Parrot®

Bluetooth Low Energy Interface Specification

Table of contents

1 General Description 3

2 Connectivity 4

2.1 FlowerPower device Scan 4
2.2 FlowerPower devices identification 4
2.2.1 Identify that a device is an FlowerPower device 4
2.2.2 Retrieve the FlowerPower device system ID 4
2.3 Connect to the desired device 5
2.4 Disconnecting from a device 5

3 Control and monitor the FlowerPower devices 6

3.1 Identify a FlowerPower device 6
3.2 Retrieve Battery status 6
3.3 Retrieve current time 7
3.4 Control and retrieve live measurements 7
3.5 Control the LED of the FlowerPower device 8
3.6 Retrieve the last detected move time 8

4 Retrieve the history file 9

4.1 Retrieve the history file information trough the History Service 9
4.2 Retrieve the history file content through the Upload Service 9
4.3 History retrieving procedure 17

5 Background connections process 21

6 Last Move date retrieving procedure 22

Appendix A - FlowerPower Application GATT services description 24

Appendix B - Advertising frames description 28

1 General Description

This document specifies the messages exchanged between the FlowerPower device and the smartphone or table application trough Bluetooth Low Energy (BLE).

We will call the smartphone or tablet application the client application in this document.

Bluetooth Low energy main functions

Connectivity

- Scanning of the BLE devices available.
- Recognizing FlowerPower BLE devices.
- Establish a link with a FlowerPower BLE device.
- Handle the disconnections started by the FlowerPower device.

Monitor live data

- Retrieve Battery status.
- Retrieve current device time.
- Configure and retrieve the FlowerPower live measurements.
- Retrieve the last detected move time.

Control the FlowerPower LED

- Switch on/off the LED of the device at user's request.

Retrieve a measurement history file

- Retrieve information about the history stored into the FlowerPower device.
- Receive and store the file containing the measure history.

Retrieve the date of the last time the device was moved

- The device has the ability to detect when it has been moved.

2 Connectivity

The FlowerPower device communicates with the client application using BLE.

It will act as a peripheral and uses GAP and GATT profiles. The client application should configure its BLE stack as central.

2.1 FlowerPower device Scan

When not connected, the FlowerPower device is in advertising mode. The configured connection interval is 2 second, which means that an advertising frame will be sent every 2 seconds on each of the 3 advertisement channels.

The client application should scan for all BLE devices.

2.2 FlowerPower devices identification

2.2.1 Identify that a device is an FlowerPower device

- The client application must identify a BLE device as a FlowerPower device when:
  - It declares to support the live service in the advertisement data (39e1**FA00**-84a8-11e2-afba-0002a5d5c51b is included in the list of available service UUID).
- The scan response data is obtained by performing active scanning.

2.2.2 Retrieve the FlowerPower device system ID

A Flower Power is identified by its **System ID**. This is a unique identifier assigned to Flower Power devices. It can be retrieved in two ways:

- By building it from the device Bluetooth address (preferred method).
- By building it from the advertisement frame information (for Firmware version 1.1 or higher).
- By reading the “system ID” characteristic which is part of the device information service (0x180A).

  ⚠ This method shall only be used when the current Bluetooth implementation does not give access to the device Bluetooth address.

The system ID is an 8 bytes buffer. When retrieved, it should be converted into a string representing the system ID encoded in hexadecimal format.

The following table describes the correspondence between the Bluetooth Mac address and the Flower Power system ID.

Field	Byte correspondence	Example
Mac address IEEE 802 format	b0:b1:b2:b4:b4:b5	90:03:b7:c7:34:e9
Mac address hexadecimal format	0xb0b1b2b3b4b5	0x9003b7c734e9
system ID desired format	"b0b1b20000b3b4b5"	"9003b70000c734e9"

2.3 Connect to the desired device

When one or several devices have been identified as FlowerPower devices, the application could establish a connection to one of them.

There are 3 reasons for the application to connect to the FlowerPower device:

- The "move detected" flag is set on the scan response data (see Appendix B for scan response description).
- The "unread entries" flag is set on the scan response data.
- The user starts a live session on that device.

2.4 Disconnecting from a device

For now the Apple devices stay connected to the devices event when the application requested a disconnection. In order to limit connection time, the FlowerPower device may disconnect from the application after a certain amount of time without incoming BLE request.

It has been decided (arbitrary) to set this time to 1 second, but we may change this timeout value if needed.

3 Control and monitor the FlowerPower devices

3.1 Identify a FlowerPower device

A FlowerPower sensor is identified by its **System ID**. This is a unique identifier assigned to FlowerPower devices. It can be retrieved in two ways:

- By building it from the device Bluetooth address (preferred method).
- By reading the "system ID" field of the device information BLE service.
  ⚠ This method shall only be used when the current Bluetooth implementation does not give access to the device Bluetooth address.

The system ID is an 8 bytes buffer. When retrieved, it should be converted into a string representing the system ID encoded in hexadecimal format.

The following table describes the correspondence between the Bluetooth Mac address and the Flower Power system ID.

Field	Byte correspondence	Example
Mac address IEEE 802 format	b0:b1:b2:b4:b4:b5	90:03:b7:c7:34:e9
Mac address hexadecimal format	0xb0b1b2b3b4b5	0x9003b7c734e9
system ID desired format	"b0b1b20000b3b4b5"	"9003b70000c734e9"

3.2 Retrieve Battery status

The FlowerPower device includes the Battery service UUID as specified by the Bluetooth specification:

http://developer.bluetooth.org/gatt/services/Pages/ServiceViewer.aspx?u=org.bluetooth.service.battery_service.xml

The battery service implemented in the FlowerPower device allows to retrieve the current battery status (in %), and has the ability to send a notification when the battery level is critical.

- When connecting to the device, the application reads the battery status (the characteristic with UUID 0x2A19).
- When connecting to the device, the application registers to notifications for the battery status notifications.
- When a notification is received for the battery status from the FlowerPower device, the client application should report it to the user.

3.3 Retrieve current time

The device communicates its current time value using the custom FlowerPower time profile (UUID 0xFD00)
The date (UUID 0xFD01) is a 32 bits unsigned integer coded in little endian.
The retrieving of the current time allows to determine the device startup time, and so to convert all the time information given by the device into UTC time.

3.4 Control and retrieve live measurements

Live measurements shall be controlled by the client application trough the live profile.
Basically, the application has to register to the notifications for the sensor values attributes and configure the "live measure period" value to the desired measure period (when set to 0, no live measure is taken).

