How to hack mini RC-car

Arduino, Go Language, Web Bluetooth
Rocket Hackathon 2016 Powered =)

Maksim Naumov

- Software engineer, work for Rocket Labs
- Open source contributor
- Survived in Asia
- World Citizen, I hate the borders
- Go-kart racer
- Coffee lover

Follow me:
https://github.com/digitalcrab
https://twitter.com/digital_crab

Agenda

- Rocket Hackathon (some pictures)
- Project description: Arduino, Go language, Microelectronics
- Demo #1
- Improved version of the project: Web Bluetooth (BLE)
- Demo #2
- Questions & Answers
- Hangout

Rocket Hackathon 2016

Many thanks to

ROCKETINTERNET !!!

- Amazing prizes
- Well organized
- A lot of food and drinks =)

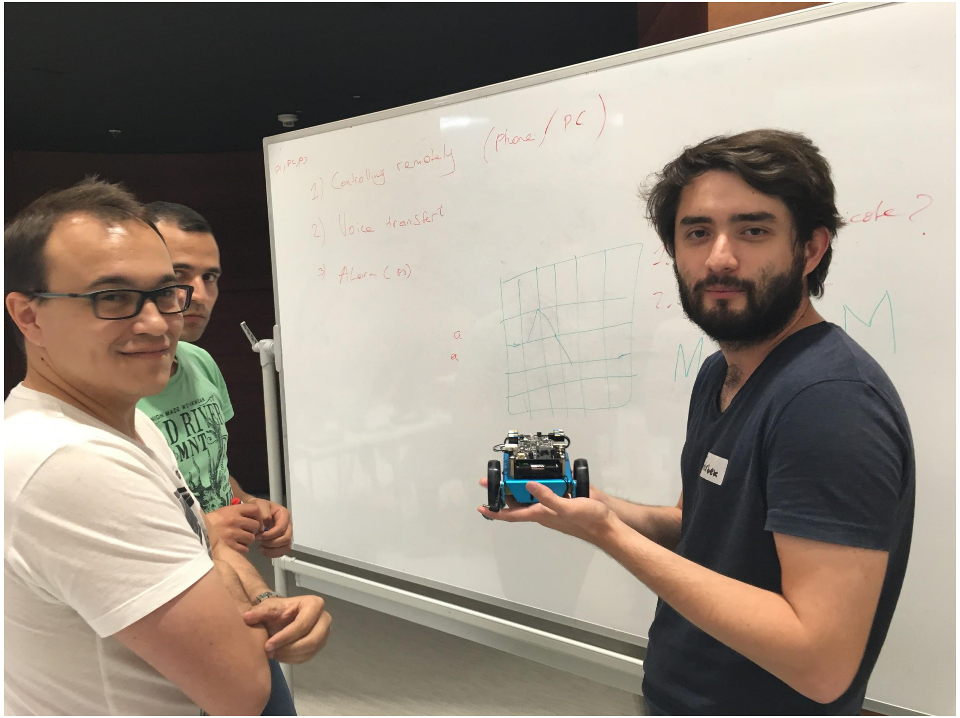
Rocket Hackathon: Road King Team

At the moment of this picture we had no name of the team.

Anyway, we are:

- Maksim Naumov
- Sardorbek Pulatov
- And the Guy in a green T-shirt (dropout)

We had no plan what to do and how. Decided to drink some beer and have some fun with the little blue robot (Sardo is holding it).



Rocket Hackathon: Idea !?

Next morning we came up with the idea!
And the team name =)

- Racing track with baby tiny cars
- Control the cars through the Internet
- Stream the video online

Time for shopping ...

Hackathon will over in 24 hours....

Rocket Hackathon: Working hard ...

The algorithm was almost like this:

```
while (true) {
    Write code;
    Solder wires;
    Test everything;
    Drink Beer and Club Mate;
    Eat pizza and sandwiches;
}
```

And we got ...

Rocket Hackathon: 2nd place =)

- Gain 4 kg
- Drank 5 liters of Club Mate
- Drank 5 liters of Beer
- Drank 5 liters of Coffee
- Eat tons of pizza and other food
- Got eye twitching
- Won Apple Watch and Samsung Gear =)

What is behind the stage?

Mini RC-car for 8€

AliExpress™

Smarter Shopping, Better Living!

