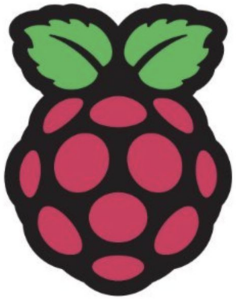
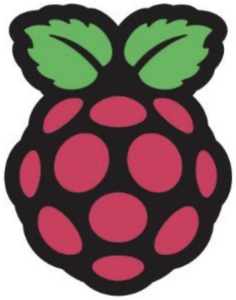


Pi Pico-w 주행로봇 프로젝트

2245051 이승원

Computer Science and Engineering

목차

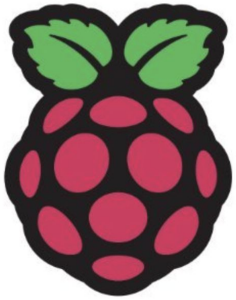


프로젝트 배경

1. 프로젝트 배경

Embedded System

1. 특정한 기능에 초점을 맞추고 최적화되어 있음.
2. 제한된 자원(메모리, 연산능력, 전력 등)을 가지고 효율적으로 작동.
3. 외부 환경과의 인터페이스 및 상호작용이 필요한 경우가 많음.
4. 실시간 요구 사항을 충족해야 하는 경우가 있음.
5. 높은 안정성과 신뢰성이 요구됨.



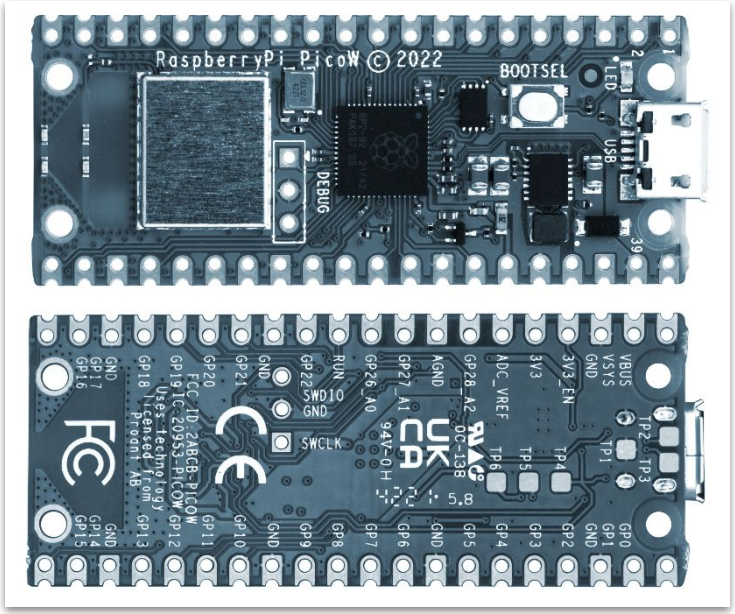
프로젝트 기술개요

HW/SW

Computer Science and Engineering

2. 프로젝트 기술 개요

Raspberry Pi-Pico w

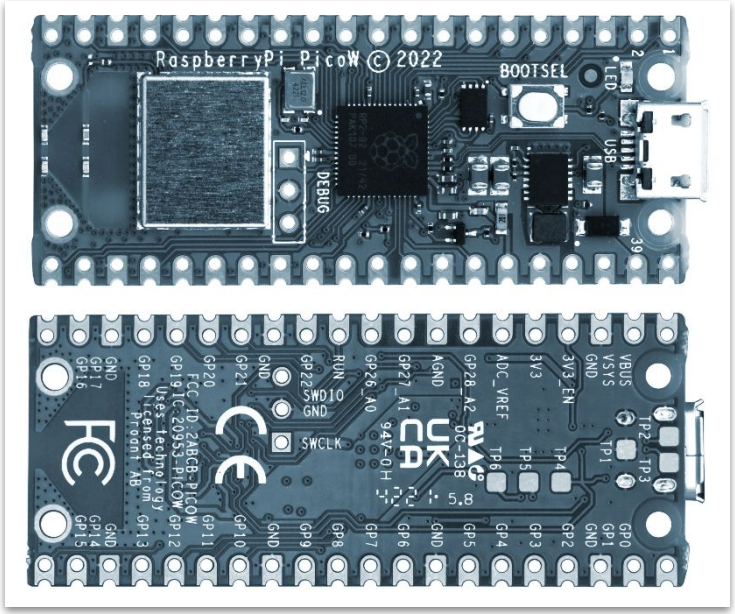


1. 프로세서: **RP2040**
   - 듀얼코어 **ARM Cortex-M0+** 프로세서
   - 최대 클럭 속도 **133MHz**

2. 메모리:
   - **264KB** 내장 **SRAM**
   - 외부 **QSPI** 플래시 메모리 지원 **(2MB** 플래시 메모리가 기본으로 포함되어 있음)

