UAD

MODUL PRAKTIKUM SISTEM EMBEDDED Laboratorium Robotika

Ir. Alfian Ma'arif, S.T., M.Eng Ir. Arsyad Cahya , S.T., M.T. Muchammad Ardan R., S.T.



PETUNJUK PRAKTIKUM SISTEM EMBEDDED DENGAN MICROPYTHON



Disusun Oleh :

Ir. Alfian Ma'arif, S. T., M. Eng. Ir. Arsyad Cahya Subrata, S. T., M. T.

Muchammad Ardan Ramadhani, S.T.

LABORATORIUM ROBOTIKA PROGRAM STUDI TEKNIK ELEKTRO FAKULTAS TEKNOLOGI INDUSTRI UNIVERSITAS AHMAD DAHLAN 2025

KATA PENGANTAR

Segala puji dan syukur mari kita panjatkan kepada Allah SWT atas segala limpahan rahmat, inayah, taufik dan hidayah-Nya sehingga sehingga Modul Praktikum Sistem Embeded ini dapat diselesaikan. Semoga modul ini dapat digunakan sebagai pedoman menjalankan mata kuliah Praktikum Sistem Embeded sebagai matakuliah wajib mahasiswa Program Studi Teknik Elektro, Fakultas Teknologi Industri, Universitas Ahmad Dahlan.

Sistem Embeded merupakan salah satu kemampuan/kompetensi dasar yang harus dikuasai di bidang Teknik Elektro. Kompetensi dapat dicapai oleh mahasiswa melalui mata kuliah Sistem Embeded (dua SKS) dan Praktikum Sistem Embeded (satu SKS). Materi praktikum disusun seacara sejalan dengan kuliah agar praktikan mampu memiliki gambaran yang jelas tentang aplikasi nyata.

Praktikum Sistem Embeded terdiri dari sepuluh modul, yaitu : (1) Pengenalan Pemrograman Python dan MicroPython; (2) Pemrograman MicroPython dan Input Output; (3) Pemrograman Input Output Micropython dengan ESP32; (4) Analog to Digital Converter (ADC); (5) I2C LCD; (6) Mini Projek; (7) Menampilkan huruf dan Angka Pada LCD 2x16; (8) Lampu Lalu-Lintas; (9) Sensor Ultrasonik; (10) Motor Servo. Hal ini diharapkan mampu memberikan gambaran kepada praktikan tentang Sistem Embeded dalam penggunaan dan penerapannya.

Kepada semua pihak yang telah membantu penyusunan modul praktikum ini dan membantu uji coba peralatan praktikum, kami ucapkan terima kasih. Kami sadar masih ada kekurangan pada modul ini sehingga kami mengharapkan masukan berupa kritik dan saran yang membangun terhadap materi praktikum maupun metode serta modul praktikum untuk perbaikan terus menerus.

Yogyakarta, Maret 2025

Penyusun

No	Riwayat Penulisan	Keterangan	Penulis	
1	Maret 2021	Modul Praktikum PSE (10 unit percobaan praktikum)	Alfian Ma'arif, S. T., M. Eng.	
2	Maret 2025	Revisi (Penambahan Riwayat Penulisan, Daftar Isi, Format Laporan)	Arsyad Cahya Subrata, S.T., M.T. Muchammad Ardan Ramadhani, S.T.	

RIWAYAT PENULISAN

ALOKASI WAKTU

DAN

KESEHATAN KESELAMATAN KERJA

I. Waktu Praktikum

Pelaksanaan Waktu praktikum di laboratorium Teknik Elektro berlangsung selama 180 menit dengan pembagian waktu diatur sebagai berikut

- 1. Tugas Pendahuluan/Pre Test : 20 Menit
- 2. Percobaan Praktikum : 100 Menit
- 3. Laporan Praktikum : 60 Menit

II. Keselamatan

Pada prinsipnya, untuk mewujudkan praktikum yang aman diperlukan partisipasi seluruh praktikan dan asisten pada praktikum yang bersangkutan. Dengan demikian, kepatuhan setiap praktikan terhadap uraian panduan pada bagian ini akan sangat membantu mewujudkan praktikum yang aman.

III. Bahaya Listrik

Perhatikan dan pelajari tempat-tempat sumber listrik (stop-kontak dan circuit breaker) dan cara menyala-matikannya. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, laporkan pada asisten.

- 1. Hindari daerah atau benda yang berpotensi menimbulkan bahaya listrik (sengatan listrik) secara tidak disengaja, misalnya kabel jala-jala yang terkelupas dll.
- 2. Tidak melakukan sesuatu yang dapat menimbulkan bahaya listrik pada diri sendiri atau orang lain.
- 3. Keringkan bagian tubuh yang basah karena, misalnya, keringat atau sisa air wudhu.
- 4. Selalu waspada terhadap bahaya listrik pada setiap aktivitas praktikum.

Kecelakaan akibat bahaya listrik yang sering terjadi adalah tersengat arus listrik. Berikut ini adalah hal-hal yang harus diikuti praktikan jika hal itu terjadi:

- 1. Jangan panik,
- 2. Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing dan di meja praktikan yang tersengat arus listrik,
- 3. Bantu praktikan yang tersengat arus listrik untuk melepaskan diri dari sumber listrik,
- 4. Beritahukan dan minta bantuan asisten, praktikan lain dan orang di sekitar anda tentang terjadinya kecelakaan akibat bahaya listrik.

