# **Open source intensity Verilog cross-correlator and intensity spectrograph firmware**

I. Platone1

1Ilia Platone, 47841 RN Italy, [info@iliaplatone.c](mailto:info@iliaplatone.com)om

## **ABSTRACT**

Logging and pulse counts reporting require simple and efficiently fast protocols. Intensity cross-correlators and spectrographs use logging protocols that most of the times are proprietary and closed-source or even patent protected. This paper describes an intensity cross-correlator and spectrograph protocol and Verilog firmware.

Such firmware aims to be a standard and efficient way to solve most of the problems in pulse logging and cross-correlation reports, to be complete and exhaustive, and reasonably fast and easy for interpretation.

This work is composed by a protocol part, a firmware for FPGAs or CPLDs and an hardware implementation of this solution.

## **1. Protocol parts and expansion chances**

### 1.*1* The protocol header structure and payload types

This protocol is composed by various parts, it is customizable from source code and is expandable to more parts. The payload types are described by a header, which sets lengths and data size of each element and payloads, and the payloads itselves. The header length is 64 bits, and contains the data word size value, the number of the input channels, the delay amount, the jitter or cross-delay lines and the clock frequency. Each value is transmitted as ASCII values expressed in hexadecimal notation.

The payloads are sent in this sequence and the sampling time corresponds to the total payloads and header transmission time:

* Pulse counts of each channel with the selected delay time.
* Pulse counts of the autocorrelated flux of each channel.
* Pulse counts of the correlations between each channel with others independently expressed with delay jitter range correlations.

### 1.*2* The protocol header structure and payload types

Each payload can be used for different operating modes. This firmware can be used for intensity correlations, spectrograms and time-tagged cross-correlations.

Each packet contains all of these payloads, and the header describes each payload length. Only the cross-correlator payload has a length defined by all the possible baselines that can be obtained with the input channels and jitter delay lines.

Pulse counts are subject to a pulse bandwidth dependent to the clock speed of the FPGA or CPLD programmed with the firmware. The output stream can be transmitted using an UART connection, even with high or custom baud rates. Each packet generates a sampling clock pulse that can be used as external strobe for microcontrollers or computers.

### 1.*3* *INDI driver or 3rd-party software support*

There is a software that can read and drive this firmware or a device programmed with this code: INDI library already contains this as driver and that can dinamically configure itself to read as many channels as programmed into the chip.

This driver can also do an aproximate calculation of an hypotetic star or stellar object magnitude or ideal photon flux.

There is also the possibility to use this driver to draw a test plot of the correlation Fourier plane and estimate an interferometric observation.

## **2. Operating modes**

### *2.1* *Cross-correlation mode*

This firmware’s main usage is as cross-correlator impulsive detector. Each payload contains a list of the baselines’ cross-correlations during the sampling time defined by the UART or packet transmission time.

This mode is mainly useful for particle photonics and communication tests, or cross-correlation between bosons or fermions, expecially in electron or photon correlation and anti-correlation detection or measurements.

This mode can also be used in quality tests in radio interferometry pulsing mode by using special high-pass detectors or noise-scattering or geiger mode radio or photon receivers.

Since the bandwidth is generally higher than sampling frequency, this mode is not be useful for dosimetry or cross-correlation between alpha/beta radiation revealers, althought can be used in neutrin detectors or muon counters.

### 2.2 *Fluorescence spectrograph*

This firmware is able to generate a mutichanner fluorescence spectrograph by connecting it to an array of PMTs with adapted input voltage levels. Its autocorrelation payload can reach up to 4096 delay lines for a linear scale frequency analysis.

Indide its code there’s a parameter, which is common in this mode and cross-correlation mode, that generates the fluorescence graph payload and the cross-correlation delay line array.

### 2.*3* *Intensity correlator mode*

Each packet’s first payload consists to the pulse counts of each input within the sampling time. Each input ‘s pulse counts can be used in intensity interferometry as in high-speed photometry both in laboratory or photonics, telecommunications and in intensity cross-correlation of each node intensities.

### 2.*4* *Time deviation correlator*

The firmware also can use each baseline as time deviation measurement tool, by using the jitter delay lines parameter.

This parameter generates a series of cross-correlation blocks that range from the center (fully correlated) to the number of lines deviation cross-correlations.

The values obtained from the time deviation comparison are expressed in pulse counts, which can be summed to reduce the bandwidth or can be used for comparison of the time deviation or delay between the lines of the 2nd order baseline in subject.

## **3. Hardware examples and existing implementations**

### *3.1* *12-channel cross-correlator*

Github.com hosts a repository that contains the firmware Verilog code and a description of the protocol in the README.md.

This firmware contains all the work described here and is freely downloadable and customizable.

Licensed with the GPL3 open source license, can be also freely distributed and used in proprietary projects.

The cross-correlator firmware is part of a larger project for an international amateur driven interferometer network, and can also be used in local smaller interferometers. Small associations and amateur astronomer clubs are invited to use this code and possibly these products for small interferometers or school projects.

### *3*.*2* *The AHP XC series*

AHP already has a product that uses this code. The XC series are a series of cross-correlators with 4,8, and 12 inputs, 20 or 16 bits sampling words, no delay or jitter calculations.

These products are aimed for pulse or noise-scattering cross-correlation measurements of stars or astronomical objects.

### *3*.*2* *The AHP LC series*

Future implementations of this code are aimed for gamma spectrography with fluorescence detectors or revealers, expecially studied for gamma emissions or gamma ray bursts from amateur plants.

## **4. Next Chapter**

### *4.1* *Next paragraph*

Next description.