Электромобильдерге арналған микроконтроллер жүйелерін бағдарламалау

Дәуренов ӘД

Дәріс 10. Real-Time Embedded Software

маңызды сұрақ:

Егер бір микроконтроллер бір уақытта көптеген тапсырманы орындауы керек болса, олардың орындалу уақытын қалай ұйымдастырамыз?

Дәрістің мақсаты

Бұл дәрістің мақсаты:

- Real-Time ұғымын түсіндіру;
- Super-loop архитектурасын қарастыру;
- периодтық тапсырмалар ұғымын түсіндіру;
- Scheduler қызметін түсіндіру;

Real-Time дегеніміз не?

Real-Time System — нақты уақыт жүйесі — сыртқы оқиғаға белгіленген уақыт ішінде жауап беруі тиіс жүйе.

Маңызды:

Real-Time дегеніміз міндетті түрде «өте жылдам» деген сөз емес.

Негізгі талап:

Жауап қажетті уақыт шегінде берілуі керек.

Мысалы:

```
Оқиға пайда болды
        ↓
Жүйе оны анықтады
        ↓
Есептеу жүргізді
        ↓
Жауап берді
        ↓
Deadline-нан кешікпеді
```

Басқару жүйесіндегі өңдеу уақыты (Response Time / Deadline)

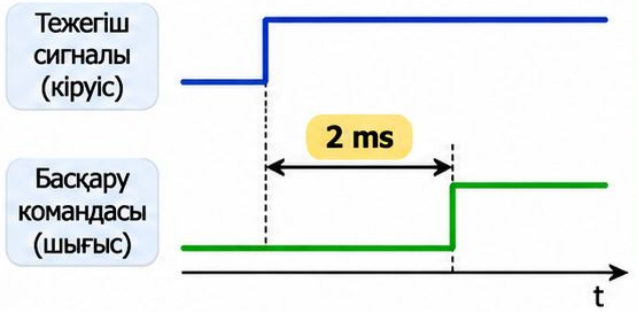
Жалпы ақпарат

Егер автомобиль контроллері тежегіш сигналының өзгеруін **2 ms** ішінде өңдесе, бұл оның басқару жүйесіне жауап беру уақыты (Response Time) болып табылады.

Өңдеу уақытына қойылатын талап

Өңдеу уақыты	Бағалау
1 ms	✓ Жақсы
1.5 ms	✓ Жақсы
2 ms	! Рұқсат етілген
5 ms	⊗ Кеш (қабылданбайды)

Уақыттық диаграмма (мысал)



Маңызы

- Басқару жүйесіндегі қауіпсіздікті қамтамасыз етеді.
- Жетекші құрылғылардың (тежеу, тарту) уақтында және сенімді жұмыс істеуін қамтамасыз етеді.
- Көлік динамикасы мен тұрақтылығына тікелей әсер етеді.

Негізгі ұғымдар

Response Time (жауап беру уақыты) — Кіруіс сигналы өзгерген сәттен бастап, контроллердің басқару командасын шығарғанға дейінгі уақыт.

Deadline (мерзім) — Басқару командасы орындалуы тиіс ең ұзақ рұқсат етілген уақыт.

Қорытынды

✓ Өңдеу уақыты неғұрлым аз болса, басқару жүйесінің жауап беруі соғұрлым жылдам, ал көліктің жұмысы қауіпсіз және тұрақты болады.

Deadline, Hard Real-Time және Soft Real-Time дегеніміз не?

Deadline — тапсырма орындалып бітуі тиіс ең соңғы уақыт шегі.
Real-Time (нақты уақыт режимі) — микроконтроллердің белгілі бір тапсырманы белгіленген уақыт ішінде орындауы.

Мұнда ең маңыздысы — есептеуді дұрыс орындау ғана емес, оны уақытында орындау.
Мысалы, электромобильдің микроконтроллері тежегіш педалінің басылғанын анықтады. Ол бұл ақпаратты уақытында өңдеп, қажетті басқару әрекетін орындауы керек.
Real-Time жүйелерінің екі негізгі түрін қарастырайық.

Hard Real-Time және Soft Real-Time режимдері

1 Hard Real-Time — қатаң нақты уақыт режимі

Жалпы ақпарат

Hard Real-Time — тапсырма белгіленген уақыттан кеш орындалмауы тиіс жүйе.

