



Московский Государственный Технический Университет им. Н.Э.Баумана



Будущее информационных технологий и Интернет вещей

Алексей Юрьевич Попов

*Доцент кафедры «Компьютерные системы и сети»
МГТУ им. Н.Э.Баумана*



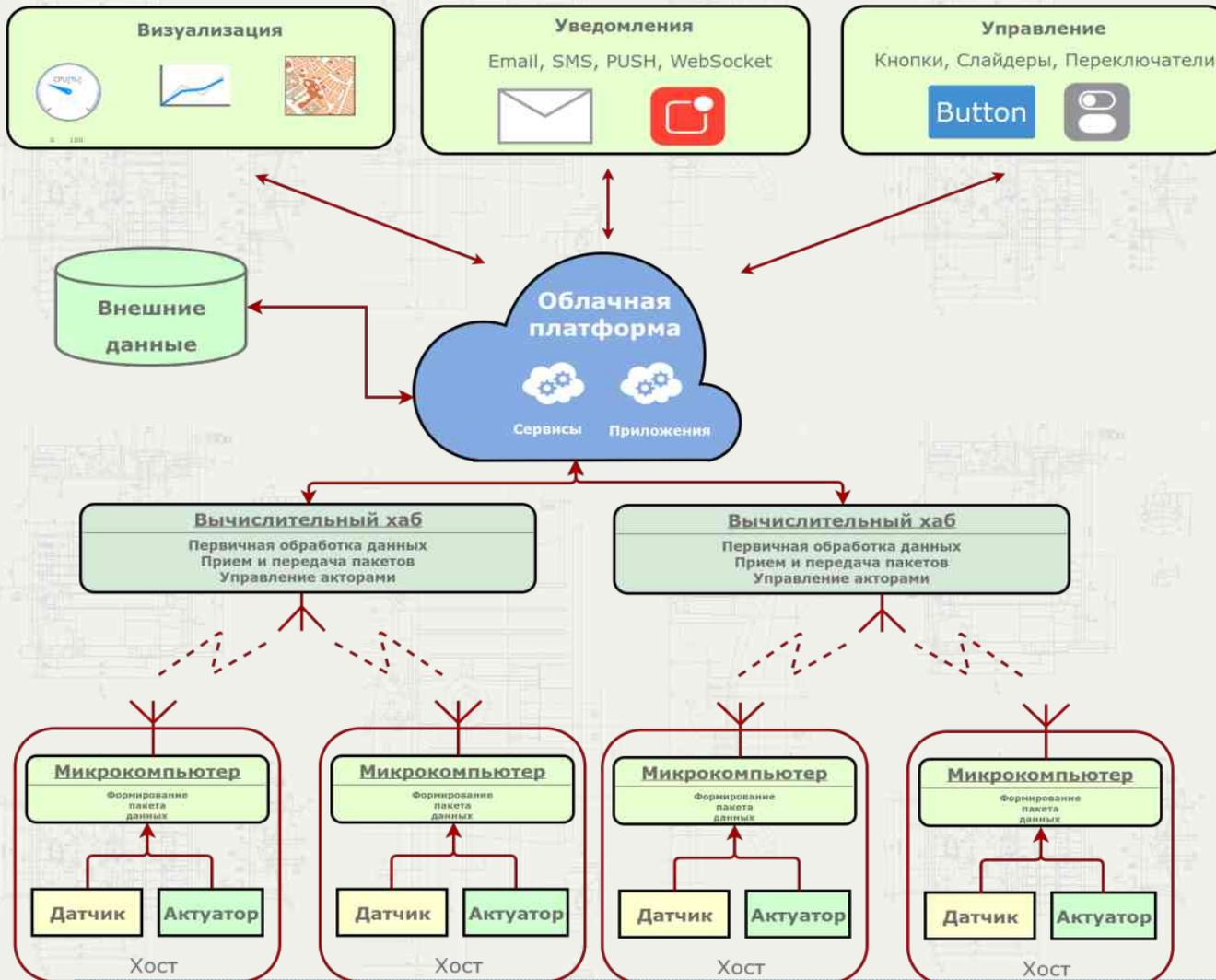
Московский Государственный Технический Университет им. Н.Э.Баумана



