

---

# Contents

<b>1 Patchmaster for Pulse Users</b>	<b>1</b>
1.1 General . . . . .	1
1.2 Major Changes . . . . .	3



---

# 1. Patchmaster for Pulse Users

This section is intended for those users familiar with the PULSE acquisition software. We will quickly summarize the most relevant differences between PULSE and PATCHMASTER. For a detailed description of all functions, please refer to the relevant chapters of the PATCHMASTER reference manual.

## 1.1 General

Although in some respects PATCHMASTER *looks* similar to PULSE, it is a completely new program; following an improved strategy for process handling and programming. Therefore, sometimes PULSE users may miss some typical PULSE behavior of the program until it becomes clear what the benefits of the new features are. Obviously, a substantially increased realm of functions and flexibility comes at some price. In most cases, this price is to set some definitions before using PATCHMASTER in order to customize the program according to the individual needs. Thus, this tutorial tries to explain why some things are different although the old way worked very well.

There are three major changes that have several consequences:

- Support of multiple output and input channels.
- Partial support of parallel task processing, e.g. type text while performing acquisition.
- Removal of implicit functions and full capabilities for task automation.

The increased number of channels requires specifying which channels are used, displayed, analyzed etc. If only one or two channels are to be used

---

(like in PULSE), not too many things have to be adjusted (for details see below).

PATCHMASTER tries to provide at least partial support of parallel task processing. This is useful if one wants to edit some text or change windows while the program is doing something else.

Since PATCHMASTER is a program with full automation capability, all functions have to be capable of being called from a **Macro** or by the **Protocol Editor**. Therefore, all implicit functions and key assignments (as partially used in PULSE) had to be removed. Key assignments are now only done via direct links to buttons in windows and entries in drop-down menus. All these assignments are stored in `keys.mac`. With the default settings of this file, as supplied with the software, most keys behave like they did in PULSE. Some examples:

**Pipette pressure.** In PULSE there were implicit key assignments for setting pipette pressure (e.g., 'S'=suction, 'P'=positive pressure, etc.) They have been removed.

To achieve a similar behavior in PATCHMASTER the user can record macros increasing or decreasing the voltage of the DA channel connected to the pipette pressure controller. The execution of the macros (corresponding macro buttons) can be assigned to keys. Here an example: Let's assume that the pipette pressure controller is connected to DA-3. Start "Macro Recording", press the button "Relative Value" in the amplifier window and then increase the voltage of the DA-3 channel by the desired value (e.g. 0.1 V). Finally, stop macro recording and store the macro at index 10 under the name "Pressure". Then, you create a second macro in which you decrease the voltage by the same value. Store this macro at index 11 under the name "Suction". Now you can assign the key "p" to macro button number 10 and key "s" to macro button 11. Store the new macros and key assignment in your default macro and key files.

**Store pipette resistance.** In PULSE the value of `SealResistance` could be stored into the variable `PipetteResistance` by typing 'W'. Now this function is accomplished by a new button in the **Amplifier Window** (`R.Seal -> R.Pipette`). The key 'W' can now be assigned to this field in order to obtain the same behavior as in PULSE (this is indeed done in the default `keys.mac` settings).

## 1.2 Major Changes

**Patchmaster is a multi-channel acquisition program** PULSE could stimulate up to 1 DA channels and could take data via 2 AD channels. In order to support trigger pulses, up to three separate trigger channels could be defined. These trigger channels, however, could only be used for very simple pulse paradigms. PATCHMASTER supports a (theoretically) unlimited number of input and output channels. Currently 16 input and output channels are supported by PATCHMASTER. This number, however, is limited by the hardware used. Given an Epc10, there are 4 DA channels (1 stim-out and 3 free DA channels on the front panel) plus 16 digital output channels (three are available at the front panel). The number of input channels corresponds to the number of available AD channels, i.e. in most cases 8, plus the number of digital inputs (16 digital inputs).

This extension of the number of channels required a substantial redesign of pulse generation and data acquisition. As a result, independent pulse patterns can be output via the selected DA channels. Since there are no separate trigger channels anymore, short pulses to be used as trigger signals have to be designed with a regular output channel. In order to simplify synchronization of parallel output channels and for subsequent analysis purposes it can be helpful to generate such output patterns with a fixed pulse segment paradigm. For this purpose, the **PulseGenerator** of PATCHMASTER offers the feature **Common Timing**. When this option is selected, the durations only of the segments in the first stimulation channel can be altered – the corresponding segments of all other channels will be treated in the same way.