DX

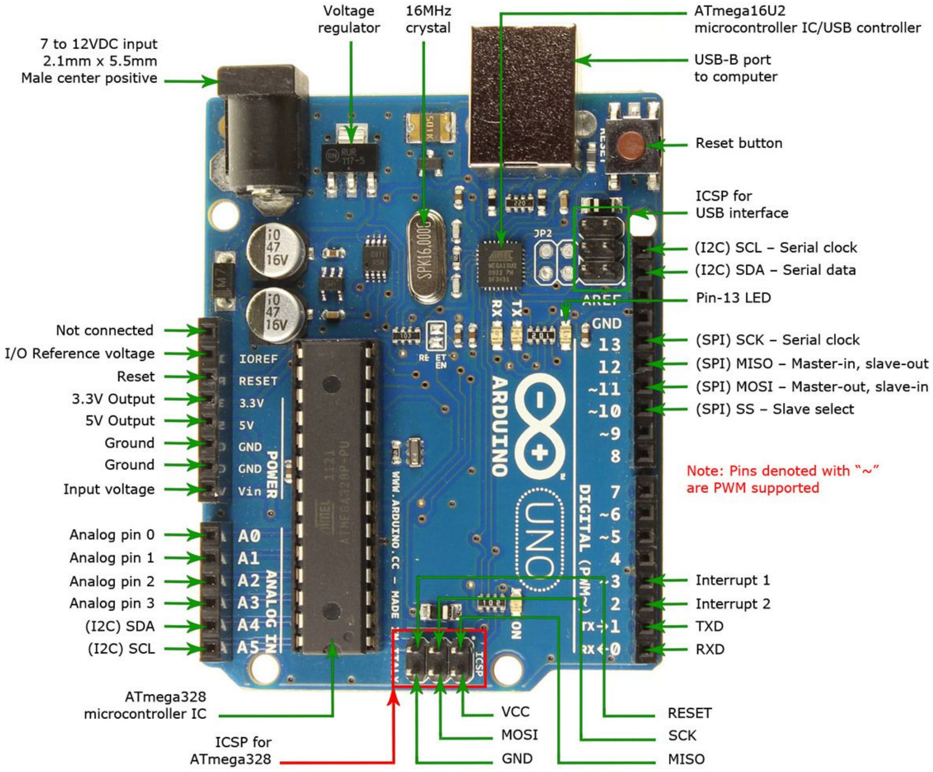
dealextreme

Arduino: microcontroller board

Arduino Uno is a microcontroller board based on the ATmega328P.

- 14 digital input/output pins (PWM included)
- 6 analog inputs
- 16 MHz quartz crystal
- USB connection

The UNO is the best board to get started with electronics and coding. The UNO is the most used and documented board of the whole Arduino family.