- When the connection has been established, the application must:
  - Register to notifications for the Light characteristic
  - Register to notifications for the Soil EC characteristic
  - Register to notifications for Soil Temp characteristic
  - Register to notifications for the Air temp characteristic
  - Register to notifications for the Soil VWC characteristic
- When the client application has to activate the live measurements, the client application must :
  - Write the value 1 into the characteristic with UUID 0xFA06 (in order to receive notifications every 1 sec).
- When the client application has to de-activate the live measurements, the client application must:
  - Write the value 0 into the measure period characteristic.
- When a notification is received for one of the characteristics previously written, the application must convert the received value to a value to display, using the formulas described in the following table:

UUID	Description	Value to display
0xFA01	Light	characteristic_value
0xFA02	Soil EC	(characteristic_value * 3.3) / (2^{11}-1)
0xFA03	Soil Temp	(characteristic_value * 3.3) / (2^{11}-1)
0xFA04	Air Temp	(characteristic_value * 3.3) / (2^{11}-1)
0xFA05	Soil VWC	(characteristic_value * 3.3) / (2^{11}-1)

The live measures must be deactivated when the application is not displaying live values.

3.5 Control the LED of the FlowerPower device

The LED of the FlowerPower device is controlled by the characteristic which UUID is 0xFA07:

- Writing 1 on the LED state (0xFA07) characteristic value turns the LED on.
- Writing 0 on the LED state (0xFA07) characteristic value turns the LED off.

Note that the LED is automatically turned off when the device is disconnected.

The LED is turned on when the user requests it to the application. The led may not stay turned on for a long time since it consumes power.

3.6 Retrieve the last detected move time

The last detected move time can be retrieved using the Live service (UUID 0xFA08).

The “move detected” flag is set on the BLE scan response information when a move has been detected and not been read by the application. The application should be able to see that that flag has been set and start the following procedure:

1. Connect to the device.
2. Read the value of the last move date.
3. Convert this value into a UTC time value (adding it the device startup time in UTC format)
4. Communicate this value to the web server.

Note that the move detect flag is automatically unset when the *last move date* value is read by the application.

4 Retrieve the history file

Retrieving the history file uses 2 services:

5. the history service
6. the upload service

4.1 Retrieve the history file information trough the History Service

The FlowerPower device includes a custom service named “History”. This service allows knowing the current status of the history recording process of the FlowerPower device and is used to configure the upload process.

This history service provides the following characteristics:

UUID	Description	Type	Access
0xFC01	Nb entries	Little endian U16	Read
0xFC02	Last entry Index	Little endian U32	Read
0xFC03	Transfer Start Index	Little endian U32	Read/Write

The client application can specify the “*transfer start index*” value and write any value in the range [0-65535].

Note that the device will not always strictly starts the history file using the transfer start index value, but can start before that value (it will not start later that value).

4.2 Retrieve the history file content through the Upload Service

To retrieve the history file content, the client application should follow the upload protocol.

The upload protocol allows the application to retrieve a file of a random size. It is an upper layer of the GATT profile and uses the custom “upload” profile (UUID 0xFB00) which characteristic table is the following:

UUID	Description	Type	Access	Comment
0xFB01	Tx buffer	20 bytes buffer	Notify	Send raw data threw notifications
0xFB02	Tx Status	U8	Read/Notify	Adverts the central of status changes
0xFB03	Rx Status	U8	Read/Write	Update application status

Basically, the central (application) and the peripheral (FlowerPower) synchronize themselves using the Tx status and Rx status characteristics, and the peripheral sends the file data threw the Tx buffer characteristic. Periodically during the transfer, a handshake procedure is done in order to ensure that the last data sent have been correctly received by the application.

Transmitter behavior

The Tx status describes the current status of the transmitter (FlowerPower device) and can take the following values:

0 IDLE (initial state)
1 Transferring
2 Waiting Ack

The FlowerPower device uses this characteristic in order to advice the receiver (Application) of its current status. The application should enable notifications on this field in order to be advices of status changes.
The following picture shows the state machine of the Transmitter (FlowerPower device):

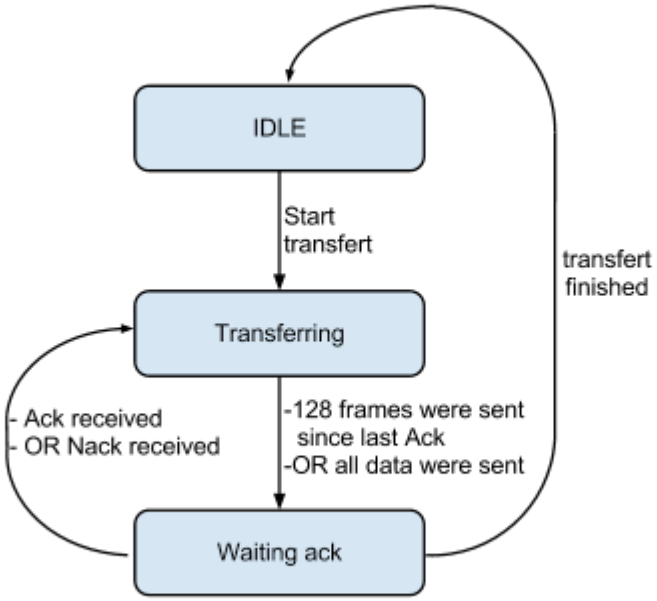


Figure 1: Transmitter state machine

When the transmitter is on *transferring* state, it sends data threw notifications on the *Tx buffer* characteristic. When 128 *Tx buffer* notifications has been sent, the transmitter goes into *waiting ack* status and wait for an acknowledgement of the receptor before sending the rest of the file. When all data has been transmitted, the transmitter goes into *waiting ack* status.