For later analysis, AD channels have to be logically linked to DA channels. For this purpose, for each input channel such a link has to be specified.

Due to the increased number of DA and AD channels, at several places in the program the channels of interest have to be selected explicitly (e.g., in the **Display** and **Online Analysis**).

**Input channels do not need to have identical sampling intervals** During data acquisition, all input channels are read with an identical Sampling Interval. For storage of the data, however, these input channels can be compressed. A compression factor and mode have to be specified for

each channel. In addition, using the feature of **Virtual Traces**, derivative input channels can be generated from other channels.

**Zeroline subtraction can be performed on any segment** Unlike in PULSE, zeroline subtraction can be performed based on any segment to be specified for each input channel.

**Leak handling** Leak pulses can be generated for individual output channels. Turn **Leak** on, and specify leak parameters as usual. In the linked AD channel, **Leak** also has to be turned on to tell the program to acquire leak signals. In addition, it has to be specified whether and how these leak pulses are to be stored (no storage, store average – like in PULSE, or store all individual leak responses).

**Start segment** Since there are no separate trigger channels anymore, PATCHMASTER does not need the implicit assumption that data storage starts after the first trigger. Instead, data storage starts at the specified **StartSegment** and **StartTime**.

**What's new in Segments?** The segments are arranged a little differently to PULSE. Major changes are that values for **voltage** and **duration** can be replaced by global variables P0, ..., P9). In addition, the logarithmic increment mode can be specified for each incrementing variable separately.

There are no **Conditioning Segments** anymore. Instead, for each segment it can be specified whether or not it is to be stored. **Store=off** largely behaves like the previous **Conditioning Segment**. The implementation and details, however, are different (also see chapter ?? on page ??). Like in PULSE, for **Store=off** segments no P/n leak pulses are generated. Unlike in PULSE, the **Store=off** segments are always explicitly output and sampled. Only after sampling the corresponding data are removed from the traces. The big advantage is that the durations of the **Store=off** segments are now precise like all other segments. In addition, non-constant segments can be used (e.g. conditioning stimulation with a sine wave). For long conditioning segments huge data arrays may be required. Therefore, the user has to make sure to set the maximal number of sample points accordingly (in **Configuration Window**, just limited by the amount of available RAM).

**Global parameters** As already mentioned, in addition to **Holding** and specific values, amplitude and duration of segments can be filled in with global parameters P0, ..., P9. At the time of execution, these parameters

are filled into the pulse patterns. This makes it much easier changing many segments at once when these have identical parameters (e.g. a train of pulses to a given voltage P0) (also see chapter ?? on page ??).

**Sequence Timing** PULSE users may wonder where they could find the sequence timing parameters **Linked Sequence** and **Repeat**. The sequence timing has been removed from the pulse generator in PATCHMASTER. Instead, a much more flexible sequence timing is now provided within the **Protocol Editor** (see below).

**Amplifier adjustments, updates** Like sequence timing, amplifier adjustments such as updates of **C-fast**, **C-slow**, **G-series** etc., have been removed from the **Pulse Generator** and now have to be called from the **Protocol Editor**.

**Protocol Editor** The protocol editor is completely new. It allows for a versatile definition of complex protocols. See reference manual.

**Display** The display has not changed very much. Since there are now many traces to be shown simultaneously, scaling parameters, colors etc., have to be specified for each trace separately. Several features were previously supported as buttons in the **Oscilloscope Window**. These functions have now been moved to the **Display** drop-down menu (e.g. **Subtract ZeroCurrent**). Key assignments to the **Display** menu entries e.g. **ShowLeakPulses** have to be defined individually by the user.

Limitations of **Display** with respect to PULSE:

- No average

**Pipette pressure** Pipette pressure is not supported as an implicit output channel anymore. One now has to use the **I/O Control Window** for setting the corresponding pressure.

**Online Analysis** The online analysis functions have been redesigned completely. As a result, the analysis has become much more flexible and powerful. The immediate consequence for PULSE users is that there is no default analysis anymore. Thus, without definition of online functions (see **Online Analysis**), there will be no online analysis. The major improvements with respect to PULSE are:

- Arbitrary number of analyses and a much greater set of analysis functions.
- Generation of derivative data.
- Analysis results do not have to be shown in the Notebook window.
- Analysis results can be plotted in multiple graphs placed in up to two windows.
- An unlimited number of analysis protocols can be stored.
- Online Analysis can be directly triggered by the incoming data (Analysis method can be specified in the pulse generator).

Temporary limitations of **Online Analysis** with respect to **PULSE**:

- Currently, there is no reference analysis.