

Вбудовані системи

Вбудована система (ВС) – це спеціалізована комп'ютерна система (або обчислювальний пристрій), призначена для виконання обмеженої кількості функцій, часто, з обмеженнями реального часу. Комбінація апаратного та програмного забезпечення, можливо, з механічними або іншими частинами, призначена для виконання окремої функції.

У зв'язку з тим, що ВС розміщується всередині більш складного пристрою, при її розробці ключову роль відіграють такі чинники:

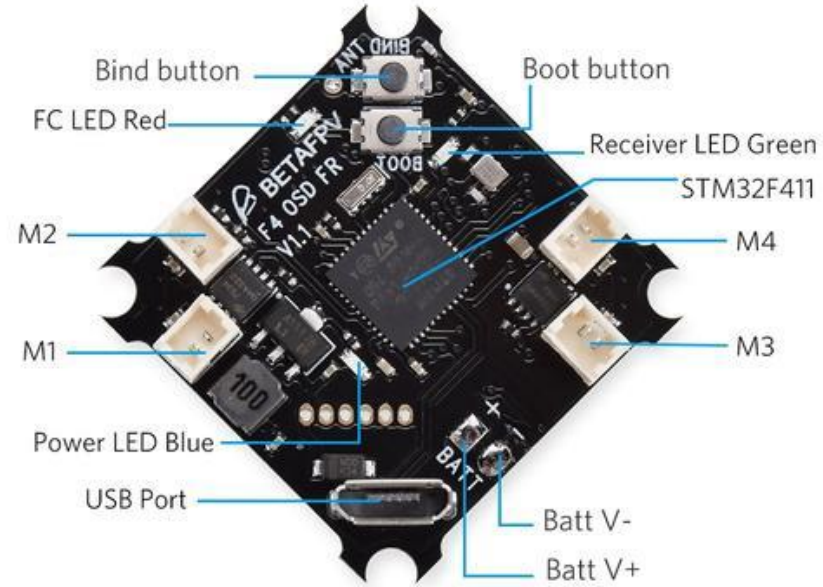
- мінімальне власне енергоспоживання (можливо, автономне живлення);
- процесор і системна логіка, а також ключові вузли по можливості поєднані на одному кристалі;
- мінімальні власні габарити і вага;
- спеціальні військово-космічні вимоги з радіаційної та електромагнітної стійкості, працездатність в вакуумі, гарантований час напрацювання, термін доступності рішення на ринку і т. д.
- функції відведення тепла, (охолодження) повинні забезпечувати мінімум вимог, до теплових режимів. (Якщо щільність теплового потоку не перевищує $0,5 \text{ мВт} / \text{см}^2$, перегрів поверхні пристрою щодо навколишнього середовища не перевищить $0,5 \text{ }^{\circ} \text{C}$, така апаратура не вимагає спеціальних схем охолодження.)