Receiver behavior

The application writes values in the Rx status to indicate its current status (when changed). The possible values for the Rx status field are the following:

0 Standby (initial state)
1 Receiving
2 Ack
3 Nack
4 Cancel
5 Error

The following picture shows the state machine of the Receptor (Application)

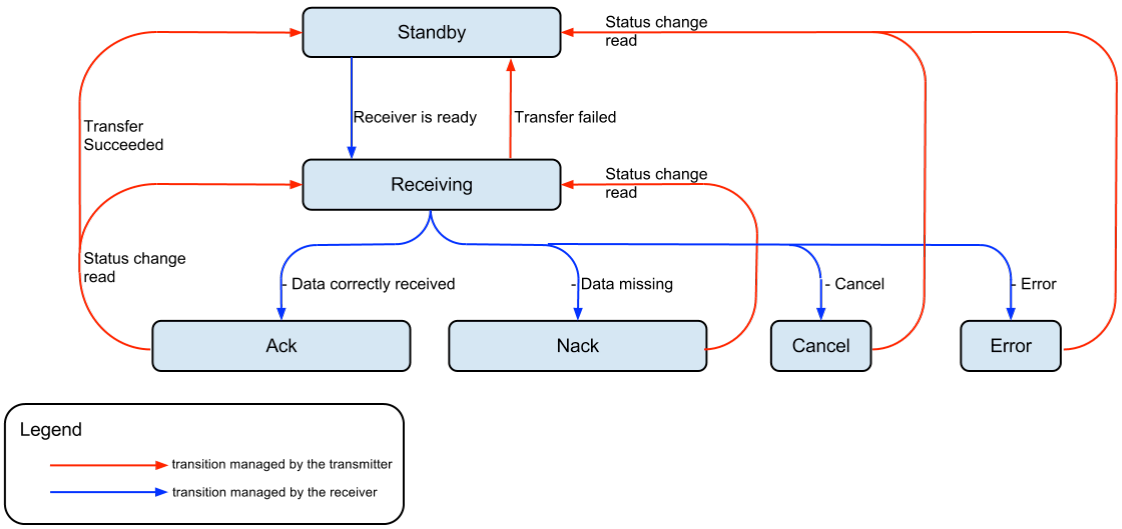


Figure 2: Receptor state machine

The first data buffer transferred contains a header indicating the size of the file which will be sent. This allows the receiver to determine when the transfer finished and manage the last acknowledge procedure.

First Tx buffer frame notification description

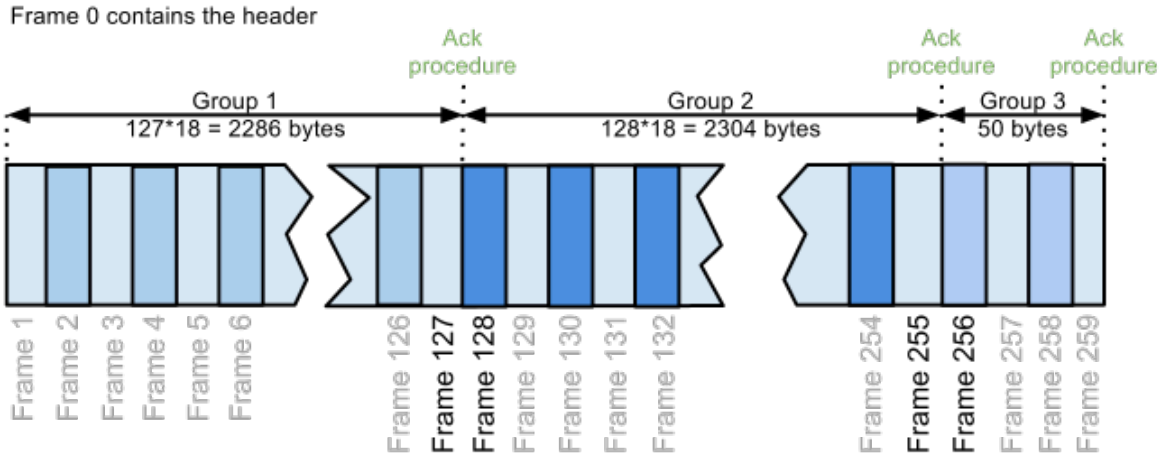
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Bytes [6 :20]
Frame index Big endian		**Header**				
		File length Big endian				reserved
Index L	Index H	File size 1	File size 2	File size 3	File Size 4	

Note that the *frame index* of the first Tx buffer frame is always 0.

Others Tx buffer frame notification description

Byte 0	Byte 1	Bytes [2:20]
Frame index big endian		**Payload Data**
Index L	Index H	

Example of 4640 byte sized file splitting for sending



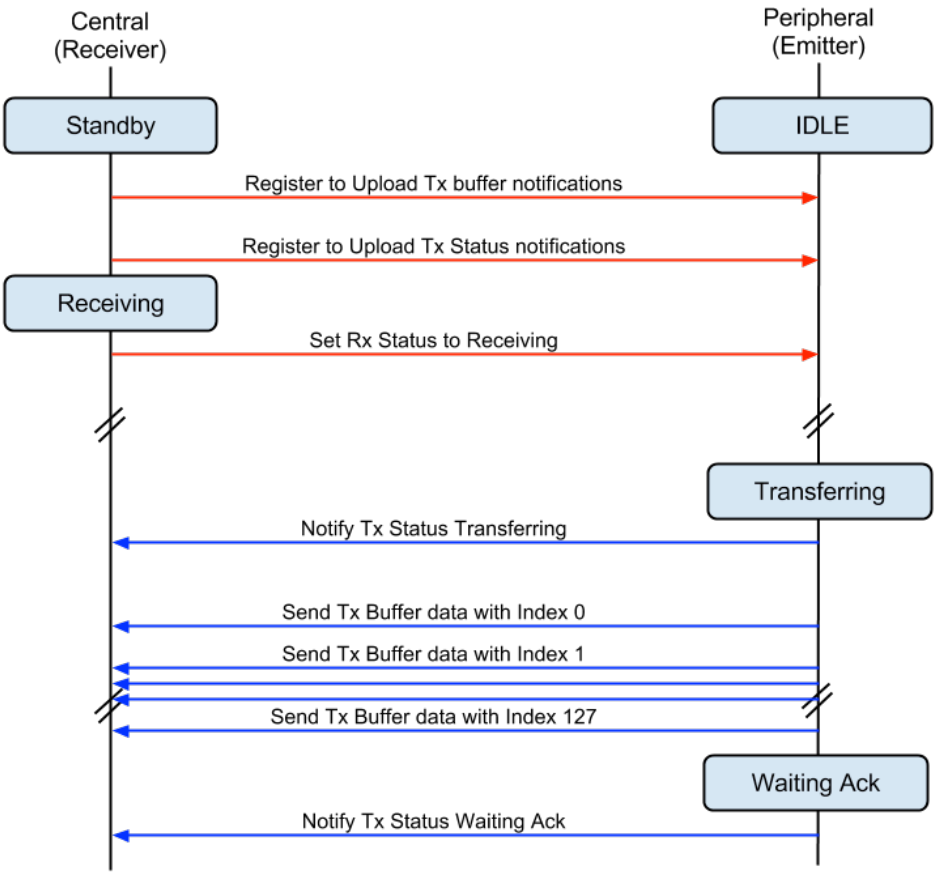
- The frame index is incremented at each notification.
- The frame index following 35535 is 0.
- Note that the application must handle correctly the following cases:
    - A Tx buffer frame can be sent twice.
    - The Tx buffer frames can be sent in the wrong order (a frame won't move more than 4 positions) in a group of frames.

