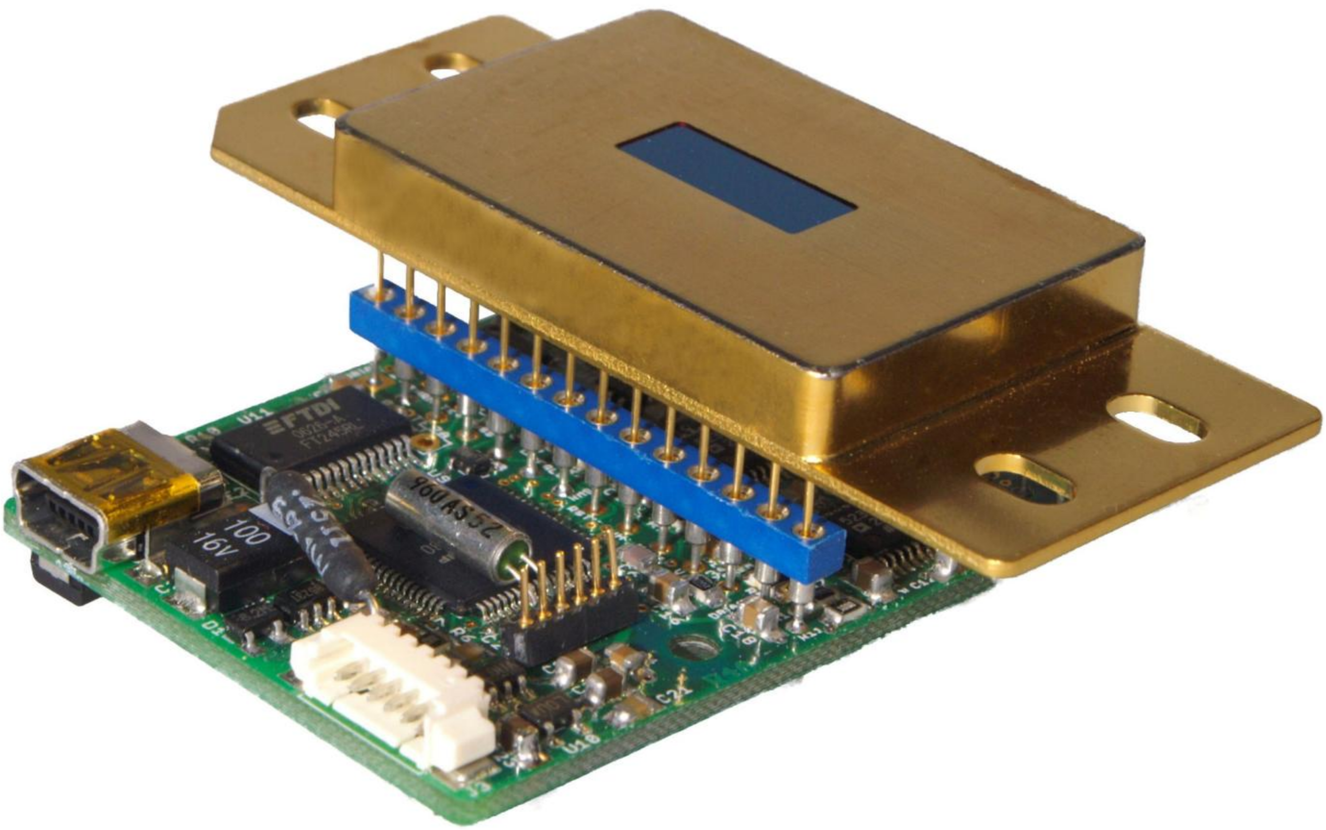
Cal Sensors, Inc.
5460 Skylane Blvd.
Santa Rosa, CA 95403

Phone: 707-545-4181
Fax: 707-545-5113
Web: www.calsensors.com

# 256 Element Array

## Programming Manual



# Table of Contents



# Table of Figures

Section
# 1

# Cal Sensors

## 256 Element Array Programming Manual

## 1. Introduction

The purpose of this manual is to provide our customers with the basic operating principles of the Cal Sensors 256 Element Multiplexed Array as interfaced through the Electronics Interface Board. With this information and the ADIC Library Documentation the user will be able to write code to operate the array in a variety of programming languages. The ADIC Library Documentation provides detailed information on each function, as well as information regarding specific language interfacing, error code definitions and documentation on working with multiple arrays on a single computer.

