar 4. 2. Komunikasi SPI



Gambar 4. 3. Komunikasi I2C

Kelebihan komunikasi I2C adalah hanya dengan dua buah jalur komunikasi yaitu SDA dan SCL, sebuah mikroprosesor dapat terhubung dengan beberapa perangkat sekaligus. Contohnya sebuah mikroprosesor dapat terhubung dengan LCD, dan beberapa sensor sekaligus dengan hanya menggunakan komunikasi I2C. Oleh karena itu dengan menggunakan komunikasi I2C dapat menghemat Pin yang digunakan.

Prinsip kerja dari komunikasi i2c adalah mengakses alamat perangkat, mengirimkan perintah dan menerima data dari perangkat. Setiap perangkat dalam jalur komunikasi I2C memiliki alamat masing-masing. Beberapa perangkat sekaligus dapat terhubung dalam komunikasi I2C karena memiliki alamat yang berbeda. Setelah alamat perangkat dikirimkan, kemudian perintah untuk mengakses data dikirimkan ke alamat tersebut. Perangkat yang dituju kemudian mengirimkan data ke master jika perintahnya benar.

Library LCD I2C

Library I2C Ditunjukkan pada Program 4.1.

Program 4. 1.	
<code>import machine</code>	<code>#Memasukkan library machine</code>
<code>import framebuf</code>	<code>#Memasukkan library framebuf</code>
<code>scl = machine.Pin('X9')</code>	<code>#Mendefinisikan SCL sebagai PIN X9</code>
<code>sda = machine.Pin('X10')</code>	<code>#Mendefinisikan SDA sebagai PIN X10</code>
<code>i2c = machine.I2C(scl=scl, sda=sda)</code>	<code>#Mendefinisikan i2c sebagai machine.i2c</code>
<code>fbuf = framebuf.FrameBuffer(bytearray(64 * 32 // 8), 64, 32, framebuf.MONO_HLSB)</code> <code># Mendefinisikan fbuf sebagai buffer array LCD dengan ukuran resolusi 64x32</code>	
<code>fbuf.pixel(x,y,c)</code>	<code># Perintah membuat titik piksel, x adalah letak koordinat x</code> <code># y adalah letak koordinat y</code> <code># c dapat bernilai 1 berarti muncul titik dan 0 berarti titik hilang</code>
<code>fbuf.rect(x,y,w,h,c)</code>	<code># Perintah untuk membuat persegi, h=height/ tinggi</code> <code># w=width/ lebar, x=koordinat x, y=koordinat y</code>
<code>fbuf.text(s,x,y,c)</code>	<code># Perintah untuk membuat teks, s=teks</code>
<code>fbuf.hline(x,y,w,c)</code>	<code># Perintah untuk membuat garis horisontal</code>
<code>fbuf.vline(x,y,h,c)</code>	<code># Perintah untuk membuat garis vertikal</code>
<code>fbuf.line(x1,y1,x2,y2,c)</code>	<code># Perintah untuk garis</code>
<code>i2c.writeto(8, fbuf)</code>	<code># Perintah untuk mengirim variabel fbuf sebesar 8bit melalui</code> <code># jalur komunikasi I2C</code>

Pre Test

Bacalah dasar teori pada sub bab sebelumnya lalu kerjakan soal Pre Test berikut dan tuliskan jawaban pada Laporan.

1. Apa itu protokol Komunikasi I2C?
2. Apa itu PIN SDA dan SCL pada I2C?
3. Bagaimana cara kerja komunikasi I2C?

Langkah Praktikum dan Post Test

Silahkan buka website MicroPython Berikut.

<http://micropython.org/unicorn/>

Menampilkan Piksel pada LCD I2C

Buatlah konfigurasi sesuai Gambar 4.4. Pilih pilihan I2C LCD pada bagian Peripherals. Ketikkan Program 4.2 untuk menampilkan piksel pada LCD. Lalu tekan tombol Run Script untuk menjalankan Program dan melihat tampilan pada I2C LCD.

Program 4. 2.
<pre>import machine, framebuf scl = machine.Pin('X9') sda = machine.Pin('X10') i2c = machine.I2C(scl=scl, sda=sda) fbuf = framebuf.FrameBuffer(bytearray(64 * 32 // 8), 64, 32, framebuf.MONO_HLSB) fbuf.pixel(0,0,1) i2c.writeto(8, fbuf)</pre>



Gambar 4. 4.

