Duo & Duo R
User Guide
**FLIR Duo User Guide**

**Table of Contents**

1 Introduction ................................................................................................................................................. 2

1.1 Scope ...................................................................................................................................................... 2

1.2 Revision History .................................................................................................................................... 2

2 Resources .................................................................................................................................................. 3

2.1 FLIR Website / Tech Support Information.............................................................................................. 3

2.2 FLIR Systems Documents (available on website) .................................................................................... 3

2.3 External Documents ............................................................................................................................... 3

2.4 Abbreviations / Acronyms ..................................................................................................................... 4

3 What’s in the box ....................................................................................................................................... 5

3.1 Unpacking Your Camera ....................................................................................................................... 5

3.1.1 FLIR Duo Camera ............................................................................................................................ 5

3.1.2 Bench Cable ...................................................................................................................................... 6

4 Connecting to the Camera .......................................................................................................................... 7

4.1 Mechanical Interface ............................................................................................................................ 7

4.1.1 Size / Weight .................................................................................................................................... 7

4.1.2 Mounting .......................................................................................................................................... 7

4.2 Electrical Interface ............................................................................................................................... 8

4.2.1 Bench Testing .................................................................................................................................. 8

4.2.2 Mini-USB Cables ............................................................................................................................ 8

4.3 Software Interface .................................................................................................................................. 10

4.4 Camera Operation ................................................................................................................................. 10

---

*Information on this page is subject to change without notice*
# 4.5 Camera Troubleshooting

- Troubleshooting Guide

# 5 The FLIR UAS App

5.1 Home Screen

- 5.1.1 MSX
  - 5.1.1.1 MSX Strength
  - 5.1.1.2 Alignment
  - 5.1.1.2 Recommended MSX Alignment procedure

- 5.1.2 IR Color Palette

- 5.1.3 Display Video Mode

- 5.1.4 Video/Still Image
  - 5.1.4.1 Video
  - 5.1.4.2 Still Images

- 5.1.5 Record

- 5.1.6 FFC (Recalibrate)

- 5.1.7 Settings
  - 5.1.7.1 Main
  - 5.1.7.2 Accy. Port
  - 5.1.7.3 Radiometry
  - 5.1.7.4 About

# 6 PWM, MAVLink, and TCP/IP Operation

6.1 PWM

- 6.1.1 Connecting to a PWM Compatible Flight Controller
- 6.1.2 Configuring PWM Connection
List of Figures

Figure 1. Duo with Camera Mount.................................................................................................................. 5
Figure 2. Bench Cable ...................................................................................................................................... 6
Figure 3. Core dimensions................................................................................................................................. 8
Figure 4: Mini-USB 10-pin Layout..................................................................................................................... 9
Figure 5. Camera Status and Record LED Description.................................................................................... 10
Figure 6: FLIR launch screen............................................................................................................................. 12
Figure 7. Home Screen...................................................................................................................................... 13
Figure 8. Thermal IR Color Palettes.................................................................................................................. 15
Figure 9. Video Display mode selection .......................................................................................................... 16
Figure 10: Video/Still Images and Record, shown in Still Images mode at 5-second intervals...................... 17
Figure 11. Main Settings (Radiometry and Spot Meter function (Duo R only) not shown).............................. 19
Figure 12. Radiometry Tab................................................................................................................................. 21
Figure 13. Spot Meter OSD, with “Temperature Unit” set to Fahrenheit......................................................... 22
Figure 14. About Page....................................................................................................................................... 24
Figure 15. PixHawk Flight Controller for PWM............................................................................................. 26
Figure 16. PWM Settings allow for control of Recording (start/stop and mode), IR Color Palette selection, Recalibrate, and streaming Display Video Mode........................................................................... 27
Figure 17. Function Selection............................................................................................................................ 28
Figure 18. PixHawk Flight Controller for MAVLink.......................................................................................... 29
Figure 19: FLIR Tools can be used for advanced thermal analysis on Radiometric JPEGs............................. 33
List of Tables

Table 1: Mini-USB Port 10-pin Assignment ............................................................ 9
Table 2: PWM Connection ..................................................................................... 26
Table 3: MAVLink Connection .............................................................................. 29
Table 4: Visible Sensor Image Formats Recorded on microSD Card ..................... 34
Table 5: Thermal IR Sensor Image Formats Recorded on microSD Card .................. 35
1 Introduction

1.1 Scope

Thank you so much for your purchase of the FLIR Duo or Duo R! Designed for commercial use, the Duo camera system is more than a simple thermal camera. It is a visible + thermal imaging instrument and data recorder that can add tremendous value to small Unmanned Aerial System (sUAS) operations and services.

This guide shows how to get your plug-and-play FLIR Duo camera mounted, connected, and collecting images & video.

Unless specifically stated otherwise, all features and functions of the Duo also apply to the Duo R. The Duo R functions identically to the Duo, with the addition of radiometric functionality to make non-contact temperature measurements.

1.2 Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>01/02/2017</td>
<td>Initial Release</td>
</tr>
<tr>
<td>200</td>
<td>09/28/2017</td>
<td>Updated App screenshots, Added Appendix C (TCP/IP Protocol)</td>
</tr>
<tr>
<td>201</td>
<td>10/13/2017</td>
<td>Updated Radiometry information (convert TIFF to temperature values)</td>
</tr>
</tbody>
</table>
2 Resources

There are many resources to help you operate your FLIR Duo:

2.1 FLIR Website / Tech Support Information

In several locations throughout this document, the FLIR Duo website is referenced as a source of additional information. This website can be accessed via the following URL:

http://www.flir.com/suas

Additionally, FLIR’s Technical Support department is referenced as a resource for obtaining additional help or information. The department can be accessed via the following phone number: (866) 667-7732.

2.2 FLIR Systems Documents (available on website)

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>FLIR Duo Quick Datasheet</td>
</tr>
<tr>
<td>436-0100-01-19</td>
<td>FLIR Duo Technical Drawing</td>
</tr>
<tr>
<td>n/a</td>
<td>FLIR Duo STEP File</td>
</tr>
<tr>
<td>102-9012-01</td>
<td>Interface Requirements Specification for FLIR TIFF File Format (Web Link)</td>
</tr>
<tr>
<td>n/a</td>
<td>FLIR UAS Radiometry Tech Note (Web Link)</td>
</tr>
</tbody>
</table>

2.3 External Documents

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>n/a</td>
<td>MAVLink Protocol (Web Link)</td>
</tr>
</tbody>
</table>
## Abbreviations / Acronyms

<table>
<thead>
<tr>
<th>Abbreviation/Acronym</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGC</td>
<td>Automatic Gain Control</td>
</tr>
<tr>
<td>ESD</td>
<td>Electrostatic Discharge</td>
</tr>
<tr>
<td>FFC</td>
<td>Flat Field Correction</td>
</tr>
<tr>
<td>FOV</td>
<td>Field of View</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>I/O</td>
<td>Input / Output</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>NTSC</td>
<td>National Television System Committee</td>
</tr>
<tr>
<td>OSD</td>
<td>On Screen Display</td>
</tr>
<tr>
<td>PAL</td>
<td>Phase Alternating Line</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>µSD</td>
<td>MicroSD (memory card)</td>
</tr>
</tbody>
</table>
3 What’s in the box

Your FLIR Duo comes with the Duo camera and a multi-function bench cable for testing the device.

Please note that disassembling the camera can cause permanent damage and will void the warranty. Operating the camera outside of the specified input voltage range or the specified operating temperature range can cause permanent damage. The camera housing is not sealed. Avoid exposure to dust and moisture.

3.1 Unpacking Your Camera

The FLIR Duo comes with the following components:

3.1.1 FLIR Duo Camera

Figure 1. Duo with Camera Mount
3.1.2 Bench Cable

The Bench Cable should be used for connecting the FLIR Duo to a computer USB port for power\(^1\) and file transfer, as well as video output testing with an analog video monitor. Analog video output is accessed by connecting the yellow RCA pigtail to a video display. Digital video is not accessible from the miniUSB port, but can be displayed using the microHDMI port.

