Proof of Concept Test Demo Sheet

TEAM NAME: metaTait **Test Date & Time:** 12/1/16 at 10:30 am

Test # and Name: #3-Image Processing/Data Packet Creation Program **Test Type:** Information Gathering

1) Purpose of Test and How it Relates to Project

- Characterization: Critical to project implementation.
- Purpose: A computer program must efficiently pre-process a varied amount of arbitrarily sized images to meet the physical resolution and ordering requirements determined by the LED and cage dimensions. This pre-processing program must down sample and adjust images to the LED pixel resolution dictated by the physical dimensions of the cage, and create a data packet in a text file.
- Without the pre-processing (color extraction, down sampling, aspect ratio adjustment and pre-assignment of image sections to LED strips) and creation of a data packet already formatted to the cage parameters, the on-board MCU would have too large of a processing burden to maintain the refresh rates the LEDs require to display an image upon cage rotation.

2) Test Setup, Pre-conditions, and Procedure:

Test Setup:

- ✓ Install Matlab
- ✓ Working directory must contain: a folder holding images in a standard image format (ex. .jpeg, .png), datapacket.txt, AdjustToAspectRadio.m, CalcDiscreteTimePostData.m, CreateDataPacket2.m, CreateImageMatrix.m, DownsampleImage.m, MainProgram.m and SavePacketToFile.m
- ✓ Set Matlab's current folder/path to this working directory

Pre-conditions:

- ✓ Pick desired image resolution height H and width W that will display on the cage. H = 44 and W = 72 matches our current physical design.

Procedure:

- ✓ Set a breakpoint at function 'end' in MainProgram.m (currently Line 52)
- ✓ Navigate to Matlab's Command Line
- ✓ Execute program: >> MainProgram('ImageFolderName', height, width)
- ✓ Examine Matlab Figure outputs, Workspace Variables and datapacket.txt file

3) Design Info:

- ✓ If N images are in the images folder, N downsampled and aspect adjusted images displays in a Matlab figure. The Nth image is Workspace variable *downsampledImageMatrix* of type uint8 and should have dimensions HxWx3
- ✓ The *dataPacket* variable is a 2D matrix of type double with dimensions 6 x (H*(W/6)*N*3)
- ✓ datapacket.txt is updated to hold the *dataPacket* variable written as comma-separated Hex values.
- ✓ The *CalcDiscreteTimePostData* function pieces together and displays N new figures. These figures concatenate 6 vertical sections of a subset of the N processed images according to discrete time increments t1-tN. Neighboring sections of an image should be located at neighboring times.

4) Instructional Team Notes:

5) **Test outcome and what was learned:**