2-ДӘРІС

MicroPython ортасы және бағдарламалау негіздері

Thonny IDE, айнымалылар, басқару құрылымдары және функциялар

әл-Фараби атындағы Қазақ ұлттық университеті

Физика-техникалық факультет

Дәрістің мақсаты және күтілетін нәтижелер

- Негізгі ұғымдарды түсіндіру: MicroPython, REPL, main.py.
- 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

MicroPython

Микроконтроллер ресурстарына бейімделген Python 3 іске асырылымы.

REPL

Командаларды интерактивті енгізу және нәтижені бірден бақылау ортасы.

main.py

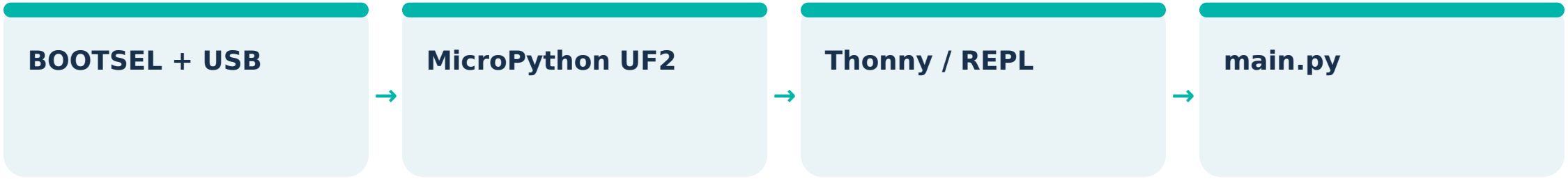
Құрылғы жүктелгенде автоматты орындалатын пайдаланушы бағдарламасы.

Теориялық негіздер: басты қағидалар

- UF2 микробағдарламасын BOOTSEL режимінде орнату; Thonny-де MicroPython (Raspberry Pi Pico) интерпретаторын таңдау.
- Python шегінісі синтаксистің бөлігі: if, for, while және def блоктары бірізді жазылады.
- Құрылғыдағы файлдар мен компьютердегі файлдарды ажырату; main.py қателерін REPL арқылы түзету.

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.

Алғашқы бағдарлама

```
print("Salem, Pico!")
from machine import Pin
led = Pin("LED", Pin.OUT)
led.on()
```

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

Цикл және шарт

```
from machine import Pin
from time import sleep
led = Pin("LED", Pin.OUT)
for i in range(5):
    led.toggle()
    sleep(0.5)
```

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

Функцияларды қолдану

```
from machine import Pin
from time import sleep_ms
led = Pin("LED", Pin.OUT)
def blink(count, period_ms):
    for _ in range(count):
        led.on(); sleep_ms(period_ms)
        led.off(); sleep_ms(period_ms)
blink(3, 300)
```

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

Практикалық қолдану сценарийлері

- REPL-де арифметикалық өрнек пен айнымалы мәнін тексеру.
- main.py файлын сақтап, қайта жүктегеннен кейін орындалуын бақылау.
- Тәжірибе барысында алынған нәтижені кесте немесе қысқа техникалық есеп түрінде рәсімдеу.

Жиі кездесетін қателер және қауіпсіздік

- Thonny интерпретаторының қате таңдалуы.
- Шегіністердің біркелкі болмауы және шексіз циклден шыға алмау.
- Құрылғыны жалғар алдында қорек көзін ажыратыңыз; 3,3 В логикалық деңгей шегін сақтаңыз.

Бақылау сұрақтары және талқылау

- REPL мен main.py орындалуының айырмашылығы қандай?
- MicroPython-та импорт не үшін қолданылады?
- Құрылғыда бағдарламаны іске қосудың қандай тәсілдері бар?

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.