8-Channel Pulse Generator x2

(Release -)

|  |  |
| --- | --- |
| **Prepared for the laboratory of**  **Dr. Marta Zlatic**  **Group Leader**  **Email:** [zlaticm@janelia.hhmi.org](mailto:zlaticm@janelia.hhmi.org)  **Voice:** 571.209.4197 | **Principle Engineers**  **Lakshminarayanan Ramasamy, PhD**  **Engineering Specialist II, Electrical Engineer**  **Email:** [ramasamyl@janelia.hhmi.org](mailto:ramasamyl@janelia.hhmi.org)  **Voice:** 571.209.4361 |

# Revision Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rev.** | **Date** | **Engineer** | **Lab** | **Comments** |
| 0 | 05/05/2015 | Lakshminarayanan Ramasamy, PhD | ID & F | Draft |

# What does this pulse generator do?

8-hannel pulse generator x2 is a digital pulse generator (version: Release-) that can generate digital pulses independently on 8 channels with a timing accuracy of ±200 µs. Timing diagram is shown in Figure 1. USER can define the timing parameters shown in Figure 1 via PC. Pulse generator communicates with PC through serial interface protocol.

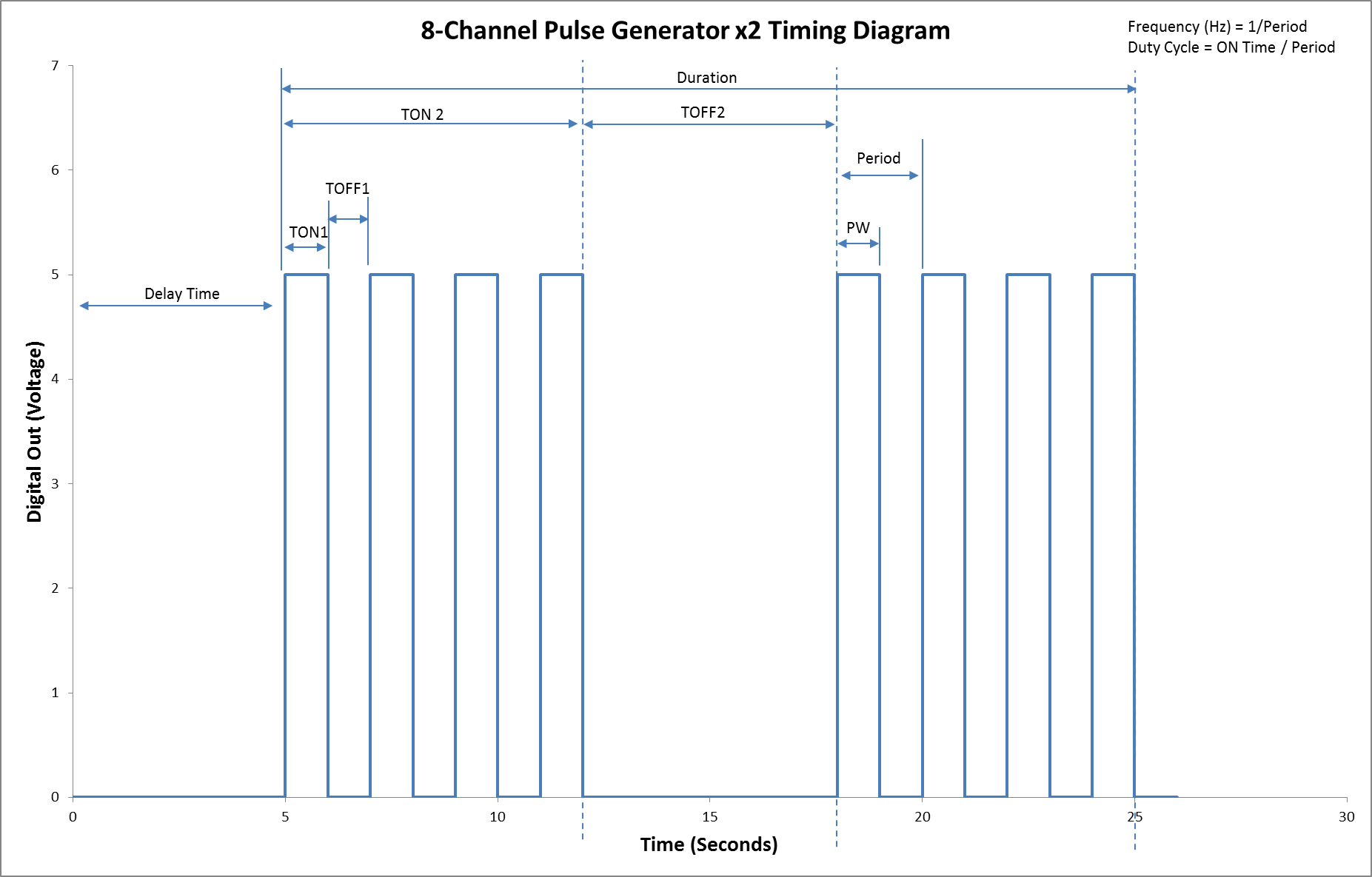


Figure 1 Timing diagram

# Hardware Description

8-channel pulse generator x2 is designed using Teensy 3.1 development board. Figure 2 & 3 depicts the front and back panel. USB interface on the front panel is used for PC interface. 8 BNC connectors 1 to 8 on the back panel are connected to PIN 14 to 21 respectively. Teensy-3.1 development board is shown in Figure 4.