The Bench Cable can also be used for connecting FLIR Duo to standard Radio/Control (R/C) Pulse Width Modulation (PWM) outputs. See Table 1 for pin definitions.

Connecting to a MAVLink-compatible autopilot is possible by building a customized cable based on accessing miniUSB pin1 and pin3 for MAVLink TX and RX. The MAVLink interface operates at a default data rate of 57.6 kbps, with the ability to switch to 115.2 kbps, 230.4 kbps through the user App. See Section 6 for more information.

Figure 2. Bench Cable

\(^1\) 5 V, 0.44 A (average), 0.66 A (peak)
4 Connecting to the Camera

This section describes the properties and methods of interfacing with the FLIR Duo, both mechanically and through software.

4.1 Mechanical Interface

4.1.1 Size / Weight

The overall size of the Duo is 58.9 mm x 40.9 mm x 30 mm (2.32 x 1.61 x 1.19 in), including the lens but excluding cables and buttons (which protrude slightly). Weight is approximately 65 grams (2.3 oz.). These mechanical specifications do not take into account any additional mounting hardware. For additional details, please see the Duo Technical Drawing referenced in Section 2.2.

4.1.2 Mounting

The Duo camera form-factor conveniently fits with most popular action-camera accessories: there are many available commercial off-the-shelf (COTS) gimbals and mounts available for mounting the Duo onto different UAS airframes. Refer to the Duo Technical Drawing for additional information.
4.2 Electrical Interface

4.2.1 Bench Testing

For initial bench testing, connect the 10-pin mini-USB connector on the Bench Cable (see Section 3.1.2) to the mini-USB port on the Duo camera. Connect the RCA plug on the Bench Cable to an analog video monitor. Connect the USB (Type-A) connector on the Bench Cable to an available USB port on your computer. Connect the PWM1 and PWM2 to a standard R/C PWM source, if needed. If using an HDMI display, connect the camera to the display using an HDMI cable (not included). If HDMI is used, the camera will automatically detect the connection and switch over to HDMI video output.

4.2.2 Mini-USB Cables

FLIR Duo is compatible with many commercial off-the-shelf (COTS) 10-pin mini-USB cables that are used to provide power to, and receive video from, action cameras commonly mounted on small unmanned aerial systems (sUAS). The included bench cable is for initial setup and configuration of the Duo. The internal wires of the FLIR bench cable are color coded to help you build a custom sUAS.
cable that can be used to integrate Duo to your airframe. Please refer to the Duo Technical Drawing for electrical interface guidelines and USB connector pinout.

Simply plug the chosen cable into the mini-USB port on the FLIR Duo, connect power to an appropriate filtered and regulated power supply, and video to a downlink, if desired. Approximate operating current of the Duo camera is 440 mA at 5Vdc, (2.2 Watts). Peak current can reach as high as 660 mA (3.3 Watts).

Note: FLIR Duo has over-voltage and reverse polarity protection on power pins. However, exceeding the wide voltage specification (5-26 Vdc) will damage the camera and void the warranty. Applying reverse polarity should not damage the unit, but it will prevent the camera for powering on and is not recommended.

Figure 4: Mini-USB 10-pin Layout

Table 1: Mini-USB Port 10-pin Assignment

<table>
<thead>
<tr>
<th>Mini-USB Port</th>
<th>Bench Cable Internal</th>
<th>Wire Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Main Power</td>
<td>red</td>
</tr>
<tr>
<td>4</td>
<td>Data Low</td>
<td>white</td>
</tr>
<tr>
<td>6</td>
<td>Data High</td>
<td>green</td>
</tr>
<tr>
<td>8</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Main Power GND</td>
<td>black</td>
</tr>
<tr>
<td>1</td>
<td>PWM_1 / MAVLink TX Out</td>
<td>yellow</td>
</tr>
<tr>
<td>3</td>
<td>PWM_2 / MAVLink RX In</td>
<td>purple</td>
</tr>
<tr>
<td>5</td>
<td>Video Low</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Video High</td>
<td></td>
</tr>
</tbody>
</table>
4.3 Software Interface

The Duo camera is configured and operated through the FLIR UAS mobile device app. See Section 5 for additional details.

*Note: The Duo camera is not compatible with the FLIR Camera Controller User Interface software. When connected to the USB port on a computer, the camera appears as a mass storage device only.*

There is no Software Developer’s Kit or API available for the camera. However, advanced camera control can be arranged using PWM, MAVLink, or TCP/IP integration (see Section 6).

4.4 Camera Operation

![Camera Status and Record LED Description](image)

**Figure 5. Camera Status and Record LED Description**

The Duo does not have an on/off switch. When power is applied the camera will boot automatically; this takes less than 30 seconds.

- When power is applied to the camera, the Status LED blinks red for approximately 20 seconds, then changes to solid blue, indicating that Bluetooth is enabled.
The Record LED will go from blinking orange to solid green, indicating that the camera is ready for operation
  - Note 1: If connected to a computer, the “Record” LED will continue blinking orange, indicating that the device is in “USB Mass Storage” mode
  - Note 2: If “Record” LED continues blinking orange while connected to a battery or bench power supply, this indicates that there is no SD card in the camera

Launch the FLIR UAS app on your mobile device.
  - Use the app to configure camera settings as desired
    - See Section 5 for information on camera settings & operating modes
  - Disable the Bluetooth radio in preparation for flight
    - Simply press the Bluetooth button on the side of the Duo, and confirm that the Status LED turns green
    - The Duo will automatically disable the Bluetooth radio after 2 minutes if no commands are received from the app. It can be re-started by pressing the Bluetooth button on the camera

*Note:* The Bluetooth interface is only intended for use during ground operations. FLIR does not recommend flying while Bluetooth is enabled. Bluetooth interface has limited range and high latency.

Your FLIR Duo is now ready for use. Press the Record button on the camera or in the app to start recording, or use PWM signals. The “Record” LED blinks red while recording video, or flashes red each time it captures a still image.

### 4.5 Camera Troubleshooting

- The Duo can be rebooted & reset to factory defaults through the FLIR UAS App
  - With the App connected to the Duo, go to:
    - “Settings” -> “About” -> “Reset To Factory Defaults”

- To reboot the system manually:
  - Press and hold the “Record” button on the Duo for at least 5 seconds

- To reboot and perform a factory reset to all default settings manually:
  - Press and hold both the “Record” and “Bluetooth” buttons for 10 seconds
The FLIR UAS app is the primary configuration interface for the Duo camera. It is compatible with mobile devices equipped with Bluetooth LE running iOS 8.0 or later and Android V 4.3 or above. The screenshots shown in this user manual are for the iOS App version; however, the menu system for Android is quite similar.

Figure 6: FLIR launch screen
5.1 Home Screen

When launched, the FLIR UAS APP will automatically identify local FLIR UAS cameras; Confirm that Bluetooth is enabled on your Duo device, then select the camera from the camera select screen. After connecting to the camera, the Home screen will be displayed. The Home screen shows camera status and allows the user to adjust the more frequently-used camera settings. The current PWM states are also displayed at the top of the Home screen.

Note: The “Measure” section allows for control over parameters related to radiometry, and will only appear for Duo R users.

![Home Screen](Image)

Figure 7. Home Screen
5.1.1 MSX

The Duo has the ability to enhance the IR imagery using FLIR’s patented MSX image blending technology. Enable this feature to record real-time MSX-enhanced thermal imagery, or disable it to record raw thermal image data. To activate, press the “Enable” or “Disable” text on the App, and select the desired mode.

To adjust MSX settings, simply press the MSX bar on the App home screen to access the MSX drop-down menu.

Note: To record video with MSX enhancements, be sure to set the IR Video mode to H264. MSX will not be recorded on 14-bit (TIFF Sequence) IR imagery, since this is recorded as raw video data.

5.1.1.1 MSX Strength

Use this feature to adjust the degree to which the MSX details show up in the IR imagery. A lower setting results in a softer effect, while a higher setting results in a stronger effect.