Сфери застосування вбудованих систем



Приклади вбудованих систем

BOTTOM FRONT ↑



Boot button

Receiver LED Green

STM32F411

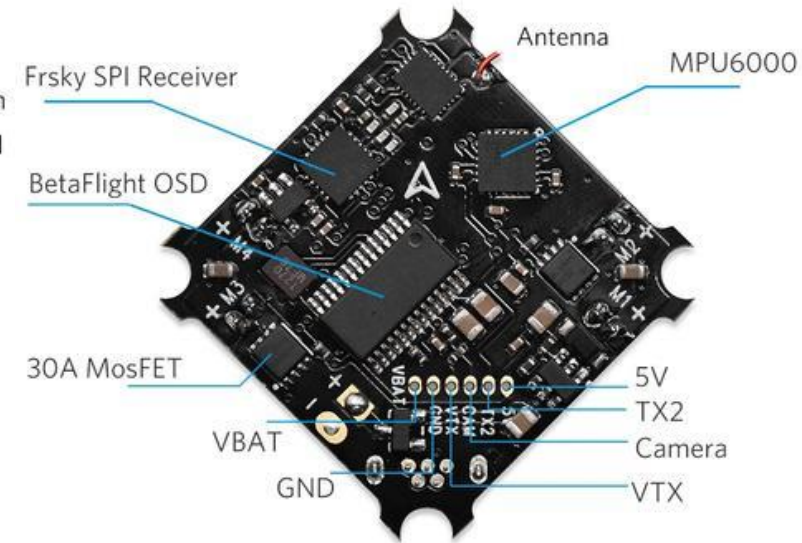
M4

M3

Batt V-

Batt V+

TOP FRONT ↑

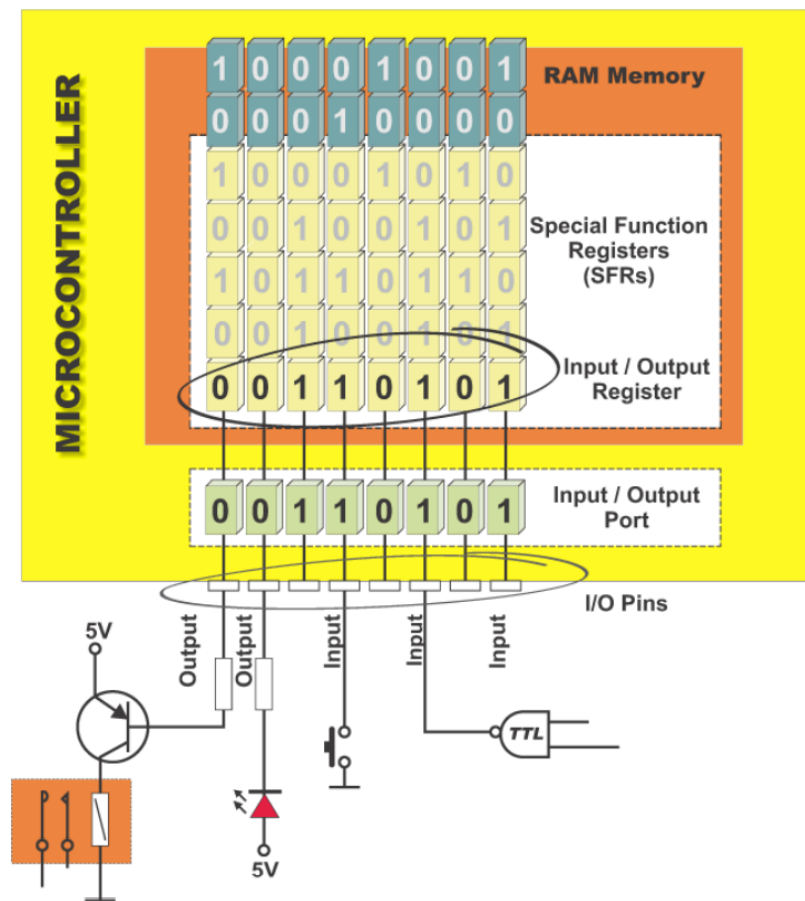
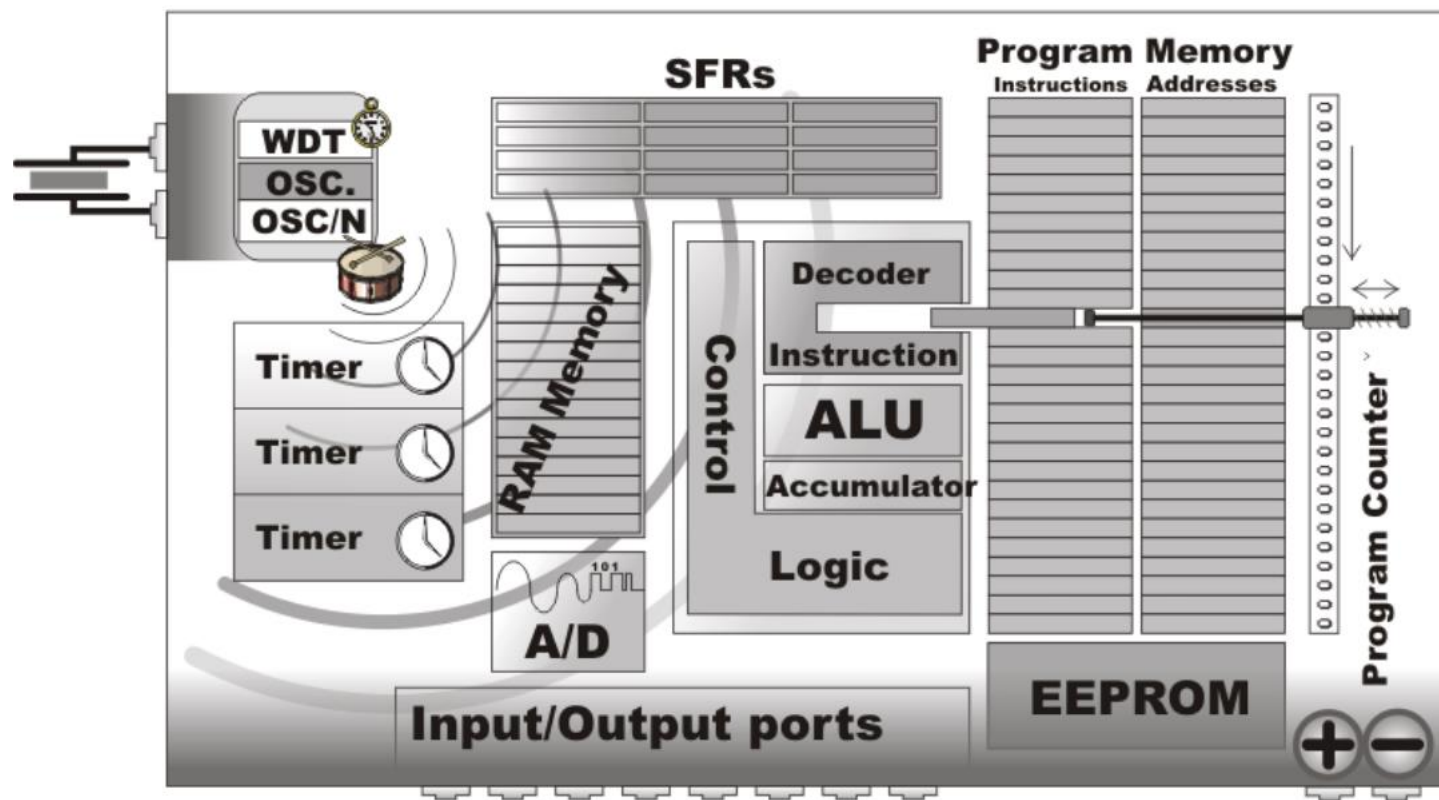