TATA TERTIB PRAKTIKUM

LABORATORIUM TEKNIK ELEKTRO

I. Sebelum Praktikum

- 1. Lima menit sebelum kegiatan praktikum dimulai, Mahasiswa diwajibkan sudah berada dilaboratorium.
- 2. Praktikan datang terlambat lebih dari 15 menit, dilarang mengikuti praktikum.
- 3. Praktikan wajib memahami aturan keselamatan kerja.
- 4. Mahasiswa/praktikan memakai sepatu, berpakaian rapi (tidak berkaos oblong) dan celana rapi tidak sobek.
- 5. Menaruh tas pada tempatnya (tidak di atas meja kerja)
- 6. Pengecekan peralatan sebelum dipinjam.
- 7. Menyerahkan laporan Praktikum kepada asisten Praktikum

II. Selama Praktikum

- 1. Menjaga ketertiban Lab./tidak mengganggu kegiatan praktikum yang sedang berlangsung.
- 2. Dilarang makan, minum dan merokok dalam melaksanakan praktikum.
- 3. Tidak diperkenankan memindahkan, mengambil alat/bahan dan fasilitas lab tanpa seijin laboran
- 4. Semua pengguna/pemakai peralatan laboratorium wajib menjaga kebersihan dan keamanan peralatan laboratorium.
- 5. Jika terdapat kerusakan kehilangan alat, segera melapor kepada laboran atau dosen koordinator praktikum.
- 6. Membuat laporan sementara yang ditandatangani asisten.
- 7. Praktikum susulan (INHAL) hanya diberikan maksimal 3 X percobaan , peserta yang tidak mengikuti praktikum lebih dari 3 percobaan dinyatakan gagal dalam praktikum

III. Sesudah Praktikum

- 1. Sebelum peralatan dikembalikan di chek terlebih dahulu kondisi, perlengkapan dan jumlah peralatan yang dipinjam.
- 2. Kembalikan peralatan kepada laboran.
- 3. Jika terjadi kerusakan dan kehilangan peralatan laboratorium, maka pengguna/praktikan yang merusakkan atau menghilangkan alat tersebut wajib mengganti atau memperbaiki peralatan tersebut.

IV. Sanksi

Melanggar tata aturan Laboratorium dan praktikum dikenakan sanksi berupa :

- 1. Teguran
- 2. Tidak diijinkan mengikuti praktikum / menggunakan sarana dan prasana Laboratorium.

DAFTAR ISI

KATA PENGANTAR	ii
RIWAYAT PENULISAN	iii
ALOKASI WAKTU DAN KESEHATAN KESELAMATAN KERJA	iv
TATA TERTIB PRAKTIKUM	v
DAFTAR ISI	vi
PRAKTIKUM I PENGENALAN BAHASA PEMROGRAMAN PYTHON DAN MICROPYTHON	
PRAKTIKUM II BAHASA PEMROGRAMAN PYTHON DAN INPUT OUTPUT	
PRAKTIKUM III	
I/O MICROPYTHON DENGAN ESP32	
PRAKTIKUM IV	
ANALOG TO DIGITAL CONVERTER (ADC)	
PRAKTIKUM V	
I2C LCD	27
PRAKTIKIIM VI	
MINI PROJEK	
DDARTIKIIM VII	
MENAMPILKAN HURUF DAN ANGKA PADA LCD 2x16	
PRAKTIKUM VIII	
LAMPU LALU-LINTAS	
PRAKTIKUM IX	
SENSOR ULTRASONIK	
PRAKTIKUM X	
MOTOR SERVO	

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 pr aktikum 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. 2. Terdapat beberapa bagian pada halaman tersebut, keterangan bagian-bagian website tersebut ditunjukkan pada Tabel 1.1.

	croPython ном	IE FORUM D	DOCS QUICK-REF	DOWNLOAD	STORE	CONTACT
a and b are equal MicroPython cd2f742 on 2017-11-29; ur Type "help()" for more information. >>> help() Welcome to MicroPython running under Control commands: CTRL-A on a blank line, or CTRL-B on a blank line, or CTRL-C interrupt a runnin CTRL-C on a blank line, or CTRL-E or or blank line, or blank line, or CTRL-E or or blank line, or bla	nicorn with Cortex-M: Unicorn! enter raw REPL mode enter normal REPL mod ng program do a soft reset of th enter paste modej e help('modules') micropython ucc pyb ui sys us time ut	3 de he board pllections ptruct ime	1 print("Heli	o World!")		
BINARY : PYBOARD RAM : 64KB STAT CLOCK SPEED 0.00 MHz PERIPHERALS : LED I2C LCD SERVO ADC	CK: BKB RESET		RUN SCRIPT	EDS		

Gambar 1. 2. Tampilan Website 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

Tabel 1. 1. Bagian-Bagian	Website Unicorn	MicroPython
---------------------------	-----------------	-------------

Tombol konfigurasi ditunjukkan pada Gambar 1.3. Tombol Reset berfungsi untuk mengulang program dari awal. Tombol Run Script berfungsi untuk menjalankan program yang telah dituliskan pada editor.

RESET



Gambar 1. 3. Tombol Konfigurasi

Bagian-Bagian Board PYBv1.1. ditunjukkan pada Gambar 1.4. 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.			
from pyb import LED	# memanggil fungsi LED pada library pyb		
led = LED(1)	# 1=merah, 2=hijau, 3=kuning, 4=biru		
led.on()	# menyalakan lampu LED		
led.off()	# mematikan lampu LED		

Delay/Waktu Tunda

Library Delay/Waktu tunda internal ditunjukkan pada Program 1.2.

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

Tombol Internal

Library Delay/Waktu tunda internal ditunjukkan pada Program 1.3.

Program 1. 3.			
from pyb import Switch # memanggil fungsi Switch pada library pyb			
$s_{W} - S_{W}itch()$			
sw.value()	# mengembalikan nilai True(1) atau False(0)		

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

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.

1 IUgrain 1. H.

print("Hello World!")

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.
a=10
b=5
if a>5:
print("variabel a lebih besar dari b")

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

<u>While</u>

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

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

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

<u>For</u>

Ketikan program 1.7 pada editor, kemudian jalankan program tersebut.

