



Building a robot army with **Angular** and **Nativescript**

@sebawita

Sebastian Witalec

Jack of all Angular
Master of Robots

@sebawita



Squirrels are cute



But are they?



Modern squirrel warfare



Stormtroopers came
to the rescue



Trying to subdue
the rebellion



... and failed miserably



Call for Help



Robots answered the call



But they spoke a different language

WILLKOMMEN
स्वागत
Tervetuloa 欢迎 BIENVENIDA
WELCOME
BIENVENUE 0x48, 0x69
добро пожаловать
ترحيب BEM-VINDO



But they spoke a different language

WILLKOMMEN
स्वागत
Tervetuloa 欢迎 BIENVENIDA

WELCOME

BIENVENUE 0x48, 0x69 ←

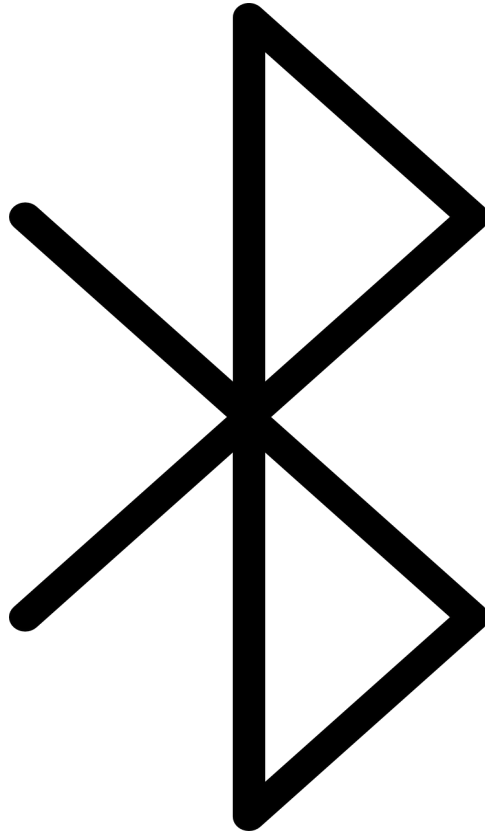
добро пожаловать

ترحيب BEM-VINDO

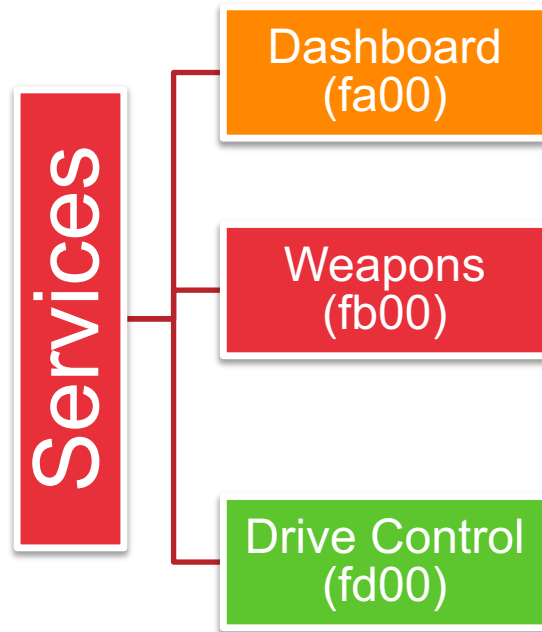
“Hi”
in ASCII



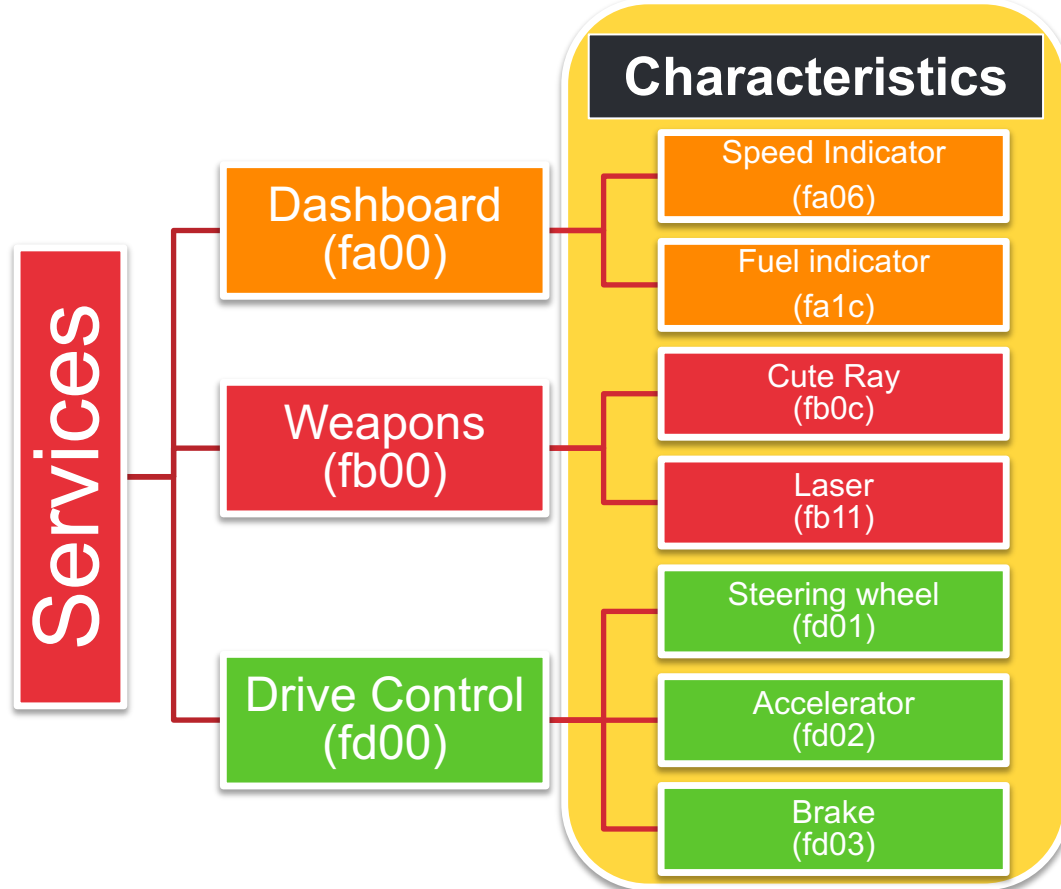
Bluetooth



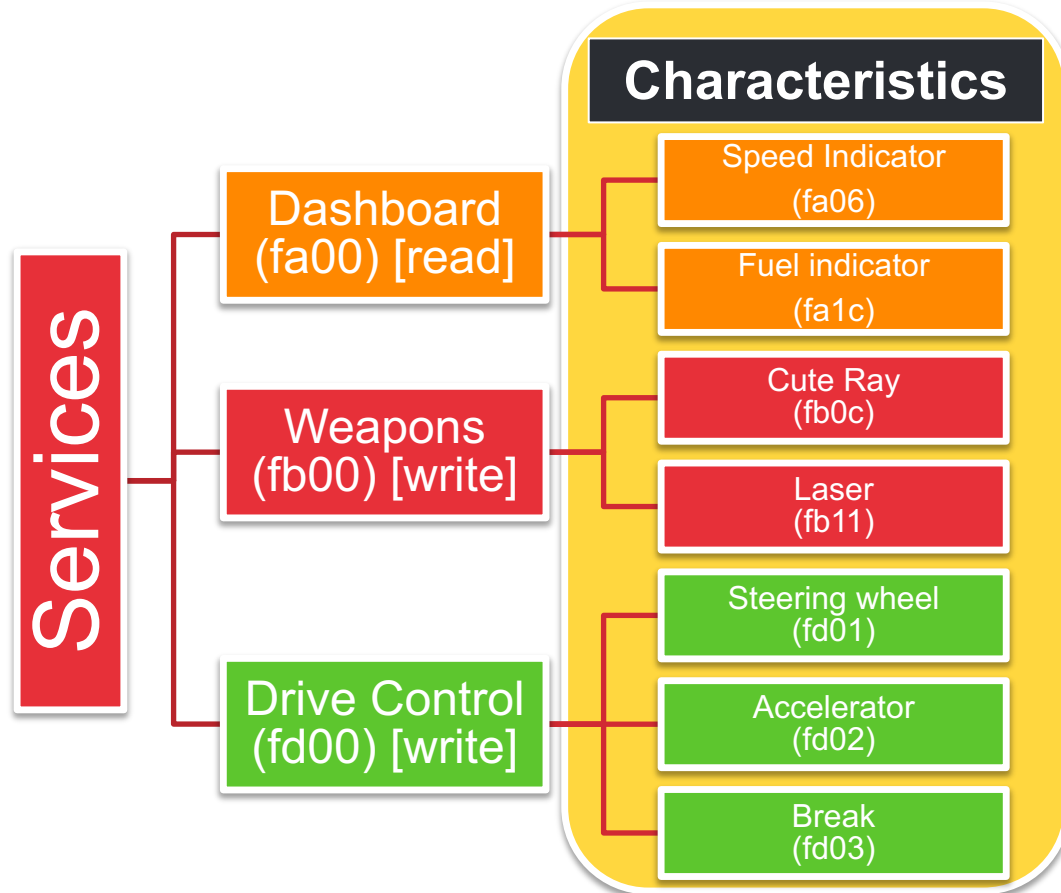
