Электромобильдерге арналған микроконтроллерлік жүйелерді бағдарламалау

Дәуренов ӘД

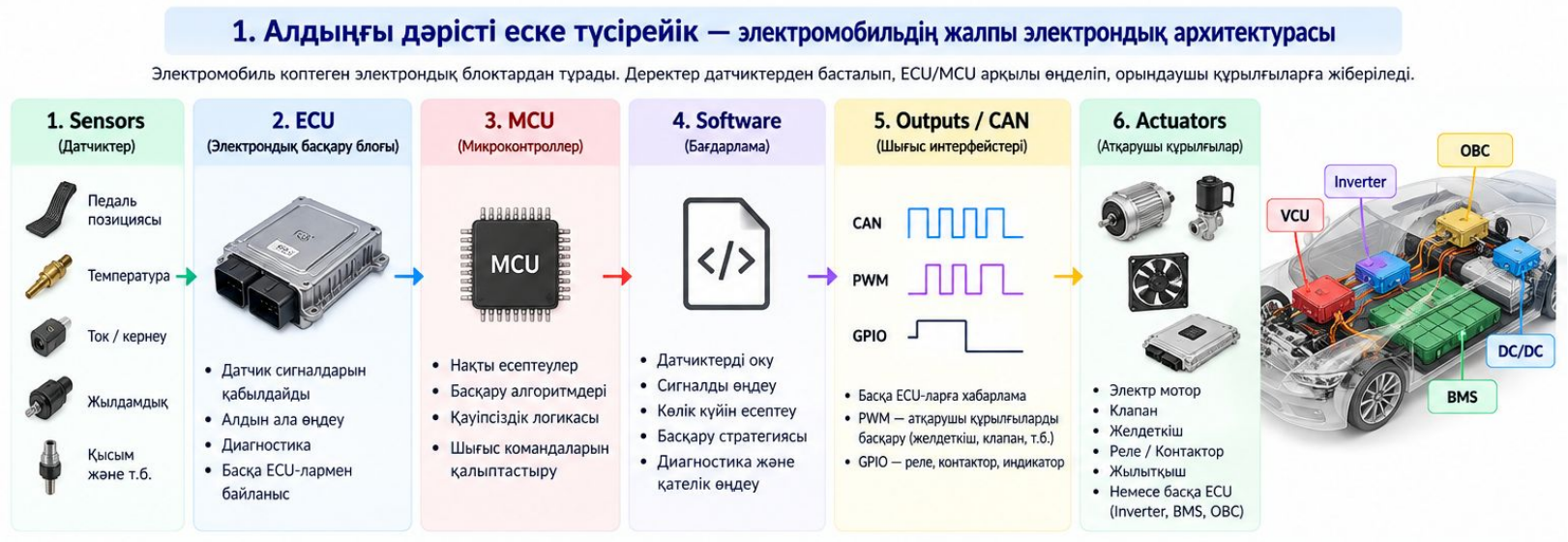
Дәріс №2. Автомобильдік микроконтроллерлердің архитектурасы

Дәрістің мақсаты

Студенттерді автомобильдік микроконтроллердің ішкі архитектурасымен таныстыру және оның жеке блоктарының электромобильді басқару кезінде қандай міндет атқаратынын түсіндіру.

Дәріс соңында студент:

- MCU архитектурасын түсіндіре алады;
- CPU, Flash, RAM және регистрлердің қызметін ажырата алады;
- микроконтроллер периферияларының қызметін түсінеді;
- Clock, Reset және Watchdog қызметін біледі;
- MCU және MPU арасындағы айырмашылықты түсінеді;
- real-time ұғымын түсіндіре алады;
- automotive-grade MCU ерекшеліктерін біледі;
- NXP S32K, Infineon AURIX және Renesas RH850 сияқты нақты автомобильдік MCU отбасыларын таниды.

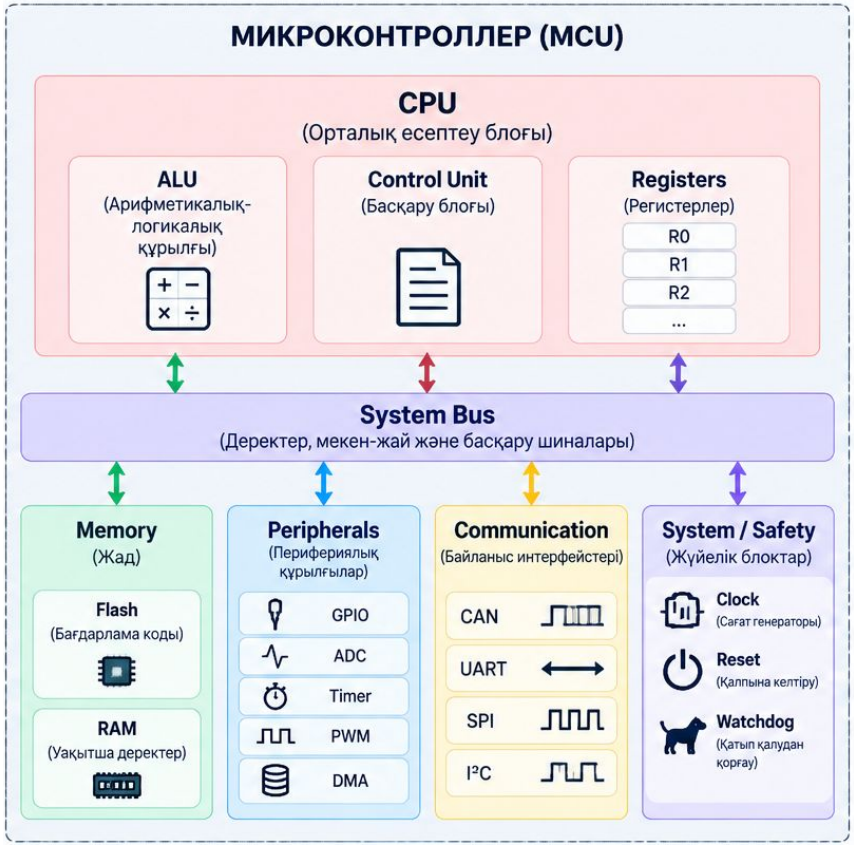


Біз MCU-ды ECU-дың есептеу ядросы деп қарастырдық.

