

BA-4000 PROGRAMMING GUIDE

REV 2.0



Version Note

Version	Description	Release Date
1.0	Built up	
2.0	Create FGC Function <ul style="list-style-type: none">• SwitchPPGEDOptionsV4• ReadBERResultMarginV3• ReadBERResultMarginV4• ReadSelectedPPGOptionsV2• ReadSelectedPPGOptionsV3 Update FGC information into Appendix A	2021/07/28

How to get API

1. Open the **INO-BA GUI** which you are using
2. Select from system bar [Help] ➔ [Programming Guide]
3. Save BA_API.zip
4. *Unzip BA_API.zip

*Folder include C# and Python Sample Code, dll file ,and header file.

How to use API

1. Open the BA_API folder which you just download
2. Copy BA_API.dll
3. Paste BA_API.dll into your programming folder
4. *Import and use dll in your programming

* For detailed usage, please refer to the sample code.

How to use SCPI

1. Open the BA_API folder which you just download
2. Copy BA_API.dll
3. Paste BA_API.dll into **BA_SCPI_Converter** folder
4. *Open **BA_SCPI_Converter.exe**
- 5-1. **(VISA socket) Copy VISA Address to your programming
- 5-2. **(TCP socket) Copy the IP and port of the VISA Address to your programming

* Please keep it on when using SCPI control, don't turn it off.

**For detailed usage, please refer to the sample code

Information about Command Version

Please use the latest version of command for the full functionality.

For example, there are 3 versions of the connection commands.

- Connect
- ConnectV2
- ConnectV3

ConnectV3 is the latest.

Connect and ConnectV2 are kept to support the codes using previous API versions.

Function Description

[Description]

User needs to use connect function to connect BA-4000, and then use any function to control BA-4000.

[API Command]

```
public static extern bool Connect(string IP, ref bool FWUpgradeRequired);
```

[API Parameter]

- IP : IP Address
- FWUpgradeRequired :
 - Return True : Use GUI to Upgrade
 - Return False : Not Required

Connect

This command is kept to support the codes using previous API versions.

[SCPI Command]

:Connect:{IP}

[SCPI Parameter]

- IP : IP Address

[SCPI Example]

:Connect:172.16.81.91

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail
- Array [32] : Return FWUpgradeRequired
 - 1 : Use GUI to Upgrade
 - 0 : Not Required

[Description]

User needs to use connect function to connect BA-4000, and then use any function to control BA-4000.

[API Command]

```
public static extern bool ConnectV2(string IP, ref byte FWStatus);
```

[API Parameter]

- IP : IP Address
- FWStatus :
 - Return 2 : Update API / GUI
 - Return 1 : Use GUI to Upgrade
 - Return 0 : Not Required

ConnectV2

This command is kept to support the codes using previous API versions.

[SCPI Command]

:ConnectV2:{IP}

[SCPI Parameter]

- IP : IP Address

[SCPI Example]

:ConnectV2:172.16.81.91

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail
- Array [32] : Return FWStatus
 - 2 : Update API / GUI
 - 1 : Use GUI to Upgrade
 - 0 : Not Required

ConnectV3

[Description]

User needs to use connect function to connect BA-4000, and then use any function to control BA-4000.

[API Command]

```
public static extern bool ConnectV3(string IP, ref byte FWStatus, ref byte hwMajorVr, ref byte hwMinorVr);
```

[API Parameter]

- IP : IP Address
- FWStatus :
 - Return 2 : Update API / GUI
 - Return 1 : Use GUI to Upgrade
 - Return 0 : Not Required
- hwMajorVr : Return Hardware Major Revision Number
- hwMinorVr : Return Hardware Minor Revision Number

ConnectV3

[SCPI Command]

:ConnectV3:{IP}

[SCPI Parameter]

- IP : IP Address

[SCPI Example]

:ConnectV3:172.16.81.91

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail
- Array [32] : Return FWStatus
 - 2 : Update API / GUI
 - 1 : Use GUI to Upgrade
 - 0 : Not Required
- Array [33] : Return Hardware Major Revision Number
- Array [34] : Return Hardware Minor Revision Number

Disconnect

[Description]

User needs to use disconnect function before stop or close program.

[API Command]

```
public static extern bool Disconnect();
```

Disconnect

[SCPI Command]

:Disconnect

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

IsConnected

[Description]

Check connection status.

[API Command]

```
public static extern bool IsConnected();
```


IsConnected

[SCPI Command]

:IsConnected

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

HWIsConnected

[Description]

Check connection status.

[API Command]

```
public static extern bool HWIsConnected();
```

HWIsConnected

[SCPI Command]

:HWIsConnected

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

ChangeIP

[Description]

Change machine IP by 1x4 byte array. It will auto restart after setting.

[API Command]

```
public static extern bool ChangeIP(ref byte IP);
```

[API Parameter]

➤ IP : IP Address

ChangelP

[SCPI Command]

:ChangelP:{IP}

[SCPI Parameter]

- IP : IP Address

[SCPI Example]

:ChangelP:172.16.77.99

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail
- Array [32-35] : Return Change IP Address

SwitchPPGEDOptions

This command is kept to support the codes using previous API versions.

[Description]

Switch PPG and ED Option to MR,1.5V, Basic, FEC or 56G.

[API Command]

```
public static extern bool SwitchPPGEDOptions(ref bool basic, ref bool multiRate, ref bool OneP5Vpp, ref bool FEC
, ref bool FiftyThreeG, ref bool isPAM4, ref bool is8Ch);
```

[API Parameter]

- basic : Basic mode Enable/Disable
- multiRate : MultiRate mode Enable/Disable
- OneP5Vpp : OneP5Vpp mode Enable/Disable
- FEC : FEC mode Enable/Disable
- FiftyThreeG : FiftyThreeG mode Enable/Disable
- isPAM4 : Return current BA support PAM4 or not [Read Only]
- is8Ch : Return current BA is 8 or 4 channels [Read Only]

SwitchPPGEDOptions

This command is kept to support the codes using previous API versions.

[SCPI Command]

:SwitchPPGEDOptions:{basic},{multiRate},{OneP5Vpp},{FEC},{FiftyThreeG},{isPAM4},{is8Ch}

[SCPI Parameter]

- basic : Basic mode Enable/Disable
- multiRate : MultiRate mode Enable/Disable
- OneP5Vpp : OneP5Vpp mode Enable/Disable
- FEC : FEC mode Enable/Disable
- FiftyThreeG : FiftyThreeG mode Enable/Disable
- isPAM4 : Return current BA support PAM4 or not [Read Only]
- is8Ch : Return current BA is 8 or 4 channels [Read Only]

[SCPI Example]

:SwitchPPGEDOptions:1,0,0,1,0,0,0

SwitchPPGEDOptions

This command is kept to support the codes using previous API versions.

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail
- Array [32] : Return Basic mode
 - 1 : Enable
 - 0 : Disable
- Array [33] : Return multiRate mode
 - 1 : Enable
 - 0 : Disable
- Array [34] : Return OneP5Vpp mode
 - 1 : Enable
 - 0 : Disable
- Array [35] : Return FEC mode
 - 1 : Enable
 - 0 : Disable
- Array [36] : Return FiftyThreeG mode
 - 1 : Enable
 - 0 : Disable
- Array [37] : Return isPAM4
 - 1 : Support PAM4
 - 0 : No support PAM4
- Array [38] : Return is8Ch
 - 1 : the board is 8 Channels
 - 0 : the board is 4 Channels

SwitchPPGEDOptionsV2

This command is kept to support the codes using previous API versions.

[Description]

Switch PPG and ED Option to MR,1.5V, Basic, FEC or 56G.

