s0-meter - Der Zähler für S0-Impulse an den RPi-GPIOs

Status

Funktion

Das Programm s0-meter registriert alle an den configurierten RPi-GPIOs auftretenden Pegeländerungen, speichert diese intern in einer Liste und gibt diese über ein REST-API per HTTP-Request im JSON-Format aus.

Das Programm zeichnet alle registrierten Pegeländerungen auf. Eine eventuell notwendige Entprellung der Pulse muss der Client-Applikation erfolgen.

Installation

Das Programm s0-meter kann in ein beliebiges Verzeichnis abgelegt werden.

Konfiguration

GPIOs

Die Konfiguration der S0-Kanäle und der zugehörigen RPi-GPIOs erfolgt in einer Konfigurationsdatei mit folgendem Format:

```
[[channel]]
    id = 1
    gpio = 17

[[channel]]
    id = 2
    gpio = 12
```

Für jeden Kanal wird die Kanal-ID und die Nummer des RPi-GPIO angegeben. Die Kanal-IDs sind frei wählbar und müssen nicht fortlaufen sein. Für die RPi-GPIOs muss die Nummer angegeben werden, die am RPi-Pfostenstecker steht.

Standardmäßig wird die Konfigurationsdatei `/etc/s0-meter.cfg` gelesen. Mit dem Kommandozeilenparameter `-c` kann eine andere Datei angegeben werden.

Kommandzeilenargumente

Die möglichen Kommandozeilenargumente können wie folgt ermittelt werden:

```
$ ./s0-meter --help
s0-meter 0.1.0
Harald Kube <harald.kube@gmx.de>
Listen for S0 pulses at the given GPIO pins

USAGE:
    s0-meter [OPTIONS]

FLAGS:
    -h, --help       Prints help information
    -V, --version    Prints version information

OPTIONS:
    -c, --config <config>    The name of the config file (default:
/etc/s0_logger.cfg)
```

Beenden des Programms

Das Programm kann beeendet werden indem z. B. das Signal TERM geschickt wird.

```
# kill -TERM <PID des Programms>
```

REST-API

Zugang

Das REST-API ist nur vom localhost unter Port 6310 erreichbar.

Format

Über das HTTP-Protokoll lassen sich vom REST-API folgende Informationen auslesen:

Abfrage der Kanal-IDs:

```
# curl http://localhost:6310/v1/channels
```

Antwort:

```
{
   "channels" : [
      1,
      2
   ]
}
```

Abfrage der Pulse für den S0-Kanal mit der ID 2:

```
# curl http://localhost:6310/v1/channel/2/pulses
```

Antwort:

```
{
   "channel" : 2
   "pulses" : [
      {
         "channel_id" : 2,
         "timestamp_ns" : 543838417704
         "level" : false,
      },
      {
         "channel_id" : 2,
         "timestamp_ns" : 543917859198,
         "level" : true
      }
   ],
}
```

Die Pulse müssen für jeden konfigurierten Kanal einzeln abgefragt werden.
Für jeden Änderung des Eingangspegels am GPIO wird ein Datensatz zurückgegeben der folgende Informationen enthält:

Name	Inhalt
channel_id	Die ID des Kanals
timestamp_ns	Der Zeitpunkt des Pegeländerung seit dem Programmstart in Nanosekunden
level	Der neue Pegel am GPIO

Es sollten sich die Pegel in den aufeinanderfolgenden Datensätzen immer abwechseln - also: true -> false -> true. Wenn aber Pegeländerungen sehr schnell hintereinander auftreten kann es vorkommen, dass zwei oder mehrere Datensätze mit gleichen Pegeln existieren - z. B.: false -> false. Dann war dazwischen ein sehr kurzer High-Puls, der nicht aufgezeichnet wurde.

Achtung: Die Pulse werden nach der Abfrage nicht gelöscht sonder müssen über den REST API call "delete_upto_timestamp_ns" gelöscht werden!

Löschen der Pulse bis zum Zeitstempel 1234 für den S0-Kanal mit der ID 2:

```
# curl http://localhost:6310/v1/channel/2/delete_upto_timestamp_ns/1234
```

Antwort:

```
{}
```

Die Pulse müssen für jeden konfigurierten Kanal einzeln gelöscht werden.