The 256 Element Array package can be interfaced to directly without the interface board, but requires a significant amount of digital logic and various voltages. The interface board simplifies the operation of the array by providing all necessary voltages, timing and thermoelectric cooler control functions. The array interface is then accomplished by sending commands over a USB interface using a standard Windows dynamic linked library (dll).

This manual will first discuss the operation of the array, then the operation of the interface board.

Section
2

# Cal Sensors

256 Element Array Programming Manual

## 2. Array – Principles of Operation

### Introduction

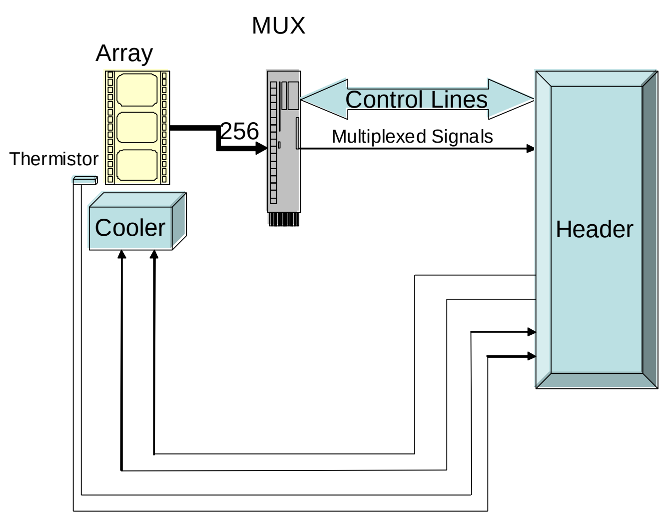
The array consists of the following items, all housed within an industry standard 28 pin package:

1. Thermoelectric cooler – used for thermal stabilization. **Note: Sufficient heatsinking must be provided to prevent damage to the array.**

2. Thermistor – used to provide feedback to control the temperature of the thermoelectric cooler.

3. Multiplexing Interface IC – used to condition the array output and multiplex all 256 channels onto a single output line.

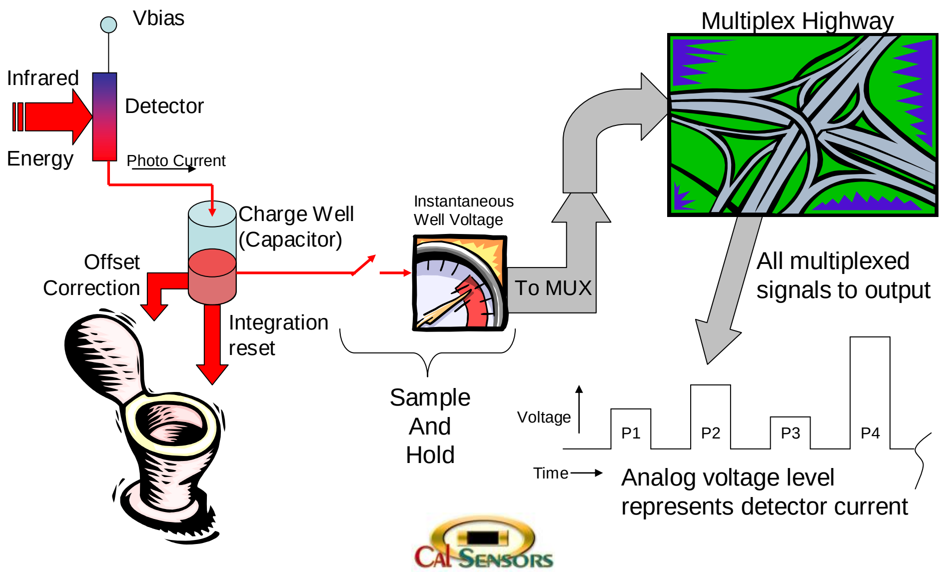
4. 256 Element Photoconductive Infrared Array – used to convert infrared photons into an electrical output.