2. 프로젝트 기술 개요

Raspberry Pi-Pico w



3. **GPIO 및 통신:**

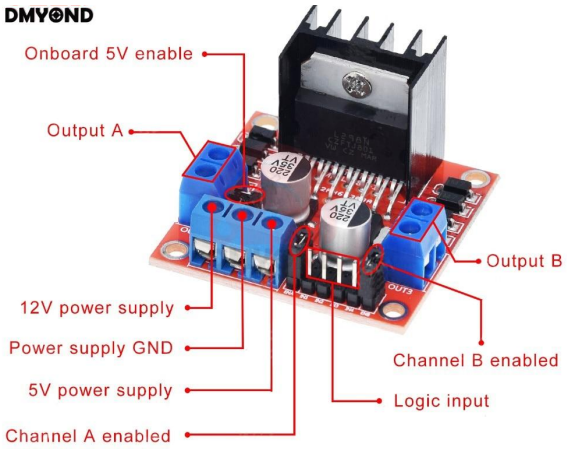
- **26개의 다기능 GPIO 핀**
- **하드웨어 지원: UART, SPI, I2C**
- **16개의 PWM 채널**
- **3개의 12비트 ADC (아날로그 디지털 변환기)**

4. **전원 공급:**

- **입력 전압 범위: 1.8V - 5.5V**
- **마이크로 USB 포트를 통한 전원 공급 가능**
- **별도의 전원 공급을 위한 핀(VSYS) 제공**

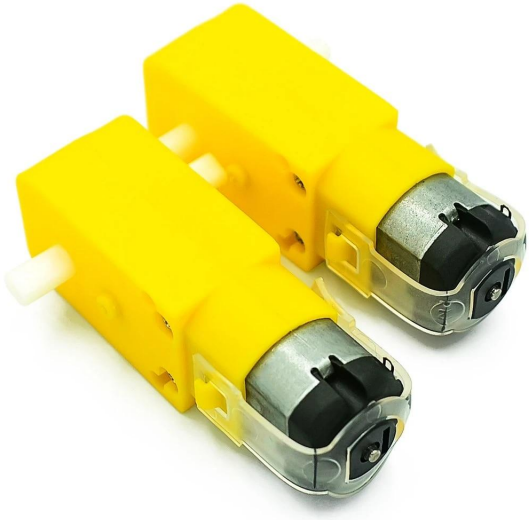
2. 프로젝트 기술 개요

Motor Driver L298N



1. 전원 공급 전압 범위:
   - 로직 공급 전압 (Vss): 4.5V - 7V
   - 모터 공급 전압 (Vs): 최대 46V
2. 출력 전류:
   - 연속 전류: 최대 2A (각 채널당)
   - 최대 전류: 3A (각 채널당, 펄스로 제공)
3. 듀얼 H-브리지:
   - 두 개의 독립적인 H-브리지를 포함
   - 두 개의 DC 모터를 양방향으로 구동
4. 보호 기능:
   - 역전압 보호
   - 온도 보호
   - 과전류 보호

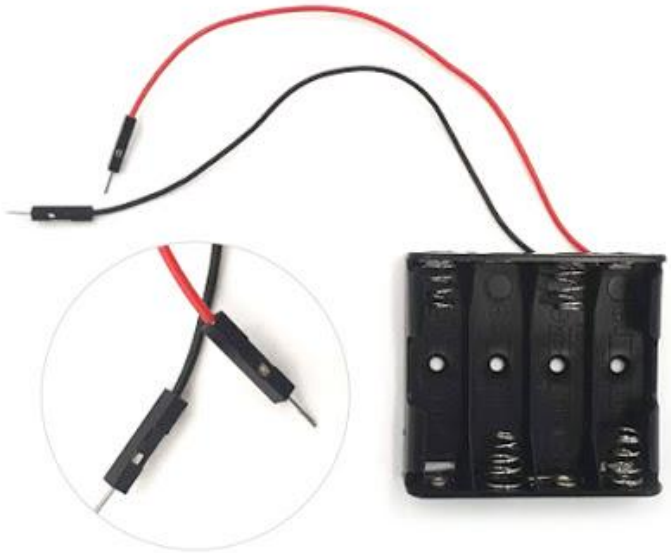
2. 프로젝트 기술 개요



TT motor

1. 작동 전압: **3V ~ 12V DC** (권장 작동 전압 약 **6 ~ 8V**)
2. 최대 토크: 최소 **800gf cm** (**3V** 시)
3. 무부하 속도: **1:48** (**3V** 시)
4. 부하 전류: **70mA** (최대 **250mA**) (**3V** 일 때)
5. **EMC**, 간섭 방지 기능

