Digitally-Adjustable Phrenic Nerve (DAPhNe) Stimulator System

Technical Reference: DAPhNe Stimulator Prototype v1.0

Created by: Alexey Revinski

Revision: 1.1

Date: 18 September 2017

Version History

Version	Date	Comment
1.0	08/29/2017	First release of the document
1.1	09/18/2017	Added pinout information for DAPhNe MCU and DAPhNe STIM boards

Table of Contents

Table of Abbreviations i
1 Introduction 1
2 Project Background 1
3 Design Overview 3
4 Electrical and Physiological Considerations 4
4.1 Constant-Current Stimulation 4
4.2 Charge Balancing: Biphasic Stimulation 4
4.3 Device Output 5
4.3.1 Programmable Signal Parameters 5
4.3.1 Output Waveform 5
4.4 Stimulation Control Signals 7
5 Hardware Design 8
5.1 General Overview 8
5.2 Stimulation Circuit 9
5.2.1 Rationale 9
5.3 Device Circuit Diagram 11
5.4 Prototype Hardware Design 13
5.4.1 Battery Information 14
5.4.2 3.3V DC Supply 14
5.4.3 Charging Circuitry 15
5.4.4 NFC Memory Antenna Board 15
5.4.5 Breakout Board Components 15
5.4.6 DAPhNe MCU v1.0 Printed Circuit Board 16
5.4.7 DAPhNe STIM v1.0 Printed Circuit Board 18
6 Software Design 20
6.1 Software Requirements 20
6.2 Power Consumption Considerations 21
6.3 Software Architecture 21
6.4 Reducing CPU duty cycle 22
6.4.1 Inspiratory Mode: WFI Mode 22
6.4.2 Expiratory Mode: Active Halt Mode 22
6.5 Peripheral Chaining 23
6.5.1 Peripheral Event Flow 23
6.5.2 TIM1 25
6.5.3 TIM2 25
6.5.4 TIM4 26
6.5.5 DAC 26
6.5.6 DMA 27
6.5.7 Peripheral Configuration Buffers 27
6.6 Safely Ending IM 28
6.7 NFC Data Exchange 28

Table of Abbreviations

ANSI	American National Standards Institute
bpm	Breaths per Minute (unit of BR)
BR	Breathing Rate
CAMP	Center for Autonomic Medicine in Pediatrics
CC	Capture/Compare (Memory Register)
CCHS	Congenital Central Hypoventilation Syndrome
CCS	Constant-Current Stimulation
CPU	Central Processing Unit
CVS	Constant-Voltage Stimulation
DAC	Digital-to-Analog Converter
DAPhNe	Digitally-Adjustable Phrenic Nerve (Stimulator)
DIP	Dual In-line Package
DMA	Direct Memory Access (Controller)
EM	Expiratory Mode
I2C	Inter-Integrated Circuit (Communication Protocol)
IC	Integrated Circuit (Technology)
IM	Inspiratory Mode
IPR	Inter-Pulse Resting (Period)
IT	Inspiratory Time
LED	Light Emitting Diode
MCU	Microcontroller Unit
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NFC	Near-Field Communication
OC	Output Compare (Memory Register)
PCB	Printed Circuit Board
PF	Pulse Frequency
PL	Pulse Length
PM	Pulse Magnitude
RAM	Random-Access Memory
RF	Radio Frequency (signal)
RTC	Real Time Clock
SoC	System-on-a-Chip
SPDT	Single Pole Dual Throw (Switch)
SPST	Single Pole Single Throw (Switch)
SRS	Stimulation-to-Recharge Scaling (Ratio)
TIM1	Timer 1
TIM2	Timer 2
UVLO	Under Voltage Lock-Out (Feature)
Vmag	Pulse Magnitude control signal
Vpol	Pulse Polarity control signal
Vtim	Pulse Timing control signal
WFI	Wait for Interrupt (Low-Power Mode)

1 Introduction

This is a technical reference document for the Digitally-Adjustable Phrenic Nerve (DAPhNe) Stimulator System, developed by a team of undergraduate students at Northwestern University as part of their Biomedical Engineering capstone design project in 2016-2017 academic year.

This document provides a thorough walkthrough of the hardware and software design of the DAPhNe system to date, providing technical specifications and rationale behind most of the design decisions. This document is **not** aimed to provide extensive physiological information that guided this project.

2 Project Background

The DAPhNe Stimulator System is aimed towards patients who acquire, genetically or otherwise, severe autonomic nervous system disorders that result in prolonged hypoventilation. Part of the system is implantable; it electrically stimulates the patient's phrenic nerves, which innervate the diaphragm muscle and provide a natural way for the patients to breathe. As opposed to mechanical ventilation, this approach uses the body's own electrical structures to provide ventilation support to the patient.

The system consists of an implantable stimulator and an external controller that wirelessly communicates with the stimulator. The scope of this project was the development of the essential hardware and software of the implantable stimulator; the external controller device is yet to be designed.

The system was designed for and in collaboration with the Center for Autonomic Medicine in Pediatrics (CAMP) at Ann & Robert H. Lurie Children's Hospital of Chicago. The disease that the project initially targeted was Congenital Central Hypoventilation Syndrome (CCHS), a rare and devastating genetic disorder that causes autonomic nervous system dysregulation, and, among other symptoms, mild to severe lack of breathing control, which causes hypoventilation. However, the developed system is applicable to any other hypoventilation-inducing conditions, in which a long-term solution is necessary or desired.

The current mainstream solution to the effects of CCHS and other hypoventilation-causing conditions is positive pressure ventilation by means of an external mechanical ventilator connected to the patient's respiratory system via a tracheostomy.

A more physiologically-appropriate approach is phrenic nerve stimulation (pacing). The phrenic nerve innervates the diaphragm; by providing stimulation pulses to the nerve, the stimulator causes the diaphragm to contract. This develops negative pressure (with respect to atmospheric) inside the patient's lungs, drawing oxygenated air inside. Upon rest, the diaphragm expels de-oxygenated air from the lungs.

The current FDA-approved device that accomplishes this is the Avery Breathing Pacemaker, manufactured by Avery Biomedical Devices. It includes bilaterally-implanted short-range RF receivers that condition RF signals sent by an external transmitter and send the output to their respective phrenic nerves via specially designed electrodes. The external receiver takes on all of the control functionality – without it actively transmitting in immediate range or the receivers, stimulation is not possible.

While the benefits of the Avery Breathing Pacemaker system over mechanical ventilation are numerous, its drawbacks are just as extensive. This system:

- Requires the patient to be tethered to two antenna coils of the external transmitter; these antennas are placed right over the subcutaneous bilaterally-implanted receivers.
- Offers only one respiration mode per transmitter; for example, to switch from "walking" to "running" respiration mode, the patient needs to switch between two different transmitters.
- Was designed in 1960s using outdated analog technology, seemingly without the end-users (doctors and patients) in immediate sight. The external transmitter offers no reliable metrics for adjusting its respiration settings – the doctor has to manually adjust the system by hand, viewing the output using an EMG setup.

3 Design Overview

The DAPhNe Stimulator System takes aim at the above drawbacks and was designed provide the patient with a reliable, semi-autonomous alternative to the current standard.

The designed system consists of an implantable phrenic nerve stimulator and an external digital controller (Figure 1). The internal device is battery powered, and is responsible for stimulation output generation and proper breathing control on its own. The external controller is only needed to switch the implantable device to a different respiration mode. Upon changing breathing modes (i.e. from “walking” to “running” or “sitting”, “eating”, etc.), the external device communicates new breathing parameters to the implant via NFC technology. The implant autonomously provides breathing control at the desired settings, while the external controller is only used to switch the implant from one mode to another on per-breath basis.

