

MBARI Zeiss MMS Spectrometer Controller API

Zeiss MMS Detector

The free running pixel clock P1 (PSoC pin 62) should be set at 62.5kHz - 16us. The following signals should also be free running with fixed timing relative to P1: RESET (PSoC pin 65), CONVERT (PSoC pin 2), and SERCLK (PSoC pin 3). The START signal (PSoC pin 64) has fixed timing relative to P1 too but its period is programmable. The START signal is only asserted when the detector is ready to be read – the time interval between start pulses is the integration period.

Physical Connection

Power and communications are provided via a DB-9F connector. The controller connector and pin mapping conform to an RS-232 DCE (Data Communications Equipment) device. Connection between the MMS controller and a remote PC (Data Terminal Equipment) is detailed below.

MMS (DCE)	DB-9F		PC (DTE)	DB-9M
DCE_TXD	2	->	DTE_RXD	2
DCE_RXD	3	<-	DCE_TXD	3
DCE_CTS	7	<-	DTE_RTS	7
DCE_RTS	8	->	DTE_CTS	8
Vin (5V)	9		NC	-
Ground	5		NC	-

Initialization

When power is first applied the device will immediately set RTS (PSoC pin 60 low) as part of power on initialization and will clear RTS (PSoC pin 60 high) upon entering the command processor. The computer/controller connected will monitor RTS to validate power up.

Command interface

Control of the spectrometer is done using simple one letter commands, with some commands (A, I, P, R) including additional parameters. All commands are sent using ASCII characters. All commands must be terminated with a carriage return (CR). The device only acts on the command after receiving the CR. Sending the ZZZ command resets the PSoC controller.

List of commands

- A Set Baud Rate
- B Read back
- G Get spectrum data
- H Get Temp/Humidity sensor data
- I Set Integration period
- P Set number of Prescans
- R Set number of Readouts
- S Start a measurement cycle
- V Get firmware version info

A Command

This is used to set the baud rate of the device. The letter "A" is sent followed by a single digit number with optional exclamation point character (see write and store examples below). If the Baud rate has been changed then it is necessary to change the baud of the computer/controller connected to match the new baud rate. The default baud rate is read from eeprom during device initialization. The default baud rate is typically 115200.

A1 Set baud rate to 9600.

A2 Set baud rate to 19200.

A3 Set baud rate to 38400.

A4 Set baud rate to 57600

A5 Set baud rate to 115200.

Format: An[!]

Default: read from [PSoC eeprom](#) during device initialization

Range: [1, 5]

Read: A?

Write: A5 (sets the current baud rate)

Store: A5! (sets the current baud rate and stores the default baud rate in eeprom)

B Command

This is used to check whether a device is connected to the serial interface. Whatever is sent to the device AFTER the B is echoed back to the sender.

Example:

BANYONETHERE? (returns ANYONETHERE?)

G Command

This tells the device to send one scan of measurement data using both RTS (PSoC pin 60) and CTS (PSoC pin 61) to synchronize transmission with the host PC/controller (see pseudo code below). If multiple scans are required, the G command must be repeated for every scan required. If the device was set to make 5 readouts (see R command) then the G command would need to be sent 5 times to retrieve all the scans. The device's detector has 256 pixels measured with a 16bit A/D, which means 512 data bytes are returned to the sender. Data bytes are sent high byte first as binary values.

Use G? to read the number of data scans available in the data FIFO. The number of data scans pending in the FIFO is set during execution of the S Command.

H Command

This tells the device to initiate a Humidity and Temperature Data Fetch from the onboard Honeywell HIH6130 Humidity/Temperature Sensor - [see Figure 4. Humidity and Temperature Data Fetch, Four Byte Data Read](#). Data will be sent in hexadecimal using the format provided below.

Returned Data Format: StatusUchar, HDataUshort, TDataUshort

```
Status = (uchar)( (Hum_h >> 6) & 0x03 );  
HData = (ushort)( (((ushort)(Hum_h) << 8) | Hum_l ));  
TData = (ushort)( (((ushort)Temp_h) << 8) | Temp_l >> 2 );
```

I Command

The I command is used to set the Integration period for the measurement scan(s). The I is followed by a number which represents the integration period in milliseconds.

Format: Innnn[!]

Default: read from [PSoC eeprom](#) during device initialization

Range: [20, 4000] milliseconds

Read: I?