• Figure 1 - Array Functional Block Diagram

## Operation

A simplified pictorial representation of the array operation is shown in Figure 2. This should be referred to throughout the following discussion.




• Figure 2 - Array Operation

### Summary

Each pixel has a bias voltage applied, generating a current which is collected in a charge well (capacitor). The offset current is subtracted from the incoming current, leaving only the signal current on the charge well. The signal current accumulates on the charge well for the integration period, at which time the voltage is sampled, the charge on the capacitor is removed and the accumulation cycle starts again. The value from all 256 sample-and-hold circuits are multiplexed onto a single line and sent out.

### Detailed Description

Each detector element has a bias voltage, Vbias, applied to it. The nature of photoconductive detectors is that they change resistance with incident photons. With an applied bias voltage the electrical current out of a photoconductive detector will consist of two parts. There will be an amount of current that exists when the element has no incident radiation (dark current) and a different amount of current when the element is exposed to the "signal" radiation. A higher bias voltage will produce more overall current. The "signal" current therefore can be determined by subtracting a known dark current from the total current coming from the detector element.

The difficulty with Lead Salt photoconductors (PbS and PbSe) is that the dark current is typically quite large compared to the signal current. In addition the bulk resistance of the film changes significantly with changing temperature. To stabilize the temperature the Cal Sensors array contains a thermoelectric cooler and a thermistor, which is strategically located. With feedback from the thermistor, the cooler current can be modulated to accurately control the temperature of the detector array.

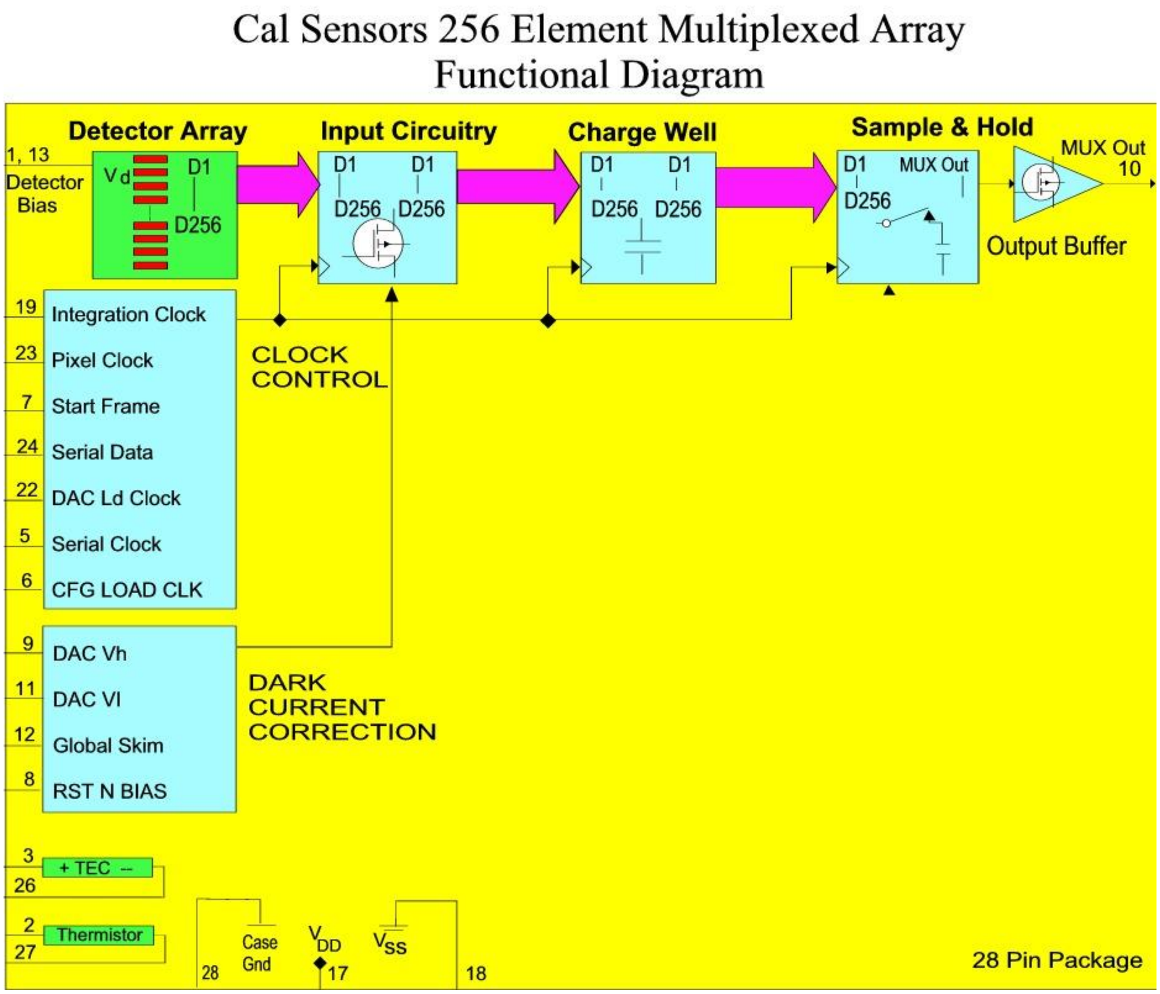
With the variability due to temperature removed the task of the multiplexer is to determine the dark current for each pixel and eliminate it so the signal current can be collected. This is performed through a calibration process where the array element is covered, removing all signal photons, and the output current is measured. This dark current can then be "skimmed" from the total current revealing the signal current. In this system there are actually two methods to remove the dark current. The first is Global Skim, where a fixed amount is skimmed from all 256 channels. The second is an individual correction for each pixel, typically required to accommodate the resistance variability of the photoconductive elements. This individual correction is always used, whereas Global Skim is often not used. Both global and individual dark current corrections are included as Offset Correction in Figure 2.

