Руководство по интеграции модулей OR-AVR-M128-* в среду Arduino IDE

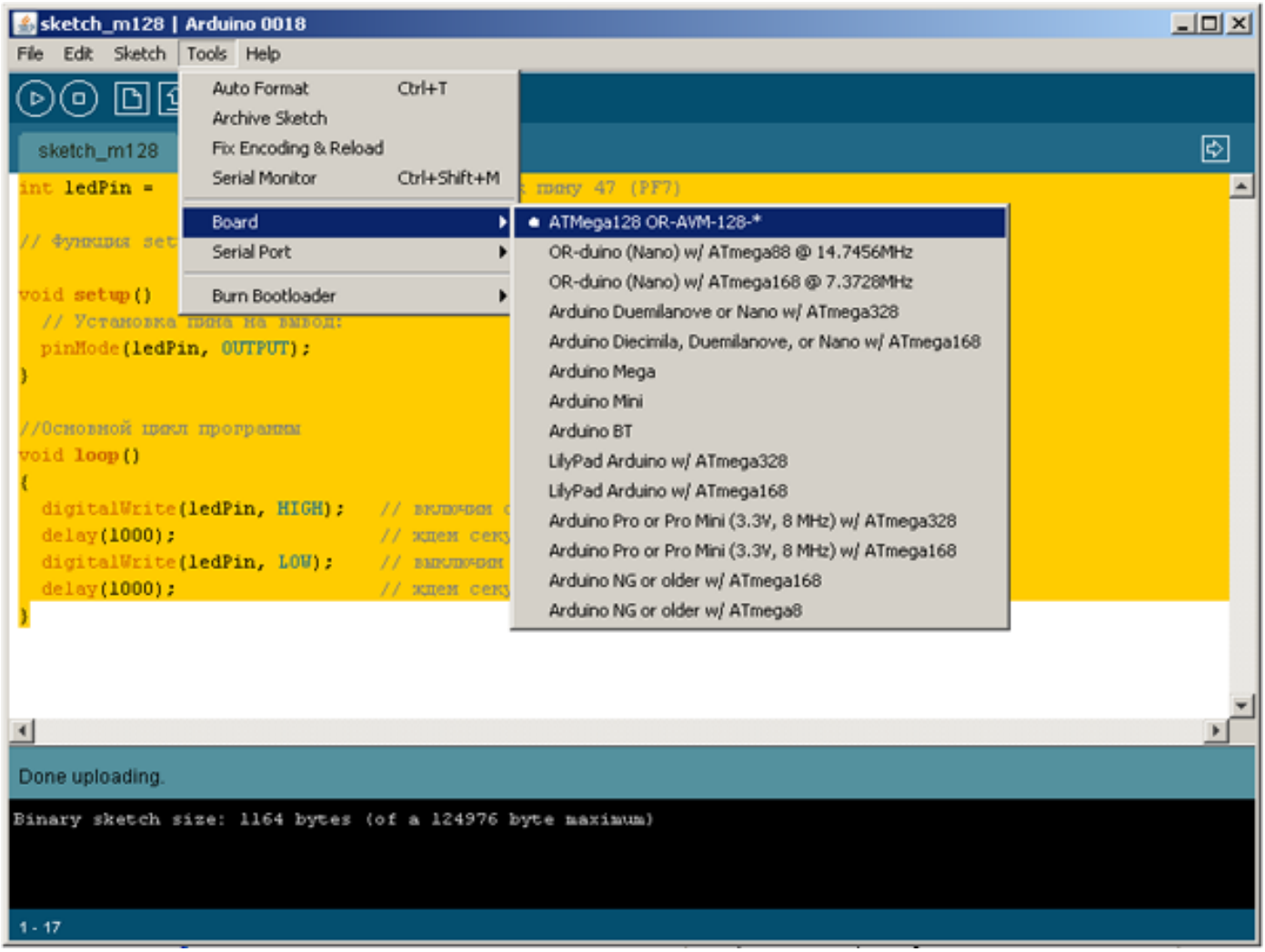
Для реализации интеграции модуля OR были использованы следующие аппаратные и программные средства:

1) Модуль OR-AVR-M128-S
2) Среда разработки Arduino IDE v. 0018

Порядок установки среды Arduino IDE и интеграции в неё модулей OR-AVR-M128-*

Шаг 1. Установка среды Arduino IDE

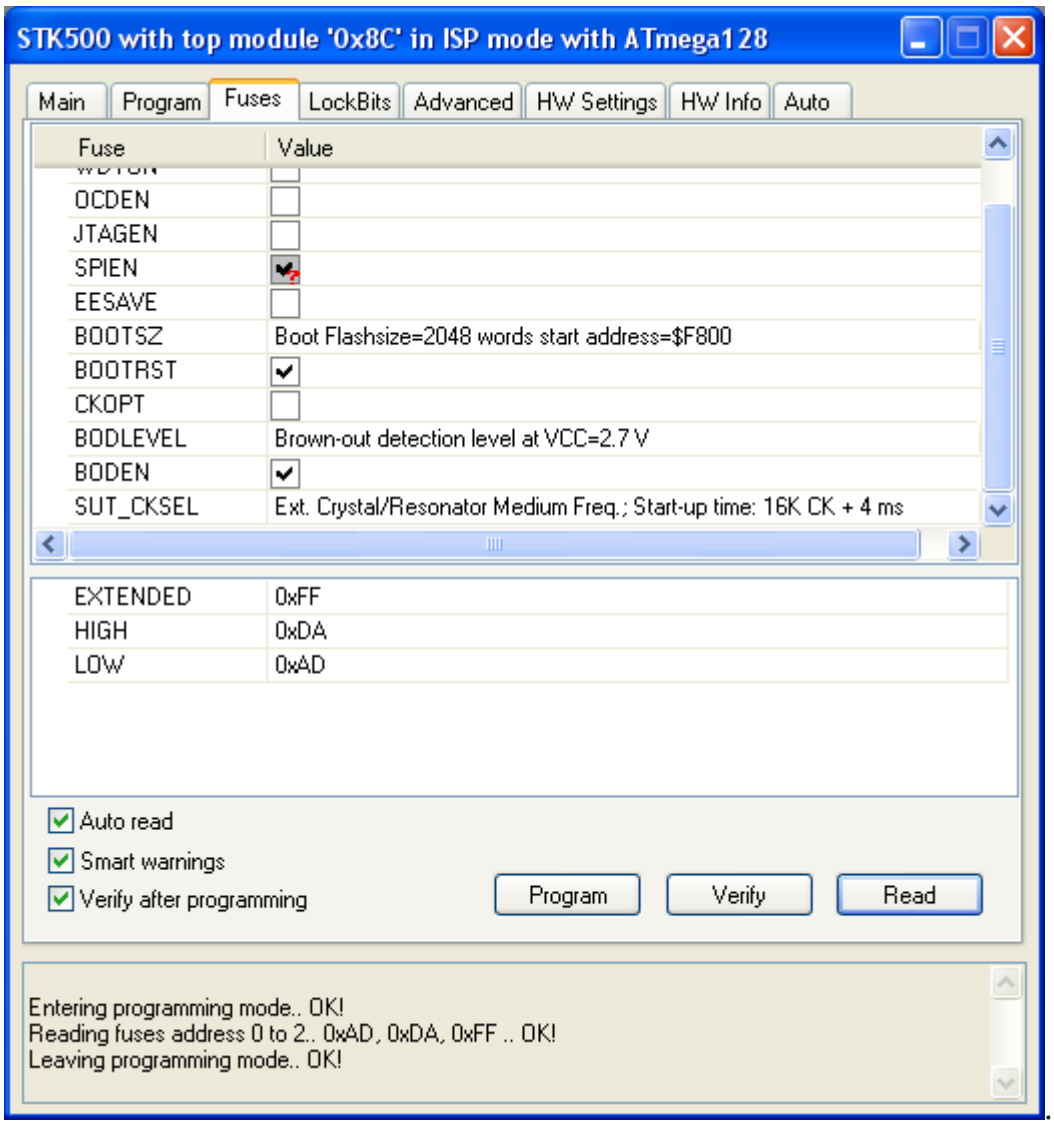
Для начала заходим на сайт Arduino: http://www.arduino.cc/ и скачиваем последнюю версию Arduino IDE. Полученный архив распаковываем и сразу же производим следующую модификацию - распаковываем архив arduino-m128.zip в папку arduino-00XX\hardware\. После этого среда начинает «понимать» контроллеры OR-AVR-M128-*. Запускаем IDE и убеждаемся в появлении пункта меню **OR-AVR-M128-* w/ATMega128 @ 7.3728MHz:**