[API Command]

```
public static extern bool SwitchPPGEDOptionsV2(ref bool basic, ref bool multiRate, ref bool OneP5Vpp, ref bool FEC
, ref bool FiftyThreeG, ref bool isPAM4, ref bool is8Ch , ref bool isMA);
```

[API Parameter]

- basic : Basic mode Enable/Disable
- multiRate : MultiRate mode Enable/Disable
- OneP5Vpp : OneP5Vpp mode Enable/Disable
- FEC : FEC mode Enable/Disable
- FiftyThreeG : FiftyThreeG mode Enable/Disable
- isPAM4 : Return current BA support PAM4 or not [Read Only]
- is8Ch : Return current BA is 8 or 4 channels [Read Only]
- isMA : Return instrument is BA or MA [Read Only]

SwitchPPGEDOptionsV2

This command is kept to support the codes using previous API versions.

[SCPI Command]

:SwitchPPGEDOptionsV2:{basic},{multiRate},{OneP5Vpp},{FEC},{FiftyThreeG},{isPAM4},{is8Ch},{isMA}

[SCPI Parameter]

- basic : Basic mode Enable/Disable
- multiRate : MultiRate mode Enable/Disable
- OneP5Vpp : OneP5Vpp mode Enable/Disable
- FEC : FEC mode Enable/Disable
- FiftyThreeG : FiftyThreeG mode Enable/Disable
- isPAM4 : Return current BA support PAM4 or not [Read Only]
- is8Ch : Return current BA is 8 or 4 channels [Read Only]
- isMA : Return instrument is BA or MA [Read Only]

[SCPI Example]

:SwitchPPGEDOptions:1,0,0,1,0,0,0,0

SwitchPPGEDOptionsV2

This command is kept to support the codes using previous API versions.

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail
- Array [32] : Return Basic mode
 - 1 : Enable
 - 0 : Disable
- Array [33] : Return multiRate mode
 - 1 : Enable
 - 0 : Disable
- Array [34] : Return OneP5Vpp mode
 - 1 : Enable
 - 0 : Disable
- Array [35] : Return FEC mode
 - 1 : Enable
 - 0 : Disable
- Array [36] : Return FiftyThreeG mode
 - 1 : Enable
 - 0 : Disable
- Array [37] : Return isPAM4
 - 1 : Support PAM4
 - 0 : No support PAM4
- Array [38] : Return is8Ch
 - 1 : the board is 8 Channels
 - 0 : the board is 4 Channels
- Array [39] : Return isMA
 - 1 : MA
 - 0 : BA

SwitchPPGEDOptionsV3

This command is kept to support the codes using previous API versions.

[Description]

Switch PPG and ED Option to MR,1.5V, Basic, FEC or 56G.

[API Command]

```
public static extern bool SwitchPPGEDOptionsV3(ref bool basic, ref bool multiRate, ref bool OneP5Vpp, ref bool FEC
, ref bool FiftyThreeG, ref bool isPAM4, ref bool is8Ch , ref bool isMA , ref bool is25GNRZ);
```

[API Parameter]

- basic : Basic mode Enable/Disable
- multiRate : MultiRate mode Enable/Disable
- OneP5Vpp : OneP5Vpp mode Enable/Disable
- FEC : FEC mode Enable/Disable
- FiftyThreeG : FiftyThreeG mode Enable/Disable
- isPAM4 : Return current BA support PAM4 or not [Read Only]
- is8Ch : Return current BA is 8 or 4 channels [Read Only]
- isMA : Return instrument is BA or MA [Read Only]
- is25GNRZ : Return instrument is 28-NRZ series or not [Read Only]

SwitchPPGEDOptionsV4

To switch to FGC mode, please use this function!

[Description]

Switch PPG and ED Option to MR, 1.5V, Basic, FEC, FGC or 56G.

[API Command]

```
public static extern bool SwitchPPGEDOptionsV4(ref byte options);
```

[API Parameter]

- Array[0] = **Basic** mode Enable (1) /Disable (0)
- Array[1] = **MultiRate** mode Enable (1) /Disable (0)
- Array[2] = **OneP5Vpp** mode Enable (1) /Disable (0)
- Array[3] = **FEC** mode Enable (1) /Disable (0)
- Array[4] = **FGC** mode Enable (1) /Disable (0)
- Array[5] = **FiftyThreeG** mode Enable (1) /Disable (0)
- Array[6] = **isPAM4** return current BA support PAM4 or not [Read Only]
- Array[7] = **is8Ch** return current BA is 8 or 4 channels [Read Only]
- Array[8] = **isMA** return instrument is BA or MA [Read Only]
- Array[9] = **is25GNRZ** return instrument is 28-NRZ series or not [Read Only]
- Array[10] = **is53GB** return instrument is 56-PAM series or not [Read Only]
- Array[11 – 31] = Reserved

SetBAConfig

This command is kept to support the codes using previous API versions.

[Description]

Set data rate, signal type (NRZ/PAM4), clock, ED sensitivity and CTLE.

[API Command]

```
public static extern bool SetBAConfig(byte signalMode, int dataRate, byte clockDiv, byte rxSensitivity, ushort ctle);
```

[API Parameter]

- signalMode :
 - 1 : NRZ
 - 0 : PAM4
- dataRate : set DataRate in MBd
 - NRZ : 28900000, 28050000, 26562500, 25781250, 24330240, 14025000, 12500000, 11317600, 10709000, 10312500, 9953280
 - PAM4 : 57800000, 53125000, 28900000, 28050000, 26562500
- clockDiv :
 - 0 : Rate/8
 - 1 : Rate/16
 - 2 : Rate/32
 - 3 : Rate/64
- rxSensitivity :
 - 0 : High Sensitivity
 - 1 : Optimized
 - 2 : High Power
- ctle : set CTLE in dB

SetBAConfig

This command is kept to support the codes using previous API versions.

[SCPI Command]

:SetBAConfig:{signalMode},{dataRate},{clockDiv},{rxSensitivity},{ctle}

[SCPI Parameter]

- signalMode :
 - 1 : NRZ
 - 0 : PAM4
- dataRate : set DataRate in MBd
 - NRZ : 28900000, 28050000, 26562500, 25781250, 24330240, 14025000, 12500000, 11317600, 10709000, 10312500, 9953280
 - PAM4 : 57800000, 53125000, 28900000, 28050000, 26562500
- clockDiv :
 - 0 : Rate/8
 - 1 : Rate/16
 - 2 : Rate/32
 - 3 : Rate/64
- rxSensitivity :
 - 0 : High Sensitivity
 - 1 : Optimized
 - 2 : High Power
- ctle : set CTLE in dB

SetBAConfig

This command is kept to support the codes using previous API versions.

[SCPI Example]

:SetBAConfig:0,26562500,0,1,2

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

SetBAConfigAdv

This command is kept to support the codes using previous API versions.

[Description]

Set data rate, signal type (NRZ/PAM4), clock, ED sensitivity and CTLE.

[API Command]

```
public static extern bool SetBAConfigAdv(byte signalMode, int dataRate, byte clockDiv
                                         , byte rxSensitivity, ushort ctle , byte mapping, byte reserved);
```

[API Parameter]

- signalMode :
 - 0 : PAM4
 - 1 : NRZ
- dataRate : set DataRate in MBd
 - NRZ : 28900000, 28050000, 26562500, 25781250, 24330240, 14025000, 12500000, 11317600, 10709000, 10312500, 9953280
 - PAM4 : 57800000, 53125000, 28900000, 28050000, 26562500
- clockDiv :
 - 0 : Rate/8
 - 1 : Rate/16
 - 2 : Rate/32
 - 3 : Rate/64
- rxSensitivity :
 - 0 : High Sensitivity
 - 1 : Optimized
 - 2 : High Power
- ctle : set CTLE in dB
- mapping :
 - 0 : Gray Mapping
 - 1 : Linear Mapping
- reserved :
 - 0 : Off
 - 1 : On

SetBAConfigAdv

This command is kept to support the codes using previous API versions.

