Electronic Supplementary Information (ESI) for: Remotely controlled nanofluidic implantable platform for tunable drug delivery

*Nicola Di Trani[1,2], Antonia Silvestri[1,3], Giacomo Bruno[1,3], Thomas Geninatti[1], Corrine Ying Xuan Chua[1], April Gilbert[1], Giulia Rizzo[1,3], Carly S. Filgueira[1], Danilo Demarchi[3], Alessandro Grattoni[1,4,5]**

[1]Nanomedicine Department, Houston Methodist Research Institute, Houston, TX, USA.
[2]University of Chinese Academy of Science (UCAS), Shijingshan, 19 Yuquan Road, Beijing 100049, China

[3]Department of Electronics and Telecommunications, Politecnico di Torino, Turin 10024, Italy
[4]Department of Surgery, Houston Methodist Hospital, 6550 Fannin Street, Houston, TX, 77030, USA.
[5]Department of Radiation Oncology, Houston Methodist Hospital, 6550 Fannin Street, Houston, TX, 77030, USA.

*Corresponding Author, agrattoni@houstonmethodist.org, phone: +1 (713) 441-7324

Supplementary note 1: Power consumption analysis

The high-power consumption scenario investigated consists of: advertising signal set with a 100 ms interval; a connection of 30 s every hour; constantly active electrodes with a current equal to the maximum peak value registered during the in vitro release (100 μA).

Supplementary note 2: Remote communication via Bluetooth with implant

To establish radiofrequency-based communication with our implant, we employed the Bluetooth Low Energy (BLE 4.0) protocol due to its high flexibility and low power consumption. On the server (laptop), an ad-hoc MATLAB® script was used to remotely change the implant behavior according to a predetermined schedule, by communicating through a serial port to a BLE-enabled USB dongle. In the microcontroller unit (MCU) of the PCB, we implemented a program based on the same BLE protocol, which can communicate with other BLE-enabled implants through the integrated antenna. This program simultaneously handles BLE communication and voltages to apply to the electrodes.

To manage communications, the implant can be in either of 2 states: advertising or connected. During the advertising state, the controller will send advertising packets on three different radio-frequency channels. These packets will be used by the server (laptop) to acknowledge the presence of the client (implant). The frequency of these advertising packets plays a role in determining the time needed to create a connection between a server and a client. High frequencies result in fast responses and high energy usage, while low frequencies save energy but may lead to response lags in a connection request. Keeping in mind a balance between power consumption and response lag, we found that an advertising time of 1-10 s can offer considerable energy saving with negligible communication lags. When the server acknowledges the presence of the client, it may request a connection, during which the two implants exchange copious amounts of packets. During a connection, the server can read or change the value in a register inside the MCU, which will then be used to act on the electrodes potential.

As a security measure, every time a connection to an implant is performed by the MATLAB® script, the confirmation of successful communication is requested by reading back the values of the transmitted data, just before closing the communication. If the value read is different from the one written, an error is detected, and the register value is rewritten.

Supplementary note 3: Applications for remote control of the device

Figure S1 shows the interface of an iOS application that exploits the integrated BLE module, which is present in most smartphones, to establish a connection with the device and change parameters as needed. We also established a graphical user interface of the web application based on the Node-RED framework, that allows real-time monitoring of all active devices (Figure S2). It shows their identification code, the received signal strength indication (RSSI) and the code of the last known administration regimen applied. Furthermore, it allows the user to select from a list of available devices and set a new release regimen. This web interface was developed aiming at low energy wireless communications to extend lifespan of battery-powered devices. As a matter of fact, we implemented the asynchronous MQTT (Message Queuing Telemetry Transport) communication protocol for all the information exchanges because it is simple, light weight and offers nearly instantaneous data exchange with small packets. The communication between the web interface and the device is mediated by a Python script hosted on a Raspberry Pi 3, which is a miniaturized low-cost portable computer that could be used to share device information with the physician over the internet.

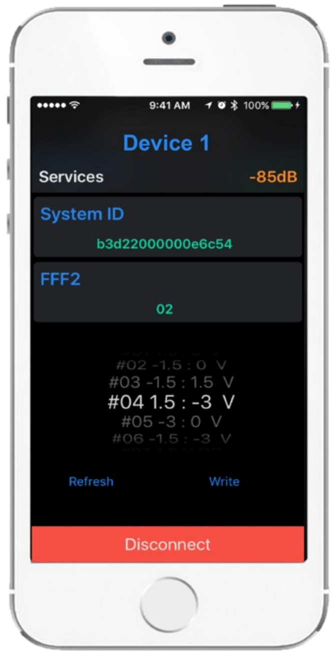

Figure S1. iOS application interface.

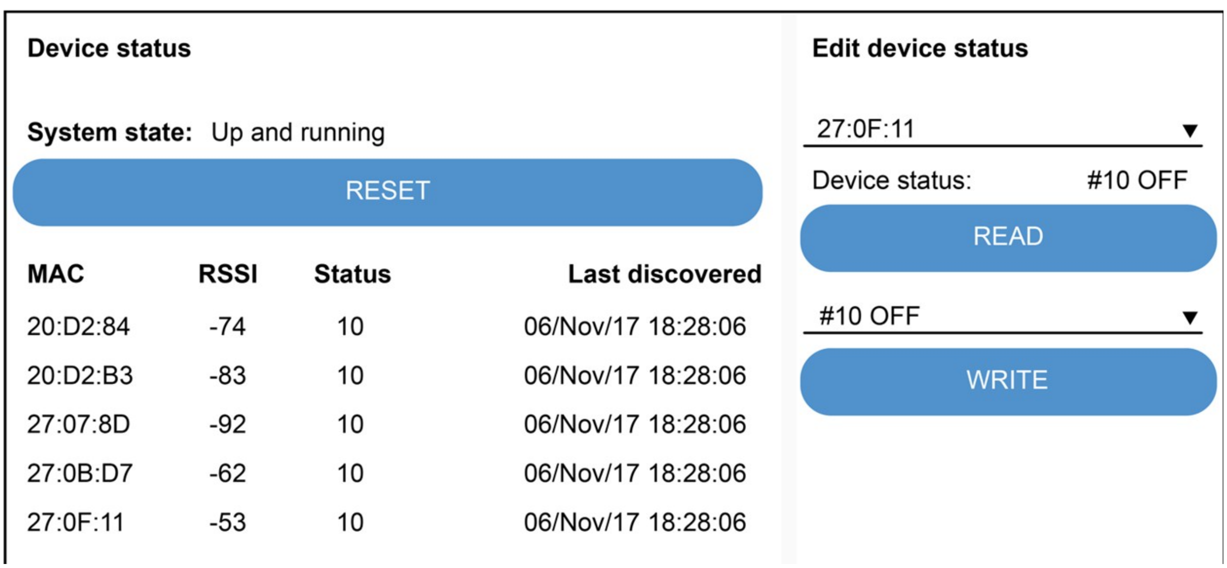

Figure S2. Web based application interface.