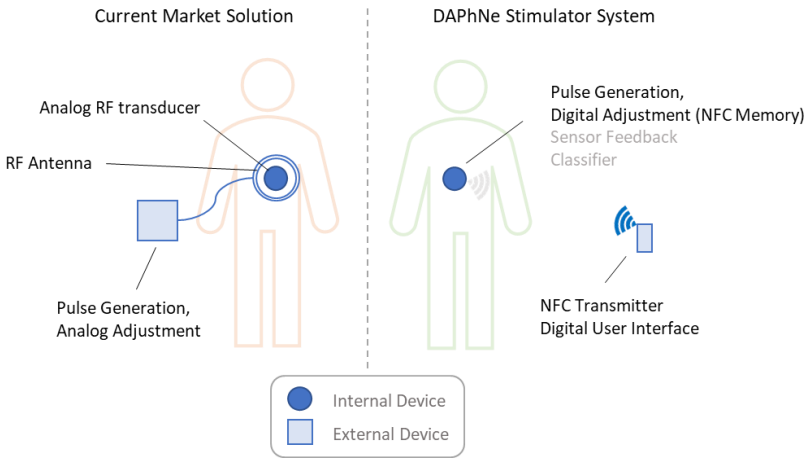


Figure 1: System Diagram: Current Solution vs. DAPhNe Stimulator System
(functions in gray are planned in future iterations of the device)

At this point in development, the current prototype of the device does not yet use any biological feedback. Bio-feedback mechanisms in future development may allow the system to become almost completely autonomous, eliminating the need for regular control with an external device.

4 Electrical and Physiological Considerations

4.1 Constant-Current Stimulation

To stimulate neural tissue, most devices either apply a constant current through the nerve of interest, or a constant voltage across it. The two approaches are called constant-current stimulation (CCS) and constant-voltage stimulation (CVS). Both have advantages and drawbacks. CCS pulls a constant current through the load no matter the tissue impedance, which ensures proper neural excitation at each stimulation pulse. However, this leads to larger power consumption than CVS. CVS, on the other hand, conserves current at higher tissue impedances at the expense of reliable stimulation. Because the focus of the DAPhNe system is on patient treatment, support, and reliability, the stimulator employs CCS to achieve phrenic nerve excitation. Hence, it pulls a programmable current through tissue, limited only by the device's rail-to-rail power supply.

4.2 Charge Balancing: Biphasic Stimulation

As the charge is injected into the tissue, it disrupts the electrical potential balance in neural cells. The body naturally achieves ionic balance in the tissue via cellular pumps and channels, but this process happens gradually. Stimulation devices usually do not balance the charge they inject into the tissue. While this is a tolerable short-term solution, studies have shown that prolonged unbalanced stimulation can lead to neural damage. In effect, solutions like these hurt the patient in the long term.

To combat this issue, some researchers have used a biphasic stimulation pattern using dual power supplies. In effect, a negative stimulation pulse depolarizes the nerve, and the following positive re-charge pulse balances out the injected charge. The DAPhNe Stimulator provides a solution to this as well – it generates a positive re-charge pulse that has a much smaller magnitude but takes place for a longer time period than the negative stimulation pulse. By matching the total amount of charge exchanged during each positive and negative pulse, the device takes care of the charge-balancing problem, but also prevents the unwanted anodic stimulation.

4.3 Device Output

Based on the above considerations, the output of the device consists of biphasic pulses depicted in Figure 2. The device has full control of all the depicted parameters.

4.3.1 Programmable Signal Parameters

Table 1 shows the breathing parameter settings satisfied by the current prototype of the device.

Table 1: Device Output Parameters

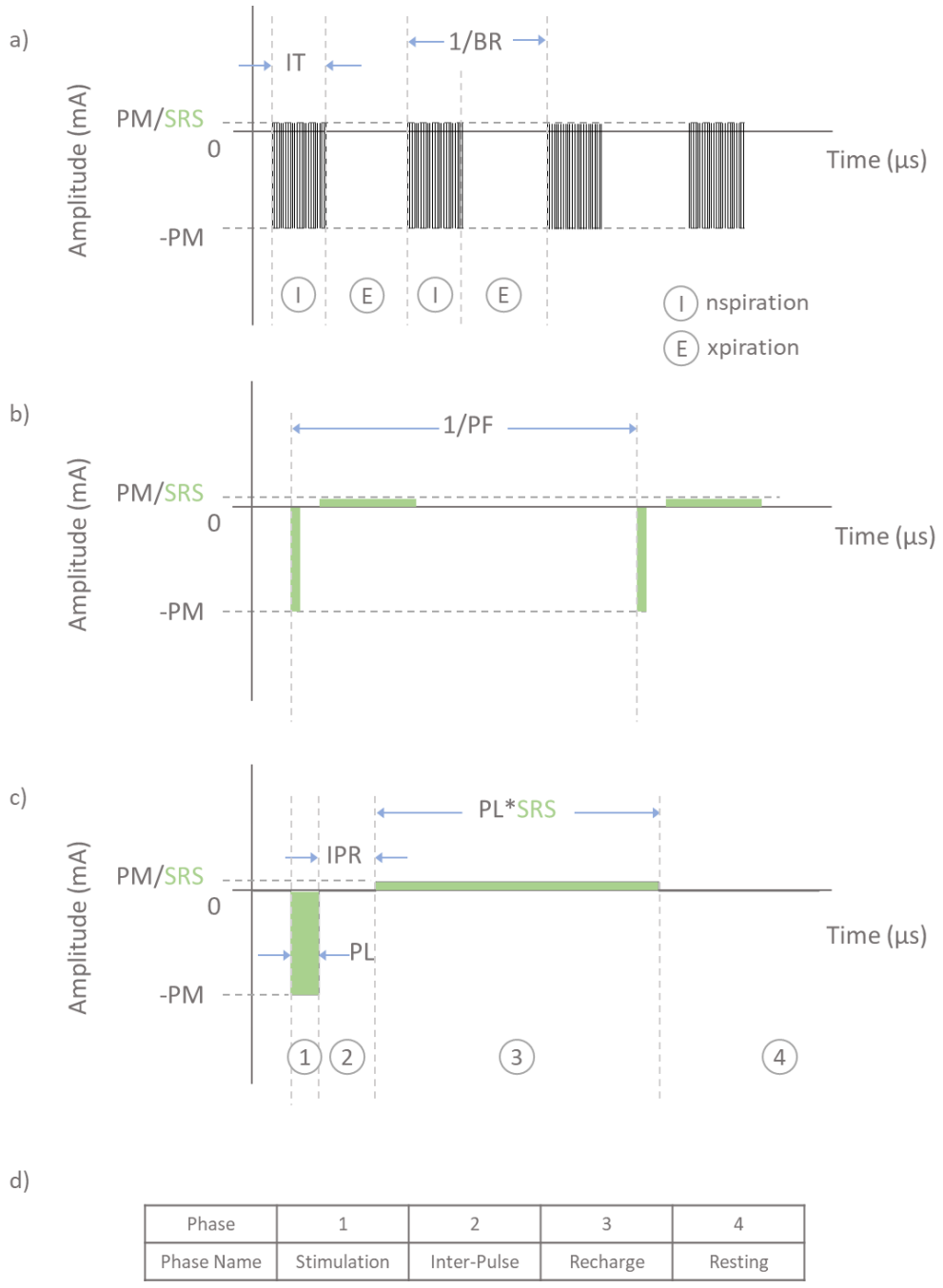
Acronym	Full Name	Min*	Typ	Max*
BR	Breathing Rate	10bpm	20bpm	35bpm
IT	Inspiratory Time	0.8s	1.2s	2.0s
PF	Pulse Frequency		20Hz	
PL	Pulse Length	10us	150us	1000us
PM	Pulse Magnitude	10uA	2-5mA	7mA
IPR	Inter-Pulse Resting Period	100us	3*PL	1200us
SRS	Stimulation-to-Recharge Ratio		10	

* Most minimum and maximum values are limited in software to ensure patient safety. Further evaluation is required to determine absolute min/max values that should be allowed.

4.3.1 Output Waveform

On the breath-to-breath scale (Figure 2.a), the device has full control of the patient's breathing rate and inspiratory time. For optimal physiological stimulation, the pulse cycles are generated at 20Hz, with the higher frequencies allowed to smooth out the contraction curve (Figure 2.b).