1. Gambarkan letak titik piksel Program 4.1.
2. Gambarkan diagram blok menampilkan Piksel pada LCD tersebut.

3. Modifikasi bagian `fbuf.pixel(0,0,1)` sesuai Tabel 4.1. Gambarkan letak piksel pada kolom kedua.

Tabel 4. 1.

Perintah Membuat Titik Piksel	Letak Piksel pada LCD
<code>fbuf.pixel(63,31,1)</code>	
<code>fbuf.pixel(63,0,1)</code>	
<code>fbuf.pixel(0,31,1)</code>	
<code>fbuf.pixel(32,16,1)</code>	

4. Bagian mana dari perintah `fbuf.pixel(63,0,1)` yang merupakan koordinat sumbu x dan koordinat sumbu y?

Menampilkan Persegi pada LCD I2C

Ketikan Program 4.3 untuk menampilkan persegi pada LCD. Lalu tekan tombol Run Script untuk menjalankan Program dan melihat tampilan pada I2C LCD.

Program 4. 3.
<pre>import machine import framebuf scl = machine.Pin('X9') sda = machine.Pin('X10') i2c = machine.I2C(scl=scl, sda=sda) fbuf = framebuf.FrameBuffer(bytearray(64 * 32 // 8), 64, 32, framebuf.MONO_HLSB) fbuf.rect(0,0,5,5,1) i2c.writeto(8, fbuf)</pre>

5. Gambarkan letak persegi Program 4.3.
6. Modifikasi Program 4.3 sesuai Tabel 4.3, lalu lengkapi tabel tersebut.

Tabel 4. 2.

Perintah Menggambar Persegi	Gambar Persegi pada LCD
<code>fbuf.rect(0,0,10,10,1)</code>	
<code>fbuf.rect(10,10,15,15,1)</code>	

7. Bagian mana dari perintah `fbuf.rect(0,0,5,5,1)` yang merupakan koordinat x, koordinat y, panjang persegi dan lebar persegi?

Menampilkan Teks pada LCD I2C

Ketikan Program 4.4 untuk menampilkan teks pada LCD. Lalu tekan tombol Run Script untuk menjalankan Program dan melihat tampilan pada I2C LCD.

Program 4. 4.
<pre>import machine, framebuf scl = machine.Pin('X9') sda = machine.Pin('X10') i2c = machine.I2C(scl=scl, sda=sda) fbuf = framebuf.FrameBuffer(bytearray(64 * 32 // 8), 64, 32, framebuf.MONO_HLSB) fbuf.text('Hello',0,0,1) fbuf.text('Elektro',0,9,1) fbuf.text('UAD',0,18,1) i2c.writeto(8, fbuf)</pre>

8. Gambarkan hasil Program 4.4, lalu tuliskan program 4.4 dan berikan analisis setiap baris.

Menampilkan Garis pada LCD I2C

Ketikan Program 4.5 untuk menampilkan teks pada LCD. Lalu tekan tombol Run Script untuk menjalankan Program dan melihat tampilan pada I2C LCD.

Program 4. 5.
<pre>import machine, framebuf scl = machine.Pin('X9') sda = machine.Pin('X10') i2c = machine.I2C(scl=scl, sda=sda) fbuf = framebuf.FrameBuffer(bytearray(64 * 32 // 8), 64, 32, framebuf.MONO_HLSB) fbuf.vline(5,9,12,1) fbuf.text('Elektro',7,10,1) fbuf.hline(6,20,23,1) fbuf.line(15,5,45,7,1) i2c.writeto(8, fbuf)</pre>

9. Gambarkan hasil Program 4.5, lalu tuliskan program tersebut dan berikan analisis setiap baris.

Praktikum Sistem Embedded Unit 5
Mini Projek
oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 5 adalah

1. Mempelajari tentang pemrograman micropython untuk mini projek atau projek sederhana dengan kombinasi dengan beberapa perangkat dan kombinasi beberapa perintah pemrograman.

Langkah Praktikum dan Post Test

Silahkan buka website MicroPython berikut <http://micropython.org/unicorn/>

Buka kembali materi pada unit-unit sebelumnya untuk membantu dalam menjawab pertanyaan-pertanyaan pada Post Test berikut. Rangkuman beberapa program adalah sebagai berikut.

