Tetracam
μ-MCA
User's Guide

Tetracam Inc
21601 Devonshire Street Suite 310
Chatsworth, CA 91311

Notices

The content of this User's Guide may differ in some details from the product or its software. For best results, make sure you are running the latest version of the PixelWrench2 application software.

All information in this document is subject to change without notice.

Unless otherwise provided by written agreement with Tetracam Inc., this publication is provided “as is” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Some states do not allow disclaimer of expressed or implied warranties in certain transactions, so this statement may not apply to you.

While reasonable efforts have been made to assure the accuracy of this document, in no event will Tetracam be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect in this publication or the associated software. Tetracam Inc. reserves the right to change this document at any time without obligation to notify anyone.

Trademarks

Windows™ is a registered trademark of Microsoft Corporation.
Photoshop™ is a registered trademark of Adobe Systems.

Other brand or product names are trademarks of their respective holders.

Version 1.1 - Last Modified March 1, 2015

INC

Tetracam Inc
21601 Devonshire Street Suite 310
Chatsworth, CA 91311

Table of Contents

Introduction

This User's Guide contains general information about Tetracam's Micro Multi-Camera Array (μ-MCA) and Micro Multi-Camera Array Snap (μ-MCA Snap) imaging systems. It covers installation, operation, options and accessories, warranties, and technical support. The information is specific to firmware version 5.182 and later – users with earlier firmware should upgrade their firmware so that their units conform to the information herein.

The difference between the μ-MCA and μ-MCA Snap systems is the sensors they employ. The μ-MCA cameras use rolling shutter sensors. These scan each image, line-by-line, top to bottom. Rolling shutter sensors are less expensive and suitable for fixed installations or applications where motion is modest such as aboard a manned fixed wing aircraft at altitudes greater than 2,500 feet AGL, or on a slow, very stable UAV platform. μ-MCA Snap system sensors expose the entire image at the same instant in time. Global shutter sensor μ-MCA Snap cameras are preferred for use where reducing motion artifacts is essential such as in manned aircraft at altitudes lower than about 2,500 feet, or for use in faster and less stable UAV platforms. Because these two models are alike in every other way, all references to μ-MCA in the text throughout this manual shall refer to both the μ-MCA and μ-MCA Snap systems unless otherwise noted.

Tetracam's μ-MCA is an updated version of the company's previously-manufactured mini MCA. It is 20% lighter than the mini MCA. It is delivered with a 10X faster USB interface (enhanced USB 2.0) and a larger default image memory (16 GB rather than 2 GB per channel). The larger memory enables users to capture thousands of additional images during each imaging mission. Like mini MCA systems, all μ-MCA systems are available in three sub-models. The μ-MCA4 contains four digital cameras. The μ-MCA6 contains six digital cameras. The μ-MCA12 contains twelve digital cameras.

Each system's digital cameras are organized in an array, with discrete filters installed in each camera in the array between the camera's lens and sensor. The primary use of this product is to capture spectroscopic signatures of vegetation, chemicals and geology using a set of filters tailored to the targeted substance.

The purpose of this document is:

1. To guide the user through the installation of the product and its supporting software on its target host system.
2. To describe the basic camera operating procedures.
3. To describe the interaction between the camera's interface software and the image editing and archiving software it may be used with.

This document assumes that the user is familiar with the operation of an IBM compatible personal computer running the Windows 8, Windows 7, Windows Vista, or Windows XP operating system. The user should be familiar with the use of USB serial ports and USB disk devices, and with the use of spectroscopic signatures to identify materials of interest.

Quick Start Instructions

Note: A PDF version of this manual is supplied on the installation CD.

Connect the unit to a DC power source. The μ-MCA system will accept a DC voltage input between + 9.0 VDC and + 16.5 VDC, such as the power commonly available from vehicle DC batteries.

Power-On Note: This unit is designed to operate with a DC battery. The first time that you power on an μ-MCA system, the unit presents a large instantaneous capacitive load. This is not a problem for DC batteries that can deliver high peak instantaneous current but it can cause a μ-MCA to fail power-on diagnostics when using a power supply driven by AC power. If the unit fails to come Ready when using an AC driven power supply, we suggest unplugging the system and immediately re-applying power.

Install PixelWrench2 before connecting the camera to the computer. This program is needed to manage connections to the camera and to extract useful data from the sets of visible light and NIR images the camera captures.

With the camera control switches and a video display the camera settings, diagnostic results and image storage may be reviewed.

To view your pictures on a computer, you may remove the memory cards and install them in a separate card reader, or plug the camera into the USB interface on your computer.

When connecting the μ-MCA system to a computer's USB port, Windows will recognize the system as multiple USB Mass Storage Devices, one device per camera in the array. From PixelWrench2, you may open previews of the images on the memory card in the camera, and extract them for viewing and analysis. Consult the PixelWrench2 Help Menu for more detailed instructions.

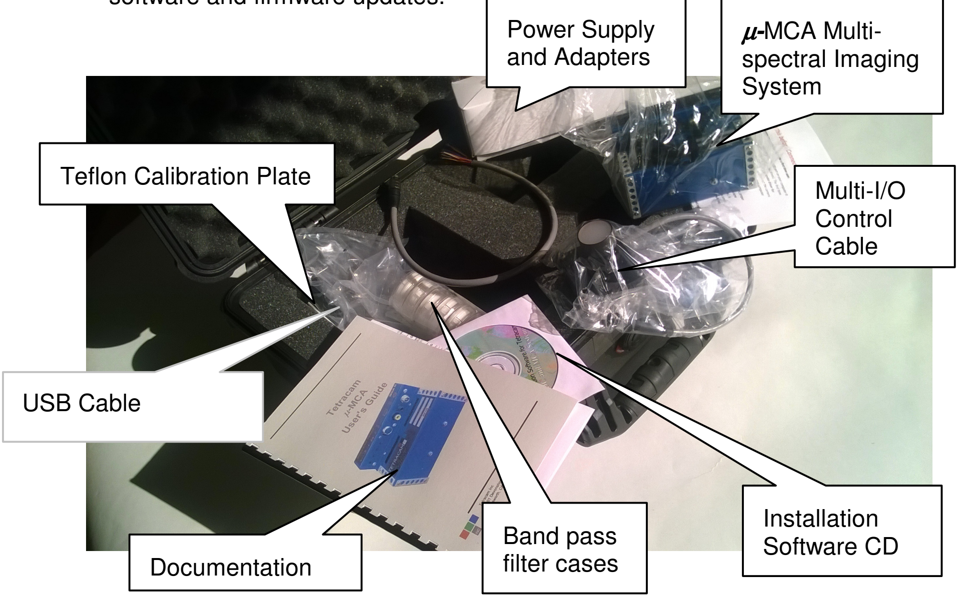
When you are done with the camera, turn it off by disconnecting the power.

Unpacking The Box

Your new μ-MCA multi-spectral imaging system is packaged in a rugged plastic carrying case. This is what you should find in the case.

- A μ-MCA Digital Still Camera
- A CDROM with the installation software supporting the camera
- Product and Accessory Documentation
- A USB interconnection cable
- A Control Cable
- A Secure Data Memory Card for each camera in the array
- An AC Power Adapter and Power input cable
- A White Teflon Calibration Plate
- A six slot memory card USB reader (Optional)
- There may optionally also be an ILS sensor and GPS receiver

Your μ-MCA comes with a one-year warranty against defects. You should send in the warranty card to register the camera and qualify for additional software and firmware updates.



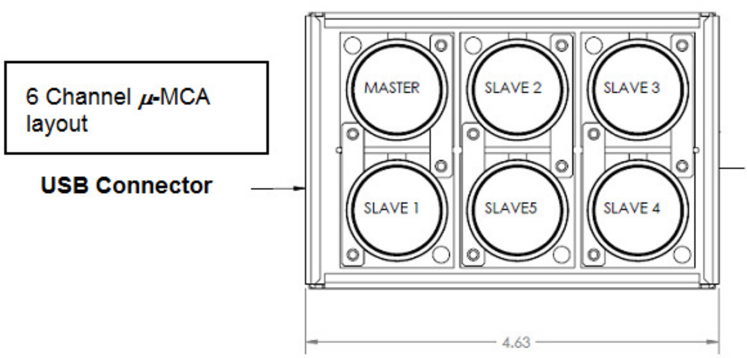
Getting Started

Becoming Familiar with your μ-MCA System

The μ-MCA multi-spectral imaging system consists of a set of digital still cameras compactly packaged and synchronized so they can all take pictures at the same time with only minor registration errors between images. These registration errors are corrected during factory alignment of the system. The alignment file software is supplied on the CD included with the unit.

