

Отладочные средства для микроконтроллеров STM32 от компании «Амперка»

Олег Вальпа (sandh@narod.ru)

В статье приведено описание отладочной платы Iskra JS и свободной среды разработки программ Espruino Web IDE от компании «Амперка». Представленные продукты предназначены для практического изучения микроконтроллеров серии STM32 компании STMicroelectronics.

ВВЕДЕНИЕ

Одним из ключевых факторов развития отечественного приборостроения, робототехники и других высокотехнологичных отраслей является привлечение молодых специалистов к работе с современной микропроцессорной техникой, а для этого следует обеспечить наличие необходимых источников информации и доступных аппаратных и программных средств обучения и разработки.

Для решения подобных задач многие компании предлагают широкий выбор отладочных плат и программных инструментов. Одной из них является компания «Амперка» [1], выпускающая отладочные платы для различных микроконтроллеров и периферийные устройства, которые легко подключаются к данным платам и имеют программную поддержку. Проекты, развиваемые компанией, во многом схожи с хорошо извест-

ным и распространённым проектом Arduino [2]. В статье представлен один из новых продуктов «Амперки» – отладочная плата Iskra JS, позволяющая быстро приобрести практические навыки работы с микропроцессорами серии STM32 [3].

Отладочная плата ISKRA JS

Отладочная плата Iskra JS является флагманской платой «Амперки», имеет встроенный интерпретатор JavaScript и представляет собой развитие платформы Espruino. Iskra JS обеспечивает высокую скорость выполнения программ, комфортность разработки, максимальную совместимость с платами расширения (в т.ч. с платами Arduino), сенсорами и другими электронными модулями. Внешний вид отладочной платы Iskra JS представлен на рисунке 1.

Основой платы Iskra JS является 32-битный микроконтроллер STM32F405RG с ядром ARM Cortex M4. Плата имеет встро-

енный в микроконтроллер интерфейс USB и не требует для подключения к компьютеру дополнительного моста USB-UART. Iskra JS работает на частоте 168 МГц, имеет 1 МБ флэш-памяти для хранения программы интерпретатора JavaScript и программы пользователя, а также 192 КБ оперативной памяти. Такого объёма памяти хватает для обработки JS-кода и решения множества задач, таких как управление роботом, промышленная автоматика, системы умного дома и т.д. На плате имеются совместимые с Arduino контактные соединители, расположенные в соответствии со стандартной компоновкой Arduino R3.

Напряжения логических уровней сигналов платы Iskra JS составляют 3,3 В, но многие выводы толерантны к 5 В уровням сигналов. Это обеспечивает электрическую совместимость с большей частью периферийных устройств Arduino.

Плата имеет 36 доступных управляющих контактов: 12 для считывания аналогового сигнала, 22 для генерации сигнала широтно-импульсной модуляции, 2 аналоговых выхода.

На платформе имеется 4 последовательных порта, 2 интерфейса SPI и 3 интерфейса I²C для взаимодействия с дисплеями, гироскопами, акселерометрами и другими сложными модулями. При этом по одному интерфейсу SPI и I²C вынесено на штатные разъёмы согласно стандарту Arduino R3 без создания конфликта с остальными выводами. Расположение выводов отладочной платы Iskra JS показано на рисунке 2. Выводы отладочной платы имеют следующее назначение:

- P0 – вывод GPIO (по этому имени можно обратиться к выводу из программы);
- B11 – вывод микроконтроллера STM32F405RG;
- 3.3V – вывод питания;
- PWM – вывод ШИМ сигнала;
- ADC – вывод АЦП микроконтроллера для чтения аналогового сигнала;
- DAC – вывод ЦАП микроконтроллера для формирования аналогового сигнала;
- FVT – вывод, толерантный к напряжению 5 В;