Енді келесі сұрақ туындайды:

MCU ішінде нақты не бар және ол бағдарламаны қалай орындайды?

2. Микроконтроллер дегеніміз не?



MCU (Microcontroller Unit) — процессорды, жадты және перифериялық құрылғыларды бір микросхема ішінде біріктіретін бағдарламаланатын электрондық құрылғы.

MCU электромобильде не істейді?

- **температураны өлшеу;**
- **CAN хабарламаларын қабылдау;**
- **басқару алгоритмін орындау;**
- **ақауларды анықтау;**
- **PWM сигналын генерациялау;**
- **CAN арқылы басқа ECU-ларға хабарлама жіберу.**

MCU-дың негізгі блоктары

Микроконтроллердің (MCU) негізгі құрылымдық блоктары оның бір кристаллда орналасқан толыққанды шағын компьютер ретінде жұмыс істеуін қамтамасыз етеді.

MCU келесі негізгі блоктардан тұрады:

- **Орталық процессорлық блок (CPU):** Микроконтроллердің «миы». Ол бағдарлама нұсқауларын орындайды, математикалық және логикалық амалдарды жүргізеді және басқа блоктардың жұмысын басқарады.
- **Жады (Memory):** Екі негізгі түрге бөлінеді:
  - Flash (немесе ROM): Бағдарлама коды мен тұрақты деректер сақталатын энергияға тәуелсіз жады.
  - RAM (Жедел жады): Процессор жұмыс істеу барысында уақытша деректерді сақтауға арналған жылдам, бірақ энергияға тәуелді жады.
- **Енгізу-шығару порттары (GPIO - General Purpose Input/Output):** Сыртқы әлеммен (түймелер, жарық диодтары, датчиктер) байланыс орнатуға арналған істікшелер (пиндер).
- **Таймерлер және санауыштар (Timers/Counters):** Уақыт аралықтарын өлшеуге, сигналдарды санауға және ШИМ (PWM) сигналдарын қалыптастыруға жауапты.

- **Аналогты-цифрлық және цифрлы-аналогты түрлендіргіштер (ADC / DAC):** Сыртқы аналогты сигналдарды (мысалы, температура датчигінен келетін кернеуді) процессор түсінетін цифрлық кодқа (ADC) немесе керісінше (DAC) айналдырады.
- **Байланыс интерфейстері (Serial Communication):** Басқа құрылғылармен деректер алмасуға арналған хаттамалар блоктары: **UART**, **SPI**, **I2C**, **CAN** немесе **USB**.
- **Үзілімдерді басқару жүйесі (Interrupt Controller):** Сыртқы немесе ішкі маңызды оқиғалар болғанда, негізгі бағдарламаны уақытша тоқтатып, шұғыл тапсырманы орындауға мүмкіндік береді.
- **Тактілік генератор (Clock Circuit/Oscillator):** Микроконтроллердің жұмыс істеу жылдамдығын (тактілік жиілігін) анықтайтын және барлық блоктарды синхрондайтын жүйе.

Микроконтроллердің (MCU) негізгі құрылымдық блоктары

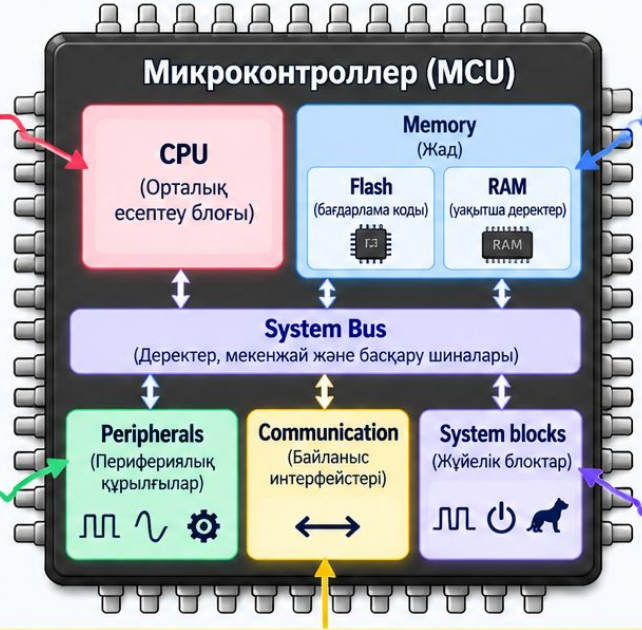
MCU — процессорды, жадты, перифериялық құрылғыларды және басқару блоктарын бір кристаллда біріктіретін толыққанды шағын компьютер.

1 CPU (Орталық процессорлық блок)

- Бағдарлама нұсқауларын орындайды.
- Арифметикалық және логикалық амалдарды жүзеге асырады.
- Басқа блоктардың жұмысын басқарады.

CPU

- **ALU** (есептеу)
- **Control Unit** (басқару)
- **Registers** (регистрлер): R0, R1, R2, ...



2 Жад (Memory)

- Бағдарлама мен деректерді сақтайды.
- **Flash** — бағдарлама коды (энергия өшірілгенде де сақталады).
- **RAM** — уақытша деректер (жұмыс кезінде қолданылады).

Flash (бағдарлама коды)



Энергияға тәуелсіз (тұрақты жады)

RAM (уақытша деректер)



Жұмыс кезінде қолданылады (жылдам, бірақ энергияға тәуелді)

3 Перифериялық құрылғылар (Peripherals)

- Сыртқы әлеммен жұмыс істейді (датчиктер мен атқарушылар).
- Нақты уақыттық функцияларды орындайды.
- Аналогты/цифрлық сигналдарды өңдейді.