One of the cameras in the system is set up as the MASTER camera. The master camera is responsible for synchronizing the other cameras (SLAVES), calculating exposure requirements, and logging GPS geo-referencing information.

Control connectors are provided to allow the camera to be remotely triggered, and to provide connection to an external GPS receiver. There is also an output video signal that can be used to monitor the framing of the image. Monitoring can take place remotely, using a commonly available video transmitter for RC aircraft, or locally, in a manned aircraft, using a video monitor. Each camera has its own file system for saving images. In a six-camera system there will be six Micro SD memory cards installed in the unit. Each will have a unique volume ID and naming convention for images that allows the sets of pictures to be separated on the host PC. The software supplied with the camera is able to combine the images into multi-channel TIFF image file for convenient extraction of the data at any time.



Each camera has its own band pass filter – typically a spectroscope filter commercially available from companies like Andover or Sigma. A wide variety of pass filters are available. Bandwidths of 10 nanometers and above will supply acceptable images. Narrower band pass filters (less than 10 nm) can produce some artifacts in the images. These must be corrected on the host PC using PixelWrench2 software supplied with the system. Band pass filters may be anywhere in the range of ~425 nm (blue) to ~1000 nm (near infrared). The spectral response graph later in this document shows the working range of the sensors. A set of four, six or twelve standard filters is included with the unit. Alternate custom filters are available upon request with a quotation.

Images can be transferred to a host PC by removing and reading the memory cards, or by plugging in a USB cable to the USB Hub controller built into the unit. In a six-channel unit, six separate devices will appear when the unit is connected.

An optional incident light sensor (ILS) module is available that includes an additional camera and matching filters for the set in the MCA. The ILS connects to the main assembly by way of a serial port, and has its own memory card that contains the calibration pictures associated with each image taken by the master and slave cameras.

Hardware Overview

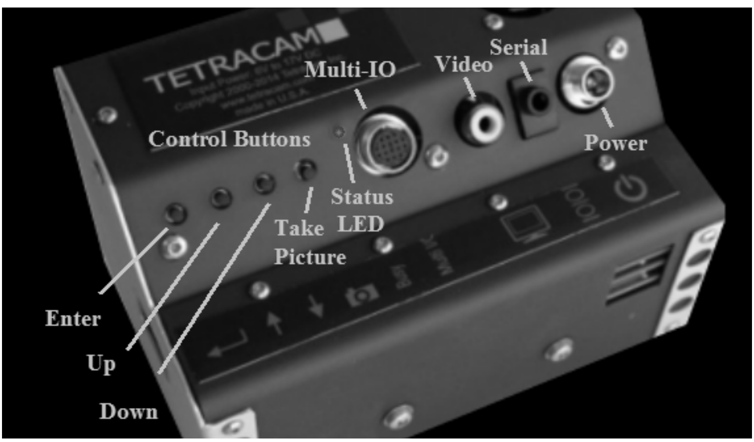
System Control Panel

The μ-MCA system hardware is housed within a rugged sheet metal assembly. The μ-MCA is designed for use in aerial manned and unmanned aerial photographic missions. The unit weighs 20% less than the prior model mini MCA and its interface is specifically designed to accommodate aerial missions. In the air, system operation is generally performed by remote commands but the unit contains System Control Buttons for testing and configuration on the ground.

The System Control Buttons are readily accessible on a raised plate located opposite the system's down-facing cameras. From right to left, the first button enables the user to manually take a picture. The next three buttons to the left enable users to move up or down through menu selections or to select an entry in the System Menu. The System Menu may be viewed by monitoring the video signal on the unit's Video Out port (see below).

To the right of the System Control Buttons, the ***μ***-MCA contains discrete connectors and a multifunction control connector. All of the signals on the discrete connectors, including power, are available on the multifunction connector. In the discussion of interconnection signals that follows, the signals may be found on the multifunction connector, and also on the separate connectors.

The Status LED indicates whether the system is ready or not ready to take a new picture. When the Status LED is green, the system is ready to take a new picture. When the Status LED is red, the camera is busy capturing and saving images. After saving images, this LED returns to green indicating the camera is ready to capture a new picture.



Power – The right-most of the system connectors is the ***μ***-MCA power supply input. The ***μ***-MCA contains a switching power supply so it is able to handle a variety of voltage input levels. The unit runs nominally on +12 VDC external power but can handle input voltages between +9.0 VDC and +16.5 VDC. The power connectors are center positive. The total current drawn by the unit at any moment depends upon its model number and level of system activity (i.e., whether the system is capturing and saving an image or not). The ***μ***-MCA4 typically consumes 580 ma. The *μ*-MCA6 typically consumes 920 ma. The ***μ***-MCA12 consumes ~1800 ma. See Power On Note on Page 3.

Serial Port – To the immediate left of the Power input is the RS232 Serial port. This is used for connecting an optional GPS receiver or for sending serial commands to the camera from auxiliary devices such as a Bluetooth transmitter/receiver. The connector is a 3.5mm stereo phone with plug tip, ring and sleeve carrying camera receive, camera transmit, and ground, respectively.

Video Out – To the left of the Serial Port is the video signal output. Video on this port is configurable. It may be run in NTSC or PAL format (configurable via the System Menu). The Video Out Port is able to act as the Master camera's viewfinder, displaying the image in the view by the Master camera, or it is able to display the System Menu.

Multi-function I/O – This 16-pin circular connector carries signals to enable viewing video, navigating the menu, taking pictures, and connecting GPS

USB Connector and Image Memory Card Access – Access to the USB 2.0 data connector and micro SD image memory cards is provided on the left and right sides of the system. The USB 2.0 data connector is located on the right side of the unit. This connector provides the data pathway that is used to transfer images between the system and a Windows-based host computer following completion of imaging missions. Visible on this side of the unit as well is a cut-out that provides physical access to one half of the system's micro SD image memory cards.

Each micro SD card stores images captured from one camera in the array. The system's Master micro SD card and the micro SD card for Slave #1 are located on upper-most circuit card assembly. Also accessible here are circuit card assemblies below the Master. These hold the balance of odd-numbered slave micro SD Cards. These are numbered from top to bottom: Slave # 3 (for all systems); Slave # 5 (for the pictured μ-MCA6 and μ-MCA12) and Slave #'s 7, 9 and 11 (for the μ-MCA12). On the left side of the unit (not pictured), access is provided to the balance of the system's even-numbered slave micro SD Cards. These are numbered from top to bottom; Slave # 2 (for all systems); Slave #' 4 (for the μ-MCA6 and μ-MCA12), and Slave #'s 6, 8 and 10 (for the μ-MCA12).

CAUTION: THE μ-MCA MUST BE IN A READY STATE PRIOR TO CONNECTING A USB CABLE. DO NOT CYCLE POWER WHEN THE USB CABLE IS CONNECTED.

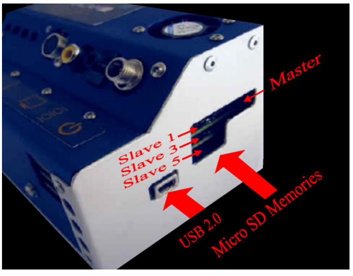


Image Memory

The μ-MCA employs one 16 GB micro SD Card for the images captured by each of its cameras. Each card in the array must have the same storage capacity. Micro SD cards with other values may be substituted if desired. The image memories store images in one of four native image file formats as they are acquired during imaging missions. These are: DCM10, RAW10 and RAW8 formats for all systems plus RWS10 for μ-MCA Snap systems. When images are transferred from each sensor to each image memory, the system's Status LED turns from green to red signaling that the system is busy and unable to capture additional images. The image memory also stores GPS coordinates streamed to the system from an optional external GPS receiver. Upon completion of each mission, the content of each image memory is transferred to the host computer by the user for processing by PixelWrench2. We recommend that the unit's power be turned off when any micro SD memory card is removed from the system.

μ-MCA Optical Channels (Lenses, Filters and Sensors)

Each μ-MCA optical channel consists of a 9.6 mm lens, a narrow band optical filter and a CMOS optical sensor. The lens focuses visible or near-infrared radiation through the filter and on to a 1.3 mega-pixel CMOS sensor. The filter limits the wavelengths that are able to pass through it to a specific band of electromagnetic radiation. The CMOS sensor accumulates electrons proportional to the amount of radiation impinging upon the sensor at each of its 1280 X 1024 pixel locations. Each well is cleared of its electron contents immediately prior to image exposure. This ensures that each image has high spectral fidelity and low image noise.

Like the mini MCA, the μ-MCA uses small miniature lenses with fixed apertures and focal lengths. Each channel of the μ-MCA is equipped with the same type of 9.6 mm focal length lens so the images can be merged with a minimum of error and distortion.

