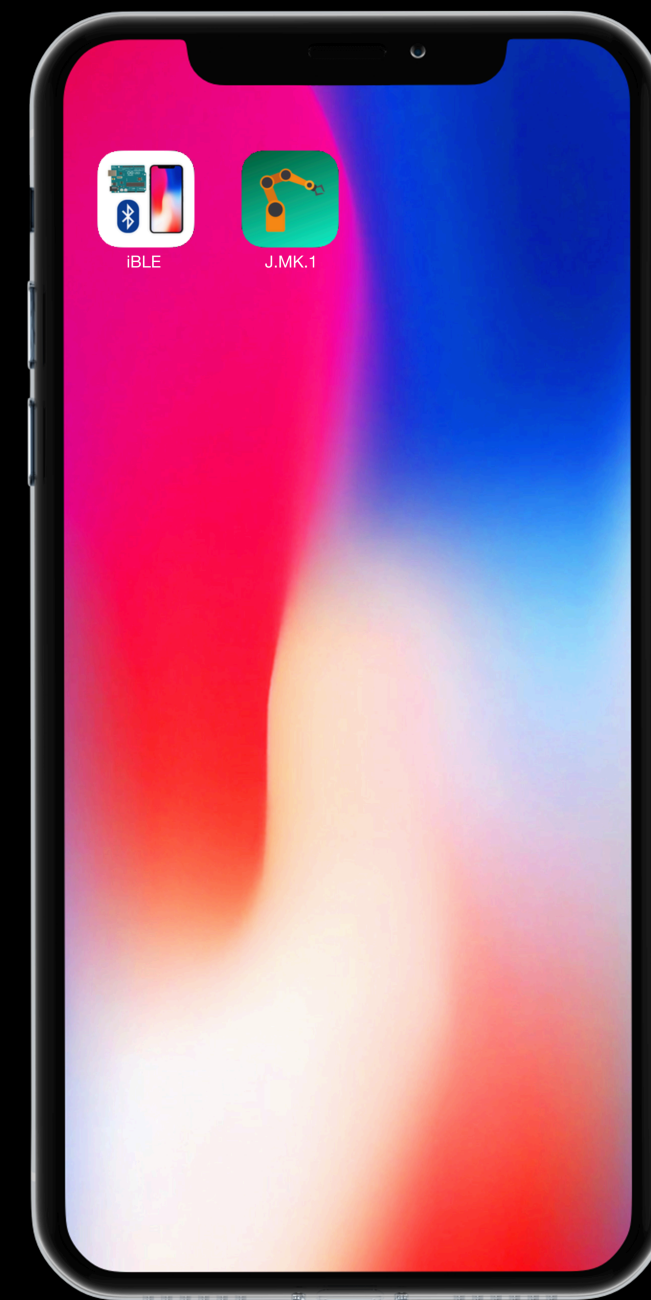
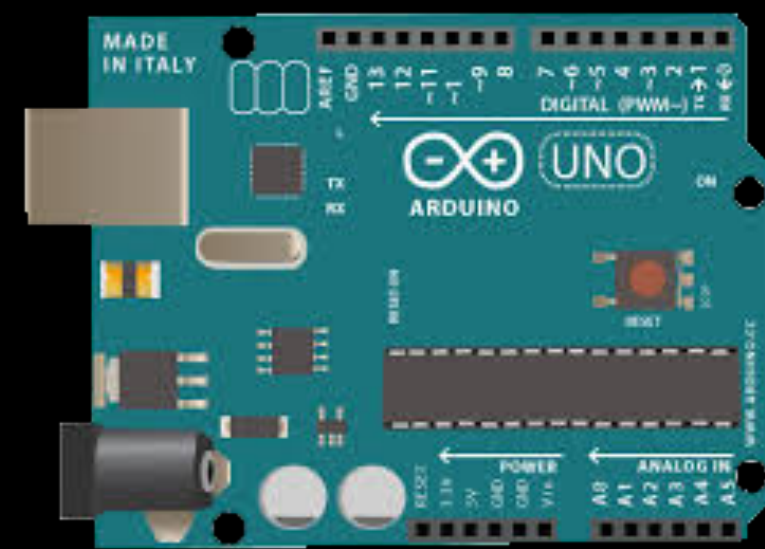


