

# Iskra JS: JavaScript В МИКРОКОНТРОЛЛЕРЕ

Игорь Зотов, разработчик

twitter: amperkaru

email: morgan@amperka.ru

youtube: AmperkaRu

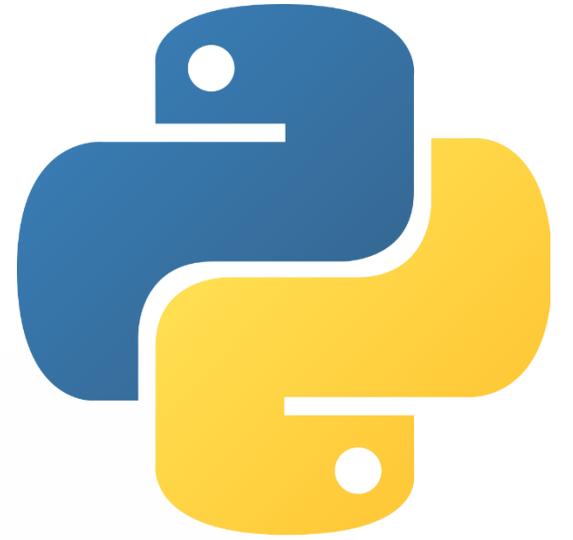
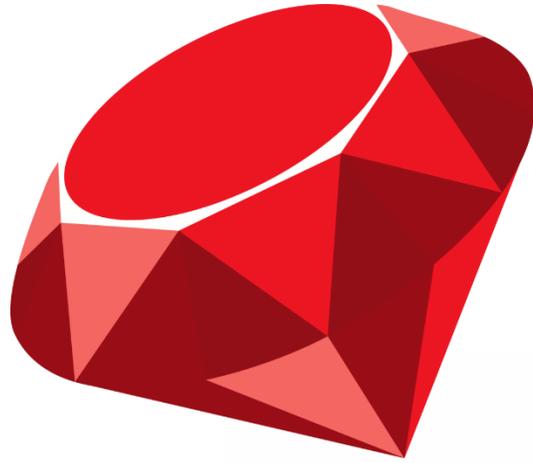