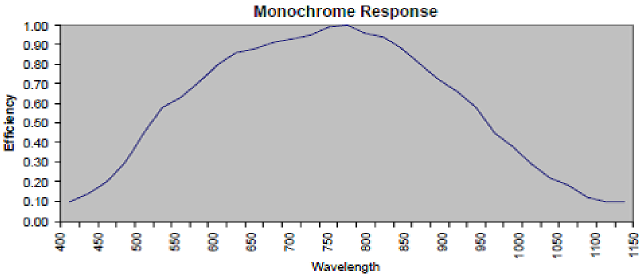
Each channel has a receptacle to hold the optical filter in place immediately below the lens. Positioning the filters below the lens limits the angle that radiation enters the channel and inhibits image vignetting,

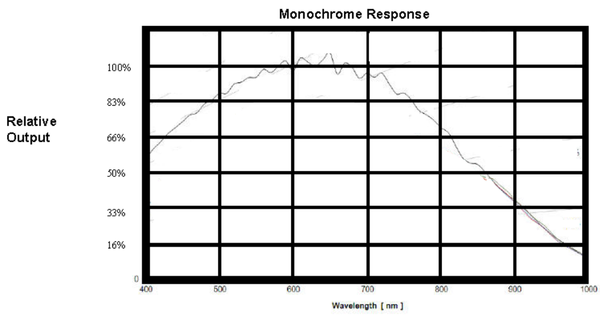
Filters may be obtained from Tetracam or a commercial supplier such as Andover. Tetracam supplies a standard set of filters with each system. These are selected for their relevance to remote sensing of vegetation. The customer may specify alternate or additional filters with their order. These may change the overall price of the system. Consult Tetracam's web site for the current list of standard filters for each system.

CMOS Sensor Relative Sensitivity

The μ-MCA filters are customer designated at the time of order. When they are installed at the factory, the filter that produces the greatest number of electrons in response to the radiation that it passes is selected as the Master filter. Each slave channel's relative exposure time is referenced as a multiple of the Master exposure time.

The graphs below show the relative efficiency of the μ-MCA rolling shutter sensor (above) and Snap sensor (below) for different bands of radiation.





Users may replace filters in the field with alternative alternate filters in order to re-task the system with the following limitations. A replacement filter should only replace a filter that is near the center wavelength of the filter originally installed in the system in that channel position. In the visible spectrum, replacing a filter with a new one that is within 025±50 nm of the original is acceptable. Beyond 700 nm replacement filters should be within ±25 nm of the original.

Replacement of filters beyond this range generally requires factory re-calibration of the unit. Contact Tetracam Technical Support for guidelines on customer re-calibration of the system or to obtain a Return Material Authorization number to return the unit to Tetracam for re-calibration.

Software Overview

PixelWrench2

The CD included with your system contains PixelWrench2 as well as other utilities, files and documentation that are helpful for setting up and operating your system. PixelWrench2 (PW2) is a powerful image editing program that is included with each μ-MCA system on This software runs on any IBM-compatible personal computer running Windows 8, 7, Vista or XP operating system (see PC System Requirements for additional details). The licensed software includes tools specific to multi-spectral images.

PixelWrench2 oversees transfer of images from the μ-MCA system's image memories to the host computer. It provides tools for managing μ-MCA image sets, complex batch processing tools, extraction of several vegetation index and canopy segmentation images as well as a comprehensive suite of image editing tools. PixelWrench2 translates Tetracam's proprietary DCM10, RAW10 and RAW8 image files into standard image file types (BMP, JPEG, TIFF, PNG etc.)

PixelWrench2 includes various tools that support image processing operations. TiffExport, for example, is a tool that converts multiple TIFF images into a single page multi-layered TIFF file of any bit depth (usable by GIS applications). Mousing over any pixel in the image enables users to view a graph that shows the luminance level from each band at that pixel location. This enables users to differentiate plant species or recognize other plant, soil or chemical conditions that are, in each case, able to be identified by their unique spectral signature.

The GPS Log Distiller in PW2 provides controls for setting all important camera operating parameters and creating and editing camera alarm files. Distiller provides a multi-threaded file transfer function for moving image and log files from the camera(s) to hard disk. Distiller also refines the camera-produced log files into compact *.GPS files and KML files useable with Google Earth.

PixelWrench2 also includes a Field of View Optical Calculator. This tool enables users to input camera parameters along with the distance from the camera to the subject in order to calculate variables including the observable field of view (FOV) in meters and the image resolution in millimeters per pixel. Additional variables including the required speed to fly to obtain a desired image overlap may also be calculated using this tool.

PC System Requirements

Any IBM-compatible personal computer with a free USB serial jack can be used to retrieve pictures from the μ-MCA system. The unit produces sets of images synchronized for simultaneous capture. The images from the set can be displayed three channels at a time using RGB format for false color rendition. Each RGB rendering is about 3.9 Megabytes. You should select a computer with resources that can support manipulation of images that are this large. Our recommendations for a minimum configuration are:

1 GHz or better processor, Intel or AMD

Windows 8, 7, Vista or XP operating system

512 megabytes of SDRAM

24 bit color graphics adapter at 1024 x 768 or better resolution

1024 x 768 or higher display

Large hard disk drive with 10 GB or more free space

Software Installation

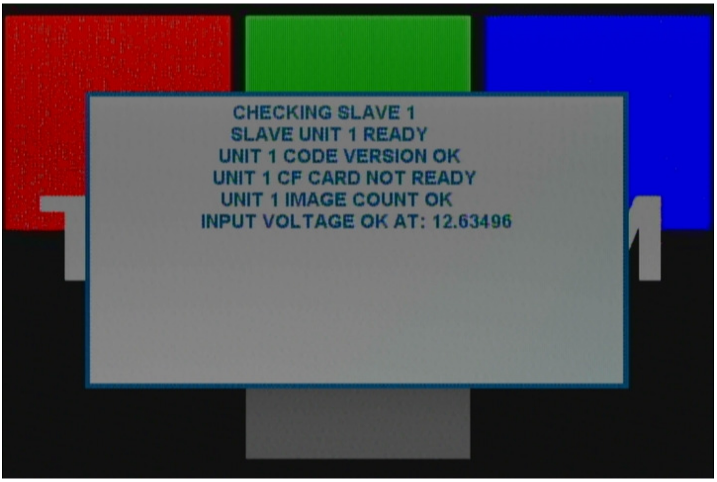
In order to install PW2, download the latest version of the software applicable to your computer (either the PW2 x32 Update file or the PW2 x64 Update file. Run the file Setup.exe located in the PW2 folder. The root folder of the CD also contains the image alignment file for your particular camera. Its name is xxxxxxglobal.MCA where xxxxxx represents your camera's serial number.

If your computer does not have the Microsoft .NET 3.5 framework installed, the PixelWrench2 installer will try to open Microsoft.com and download a file called dotNetfx.exe. This is the installer for .NET 3.5. This file is also on the CD in the root directory where you can run it directly prior to installing PixelWrench2.

Configuring the μ-MCA System for Operation

The Camera Viewfinder

Once the camera is powered on, it enters a diagnostic mode in which the slave and master cameras in the system are checked for consistency and operability. Included in the consistency checks are image count, memory card status and firmware revision. The results of each test cycle are displayed on the video output as they occur



Camera Control

There are buttons built into the camera to be used to setup and manually control the camera. Separate connectors are provided for Video out and RS232 communications. A trigger button is provided for manually taking pictures, or starting and stopping the continuous capture process.

Controlling the Viewfinder

In daylight conditions, a properly set exposure will produce a good viewfinder image so that NIR data is not over-exposed. Exposure can be set automatically by the camera, or can be entered as a fixed value. Once the viewfinder image is satisfactory, capture can be initiated with the TAKE PIC button on the controller. The Status LED will turn from green to red, indicating that the camera is busy capturing and saving the image. After a capture, the camera completes the compression and storage of the image and the status LED returns to green indicating the camera is ready.

More about the Viewfinder

The live viewfinder screen is the normal ready-state of the camera. From this state, the user may enter the menu system using the MENU / SELECT button, or capture images using the TAKE PIC button.

With an external video monitor connected to the camera's video out port, a live video image of what the camera "sees" will be shown on the screen.

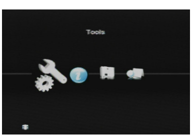
GPS Position – With a GPS receiver connected and communicating with the camera, position coordinates will be displayed. If the GPS receiver loses its signal lock, "WAITING FOR GPS" will be displayed in RED until it recovers its signal lock.