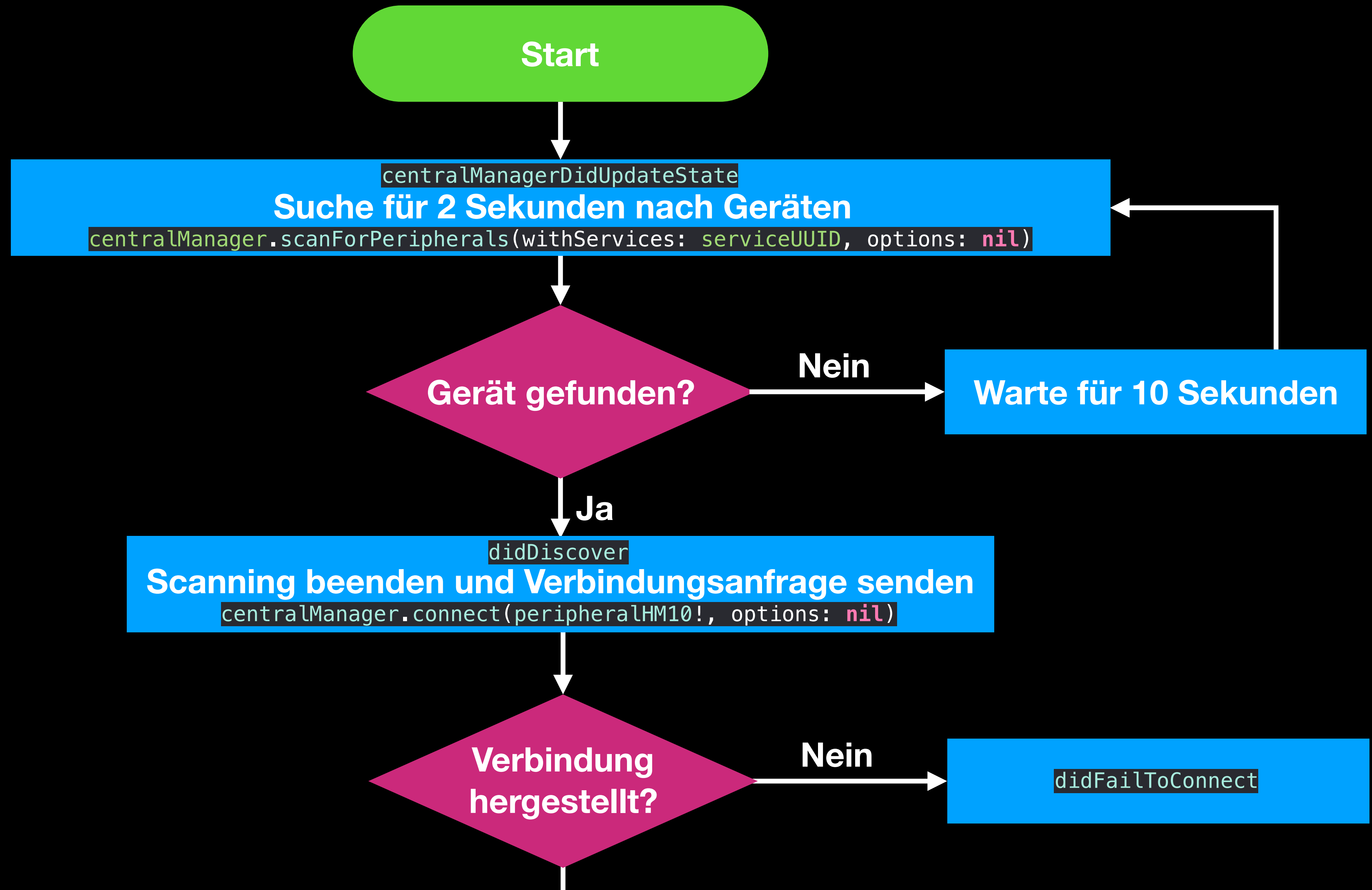
Entwicklung einer iOS-App für eine Bluetooth Low Energy Verbindung zum Arduino

