МПиМПС

Лекция 6.

Программирование на С

Функция main.

```
#include <avr/io.h>
int main(void)
{
    /* Replace with your application code */
    while (1)
    {
    }
}
```

Работать с памятью и стеком напрямую больше не нужно.

Компилятор может оптимизировать ваши переменные

Обращаться к регистрам нужно по прежнему по именам!

```
PORTB |= (1 << 3);
```

Прерывания

Подключить с помощью заголовочного файла

```
#include <avr/interrupt.h>
```

Использование:

```
ISR(USART_RXC_vect)
{
    switch(UDR)
    {
        case '1': PORTB |= (1 << 3);
        case '0': PORTB &= ~(1 << 3);
        default: break;
    }
}
```

Просмотр сгенерированного кода в виде ассемблерных команд

Disassembly | main.c

Address: main

Viewing Options

```
0000003A  PUSH R1       Push register on stack
0000003B  PUSH R0       Push register on stack
0000003C  IN R0,0x3F        In from I/O location
0000003D  PUSH R0       Push register on stack
0000003E  CLR R1        Clear Register
0000003F  PUSH R24      Push register on stack
    switch(UDR)
00000040  IN R24,0x0C       In from I/O location
00000041  CPI R24,0x30      Compare with immediate
00000042  BREQ PC+0x04      Branch if equal
00000043  CPI R24,0x31      Compare with immediate
00000044  BRNE PC+0x03      Branch if not equal
        case '1': PORTB |= (1 << 3);
00000045  SBI 0x18,3         Set bit in I/O register
        case '0': PORTB &= ~(1 << 3);
00000046  CBI 0x18,3         Clear bit in I/O register
}
00000047  POP R24       Pop register from stack
00000048  POP R0        Pop register from stack
00000049  OUT 0x3F,R0       Out to I/O location
0000004A  POP R0        Pop register from stack
0000004B  POP R1        Pop register from stack
0000004C  RETI      Interrupt return
```

Disassembly | main.c

Address: main

Viewing Options

```
--- No source file -------------------------------
00000000  JMP 0x0000002A        Jump
00000002  JMP 0x00000034        Jump
00000004  JMP 0x00000034        Jump
00000006  JMP 0x00000034        Jump
00000008  JMP 0x00000034        Jump
0000000A  JMP 0x00000034        Jump
0000000C  JMP 0x00000034        Jump
0000000E  JMP 0x00000034        Jump
00000010  JMP 0x00000034        Jump
00000012  JMP 0x00000034        Jump
00000014  JMP 0x00000034        Jump
00000016  JMP 0x0000003A        Jump
00000018  JMP 0x00000034        Jump
0000001A  JMP 0x00000034        Jump
0000001C  JMP 0x00000034        Jump
0000001E  JMP 0x00000034        Jump
00000020  JMP 0x00000034        Jump
00000022  JMP 0x00000034        Jump
00000024  JMP 0x00000034        Jump
00000026  JMP 0x00000034        Jump
00000028  JMP 0x00000034        Jump
--- ../../../../crt1/gcrt1.S ---------------------
0000002A  CLR R1        Clear Register
0000002B  OUT 0x3F,R1       Out to I/O location
0000002C  LDI R28,0x5F      Load immediate
0000002D  LDI R29,0x04      Load immediate
0000002E  OUT 0x3E,R29      Out to I/O location
0000002F  OUT 0x3D,R28      Out to I/O location
```