Programmable Digital Pulse Generator

Firmware Version 1-00

User Manual

Document version 0, created on Dec-06-2019

Contents

Figure List

Table List

Technical Data

Characteristics

- 24 digital pulse generators, 1 digital oscillator
  resolution: 20 ns, length 32 bit (delay up to 85 s)
- arbitrary configuration by selectable
  32 trigger and 32 output sources
- control of 6 signal switches and 5 digital I/O modules
- 6 monitoring outputs
  connectors: LEMO
  signal level: TTL, log. 0: 0..0.4 V, log. 1: 2.4..5.0 V
  output impedance: 50 Ω
- memory data space: 256 KB
- maximum number of stored configurations: 500

Digital Interface

- USB interface according to USB 2.0 standard
  connector: USB plug type B
  data transfer rate: up to 12 MBit/s (*Full Speed*)
  effective data transfer rate: >100 kBit/s

Description

The pulse controller produces 32 digital signals that can be used to control up to 6 signal switches, 5 I/O modules, or 6 monitor outputs at the controller front panel. The first channel of each digital I/O module can be used as a trigger source.

The device integrates one digital oscillator and 24 digital pulse generators (see Fig. 1). These modules are clocked by 50 MHz, thus provide a time resolution of 20 ns. The maximum pulse delay, pulse width or oscillation period are 2^{32} clock pulses, i.e. about 85 s.

The digital oscillator (module Oscillator in Fig. 1) is a free running multivibrator with a period defined by a 32-bit long integer number. The oscillator can be stopped or started at any time by the control signal Enb. If enabled, it provides at its output a 1-clock (20 ns) wide positive pulse at the end of the programmed period.

The pulse generators (modules Monoflop 0-23 in Fig. 1) are digital monoflops. They are triggered by a rising slope at the trigger input and produce a positive pulse with a specified width (32-bit integer numbers Width0-23) after a specified delay (32-bit integer numbers Delay0-23). The polarity of the trigger signal can be inverted by the control signal Inv0-23. The trigger source is selected by a 32-channel multiplexor (control by a 5-bit integer number Select0-23). As trigger input, any output of the pulse generators (signals Del0-23), output of the oscillator (signal Osc), or the external trigger sources, i.e. the first channels of the I/O modules (signals Trig0-4) can be selected. The trigger input can also be set to 0 or 1. If the level 0 is selected, the particular pulse generator is stopped by the signal Clr. The trigger level 1 can be used to trigger the pulse generator by the software.

The symmetrical architecture of the pulse generators offers a large variability of configurations. The pulse generators can be chained to produce complex pulse sequences. They can be triggered periodically by the internal oscillator, by an external event, or by the software. Unused channels can be disabled.

