

Summary - Baud Tracking Simple

Name	baudtracking_simple
Worker Type	Application
Version	v1.4
Release Date	September 2018
Component Library	ocpi.assets.dsp_comps
Workers	baudtracking_simple.rcc
Tested Platforms	centos7, xilinx13.3

Worker Implementation Details

baudtracking_simple.rcc

The input data to this worker is expected to be pulse shaped samples. The full length of the baud in samples (SPB) is averaged over BaudAvrCount bauds. The maximum of this array of averages is then calculated, if this is not in the mid point of the array, then a drift is calculated. The midpoint of the next SPB envelope is then shifted based on the drift value.

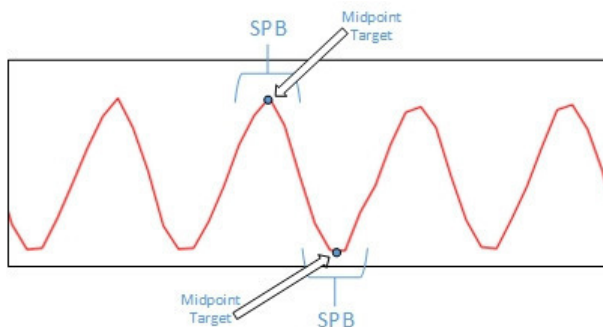
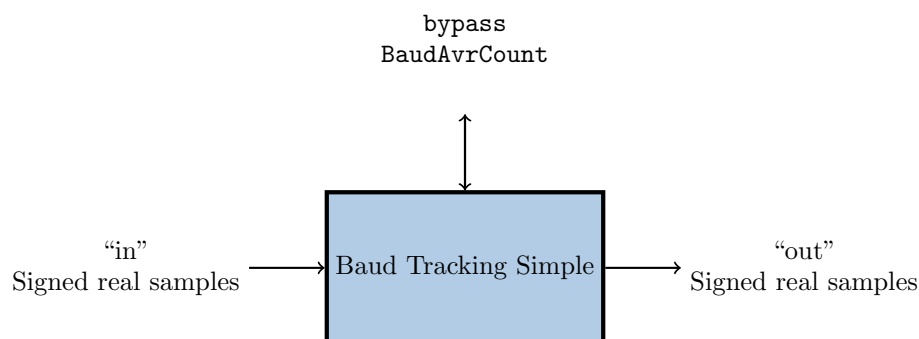


Figure 1: Functionality Diagram

For a consistent signal, the algorithm will lock on after a few iterations followed by a few small adjustments. If you are not exactly sure of the SPB, the algorithm will constantly adjust and find the peak values as long as the SPB is not off by too much.

Block Diagrams

Top level



Source Dependencies

baudtracking_simple.rcc

- `ocpi.assets/components/dsp_comps/Baudtracking_simple.rcc/Baudtracking_simple.c`

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
bypass	Bool	-	-	Readable, Writable	Standard	-	When set to true the worker will pass its input to its output without any processing. False means normal operation.
BaudAvrCount	Short	-	-	Readable, Writable	5-34767	-	The number of bauds to average over to determine drift(10 is a good place to start). If this number is too high the algorithm won't be able to compensate quickly enough if the average number is too low it will adjust back and forth without enough averages to make an intelligent decision.

Worker Properties

baudtracking_simple.rcc

Control Operations: Start

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	SPB	UShort	-	-	Readable,Writable	3-100	-	The expected number of samples per baud. This number can be slightly off and the algorithm will work correctly. As long as it is close, the averaging mechanism will compensate.

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
in	false	rstream_protocol	false	-	Signed real samples
out	true	rstream_protocol	false	-	Signed real samples

Performance and Resource Utilization

baudtracking_simple.rcc

Processor Type	Processor Frequency	Run Function Time
TBD	TBD	TBD

Test and Verification

No unit test currently exists for this worker.