Принципы построения кибер- физических систем Интернета вещей

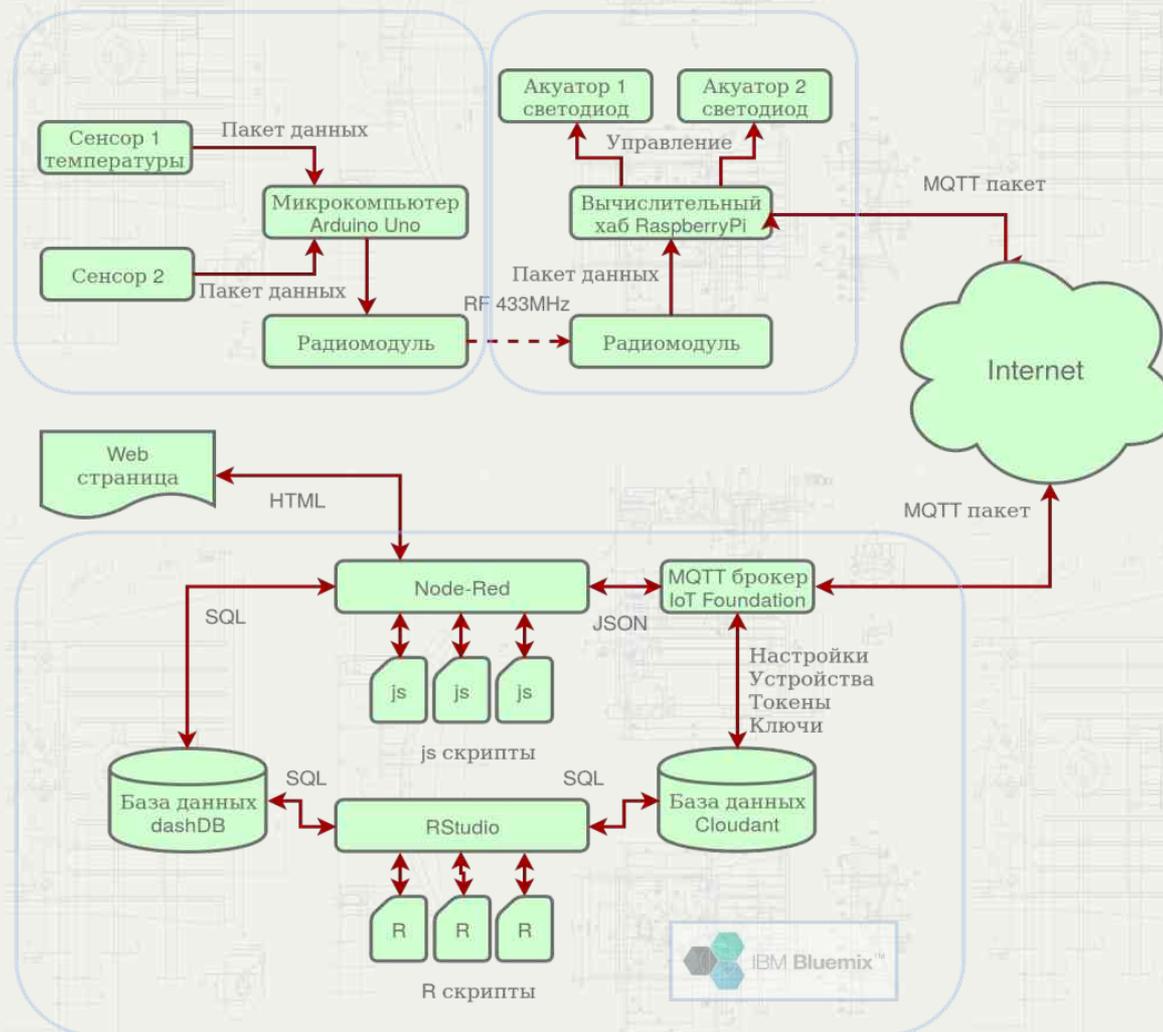
Архитектура типового решения

Типовая инфраструктура решения Интернета вещей



- Функции типового решения:**
- i. Сбор первичных данных.
 - ii. Управление объектами через актуаторы.
 - iii. Передача первичных данных от микрокомпьютеров в вычислительный хаб и в обратном направлении.
 - iv. Первичная обработка данных в вычислительном хабе, формирование пакетов данных для передачи их в облако.
 - v. Получение и хранение данных в облаке.
 - vi. Аналитическая обработка в облаке.
 - vii. Визуализация данных.
 - viii. Прием команд от внешних управляющих консолей.
 - ix. Принятие решений по управлению актуаторами.
 - x. Передача управляющих сообщений в вычислительные хабы.