Figure 2 8-channel pulse generator x2 front panel



Figure 3 8-channel pulse generator x2 back panel

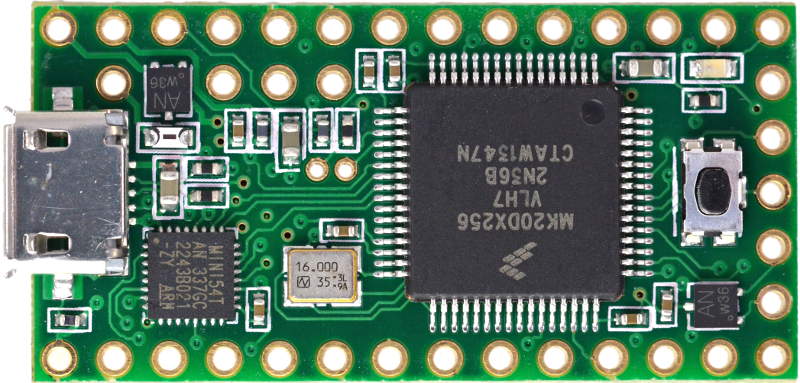


Figure 4 Teensy 3.1

# Wring Instruction

PIN 14 to 21 are wired to center pin of BNC connectors 1 to 8 respectively. All ground pins of BNC connectors are connected to GND pin on Teensy3.1 board.

# Application Program Interface

Pulse generator communicates with a HOST computer via serial protocol. Settings for the serial protocol are as below:

Baud rate: 115200

Signal DTR: Disabled (On certain microcontrollers, DTR signal line is tied to RESET pin. So, it is recommended to turn this signal off)

NOTE: USER must identify the COM port number from windows device manager.

Every command has the following list of components:

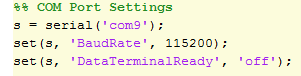
1. Protocol ID: Datatype = 8-bit integer constant. Value = 255. Length = 1 byte.
2. Command ID: Data type = ASCII character. Value = a to z. Length = 1 byte.
3. Channel ID: Data type = ASCII character. Value = 0 to 7. Length = 1byte.
4. Parameter: Data type = ASCII character. Value = 0 to 9. Length = varies depending on the command. For command “start” and “stop”, no parameters are needed. For command “polarity”, one byte length parameter is sent. For rest of the commands, 6 byte length parameter value is sent.

Table 1 Full Command List

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Header** | **Parameter** | **Comments** |
| ON time  (TON1) | <255><’a’><’n’> | <’000000’> | ASCII letter ‘a’ is the command identifier for setting ON time (TON1). ‘n’ is a ASCII character ‘0’ to ‘7’ and defines the channel number. 6 byte long parameter defines the ON time in units of milliseconds. |
| OFF time  (TOFF1) | <255><’b’><’n’> | <’000000’> | ASCII letter ‘b’ is the command identifier for setting OFF time (TOFF1). ‘n’ is a ASCII character ‘0’ to ‘7’ and defines the channel number. 6 byte long parameter defines the OFF time in units of milliseconds. |
| ON time  (TON2) | <255><’c’><’n’> | <’000000’> | ASCII letter ‘a’ is the command identifier for setting ON time (TON2). ‘n’ is a ASCII character ‘0’ to ‘7’ and defines the channel number. 6 byte long parameter defines the ON time in units of milliseconds. |
| OFF time  (TOFF2) | <255><’d’><’n’> | <’000000’> | ASCII letter ‘b’ is the command identifier for setting OFF time (TOFF2). ‘n’ is a ASCII character ‘0’ to ‘7’ and defines the channel number. 6 byte long parameter defines the OFF time in units of milliseconds. |
| Duration | <255><’e’><’n’> | <’000000’> | ASCII letter ‘c’ is the command identifier for setting Duration. ‘n’ is a ASCII character ‘0’ to ‘7’ and defines the channel number. 6 byte long parameter defines the Duration in units of milliseconds. |
| Delay Time | <255><’f’><’n’> | <’000000’> | ASCII letter ‘f’ is the command identifier for setting Delay Time. ‘n’ is a ASCII character ‘0’ to ‘7’ and defines the channel number. 6 byte long parameter defines the Delay Time in units of milliseconds. |
| Polarity | <255><’g’><’n’> | <’0’> or <’1’> | ASCII letter ‘g’ is the command identifier for Polarity setting. 1 byte long parameter ‘0’ or ‘1’ sets output polarity to active high or active low respectively. |
| Start | <255><’s’><’n’> | NONE | ASCII letter ‘s’ is the command identifier for Start command. It starts the pulse generation on channel ‘n’. |
| Stop | <255><’t’><’n’> | NONE | ASCII letter ‘t’ is the command identifier for Stop command. It stops the pulse generation on channel ‘n’. |

**Example MATLAB program:**

1. Initializing COM port



1. Setting channel 0 ON time (TON1) to 10 milliseconds



1. Setting channel 0 OFF time (TOFF1) to 10 milliseconds



1. Setting channel 0 ON time (TON2) to 100 milliseconds



1. Setting channel 0 OFF time (TOFF2) to 100 milliseconds



1. Set Duration to 300 milliseconds



1. Set Delay time to 10 milliseconds



1. Invert polarity of channel 0 to active low



1. Start pulse generation on channel 0



The measured output of the above example MATLAB code is shown in Figure 5.

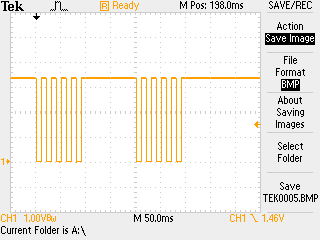


Figure 5 Measured output waveform

# Source code

Arduino source code is embedded below.