Voltage regulator

16MHz crystal

ATmega16U2 microcontroller IC/USB controller

7 to 12VDC input
2.1mm x 5.5mm
Male center positive

USB-B port to computer

Reset button

ICSP for USB interface

(I2C) SCL – Serial clock

(I2C) SDA – Serial data

Pin-13 LED

(SPI) SCK – Serial clock

(SPI) MISO – Master-in, slave-out

(SPI) MOSI – Master-out, slave-in

(SPI) SS – Slave select

Note: Pins denoted with "~" are PWM supported

Not connected

I/O Reference voltage

Reset

3.3V Output

5V Output

Ground

Ground

Input voltage

Analog pin 0

Analog pin 1

Analog pin 2

Analog pin 3

(I2C) SDA

(I2C) SCL

Interrupt 1

Interrupt 2

TXD

RXD

ATmega328 microcontroller IC

ICSP for ATmega328

VCC

MOSI

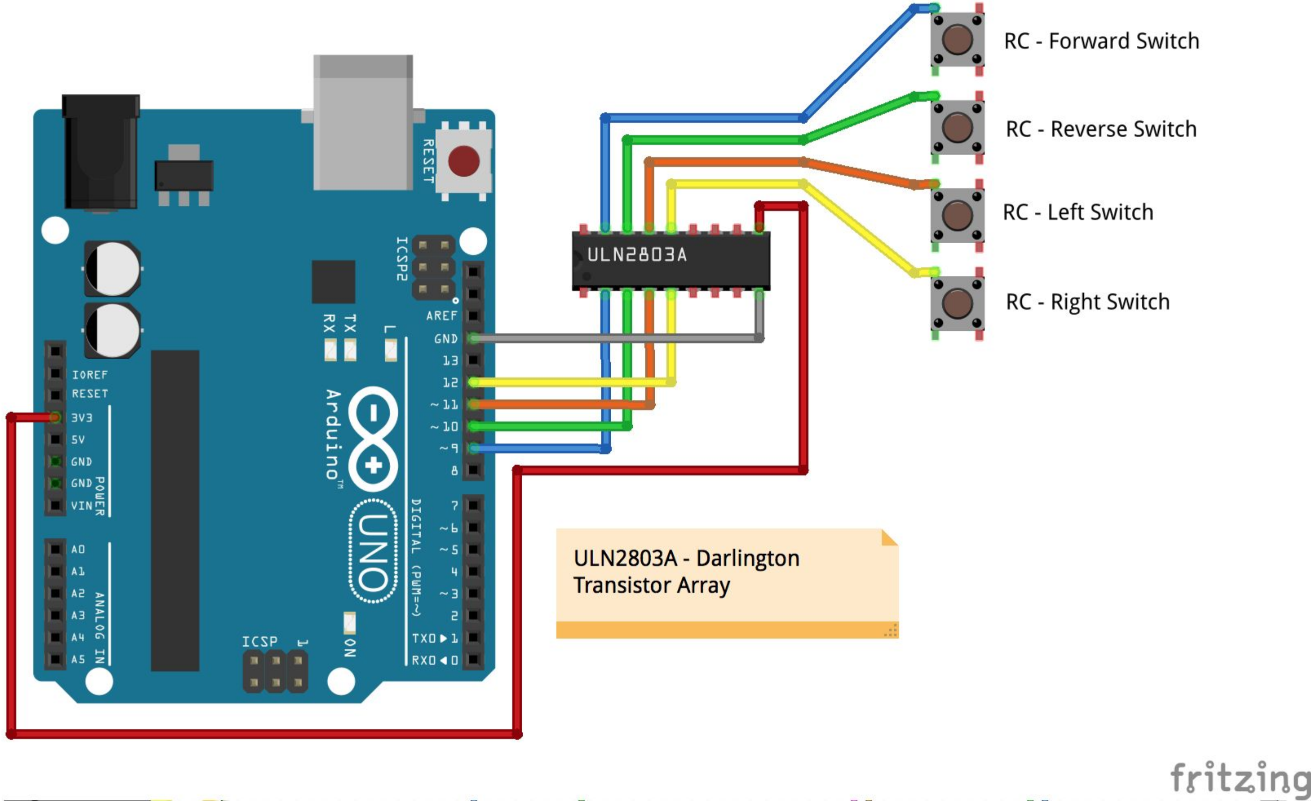
GND

RESET

SCK

MISO

Connection schema: USB version



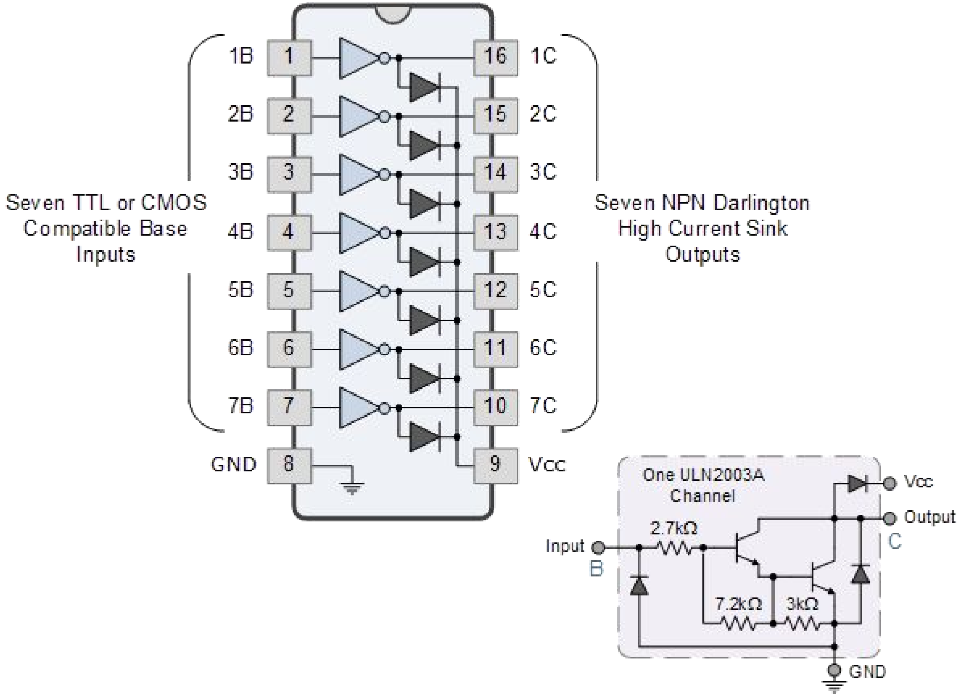
ULN2803A - Darlington Transistor Array

ULN2803A: Darlington Transistor Array

Contains seven darlington transistor pairs each with an input pin on the left and an output pin opposite it on the right as shown.

The Darlington Transistor named after its inventor, Sidney Darlington is a special arrangement of two standard NPN bipolar junction transistors (BJT)

BJTs are current amplifiers. The more current travels through the Base-Emitter path, the more current is permitted to pass through the Collector-Emitter path. The relationship between



ULN2003A on the picture.

Darlington Transistor

Like **water valve**!

Apply small power to the handle - open the big power of the water =)

Easy?

Talk is cheap. Show me the code!

(c) Linus Torvalds

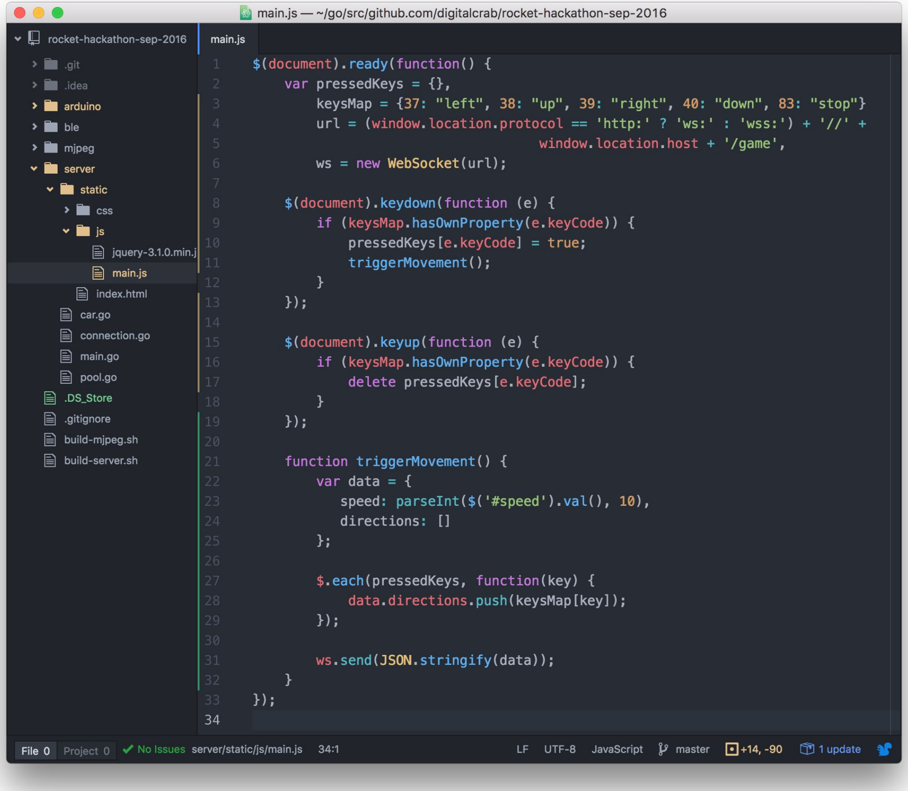
Web part: JavaScript

Flow

- Open WebSocket connection
- Bind keyboard even
- Collect pressed keys
- Collect value of the current speed
- Build JSON
- Send to the server

