TECHNICAL GUIDE

Contents

1 Working with HDF5 Files

HDF5 is the preferred format for storing APDM movement monitor data. It is a standard format for scientific data that is efficient and widely supported. It uses less space than CSV, is faster to load, and supports more structured data. This section will cover the organization of the APDM movement monitor data and the basics of reading HDF5 files in Python.

1.1 HDFView

A free program called HDFView (http://www.hdfgroup.org/hdf-java-html/hdfview/) can be used to explore, plot, and export this data into other formats. Many free and commercial products are capable of reading, writing, and editing HDF data, including Matlab, Python, and LabView. Many programming languages are also supported, including, C, C++, Java, and Python.

1.2 Python

APDM embraces Python as a Mathematical scripting language. It has excellent support for reading, writing, editin HDF files through the h5py package. It is also widely used and open source, which makes it available to our entire customer base.

1.3 Data Organization

HDF5 files are organized like a file structure. The root of the file contains one or more attributes. One is the version number for the organization of the HDF 5 file. APDM is currently on version 5 of our HDF format.

1.4 File Structure

Note about the latest version: Version 5 of APDM's HDF format departs significantly from Version 4. It is specific to our second iteration of hardware, so if you are using v1 hardware (without the displays), your recordings will follow the Version 4 HDF convention. The information contained is largely identical, but it is organized differently to make it more extensible. If you have created custom parsers for your recordings or have used APDM's examples for post-processing your recordings, you will likely have to change these scripts to adapt. You can use the version attribute to modify your scripts to handle both versions from the same script.

1.4.1 Version 5

- **Annotations** Table containing annotations and button events
  - **Time** Annotation time in epoch microseconds
  - **Case ID** A movement monitor case ID associated with the annotation
  - **Annotation** The annotation string
- **Sensors** Group containing sensor data
  - **XI-XXXXXX** Group containing data from the monitor with this case ID. There is one of these groups for each sensor
    - **Accelerometer** Dataset containing accelerometer data
      - **Lower Limit** Attribute specifying the lower limit of the sensor
      - **Upper Limit** Attribute specifying the upper limit of the sensor
      - **Name** Attribute containing the name of the sensor
      - **Units** Attribute containing the units used to report the sensor measurements
    - **Barometer** Dataset containing barometer data
      - **Lower Limit** Attribute specifying the lower limit of the sensor
      - **Upper Limit** Attribute specifying the upper limit of the sensor
      - **Name** Attribute containing the name of the sensor
      - **Units** Attribute containing the units used to report the sensor measurements
    - **Gyroscope** Dataset containing gyroscope data
      - **Lower Limit** Attribute specifying the lower limit of the sensor
      - **Upper Limit** Attribute specifying the upper limit of the sensor
      - **Name** Attribute containing the name of the sensor
      - **Units** Attribute containing the units used to report the sensor measurements
    - **Magnetometer** Dataset containing magnetometer data
      - **Lower Limit** Attribute specifying the lower limit of the sensor
      - **Upper Limit** Attribute specifying the upper limit of the sensor
      - **Name** Attribute containing the name of the sensor
      - **Units** Attribute containing the units used to report the sensor measurements
    - **Temperature** Dataset containing internal temperature data
      - **Lower Limit** Attribute specifying the lower limit of the sensor
      - **Upper Limit** Attribute specifying the upper limit of the sensor
      - **Name** Attribute containing the name of the sensor
      - **Units** Attribute containing the units used to report the sensor measurements
    - **Time** Dataset containing timestamps in units of microseconds since 0:00 Jan 1, 1970 UTC
      - **Units** Attribute specifying the units used to store the timestamps
    - **Configuration** Group containing attributes that specify the configuration of the sensor
      - **Accelerometer Enabled** 1 for enabled. 0 for disabled
      - **Gyroscope Enabled** 1 for enabled. 0 for disabled
      - **Magnetometer Enabled** 1 for enabled. 0 for disabled
      - **Barometer Enabled** 1 for enabled. 0 for disabled
      - **Button Event 0 =** String configured to represent a button 0 event
      - **Button Event 1 =** String configured to represent a button 1 event
      - **Button Event 2 =** String configured to represent a button 2 event

        - **Button Event 3 =** String configured to represent a button 3 event
        - **Calibration Disabled** 1 to apply only the sensor data sheet typical conversion factors. 0 to use calibration data stored on the device
        - **Label 0** Custom label for this sensor. This is what is displayed on the LCD
        - **Location** Sensor location on the body
        - **Sample Rate** Sensor output data rate
        - **Timezone** Timezone code
        - **Timezone Offset** Timezone offset in units of hours from UTC
        - **Wireless Channel** Wireless channel
        - **Wireless Protocol** 0 for wireless disabled, 1 for synchronized logging, 2 for wireless streaming
        - **Wireless Latency (ms)** Configured maximum acceptable wireless latency
        - **Wireless rate divider** Data is streamed at the sample rate divided by this factor
      - **Metrics** Group containing information on the status of the sensor
- **Processed** Group containing derived signals from the sensor data
  - **XI-XXXXXX** Group containing derived signals from the sensor with this case ID
    - Orientation Dataset containing quaternions which can be used to rotate sensor data from the sensor frame to a local NWU reference frame

1.4.2 Versions 3 and 4

- **MonitorLabelList** Attribute containing an array of monitor labels in the same order as the CaseIdList
- **CaseIdList Attribute** containing an array of monitor case IDs in the same order as the MonitorLabelList
- **FileFormatVersion** Attribute containing the file format version (3)
- **Annotations** Table containing annotations
  - **Time** Annotation time in epoch microseconds
  - **Case ID** A movement monitor case ID associated with the annotation
  - **Annotation** The annotation string
- **AA-XXXXXX** A group is included in the file for each monitor in the CaseIdList, with the name equal to the case ID
  - **FilteredDataPopulated (version 4)** Attribute indicating the present of the filtered data group
  - **SampleRate** Attribute containing the output data rate for the monitor
  - **DecimationFactor** Decimation factor for the monitor's internal processing
  - **ModuleID** The module ID for the monitor
  - **TimeGood** Flag indicating whether the time has been set on the monitor since it powered on
  - **RecordingMode** One of: "Wireless streaming", "Synchronized logging", or "Unsynchronized logging"
  - **DataMode** Indicates whether the data was retrieved wirelessly or copied from the monitor's internal storage while docked. One of: "Streamed wirelessly" or "Logged to monitor"
  - **AccelerometersEnabled** 1 for enabled, 0 for disabled
  - **GyroscopesEnabled** 1 for enabled, 0 for disabled
  - **MagnetometersEnabled** 1 for enabled, 0 for disabled
  - **DecimationBypass** Internal use, deprecated
  - **CalibrationVersion** Version of the calibration data used to convert from raw samples to calibrated SI units
  - **VersionString1** Firmware version string 1
  - **VersionString2** Firmware version string 2

