BUFFER ISSUE RESOLUTION DOCUMENT (BIRD)
Rev 1
3/21/15

BIRD NUMBER: *TBD*
ISSUE TITLE: *Backchannel_Protocol "Basic"*
REQUESTOR: *Walter Katz, Signal Integrity Software, Inc.*

DATE SUBMITTED:
DATE REVISED:
DATE ACCEPTED BY IBIS OPEN FORUM:

ANALYSIS PATH/DATA THAT LED TO SPECIFICATION:

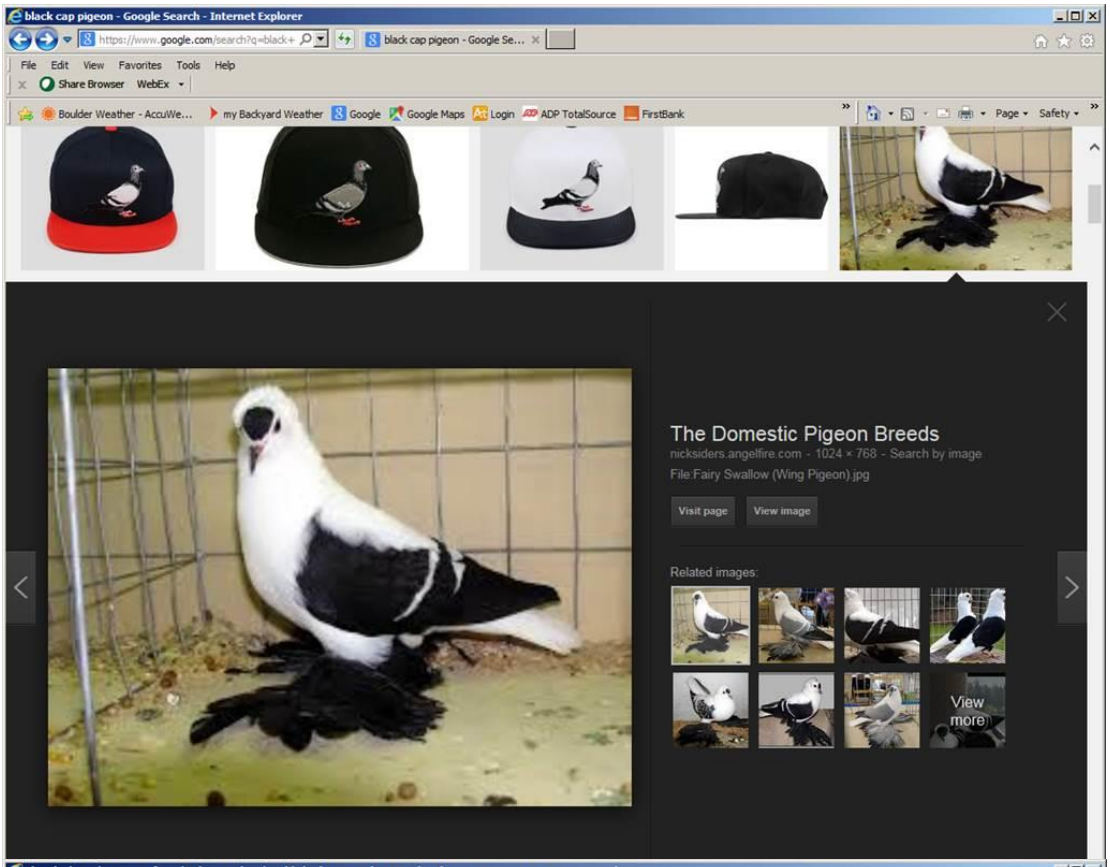
Define a simple BCI protocol that should represent most Tx models.