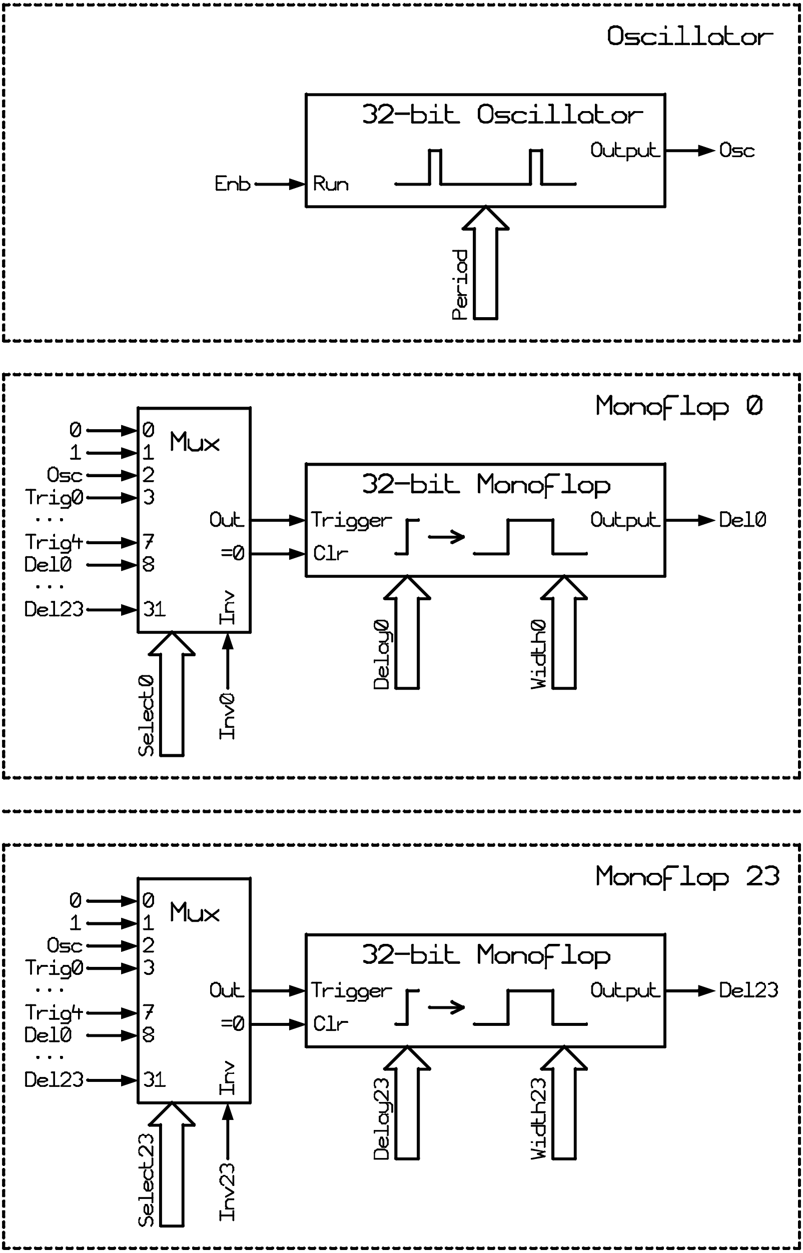

Fig. 1. Pulser Block Diagram.

Each pulse generator is a combination of two coupled non-retriggerables monoflops. The first one with a pulse width defined by one of the numbers Delay0-23 is triggered by the input signal Trigger. When its delay finishes, the second monoflop with a pulse width defined by one of the numbers Width0-23 is launched and the output of the particular module is activated. The minimum delays of the monoflops are 3 clock pulses, i.e. 60 ns.

The pulse controller controls 32 outputs (see Fig. 2). These output signals can control 6 signal switches, 5 I/O modules with 4 outputs each, or 6 monitor outputs at the controller front panel. Each output can be configured in a similar way like the trigger inputs of the pulse generators. The output may be permanently set to 0 or 1, connected to the output of the oscillator (signal Osc), to the external trigger source (signals Trig0-4) or to the pulse generator (signals Del0-23). The polarity of the output signal can be inverted by the control signal Inv0-31.

The configuration of the pulse controller is controlled by software. The current configuration is stored in a non-volatile memory and is automatically restored when the device is started. The user can define and save up to 500 configurations in the non-volatile memory, they can be easily applied by a software command. Beside its number, each configuration can be labeled by a unique name or description text. Using the stored configurations, the pulse controller can be rapidly reconfigured for a new application or a different measurement procedure.

The controller also provides control signals for auxiliary power supplies of the signal switches. The function of these signals cannot be configured. The user can just decide whether the control signals are modulated using a dithering technique to reduce the spectral noise amplitude at the switch outputs.

The pulse controller is equipped with a USB data interface that allows to transfer configuration data to or from the device and remotely control it. The USB data interface is galvanically connected to the device case. When making a connection to a host computer, a large ground loop is created that can influence the performance of the experimental setup.

