9-ДӘРІС

Raspberry Pi Pico W және IoT

Wi-Fi, HTTP, MQTT және желілік деректер қауіпсіздігі

әл-Фараби атындағы Қазақ ұлттық университеті

Физика-техникалық факультет

Дәрістің мақсаты және күтілетін нәтижелер

- Негізгі ұғымдарды түсіндіру: IoT, HTTP, MQTT.
- 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

IoT

Физикалық құрылғылардың желі арқылы деректер алмасуы.

HTTP

Клиент–сервер сұраныс-жауап протоколы.

MQTT

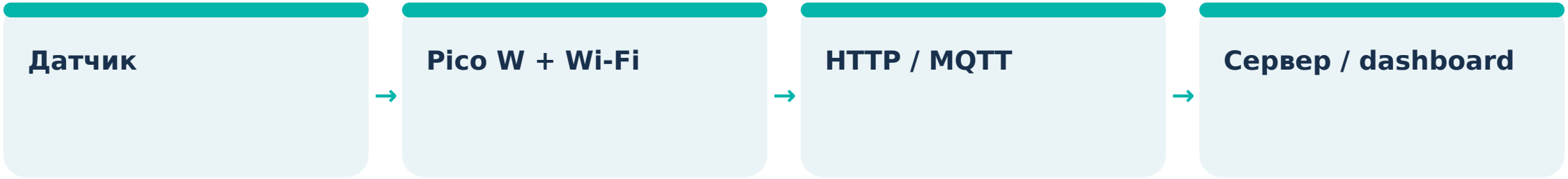
Publish/subscribe моделіндегі жеңіл хабар алмасу протоколы.

Теориялық негіздер: басты қағидалар

- Wi-Fi тек Raspberry Pi Pico W/WH нұсқаларында бар; кәдімгі Pico-да кіріктірілген Wi-Fi жоқ.
- Қосылу алдында WLAN.active(True) және connect() шақырылады; желі күйін тексеру керек.
- Құпиясөздерді жария репозиторийге жазбау; TLS, аутентификация және желілік үзілістерді өңдеу қажет.

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.

Wi-Fi желісіне қосылу

```
import network, time
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect("SSID", "PASSWORD")
for _ in range(20):
    if wlan.isconnected(): break
    time.sleep(0.5)
print(wlan.ifconfig() if wlan.isconnected() else "Жоқ")
```

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

HTTP GET сұранысы

```
import urequests
# Wi-Fi қосылған болуы тиіс
response = urequests.get("http://example.com")
print(response.status_code)
response.close()
```

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

MQTT жариялау үлгісі

```
from umqtt.simple import MQTTClient
# umqtt.simple модулі қажет
client = MQTTClient("pico-demo", "broker.example.com")
# Сервер, TLS және аутентификацияны баптаңыз
# client.connect()
# client.publish(b"lab/temp", b"24.1")
```

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

Практикалық қолдану сценарийлері

- Pico W құрылғысын жергілікті Wi-Fi желісіне қосу.
- Желілік қате болғанда қайта қосылу стратегиясын сипаттау.
- Тәжірибе барысында алынған нәтижені кесте немесе қысқа техникалық есеп түрінде рәсімдеу.

Жиі кездесетін қателер және қауіпсіздік

- Pico мен Pico W мүмкіндіктерін шатастыру.
- Құпиясөзді кодта ашық қалдыру және қорғалмаған брокер қолдану.
- Құрылғыны жалғар алдында қорек көзін ажыратыңыз; 3,3 В логикалық деңгей шегін сақтаңыз.

Бақылау сұрақтары және талқылау

- HTTP пен MQTT қандай сценарийлерде тиімді?
- IoT құрылғысы желіден ажыраған да не істеуі керек?
- Pico W-дің Pico-дан негізгі айырмашылығы қандай?

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.