Амперка

**THE**

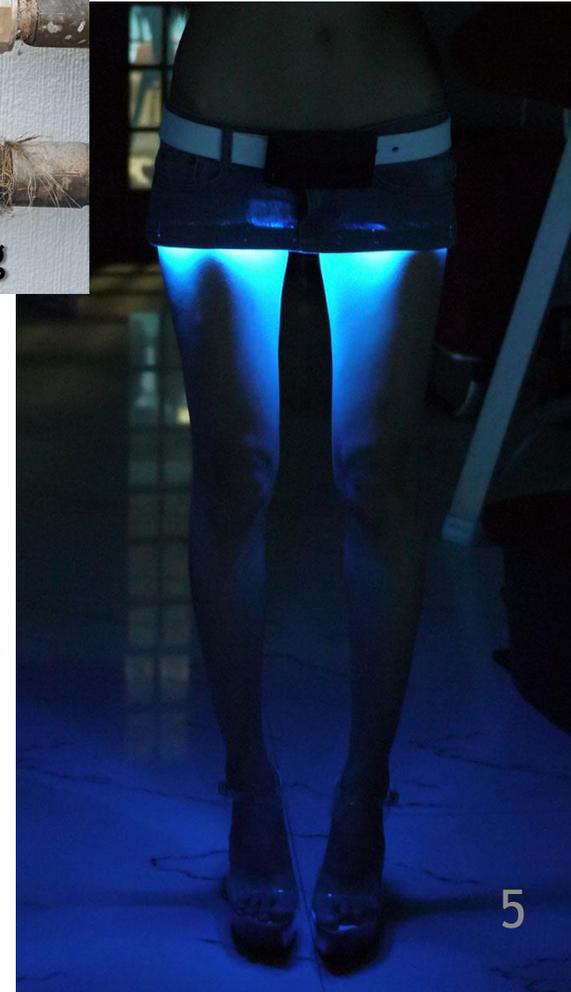
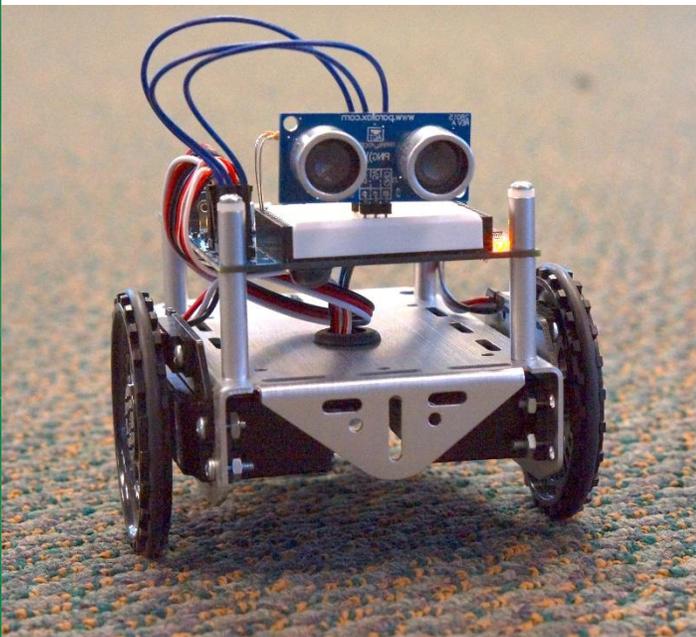
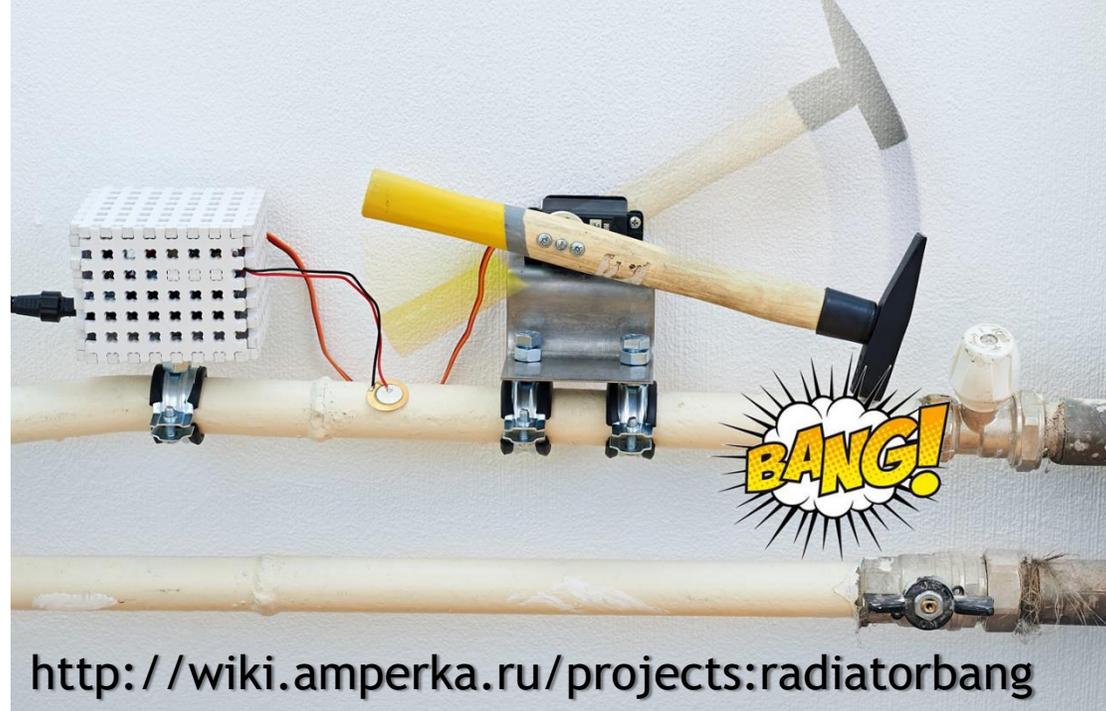
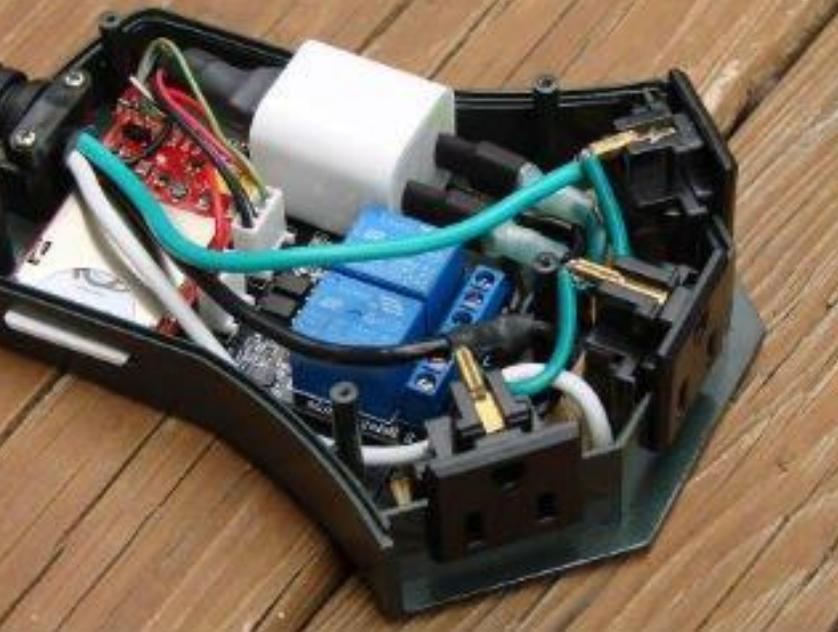
**C**