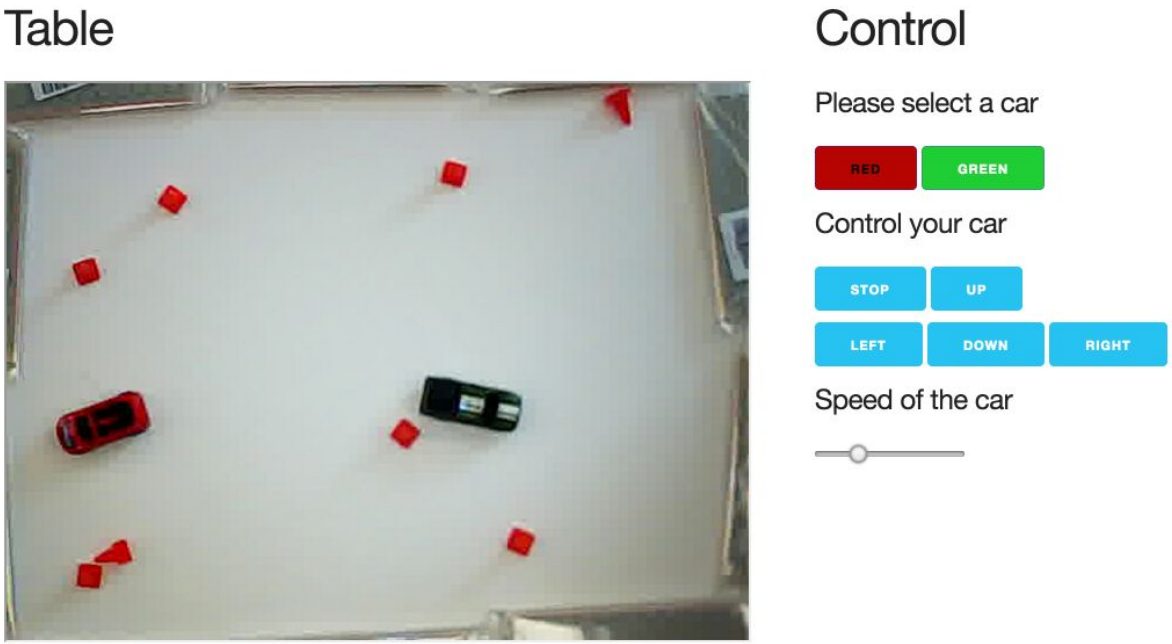
Libraries

- Native WebSockets
- jQuery (actually no sense to use it)



```javascript
$(document).ready(function() {
    var pressedKeys = {},
        keysMap = {37: "left", 38: "up", 39: "right", 40: "down", 83: "stop"}
        url = (window.location.protocol == 'http:' ? 'ws:' : 'wss:') + '//' +
                                    window.location.host + '/game',
        ws = new WebSocket(url);

    $(document).keydown(function (e) {
        if (keysMap.hasOwnProperty(e.keyCode)) {
            pressedKeys[e.keyCode] = true;
            triggerMovement();
        }
    });

    $(document).keyup(function (e) {
        if (keysMap.hasOwnProperty(e.keyCode)) {
            delete pressedKeys[e.keyCode];
        }
    });

    function triggerMovement() {
        var data = {
           speed: parseInt($('#speed').val(), 10),
           directions: []
        };

        $.each(pressedKeys, function(key) {
            data.directions.push(keysMap[key]);
        });

        ws.send(JSON.stringify(data));
    }
});
```

Web part: UI



Table

Control

Please select a car

Control your car

Speed of the car

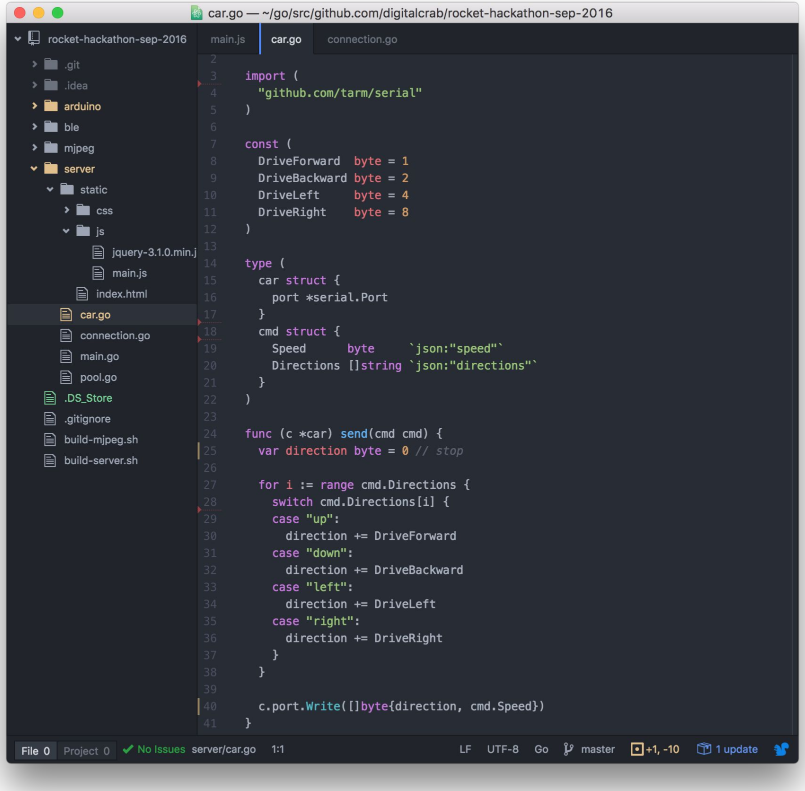
Server part: GoLang

Flow

- Start WebSocket (WS) server
- Open serial port (USB in our case)
- Handle WS connection
- Decode message
- Convert text representation or direction to the bytes
- Send data into the serial port

