

iVn Video Capture Interface

Hardware Reference



Updated 2024-12-19

© 2016-2024 Tucker-Davis Technologies, Inc. (TDT). All rights reserved.

Tucker-Davis Technologies
11930 Research Circle
Alachua, FL 32615 USA
Phone: +1.386.462.9622
Fax: +1.386.462.5365

Notices

The information contained in this document is provided "as is," and is subject to being changed, without notice. TDT shall not be liable for errors or damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

The latest versions of TDT documents are always online at <https://www.tdt.com/docs/>

Table of Contents

iVn Video Capture Interface

Camera Options	4
USB3CAM Kit for iV2	4
USB3CAM_MONO	4
iV1_CAM Kit	5
iV1_CAM_MONO Kit	5
Status Lights	5
LCD Screen	5
Frame Buffer	6
Temperature Indicator	6
Camera Status	6
Compression	7
iV2 Example Frame Rates	7
iV1 Example Frame Rates	7
IR Support	8
Software Updates	8
Software Control	8
Converting MP4 to AVI	8

iVn Video Capture Interface



The iV1/iV2 iCon modules connect to USB3 cameras to record video frames synchronized with all behavioral, electrophysiology, fiber photometry, and other research data. Record video from up to two cameras simultaneously per iV2 module or one camera per iV1 module. The iV2 also features real-time subject tracking.

Camera Options

USB3CAM Kit for iV2

Each kit includes:

- One [Basler ace 2 a2A1920-160ucBAS](#) color camera w/ IR filter and lens
- A 3 meter locking USB cable and camera sync cable

USB3CAM_MONO

Each kit includes:

- One [Basler ace 2 a2A1920-160umBAS](#) monochrome camera and lens*
- A 3 meter locking USB cable and camera sync cable

* requires iV2 v1.3 firmware or above. See [Software Updates](#)

iV1_CAM Kit

Each kit includes:

- One **Basler dart daA1920-30uc** color camera and IR filter lens
- A 3 meter locking USB cable and camera sync cable

iV1_CAM_MONO Kit

Each kit includes:

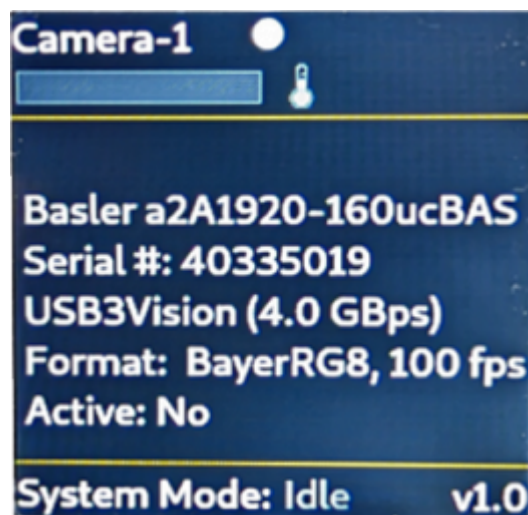
- One **Basler dart daA1920-30um** monochrome camera and lens
- A 3 meter locking USB cable and camera sync cable

Status Lights

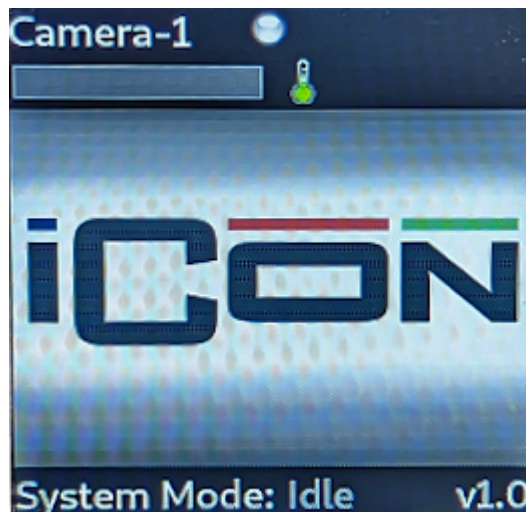
Each camera has an Error and Active LED. All Err and Act LEDs will flash during the boot process, or if the iCon has no communication.