The individual pixel correction is implemented through a series of 256 digital to analog converters (DACs). All of the DACs operate between a high and low voltage (DAC Vh and DAC Vl). Each 8 bit DAC divides this voltage range into 256 steps and applies the appropriate correction to each pixel depending on the individual dark currents. The process of determining the DAC correction values for each pixel is referred to as Calibration. The correction values are dependent on other settings (integration time, bias voltage, etc.) so these values must be set and all signal radiation blocked prior to performing a calibration. In theory, if the temperature of the array and all other factors remain constant, a given calibration would never need to be repeated. In practice, because of changing temperature and other factors, the calibration process will need to be repeated at some interval. This needs to be performed more or less often depending on the stability or resolution required by a particular application.

The resulting signal current is accumulated on the charge well capacitor. The size of this capacitor can be set to 1pF, 4pF, 7pF, 10pF, 11pF, 14pF, 17pF or 20pF. Decreasing the charge well size effectively increases the gain of the system, in that the resultant voltage on the charge well after a certain integration time will be greater with a smaller capacitor. Since the total charge accumulated depends on bias voltage, detector resistance, dark current subtraction accuracy and integration time it is important to select a capacitor size that can accommodate the charge accumulation without "filling up", otherwise signal clipping will occur.

Once the integration time is completed the charge well is sampled and the resulting voltage held in a sample-and-hold circuit. This sampled voltage is proportional to the signal photons on the detector element. After the sampling is accomplished, the charge well is reset and the charge accumulation repeats. The resulting 256 voltage levels are multiplexed at a fixed clock rate onto the single output line.

Figure 3 shows the array pin assignments with the associated functional block diagram. Many of the pins are associated with clocking and control functions and require digital logic to implement. For systems that already contain digital logic or microcontroller functions this may be practical to provide. Many systems, however, are computer based and need to interface through standard ports. This functionality is provided by the Interface Electronics Board described in the next section.