Libraries

- github.com/gorilla/websocket
- github.com/tarm/serial



```go

import (
  "github.com/tarm/serial"
)

const (
  DriveForward  byte = 1
  DriveBackward byte = 2
  DriveLeft     byte = 4
  DriveRight    byte = 8
)

type (
  car struct {
    port *serial.Port
  }
  cmd struct {
    Speed      byte     `json:"speed"`
    Directions []string `json:"directions"`
  }
)

func (c *car) send(cmd cmd) {
  var direction byte = 0 // stop

  for i := range cmd.Directions {
    switch cmd.Directions[i] {
    case "up":
      direction += DriveForward
    case "down":
      direction += DriveBackward
    case "left":
      direction += DriveLeft
    case "right":
      direction += DriveRight
    }
  }

  c.port.Write([]byte{direction, cmd.Speed})
}
```

Board part: Arduino

Setup

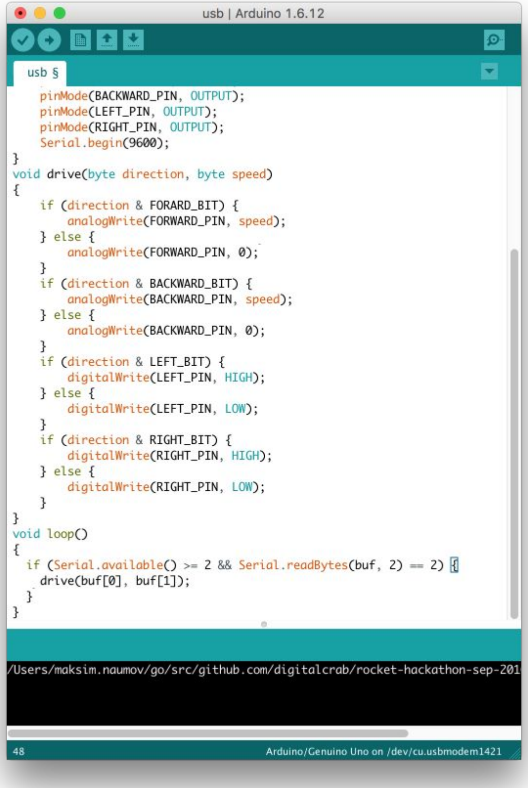
- Set pins to the output mode
- Set the data rate for serial data transmission

Loop

- Get the number of bytes available
- Read 2 bytes (direction and speed)

Drive

- Write value to the pins



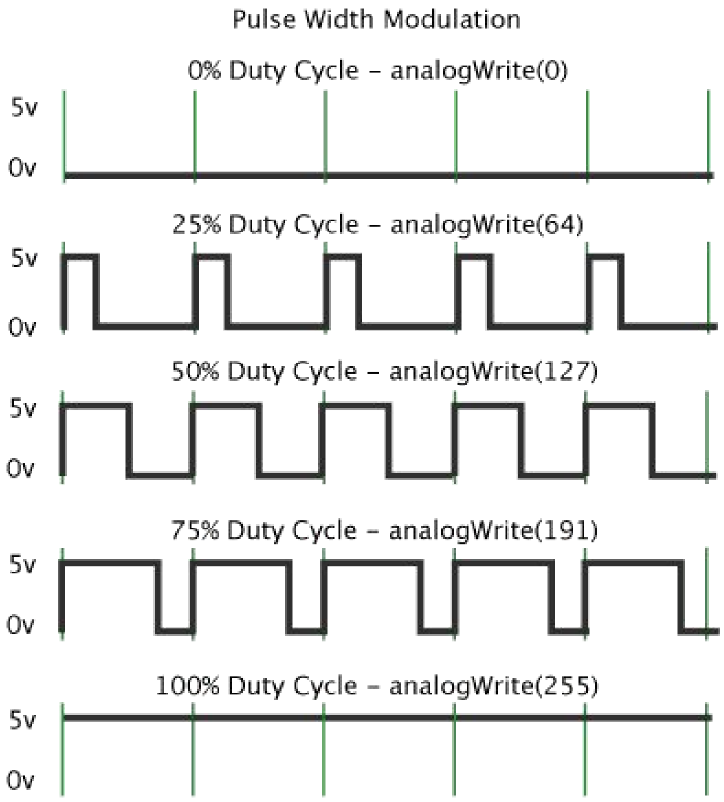
```
    pinMode(BACKWARD_PIN, OUTPUT);
    pinMode(LEFT_PIN, OUTPUT);
    pinMode(RIGHT_PIN, OUTPUT);
    Serial.begin(9600);
}
void drive(byte direction, byte speed)
{
    if (direction & FORARD_BIT) {
        analogWrite(FORWARD_PIN, speed);
    } else {
        analogWrite(FORWARD_PIN, 0);
    }
    if (direction & BACKWARD_BIT) {
        analogWrite(BACKWARD_PIN, speed);
    } else {
        analogWrite(BACKWARD_PIN, 0);
    }
    if (direction & LEFT_BIT) {
        digitalWrite(LEFT_PIN, HIGH);
    } else {
        digitalWrite(LEFT_PIN, LOW);
    }
    if (direction & RIGHT_BIT) {
        digitalWrite(RIGHT_PIN, HIGH);
    } else {
        digitalWrite(RIGHT_PIN, LOW);
    }
}
void loop()
{
  if (Serial.available() >= 2 && Serial.readBytes(buf, 2) == 2) {
    drive(buf[0], buf[1]);
  }
}
```

```
/Users/maksim.naumov/go/src/github.com/digitalcrab/rocket-hackathon-sep-201
```