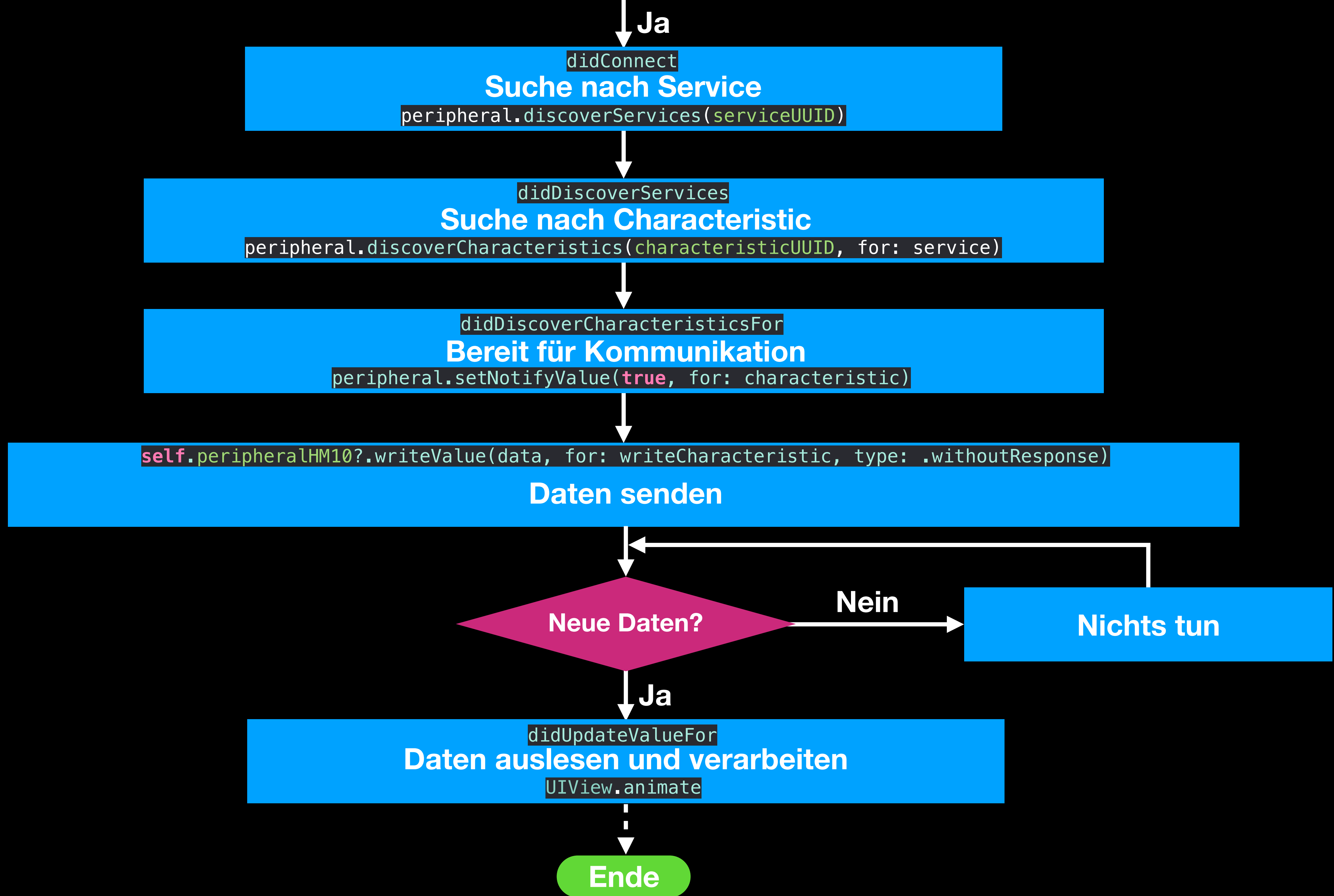


Gliederung

- Programmablaufplan (PAP) zum Verbindungsaufbau
- Demo zum Aufbau der BLE Verbindung
- Inwiefern ermöglicht diese Arbeit neue, zukünftige BLLs?
- Live-Demo mit dem selbstkonstruierten Roboterarm
- Fragen