**PROGRAMMING  
LANGUAGE**



# Что такое программирование МК?

- Долго
- Сложно
- Неудобно



```
#include <SoftwareSerial.h>
#include <GPRS_Shield_Arduino.h>

#include <Adafruit_NeoPixel.h>

#define PIXEL_PIN 6
#define PIXEL_COUNT 200

#define MESSAGE_OK "Well done! "\
"You are hacked our Christmas Tree. Happy New Year 2016!"
#define MESSAGE_ERROR "Whoops! Something wrong."\
>Your attempt is failed."
#define MESSAGE_BONUS "You are hacked our Christmas Tree."\
"Secret mod has been activated. Happy New Year 2016!"

#define MESSAGE_LENGTH 160

char message[MESSAGE_LENGTH];
char phone[16];
char datetime[24];

GPRS gprs;

Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL_COUNT, PIXEL_PIN, NEO_BRG + NEO_KHZ800);
```

```

int r, g, b;

void setup()
{
  strip.begin();
  strip.show();
  gprs.powerUpDown();
  while (!gprs.init()) {
    delay(1000);
  }
  colorWipe(strip.Color(190, 0, 255), 50);
  colorWipe(strip.Color(0, 0, 0), 50);
}

void loop()
{
  if (gprs.ifSMSNow()) {
    gprs.readSMS(message, phone, datetime);
    if (checkSMS() == 0) {
      gprs.sendSMS(phone, MESSAGE_ERROR);
      return;
    }
  }
}

```

```

int value = atoi(message);
if (value >= 0 && value <= 360) {
  gprs.sendSMS(phone, MESSAGE_OK);
  f_HSV_to_RGB(value, 255, 255);
  colorWipe(strip.Color(r, g, b), 50);
} else if (value == 2016) {
  gprs.sendSMS(phone, MESSAGE_BONUS);
  theaterChaseRainbow(50);
  colorWipe(strip.Color(r, g, b), 50);
} else {
  gprs.sendSMS(phone, MESSAGE_ERROR);
}
}

bool checkSMS() {
  int len = strlen(message);
  if (len == 0 || len > 4) {
    return false;
  }
  for (int i = 0; i < len; i++) {
    if (message[i] < '0' || message[i] > '9') {
      return false;
    }
  }
}