Program Menyalakan lampu LED dengan Tombol

```
import time, pyb

while True:
    if pyb.Switch().value():
        pyb.LED(1).on()
        time.sleep_ms(50)
    else:
        pyb.LED(1).off()
        time.sleep_ms(50)
```

Program Membaca data ADC dan Menampilkan pada Terminal

```
import machine, pyb

y4 = machine.Pin('Y4')
adc = pyb.ADC(y4)
print(adc.read())
```

Program Menggerakkan Motor Servo

```
import machine, pyb

servo = pyb.Servo(1)
servo.angle(90, 1000)
```

Program Menampilkan Teks pada LCD I2C

```
import machine, framebuf

scl = machine.Pin('X9')
sda = machine.Pin('X10')
i2c = machine.I2C(scl=scl, sda=sda)

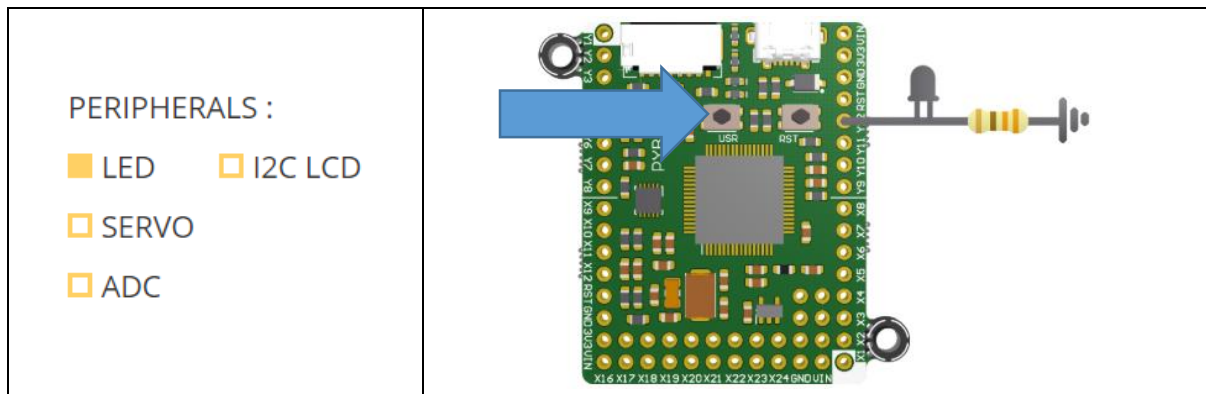
fbuf = framebuf.FrameBuffer(bytearray(64 * 32 // 8), 64, 32, framebuf.MONO_HLSB)

fbuf.text('UAD',0,18,1)
i2c.writeto(8, fbuf)
```

1. Menyalakan Lampu LED dengan Tombol

Buatlah konfigurasi seperti pada Gambar 5.1, lalu ketikkan Program 5.1 pada editor. Tekan tombol Run Script untuk menjalankan Program. Tekan tombol USR pada mikrokontroler seperti ditunjukkan tanda panah pada Gambar 5.1 untuk menyalakan semua LED.

Program 5. 1.
<pre>import time, pyb from machine import Pin p12 = Pin('Y12', Pin.OUT) while True: if pyb.Switch().value(): pyb.LED(1).on() pyb.LED(2).on() pyb.LED(3).on() pyb.LED(4).on() p12.value(1) time.sleep_ms(100) else: pyb.LED(1).off() pyb.LED(2).off() pyb.LED(3).off() pyb.LED(4).off() p12.value(0) time.sleep_ms(100)</pre>



Gambar 5. 1. Konfigurasi Menyalakan LED dengan Tombol

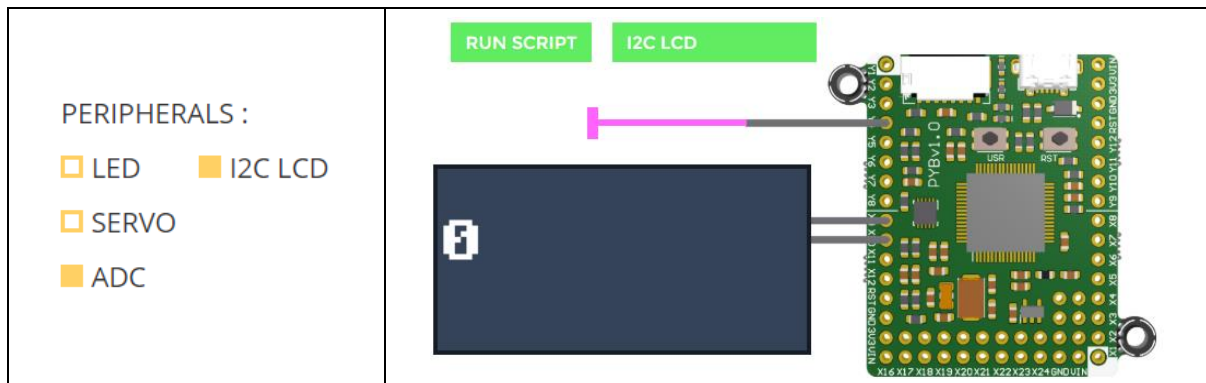
- 1.1. Perintah mana yang berfungsi untuk menyalakan dan mematikan semua lampu LED?
- 1.2. Perintah mana yang berfungsi untuk mendeteksi tombol ditekan untuk menyalakan lampu LED?
- 1.3. Jika tombol tidak ditekan, perintah mana yang bekerja?
- 1.4. Apa fungsi perintah `p12 = Pin('Y12', Pin.OUT)`?
- 1.5. Gambarkan diagram blok sistem Gambar 5.1. dan diagram alir Program 5.1.