Типовая инфраструктура решения Интернета вещей





Московский Государственный Технический Университет им. Н.Э.Баумана



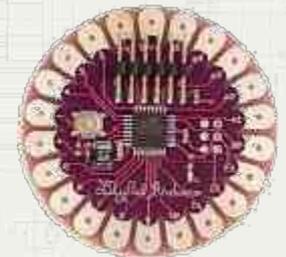
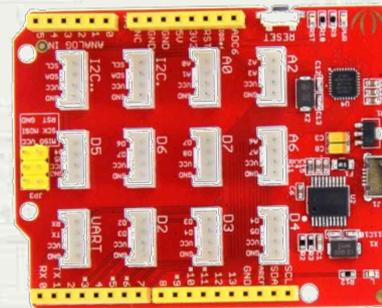
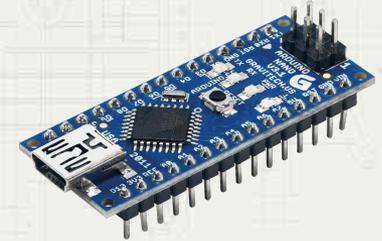
Принципы построения кибер- физических систем Интернета вещей

*Получение телеметрической
информации с помощью
микроконтроллеров*

Аппаратная платформа Arduino



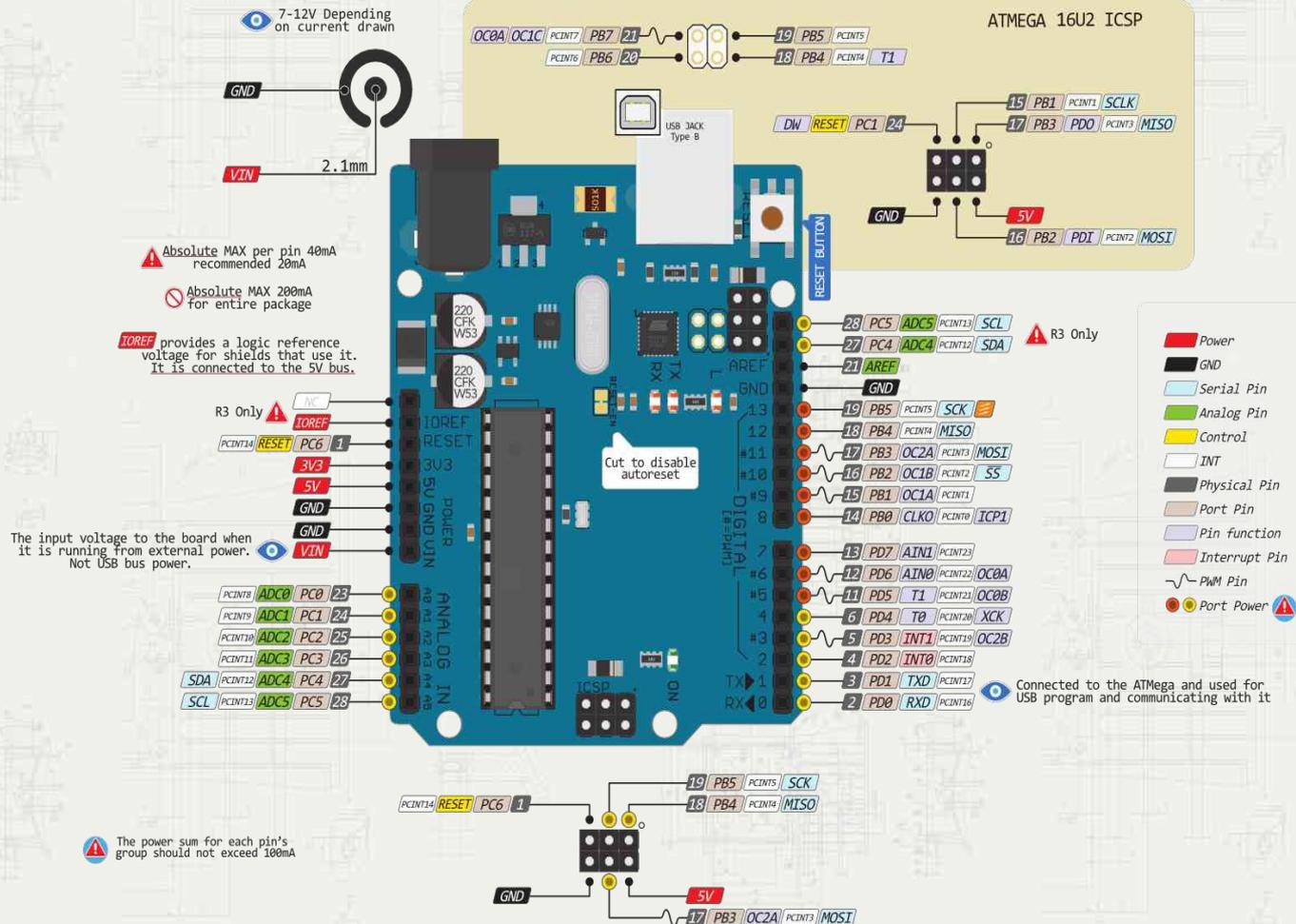
- ✓ Arduino Nano
- ✓ Arduino Micro
- ✓ Arduino Mini
- ✓ Arduino Uno
- ✓ Arduino Leonhard
- ✓ Arduino Mega
- ✓ Arduino LilyPad
- ✓ Seeduino Lotus