- **GPIO** (кіру/шығыс)
- **ADC** (аналогтық кіріс)
- **Timer** (таймерлер)
- **PWM** (ШИМ)
- **DMA** (тікелей жадыға көшіру)

4 Байланыс интерфейстері (Communication)

- Басқа құрылғылармен деректер алмасуға арналған.
- Автомобильдегі басқа ECU-лармен, датчиктермен, модульдермен байланысады.

- **CAN** (автомобиль шинасы)
- **UART** (сериялық байланыс)
- **SPI** (жылдам деректер алмасу)
- **I²C** (сенсорлармен байланыс)
- **USB** (қажет болса)

5 Жүйелік блоктар (System blocks)

- **Clock** — тактілік жиілікті береді.
- **Reset** — жүйені бастапқы күйге келтіреді.
- **Watchdog** — бағдарламаның “қатып қалуын” бақылайды.

- **Clock** (сағат генераторы)
- **Reset** (қалпына келтіру)
- **Watchdog** (қатып қалудан қорғау)

MCU = CPU + Memory + Peripherals + Communication + System blocks

бір кристаллда орналасқан толыққанды шағын компьютер.

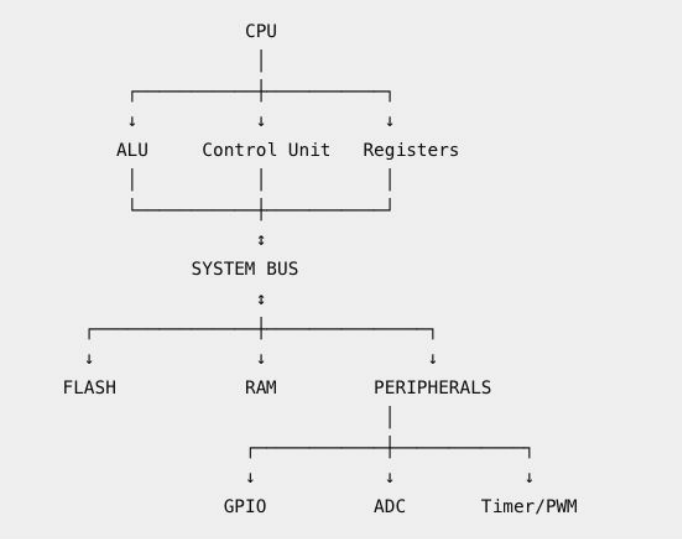
CPU — Central Processing Unit

CPU (Central Processing Unit) — орталық процессорлық блок. Ол микроконтроллердегі бағдарламаның командаларын орындайтын негізгі есептеу блогы.

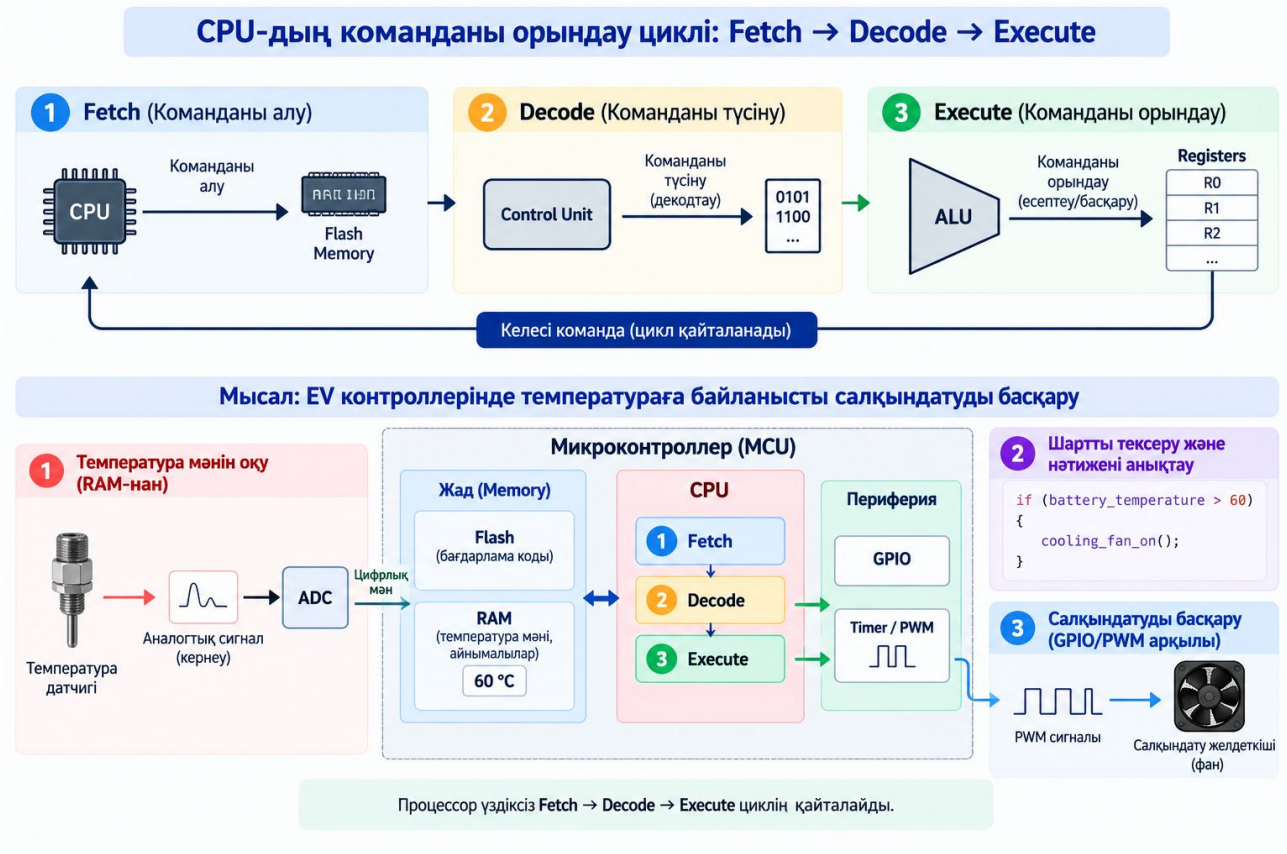
CPU-дың негізгі бөліктері:

- **ALU (Arithmetic Logic Unit)** — арифметикалық және логикалық операцияларды орындайды.
- **Control Unit** — командалардың орындалуын басқарады.
- **Registers** — CPU ішінде деректерді уақытша сақтайтын өте жылдам жады.

