

Mirrorcle Technologies, Inc.

Application Note AN009

ANALOG INPUT TO OUTPUT DATAMODE

Last Revised: Oct. 2020

Mirrorcle Technologies, Inc.



Overview

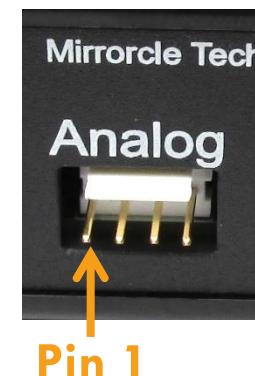
- The “Analog_Input_To_Output” DataMode allows the device to sample the analog inputs at maximum speed and use these values as normalized coordinates to drive the output.
- In enabling this DataMode, it is important to be familiar with the following:
 - The USB-SL MZ Pinout
 - Safely transitioning between DataModes
- An example can be found in the C++ SDK inside MTIDevice-Demo.cpp.

USB-SL MZ Pinout

- When enabling Analog Input to Output DataMode, it is important to connect to the analog connector according to the pinout.
 - Pin 1 is a +3.3V Output, NOT an input
 - Pin 2 is Analog Input Channel 0
 - Pin 3 is Analog Input Channel 1
 - Pin 4 is the ground



Analog Connector: 4 - Pin Header		
J7-Pin	Name	Description
1	+3.3V	+3.3V Output, limited to 25mA (Do not input 3.3V. Output Pin for Optional Customer Use)
2	AI0	Analog Input Channel 0
3	AI1	Analog Input Channel 1
4	Gnd	Ground



Set Device Parameters before Enabling

- **MTIDeviceParams** must be set **before** switching DataModes

```
void AnalogInputToOutputDemo ( MTIDevice* mti )
{
    MTIDeviceParams sParams;
    mti->GetDeviceParams(&sParams); //back up current params from controller
    //Set any MEMS Driver Settings for Analog Input to Output Mode
    MTIDeviceParams tParams;
    mti->GetDeviceParams(&tParams); //create temp params for controller
    tParams.DataScale = 1.0;
    tParams.Vbias = 70;
    tParams.VdifferenceMax = 100;
    tParams.HardwareFilterBw = 200;
    tParams.SampleRate = 50000;
    mti->SetDeviceParams( &tParams );

    mti->ResetDevicePosition();
    mti->StopDataStream();
    mti->SetDeviceParam( MTIParam::DataMode, MTIDataMode::Analog_Input_To_Output );

    system(CLEARSCREEN);
    printf("Device will read analog inputs and convert the signals to MEMS drive voltages\n");
    printf("Press ANY KEY to set controller into Analog_Input_To_Output DataMode...\n");
    _getch();
    mti->StartDataStream ( );

    printf("Press ANY KEY to return controller to Sample_Output DataMode...\n");
    _getch();
    mti->StopDataStream();
    mti->SetDeviceParam( MTIParam::DataMode, MTIDataMode::Sample_Output );
    mti->StartDataStream();
    mti->ResetDevicePosition ();
    mti->SetDeviceParams( &sParams ); //Restore original params back to controller
}
```

Stop Data Stream before Switching

- It is important to call **ResetDevicePosition** and **StopDataStream** before calling the **SetDeviceParam** method to switch to Analog Input to Output DataMode

```
void AnalogInputToOutputDemo ( MTIDevice* mti )
{
    MTIDeviceParams sParams;
    mti->GetDeviceParams(&sParams); //back up current params from controller
    //Set any MEMS Driver Settings for Analog Input to Output Mode
    MTIDeviceParams tParams;
    mti->GetDeviceParams(&tParams); //create temp params for controller
    tParams.DataScale = 1.0;
    tParams.Vbias = 70;
    tParams.VdifferenceMax = 100;
    tParams.HardwareFilterBw = 200;
    tParams.SampleRate = 50000;
    mti->SetDeviceParams( &tParams );

    mti->ResetDevicePosition();
    mti->StopDataStream();
    mti->SetDeviceParam( MTIParam::DataMode, MTIDataMode::Analog_Input_To_Output );

    system(CLEARSCREEN);
    printf("Device will read analog inputs and convert the signals to MEMS drive voltages\n");
    printf("Press ANY KEY to set controller into Analog_Input_To_Output DataMode...\n");
    _getch();
    mti->StartDataStream ( );

    printf("Press ANY KEY to return controller to Sample_Output DataMode...\n");
    _getch();
    mti->StopDataStream();
    mti->SetDeviceParam( MTIParam::DataMode, MTIDataMode::Sample_Output );
    mti->StartDataStream();
    mti->ResetDevicePosition ();
    mti->SetDeviceParams( &sParams ); //Restore original params back to controller
}
```

