

BLUETOOTH Characteristics 2

Goals of this challenge

- Flashing the firmware
- Solving the challenge

Challenge Description: A confidential message is stored on the firmware but protected by a password. The goal is to provide a screenshot of the confidential message.

Flashing the firmware

- Connect the **DVID board** to the **computer** using the **USB AVR Programmer**, and start the script
- When the AVR programming has completed, the **board should restart** and you should the following
- Flash.sh

```
#!/bin/bash

if [[ ! -d "./DVID/" ]]; then
    git clone https://github.com/vulcainreo/DVID
fi

pushd ./DVID/trainings/bluetooth/characteristics2/
avrdude -F -v -p atmega328p -P /dev/ttyUSB0 -c
usbasp -u -U
flash:w:characteristics2.ino.with_bootloader.arduino_standard.hex
popd
```

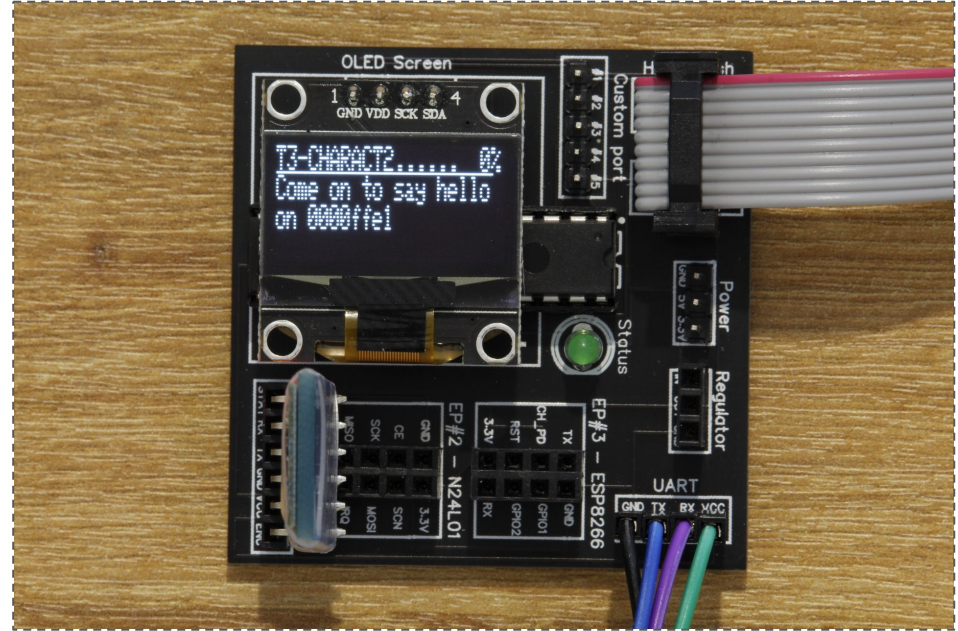
Note:

Avrdude for windows

<https://github.com/mariusgreuel/avrdude/releases/download/v7.0-windows/avrdude-v7.0-windows-windows-arm64.zip>

Flashing the firmware

- Now, we will connect the DVID board to our computer using the USB AVR programmer, and start the script. When the AVR programming has completed, the board should restart and you should see the following



Note:

Avrdude for windows

<https://github.com/mariusgreuel/avrdude/releases/download/v7.0-windows/avrdude-v7.0-windows-windows-arm64.zip>

Solving the challenge

- In this challenge, the IoT device sends informations using bluetooth characteristics on `0000ffe1`
- Firstly, we need to do a **bluetooth scan** to find the device
- We can do this inside **bluetoothctl** using **scan on** command. (and **scan off** to stop)

```
[DVID:characteristic]
[DVID:characteristic]$ bluetoothctl
Agent registered
[CHG] Controller B8:76:3F:AE:38:78 Pairable: yes
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:76:3F:AE:38:78 Discovering: yes
[NEW] Device 00:13:AA:00:25:93 dvid-807
[bluetooth]# scan off
Discovery stopped
[CHG] Controller B8:76:3F:AE:38:78 Discovering: no
[CHG] Device 00:13:AA:00:25:93 TxPower is nil
[CHG] Device 00:13:AA:00:25:93 RSSI is nil
[bluetooth]#
```

Solving the challenge

- Now that we found the IoT device, we can connect to it by typing **connect** followed by the device's bluetooth address
- To use **bluetooth characteristics**, we need to use a specific menu, called **gatt**

```
[DVID:characteristic]
[bluetooth]# connect 00:13:AA:00:25:93
Attempting to connect to 00:13:AA:00:25:93
[CHG] Device 00:13:AA:00:25:93 Connected: yes
Connection successful
[NEW] Primary Service (Handle 0x22a1)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service000c
00001801-0000-1000-8000-00805f9b34fb
Generic Attribute Profile
[NEW] Characteristic (Handle 0x22a1)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service000c/char000d
00002a05-0000-1000-8000-00805f9b34fb
Service Changed
[NEW] Descriptor (Handle 0x2134)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service000c/char000d/desc000f
00002902-0000-1000-8000-00805f9b34fb
Client Characteristic Configuration
[NEW] Primary Service (Handle 0x22a1)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0010
0000180a-0000-1000-8000-00805f9b34fb
Device Information
[NEW] Characteristic (Handle 0x22a1)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0010/char0011
00002a23-0000-1000-8000-00805f9b34fb
System ID
[NEW] Characteristic (Handle 0x22a1)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0010/char0013
00002a24-0000-1000-8000-00805f9b34fb
Model Number String
[NEW] Characteristic (Handle 0x22a1)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0010/char0015
00002a25-0000-1000-8000-00805f9b34fb
Serial Number String
```