Ерекшелігі

Егер тапсырма уақытында орындалмаса, бұл жүйенің дұрыс жұмыс істемеуіне немесе кауіпті жағдайға әкелуі мүмкін.

Мысал

Электромобильдің инверторын басқару.

Орындау тәртібі

Инвертордың басқару алгоритмі әрбір 1 мс сайын орындалуы керек деп алайық (шартты оқу мысалы).

Тапсырманың орындау мерзімі — 1 мс



Қорытынды (негізгі тұжырымдар)

- Егер алгоритм 0,8 мс ішінде орындалса — талап сақталады.
- Егер 1,2 мс ішінде орындалса — белгіленген мерзім бұзылады.
- Қатаң нақты уақыт жүйесінде мұндай кешігуге жол берілмейді.

2 Soft Real-Time — икемді нақты уақыт режимі

Жалпы ақпарат

Soft Real-Time — тапсырманы уақытында орындау маңызды, бірақ аздаған кешігу жүйенің толық істен шығуына міндетті түрде әкелмейді.

Ерекшелігі

- Кейбір тапсырмаларда аздаған кешігуге жол беріледі.
- Жүйе жұмысы жалғаса береді, бірақ сапасы төмендеуі мүмкін.

Мысал

Электромобиль дисплейндегі жылдамдықты жаңарт

Орындау тәртібі

Дисплей жылдамдықты әрбір 100 мс сайын жаңартуы керек делік.

Дисплейді жаңарту мерзімі — 100 мс



Қорытынды (негізгі тұжырымдар)

- Егер дисплей 100 мс орнына 120 мс кейін жаңартылса, жүргізуші бұл айырмашылықты байқамауы да мүмк
- Яғни, кешігу жағымсыз болғанымен, жүйе жұмысын жалғастыра алады.

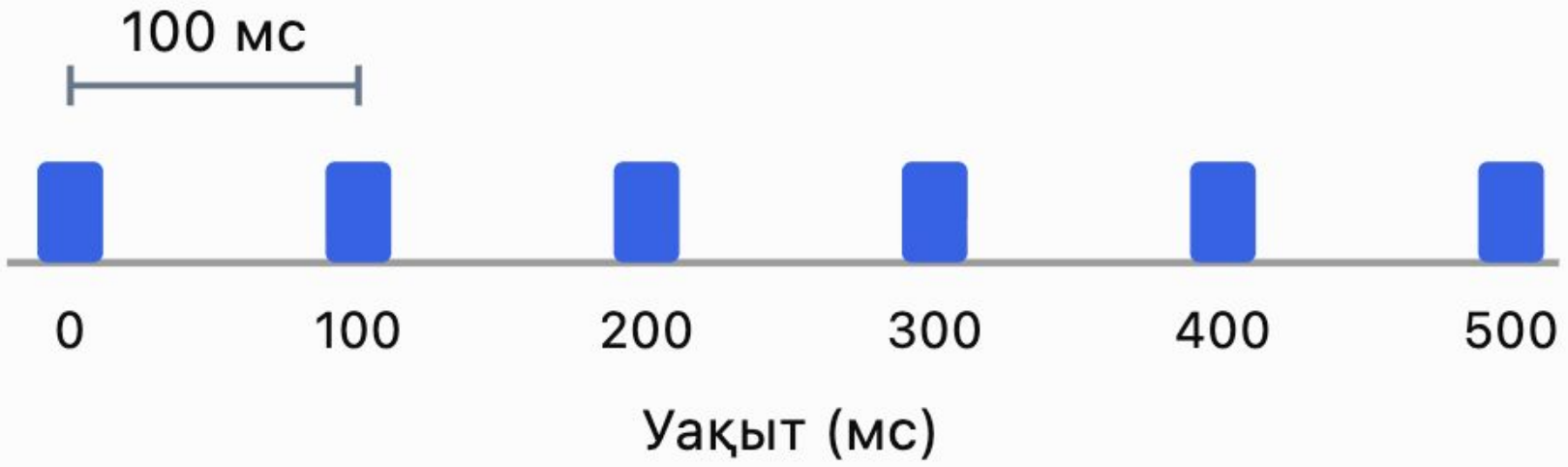
Periodic Task — периодтық тапсырма

Акселератор датчигін тек бір рет тексеру жеткіліксіз. Жүргізуші педальды кез келген уақытта басуы мүмкін.