5.1.1.2 Alignment

The “Horizontal Align” and “Vertical Align” features allow the user to adjust the alignment between the visible image detail and the IR imagery with MSX enabled. Objects at different distances may require slight tuning of the alignment settings to achieve an optimal image, particularly for objects close to the camera.

5.1.1.2 Recommended MSX Alignment procedure

The MSX alignment adjustment can be used to optimize the MSX alignment for different use cases. Objects at different distances will require slightly different alignment settings to achieve an ideal image. The following method is a useful way to tune the MSX alignment prior to recording:

1) With the camera stationary, point at an easily-identifiable object at a distance similar to the intended flight height.
   a. For example, if you intend to fly at 20m above ground for a given video capture, focus on an object 20m away
2) While viewing the IR image with MSX enabled on a display monitor, adjust the MSX alignment settings to achieve optimal alignment for the test subject
3) Proceed with flight and data recording, and attempt to keep the Duo camera at the intended flight height to obtain optimal MSX-enhanced imagery.
5.1.2 IR Color Palette

The Duo detects and images long wave infrared radiation. Within the camera, this radiation is mapped to a range of 255 colors. Using black and white palette, such as White Hot, this range is converted to shades of gray with 0 being totally black and 255 being totally white. Different palettes are available to change the appearance of the image. The most common selection is typically White Hot (hotter objects appear lighter in color than cooler objects) or HotMetal (hotter objects appear reddish white as oppose to cooler objects in the scene).

Figure 8. Thermal IR Color Palettes

There are two types of palettes offered in the Duo camera.

1. **Linear**: Scene radiation is mapped to pixel values where color is uniformly distributed from one shade to another as pixel intensity increases. These palettes are the most common palettes for general use. ([WhiteHot & HotMetal palettes])
2. **Contrast**: Enhanced observed contrast by mapping pixel values to divergent color schemes. Subtle differences between object temperatures become more apparent. ([Rainbow palette])
5.1.3 Display Video Mode

The Duo streams analog or HDMI video from the output ports. Selection of the display video mode toggles between a **Visible** stream only, thermal infrared (**IR**) stream only, and a **Picture-In-Picture** stream that shows both visible and thermal. Note that this setting does NOT impact what video is recorded to disk – see the “Settings” -> “Main” -> “Recorded Video” setting for information on that feature.

This feature will also impact the image thumbnail for the Radiometric JPEG file format: set to “IR” to use the IR imagery as the thumbnail, or set to “Visible” or “Picture-In-Picture” for a visible image thumbnail. Note that both IR & Visible image components are still recorded, and accessible through FLIR Tools.

![Select Video Mode](image-url)

**Figure 9. Video Display mode selection**
5.1.4 Video/Still Image

The **Video** and **Still Images** tabs configure the Duo recording mode. The system cannot capture a still image during recording of video (or vice versa). The type of video or image files can be configured in **Settings**.

### 5.1.4.1 Video

Video recordings can record either visible, infrared, or both simultaneously. File type settings are adjustable in the **Settings** screen.

### 5.1.4.2 Still Images

Configures image capture at a user-selectable interval from one frame per second (slider setting of “1s”) to one frame every 60 seconds (slider setting of “60s”). Duo can be configured to record visible and IR as separate data files or as one combined FLIR Radiometric JPEG file.

Use the “Single” slider setting (far left) to record a single still image.

---

**Figure 10**: Video/Still Images and Record, shown in Still Images mode at 5-second intervals.
5.1.5 Record

The “Record” button either captures a still image, or starts/stops live video recording. See Section 7 for details on available file formats.

5.1.6 FFC (Recalibrate)

To maintain optimum performance, thermal imaging cameras must occasionally perform an internal recalibration (also known as a Flat Field Correction, or FFC). This is accomplished using an internal shutter, and takes less than one second. During recalibration, a subtle audible “click” can be heard, and live video is momentarily frozen. This function takes place automatically (based on internal camera parameters), and can be manually initiated through user command via the “Recalibrate” button on the home screen, or through an appropriately-configured PWM channel.

Recalibrating prior to taking critical measurements will ensure the most consistent image contrast.
### 5.1.7 Settings

The “Settings” button is used to configure additional operational parameters and camera modes.

#### 5.1.7.1 Main

![Main Settings](image)

- **Capture Mode**: Video
- **Photo Format**: Radiometric JPEG
- **IR Video Format**: 8-bit (H264)
- **Recorded Video**: IR & Visible
- **Sound When Start/Stop Record**: Off
- **Status Light**: On
- **Flip Image**: Off
- **Thermal**
  - **Color Palette**: WhiteHot
- **Video Output**
  - **Analog Output Format**: NTSC
  - **HDMI**: 1080p60

Figure 11. Main Settings (Radiometry and Spot Meter function (Duo R only) not shown)
• **Capture Mode**: Select the recording mode: video, still, or multiple.

• **Photo Format**: Select formats for still images and video saved to the µSD card (See Section 7)

• **IR Video Format**: Infrared video can be saved as 8-bit (H264) that has FLIR AGC (automatic gain correction), MSX, Spot Meter (Duo R only), and color palette applied, or 14-bit (TIFF Sequence) raw sensor data. See Section 7 for more details.

• **Recorded Video**: This setting determines the type of sensor data that is saved to the µSD card. The Duo can be setup to record Visible Only, IR Only, or IR & Visible.

• **Sound When Start/Stop Record**: Turns on/off the audible indicator for video recording. Even when turned off, the speaker will beep three times when the camera is ready after power-up.

• **Status Light**: Turns on/off the Status and Bluetooth LEDs. LED indicators will still be displayed while camera is booting up, even if this setting is turned off.

• **Camera Orientation Flip**: Select image orientation. If the camera will be mounted upside-down, enable this setting to vertically flip the images recorded by the Duo.

• **Color Palette**: Select the IR color palette

• **Radiometry (Duo R only)**: Enter the Radiometry parameter adjustment menu

• **Spot Meter (Duo R only)**: Enable on-screen spot meter that shows the temperature of the object in the center of the image

• **Analog Output Format**: Select NTSC or PAL as the format of the video that is output to the 10-pin mini USB port. This setting does not affect the format of the video that is saved to the microSD card.
**HDMI**: Select digital output stream format (through the microHDMI port). Available options are **720p60** (60hz, 720p), **1080p30** (30hz, 1080p), and **1080p60** (60hz, 1080p). Note that this does not change the actual recording frequency, only the display rate frequency when connected to a digital monitor.

5.1.7.2 Accy. Port

See Section 6 for information on connecting via PWM and MAVLink.

5.1.7.3 Radiometry

The Radiometry tab provides access to all the temperature measurement functions and settings available on the Duo R. If connecting to a standard Duo camera, this tab will not be visible and these features are unavailable. See FLIR’s UAS Radiometry Tech Note (see Section 2.2 for the link) for a detailed discussion on how to obtain accurate temperature readings for UAS applications.

![Radiometry Tab](image)

**Figure 12. Radiometry Tab**

- **Temperature Unit** – Units of measure displayed on the analog video. Select between Celsius and Fahrenheit.
- **Spot Meter** – Turns on or off the spot meter (fixed 4x4 pixel array) in the center of the image and the digital temperature in the lower left corner of the image.

![Spot Meter OSD](image)

Figure 13. Spot Meter OSD, with “Temperature Unit” set to Fahrenheit

- **Sky Condition** – Measure of the cloud cover above the operating site. This affects the background radiation incident on the scene. Choose between Clear, Scattered, and Cloudy.
- **Humidity** – Relative moisture content of the air. Three settings are available; Low (<30%), Medium (~50%), High (>75%).
- **Air Temp.** – Ambient temperature of the operating environment. Values from 0 to 40°C (32 to 104°F) can be configured.
- **Emissivity** – Measure of the target surface ability to emit thermal energy. Values from 50-100% can be configured.
- **Subject Distance** – Distance from the camera to the subject in the scene. Values from 0-200m (0-218 yards) can be configured.
5.1.7.3.1 Radiometry File Formats

The different available file formats have different characteristics with regards to recording radiometric data, as follows:

- **Video (8-bit H264)** – If spot meter is enabled, this format will record the spot meter reading from the center of the screen. This video is not editable for post-processing.