Functionality is grouped into Services



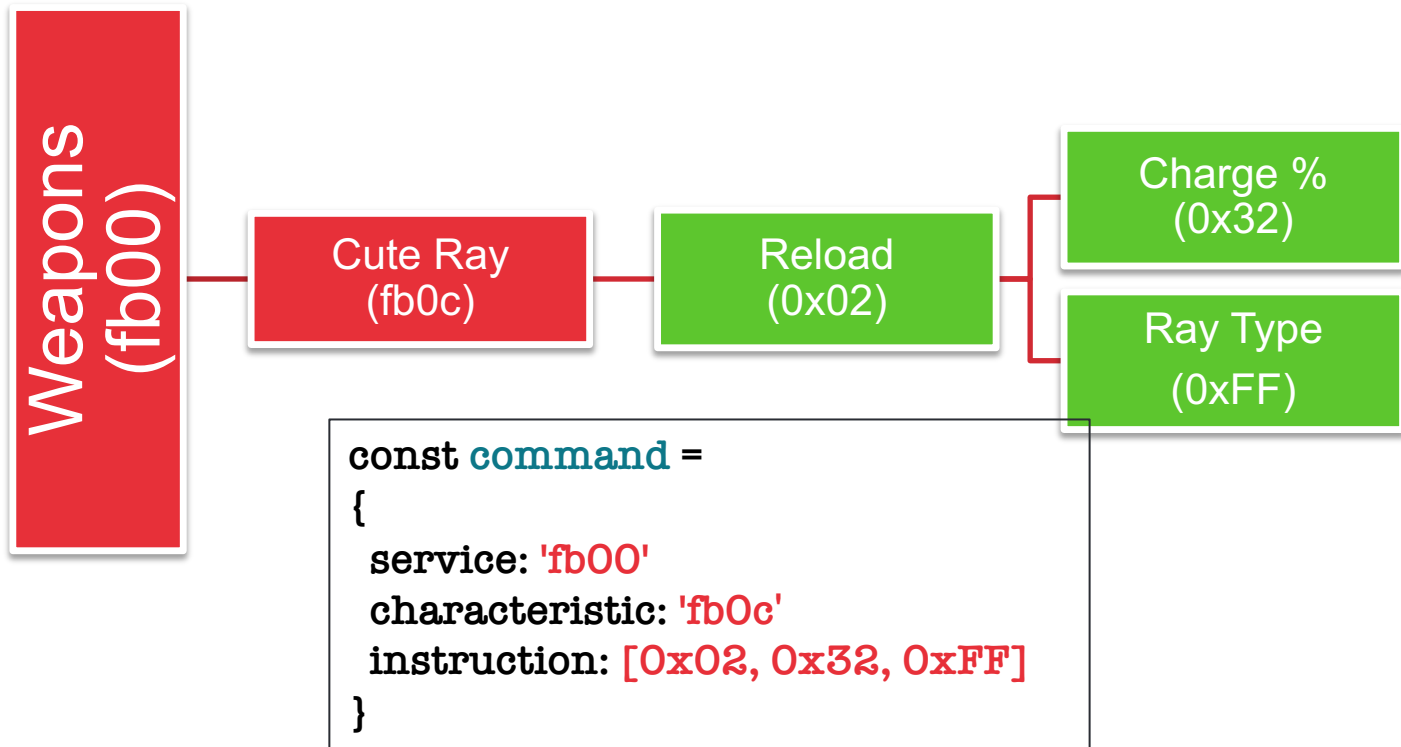
Services contain characteristics



Characteristics interactions



Sending commands



Run Bluetooth from

Web
Mobile



Web Bluetooth



Web bluetooth in 5 easy steps

0. Add `web-bluetooth` Typing
1. `Scan` for device
2. `Connect` to it
3. `Get` the `Service` you need
4. `Get` the `Characteristic` you need
5. `Read` / `Write` from the `Characteristic`