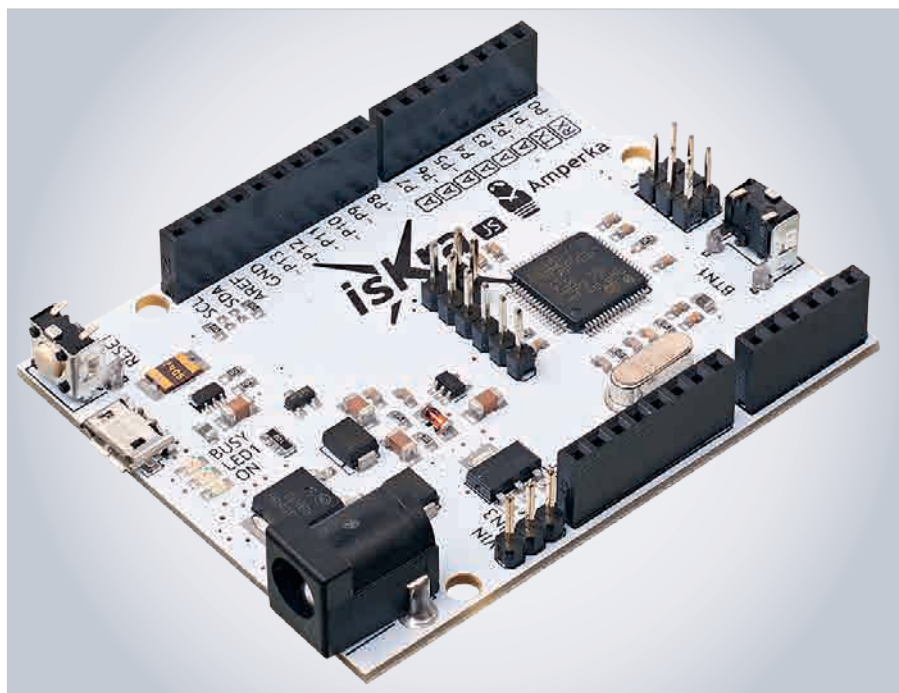


Рис. 1. Внешний вид отладочной платы Iskra JS

- SPI2 SCK – вывод сигнала синхронизации интерфейса SPI2 (программно доступен по имени PrimarySPI);
 - I2C1 SDA – вывод сигнала данных интерфейса I²C (программно доступен по имени PrimaryI2C);
- USART3 TX – вывод сигнала интерфейса UART/USART (программно доступен по имени PrimarySerial).

Такую отладочную плату можно приобрести или изготовить самостоятельно. Принципиальная электрическая схема отладочной платы Iskra JS приведена на рисунке 3, а её сборочный чертёж – на рисунке 4.

Питание платы можно осуществлять как от компьютера через порт microUSB, так и через гнездо питания 2,1 мм. Источник питания определяется платой автоматически. В случае питания через гнездо рекомендуемое входное напряжение составляет от 7 до 15 В. Встроенный в плату регулятор напряжения сформирует необходимые для платы напряжения 5 и 3,3 В. В качестве источника питания можно использовать сетевой адаптер, батарейки или аккумуляторы. Для питания

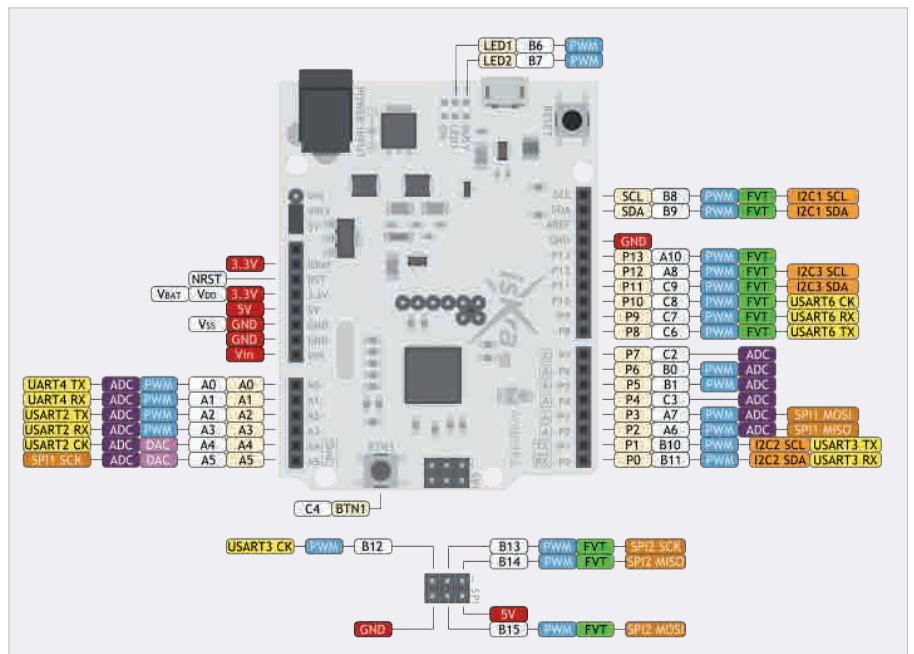


Рис. 2. Расположение выводов отладочной платы Iskra JS

подключённой периферии используются выводы 5V и 3.3V, которые выдают соответствующие напряжения. При использовании внешнего источника питания с вывода 5V можно получить

суммарный ток до 1000 мА, а с вывода 3.3V – до 300 мА. Если для внешних устройств не требуется напряжение 5 В, то можно запитать плату от источника напряжением от 3,6