- **Video (14-bit TIFF Sequence)** – This video format will record temperature values for every pixel. It can be processed using many different software tools, such as FLIR Tools, ResearchIR, ImageJ, MATLAB, and others.
  - To convert the pixel values to degrees Celsius, multiply the entire image by 0.01 and then subtract 273.15.

- **Photo (Radiometric JPEG)** – This photo format will record temperature values for every pixel. It can be processed using FLIR Tools and ResearchIR.
  - A major benefit to this file format is that all radiometry parameters, such as emissivity and subject distance, can be edited in post-processing. This is not true for the TIFF file formats.

- **Photo (14-bit TIFF)** – This photo format will record temperature values for every pixel.
  - To convert pixel values to degrees Celsius, multiply the entire image by 0.01 and then subtract 273.15.

5.1.7.4 About

The “About” screen provides information on the current software installed on the Duo, as well as the App version.
### Figure 14. About Page

- **Scan for Camera**: Searches for any FLIR UAS camera in the area and provides user ability to change cameras.

- **USB Interface**: Use this setting to select between USB Mass Storage mode (the default setting) and Ethernet mode. Recommended for advanced users only – see section 6 for more detail on controlling the camera via a virtual Ethernet connection.

- **USB Boot Mode**: Sets which USB Interface mode the camera is defaulted to upon power-on.

- **Persistent Bluetooth** – Prevents the Bluetooth interface from shutting off after 2 minutes by keeping the APP active.

- **Bluetooth Power** – Adjusts the Bluetooth transmitter power and affects the maximum range at which the Duo camera can connect to the mobile device. Power values are expressed in percent.
FLIR Duo User Guide

- **Turn Off Camera’s Bluetooth:** Powers down the Bluetooth radio on the camera. The camera will normally disable the Bluetooth after two minutes of inactivity. See Persistent Bluetooth.
- **Serial No.** – Serial number of the Duo camera currently connected to the app. A “D” indicates a Duo, while a “DR” indicates a Duo R.
- **Part No.** – FLIR part number of the Duo camera currently connected to the app.
- **Firmware Version** – Version of firmware currently loaded in the Duo camera.
- **FW Upgrade Version** - Detects if a camera upgrade file exists on the µSD card. If one is available, the “Upgrade Available” dialog will start. Choose “Continue” to begin the firmware upgrade process. See Appendix A for details on how to upgrade camera firmware.
- **App Version** - Version of software currently loaded on the mobile device.
- **App Upgrade Version** – If the mobile device is connected to the Internet, the Duo app will automatically search for any available app updates. If one exists, the user will have the option to update by following the link to the FLIR web page.
- **Terms and Conditions** – Legal statements pertaining to the Duo product and FLIR UAS App
- **Reset To Factory Defaults** – Resets settings in the Duo camera to original factory configuration. This is required during most FW updates as the internal memory map will change as new features are added.

6 PWM, MAVLink, and TCP/IP Operation

6.1 PWM

6.1.1 Connecting to a PWM Compatible Flight Controller
The FLIR Duo camera is software- and connector-compatible with standard R/C PWM controllers using the MiniUSB pinouts. An illustrative PWM implementation connection is described in Table 2. Other flight controllers or I/O modules may require different cables or connectors. Refer to the FLIR Duo Technical Drawing for detailed information.
Figure 15. PixHawk Flight Controller for PWM

Table 2: PWM Connection

<table>
<thead>
<tr>
<th>MiniUSB Port</th>
<th>Bench Cable Internal Wire Color</th>
<th>Aircraft Power Source</th>
<th>P1 - AUX Out (M20-1060300)²</th>
<th>P2 - AUX Out (M20-1060300)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Main Power</td>
<td>red</td>
<td>5-26 Vdc</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Main Power GND</td>
<td>black</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>PWM_1</td>
<td>yellow</td>
<td></td>
<td>PWM_1</td>
</tr>
<tr>
<td>10</td>
<td>GND</td>
<td>black</td>
<td></td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>PWM_2</td>
<td>purple</td>
<td></td>
<td>PWM_2</td>
</tr>
<tr>
<td>10</td>
<td>GND</td>
<td>black</td>
<td></td>
<td>GND</td>
</tr>
</tbody>
</table>

² All cable plug part numbers are Hirose

Information on this page is subject to change without notice
6.1.2 Configuring PWM Connection

Using the accessory port on the Duo and the included Bench Cable, camera functions can be controlled directly from the sUAS flight controller via PWM signals (refer to the controller manual for configuration instructions). Select the desired setting from the list of available PWM functions. **No two channels can be assigned to the same function.** Any functions currently assigned to PWM channels will not appear in the list of available functions for other channels.

To configure, select the “Settings” button on the Main page. Select the **Accy. Port** tab at the top of the page. Ensure Serial Protocol is set to PWM. Select a PWM channel to configure (Figure 16). Select the desired function. Select the number of states (Figure 17). The number of states will depend on the type of switch that is being mapped on the flight controller. For a three-position switch, the number of states will be three. Assign functions to the available states for each signal. Next, program the flight controller to provide signals to the Aux Out ports for the PWM connectors on the Bench Cable.

The Duo accepts standard PWM inputs for the R/C industry, 3.3-5vdc, 50Hz. LOW = 1ms/20ms, MID = 1.5ms/20ms, HIGH = 2ms/20ms.

![Select PWM1 Function](image)

**Figure 16.** PWM Settings allow for control of Recording (start/stop and mode), IR Color Palette selection, Recalibrate, and streaming Display Video Mode
6.2 MAVLink

6.2.1 Connecting to a MAVLink Compatible Flight Controller

The FLIR Duo camera is compatible with many MAVLink autopilots using the MiniUSB pinouts, illustrated in Figure 18 and Table 3. Other flight controllers or I/O modules may require different cables or connectors. Refer to the FLIR Duo Technical Drawing for detailed information.
Table 3: MAVLink Connection

<table>
<thead>
<tr>
<th>MiniUSB Port</th>
<th>Bench Cable Internal Wire Color</th>
<th>P2 - MAVLink (DF13-6S-1.25C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Main Power red</td>
<td>1 Main Power: 5 Vdc</td>
</tr>
<tr>
<td>10</td>
<td>Main Power GND black</td>
<td>6 Main Power GND</td>
</tr>
<tr>
<td>1</td>
<td>MAVLink TX Out yellow</td>
<td>3 MAVLink RX In</td>
</tr>
<tr>
<td>3</td>
<td>MAVLink RX In purple</td>
<td>2 MAVLink TR Out</td>
</tr>
</tbody>
</table>

6.2.2 Configuring MAVLink Connection

Many UAS flight controllers support the MAVLink serial protocol to provide an interface with external components. The Duo can be configured to use this bus to capture available telemetry data provided by GPS, altimeter, accelerometers, etc. This data is saved as standard EXIF metadata in all Still Image files. This data is not saved to Video files as there is no current standard for recording.

3 All cable plug part numbers are Hirose
FLIR Duo User Guide

video metadata. Metadata is accessible through standard photo editing applications, file explorers, and suggested mapping applications.