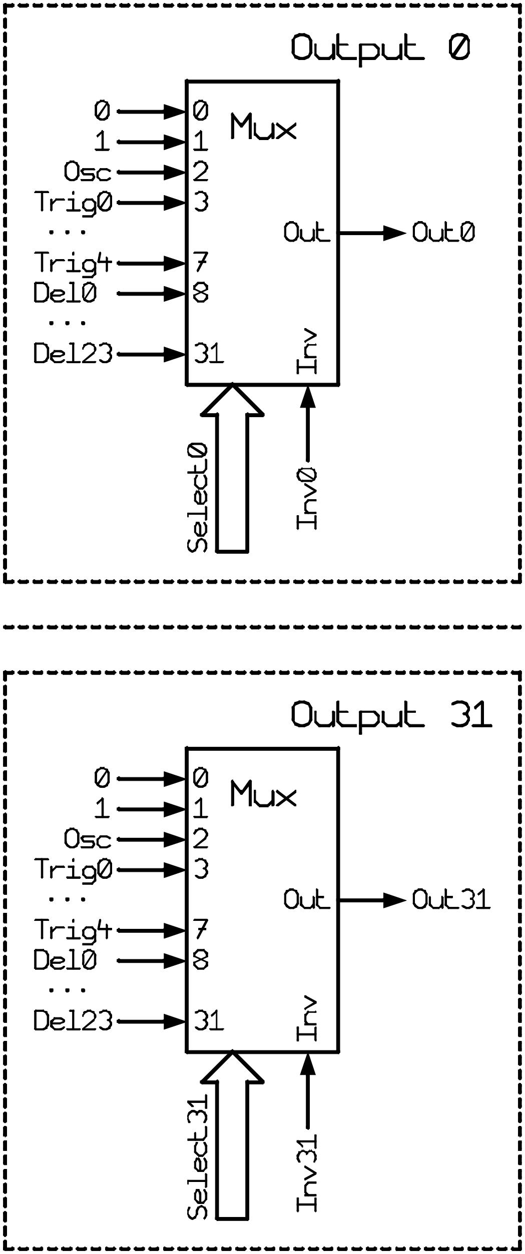

Fig. 2. Output Block Diagram.

The software package for controlling the device contains several utilities for uploading or downloading data (see sections "Software Utilities" and "Backing Up and Restoring the Data") and upgrading the firmware (see the section "Utility FlashLoader").

Quick Setup Guide

The device is delivered with cleared non-volatile memory. To setup the device, follow the next steps:

-
-

If you encounter problems, read carefully the corresponding section in this manual.

Backing Up and Restoring the Data

The data stored in the device's non-volatile memory can be backed up or restored. The data includes all configuration settings. This means that a restore procedure rolls back the device exactly to the state it was in at the backup time. Thus, if any tuning of the settings are planned, it is advisable to create a backup before with which the original state can be restored.

To perform a data backup, locate the program `PulseLoader.exe` in the directory "Program" of the enclosed software package. To back up the system memory into a data file `Memory.txt`, execute the following command in a Windows™ command shell[†]:

```
PulseLoader 6 -y MemoryData.txt -t
```

This command downloads the memory data from the device connected to the virtual port COM6 into the file `Memory.txt`.

To restore the data, execute the following command:

```
PulseLoader 6 -Y MemoryData.txt -t
```

This command uploads the memory data from the file `Memory.txt` to the device connected to the virtual port COM6.

[†] Select "Run" in the start menu of Windows™ and type "cmd". Then change the directory to that with the program files using the command "cd". Finally, execute the given command by copy & pasting and pressing "Enter". A better and more comfortable alternative to the Windows™ command shell are utilities like "File Commander/W" or "File and archive manager (FAR)". Please use the search utilities to find out how to obtain these applications.

Software Utilities

The software utilities can be found in the directory "Program" of the enclosed software package. Before using them, the virtual USB port driver must be installed (see sections "Driver Installation"). The utilities do not require any additional installation, you need only to copy them to a suitable directory on your computer. Before starting the utilities, you need to obtain the virtual port number, as described in the section "Driver Installation".

Utility PulseLoader

The PulseLoader is a simple Windows™ program that runs in text mode. It enables you to backup and restore data of the pulse controller. Launching the utility `PulseLoader.exe` without any parameters or with the parameter `-?` displays a simple help:

```
PulseLoader -?
```

To start the program without any error message, at least the number of the COM port must be given:

```
PulseLoader 6
```

This command starts the utility PulseLoader and assumes that the device is connected to the port COM6. On success, the utility reports the following message:

```
Press '?' for help
```

and waits for command input.

In case of any problem, check whether the port number matches the system settings (see sections "Driver Installation") and the connected device is powered on and working properly. If an error occurs, please consult the section "Error Codes". The tables 2 and 3 explain the possible error messages; they should help you to localize the reason for the software failure.

To check the communication, press the key 'p' to obtain the product identification text. The device should respond as follows:

```
Product identification: PULS-CTRL24 1-00
```

Tab. 1. Command line parameters of the program PulseLoader.

Parameter	Explanation
`-y FileName`	backup memory data into a text file with the name `FileName`
`-Y FileName`	restore memory data from a text file with the name `FileName`
`-s, -S`	get the device status
`-k, -K`	simulate the keypad
`-d, -D`	get the device firmware date
`-v, -V`	get the device firmware version
`-p, -P`	get the product identification text
`-n, -N`	get the product number
`-u, -U`	get the device uptime
`-b, -B`	get the device buffer status
`-z, -Z`	purge the communication
`-t, -T`	terminate the program
`-q, -Q`	quiet mode

`-g`
`-G`
`-?`
	debug mode
	debug mode with output into `Debug.txt`
	show the online help

If the device responds properly, you may try other program commands. Press '?' to obtain the help listing of all available commands[‡].