- **VersionString3** Firmware version string 3
- **CalibratedDataPopulated (version 3)** 1 for populated, 0 for unpopulated
- **CalibratedData (version 4)** Calibration data for the monitor
- **LocalTimeOffset** Time in microseconds to add to UTC to convert to local time
- **SyncValue** Dataset containing the internal sync value for each sample
  - **Units** Attribute string containing the timestamp units (1/2560th of a second since 0:00 Jan 1, 1970 UTC)
- **Time** Dataset containing a timestamp for each sample
  - **Units** Attribute string containing the units (microseconds since 0:00 Jan 1, 1970 UTC)
- **ButtonStatus** Dataset containing the button status for each sample (1==pressed, 0==unpressed)
- **Calibrated** Group containing calibrated data
  - **Accelerometers** Dataset containing accelerometer data (N$\times$3)
    - **Units** Attribute string containing the accelerometer units (m/s^2)
    - **Range** Attribute containing the range setting for the accelerometer (2g or 6g)
    - **Gravity (version 4)** Attribute indicating the gravity constant used in orientation estimation
  - **Gyroscopes** Dataset containing gyroscope data (N$\times$3)
    - **Units** Attribute string containing the gyroscope units (rad/s)
  - **Magnetometers** Dataset containing magnetometer data (N$\times$3)
    - **Units** Attribute string containing the magnetometer units (μT)
  - **Temperature** Dataset containing the temperature (N$\times$1)
    - **Units** Attribute string containing the temperature units ($^\circ$C)
    - **EarthFieldMagnitude (version 4)** The field constant used in orientation estimation
  - **TemperatureDerivative** Dataset containing the temperature derivative (N$\times$1)
    - **Units** Attribute string containing the temperature derivative units ($^\circ$C/s)
  - **Orientation** Dataset containing the orientation quaternion (Nx4). The orientation is relative to a (magnetic) north, west, up reference frame. The scalar component of the quaternion is the first element.
- **Raw** Group containing raw data if selected during import
  - **Accelerometers**
  - **Gyroscopes**
  - **Magnetometers**
  - **DataFlags**
  - **OptData**
  - **Temperature**
  - **TemperatureDerivative**
- **Filtered (version 4)** Filtered data set. This set is intended for post-processed data. Currently the gyro biases are removed from the gyroscope signals. In the future, additional filtering may be implemented.
  - **Accelerometers**
  - **Gyroscopes**
  - **Magnetometers**

1.4.3 Version 3

- **MonitorLabelList** Attribute containing an array of monitor labels in the same order as the CaseIdList
- **CaseIdList Attribute** containing an array of monitor case IDs in the same order as the MonitorLabelList

- **FileFormatVersion** Attribute containing the file format version (3)
- **Annotations** Table containing annotations
  - **Time** Annotation time in epoch microseconds
  - **Case ID** A movement monitor case ID associated with the annotation
  - **Annotation** The annotation string
- **AA-XXXXXX** A group is included in the file for each monitor in the CaseIdList, with the name equal to the case ID
  - **SampleRate** Attribute containing the output data rate for the monitor
  - **DecimationFactor** Decimation factor for the monitor's internal processing
  - **ModuleID** The module ID for the monitor
  - **TimeGood** Flag indicating whether the time has been set on the monitor since it powered on
  - **RecordingMode** One of: "Wireless streaming", "Synchronized logging", or "Unsynchronized logging"
  - **DataMode** Indicates whether the data was retrieved wirelessly or copied from the monitor's internal storage while docked. One of: "Streamed wirelessly" or "Logged to monitor"
  - **AccelerometersEnabled** 1 for enabled, 0 for disabled
  - **GyroscopesEnabled** 1 for enabled, 0 for disabled
  - **MagnetometersEnabled** 1 for enabled, 0 for disabled
  - **DecimationBypass** Internal use, deprecated
  - **CalibrationVersion** Version of the calibration data used to convert from raw samples to calibrated SI units
  - **VersionString1** Firmware version string 1
  - **VersionString2** Firmware version string 2
  - **VersionString3** Firmware version string 3
  - **CalibratedDataPopulated** 1 for populated, 0 for unpopulated
  - **LocalTimeOffset** Time in milliseconds to add to UTC to convert to local time
  - **SyncValue** Dataset containing the internal sync value for each sample
    - **Units** Attribute string containing the timestamp units (1/2560th of a second since 0:00 Jan 1, 1970 UTC)
  - **Time** Dataset containing a timestamp for each sample
    - **Units** Attribute string containing the units (microseconds since 0:00 Jan 1, 1970 UTC)
  - **ButtonStatus** Dataset containing the button status for each sample (1==pressed, 0==unpressed)
  - **Calibrated** Group containing calibrated data
    - **Accelerometers** Dataset containing accelerometer data (N$\times$3)
      - **Units** Attribute string containing the accelerometer units (m/s^2)
      - **Range** Attribute containing the range setting for the accelerometer (2g or 6g)
    - **Gyroscopes** Dataset containing gyroscope data (N$\times$3)
      - **Units** Attribute string containing the gyroscope units (rad/s)
    - **Magnetometers** Dataset containing magnetometer data (N$\times$3)
      - **Units** Attribute string containing the magnetometer units (μT)
    - **Temperature** Dataset containing the temperature (N$\times$1)
      - **Units** Attribute string containing the temperature units (°C)
    - **TemperatureDerivative** Dataset containing the temperature derivative (N$\times$1)
      - **Units** Attribute string containing the temperature derivative units (°C/s)
    - **Orientation** Dataset containing the orientation quaternion (N$\times$4). The orientation is relative to a (magnetic) north, west, up reference frame. The scalar component of the quaternion is the first element.
  - **Raw** Group containing raw data if selected during import

        * **Accelerometers**
        * **Gyroscopes**
        * **Magnetometers**
        * **DataFlags**
        * **OptData**
        * **Temperature**
        * **TemperatureDerivative**

1.4.4 Version 2

- **MonitorLabelList** Attribute containing an array of monitor labels in the same order as the CaseIdList
- **CaseIdList Attribute** containing an array of monitor case IDs in the same order as the MonitorLabelList
- **FileFormatVersion** Attribute containing the file format version (2)
- **Annotations** Table containing annotations
    - **Time** Annotation time in epoch microseconds
    - **Case ID** A movement monitor case ID associated with the annotation
    - **Annotation** The annotation string
- **AA-XXXXXX** A group is included in the file for each monitor in the CaseIdList, with the name equal to the case ID
    - **SampleRate** Attribute containing the output data rate for the monitor
    - **DecimationFactor** Decimation factor for the monitor's internal processing
    - **ModuleID** The module ID for the monitor
    - **TimeGood** Flag indicating whether the time has been set on the monitor since it powered on
    - **RecordingMode** One of: "Wireless streaming", "Synchronized logging", or "Unsynchronized logging"
    - **DataMode** Indicates whether the data was retrieved wirelessly or copied from the monitor's internal storage while docked. One of: "Streamed wirelessly" or "Logged to monitor"
    - **AccelerometersEnabled** 1 for enabled, 0 for disabled
    - **GyroscopesEnabled** 1 for enabled, 0 for disabled
    - **MagnetometersEnabled** 1 for enabled, 0 for disabled
    - **DecimationBypass** Internal use, deprecated
    - **CalibrationVersion** Version of the calibration data used to convert from raw samples to calibrated SI units
    - **VersionString1** Firmware version string 1
    - **VersionString2** Firmware version string 2
    - **VersionString3** Firmware version string 3
    - **CalibratedDataPopulated** 1 for populated, 0 for unpopulated
    - **LocalTimeOffset** Time in milliseconds to add to UTC to convert to local time
    - **SyncValue** Dataset containing the internal sync value for each sample
        * **Units** Attribute string containing the timestamp units (1/2560th of a second since 0:00 Jan 1, 1970 UTC)
    - **Time** Dataset containing a timestamp for each sample
        * **Units** Attribute string containing the units (microseconds since 0:00 Jan 1, 1970 UTC)
    - **Calibrated** Group containing calibrated data
        * **Accelerometers** Dataset containing accelerometer data (N$\times$3)
            · **Units** Attribute string containing the accelerometer units (m/s^2)
            · **Range** Attribute containing the range setting for the accelerometer (2g or 6g)