Сондықтан микроконтроллер датчиктерді қайта-қайта тексеріп отыруы керек.
Periodic Task (периодтық тапсырма) — белгілі бір уақыт аралығында тұрақты түрде қайталанып орындалуға жоспарланған тапсырма.

Мысалы, датчиктерді әрбір 100 мс сайын тексеру.

Бір периодтық тапсырманың уақыт шкаласы



Көк белгілер тапсырманың жоспарланған іске қосылу сәттерін көрсетеді.

Мұнда:

- Период (T) — 100 мс.
- Орындау уақыты (C) — тапсырманың аяқталуына кететін уақыт, мысалы 5 мс.

Бұл екеуі бірдей емес.

Мысалы, тапсырма әрбір 100 мс сайын іске қосылуға жоспарланса да, оны орындауға тек 5 мс кетуі мүмкін.

Демек, Periodic Task тапсырманың қаншалықты жиі қайталанатынын анықтайды.

Multiple Periodic Tasks — бірнеше периодтық тапсырма

Нақты электромобильде микроконтроллер тек акселератор датчиктерін ғана бақыламайды.

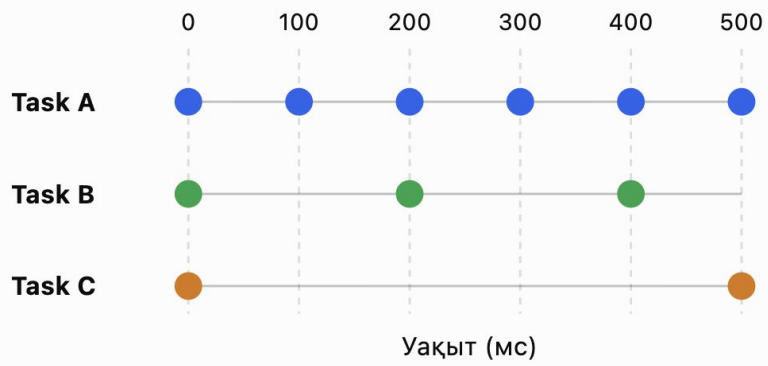
Ол батарея температурасын өлшейді, басқа басқару блоктарымен ақпарат алмасады және диагностикалық мәліметтерді шығарады.

Бұл тапсырмалардың барлығын бірдей жиілікпен орындау міндетті емес.

Мысалы:

Тапсырма	Период
Task A — акселератор датчиктерін тексеру	100 мс
Task B — батарея температурасын өлшеу	200 мс
Task C — диагностикалық ақпаратты шығару	500 мс

Бірнеше периодтық тапсырманың жоспарланған іске қосылуы



Әр тапсырманың өз периоды бар. График олардың орындалу ұзақтығын немесе орындалу ретін көрсетпейді.

Multiple Periodic Tasks — әртүрлі периодпен қайталанатын бірнеше тапсырманың жиынтығы.

Бірақ мұнда жаңа мәселе пайда болады.
Мысалы, 1000 мс уақытында үш тапсырманың да орындалу уақыты келді делік.

Микроконтроллер қай тапсырманы бірінші орындауы керек?
Осы сұраққа келесі ұғым жауап береді.

Scheduler — тапсырмаларды жоспарлаушы

Scheduler — тапсырмалардың қашан және қандай ретпен орындалатынын анықтайтын бағдарламалық механизм.
Оны қарапайым тілмен диспетчер деп түсінуге болады.

Мысалы, университетте бірнеше пән бар. Әр пәннің өз сабақ уақыты болады. Бірақ сабақтарды бір-бірімен сәйкестендіріп, ортақ кесте құру қажет.

Scheduler де микроконтроллердегі тапсырмаларды осылай ұйымдастырады.

Multiple Periodic Tasks пен Scheduler арасындағы айырмашылық:

Multiple Periodic Tasks — қандай тапсырмалар бар және олардың периодтары қандай екенін сипаттайды.

Scheduler — сол тапсырмалардың орындалуын ұйымдастырады.

Scheduler-дің бірнеше түрі бар. Біз ең қарапайым түрін қарастырамыз.

Циклдік жоспарлаушы (Cyclic Scheduler)

Жалпы ақпарат

Cyclic Scheduler — тапсырмаларды алдын ала белгіленген, қайталанатын уақыт кестесі бойынша орындайтын жоспарлаушы.