[SCPI Command]

:SetBAConfigAdv:{signalMode},{dataRate},{clockDiv},{rxSensitivity},{ctle},{mapping},{reserved}

[SCPI Parameter]

- signalMode :
 - 0 : PAM4
 - 1 : NRZ
- dataRate : set DataRate in MBd
 - NRZ : 28900000, 28050000, 26562500, 25781250, 24330240, 14025000, 12500000, 11317600, 10709000, 10312500, 9953280
 - PAM4 : 57800000, 53125000, 28900000, 28050000, 26562500
- clockDiv :
 - 0 : Rate/8
 - 1 : Rate/16
 - 2 : Rate/32
 - 3 : Rate/64
- rxSensitivity :
 - 0 : High Sensitivity
 - 1 : Optimized
 - 2 : High Power
- ctle : set CTLE in dB
- mapping :
 - 0 : Gray Mapping
 - 1 : Linear Mapping
- reserved :
 - 0 : Off
 - 1 : On

SetBAConfigAdv

This command is kept to support the codes using previous API versions.

[SCPI Example]

:SetBAConfigAdv:0,26562500,0,1,2,0,0

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

SetBAConfigAdvV2

[Description]

Set data rate, signal type (NRZ/PAM4), clock, ED sensitivity, CTLE, and PPM.

[API Command]

`Public static extern bool SetBAConfigAdvV2(byte signalMode, int dataRate, byte clockDiv, byte rxSensitivity, ushort ctle, byte mapping, byte reserved, ushort PPMTuneVal);`

[API Parameter]

- signalMode :
 - 0 : PAM4
 - 1 : NRZ
- dataRate : set DataRate in MBd
 - NRZ : 28900000, 28050000, 26562500, 25781250, 24330240, 14025000, 12500000, 11317600, 10709000, 10312500, 9953280
 - PAM4 : 57800000, 53125000, 28900000, 28050000, 26562500
- clockDiv :
 - 0 : Rate/8
 - 1 : Rate/16
 - 2 : Rate/32
 - 3 : Rate/64
- rxSensitivity :
 - 0 : High Sensitivity
 - 1 : Optimized
 - 2 : High Power
- ctle : set CTLE in dB
- mapping :
 - 0 : Gray Mapping
 - 1 : Linear Mapping
- reserved :
 - 0 : Off
 - 1 : On
- PPMTuneVal : tune PPM rate between “+ - 300 ppm”

SetBAConfigAdvV2

[SCPI Command]

:SetBAConfigAdvV2:{signalMode},{dataRate},{clockDiv},{rxSensitivity},{ctle},{mapping},{reserved},{PPMTuneVal}

[SCPI Parameter]

- signalMode :
 - 0 : PAM4
 - 1 : NRZ
- dataRate : set DataRate in MBd
 - NRZ : 28900000, 28050000, 26562500, 25781250, 24330240, 14025000, 12500000, 11317600, 10709000, 10312500, 9953280
 - PAM4 : 57800000, 53125000, 28900000, 28050000, 26562500
- clockDiv :
 - 0 : Rate/8
 - 1 : Rate/16
 - 2 : Rate/32
 - 3 : Rate/64
- rxSensitivity :
 - 0 : High Sensitivity
 - 1 : Optimized
 - 2 : High Power
- ctle : set CTLE in dB
- mapping :
 - 0 : Gray Mapping
 - 1 : Linear Mapping
- reserved :
 - 0 : Off
 - 1 : On
- PPMTuneVal : tune PPM rate between “+ - 300 ppm”

SetBAConfigAdvV2

[SCPI Example]

:SetBAConfigAdvV2:0,26562500,0,1,2,0,0,100

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

SetPattern

[Description]

Set TX/RX Pattern

[API Command]

`Public static extern bool SetPattern(byte channelIndex, byte patternSelect, UInt64 userPattern, bool isAutoLock, byte fecType);`

[API Parameter]

➤ channelIndex :

- 0 : All Channels
- 1-8 : Per Channel

➤ patternSelect :

- | | | |
|-----------|-------------------|------------|
| • NRZ : | • User Pattern: 9 | • PAM4 : |
| 0 : PN7 | | 10 : PN7Q |
| 1 : SSPRQ | | 11 : SSPRQ |
| 2 : PN9 | | 12 : PN9Q |
| 3 : PN11 | | 13 : PN11Q |
| 4 : PN13 | | 14 : PN13Q |
| 5 : PN15 | | 15 : PN15Q |
| 6 : PN16 | | 16 : PN16Q |
| 7 : PN23 | | 17 : PN23Q |
| 8 : PN31 | | 18 : PN31Q |

SetPattern

[API Parameter]

- userPattern : user pattern max length is 64 bits
- isAutoLock :
 - False : Auto Lock Off
 - True : Auto Lock On
- fecType :
 - NRZ :
 - 0 : Single KR4
 - 1 : Combined KR4
 - PAM4 :
 - 2 : Single KP4
 - 3 : Combined KP4

SetPattern

[SCPI Command]

:SetPattern:{channelIndex},{patternSelect},{userPattern},{isAutoLock},{fecType}

[SCPI Parameter]

➤ channelIndex :

- 0 : All Channels
- 1-8 : Per Channel

➤ patternSelect :

- | | | |
|-----------|-------------------|------------|
| • NRZ : | • User Pattern: 9 | • PAM4 : |
| 0 : PN7 | | 10 : PN7Q |
| 1 : SSPRQ | | 11 : SSPRQ |
| 2 : PN9 | | 12 : PN9Q |
| 3 : PN11 | | 13 : PN11Q |
| 4 : PN13 | | 14 : PN13Q |
| 5 : PN15 | | 15 : PN15Q |
| 6 : PN16 | | 16 : PN16Q |
| 7 : PN23 | | 17 : PN23Q |
| 8 : PN31 | | 18 : PN31Q |

SetPattern

[SCPI Parameter]

- userPattern : user pattern max length is 64 bits
- isAutoLock :
 - 0 : Auto Lock Off
 - 1 : Auto Lock On
- fecType :
 - NRZ :
 - 0 : Single KR4
 - 1 : Combined KR4
 - PAM4 :
 - 2 : Single KP4
 - 3 : Combined KP4

[SCPI Example]

:SetPattern:0,18,0,1,3

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

SeRXPattern

[Description]

Set RX Pattern

[API Command]

`Public static extern bool SeRXPattern(byte channelIndex, byte patternSelect, bool isAutoLock, byte fecType);`

[API Parameter]

➤ channelIndex :

- 0 : All Channels
- 1-8 : Per Channel

➤ patternSelect :

- | | | |
|-----------|-------------------|------------|
| • NRZ : | • User Pattern: 9 | • PAM4 : |
| 0 : PN7 | | 10 : PN7Q |
| 1 : SSPRQ | | 11 : SSPRQ |
| 2 : PN9 | | 12 : PN9Q |
| 3 : PN11 | | 13 : PN11Q |
| 4 : PN13 | | 14 : PN13Q |
| 5 : PN15 | | 15 : PN15Q |
| 6 : PN16 | | 16 : PN16Q |
| 7 : PN23 | | 17 : PN23Q |
| 8 : PN31 | | 18 : PN31Q |

SeRXPattern

[API Parameter]

- isAutoLock :
 - False : Auto Lock Off
 - True : Auto Lock On
- fecType :
 - NRZ :
 - 0 : Single KR4
 - 1 : Combined KR4
 - PAM4 :
 - 2 : Single KP4
 - 3 : Combined KP4