Далее на контроллер устанавливаем загрузчик

Шаг 2. Установка загрузчика

Берем из каталога arduino-00XX\hardware\arduino\bootloaders\atmega128 файл ATmegaBOOT_128_atmega128.hex и прошиваем в контроллер OR-AVR-M128-* со следующими фьюз-битами:



Внимание!!! Обязательно нужно запрограммировать фузии бит M103C – нам не нужна поддержка этого вида МК, потому как с ней бутлоадер не работает.

Внимание!!! При прошивке программы в МК из IDE, если вы не пользуетесь модулем OR-USB-UART, после появления сообщения компилятора о размере кода, следует нажать и удерживать 3-4 сек кнопку Reset на контроллере.

Параметры загрузчика:
Микроконтроллер: ATMega128
Тактовая частота: 7,3728 МГц от внешнего кварца
Размер загрузочной области: 2048 Б
Программатор: stk500v2
Параметры для подключения: 115200 8N1

Работа с IDE, примеры

Для того чтобы прошить МК из IDE, выполняем следующие действия:

1) Открываем скетч
2) В меню Tools->Serial Port выбираем порт к которому подключена плата OR
3) В меню Tools->Board выбираем пункт ATMega128 OR-AVM-128-S
4) Собственно открываем скетч: File->Open
5) Проверяем программу на отсутствие ошибок Sketch->Verify / Compile
6) Прошиваем МК File->Upload to I/O Board
7) На плате OR нажимаем Reset и наблюдаем выполнение программы

2 примера (скетча): мигание светодиодом и пример работы кнопки

Пример 1. blink_led

```
/*
  Скетч blink_led
  Зажигает светодиод на 1 секунду и тушит его в цикле
  Схема:
  Светодиод подключен к пину PF7 одной ножкой и на землю другой
  Распиновка ATmega128:
  8-15 PC0-PC7
  16-23 PA0-PA7
  24-31 PB0-PB7
  32-39 PE0-PE7
  40-47 PF0-PF7
  48-50 PG0-PG2
  0-7 PE0-PE7
*/

int ledPin =  47;    // Светодиод подключен к пину 47 (PF7)

// Функция setup() вызываетс раз при запуске скетча

void setup()   {
  // Установка пина на вывод:
  pinMode(ledPin, OUTPUT);
}

//Основной цикл программы
void loop()
{
  digitalWrite(ledPin, HIGH);   // включим светодиод
  delay(1000);                  // ждем секунду
  digitalWrite(ledPin, LOW);    // выключим светодиод
  delay(1000);                  // ждем секунду
}
```

Пример 2 button_ex

```
/*
  Button_ex

  Зажигает и отключает светодиод, подключенный к PF7 пин 47,
  когда нажата кнопка, подключенная к PF6 пин 46.
  Схема:
  Светодиод подключен к пину PF7 одной ножкой и на землю другой
  Кнопка подключена к PF6 одной ножкой и на питание другой
  Подтягивающий резистор подключен к PF6 одной ножкой и к земле другой
  Распиновка ATmega128:
  8-15 PC0-PC7
  16-23 PA0-PA7
  24-31 PB0-PB7
  32-39 PE0-PE7
  40-47 PF0-PF7
  48-50 PG0-PG2
  0-7 PE0-PE7
 */

const int buttonPin = 46;     // Кнопка подключена к пину 46 (PF6)
const int ledPin =  47;      // Светодиод подключен к пину 47 (PF7)

int buttonState = 0;         // Статус кнопки

void setup() {
  // установка пина на вывод
  pinMode(ledPin, OUTPUT);
  // установка пина на ввод
  pinMode(buttonPin, INPUT);
}
//Основной цикл программы
void loop(){
  // читаем состояние кнопки
  buttonState = digitalRead(buttonPin);

  // нажата ли кнопка?
  // если да, то на пине будет 1
  if (buttonState == HIGH) {
    // включаем светодиод
    digitalWrite(ledPin, HIGH);
  }
  else {
    // выключаем светодиод
    digitalWrite(ledPin, LOW);
  }
}
```