2. 프로젝트 기술 개요



4xAA Serial Battery Holder

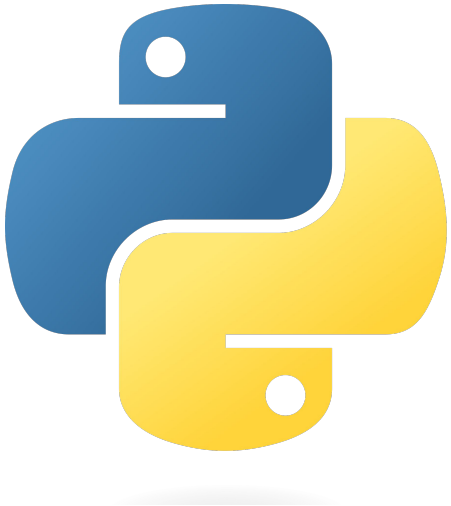
1. 배터리 호환성: **AA** 크기 배터리 (일반적으로 각각 **1.5V**)
2. 구성: 직렬 연결 (배터리가 끝에서 끝으로 연결됨)
3. 출력 전압: **6V (4 x 1.5V) -** 사용되는 배터리의 실제 전압에 따라 다를 수 있음
4. 출력 전류: 사용되는 배터리의 용량 및 방전률에 따라 다름
5. 커넥터 유형: 일반적으로 와이어 리드가 있음

2. 프로젝트 기술 개요

HC-SR04 초음파센서

1. 작동 전압: **5V DC**
2. 전류 소모: **15mA** (작동 중)
3. 측정 각도: 약 **15**도
4. 측정 범위: **2cm - 400cm** (약 **0.8**인치 - **13**피트)
5. 정밀도: 약 **3mm**
6. 출력 유형: 디지털 출력 (펄스 폭 변조, **PWM**)
7. 트리거 및 에코 핀: **2**개의 **I/O** 핀 (**1**개의 입력 트리거 핀, **1**개의 출력 에코 핀)
8. 최소 트리거 시간 간격: **10ms** (최적의 성능을 위해 **60ms** 이상 권장)

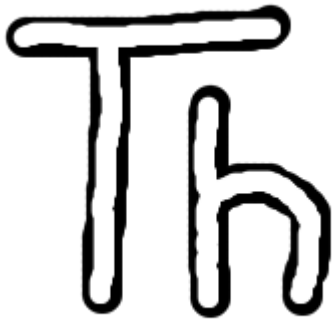
2. 프로젝트 기술 개요



microPython

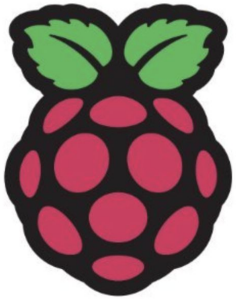
a. 제한된 환경에서 작동, 메모리 효율적 사용
b. 축소된 표준 라이브러리
c. 하드웨어 제어를 위한 내장 기능을 제공, **GPIO** 핀, **I2C, SPI, UART** 등의 통신 프로토콜을 지원