The following MAVLink 3.0 messages are currently supported by the Duo FW 1.2.4 or newer:

<table>
<thead>
<tr>
<th>Command</th>
<th>MAVLink Message ID</th>
<th>R/S (Receive / Send)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEARTBEAT</td>
<td>0</td>
<td>R/S</td>
<td></td>
</tr>
<tr>
<td>PING</td>
<td>4</td>
<td>R/S*</td>
<td>If PING is received, then will reply with a PING</td>
</tr>
<tr>
<td>SYSTEM_TIME</td>
<td>2</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>ATTITUDE</td>
<td>30</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>GLOBAL_POSITION_INT</td>
<td>33</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>GLOBAL_POSITION_INT_COV</td>
<td>63</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>HIL_GPS</td>
<td>113</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>MOUNT_STATUS</td>
<td>158</td>
<td>R</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>MAVLink Component ID</th>
<th>R/S (Receive / Send)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAV_COMP_ID_CAMERA</td>
<td>100</td>
<td>/</td>
<td></td>
</tr>
</tbody>
</table>

Information on this page is subject to change without notice.
FLIR Duo User Guide

| MAV_CMD_DO_CONTROL_VIDEO                | 200 | R   | Custom FLIR parameters |
| MAOV_CMD_DO_DIGICAM_CONFIGURE          | 202 | R   | Custom FLIR parameters |
| MAV_CMD_DO_DIGICAM_CONTROL             | 203 | R   | Custom FLIR parameters |
| MAV_CMD_IMAGE_START_CAPTURE            | 2000| R   | Custom FLIR parameters |
| MAV_CMD_IMAGE_STOP_CAPTURE             | 2001| R   |                     |
| MAV_CMD_VIDEO_START_CAPTURE            | 2500| R   | Custom FLIR parameters |
| MAV_CMD_VIDEO_STOP_CAPTURE             | 2501| R   |                     |
| MAV_CMD_USER_1                         | 31010| R   | Custom FLIR parameters |

FLIR Duo communicates on the MAVLink bus at 57600 baud which is standard for most devices, as default, but is configurable from the app. Where possible, ensure RTSCTS is disabled as this is known to cause issues with communication. If all available flight controller ports are full, you may need to investigate using a splitter cable to attach additional devices. Please refer to Table 1 for miniUSB Port pin-out when building custom cables.

Duo can operate in a receive-only configuration much like an onscreen display (OSD), and therefore does not require the TX pin from the camera to be connected to the RX pin on a flight controller. Please note that all flight controllers will sleep a serial port if no connection is detected. The TX port can only be disconnected when used in a Y-harness with a device providing a “heartbeat”.

Please see Appendix B for the full MAVLink protocol implementation.

6.2.3 Duo-specific Custom MAVLink Commands

In addition to the MAVLink commands listed above, the Duo camera supports certain customized MAVLink commands to control various parameters of the camera. Experienced users familiar with MAVLink can update the communication protocol on their Ground Station Controller to enable greater in-flight control of the Duo camera.

Please see Appendix B for the custom MAVLink protocol implementation.
6.3 TCP/IP

For all Duo FW versions 2.1.4 or later, the camera now supports a TCP/IP communication protocol for command/control and small file transfer over USB.

To enable TCP/IP communication with your Duo, set “USB Interface” to “Ethernet” with the mobile APP or over MAVlink, and then send commands from your host computer. The TCP/IP and MAVlink interfaces can operate simultaneously, and provide complementary functionality in a fully integrated system.

Please see Appendix C for the TCP/IP protocol implementation.
7 File Formats

The Duo can save image data in a number of file formats.

7.1 Radiometric JPEG (FLIR Tools)

The Radiometric JPEG (RJPEG) format has both the compressed visible JPEG image and 14-bit raw IR sensor data in a single file. Although this image can be viewed with any JPEG viewer, accessing the full-data will require FLIR Tools or FLIR Research IR software. If the camera is setup for MAVLink integration, telemetry will be captured and saved in standard metadata fields.

Figure 19: FLIR Tools can be used for advanced thermal analysis on Radiometric JPEGS
7.2 JPEG

The JPEG format of the Duo stores the compressed visible image. This is the full-resolution visible image that can be viewed with any JPEG viewer. If the camera is setup for MAVLink integration, telemetry will be captured and saved in standard metadata fields.

7.3 TIFF, TIFF-Sequen ce

TIFF is uncompressed 14-bit raw IR sensor data with no post processing; radiometric count values are recorded for each pixel. If the camera is setup for MAVLink integration, telemetry will be captured and saved in standard metadata fields.

The following links provide information on viewing 14-bit TIFF images and TIFF-sequence videos.

Table 4: Visible Sensor Image Formats Recorded on microSD Card

<table>
<thead>
<tr>
<th>File Format</th>
<th>Feature</th>
<th>Use</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Visible Image</td>
<td>IR 8-bit Colorized Palette</td>
<td>IR Raw 14-bit</td>
</tr>
<tr>
<td>Visible Still Image File Type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPEG</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visible Video File Type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV (H.264)</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

⁴ Requires MAVLink integration

Information on this page is subject to change without notice
## Table 5: Thermal IR Sensor Image Formats Recorded on microSD Card

<table>
<thead>
<tr>
<th>File Format</th>
<th>Feature</th>
<th>Use</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Visible Image</td>
<td>IR 8-bit Colorized Palette</td>
<td>IR Raw 14-bit</td>
</tr>
<tr>
<td><strong>Thermal IR Still Image File Type</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPEG</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TIFF</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>Thermal IR Video File Type</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV (H.264)</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>TIFF-Seq</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

3 Requires MAVLink integration

6 IR 8-bit Thermal Colorized Recording only in **Display Video Mode: IR**. Always accessible using FLIR Software.
7.4 Recommended Application Links

- FLIR Tools: http://www.flir.com/instruments/display/?id=54865
- FLIR ResearchIR: http://www.flir.com/Science/display/?id=51371
- Pix4D: https://pix4d.com/
- MATLAB: http://www.mathworks.com/products/matlab/
  - When using MATLAB it is recommended that the FLIR Atlas SDK is installed to expose the full metadata set available from FLIR radiometric JPEG files: http://support.flir.com/resources/atlas-matlab
- Software Developers can access full metadata from FLIR radiometric JPEGs by integrating the Atlas SDK: https://flir.custhelp.com/app/answers/detail/a_id/1043/kw/atlas%20sdk
8 Care of FLIR Duo

- Power your FLIR Duo with a regulated 5-26 Vdc power source. Using more than 26 Vdc will damage the camera and void the warranty.

- Do not touch the lens. If the lens gets dirty, a light dusting of air should dislodge any dust particles. If the lens is still noticeably dirty, use 75% Isopropyl alcohol and lens tissue. Use light wiping motions with a fresh section of lens tissue with each swipe so as not to drag dust or dirt particles back over the lens surface.

- FLIR Duo cameras have been focused at the factory and optimized for the maximum UAS range. Opening the camera may compromise the external seal of the camera, and factory focus will be lost. This also voids the camera warranty.

- The FLIR Duo is neither water nor dust resistant. Care for it as you would any valuable piece of electronics equipment.

If you have questions about your Duo camera, contact FLIR Tech Support at 1-866-667-7732.
Appendix A - Software and Firmware Update

How to Update App

Download the latest handset application from the appropriate web store for your platform. Some handy links are listed below. Also be sure to keep the firmware and Mobile App up to date on your device to receive continuous improvements.

iOS APP store (Web Link)

Android APP store (Web Link)

Android APK file direct DL (Web Link)

How to update Firmware (FW)

Please download the latest firmware from the FLIR web site here. (http://www.flir.com/suas/duo/software)

Duo Firmware Upgrade Procedure

Instructions:

1) Connect to the Duo with the Bluetooth application and make note of all your settings. The firmware upgrade will require resetting the camera settings to defaults.