    * **Gyroscopes** Dataset containing gyroscope data (N$x3$)
        - **Units** Attribute string containing the gyroscope units (rad/s)
    * **Magnetometers** Dataset containing magnetometer data (N$x3$)
        - **Units** Attribute string containing the magnetometer units (μT)
    * **Temperature** Dataset containing the temperature (N$x1$)
        - **Units** Attribute string containing the temperature units (°C)
    * **TemperatureDerivative** Dataset containing the temperature derivative (N$x1$)
        - **Units** Attribute string containing the temperature derivative units (°C/s)
  - **Raw** Group containing raw data if selected during import
    * **Accelerometers**
    * **Gyroscopes**
    * **Magnetometers**
    * **DataFlags**
    * **OptData**
    * **Temperature**
    * **TemperatureDerivative**

1.4.5 Version 1

This version is deprecated. All new files created will use the most recent version.

- **Device_List** Attribute containing a list of monitors present in the file
- **File_Format_Version** Attribute containing the file version
- **Annotations** Table containing annotations
  - **Time** Annotation time in epoch microseconds
  - **Device ID** A movement monitor ID associated with the annotation
  - **Annotation** The annotation string
- **Opal_xxx/** Group containing information about and data from monitor ID xxx
  - **Sample_Rate** Attribute containing the output data rate for the monitor
  - **Decimation_Factor** Decimation factor for the monitor's internal processing
  - **Time_Good** Flag indicating whether the monitor has had its time set since turning on
  - **Decimation_Bypass** Internal use, deprecated
  - **Calibration_Version** Version of the calibration data used to convert from raw samples to calibrated SI units
  - **Version_String1** Firmware version string 1
  - **Version_String2** Firmware version string 2
  - **Version_String3** Firmware version string 3
  - **Acceleration** Dataset containing data from the accelerometers (N$x3$)
    * **Units** Attribute string containing the acceleration units (m/s^2)
  - **Angular_Velocity** Dataset containing data from the gyroscopes (N$x3$)
    * **Units** Attribute string containing the angular velocity units (rad/s)
  - **Magnetic_Field** Dataset containing data from the magnetometers (N$x3$)
    * **Units** Attribute string containing the magnetic field units (a.u.)

  - **Temperature** Dataset containing the temperature of the monitor (Nx1)
    * **Units** Attribute string containing the temperature units (°C)
  - **Temperature_Derivative** Dataset containing the rate of change of temperature
    * **Units** Attribute string containing the temperature derivative units (°C/s)
  - **Sync_Value** Dataset containing the internal timestamp of each sample
    * **Units** Attribute string containing the timestamp units (1/2560th of a second since 0:00 Jan 1, 1970 UTC)
    * **Time** Dataset containing the time for each sample in microseconds since 0:00 Jan 1, 1970 UTC

Additional fields present when raw data is also stored:

- **Opal_XX/**
  - **Calibration_Data** Attribute containing binary block of calibration data
  - **Raw_File_Version** Attribute containing the version string of the raw file (if this was converted from a .apdm file instead of streamed)
  - **Accelerometers_Raw** Dataset containing raw accelerometer data ($Nx3$)
  - **Gyroscopes_Raw** Dataset containing raw gyroscope data ($Nx3$)
  - **Magnetometers_Raw** Dataset containing raw magnetometer data ($Nx3$)
  - **Data_Flags** Dataset containing flags used for processing the raw data
  - **Opt_Data** Dataset containing several measurements taken at a low data rate
  - **Temperature_Raw** Dataset containing lowpass filtered, but uncalibrated temperature data ($Nx1$)

1.5 Working with HDF 5 in MATLAB

MATLAB contains native functionality for working with HDF5 files. Additional help and examples are included in the built in help documentation for these functions. The first example below uses an older Matlab interface (hdf5read,...) which is compatible with Matlab 2008b and later. It also assumes version 4 of APDM's HDF format. The second example uses Matlab's updated interfacd (h5read, ...) and assumes version 5 of APDM's HDF format.

1.5.1 Matlab, hdf5read, Version 4 of APDM's HDF format

```
filename = 'example.h5';
try
   vers = hdf5read(filename, '/FileFormatVersion');
catch
   try
      vers = hdf5read(filename, '/File_Format_Version');
   catch
      error('Couldn''t determine file format');
   end
end
if vers< 2
```

```
   error('This example only works with version 2 or later of the data file')
end
caseIdList = hdf5read(filename, '/CaseIdList');
groupName = caseIdList(1).data;
accPath = [groupName '/Calibrated/Accelerometers'];
fs = hdf5read(filename, [groupName '/SampleRate']);
fs = double(fs);
acc = hdf5read(filename, accPath)'; %Transposed to make Nx3 in MATLAB}
t = (1:size(acc,1))/fs;
figure;
plot(t,acc);
```

A more complicated example using the flexibility of HDF5 to load and process only part of a data set. This can be useful when the data set is too large to fit into memory. Care is taken not to attempt to read beyond the end of the file.

```
filename = 'example.h5';
try
    vers = hdf5read(filename, '/FileFormatVersion');
catch
    try
        vers = hdf5read(filename, '/File_Format_Version');
    catch
        error('Couldn''t determine file format');
    end
end
if vers < 2
    error('This example only works with version 2 or later of the data file')
end
idList = hdf5read(filename, '/CaseIdList');
groupName = idList(1).data;
accPath = [groupName '/Calibrated/Accelerometers'];
fs = hdf5read(filename, [groupName '/SampleRate']);
fs = double(fs);
fhandle = H5F.open(filename, 'H5F_ACC_RDONLY', 'H5P_DEFAULT');
dset = H5D.open(fhandle, [groupName '/Calibrated/Accelerometers'], 'H5P_DEFAULT');
dspace = H5D.get_space(dset);
[ndims, dims] = H5S.get_simple_extent_dims(dspace);
nSamples = dims(1);
nSamplesRead = min(nSamples, 60*fs); %read at most one minute of data
accSegment = hdf5readslab(filename, accPath, [0,0], [nSamplesRead, 3])';
t = (1:nSamplesRead)/fs;
figure;
plot(t,accSegment);
```

1.5.2 Matlab, h5read, Version 5 of APDM's HDF format

Plot the accelerometer dataset for the sensor labeled "Lumbar".

```
filePath = '/my_files/20170329-090555_TUG.h5'

fileFormat = h5readatt(filePath, '/', 'FileFormatVersion');

if fileFormat < 5
    error('TruncateHDF only works on fileFormat versions 5+');
end

sensors = h5info(filePath, '/Sensors')

for i = 1:length(sensors.Groups)
    label = h5readatt(filePath, [sensor(i).Name '/Configuration'], 'Label 0');
    if strfind(label, 'Lumbar')
        acc = h5read(filePath, [sensor(i).Name '/Accelerometer'])
        figure;plot(acc')
    end
end
```