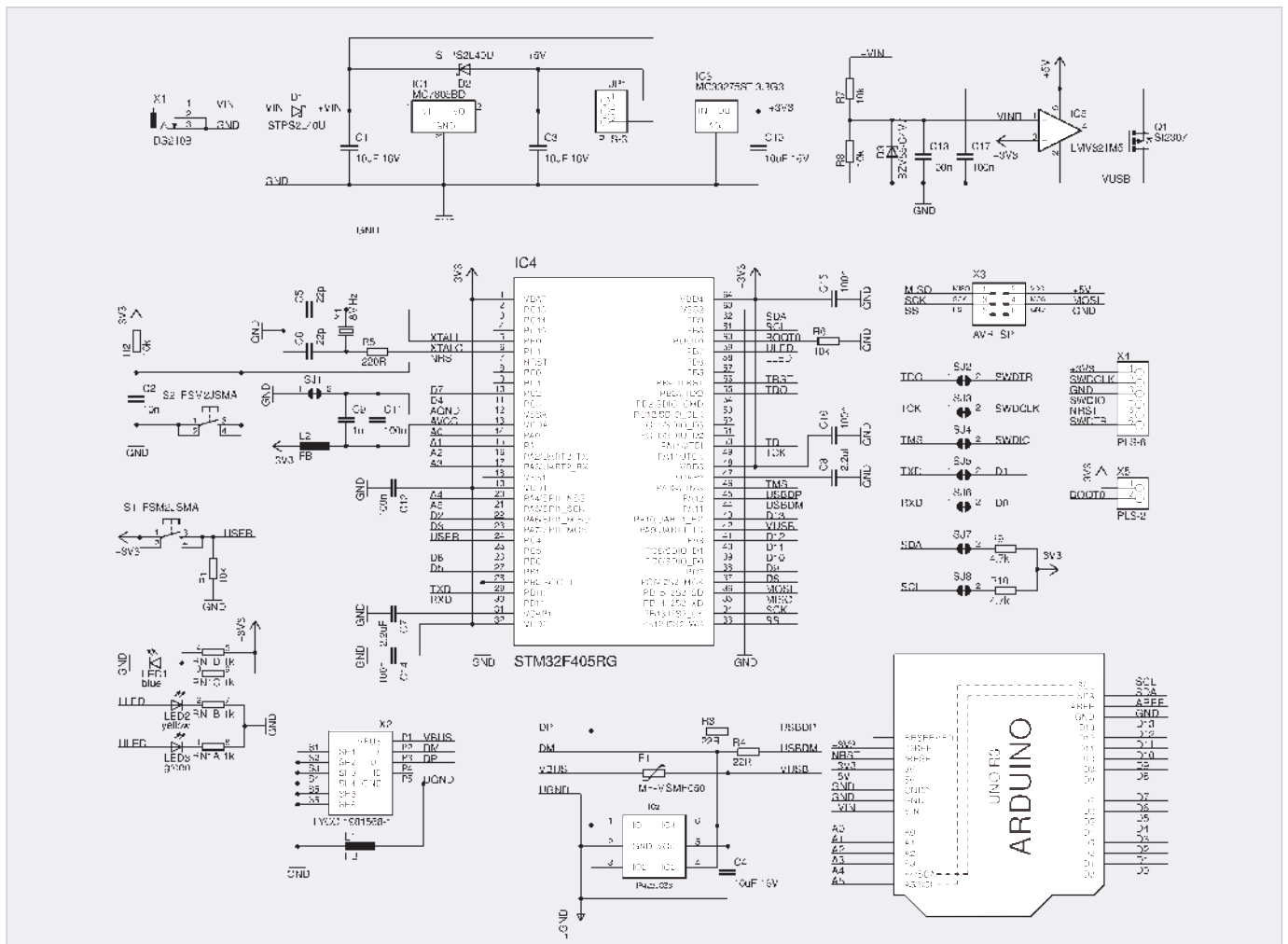


Рис. 3. Принципиальная электрическая схема отладочной платы Iskra JS

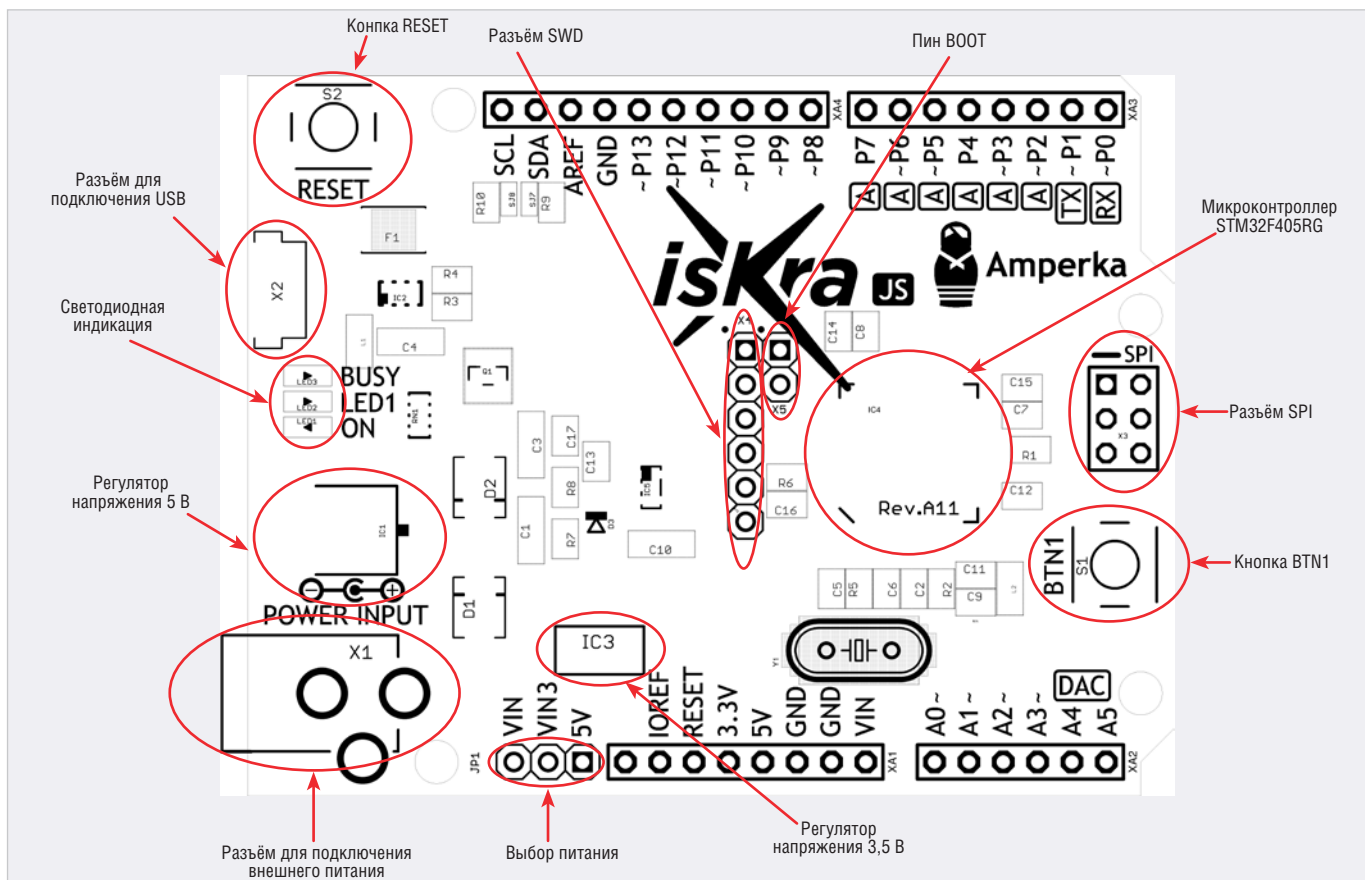


Рис. 4. Сборочный чертёж отладочной платы Iskra JS

до 12 В, например от литий-ионного аккумулятора.

Плата Iskra JS подключается к компьютеру кабелем microUSB. При подключении платформа обнаруживается компьютером как виртуальный последовательный порт. По этому порту загружаются пользовательские программы и осуществляется чтение отладочной информации. Данные процессы автоматизированы в среде разработки Iskra IDE. Кроме того, возможно запрограммировать плату Iskra JS как HID-устройство, то есть эмулировать с его помощью работу клавиатуры, манипулятора мыши, джойстиков и другого компьютерного оборудования.

Плата Iskra JS снабжена предохранителем, защищающим USB-порты компьютера от перенапряжения и коротких замыканий. Хотя большинство компьютеров обладает собственными средствами защиты, предохранитель разрывает соединение, если от USB-порта потребляется ток более 500 мА, и восстанавливает его после нормализации ситуации.