Solving the challenge

- To access it in **bluetoothctl**, type **menu gatt** to enter the gatt menu and **back** to exit the menu
- In the gatt menu we have access to many advanced features, such as listing attributes of a device, reading and writing data to bluetooth services

```
[dvid-807]# menu gatt
Menu gatt:
Available commands:
-----
list-attributes [dev/local]           List attributes
select-attribute <attribute/UUID>    Select attribute
attribute-info [attribute/UUID]      Select attribute
read [offset]                        Read attribute value
write <data=xx xx ...> [offset] [type] Write attribute value
acquire-write                         Acquire Write file descriptor
release-write                        Release Write file descriptor
acquire-notify                       Acquire Notify file descriptor
release-notify                       Release Notify file descriptor
notify <on/off>                      Notify attribute value
clone [dev/attribute/UUID]           Clone a device or attribute
register-application [UUID ...]       Register profile to connect
unregister-application               Unregister profile
register-service <UUID> [handle]      Register application service.
unregister-service <UUID/object>     Unregister application service
register-includes <UUID> [handle]     Register as Included service in.
unregister-includes <Service-UUID><Inc-UUID> Unregister Included service.
register-characteristic <UUID> <Flags=read,write,notify...> [handle] Register application characteristic
unregister-characteristic <UUID/object> Unregister application characteristic
register-descriptor <UUID> <Flags=read,write...> [handle] Register application descriptor
unregister-descriptor <UUID/object>  Unregister application descriptor
back                                 Return to main menu
version                             Display version
quit                                 Quit program
exit                                 Quit program
help                                Display help about this program
export                              Print environment variables
[dvid-807]#
```

Solving the challenge

- We will now list attributes offered by the bluetooth device using **list-attributes** command

```
[dvid:characteristic2]
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0010/char001b
00002a28-0000-1000-8000-00805f9b34fb
Software Revision String
Characteristic (Handle 0xdea4)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0010/char001d
00002a29-0000-1000-8000-00805f9b34fb
Manufacturer Name String
Characteristic (Handle 0xdea4)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0010/char001f
00002a2a-0000-1000-8000-00805f9b34fb
IEEE 11073-20601 Regulatory Cert. Data List
Characteristic (Handle 0xdea4)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0010/char0021
00002a50-0000-1000-8000-00805f9b34fb
PnP ID
Primary Service (Handle 0x6920)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0023
0000ffe0-0000-1000-8000-00805f9b34fb
Unknown
Characteristic (Handle 0x4314)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0023/char0024
0000ffe1-0000-1000-8000-00805f9b34fb
Unknown
Descriptor (Handle 0x0015)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0023/char0024/desc0026
00002902-0000-1000-8000-00805f9b34fb
Client Characteristic Configuration
Descriptor (Handle 0x0015)
/org/bluez/hci0/dev_00_13_AA_00_25_93/service0023/char0024/desc0027
00002901-0000-1000-8000-00805f9b34fb
Characteristic User Description
[dvid-807]#
```


Solving the challenge

- Looking at the screen of the **DVID device**, we can see a message "**Something is leaking on 0000ffe1**", therefore we will try to interact with the characteristic **0000ffe1-0000-1000-8000-00805f9b34fb**. To do this, we need to select the attribute using **select-attribute <uuid>** command

```
[DVID:characteristic]
[dvid-807]# select-attribute 0000ffe1-0000-1000-8000-00805f9b34fb
[dvid-807:/service0023/char0024]#
```

- Then we need to **acquire-write** to connect to the stream and be able to read the values from the **service/characteristic**. We can then type read to read the stream of values coming from the **service/characteristic**.

Solving the challenge

```
[DVID:characteristic2]
[dvid-807:/service0023/char0024]# acquire-write
[CHG] Attribute /org/bluez/hci0/dev_00_13_AA_00_25_93/service0023/char0024 WriteAcquired: yes
AcquireWrite success: fd 7 MTU 23
[dvid-807:/service0023/char0024]#
```

- Now we can use the **write** command to **send data** to the **service/characteristic**. The syntax is **write "0xaa 0xbb 0xcc ..."** with the **hex values of the data you want to send**. For example **hello** is **"0x68 0x65 0x6c 0x6c 0x6f"**. Therefore, the final command is **write "0x68 0x65 0x6c 0x6c 0x6f"**

```
[DVID:characteristic2]
[dvid-807:/service0023/char0024]# write "0x68 0x65 0x6c 0x6c 0x6f"
Attempting to write fd 7
[dvid-807:/service0023/char0024]#
```

Flashing the firmware

- We can now see the flag on the OLED screen