The pulses themselves are generated having four phases, as shown in Figure 2.c. The actual stimulation pulse is followed by a period of rest (IPR), followed by a recharge pulse smaller than stimulation pulse in magnitude and larger in length by a certain SRS ratio. For example, with SRS = 10, the recharge pulse is 10 times longer than the stimulation pulse, but also 10 times smaller in magnitude. This allows for the positive pulse to assume a low magnitude, yet completely balance the charge injected into tissue. The fourth phase is the resting phase, during which no signals are generated until the next pulse cycle.



Phase	1	2	3	4
Phase Name	Stimulation	Inter-Pulse	Recharge	Resting

Figure 2: Stimulation electrode voltage potential with respect to body ground, normalized by tissue impedance; a) Time scale: seconds – multiple breaths, showing inspiratory and expiratory phases; b) Time scale: tens of milliseconds – two pulse cycles; c) Time scale: hundreds of microseconds – a single pulse cycle; d) Pulse phases referenced throughout the document. For detailed acronym specification, refer to Table 1**.**

4.4 Stimulation Control Signals

To achieve the device output shown in Figure 2, an embedded microcontroller provides three control signals to a hardware stimulation circuit (refer to Hardware Design section):

- Pulse timing logic (on/off), called $\mathbf{V_{TIM}}$ in the rest of the document
- Pulse polarity logic (positive/negative), called $\mathbf{V_{POL}}$ in the rest of the document
- Pulse magnitude analog signal, called $\mathbf{V_{MAG}}$ in the rest of the document

Figure 3 shows how the three signals control device output during all four phases of the pulse cycle.

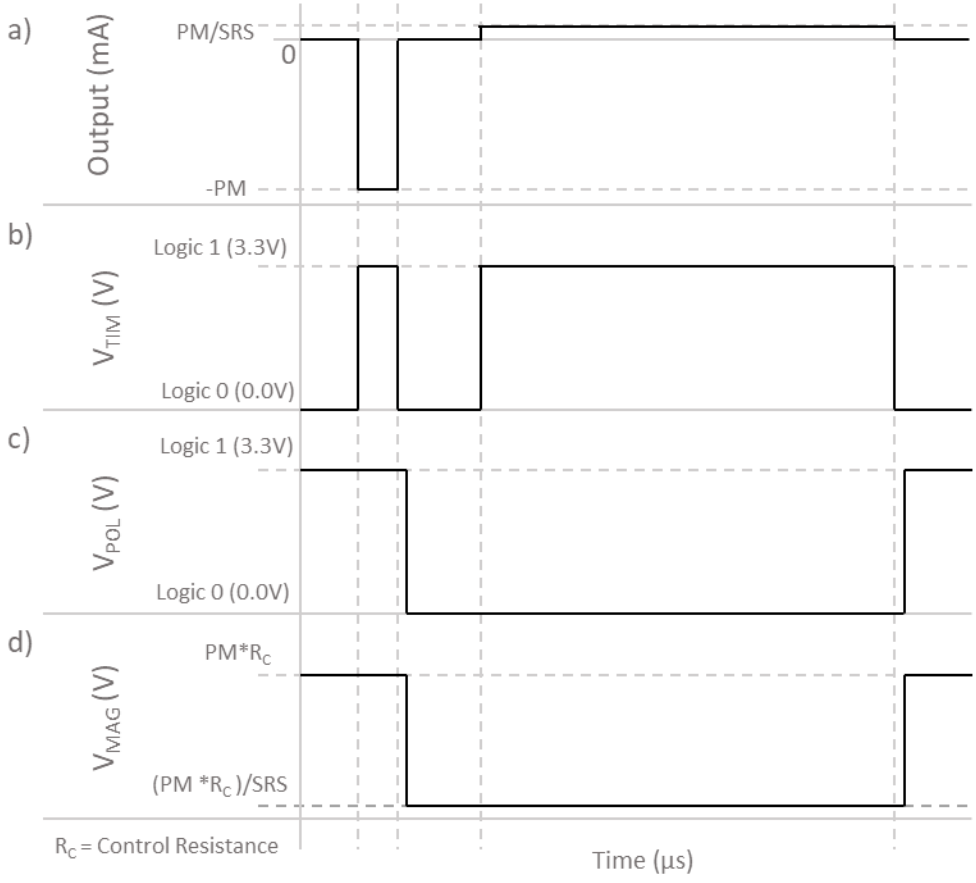


Figure 3: Stimulation control signals; a) Desired device output signal, b) Pulse timing logic input, c) Pulse polarity logic input, d) Pulse magnitude control input

5 Hardware Design

The current prototype consists of multiple printed circuit boards. Custom printed circuit designs were developed using EagleCAD, recently bought out by Autodesk. Otherwise, breakout boards were used for multiple IC components, and a breadboard setup (see sections below) was used as the main platform for development of this prototype of the DAPhNe stimulator. Circuit diagrams used for this document were generated using Scheme-It – a free design tool provided by DigiKey Electronics.

5.1 General Overview

Figure 4 shows the overall hardware diagram. The device is powered by a 3.7V Lithium Ion battery, which is charged wirelessly using a Qi-capable power receiver SoC through the patient's thin skin layer. The battery supplies energy to two voltage supplies. The 3.3V supply powers the embedded microcontroller (MCU) and the wireless NFC memory; the 10V boost supply powers the stimulation circuit. The device draws current through the tissue by means of a hemi-cuff electrode and a subcutaneous anode plate. The latter two were provided by Avery Biomedical Devices.

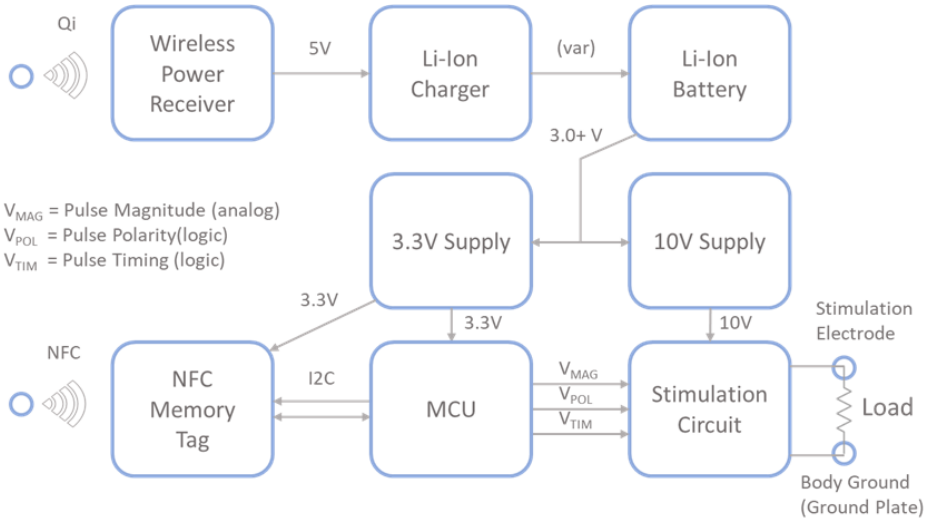


Figure 4: Hardware Module Diagram

5.2 Stimulation Circuit

Referring to Section 4.4, the microcontroller outputs three control signals that interface with a stimulation circuit. The theoretical stimulation circuit, along with a "summarized" depiction of the microcontroller is shown below in Figure 5.