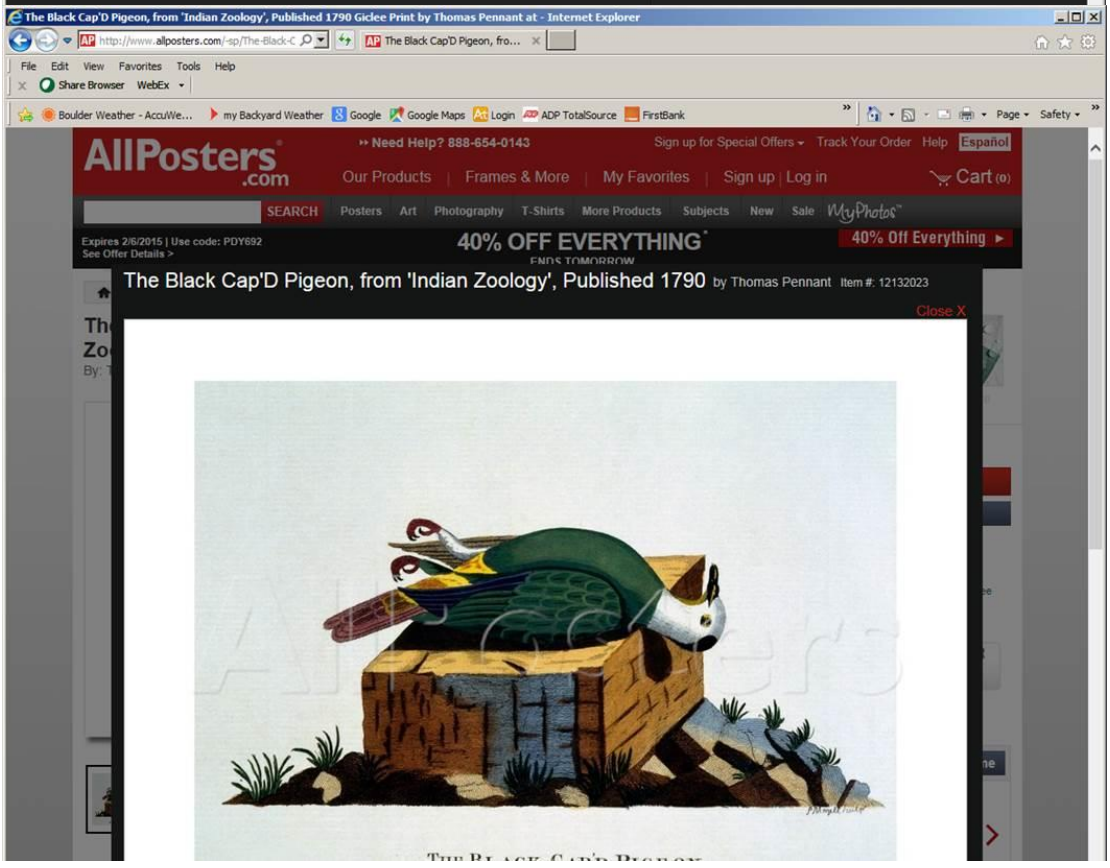
ANY OTHER BACKGROUND INFORMATION:

A Pigeon describes a Back Channel Protocol that defines the messages sent between the Rx and Tx models. Messenger Pigeons are birds that carry messages.

Back Channel Pigeon Mascot

Black Cap'D Pigeon


black cap pigeon - Google Search - Internet Explorer
The Domestic Pigeon Breeds
nicksiders.angelfire.com - 1024 × 768 - Search by image
File:Fairy Swallow (Wing Pigeon).jpg
Visit page
View image
Related images:
View more


The Black Cap'D Pigeon, from 'Indian Zoology', Published 1790 Giclee Print by Thomas Pennant at - Internet Explorer
AllPosters.com
Need Help? 888-654-0143
Sign up for Special Offers
Track Your Order
Help
Español
Our Products
Frames & More
My Favorites
Sign up | Log in
Cart (0)
SEARCH
Posters
Art
Photography
T-Shirts
More Products
Subjects
New
Sale
MyPhotos
Expires 2/6/2015 | Use code: PDY692
See Offer Details >
40% OFF EVERYTHING
40% Off Everything
The Black Cap'D Pigeon, from 'Indian Zoology', Published 1790 by Thomas Pennant Item #: 12132023
Close X

Backchannel_Protocol Basic

The output of Tx AMI_Init shall contains in the BCI branch a branch for each tap containing the parameters min_gain, max_gain, gain_step and gain as well as the parameter tx_swing. The parameters gain and the parameter tx_swing determine the equalization and amplitude used by the Tx to generate the Impulse Response output of the Tx AMI_Init function. This represents an FFE filter with the output of each tap being tx_swing times the gain of the tap. The EDA tool shall copy this BCI branch from the Tx AMI_Init output to the Rx AMI_Init AMI_Parameters_In input string. The following BNF format describes the message set that the Tx AMI_Init function outputs in the (BCI …) branch.

```
(BCI  (tap_filter
            (<tap number>
                  (min_gain <min gain>)
                  (max_gain <max gain>)
                  (gain_step <gain step>)
                  (gain <gain used to generate IR))
       (tx_swing <peak to peak voltage used to generate IR>))
```

In the Tx AMI_Init AMI_parameters_in string this will appear as (In this example there is one pre-cursor tap (-1) and one post cursor tap (1)):

```
(BCI  (tap_filter
            (-1 (min_gain -0.2)(max_gain 0.2)(gain_step 0.01)(gain 0.0))
            ( 0 (min_gain  0.2)(max_gain 1.0)(gain_step 0.01)(gain 1.0))
            ( 1 (min_gain -0.2)(max_gain 0.2)(gain_step 0.01)(gain 0.0))
      (tx_swing 1.0))
```