2) Download the latest firmware update from http://www.flir.com/suas/duo/software

3) Connect the Duo to your computer via miniUSB
   a) The Duo will appear as a removable storage device on your computer
   b) Drag & drop latest firmware (.ITM file) onto the Duo camera
   c) Connect to the Duo over Bluetooth via the FLIR UAS App
   d) Follow the automated in-app prompts to update device FW
      i. Do not restart camera during FW update
      ii. This step can also be initiated through “Settings” -> “About” -> “FW Upgrade Version”
   e) After the device reboots, connect again over Bluetooth via the FLIR UAS App

4) Confirm that the correct FW has been uploaded by again viewing the “Settings” -> “About” tab.
5) Restore User Settings. From your notes in Step 1 reconfigure any Main, Accessory Port, and Radiometry (Duo R only) settings for your specific application. You can also delete the firmware update file from your microSD card (it does not automatically delete itself).
### Appendix B - MAVLink Implementation

<table>
<thead>
<tr>
<th>CMD ID</th>
<th>Field Name</th>
<th>Data Type</th>
<th>Description (MAVlink published data)</th>
<th>FLIR Implementation</th>
<th>Device Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HEARTBEAT</td>
<td></td>
<td>Shows the system is present and responding.</td>
<td></td>
<td>FLIR Duo / R</td>
</tr>
<tr>
<td></td>
<td>System ID</td>
<td>MAVLink packet Byte 3</td>
<td></td>
<td>Standard</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Component ID</td>
<td>MAVLink packet Byte 4</td>
<td></td>
<td>Standard</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Type</td>
<td>uint8_t</td>
<td>MAV_TYPE_GENERIC - Generic micro air vehicle.</td>
<td>Standard</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Autopilot</td>
<td>uint8_t</td>
<td>MAV_AUTOPILOT_INVALID - No valid autopilot, e.g. a GCS or other MAVLink component</td>
<td>Standard</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Base Mode</td>
<td>uint8_t</td>
<td>MAV_MODE_FLAG_MANUAL_INPUT_ENABLED - 0b01000000 remote control input is enabled.</td>
<td>Standard</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>Custom Mode</td>
<td>uint32_t</td>
<td>A bitfield for use for autopilot-specific flags.</td>
<td>Standard</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>System Status</td>
<td>uint8_t</td>
<td>MAV_STATE_ACTIVE - System is active and might be already airborne. Motors are engaged.</td>
<td>Standard</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>MAVLink Version</td>
<td>uint8_t</td>
<td>MAVLink version, not writable by user</td>
<td>Standard</td>
<td>Standard</td>
</tr>
<tr>
<td>2</td>
<td>SYSTEM_TIME</td>
<td></td>
<td>System master clock time.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time Unix</td>
<td>uint64_t</td>
<td>Timestamp of the master clock in microseconds since UNIX epoch.</td>
<td>Standard</td>
<td>If time is valid, it will be set to the device.</td>
</tr>
<tr>
<td></td>
<td>Time Boot</td>
<td>uint32_t</td>
<td>Timestamp of the component clock since boot time in milliseconds.</td>
<td>Standard</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

Information on this page is subject to change without notice
**PING**

System latency diagnostics.

- **Time Unix**
  - `uint64_t`
  - Timestamp (microseconds since UNIX epoch or microseconds since system boot)

- **Sequence**
  - `uint32_t`
  - PING sequence

- **System ID**
  - `uint8_t`
  - 0: request ping from all receiving systems, if greater than 0: message is a ping response and number is the system id of the requesting system

- **Component ID**
  - `uint8_t`
  - 0: request ping from all receiving components, if greater than 0: message is a ping response and number is the system id of the requesting system

---

**ATTITUDE**

The attitude in the aeronautical frame.

- **Time Boot**
  - `uint32_t`
  - Timestamp (milliseconds since system boot)

- **Roll**
  - `float`
  - Roll angle (rad, -π..+π)

- **Pitch**
  - `float`
  - Pitch angle (rad, -π..+π)

- **Yaw**
  - `float`
  - Yaw angle (rad, -π..+π)

- **Roll Rate**
  - `float`
  - Roll angular velocity (rad/s)

- **Pitch Rate**
  - `float`
  - Pitch angular velocity (rad/s)

- **Yaw Rate**
  - `float`
  - Yaw angular velocity (rad/s)

---

**GLOBAL_POSITION_INT**

Filtered GPS position.

- **Time Boot**
  - `uint32_t`
  - Timestamp (milliseconds since system boot)

- **Latitude**
  - `int32_t`
  - Latitude, expressed as * 1E7

- **Longitude**
  - `int32_t`
  - Longitude, expressed as * 1E7

- **Altitude (MSL)**
  - `int32_t`
  - Altitude in meters, expressed as * 1000 (millimeters), AMSL (not WGS84 - note that virtually all GPS modules provide the AMSL as well)

- **Altitude (Relative)**
  - `int32_t`
  - Altitude above ground in meters, expressed as * 1000 (millimeters)
### Velocity X
- **Type**: `int16_t`  
- **Description**: Ground X Speed (Latitude), expressed as m/s * 100  
- **Metadata**: calculated  
- **Category**: GPS Speed

### Velocity Y
- **Type**: `int16_t`  
- **Description**: Ground Y Speed (Longitude), expressed as m/s * 100  
- **Metadata**: calculated  
- **Category**: GPS Speed

### Velocity Z
- **Type**: `int16_t`  
- **Description**: Ground Z Speed (Altitude), expressed as m/s * 100  
- **Metadata**: calculated  
- **Category**: GPS Speed

### Heading
- **Type**: `uint16_t`  
- **Description**: Compass heading in degrees * 100, 0.0..359.99 degrees. If unknown, set to: UINT16_MAX  
- **Metadata**: calculated  
- **Category**: GPS Track

---

### GLOBAL_POSITION_INT_COV

- **Global Position with Higher Resolution**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
<th>Metadata</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Boot</td>
<td><code>uint32_t</code></td>
<td>Timestamp (milliseconds since system boot)</td>
<td>Ignored</td>
<td>Ignored</td>
</tr>
<tr>
<td>Time Unix</td>
<td><code>uint64_t</code></td>
<td>Timestamp (microseconds since UNIX epoch in UTC. 0 for unknown.</td>
<td>Ignored</td>
<td>Ignored</td>
</tr>
<tr>
<td>estimator_type</td>
<td><code>uint8_t</code></td>
<td>Class id of the estimator this estimate originated from.</td>
<td>Ignored</td>
<td>Ignored</td>
</tr>
<tr>
<td>Latitude</td>
<td><code>int32_t</code></td>
<td>Latitude, expressed as * 1E7</td>
<td>Metadata</td>
<td>GPS Latitude</td>
</tr>
<tr>
<td>Longitude</td>
<td><code>int32_t</code></td>
<td>Longitude, expressed as * 1E7</td>
<td>Metadata</td>
<td>GPS Longitude</td>
</tr>
<tr>
<td>Altitude (MSL)</td>
<td><code>int32_t</code></td>
<td>Altitude in meters, expressed as * 1000 (millimeters). AMSL (not WGS84 - note that virtually all GPS modules provide the AMSL as well)</td>
<td>Metadata</td>
<td>GPS Altitude</td>
</tr>
<tr>
<td>Altitude (Relative)</td>
<td><code>int32_t</code></td>
<td>Altitude above ground in meters, expressed as * 1000 (millimeters)</td>
<td>Metadata</td>
<td>MAVRelativeAltitude</td>
</tr>
<tr>
<td>Velocity X</td>
<td><code>float</code></td>
<td>Ground X Speed (Latitude), expressed as m/s * 100</td>
<td>Metadata</td>
<td>GPS Speed</td>
</tr>
<tr>
<td>Velocity Y</td>
<td><code>float</code></td>
<td>Ground Y Speed (Longitude), expressed as m/s * 100</td>
<td>Metadata</td>
<td>GPS Speed</td>
</tr>
<tr>
<td>Velocity Z</td>
<td><code>float</code></td>
<td>Ground Z Speed (Altitude), expressed as m/s * 100</td>
<td>Metadata</td>
<td>GPS Speed</td>
</tr>
<tr>
<td>covariance</td>
<td><code>float[36]</code></td>
<td>Covariance matrix (first six entries are the first ROW, next six entries are the second row, etc.)</td>
<td>Ignored</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