SeRXPattern

[SCPI Command]

:SeRxPattern:{channelIndex},{patternSelect},{isAutoLock},{fecType}

[SCPI Parameter]

➤ channelIndex :

- 0 : All Channels
- 1-8 : Per Channel

➤ patternSelect :

- | | | |
|-----------|-------------------|------------|
| • NRZ : | • User Pattern: 9 | • PAM4 : |
| 0 : PN7 | | 10 : PN7Q |
| 1 : SSPRQ | | 11 : SSPRQ |
| 2 : PN9 | | 12 : PN9Q |
| 3 : PN11 | | 13 : PN11Q |
| 4 : PN13 | | 14 : PN13Q |
| 5 : PN15 | | 15 : PN15Q |
| 6 : PN16 | | 16 : PN16Q |
| 7 : PN23 | | 17 : PN23Q |
| 8 : PN31 | | 18 : PN31Q |

SeRXPattern

[SCPI Parameter]

- isAutoLock :
 - 0 : Auto Lock Off
 - 1 : Auto Lock On
- fecType :
 - NRZ :
 - 0 : Single KR4
 - 1 : Combined KR4
 - PAM4 :
 - 2 : Single KP4
 - 3 : Combined KP4

[SCPI Example]

:SeRxPattern:0,18,1,3

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

UpdateTaps

[Description]

Set Amplitude, Pre Cursor, Post Cursor, Upper Eye, Lower Eye

[API Command]

```
Public static extern bool UpdateTaps(byte channelIndex, double pre, double main, double post  
    , double upperEyeHeight, double lowerEyeHeight);
```

[API Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- pre : Pre cursor -100 to 100 (%)
- main : Amplitude 0 to 1000 (mV)
- post : Post cursor -100 to 100 (%)
- upperEyeHeight : Upper Eye Height -100 to 100 (%)
- lowerEyeHeight : Lower Eye Height -100 to 100 (%)

UpdateTaps

[SCPI Command]

:UpdateTaps:{channelIndex},{pre},{main},{post},{upperEyeHeight},{lowerEyeHeight}

[SCPI Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- pre : Pre cursor -100 to 100 (%)
- main : Amplitude 0 to 1000 (mV)
- post : Post cursor -100 to 100 (%)
- upperEyeHeight : Upper Eye Height -100 to 100 (%)
- lowerEyeHeight : Lower Eye Height -100 to 100 (%)

[SCPI Example]

:UpdateTaps:0,0,500,0,0,0

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

Update7Taps

[Description]

Set Amplitude, Pre Cursor, Post Cursor, Upper Eye, Lower Eye

[API Command]

```
Public static extern bool Update7Taps(byte channelIndex, double pre1, double pre2, double pre3, double main, double post1, double post2, double post3, double upperEyeHeight, double lowerEyeHeight);
```

[API Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- pre1, pre2, pre3 : Pre cursor -100 to 100 (%)
- main : Amplitude 0 to 1000 (mV)
- post1, post2, post3 : Post cursor -100 to 100 (%)
- upperEyeHeight : Upper Eye Height -100 to 100 (%)
- lowerEyeHeight : Lower Eye Height -100 to 100 (%)

Update7Taps

[SCPI Command]

:Update7Taps:{channelIndex},{pre1},{pre2},{pre3},{main},{post1},{post2},{post3},{upperEyeHeight},{lowerEyeHeight}

[SCPI Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- pre1, pre2, pre3 : Pre cursor -100 to 100 (%)
- main : Amplitude 0 to 1000 (mV)
- post1, post2, post3 : Post cursor -100 to 100 (%)
- upperEyeHeight : Upper Eye Height -100 to 100 (%)
- lowerEyeHeight : Lower Eye Height -100 to 100 (%)

[SCPI Example]

:Update7Taps:0,0,0,0,500,0,0,0,0,0

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

SetCTLE

[Description]

Set CTLE value

[API Command]

```
Public static extern bool SetCTLE(byte channelIndex, ushort ctleValue);
```

[API Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- ctleValue : ctleValue 0 to 8 (dB)

SetCTLE

[SCPI Command]

:SetCTLE :{channelIndex},{ctleValue}

[SCPI Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- ctleValue : ctleValue 0 to 8 (dB)

[SCPI Example]

:SetCTLE:0,2

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

SetRXSensitivity

[Description]

Set Rx Sensitivity

[API Command]

`Public static extern bool SetRXSensitivity(byte channelIndex, byte rxSensitivity);`

[API Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- rxSensitivity :
 - 0 : High Sensitivity
 - 1 : Optimized
 - 2 : High Power

SetRXSensitivity

[SCPI Command]

:SetRXSensitivity:{channelIndex},{rxSensitivity}

[SCPI Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- rxSensitivity :
 - 0 : High Sensitivity
 - 1 : Optimized
 - 2 : High Power

[SCPI Example]

:SetRXSensitivity:0,1

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

SetBERSettings

[Description]

Set BER type and timer settings. Please call this function before "StartBERTest" function

[API Command]

```
Public static extern bool SetBERSettings(byte channelIndex, byte BERTType, bool realtimeUpdate  
    , int days, int hours, int minutes, int seconds);
```

[API Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- BERTType :
 - 0 : Timed
 - 1 : Repetitive
 - 2 : Infinite
- realtimeUpdate :
 - False : Update BER on timer finish
 - True : Update BER every 1s
- days : Counter days
- hours : Counter hours
- minutes : Counter minutes
- seconds : Counter seconds

SetBERSettings

[SCPI Command]

:SetBERSettings:{channelIndex},{BERTType},{realtimeUpdate},{days},{hours},{minutes},{seconds}

[SCPI Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- BERTType :
 - 0 : Timed
 - 1 : Repetitive
 - 2 : Infinite
- realtimeUpdate :
 - False : Update BER on timer finish
 - True : Update BER every 1s
- days : Counter days
- hours : Counter hours
- minutes : Counter minutes
- seconds : Counter seconds

[SCPI Example]

:SetBERSettings:0,0,1,0,0,0,10

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

StartBERTest

[Description]

Start BER Test. When you click "RUN" it will do ClearBERTest and StartBERTest function than start to read BER test result.

[API Command]

```
Public static extern bool StartBERTest();
```

[SCPI Command]

:StartBERTest

[SCPI Example]

:StartBERTest

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

StopBERTest

[Description]

Stop BER Test.

[API Command]

```
Public static extern bool StopBERTest();
```

[SCPI Command]

:StopBERTest

[SCPI Example]

:StopBERTest

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

IsBERTRunning

[Description]

Is BER Test is running?

[API Command]

`Public static extern bool IsBERTRunning();`

[SCPI Command]

:IsBERTRunning

[SCPI Example]

:IsBERTRunning

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

ClearBERTest

[Description]

Clear BER Result.

[API Command]

```
Public static extern bool ClearBERTest(byte channelIndex);
```

[API Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel

ClearBERTest

[SCPI Command]

:ClearBERTest:{channelIndex}

[SCPI Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel

[SCPI Example]

:ClearBERTest:0

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

Relock

[Description]

Re-lock All Channels

[API Command]

```
Public static extern bool Relock();
```

[SCPI Command]

:Relock

[SCPI Example]

:Relock

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

ForceRelock

[Description]

Force re-lock All Channels

[API Command]

```
Public static extern bool ForceRelock();
```

[SCPI Command]

:ForceRelock

[SCPI Example]

:ForceRelock

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

PPMTune

[Description]

PPM Tuning. Re-lock is required when wide range tuning is applied.