```

```

void f_HSV_to_RGB(int hue, int sat, int val)
{
    int base;
    if (sat == 0) {
        r = val;
        g = val;
        b = val;
    } else {
        base = ((255 - sat) * val) >> 8;
        switch (hue / 60) {
            case 0: {
                r = val;
                g = (((val - base) * hue) / 60) + base;
                b = base;
                break;
            }
            case 1: {
                r = (((val - base) * (60 - (hue % 60))) / 60) + base;
                g = val;
                b = base;
                break;
            }

```

```

        case 2: {
            r = base;
            g = val;
            b = (((val - base) * (hue % 60)) / 60) + base;
            break;
        }
        case 3: {
            r = base;
            g = (((val - base) * (60 - (hue % 60))) / 60) + base;
            b = val;
            break;
        }
        case 4: {
            r = (((val - base) * (hue % 60)) / 60) + base;
            g = base;
            b = val;
            break;
        }
        case 5: {
            r = val;
            g = base;
            b = (((val - base) * (60 - (hue % 60))) / 60) + base;
            break;
        }

```

```

case 6: {
    r = val;
    g = 0;
    b = base;
    break;
}
}
}
}
void colorWipe(uint32_t c, uint8_t wait)
{
    for (uint16_t i = 0; i < strip.numPixels(); i++) {
strip.setPixelColor(i, c);
        strip.show();
delay(wait);
    }
}

```

```

void theaterChaseRainbow(uint8_t wait)
{
    for (int j = 0; j < 256; j++) {
        for (int q = 0; q < 3; q++) {
            for (int i = 0; i < strip.numPixels(); i = i + 3) {
strip.setPixelColor(i + q, wheel((i+j) % 255));
            }
            strip.show();
delay(wait);
            for (int i = 0; i < strip.numPixels(); i = i + 3) {
strip.setPixelColor(i+q, 0);
            }
        }
    }

uint32_t wheel(byte wheelPos)
{
    wheelPos = 255 - wheelPos;
    if (wheelPos < 85) {
        return strip.Color(255 - wheelPos * 3, 0, wheelPos * 3);
    }
    if (wheelPos < 170) {
        wheelPos -= 85;
        return strip.Color(0, wheelPos * 3, 255 - wheelPos * 3);
    }
}

```

# А потом появился Espruino

- Простой
- Гибкий
- Ёмкий

# А потом появился Espruino

- Простой
- Гибкий
- Ёмкий

*На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами.*

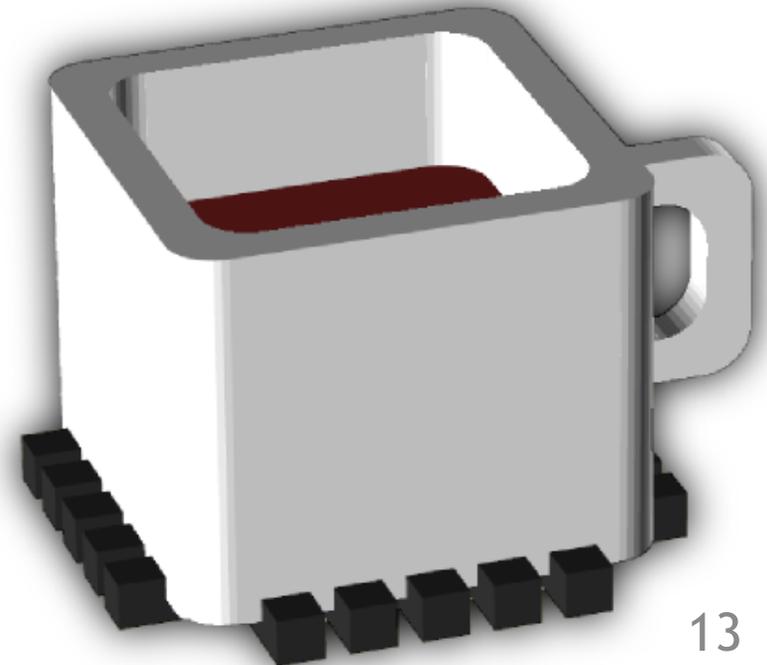
*- wikipedia.*

```
1 // SMS
2 var sim = require('@amperka/Sim900r').connect();
3 sim.powerOn();
4 sim.on('sms', function(sms) {
5     var rgb = sms.text.split(' ');
6     var r = +rgb[0];
7     var g = +rgb[1];
8     var b = +rgb[2];
9     if (isNaN(r + g + b)) return;
10    fadeTo(r, g, b);
11 });
12
13 // Лента
14 SPI2.setup({baud:3200000, mosi:B15});
15 var LED_COUNT = 150;
16 var colors = new Uint8ClampedArray(LED_COUNT * 3);
17
18 function fadeTo(r, g, b) {
19     var n = 0;
20     var ivalID = setInterval(function() {
21         colors[n*3 + 0] = b;
22         colors[n*3 + 1] = r;
23         colors[n*3 + 2] = g;
24         SPI2.send4bit(colors, 0x01, 0x03);
25         if (n++ >= LED_COUNT) {
26             clearInterval(ivalID);
27         }
28     }, 50);
29 }
```