---

### HIL_GPS

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
<th>Metadata</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>time_usec</td>
<td><code>uint64_t</code></td>
<td>Timestamp (microseconds since UNIX epoch or microseconds since system boot)</td>
<td>Ignored</td>
<td>Ignored</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
<td>Metadata</td>
<td>Notes</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>fix_type</td>
<td>uint8_t</td>
<td>0: no fix, 2: 2D fix, 3: 3D fix. Some applications will not use the value of this field unless it is at least two, so always correctly fill in the fix.</td>
<td>Ignored</td>
<td></td>
</tr>
<tr>
<td>lat</td>
<td>int32_t</td>
<td>Latitude (WGS84), in degrees * 1E7</td>
<td>Metadata</td>
<td>GPS Latitude</td>
</tr>
<tr>
<td>lon</td>
<td>int32_t</td>
<td>Longitude (WGS84), in degrees * 1E7</td>
<td>Metadata</td>
<td>GPS Longitude</td>
</tr>
<tr>
<td>alt</td>
<td>int32_t</td>
<td>Altitude (AMSL, not WGS84), in meters * 1000 (positive for up)</td>
<td>Metadata</td>
<td>GPS Altitude</td>
</tr>
<tr>
<td>eph</td>
<td>uint16_t</td>
<td>GPS HDOP horizontal dilution of position in cm (m*100). If unknown, set to: 65535</td>
<td>Metadata</td>
<td>GPSXYAccuracy</td>
</tr>
<tr>
<td>epv</td>
<td>uint16_t</td>
<td>GPS VDOP vertical dilution of position in cm (m*100). If unknown, set to: 65535</td>
<td>Metadata</td>
<td>GPSZAccuracy</td>
</tr>
<tr>
<td>vel</td>
<td>uint16_t</td>
<td>GPS ground speed (m/s * 100). If unknown, set to: 65535</td>
<td>Ignored</td>
<td></td>
</tr>
<tr>
<td>vn</td>
<td>int16_t</td>
<td>GPS velocity in cm/s in NORTH direction in earth-fixed NED frame</td>
<td>Ignored</td>
<td></td>
</tr>
<tr>
<td>ve</td>
<td>int16_t</td>
<td>GPS velocity in cm/s in EAST direction in earth-fixed NED frame</td>
<td>Ignored</td>
<td></td>
</tr>
<tr>
<td>vd</td>
<td>int16_t</td>
<td>GPS velocity in cm/s in DOWN direction in earth-fixed NED frame</td>
<td>Ignored</td>
<td></td>
</tr>
<tr>
<td>cog</td>
<td>uint16_t</td>
<td>Course over ground (NOT heading, but direction of movement) in degrees * 100, 0.0..359.99 degrees. If unknown, set to: 65535</td>
<td>Ignored</td>
<td></td>
</tr>
<tr>
<td>satellites_visible</td>
<td>uint8_t</td>
<td>Number of satellites visible. If unknown, set to 255</td>
<td>Ignored</td>
<td></td>
</tr>
</tbody>
</table>

**MOUNT_STATUS**

- **pointing_a**
  - uint32_t: Pitch (deg*100)  
  - Metadata: Orientation of gimbal  
  - Yaw (camera)
- **pointing_b**
  - uint32_t: Roll (deg*100)  
  - Metadata:  
  - Pitch (camera)
- **pointing_c**
  - uint32_t: Yaw (deg*100)  
  - Metadata:  
  - Roll (camera)
- **Target System**
  - uint8_t: System ID  
  - Ignored
- **Target Component**
  - uint8_t: Component ID  
  - Ignored

---

Information on this page is subject to change without notice
## MAV_CMD_DO_CONTROL_VIDEO

200 **MAV_CMD_DO_CONTROL_VIDEO**

Control onboard camera system.

<table>
<thead>
<tr>
<th>Mission Param #1</th>
<th>float</th>
<th>Camera ID (-1 for all)</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission Param #2</td>
<td>float</td>
<td>Transmission: 0: disabled, 1: enabled compressed, 2: enabled raw</td>
<td>Video signal made available to Analog video or HDMI port</td>
</tr>
<tr>
<td>Mission Param #3</td>
<td>float</td>
<td>Transmission mode: 0: video stream, &gt;0: single images every n seconds (decimal)</td>
<td>Ignore</td>
</tr>
<tr>
<td>Mission Param #4</td>
<td>float</td>
<td>Recording: 0: disabled, 1: enabled compressed, 2: enabled raw</td>
<td>Video data to record to disk</td>
</tr>
<tr>
<td>Mission Param #5</td>
<td>float</td>
<td>Empty</td>
<td>Ignored</td>
</tr>
<tr>
<td>Mission Param #6</td>
<td>float</td>
<td>Empty</td>
<td>Ignored</td>
</tr>
<tr>
<td>Mission Param #7</td>
<td>float</td>
<td>Empty</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mission Param #1</th>
<th>float</th>
<th>Modes: P, TV, AV, M, Etc</th>
<th>Set Scene (AGC) mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission Param #2</td>
<td>float</td>
<td>Shutter speed: Divisor number for one second</td>
<td>Set Color Palette</td>
</tr>
<tr>
<td>Mission Param #3</td>
<td>float</td>
<td>Aperture: F stop number</td>
<td>Set Scene ROI range</td>
</tr>
<tr>
<td>Mission Param #4</td>
<td>float</td>
<td>ISO number e.g. 80, 100, 200, Etc</td>
<td>Set camera Gain Mode</td>
</tr>
<tr>
<td>Mission Param #5</td>
<td>float</td>
<td>Exposure type enumerator</td>
<td>Rotate image 0 or 180 deg</td>
</tr>
</tbody>
</table>

## MAV_CMD_DO_DIGICAM_CONFIGURE

202 **MAV_CMD_DO_DIGICAM_CONFIGURE**

Configure an on-board camera.

<table>
<thead>
<tr>
<th>Mission Param #1</th>
<th>float</th>
<th>Modes: P, TV, AV, M, Etc</th>
<th>Set Scene (AGC) mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission Param #2</td>
<td>float</td>
<td>Shutter speed: Divisor number for one second</td>
<td>Set Color Palette</td>
</tr>
<tr>
<td>Mission Param #3</td>
<td>float</td>
<td>Aperture: F stop number</td>
<td>Set Scene ROI range</td>
</tr>
<tr>
<td>Mission Param #4</td>
<td>float</td>
<td>ISO number e.g. 80, 100, 200, Etc</td>
<td>Set camera Gain Mode</td>
</tr>
<tr>
<td>Mission Param #5</td>
<td>float</td>
<td>Exposure type enumerator</td>
<td>Rotate image 0 or 180 deg</td>
</tr>
</tbody>
</table>

Information on this page is subject to change without notice.
### FLIR Duo User Guide

**Mission Param #6**
- **Type:** float
- **Command Identity:** Set MSX function
- **Value:**
  - 0 = Disabled
  - 1 = Enabled

**Mission Param #7**
- **Type:** float
- **Main engine cut-off time before camera trigger in seconds/10 (0 means no cut-off):** Set MSX strength
- **Value:** 0 thru 100, inclusive

**203 MAV_CMD_DO_DIGICAM_CONTROL**
- **Command on-board camera:** Adjust FLIR camera image parameters

**Mission Param #1**
- **Type:** float
- **Session control e.g. show/hide lens:** Command FFC
- **Value:**
  - 1 = Command FFC

**Mission Param #2**
- **Type:** float
- **Zoom's absolute position:** Set eZoom level
- **Value:** Ignored

**Mission Param #3**
- **Type:** float
- **Zooming step value to offset zoom from the current position:** Ignored

**Mission Param #4**
- **Type:** float
- **Focus Locking, Unlocking or Re-locking:** Set Scene image tuning parameter 1
- **Value:** Ignored

**Mission Param #5**
- **Type:** float
- **Shooting Command:** Set Scene image tuning parameter 2
- **Value:** Ignored

**Mission Param #6**
- **Type:** float
- **Command Identity:** Set Scene image tuning parameter 3
- **Value:** Ignored

**Mission Param #7**
- **Type:** float
- **Empty:**

---

**2000 MAV_CMD_IMAGE_STAR T_CAPTURE**
- **Start image capture sequence:** Command Still image capture

**Mission Param #1**
- **Type:** float
- **Duration between two consecutive pictures (in seconds):** Interval between captures
- **Value:**
  - 0 through 60, inclusive