[API Command]

```
Public static extern bool PPMTune(bool read, ref short ppmValue);
```

[API Parameter]

- read :
 - False : write
 - True : read
- ppmValue : Data rate PPM tune -300 to 300 (ppm)

PPMTune

[SCPI Command]

:PPMTune:{read},{ppmValue}

[SCPI Parameter]

- read :
 - 0 : write
 - 1 : read
- ppmValue : Data rate PPM tune -300 to 300 (ppm)

[SCPI Example]

:PPMTune:0,100

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail

ErrorInjection

[Description]

When start BER testing, you can call "ErrorInjection" to injection error bit

[API Command]

```
Public static extern bool ErrorInjection(byte channelIndex, byte bitSelect, byte type, byte gap, byte PktCount);
```

[API Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- bitSelect :
 - 0 : MSB
 - 1 : LSB
 - 2 : MSB & LSB or NRZ
- type :
 - 0 : Single Error Per Packet (64bit)
 - 1 : Burst Error Per Packet
- gap : 0-127 Error Free Packets Gap between injected packets
- PktCount : 0-127 Packet injected with Errors

ErrorInjection

[SCPI Command]

:ErrorInjection:{channelIndex},{bitSelect},{type},{gap},{PktCount}

[SCPI Parameter]

- channelIndex :
 - 0 : All Channels
 - 1-8 : Per Channel
- bitSelect :
 - 0 : MSB
 - 1 : LSB
 - 2 : MSB & LSB or NRZ
- type :
 - 0 : Single Error Per Packet (64bit)
 - 1 : Burst Error Per Packet
- gap : 0-127 Error Free Packets Gap between injected packets
- PktCount : 0-127 Packet injected with Errors

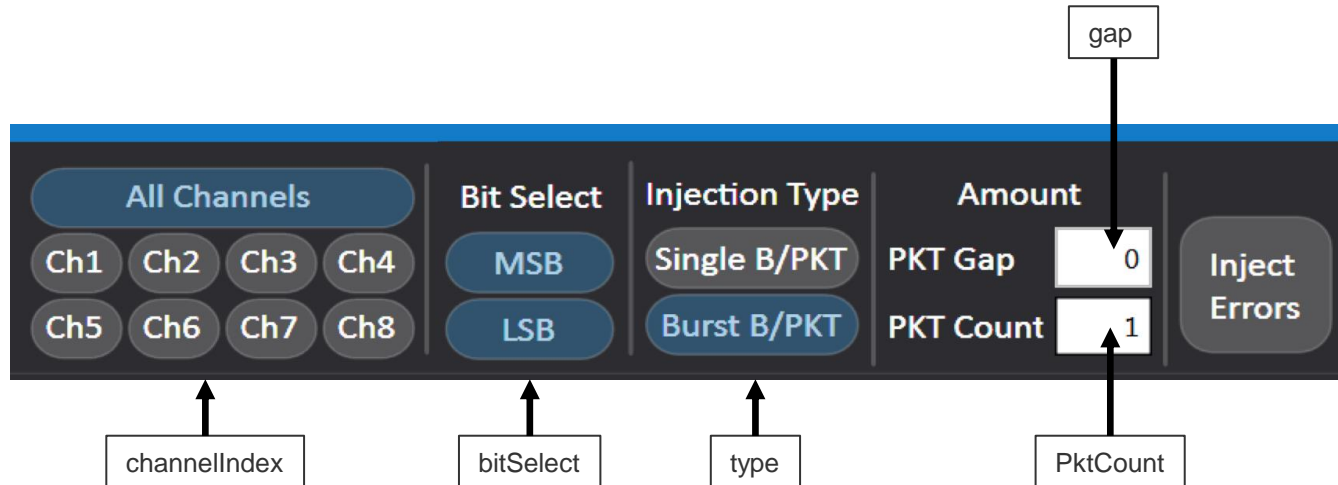
ErrorInjection

[SCPI Example]

:ErrorInjection:0,0,0,1,0

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail



ReadBERResult

This command is kept to support the codes using previous API versions.

[Description]

After BER test starting, using this function to read BER test result.

[API Command]

```
Public static extern bool ReadBERResult(ref long captureTimeIns, byte[] rxPatternMSB, byte[] rxPatternLSB
    , byte[] rxLockMSB, byte[] rxLockLSB, byte[] rxLock
    , byte[] rxInvertMSB, byte[] rxInvertLSB
    , UInt64[] bertErrorCountMSB, UInt64[] bertErrorCountLSB
    , UInt64[] bertErrorCount, UInt64[] bertBitCount
    , double[] realTimer, double[] bertValue, UInt64[] fecCOR
    , double[] fecBertValues, UInt64[] fecResults);
```

[SCPI Command]

:ReadBERResult:{channelIndex}

[SCPI Parameter]

- channelIndex :
 - 1-8 : Per Channel

***Note : SCPI Not support read all channels**

ReadBERResultMargin

This command is kept to support the codes using previous API versions.

[Description]

After BER test starting, using this function to read BER test result.

[API Command]

```
Public static extern bool ReadBERResultMargin(ref long captureTimeIns, byte[] rxPatternMSB, byte[] rxPatternLSB, byte[] rxLockMSB, byte[] rxLockLSB, byte[] rxLock, byte[] rxInvertMSB, byte[] rxInvertLSB, UInt64[] bertErrorCountMSB, UInt64[] bertErrorCountLSB, UInt64[] bertErrorCount, UInt64[] bertBitCount, double[] realTimer, double[] bertValue, UInt64[] fecCOR, double[] fecBertValues, UInt64[] fecResults, double[] margin, double[] marginPct, sbyte[] taps);
```

[SCPI Command]

:ReadBERResultMargin:{channelIndex}

[SCPI Parameter]

- channelIndex :
 - 1-8 : Per Channel

***Note : SCPI Not support read all channels**

ReadBERResultMarginV2

This command is kept to support the codes using previous API versions.

[Description]

After BER test starting, using this function to read BER test result. Update channel response UI supports up to UI-16.

[API Command]

```
Public static extern bool ReadBERResultMarginV2(ref long captureTimeIns, byte[] rxPatternMSB, byte[] rxPatternLSB, byte[] rxLockMSB, byte[] rxLockLSB, byte[] rxLock, byte[] rxInvertMSB, byte[] rxInvertLSB, UInt64[] bertErrorCountMSB, UInt64[] bertErrorCountLSB, UInt64[] bertErrorCount, UInt64[] bertBitCount, double[] realTimer, double[] bertValue, UInt64[] fecCOR, double[] fecBertValues, UInt64[] fecResults, double[] margin, double[] marginPct, sbyte[] taps);
```

[SCPI Command]

:ReadBERResultMarginV2:{channelIndex}

[SCPI Parameter]

- channelIndex :
 - 1-8 : Per Channel

***Note : SCPI Not support read all channels**

ReadBERResultMarginV3

[Description]

After BER test starting, using this function to read BER test result.

[API Command]

```
Public static extern bool ReadBERResultMarginV3(ref long captureTimeIns, byte[] rxPatternMSB, byte[] rxPatternLSB
, byte[] rxLockMSB, byte[] rxLockLSB, byte[] rxLock
, byte[] rxInvertMSB, byte[] rxInvertLSB
, UInt64[] bertErrorCountMSB, UInt64[] bertErrorCountLSB
, UInt64[] bertErrorCount, UInt64[] bertBitCount
, double[] realTimer, double[] bertValue, UInt64[] fecCOR , UInt64[] fecUnCOR
, double[] fecBertValues, UInt64[] fecResults, double[] margin
, double[] marginPct, sbyte[] taps);
```


ReadBERResultMarginV4

In FGC mode, please use this function!

[Description]

After BER test starting, using this function to read BER test result.