Write: I765 (updates the integration period value in sram)

Store: I765! (stores the integration period value in onboard eeprom and updates value in sram)

P Command

The P command sets the number of prescans in each measurement. The prescans are used to clear the device's detector of residual charge prior to collecting measurement scans (see R and S commands). The period for each prescan will be approx. 4ms (16us pixel clock * 256 pixels).

Format: Pn[!]

Default: read from [PSoC eeprom](#) during device initialization

Range: [1, 6] prescans (each 4 milliseconds long)

Read: P?

Write: P3 (updates the prescan value in sram)

Store: P3! (stores the prescan value in onboard eeprom and updates value in sram)

R Command

The R command sets the number of scans in each measurement. By making multiple scans and then making an average of these scans significant improvement in the S/N can be achieved. The R is followed by a number which determines the number of readouts.

Format: Rn[!]

Default: read from [PSoC eeprom](#) during device initialization

Range: [1, 10] data scans (each n milliseconds long – see I command)

Read: R?

Write: R5 (updates the readout value in sram)

Store: R5! (stores the readout value in onboard eeprom and updates value in sram)

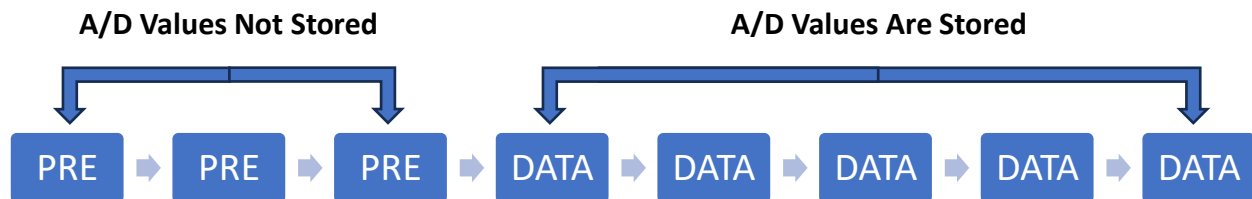
S Command

After the device has been set up using the I, P, and R commands the device can be triggered to make a measurement using the S command. Just sending S followed by CR will cause the device to make a measurement using RTS (PSoC pin 60) to synchronize data acquisition with the host PC/controller (see measurement cycle diagram and pseudo code below).

Example: P = 3 and R = 5 Measurement

Integration Period = 4 ms for **PRE** scans

Integration Period = I Command Value for **DATA** scans



The time interval between START pulses (PSoC pin 64) defines the detector integration period - see [Hamamatsu S3904](#) datasheet and attached Current PSoC Detector Timing.

V Command

Output Device's firmware version and build info (see pseudo code below).

ZZZ Command

Reset the PSoC controller.

Errors

If a command is sent that is not understood, the reply from the device will be E:x.
Where x is the command that was not understood.

NOTE

DESIGN FILES AND FURTHER DOCUMENTATION IS AVAILABLE ON [GOOGLE DRIVE](#)

Luke Coletti
(831) 775-1831
coletti@mbari.org
rev. December 5, 2023

PSoC pseudo code for the 'S' command:

```
output_bit(RTS,0);    //set RTS, signal controller we're scanning
output_bit(LED_GRN, 0); //GRN LED ON

InitializeDigitalLogic()

TakePreAndDataScans()    //set PSoC to take P prescans followed R data scans

MonitorScanProgress      //wait for detector to finish last scan

UpdateDataFIFOscansAvailable

output_bit(RTS,1); //clear RTS, signal the controller we're done scanning
output_bit(LED_GRN, 1); //GRN LED OFF
```

PSoC pseudo code for the 'G' command:

```
output_bit(RTS,0);    //set RTS, signal controller it's time to start burping the data out
output_bit(LED_RED, 0); //RED LED ON

    for(x=0; x < 512; x++){

        while (CTS_Read()); //wait for CTS input to go low

        UART_PutChar (spectra[x]);
        toggle_bit(LED_RED); //RED LED TOGGLE

        while (!CTS_Read()); //wait for CTS input to go high

    }

output_bit(RTS,1);    //clear RTS, signal controller we're done burping the data out
output_bit(LED_RED, 1); //RED LED OFF
```

PSoC pseudo code for the 'V' command:

```
const char Date[15] = __DATE__;
const char Time[15] = __TIME__;
const char Revision[10] = "1.1";

printf("MBARI-ISUS %s, %s, %s\n\r", Revision, Date, Time);
```

Current PSoC Detector Timing