1.5.3 Notes

- Arrays in MATLAB use the FORTRAN convention of storing them in memory by column then row, instead of the C convention (used by HDF 5) of row then column. This has the effect of making the returned arrays transposed from how this document (and many other interfaces to HDF5) claim they are laid out.
- Older versions of MATLAB (before 2009a) did not support the compression used in Motion Studio's HDF 5 files. If you are using one of these older versions, the free h5repack utility available from the HDF Group can remove the compression. This utility is available at:
  http://www.hdfgroup.org/HDF5/release/obtain5.html
  The command to repack the file is:
  `h5repack -f NONE example.h5 example_no_compression.h5`

1.6 Working with HDF 5 in Python

Swap labels on a recording. For example, swap the ”Right Arm” with the ”Right Foot” Opal.

```
import h5py
def swapLabels(fn, oldLabel='Right', newLabel='Left'):
    with h5py.File(fn, 'r+') as f:
        for s in f['Sensors']:
            l = f['Sensors'][s]['Configuration'].attrs['Label 0']
            f['Sensors'][s]['Configuration'].attrs.modify('Label 0', l.replace(oldLabel, newLabel))
```

2 Firmware Updates

Firmware controls the various hardware components of your APDM product line (monitors, access points, and docking stations). It is important to keep the firmware up to date to ensure that your system gets the latest bug fixes and has access to the latest features. Firmware updates are bundled with updates to our software products. Firmware can be updated either automatically or manually.

2.1 Automatic Firmware Updates

Whenever you configure your system, your hardware is first checked to ensure that the latest firmware is installed. If not, you will be prompted to automatically update your hardware to the latest versions of the firmware bundled with your system.

2.2 Manual Firmware Updates

Firmware can be updated manually as well. This functionality can be used to either flash the default firmware to one of the hardware components, or to flash a different version. To access the “Update Firmware” dialog, click on “Tools→ Update Firmware” in the menu bar.

2.2.1 Flash Default Firmware

Your system comes bundled with an up to date version of the firmware. This option will re-flash this version of the firmware onto the specified monitor.

2.2.2 Flash Alternate Firmware

For testing purposes or to address an issue in a timely fashion, it may be necessary to flash a monitor with a version of the firmware that is different than the bundled version. You will have to specify the alternate firmware file to use with this option.

2.2.3 Force Update

When using either of the options above, if the firmware version on the target device(s) matches the firmware version to be flashed, the device will be skipped. If the “Force update even if versions match” checkbox is selected, however, the firmware will be flashed even if the versions match. This may be necessary in some cases to recover a malfunctioning device.

3 Calibration

APDM takes special care to ensure that your Opal's are accurate. This includes custom, in house calibration of each individual Opal to account for small variances in the integrated sensors and their exact placement within the device. Each monitor is calibrated individually in a procedure that determines optimal scaling factors and offsets for the accelerometers, gyroscopes, and magnetometers across a wide range of orientations and temperatures.

3.1 Sensor Error Models

The errors modeled and compensated for by the calibration are: scale factor, cross axis sensitivity, sensor misalignment, and bias. For scale factor, there is a linear temperature model, and for bias, a look up table based temperature model. The notation is reused, but each type of sensor has distinct calibration parameters. For example, the scale factor matrix S_T for the accelerometers is different from the one for the gyroscopes, and from the one used for the magnetometers. APDM factory calibration does not compensate for misalignment between the sensors and the case, only misalignment between the accelerometers and the other two sensors.

3.1.1 Accelerometers

The calibrated accelerometer measurements are calculated as

$$\vec{a}_{cal} = CS_T(\vec{a}_{raw} - \vec{b}_T)$$

$$C = \begin{bmatrix} \cos s_{xy} \cos s_{xz} & \sin s_{xy} & \sin s_{xz} \\ \sin s_{xy} & \cos s_{xy} \cos s_{yz} & \sin s_{yz} \\ \sin s_{xz} & \sin s_{yz} & \cos s_{xz} \cos s_{yz} \end{bmatrix}$$

$$S_T = \begin{bmatrix} s_x + T s_{x,T} & 0 & 0 \\ 0 & s_y + T s_{y,T} & 0 \\ 0 & 0 & s_z + T s_{z,T} \end{bmatrix}$$

$$\vec{b}_T = \begin{bmatrix} b_{x,T} \\ b_{y,T} \\ b_{z,T} \end{bmatrix}$$

where C is the cross axis sensitivity matrix, S_T is the temperature dependent scale factor matrix, and $\vec{b}_T$ is the temperature dependent bias vector. There is a look up table for the temperature effect on bias for each sensor axis. The bias value for a particular temperature is linearly interpolated from this table.

3.1.2 Gyroscopes

The calibrated gyroscope measurements are calculated as

$$\vec{\omega}_{cal} = MCS_T(\vec{\omega}_{raw} - \vec{b}_T)$$

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_r & -\sin\theta_r \\ 0 & \sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} \cos\theta_p & 0 & \sin\theta_p \\ 0 & 1 & 0 \\ -\sin\theta_p & 0 & \cos\theta_p \end{bmatrix} \begin{bmatrix} \cos\theta_y & \sin\theta_y & 0 \\ -\sin\theta_y & \cos\theta_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} \cos s_{xy} \cos s_{xz} & \sin s_{xy} & \sin s_{xz} \\ \sin s_{xy} & \cos s_{xy} \cos s_{yz} & \sin s_{yz} \\ \sin s_{xz} & \sin s_{yz} & \cos s_{xz} \cos s_{yz} \end{bmatrix}$$

$$S_T = \begin{bmatrix} s_x + T s_{x,T} & 0 & 0 \\ 0 & s_y + T s_{y,T} & 0 \\ 0 & 0 & s_z + T s_{z,T} \end{bmatrix}$$

$$\vec{b}_T = \begin{bmatrix} b_{x,T} \\ b_{y,T} \\ b_{z,T} \end{bmatrix}$$

where M is the misalignment matrix, C is the cross axis sensitivity matrix, S_T is the temperature dependent scale factor matrix, and $\vec{b}_T$ is the temperature dependent bias vector. There is a look up table for the temperature effect on bias for each sensor axis. The bias value for a particular temperature is linearly interpolated from this table.

3.1.3 Magnetometers

The calibrated magnetometer measurements are calculated as

$$\vec{m}_{cal} = MCS_T(\vec{m}_{raw} - \vec{b}_T)$$

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_r & -\sin\theta_r \\ 0 & \sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} \cos\theta_p & 0 & \sin\theta_p \\ 0 & 1 & 0 \\ -\sin\theta_p & 0 & \cos\theta_p \end{bmatrix} \begin{bmatrix} \cos\theta_y & \sin\theta_y & 0 \\ -\sin\theta_y & \cos\theta_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} \cos s_{xy}\cos s_{xz} & \sin s_{xy} & \sin s_{xz} \\ \sin s_{xy} & \cos s_{xy}\cos s_{yz} & \sin s_{yz} \\ \sin s_{xz} & \sin s_{yz} & \cos s_{xz}\cos s_{yz} \end{bmatrix}$$

$$S_T = \begin{bmatrix} s_x + T s_{x,T} & 0 & 0 \\ 0 & s_y + T s_{y,T} & 0 \\ 0 & 0 & s_z + T s_{z,T} \end{bmatrix}$$

$$\vec{b}_T = \begin{bmatrix} b_{x,T} \\ b_{y,T} \\ b_{z,T} \end{bmatrix}$$

where M is the misalignment matrix, C is the cross axis sensitivity matrix, S_T is the temperature dependent scale factor matrix, and $\vec{b}_T$ is the temperature dependent bias vector. There is a look up table for the temperature effect on bias for each sensor axis. The bias value for a particular temperature is linearly interpolated from this table.