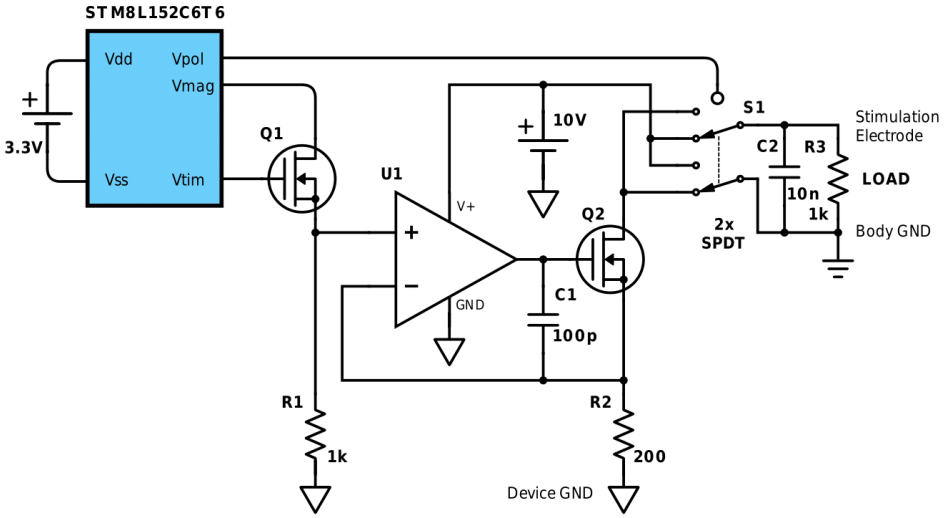


Figure 5: Theoretical Stimulation Circuit

5.2.1 Rationale

The circuit consists of a programmable current sink, a polarity switching circuit, and a "valve" MOSFET switch (Q1) controlling the current sink.

5.2.1.1 Current Sink Circuit

The programmable current sink is implemented with a precision op-amp, a low Rdson MOSFET, and a high-precision current flow control resistor (R2). The voltage potential applied to the positive terminal of the op-amp appears at the R2-Q2 node in the circuit via negative feedback; this establishes a certain current over R2. With a value of 200Ω, 1V at the positive terminal of the op-amp results in 5mA of current flowing through R2. The op-amp outputs just enough voltage to the gate of Q2 as to establish this current through R2 and equate the potential at its the positive and negative terminals.

Current flowing through R2 comes only from the source of Q2. Assuming the dual SPDT switch is configured in such a way that the load is in series with Q2, all of this current also flows through the load. In this way, a programmable current is pulled through the load, no matter its actual impedance.

5.2.1.2 Control Switch

The Q1 MOSFET acts as a "valve" for the V_{MAG} signal output from the microcontroller; the gate of Q1 is connected to V_{TIM} output. So, when V_{TIM} is high, Q1 fully conducts (V_{TIM} is always greater than V_{MAG}, which is in the range of 0-2V), and V_{MAG} appears at the positive terminal of the op-amp, forcing current to flow through the load. When V_{TIM} is low, Q1 does not conduct; the positive terminal of the op-amp is pulled to device ground by R1, and no current flows through the load.

5.2.1.3 Polarity Switching Circuit

The signal V_{POL} controls the orientation of the dual SPDT switch. Both SPDT branches of the switch are tied together to V_{POL} – they are both switched when V_{POL} changes.

When V_{POL} is high, the equivalent circuit is shown in Figure 6. Body ground is connected to the 10V source, and stimulation electrode – to the drain of Q2. When Q2 conducts, current flows from body ground to the stimulation electrode – electrode is driven low with respect to body ground.

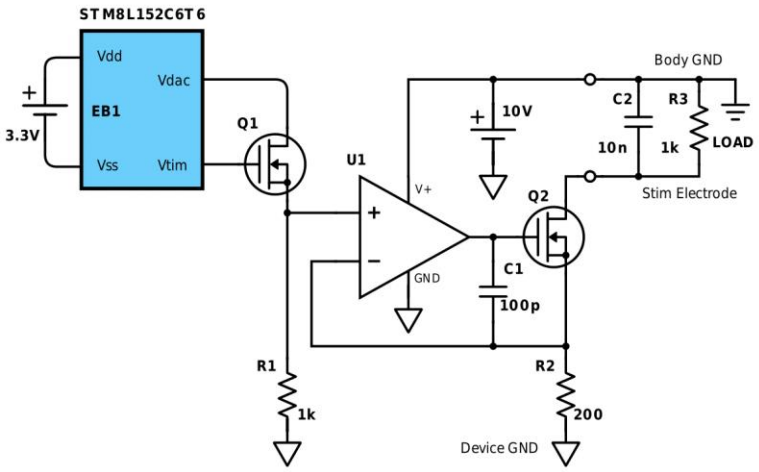


Figure 6: Equivalent circuit when V_{POL} is high

Alternatively, when V_{POL} is driven low, the equivalent circuit is shown in Figure 7. In this case, the stimulation electrode is connected to the 10V source, and body ground – to the drain of Q2. When Q2 conducts, the electrode is driven higher than body ground, and current flows in the opposite direction, balancing the charge pulled through tissue during a previous negative pulse.

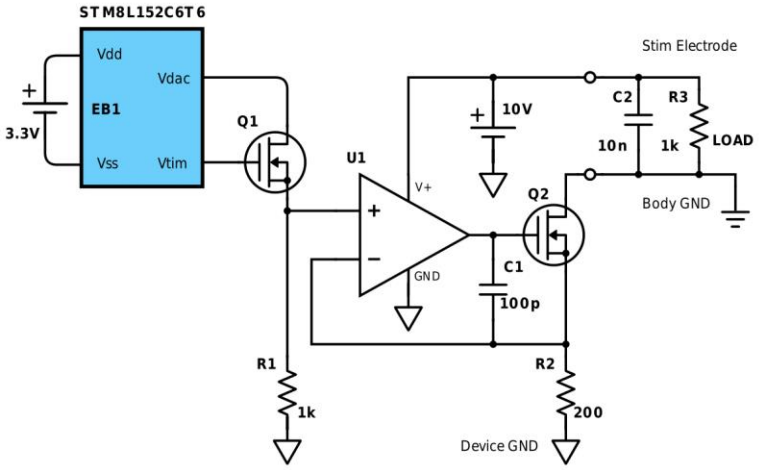


Figure 7: Equivalent circuit when V_{POL} is low

Capacitor C1 prevents spikes appearing at the output of the op-amp as a result of sudden switching of Q1, and C2 smoothes out the spikes caused by sudden switching of Q2. Both capacitors are there to smooth out current spikes through the tissue.

5.3 Device Circuit Diagram

Figure 8 shows the overall device circuit diagram. No device protection features have been incorporated into the design yet – only functionally relevant components described below. For a list of components used, please refer to Table 2.

