

Near-Infrared Single-Photon Detector - IRSPD1 by S-Fifteen Instruments

1 Features

- 10% Photon detection efficiency
- Single-photon sensitivity from 1020 nm to 1650 nm
- Free-Running
- Dark counts below 20 000 s⁻¹
- Detection timing jitter below 300 ps
- User-adjustable diode parameters (temperature, threshold, hold-off time, reverse-bias voltage)
- Integrated detection events counter



Applications: LIDAR & LADAR, Ranging, Optical communications, Ultra-sensitive fluorescence measurements, Environmental analysis, Biomedical devices, Quantum key distribution

2 General Description

The IRSPD1 is a single-photon detector module sensitive to the near-infrared spectrum. The photodetection is performed by a passively quenched InGaAs/InP avalanche photodiode in Geiger mode. A monolithically integrated quenching resistor provides a stable high-gain single-photon response and low dark counts.

3 Specifications

Table 1: Device specifications

| Diode properties – Cooled Negative Feedback Avalanche Diode | |
|--|--|
| Photon detection efficiency | 10% @ 1320 nm, -40°C device temperature |
| Spectral response range | 1020 nm to 1650 nm |
| Dark count (dc) rate | <20 000 dc/s, typical 15 000 dc/s |
| Diode temperature | Min. -50°C with the device case at 25°C, Software controllable |
| Breakdown voltage V_B | 70 V – 85 V |
| Temperature coefficient of V_B | 0.1 V/K |
| Available reverse-bias range | 0 – 100 V, Software controllable |
| Hold-off time | 0 – 1 μ s, Software controllable |
| Internal quenching resistor | 350 000 Ω |
| Fiber pigtail | SMF28 FC/UPC (other options available upon request) |
| Software Interface | |
| Physical port | USB 2.0, Type B |
| Communication | Serial communication via virtual COM port / USB CDC ACM class |
| Electrical Specifications | |
| Output pulse | NIM (Negative True -0.8V, False 0V, 50 Ω Load) or TTL (Positive True 2V, False 0V, 50 Ω Load), Software controllable |
| Pulse rise/fall time (20% to 80%) | <2 ns |
| Timing jitter | <300 ps |
| Connector | LEMO 00 (NIM-CAMAC CD/N 549) or SMA, Selectable at product purchase |
| Power supply | 4 W, 12 V DC, 2.1 mm Barrel socket |
| Physical dimensions | |
| Size (W x L x H) | 107 mm x 160 mm x 58 mm |
| Weight | 500 g |

4 Software control commands

Send plain-text commands separated by newline/cr or semicolon via the virtual COM port. An eventual reply comes terminated with cr+lf. The commands are case-insensitive.

Table 2: List of serial commands/queries available on the detector

| Command | Description |
|--------------------|---|
| *IDN? | Returns device identifier |
| *RST | Resets device |
| TEMP? | Returns the current diode temperature |
| HVOLT <value> | Sets the diode reverse bias to <value> Volt. |
| HVOLT? | Returns the current voltage setting |
| PVOLT <value> | Sets the Peltier element to <value> Volt. |
| SETTEMP <value> | Sets the APD temperature to <value> Celsius. |
| SETTEMP? | Returns the set temperature. |
| CONSTP <value> | Sets the P loop constants in V/K. |
| CONSTI <value> | Sets the I loop constants in V/s/K. |
| CONSTD <value> | Sets the D loop constants in Vs/K. |
| LOOP <s> | Sets the temp loop status <s>. 1: on, 0: off |
| LOOP? | Returns the temperature loop status. |
| THRESHVOLT <value> | Sets the threshold to <value> Volt. Can be both polarities. |
| THRESHVOLT? | Returns current threshold voltage. |
| TIME <t> | Sets the counter integration time to <t> millisecc. |
| TIME? | Returns the counter integration time in millisecc. |
| COUNTS? | Returns the events counted in an integration time. |
| SAVE | Saves the current settings. |
| HELP | Print help text. |

NOTE: Command to set the output pulse type (NIM/TTL) is missing.

4.1 Python Library for S-Fifteen Instruments

A python class encapsulating all serial commands can be found on <https://github.com/s-fifteen-instruments/pyS15>. To install the library follow the instructions on GitHub.

Here a code example to count photon detections for 1 second:

```
from S15lib.instruments import SinglePhotonDetector
detector = SinglePhotonDetector('/dev/tty.usbmodemISP1_0101')
detection_events = detector.counts(1)
```