3.1.4 Temperature

The calibrated temperature measurements are calculated as

$$T_c = s(T_r - b_{20}) + 20,$$

where s is the scale factor, T_r is the raw sensor reading, and b_{20} is the raw temperature value at 20 degrees Celsius.

3.1.5 Magnetometer Recalibration

To perform this task, click on the “Tools”→“Recalibrate Magnetometer” option in the menu bar. This wizard will guide you through the process of recalibrating the magnetometers on your monitor(s). The wizard asks that you only undock and collect calibration data one monitor at a time, because they must each be moved independently away from other objects that may disrupt the magnetic field (including other monitors).

4 External Synchronization

4.1 Synchronization Overview

Opal recordings can be precisely time synchronized with external equipment. There are two supported synchronization modes:

- Output synchronization: when the Opal system begins recording data, it will output a signal to external equipment when the recording starts. Output signals can be edge or level triggered.
- Input synchronization: the Opal system will immediately begin recording when it receives a signal from external equipment. Input signals can be edge or level triggered as well.

External Synchronization requires:

- An APDM External Synchronization Box, or “Sync Box”
- A 3.5 mm 4-conductor cable to connect the Sync Box to the access point
- An APDM access point
- That the Opal system is configured for wireless streaming

4.2 Synchronization Hardware

APDM’s Sync Box (v2)

Access Point $\leftrightarrow$ Sync Box Connectors and Cable Both the access point and Sync Box have a 4-conductor 3.5 mm receptacle. A standard 1 m long, 4-conductor, 3.5 mm cable is included with the Sync Box to connect it to the access point. These cables are often called “AV” cables becaus they contain 4, not the usual 3, conductors. A regular stereo audio cable will *not* operate correctly. A longer 4-conductor, 3.5 mm cable may be used, up to 10 m. This cable provides power from the access point to the Sync Box, and provides bidirectional communication between the two (using the Controller Area Network, or CAN, protocol). A green light on the side of the access point, near the 3.5 mm receptable, indicates that the access point is providing power to the Sync Box.

Mode Switch A toggle switch on the top of the Sync Box allows you to switch the operational “mode” of the Sync Box:

- **Normal Mode:** When in normal mode, the In and Out signals are controlled by software, and the square buttons on the top of the Sync Box are ignored.
- **Manual Mode:** When in manual mode, the square push buttons just below the “Out” and “In” LEDs allow the user to manually trigger input and output events. Pressing the “Out” button will toggle the output synchronization signal on the “Out” BNC connector high and low. Pressing the “In” button will toggle the input synchronization signal that goes into the access point high and low. In Manual mode, output signals sent by APDM software and actual input signals from external equipment are ignored.

Status indicator The Status LED is illuminated when the Sync Box has power. It is green when the Sync Box is in “Normal” mode, and blue when in “Manual” mode (described above).

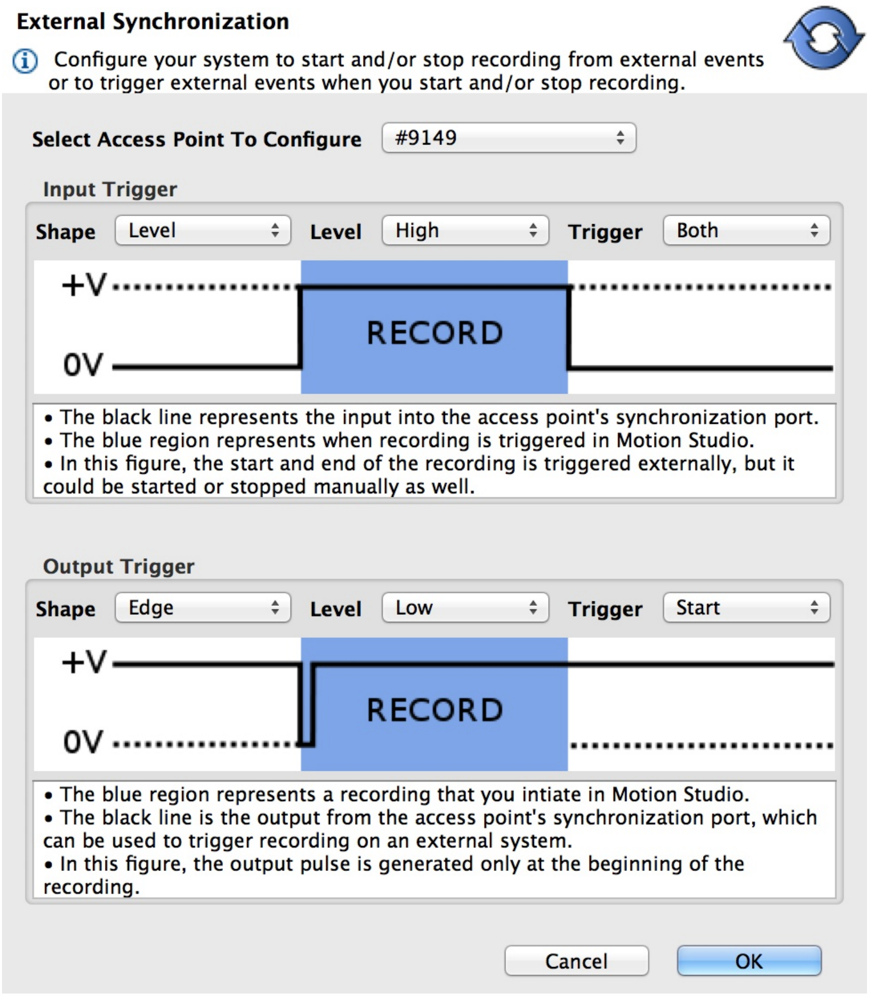
External Synchronization Connectors (BNC Connectors) The Sync Box has two external BNC connectors, labeled “In” and “Out”.

- **In BNC Connector:** The In signal is designed to receive a 0 to 3.3 V (5V maximum) signal from external equipment. The logic high threshold is at 2.3V and the logic low threshold is 0.99V. This means a valid input signal must be below 0.99V when low, and above 2.3V when high. When the input signal is low, the In LED is off. When input is high), the In LED is green. For more technical details on the In signal, please see the *Schematic of Sync Box Input and Output Signals* (below).
- **Out BNC Connector:** The Out signal sends a 0 to 3.3V signal to external equipment. When not excessively loaded, the output high signal is greater than or equal to 2.9V and is typically 3.3V. When the output signal is low, it will be 0.4V or less. When output is low, the Out LED is off, and when output is high, the Out LED is green. For more technical details on the Out signal, please see the *Schematic of Sync Box Input and Output Signals* (below).

Firmware Updates If there is ever a need to update the Sync Box’s firmware in the field, there is a small panel on the side of the Sync Box which exposes a USB port that can be used to perform the update.