2. 프로젝트 기술 개요



ThonnyIDE

a. 사용자 친화적 인터페이스
b. 마이크로파이썬 지원
c. 코드 편집기



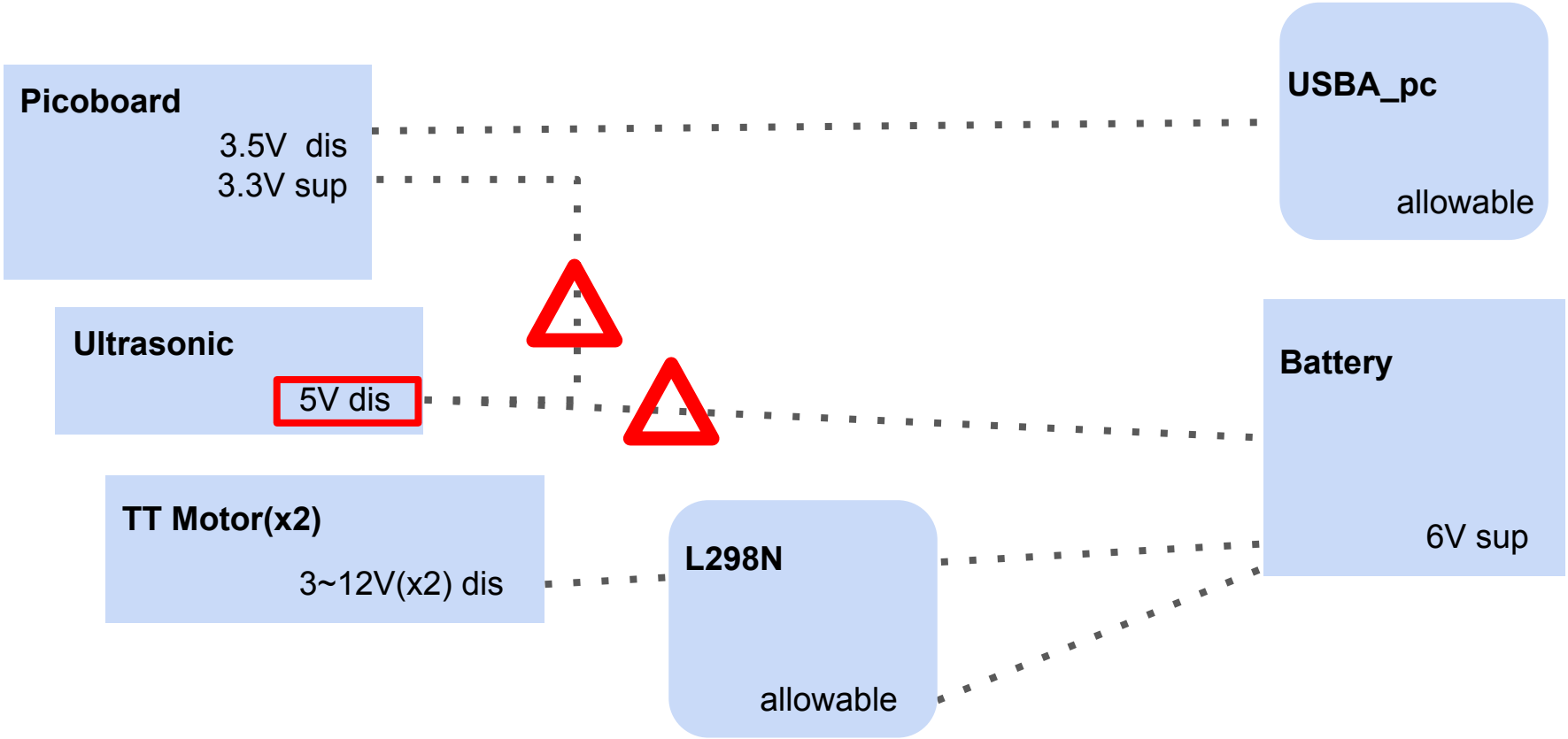
설계 및 구현

전류설계 중요성을 중심으로

Computer Science and Engineering

3. 설계 및 구현

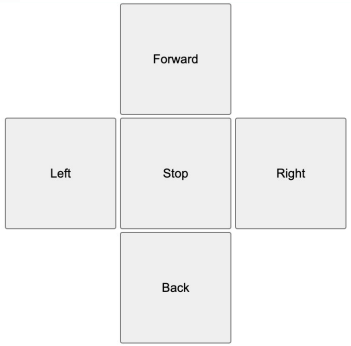
다음 전류흐름도와 코드를 이용해 설명하겠습니다.



*allowable : 시스템 전원에 영향을 미치지 않을 만큼 가변적임

```
eddedmain.py   <untitled>   [ main.py
def move_forward():

    Mot_A_Forward.value(0)
    Mot_B_Forward.value(0)
    Mot_A_Back.value(1)
    Mot_B_Back.value(1)

def move_backward():
    Mot_A_Forward.value(1)
    Mot_B_Forward.value(1)
    Mot_A_Back.value(0)
    Mot_B_Back.value(0)

def move_stop():
    Mot_A_Forward.value(0)
    Mot_B_Forward.value(0)
    Mot_A_Back.value(0)
    Mot_B_Back.value(0)

def move_left():
    Mot_A_Forward.value(0)
    Mot_B_Forward.value(1)
    Mot_A_Back.value(1)
    Mot_B_Back.value(0)

def move_right():
    Mot_A_Forward.value(1)
    Mot_B_Forward.value(0)
    Mot_A_Back.value(0)
    Mot B Back.value(1)
```

```
%Run -c $EDITOR_CONTENT
결됨 192.168.213.21
aceback (most recent call last):
File "<stdin>", line 142, in <module>
File "<stdin>", line 119, in serve
Error: [Errno 104] ECONNRESET
```