For example the following sequences of frame index could happen:

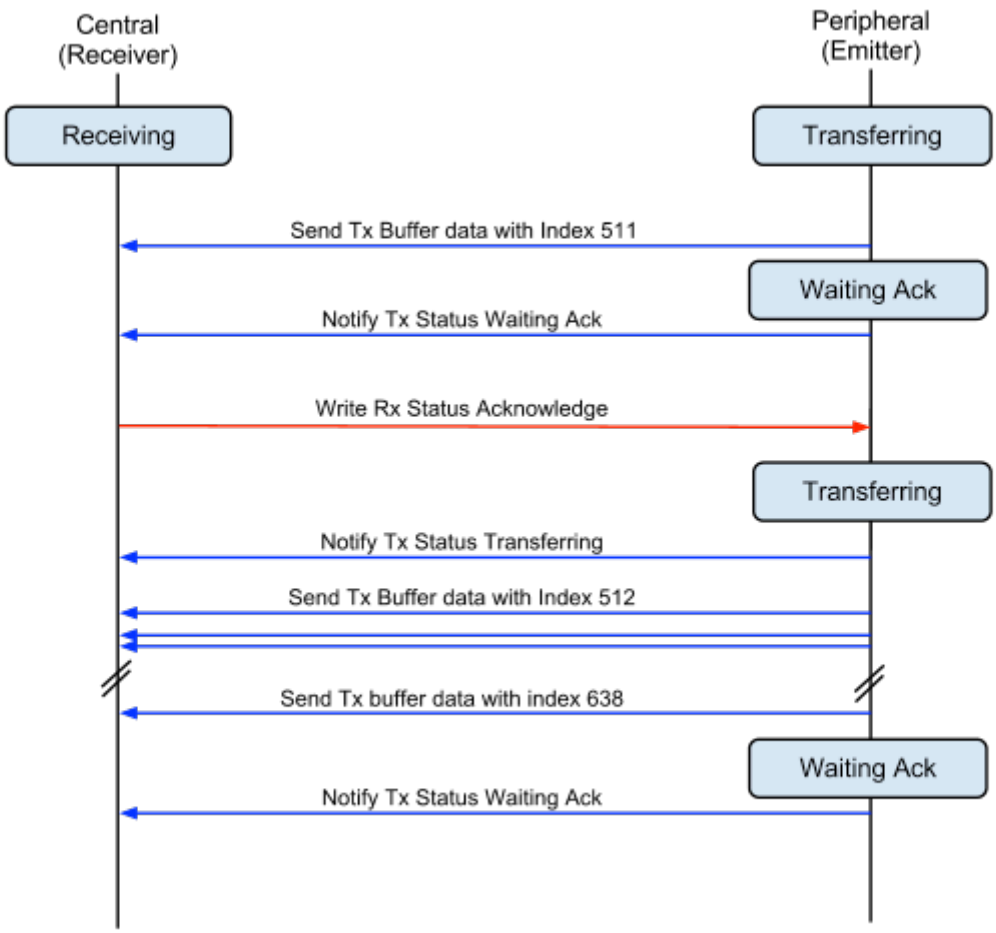
- [1 - 3 - 2 - 4]
- [1 - 4 - 2 - 3]
- [1 - 1 - 2 - 3]
- [2 - 3 - 1 - 4]