Аппаратные ресурсы Arduino



UNO PINOUT



Среда разработки Arduino CC



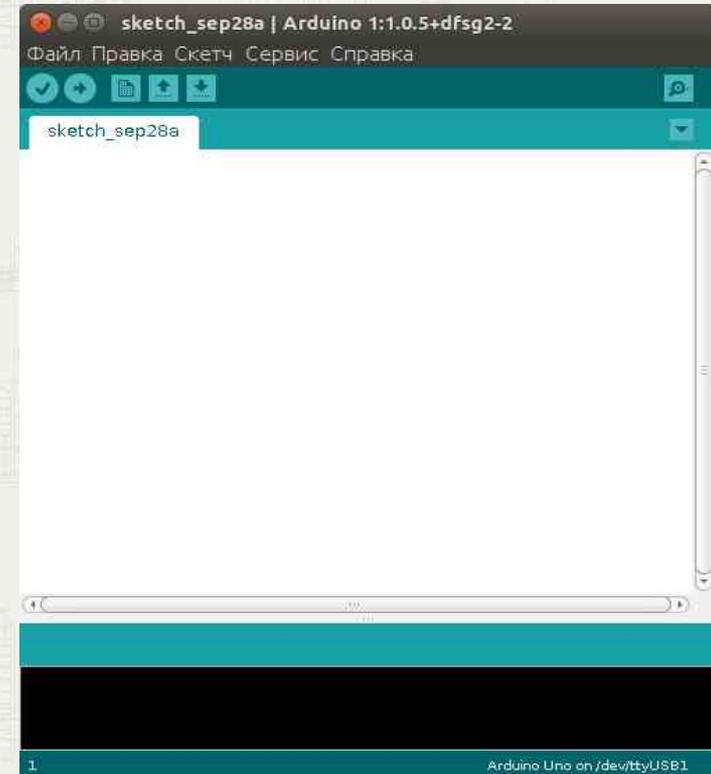
Инструкция по установке: http://arduino.ru/Arduino_environment

Простой пример:

```
int sensorPin = A0; // select the input pin for the potentiometer
int ledPin = 13;    // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the
sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(sensorValue);
}
```





Московский Государственный Технический Университет им. Н.Э.Баумана



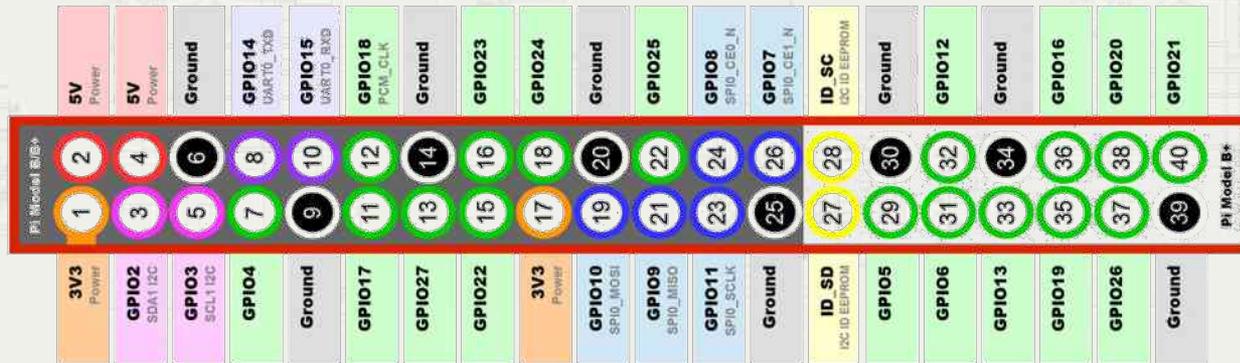
Принципы построения кибер- физических систем Интернета вещей