LCD Screen

Press the Info button to toggle between live view and camera information



Camera Information



Recording Status

Frame Buffer

Synapse triggers asynchronous frame grabs during acquisition. The iVn has an onboard 10 second frame buffer that allows it to catch up after periods of rapidly changing images. A small progress bar indicator on the LCD screen shows the state of this buffer. If it fills up, you will see Frame Errors in the Synapse run-time tab, indicating lost frames. The frame buffer is shared by both cameras on an iV2.

Temperature Indicator

Next to the status bar is a temperature sensor. Make sure the iCon housing the iVn has adequate room to dissipate heat.

Camera Status

LED status	Camera Status
hidden	no camera detected
gray	detected but not running
red	running with errors
yellow	running but not capturing frames
blue	running and capturing frames

Compression

The iVn streams data at a fixed bit rate, determined by the Compression Quality setting in Synapse. The value ranges from 1 to 5, which corresponds to 2.5 MB/minute to 22 MB/minute. Therefore the size of a 30-minute video ranges from ~75 MB to ~660 MB depending on this setting. The actual quality of each frame (pixelation) will depend on how much movement there is in the image, like you would see with any live camera feed streaming at a fixed bit rate.

Compression	MB/minute
1	2.5
2	3.7
3	7
4	14.5
5	22

The iVn generates mp4 files with three choices of encoding: H264, H265, MJPEG. The max FPS is limited by the choice of the encoder. H264 is around twice as efficient as H265 or MJPEG without a perceived difference in quality, so you can get twice the frame rate using H264.

iV2 Example Frame Rates

A single camera running on the iV2 using H264 encoding can record 1920 x 1200 @ 40 FPS, 1280 x 1024 @ 60 FPS, or 640 x 400 @ 200 FPS. If you're running two cameras per iV2, then cut those FPS numbers in half. Each camera can run at independent resolutions and frame rates.

iV1 Example Frame Rates

The iV1 camera can record at 1920 x 1080 @ 25 FPS, 1280 x 1024 @ 40 FPS, or 640 x 400 @ 120 FPS. There may be some frame errors during initial startup at higher frame rates.

IR Support

In addition to the monochrome [camera options](#) listed above, current users with the a2A1920-160ucBAS color camera can also optionally remove the IR filter from the camera itself following the [instructions here](#).

Software Updates

In case the iVn needs a software update, here is the procedure:

1. Copy the files provided by TDT to a USB drive.
2. Insert the USB stick into the iVn.
3. The iVn will automatically update itself. After ~5 minutes, the iVn screen will indicate that it is finished.
4. Remove the USB drive, and the iVn will reboot itself to complete the process.
5. Verify that the new software version number is correct on the iVn screen.

Software Control

See the [Synapse Manual](#) for software control, including how to set up real-time subject tracking for iV2.

Converting MP4 to AVI

The iVn only records MP4 files. To convert MP4 to AVI, you can use FFmpeg. This example will convert all the MP4 files in a folder/subfolders into AVI files with similar quality.

1. Download FFmpeg (<https://www.ffmpeg.org/download.html>). On Windows, the latest version can be downloaded from this [github link](#).
2. Extract the contents onto your hard drive. In the example below, FFmpeg was extracted onto the `D:\` drive, and the tanks that will be converted are located in the `D:\Tanks` directory.
3. Copy the code below into Notepad, adjust the "FFMPEG" and "SOURCE" variables as needed. Save it as MP4toAVI.bat on your desktop.
4. Close the MP4toAVI.bat file and then double-click it on your desktop to run the commands.


```
@echo off
setlocal enabledelayedexpansion
set FFMPEG=D:\ffmpeg-master-latest-win64-gpl\bin\ffmpeg.exe

REM Set the source folder path
set "SOURCE=D:\Tanks"

REM Loop through each MP4 file in the source folder and its subfolders
for /r "%SOURCE%" %%F in (*.mp4) do (
    REM Extract the base name of the file (without extension)
    set "basename=%%~nxF"

    REM Remove the .mp4 extension and add .avi
    set "outputfile=!basename:.mp4=.avi!"

    REM Construct the full path for the output file
    set "outputpath=%%~dpF!outputfile!"

    REM Output the input and output file names to the console
    echo Converting: %%F to !outputpath!

    REM Perform the conversion using FFmpeg
    %FFMPEG% -y -i "%%F" -vcodec copy -acodec copy "!outputpath!"
)

endlocal
```