# JS

# Ядро Espruino

- Espruino Web IDE — среда программирования;
- Espruino Firmware — JavaScript-машина
- Стандартная библиотека и внешние библиотеки;
- Железо — сами платы;
- Документация.





# Espruino Web IDE

This app is designed for [Espruino JavaScript Microcontrollers](#). Please see the [website](#) and [Quick Start guide](#) for more information, tutorials and example projects.

For information on how to use this IDE, see our [guided tour](#).

Espruino is [Open Source](#). Please support us by [donating](#) or [buying one of our boards!](#)



```
1 var on = false;
2 setInterval(function() {
3   on = !on;
4   LED1.write(on);
5 }, 500);
```



# Espruino Web IDE

This app is designed for [Espruino JavaScript Microcontrollers](#). Please see the [website](#) and [Quick Start guide](#) for more information, tutorials and example projects.

For information on how to use this IDE, see our [guided tour](#).

Espruino is [Open Source](#). Please support us by [donating](#) or [buying one of our boards!](#)



```
1 var on = false;
2 setInterval(function() {
3     on = !on;
4     LED1.write(on);
5 }, 500);
```



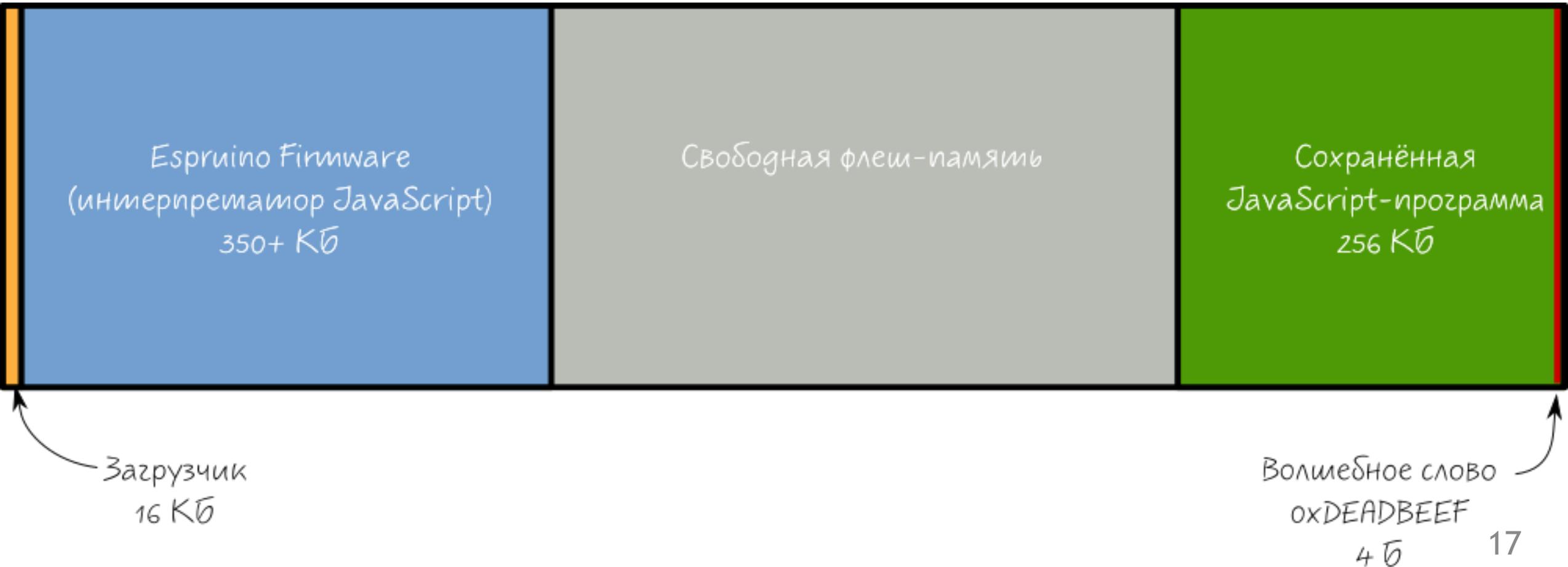
Connected

&gt;



```
1 var on = false;
2 setInterval(function() {
3   on = !on;
4   LED1.write(on);
5 }, 500);
```

# Espruino Firmware



# Минификатор

- Esprima (built-in)
- The Closure Compiler (online)



# ВОЗМОЖНОСТИ

- Стандартные

- `Array, Function, Number, Object, String`
- `pinMode, digitalRead, setWatch, digitalPulse`

- Встроенные модули

- `fs, http, Waveform, crypto, Flash`
- ```
var http = require('http');
```

- Внешние библиотеки

- `accelerometer, gprs-shield, robot`
- ```
var acc = require('@amperka/accelerometer');
```

# Отличия JavaScript от стандарта

- Точки с запятыми не расставляются автоматически на месте переноса строк
- Отсутствуют метки для инструкций break
- Отсутствуют регулярные выражения
- Символы строк хранятся в кодировке UTF-8, символы вне ASCII-диапазона занимают по 2 байта:

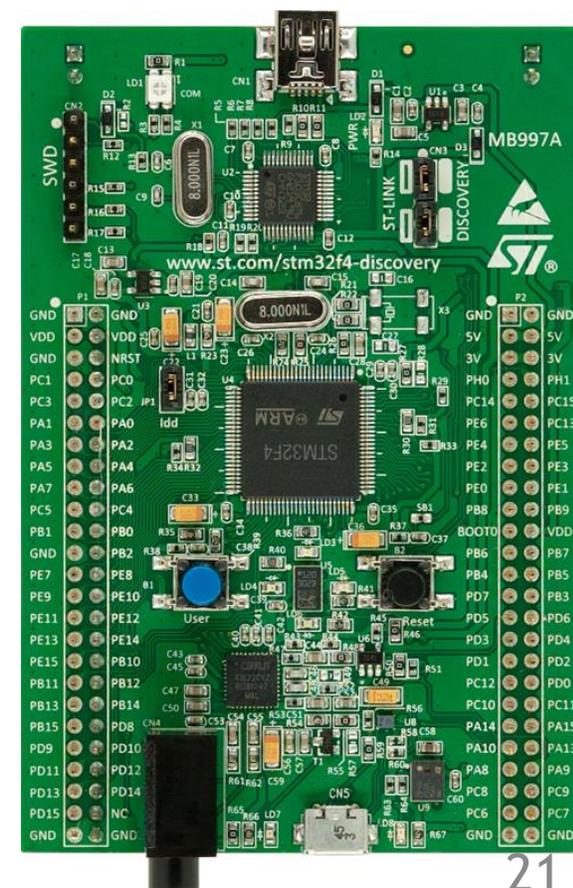
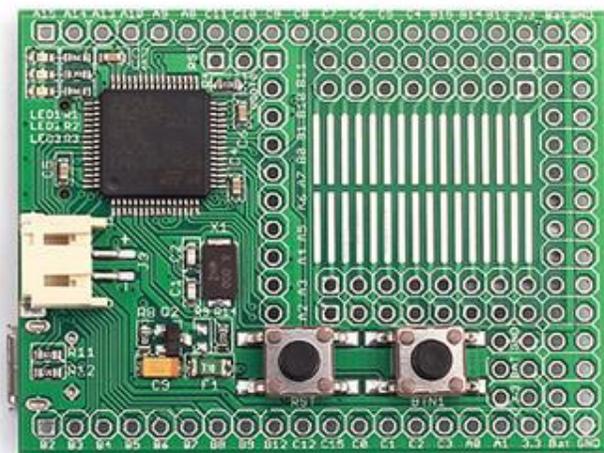
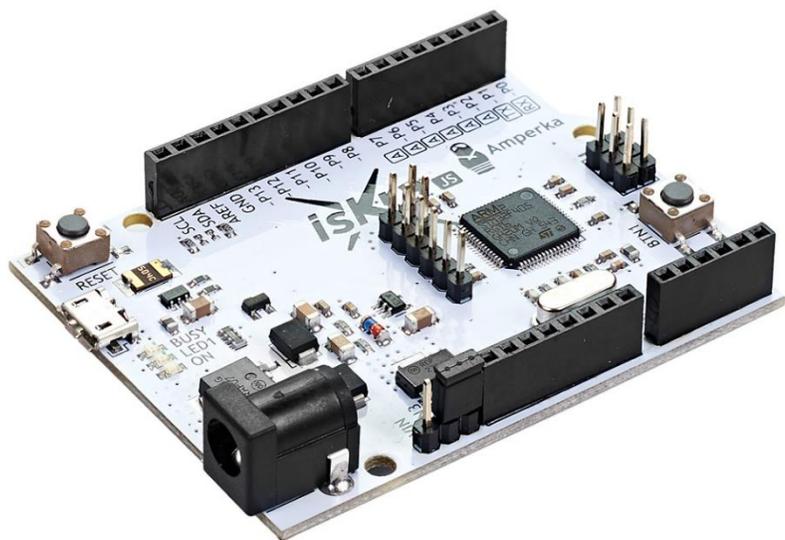
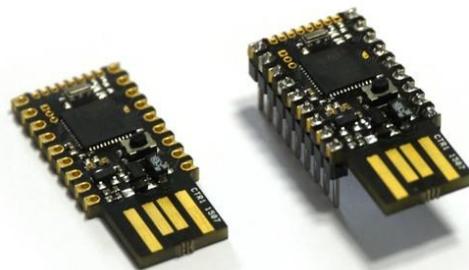
```
"SWD".length === 3;
```