Arduino: digitalWrite vs analogWrite

digitalWrite - Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode, its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.

analogWrite - Writes an analog value (PWM wave) to a pin. Can be used to drive a motor at various speeds. After a call to analogWrite, the pin will generate a steady square wave of the specified duty cycle until the next call to analogWrite.


Pulse Width Modulation
0% Duty Cycle – analogWrite(0)
5v
0v
25% Duty Cycle – analogWrite(64)
5v
0v
50% Duty Cycle – analogWrite(127)
5v
0v
75% Duty Cycle – analogWrite(191)
5v
0v
100% Duty Cycle – analogWrite(255)
5v
0v

Demo time! =)

Cast: USB, Arduino, Go Server, RC-control, Car

How to improve it?

Unnecessary things:

- USB connection between Arduino and computer
- Go server. Handle WebSockets. Do we need it?
- RC-control. This ugly board with tons of wires...

Things could be better:

- Arduino is too big
- RC-control bounded to the specific frequency (27 Mhz, 40 Mhz, etc...)

Bluetooth Low Energy (BLE)

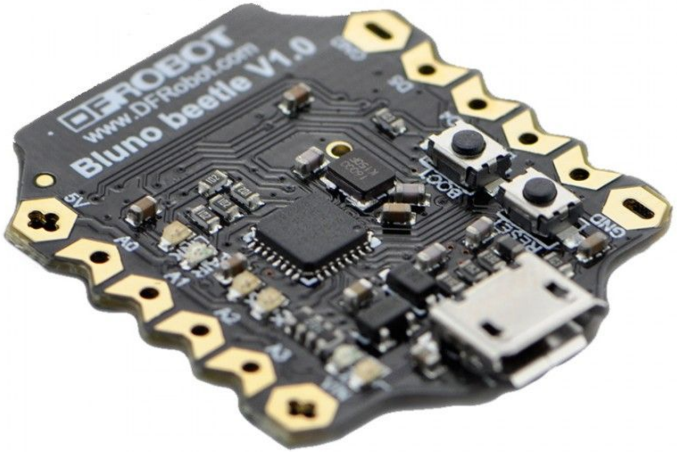
Bluetooth low energy (BLE) is the power- version of Bluetooth that was built for the Internet of Things (IoT).

The power-efficiency of Bluetooth with low energy functionality makes it perfect for devices that run for long periods on power sources, such as coin cell batteries or energy-harvesting devices.