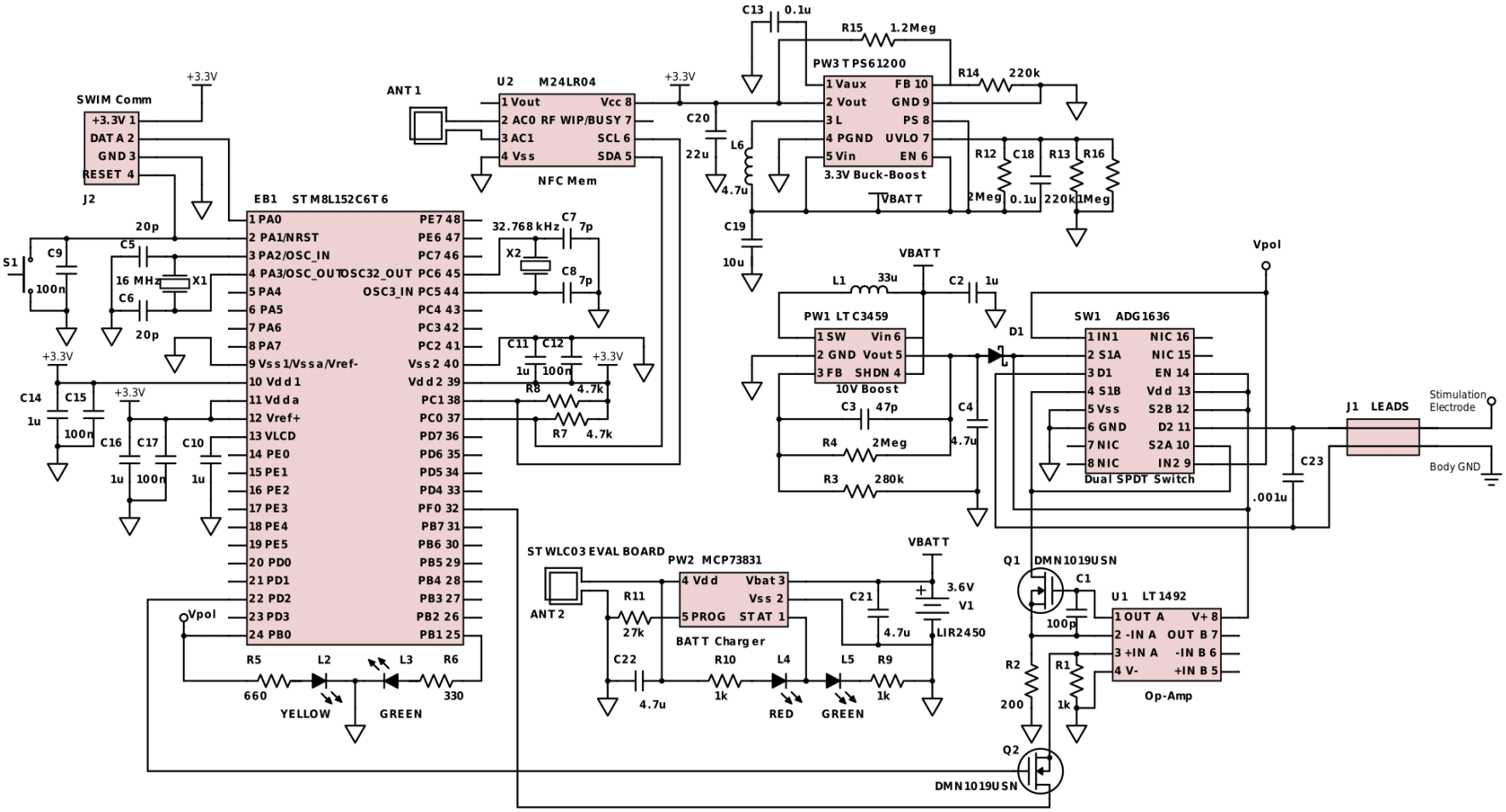


Figure 8: Circuit Diagram of DAPhNe Stimulator Prototype v1.0

5.4 Prototype Hardware Design

The current prototype incorporates some of the above components in custom-designed PCBs, and others as part of third-party solutions, some of which are open-source devices. A picture of the current prototype is shown in Figure 9. Table 2 lists the active and special passive components used in the prototype.

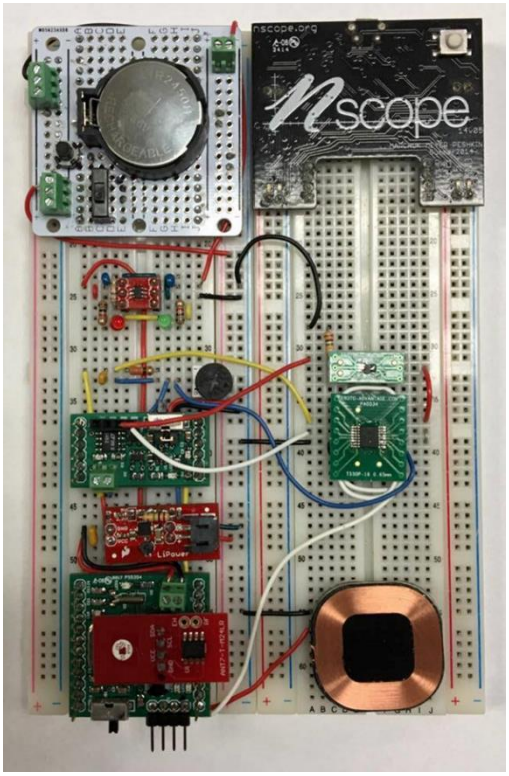


Figure 9: Current Prototype of the DAPhNe Stimulator

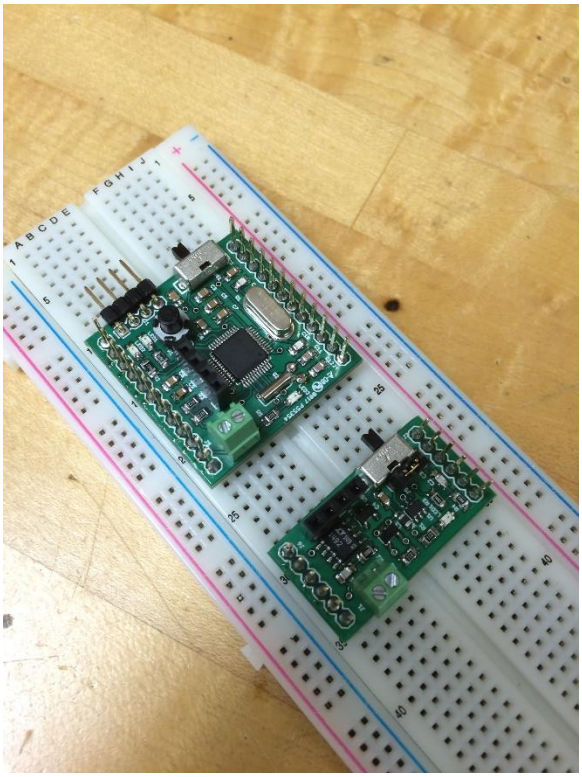


Figure 10: Custom PCBs: DAPhNe MCU (top left), and DAPhNe STIM (bottom right)

Table 2: Active and Special Passive Components used in DAPhNe Stimulator Prototype v1.0

Component Model	Description	PCB Solution
STM8L152C6T6	16MHz Low-power MCU with 12-bit DAC, 32KB FLASH, 2KB RAM, and 41 GPIOs	DAPhNe MCU
M24LR04	14x14mm double layer antenna reference board for the M24LR04E Dual Interface EEPROM	ANT7-T-M24LR04
STWLC03	Dual-mode Qi/ PMA wireless power receiver	STEVAL-ISB039V1
MCP73831	Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controller	(Breakout Board)
LTC3459	10V Micropower Synchronous Boost Converter	DAPhNe STIM
LT1492	5MHz, 3V/us, Low Power Single Supply, Dual Precision Op Amp	DAPhNe STIM
DMN1019USN	12v N-Channel Enhancement Mode MOSFET	DAPhNe STIM
TPS61200	0.3V Input Voltage Boost Converter with 1.3A Switches and 'Down Mode' in 3x3 QFN	Sparkfun PRT-10255
ADG1636	1 Ω Typical On Resistance Dual SPDT Switch	(Breakout Board)

5.4.1 Battery Information

The battery powering the current prototype is LIR2450H, a 190 mAh rechargeable Li-Ion coin cell. A breakout board was designed for this battery size (large square white-colored prototyping board on top left of Figure 9). It features a 2450-size battery holder, a push button (SPST) and an SPDT switch, with multiple screw-in ports for the positive and negative terminals of the battery.

5.4.2 3.3V DC Supply

The prototype is powered by a Li-Ion battery; batteries of this type can assume close to 4.0 V when fully charged, and go down to 3.0V when discharged. Yet, the chosen microcontroller is most stable at a 3.3V supply. So, a simple buck converter or a linear regulator cannot be used, as the battery voltage can dip below 3.3V for some portion of the discharge cycle. A buck-boost converter, however, is very suitable.

The DAPhNe Stimulator prototype uses a 3.3V DC supply board by SparkFun, LiPower PRT-10255. It powers the microcontroller, NFC memory chip, and related circuitry. At the heart of the board is the TPS61200 buck/boost converter from Texas Instruments, which features programmable voltage output, overtemperature protection, and an undervoltage lockout (UVLO) feature.

The latter feature makes sure the battery is not overdrawn by the rest of the circuit. At a certain programmable low voltage threshold, the converter essentially disconnects the battery supply from the rest of the circuit. On the LiPower board, this threshold was adjusted to about 3.0V by soldering a 1MΩ resistor (R16) in parallel with R13 – refer to the circuit diagram in Figure 8.

This power supply board from SparkFun is a great short-term solution. In later development, a microcontroller running on a sub-2V supply should be chosen – in that case, a simple buck-converter with a UVLO-like feature may be used.

