



## Operation Manual

### For controlling IQ Bias Controllers using UART

*Applicable for MBC-IQ series*

#### 1) Connection



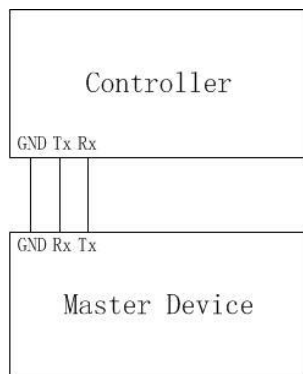
This blue unit, which is USB-UART Converter, transforms between USB and UART(TTL232). Connect it with your PC through USB. It will translate the USB data into UART(TTL232) form, which will be understood by Bias Controllers.

It can be seen from above diagram that GND, TXD and RXD are connected with wires.

GND: ground

TXD: transmit

RXD: receive



**(In this case, Master Device is UART-USB converter)**

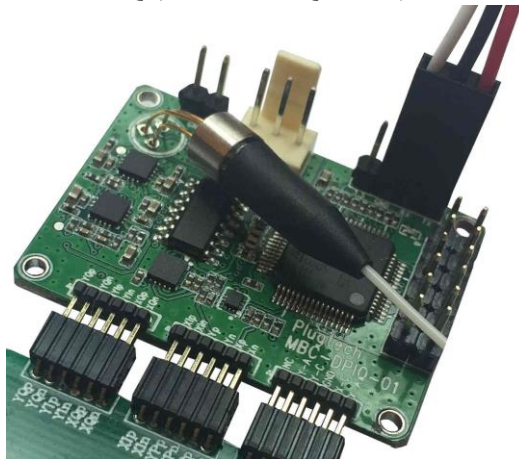
TX of this blue unit shall be connected with the RX of bias controller.

RX of this blue unit shall be connected with the TX of bias controller.

For SP-IQ (MBC-IQ series) controller: (Please take care of the color)



For DP-IQ (MBC-DPIQ series) controller:



## 2) PC Driver for the USB-UART Converter

To use USB-UART Converter, which is the blue unit in above diagram, properly on your computer, essential drivers must be installed properly.

Driver can be found in the Driver folder, PL2303\_Prolific\_DriverInstaller\_v1.8.0.exe is the execution file to install the driver. It can also be downloaded at following links.

For PL2303 Module:

Driver download address:

[http://www.waveshare.net/w/upload/6/64/PL2303\\_Windows\\_Driver.7z](http://www.waveshare.net/w/upload/6/64/PL2303_Windows_Driver.7z)

Other information:

[http://www.waveshare.net/wiki/PL2303-USB-UART-Board\\_Software](http://www.waveshare.net/wiki/PL2303-USB-UART-Board_Software)

For FT232 Module:

Driver download address (32 bit windows):

[http://www.waveshare.net/w/upload/1/1f/FT232\\_Driver.7z](http://www.waveshare.net/w/upload/1/1f/FT232_Driver.7z)

Driver download address (64 bit windows):

[http://www.waveshare.net/w/upload/4/49/CDM\\_v2.12.06\\_WHQL\\_Certified.zip](http://www.waveshare.net/w/upload/4/49/CDM_v2.12.06_WHQL_Certified.zip)

Other information:

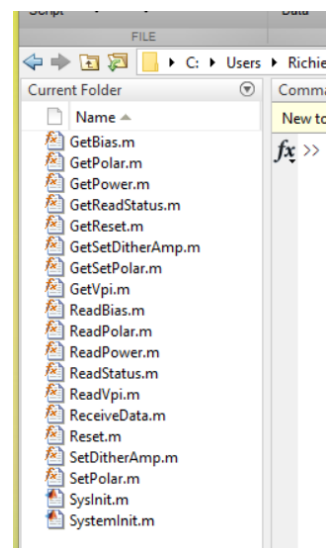
<http://www.waveshare.net/w/upload/d/d0/FT232-USB-UART-Board-UserManual.pdf>

### 3) Sample for changing Dither Amplitude of bias controllers

a) Firstly, change Matlab working directory to MatlabFiles



Then, you will find several functions available at the left navigation bar.