• Figure 3 - Array Pin Functions

Section

# 3

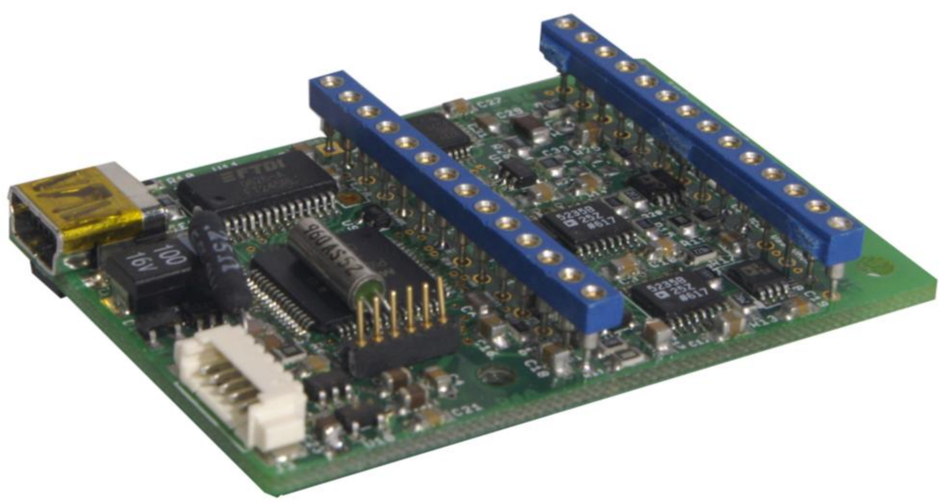
# Cal Sensors

256 Element Array Programming Manual

## 3. Interface Board – Principles of Operation

### General Description

The interface electronics board contains a microcontroller that supplies all timing and cooler control for the system along with USB interface circuitry and a current switch for powering the cooler. System control is provided by sending commands over the USB interface. The cooler is controlled by Pulse Width Modulating the supplied cooler current based on feedback from the thermistor. The set temperature is determined by a resistor located on the Interface Electronics Board. This set temperature is approximately -4° C.



## Features

- Standard USB interface using Windows dynamic linked library for interface functions.
- Small size (1.4” x 2.1”)
- 10 bit linear precision array voltage settings, providing all of the required array voltages from a single 12V input.
- 16bit output signal digitization at 500K samples/second
- Non-volatile storage of array settings and user data
- High efficiency thermoelectric control to fixed temperature
- Generation of all required timing and control signals