Flash және RAM — MCU жады, олар CPU-дың құрамына кірмейді. **CPU жадымен және перифериялық құрылғылармен System Bus (жүйелік шина) арқылы өзара әрекеттеседі.**

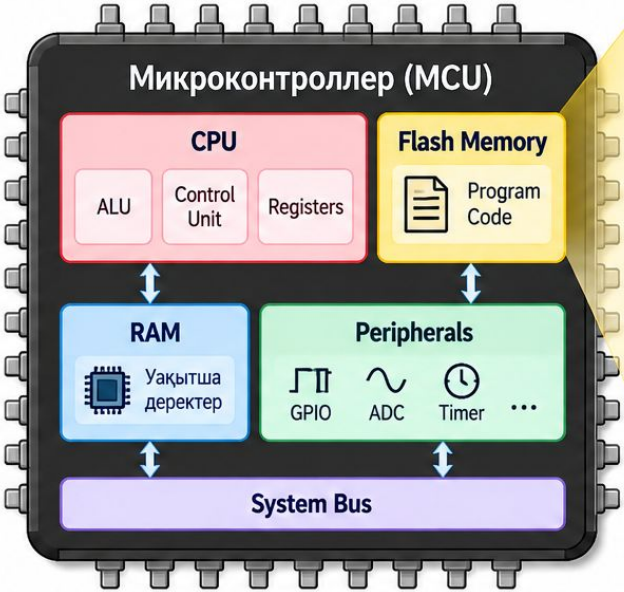


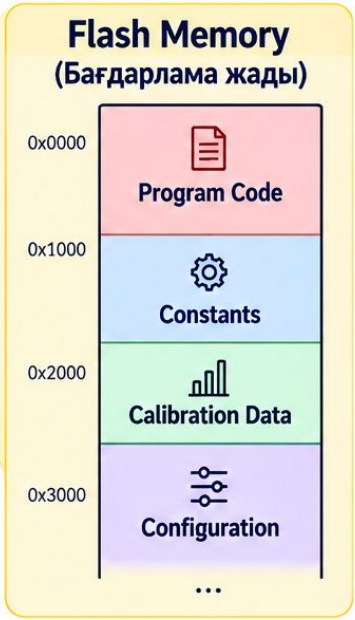
CPU бағдарламаны орындау кезінде **Fetch → Decode → Execute** циклін үздіксіз қайталайды·



7. Flash Memory

Flash — энергияға тәуелсіз жады. Қуат өшіріглен кезде оның ішіндегі ақпарат жоғалмайды.





Flash ішінде сақталатын негізгі мәліметтер

Program Code (Бағдарлама коды)
- Микроконтроллердің негізгі бағдарламасы
- Функциялар, командалар

Constants (Тұрақты мәндер)
- Тұрақты айнымалылар
- Кестелер (lookup table)
- Параметрлер

Calibration Data (Калибрлеу деректері)
- Датчиктерді калибрлеу мәліметтері
- Өндірістік параметрлер

Configuration (Конфигурация)
- Жүйе параметрлері
- Орнатулар (settings)
- Пайдаланушы конфигурациясы

Мысал: Flash-та сақталатын тұрақты мәндер

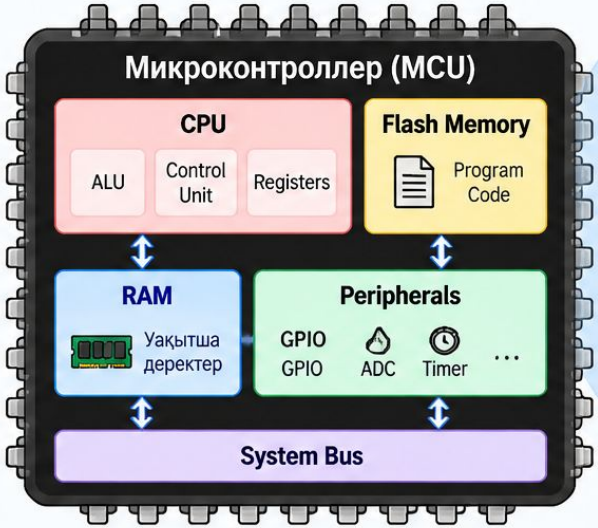
```
#define MAX_TEMP 60       // Максималды температура (°C)
#define MAX_VOLTAGE 420   // Максималды кернеу (V)
```

Жұмыс принципі

- Микроконтроллер қосылған кезде CPU бағдарлама кодын Flash жадыдан оқып, орындала бастайды.
- Flash жадының мазмұны қуат өшірігленде жоғалмайды (энергияға тәуелсіз жады).
- Flash әдетте оқу кезінде қолданылады, жазу/өшіру сирек жүргізіледі (шектеулі цикл саны бар).

8. RAM — Random Access Memory (Жедел жады)

RAM — бағдарламаның жұмысы кезінде қолданылатын уақытша жады.



RAM (Жедел жады)

Адрес	Деректер	
0x2000 0000	...	
0x2000 0004	temperature (float)	34.7
0x2000 0008	accelerator (float)	42.5
0x2000 000C	battery_voltage (float)	382.1
0x2000 0010	motor_speed (uint16_t)	3150
...	...	...