1. Wi-Fi 연결을 설정합니다.
2. 웹 페이지 생성, 웹 서버 설정 후 클라이언트 요청에 응답합니다.
3. 각 버튼(전진, 후진, 왼쪽, 오른쪽, 정지)에 대한 웹 요청을 처리하고 로봇에게 이동을 지시합니다.

- move_forward(), move_backward(), move_left(), move_right(), move_stop() 함수: 이들 함수는 각각 전진, 후진, 왼쪽, 오른쪽 및 정지 명령을 처리합니다. 이들 함수는 모터 드라이버 핀에 적절한 출력을 제공하여 로봇의 움직임을 제어합니다.
- connect() 함수: Wi-Fi 네트워크에 연결하는 데 사용됩니다. 연결이 완료되면 IP 주소를 반환합니다.
- open_socket() 함수: 주어진 IP 주소를 사용하여 소켓을 열고 연결을 수신합니다.
- webpage() 함수: 웹 인터페이스를 위한 HTML 페이지를 생성합니다. 이 페이지에는 로봇을 제어하는 데 사용되는 버튼이 포함되어 있습니다.
- serve() 함수: 웹 서버를 시작하고 클라이언트 요청을 처리합니다. 각 요청은 로봇이 이동해야 하는 방향을 나타내는 URL 경로를 포함합니다. 요청에 따라 적절한 이동 함수가 호출됩니다.

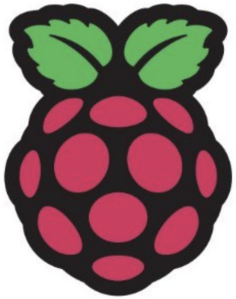
https://github.com/Guitarman9119/Raspberry-Pi-Pico- / https://www.explainingcomputers.com/

```
#핀모터 매소드를 정의합니다.
Mot_A_Forward = Pin(2, Pin.OUT)
Mot_A_Back = Pin(3, Pin.OUT)
Mot_B_Forward = Pin(7, Pin.OUT)
Mot_B_Back = Pin(8, Pin.OUT)
ENA = Pin(4, Pin.OUT)
ENB = Pin(6, Pin.OUT)
pwm_A = PWM(ENA)
pwm_B = PWM(ENB)
#echo_pin = machine.Pin(9, machine.Pin.IN)
#trigger_pin = machine.Pin(5,machine.Pin.OUT)
```

```
# PWM 듀티 사이클을 설정합니다. (0-65535 사이의 값)
pwm_A.duty_u16(45875)  # 70% 듀티 사이클
pwm_B.duty_u16(45875)  # 70% 듀티 사이클
```

```
def webpage():
    #HTML 디자인
    html = f"""
    """

    #초음파센서 매소드를 정의합니다.
    def measure_distance():
        trigger_pin.value(1)
        time.sleep_us(10)
        trigger_pin.value(0)
```

```
try:
    ip = connect()
    connection = open_socket(ip)
    serve(connection)
```