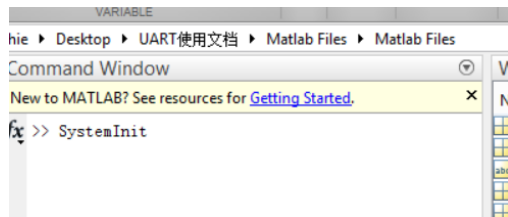
b) Open SystemInit.m file, change comName to the com port name that the USB-UART Converter installed on your computer. You can check the COM port number in device manager if you are using Windows. In this case, we use COM7. There are 6 variables representing the maximum and minimum output voltage of controller. User should update the value according to the test report. For example, on test report, I-DAC output max is 14.4V, then variable upperLimit\_I in SystemInit.m should be changed to 14.4.

```
8- comName = 'COM7';
9- lowerLimit_I = -14.604;
0- upperLimit_I = 14.467;
1- lowerLimit_Q = -14.546;
2- upperLimit_Q = 14.372;
3- lowerLimit_P = -14.498;
4- upperLimit_P = 14.349;
```

Test Report Sample

16	I-DAC output range	Max: Min:
17	Q-DAC output range	Max: Min:
18	P-DAC output range	Max: Min:

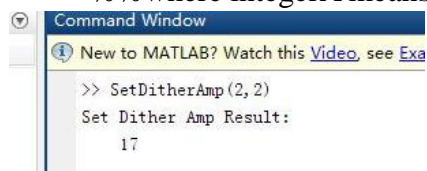
### c) Run **SystemInit**



Succeed message will show up.



### d) Run **SetDitherAmp (integerA, integerA);** %%where integerA means an integer constant.



### **SetDitherAmp (integerA, integerA):**

integerA means the dither amplitude will be set to integerA % of Vpi(DC). By default, integerA is equal to 1. It shall work for most cases. IntegerA has to be integers and range from 1 to 20.

In above diagram, dither amplitude has been set to 2% of Vpi. The return value is 17, which indicates that the operation is successful.

By this, the dither amplitude of the controller has been set. The controller will restart immediately.

### **4) Sample for Change modulation format seamlessly**

#### **Operation principle:**

For different modulation formats, such as QPSK or QAM, the controller has different control parameters to stabilize the whole system. If one uses QPSK control parameters for QAM modulation, the system performance would be harmed.

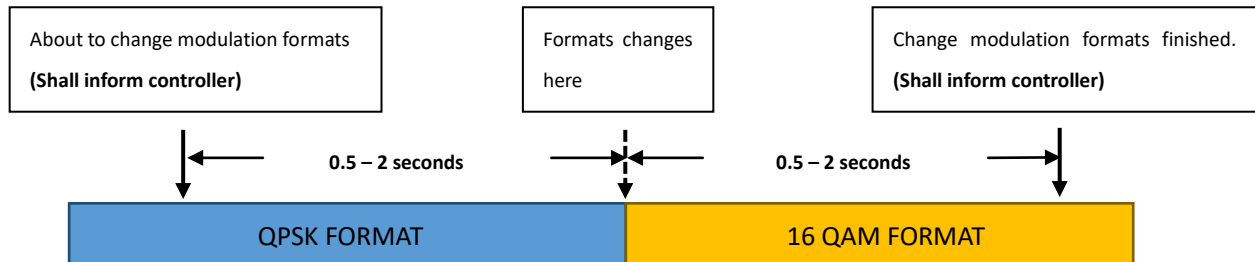
The control parameters shall be recorded for different modulation formats first and stored in controller. When user needs to switch between different modulation formats, corresponding control parameters will be retrieved in advance. In this controller, five different sets of control parameters could be stored.

This function records control parameters for different modulation formats. When user changes modulation formats, controller shall be informed in advance, then controller

will update control parameters accordingly.

*Notes: The phase part of the modulator for different modulation formats shall be the same ( $Q+$  or  $Q-$ ). Otherwise, the phase part of the control voltage will have to transit between  $Q+$  and  $Q-$ . And this transit can hardly be synchronized with transit of modulation formats.*

### Timing diagram:



### Procedures:

- i) At the very beginning, the control parameters shall be recorded.

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read
>> StoreParam(1)
Store Parameter Result:
17
```

In this case, StoreParam(1) was used. It means that control parameters has been stored in the controller. And the indicator for this set of control parameters is 1. A total of 5 sets of control parameters can be stored.

- ii) A period of time(0.5 to 2 seconds) before changing the modulation formats, the controller shall be informed that a modulation format switch is about to happen using *ChangeOperationParam(setNo, from 1 to 5);*

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read
>> ChangeOperationParam(1)
Change Operation Param Result:
17
```

- iii) A short while (0.5 to 2 seconds) after changing the modulation format already, the controller shall be informed again that the modulation format has been updated successfully using *ChangeOperationParamFinish;*

```
Command Window
New to MATLAB? Watch this Video, see Examples, or read
>> ChangeOperationParamFinish
Change Operation Param Finish Result:
17
```

### 5) You can also try other functions. Such as

**Reset**      %% The system will be reset.

**ReadBias**    %%Current bias voltages will be returned.

**ReadVpi**    %%Vpi of each arm will be returned.

And many more! We also provide advanced functions like:

- a. Control at ANY point around null point
- b. Modulation format seamlessly transition
- c. Customized functions

**6) Detailed function descriptions can be found at *UART Operation Manual*.**

7) PS: Do not use functions start with Get, such as GetSetDitherAmp. These functions are for system use purposes.