*Обработка данных в вычислительном
хабе*

Аппаратные ресурсы RaspberryPi



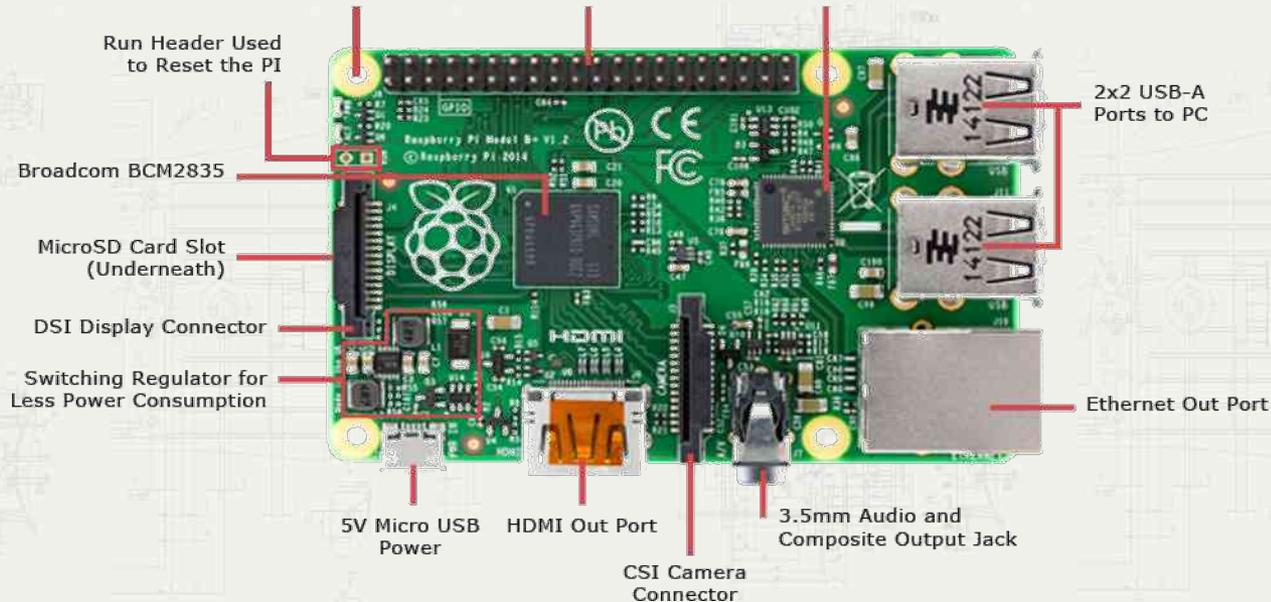
GPIO Pinout Diagram



4 Squarely Placed Mounting Holes

40 GPIO Headers

SMSC LAN9514 USB Ethernet Controller



Операционные системы для RaspberryPi

* - всего применяется более 50 ОС (на базе Linux, Unix, RTOS, Windows)



1) **Raspbian** - Свободная операционная система на основе Debian, оптимизированная для аппаратных возможностей Raspberry Pi.

www.raspberrypi.org/downloads

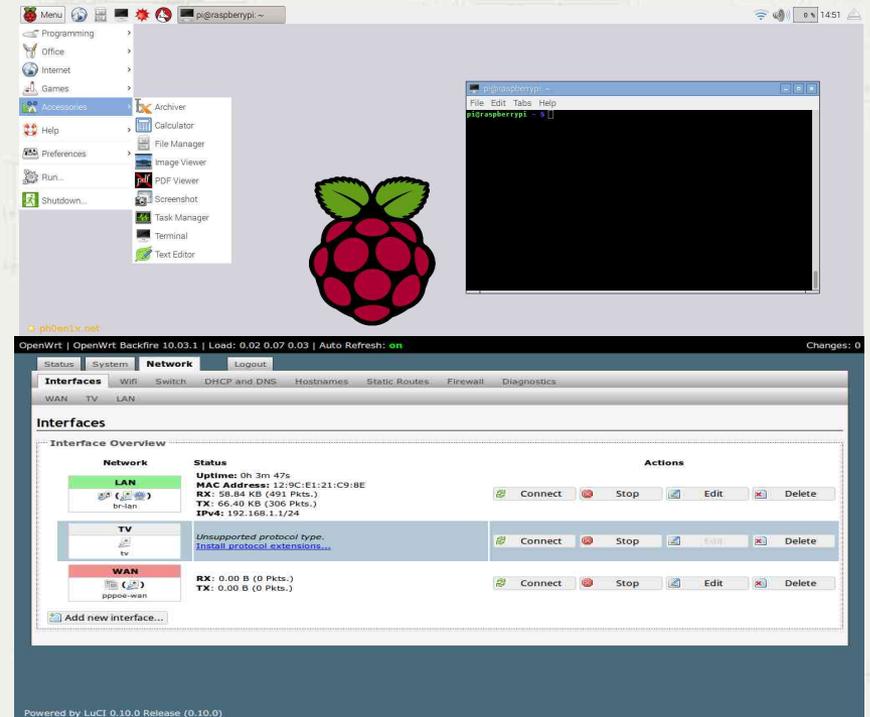
2) **OpenWrt** - встроенная операционная система, основанная на ядре Linux, и предназначенная, в первую очередь, для домашних маршрутизаторов.

wiki.openwrt.org/toh/raspberry_pi

3) **Windows 10 IoT** - особая операционная система Microsoft, оптимизированная для мини-компьютеров с ограниченным набором системных ресурсов