실험 및 평가

영상

Computer Science and Engineering

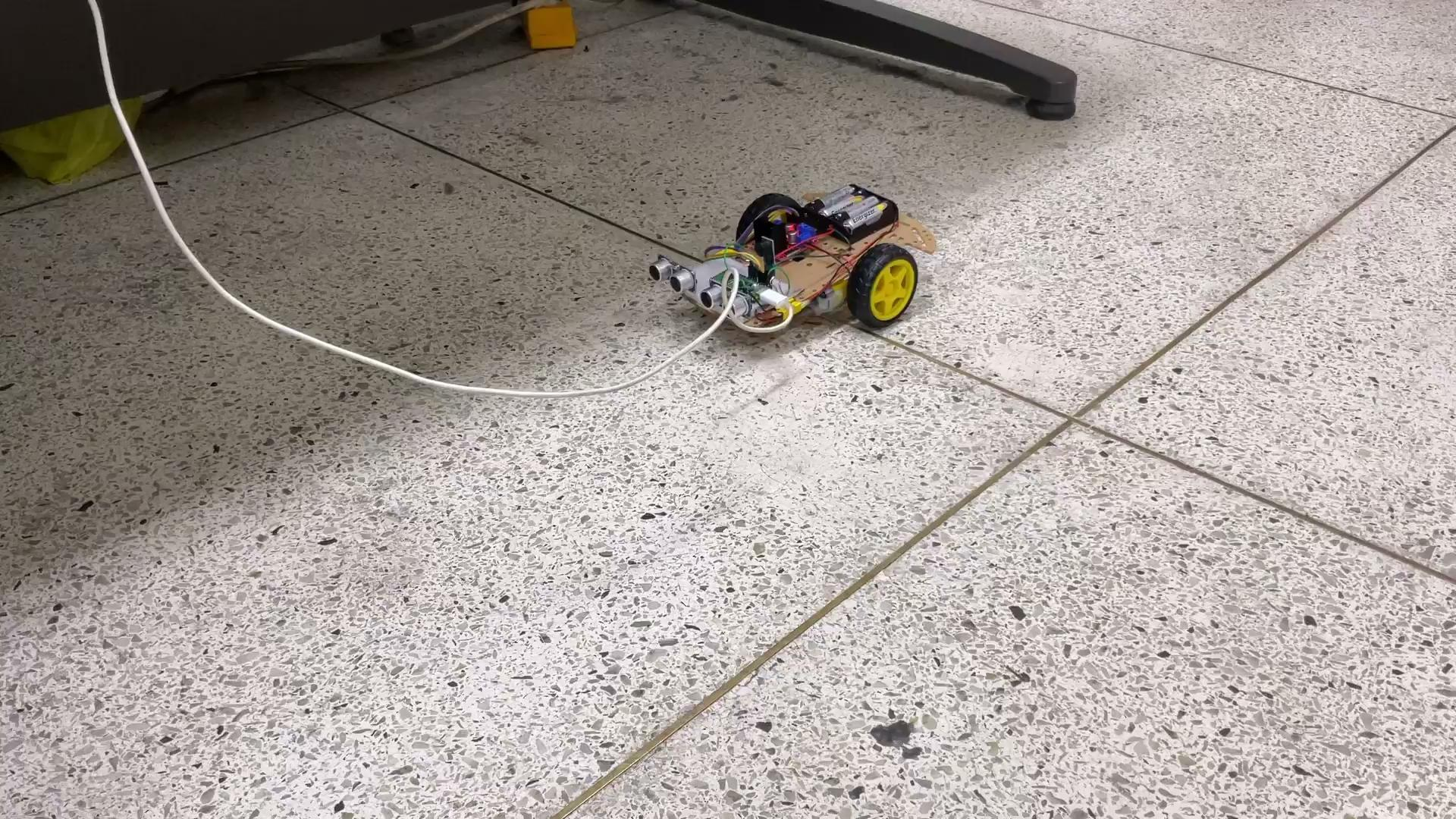
3. 실험 및 평가

Case1

PWM 50%

WIFi 양호

다음 영상을 통해 설명하겠습니다.