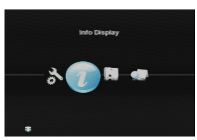
The live view screen is active until a still image is captured. Live view resumes when the image has been completely stored. If there is little or no time between pictures in continuous capture mode, the unit will not display video between captures. In either RAW or DCM format, the picture interval needs to be at least 4-6 seconds if a useful amount of video display between shots is required.

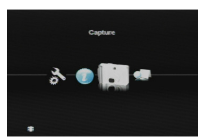
Using the Graphical User Interface

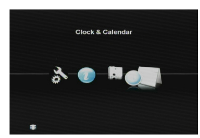
The camera configuration values can also be managed from the graphical user interface. The interface is accessed from the four buttons on the camera with feedback from the video display.

Navigation begins when the enter key is pressed. The first screen shows icons for the setup screens that are available to the user. The icons start small (1/2 size) and get larger when one of them is selected with up and down arrow keys. In the illustration below, the tools setup icon has been chosen. This screen is used to delete images and format memory cards. These settings are preserved through power cycles. Pressing enter activates the tools screen.





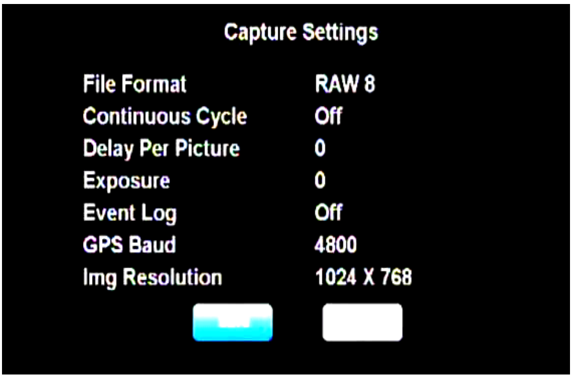




Inside the text screens, the enter button is used to move from field to field, and the up down arrow are used to change the value of a field. The picture taking switch is used to complete the operation. The only time that the changes made are saved is when the picture taking switch (Exit) is used and the SAVE button is selected. When the Exit switch is pressed while viewing the icon screen, live view is resumed. The tool screen consists of four action buttons, each labeled with the appropriate action:

The Tools Icon provides an entry point for deleting images in the camera and restoring factory defaults. The selection dialog is presented on the right:

In the screen below, the capture icon has been selected, which brings up the Capture setup text screen. The values controlled here determine how the camera will take pictures on its mission.



File Format selects if picture data will be compressed, or saved as it comes out of the sensor (RAW) for speed.

Continuous capture mode causes the camera to continue to take pictures after a single button press, with a delay between them, as controlled by "Delay Per Picture"

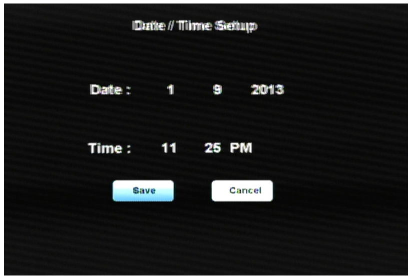
Exposure allows the user to select a fixed exposure time for the mission, so that the brightness of different shots will not change as the subject changes. If a fixed exposure is not set, the camera's automatic exposure feature will calculate an optimum exposure for each shot.

The Event Log, discussed in an earlier section can be turned on and off in this setup screen.

The GPS baud can be one of 6 values: 4800, 9600, 19200, 38400, and 115200.

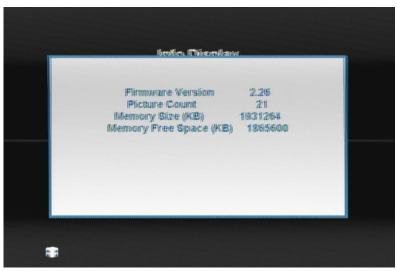
The camera can capture still images at reduced resolution to speed up the camera cycle time. Choose 2048 X 1536 for full resolution shots and 2+ second cycle time. Choose 1024 X 768 for half resolution and .8 second cycle time.

The camera has a real time clock which can be set from the user screen. Normally, it is better (more accurate) to allow the camera to set the clock from GPS messages, but when no GPS is available, or the GPS cannot be configured to emit the GGZDA string, manual setting can be carried out as shown in the example. As with all setup screens, enter navigates between fields, and up/down arrow changes the value. Pressing the shutter switch (exit) when the save button is selected makes the changes permanent.



The clock settings are standard except for the UTC offset field. Here the user enters the difference in hours between local time and UTC (Greenwich, U.K.). The default UTC value is pacific daylight time (Chatsworth, CA), or -8.

Selecting the Info Icon results in a display of camera status: Available memory, picture inventory, and camera firmware revision.



Choosing an Image Format

The μ-MCA and μ-MCA Snap systems can save images in a variety of native image file formats that are selectable via the system menu. Both systems can save images in 10-Bit RAW file format, 8-Bit RAW file format or DCM file format.

For standard μ-MCA systems, the10-Bit RAW file format provides the highest rate of capture, at about one picture per second. The speed depends in part on the features of the micro SD card. For users who want to use less storage, the 8 bit RAW format is the next fastest.

The RAW files are quite large – 6 megabytes for the10 bit-format and 3 megabytes for the 8-bit format. Compression (DCM format) cuts the size of the files in half, but takes longer to capture. Besides the smaller file size, another advantage of the DCM format is that the files contain previews that speed up the image access speed using PW2.

μ-MCA Snap system users can choose to save files in one additional native format, 10-Bit RWS format. The reason for this is dictated by requirements of the Snap sensor itself. The order of the pixels shifted out of the Snap Sensor is scrambled. De-scrambling pixels so that they are arranged in standard RAW or DCM formats lengthens the time required to save images in those formats. No such de-scrambling takes place during saving of RWS images. For this reason, this format allows sub-second save operations for μ-MCA Snap systems. De-scambling of RWS images takes place during post-processing of the images using PixelWrench2. Once processed, all native image file formats may be translated via PixelWrench2 into other common image file types such as BMP, JPEG, TIFF, PNG, etc.

Configuring Continuous Capture Mode

The simplest way to map large areas with the μ-MCA is to place the camera in Continuous Capture mode with a delay that will ensure adequate overlap of adjacent images. The recommended image overlap is at least 75% frontal overlap (along the line of flight) and at least 60% side overlap (between adjacent flying tracks). You can use the FOV calculator in PixelWrench2 to determine the altitude above ground level (AGL) that you will need to fly your plane in order to attain the desired resolution and the best delay to set up between successive shots given your aircraft's airspeed.

Say you wanted to know how much detail your image would show when flying a UAV at 400 feet above the ground. The FOV calculator tells you that the field of view that each camera sees at 400' AGL is an area 75 meters wide (246 feet wide) by 67 meters high (220 feet high) or, in other words, an area approximately equal to three rows of six standard tennis courts 23.78 meters (78.0 feet) long by10.97 meters (36.0 feet) wide. It also tells you that camera's resolution at that height is 66 mm (2.5 inches) per pixel or about the size of a standard tennis ball. If you wanted a different resolution you could use the FOV calculator to determine a different height AGL to fly your aircraft.

The μ-MCA is able to capture one Raw 10 bit image every second. If the delay between pictures were set to one second then in order to obtain a 75% image overlap along the line of flight, the camera would have to be ready to take a new picture before it had traveled ¼ of the distance into the new picture. Since the cameras are generally set up to capture images perpendicular to the line of flight of the aircraft, the overlap would be along the vertical axis of the image. So, in the one second the aircraft would have to traverse no further than ¼ of the image height at 220 per second/4 (or 55 feet per second or 37.5 mph) in order to attain that overlap. If it flew faster, it would pass the 75% overlap point before one second elapsed. If the aircraft flew slower then the Continuous Capture Delay setting would need to be set to a longer delay between pictures to attain the 75% overlap.

GPS Option Installation and Use

Your μ-MCA will capture and append the most recent GPS data string to each image as it is taken. The following requirements apply; Your GPS receiver must be configured to output the standard NMEA RMC and/or GGA sentences. The default output protocol for NMEA sentences is 4800 baud, 8 data bits, 1 stop bit, no parity. Your receiver should allow you to configure it for RMC and/or GGA at 4800:8:1:N. If your GPS receiver can be configured for a higher Baud, you should take advantage of the feature, since it will make the GPS data more accurate since less time would be lost transferring the messages. The menu in the camera has an entry for the GPS Baud. There is also an advanced setup screen, accessible via Pixelwrench2 that save a higher baud rate.