[API Command]

```
Public static extern bool ReadBERResultMarginV4(ref long captureTimeIns, byte[] rxPatternMSB, byte[] rxPatternLSB, byte[] rxLockMSB, byte[] rxLockLSB, byte[] rxLock, byte[] rxInvertMSB, byte[] rxInvertLSB, UInt64[] bertErrorCountMSB, UInt64[] bertErrorCountLSB, UInt64[] bertErrorCount, UInt64[] bertBitCount, double[] realTimer, double[] bertValue, UInt64[] fecCOR, UInt64[] fecUnCOR, double[] cerValues, double[] flrValues, UInt64[] fecResults, double[] margin, double[] marginPct, sbyte[] taps);
```

ReadSelectedPPGOptions

This command is kept to support the codes using previous API versions.

[Description]

Read Selected PPG Options

[API Command]

```
public static extern bool ReadSelectedPPGOptions(ref bool basic, ref bool multiRate, ref bool OneP5Vpp, ref bool FEC, ref bool FiftyThreeG, ref bool isPAM4, ref bool is8Ch, ref bool isMA);
```

[API Parameter]

- basic : Basic mode Enable/Disable
- multiRate : MultiRate mode Enable/Disable
- OneP5Vpp : OneP5Vpp mode Enable/Disable
- FEC : FEC mode Enable/Disable
- FiftyThreeG : FiftyThreeG mode Enable/Disable
- isPAM4 : Return current BA support PAM4 or not
- is8Ch : Return current BA is 8 or 4 channels
- isMA : Return instrument is BA or MA

ReadSelectedPPGOptions

This command is kept to support the codes using previous API versions.

[SCPI Command]

:ReadSelectedPPGOptions

[SCPI Example]

:ReadSelectedPPGOptions

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail
- Array [32] : Return Basic mode
 - 1 : Enable
 - 0 : Disable
- Array [33] : Return multiRate mode
 - 1 : Enable
 - 0 : Disable
- Array [34] : Return OneP5Vpp mode
 - 1 : Enable
 - 0 : Disable
- Array [35] : Return FEC mode
 - 1 : Enable
 - 0 : Disable
- Array [36] : Return FiftyThreeG mode
 - 1 : Enable
 - 0 : Disable
- Array [37] : Return isPAM4
 - 1 : Support PAM4
 - 0 : No support PAM4
- Array [38] : Return is8Ch
 - 1 : the board is 8 Channels
 - 0 : the board is 4 Channels
- Array [39] : Return isMA
 - 1 : MA
 - 0 : BA

ReadSelectedPPGOptionsV2

[Description]

Read Selected PPG Options

[API Command]

```
public static extern bool ReadSelectedPPGOptionsV2(ref bool basic, ref bool multiRate, ref bool OneP5Vpp, ref bool FEC
, ref bool FiftyThreeG, ref bool isPAM4, ref bool is8Ch , ref bool isMA
, ref bool is25GNRZ);
```

[API Parameter]

- basic : Basic mode Enable/Disable
- multiRate : MultiRate mode Enable/Disable
- OneP5Vpp : OneP5Vpp mode Enable/Disable
- FEC : FEC mode Enable/Disable
- FiftyThreeG : FiftyThreeG mode Enable/Disable
- isPAM4 : Return current BA support PAM4 or not
- is8Ch : Return current BA is 8 or 4 channels
- isMA : Return instrument is BA or MA
- is25GNRZ : Return instrument is 28-NRZ series or not

ReadSelectedPPGOptionsV3

In FGC mode, please use this function!

[Description]

Read Selected PPG Options

[API Command]

```
public static extern bool ReadSelectedPPGOptionsV3(ref byte options);
```

[API Parameter]

- Array[0] = **Basic** mode Enable (1) /Disable (0)
- Array[1] = **MultiRate** mode Enable (1) /Disable (0)
- Array[2] = **OneP5Vpp** mode Enable (1) /Disable (0)
- Array[3] = **FEC** mode Enable (1) /Disable (0)
- Array[4] = **FGC** mode Enable (1) /Disable (0)
- Array[5] = **FiftyThreeG** mode Enable (1) /Disable (0)
- Array[6] = **isPAM4** return current BA support PAM4 or not [Read Only]
- Array[7] = **is8Ch** return current BA is 8 or 4 channels [Read Only]
- Array[8] = **isMA** return instrument is BA or MA [Read Only]
- Array[9] = **is25GNRZ** return instrument is 28-NRZ series or not [Read Only]
- Array[10] = **is53GB** return instrument is 56-PAM series or not [Read Only]
- Array[11 – 31] = Reserved

ReadDataRate

This command is kept to support the codes using previous API versions.

[Description]

Read data rate, signal type (NRZ/PAM4), clock and mapping.

[API Command]

```
public static extern bool ReadDataRate(ref byte signalMode, ref int dataRate, ref byte clockDiv  
                                     , ref byte mapping, ref byte coding);
```

[API Parameter]

➤ signalMode :

- 0 : PAM4
- 1 : NRZ

➤ dataRate : Read DataRate in MBd

- NRZ : 28900000, 28050000, 26562500, 25781250, 24330240, 14025000
 , 12500000, 11317600, 10709000, 10312500, 9953280
- PAM4 : 57800000, 53125000, 28900000, 28050000, 26562500

➤ clockDiv :

- 0 : Rate/8
- 1 : Rate/16
- 2 : Rate/32
- 3 : Rate/64

➤ mapping :

- 0 : Gray Mapping
- 1 : Linear Mapping

➤ coding :

- 0 : Off
- 1 : On

ReadDataRate

This command is kept to support the codes using previous API versions.

[SCPI Command]

:ReadDataRate

[SCPI Example]

:ReadDataRate

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail
- Array [32] : Return signalMode
 - 0 : PAM4
 - 1 : NRZ
- Array [33] : Return DataRate in MBd
- Array [34] : Return clockDiv
 - 0 : Rate/8
 - 1 : Rate/16
 - 2 : Rate/32
 - 3 : Rate/64
- Array [35] : mapping
 - 0 : Gray Mapping
 - 1 : Linear Mapping
- Array [36] : coding
 - 0 : Off
 - 1 : On

ReadDataRateAdv

[Description]

Read data rate, signal type (NRZ/PAM4), clock and mapping.

[API Command]

```
public static extern bool ReadDataRateAdv(ref byte signalMode, ref int dataRate, ref byte clockDiv  
    , ref byte mapping, ref byte preCoding);
```

[API Parameter]

- signalMode :
 - 0 : PAM4
 - 1 : NRZ
- dataRate : Read DataRate in MBd
 - NRZ : 28900000, 28050000, 26562500, 25781250, 24330240, 14025000
 , 12500000, 11317600, 10709000, 10312500, 9953280
 - PAM4 : 57800000, 53125000, 28900000, 28050000, 26562500
- clockDiv :
 - 0 : Rate/8
 - 1 : Rate/16
 - 2 : Rate/32
 - 3 : Rate/64
- mapping :
 - 0 : Gray Mapping
 - 1 : Linear Mapping
- preCoding :
 - 0 : Off
 - 1 : On

ReadDataRateAdv

[SCPI Command]

:ReadDataRateAdv

[SCPI Example]

:ReadDataRateAdv

[SCPI Return]

- Array [0-30] : Return Function Name (ASCII)
- Array [31] : Function Return
 - 1 : Function Success
 - 0 : Function Fail
- Array [32] : Return signalMode
 - 0 : PAM4
 - 1 : NRZ
- Array [33] : Return DataRate in MBd
- Array [34] : Return clockDiv
 - 0 : Rate/8
 - 1 : Rate/16
 - 2 : Rate/32
 - 3 : Rate/64
- Array [35] : mapping
 - 0 : Gray Mapping
 - 1 : Linear Mapping
- Array [36] : preCoding
 - 0 : Off
 - 1 : On

ReadPatternData

[Description]

Read data rate, signal type (NRZ/PAM4), clock and mapping.