Protocol description

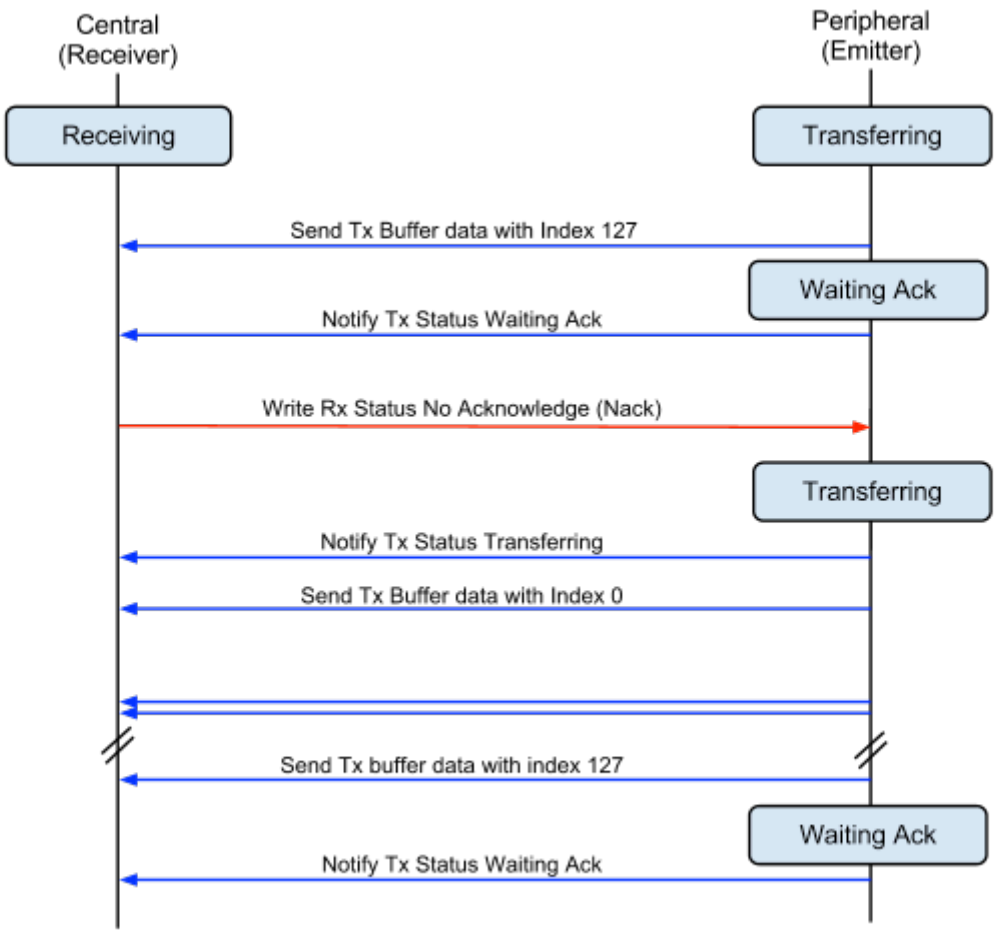
Transfer starting procedure



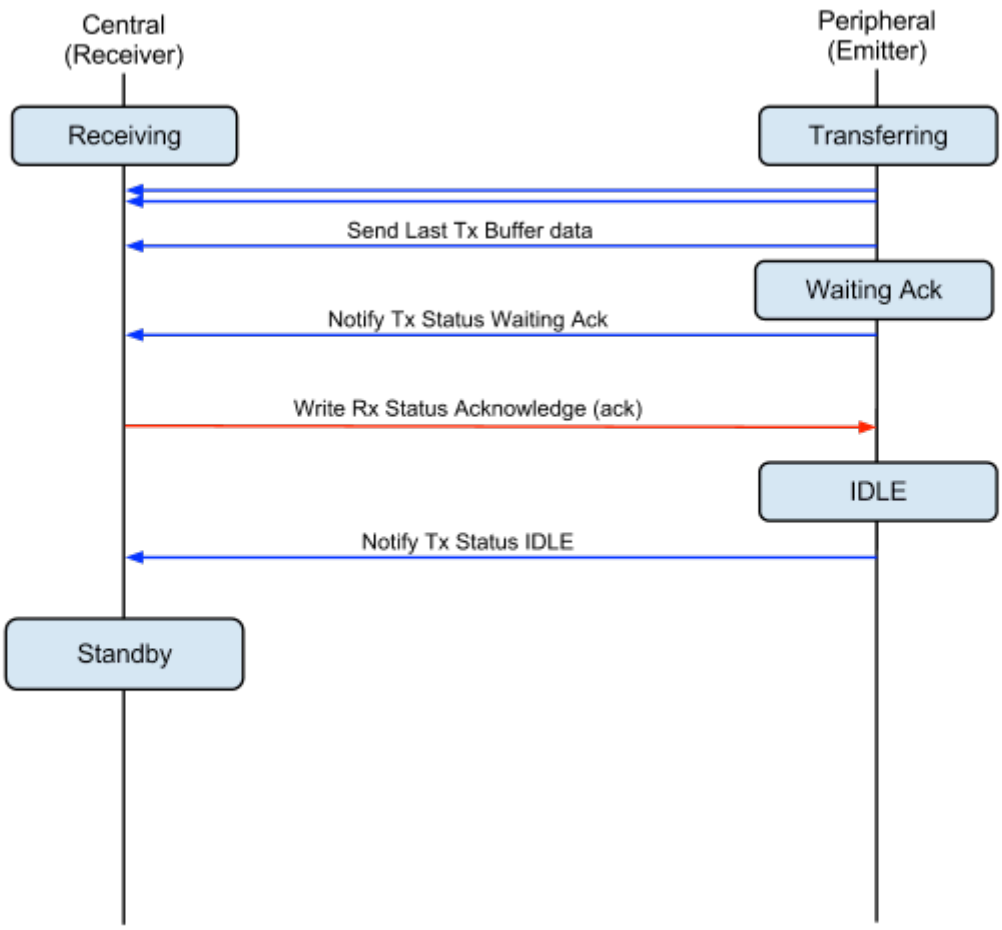
Acknowledge procedure for end of frame group 4.



Not acknowledge procedure example at the end of the first frame group



Transfer ending procedure



Requirements:

- When the connection is established, the client application must register to notifications for “Tx buffer”
- When the connection is established, the client application must register to notifications for “Tx status”
- After 3 failures when trying to receive a frame group, the client application must write the *error* state into *Rx status* and the transfer will be stopped.
- If the receiver has been waiting for a frame (except the first *Tx status transferring* state of a transfer) for more than 1 second, the client application must write the *error* state into *Tx status* and the transfer will be stopped.

4.3 History retrieving procedure

The history management allows client application to indicate to the FlowerPower device the beginning of the history file upload using the *Transfer Start Index*. The value to set in the transfer start index

The upload profile manages the upload Process.

1. The client application initiates the transfer procedure (by setting *Rx status* to Ready).

2. The client application needs to manages the reception of the history file including:
    - Receiving and order the data frames.
    - Managing the acknowledgement (or not acknowledgement) process.
    - Storing the received file.

The procedure for the history upload is the following:

Initial state

Device is disconnected from the device. If the 'unread entries' flag is set, the procedure is started automatically by the application.

Procedure

1. Connect to the device
2. Retrieve information to send to the server
    a. Retrieve FlowerPower clock (using the FlowerPower clock service). Compute the device startup time using the following formula:

startupTime = currentTime –

    currentTime Current number of seconds since the 1st Jan 1970 00:00 (UTC time)
    FlowerPowerClock Value retrieved from the device using BLE