The contents of the BCI branch can also be described in the format used to define parameters in the .ami file:

```
(BCI
  (tap_filter
    (-1
      (min_gain (Value -0.2)(Usage Out)(Type Float)
            (Description "Pre-cursor min gain"))
      (max_gain (Value 0.2) (Usage Out)(Type Float)
            (Description "Pre-cursor max gain"))
      (gain_step (Value 0.01) (Usage Out)(Type Float)
            (Description "Pre-cursor gain step"))
      (gain (Range 0.0 -0.2 0.2) (Usage Out)(Type Float)
```

```
            (Description "Pre-cursor gain used to generate impulse response")))
    ( 0
      (min_gain (Value 0.2)(Usage Out)(Type Float)
            (Description "Main-cursor min gain"))
      (max_gain (Value 1.0) (Usage Out)(Type Float)
            (Description "Main-cursor max gain"))
      (gain_step (Value 0.01) (Usage Out)(Type Float)
            (Description "Main-cursor gain step"))
      (gain (Range 1.0 0.0 1.0) (Usage Out)(Type Float)
            (Description "Main-cursor gain used to generate impulse response")))
    ( 1
      (min_gain (Value -0.2)(Usage Out)(Type Float)
            (Description "Post-cursor min gain"))
      (max_gain (Value 0.2) (Usage Out)(Type Float)
            (Description "Post-cursor max gain"))
      (gain_step (Value 0.01) (Usage Out)(Type Float)
            (Description "Post-cursor gain step"))
      (gain (Range 0.0 -0.2 0.2) (Usage Out)(Type Float)
            (Description "Post-cursor gain used to generate impulse response")))
  (tx_swing (Value 1.0) (Usage Out)(Type Float)
            (Description "Tap coefficient is the tap gain times tx_swing"))))
```

The following describes in BNF format the message set of the BCI_Command_Set "Basic" that is communicated from the Rx AMI_Init and Rx_Ami_GetWave to the Tx AMI_Init and Tx AMI_GetWave functions. This branch describes the changes that the Rx wants the Tx to make. Note that this branch cannot contain both gain and increment. These are optional, and if not present, then that parameter will not change.

```
(BCI  (tap_filter
            (<tap number>
                  (increment <increment>)
                  (gain <gain used to generate IR))
       (tx_swing <scales the value of each gain>))
```

Example of actual BCI branches generated by the Rx:

Coefficient method (eg. PCIeG3)

```
(BCI(tap_filter (-1 (gain -.1))(0 (gain 0.85))(1 (gain -.05)))
(BCI(tap_filter (-1 (gain -.1))(1 (gain -.05)))
(BCI(tx_swing .8))
```

Increment method (eg. 802.3KR)

```
(BCI(tap_filter (-1 (increment -2))(0 (increment -3))(1 (increment -1)))
(BCI(tap_filter (-1 (increment -2))(1 (increment -1)))

In .ami format
(BCI
  (tap_filter
    (-1
      (increment (Range 0 -4 4) (Usage Out)(Type Integer)
            (Description "Pre-cursor increment requested by Rx"))
      (gain (Range 0.0 -0.2 0.2) (Usage Out)(Type Float)
            (Description "Pre-cursor gain requested by Rx")))
    ( 0
      (increment (Range 0 -4 4) (Usage Out)(Type Integer)
            (Description "Main-cursor increment requested by Rx"))
       gain (Range 1.0 0.2 1.0) (Usage Out)(Type Float)
            (Description "Main-cursor gain requested by Rx")))
    ( 1
      (increment (Range 0 -4 4) (Usage Out)(Type Integer)
            (Description "Post-cursor increment requested by Rx"))
      (gain (Range 0.0 -0.2 0.2) (Usage Out)(Type Float)
            (Description "Post-cursor gain requested by Rx")))
  (tx_swing (Value 1.0) (Usage Out)(Type Float)
            (Description "tx_swing requested by Rx"))))
```

The Tx AMI_Init or AMI_GetWave function applies these requested changes and shall return the BCI branch containing the values of gain and tx_swing that were used to generate the IR output of Tx AMI_Init or the equalization that the Tx AMI_GetWave applies to the stimulus input. The values of increment shall either be 0, -1 or 1. If -1the tap has reached its lower limit and can only be increased, if 1 the tap has reached its upper limit and can only be decreased, and if 0 the tap can be increased or decreased.

Example of the Tx BCI branch output:

```
(BCI(tap_filter   (-1 (gain -.11) (increment 0))
```

```
                    ( 0 (gain 0.83) (increment 0))
                    ( 1 (gain -.06) (increment 0)))
    (tx_swing 1.0))

In .ami format
(BCI
  (tap_filter
    (-1
      (increment (Range 0 -1 1) (Usage Out)(Type Integer)
            (Description "Pre-cursor increment implemented by Tx"))
      (gain (Range 0.0 -0.2 0.2) (Usage Out)(Type Float)
            (Description "Pre-cursor gain implemented by Tx")))
    ( 0
      (increment (Range 0 -1 1) (Usage Out)(Type Integer)
            (Description "Main-cursor increment implemented by Tx"))
       gain (Range 1.0 0.2 1.0) (Usage Out)(Type Float)
            (Description "Main-cursor gain implemented by Tx")))
    ( 1
      (increment (Range 0 -1 1) (Usage Out)(Type Integer)
            (Description "Post-cursor increment implemented by Tx"))
      (gain (Range 0.0 -0.2 0.2) (Usage Out)(Type Float)
            (Description "Post-cursor gain implemented by Tx")))
  (tx_swing (Value 1.0) (Usage Out)(Type Float)
            (Description "tx_swing implemented by Tx"))))
```