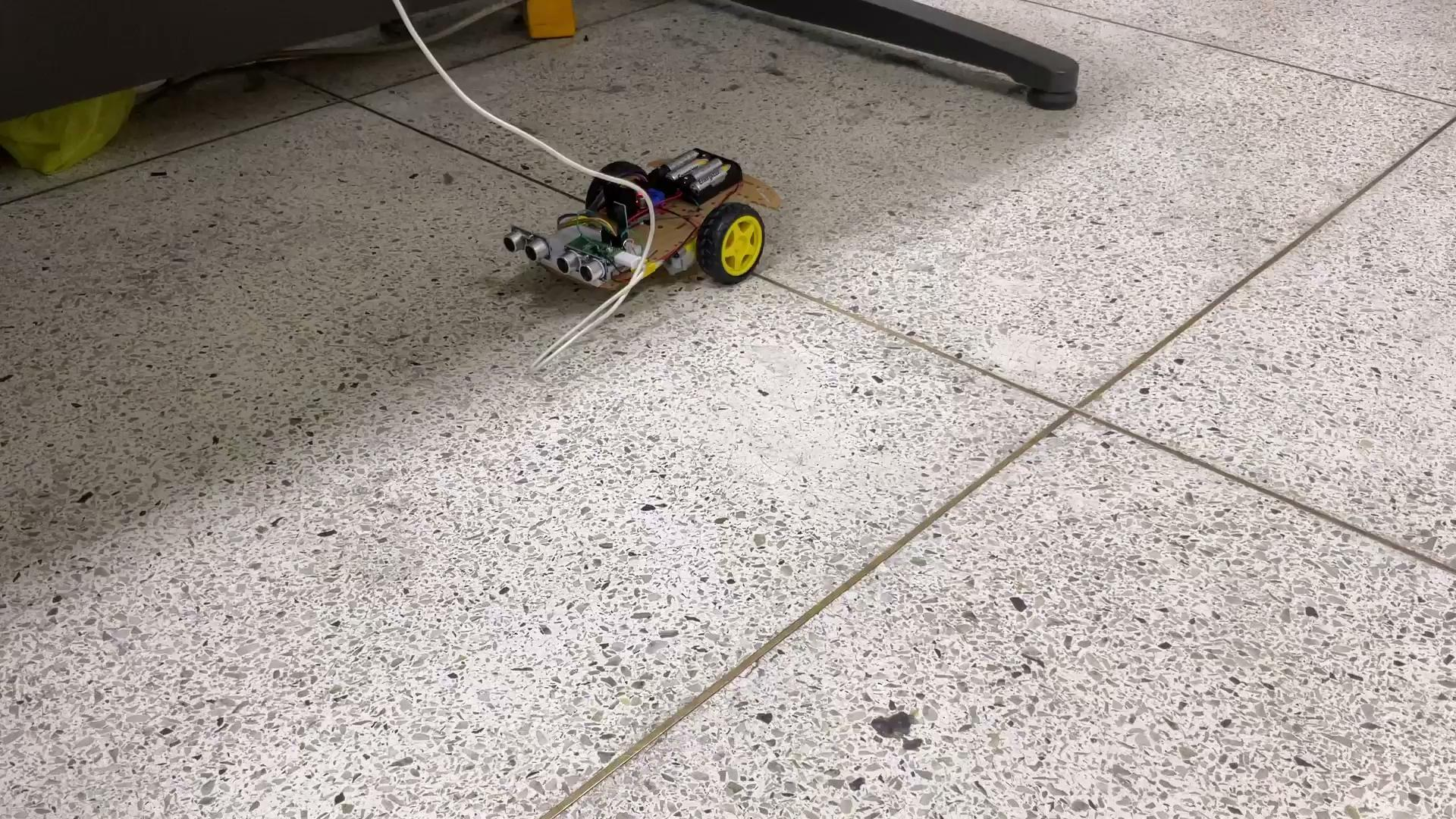
3. 실험 및 평가

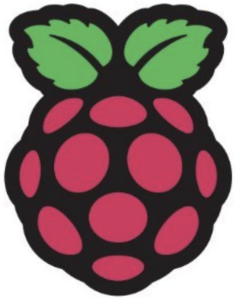
Case2

PWM 70%

WIFI 미흡

다음 영상을 통해 설명하겠습니다.





프로젝트 요약

한계와 의의

Computer Science and Engineering

4. 프로젝트 요약 - 의의

- 같은 Wifi IP를 공유하는 환경 내에서 무선조종이 가능했습니다.
- HTML 웹서버를 통해 시스템간 통신이 가능했습니다.
- 전,후,좌,우 와 같은 기본 동작에 문제가 없습니다.
- main.py를 통해 무선 환경에서 자동으로 구동 가능했습니다.
- PWM을 통해 모터를 안정적으로 컨트롤했습니다.

4. 프로젝트 요약 - 한계

- 기본적인 배터리-전류 설계에 문제가 있었습니다.
- 전압 문제로 인해 설계안대로 초음파센서를 활용하지 못했습니다.
- 메인보드에 안정적인 전류를 공급해줄 수 있는 가벼운 보조배터리가 필요했습니다.
- 사전에 준비한 부저버터, 카메라모듈, 아트메가128과 같은 하드웨어를 활용하지 못했습니다.

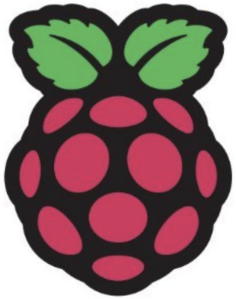
4. 프로젝트 요약 - 개선 가능성

- 카메라모듈을 활용한 이미지 처리, 데이터학습이 가능합니다.
  (RNN,CNN,YOLO등)
- 재 전류설계를 통해 초음파센서 두 개와 카메라모듈을 동시에 활용하여
  충돌예측, 표시 감지 자율주행시스템을 구현할 수 있습니다.

4. 프로젝트 요약 - 느낀점, QnA

이번 Embedded System 중간 평가 과제를 통해 전류설계의 중요성을 다시 체감하였습니다. 또한 시중에 파는 일반적인 4xAA RC카를 재평가하게 되었습니다.

QnA



감사합니다.