Step 0 – Add web-bluetooth

```
npm install --save @types/web-bluetooth
```



Step 1 - Find a matching Device

```
/// <reference types="web-bluetooth" />
```

```
let device =  
  await navigator.bluetooth.requestDevice(  
    {  
      filters: [{ namePrefix: 'Mambo' }],  
      optionalServices: [serviceUUID]  
    })
```



Steps 2-4 - Connect and get characteristic

```
let server =  
    await device.gatt.connect();
```

```
let service =  
    await server.getPrimaryService(serviceUUID);
```

```
let characteristic =  
    await service.getCharacteristic(characteristicUUID);
```



Step 5 - Read

```
const value =  
    await characteristic.readValue();  
  
console.log('val:' + value.getUint8(0));
```



Step 5 - Write

```
const instructions = new Int8Array([0x02, 0x32, 0xFF]);  
  
characteristic.writeValue(instructions);
```



First volunteers arrive





find packages

★ mambo-angular-service public

Angular Service to control Parrot Mambo drone

```
this.drone = await this.mamboService.search();  
// wait until ready...  
this.drone.takeOff();
```

- fire
- land
- setAltitude
- setMaxAltitude
- setMaxRotationSpeed
- setMaxTilt
- setMaxVerticalSpeed
- setPitch
- setRoll
- setYaw
- takeOff (method) Drone.takeOff(): void
Instructs the drone to take off
- updateFlightParams

<https://www.npmjs.com/package/mambo-angular-service>



Time to send the drone to scout on the enemy



What about going beyond Chrome?



Bluetooth from a mobile app with NativeScript



NativeScript is...

an **open source** framework for building **truly native** mobile apps with **JavaScript**. Use web skills, like TypeScript, **Angular** and CSS, and **get native UI and performance** on iOS and Android.



NativeScript Bluetooth in 4 easy steps

0. Add `nativescript-bluetooth`
1. Request permissions
2. Scan for device
3. Connect to it
4. Read / Write from the Characteristic



Step 0 – Add nativescript-bluetooth

```
tns plugin add nativescript-bluetooth
```



Step 1 – Request permissions

```
import bluetooth = require('nativescript-bluetooth');  
  
bluetooth.requestCoarseLocationPermission().then(  
  () => console.log('Location permission requested')  
);
```



Step 2 - Find a matching Device

```
bluetooth.startScanning({  
    seconds: 3,  
  
})
```



Step 2 - Find a matching Device

```
bluetooth.startScanning({  
  seconds: 3,  
  onDiscovered: (peripheral: Peripheral) => {  
    console.log('Device UUID: ' + peripheral.UUID);  
    console.log('Device name: ' + peripheral.name);  
  }  
})
```



Step 3 - Connect to the Device

```
bluetooth.connect({  
  UUID: deviceUUID,  
  
});
```



Step 3 - Connect to the Device

```
bluetooth.connect({  
  UUID: deviceUUID,  
  onConnected: ((peripheral: Peripheral) => {  
    console.log('Device connected: ' + peripheral.UUID);  
  } ),  
  
});
```



Step 3 - Connect to the Device

```
bluetooth.connect({  
  UUID: deviceUUID,  
  onConnected: ((peripheral: Peripheral) => {  
    console.log('Device connected: ' + peripheral.UUID);  
  }),  
  onDisconnected: ((peripheral: Peripheral) => {  
    console.log('Device disconnected');  
  })  
});
```



Step 4 - Read

```
bluetooth.read({  
  peripheralUUID: deviceUUID,  
  serviceUUID: serviceUUID,  
  characteristicUUID: characteristicUUID  
})
```



Step 4 - Read

```
bluetooth.read({  
  peripheralUUID: deviceUUID,  
  serviceUUID: serviceUUID,  
  characteristicUUID: characteristicUUID  
})  
.then((result: ReadResult) => {  
  const data = new Uint8Array(result.value);  
});
```



Step 4 - Write

```
const bluetoothOptions: WriteOptions = {  
  peripheralUUID: deviceUUID,  
  serviceUUID: serviceUUID,  
  characteristicUUID: characteristicUUID,  
  value: '0x02, 0x32, 0xFF'  
}
```



Step 4 - Write

```
const bluetoothOptions: WriteOptions = {  
  peripheralUUID: deviceUUID,  
  serviceUUID: serviceUUID,  
  characteristicUUID: characteristicUUID,  
  value: '0x02, 0x32, 0xFF'  
}  
bluetooth.writeWithoutResponse(bluetoothOptions);
```



Another hero joins the quest





find packages

★ nativescript-mip-ble public

NativeScript plugin that allows to control a Mip robot.

```
var mip = new MipDevice(UUID, name, status);  
mip.
```