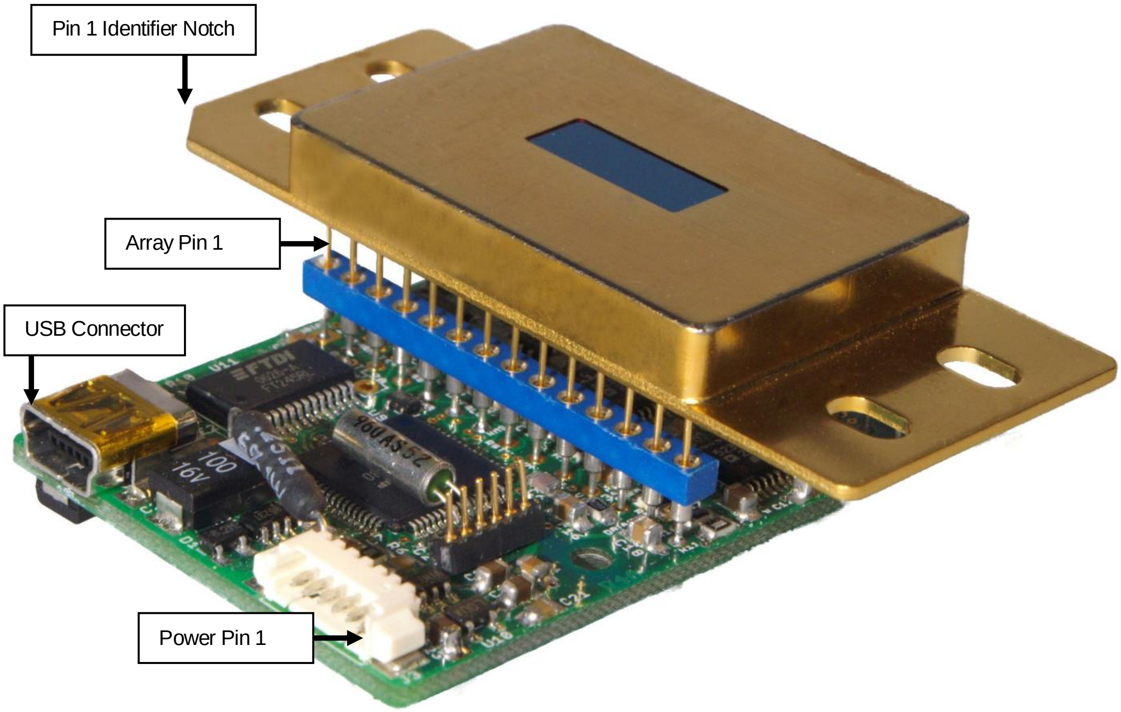
## Power connection

Power to the Interface Electronics Board is supplied through a six pin connector. Power supplied to this board also supplies power to the detector array and the thermoelectric cooler.

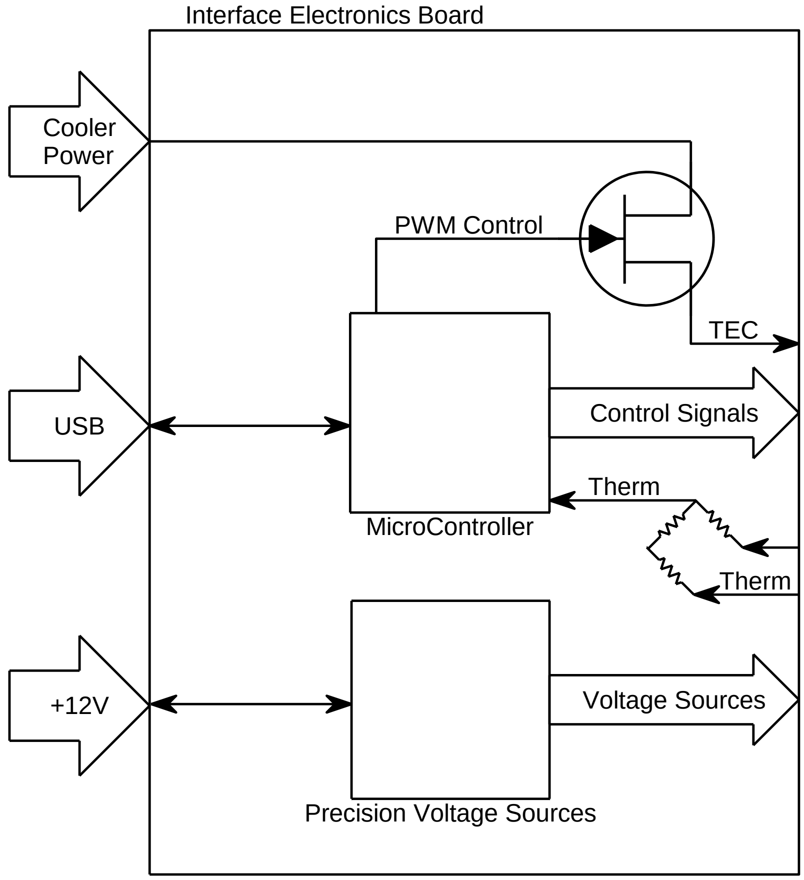
Connector Pin Assignment:

Pin 1: BLACK Wire - Ground Return for +12V power (100mA)
Pin 2: RED Wire - +12VDC @ 100mA System Power
Pin 3 & 4: BLUE Wire - Ground Return for TEC Power (approx 2A)
Pin 5 & 6: YELLOW Wire - +5VDC @ 2 Amps TEC Power

