Questions:

1. What specifications should we stipulate while ordering Sentera Senor?

* You should specify what mode (ie filter type) that you want.  Since this is a custom integration, we should also discuss what connections you wish to use on the sensor and we can either provide connector part numbers, or build cables with flying leads for you to connect to your system.
* For your application, you will most likely just need power and optionally Ethernet/Serial
* You may also need to specify if you want the camera by itself, or with a gimbal.

1. What is the linkage between the Sentera and the HoloLens---hardwire or wireless?

* The primary interface is over 10/100/1000 ethernet, however the camera can also act as a wireless access point that the hololens could connect to.
* Depending on your application, we also have 3.3V serial and i2c, however these are limited in what protocols they support.

1. What is the FOV of the Sentera for this application?

* 59.7 degree horizonal
* 46.6 degree vertical

1. Does it have an angling/panning capability as it needs to see as required; can that be controlled?

* We do sell versions that have gimbals that you could potentially control.  However most of these are designed to mount to existing aircraft with very specific connectors.  If you are interested in a gimballed solution, we would have to discuss custom options.

1. Can it do both single picture and video

* Yes, you can stream/record video at the same time you capture full frame images
* The main llimitation is that the resolution and bit rate you are streaming at determines the resolution and bit rate of the video recording.

1. What is the power requirement for the Sentera; and everything related to power

* We accept 6-26V on the input power side, and pull up to 18W.
* ~12W is typical when streaming from both sensors and captures images at 1 Hz
* To avoid excessive current, we recommend >=12V

1. What about the trigger: manual, remote etc?

* Triggering can be done in a number of ways.
  + Trigger via UDP packets over ethernet/wireless
  + Trigger on a fixed time interval
  + Trigger based on GPS location
  + Trigger via the built in web page
  + Trigger via a a REST api over http

QUESTIONS:

(A)    Can it connect to a pre-existing Wifi network? ? By the sound of it, it can only act as an access point to its own network, which means we would not be able to connect to it wirelessly (if we wanted to) and still communicate with other wireless devices??

* Currently, the firmware only allows it to be set up as an access point, so it cannot connect to an existing wifi network.
* It could be capable of connecting as a client, but that would require additional firmware development.  If this important to you, we could discuss pricing on adding this firmware feature.  Otherwise, the ethernet interface is available.

(B)    As for power connectors, as long as we have bare leads, we can assemble the power cables. The easiest approach then would be to have a cable with the connector pre-installed on the sensor's mating end and bare wire on the other end so we can crimp on whatever we need.

* What length of flying leads would you need on the power cable?
* If you use ethernet, what length ethernet cable would you need?

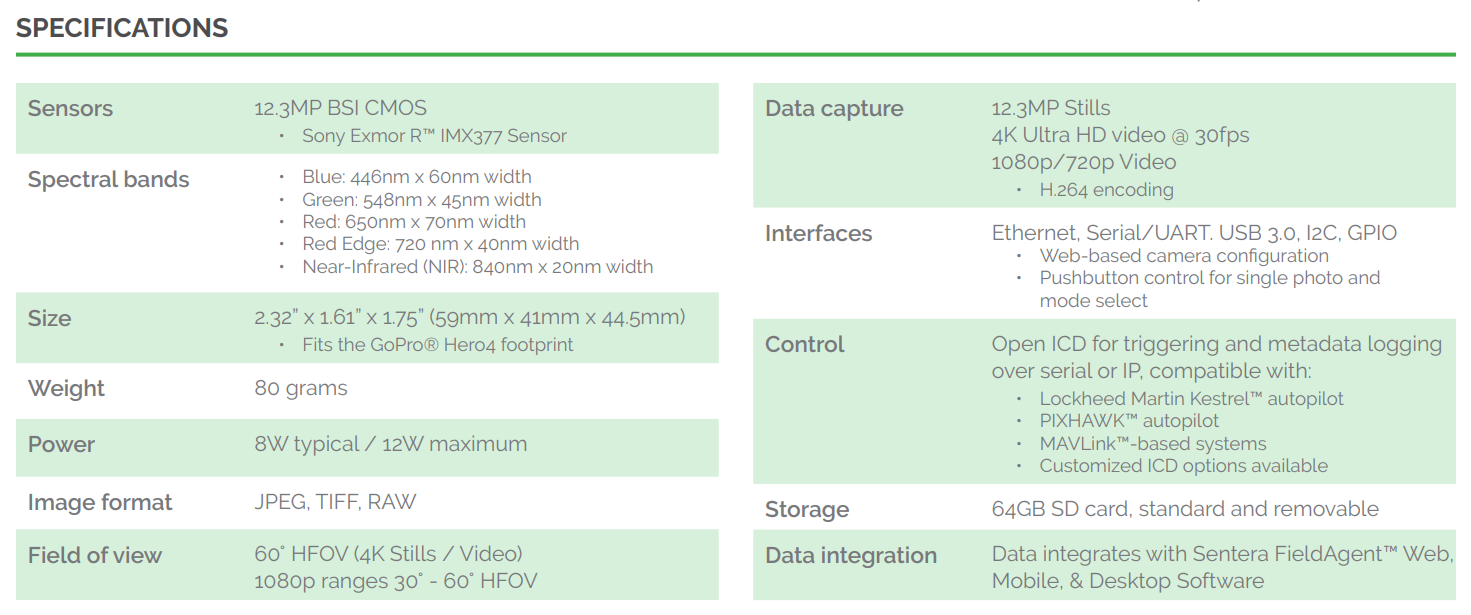
SPECS WE NEED:

A.      4K 5 Band Multispectral Imager

B.      Ethernet, Power and Wifi connection

C.      Trigger via Ethernet UDP

D.     No Gimbal



Sensor:

* Sony Exmor R IMX377

Wifi 802.11ac vs 802.11n

* <https://www.extremetech.com/computing/160837-what-is-802-11ac-and-how-much-faster-than-802-11n-is-it>

Use Pi3 with USB 3.0 Gigabit Adapter

* <https://www.jeffgeerling.com/blogs/jeff-geerling/getting-gigabit-networking>
* I tested four different 802.11n WiFi USB adapters I had laying around, and they got between 30-45 Mbps.
* I tested the internal 10/100 NIC on two different B+s (I don't yet have my Pi 2), and it got about 94 Mbps
* I tested a TRENDnet Gigabit USB 3.0 adapter, and it got a whopping 222 Mbps (nearly 30 MB/sec)!

