

# PulseDecci

2 channel pulse generator with an optional trigger  
input for Raspberry Pi

Hajime Hirase, RIKEN Brain Science Institute  
hirase@brain.riken.jp

# Caution

- THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- PulseDecci is a python3 program running on a Raspbian OS (a linux). Therefore, it is not suitable for timing-critical real-time applications.
- Both trigger input and channel output uses CMOS logic (i.e. 0 and 3.3 V)

# Installation

- Raspbian Jessie OS is assumed, in which the user pi has access to GPIOs.
- Install the source code (Getch.py HPulseDecci.py PulseDecci.py) in /home/pi/PulseDecci
- PulseDecci should be executable by pi user. If not, run (as user pi)

```
chmod u+x /home/pi/PulseDecci/PulseDecci.py
```

- For convenience, make a symbolic link

```
sudo ln -s /home/pi/PulseDecci/PulseDecci.py /usr/local/bin/pulsedecci
```

- You can run PulseDecci as pulsedecci at command prompt.
- It probably runs better if your raspi does not run GUIs (i.e. X).

```
sudo raspi-config
```

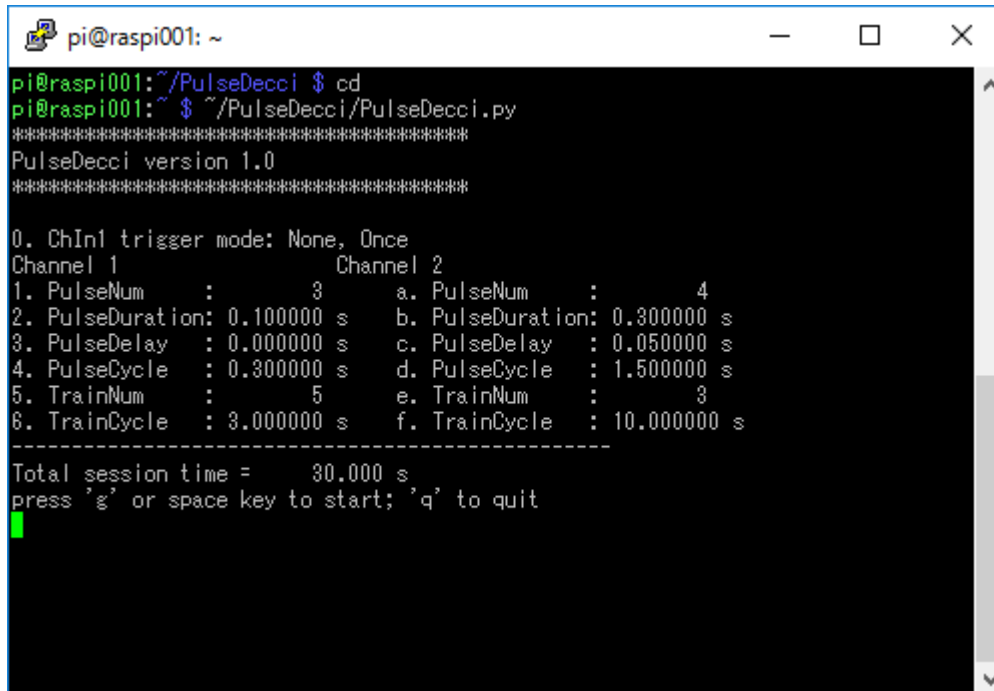
```
Boot Options -> Console
```

# Execution

Assuming the pi user, type “~/PulseDecci/PulseDecci.py” (without quotation marks) and hit an enter key at command prompt.

Alternatively, “pulsedecci” will start the program if you installed the symbolic link (see Installation in the previous page)

After execution, a menu screen should appear.



```
pi@raspi001: ~  
pi@raspi001:~/PulseDecci $ cd  
pi@raspi001:~/PulseDecci $ ./PulseDecci.py  
*****  
PulseDecci version 1.0  
*****  
0. ChIn1 trigger mode: None, Once  
Channel 1          Channel 2  
1. PulseNum       :      3      a. PulseNum       :      4  
2. PulseDuration : 0.100000 s  b. PulseDuration : 0.300000 s  
3. PulseDelay    : 0.000000 s  c. PulseDelay    : 0.050000 s  
4. PulseCycle    : 0.300000 s  d. PulseCycle    : 1.500000 s  
5. TrainNum      :      5      e. TrainNum      :      3  
6. TrainCycle    : 3.000000 s  f. TrainCycle    : 10.000000 s  
-----  
Total session time =      30.000 s  
press 'g' or space key to start; 'q' to quit  
█
```

# Channel definition (default)

All logics are cMOS (i.e. 0 or 3.3V)

ChIn1 Trigger input (pulldown)