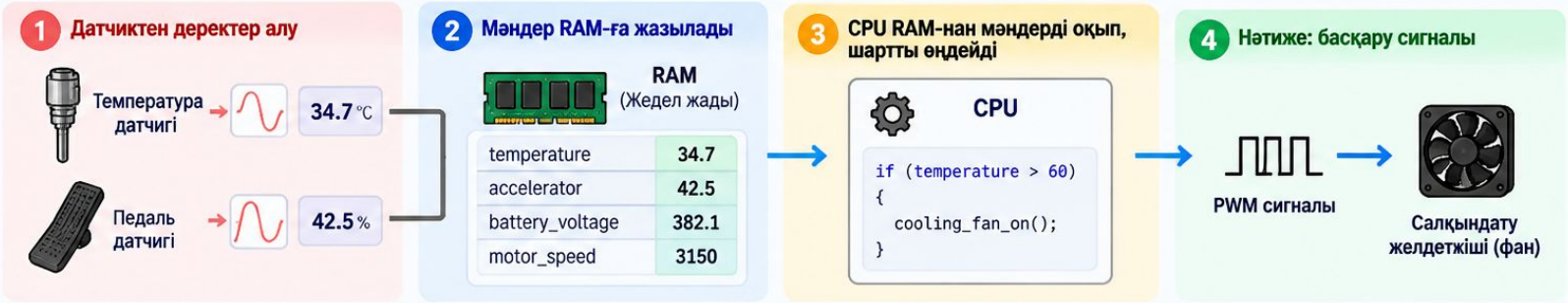
Мысал: RAM ішіндегі айнымалылар

```
float temperature;
float accelerator;
float battery_voltage;
uint16_t motor_speed;
```

RAM ішіндегі мәндер (жұмыс кезінде)

```
temperature     = 34.7
accelerator     = 42.5
battery_voltage = 382.1
motor_speed     = 3150
```

Жұмыс процесі: датчиктен алынған мәндер RAM-да сақталады және CPU оларды пайдаланады



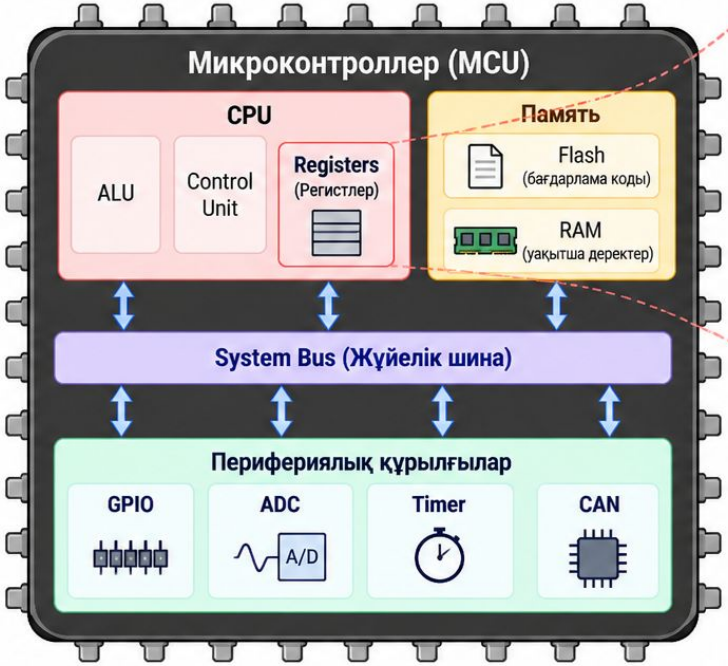
9. Flash пен RAM айырмашылығы

	Flash	RAM
Бағдарлама сақтала ма?	Иә	Әдетте орындалу кезіндегі деректер
Қуат өшкенде ақпарат	Сақталады	Жоғалады
Негізгі мақсаты	Code / тұрақты деректер	Уақытша деректер
EV мысалы	VCU firmware	Ағымдағы sensor values

Қарапайым аналогия: FLASH – "Не істеу керек?" RAM — "Қазір қандай мәндер бар?"

10. Регистр дегеніміз не?

Регистр — CPU немесе перифериялық құрылғы тікелей жұмыс істейтін өте жылдам сақтау элементі. Микроконтроллердің көптеген аппараттық блоктары регистрлер арқылы басқарылады.



CPU регистрлері (мысал)

R0	0x2000 0010
R1	0x0000 003C
R2	0x0000 0001
R3	0x0800 1234
...	...

Периферийлық регистрлер (мысалдар)

GPIO Register	0x4002 0000
ADC Register	0x4001 2400
Timer Register	0x4000 1000
CAN Register	0x4000 6400

Регистермен жұмыс істеу мысалы

```
// GPIO регистріне жазу
#define GPIO_BASE    0x40020000
#define GPIO_ODR     (GPIO_BASE + 0x14)

uint32_t value = 0x01;      // LED қосу
*(volatile uint32_t *)GPIO_ODR = value;
```



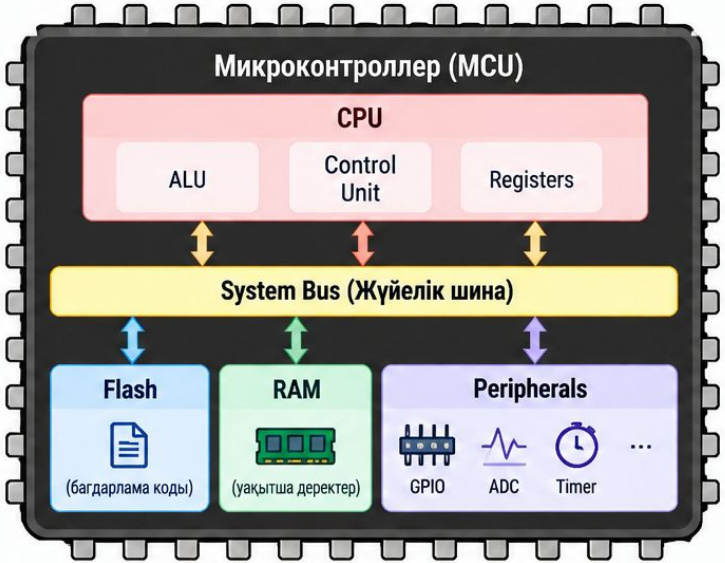
Бағдарлама регистр арқылы аппараттық блоктың жұмысын өзгертеді



Software (Бағдарлама) → **Регистерге жазу** (оқу/жазу) → **Аппараттық блок** (периферия) → **Нәтиже** (LED қосу, таймерді іске қосу, деректерді оқу және т.б.)