Мысал: тапсырмалар мен периодтар

Мысалы, Task A, B және C тапсырмаларының периодтары сәйкесінше 100, 200 және 500 мс.

Олардың ортақ қайталану кестесі 1000 мс-тан кейін қайта басталады.

Бұл уақыт аралығы **Major Cycle (үлкен цикл)** деп аталады.

Тапсырмалардың орындау уақыты

Тапсырмалардың орындалу ұзақтығы мынадай болсын:

- Task A — 5 мс.
- Task B — 10 мс.
- Task C — 15 мс.

Барлық тапсырмаларды орындау

$$T = 5 + 10 + 15 = 30 \text{ мс}$$

уақыт кетеді.

Cyclic Scheduler — бір толық жоспарлау циклі

Уақыт	Орындалатын тапсырмалар
100 мс	A
200 мс	A → B
300 мс	A
400 мс	A → B
500 мс	A → C
600 мс	A → B
700 мс	A
800 мс	A → B
900 мс	A
1000 мс	A → B → C

1000 мс өткен соң, дәл осындай кесте келесі 1000 мс аралығында қайталанады.

Тапсырмалар бір уақытта дайын болса не болады?

Мысалы, 1000 мс кезінде үш тапсырманың да кезегі келді.

Бір ядролы микроконтроллердегі қарапайым циклдік жоспарлаушы оларды бірінен кейін бірін орындайды.

Scheduler — тапсырмалардың нақты орындау реті



Қорытынды

- Демек, тапсырмалар 1030 мс сәтінде аяқталады, ал келесі 100 мс белгісі 1100 мс кезінде келеді.
- Осы мысалда тапсырмалар келесі уақыт белгісіне дейін аяқталып үлгереді.

Негізгі ерекшелік

Cyclic Scheduler-дің негізгі ерекшелігі — тапсырмалардың орындалу кестесі алдын ала белгілі және белгілі бір уақытта қайталанып отыратын (периодтық) бір үлкен циклмен ұйымдастырылуы.

Super-Loop — шексіз цикл архитектурасы

Енді тағы бір сұрақ туындайды:

Scheduler жоспарлаған тапсырмаларды микроконтроллер бағдарламасының қай бөлігінде орындаймыз?

Мұның қарапайым тәсілі — Super-Loop архитектурасын пайдалану.

Super-Loop — микроконтроллер бағдарламасының тапсырмаларды шексіз цикл ішінде орындайтын құрылымы.

C++ ортасында бұл loop() функциясы арқылы жүзеге асырылады.

```
void loop()
{
  readSensors();

  checkSignals();

  updateControl();
}
```

Бұл қарапайым Super-Loop. Тапсырмалар аяқталғаннан кейін цикл бірден қайта басталады.
Ал егер біз тапсырмаларды әртүрлі периодпен орындағымыз келсе, Super-Loop ішіне Scheduler логикасын енгізе аламыз.
Сонда Timer-based Super-Loop архитектурасы пайда болады.

Timer-based Super-Loop — таймерге негізделген шексіз цикл

Бұл тәсілде TIM2 таймері белгілі бір уақыт аралығында белгі береді.

Scheduler осы белгілер бойынша қай тапсырманың орындалуы керектігін анықтайды. Ал Super-Loop тапсырмаларды орындайды.

Timer, Super-Loop, Scheduler және Periodic Tasks — бір кодта

1. Негізгі ұғымдар

Timer (TIM2) — Әр 100 мс сайын үзіліс тудырып, белгі береді (**task100ms** = true).

Super-Loop — Бағдарламаның негізгі шексіз циклі (`loop()`).

Scheduler — Қай тапсырманың орындалу уақыты келгенін анықтайды (`tickCount` және `if` шарттары).