4k Video @ 30fps

* 35-45 Mbps maybe more

Henry,

To answer a few of your questions:

**1. We saw that the sensor hardware can compress to JPEG. With 5 bands of data, how are they compressed?  RGB + 2 bands? Will we be able to specify JPEG compression quality?**

The data from the sensors comes off as two separate RGB images.  The bands in the jpeg correspond to a different mix of each wavelength, which can be separated by performing some band math across the image.  I have created an appendix at the bottom of this email with details on the calculations that we do to extract different bands.

We would also have the capability to perform the band math on board the sensor before saving the image if it is required for you application, though we would have to spend some development time on adding the feature to the current firmware.

You will be able to specify the compression quality.

You can also stream the video feed over UDP for one or both cameras as a raw H264, H264 w/ RTP headers, or Mpeg2ts if you are interested in video rather than stills.

**2. What is the general procedure/protocol for pulling images over I2C/BT/WiFi/Ethernet? We will primarily be working with ethernet but understanding the other options is convenient as well.**

Pulling images from the camera is typically done in one of 3 ways:

* Removing the SDcard to copy the files directly
  + Not really applicable in your application
* File transfer via a Samba Server over TCP/IP
  + This is easier to use in a windows environment, but is typically slower than http
* Http requests to the web server to download a given image
  + Lower overhead, so can be quite a bit faster

The two Ethernet options can be done over either the Wifi or Ethernet interface.  We do not support I2C or Bluetooth for communication or data transfer.

The customers that have been interested in pulling down the data as it was taken have typically used the UDP Sentera Protocol (see attached) to watch for new image notification.  They then use the url in the new image notification to download the captured image.

The packet you would be interested in is Image Data Ready (0x85) which gets pushed whenever a new image file is saved.

**3. Do you have a full datasheet with registers for configuration and control of the sensor?**

See attached.

The UDP command and control is the interface that has undergone the most testing.  However, if you want to use a TCP/IP interface we do have some http/json calls that we use for our internal web page.  If that is something you are interested in, I could write up a short document on what it controls.  It only contains a subset of the available commands over UDP, but it would be an option.

**---------------- Multispectral Camera Band Details -----------------------------**

Our cameras use a combination of an external filter and the sensors bayer filter to produce an image in the wavelengths mentioned by the datasheet.

For the RGB side of the camera, the stacked filters result in the following spectral response:

In the above chart, red/green/blue correspond to channels of the resulting jpeg image.  Note that there is some cross talk between the channels.  That is, the blue channel has primarily blue wavelengths, but also has small contributions from the green and red wavelengths as well.

If it is necessary to separate the channels more and the object your are images does not have impulse like wavelengths present in the spectrum, you can use a system of equations to subtract out the effect of the out of band channels on each band.  At the same time, you can add a constraint that the power across the red/green/blue should be equalized (to make calculations between the bands relavent).

This results in the following:

DN[color] = digital number of that band (from the jpeg)

color = Corrected color after subtracting out of band colors

Blue  =  1.377\*DNblue - 0.182\*DNGreen - 0.061 DNRed

Green = -0.199\*DNblue + 1.420\*DNGreen - 0.329 DNRed

Red   = -0.034\*DNblue - 0.110\*DNGreen + 1.150 DNRed

For the NIR/Red Edge camera, we have the following combined spectral response:

Note that from this imager, we only use the Blue channel for NIR and the Red channel for Red Edge.  The green channel is discarded when performing index and band calculations.  This is because there is not any additional useful information in the green band that isn't already captured by the blue and red bands.

Similar to the RGB method above (and subject to the same restrictions), we can subtract out the our of band wavelengths in the NIR and Red Edge portions via the following:

RedEdge = -0.956\*DNblue + 1.000\*DNred

NIR     =  2.426\*DNblue - 0.341\*DNred

When attempting to calculate indices that use both cameras, both the camera settings and the normalization factors used in the band math equations need to be taken into account during any index computations you may make.

You will first have to normalize each of your images for the total exposure opportunity.  This will allow images with differing ISO and exposure times to be compared properly.

DN = Digital Number

Gain = ISO / 100 [Exif Tag]

Shutter = Shutter time in seconds [ Exif Tag ]

ExposureOpportunity = DN \* Gain \* Shutter

The normalization values for each band separation matrix were chosen independently for each camera to keep 8 bit values in their proper ranges (typically 0-255). A normalization value of 1/750 was used for RGB, and 1/277.7 was used for RedEdge.  This means that a step increase in RGB values implies a 2.7x step in the RedEdge values.  So, if you want to compute indices that use both cameras you have to take into account our band math gains.  This is done after the bands are separated using the above system of equations.  So you end up with the following:  
  
NDVI      =  (2.700 \* NIR\_2 - Red\_1) / (2.700 \* NIR\_2 + Red\_1)

NDRE      =  (NIR\_2 - RedEdge\_2) / (NIR\_2 + RedEdge\_2)