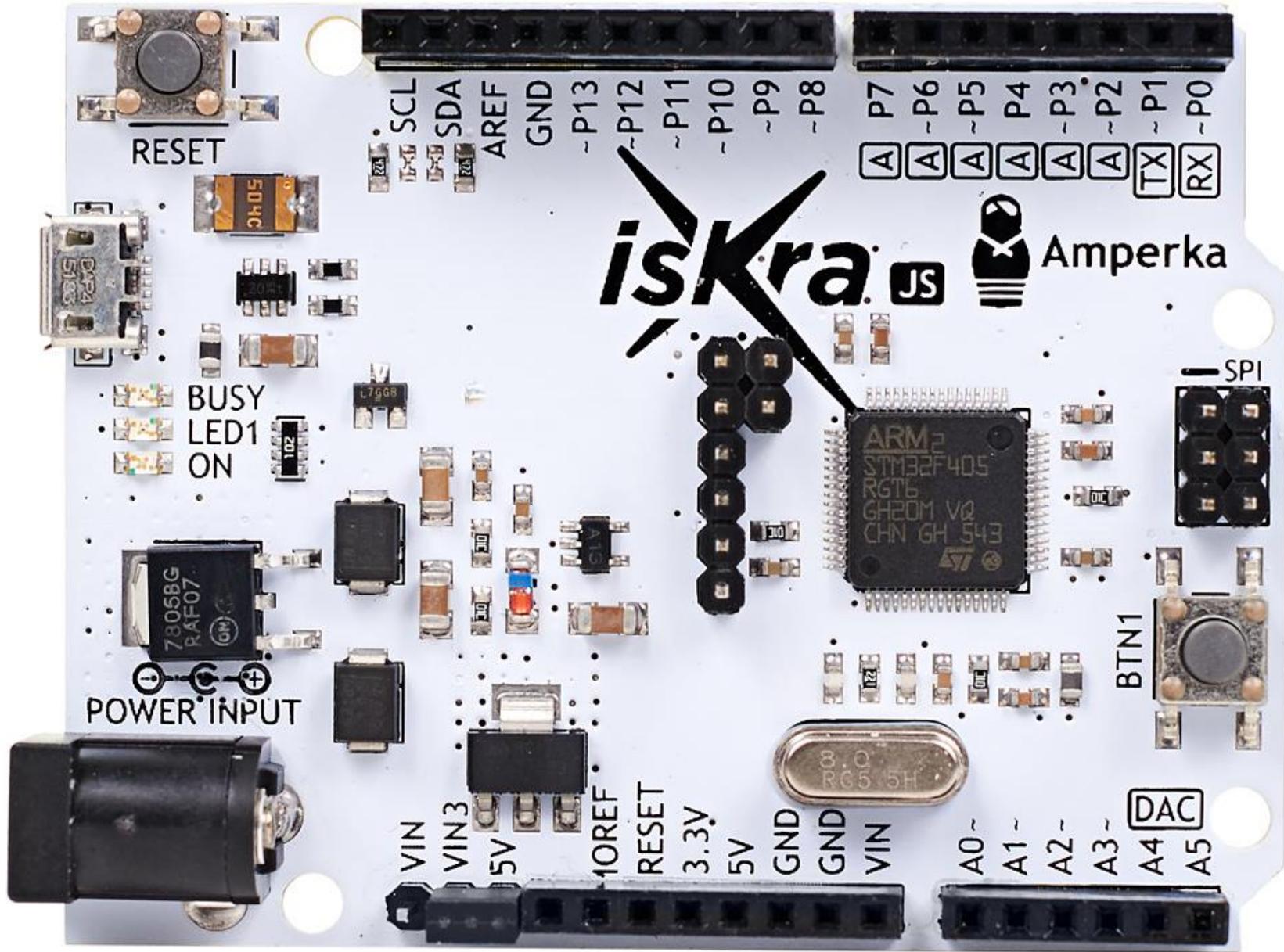
```
"ЫЙЦ".length === 6;
```

```
"SWЙ".length === 4;
```