2. Menampilkan data ADC pada LCD I2C

Buatlah konfigurasi sesuai Gambar 5.2, lalu ketikkan Program 5.2 pada editor. Tekan tombol Run Script untuk menjalankan program. Geser slider ke kiri dan ke kanan kemudian perhatikan nilai yang ditampilkan pada terminal dan LCD 12C.

Program 5. 2.
<pre> import machine, pyb, framebuf y4 = machine.Pin('Y4') scl = machine.Pin('X9') sda = machine.Pin('X10') i2c = machine.I2C(scl=scl, sda=sda) fbuf = framebuf.FrameBuffer(bytearray(64 * 32 // 8), 64, 32, framebuf.MONO_HLSB) while True: adc = pyb.ADC(y4) dataadc=adc.read() print(dataadc) fbuf.fill(0) fbuf.text(str(dataadc),0,9,1) i2c.writeto(8, fbuf) </pre>

*Perintah `fbuf.fill(0)` berfungsi untuk menghapus semua gambar pada LCD. Perintah `str()` berfungsi untuk mengonversi data integer ke data teks.



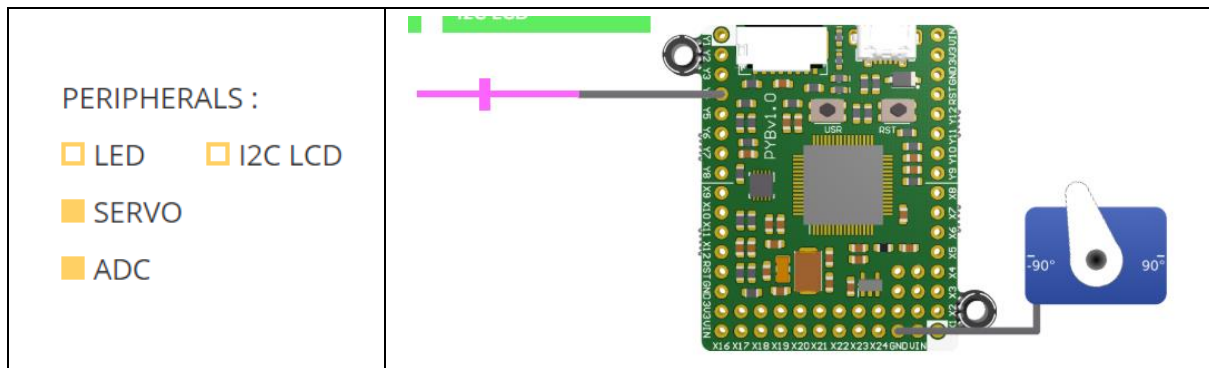
Gambar 5. 2. Konfigurasi Menyalakan Menampilkan data ADC pada LCD I2C

- 2.1. Perintah mana yang berfungsi untuk membaca data ADC dari slider?
- 2.2. Variabel apa yang berfungsi untuk menyimpan data ADC?
- 2.3. Perintah mana yang berfungsi untuk menampilkan data ADC pada terminal dan pada LCD I2C?
- 2.4. Gambarkan diagram blok sistem Gambar 5.2. dan diagram alir Program 5.2.

3. Menggerakkan Motor Servo dengan Slider ADC

Buatlah konfigurasi sesuai Gambar 5.3, lalu ketikkan Program 5.3 pada editor. Tekan tombol Run Script untuk menjalankan program. Geser slider ke kiri dan ke kanan kemudian perhatikan gerakan sudut motor servo.