Program 1. 7.	
for i in range(0,10):	
print(i)	

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

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

Program 1. 8.
from pyb import LED
LED(1).on()

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.
LED(1).off()	

9. Amati lampu LED pada Board, apa fungsi perintah LED(1).off()? Tekan tombol Reset untuk mengulang program.

Menyalakan Beberapa Lampu LED

Ketikan Program 1.10 pada Editor lalu tekan tombol Run Script.

Program 1. 10.
from pyb import LED
LED(1).on()
LED(2).on()

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

Program 1. 12.
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)

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

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 asisten untuk dinilai. 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.

Conton Laporan Praktikum Embedded

Praktikum Sistem	Embedded Unit 1
Pengenalan Sis	tem Embedded
Nama (NIM)	
<u>Tujuan Praktikum</u>	Jawaban Post-Test
1.	1.
	2.
<u>Alat dan Bahan Praktikum</u>	3.
1. Komputer/ Laptop	4.
	5.
<u> Jawaban Pre-Test</u>	6.
1.	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 Ouput 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 2.1.

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

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 didefiniskan dengan standar C. Contoh perintah pada library math ditunjukkan pada Program 2.2.

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

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

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

Program 2.4.
mport 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))

- 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 pi, pi/2, pi/3, pi/6 radian ke derajat. Tuliskan hasil pada Tabel 2.1 pada kolom nilai derajat

Nilai Desimal	Nilai Radian	Nilai Derajat
pi=	pi	
pi/2=	pi/2	
pi/3=	pi/3	
pi/4=	pi/4	
pi/6=	pi/6	

Tabel 2.1. Nilai Konversi Radian ke Derajat

- 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 2.2. Pilih pilihan LED pada bagian Peripherals. Ketikan Program 2.5 untuk menyalakan lampu LED kemudian amati apakah lampu LED menyala. Untuk mematikan lampu LED ubah perintah pada bagian p12.value(0).

Program 2.5.

from machine import Pin

p12 = Pin('Y12', Pin.OUT)
p12.value(1)

PERIPHERALS :	
LED I2C LCD	
SERVO	
ADC	

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 berkedit dengan waktu tunda 100 milisekon.

Program 2.6.
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)

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 ketikan Program 2.7 pada Editor, kemudian jalankan program tersebut dengan menekan tombol Run Script.



Gambar 2.3. Konfigurasi Motor Servo

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

- 12. Gambarkan posisi sudut awal sebelum dan setelah program 2.6 dijalakan. 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. Ketikan 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 Input Output MicroPython dengan ESP32 oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 3 adalah

- 1. Mempelajari tentang pemrograman MicroPython dan Board ESP32.
- 2. Mempelajari tentang Pin Input dan Output pada MicroPython dengan konfigurasi aktif high dan aktif low dengan ESP32.

Mikrokontroller ESP32

ESP32 adalah Sistem-on-Chip (SoC) WiFi dan Bluetooth yang populer dari Espressif Systems. Menggunakan MicroPython adalah cara yang bagus untuk mendapatkan hasil maksimal dari board ESP32. Sebaliknya, chip ESP32 adalah platform yang bagus untuk menggunakan MicroPython. Penampang Board ESP32 ditunjukkan pada Gambar 3.1.

Beberapa spesifikasi dari ESP 32 adalah menggunakan prosesor Xtensa Dual-Core 32-bit LX6, Frekuensi CPU sampai 240Mhz, ROM 448kb, RAM 528kb, ADC 12bits sampai 18 channel, mendukung komunikasi I2C, SPI, UART.



Gambar 3.1. Board ESP32

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 3.2.



Gambar 3.2. Konfigurasi Ouput 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

	Program 3.1.
from machine import Pin	# Memanggil fungsi Pin pada Library pyb
<pre>p1 = Pin(1, Pin.OUT) p1.value(1) p1.value(0)</pre>	# Mendefinisikan bahwa Pin 1 adalah output # Memberikan logika high atau 1 # Memberikan logika low atau 0
$p2 = Pin(2, Pin.IN, Pin.PULL_UP)$ $p2 = Pin(2, Pin,IN, Pin,PULL_DOWN)$	# Mendefinisikan bahwa Pin 2 adalah input
print(p2.value())	# Mendapatkan nilai 0 atau 1
p5 = Pin(5, Pin.OUT, value=1)	# Mendefinisikan Pin 5 adalah output kondisi 1

Pre Test

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

1. Tuliskan program untuk mengonfigurasikan output aktif high dan input aktif low.

2. Tuliskan program untuk mengonfigurasikan input aktif high dan input aktif low.

Langkah Praktikum dan Post Test

Silahkan buka website MicroPython pada ESP32 Berikut. https://wokwi.com/projects/new/micropython-esp32

Menyalakan Lampu LED (Konfigurasi Aktif High dan Aktif Low)

Buatlah konfigurasi sesuai Gambar 3.3. Pin yang digunakan adalah D4 atau Pin Nomor 4. Tambahkan LED dengan menekan tombol bulatan berwarna ungu. Ketikan Program 3.2 untuk memberikan perintah menyalakan lampu LED kemudian amati apakah lampu LED menyala. Tekan tombol bulatan hijau untuk menjalankan program. Untuk mematikan lampu LED ubah perintah pada bagian led.value(0).

Program 3.2.	
rom machine import Pin	
ed = Pin(4, Pin.OUT)	
ed.value(1)	

- 1. Mengapa logika 1 pada led.value(1) dapat menyalakan lampu LED dan logika 0 pada led.value(0) dapat mematikan lampu LED? Berikan penjelasan.
- 2. Gambarkan rangkaian pada Gambar 3.3 pada laporan kemudian berikan penjelasan mengapa logika 1 dapat menyalakan lampu LED.
- 3. Apakah konfigurasi pada Gambar 3.3 merupakan konfigurasi aktif low atau aktif High? Berikan alasan dan penjelasan.