- move (method) MipDevice.move(speed: any, turn: any): void
- moveBack
- moveForward
- name
- notify
- notifyPropertyChange
- off
- on
- removeEventListener
- set
- setHeadLED
- state

<https://www.npmjs.com/package/nativescript-mip-ble>



General Mipuś reporting for duty
AKA demo time



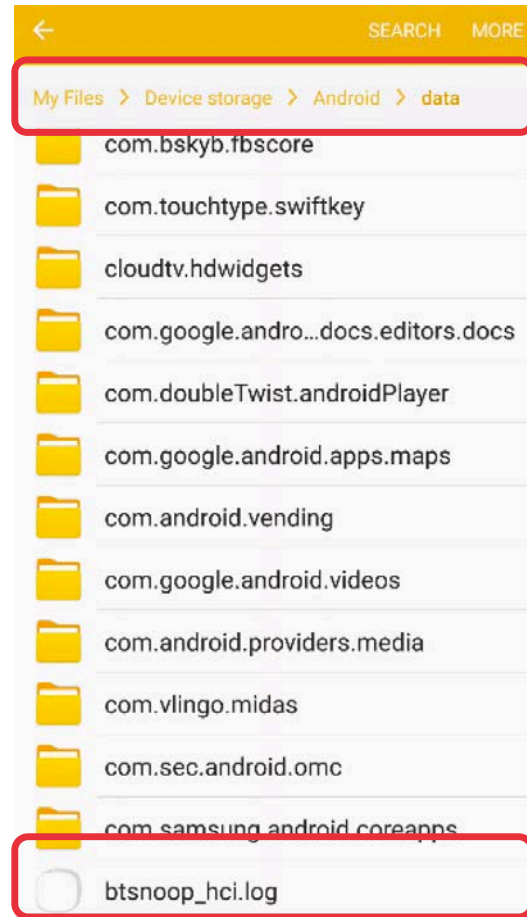
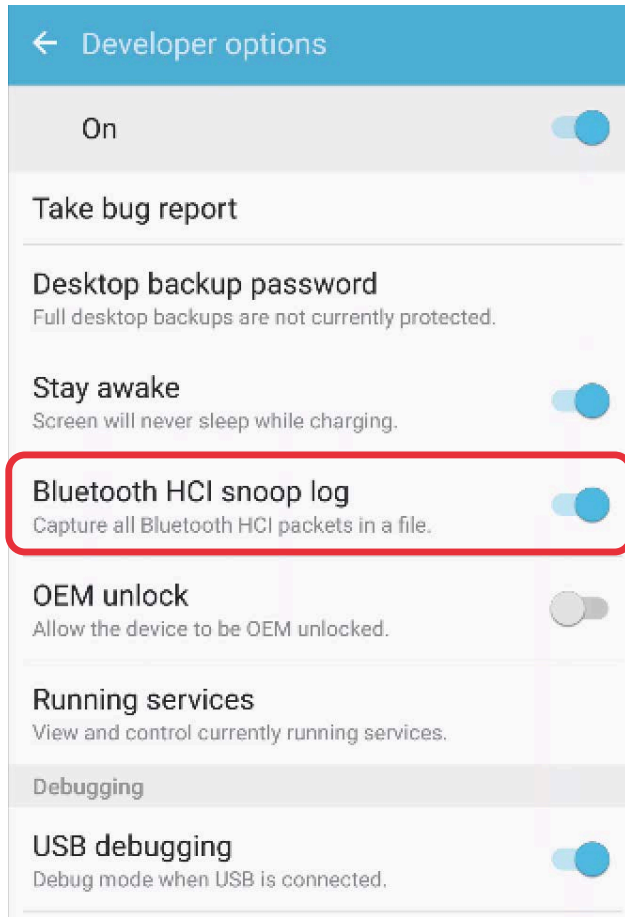
Reverse Engineering

```

    <a> .sf-sub-indicator {
    parent .cart-menu .cart-icon-wr
    outer-outer.transparent header#top
    nav .sf-menu > li.current_page
    nav .sf-menu > li.current-menu
    nav > ul > li > a: hover > .sf-sub
    nav ul #search-btn a: hover span, #
    nav .sf-menu > li.current-menu-it
    : hover .icon-salient-cart, .ascend
    !important; color: #ffffff !impo
    parent header#top nav > ul > li.but
    .yc-widget-area-toggle a i
    .filter-filter transparent

```

Android Developer Tools



Wireshark

Wireshark interface showing a packet capture file named `btsnoop_hci-mip.log.pcap`. The filter bar displays `btatt.opcode.method==0x12`.