[API Command]

```
public static extern bool ReadPatternData(byte[] patternTX, double[] preCursor, double[] amplitude, double[] postCursor, double[] upperEyeHeight, double[] lowerEyeHeight, byte[] txInvert, byte[] patternRX, byte[] patternRXLSB, bool[] rxAutoLock, byte[] rxInvert, byte[] fecModeSelect, byte[] rxSensitivity, ushort[] ctle);
```

[SCPI Command]

:ReadPatternData:{channelIndex}

[SCPI Parameter]

- channelIndex :
 - 1-8 : Per Channel

***Note : SCPI Not support read all channels**

ReadPatternData7T

[Description]

Read data rate, signal type (NRZ/PAM4), clock and mapping.

[API Command]

```
public static extern bool ReadPatternData7T(ref byte is7Taps, byte[] patternTX, double[] preCursor1, double[] preCursor2, double[] preCursor, double[] amplitude, double[] postcursor, double[] postCursor2, double[] postCursor3, double[] upperEyeHeight, double[] lowerEyeHeight, byte[] txInvert, byte[] patternRX, byte[] patternRXLSB, bool[] rxAutoLock, byte[] rxInvert, byte[] fecModeSelect, byte[] rxSensitivity, ushort[] ctle);
```

[SCPI Command]

:ReadPatternData7T:{channelIndex}

[SCPI Parameter]

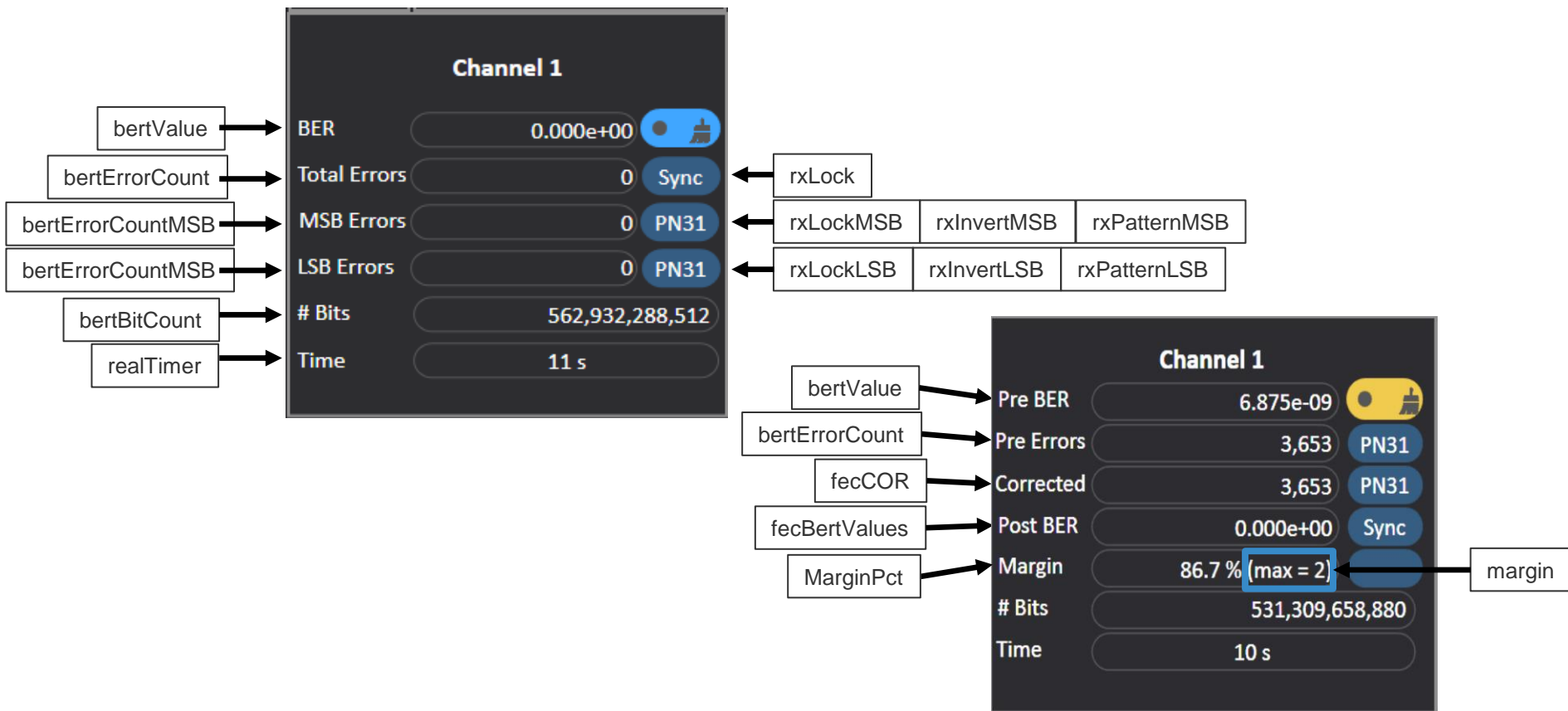
- channelIndex :
 - 1-8 : Per Channel

***Note : SCPI Not support read all channels**

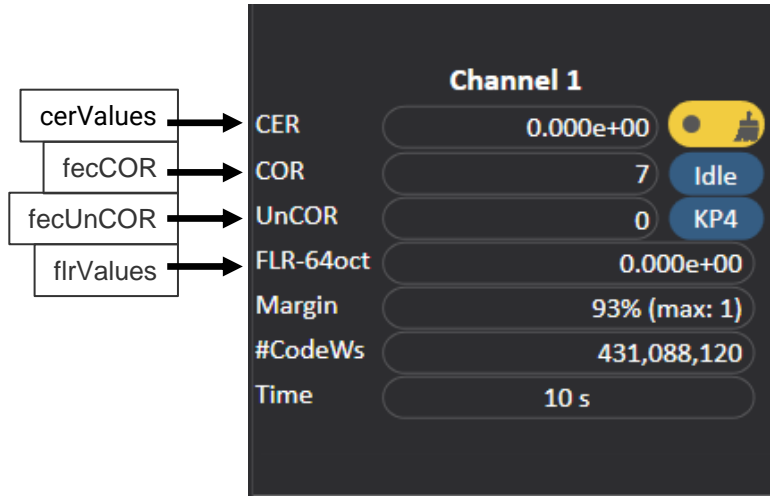
Appendix A

Read BER Function Description

Description of various read BER function



Description of various read BER function



Description of various read BER function

[Parameter]

- captureTimeIns : Capture Time in s (date = capture time ins seconds since 1970)
- rxPatternMSB : RX Pattern Status MSB (Array[9])
- rxPatternLSB : RX Pattern Status LSB (Array[9])

- NRZ :
 - 0 : PN7
 - 1 : SSPRQ
 - 2 : PN9
 - 3 : PN11
 - 4 : PN13
 - 5 : PN15
 - 6 : PN16
 - 7 : PN23
 - 8 : PN31
- PAM4 :
 - 10 : PN7Q
 - 11 : SSPRQ
 - 12 : PN9Q
 - 13 : PN11Q
 - 14 : PN13Q
 - 15 : PN15Q
 - 16 : PN16Q
 - 17 : PN23Q
 - 18 : PN31Q

Note 1: This param need a array (array length 9) to get the output data

Index	0	1	2	3	4	5	6	7	8
	NA	Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	Ch7	Ch8

Description of various read BER function

[Parameter]