Размеры платы составляют 69×53 мм, гнёзда для внешнего питания и USB выступают на 2 мм за её габариты. По краям платы расположены отверстия

для винтового крепления. Расстояние между контактами составляет 2,54 мм.

Плата Iskra JS имеет следующие технические характеристики:

- микроконтроллер – 32-битный STM32F405RG с ARM-ядром Cortex M4;
- тактовая частота – 168 МГц;
- объём флэш-памяти – 1024 КБ;
- объём флэш-памяти для хранения программы – 256 КБ;
- объём SRAM-памяти – 192 КБ;
- количество портов ввода-вывода общего назначения – 26;
- количество портов с поддержкой ШИМ – 22;
- количество портов АЦП – 12;
- разрядность АЦП – 12 бит;
- количество портов ЦАП – 2;
- разрядность ЦАП – 12 бит;
- количество портов, толерантных к 5 В, – 11;
- доступные аппаратные интерфейсы – 4×UART/Serial, 3×I²C/TWI, 2×SPI;
- номинальное рабочее напряжение – 3,3 В;
- рекомендуемое входное напряжение – 7...15 или 3,6...12 В;
- максимальный ток от источника 5 В – 1000 мА;
- максимальный ток от источника с шины 3,3 В – 300 мА (включая питание микроконтроллера);

- максимальный ток на выводе – 25 мА;
- максимальный суммарный ток выводов – 240 мА;
- рабочий температурный режим –20...+70°C;
- размеры – 69×53 мм.