Periodic Tasks — Белгілі периодпен қайталанатын нақты тапсырмалар:

- Task A — 100 мс
- Task B — 200 мс
- Task C — 500 мс

2. Код (түстермен белгіленген)

```
1  volatile bool task100ms = false;
2
3  // Timer (TIM2) үзіліс қызметі
4  void timerISR() {
5      task100ms = true;
6  }
7
8  int tickCount = 0;
9
10 // Super-Loop
11 void loop() {
12   if (task100ms) {
13     task100ms = false;
14     tickCount++;
15
16     // Task A — 100 мс
17     processSensors();
18     updateControl();
19
24     // Task B — 200 мс
21     if (tickCount % 2 == 0) {
22       readBatteryTemperature();
24     }
26     // Task C — 500 мс
28     if (tickCount % 5 == 0) {
27       printStatus();
28     }
29   }
20 }
```

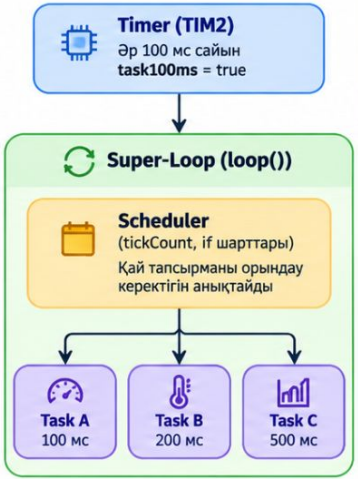
Timer
100 мс сайын белгі береді

Super-Loop
Негізгі шексіз цикл

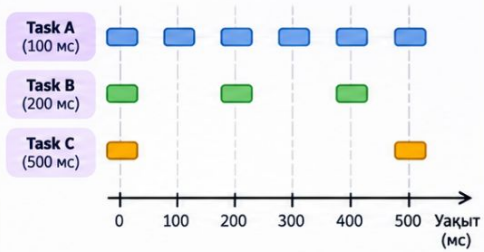
Scheduler
Қай тапсырманы орындау керектігін анықтайды (`tickCount` және шарттар)

Periodic Tasks
- **Task A** — 100 мс
- **Task B** — 200 мс
- **Task C** — 500 мс

3. Жұмыс принципі



4. Уақыт шкаласы (орындалу реті)



Нәтиже

- **Timer:** әр 100 мс белгі береді.
- **Super-Loop:** осы белгілерді өңдеп, циклді қайталайды.
- **Scheduler: tickCount** арқылы қай тапсырма орындалатынын анықтайды.
- **Periodic Tasks: A (100 мс)**, B (200 мс), C (500 мс) жоспарланған уақытта орындалады.

Барлық ұғымдардың өзара байланысы

Ұғым	Негізгі сұрақ	Мағынасы
Real-Time	Тапсырма уақытында орындала ма?	Уақыт талаптары
Hard Real-Time	Кешігуге бола ма?	Белгіленген мерзімді бұзуға болмайды
Soft Real-Time	Аздаған кешігуге бола ма?	Кейбір кешігулерге төзуге болады
Periodic Task	Бір тапсырма қаншалықты жиі қайталанады?	Бір периодтық тапсырма
Multiple Periodic Tasks	Қандай қайталанатын тапсырмалар бар?	Бірнеше периодтық тапсырма
Scheduler	Қай тапсырманы қашан және қандай ретпен орындаймыз?	Жалпы жоспарлау механизмі
Cyclic Scheduler	Тапсырмаларды қайталанатын кестемен қалай жоспарлаймыз?	Scheduler түрі
Super-Loop	Тапсырмалар бағдарламада қайда орындалады?	Шексіз цикл архитектурасы
Timer-based Super-Loop	Циклдегі тапсырмаларды уақытпен қалай байланыстырамыз?	Таймер мен Super-Loop біріктірілген тәсіл

Жалпы мысал

Елестетейік, электромобильде STM32 микроконтроллері жұмыс істеп тұр.
Бірінші: микроконтроллер акселератор датчиктерін уақытында тексеруі қажет. Бұл — Real-Time талабы.
Екінші: датчиктерді әрбір 100 мс сайын тексеру жоспарланған. Бұл — Periodic Task.
Үшінші: батарея температурасын әрбір 200 мс сайын өлшеу және диагностиканы әрбір 500 мс сайын шығару да қажет. Бұл — Multiple Periodic Tasks.
Төртінші: осы тапсырмалардың қайсысын қашан орындау керектігін Scheduler анықтайды.
Бесінші: егер тапсырмалар алдын ала белгіленген, қайталанатын кестемен жоспарланса, бұл — Cyclic Scheduler.
Алтыншы: тапсырмалар loop() шексіз циклінің ішінде орындалса, бұл — Super-Loop архитектурасы.
Жетінші: TIM2 таймері әрбір 100 мс сайын уақыт белгісін берсе, біз Timer-based Super-Loop тәсілін қолданамыз.