[http://go.microsoft.com/fwlink/?](http://go.microsoft.com/fwlink/?LinkId=616847)

[LinkId=616847](http://go.microsoft.com/fwlink/?LinkId=616847)





Московский Государственный Технический Университет им. Н.Э.Баумана



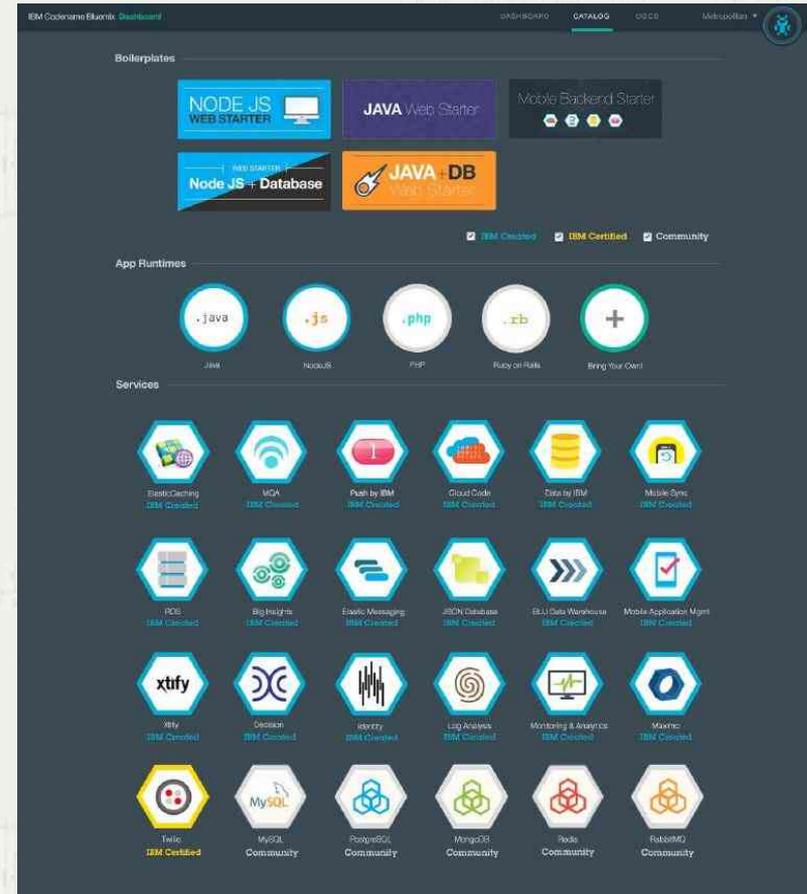
Принципы построения кибер- физических систем Интернета вещей

Облачная платформа IBM Bluemix

Преимущества облачной платформы IBM Bluemix



- i. Простота использования для начинающих разработчиков.
- ii. Наличие большого количества установленных встроенных сервисов и приложений.
- iii. Развитый графический интерфейс и командное управление через CloudFoundry консоль.
- iv. Встроенную веб среду разработки.
- v. Бесплатное предоставление веб-хостинга.
- vi. Для преподавателей и студентов: возможность длительного использования (на 1 год и более).
- vii. Поддержка протоколов Интернета вещей.