Gambar 3.3. Konfigurasi Output LED Aktif High

- 4. Lengkapi konfigurasi pengkabelan pada Gambar 3.4, buatlah rangkaian tersebut pada simulasi dan buat program output aktif low. Pergunakan pin tegangan 3v3 sebagai sumber tegangan positif.
- 5. Gambarkan rangkaian pada Gambar 3.4 pada laporan kemudian berikan penjelasan mengapa logika 0 dapat menyalakan lampu LED.



Gambar 3.4. Konfigurasi Output LED Aktif Low

Membaca Input Aktif High dan Aktif Low

Buatlah konfigurasi sesuai Gambar 3.5. Pin yang digunakan adalah D4 atau Pin Nomor 4. Tambahkan Pushbutton dengan menekan tombol bulatan berwarna ungu. Ketikan Program 3.3 untuk membaca input pada pushbutton kemudian amati logika pada terminal. Tekan tombol bulatan hijau untuk menjalankan program. Amati perubahan logika pada pushbutton.

Program 3.3.
from machine import Pin
<pre>button = Pin(4, Pin.IN, Pin.PULL_UP) print(button.value())</pre>
while True: print(button.value())



Gambar 3.5. Konfigurasi Input Aktif Low

7. Apa nilai logika ketika pushbutton belum ditekan pada terminal? Apa nilai logika ketika pushbutton ditekan pada terminal? Tuliskan alasannya.

8. Termasuk konfigurasi input aktif low atau input aktif high untuk rangkaian pada Gambar 3.5?

9. Modifikasilah rangkaian pada Gambar 3.5 dan Program 3.3 agar menjadi rangkaian input aktif high. Ubahlah bagian PULL_UP menjadi PULL_DOWN. Manfaatkan Pin 3v3 untuk memberikan tegangan positif atau logika high. Gambarkan rangkaian dan tuliskan program di laporan.

Menyalakan Lampu LED dengan Tombol

Buatlah konfigurasi seperti pada Gambar 3.6 dengan menambahkan LED dan pushbutton dengan menekan tombol ungu bulat. Ketikkan Program 3.4 lalu Jalankan simulasi dengan menekan tombol bulat hijau. Amati perubahan logika pada terminal dan nyala lampu LED.

10. Buatlah diagram blok dan diagram alir untuk Program 3.4.

11. Buatlah analisa per baris pada Program 3.4.

Program 3.4.	
from machine import Pin	
led = Pin(13, Pin.OUT, value=0)	
button = Pin(4, Pin.IN, Pin.PULL_UP)	
while True:	
if not button.value():	
led.value(1)	
else:	
led.value(0)	
print (button.value())	



Gambar 3.6. Konfigurasi Menyalakan LED dengan Button

Praktikum Sistem Embedded Unit 4 Analog to Digital Converter (ADC) oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 4 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 dirubah 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, sala 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

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

 2^n

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

Tabel 4. 1. Ukuran Bit dan Jangkauan Datanya

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

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

(1)

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

$$Data ADC = \frac{Tegangan Input}{Tegangan Referensi} * Ukuran bit$$
$$= \frac{2}{5} * 255 = 102$$

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

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

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

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

Oleh karena itu dapat diperoleh tegangan asli nilai ADC 102 adalah

$$Tegangan Input = \frac{Data ADC}{Ukuran bit} * Tegangan Referensi$$
$$= \frac{102}{255} * 5 = 2volt$$

Library ADC

Library ADC ditunjukkan pada Program 4.1.

	Program 4. 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 jangkaian 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. <u>http://micropython.org/unicorn/</u>

Membaca ADC

Buatlah konfigurasi sesuai Gambar 4.1. Pilih pilihan ADC pada bagian Peripherals. Ketikan Program 4.2 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 4. 2.
import machine
import pyb
y4 = machine.Pin('Y4')
adc = pyb.ADC(y4)
print(adc.read())



Gambar 4. 1. Konfigurasi Penggunaan Resistor Geser untuk ADC

1. Tuliskan Program 4.2, lalu berikan arti perintah pada setiap baris tersebut.

- 2. Geser slider pada posisi tengah, berapa nilai ADC yang ditampilkan pada terminal?
- 3. Geser slider pada posisi ujung kanan dan ujung kiri. Berapa nilai ADC pada masingmasing ujung slider tersebut? Berdasarkan data tersebut, berapa bit ukuran ADC pada mikroprosesor tersebut?
- 4. Masukkan Program 4.3 pada editor, lalu geser slider sesuai dengan gambar pada kolom pertama Tabel 4.2. Tekan tombol Run Script dan amati hasilnya pada Terminal.

Pada Tabel 4.2, 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.

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

Tabel 4. 2.

Program 4. 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 4.3 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 4.3, untuk nilai tegangan referensi bernilai 5 volt. Tuliskan program tersebut pada laporan lalu lengkapi Tabel 4.3. Tuliskan tabel tersebut pada laporan.

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

Tabel 4. 3.

8. Buatlah diagram blok dan flowchart/ diagram alir Program 4.3 dan Program pada nomor 7.

Praktikum Sistem Embedded Unit 5 I2C LCD oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 5 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 5.1, 5.2 dan 5.3. 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 5.3.