Programmablaufplan





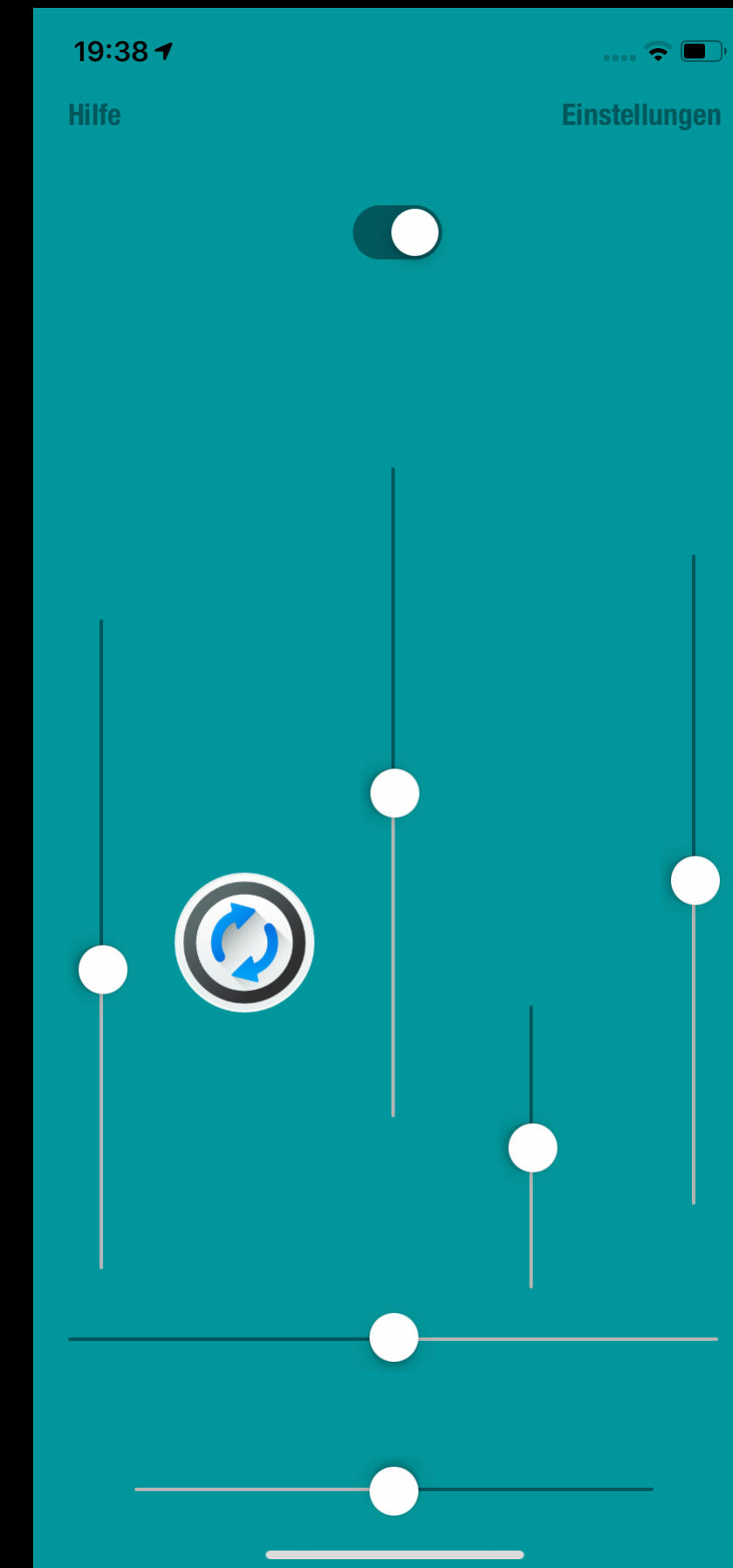
Demo zum Aufbau der BLE Verbindung

- Bildschirmaufnahme
- Demonstration mithilfe des Debuggers von Xcode
- Breakpoints
- Schrittweiser Aufbau
- Theorieteil essentiell fürs Verständnis

Inwiefern ermöglicht diese Arbeit neue, zukünftige BLLs?

- Fundament, um iOS Geräte mit Arduino Mikrocontrollern kommunizieren zu lassen
- Beispiele
 - Wetterstation
 - Roboter aller Art
 - Geräte aus dem Haushalt sparsam vernetzen (IoT)

Live-Demo Roboterarm





```
1 //
2 // ViewController.swift
3 // BLL_1.3
4 //
5 // Created by Paul Knausst on 04.07.18.
6 // Copyright © 2018 Paul. All rights reserved.
7 //
8
9 import UIKit
10 import CoreBluetooth
11 import CoreMotion
12 class ViewController: UIViewController, CBCentralManagerDelegate, CBPeripheralDelegate,
    UIPickerViewDataSource, UIPickerViewDelegate, UITextFieldDelegate {