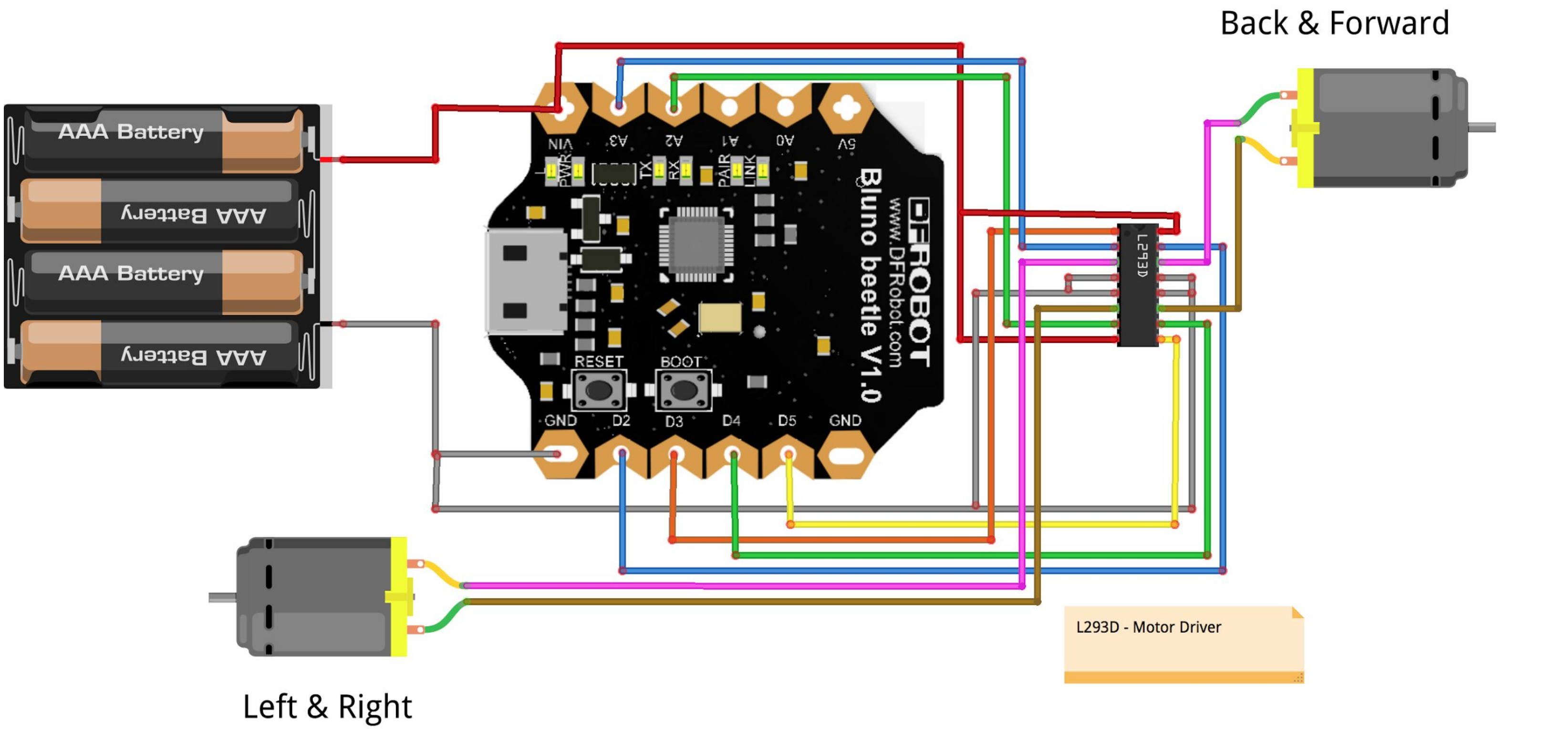
Bluno beetle: Tiny Arduino + Bluetooth

Bluno Beetle is an Arduino Uno based board with bluetooth 4.0 (BLE). It is probably the smallest Arduino BLE board in the market.

- ATmega328@16MHz
- Bluetooth Low Energy (BT 4.0)
- Micro USB port
- Super Compact Size
- Support Bluetooth HID and ibeacon
- Support Wireless Programming



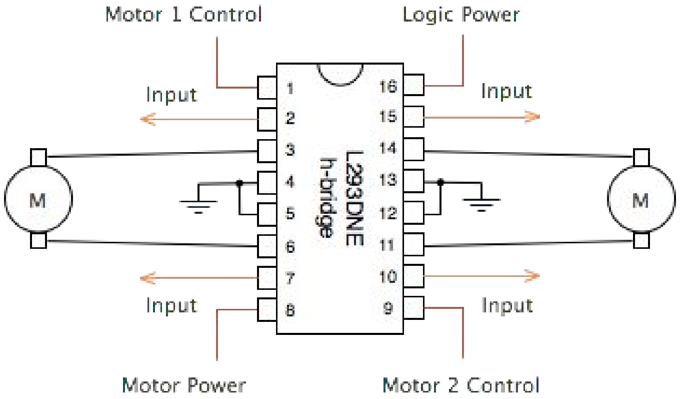
Connection schema: BLE version

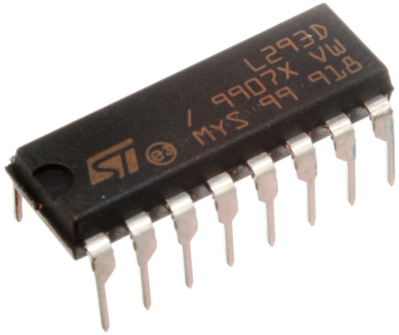


L293D: H-bridge motor driver

- Pin 16 - Logic power
- Pin 8 - Motors power
- Pins 1 and 9 - Motors control
- Pins 2 and 7 - Input logic (motor 1)
- Pins 15 and 10 - Input logic (motor 2)

	Input 1	Input 2
Break	0	0
Break	1	1
Forward	1	0
Reverse	0	1





Web part: JavaScript

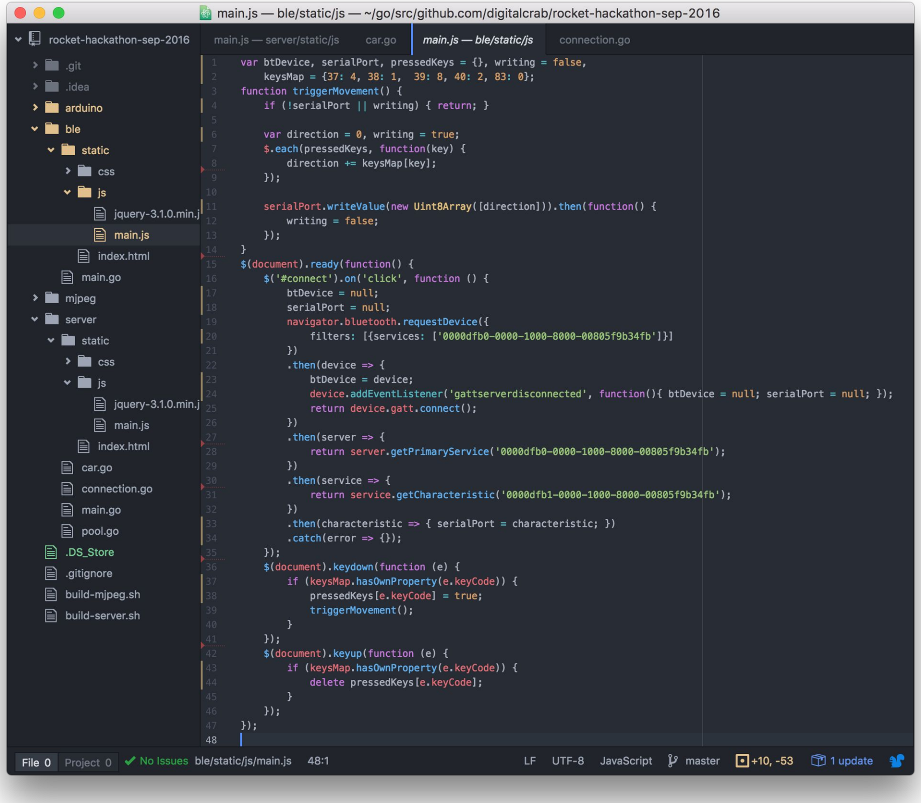
The Web Bluetooth API relies heavily on JavaScript **Promises**.

