5-ДӘРІС

Аналогтық сигналдар және ADC

Кванттау, өлшеу дәлдігі және аналогтық датчиктер

әл-Фараби атындағы Қазақ ұлттық университеті

Физика-техникалық факультет

Дәрістің мақсаты және күтілетін нәтижелер

- Негізгі ұғымдарды түсіндіру: ADC, Кванттау, Сүзгілеу.
- Raspberry Pi Pico құрылғысында тиісті MicroPython API құралдарын қолдану.
- Аппараттық қосылымның қауіпсіздігін және бағдарламаның дұрыстығын бағалау.
- Өлшеу немесе басқару алгоритмін тәжірибелік міндетке бейімдеу.

ТЕОРИЯ

Негізгі ұғымдар және анықтамалар

ADC

Аналогтық кернеуді сандық кодқа түрлендіретін модуль.

Кванттау

Үздіксіз амплитуданы дискретті деңгейлермен бейнелеу.

Сүзгілеу

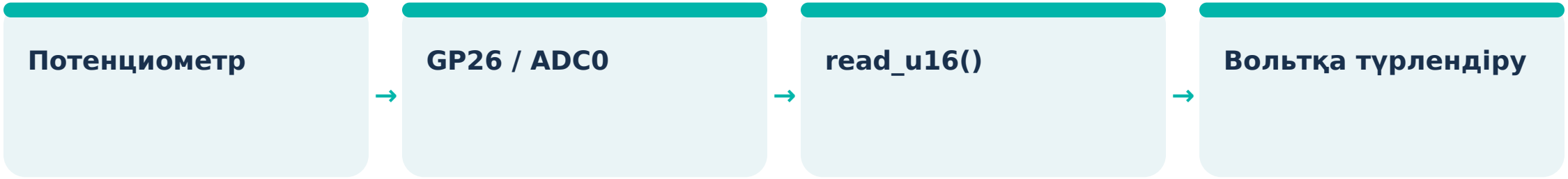
Өлшеудегі кездейсоқ ауытқуды азайтуға арналған өңдеу.

Теориялық негіздер: басты қағидалар

- RP2040 аппараттық ADC ажыратымдылығы 12 бит; MicroPython read_u16() нәтижені 0–65535 шкаласына келтіреді.
- ADC кірісіндегі кернеу 0–3,3 В аралығында болуы керек; GP26–GP28 сыртқы ADC арналарына қолданылады.
- Орташа мән шу әсерін азайтады, бірақ жылдам өзгерістерге жауапты баяулатады.

АРХИТЕКТУРА

Жүйенің функционалдық құрылымы



Блок-схема сигналдың немесе деректердің негізгі қозғалыс бағытын көрсетеді. Нақты қосылым пиндері және қоректендіру талаптары қолданылатын модульдің техникалық құжаттамасымен салыстырылады.

ТАЛДАУ

Аппараттық және бағдарламалық байланыс

Аппараттық қабат

GPIO, қоректендіру, датчик, шина және физикалық сигнал деңгейлері.

Бағдарламалық қабат

MicroPython кітапханалары, алгоритм, деректерді өңдеу және қате жағдайлары.

Инженерлік қағида: бағдарламаның дұрыс орындалуы электрлік қосылымның қауіпсіздігін автоматты түрде қамтамасыз етпейді.

ADC мәнін оқу

```
from machine import ADC, Pin
from time import sleep_ms
adc = ADC(Pin(26))
while True:
    raw = adc.read_u16()
    voltage = raw * 3.3 / 65535
    print(raw, round(voltage, 3))
    sleep_ms(200)
```

Ескерту: кодты іске қоспас бұрын пиндер мен модуль түрін тексеріңіз.

Орташа мәнді есептеу

```
from machine import ADC, Pin
adc = ADC(Pin(26))
def average(n=16):
    return sum(adc.read_u16() for _ in range(n)) / n
print(average())
```

Ескерту: кодты іске қоспас бұрын пиндер мен модуль түрін тексеріңіз.

Шекті мәнге жауап

```
from machine import ADC, Pin
adc = ADC(Pin(26))
led = Pin(15, Pin.OUT)
while True:
    led.value(adc.read_u16() > 32767)
```

Ескерту: кодты іске қоспас бұрын пиндер мен модуль түрін тексеріңіз.

Практикалық қолдану сценарийлері

- Потенциометрді 3V3, GND және GP26 желілеріне жалғау.
- Кернеу мен сандық код арасындағы тәуелділікті кестеге түсіру.
- Тәжірибе барысында алынған нәтижені кесте немесе қысқа техникалық есеп түрінде рәсімдеу.

Жиі кездесетін қателер және қауіпсіздік

- ADC кірісіне 3,3 В-тан жоғары кернеу беру.
- read_u16() мәнін тікелей 12 биттік нәтиже деп қабылдау.
- Құрылғыны жалғар алдында қорек көзін ажыратыңыз; 3,3 В логикалық деңгей шегін сақтаңыз.

Бақылау сұрақтары және талқылау

- 12 биттік ADC неше кванттау деңгейін береді?
- read_u16() нәтижесін кернеуге қалай аударамыз?
- Орташалау қандай артықшылық пен кемшілік береді?

Қорытынды және ұсынылатын әдебиеттер

Дәріс бойынша қорытынды: теориялық ұғымдар MicroPython коды және аппараттық жүйе жұмысы арқылы бекітіледі.

- Raspberry Pi Ltd. Raspberry Pi Pico-series documentation: raspberrypi.com/documentation/microcontrollers/
- Raspberry Pi Ltd. RP2040 Datasheet: datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf
- MicroPython documentation: docs.micropython.org/en/latest/rp2/quickref.html

- Қосымша: пайдаланылатын датчиктер мен модульдердің өндіруші дерекпарақтары.