Program 5. 3.
<pre> import machine import pyb y4 = machine.Pin('Y4') adc = pyb.ADC(y4) servo = pyb.Servo(1) while True: dataadc=int(((adc.read()-127)*100)/127) sudut=int(dataadc*90/100) servo.angle(sudut,100) </pre>



Gambar 5. 3. Konfigurasi Menggerakan Motor Servo dengan Slider

- 3.1. Perintah mana pada Program 5.3 yang berfungsi untuk menggerakan motor servo?
- 3.2. Lakukan modifikasi pada Program 5.3 untuk menampilkan nilai sudut pada terminal. Perintah apa yang digunakan?
- 3.3. Gambarkan diagram blok sistem pada Gambar 5.3. Perangkat apa pada sistem Gambar 5.3, yang berfungsi sebagai input, pemroses dan output?

Buka halaman berikutnya untuk melihat percobaan 4. Berikutnya >>

4. Menggerakkan Motor Servo dengan Slider ADC dan menampilkan sudut pada LCD I2C disertai dengan indikator LED

Buatlah konfigurasi sesuai Gambar 5.4, lalu ketikkan Program 5.4 pada editor. Tekan tombol Run Script untuk menjalankan program. Geser slider ke kiri dan ke kanan kemudian perhatikan nilai sudut motor servo pada LCD I2C. Perhatikan juga kondisi lampu LED. Pindahkan slider ke kiri dan ke kanan, ketika sudut kurang dari nol, lampu LED akan mati, sebaliknya ketika sudut lebih dari nol, lampu LED akan menyala.

Program 5. 4.

```
import machine
import pyb
import framebuf

from machine import Pin

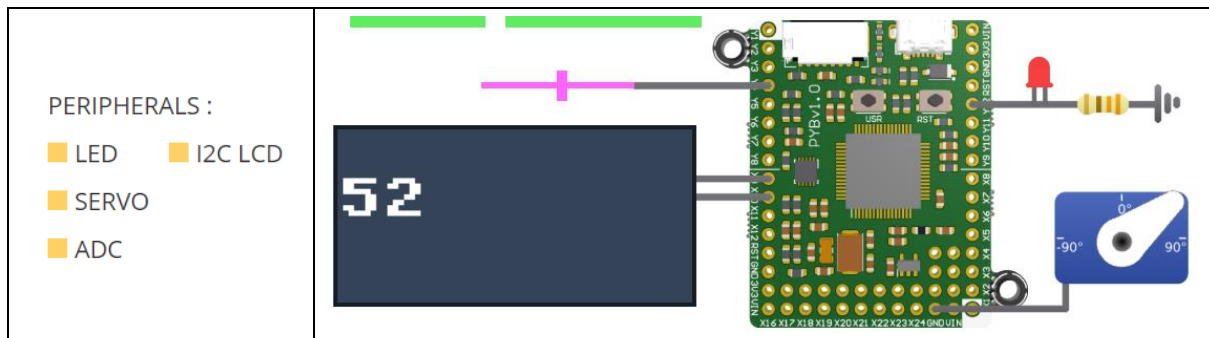
p12 = Pin('Y12', Pin.OUT)

y4 = machine.Pin('Y4')
adc = pyb.ADC(y4)
servo = pyb.Servo(1)

scl = machine.Pin('X9')
sda = machine.Pin('X10')
i2c = machine.I2C(scl=scl, sda=sda)

fbuf = framebuf.FrameBuffer(bytearray(64 * 32 // 8), 64, 32, framebuf.MONO_HLSB)

while True:
    dataadc=int(((adc.read()-127)*100)/127)
    sudut=int(dataadc*90/100)
    if sudut > 0:
        p12.value(1)
    else:
        p12.value(0)
    fbuf.fill(0)
    fbuf.text(str(sudut),0,9,1)
    i2c.writeto(8, fbuf)
    servo.angle(sudut,100)
```



Gambar 5. 4. Konfigurasi Menggerakkan Motor Servo dengan Slider dan menampilkan nilai sudut pada LCD I2C disertai indikator LED

- 4.1. Buatlah diagram blok sistem pada Gambar 5.4 dan diagram alir dari program 5.4. Ceritakan cara kerja sistem berdasarkan pada diagram blok dan diagram alir tersebut.

Daftar Pustaka

<https://micropython.org/>

<https://micropython.org/unicorn/>

<http://docs.micropython.org/en/latest/>

<http://docs.micropython.org/en/latest/pyboard/quickref.html>