ChOut1

ChOut2

3.3V	1	2	5V
GPIO 2 (I2C1_SDA)	3	4	5V
GPIO 3 (I2C1_SCL)	5	6	GND
GPIO 4 (GPCLK0)	7	8	GPIO 14 (UART_TXD)
GND	9	10	GPIO 15 (UART_RXD)
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
GPIO 10 (SPI_MOSI)	19	20	GND
GPIO 9 (SPI_MISO)	21	22	GPIO 25
GPIO 11 (SPI_SCLK)	23	24	GPIO 8 (SPI_CE0)
GND	25	26	GPIO 7 (SPI_CE1)
ID_SD	27	28	ID_SC
GPIO 5	29	30	GND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GND
GPIO 19	35	36	GPIO 16
GPIO 26	37	37	GPIO 20
GND	39	40	GPIO 21

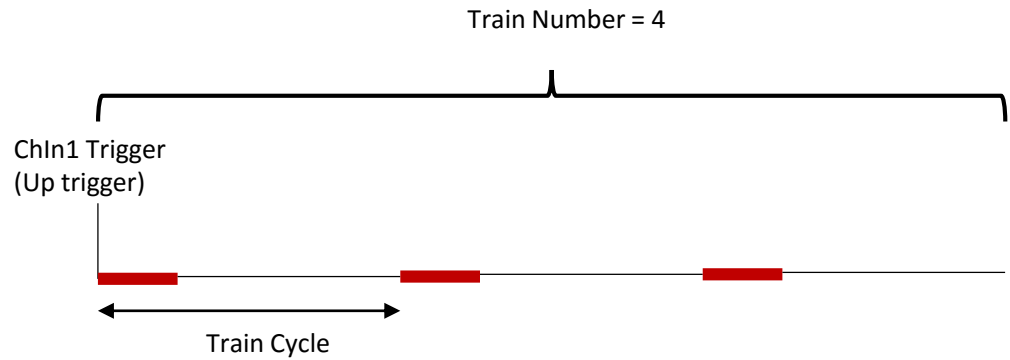
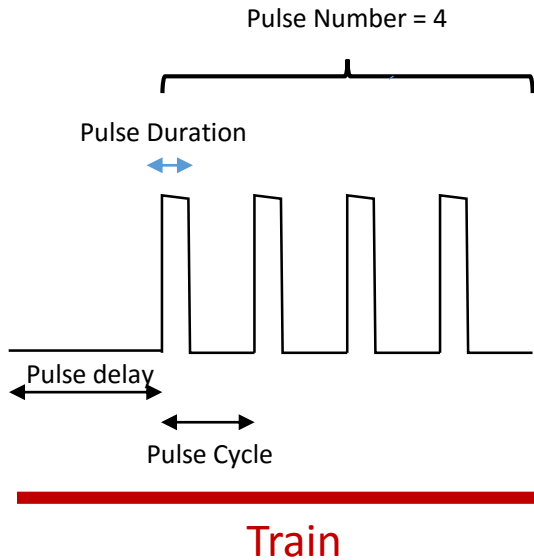
Key

- Power (5 Volts)
- Power (3.3 Volts)
- Ground
- General Inputs/Outputs
- I2C Interface
- SPI Interface
- UART Interface
- ID EEPROM Interface

Pin 1



# Pulse train parameter description



# Operation modes

After start key ('g') is pressed...

- Trigger mode: None, Once

The programmed pulse train is produced once and returns to menu.

- Trigger mode: None, Loop

The programmed pulse train is produced repeated until a keyboard interrupt (Ctrl-C) is made. Repeated keyboard interrupts will terminate the program to linux command prompt.

- Trigger mode: Trig, Once

Ch1Input triggers the programmed pulse train and returns to menu.

- Trigger mode: Trig, Loop

Ch1Input triggers the programmed pulse train and the train is repeated. until a keyboard interrupt (Ctrl-C) is made. Repeated keyboard interrupts will terminate the program to linux command prompt.

# Frequently asked questions

Q. How do I change the default pulse train parameters?

A. Edit HPulseDecci.py parameter assignments for ChOut1PulseNum, ChOut1PulseDuration, ... etc. that comes in the beginning of the file. (Lines 30-50 ish).

Q. Can I assign different GPIOs for input and/or output channels?

A. Yes, edit ChIn1, ChOut1, and ChOut2 assignments in HPulseDecci.py. Note that naming convention is the Broadcom GPIO numbers (BCM), not pin numbers.

Q. Do I need to step-up the output voltage to 5V to trigger TTL devices?

A. Most probably not. 3.3V is usually enough to trigger modern commercial TTL devices.

Q. Do I need to step-down the input voltage from 5V to 3.3V?

A. Yes. A voltage divider consisting of 2.2 kOhm and 3.3 kOhm resistors will do.

\* For level conversion, you can probably use bidirectional level converters like MAX3000 series (MAXIM).

Q. Is InputCh1 pulled down?

A. Yes. (PulseDecci.py Line 30: `GPIO.setup(HPulseDecci.ChIn1,GPIO.IN, pull_up_down=GPIO.PUD_DOWN)`)



# Acknowledgments and other notes

Donation to RIKEN is always welcome

<http://www.riken.jp/en/about/support/>

(日本語) <http://www.riken.jp/about/support/>