

ROB3111 БИОТЕХНИКАЛЫҚ ЖҮЙЕЛЕРДЕГІ **БАСҚАРУ**

Сабақтың тақырыбы:

Arduino-да таймерлерді пайдалану

Оқытушы: Исабеков Жанібек Назарбекұлы

Сабақтың мақсаты: Arduino тақталарының таймерлерімен жұмыс.

Сабақ жоспары:

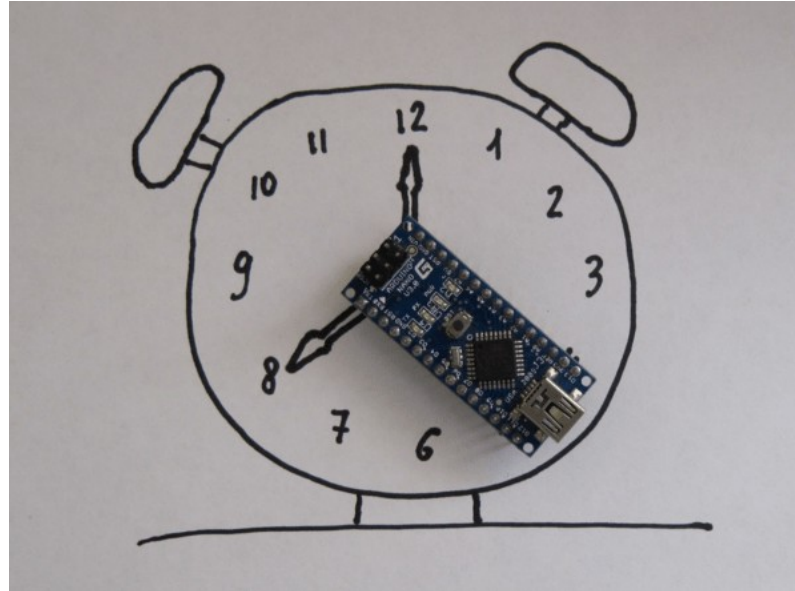
- Таймерлер
- TimerOne кітапханасы

Таймерлер

Микроконтроллерлердегі күнделікті өмірдегі сияқты, таймер - бұл сіз орнатқан сәтте болашақта сигнал бере алатын нәрсе. Бұл нүкте келгенде, микроконтроллердің үзілуі шақырылады, оған кодтың белгілі бір бөлігін орындау сияқты бірдеңе істеу керектігін ескертеді.

Таймерлер, сыртқы үзілістер сияқты, негізгі бағдарламадан тәуелсіз жұмыс істейді. Циклдарды орындаудың немесе `millis()` кідірісін қайталап шақырудың орнына, кодыңыз басқа әрекеттерді орындаған кезде таймерге өз жұмысын орындауға тағайындай аласыз.

Таймерлер



Сонымен, жарық диодты әр 5 секунд сайын жыпылықтау сияқты бірдеңе істеу керек құрылғы бар делік. Егер сіз таймерлерді пайдаланбасаңыз, бірақ кәдімгі кодты жазсаңыз, онда айнымалы мәнді жарық диоды жанған кезде орнатып, оны ауыстыру сәтінің келгенін үнемі тексеріп отыруыңыз керек. Таймердің үзілісімен үзілісті орнату, содан кейін таймерді іске қосу жеткілікті. Негізгі бағдарламаның әрекеттеріне қарамастан, жарық диоды дәл уақытында жыпылықтайды.

Таймердің жұмыс істеу принципі

Ол санау регистрі деп аталатын айнымалыны көбейту арқылы әрекет етеді. Санақ регистрі оның мөлшеріне байланысты белгілі бір мөлшерге дейін санай алады. Таймер есептегішті максималды мәнге жеткенше қайта-қайта арттырады, осы кезде есептегіш толып, нөлге қайтарылады. Таймер әдетте толып кетудің орын алғанын хабарлау үшін жалаушаның битін орнатады.

Сіз бұл жалаушаны қолмен тексере аласыз немесе таймер қосқышын жасай аласыз — жалауша орнатылған кезде үзілісті автоматты түрде шақыра аласыз. Кез келген басқа үзілістер сияқты, таймер толып кеткенде алдын ала орнатылған кодты орындау үшін қызметтік үзіліс тәртібін (Interrupt Service Routine немесе ISR) тағайындауға болады. ISR толып кету жалаушасын өзі қалпына келтіреді, сондықтан үзілістерді пайдалану әдетте қарапайымдылық пен жылдамдыққа байланысты ең жақсы таңдау болып табылады.