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 degnan 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

Program 5. 1.			
import machine		#Memasukkan library machine	
import framebuf		#Memasukkan library framebuf	
<pre>scl = machine.Pin('X</pre>	K9')	#Mendefinisikan SCL sebagai PIN X9	
sda = machine.Pin('X10')	#Mendefinisikan SDA sebagai PIN X10	
i2c = machine.I2C(scl=scl, sda=sda)		#Mendefinisikan i2c sebagai machine.i2c	
fbuf = framebuf.FrameBuffer(bytearray(64 * 32 // 8), 64, 32, framebuf.MONO_HLSB) # Mendefinisikan fbuf sebagai buffer array LCD dengan ukuran resolusi 64x32			
fbuf.pixel(x,y,c)	# Perintah membua	t titik piksel, x adalah letak koordinat x	
	# y adalah letak koo	ordinat y	
	# c dapat bernilai 1	berarti muncul titik dan 0 berarti titik hilang	
fbuf.rect(x,y,w,h,c)	# Perintah untuk membuat persegi, h=height/ tinggi # w=width/ lebar, x=koordinat x, y=koordinat y		
fbuf.text(s,x,y,c) # Perintah untuk membuat teks, s=teks			
fbuf.hline(x,y,w,c)	c) # Perintah untuk membuat garis horisontal		
fbuf.vline(x,y,h,c)	# Perintah untuk membuat garis vertikal		
fbuf.line(x1,y1,x2,y2	2,c) # Perintah untu	k garis	
i2c.writeto(8, fbuf)	# Perintah untuk me # jalur komunikasi 1	engirim variabel fbuf sebesar 8bit melalui 12C	

Library I2C Ditunjukkan pada Program 5.1.

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

Menampilkan Piksel pada LCD I2C

Buatlah konfigurasi sesuai Gambar 5.4. Pilih pilihan I2C LCD pada bagian Peripherals. Ketikan Program 5.2 untuk menampilkan piksel pada LCD. Lalu tekan tombol Run Script untuk menjalankan Program dan melihat tampilan pada I2C LCD.







- 1. Gambarkan letak titik piksel Program 5.1.
- 2. Gambarkan diagram blok menampilkan Piksel pada LCD tersebut.

3. Modifikasi bagian fbuf.pixel(0,0,1) sesuai Tabel 5.1. Gambarkan letak piksel pada kolom kedua.

Tabel 5. 1.

Perinah Membuat Titik Piksel	Letak Piksel pada LCD
fbuf.pixel(63,31,1)	
fbuf.pixel(63,0,1)	
fbuf.pixel(0,31,1)	
fbuf.pixel(32,16,1)	

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

Program 5. 3.
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)

5. Gambarkan letak persegi Program 5.3.

6. Modifikasi Program 5.3 sesuai Tabel 5.2, lalu lengkapi tabel tersebut.

Tabel 5. 2.

Perinah Menggambar Persegi	Gambar Persegi pada LCD
fbuf.rect(0,0,10,10,1)	
fbuf.rect(10,10,15,15,1)	

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

Program 5. 4.
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)

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

Menampilkan Garis pada LCD I2C

Ketikan Program 5.5 untuk menampilkan teks pada LCD. Lalu tekan tombol Run Script untuk menjalankan Program dan melihat tampilan pada I2C LCD.

```
Program 5. 5.

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)
```

9. Gambarkan hasil Program 5.5, lalu tuliskan program tersebut dan berikan analisis setiap baris.

Praktikum Sistem Embedded Unit 6 Mini Projek oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 6 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 <u>http://micropython.org/unicorn/</u>

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 6.1, lalu ketikan Program 6.1 pada editor. Tekan tombol Run Script untuk menjalankan Program. Tekan tombol USR pada mikrokontroler seperti ditunjukkan tanda panah pada Gambar 6.1 untuk menyalakan semua LED.

Program 6. 1.
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)



Gambar 6. 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 6.1. dan diagram alir Program 6.1.

2. Menampilkan data ADC pada LCD I2C

Buatlah konfigurasi sesuai Gambar 6.2, lalu ketikan Program 6.2 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 6. 2.

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.fill(0)

fbuf.text(str(dataadc),0,9,1)

i2c.writeto(8, fbuf)
```

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



Gambar 6. 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 6.2. dan diagram alir Program 6.2.

3. Menggerakkan Motor Servo dengan Slider ADC

Buatlah konfigurasi sesuai Gambar 6.3, lalu ketikan Program 6.3 pada editor. Tekan tombol Run Script untuk menjalankan program. Geser slider ke kiri dan ke kanan kemudian perhatikan gerakan sudut motor servo.

Program 6. 3.
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)



Gambar 5. 3. Konfigurasi Menggerakan Motor Servo dengan Slider

- 3.1. Perintah mana pada Program 6.3 yang berfungsi untuk menggerakan motor servo?
- 3.2. Lakukan modifikasi pada Program 6.3 untuk menampilkan nilai sudut pada terminal. Perintah apa yang digunakan?
- 3.3. Gambarkan diagram blok sistem pada Gambar 6.3. Perangkat apa pada sistem Gambar 6.3, 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 6.4, lalu ketikan Program 6.4 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 6.4.
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 6. 4. Konfigurasi Menggerakan Motor Servo dengan Slider dan menampilkan nilai sudut pada LCD I2C disertai indikator LED

4.1. Buatlah diagram blok sistem pada Gambar 6.4 dan diagram alir dari program 6.4. 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

Praktikum Sistem Embedded Unit 7 MicroPython dengan ESP32 Menampilkan Huruf dan Angka pada LCD 2x16 oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 7 adalah

1. Mempelajari tentang pemrograman MicroPython dan Board ESP32 untuk menampilkan tulisan dan data pada LCD 2x16.

Liquid Crystal Display (LCD)

Liquid Crystal Display (LCD) adalah sebuah layar penampil berbahan kristal cair. Pada mikrokontroler terdapat beberapa jenis LCD dengan ukuran dan jenis yang bervariasi seperti yang ditunjukkan pada Gambar 7.1. Misalnya LCD 2x16, LCD 2x20, LCD OLED dsb. Ukuran 2x16 menunjukkan jumlah baris dan kolom yang berarti LCD memiliki 2 baris dan 16 kolom. Sementara LCD OLED adalah jenis LCD yang memiliki titik lebih banyak dibandingkan LCD biasa seperti LCD OLED 128x64.

LCD memiliki dua jenis konfigurasi pengkabelan yaitu konfigurasi standar dan konfigurasi I2C. Konfigurasi standar memerlukan jumlah kabel yang lebih banyak daripada konfigurasi I2C. Komunikasi LCD dengan I2C menggunakan jalur komunikasi Pin SDA dan SCl pada mikrokontroler. Umumnya, pada komunikasi I2C, masing-masing perangkat LCD memiliki alamat heksa untuk mengakses perangkat LCD. Oleh karena itu perlu diketahui alamat heksa tersebut sebelum digunakan.



Gambar 7. 1. Jenis-Jenis LCD

Pre Test

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

- 1. Apa kepanjangan dari LCD dan Apa kepanjangan dari I2C?
- 2. Apa nama Pin yang digunakan untuk komunikasi I2C?

Langkah Praktikum dan Post Test

Silakan buka website MicroPython pada ESP32 Berikut. https://wokwi.com/projects/new/micropython-esp32

Membaca Alamat LCD I2C

Buatlah konfigurasi sesuai Gambar 7. 2. Pin yang digunakan adalah D21 dan D22, sambungkan kedua Pin tersebut ke Pin SDA dan SCL pada LCD. Ketikan Program 7.1 dan jalankan program tersebut untuk mengetahui alamat heksa LCD.



Gambar 7. 2. Konfigurasi