13
14
15     let serviceUUID = [CBUUID(string:"FFE0")]
16     let characteristicUUID = [CBUUID(string:"FFE1")]
17
18     let timerPauseInterval:TimeInterval = 10.0
19     let timerScanInterval:TimeInterval = 2.0
20
21     let defaults = UserDefaults.standard
22
23     var centralManager: CBCentralManager!
24     var peripheralHM10: CBPeripheral?
25     var writeCharacteristic: CBCharacteristic?
26
27     var keepScanning = false
28     var state = ""
29     var peripheralName1 = ""
30     var timer = Timer()
31
32     @IBOutlet weak var textInput: UITextField!
33     @IBOutlet weak var pickerView: UIPickerView!
```

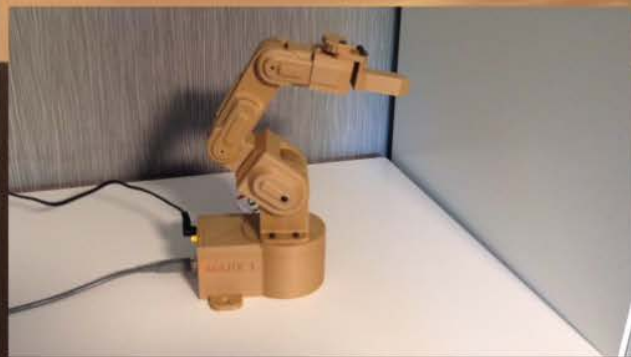
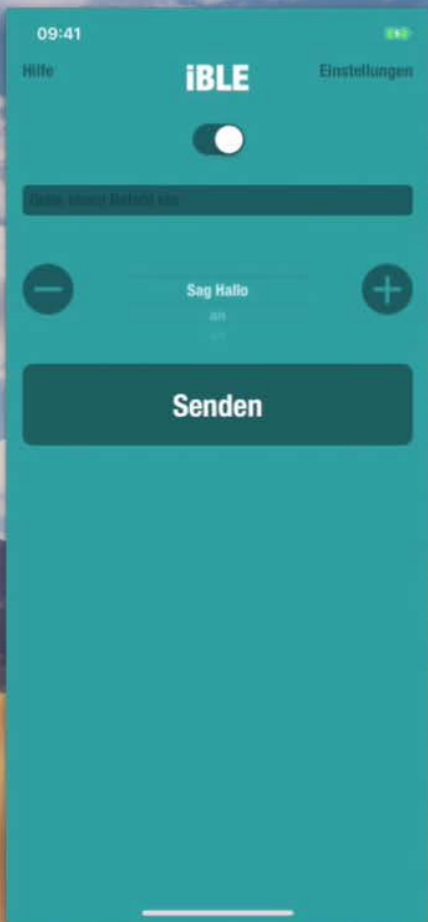
Identity and Type
Name: ViewController.swift
Type: Default - Swift Source
Location: Relative to Group
View Controller.swift
Full Path: /Users/.../Desktop/BLL_1.3/ViewController.swift