На рисунке 4 приведено расположение органов управления и коммутации отладочной платы Iskra JS. Переключатель выбора питания определяет, как осуществляется преобразование входного напряжения. Она может быть установлена в одном из двух положений:

1. VIN3+5V – внешнее питание сначала проходит регулятор на 5 В, а он, в свою очередь, питает регулятор на 3,3 В; работают обе шины: 5 В и 3,3 В; допустимое входное напряжение в этом случае составляет от 7 до 15 В.
2. VIN3+VIN – внешнее питание сразу подаётся на регулятор 3,3 В; шина 5 В не используется для получения напряжения 3,3 В; допустимое входное напряжение – от 3,6 до 12 В; данное положение рекомендуется использовать при питании схемы от литий-ионного аккумулятора.

Когда плата подключена к внешнему источнику питания и переключатель выбора питания установлена в положение VIN3+5V, напряжение прохо-

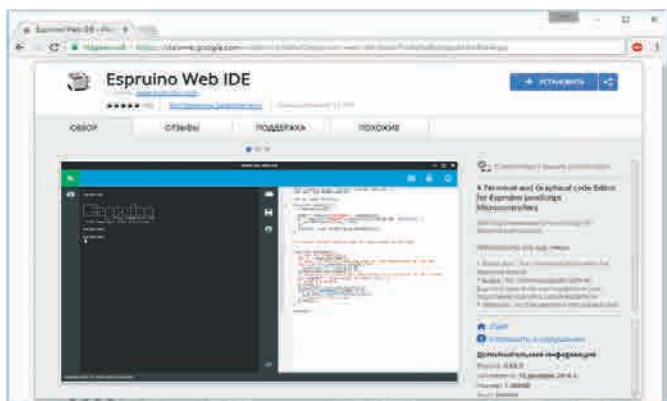


Рис. 5. Установка среды разработки

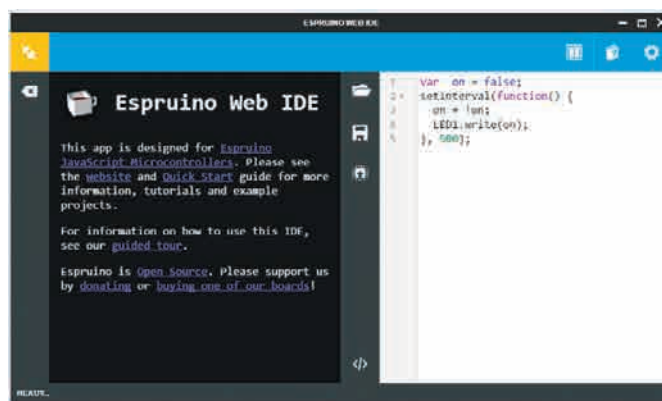


Рис. 6. Окно среды разработки

дит через стабилизатор MC7805BD. Выход стабилизатора соединён с выводом 5V. Максимальный выходной ток составляет 1000 мА. Стабилизатор MC33275ST с выходным напряжением 3,3 В обеспечивает питание микроконтроллера STM32F405RG. Максимальный выходной ток составляет 300 мА.

На плате установлено 3 светодиода:

1. BUSY – индикатор «занятости» системы.
2. LED1 – светодиод свободного назначения.
3. ON – индикатор наличия питания на микроконтроллере.

Кнопка *RESET* служит для сброса микроконтроллера, кнопка *BTN1* программируется пользователем и подключена к выводу PC4 микроконтроллера.

Разъём SPI служит для подключения плат расширения к интерфейсу SPI. Согласно стандарту Arduino R3 разъём SPI должен быть вынесен в виде соединителя 3×2 вывода в определённом месте. Интерфейс SPI не предназначен для внутрисхемного программирования микроконтроллера STM32F405RG и может использоваться только для связи с другими SPI-устройствами.

Разъём SWD предназначен для внутрисхемного программирования микроконтроллера STM32F405RG через программатор, например ST-Link.

Вывод BOOT0 определяет режим загрузки микроконтроллера отладочной платы. Если замкнуть эту пару выводов, микроконтроллер загрузится в режиме DFU. Это позволит записать программу в микроконтроллер без программатора через порт USB.

СРЕДА РАЗРАБОТКИ

Для программирования отладочной платы Iskra JS разработана специальная среда разработки Espruino Web IDE, которая позволяет создавать програм-

мы, загружать их в микроконтроллер платы, отображать результаты работы на консоль и отлаживать программы. Данное приложение работает на платформе Google Chrome [4] и доступно для операционных систем Windows, Mac OS, Linux и Chrome OS.

Для программирования платы Iskra JS используется язык программирования JavaScript, особенности использования которого подробно описаны в [5]; также доступны исходные файлы JavaScript ядра Espruino [6]. Язык программирования JavaScript является популярным и мощным средством разработки. Простой, выразительный и гибкий, он обычно используется для создания интернет-приложений, но этим сфера его применения не ограничивается.


Для установки среды программирования на компьютер потребуется выполнить несколько простых шагов. Сначала необходимо установить на компьютер интернет-браузер Google Chrome и открыть с его помощью ссылку [7] для загрузки среды. Далее необходимо применить дополнительные настройки IDE, кликнув мышкой по программной кнопке «Установить» (см. рис. 5). Можно установить среду разработки, используя другой браузер, но при этом придётся настроить систему вручную [8]. Выполнение настройки необходимо для корректной работы библиотек и плат «Амперки».