```
Program 7. 1.

import machine

sdaPIN=machine.Pin(21) #for ESP32

sclPIN=machine.Pin(22)

i2c=machine.I2C(sda=sdaPIN, scl=sclPIN, freq=10000)

devices = i2c.scan()

if len(devices) == 0:

print("Tidak ada Perangkat I2C !")

else:

print("Perangkat I2C Ditemukan = ",len(devices))

for device in devices:

print("Alamat Heksa Perangkat = ",hex(device))
```

1. Pada terminal, berapa alamat heksa LCD yang tertulis? Ingat alamat heksa LCD tersebut untuk program berikutnya.

2. Apakah alamat heksa tersebut sama dengan alamat heksa praktikan lain? Berapa alamat heksa praktikan lain?

Menuliskan Huruf dan Angka pada LCD I2C

Tambahkan 2 buah library pada dengan cara menekan menu New File pada Library Manager seperti ditunjukkan pada Gambar 7. 3. Beri nama lcd_api.py dan i2c_lcd.py untuk kedua file tersebut.



Gambar 7. 3. Membuat Library

Isikan library lcd_api.py dengan copy paste semua program dari alamat URL https://peppe8o.com/download/micropython/LCD/lcd_api.py

Isikan library i2c_lcd.py dengan copy paste semua program dari alamat URL https://peppe8o.com/download/micropython/LCD/i2c_lcd.py

Ketikkan Program 7. 2 lalu jalankan program tersebut. Amati tulisan dan angka yang tertampil pada LCD.

Program 7. 2.import machinefrom machine import Pin, SoftI2Cfrom lcd_api import LcdApifrom i2c_lcd import I2cLcdfrom time import sleepI2C_ADDR = 0x27totalRows = 2totalColumns = 16i2c = SoftI2C(scl=Pin(22), sda=Pin(21), freq=10000)lcd = I2cLcd(i2c, I2C_ADDR, totalRows, totalColumns)

while True:
lcd.putstr("I2C LCD Tutorial")
sleep(2)
lcd.clear()
lcd.putstr("Lets Count 0-10!")
sleep(2)
lcd.clear()
for i in range(11):
lcd.putstr(str(i))
sleep(1)
lcd.clear()

3. Perintah apa yang digunakan untuk menuliskan teks dan angka ke LCD?

4. Perintah apa yang digunakan untuk menghapus tulisan pada LCD?

5. Perintah apa yang digunakan untuk menahan tulisan di LCD agar tetap ditampilkan sebelum dihapus?

6. Pada bagian apa pada Program 7. 2 agar dapat tertampil angka sampai 15?

Menuliskan Data Sensor DHT22 di LCD I2C

Buatlah konfigurasi simulasi seperti pada Gambar 7. 4. Lalu ketikan Program 7. 3 dan jalankan program tersebut. Amati hasilnya pada LCD.

Program 7. 3.
import machine
from machine import Pin, SoftI2C
from lcd_api import LcdApi
from i2c_lcd import I2cLcd
from time import sleep
import dht
sensor = dht.DHT22(Pin(14))
$I2C_ADDR = 0x27$
totalRows = 2
totalColumns = 16
i2c = SoftI2C(scl=Pin(22), sda=Pin(21), freq=10000)
lcd = I2cLcd(i2c, I2C_ADDR, totalRows, totalColumns)

while True:	
sensor.measure()	
temp = sensor.temperature()	
hum = sensor.humidity()	
lcd.move_to(0,0)	
lcd.putstr("Sensor DHT 22")	
lcd.move_to(0,1)	
lcd.putstr(str(temp))	
sleep(1)	
lcd.clear()	



Gambar 7. 4. Membuat Library

7. Perintah apa yang digunakan untuk memindahkan tulisan pada LCD ke baris kedua?

8. Tuliskan library dan variabel yang berkaitan dengan sensor DHT22.

9. Perintah apa yang digunakan untuk menentukan panjang baris dan panjang kolom LCD?

10. Modifikasilah Program 7. 3 agar pada LCD ditampilkan nilai kelembapan seperti pada Gambar 7. 5. Tuliskan modifikasi program pada laporan.



Gambar 7. 5. Membuat Library

Praktikum Sistem Embedded Unit 8 MicroPython dengan ESP32 Lampu Lalu Lintas oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 8 adalah

1. Mempelajari tentang pemrograman MicroPython dan Board ESP32 untuk membuat lampu lalu lintas 1 jalur, 2 jalur dan 3 jalur.

Pre Test

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

- 1. Bagaimana cara mengonfigurasi Pin Output pada Micropython?
- 2. Apa itu konfigurasi input aktif high dan aktif low? Apa nilai logika awal untuk konfigurasi aktif low dan aktif high? 0 atau 1?

Langkah Praktikum dan Post Test

Silakan buka website MicroPython pada ESP32 Berikut. https://wokwi.com/projects/new/micropython-esp32

Membuat Lampu Lalu Lintas 1 Jalur

Buatlah konfigurasi sesuai Gambar 8. 1. Pin yang digunakan adalah D23, D22 dan D21, sambungkan ketiga Pin tersebut ke kaki Anode LED merah, kuning dan hijau. Ketikan Program 8. 1 dan jalankan program tersebut untuk mengetahui hasil lalu lintas 1 jalur.



Gambar 8. 1. Lalu Lintas 1 Jalur

Program 8. 1.
from machine import Pin
from time import sleep
ladMarah1 - Din(22) Din OUT value = 0)
ledWerlan1 = Pln(23, Pln, OUT, value=0)
ledKuning1 = Pin(22, Pin.001, value=0)
ledHijau1 = Pin(21, Pin.OUT, value=0)
while True:
ledMerah1.value(1)
ledKuning1.value(0)
ledHijau1.value(0)
sleep(1)
ledMerah1.value(0)
ledKuning1.value(1)
ledHijau1.value(0)
sleen(1)
ledMerah1.value(0)
ledKuning1.value(0)
$ledHii_{2} + 1 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2$
sieep(1)

1. Buatlah flowchart program lampu lalu lintas 1 jalur.

2. Bagian mana pada program yang harus dimodifikasi untuk membuat lampu LED menyala lebih lama?

3. Termasuk konfigurasi aktif high atau aktif low rangkaian tersebut? Tuliskan penjelasannya.

Membuat Lampu Lalu Lintas 2 Jalur

Buatlah konfigurasi sesuai Gambar 8. 2. Pin yang digunakan adalah D23, D22, D21, D19, D18, dan D5 sambungkan keenam Pin tersebut ke kaki Anode LED merah, kuning dan hijau jalur satu dan jalur dua.

5. Buatlah flowchart dan program lampu lalu lintas 2 jalur dengan aturan sebagai berikut.

- Ketika jalur 1 merah, jalur 2 hijau
- Ketika jalur 2 merah, jalur 1 hijau

- Ketika perpindahan dari lampu hijau ke lampu merah, terlebih dahulu menyala kuning, berlaku di kedua jalur

- Perpindahan nyala lampu dari merah ke hijau, langsung tanpa melalui kuning

Sebagai conton.								
Sleep	1	1	1	1	1	1	1	1
Jalur 1								
Jalur 2								