**Note that the Ground Returns for both the +12V power and the TE power (pins 1, 3 and 4) must be connected together at the system common ground point otherwise damage may result to the controller board.**




• Figure 4 - Interface Board Connectors




• Figure 5 - Interface Board Block Diagram

Section

4

# Cal Sensors

256 Element Array Programming Manual

## 4. Operation

### Overview

Because of the high level of functional integration contained in the Interface Electronics Board, the operation of the array is simply a matter of supplying mechanical connection (including heatsinking), electrical connection (power and USB) and then sending the appropriate commands from the host computer. Therefore, this section is primarily a description of how to implement the software commands. Since a standard Windows interface library is provided any standard programming language may be used. Cal Sensors provides an application written in National Instrument's LabView with the Development System. Because of the generic nature of the interface library no particular programming language will be targeted in this section. Specific language implementations are shown in the ADIC Library Documentation, and that document should be referred to for details on each of the library commands.

### Functions

| | |
|---|---|
| ArrayAdd | SaveE2Block |
| ArrayPercentDiff | SendAllCoeff |
| ArraySubtract | SetConversionRef |
| CloseDevice | SetConversionRef_Brd |
| DoCalibrate | SetDacGSkim |
| Flux_P | SetDacReferences |
| Flux_W | SetDacReferences_Brd |
| GetBoard_Handle | SetDacVH |
| GetData | SetDacVL |
| GetData2 | SetDetBias |
| GetDeviceHandle | SetGlobalSkimVal |
| GetNumBoards | SetGlobalSkimVal_Brd |
| GetQuery3 | SetIntegration |
| HideBadPixels | SetMuxSize |
| HideBadPixels_Brd | SetWellSize |
| MarkBadPixel | StoreAll |
| ReadE2Block | SuppressCalStat |
| ReadPWM | SuppressErrors |
| ReadTemp | SyncPC |
| RestoreAll | TECoolerPower |

Several of these functions simply perform calculations that are convenient for a particular application.

**Calculation Functions**

| | |
|---|---|
| ArrayAdd | Adds two single dimension arrays. |
| ArrayPercentDiff | Percent difference between two single dimension arrays. |
| ArraySubtract | Subtracts two single dimension arrays. |
| Flux_P | Integrates Plank's blackbody equation in Photons/(sec•cm$^2$•sr). |
| Flux_W | Integrates Plank's blackbody equation in Watts/(cm$^2$•sr). |

The remainder of the functions are involved with controlling the array. Many of these have default conditions that are usually left at their default value, so there is another subset of functions that are most commonly used for systems that use a single array connected to a computer.

**Common Functions**

| | |
|---|---|
| CloseDevice | Closes communication with the device. This should always be the last command sent. |
| DoCalibrate | Calibrates the array, setting the DAC coefficient array of correction values for each pixel. In the default condition Global Skim will not be used and DAC Vh and Vl will be determined. |
| GetData | Reads the digital data values for each pixel. |
| GetDeviceHandle | Returns an address to the device used in all other function calls. This should always be the first command sent. |
| MarkBadPixel | Sends the bad pixel map to the board. The board will not use values from bad pixels, instead returning the average of the two adjacent pixels with GetData. |
| RestoreAll | Restores settings from the non-volatile RAM on the board previously saved with the StoreAll command. |
| SetDetBias | Sets the bias level for the array. |
| SetIntegration | Sets the integration time for the array. |
| SetMuxSize | Sets the size of the Mux. This should always be set to 256. |
| SetWellSize | Sets the charge well to 1pF, 4pF, 7pF or 10pF. |
| StoreAll | Stores array settings to non-volatile RAM on the board. |
| TECoolerPower | Turns on or off the cooler. When on the temperature of the array will be held to the set point, (approximately -4°C), |

Before any functions are called up, there are several data structures that need to be established. These structures are passed to various functions and should be initialized before us in a function. Most functions are passed the Device Handle (obtained from GetDeviceHandle) and return a status variable which is 1 if the function completes successfully. Please see the appendix of ADIC Library Documentation for a list of error codes.