The GGA sentence is emitted once per second and contains the following fields:

1. Time UTC
2. Latitude and Longitude
3. Fix quality
4. Number of satellites tracked
5. Horizontal dilution of position
6. Altitude in meters MS
7. Height above MSL

Attach the optional serial cable to the small serial connector (see the illustration in the Hardware Installation section of this manual). Attach the other end to the serial port of the receiver.

The most recent GPS sentence sent to the camera will be appended to the image data file. You can view the GPS data in the image using Pixelwrench2. The camera firmware also supports an event, or position, logging system that will exactly place the GPS locations at the time pictures are taken, with a resolution of 10 milliseconds.

The camera also has two features you can turn on in the setup called GPS HEARTBEAT and GPS DISPLAY, which will toggle an indicator on the viewfinder screen each time it receives a new GPS sentence and will show the last GPS position received. This is useful to see that the camera is properly receiving GPS data.

The GPS messages are saved in each image as it is taken, and in the camera log file. Bad GPS sentences are not included in the log file unless logging is set to “verbose” mode in the capture setup screen.

Event Log File

When LOG EVENTS is set to ON in the CAPTURE METHOD screen, The camera will maintain a file with a record of key events that can be used to accurately position the location at which the picture was taken. Generally, a GPS receiver is connected to the camera serial port that sends $GGA… and $RMC…position strings to the camera.

This feature is used most often with aerial photography, when the GPS point is directly below the camera, so that both the camera and the image are at the same coordinate.

When the option is enabled, the camera creates the file CURRENT.LOG on the memory card in root folder. If there is a pre-existing CURRENT.LOG file, the file is moved to the image folder (TTCMCA0 in the case of the ***μ***-MCA master camera) and renamed according to the image numbers that were captured while the camera was last in operation. Only the master camera channel logs GPS data and image events.

For Example: If images 31, 32, 33, and 34 were captured, there will be event log records for each of those captures in the file. The file is scanned image capture records, and the smallest and largest image numbers found are used to compose a file name. In this case, the file would be renamed to 00310034.LOG. The first four characters of the new file name are the lowest image capture record in the file; the second four letters are the highest image capture record in the file.

A typical Event Log file is shown on the next page. Each line shown is one record in the file, terminated by a newline character and NULL. The NULL characters are hidden, and additional line feeds are added for clarity in the illustration.

Log files are much easier to manage if the camera is set up to operate in the USB Disk mode described earlier. The Log files are not directly accessible from Pixelwrench2 using the stream interface. When the camera shows up as a folder window on the desktop, as it does in USB Disk mode, the moving and deleting files is trivial.

You can use the GPS Distiller tool to manage image and GPS log files. See the PW2 help file for more information on how PW2 uses the distilled log files to refine the embedded GPS data in each image.

The CLK record is added when the camera powers up. It shows the camera date and time.

Every Record has a “Ticks” Field that shows the cameras internal clock count of 10 millisecond ticks. The count 104 means that che CLK record was written 1.040 seconds after power on.

CLK 000000104 Date/Time: 10/22/2009 15:15:01

GPS 000002006 00217 $GPRMC,192254.00,A,2942.79012,N,08223.306

GPS 000002064 00217 $GPRMC,192254.00,A,2942.79012,N,08223.30667,W,000.0,000.0,221009,03.3,W,A*0B
$GPGGA,192255.00,2942.79047,N,08223.30663,W,1,04,2.58,00040,M,-031,M,,*5D

IMG 000003049 00218

GPS 000003102 00218 $GPRMC,192302.00,A,2942.79461,N,08223.30899,W,000.0,

GPS 000003280 00219 $GPRMC,192302.00,A,2942.79461,N,08223.30899,W,000.0,
$GPGGA,192303.00,2942.79517,N,08223.30922,W,1,04,2.58,00062,M,-031,M,,*55

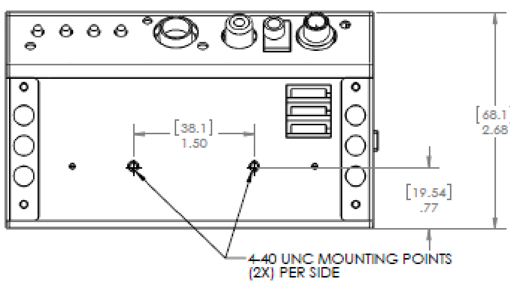
If a GPS is connected, an entry is made each time a GPS update string is received. Different GPS messages are concatenated as they come in.

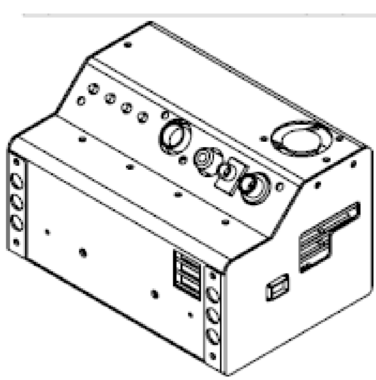
When a picture is taken, a record is written showing the system ticks at the end of integration. The camera can only do one thing at a time, so there will always be a system ticks offset between capturing a picture and the GPS messages. The actual position of the camera when the picture is captured can be approximated by interpolating between the two GPS messages using the system ticks.

Integrating the μ-MCA System with Aircraft

Mounting the unit

The μ-MCA contains four mounting holes in the front and rear of the unit (see drawing below). These should be used to mount the μ-MCA system into manned or unmanned aircraft. The system should be isolated from vibration transferred to it from aircraft. The μ-MCA does not contain its own power supply so power must be supplied from the aircraft by using the supplied Multi-I/O Control Cable. Connect +9.0 VDC and +16.5 VDC and Ground to Pin 15 u (see 16-Pin Multi-I/O Connector Pin Outs next page). In addition, the camera metal chassis should be grounded by connecting Ground to one of the mounting screws. If the camera is mounted on non-conducting vibration isolators then a ground strap should be provided. Dress and restrain all interconnect cables to prevent snagging or undue disturbance from propeller blast, etc. The μ-MCA housing and optics are not weatherproof. If the camera is mounted externally, weather protection should be provided.





16-Pin Multi I/O Conne ctor Pin Outs

The following describes the 15-pin functions of the Multi I/O connector:

1 –Power
2, 3, 4 & 5 – MENU/SELECT, UP, DOWN, and TAKE PIC buttons, respectively. Momentarily short to ground for button activation.
6 – power switch / external event trigger
7 – RS-232 Transmit (GPS)
8 – RS-232 Receive (GPS)
9 – Red LED: logic high when camera is busy
10 – Green LED: logic high when camera is on / idle
11 – (NC)
12, 13 – NTSC video signal and ground, respectively
14 – 3.3 V (logic high)
15 – Ground
16 – (NC)

Triggering the μ-MCA at Specific GPS Coordinates

Images may be captured by the μ-MCA system at specific locations using one of two methods. If the system contains an autopilot that has a programmable port, then program the aircraft's autopilot with the waypoints where you want the system to capture images. Program the autopilot's port to momentarily output a low logic level (ground) as it reaches each waypoint. Connect the autopilot's port to Pin 5 (TAKE PIC) on the Multi-I/O Control Cable. At each waypoint, the grounded TAKE PIC signal line will trigger the system to capture images.

In order to record the GPS locations where images are captured connect an external GPS system such as Tetracam's 'plug and play' FirePoint 100 to the μ-MCA by using the Multi-I/O Control Cable. Connect the GPS system's RS232 Transmit signal line to the RS232 Receive Pin (Pin 8). Connect the GPS system's RS232 Receive signal line to the RS232 Transmit Pin (Pin 7). Consult the previous section of GPS Option Installation regarding GPS configuration.

The μ-MCA system may alternatively be triggered at specific locations by sending an <ESC> T command to the system via Pins 7 and 8 (RS232 Transmit and Receive) using the Multi-I/O Control Cable (see Tetracam RS232 Serial Control Commands ahead). This is the method that is employed Tetracam's Wireless Camera Interface. In this, an Android tablet or cell phone is inserted into the aircraft. The Android uses Tetracam's free S-Link App to calculate waypoints and the Android's own Bluetooth® interface to trigger the camera at each waypoint.

Capturing and Processing μ-MCA Images

The Calibration Image

In order for the μ-MCA system to capture accurate multi-spectral images it is necessary to obtain a reference calibration image prior to or during each imaging mission. Calibration consists of taking an image of the Teflon calibration tag under the same lighting conditions as the images under study. This image is used to teach the application software what the spectral balance of that day's sunlight is. The ratio of red/NIR or green/NIR is then applied as an offset to the calculation of the various vegetation indices. Note: if a calibration image is not taken within an hour or two of pictures in the field, the vegetation index calculations will not be very accurate, and the pictures may not be useful.