In practice, you may prefer to use the command line mode instead of the interactive mode. The former mode allows you, for instance, to save the complete commands in batch files for repeated usage.

Table 1 summarizes all allowable command line parameters of the program PulseLoader. The parameters are processed from left to

[‡] Note that keyboard layouts different to the US one may cause issues when evaluating several characters. We recommend to switch to the US layout when using the utility PulseLoader in the interactive mode.

right. If an error in the command line is encountered, the program stops with an error text showing the allowed values of the parameters.

If the parameter `-t` is found, the program stops without processing any following parameter. If you do not specify the parameter `-t` at all, the program does not stop and enters the interactive mode after having processed the complete command line.

If you wish to specify a name parameter containing spaces or special characters, use the conventions valid for your operating system. In Windows™ systems, for instance, enclose the name in quotation marks.

The quiet program mode reduces the program text output, contrary to that, the debug mode provides a detailed output for error analysis.

Tips

To backup the complete device memory, execute the command:

```
PulseLoader 6 -y MemoryData.txt -t
```

The text file `MemoryData.txt` will contain the memory data in hexadecimal format. To restore the device memory, use the command:

```
PulseLoader 6 -Y MemoryData.txt -t
```

It restores the data immediately without any confirmation, thus be careful when executing it.

Note also that the backup and restore needs several minutes to finish.

Utility FlashLoader

The FlashLoader is a simple Windows™ program running in text mode. It enables you to upgrade the firmware of the pulse controller. You should perform the upgrade if you have received or downloaded a new firmware file from the device manufacturer. Launching the utility `FlashLoader.exe` without any parameters displays a simple help text with the expected syntax of the command line.

Before upgrading the firmware, you should first test the device and the communication by verifying the current firmware version. To do so, start the following command:

```
FlashLoader 6 Firmware.txt -v
```

where `Firmware.txt` is the file containing the current firmware and the number 6 indicates the port COM6 to which the device is connected. The program should produce the following output:

```
Code file R:Final.txt from 11/04/2019, 12:00:00
Flash Loader 1.12
Verifying code file Final.txt
Verifying finished at Tue, 11/05/2019, 15:00:00
100495 (1888Fh) bytes processed, 99584 (18500h)
bytes verified
Resetting the target
Program finished ok
```

During the verify procedure, a message box is displayed at the device display informing the user that the flash loader has been activated. When the verify finishes without any error, the device is restarted.

! **Attention:** To be sure that the device cannot activate the attached switches when the FlashLoader is active, disconnect the output cables of the switches or remove these plugin modules from the chassis.

If any error occurs, do not proceed with the firmware upgrade. If you cannot resolve the issues, contact the manufacturer. Note that even if the verify fails and the flash loader at the device remains active, it is safe to power the device off to restart it. However, a more safe and comfortable alternative to that is to execute the following command:

```
FlashLoader 6 -i -f
```

This prevents the utility at the host computer from initializing the flash loader utility at the microcontroller again and sends the reset command to the device.

If the verify has succeeded, you may start the firmware upgrade by entering the command:

```
FlashLoader 6 Firmware.txt
```

where `Firmware.txt` is the file with the new firmware. The program should produce the following output:

```
Code file R:Final.txt from 11/04/2019, 12:00:00
Flash Loader 1.12
Programming code file Final.txt
Programming finished at Tue, 11/05/2019, 15:00:00
```

```
100495 (1888Fh) bytes processed, 99584 (18500h)
bytes programmed
Resetting the target
Program finished ok
```

During the programming procedure, a message box is displayed at the device display. When the programming finishes, the device is restarted with the new firmware.

If an error occurs, the flash loader utility at the microcontroller may remain active. This is the case if the message box at the device display is still present and the device did not restart. In this case, you may retry the action with the command line parameter '-i':

```
FlashLoader 6 Firmware.txt -i
```

This will prevent the utility at the host computer from initializing the flash loader utility at the microcontroller again and it will just try to reprogram the file `Firmware.txt`. If the error persists, contact the manufacturer.

! **Attention:** You must not power down the device if the firmware upgrade did not succeed. Otherwise, the device will not operate properly or might even not restart at all. Did this happen, it would be necessary to reprogram the device in the factory.

