

OpenCPI

Advanced Pattern Component Data Sheet

OpenCPI Release: v2.4.7

Summary - Advanced Pattern

Name	advanced_pattern
Worker Type	Application
OpenCPI Release	v2.4.7
Last Update	4/2019
Component Library	ocpi.assets.util_comps
Workers	advanced_pattern.rcc
Tested Platforms	CentOS7, xilinx13.3 (limited)

Functionality

The Advanced Pattern Component provides predefined data to assist in the testing of other Components.

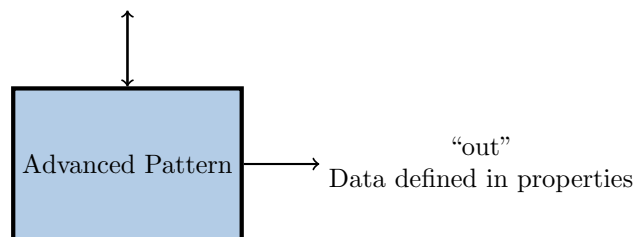
The data can be arranged in messages of up to 2048 bytes at a time with each block having a specific opcode. By default, 32 of these messages are available, but that configuration is exposed as a build-time parameter with a default configuration building additional Workers allowing for 64, 128, and 256 messages.

This Component provides *minimal* error checking and is **not recommended for production use**, but is only intended for prototyping and testing of other Components.

Block Diagrams

Top level

See property table



Source Dependencies

advanced_pattern.rcc

- <assets>/components/util_comps/advanced_pattern.rcc/advanced_pattern.cc

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
maxPatternLength	ULong	-	-	Parameter	Standard	32	Maximum “ Pattern ” sequence length to allow ¹
Pattern	Struct	maxPatternLength	-	Initial, Readable ²	-	-	Message to send
Pattern.Opcode	UChar	-	-	”	Standard	0	Opcode metadata to send with this message’s data
Pattern.Bytes	UChar	2048	-	”	Standard	0	Data to send
LoopCount	ULongLong	-	-	Initial, Readable ²	Standard	1	How many times to repeat the “ Pattern ” sequence ³
ZLM	UShort	-	-	Initial, Readable ²	0 ... 256	0	Opcode for a Z ero L ength M essage with when finished. ⁴
current	Struct	-	-	Volatile	-	-	Current statistics for each opcode
current.Total	Struct	-	-	”	-	-	Statistics across <i>all</i> opcodes
current.Total.bytes	ULongLong	-	-	”	Standard	-	Number of bytes received
current.Total.messages	ULongLong	-	-	”	Standard	-	Number of messages received
current.Opcode	Struct	-	256	”	-	-	Statistics for <i>each</i> opcode
current.Opcode.*	Various	-	-	”	-	Various	Various ⁵

¹Each **Pattern** entry requires about 2K of RAM.
²“Readable” is deprecated and superfluous here. It will be removed in a future release.
³0 will continue as long as Worker is running.
⁴Default is opcode 0; set to invalid opcode 256 if this feature is *not* desired.
⁵Internal structure equivalent to **current.Total** and not explicitly shown.

Worker Properties

There are no implementation-specific properties for this component.

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
out	true	-	-	numberofopcodes=256	Data defined in properties

Worker Interfaces

There are no implementation-specific interfaces for this component.

Test and Verification

The `advanced_pattern` worker is tested using five cases. All cases are verified using the md5 hashes of different properties and or the output file against the expected md5 hashes. A sixth case, `test_05`, is present but not tested due to sending a ZLM with opcode 1. The `file_write` component cannot currently stop on ZLM with opcodes other than 0.

`case00` runs default values and sends a single ZLM with opcode 0. The output file and the contents of the `current` property are compared to the expected.

`case01` also runs default values but does not send a ZLM and instead times out after 10 seconds. The `current` property is compared to the expected.

`case02` generates the `Pattern` property (`generate.py 2`) and sends three bytes every tenth opcode. The `maxPatternLength` property is explicitly set to be 256. The output file, contents of the `current` property, and, contents of the `maxPatternLength` property are compared to the expected.

`case03` generates the `Pattern` property (`generate.py 3`) and sends three bytes every tenth opcode, 50 times. There is no ZLM sent and the test case times out after 10 seconds. The `maxPatternLength` property is set to 64. For clarity, an `Output_03.out` file is created to see from where the output hashes are gotten. The output file, contents of the `current` property, and, contents of the `maxPatternLength` property are compared to the expected.

`case04` generates 2048 bytes of data for the `Pattern` property (`generate.py 4`) and sends 3 bytes with opcode 0 for 10 seconds. The contents of the `maxPatternLength` property, and the end cut of the `current` property are compared to the expected.