No.	Time	Source	Length	Value	Service UUID	UUID
1418	218.436389	SamsungE_13:35:1d (Galaxy S6 edge+)	30	06230000000000000000000000000000	65509	65513
1420	220.585078	SamsungE_13:35:1d (Galaxy S6 edge+)	30	06140000000000000000000000000000	65509	65513
1423	236.118518	SamsungE_13:35:1d (Galaxy S6 edge+)	15	71188c	65509	65513
1425	238.075666	SamsungE_13:35:1d (Galaxy S6 edge+)	15	72188c	65509	65513

Frame 1423: 15 bytes on wire (120 bits), 15 bytes captured (120 bits)

- Bluetooth
- Bluetooth HCI H4
- Bluetooth HCI ACL Packet**
- Bluetooth L2CAP Protocol
- Bluetooth Attribute Protocol
 - Opcode: Write Command (0x52)
 - Handle: 0x0013 (Unknown: Unknown)
 - [Service UUID: Unknown (0xffe5)]
 - [UUID: Unknown (0xffe9)]
 - Value: 71188c**

0000 02 40 00 0a 00 06 00 04 00 52 13 00 71 18 8c .@..... .R..q..

Looking for the bullet:

Test instructions

1. Fire
2. Fire
3. TakeOff
4. Land
5. Fire
6. Fire
7. Fire
8. TakeOff
9. Land



Looking for the bullet:

Test instructions

TakeOff: [4, **step**, 2, 0, 1, 0]

Land: [4, **step**, 2, 0, 3, 0]



Looking for the bullet:

Test instructions

TakeOff: ^{prefix} [4, **step**, 2, 0, 1, 0]

Land: [4, **step**, 2, 0, 3, 0]



Looking for the bullet:

Test instructions

TakeOff: $\left[4, \text{step}, 2, 0, 1, 0 \right]$

Land: $\left[4, \text{step}, 2, 0, 3, 0 \right]$

Talk to the
(wire) shark



No.	Time	Source	Length	Value	Service UUID	UUID
2910	89.944749	localhost ()	27	026a020002000000000000deba0000	9a66fa000800919111e4012d1540cb8e	9a66fa0a0800919111e4012d1540cb8e
2912	90.275792	localhost ()	27	026b020002000000000000fbc0000	9a66fa000800919111e4012d1540cb8e	9a66fa0a0800919111e4012d1540cb8e
2914	90.632524	localhost ()	27	026c020002000000000000fbd0000	9a66fa000800919111e4012d1540cb8e	9a66fa0a0800919111e4012d1540cb8e
574	14.955036	localhost ()	29	040100040100323031372d30332d323900	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
577	15.176630	localhost ()	31	040200040200543135313032362b3031303000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
580	15.292290	localhost ()	18	040300020000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
620	16.494624	localhost ()	18	040400040000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
649	17.211076	localhost ()	25	0405020e01007a656e6c746500	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
655	17.353535	localhost ()	27	0406020e0000534d2d473932384600	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
660	17.486152	localhost ()	19	0407020a000001	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
670	17.904194	localhost ()	22	04080208010000007041	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
678	18.047565	localhost ()	22	0409020100000000803f	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
684	18.130421	localhost ()	22	040a020101000000f042	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
689	18.212033	localhost ()	22	040b0208000000002041	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
891	22.399162	localhost ()	22	040c001a000003000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1603	51.507254	localhost ()	29	040100040100323031372d30332d323900	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1607	51.633213	localhost ()	31	040200040200543135313130332b3031303000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1612	51.701444	localhost ()	18	040300020000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1655	52.321859	localhost ()	18	040400040000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1703	52.827726	localhost ()	25	0405020e01007a656e6c746500	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1709	52.900106	localhost ()	27	0406020e0000534d2d473932384600	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1716	52.962773	localhost ()	19	0407020a000001	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1722	53.016524	localhost ()	22	04080208010000007041	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1732	53.008035	localhost ()	22	0409020100000000803f	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1736	53.124146	localhost ()	22	040a020101000000f042	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1743	53.190440	localhost ()	22	040b0208000000002041	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2507	55.639874	localhost ()	22	040c001a000003000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2575	59.755096	localhost ()	23	040d021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2604	62.815980	localhost ()	23	040e021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2630	66.235318	localhost ()	18	040f02000100	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2703	69.873556	localhost ()	18	041002000300	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2740	73.967724	localhost ()	23	0411021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2762	76.459208	localhost ()	23	0412021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2818	79.376117	localhost ()	23	0413021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2852	83.304751	localhost ()	18	041402000100	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2868	84.886739	localhost ()	18	041502000300	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
582	15.343926	localhost ()	15	010101	9a66fa000800919111e4012d1540cb8e	9a66fa1e0800919111e4012d1540cb8e
588	15.443673	localhost ()	15	010202	9a66fa000800919111e4012d1540cb8e	9a66fa1e0800919111e4012d1540cb8e
591	15.518958	localhost ()	15	010303	9a66fa000800919111e4012d1540cb8e	9a66fa1e0800919111e4012d1540cb8e
593	15.604010	localhost ()	15	010404	9a66fa000800919111e4012d1540cb8e	9a66fa1e0800919111e4012d1540cb8e

