4-ДӘРІС

Таймерлер, үзілістер және PWM

Уақытты басқару, оқиғаларды өңдеу және импульстік модуляция

әл-Фараби атындағы Қазақ ұлттық университеті

Физика-техникалық факультет

Дәрістің мақсаты және күтілетін нәтижелер

- Негізгі ұғымдарды түсіндіру: PWM, IRQ, Duty cycle.
- 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

PWM

Импульстің толтырылу коэффициенті арқылы орташа қуатты басқару.

IRQ

Оқиға туындағанда шақырылатын үзіліс өңдегіші.

Duty cycle

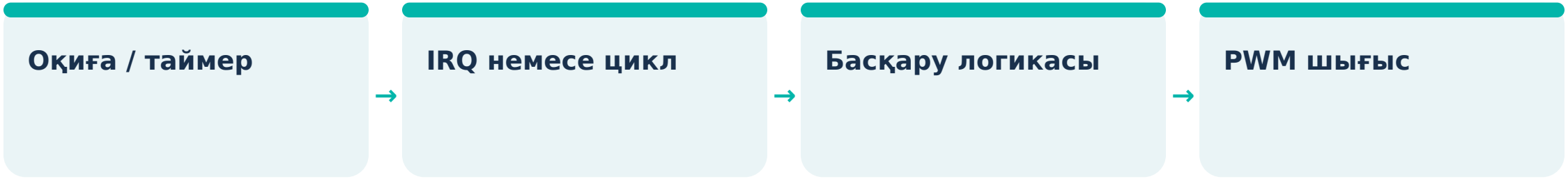
Импульс белсенді уақытының толық периодқа қатынасы.

Теориялық негіздер: басты қағидалар

- PWM жиілігі және толтырылу коэффициенті — өзара бөлек параметрлер.
- Үзіліс өңдегіші қысқа әрі блоктамайтын болуы керек; күрделі есептеуді негізгі циклге беру дұрыс.
- time.ticks_ms() монотондық уақыт аралығын өлшеуге мүмкіндік береді.

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.

PWM арқылы жарықтық

```
from machine import Pin, PWM
from time import sleep_ms
pwm = PWM(Pin(15))
pwm.freq(1000)
for duty in range(0, 65536, 1024):
    pwm.duty_u16(duty)
    sleep_ms(20)
```

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

Батырма үзілісі

```
from machine import Pin
button = Pin(14, Pin.IN, Pin.PULL_UP)
flag = False
def on_press(pin):
    global flag
    flag = True
button.irq(trigger=Pin.IRQ_FALLING, handler=on_press)
```

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

Блоктамайтын тайминг

```
from time import ticks_ms, ticks_diff
from machine import Pin
led = Pin("LED", Pin.OUT)
last = ticks_ms()
while True:
    now = ticks_ms()
    if ticks_diff(now, last) >= 500:
        led.toggle()
        last = now
```

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

Практикалық қолдану сценарийлері

- PWM толтырылуын 10%, 50%, 90% өзгертіп, LED жарықтығын салыстыру.
- Батырма басылуын IRQ арқылы тіркеп, дірілді (debounce) талдау.
- Тәжірибе барысында алынған нәтижені кесте немесе қысқа техникалық есеп түрінде рәсімдеу.

Жиі кездесетін қателер және қауіпсіздік

- IRQ ішінде ұзақ sleep() шақыру.
- PWM жиілігін сервомотор үшін дұрыс таңдамау.
- Құрылғыны жалғар алдында қорек көзін ажыратыңыз; 3,3 В логикалық деңгей шегін сақтаңыз.

Бақылау сұрақтары және талқылау

- PWM жиілігі мен duty cycle айырмашылығы қандай?
- IRQ өңдегіші неге қысқа болуы тиіс?
- ticks_diff() не үшін керек?

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.