Place the calibration target on the ground, or hold it level with the ground, and photograph it. It need not fill the entire frame and it must not be overexposed.

Make sure to avoid a direct reflection of the sun. The sample image on the left is of a properly exposed calibration target. When the pictures are imported to the host computer, the calibration image will be used to refine the vegetation index calculations.

In place of the Teflon Calibration Tag supplied with each system, you may also use Multi-spectral Ground Calibration Targets that provide a relatively flat response to incident radiation throughout the VIS-NIR wavelengths. These have that added advantage of acting as Ground Control Points for geo-referencing images.

Flying Aerial Imaging Missions

Typically, a user's initial concern at the start of an aerial imaging mission is the ground resolution they will need to achieve in order to identify the detail they require in their images. Use the FOV Calculator in PixelWrench2 to determine the above ground level altitude you need to fly in order to obtain that resolution. You can also use this calculator to determine the airspeed you will need to fly in order to obtain sufficient frontal and side image overlap.

Aerial missions involve making successive passes back and forth across an area of interest usually in an east-west direction. This flight line ensures consistency in orientation and sun angle. The orientation of images is generally at 90° to the flight lines. Images are generally captured at a nadir angle with the μ-MCA system pointing straight-down. Using the nadir angle minimizes occlusion of subject matter by hills or other intervening objects. In order to permit splicing multiple images together, vendors of mosaicking software recommend a high overlap percentage for adjacent images. Pix4D, for example, recommends a minimum of 75% forward overlap and 60% side overlap.

Images are captured on 16 GB micro SD memory cards. One card is used for each of the system's optical channels. Tetracam recommends that the memory cards be formatted on a PC prior to a camera mission. They should be formatted as FAT32 cards with a 32KB cluster size for high performance Care must be taken to maintain the label on the volume: TTCDISK[1-8], as it comes from the factory, during formatting on the PC. The volume information is used by PixelWrench2 to identify the device as a Tetracam camera, and identify the channels when it appears as a USB Disk. If desired, move lower resolution video images out of the system for display on a cabin flight monitor in manned aircraft or, with a supplementary video transmitter, you can send video down to a ground control station from an unmanned aircraft.

Managing and Processing μ-MCA Images in PixelWrench2

Upon landing, images may be transferred from the μ-MCA system to the host computer. There are three ways to retrieve images from the μ-MCA:

1. Remove the cards from the system and copy their contents to a single folder on your computer. The native image files can then be opened directly in PixelWrench2.

2. Use the GPS Log Distiller in PixelWrench2. Run Transfer Images.

3. Connect the camera directly to a host computer via its USB 2.0 interface. The cameras will all show up, each as a separate USB disk device with a unique volume ID. Files can be dragged to and from the disks, and Pixelwrench2 can be used to open the files directly.

Tetracam supplies a Six Pack micro SD Card Reader accessory that allows up to 7 cards to be installed in a system via the USB 2.0 interface (see page 27).

Read the PixelWrench2 Help file for a detailed description of this software and its use with Tetracam multi-spectral imaging systems.

Options and Accessories

GPS Receiver

Tetracam supplies a pre-configured FirePoint 100 GPS receiver that provides the NEMA strings needed to reference captured images geographically. The unit comes with a cable wired correctly for the μ-MCA serial port, and has a separate 8V to 14V DC power connection. A 2mm plug and cable is shipped with the GPS unit to simply installation of the unit. The GPS and camera ship from the factory configured for transferring data at a 4800 Baud rate.

Electronic Incident Light Sensor (e-ILS)

The μ-MCA e-ILS is an up-looking imaging sensor consisting of a standard 16 GB micro SD image memory, a fast USB 2.0 interface, a single Mini-MCA 1280 x 1024 CMOS sensor and 8.0 mm filters, each with the same center wavelength and bandwidth values as those present in the down-looking μ-MCA. The filters are contained beneath a translucent lens that diffuses incoming radiation. The e-ILS is placed in a position that affords a clear un-obscured view of the sky. It is attached to the μ-MCA by means of an electrical umbilical cable that is plugged into to a circular connector at the rear of the μ-MCA. The umbilical carries power, ground and a serial trigger signal between the e-ILS and the μ-MCA system. Extension cables are available for mounting the e-ILS up to seven meters away from the μ-MCA system.

The e-ILS gathers down-welling radiation at the same wavelengths as the up-welling reflected radiation captured by the μ-MCA. The e-ILS saves a calibration image of the sky at the same time that the μ-MCA master camera signals all of the other cameras in the array to capture images. Once the mission is completed, the calibration images can be extracted from the e-ILS over the device's USB 2.0 connection or by physically removing the 16 GB micro SD image memory from the e-ILS camera. When processed by Pixelwrench2, every pixel in every image in the array will have two values to process; the reflected value from the objects in the scene and the e-ILS calibration value. Comparing the two enables the software to represent each pixel by the exact fraction of radiation that it reflects back to the camera.

The ratio of upwelling reflected radiation to down-welling incident light for each pixel in the image is expressed as some fraction of either 255 for an 8-bit image or 1023 for a 10-bit image. The resulting image accurately represents the reflectance value of the radiation at each pixel in the image.

Six Pack micro SD Card Reader

The Six Pack micro SD Card Reader is provided to facilitate transferring micro SD memory cards from a μ-MCA after a mission is complete (and thereby, freeing the μ-MCA for further remote sensing operations). The reader enables simultaneously linking six image memory cards to a host computer for processing by PixelWrench2.

When the Six Pack is populated with a set of cards from a mission, it looks exactly like the camera array to PixelWrench2, and all camera processing operations can proceed as if the host computer were connected to a μ-MCA.

Each Six Pack is provided with a two-amp 24 watt power supply. This delivers sufficient power to connect up to six micro SD memory cards simultaneously plus a seventh micro SD card that may be plugged into a Daisy-Chain socket on the side of the unit.

The Six Pack contains an extension socket for “Daisy Chaining” another Six Pack. This would be necessary when using two Six-Pack units with a μ-MCA12. The Daisy-chain Socket is also used if a μ-MCA has an ILS or e-ILS. The memory card from the ILS should be installed in the daisy chain connector using the single card SD reader that is included with the ILS unit.

Programmers Reference – File Formats

The μ-MCA system uses proprietary formats for lossless data storage. DCM files are compressed using differential encoding and Huffman compression. RAW files are the array of captured pixel values with header and trailer information. The exact format of the file in 8 and 10 bit form is shown below.

10 Bit Raw File Format

The RAW file format contains both Header and trailer information. For values greater than 255, two bytes are used in little endian (Intel) configuration for header, trailer and pixel values.

Byte 0-3 Size of raw image in bytes – 32 bit value
Byte 4 Bits per pixel – 10 for this format
Byte 5 Format tag – 16 for RAW files
Bytes 6-7 Pixel Columns – 16 bit value. This is pixels not bytes
Bytes 8-9 Pixel Rows – 16 bit values
Bytes 10-(image size + 10) PIXEL DATA – 16 bit values
Bytes (image size + 10)-(EOF - 28) GPS data. $GGA and $RMC strings
Last 28 Bytes – ASCII exposure string
formatted: "EXPOSURE:%08ld uSeconds\n"

8 Bit Raw File Format

Byte 0-3 Size of raw image in bytes – 32 bit value
Byte 4 Bits per pixel – 8 for this format
Byte 5 Format tag – 16 for RAW files
Bytes 6-7 Pixel Columns – 16 bit value. This is pixels not bytes
Bytes 8-9 Pixel Rows – 16 bit values
Bytes 10-(image size + 10) PIXEL DATA – 8 bit values
Bytes (image size + 10)-(EOF - 28) GPS data. $GGA and $RMC strings
Last 28 Bytes – ASCII exposure string
formatted: "EXPOSURE:%08ld uSeconds\n"

10 Bit DCM File Format

Byte 0-3 Size of image data, GPS data, and various tags in bytes – 32 bit value This value can be used to calculate a pointer to the JPG preview data
Byte 4 Bits per pixel – 12 for this format (2 padded)
Byte 5 Format tag – 16 for RAW files
Bytes 6-7 Pixel Columns – 16 bit value. This is pixels not bytes
Bytes 8-9 Pixel Rows – 16 bit values
Bytes 10-(data size + 10) DATA – 8 bit values
Bytes (data size+10)-EOF JPEG Preview image.

Looking backwards into the data encompassed by the size value in the header there are several fixed length fields, given below with their sizes.

GPS data – 1024 Bytes
Tags for temperature and clock ticks – 16 bytes

We do not recommend trying to process the DCM files with your own code. Contact Tetracam for assistance with sample ‘C’ source files if reading the DCM file data is absolutely necessary.