btatt.opcode.method==0x12						
No.	Time	Source	Length	Value	Service UUID	UUID
574	14.955036	localhost ()	29	040100040100323031372d30332d323900	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
577	15.176630	localhost ()	31	040200040200543135313032362b3031303000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
580	15.292290	localhost ()	18	040300020000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
620	16.494624	localhost ()	18	040400040000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
649	17.211076	localhost ()	25	0405020e01007a656e6c746500	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
655	17.353535	localhost ()	27	0406020e0000534d2d473932384600	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
660	17.486152	localhost ()	19	0407020a000001	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
670	17.904194	localhost ()	22	04080208010000007041	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
678	18.047565	localhost ()	22	0409020100000000803f	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
684	18.130421	localhost ()	22	040a020101000000f042	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
689	18.212033	localhost ()	22	040b0208000000002041	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
891	22.399162	localhost ()	22	040c001a000003000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1603	51.507254	localhost ()	29	040100040100323031372d30332d323900	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1607	51.633213	localhost ()	31	040200040200543135313130332b3031303000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1612	51.701444	localhost ()	18	040300020000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1655	52.321859	localhost ()	18	040400040000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1703	52.827726	localhost ()	25	0405020e01007a656e6c746500	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1709	52.900106	localhost ()	27	0406020e0000534d2d473932384600	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1716	52.962773	localhost ()	19	0407020a000001	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1722	53.016524	localhost ()	22	04080208010000007041	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1732	53.008035	localhost ()	22	0409020100000000803f	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1736	53.124146	localhost ()	22	040a020101000000f042	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
1743	53.190440	localhost ()	22	040b0208000000002041	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2507	55.639874	localhost ()	22	040c001a000003000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2575	59.755096	localhost ()	23	040d021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2604	62.815980	localhost ()	23	040e021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2630	66.235318	localhost ()	18	040f02000100	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2703	69.873556	localhost ()	18	041002000300	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2740	73.967724	localhost ()	23	0411021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2762	76.459208	localhost ()	23	0412021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2818	79.376117	localhost ()	23	0413021002000000000000	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2852	83.304751	localhost ()	18	041402000100	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e
2868	84.886739	localhost ()	18	041502000300	9a66fa000800919111e4012d1540cb8e	9a66fa0b0800919111e4012d1540cb8e

040c001a000003000000
040d021002000000000000
040e021002000000000000
040f02000100
041002000300
0411021002000000000000
0412021002000000000000
0413021002000000000000

Take Off
Land



041402000100
041502000300

Take Off
Land



Take Off
Land



1a000003000000
1002000000000000
1002000000000000
000100
000300
1002000000000000
1002000000000000
1002000000000000
000100
000300