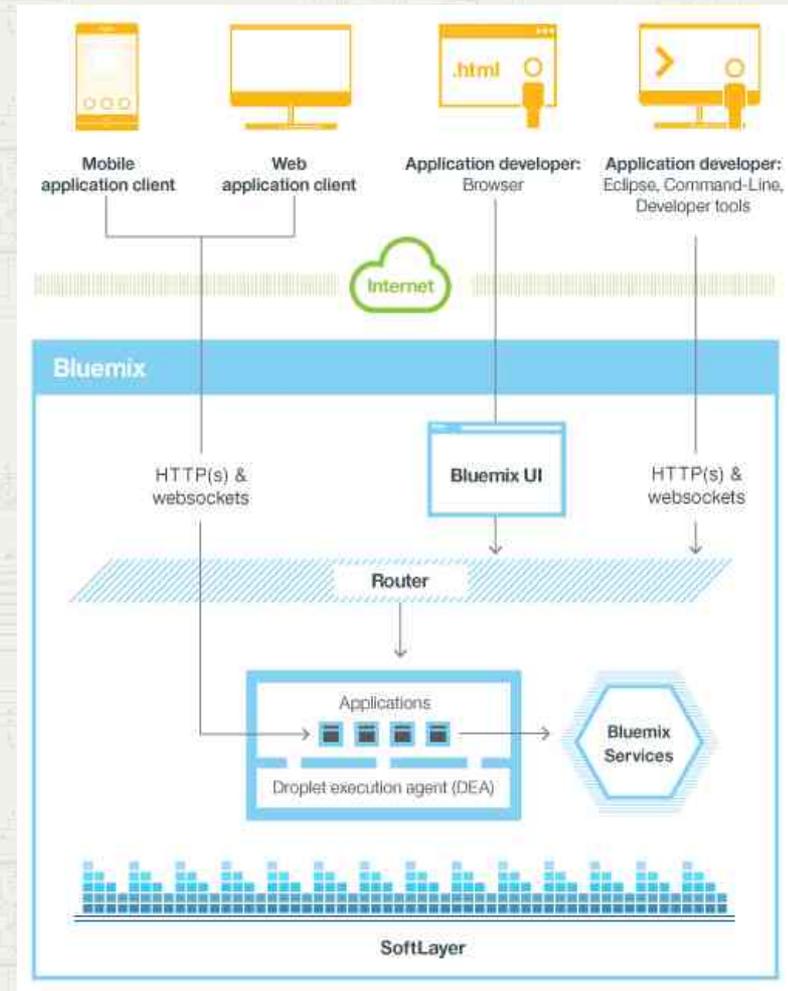


Регистрация: <https://console.bluemix.net>



Концепция IBM Bluemix

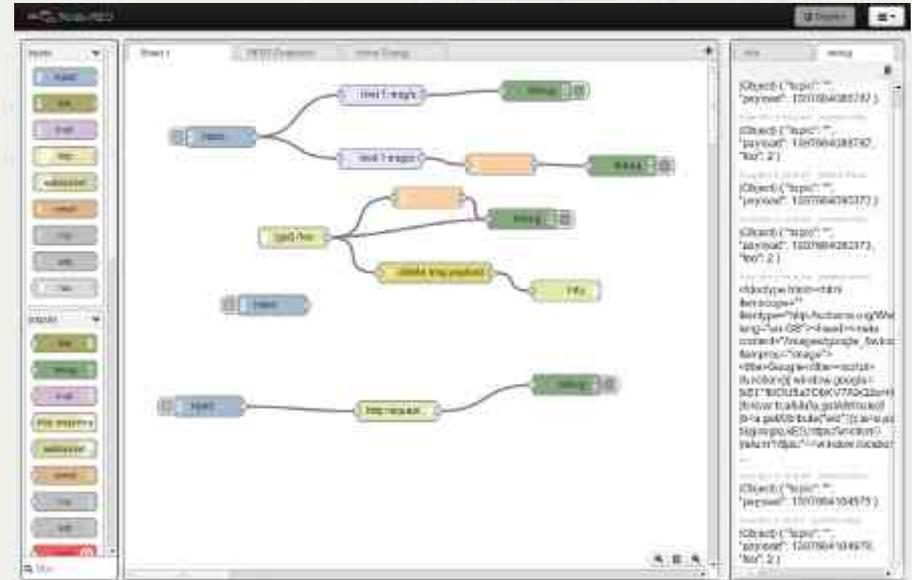
- В терминологии Bluemix приложение (**application**) — это созданный вами артефакт, т. е. весь программный код (исходный код или исполняемые двоичные файлы), который необходимо запустить или на который необходимо сослаться в процессе исполнения.
- Сервис (**service**) — это код, работающий на платформе Bluemix и предлагающий некоторую функциональность, которую могут использовать приложения. Это может быть готовый сервис, используемый непосредственно — например, push-уведомления для мобильных приложений или эластичное кэширование для веб-приложения. Вы также можете создавать собственные сервисы в диапазоне от простых служебных функций до сложной бизнес-логики.



Среда визуальной разработки JavaScript приложений Node-RED



- **Нод (node)** - функционально законченный блок.
- **Поток обработки (flow)** - цепь соединенных нодов и соответствующий им конфигурационный нод.
- **Входной нод (input node)** - нод, принимающий внешние данные и задающий начало потоку обработки (flow). Входной нод имеет один или несколько выходных портов (output ports).
- **Выходной нод (output node)** - нод, завершающий поток обработки и передающий результаты во внешнюю среду.
- **Функциональный нод (function node or query node)** - нод, находящийся внутри потока обработки и имеющий один входной порт и один или несколько выходных портов.
- **Конфигурационный нод (config node)** - нод, содержащий конфигурационную информацию, используемую в других нодах. Конфигурационный нод не связан портами с другими типами нодов.
- **Набор нодов (node set)** - ноды, включенные в связанные js/html файлы (пары соответствующих друг другу файлов, содержащих связанный код JavaScript и HTML). Ошибка в js/html паре приводит к неработоспособности всех нодов набора.
- **Модуль нодов (node module)** - множество связанных js/html файлов (и, соответственно, наборов нодов), где каждая пара описана в package.json файле.
- **Пакет нодов (node pack)** - коллекция связанных по функциональности нодов, которые могут быть использованы разработчиком в рамках одного проекта.