Коротка історія розвитку мікроконтролерів

Термін «мікроконтролер» (МК) витіснив раніше використовуваний термін «однокристальна мікро-ЕОМ». Перший патент на однокристальну мікро-ЕОМ був виданий в 1971 році інженерам М. Кочрену і Г. Буну, співробітникам компанії Texas Instruments. Саме вони запропонували на одному кристалі розмістити не тільки мікропроцесор, але й пам'ять, пристрої вводу-виводу. З появою однокристальних мікро-ЕОМ пов'язують початок ери комп'ютерної автоматизації в галузі керування. Мабуть, ця обставина і визначило термін «мікроконтролер» (Control - керування).

У 1980 році фірма Intel випустила мікроконтролер *i8048*. Трохи пізніше у цьому ж році Intel випускає наступний мікроконтролер: *i8051*. Вдалий набір периферійних пристроїв, можливість гнучкого вибору зовнішньої або внутрішньої програмної пам'яті і прийнятна ціна забезпечили цьому мікроконтролеру успіх на ринку. З точки зору технології, мікроконтролер *i8051* був для свого часу дуже складним виробом - в кристалі було використано 128 тис. транзисторів, що в 4 рази перевищувало кількість транзисторів в 16-розрядному мікропроцесорі *i8086*.

Демонстрація роботи мікроконтролера та портів вводу/виводу



Сімейства мікроконтролерів

Всі мікроконтролери об'єднуються в сімейства. До одного сімейства відносять вироби, які мають однакове процесорне ядро - сукупність таких понять, як система команд, циклограма роботи ЦП, організація пам'яті програм і пам'яті даних, система переривань і базовий набір периферійних пристроїв. Відмінності між різними представниками одного сімейства полягають, в основному, у наборі периферійних пристроїв і обсязі пам'яті програм та даних. Найбільш важлива особливість сімейства - програмна сумісність на рівні двійкового коду всіх МК які входять в сімейство.

MCS-51

Мікроконтролер Intel 8051, що вийшов у 1980 році, став класичним зразком пристроїв даного класу. Цей 8-бітний мікроконтролер поклав початок цілому сімейству мікроконтролерів з архітектурою CISC, які панували на ринку аж до недавнього часу. Більшість фірм виробників мікроконтролерів і сьогодні випускають пристрої, засновані на цій архітектурі. Серед них Philips, Atmel, Dallas, Silicon Labs, Siemens.

Microchip (PIC)

Мікроконтролери сімейства PIC (Peripheral Interface Controller) компанії Microchip об'єднують всі передові технології мікроконтролерів: електрично програмовані користувачем ППЗП, мінімальне енергоспоживання, високу продуктивність, добре розвинену RISC-архітектуру, функціональну закінченість і мінімальні розміри. Широка номенклатура виробів забезпечує використання мікроконтролерів в пристроях, призначених для різноманітних сфер вживання. Перші мікроконтролери компанії Microchip PIC16C5x з'явилися в кінці 1980-х років і завдяки своїй високій продуктивності і низькій вартості склали серйозну конкуренцію 8-розрядним МК, що вироблялися у той час, з CISC-архітектурою.

Atmel (AVR)

Справжня революція в світі мікроконтролерів сталася у 1996 році, коли корпорація Atmel представила своє сімейство чіпів на новому прогресивному ядрі AVR. Більш продумана архітектура AVR, швидкодія, що перевершує контролери Microchip, вдала цінова політика, сприяли відтоку симпатій багатьох розробників від недавніх претендентів на звання контролера номер один.

Мікроконтролери AVR мають більш розвинену систему команд, яка налічує до 133 інструкцій, продуктивність, що наближається до 1 мільйону операцій на 1 МГц, Flash ПЗП програм з можливістю внутрішньо-схемного програмування. AVR-архітектура оптимізована для мови високого рівня Cі.