4.3 Configuration

External synchronization options are selected using the External Synchronization Configuration dialog. You must specify the access point that will be connected to the Sync Box. Only one access point (and thus one Sync Box) can be specified at a time.



The External Synchronization Configuration Dialog

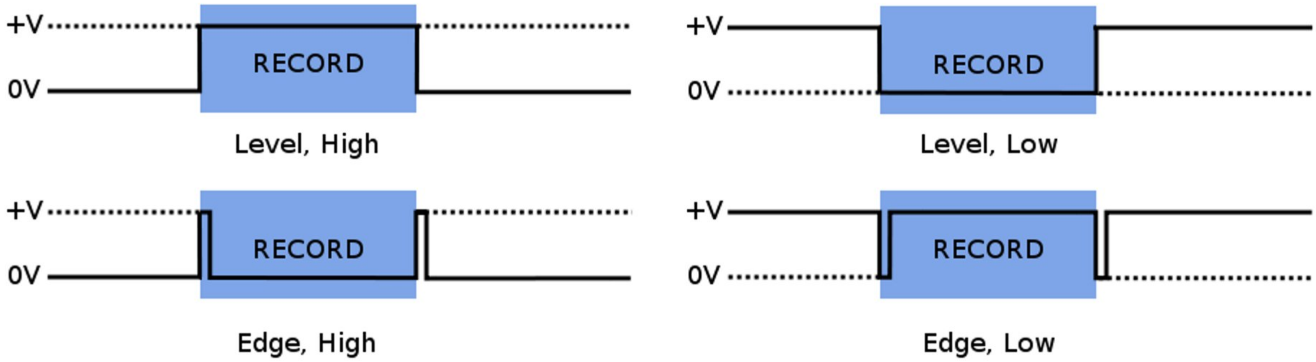
4.3.1 Input Trigger Shape

The input trigger shape indicates the type of signal that will be input into the specified access point and how you want your Opal system to respond. In the figure above, the four basic trigger shapes are shown. The solid black line represents the external synchronization signal being sent to the access point. The blue shaded region represents the period that will be recorded by your Opals.

4.3.2 Input Trigger Level

Input triggers can be either low or high, depending on the nature of the signal generated by your external synchronization source.

4.4 Input Synchronization



Input synchronization trigger types

4.4.1 Input Trigger

There are three input trigger options available:

- Start: The external trigger will only be used to start recording by your Opals.
- End: The external trigger will only be used to stop recording by your Opals.
- Both: The external trigger will be used to start and stop recording by your Opals.

4.4.2 Sample Selection with External Input Trigger Events

The time of the external input trigger events may not align exactly with the time of an individual samples being collected in Motion Studio due to the discreet sampling interval. If the start trigger event time does happen to align exactly with a sample captured in Motion Studio, the first sample recorded will correspond exactly to the time of the start trigger event. If these do not align exactly (as will generally be the case) the sample following the start trigger event will be the first sample recorded. Similarly, if the stop trigger event aligns exactly with a sample captured in Motion Studio, the last sample recorded will correspond exactly to the time of the stop trigger event. If these do not align exactly, the sample preceeding the stop trigger event will be the last sample recorded.

4.4.3 Annotation of Externally Triggered Recordings

Note: Annotations are implemented for the HDF file format only. When an external “Start” trigger event is detected, an annotation is added to the recording that indicates the name of the event (in this case “External trigger start time”) along with the timestamp of the event in epoch microseconds. Similarly, when an external “Stop” trigger event is detected, a timestamped annotation is added to the recording (in this case labeled as the “External trigger stop time”). These annotations allow you to align the recording captured by your Opals with your external events in the case where the external trigger event times do not exactly align with the samples captured in your HDF file.

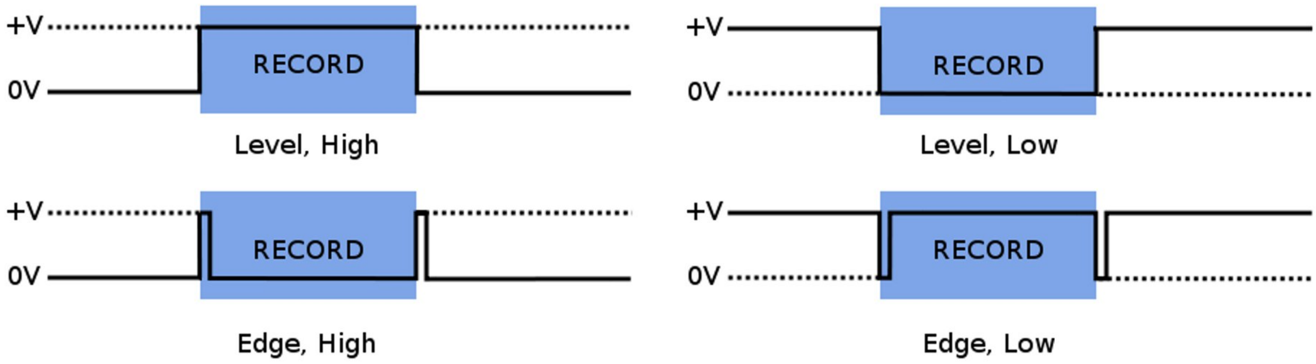
4.4.4 Output Trigger Shape

The output trigger shape indicates the type of signal that will be generated by the specified access point when recording is started and stopped by your Opals. The output trigger shapes are identical to the input trigger shapes, but in this case the solid black line in the figure above represents the signal being output by the configured access point. The blue shaded region represents the period being recorded by your Opals, initiated either through user selection of the start/stop buttons in the “Stream” dialog, use of the wireless remote, or an external synchronization event. Unlike input triggers, output triggers are processed even if the “External Sync” option is not specified in the “Record Duration” panel of the “Stream” dialog.

4.4.5 Output Trigger Level

Output triggers can be either low or high, depending on the requirements of the external system receiving the synchronization signal.