6. Buatlah tabel waktu untuk mempermudah menentukan lama waktu menyala lampu sebagai contoh.



Gambar 8. 2. Lalu Lintas 2 Jalur

Membuat Lampu Lalu Lintas 3 Jalur

- 7. Buatlah konfigurasi lampu lalu lintas 3 jalur.
- 8. Buatlah flowchart dan program lampu lintas 3 jalur.
- 9. Buatlah tabel waktu lampu lintas 3 jalur.

Praktikum Sistem Embedded Unit 9 MicroPython dengan ESP32 Sensor Ultrasonik oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 9 adalah

1. Mempelajari tentang pemrograman MicroPython dan Board ESP32 untuk membaca data jarak sensor ultrasonik.

Sensor Ultrasonik HC-SR04

Sensor ultrasonik HC-SR04 menggunakan sonar untuk mendeteksi jarak terhadap objek. Sensor tersebut dapat membaca jarak dari 2cm sampai 400 cm dengan akurasi 0,3 cm yang cukup bagus untuk sebuah proyek. Selain itu, modul khusus ini dilengkapi dengan modul pemancar dan penerima ultrasonik. Terdapat empat buah Pin pada sensor tersebut yaitu Pin sumber tegangan positif (VCC), Pin masukan trigger (Trig), Pin keluaran Echo (Echo), dan Pin sumber tegangan negatif (GND).

Cara kerja sensor ultrasonik menggunakan sonar untuk menentukan jarak suatu benda. Cara kerjanya adalah sebagai berikut (1) Pemancar ultrasonic (Pin Trigger) memancarkan suara frekuensi tinggi (40kHz). (2) Suara tersebut merambat di udara, jika menemukan sebuah objek, maka akan memantul kembali ke modul. (3) Penerima ultrasonik (Pin Echo) menerima suara yang dipantulkan.

Dengan mempertimbangkan kecepatan suara di udara dan waktu tempuh (waktu berlalu sejak pengiriman dan penerimaan sinyal), dapat dihitung jarak ke suatu objek. Persamaan untuk menghitung jarak objek adalah

jarak objek = (kecepatan suara di udara) * waktu/2

Dengan nilai kecepatan suara di udara adalah 343 m/s.

Pre Test

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

1. Apa itu sensor ultrasonik? Bagaimana cara kerja sensor ultrasonik?

2. Sebutkan lima aplikasi sistem yang menggunakan sensor ultrasonik.

Langkah Praktikum dan Post Test

Silakan buka website MicroPython pada ESP32 Berikut. https://wokwi.com/projects/new/micropython-esp32

Membaca Data Sensor Ultrasonik

Buatlah konfigurasi sesuai Gambar 9.1. Pin yang digunakan adalah D2 dan D4, sambungkan ketiga Pin tersebut ke kaki Echo dan Trigger. Ketikan Program 9.1 dan jalankan program tersebut untuk mengetahui hasilnya. Pindahkan slider pada sensor ultrasonik untuk mengubah jarak.



Gambar 9. 1. Desain Rangkaian Sensor Ultrasonik

Program 9. 1
from machine import Pin
import utime
trigger = Pin(4, Pin.OUT)
echo = Pin(2, Pin.IN)
defultra():
trigger.value(0)
utime.sleep_us(2)
trigger.value(1)
utime.sleep_us(5)
trigger.value(0)
while $echo.value() == 0$:
signaloff = utime.ticks_us()
while echo.value() == 1:
signalon = utime.ticks_us()
timepassed = signalon - signaloff
distance = $(timepassed*0.0343) / 2$
print("Jarak dari Objek adalah ", distance, "cm")
while True:
ultra()
utime.sleep(1)

- 1. Buatlah flowchart Program 9.1.
- 2. Lengkapi hasil data sensor pada Tabel 9.1. Pergunakan persamaan berikut untuk menghitung nilai error.

Error = |Jarak Objek - Nilai Jarak Digital|

- 3. Apa fungsi perintah def pada Program 9.1?
- 4. Apa fungsi perintah trigger = Pin(4, Pin.OUT) dan perintah echo = Pin(2, Pin.IN)?
- 5. Apa fungsi perintah trigger.value(0) dan perintah trigger.value(1)?
- 6. Apa fungsi perintah while True? Sama seperti perintah apa dalam pemrograman bahasa C perintah tersebut?

Data ke-	Jarak Objek (cm)	Nilai Jarak Digital (cm)	Nilai Error
1	10		
2	25		
3	40		
4	55		
5	70		
6	95		
7	110		
8	200		
9	140		
10	155		
11	170		
12	195		
13	210		
14	225		
15	250		
	Nilai Error Ra	ata-Rata	

Tabel 9. 1. Data Sensor

Praktikum Sistem Embedded Unit 10 MicroPython dengan ESP32 Motor Servo oleh Alfian Ma'arif, M. Eng.

Tujuan

Tujuan Praktikum Sistem Embedded Unit 10 adalah

1. Mempelajari tentang pemrograman MicroPython dan Board ESP32 untuk menggerakkan motor sudut servo.

Motor Servo

Motor servo adalah perangkat atau aktuator putar yang dirancang dengan sistem kendali umpan balik kalang tertutup sehingga dapat diatur untuk menentukan dan memastikan posisi sudut motor. Motor servo merupakan perangkat yang terdiri atas motor DC, serangkaian gear, rangkaian kendali dan potensiometer. Serangkaian gear yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo. Sedangkan potensiometer dengan perubahan nilai resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo. Penggunaan sistem kendali kalang tertutup pada motor servo berguna untuk mengontrol gerakan dan posisi akhir dari poros motor servo.

Motor servo dapat digunakan dengan ESP32 pada semua port output yang mendukung PWM (modulasi lebar pulsa). Versi Micropython saat ini untuk ESP32 tidak menyertakan dukungan langsung untuk motor servo, tetapi dapat menggunakan pustaka PWM untuk mengendalikannya. Motor servo memiliki tiga kaki yaitu Pin kendali sinyal, Pin tegangan positif dan Pin tegangan negatif.

Pre Test

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

1. Apa itu motor Servo?

2. Apa perbedaan antara motor DC biasa dan Motor Servo?

Langkah Praktikum dan Post Test

Silakan buka website MicroPython pada ESP32 Berikut. https://wokwi.com/projects/new/micropython-esp32

Membaca Data Sensor Ultrasonik

Buatlah konfigurasi sesuai Gambar 10. 1. Pin yang digunakan adalah D2, D4 dan D15, sambungkan ketiga Pin tersebut ke kaki tombol 1, tombol 2 dan PWM motor servo. Ketikan Program 10. 1 dan jalankan program tersebut untuk mengetahui hasilnya. Tekan tombol 1 dan tombol 2 untuk mengubah sudut motor servo.



Gambar 10. 1. Desain Rangkaian Motor Servo