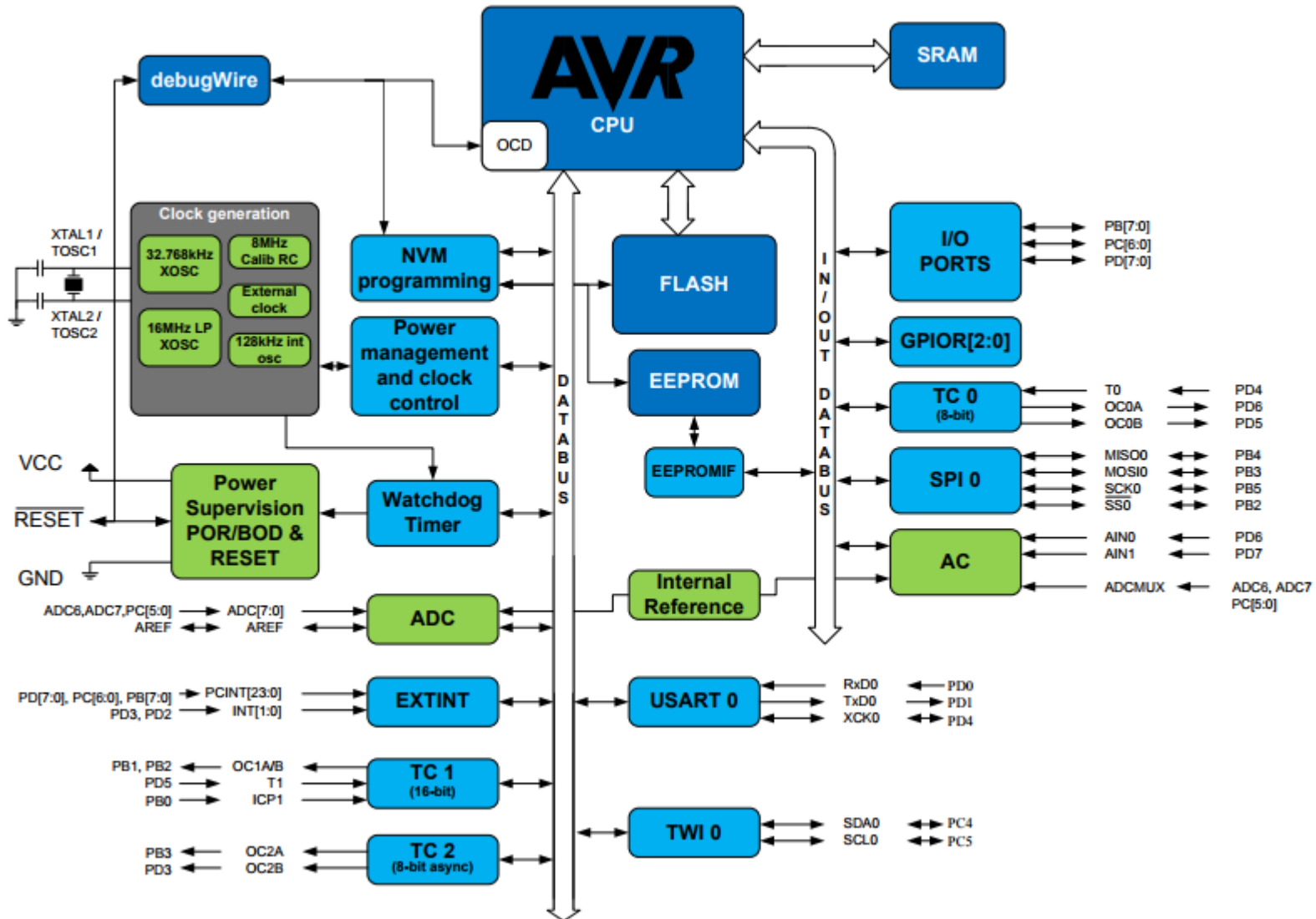
Величезну роль зіграла доступність програмного забезпечення та засобів розробки, У Atmel безкоштовно розповсюджуються багато програмних продуктів, наприклад надзвичайно вдале і абсолютно безкоштовне середовище розробки AVR Studio. Добре відомо, що розвинені засоби підтримки розробки при освоєнні і знайомстві з будь-яким сімейством мікроконтролерів грають не менш важливу роль, ніж самі мікроконтролери.