Для операционной системы Windows потребуется также загрузить файл драйвера [9] и установить его. После установки драйвера необходимо открыть каталог установки (например, C:\ProgramFiles(x86)\STMicroelectronics\Software\Virtual comport driver) и запустить файл *dpinst_x86.exe* или *dpinst_amd64.exe* для 32-разрядной или 64-разрядной версии операционной системы соответственно. Для опе-

рационных систем Mac OS и Linux драйвер не требуется.


После выполнения установки среда программирования будет автоматически запущена. Если понадобится запустить среду вновь, можно использовать её ярлык, который доступен в Google Chrome во вкладке «Сервисы». Для того чтобы создать ярлык на рабочем столе или в панели быстрого запуска, необходимо ввести в адресной строке браузера *chrome://apps/* и кликнуть правой кнопкой мыши по иконке Espruino IDE.

Окно среды разработки состоит из двух основных частей: слева – окно консоли, справа – окно исходного кода, как показано на рисунке 6. Порядок работы в установленной среде программирования следующий:

1. Подключить отладочную плату к компьютеру через USB-порт.
2. Ввести текст программы на правой панели IDE.
3. Загрузить программу в плату кнопкой .
4. Наблюдать за работой устройства и выводом отладочной информации в консоли.

Подключение платы к компьютеру осуществляется кабелем USB и нажатием программной кнопки  в верхнем левом углу окна IDE. После подключения отобразится список доступных портов и плат, например:

- COMx на Windows;
- /dev/tty.usbmodemXXX на Mac OS;
- /dev/ttyACMx на Linux;

После успешного подключения на консоли появится надпись *Connected*. Цвет фона кнопки  при этом изменится с оранжевого на зелёный, и в нижнем правом углу экрана появится окно с уведомлением о том, что плата успешно подключена к компьютеру. Теперь плата готова к выполнению команд и программ пользователя.

Например, для перезагрузки отладочной платы нужно ввести в поле консоли команду `reset()`; После перезагрузки в консоли отобразится информация о плате с указанием версии её прошивки. При необходимости можно обновить прошивку до последней версии (см. далее).

Программа может отображаться и редактироваться как в текстовом виде, так и в виде программных блоков. Переключение между этими режимами осуществляется с помощью программной кнопки `</>`.

В консоли можно работать с подключённой платой в диалоговом режиме: делать арифметические вычисления, вызывать функции, просматривать значения переменных и т.д. Например, если ввести в консоль `3 + (8-7) * 3/6` и нажать `Enter`, то отладочная плата выполнит команду и вернёт результат `3.5`. В консольном режиме можно использовать переменные:

```
>var x = 3
=3
>var y = 4
=4
>x * x + y * y + 5
=30
```

Переменные могут хранить не только числа, но и строки, логические значения, функции и составные объекты. Интерпретатор JavaScript позволяет работать с пользовательскими и встроенными функциями. Например, вызов функции `>getTime()` вернёт числовой результат времени в секундах, прошедших с момента включения платы.

В диалоговом режиме допускается управление элементами платы и опрос их состояния. Например, с помощью команды `>LED1.write(1)` подаётся питание на светодиод LED1, а посредством команды `>LED1.write(0)` питание светодиода LED1 отключается. Опрос состояния кнопки можно выполнить с помощью команды `>BTN1.read()`. Если кнопка нажата, в ответ отобразится `=true`, в противном случае выведется `=false`.

ПРИМЕРЫ ПРОГРАММ

Рассмотрим пример программы, результатом работы которой будет мигание светодиода LED1 на плате (см. листинг 1).

Данную программу можно сохранить на компьютере в виде файла с произвольным именем и расширением `js`, например `blink.js`. После загрузки программы в отладочную плату Iskra JS на плате начнёт мигать светодиод LED1 с периодичностью 500 мс.