As a security feature, discovering Bluetooth devices with browser **must** be triggered by a user gesture such as a touch or a mouse click.

Flow: scan for devices, connect to the device, read a bluetooth characteristic, write the value to the characteristic.

Magic strings:

- DF Robot Service
  0000dfb0-0000-1000-8000-00805f9b34fb
- Serial Port
  0000dfb1-0000-1000-8000-00805f9b34fb



```javascript
var btDevice, serialPort, pressedKeys = {}, writing = false,
    keysMap = {37: 4, 38: 1,  39: 8, 40: 2, 83: 0};
function triggerMovement() {
    if (!serialPort || writing) { return; }

    var direction = 0, writing = true;
    $.each(pressedKeys, function(key) {
        direction += keysMap[key];
    });

    serialPort.writeValue(new Uint8Array([direction])).then(function() {
        writing = false;
    });
}
$(document).ready(function() {
    $('#connect').on('click', function () {
        btDevice = null;
        serialPort = null;
        navigator.bluetooth.requestDevice({
            filters: [{services: ['0000dfb0-0000-1000-8000-00805f9b34fb']}]
        })
        .then(device => {
            btDevice = device;
            device.addEventListener('gattserverdisconnected', function(){ btDevice = null; serialPort = null; });
            return device.gatt.connect();
        })
        .then(server => {
            return server.getPrimaryService('0000dfb0-0000-1000-8000-00805f9b34fb');
        })
        .then(service => {
            return service.getCharacteristic('0000dfb1-0000-1000-8000-00805f9b34fb');
        })
        .then(characteristic => { serialPort = characteristic; })
        .catch(error => {});
    });
    $(document).keydown(function (e) {
        if (keysMap.hasOwnProperty(e.keyCode)) {
            pressedKeys[e.keyCode] = true;
            triggerMovement();
        }
    });
    $(document).keyup(function (e) {
        if (keysMap.hasOwnProperty(e.keyCode)) {
            delete pressedKeys[e.keyCode];
        }
    });
});
```

Board part: Arduino

Setup

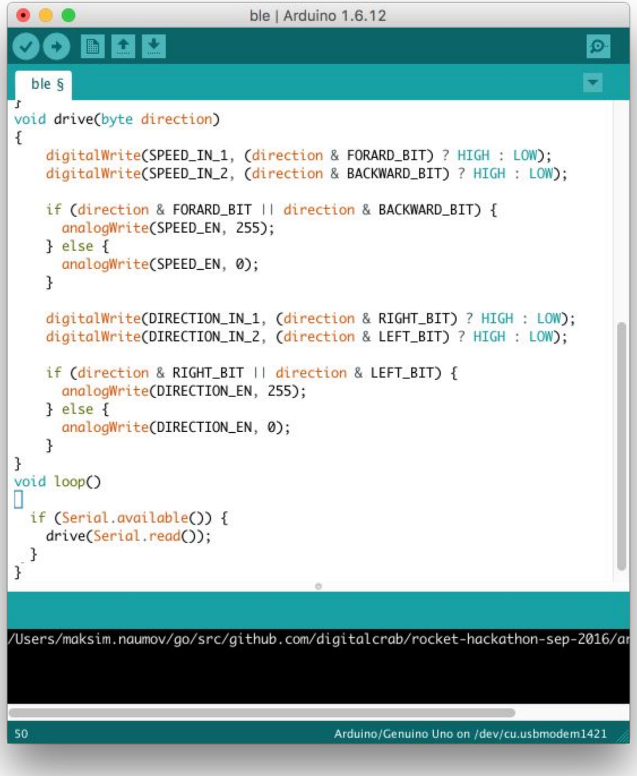
- Set pins to the output mode
- Set the data rate for serial data transmission (115200)

Loop

- Get the number of bytes available
- Read byte (direction only)

Drive

- Write value to the pins



```
}
void drive(byte direction)
{
    digitalWrite(SPEED_IN_1, (direction & FORARD_BIT) ? HIGH : LOW);
    digitalWrite(SPEED_IN_2, (direction & BACKWARD_BIT) ? HIGH : LOW);

    if (direction & FORARD_BIT || direction & BACKWARD_BIT) {
      analogWrite(SPEED_EN, 255);
    } else {
      analogWrite(SPEED_EN, 0);
    }

    digitalWrite(DIRECTION_IN_1, (direction & RIGHT_BIT) ? HIGH : LOW);
    digitalWrite(DIRECTION_IN_2, (direction & LEFT_BIT) ? HIGH : LOW);

    if (direction & RIGHT_BIT || direction & LEFT_BIT) {
      analogWrite(DIRECTION_EN, 255);
    } else {
      analogWrite(DIRECTION_EN, 0);
    }
}
void loop()
{
  if (Serial.available()) {
    drive(Serial.read());
  }
}
```

```
/Users/maksim.naumov/go/src/github.com/digitalcrab/rocket-hackathon-sep-2016/ar
```

Demo time! =)

Cast: Browser, Web Bluetooth, Bluno, Car

Future? RC-cars football

Source code

https://github.com/digitalcrab/rocket-hackathon-sep-2016

Questions?

Thank you!