12. System Bus — жүйелік шина

CPU, Memory және Peripherals бір-бірімен байланысуы керек.
Ол үшін микроконтроллер ішінде **жүйелік шиналар** қолданылады.



Жүйелік шина не істейді?

System Bus — MCU ішіндегі компоненттердің ақпарат алмасу жолы.

Адрес шинасы (Address Bus)
Қай компонентке/адреске қол жеткізу керегін көрсетеді.

Деректер шинасы (Data Bus)
Оқылатын немесе жазылатын деректерді тасымалдайды.

Басқару сигналдары (Control Signals)
Оқу/жазу, синхронизация және басқа басқару сигналдарын тасымалдайды.

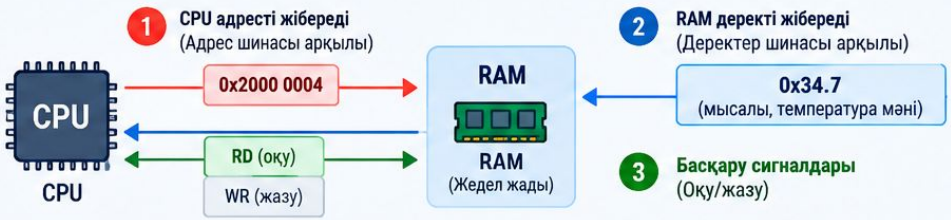
Шиналардың түрлері

Адрес шинасы (Address Bus)	n бит (мысалы, 32 бит)
Деректер шинасы (Data Bus)	m бит (мысалы, 32 бит)
Басқару сигналдары (Control Signals)	Оқу/жазу, үзілістер, такт және т.б.

Қарапайым түрде:

Bus — MCU ішіндегі компоненттердің ақпарат алмасу жолы.

Мысал: CPU RAM-нан дерек оқу

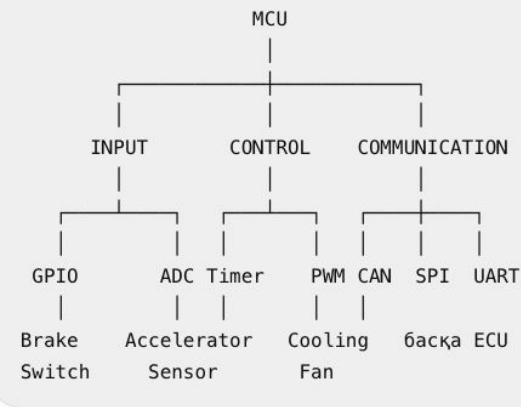


Жалпы жұмыс принципі

1. CPU адрес шинасы арқылы қажетті адрес жібереді.
2. Компонент (RAM, Flash немесе периферия) деректер шинасы арқылы деректі қайтарады немесе қабылдайды.
3. Басқару сигналдары операция түрін анықтайды (оқу/жазу және т.б.).

Перифериялық модульдер

Перифериялық модульдер (Peripherals) — микроконтроллердің сыртқы құрылғылармен және сигналдармен жұмыс істеуіне мүмкіндік беретін аппараттық блоктар.



Модуль	Негізгі қызметі	EV-дегі мысал
GPIO	Цифрлық кіріс/шығыс	Тежегіш қосқышының күйін оқу
ADC	Аналогтық сигналды цифрлық мәнге түрлендіру	Акселератор педалінің сигналын оқу
Timer	Уақытты өлшеу, периодтық оқиғаларды ұйымдастыру	Датчикті әр 10 мс сайын оқу
PWM	Импульс енін басқару	Салқындату желдеткішін басқару
CAN	ECU блоктары арасында дерек алмасу	VCU ↔ BMS ↔ Inverter
UART	Тізбекті дерек алмасу	Диагностика
SPI	Жоғары жылдамдықты құрылғыаралық байланыс	Сыртқы сенсор/IC
I²C	Бір шинадағы бірнеше құрылғымен байланыс	Сенсорлар, EEPROM

CPU есептеуді орындайды, ал перифериялық модульдер MCU-дың нақты физикалық әлеммен байланысын қамтамасыз етеді.

Clock System

Clock System (немесе **Тактылық жүйе**) — бұл қолданылу саласына байланысты бірнеше негізгі мағынаны білдіретін ұғым

Микроконтроллерлер мен процессорлардағы **Clock System** — бұл бүкіл чиптің жұмысын үндестіретін (синхрондайтын) және оған белгілі бір ырғақ (жиілік) беретін **ішкі аппараттық модуль**. Оны құрылғының «жүрек соғысы» деп атауға болады.

- **Ол не үшін керек?** Процессор ядросы, жедел жады (RAM) және таймерлер сияқты ішкі бөліктердің барлығы бір уақытта, қатесіз әрі ретімен жұмыс істеуі үшін арнайы тактылық импульстер жіберіп отырады.
- **Дереккөздері:** Жиілік импульстерін алу үшін ішкі RC-генераторлар немесе өте дәл жұмыс істейтін сыртқы кварцтық резонаторлар (crystal) қолданылады.
- **Мысалы:** STM32 микроконтроллерлерінде бұл жүйені **RCC** (Reset and Clock Control) модулі басқарады.

Reset System

Кейде микроконтроллерді бастапқы күйге қайтару қажет.

Бұл үшін **Reset System** қолданылады.

Reset келесі жағдайларда болуы мүмкін:

Power ON — RESET немесе Software Error — Watchdog — RESET

Reset орындалғаннан кейін MCU бағдарламаны белгіленген бастапқы нүктеден қайта бастайды.