Start the Data Stream

- Call `StartDataStream` to begin streaming data in the Analog Input to Output DataMode

```
void AnalogInputToOutputDemo ( MTIDevice* mti )
{
MTIDeviceParams sParams;
mti->GetDeviceParams(&sParams); //back up current params from controller
//Set any MEMS Driver Settings for Analog Input to Output Mode
MTIDeviceParams tParams;
mti->GetDeviceParams(&tParams); //create temp params for controller
tParams.DataScale = 1.0;
tParams.Vbias = 70;
tParams.VdifferenceMax = 100;
tParams.HardwareFilterBw = 200;
tParams.SampleRate = 50000;
mti->SetDeviceParams( &tParams );

mti->ResetDevicePosition();
mti->StopDataStream();
mti->SetDeviceParam( MTIParam::DataMode, MTIDataMode::Analog_Input_To_Output );

system(CLEARSCREEN);
printf("Device will read analog inputs and convert the signals to MEMS drive voltages\n");
printf("Press ANY KEY to set controller into Analog_Input_To_Output DataMode...\n");
_getch();
mti->StartDataStream ( );
```



```
printf("Press ANY KEY to return controller to Sample_Output DataMode...\n");
_getch();
mti->StopDataStream();
mti->SetDeviceParam( MTIParam::DataMode, MTIDataMode::Sample_Output );
mti->StartDataStream();
mti->ResetDevicePosition ();
mti->SetDeviceParams( &sParams ); //Restore original params back to controller
}
```

Return to Different DataMode

- Follow the same procedure that you used to switch modes
- StopDataStream before changing modes

```
void AnalogInputToOutputDemo ( MTIDevice* mti )  
{  
    MTIDeviceParams sParams;  
    mti->GetDeviceParams(&sParams); //back up current params from controller  
    //Set any MEMS Driver Settings for Analog Input to Output Mode  
    MTIDeviceParams tParams;  
    mti->GetDeviceParams(&tParams); //create temp params for controller  
    tParams.DataScale = 1.0;  
    tParams.Vbias = 70;  
    tParams.VdifferenceMax = 100;  
    tParams.HardwareFilterBw = 200;  
    tParams.SampleRate = 50000;  
    mti->SetDeviceParams( &tParams );  
  
    mti->ResetDevicePosition();  
    mti->StopDataStream();  
    mti->SetDeviceParam( MTIParam::DataMode, MTIDataMode::Analog_Input_To_Output );  
  
    system(CLEARSCREEN);  
    printf("Device will read analog inputs and convert the signals to MEMS drive voltages\n");  
    printf("Press ANY KEY to set controller into Analog_Input_To_Output DataMode...\n");  
    _getch();  
    mti->StartDataStream ( );  
  
    printf("Press ANY KEY to return controller to Sample_Output DataMode...\n");  
    _getch();  
    mti->StopDataStream();  
    mti->SetDeviceParam( MTIParam::DataMode, MTIDataMode::Sample_Output );  
    mti->StartDataStream();  
    mti->ResetDevicePosition ();  
    mti->SetDeviceParams( &sParams ); //Restore original params back to controller  
}
```

Thank You for Choosing



Additional Resources:

- [Mirracle MEMS Mirrors – Technical Overview](#)
- [Mirracle Documentation Portal](#)
- [Mirracle Web Page – Support](#)
- [Mirracle Web Page – Application Notes](#)
- [Mirracle Web Page – Publications](#)

If you have any further questions or suggestions please email us:

support@mirrorcletech.com