

Programmable pulse generator for Arduino.

The sketch will be build for Arduino NANO

The Programmable Pulse Generate will generate pulses on a number of output pins.

Programming

The Programmable Pulse Generate uses a REST approach over the serial connection (see below).

Programmeable parameters:

- Number of output pins (with a maximum of 10).
- Puls length in millis for each channel.
- Puls period in millis for each channel.[1]
- The pause between pulses in each channel "Pulse Order Interval".
- Relation between pulses on the various channels (pulseOrder).
 - Individual 'I' Pulses starts Simultaneously, but act upon their own pulse length and interval. (Default)
 - Simultaneously 'S' All pulses starts a the same time. Pulse interval is redefined to be the lognest pulse length plus the longest pulse interval.
 - Sequential 'Q' Pulses starts after another), with a programmable pause between the pulses (Pulse Order Interval).
 - Randum 'R' Pulses starts randumly om each channel and follow their own length and interval.
- The pulse length on each channel can be selected as LOW pulse or HIGH pulse.
- Number of pulses fired on each channel. Initiated by a push button function attached to pin A5 (19). When the push button is pressed once, the pulse generator stops. When the push button is pressed twice, the pulse generator restarts, at fires the configured pulses on each channel. When the push button is pressed once, after it has been pushed twice, the pusle generator restarts and runs infinitely.

Firmware Version histoty:

- 1.2.4 - Bugfix: Cannot set pulseLength. [if (value = 0 ... changed to if (value >= 0 ...] Value would never be zero:-)
- 1.2.3 - Bugfix in Sequential run and documentatino update
- 1.2.2 - Parameter for programmable delay after "Run Infinite" to "Run Limited"
- 1.2.1 - Issues regarding configurations being changed, when Puls Order is changed from I to Q, S or R. handleRest() si change to reload the stored

configuration, before adding changes.

- 1.2.0 - Configuration bugfixes
- 1.1.2 - Minor documentation corrections - No functional changes
- 1.1.0 - Adding push button functionality, which will fire a configured number of pulses on each channel.
- 1.0.0 - Prod. version.
- 0.2.0 - Versions 0.2.x is for developing Programmable interface. - Taking the REST approach for programming parameters.
- 0.1.1 - Building in functionality for handling different "pulse orders". - pinState re-defined to match ACTIVE / PASSIVE pulse instead of LOW / HIGH.
- 0.1.0 - Initial commit: Basic functionalities are build in. No parameters are programmable, but the required program structure is in place.

REST Approach

- Communication will be over serial connection (USB).
- When not connected, the sketch will start, using the last programmed configuration.

/configurations/{parameter}/{id}/{value} or /configurations/{parameter}/{value}

```
<numberOfChannels>  /<1 - 10>
<pulseOrder>         /<I, S, Q or R>
<pulseOrderInterval> /<milliseconds>
<pulseLength>        /<channel 1-10>/<milliseconds>
<pulsePeriod>        /<channel 1-10>/<milliseconds>
<pulseActive>        /<channel 1-10>/<LOW or HIGH>
<numberOfPulses>    /<channel 1-10>/<1 - 100000>
setDefault
```

Configuration examples:

- configurations/numberOfChannels/3
- configurations/pulseOrder/S
- configurations/pulseOrderInterval/250
- configurations/pulseLength/1/100
- configurations/pulsePeriod/1/500[1]
- configurations/pulseActive/1/HIGH
- configurations/numberOfPulses/1/1000
- configurations/setDefaults

Ardhino nano Pin definition/layout:***

- Pin 0: Serial RX
- Pin 1: Serial TX
- Pin 2: Channel pin 1:
- Pin 3: Channel pin 2:
- Pin 4: Channel pin 3:
- Pin 5: Channel pin 4:
- Pin 6: Channel pin 5:
- Pin 7: Channel pin 6:
- Pin 8: Channel pin 7:
- Pin 10: Channel pin 9:
- Pin 11: Channel pin 10:
- Pin 12: Channel pin 11:
- Pin 13: LED_BUILTIN.
- Pin 14: (A0): RANDUM_SEED_PIN Used for randomSeed() to generate different seed numbers each time the sketch runs.
- Pin 15: (A1): N/C
- Pin 16: (A2): N/C
- Pin 17: (A3): N/C
- Pin 18: (A4): N/C
- Pin 19: (A5): Button pin. Button to running in non continuoogious mode (file configured numer of pulses)

Project eligibility.

The sketch was build for simulating the pulses on the open collector outputs on a number of Carlo Gavazzi energy meters Type EM23 DIN and Type EM111. The project "EnergyRegistgration", which are build to register pulses on the open collector output on the above mentined meters has a problem, counting the pulses corectly. Some pulses are rigistered but not counted and some are counted too many. This project is to investigate into this problem.

1. Note: If pulseOrder is set to 'Q', puls period on channel 1 will define the pause from the last puls in the sequence to the first starts.

Project Version History:

2.0.0 - Hardware merged with Firmware (initial product version)