Библиотечные функции JavaScript для платы Iskra JS

Вызов	Результат
Случайные числа	
Инициализация: <code>var random = require('@amperka/hw-random')</code>	
<code>random.int(1, 6)</code>	целое случайное число от 1 до 6
Встроенные функции	
Инициализация: не требуется	
<code>console.log('Hello!')</code>	Вывести «Hello!»
<code>setInterval(function() { ... }, 350)</code>	Вызывать функцию каждые 350 мс
<code>setTimeout(function() { ... }, 350)</code>	Вызвать функцию через 350 мс
<code>(3.1415926).toFixed(3)</code>	Округлить до 3 знаков после запятой: 3,142
<code>getTime()</code>	Получить время в секундах с момента старта или перезагрузки
Светодиод	
Инициализация: <code>var led = require('@amperka/led').connect(P1)</code>	
<code>led.turnOn()</code>	Включить
<code>led.turnOff()</code>	Выключить
<code>led.toggle()</code>	Если выключен – включить, если включён – выключить
<code>led.blink(0.2, 0.8)</code>	Мигать: 0,2 с включён, 0,8 с выключён
<code>led.blink(0.2)</code>	Мигнуть 1 раз в течение 0,2 с
<code>led.brightness(0.42)</code>	Установить яркость 42%
Кнопка	
Инициализация: <code>var button = require('@amperka/button').connect(P3)</code>	
<code>button.on('press', function() { ... })</code>	Вызвать функцию при нажатии
<code>button.on('release', function() { ... })</code>	Вызвать функцию при отжатии
<code>button.on('click', function() { ... })</code>	Вызвать функцию при коротком клике
<code>button.on('hold', function() { ... })</code>	Вызвать функцию при длительном нажатии
Зуммер	
Инициализация: <code>var buzzer = require('@amperka/buzzer').connect(P5)</code>	
<code>buzzer.turnOn()</code>	Включить
<code>buzzer.turnOff()</code>	Выключить
<code>buzzer.toggle()</code>	Если выключен – включить, если включён – выключить
<code>buzzer.beep(0.2, 0.8)</code>	Прерывисто звучать: 0,2 с звучать, 0,8 с не звучать
<code>buzzer.beep(0.2)</code>	Звучать 1 раз в течение 0,2 с
<code>buzzer.frequency(1234)</code>	Звучать на частоте 1234 Гц
Потенциометр	
Инициализация: <code>var pot = require('@amperka/pot').connect(A0)</code>	
<code>pot.read()</code>	Считать значение от 0,0 до 1,0
Сенсор освещённости	
Инициализация: <code>var sensor = require('@amperka/light-sensor').connect(A2)</code>	
<code>sensor.read('lx')</code>	Считать значение в лк
Сервопривод	
Инициализация: <code>var servo = require('@amperka/servo').connect(P13)</code>	
<code>servo.write(42)</code>	Повернуть в положение 42°
Термометр	
Инициализация: <code>var thermometer = require('@amperka/thermometer').connect(A4)</code>	
<code>thermometer.read('C')</code>	Считать значение в °C
Ультразвуковой дальномер	
Инициализация: <code>var sonic = require('@amperka/ultrasonic').connect({trigPin: P10, echoPin: P11})</code>	
<code>sonic.ping(function(error, distance) { ... }, 'cm')</code>	Измерить значение в см и вызвать функцию
ИК-приёмник	
Инициализация: <code>var ir = require('@amperka/ir-receiver').connect(P7)</code>	
<code>ir.on('receive', function(code, repeat) { ... })</code>	Вызвать функцию при нажатии кнопки на ИК-пульте
Эмуляция клавиатуры	
Инициализация: <code>var kb = require('@amperka/usb-keyboard')</code>	
<code>kb.tap(kb.KEY_SPACE)</code>	Нажать «Пробел»
<code>kb.tap([kb.MODIFY_ALT, kb.KEY_SPACE])</code>	Нажать <code>Alt</code> + «Пробел»
<code>kb.type('Hello World!')</code>	Набрать «Hello World!»

Следующий пример – программа `lamp.js` с использованием библиотек «Амперки» – приведён в листинге 2 и предназначен для управления светодиодом, подключённым к выводу P1 отладочной платы. В первой строке программы с помощью функции `require` создаётся объект `led` для светодиода, подключаемого к выводу P1. Во второй строке программы вызывается метод `blink` объекта `led`, который заставляет светодиод светиться в течение 0,1 с, а затем гаснуть на 0,9 с в циклическом режиме.

На сайте производителя приведено множество других примеров про-

Листинг 1

```
var on = false;
setInterval(function()
{
on = !on;
LED1.write(on);
}, 500);
```

Листинг 2

```
var led = require('@amperka/led').connect(P1);
led.blink(0.1, 0.9);
```

грамм для работы с Iskra JS и периферийными платами, например из набора «Йодо» [10]. Среди доступных примеров можно найти такие проекты, как теле-

граф, автоматический диммер, «умное» освещение, терменвокс, экранный люксметр, html-термометр, ультразвуковая линейка, паркэлектроник, сканер ИК-пультов, пульт киномана, радар и др.

В таблице приведён краткий перечень библиотечных функций JavaScript, поддерживаемых отладочной платой Iskra JS, с примерами их использования.

Производитель периодически выпускает обновления интерпретатора JavaScript отладочной платы, добавляющие новые возможности и исправляющие ошибки. Обновить версию прошивки можно непосредственно в среде программирования IDE. Для это-

го необходимо перейти в настройки Flasher и нажать программную кнопку *Flash Firmware*, после чего выполнить следующий алгоритм действий:

1. Нажать кнопку *BTN1* на плате и одновременно с ней – кнопку *RESET*, затем отпустить обе кнопки. В результате попеременно начнут мигать светодиоды LED1 и BUSY – это означает, что плата успешно перешла в режим загрузчика.
2. Нажать в IDE кнопку *Next*.
3. Выбрать порт с подключённой платой.
4. Выбрать модель платы и нажать *Next*.
5. Дождаться окончания процесса прошивки.