4.5 Output Synchronization



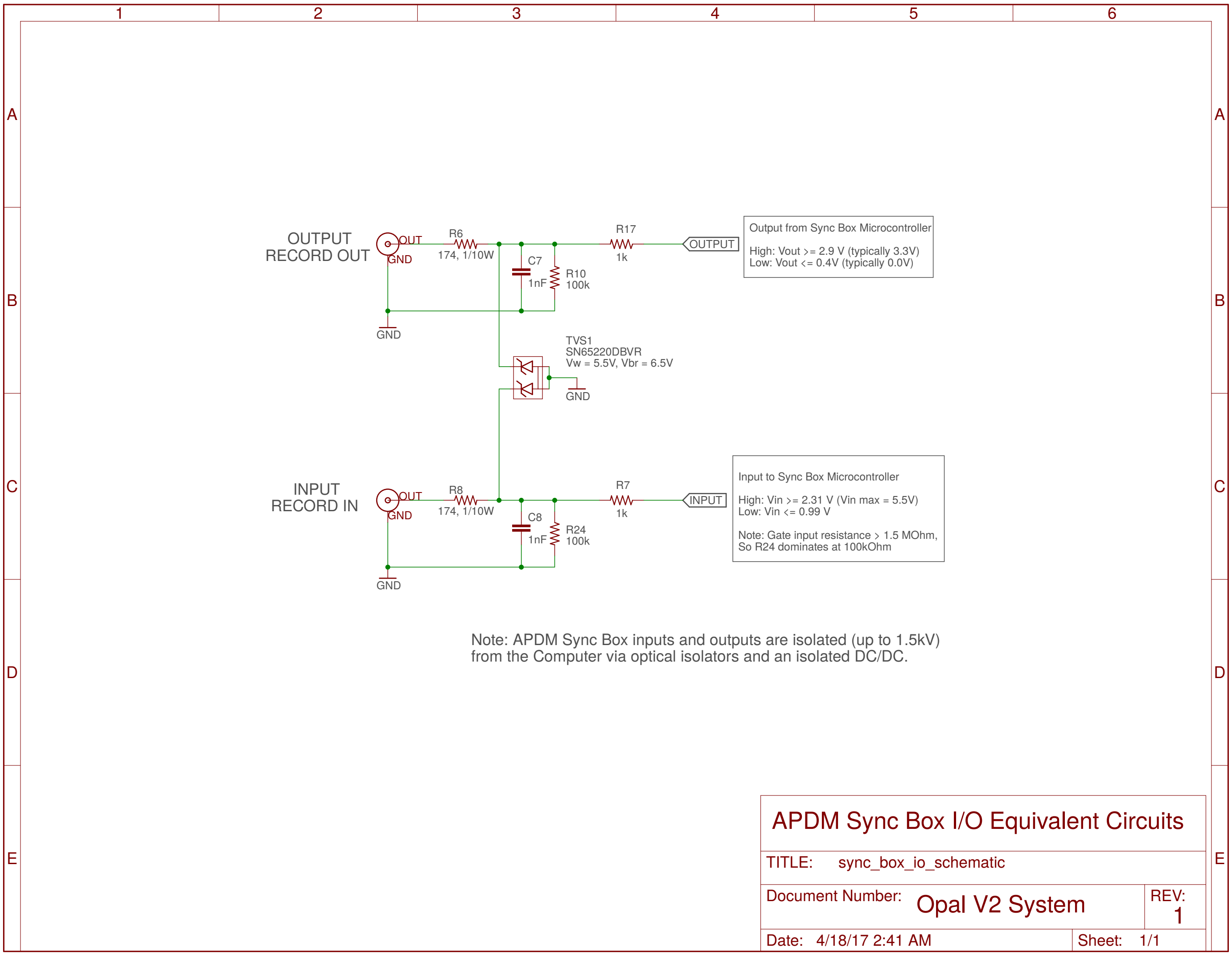
Output synchronization trigger types

4.5.1 Output Trigger

There are three output trigger options available:

- Start: The external signal will only be generated when recording is started by your Opals.
- End: The external signal will only be generated when recording is stopped by your Opals.
- Both: The external signal will be generated when recording is started and stopped by your Opals.

4.6 Schematics



Schematic of Sync Box Input and Output Signals

4.6.1 V1 Synchronization Hardware

Note: This section applies to our first major hardware iteration (v1). It includes the access point that does not have a display.

APDM's access points come fitted with a 6 pin digital I/O connector used for synchronization with external systems. To connect an access point to your external equipment, you may have to create a custom cable that can interface with both components. Below we provide the technical specifications necessary to complete this task. Feel free to contact our technical support at support@apdm.com if you require assistance or have additional questions.

The Isolated External Interface for the AP consists of an auxiliary power supply, two GPIO lines (one in, one out), and an inter-AP sync signal. All signals in the isolated external interface section (including power and ground) are isolated from the remainder of the board using an RF solution similar in operation to an opto-isolator. Further, all signals in the isolated external interface are 5V tolerant and ESD protected beyond the 15kV human body model.

The connectors used in the isolated interface consist of one standard female RCA, and one female 6 pin mini-din connector. The RCA connector mates to almost any basic RCA cable similar to those used in audio systems. When choosing an RCA mating connector, choose one that has uncovered bare shield spades to allow the connector to fit fully into the recessed hollow in the AP body.

The 6 pin mini-din connector is similar to those used for older style PS/2 keyboards and mice. Choose a connector that is small enough to fit fully inside the recessed hollow in the AP body. Some PS/2 extension cables can be cut into excellent pigtails for this connector.

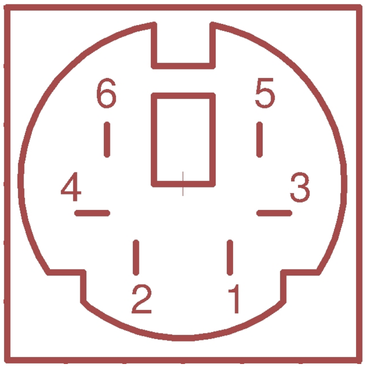
RCA Inter-AP Sync Connector

- RCA Connector: Digikey Part number RCP-021, CUI INC
- Center Pin: Inter-AP Sync
- Sheild: Isolated Ground

6 Pin Digital Input/Output Connector

- 6 Pin Mating Connector: Digikey part number CP-2060-ND, CUI Inc part number MD-60.
- 6 Pin Mating Pig Tail Cable: Digikey part number 839-1051-ND
- *Note these connectors may need the outer shell trimmed to fit into the AP case, a better solution is often pigtail cables that have over-molded ends and excellent strain relief.*

- Pin 1: Record In



AP 6 Pin Digital Connector

- Pin 2: Output Voltage Select. When connected to positive (pin 6), I/O will be in 5 volt mode. 3.3 volt mode otherwise.
- Pin 3: Isolated Ground (isolated gnd)
- Pin 4: Inter-AP synchronization output signal. 2.56 khz square wave used for synchronizing timing among multiple access points.
- Pin 5: Record Out
- Pin 6: Isolated Vdd, unregulated. 3.3 V or 5 V depending on whether it is connected to pin 2.

The auxiliary power supply is meant to provide for powered external interface solutions, allowing a small circuit to be powered directly from the AP. Accessed via pin 6 of the mini-din connector, the auxiliary power supply is rated for operation up to 250mW at 3.3V or 5V operation. While default operation is at 3.3V, 5V operation can be selected by shorting pin 2 to pin 6 of the 6 pin mini-din connector.

The inter-AP sync signal is a 2.56kHz clock signal used to keep multiple AP configurations in sync with one another. The inter-AP sync signal is available on the RCA connector, as well as pin 4 of the 6 pin mini-din connector next to it. The signal is a square wave pulse that is driven by the 'master' AP (usually the first AP to enumerate) and received by up to seven additional APs (depending on output voltage selection and cable length). In operation the signal is weakly pulled up to the isolated power rail by each AP in the system, and driven directly to ground only by the 'master' AP to produce the pulsed waveform.

Two GPIO lines are available, one input and one output. Both are pulled down by 47.5kΩ resistors, and each have a series resistance of nearly 1.2kΩ due to the methods used to protect the lines from over-voltage/overcurrent conditions. The input signal is available on pin 1 of the 6 pin mini-din connector and is typically used to start/stop data collection by the host PC. Driving the line high to 'record' and low to 'not-record' is the default operation, though this is user selectable in software to allow for other modes of operation. Similar to the input line, the output line is typically used to start/stop data capture on external systems. The line is driven high by the AP when 'start recording' is selected in software, and driven low when recording stops. Opposite high/low operation can be software selected at time of configuration for both input and output signals.