STMicroelectronics (STM32)

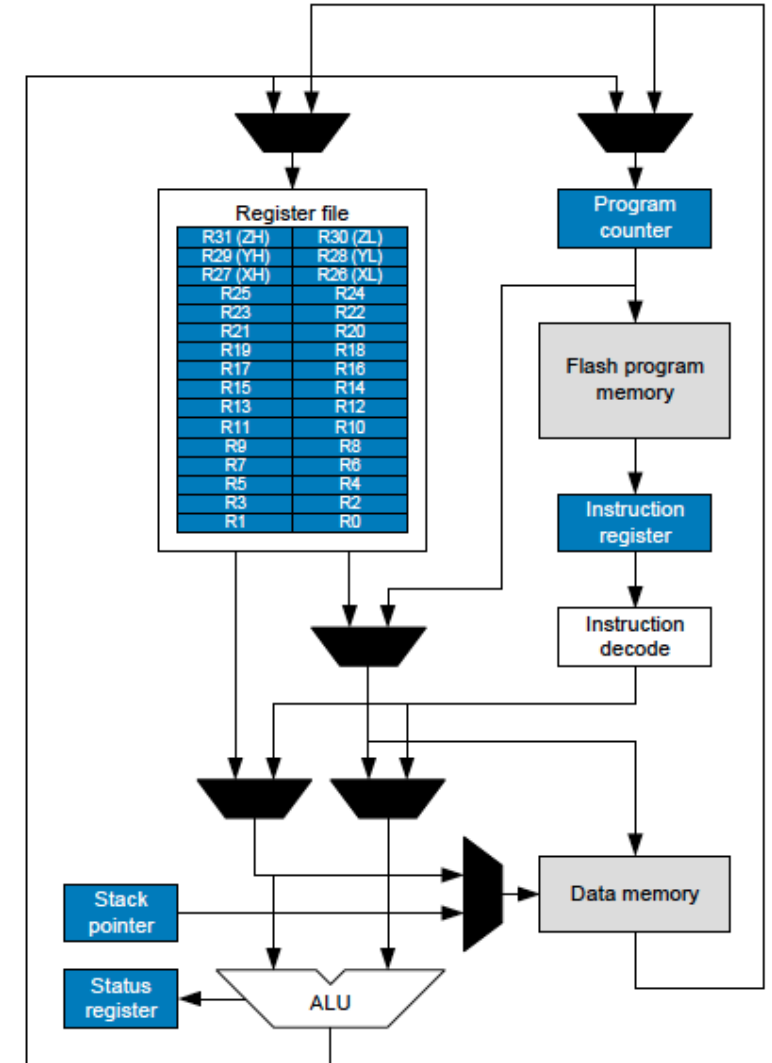
STMicroelectronics - європейська компанія, одна з найбільших, яка займаються розробкою, виготовленням і продажом різних напівпровідникових електронних і мікроелектронних компонентів. Сьогодні штаб квартира компанії знаходиться в Женеві, в той же час, її холдингова компанія STMicroelectronics зареєстрована в Амстердамі, однак компанія історично пов'язана з Італією і Францією. Компанія ST одна з перших випустила свої мікроконтролери з ядром Cortex-M0 (2007 р.) і швидко стала домінуючим гравцем на ринку 32-х бітних мікроконтролерів.

AVR мікроконтролер

Структурна схема мікроконтролера ATmega328



Структурна схема процесора AVR архітектури

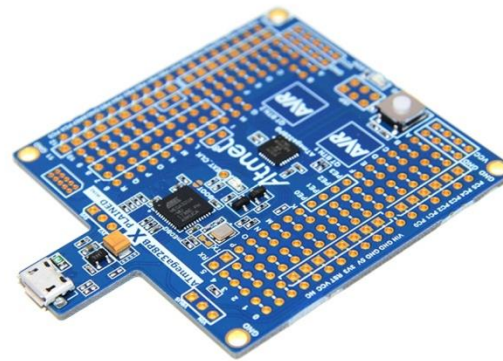


Плати для розробки (development kits)