Таймердің жұмыс істеу принципі

Есептегіш мәндерін нақты уақыт аралықтарында арттыру үшін таймерді сағат көзіне қосу керек. Сағат көзі үнемі қайталанатын сигнал шығарады. Таймер бұл сигналды анықтаған сайын есептегіш мәнін бір-бірден арттырады. Таймер сағат көзінен жұмыс істейтіндіктен, өлшенетін ең аз уақыт бірлігі сағат циклінің периоды болып табылады. Егер сіз 1 мГц жиіліктегі сағат сигналын қоссаңыз, онда таймердің ажыратымдылығы (немесе таймердің периоды) болады:

$$T = 1 / f \text{ (} f \text{ - сағат жиілігі)}$$

$$T = 1 / 1 \text{ мГц} = 1 / 10^6 \text{ Гц}$$

$$T = (1 * 10^{-6}) \text{ бірге}$$

Ардуинодағы таймерлер

Стандартты Arduino тақталарында 8 биттік AVR чипінде бірден бірнеше таймер бар. Atmega168 және Atmega328 чиптерінде Timer0, Timer1 және Timer2 үш таймері бар. Сондай-ақ оларда ақаулардан қорғау үшін немесе бағдарламалық құралды қалпына келтіру механизмі ретінде пайдалануға болатын қарауыл таймері бар. Мұнда әрбір таймердің кейбір мүмкіндіктері берілген.

Timer0:

Timer0 - 8 биттік таймер, яғни оның санау регистрі 255-ке дейінгі сандарды (яғни қол қойылмаған байттарды) сақтай алатынын білдіреді. Timer0 стандартты Arduino уақытша функцияларында қолданылады, мысалы, `delay()` және `millis()`, сондықтан егер сіз оның салдары туралы ойласаңыз, оны шатастырмағаныңыз жөн.

Timer1:

Timer1 - 16 биттік таймер, максималды санау мәні 65535 (қол қойылмаған бүтін сан). Бұл таймерді Arduino Servo кітапханасы қолданады, егер сіз оны өз жобаларыңызда қолдансаңыз, ескеріңіз.

Ардуинодағы таймерлер

Timer2:

Timer2 — 8 биттік және Timer0-ге өте ұқсас. Ол tone() функциясының Arduino-да қолданылады.

Timer3, Timer4, Timer5:

ATmega1280 және ATmega2560 чиптерінде (Arduino Mega нұсқаларында орнатылған) үш қосымша таймер бар. Олардың барлығы 16 биттік және Timer1-ге ұқсас жұмыс істейді.

TimerOne кітапханасы

`Timer1.initialize(period)`

Басқа әдістерді қолдану үшін алдымен осы әдісті шақыру керек. Мұнда таймердің кезеңін қосымша көрсетуге болады (микросекундтармен), әдепкі бойынша ол 1 секундқа тең. Бұл Arduino UNO жүйесіндегі 9 және 10 цифрлық түйреуіштер үшін `analogWrite()` функциясын бұзатынын ескеріңіз.

Timer1 таймерін 10 мс периодпен инициализациялауға мысал келтірейік.

`Timer1.initialize(10000);` // Үзіліс әрбір 10 мс

TimerOne кітапханасы

`Timer1.attachInterrupt(function, period)`

Функцияны микросекундтарда көрсетілген уақыт аралығы арқылы шақырады. Егер сіз осы функцияны соңғы параметр ретінде шақырған кезде микросекундтардағы мәнді көрсетсеңіз, периодты ерікті түрде орнатуға болатындығын ескеріңіз.

Үзілісті Timer1 мерзімді оқиғасына 1 мс тұрақты жиіліктегі Timer1_ISR өңдеуші функциясын орындайтын етіп бекітуге мысал келтірейік.

`Timer1.attachInterrupt(Timer1_ISR, 1000);`

TimerOne кітапханасы

```
#include <TimerOne.h>
```

```
void Timer1_ISR(void)
{
    digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
}
```

```
void setup(void)
{
    pinMode(LED_BUILTIN, OUTPUT);
    Timer1.initialize(10000); // Fire An Interrupt Every 10ms
    Timer1.attachInterrupt(Timer1_ISR);
}
```

```
void loop(void)
{
    // Nothing To Do
}
```