Programmer's Reference – C and Visual Basic Support

The interface to the camera is in the library SXGAMCA.DLL This library provides a number of useful camera interface functions. Developers, to incorporate the camera interface into their own programs and plug-ins, can use the interface functions embedded here. The file sxgaMCA.lib is provided in the installation directory to allow static linking to the DLL.

The "include" file loadext.h is available in the installation directory to be made part of any C or C++ program making use of the DLL. It is reprinted in part below. All requests are made by filling the PixRequest structure before the function is called. Sample source files are available from Tetracam to help with the creation of a custom application.

```
typedef struct _PXR
        {
        int requestType;          // ACTION type
        int workSilently;          // do not pop up status or hourglass the cursor
        int imageNumber;          // 0 = last image in camera or file
        char far *fileName;       // 0000:0000 = use camera - Otherwise the file to open
                                  // "" = ask userask user for file name
                                  // "xxxx" = use file xxxx.DCA for reading
        int imageBlue;             // Used for various arguments
        int imageGreen;           // Used for various arguments
        char far *statusString;   // copy camera/image status string to here
                                  // if not 0000:0000
} PixRequest;
```

In Visual Basic a wrapper function is provided which accepts the values passed in as individual variables. It then creates the required structure before calling ProgrammerPlug(). A sample calling sequence from Visual Basic is shown below the interface function definition:

```
TTCAM_API HANDLE VBProgrammerPlug
        (
        int  FAR *requestType,
        int  FAR *workSilently,
        int  FAR *imageNumber,
        char FAR *fileName,
        int  FAR *imageBlue,
        int  FAR *imageGreen,
        char FAR *statusString
        );
```

```
/* Here is what a call looks like made from Visual Basic into the DLL:
Declare Function VBProgrammerPlug% Lib "SXGAMCA"
                                        (
                                        requestType%,
                                        workSilently%,
                                        imageNumber%,
                                        fileName as Any,
                                        imageBlue%,
                                        imageGreen%,
                                        statusString as Any
                                        )

For integers, According to the VB manual for version 1.0 or thereabouts,
VB passes, by default, all arguments by reference, (or far pointers,
if your a 'C' programmer. ByVal overrides this by placing the contents
of the variable on the stack, rather than the pointer to the variable.

For strings, It appears the ByVal is the way to point to a string that is
to be modified by the DLL. The examples in the book for calling Windows
APIs that modify strings show a declaration as ByVal. See the chapter
headed "Calling DLL Routines with Specific Data Types" for details.

To pass a NULL pointer to VBProgrammerPlug, use ByVal 0& as the
parameter for fileName or statusString. To pass a pointer to a fixed length
string, use the syntax ByVal StringName$ in the argument list.
*/
The Visual Basic call ends up here after translation of the calling parameters into a PixRequest Structure
TTCAM_API HANDLE PASCAL ProgrammerPlug(PixRequest FAR *);
/*******  Multifunction DLL interface ********************************
```

IMPORTANT
The caller must always use the HOOKUP request before any other reqests are made!!!!

```
Passed a pointer to a request block this function will perform the
requested action, (see enumerated list, below) and return either the state
of the current hookup, a handle to a DIB image, or the camera or file
status string.  After processing any image controlled by the DIB handle
returned, you are responsible for freeing the memory controlled by the
DIB's handle BEFORE calling ProgrammerPlug for another image.
*/
enum {HOOKUP = 0,          // Hook up to the camera/file and prepare DLL to
                           // load images in following calls.
                           // if fileName = 0000:0000, use the camera
                           // if fileName = "xxxxx" or "xxxx.xxx", use file
                           // if fileName = "" or "*", prompt user for file name
                           // RETURNED PixRequest values:
                           //        requestType = COLOR or GRAYSCALE
                           //        depending on camera or file type
                           //    imageNumber set to # of images available
                           //
                           //        requestType and imageNumber are both set
                           //     to 0 if a file or camera I/O error occurred

                           // If a non-NULL pointer is found in statusString,
```

```
                     // the camera or file status string is copied. a ""
                     // is returned if there was an error.

  STAMP = 1,         // return a handle to the STAMP DIB
                     // RETURNED PixRequest values:
                     //   HANDLE to a D)evice I)ndependant B)itap, (DIB)
                     //   requestType = COLOR or GRAYSCALE depending on
                     //   what picture type the stamp represents
                     //   Returns a 0 on error
  GETIMAGE = 2, // standard gray scale image
                     // RETURNED PixRequest values:
                     //   HANDLE to a DIB
                     //   Returns a 0 on error

  CAPTUREBUFFER = 7,      // Returns the camera's image capture buffer
                          // as a DIB. Stretch, sharpen and scale
                          // are also done.
                          // CALL WITH:
                          // imageNumber = image type to return
                          // RETURNED PixRequest values:
                          //   HANDLE to a DIB
                          //   returns NULL on error


CAMERASTATUS = 8,    // Send the camera status string
                     // RETURNS:
                     //   Camera status string copied to statusString.
                     //   Returns "" on error

  IMAGESTATUS = 9,   // Send the imageNumber status string to
                     // RETURNS:
                     //   Image #imageNumber status copied to statuString
                     //   Returns "" on error

  SETEXPOSURE = 10, // Send the value in Blue (LSW) and Green (MSW)
                     // to camera as exposure time. 0=automatic,
  MULTISELECT = 16,  // Allow Operator to Selecting Multiple Images
                     // CALL WITH:
                     // imageNumber = Number of images pre-selected
                     // statusString= pointer to NULL terminated byte
                     //         array containing the ID numbers of
                     //         images to pre-select, in the order
                     //         desired.
                     //         ARRAY SPACE MUST BE AT LEAST
                     //         57 BYTES!!
                     // imageGreen  = Maximum number of images allowed
                     //         to be selected. If 0, the max
                     //                 is the number of images in the
                     //         file/camera.
                     // imageBlue   = TRUE=Show selection order number
                     //                 in stamp upper left corner.
                     //                 FALSE= No selection Number.
                     // RETURNED PixRequest values:
                     // imageNumber = Number of images selected, or
```

```
                                //          zero if none or error
                                // statusString= pointer to NULL terminated byte
                                //          array containing the ID numbers of
                                //          operator selected images in the
                                //          order selected. The array is
                                //          left untouched by errors.
        SNAPSHOT = 17,              // Take a picture
                                    // CALL WITH:
                                    //  Nothing
                                    // RETURNED PixRequest values:
                                    //      imageNumber = TRUE if connection made
                                    //              FALSE if comm I/O error

        FASTSHOT = 18,              // Take a fast snapshot, and return the DIB
                                    // CALL WITH:
                                    //  imageGreen  = Non-zero uses an on-screen Viewfinder


GETCOMPRESSEDDATA = 19,             // Return DIB HANDLE points to the
                                        //compressed JPEG or DPCM
                                        // Data from the file in the camera.

            ERASEIMAGES = 22,           // Erases all images in the camera
                                    // without prompting the user for confirmation.

        CLOSECAMERAPORT = 25,       // Shuts down communications thru any currently
                                    // active port
};
```

Tetracam RS232 Serial Control Commands

Camera serial port command strings consist of a lead-in character (ESC), a command character (A – Z, a - z), and a number of numeric arguments. The numeric arguments are strings of Hex ASCII digits either 4 or 8 characters long depending on the magnitude of the value (16 bit or 32 bit). Separators are not required between the argument values, or between the command character and an argument. Spaces can be used as separators if desired.

Below is a table of the command characters currently implemented, and a description of the responses to be expected from the camera. Arguments are shown as <ARG16> or <ARG32> depending on their magnitude (16 or 32 bits).

<ESC>**E**

Erase all Image files stored in the camera's file system.

<ESC>**T**

Take a picture and save the image to SD card memory.

<ESC>**X**<ARG16>

This command controls the camera exposure for the next image with the value in the argument. If the value is 0, the camera performs a light measurement operation, and calculates a reasonable exposure itself. Otherwise, the exposure is set to the number of milliseconds given in the argument.