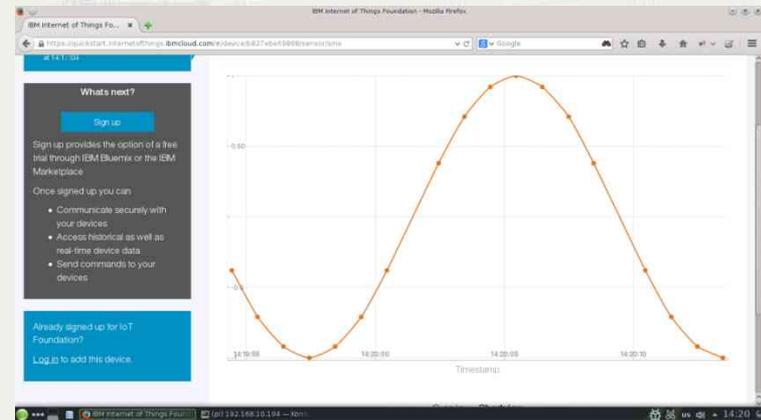
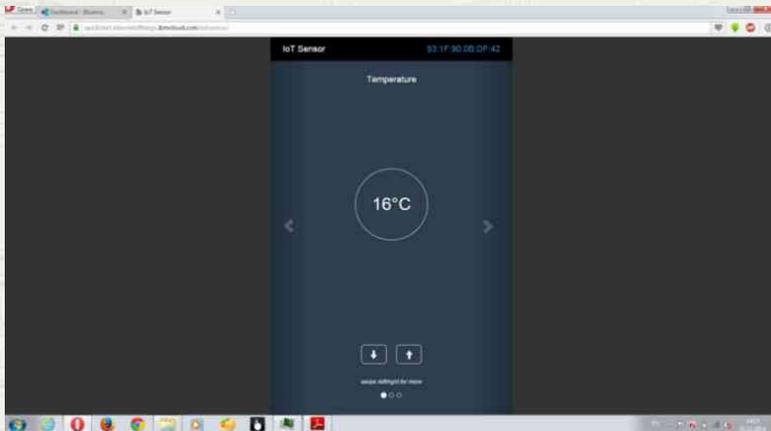
Практикум быстрому прототипированию проектов Интернета вещей



✓ Инсталляция IBM IoT клиента

```
pi@raspberrypi ~$ curl -LO https://github.com/ibm-messaging/iot-raspberrypi/releases/download/1.0.2/iot_1.0-1_armhf.deb
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 397 0 397 0 0 69 0 --:--:-- 0:00:05 --:--:-- 744
100 108k 100 108k 0 0 9306 0 0:00:11 0:00:11 --:--:-- 100k
pi@raspberrypi ~$ dpkg -i iot_1.0-1_armhf.deb
dpkg: error: requested operation requires superuser privilege
pi@raspberrypi ~$ sudo dpkg -i iot_1.0-1_armhf.deb
Selecting previously unselected package iot.
(Reading database ... 73811 files and directories currently installed.)
Unpacking iot (from iot_1.0-1_armhf.deb) ...
Setting up iot (1.0-1) ...
Starting the iot program
pi@raspberrypi ~$ service iot status
[ ok ] iot is running.
pi@raspberrypi ~$
```

✓ Тестовая визуализация данных



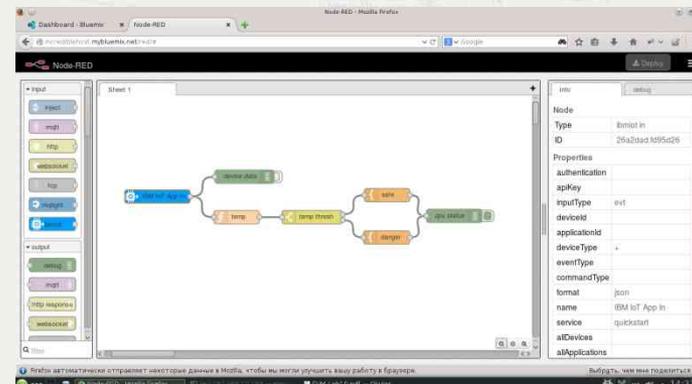
Практикум быстрому прототипированию проектов Интернета вещей



✓ Регистрация устройства в IBM Bluemix IoT



✓ Работа аналитической части в Node-RED





Московский Государственный Технический Университет им. Н.Э.Баумана



Принципы построения кибер- физических систем Интернета вещей

Обсуждение



Спасибо за внимание!