- rxLockMSB : RX Lock Status MSB (Array[9])
- rxLockLSB : RX Lock Status LSB (Array[9])
- rxLock : RX Lock Status (Array[9])
 - Return 0 : LOL
 - Return 1 : Locked
 - Return 2 : LOS

Note 1: This param need a array (array length 9) to get the output data

Index	0	1	2	3	4	5	6	7	8
	NA	Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	Ch7	Ch8

Description of various read BER function

[Parameter]

- rxInvertMSB : RX Invert Status MSB (Array[9])
- rxInvertLSB : RX Invert Status LSB (Array[9])
- bertErrorCountMSB : MSB Error Count Detected (Array[9])
- bertErrorCountLSB : LSB Error Count Detected (Array[9])
- bertErrorCount : Sum of Error Count Detected (Array[9])
- bertBitCount : Number of bits counted (Array[9])
- realTimer : Real BER Timer based on Captured Bits in seconds (Array[9])
- bertValue : BER Test Results (Array[9])
- fecCOR : FEC Correctable Errors (Array[9])
- fecUnCOR : FEC Un-Correctable Errors (Array[9])
- fecBertValues : BER Test Results After Correction (Array[9])
- margin : Margin (Array[9])
- MarginPct : Margin Percentage (Array[9])
- cerValues : codeword error ratio results (Array[9])
- flrValues : frame loss ratio (64 oct) Results (Array[9])

Note 1: This param need a array (array length 9) to get the output data

Index	0	1	2	3	4	5	6	7	8
	NA	Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	Ch7	Ch8

Description of various read BER function

[Parameter]

- fecResults : Detailed FEC Results (Array[432])

FEC Result Map

Index	Content
0~48	NA
49~80	Ch1 Codewords Count
81~96	NA
97~128	Ch2 Codewords Count
129~144	NA
145~176	Ch3 Codewords Count
175~192	NA
193~224	Ch4 Codewords Count
225~240	NA
241~272	Ch5 Codewords Count
273~287	NA
289~320	Ch6 Codewords Count
321~336	NA
337~368	Ch7 Codewords Count
369~384	NA
385~416	Ch8 Codewords Count
417~431	NA

Index	Content
0	Codewords of Symbol Error 1
1	Codewords of Symbol Error 2
2	Codewords of Symbol Error 3
3	Codewords of Symbol Error 4
4	Codewords of Symbol Error 5
5	Codewords of Symbol Error 6
6	Codewords of Symbol Error 7
7	Codewords of Symbol Error 8
8	Codewords of Symbol Error 9
9	Codewords of Symbol Error 10
10	Codewords of Symbol Error 11
11	Codewords of Symbol Error 12
12	Codewords of Symbol Error 13
13	Codewords of Symbol Error 14
14	Codewords of Symbol Error 15
15	Codewords of Symbol Error 16
16	Codewords of Symbol Error 17
17	Codewords of Symbol Error 18
18	Codewords of Symbol Error 19
19	Codewords of Symbol Error 20
20	Codewords of Symbol Error 21
21	Codewords of Symbol Error 22
22	Codewords of Symbol Error 23
23	Codewords of Symbol Error 24
24	Codewords of Symbol Error 25
25	Codewords of Symbol Error 26
26	Codewords of Symbol Error 27
27	Codewords of Symbol Error 28
28	Codewords of Symbol Error 29
29	Codewords of Symbol Error 30
30	Codewords of Symbol Error 31
31	Codewords of Saturation

Description of various read BER function

[Parameter]

- fecResults : Detailed FGC Results (Array[432])

FGC Result Map

Index	Content
0~48	NA
49~80	Ch1 Codewords Count
81~96	NA
97~128	Ch2 Codewords Count
129~144	NA
145~176	Ch3 Codewords Count
175~192	NA
193~224	Ch4 Codewords Count
225~240	NA
241~272	Ch5 Codewords Count
273~287	NA
289~320	Ch6 Codewords Count
321~336	NA
337~368	Ch7 Codewords Count
369~384	NA
385~416	Ch8 Codewords Count
417~431	NA

Index	Content
0	Codewords of Symbol Error 1
1	Codewords of Symbol Error 2
2	Codewords of Symbol Error 3
3	Codewords of Symbol Error 4
4	Codewords of Symbol Error 5
5	Codewords of Symbol Error 6
6	Codewords of Symbol Error 7
7	Codewords of Symbol Error 8
8	Codewords of Symbol Error 9
9	Codewords of Symbol Error 10
10	Codewords of Symbol Error 11
11	Codewords of Symbol Error 12
12	Codewords of Symbol Error 13
13	Codewords of Symbol Error 14
14	Codewords of Symbol Error 15
15	Codewords of Symbol Error 16
16	Codewords of Saturation

Description of various read BER function

[Parameter]

➤ UI : Ch Response UI(Array[90]) *Only for **ReadBERResultMargin**

Array[]	UI-1	UI-2	UI-3	UI-4	UI-5	UI-6	UI-7	UI-8	UI-9	UI-10
Ch1	10	11	12	13	14	15	16	17	18	19
Ch2	20	21	22	23	24	25	26	27	28	29
Ch3	30	31	32	33	34	35	36	37	38	39
Ch4	40	41	42	43	44	45	46	47	48	49
Ch5	50	51	52	53	54	55	56	57	58	59
Ch6	60	61	62	63	64	65	66	67	68	69
Ch7	70	71	72	73	74	75	76	77	78	79
Ch8	80	81	82	83	84	85	86	87	88	89

Description of various read BER function

[Parameter]

➤ UI : Ch Response UI(Array[144]) *Only for **ReadBERResultMarginV2**

Array[]	UI-1	UI-2	UI-3	UI-4	UI-5	UI-6	UI-7	UI-8	UI-9	UI-10	UI-11	UI-12	UI-13	UI-14	UI-15	UI-16
Ch1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Ch2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Ch3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Ch4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
Ch5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
Ch6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
Ch7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
Ch8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143

Appendix B

SCPI return Byte define

SCPI return Byte define

All SCPI function return byte array which include Function Name, Function Return, and Other Data.

Index	0-30	31	32-
	Function Name	Function Return	Other Data

- Function Name : Array[0] – Array[30] return Function Name, Please convert to ASCII string
- Function Return : Array[31] return function is True or False
- Other Data : After Array[32] return other data, user must convert to the corresponding data type and suitable size length.

Example for ReadBERResult:

return data	array size	Array index									
long captureTimeIns,	8	32	33	34	35	36	37	38	39		
byte[] rxPatternMSB,	1	40									
byte[] rxPatternLSB	1	41									
byte[] rxLockMSB,	1	42									
byte[] rxLockLSB,	1	43									
byte[] rxLock,	1	44									
byte[] rxInvertMSB,	1	45									
byte[] rxInvertLSB,	1	46									
UInt64[] bertErrorCountMSB,	8	47	48	49	50	51	52	53	54		
UInt64[] bertErrorCountLSB,	8	55	56	57	58	59	60	61	62		
UInt64[] bertErrorCount,	8	63	64	65	66	67	68	69	70		
UInt64[] bertBitCount,	8	71	72	73	74	75	76	77	78		
double[] realTimer,	8	79	80	81	82	83	84	85	86		
double[] bertValues,	8	87	88	89	90	91	92	93	94		
UInt64[] fecCOR,	8	95	96	97	98	99	100	101	102		
double[] fecBertValues,	8	103	104	105	106	107	108	109	110		
UInt64[] fecResults	8	111	112	113	114	115	116	117	118		

The logo for EXFO, featuring the word "EXFO" in a bold, white, sans-serif font. The letters are stylized with horizontal lines running through them, giving it a digital or network-like appearance. The background is a gradient of teal and blue, with abstract geometric shapes and network diagrams in the corners.

EXFO