Watchdog — бақылаушы таймер

Watchdog (бақылаушы таймер) — жүйенің дұрыс жұмыс істеуін бақылайтын және бағдарлама тұрып қалған жағдайда жүйені автоматты түрде қайта іске қосатын аппараттық немесе бағдарламалық механизм. Watchdog-тың негізгі міндеті — **адамның араласуынсыз жүйені қайтадан жұмыс күйіне келтіру**. Бұл серверлерде, банкоматтарда, ақылды үй жүйелерінде, ғарыш аппараттарында және қашықтан жұмыс істейтін автоматтандыру жүйелерінде өте маңызды.

Watchdog қалай жұмыс істейді?

Watchdog жұмысын үнемі қайта орнатылып отыратын таймер ретінде қарастыруға болады:

1. **Таймерді іске қосу:** жүйе кері санауды бастайды, мысалы, 60 секунд.
2. **Таймерді жаңарту (Reset/Refresh):** негізгі бағдарлама дұрыс жұмыс істеп тұрса, ол Watchdog таймерін белгілі бір уақыт сайын қайта жаңартып отырады.
3. **Бағдарламаның тұрып қалуы:** егер бағдарламада ақау пайда болып, ол орындалуын тоқтатса, Watchdog таймері жаңартылмайды.
4. **Қайта іске қосу:** таймер белгіленген уақытқа жеткенде, Watchdog **Reset** сигналын қалыптастырып, жүйені қайта іске қосады.

Interrupt — үзіліс

Interrupt (үзіліс) — процессорға **шұғыл назар аударуды қажет ететін маңызды оқиға болғанын хабарлайтын сигнал**.Мұндай сигнал келген кезде процессор ағымдағы бағдарламаның орындалуын **уақытша тоқтатады**, оның күйін сақтайды және осы оқиғаны өңдеуге арналған арнайы кодқа — **үзіліс өңдеушісіне (Interrupt Service Routine, ISR)** ауысады. Үзіліс өңделгеннен кейін процессор бұрынғы бағдарламаның орындалуын жалғастырады.



Қарапайым өмірлік мысал

Сіз **кітап оқып отырсыз** — бұл процессордың негізгі бағдарламаны орындауына ұқсайды.

1. Телефон кенеттен **шырылдайды → Interrupt**.
2. Қай бетте тоқтағаныңызды есте сақтайсыз → процессор **ағымдағы күйді сақтайды**.
3. Телефонға жауап бересіз → **ISR орындалады**.
4. Сөйлесуді аяқтайсыз → **Interrupt аяқталады**.
5. Кітапты тоқтаған жеріңізден жалғастырасыз → процессор **негізгі бағдарламаға қайтады**.

MCU және MPU айырмашылығы

MCU (Микроконтроллер) мен **MPU (Микропроцессор)** арасындағы басты айырмашылық – олардың ішкі құрылымындағы жад пен перифериялық құрылғылардың бір чипте бірігу дәрежесі және атқаратын міндеті. MCU – бұл дербес шағын компьютер, ал MPU – тек есептеуіш орталық (процессор)

Негізгі айырмашылықтар

Құрылымы (Архитектура):

- **MCU:** Бір ғана кристалл ішінде CPU (орталық процессор), жад (Flash, SRAM) және енгізу-шығару порттарын біріктіреді. Сыртқы қосымша компоненттерді қажет етпейді.
- **MPU:** Тек процессор ядросын қамтиды. Жұмыс істеуі үшін сыртқы жедел жад (RAM), тұрақты жад (Flash/ROM) және басқа да микросхемаларды қажет етеді.
-

Бағдарламалық қамтамасыз ету (ОС):

- **MCU:** Күрделі операциялық жүйелерді қажет етпейді. Тікелей «железо» деңгейінде (bare-metal) немесе жеңіл нақты уақыт режиміндегі жүйелермен (RTOS) жұмыс істейді.
- **MPU:** Linux немесе Android сияқты толыққанды операциялық жүйелерді іске қоса алады.

Өнімділік пен жиілік:

- **MCU:** Жиілігі әдетте МГц (мегагерц) диапазонында болады, қуатты аз тұтынады.
- **MPU:** Жиілігі ГГц (гигагерц) деңгейіне жетеді, көп ядролы болып келеді және өнімділігі әлдеқайда жоғары.

Қолданылу саласы:

- **MCU:** Тұрмыстық техника, сенсорлар, IoT (заттар интернеті) құрылғылары, моторды басқару жүйелері.
- **MPU:** Смартфондар, компьютерлер, өнеркәсіптік терминалдар, күрделі графикалық интерфейстері (HMI) бар жүйелер мен жасанды интеллект қондырғылары.

Real-Time

Real-Time (нақты уақыт режимі) — бұл ақпараттық технологияларда деректердің кешігусіз, болған сәтінде бірден өңделуін білдіретін термин. Басқаша айтқанда, жүйенің сыртқы оқиғаларға немесе пайдаланушының әрекетіне секундтар немесе миллисекундтар ішінде жауап беру қабілеті.

Автомобильдік жүйеде нәтиженің дұрыс болуы ғана жеткіліксіз. Ол **қажетті уақытта** орындалуы керек. Мысалы: Sensor Event – Processing – Response

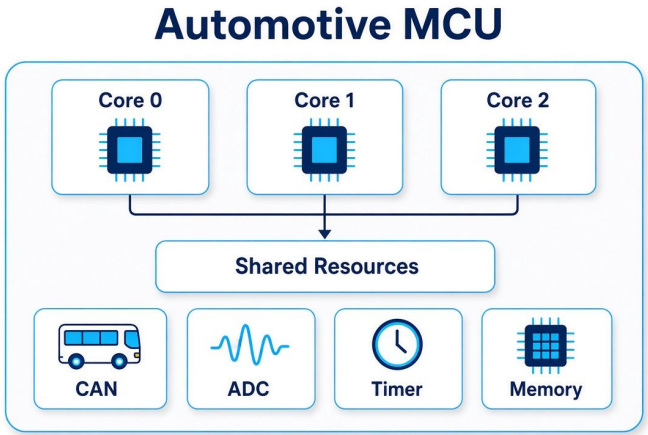
Егер жүйе реакцияны тым кеш берсе, математикалық нәтиже дұрыс болғанымен, басқару жүйесі үшін ол жарамсыз болуы мүмкін.