ЛИТЕРАТУРА

1. www.amperka.ru
2. www.arduino.ru
3. www.st.com
4. <https://www.google.ru/chrome/browser/desktop>
5. <http://wiki.amperka.ru/js:start>
6. <https://github.com/amperka/Espruino/tree/iskrajs>
7. <https://chrome.google.com/webstore/category/extensions>
8. <http://wiki.amperka.ru/js:ide:driver-error>
9. http://wiki.amperka.ru/_media/js:iskra_js:stsw-stm32102.zip
10. <http://amperka.ru/product/yodo>



НОВОСТИ МИРА

Новый осциллограф от Rohde & Schwarz

Компания Rohde & Schwarz представляет новое семейство высокопроизводительных осциллографов R&S RTP, расширяя ассортимент в соответствии с девизом «Осциллографы Rohde & Schwarz. Абсолютная уверенность в результатах».

В ходе разработки нового семейства осциллографов компания уделила особое внимание точности измерения, скорости, широкому спектру функций и перспективности применяемых технологий.

Осциллограф R&S RTP сочетает в себе различные приборы лучше, чем любой другой осциллограф данного класса. Эти возможности делают R&S RTP лучшим осциллографом для отладки, в частности, встраиваемых компонентов с быстрыми цифровыми или широкополосными ВЧ-интерфейсами.

Компания Rohde & Schwarz постоянно продвигает инновации на рынке осциллографов, и новые высокопроизводительные модели ещё раз подтверждают её приверженность высокому качеству.

Семейство высокопроизводительных осциллографов R&S RTP сочетает в себе целостность сигнала с высокой скоростью сбора данных. В стандартном режиме сбора данных прибор способен измерять 1 млн осциллограмм/с, делая это более чем в 1000 раз быстрее, чем другие осциллографы данного класса. Эта возможность поможет пользователям быстрее находить редкие возникающие ошибки.

R&S RTP – единственный прибор на рынке, который компенсирует потери при передаче сигнала от источника к осциллографу (функция исключения цепи) в реальном масштабе времени. Преимущество состоит в том, что осциллограф остаётся по-прежнему

быстрым даже при активированной коррекции сигнала, а благодаря уникальной цифровой архитектуре запуска он может с высокой точностью выполнять запуск по скомпенсированным сигналам.

Ещё одна особенность осциллографа R&S RTP, которая сэкономит разработчикам много времени, – использование аппаратного ускорения для ряда функций анализа. Испытания на соответствие маскам, гистограммы и анализ в частотной области могут выполняться с высокой скоростью, обеспечивая быстрое получение результатов с высокой статистической надёжностью.



В дополнение к аналоговым каналам R&S RTP содержит 16 логических каналов, 4 канала измерения напряжения и 4 канала измерения тока.

Прибор также поддерживает анализ протоколов интерфейсов последовательных шин и предоставляет эффективные функции анализа спектра и сигналов. Благодаря широкому спектру инструментов пользователи могут измерять множество сигналов разных типов, коррелированных друг с другом по времени в многокомпонентных цепях, и быстрее находить соответствующие ошибки с помощью одного контрольно-измерительного прибора.

R&S RTP предназначен для самого широкого круга разработчиков электроники, особенно в аэрокосмической и оборон-

ной отрасли, в области автомобилестроения, промышленности и телекоммуникаций. Прибор идеально подходит для отладки на этапе разработки электронных схем с сигналами разного типа, включая высокоскоростные шины (USB, PCI Express, MIPI и т.п.), многоканальные ВЧ-интерфейсы (для радиосвязи или радиолокации), интерфейсы DDR-памяти, сложные модули управления электропитанием и даже простые шины управления и программирования (I²C, SPI и т.п.). Осциллограф может применяться в тех случаях, когда необходимо измерять сигналы высокого разрешения во временной и частотной областях с широким динамическим диапазоном и высокой чувствительностью.

Несмотря на широкий спектр поддерживаемых функций, R&S RTP является самым компактным высокопроизводительным осциллографом на рынке, занимая на 40% меньшую площадь рабочего пространства, чем традиционные модели. Благодаря продуманной концепции охлаждения и бесшумным вентиляторам он никогда не создаст помех в работе лаборатории.

Высокопроизводительное семейство R&S RTP – очень надёжная инвестиция. Если вырастут требования к контрольно-измерительным характеристикам, полоса измерения, глубина памяти и набор функций могут быть легко модернизированы с помощью аппаратных и программных опций.

Rohde & Schwarz планирует дополнительные усовершенствования в семействе R&S RTP. Следующие модели осциллографов R&S RTP с увеличенной полосой пропускания выйдут в 2019 году.

Высокопроизводительные осциллографы R&S RTP уже доступны для заказа в компании Rohde & Schwarz в виде моделей с полосой пропускания 4, 6 или 8 ГГц.

www.rohde-schwarz.com/rtp