On Demand Resource Tags
Target Membership
BLL_1.3

Text Settings
Text Encoding: UTF-8 Explicit Encoding
Line Endings: LF
Indent Style: Spaces
Wraps: Tab, Indent, Wrap Lines

```
<fault>, <BefehlMO: 0x2030e0a0e> (entity: Befehl; id:
0xf8bec01e0cf9658a
<x-coredata://DA296AD9-8E7D-4DD1-A47B-362EE3F14038/Befehl/p2> ; data:
<fault>, <BefehlMO: 0x2836e5a40> (entity: Befehl; id:
0xf8bec01e0cf5658a
<x-coredata://DA296AD9-8E7D-4DD1-A47B-362EE3F14038/Befehl/p1> ; data:
<fault>]]
Optional("Sag Hallo")
Peripheral name: arduino_HM10
UUID: D0E81BEA-BE4A-8D71-9A2F-E99494B2B7C0
RSSI: -74
ERROR: disconnection details. The connection has timed out unexpectedly.
```


3.



```

// CBCentralManagerDelegate methods
func centralManagerDidUpdateState(_ central: CBCentralManager)
{
    switch central.state
    {
    case .poweredOn:
        keepScanning = true
        // NSTimer manages how often the CB Manager scans for peripherals (To manage battery
        // life)
        // Scan 2 seconds .. wait 10 seconds .....
        _ = Timer(timeInterval: timerScanInterval, target: self, selector:
            #selector(pauseScan), userInfo: nil, repeats: false)

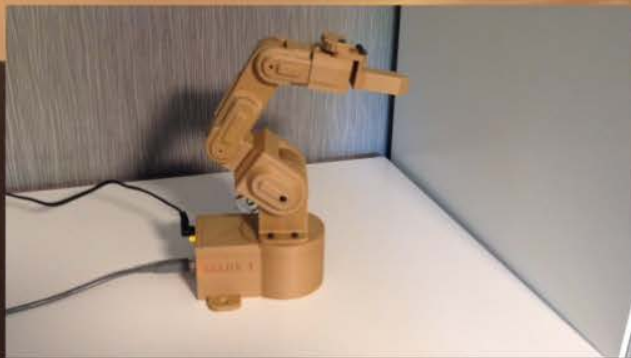
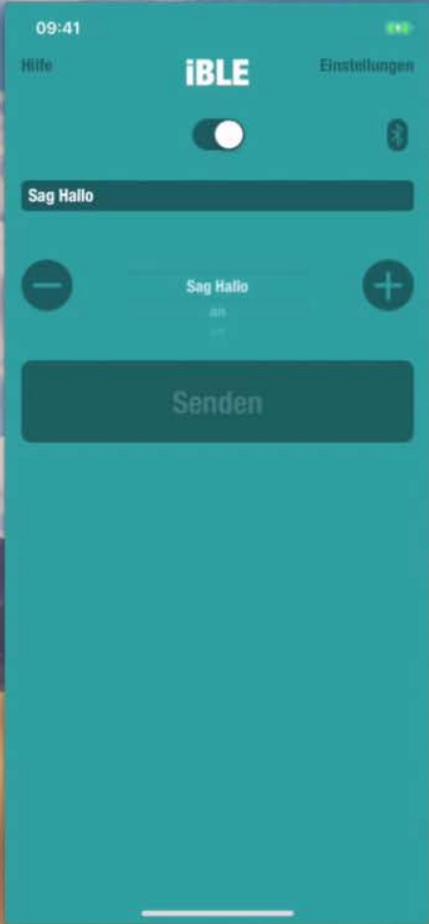
        // searches for peripheral with the service FFE0
        96 centralManager.scanForPeripherals(withServices: serviceUUID, options: nil) = Thread...
        sendButton.isEnabled = false

    case .poweredOff:
        state = "Bluetooth is powered off."
    case .unsupported:
        state = "This device does not support BLE"
    case .unauthorized:
        state = "This app is not authorized to use BLE"
    case .resetting:
        state = "BLE Manager is resetting, pending state"
    case .unknown:
        state = "The state of the BLE Manager is unknown"
    }
}
}

// invoked when central manager discovers peripherals while scanning
<<X-coredata://DA296AD9-0E7D-4DD1-A47B-362EE3F14038/Befehl/p1>> ; data:
<fault>, <BefehlMO: 0x2827ce260> (entity: Befehl; id:
0xe880f040014013f2
<X-coredata://DA296AD9-0E7D-4DD1-A47B-362EE3F14038/Befehl/p3> ; data:
<fault>, <BefehlMO: 0x2827ce210> (entity: Befehl; id:
0xe880f040014413f2
<X-coredata://DA296AD9-0E7D-4DD1-A47B-362EE3F14038/Befehl/p2> ; data:
<fault>, <BefehlMO: 0x2827ce170> (entity: Befehl; id:
0xe880f040014013f2
<X-coredata://DA296AD9-0E7D-4DD1-A47B-362EE3F14038/Befehl/p1> ; data:
<fault>}}
Optional("Sag Hallo")
(11db)

```

8.



Running BLL_1.3 on Pauls iPhone 14S

```

arduino on the RX pin
268 func writeInteger(_ position: UInt8)
269 {
270     // checks if characteristic was discovered
271     if let writeCharacteristic = self.writeCharacteristic
272     {
273         let data = Data(bytes: [position])
274         self.peripheralHM10?.writeValue(data, for: writeCharacteristic, type:
                .withoutResponse) // did not work with type:
                CBCharacteristicWriteType.withResponse
                //print("write position")
275     }
276 }
277
278
279 func writeString(_ message: String)
280 {
281     if let writeCharacteristic = self.writeCharacteristic
282     {
283         if let data = message.data(using: String.Encoding.utf8)
284         {
285             self.peripheralHM10?.writeValue(data, for: writeCharacteristic, type:
                .withoutResponse)
286         }
287     }
288 }
289
290 //print(data)
291 //print(String(data: data, encoding: String.Encoding.utf8))
292
293 // Picker methods
294
295 func numberOfComponents(in savedCommands: UIPickerView) -> Int {
296     return 1
297 }