If the current firmware is damaged so that the device is inoperable, you may try to start the flash loader utility at the microcontroller manually. Press, simultaneously, the horizontal direction keys (left and right) and power on the device. If this small part of the firmware is still working, the device's flash loader will start. Then, try to launch the utility FlashLoader with the command line parameter '-s':

```
FlashLoader 6 Firmware.txt -s
```

This does not start the flash loader utility at the microcontroller but will only try to reprogram the file `Firmware.txt`. If an error occurs that you cannot solve, contact the manufacturer.

Error Codes

Tab. 2. Return values of the interface functions

Return value	Error message	Description
0	No error	The data transfer finished successfully.
-2	Error opening the port	The port could not be opened. For the possible reasons, see Tab. 3.
-3	Error closing the port	The port could not be closed. For the possible reasons, see Tab. 3.
-4	Error purging the port	The port buffers could not be cleared.
-5	Error setting the port control lines	The port control lines could not be set.
-6	Error reading the port status lines	The port status lines could not be read.
-7	Error sending command	The data transfer to the device failed. For the possible reasons, see Tab. 3.
-8	Error sending data	
-9	Error sending termination character	
-10	Error receiving command	The data transfer from the device failed. For the possible reasons, see Tab. 3.
-11	Error receiving data	
-12	Error receiving termination character	
-13	Wrong command received	The device sent an unexpected response.
-14	Wrong argument received	
-15	Wrong argument passed to the function	One of the arguments passed to the function was out of the allowable range.

Return value	Error message	Description
-100	Device not connected	The port status lines indicate that the device is not connected.
-101	Device not ready	The port status lines indicate that the device is not ready. The communication with the device is possible only if it does not execute any process. Terminate all dialog boxes and menus at the device and retry the operation.
-102	Device state could not be set to not ready	The device did not react properly. Try to reset the communication or restart the device by powering it off and on.
-400	Error opening the file for debugging output	The file for debugging output cannot be opened for writing. Check if you have permissions to perform this action or if the file exists and is opened by another application.
-401	Error closing the file for debugging output	The file for debugging output cannot be closed. Check if the access to the file is still possible.

Tab. 3. I/O errors

Return value	Error message	Description
0	No error	The data transfer finished successfully.
1	Port has not been opened yet	You attempted to use the communication channel before having opened it.
2	Cannot open the port	The specified port could not be opened. Either the port does not exist or it is being currently used by another program.
3	Cannot get the state of the port	The system could not get the state of the port.
4	Cannot set the state of the port	The system could not set the state of the port.
5	Cannot set the timeouts for the port	The system could not set the timeouts for the port.
6	Cannot clear the port	The system could not clear the port buffers.
7	Error reading data from the port	The system could not read data from the port. Most probably, no data is available because the device is either disconnected or does not respond.
8	Error writing data to the port	The system could not write data to the port.
9	Wrong data amount written to the port	The system could not write the proper data amount to the port.
10	Error setting the control lines of the port	The system could not set the state of the port control lines.
11	Error reading the status lines of the port	The system could not get the state of the port status lines.
12	Device is busy	The system could not access the device since menus or dialog boxes are active.

Driver Installation

Installation of the Virtual Port for the USB Interface

The virtual port driver is required for the operation of the device with a USB interface. If you use the operating system Windows™, please note the following:

- Please use the update function of the operating system at the host computer or download the most recent driver from the homepage of the manufacturer of the USB adapter. The drivers are located at the following address: http://www.ftdichip.com/Drivers/VCP.htm. Please choose the correct driver version according to your operating system.
- To install the driver, administrative rights are required.
- The installation is described in detail in the "Installation Guides" available at the abovementioned address. Please read this description carefully before starting the installation.
- After the installation, the number of the virtual port can be set. You can change the settings in the device manager by opening the settings of the device *USB Serial Port (COMx)*. To modify the settings, administrative rights are required. The settings are applied immediately, you do not need to reboot the PC to activate them.

The software can also be used at computers running the Linux operating system. You can run them using the Windows™ emulator wine (see http://www.winehq.org/).

Starting with Linux Kernel 3.0.0-19, all FTDI devices are already supported without the necessity of compiling additional kernel modules. For more details, consult the homepage of the manufacturer of the USB adapter: http://www.ftdichip.com/Drivers/VCP.htm.

The system has to be configured an the following way:

- Use, for instance, the program 'dmesg' to find out to which USB port the device is attached: Look for a line similar to "FTDI USB Serial Device converter now attached to ttyUSB0"
- Link the Linux device to the virtual COM port of wine:
  ```
  ln -s /dev/ttyUSB0 ~ /.wine/dosdevices/com3
  ```
  This assumes that the device is attached to ttyUSB0 and will be linked with COM3