# Железо

- STM32, в том числе платы DISCOVERY и NUCLEO
- ESP8266
- Iskra JS





# Ура, демонстрация!

```
var on = false;
setWatch(function() {
  on = !on;
  digitalWrite(P3, on);
}, P7, {
  edge: 'falling',
  repeat: true,
  debounce: 30
});
```

# Последнее. «Аплодисметр»

```
var myLed = require('@amperka/led')  
    .connect(P8);
```

```
myLed.turnOn().brightness(0);
```

```
// Регулируем яркость светодиода  
setInterval(function() {  
    var adc = analogRead(A1);  
    console.log(adc);  
    myLed.brightness(adc);  
}, 100);
```

# Высокоуровневое. Диодная лента

```
var numberLed = 20;
var ledStrip = require('@amperka/led-strip').connect(SPI2, numberLed, 'RBG');

function fadeTo(r, g, b) {
  var n = 0;
  var ivalID = setInterval(function() {
    ledStrip.putColor(n, {
      red: r,
      green: g,
      blue: b
    }).apply();
    if (n++ >= numberLed) {
      clearInterval(ivalID);
    }
  }, 50);
}

SPI2.setup({baud:3200000, mosi:B15});
```

# Высокоуровневое. GPRS +7-903-524-07-37

```
var sim = require('@amperka/Sim900r').connect();
```

```
sim.powerOn();
```

```
sim.on('sms', function(index) {  
  print('sms');  
  sim.smsRead(index, function(err, sms) {  
    sim.smsDelete('all');  
    if (err) return;  
    var rgb = sms.text.split(' ');  
    var r = +rgb[0];  
    var g = +rgb[1];  
    var b = +rgb[2];  
    if (isNaN(r + g + b)) return;  
    fadeTo(r, g, b);  
  });  
});
```

# Хочу ещё!

<http://js.amperka.ru>

<http://espruino.com>

# Спасибо!

Игорь Зотов

twitter: amperkaru

email: morga@amperka.ru

youtube: AmperkaRu



Амперка

Q&A