5.4.3 Charging Circuitry

The battery of the device is wirelessly charged using a Qi-compatible device. In the current prototype, the wireless power receiver is an evaluation board from ST Microelectronics that uses their STWLC03 Dual Mode Qi/PMA power receiver IC, STEVAL-ISB039V1R. The external transmitter that supplies energy to the STWLC03 board is the STEVAL-ISB039V1T board, based on ST's 32-bit ARM-Cortex M0 STM32F0 microcontroller.

The STEVAL-ISB039V1R board supplies a constant 5V when the transmitting board's wireless coil is in range of reception. This 5V supply is then used by a Li-Ion charge management controller from Microchip Technology, MCP73831, to charge the LIR2450H battery. MCP73831 is a complete charging solution for Li-Ion batteries, and features multiple charging modes (Preconditioning, Fast Charge, Constant Voltage – all internally managed), and a UVLO feature like TPS61200. The charging cycle only starts once the supply voltage from STWLC03 rises above about 3.4V. The MCP73831 chip also features a tri-state status pin; LEDs L4 and L5 in the circuit diagram in Figure 8 are used in this prototype to visually monitor the status of the battery when charging.

5.4.4 NFC Memory Antenna Board

As a great substitute for a custom NFC antenna interface, the ANT7-T-M24LR04 reference board from STMicroelectronics is used as a passive NFC receiver tag in the current prototype. It features a 14mm x14mm NFC antenna, and DIP-style holes for breakout pins for its power supply and I2C interface data pins.

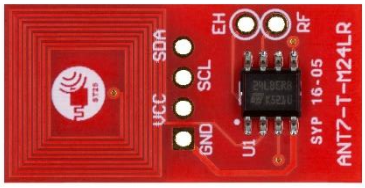


Figure 11: ANT7-T-M24LR04 NFC Antenna Board

5.4.5 Breakout Board Components

MCP73831 (Li-Ion charging IC), ADG1636 (SPDT switch IC), and a Schottky diode used to prevent current backflow into the 10V supply (when it is off) from digital inputs of ADG1636 were installed in the prototype using breakout boards.

5.4.6 DAPhNe MCU v1.0 Printed Circuit Board

A custom PCB design has been developed for the microcontroller that houses the device's software (see Figure 10, top left). The board also acts as a plug-in receiver for the ANT7-T-M24LR04 board (see Figure 9, bottom left). DAPhNe MCU is a two-layer PCB that features:

- STM8L152C6T6 microcontroller
- 16MHz external crystal oscillator used to clock CPU and other peripherals
- 32.768 kHz external crystal oscillator used for the Real Time Clock (RTC) peripheral
- 0805 package passives
- On/off switch, connecting the 3.3V supply from TPS61200 to the rest of the board
- Power status LED
- Two general-use LEDs for firmware debugging
- Screw-in port for the power supply wires
- Four-0.1"-male-header-pin interface to the ST-LINK/V2 debugger tool
- 24 0.1" DIP-style breakout pins, used partly to securely position the PCB on a breadboard and partly as signal output pins, to be interfaced with DAPhNe STIM and other boards in the prototype
- Four-0.1"-pin female connector that acts as a plug port for the ANT7-T-M24LR04 NFC board
- Push-button used to reset the MCU

After the DAPhNe MCU v1.0 was designed and incorporated into the prototype, one of the MCU pins that was merely used to drive a blue LED for debugging (PB0) actually got an essential role in the device functionality (pulse polarity control signal), and had to be connected to one of the free breakout pins on the PCB by an external wire.

5.4.6.1 DAPhNe MCU Board Pinout

The pinout of the DAPhNe MCU board is shown in Figure 12.

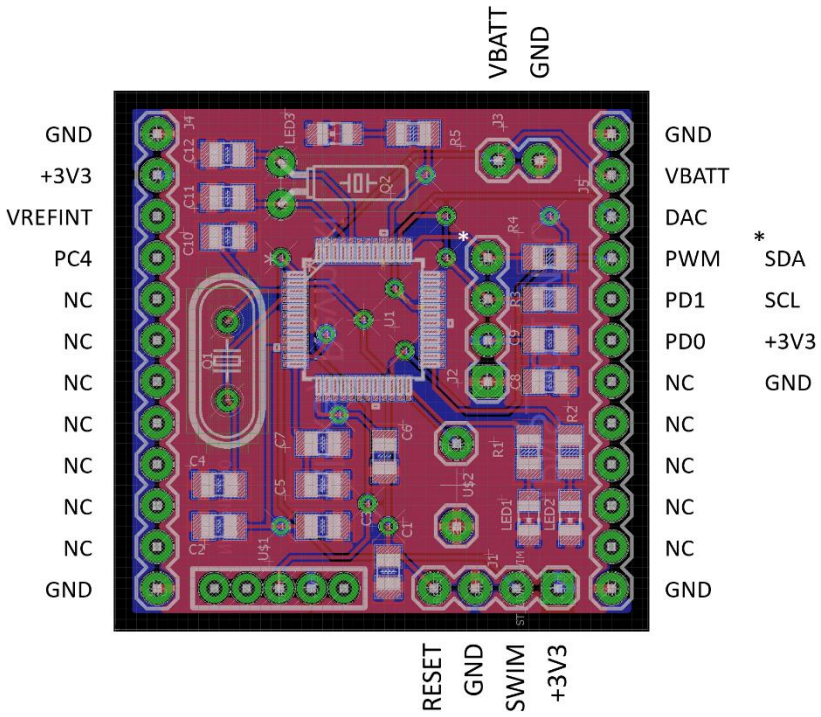


Figure 12: DAPhNe MCU Pinout

Pin	Description
GND	Device ground
VBATT	Regulated 3.3V input
+3V3	3.3V output. VBATT and 3.3V are connected with a slide switch
DAC	V_{DAC} line. Connected to DAPhNe STIM
PWM	V_{POL} line. Connected to DAPhNe MCU
SWIM	SWIM data pin
RESET	Reset line for the microcontroller. Used by debugger
VREFINT	Internal voltage reference of the microcontroller. Not used
SDA	Data line (I2C)
SCL	Clock line (I2C)
PD1, PD0, PC4	Microcontroller GPIOs. Not used.
NC	Not connected. For structural integrity only

5.4.7 DAPhNe STIM v1.0 Printed Circuit Board

A custom PCB design was developed for the stimulation circuit, excluding the polarity switch IC (see Figure 10, bottom right, and Figure 9, middle left). The polarity-switching circuit is housed externally using standard breakout boards (Figure 9, middle right). The DAPhNe STIM PCB was originally designed to plug in directly into the DAPhNe MCU board, but was installed separately in the prototype. DAPhNe STIM is a two-layer PCB that features:

- LTC3459 10V boost-converter
- LT1492 single supply dual precision op-amp
- DMN1019USN 12V N-channel enhancement mode MOSFETS
- 0805 package passives
- On/off switch that connects the battery power supply to the rest of the board
- Power status LED
- 12 0.1" DIP-style breakout pins, used partly to securely position the PCB on a breadboard and partly as signal input/output pins, to be interfaced with DAPhNe MCU and other boards in the prototype
- Four-0.1"-pin female connector that acts as a breakout connector for the polarity-switching circuit. If some pins of the four-pin connector are jumped to each other, the device would still work, but stimulation electrode will only be able to be driven lower than body ground – this is essentially a polarity-switching circuit bypass
- Screw-in mount for the stimulation electrode and body ground electrode plate, currently not used in the prototype
- Two-pin jumper breakout for development of the LTC3459's shutdown feature (not used)

5.4.7.1 DAPhNe MCU Board Pinout

The pinout of the DAPhNe STIM board is shown in Figure 13.