Atmel STK600



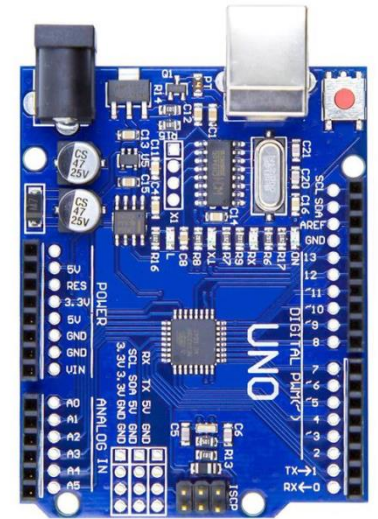
ATMEGA328PB-XMINI



Arduino nano



Arduino uno



Середовище розробки AVR Studio

The screenshot displays the AVR Studio IDE interface for an ATtiny1616 microcontroller. The main window shows the C source code for `driver_isr.c`, which is currently in the `Debug` state. The code is as follows:

```
/* Otherwise interrupt flag can be cleared by wr
 * in the STATUS register
 */
if(USART0.RXDATAH & USART_FERR_bm)
{
    idx = 0;
}

uint8_t temp = USART0.RXDATAH;

if(idx >= 0)
{
    data[idx++] = temp;
}

if(idx > 255)
{
    PWM_0_load_duty_cycle_ch0(255 - data[address]);
    PWM_0_load_duty_cycle_ch1(255 - data[address]);
    PWM_0_load_duty_cycle_ch2(255 - data[address]);
}
```

The `I/O` panel on the left shows the hardware configuration for the ATtiny1616, including the `General Purpose IO (GPIO)` and `I/O Ports (PORTA)`. The `Watch 1` panel displays the current state of the `data` array, which is a `uint8_t[10]` array. The `Disassembly` panel shows the assembly code for the `__vector_22` function, which is the interrupt service routine for the USART0 RX interrupt. The `Call Stack` panel shows the current call stack, which includes the `dmx.elf!__vector_22` function.

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Bits |
|--------|-------|------|
| DIR | 0x400 | 0x07 |
| DIRSET | 0x401 | 0x07 |
| DIRCLR | 0x402 | 0x07 |
| DIRTGL | 0x403 | 0x07 |
| OUT | 0x404 | 0x02 |
| OUTSET | 0x405 | 0x02 |
| OUTCLR | 0x406 | 0x02 |
| OUTTGL | 0x407 | 0x02 |
| IN | 0x408 | 0xFB |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |
| [3] | 254 | uint8_t |
| [4] | 105 | uint8_t |
| [5] | 0 | uint8_t |
| [6] | 0 | uint8_t |
| [7] | 0 | uint8_t |
| [8] | 0 | uint8_t |
| [9] | 0 | uint8_t |
| [10] | 0 | uint8_t |

| Name | Value | Type |
|------|--------|------------|
| data | 0x3e05 | uint8_t[2] |
| [0] | 0 | uint8_t |
| [1] | 0 | uint8_t |
| [2] | 42 | uint8_t |

Hello world! На мікроконтролері AVR

```
#include <avr/io.h>
#include <util/delay.h>
#define LED_PIN PB5

int main(void)
{
    DDRB |= 1 << LED_PIN;

    for(;;)
    {
        PORTB |= 1 << LED_PIN;
        _delay_ms(100);
        PORTB &= ~(1 << LED_PIN);
        _delay_ms(400);
    }
    return 0;
}
```

C vs Assembler

```
:100000000C940200259A05D02D9A03D02D98FBCF91
:10001000089529E236E940E84A95F1F73A95E1F783
:060020002A95D1F70895B6
:00000001FF
```

```
.equ PORTB, 0x05
.equ DDRB, 0x04

.text
.org 0x0000
    rjmp main

main:
    sbi DDRB, 5
_main_loop:
    rcall Delay
    sbi PORTB, 5
    rcall Delay
    cbi PORTB, 5
    rjmp _main_loop

Delay:
    ldi r18, 41
    ldi r19, 150
    ldi r20, 128
_Delay_loop:
    dec r20
    brne _Delay_loop
    dec r19
    brne _Delay_loop
    dec r18
    brne _Delay_loop
    ret

.end
```