```

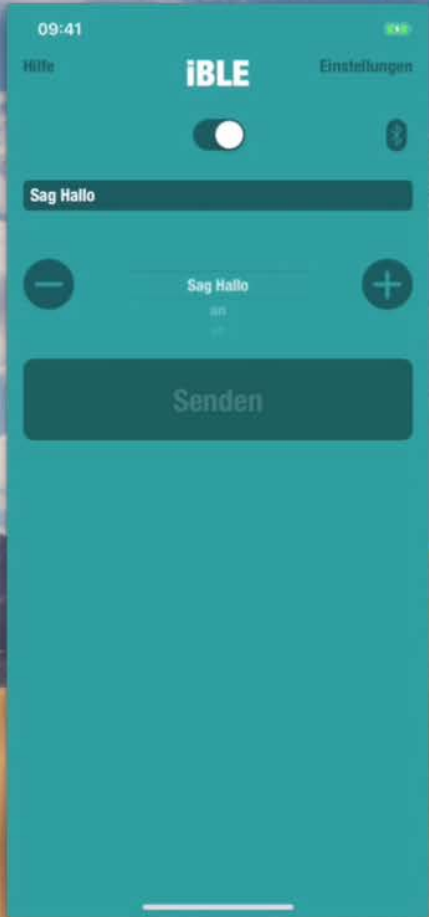
Thread 1: 0 ViewController.writeString(_:)
 0 self = 0x0000000105304870
 1 writeCharacteristic = (CBCharacteristic) 0x00000002820bc900
 2 data = (Data) 10 bytes

```

<XCOREDATA://DA296AD9-0E7D-4DD1-A47B-362EE3F14038/Befehl/p1>; data:
<fault>, <BefehlM0: 0x2827ce170> (entity: Befehl; id:
0xe880f840014813f2
<X-coredata://DA296AD9-0E7D-4DD1-A47B-362EE3F14038/Befehl/p1>; data:
<fault>]]
Optional("Sag Hallo")
Peripheral name: arduino_HM10
UUID: D0E81BEA-BE4A-8D71-9A2F-E9949482B7C8
RSSI: -76
2019-05-12 19:06:49.261635+0200 BLL_1.3[4931:1448571] [UIWorkIntervalTiming]
workIntervalStart: startTimestamp > targetTimestamp; rolling forward by
1.003333
(11db)

```


9.



Running BLL_1.3 on Pauls iPhone XR

```

226 // is invoked when the app calls the readValueForCharacteristic: method or when the
227 peripheral notifies the app that the value of the characteristic has changed
228 func peripheral(_ peripheral: CBPeripheral, didUpdateValueFor characteristic:
229     CBCharacteristic, error: Error?)
230 {
231     if error != nil
232     {
233         print(error!.localizedDescription)
234         return
235     }
236     if let data = characteristic.value
237     {
238         if let dataAsString = String(data: data, encoding: String.Encoding.utf8)
239         {
240             //print(dataAsString)
241             if dataAsString.contains("empfangen")
242             {
243                 UIView.animate(withDuration: 0.5, delay: 0.0, options: .curveLinear,
244                     animations:
245                     {
246                         self.confirmationImgView.alpha = 1.0
247                     })
248                 timer = Timer.scheduledTimer(timeInterval: 1.0, target: self, selector:
249                     #selector(hide), userInfo: nil, repeats: false)
250             }
251             //temperaturLabel.text = dataAsString + " °C";
252         }
253         else
254         {
255             print("ERROR: converting data to string.")
256         }
257     }
258 }
  
```

<x-coredata://VAZV0AUV-EE/U-MUUI-WA/B-30ZEE3F14038/Befehl/p2> ; data:
 <fault>, <BefehlM0: 0x2B27ce170> (entity: Befehl; id:
 0xe888f040014813f2
 <x-coredata://DA296AD9-0E7D-4DD1-A47B-362EE3F14038/Befehl/p1> ; data:
 <fault>]]
 Optional("Sag Hallo")
 Peripheral name: arduino_HM10
 UUID: D0EB1BEA-BE4A-8D71-9A2F-E9949482B7C0
 RSSI: -76
 2019-05-12 19:06:49.251635+0200 BLL_1.3[6931:1448571] [UIWorkIntervalTiming]
 workIntervalStart: startTimestamp > targetTimestamp; rolling forward by
 1.083333
 (lldb)

> A message from "Sag Hallo"
 > A self message from ViewController:0x0000000105304670
 > I write to "confirmatio" (CBCharacteristic) 0x00000002020c9000
 > I data = (class NSString)