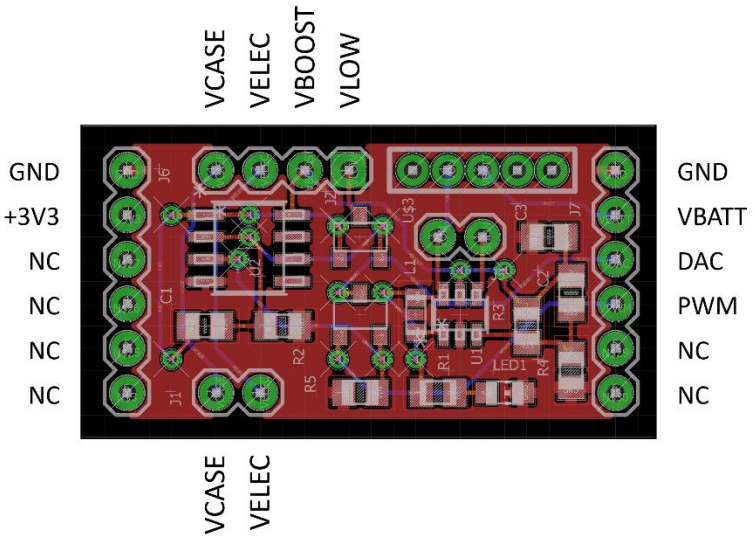


Figure 13: DAPhNe STIM Pinout

Pin	Description
GND	Device ground
VBATT	Li-Ion battery voltage input
+3V3	Residual, for DAPhNe MCU comp. Not used
DAC	V_{DAC} line. Connected to DAPhNe STIM
PWM	V_{POL} line. Connected to DAPhNe MCU
VCASE	Device case node– body ground
VELEC	Stimulation electrode node
VBOOST	10V supply line
VLOW	Drain of the current-controlling MOSFET
NC	Not connected. For structural integrity only

6 Software Design

The heart of the device is a small 8-bit microcontroller from STMicroelectronics, STM8L152C6T6. The software on this device was developed in ANSI C99 using IAR Embedded Workbench for STM8 (code-size-limited kickstart edition), in conjunction with ST's debugging tool for their STM8 and STM32 microcontrollers, called ST-LINK/V2. ST's standard peripheral code library for STM8L15x devices was used to create the source code specific to this MCU. An application code library for the M24LR04 NFC memory tag was used to communicate with it using I2C peripheral.

6.1 Software Requirements

The software requirements for this device were as follows:

- The program must cyclically generate pulse timing, pulse magnitude, and pulse polarity signals to control the external hardware of the stimulator.
- The program must respect the patient's need for inspiratory and expiratory phases of the breath cycle.
- The output of the device must have programmable pulse length (PL), pulse frequency (PF), pulse magnitude (PM), inter-pulse resting period (IPR), stimulation-to-recharge scaling ratio (SRS), breathing rate (BR), and inspiratory time (IT). See Figure 2 (in Section 4.3) for visual representation of these variables.
- The device must update its stimulation settings on the very next breath after new settings are communicated to it via the NFC interface, to ensure proper responsiveness.
- The device must remain in the lowest power states as much as possible to conserve the energy supplied to it by a battery unit.

6.2 Power Consumption Considerations

In implantable medical devices, the focus is always on power efficiency. The main consumers of energy in any embedded system are usually wireless modules and microprocessors.

Using passive NFC communication instead of ZigBee, Bluetooth Low Energy, or any other short range wireless technologies eliminates the bulk of power consumption. On another front, however, a very power-conscious microcontroller was chosen. But even when using a power-optimized microprocessor, it is its software that ultimately determines the battery life.

In any microprocessor not furnished with an on-chip wireless module, the next main power consumer is its central processing unit (CPU). In low-power applications, the CPU on the device must be kept in low-power modes as much as possible, and tasks not explicitly requiring CPU activity must be accomplished using other on-chip hardware. This has been the main focus of software development of the DAPhNe Stimulator.

6.3 Software Architecture

The software architecture of the device is essentially a two-state time-based state machine (See Figure 14). The device alternates between "inspiratory mode" (IM) and "expiratory mode" (EM); the transitions between these two states are triggered by the device's real-time clock (RTC) unit – no biological feedback is used as of yet. During IM, the device outputs stimulation and recharge pulses. During EM, the device updates its internal variables based on information read from its NFC memory and then rests until the next IM. CPU is kept inactive for as much time as possible, and RTC is used to wake up the CPU upon exiting the IM or EM.

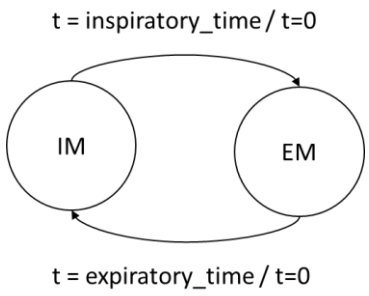


Figure 14: DAPhNe Stimulator State Machine

6.4 Reducing CPU duty cycle

By “daisy-chaining” multiple on-chip hardware peripherals, stimulation control signals are achieved without active involvement of the CPU. Besides quickly checking the NFC memory for new data using I2C during EM and starting the peripheral event chain during IM, the CPU activity is not needed. During periods of computational inactivity, the device is put into one of two low-power modes.

6.4.1 Inspiratory Mode: WFI Mode

At the beginning of IM, the CPU merely updates the RTC counter to an inspiratory time value (basically, sets a wake-up alarm for itself), and starts a hardware timer, TIM1. This timer triggers the chain of peripheral events explained in the sections below.

After updating RTC and TIM1, the CPU issues a Wait-For-Interrupt (WFI) instruction; in WFI mode, the CPU is stopped, while other peripherals are still running. As mentioned, the device “wake up” upon RTC overflow interrupt.

6.4.2 Expiratory Mode: Active Halt Mode

Upon waking up from IM’s WFI mode via an RTC interrupt, the device enters EM. Again, it updates the RTC counter value – this time, to count expiratory time. It disables TIM1 and TIM2 outputs, effectively shutting down the stimulation circuit, checks the NFC memory for new data using I2C protocol, and issues a “Halt” instruction. Because RTC has been enabled prior to the instruction, the device enters Active Halt mode (see device Datasheet for details). In this mode, CPU and all other peripherals except for RTC are stopped. This greatly reduces the device’s power consumption.

6.5 Peripheral Chaining

The firmware uses the following hardware peripherals to accomplish the above cyclic biphasic pulse generation without involvement of CPU:

- TIM1 – advanced 16-bit control timer, used to control pulse timing (V_{TIM})
- TIM2 – general purpose 16-bit timer, used to control pulse polarity (V_{POL})
- TIM4 – basic 8-bit timer, used to trigger digital-to-analog conversions
- DAC – 12-bit digital-to-analog converter, used to control pulse magnitude (V_{MAG})
- DMA – 4-channel Direct Memory Access controller, used to update peripheral memory registers

6.5.1 Peripheral Event Flow

Figure 15 on the next page shows how the hardware peripherals are chained together to produce the biphasic pulses. This chain of events happens during each of the four pulse phases shown in Figure 2.c. Following sections explain Figure 15 in great detail.