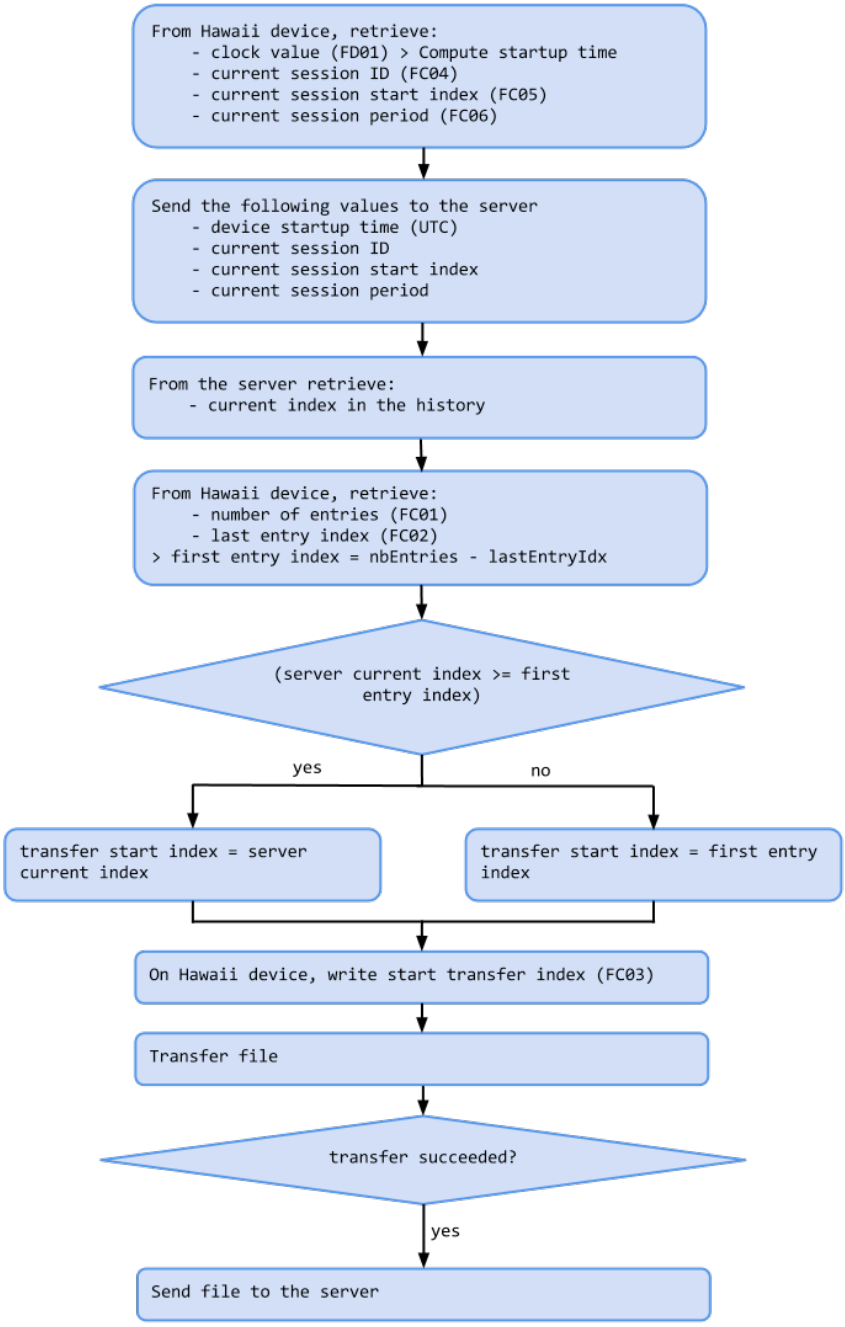
    b. Retrieve the current Session ID from the FlowerPower device using the BLE history profile.
    c. Retrieve the session measurement period from the FlowerPower device using the history BLE profile.
    d. Retrieve the session start index from the FlowerPower device using the BLE history profile.
3. Send that information (device startup time; session ID; session Start Index) to the server.
4. Retrieve *current index in the history* from the server. This value will be used (if possible) to set the value of start transfer index in the history profile.
5. Using the history profile, compute the first available index in the history.
    a. Retrieve *last entry index* from the FlowerPower device using history BLE profile.
    b. Retrieve the number of entries available in the history (*nb entries*) from the FlowerPower device using the History BLE profile.
    c. The first available entry (minimum possible value for *start transfer index*) can be computed using the following formula:

First entry index = Last entry index – nb entries + 1

6. Write the *transfer start index* value in the history profile.
    a. If the *current index in the history* value given by the server is upper or equal to the first entry Index, use this value to write in the *transfer start index* BLE characteristic.
    b. Otherwise, set the *transfer start index* to the *first entry index*.
7. Start the transfer procedure.
8. If transfer succeeded, send the raw data received to the server.

This procedure is summed-up in the following picture.