Сондықтан: **Real-Time System — нәтиженің дұрыстығымен қатар оның уақытында алынуы маңызды болатын жүйе.**

Multicore MCU

Күрделі автомобильдік микроконтроллерлерде бірнеше CPU ядросы болуы мүмкін.

Мұндай архитектура әртүрлі бағдарламалық тапсырмаларды бірнеше ядро арасында бөлуге мүмкіндік береді.



Мысалы, концептуалды түрде:

Core 0 → Control Tasks

Core 1 → Communication

Core 2 → Monitoring / Safety

Нақты бөлу нақты ECU және software architecture-ға байланысты.

Неліктен кәдімгі MCU мен Automotive MCU бірдей емес?

Кәдімгі MCU (микроконтроллерлер) мен автомобильдік (Automotive) MCU-лар бірдей емес,

өйткені олар мүлдем басқа өндірістік стандарттарға, қауіпсіздік талаптарына және жұмыс істеу ортасына негізделіп жасалады. Автомобильдік микроконтроллерлердің басты мақсаты — адам өміріне қауіп төндірмей, аса ауыр жағдайларда мінсіз жұмыс істеу.

Төмендегі кестеде олардың негізгі айырмашылықтары көрсетілген:

Көрсеткіш	Кәдімгі MCU (Коммерциялық/ Өндірістік)	Автомобильдік (Automotive) MCU
Жұмыс температурасы	Салыстырмалы түрде жұмсақ (0°C бастап +70°C немесе +85°C дейін).	Аса жоғары төзімділік (**-40°C бастап +125°C немесе +150°C** дейін).
Сапа стандарты	Стандартты өндірістік бақылау.	**AEC-Q100** стандартына сай қатаң стресс-тесттерден өтеді.
Қауіпсіздік деңгейі	Сәтсіздікке ұшыраса, құрылғы өшеді немесе қайта жүктеледі.	**ISO 26262 (ASIL)** функционалдық қауіпсіздік стандартын қолдайды.
Төзімділік	Соққыға, дірілге және электромагниттік кедергілерге төзімділігі орташа.	Дірілге, жоғары ылғалдылыққа және **электромагниттік кедергілерге (EMI)** барынша қорғалған.
Қызмет ету мерзімі	Әдетте 2–5 жыл (жаңа модельдер шыққанда өндірістен тез шығарылады).	**15 жылдан астам** уақыт бойы тұрақты жұмыс істеу және зауыттан қолдау көру кепілдігі бар.
Ақаулық деңгейі	Миллион данаға шаққанда бірнеше ақау қалыпты деп саналады (PPM).	Нөлдік ақаулыққа жақын деңгей талап етіледі (**Zero Defect** принципі).

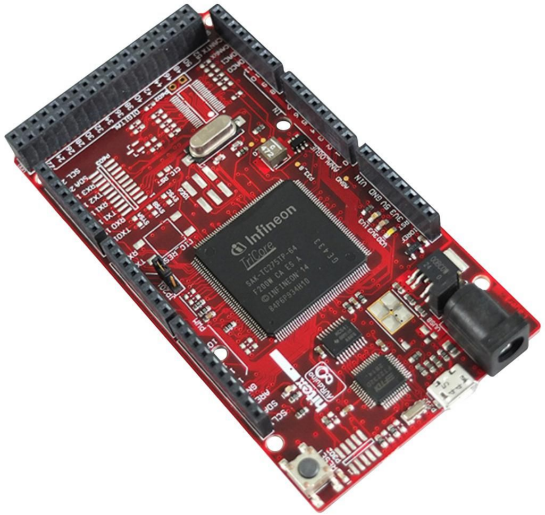
nxp s32k



Renesas RH850



Infineon AURIX



Қорытынды

Микроконтроллер — CPU, жад, перифериялық модульдер және жүйелік басқару блоктарын бір микросхемада біріктіретін есептеу жүйесі.

Автомобильдік MCU қарапайым есептеулерді ғана орындамайды. Ол датчиктерді оқиды, real-time басқару алгоритмдерін орындайды, басқа ECU-лармен байланысады және қауіпсіздікке байланысты функцияларды бақылайды.

Бақылау сұрақтары

1. MCU дегеніміз не?
2. MCU қандай негізгі блоктардан тұрады?
3. CPU қандай қызмет атқарады?
4. Flash жады не үшін қолданылады?
5. RAM жады не үшін қажет?
6. Flash пен RAM арасындағы айырмашылық қандай?
7. Регистр дегеніміз не?
8. System Bus қандай қызмет атқарады?
9. Peripheral дегеніміз не?
10. GPIO қандай міндет атқарады?
11. ADC не үшін қажет?
12. Timer қандай қызмет атқарады?
13. CAN контроллері не үшін қолданылады?
14. Clock System дегеніміз не?

15. Reset қандай қызмет атқарады?
16. Watchdog Timer не үшін қажет?
17. Interrupt дегеніміз не?
18. DMA қолданудың артықшылығы қандай?
19. MCU және MPU арасындағы негізгі айырмашылық қандай?
20. Real-Time System дегеніміз не?
21. Multicore MCU дегеніміз не?
22. Automotive MCU-ға қандай арнайы талаптар қойылады?
23. NXP S32K дегеніміз не?
24. Infineon AURIX дегеніміз не?
25. Renesas RH850 дегеніміз не?
26. Оқу кезінде STM32 қолдану automotive MCU-ларды үйренуге қалай көмектеседі?