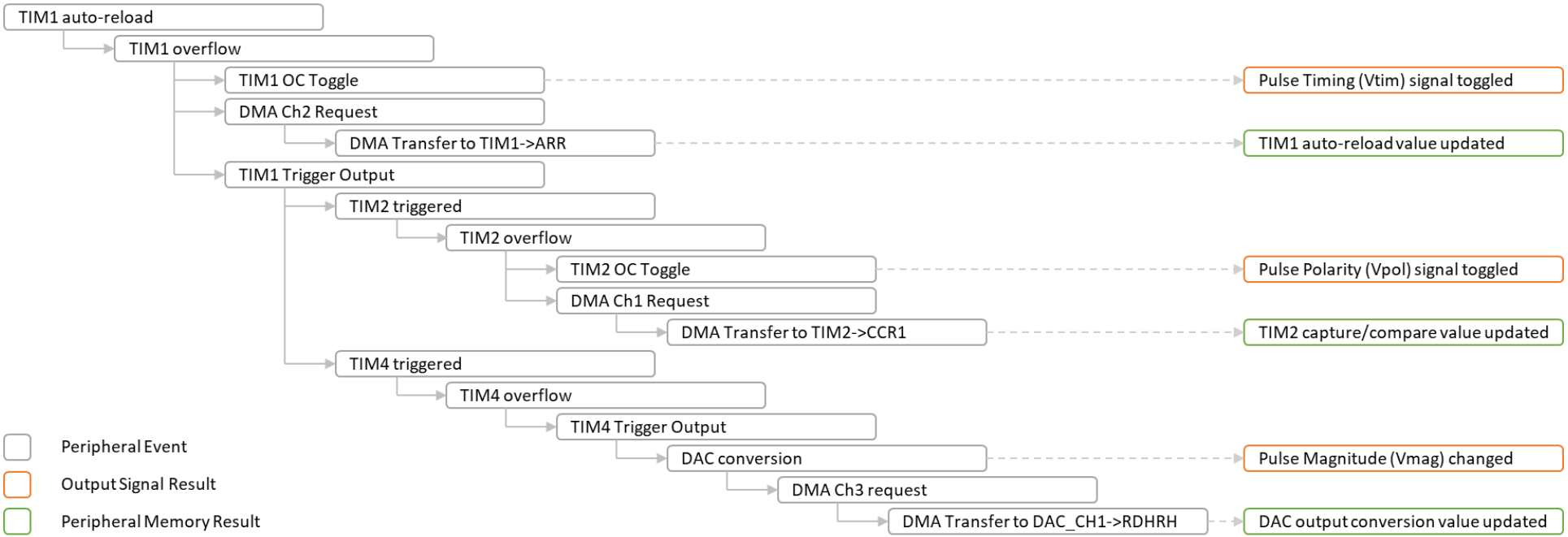


Figure 15: Peripheral Event Chain

6.5.2 TIM1

A new phase starts when TIM1 reaches its auto-reload value. During reload, it resets its counter to 0. The capture compare register of TIM1 is set to 0 permanently, so TIM1 actually overflows on the very next clock cycle. After that, it simply counts up to the auto-reload value.

Upon overflow, TIM1 issues an update event trigger flag. TIM1 toggles its pin output (effectively switching the current magnitude output signal on and off via an external MOSFET). It also issues an update event trigger flag polled by TIM2 and TIM4, and a transfer request on its DMA channel.

Upon request, DMA transfers a value from the device's RAM into the TIM1_ARR (auto-reload) register. In effect, this controls when TIM1 overflows (and the rest of the chain starts over) next. A simple four value buffer, i.e. $\{150,450,1500, (1/PF - (150 + 100 + 1500)\}$ (this example is in units of µs), supplied to the TIM1_ARR register would result in the stimulation circuit being on for 150 µs (stimulation phase), off for 450 µs (inter-pulse phase), on for 1500 µs (recharge phase), and off for the rest of the pulse period. The equivalent signal output from the TIM1 output pin can be seen in Figure 3.b.

Upon being triggered on TIM1 overflow, TIM2 and TIM4 start counting up. They are configured as slaves to TIM1, and are being synchronized with it using update event triggers.

6.5.3 TIM2

On overflow or update event trigger from TIM1, TIM2 issues its own DMA request and toggles its own pin output, switching the polarity of the stimulation signal.

The same way that DMA controller updates the auto-reload register of TIM1, DMA also controls TIM2 timing. In this case, however, DMA transfers a RAM value into TIM2's *capture compare* register instead. When TIM2's counter reaches the capture/compare (CC) value, an overflow event is generated, and a DMA transfer request is issued. It is important to note, that TIM1 and TIM2 are clocked at the same exact frequency.

In this application, polarity of the signal should be switched right after the negative stimulation pulse to positive, and right after the positive stimulation pulse to negative, as shown in Figure 3.c. The software accomplishes this by feeding the DMA TIM2 channel a buffer storing {0xFFFF, 50, 0xFFFF, 50}. When the CC value is larger than TIM1's auto-reload value (i.e. 0xFFFF), TIM2 does not have time to reach overflow, and does not toggles its output. However, the TIM1 trigger event still forces TIM2 to issue a DMA request, updating its CC register to a low value (i.e. 50 clock cycles for stability). The effect of the above is that stimulation polarity switches only during the two resting phases (inter-pulse and rest), and never during the actual pulses.

6.5.4 TIM4

On TIM1 update event, TIM4 is also triggered. Unlike TIM2, TIM4 does not control any output signals by itself; it is simply used as a gateway to trigger DAC conversions and DMA requests. TIM4 is the only trigger source for DMA in medium density STM8L devices.

Upon TIM1 trigger, TIM4 starts counting up. However, its CC register is set to 1, so TIM4 overflows very quickly. It is also configured in "One Pulse" mode, meaning that after overflowing (and issuing an update event), it does not continue counting.

On its quick overflow, TIM4 issues its own update event trigger, which is polled in hardware by DAC.

6.5.5 DAC

On TIM4 update event trigger, DAC performs a conversion of its data register to an analog voltage on its pin output. Upon conversion, DAC issues a DMA transfer request on its configured DMA channel. DMA then transfers a value from RAM to the DAC data register, which is used on the next conversion (on the next phase). By feeding in a buffer like {(high value), (low value), (low value), (high value)}, DAC output assumes the waveform shown in Figure 3.d.

6.5.6 DMA

In short, the device uses three channels of the direct memory access controller to update the TIM1, TIM2, and DAC internal registers, and thus, alter the output signal of the stimulation circuit during each phase of the pulse cycle.

6.5.7 Peripheral Configuration Buffers

Three software buffers are used to pump data into the three peripheral registers using DMA. At run-time, these are stored in RAM after each NFC memory check and settings calculation. CPU starts TIM1 and shuts down; DMA starts *cyclically* loading these RAM buffers into TIM1's auto-reload register, TIM2's capture/compare register, and DAC's data register.

Table 3 shows the three four-value buffers; first value is loaded prior to the stimulation phase, second – during inter-pulse phase, and so on. Effect of the shown values is explained below as well.

Table 3: Peripheral Configuration Buffers

Phase	Stimulation (1)	Inter-Pulse (2)	Recharge Phase (3)	Rest Phase (4)
TIM1* Buffer	PL	IPR	PL*SRS	1/BR-(PL+PL*SRS+IPR)
TIM2* Buffer	0xFFFF	50 clock cycles	0xFFFF	50 clock cycles
DAC Buffer	PM	PM/SRS	PM/SRS	PM

* TIM1 and TIM2 are initialized to low polarity – Logic 0

(1)	Pulse is of high magnitude, negative polarity, and 150 µs in length. Polarity is not switched within this period (pulse stays negative)
(2)	No pulse; polarity is almost instantly switched to positive, and magnitude is switched to "low" in preparation for upcoming re-charge pulse; duration: 450 µs
(3)	Pulse is of low magnitude and positive polarity; duration: 1500 µs. Polarity is not switched within this period (pulse stays positive)
(4)	No pulse; polarity is almost instantly switched to negative, and magnitude is switched to "high" in preparation for upcoming stimulation pulse; duration: the rest of the pulse period

6.6 Safely Ending IM

At the end of IM, depending on the programmed inspiratory time, the pulse phase may be one of the four described above, although optimally, the device should arrive into the "resting" phase (Phase 4) for the duration of EM. When IM ends, the microcontroller starts the RTC counting up do expiratory time and polls the TIM1_ARR register. At each phase switch, it compares TIM1_ARR with the software buffer value for Phase 4 timing. If comparison result is false, it keeps going through the pulse phases until Phase 4 is achieved. At this point, the device continues with EM. This ensures that the last stimulation pulse cycle ends safely in Phase 4 before the device goes into EM.

6.7 NFC Data Exchange

At this point, no specific data protocol has been developed. An external device simply transmits a string of characters containing 5 comma-separated four-digit numbers.

The string is of type {0150,0300,2000,1200,0500}, meaning "150µs PL, 3.00mA PM, 20.00 bpm BR, 1.200s IT, 500µs IPR". PF is hardcoded to 20Hz and SRS is hardcoded to 10, although both can easily be manipulated in the same way as PL, PM, BR, IT, and IPR with minor changes in code.

No backward communication protocol has been designed – reading from the NFC memory tag with an external transceiver simply returns the last communicated values.

As an external transceiver, the prototype uses a receiver board from the M24LR-DISCOVERY kit from ST Microelectronics, which contains the CR95HF-VMD5T transceiver chip. The kit comes with proprietary demo GUI software, which allows for easy data input.