참고문헌

https://blog.naver.com/PostView.naver?blogId=roboholic84&logNo=221620337215&redirect=Dlog&widgetTypeCall=true&directAccess=false라즈베리파이로 DC모터, 스텝모터 제어하기

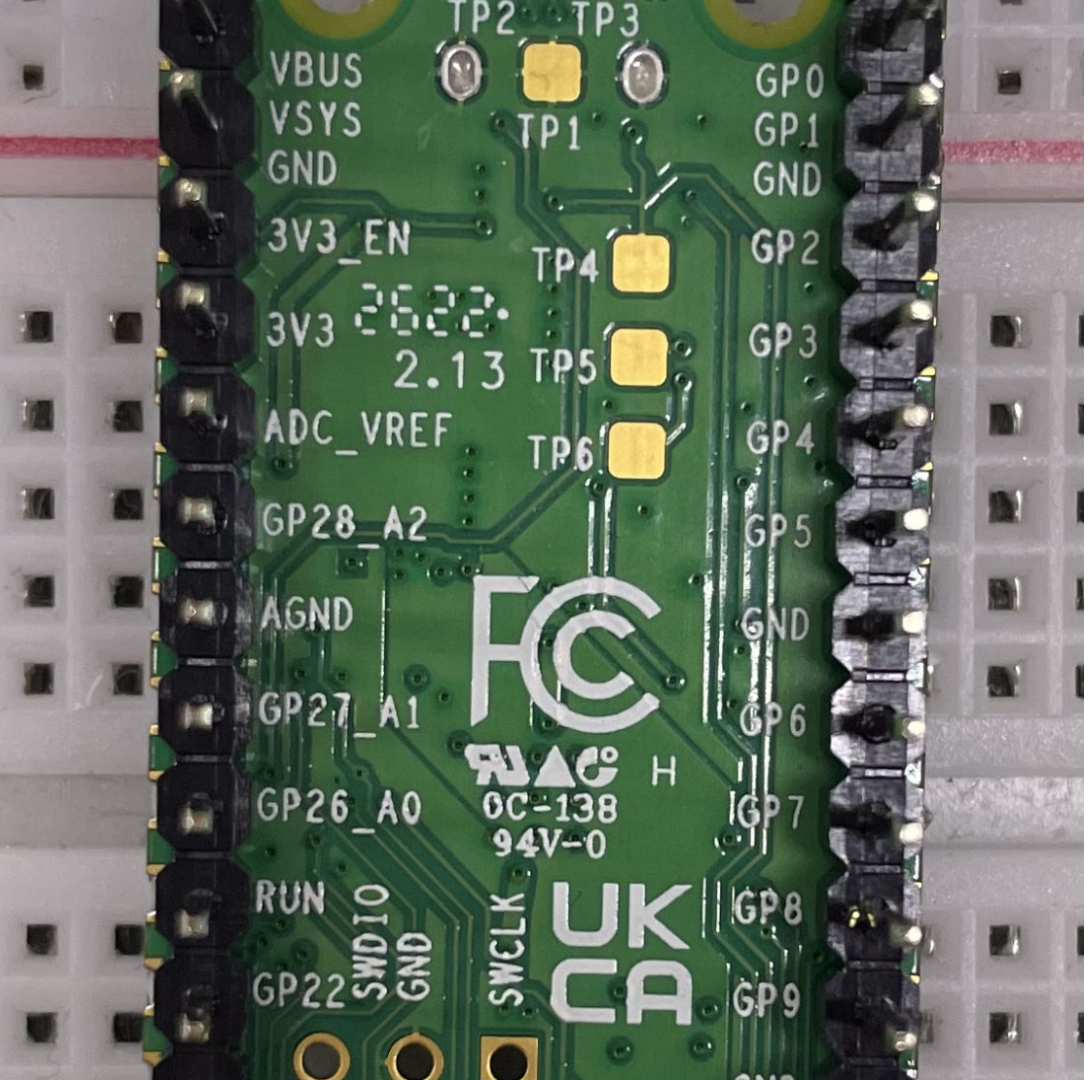
https://microcontrollerslab.com/hc-sr04-ultrasonic-sensor-raspberry-pi-pico-micropython-tutorial/HC-SR04 Ultrasonic Sensor with Raspberry Pi Pico using MicroPython

https://www.youtube.com/watch?v=I7IFsQ4tQU8 DC control tutorial

https://microcontrollerslab.com/hc-sr04-ultrasonic-sensor-raspberry-pi-pico-micropython-tutorial/ HC-SR04 Ultrasonic Sensor with Raspberry Pi Pico using MicroPython

https://blog.naver.com/no1_devicemart/221312208482 DiviceMart Official blog

https://chat.openai.com/ OpenAI GPT4, GPT3.5


TP2
TP3
VBUS
VSYS
GND
TP1
GP0
GP1
GND
3V3_EN
TP4
GP2
3V3
2622
2.13
TP5
GP3
ADC_VREF
TP6
GP4
GP28_A2
GP5
AGND
GND
GP27_A1
GP6
H
GP26_A0
0C-138
94V-0
GP7
RUN
GP8
GP22
SWDIO
GND
SWCLK
UK
CA
GP9