Also note that there are specific functions that deal with multiple attached boards. Please refer to the ADIC Library Documentation under the section Working With Multiple Controller Boards.

**Data Structures**

| | |
|---|---|
| coeffArray | The 256 element array of DAC correction coefficients. |
| bPixMapArray or bpMap | The 256 element array of bad pixels. Should be 1 at bad pixel locations, otherwise all 0. |
| MxData | The 256 element array of pixel values returned from GetData. |
| BaseSettings | A 16 element array containing array settings.<br>BaseSettings(0) = Integration Time<br>BaseSettings(1) = NOT USED, value will ALWAYS be 0<br>BaseSettings(2) = Global Skim value<br>BaseSettings(3) = NOT USED, value will ALWAYS be 0<br>BaseSettings(4) = DACVh value<br>BaseSettings(5) = NOT USED, value will ALWAYS be 0<br>BaseSettings(6) = DACVl value<br>BaseSettings(7) = NOT USED, value will ALWAYS be 0<br>BaseSettings(8) = NOT USED, value will ALWAYS be 0<br>BaseSettings(9) = Detector Bias<br>BaseSettings(10) = Mux Size<br>BaseSettings(11) = Direction Flag<br>BaseSettings(12) = Number of Bad Pixels<br>BaseSettings(13) = Integration Well Size<br>BaseSettings(14) = Left wndow address<br>BaseSettings(15) = Right window address |

## Sequence of Events

The standard sequence of events when interacting with the array is as follows:

1. Call GetDeviceHandle to get the device handle. An error will be returned if there are no devices attached, they are not powered up or there is some other communication problem. **Note: The USB drivers must be installed on any system using the array before any communication is possible.**
2. Initialize and place all zeros in the bad pixel array.
3. Set the mux size to 256 using SetMuxSize.
4. Set a well size using SetWellSize. (10pF is a good start.)
5. Set the integration time using SetIntegration. (500µS is a good start.)
6. Set the detector bias using SetDetBias. (7V is a good start.)
7. Call TECoolerPower with a teStatus of 1 if required to turn the cooler on.
    a. Note: Ensure that the array has sufficient heatsinking prior to issuing this command.
    b. Wait until the temperature has stabilized before calibrating.
8. Call DoCalibrate making sure the parameters passed have all been initialized.
    a. Remember to cover the array during calibration.
    b. A value of 1.0 is a good starting point for arraySetVal.
9. Call GetData as necessary making sure the parameters passed have all been initialized.
10. If the cooler has been turned on call TECoolerPower with a teStatus of 0 to turn the cooler off.
11. At the end of communications call CloseDevice.

If the array status was previously stored with the StoreAll command, steps 3 – 6 may be skipped by calling RestoreAll.

DoCalibrate

Since calibration is one of the most important functions, it is important to understand the process being performed by the interface board. The DoCalibrate function looks at the current being generated by each pixel and attempts to calculate values for all DACs that will make all outputs equal to a target output voltage. The actual target correction value is set by arraySetVal, which is passed to DoCalibrate. Intuitively the correction target should be as low as possible to provide the greatest range of output values. Since the multiplexer cannot operate at the lower voltage rail there must be some minimum voltage that allows the full range of DAC correction values. Typically an array can be calibrated with target values as low as 0.50V or 0.75V.

If pixels are marked as bad in the bPixMapArray sent to DoCalibrate, their values are not used to calculate the DAC values. If there is a bad pixel that is not marked typically the DAC Vh or DAC Vl will be at the rail and the bad pixel will be either very high or low. The correction for this is to mark the pixel bad and perform another calibration.

Changes in array parameters such as bias or well size will typically require a re-calibration. Changes in array operating temperature will also require a re-calibration.

## Conclusion

With the Interface Electronics Board, operating the Cal Sensors 256 Element Array becomes simply a process of issuing commands through a standard USB port. This manual has provided the hardware and software details necessary to operate the array within this environment. Please refer to the DLL library description in the ADIC Library Documentation for detailed library information.