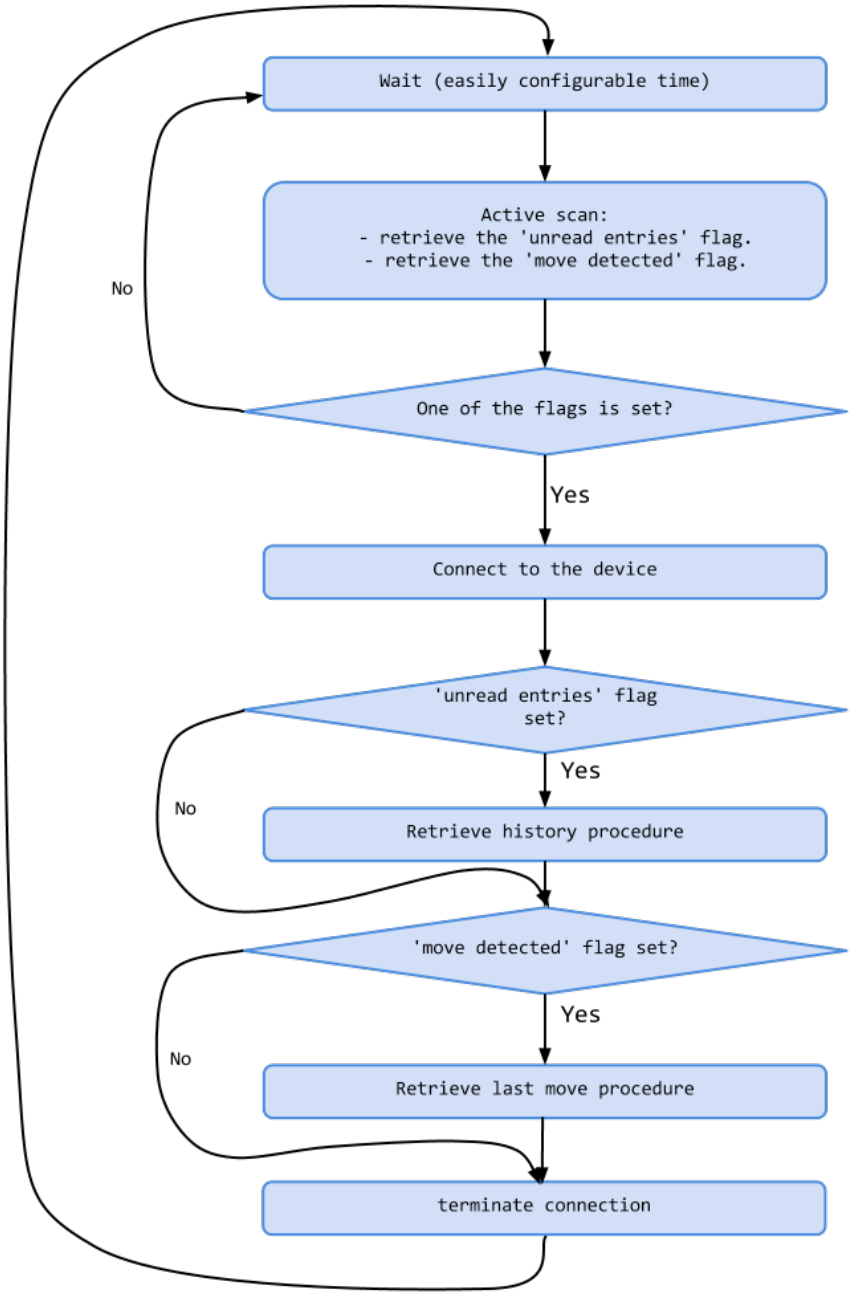
History retrieving procedure



If any failure happens during the procedure, the application should wait during 60 seconds and try again if the device is still connectable.

5 Background connections process

In order to save FlowerPower batteries, the application must connect to the device only when requested. Connections to the device will be automatically started by the application according to this scheme.



6 Last Move date retrieving procedure

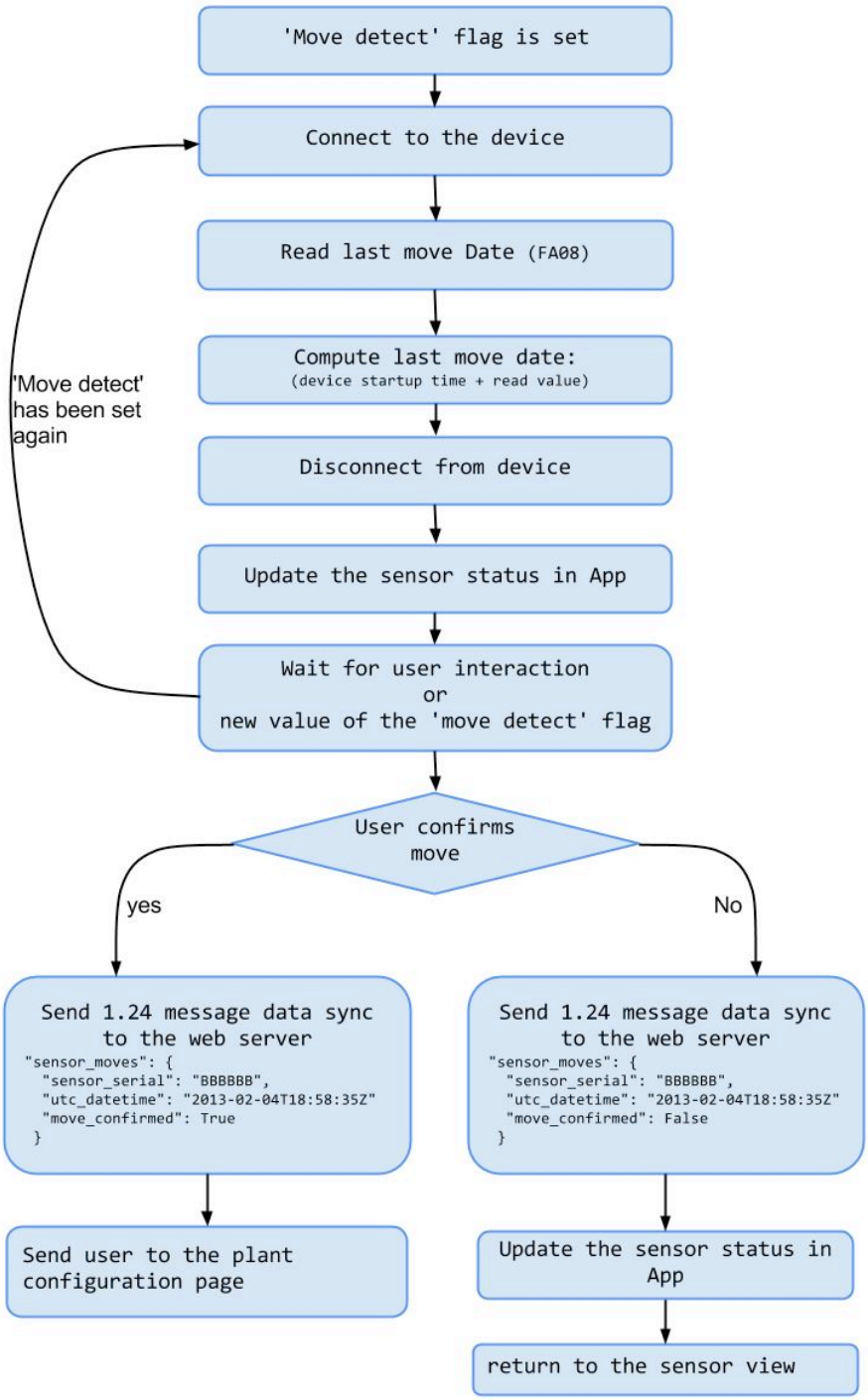
The move detection feature has been designed to determine automatically when a sensor has been moved from one plant to another one.

When the sensor detects that it has been moved while it was in the air:

- It sets the ‘move detected’ flag of the advertisement frame to 1
- It updates the ‘last move date’ characteristic (UUID FA08).

The ‘move detected’ flag is reset when the value of the ‘last move date’ is read threw BLE by the central device.

