

How to connect to a device and discovery all services and characters

井民全, Jing, mqjing@gmail.com

```
var noble = require('noble');
const SERVICE_UUID = 'xxxx';

noble.on('stateChange', function(state) {
  if (state === 'poweredOn') {
    console.log('powered On');
    console.log('Scanning');
    noble.startScanning([SERVICE_UUID]);
  } else {
    console.log('powered Off');
    noble.stopScanning();
  }
});

noble.on('discover', function(peripheral) {
  console.log('peripheral.id ' + peripheral.id);
  console.log('peripheral.address ' + peripheral.address);
  noble.stopScanning(); // connect to the first peripheral that is scanned
  var advertisement = peripheral.advertisement;
  var localName = advertisement.localName;
  if (localName) {
    console.log('local Name ' + localName);
  }

  var manufacturerData = advertisement.manufacturerData;
  if (manufacturerData) {
    console.log('manufacturerData ' + manufacturerData);
  }

  console.log(' ');

  peripheral.connect(error => {
```

```

    console.log('Connected to', peripheral.id);

    peripheral.discoverAllServicesAndCharacteristics( function (error, services,
characteristics){
    console.log('all services =', services)
    console.log('characteristics = ', characteristics)

    services.forEach(function(service, index, array){
        console.log('service ' + service + ' index ' + index)
    })

    characteristics.forEach(function(characteristic, index, array){
        console.log('characteristics ' + characteristic + ' index ' + index)
    })

    });
});
});

```

Run

```
pi@raspberrypi:~/work $ sudo node connect.js
```

Result

```

pi@raspberrypi:~/work $ sudo node connect.js
powered On
Scanning
peripheral.id 00802554c1a2
peripheral.address 00:80:25:54:c1:a2
local Name AVHCV10642
manufacturerData  

Connected to 00802554c1a2
service {"uuid":"1800","name":"Generic
Access","type":"org.bluetooth.service.generic_access","includedServiceUids":null} index 0
service {"uuid":"1801","name":"Generic
Attribute","type":"org.bluetooth.service.generic_attribute","includedServiceUids":null} index 1
service {"uuid":"180a","name":"Device

```

```
Information","type":"org.bluetooth.service.device_information","includedServiceUids":null}
index 2
service {"uuid":"fefb","name":null,"type":null,"includedServiceUids":null} index 3
characteristics {"uuid":"2a00","name":"Device
Name","type":"org.bluetooth.characteristic.gap.device_name","properties":["read","write"]} index
0
characteristics
{"uuid":"2a01","name":"Appearance","type":"org.bluetooth.characteristic.gap.appearance","prope
rties":["read"]} index 1
characteristics {"uuid":"2a04","name":"Peripheral Preferred Connection
Parameters","type":"org.bluetooth.characteristic.gap.peripheral_preferred_connection_paramet
ers","properties":["read"]} index 2
characteristics {"uuid":"2a50","name":"PnP
ID","type":"org.bluetooth.characteristic.pnp_id","properties":["read"]} index 3
characteristics {"uuid":"2a05","name":"Service
Changed","type":"org.bluetooth.characteristic.gatt.service_changed","properties":["indicate"]}
index 4
characteristics
{"uuid":"00000001000010008000008025000000","name":null,"type":null,"properties":["writeWith
outResponse"]} index 5
characteristics
{"uuid":"00000002000010008000008025000000","name":null,"type":null,"properties":["notify"]}
index 6
characteristics
{"uuid":"00000003000010008000008025000000","name":null,"type":null,"properties":["write"]}
index 7
characteristics
{"uuid":"00000004000010008000008025000000","name":null,"type":null,"properties":["indicate"]
} index 8
```