Support Info

Your μ-MCA system comes with a one-year warranty against defects or hardware failures. For help with problems, contact

Technical Support:

Tel: 818-288-4489 (8 A.M. to 5 P.M. Pacific Standard Time)
Email: support@tetracam.com
Web: www.tetracam.com

Specifications

Specification	Description/Value	Remarks
Basic		
Part Numbers	*µ-MCA* 4: TTC-1063 *µ-MCA* 6: TTC1064 *µ-MCA 12*: TTC1065 *µ-MCA* 4 Snap: TTC-1073 *µ-MCA* 6 Snap: TTC1074 *µ-MCA* 12 Snap: TTC1075 e-ILS-4: TTC-1066 e-ILS-6: TTC-1067 e-ILS-12: TTC-1068	4 multi-spectral Channels 6 multi-spectral Channels 12 multi-spectral Channels 4 multi-spectral Snap Channels 6 multi-spectral Snap Channels 12 multi-spectral Snap Channels for 4 multi-spectral systems for 6 multi-spectral systems for 12 multi-spectral systems
Typical Application	Aerial remote sensing of vegetation	
System Hardware	Configurable Camera Array of 4. 6. or 1 channels Rugged sheet metal system housing Four System Control Buttons: Menu Ent Menu Forward, Menu Backward, Take P Multi-pin I/O connector for use with accessories or testing ADC/MCA controller box & 6' Multi-I/O c Replaceable filter for each lens Six slot memory card USB reader System CD with PixelWrench2 software USB Interconnection Cable 12 VDC external power supply Power Cord (un-terminated on one end) White Teflon Calibration Plate Rugged Pelican Carrying Case	
System Inputs	Power (see below) USB 2.0 data connection, RS232 Serial (NMEA GPS sentences),	
System Outputs	USB 2.0 data connection, Real time NTSC monochrome Video for viewfinder and menu operations, Remote Shutter (External Trigger)	
System Software	PixelWrench2 is included with each purchase of a µ-MCA	
Commonly Ordered Options Available from Tetracam	FirePoint™ GPS Navigation System *µ-MCA* e-ILS Six Pack micro SD Card Reader S-Link Android App and Wireless Camera Interface (MCA) Camera Calibration Service Ground Calibration Targets Photogrammatic mosaicking software	

Specifications (Continued)

Specification	Description/Value	Remarks
Physical		
Weight	*µ-MCA 4: 497 g (1.09 pounds)* *µ-MCA 6: 530 g (1.16 pounds)* *µ-MCA 12: 1160 g (2.56 pounds)*	
Dimensions (mm)	*µ-MCA 4: 115.6 x 80.3 x 68.1* *µ-MCA 6: 115.6 x 80.3 x 68.1* *µ-MCA12: 115.6 x 155 x 68.1*	
Environmental	*Temperature* *0 degrees Celsius to 40 degrees Celsius (32 degrees Fahrenheit to 104 degrees Fahrenheit)* *Humidity* *Less than 85% relative humidity, non-condensing*	Note: the camera will operate outside of the recommended environmental range, however performance may be degraded.
Power		
Voltage Input	+ 9 VDC to + 16.5 VDC Center Positive	
Current Draw at 12V:	*µ-MCA* 4: 520 ma typical *µ-MCA* 6: 820 ma typical *µ-MCA* 12: 1650 ma typical *µ-MCA* 4 Snap: 520 ma typical *µ-MCA* 6 Snap: 820 ma typical *µ-MCA* 12 Snap: 1650 ma typical	
Sensor		
Description	1.3 mega-pixel CMOS sensor (1280 X 1024 pixels)	µ-MCA systems = rolling shutter sensors µ-MCA Snap systems = global snap shutter sensors
Quantity per System	4, 6, or 12 per system	
Sensitivity	~450 nm to ~ 1000 nm	
Dimensions	6.66 mm x 5.32 mm	
Pixel Size	5.2 microns	

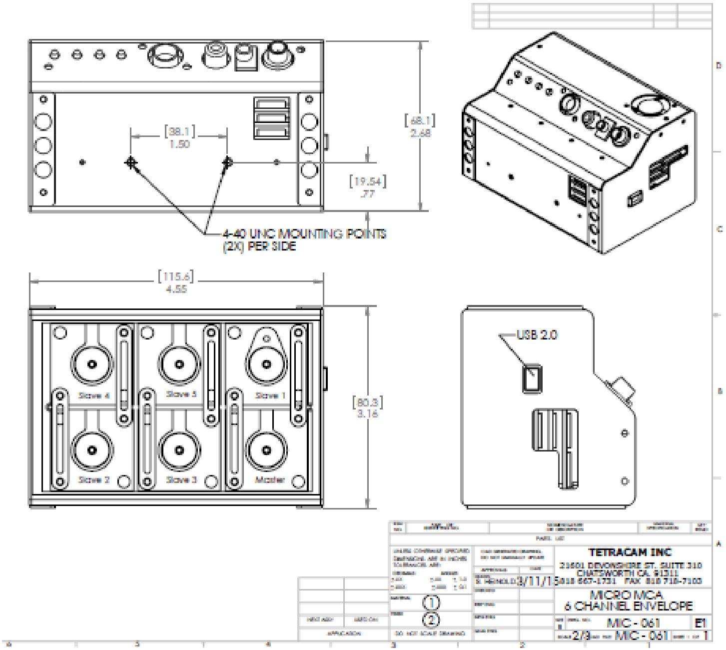
Specifications (Continued)

Specification	Description/Value	Remarks
Filters		
Filters Standard with µ-MCA 4:	- 490FS10-25 - 550FS10-25 - 680FS10-25 - 800FS10-25	Each system holds 4, 6, or 12 user-selectable field-changeable 25 mm narrowband optical filters.Typical bandwidth is 10 nm.
Filters Standard with µ-MCA 6:	- 490FS10-25 - 550FS10-25 - 680FS10-25 - 720FS10-25 - 800FS10-25 - 900FS20-25	Filters are identified by a part number. The first three digits in this identify the center wavelength in nm. The two digits following the "FS" indicate the bandwidth in nm. The two digits after the dash indicate the filter diameter in mm.
Filters Standard with µ-MCA 12:	- 490FS10-25 - 520FS10-25 - 550FS10-25 - 570FS10-25 - 671FS10-25 - 680FS10-25 - 700FS10-25 - 720FS10-25 - 800FS10-25 - 840FS10-25 - 900FS20-25 - 950FS40-25	µ-MCA filters are customer designated at the time of order. These are field-replaceable by the user with alternate 25mm optical filters in order to re-task the system with the following limitations. Since wavelength impacts focus, a replacement filter should only replace a filter that is near the wavelength of the filter originally installed in the system in that channel position. In the visible spectrum, replacing a filter with a new one that is within ±50 nm of the original is acceptable. Beyond 700 nm, replacement filters should be within ±25 nm of the original. Replacement of filters beyond this range requires factory re-focusing of the unit. Alternative band pass filters are available upon request with quotation.
Lenses		
Focal Length	9.6 mm fixed lens	
Aperture	f/3.2	

Specifications (Continued)

Specification	Description/Value	Remarks
Images		
Horizontal Angle of View	38.26 degrees	
Vertical Angle of View	30.97 degrees	
Default Depth of Field	~2 meters to infinity	
Image Triggering Mechanisms	Manual Shutter Release, Auto-Timer (Menu-settable), Remote Shutter Release (External Trigger), Computer Trigger via USB, or RS232 Serial Trigger	
Image Exposure time	Auto or menu-selectable in ms	
Default Image Dimensions	1280 x 1024 pixels (1.3 MPel)	µ-MCA image size may be adjusted to an alternate image size (1024 x 768) via menu selection
Default Image Storage	Images with metadata are stored on 16 GB Micro SD cards (1 per channel) in Tetracam native file formats	Approximately 12000 images may be stored on each 16 GB micro SD memory depending upon selected file type
Native Image File Formats **(in order of image speed)**	Micro-MCA 10-Bit .RAW file format 8-Bit .RAW file format 10-Bit .DCM Compressed file format Micro-MCA Snap 10-Bit .RWS file format 10-Bit .RAW file format 8-Bit .RAW file format 10-Bit .DCM Compressed file format	Native file formats may be converted via PixelWrench2 into common standard image file types such as BMP, JPEG, TIFF, PNG, etc. For Micro-MCA standard systems, the10-Bit .RAW file format is fastest with min image intervals ~ 1 per second (10-Bit .RWS file format is disallowed) For the Micro-MCA Snap systems, the10-Bit .RWS file format is fastest (with sub-second image intervals)
Image Capacity	(DCM10) 0.9MB per image (RAW10) 2.6MB per image (RAW8) 1.3MB per image (RWS10) 2.6MB per image	Total capacity varies as DCM capacity is variable since compressibility is content-dependent
Image Capture Rate (Single Shot) DCM10-Capture to end of cycle: RAW10-Capture to ready: RAW18-Capture to ready: RWS10-Capture to ready	 3.0 Seconds 1.0 Second 1.2 Seconds < 1.0 Second	
Maximum Image Data Transfer Rate	Via USB 2.0: 480 megabits per second (mbps) or 60 megabytes per second (MBps).	

μ-MCA 4 & μ-MCA 6 Dimensions



Index