Take Off
Land



1a000003000000

1002000000000000

1002000000000000

000100

000300

1002000000000000

1002000000000000

1002000000000000

000100

000300

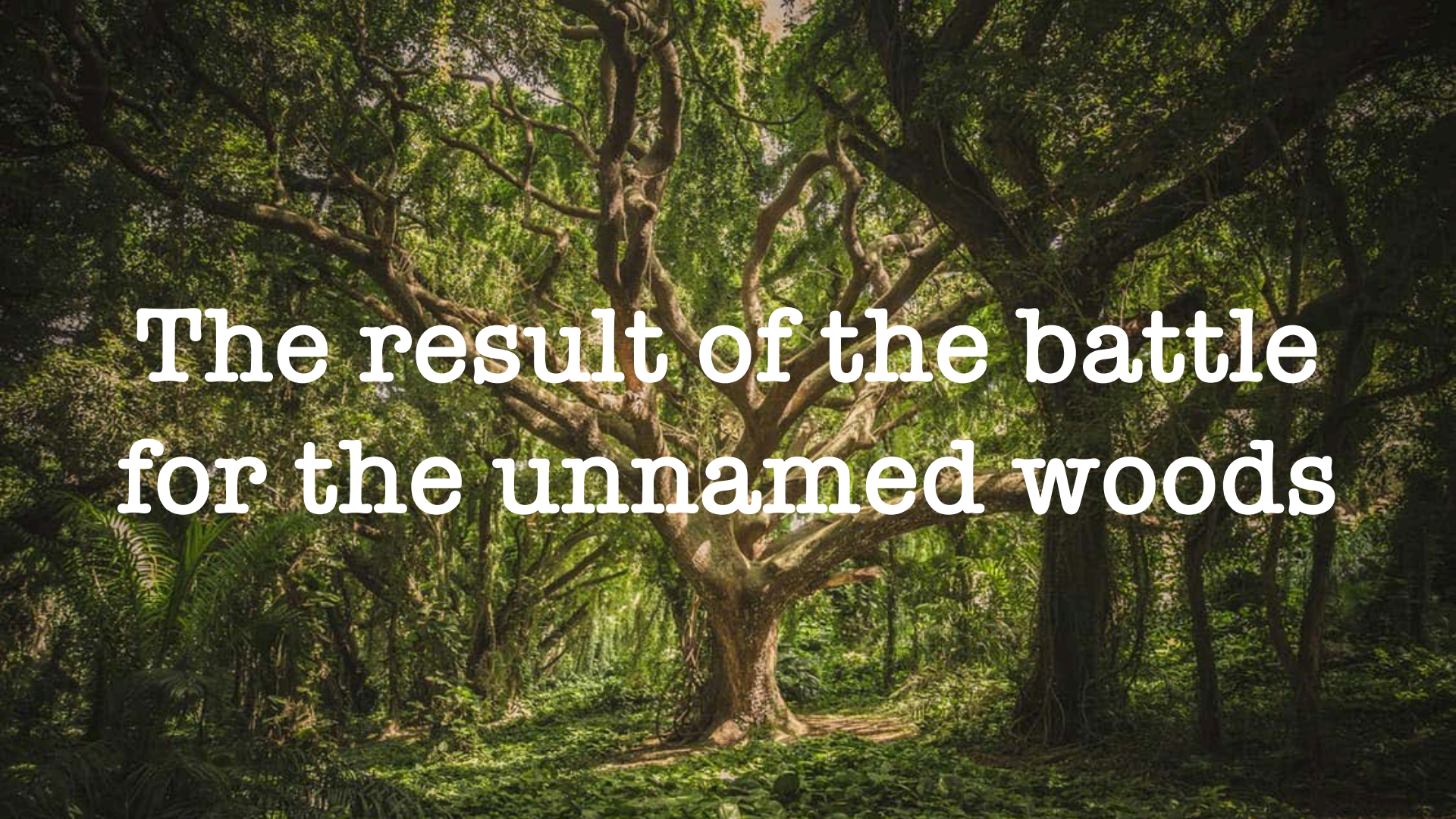


Fire Instruction

04 xx 02 10 02 00 00 00 00 00 00

```
/**  
 * Instructs the drone to fire the cannon  
 */  
public fire() {  
    this.flightCommandInstructions.writeWithoutResponse([16, 2, 0, 0, 0, 0, 0, 0]);  
}
```



A large, ancient tree with a thick, gnarled trunk and sprawling branches stands as the centerpiece in a dense, lush forest. Sunlight filters through the thick canopy of green leaves, creating dappled light on the forest floor. The ground is covered in a thick layer of green ferns and other low-lying plants. The overall atmosphere is serene and majestic, suggesting a place of great age and natural beauty.

The result of the battle
for the unnamed woods

Unconditional surrendered



Summary

Bluetooth is **fun** and **easy** to use
npm examples to get started:

- [mambo-angular-service](#)
- [nativescript-mambo-ble](#)
- [nativescript-mip-ble](#)



Summary

Use NativeScript:

- for Android and iOS apps support
- up to **10** devices connected

Use Web Bluetooth:

- to run in Chrome on Mac and Windows and Android
- only **one** device connection

For WB with Observables use:

- [@manekinekko/angular-web-bluetooth](https://github.com/manekinekko/angular-web-bluetooth)



Summary

Reverse engineering can be tricky, use:

- Android HCI Snoop log
- Together with WireShark

Advanced options:

- Pattern matching tools – group by command value
- Replay tool