Last Move date retrieving procedure



Appendix A - FlowerPower Application GATT services description

This chapter describes the GATT services specific to the FlowerPower device:

GAP service (UUID 0x1800)

UUID	Description	Type	Access
0x2A00	Name	UTF8 String	Read/Write
0x2A01	Appearance	Little endian U16	Read

Device Information (UUID 0x180A)

http://developer.bluetooth.org/gatt/services/Pages/ServiceViewer.aspx?u=org.bluetooth.service.device_information.xml

UUID	Description	Type	Access
0x2A23	System ID	8 bytes frame	Read
0x2A26	Firmware revision	UTF8 String	Read
0x2A25	Serial Number	UTF8 String	Read
0x2A27	Hardware revision (Bootloader version)	UTF8 String	Read

Live service (39e1FA00-84a8-11e2-afba-0002a5d5c51b)

UUID	Description	Type	Access
39e1**FA01**-84a8-11e2-afba-0002a5d5c51b	Light sensor Value	Little endian U16	Read/notif
39e1**FA02**-84a8-11e2-afba-0002a5d5c51b	Soil EC	Little endian U16	Read/notif
39e1**FA03**-84a8-11e2-afba-0002a5d5c51b	Soil Temperature	Little endian U16	Read/notif
39e1**FA04**-84a8-11e2-afba-0002a5d5c51b	Air temperature	Little endian U16	Read/notif
39e1**FA05**-84a8-11e2-afba-0002a5d5c51b	Soil % VWC	Little endian U16	Read/notif
39e1**FA06**-84a8-11e2-afba-0002a5d5c51b	Live measure Period	U8	Read/Write
39e1**FA07**-84a8-11e2-afba-0002a5d5c51b	Led state	U8 (value is 0 or 1)	Read/Write
39e1**FA08**-84a8-11e2-afba-0002a5d5c51b	Last move date	U32	Read/notif
Firmware version >= 1.1.0			
39e1**FA09**-84a8-11e2-afba-0002a5d5c51b	Calibrated VWC	float 32	Read/notif
39e1**FA0A**-84a8-11e2-afba-0002a5d5c51b	Calibrated air temperature	float 32	Read/notif
39e1**FA0B**-84a8-11e2-afba-0002a5d5c51b	calibrated DLI	float 32	Read/notif
39e1**FA0C**-84a8-11e2-afba-0002a5d5c51b	calibrated Ea	float 32	Read/notif
39e1**FA0D**-84a8-11e2-afba-0002a5d5c51b	calibrated Ecb	float 32	Read/notif
39e1**FA0E**-84a8-11e2-afba-0002a5d5c51b	calibrated Ec porous	float 32	Read/notif

Upload service (UUID 39e1**FB00**-84a8-11e2-afba-0002a5d5c51b**)**

UUID	Description	Type	Access
39e1**FB01**-84a8-11e2-afba-0002a5d5c51b	Tx buffer	20 bytes of raw data	Notify
39e1**FB02**-84a8-11e2-afba-0002a5d5c51b	Tx Status	U8	Read/Notify
39e1**FB03**-84a8-11e2-afba-0002a5d5c51b	Rx Status	U8	Read/Write

History service (UUID 39e1**FC00**-84a8-11e2-afba-0002a5d5c51b**)**

UUID	Description	Type	Access
39e1**FC01**-84a8-11e2-afba-0002a5d5c51b	Nb entries	Little endian U16	Read
39e1**FC02**-84a8-11e2-afba-0002a5d5c51b	Last entry Index	Little endian U32	Read
39e1**FC03**-84a8-11e2-afba-0002a5d5c51b	Transfer Start Index	Little endian U32	Read/Write
39e1**FC04**-84a8-11e2-afba-0002a5d5c51b	Current Session ID	Little endian U16	Read
39e1**FC05**-84a8-11e2-afba-0002a5d5c51b	Current Session Start Index	Little endian U32	Read
39e1**FC06**-84a8-11e2-afba-0002a5d5c51b	Current Session Period	Little endian U16	Read

FlowerPower clock service (UUID 39e1FD00-84a8-11e2-afba-0002a5d5c51b)

UUID	Description	Type	Access
39e1**FD01**-84a8-11e2-afba-0002a5d5c51b	FlowerPower current time (s)	Little endian U32	Read

FlowerPower calibration service (UUID 39e1FE00-84a8-11e2-afba-0002a5d5c51b)

UUID	Description	Type	Access
39e1**FE01**-84a8-11e2-	Calibration data	Array of little endian U16	Read

afba-0002a5d5c51b			
39e1**FE02**-84a8-11e2-afba-0002a5d5c51b	Force bond characteristic read into this make iOs devices start a pairing procedure	Dummy Byte	Read
39e1**FE03**-84a8-11e2-afba-0002a5d5c51b	Name	UTF8 String	Read/Write
39e1**FE04**-84a8-11e2-afba-0002a5d5c51b	Color	Little endian U16 **Values:** **1:** brown (Pantone 7582C) **2:** esmerald (Pantone 3272C) **3:** lemon (Pantone 389C) **4:** gray-brown (Pantone 7532C) **5:** gray-green (Pantone 5545C) **6:** classic-green (Pantone 363C) **7:** gray-blue (Pantone 549C)	Read

Over The Air Download Service (UUID F000**FFC0-**0451-4000-B000-000000000000**)**

UUID	Description	Type	Access
F000**FFC1-**0451-4000-B000-000000000000	OAD Image notify	8 bytes of raw data	Read/Write
F000**FFC2-**0451-4000-B000-000000000000	OAD Image block request/response	18 bytes of raw data	Read/write

Appendix B - Advertising frames description

FlowerPower advertisement data for Firmware versions < 1.1

Discovery mode			**List of available service UUID**			**Manufacturer specific**
Data type: 0x01 Value: - Discoverable mode - BRE/DR not supported			Data type: 0x06 () Live service UUID: FlowerPowerUuid(0xFA00)			data type: 0xFF 1 byte: Data to transmit flags
0x02	0x01	0x06	0x33	0x06	1bc5d5a50200baafe211a884**00FA**e139	Flags

Flags description

Flags		
Byte	**Name**	**Description**
Bit 0	Unread entries	One or several entries(s) have been added to the history since last connection.
Bit 1	Move Detected	The device has been moved since last time the mode date was read.
Bit 2	Starting	This flag is set when the device has started less than 3 minutes ago and has not been connected to any device. (firmware version >= 0.9.2)
Bits 3:7	-	reserved

Color description

Value	Color	Pantone value
0x01	Brown.	7582C
0x02	Esmerald	3272C
0x03	Lemon	389C
0x04	gray brown	7532C
0x05	gray green	5545C
0x06	classic-green	363C
0x07	gray-blue	549C

FlowerPower scan response data

Local name complete								**Connection interval range**						**Tx Power level**		
data type: 0x09 Default Value: “FlowerPower”								data type: 0x12 Value: to be determined						data type: 0x0A Value : 0db		
0x07	0x09	0x48	0x61	0x77	0x61	0x69	0x69	0x05	0x12	--	--	--	--	0x03	0x0A	0

Note The scan response data is obtained by performing active scanning.