- Note: The pin diagrams below show the interface on the AP and not the cable. The pin layout on the cable is the mirror image of these diagrams.
- Note: A cable designed to trigger recording in Motion Studio from an external synchronization event must make use of isolated ground (pin 3) and record in (pin 1). Assuming a voltage range of 3.3V, these are the only pins that need to be implemented. If the voltage range is 5V, pins 2 and 6 must be connected.
- Note: A cable designed to trigger an external system when recording is started or stopped in Motion Studio must make use of isolated ground (pin 3) and record out (pin 5). Assuming a voltage range of 3.3V, these are the only pins that need to be implemented. If the voltage range is 5V, pins 2 and 6 must be connected.

External Sync Box

The external sync box is meant to allow for easy access to the access point external digital expansion port. A shielded straight-through six conductor cable connects the AP to the sync box, BNC connections outside the box allow for simple connections to remote equipment.

AP External Sync Box

Three external BNC connections:

- AP-AP: This connection should only be connected to another AP, it is used to keep multiple APs in sync and can be used to connect multiple APs in a star or daisy chain configuration (both will work).
- AP In: This input to the AP can be configured via software to allow an external device to begin and stop recordings. Accepts both +3.3V and +5V logic levels.
- AP Out: This AP output can be configured via software to drive low, high, or pulse at record start/stop points allowing synchronization with an external system (such as a camera motion capture system). *Note: the default configuration for AP Out is +3.3V logic levels, though +5V levels can be selected using the voltage select switch located in the top of the box.*

Four LEDs indicators:

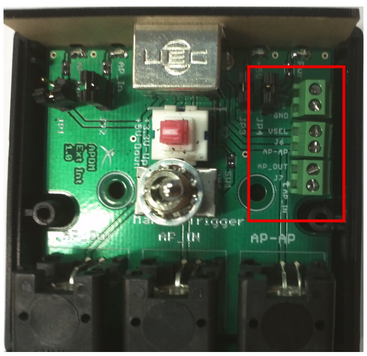
- PWR: Lights when power is applied to the external interface.
- +5V: Lights to indicate that the external interface is configured for 5V operation. Default is 3.3V (Light out)
- AP In: Lights to indicate that the APIIn signal is High
- AP Out: Lights to indicate that the AP Out signal is High

Push-button and Toggle switch:

- Push Button: Up to select 3.3V operation. Down to select 5V operation.
- Toggle: Manually ties the AP In signal to the positive voltage rail allowing for manual triggering of recordings (software configured).

Additional connections and functionality are located inside the box and can be accessed by removal of the box top: JP1 through JP4 can be removed to disconnect the corresponding LED.

Six Euro-style screw terminals can be used to directly connect to the six wires in the AP cable:



AP External Sync Box

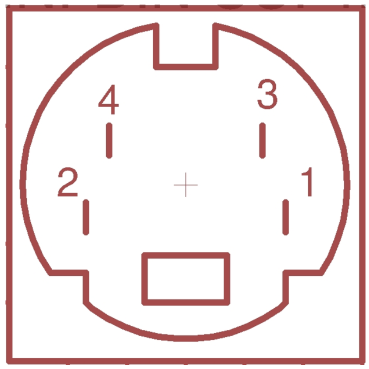
- +V: Positive voltage rail from the isolated supply located inside the AP.
- GND: Ground rail from the isolated supply located inside the AP.
- VSEL: Tie to Ground or leave floating to select output and +V operation at +3.3V, tie to +V to select +5V operation.
- AP-AP: Allows multiple AP configurations, tie only to the same port of another AP.
- AP Out: Digital output from the AP. Default is 0V to +3.3V, but can be configured for 0-5V operation.
- AP In: Digital input to the AP from an external source or the manual trigger toggle switch.

Note: The AP is able to safely source only 50mA on the +V rail.

4 Pin Analog Input/Output Connector

Note: This connector is currently reserved for future expansion.

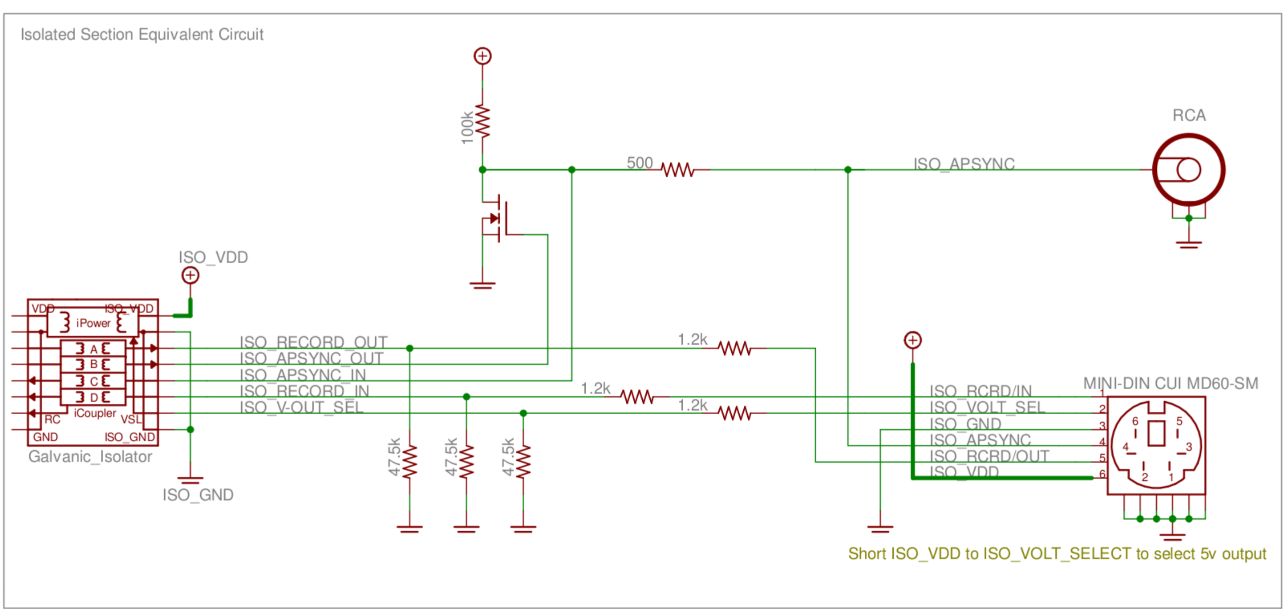
- 4 Pin Mating Connector: Digikey part number CP-2040-ND, CUI Inc part number MD-40
- 4 Pin Mating Pig Tail Cable: Digikey part number 839-1049-ND



AP 4 Pin Analog Connector

- Pin 1: Analog In (0 to 6 volts)
- Pin 2: Analog Out (0 to 5 volts or 0 to 3.3 volts depending on software controled configuration)
- Pin 3: No Connect (reserved for future use, avoid connecting this pin)
- Pin 4: Ground (gnd). This is the same ground as USB, and depending on how your USB hub and/or laptop are designed electrically, may also be the same ground as the hub and laptop. Consideration should be taken for ground loops.

4.6.2 Schematic



5 Troubleshooting

APDM is pleased to assist you with any issues you encounter using our products or questions you have about using our products.

Please contact us at:
web: support.apdm.com
email: support@apdm.com