```
Program 10. 1.
from machine import Pin, PWM
import time
pos = 25
step =1
p15 = Pin(15)
servo = PWM(p15,freq=50)
button1 = Pin(4, Pin.IN, Pin.PULL_UP)
button2 = Pin(2, Pin.IN, Pin.PULL_UP)
servo.duty(pos)
while True:
print(pos)
if not button1.value():
 if pos>25:
 servo.duty(pos)
 pos=pos-step
 time.sleep_ms(10)
if not button2.value():
 if pos<125:
 servo.duty(pos)
 pos=pos+step
  time.sleep_ms(10)
```

- 1. Berdasarkan referensi Program 10.1, buatkan program untuk menggerakkan sudut motor servo tanpa tombol.
- 2. Apa fungsi perintah servo.duty(pos)?
- 3. Gambarkan posisi sudut motor servo untuk nilai servo.duty(25), servo.duty(50), servo.duty(75), servo.duty(100), servo.duty(125).
- 4. Buatlah flowchart Program 10.1.
- 5. Apa fungsi perintah pos=pos-step? Jelaskan dengan ilustrasi angka.
- 6. Apa fungsi perintah pos=pos+step? Jelaskan dengan ilustrasi angka.



2025

CONTOH PENULISAN LAPORAN

PRAKTIKUM I

JUDUL UNIT PRAKTIKUM

A. Kompetensi Dasar

Tuliskan dasar kompetensi praktikum

B. Alat dan Bahan

Tuliskan peralatan yang digunakan dalam percobaan praktikum

C. Hasil dan Pembahasan

Tuliskan hasil dan analisa setiap pengambilan data percobaan praktikum

D. Tugas (Jika ada)

Tuliskan jawaban dari tugas yang ada

E. Kesimpulan

Tuliskan kesimpulan dari unit percobaan praktikum

F. Daftar Pustaka

Tuliskan daftar pustaka :

Nama penulis. (Tahun). Judul. Kota penerbit : Tempat penerbit

Contoh :

Ma'arif, Alfian. (2025). Petunjuk Praktikum Sistem Embeded . Yogyakarta : Universitas Ahmad Dahlan

G. Lampiran

Lampiran berupa lembar pengamatan yang telah disediakan.