**Mission Param #2**
- **Type:** float
- **Number of images to capture total - 0 for unlimited capture:** Ignored

**Mission Param #3**
- **Type:** float
- **Resolution in megapixels (0.3 for 640x480, 1.3 for 1280x720, etc):**
  - FFF = FLIR File Format, also called Radiometric JPEG
  - 1 = JPEG & TIFF
  - 2 = FFF

**2001 MAV_CMD_IMAGE_STOP_CAPTURE**
- **Stop image capture sequence:** Command Still stop capture

**Mission Param #1**
- **Type:** float
- **Reserved:**
- **Value:** Ignored

---

Information on this page is subject to change without notice.
### FLIR Duo User Guide

<table>
<thead>
<tr>
<th>Mission Param #2</th>
<th>Type</th>
<th>Description</th>
<th>Ignored</th>
<th>Ignored</th>
</tr>
</thead>
<tbody>
<tr>
<td>2500 MAV_CMD_VIDEO_START_CAPTURE</td>
<td>float</td>
<td>Starts video capture</td>
<td>Command Video capture</td>
<td></td>
</tr>
<tr>
<td>Mission Param #1</td>
<td>float</td>
<td>Camera ID (0 for all cameras, 1 for first, 2 for second, etc.)</td>
<td>Standard</td>
<td>Standard</td>
</tr>
<tr>
<td>Mission Param #2</td>
<td>float</td>
<td>Frames per second</td>
<td>Set video file type</td>
<td>1 = H264</td>
</tr>
<tr>
<td>Mission Param #3</td>
<td>float</td>
<td>Resolution in megapixels (0.3 for 640x480, 1.3 for 1280x720, etc)</td>
<td>Standard</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mission Param #2</th>
<th>Type</th>
<th>Description</th>
<th>Ignored</th>
<th>Ignored</th>
</tr>
</thead>
<tbody>
<tr>
<td>2501 MAV_CMD_VIDEO_STOP_CAPTURE</td>
<td>float</td>
<td>Stop the current video capture</td>
<td>Command Video stop capture</td>
<td></td>
</tr>
<tr>
<td>Mission Param #1</td>
<td>float</td>
<td>Reserved</td>
<td>Ignored</td>
<td>Ignored</td>
</tr>
<tr>
<td>Mission Param #2</td>
<td>float</td>
<td>Reserved</td>
<td>Ignored</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mission Param #2</th>
<th>Type</th>
<th>Description</th>
<th>Ignored</th>
<th>Ignored</th>
</tr>
</thead>
<tbody>
<tr>
<td>31010 MAV_CMD_USER_1</td>
<td>float</td>
<td>User defined command</td>
<td>Set radiometric parameters</td>
<td></td>
</tr>
<tr>
<td>Mission Param #1</td>
<td>float</td>
<td>User Defined</td>
<td>Set temperature unit</td>
<td>0 = C</td>
</tr>
<tr>
<td>Mission Param #2</td>
<td>float</td>
<td>User Defined</td>
<td>Set OSD temperature meter</td>
<td>0 = OFF</td>
</tr>
<tr>
<td>Mission Param #3</td>
<td>float</td>
<td>User Defined</td>
<td>Set Subject Emissivity</td>
<td>50 to 100, inclusive</td>
</tr>
<tr>
<td>Mission Param #4</td>
<td>float</td>
<td>User Defined</td>
<td>Set Sky Condition</td>
<td>0 to 100, inclusive</td>
</tr>
<tr>
<td>Mission Param #5</td>
<td>float</td>
<td>User Defined</td>
<td>Set Air Temperature</td>
<td>-50 to 327, inclusive</td>
</tr>
</tbody>
</table>

Information on this page is subject to change without notice.
| Mission Param #6   | float | User Defined | Set Humidity | 0 to 100, inclusive  
|                  |       |             |              | -999 = Ignore       |
| Mission Param #7   | float | User Defined | Set Subject Range | 0 to 2000, inclusive  
|                  |       |             |              | -999 = Ignore       |
Appendix C – TCP/IP Implementation

Data type definition

<table>
<thead>
<tr>
<th>Type identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Integer: int8_t, int16_t, int32_t, int64_t</td>
</tr>
<tr>
<td>N +</td>
<td>Positive integers: uint8_t, uint16_t, uint32_t, uint64_t</td>
</tr>
<tr>
<td>R</td>
<td>Float: float</td>
</tr>
<tr>
<td>C</td>
<td>Single character: char</td>
</tr>
<tr>
<td>S</td>
<td>String: string</td>
</tr>
<tr>
<td>Byte</td>
<td>Number of bytes (0 ~ 255): uchar</td>
</tr>
<tr>
<td>D</td>
<td>Date format string (2017-01-01), Beijing time: string</td>
</tr>
<tr>
<td>T</td>
<td>Time format string (01:01:01), time in Beijing: string</td>
</tr>
<tr>
<td>DT</td>
<td>Date / time format string (2017-01-01 01:01:01), Beijing time: string</td>
</tr>
<tr>
<td>B</td>
<td>Boolean type (0, 1): bool / bool</td>
</tr>
<tr>
<td>IP</td>
<td>IP address format string (192.168.2.208): string</td>
</tr>
<tr>
<td>UK</td>
<td>Not defined</td>
</tr>
</tbody>
</table>

Instruction Format

CSP uses the request / response mode. The request consists of a message header + message body.

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
<th>Instruction body</th>
<th>CRC2 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>N Bytes</td>
<td>2 Bytes</td>
</tr>
</tbody>
</table>

- **ID**: default is 0x64
- **Status code**: response status, request message can ignore the field. Is 0 for success, other representations error number
- **Instruction Number**: Each instruction has a unique number
- **Instruction length**: the length of the instruction body. If there is no instruction body, the length value is 0
**FLIR Duo User Guide**

- **CRC1 / Reserved**: Reserved field. In CSP-UART applications, for CRC check (from the beginning of the code, the checksum is calculated using the CRC16-CCITT standard in bytes 1 to 6); this field can be ignored in CSP-TCP applications.
- **message body**: see the next chapter
- **CRC2 / Reserved**: Reserved field. In CSP-UART applications, a CRC check (identification code from the beginning, in bytes 1 ~ N + 8) using CRC16-CCITT standard checksum calculation; in CSP-TCP applications can ignore this field.

[Comment]: CRC16 polynomial = $x^{16} + x^{12} + x^{5} + 1$

### Status Code

<table>
<thead>
<tr>
<th>status code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>success</td>
</tr>
<tr>
<td>0x02</td>
<td>Operation error</td>
</tr>
<tr>
<td>0x03</td>
<td>The instruction number is incorrect</td>
</tr>
<tr>
<td>0x04</td>
<td>Parameter is illegal</td>
</tr>
<tr>
<td>0x05</td>
<td>CRC1 check code error</td>
</tr>
<tr>
<td>0x06</td>
<td>CRC2 check code error</td>
</tr>
<tr>
<td>0x07</td>
<td>The file could not be found</td>
</tr>
<tr>
<td>0x08</td>
<td>The device is busy and can not respond</td>
</tr>
</tbody>
</table>

### Network Configuration

<table>
<thead>
<tr>
<th>Network Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address (USB)</td>
<td>192.168.10.19</td>
</tr>
<tr>
<td>Control command port</td>
<td>6000, TCP short connection for camera control and parameter configuration, and other messages</td>
</tr>
<tr>
<td>Event port</td>
<td>6002, TCP long connection for camera event notification</td>
</tr>
</tbody>
</table>

【Note】: Equipment as a server, network communication are used TCP request - response mode.
Camera Control

3.1 Heartbeat
Heartbeat function, used to detect the camera online status, the proposed cycle of 5 ~ 10s.

Request instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- \rightarrow</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x 0000</td>
<td>0x00</td>
</tr>
<tr>
<td>Instruction body</td>
<td>0 Bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>air</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CRC Description:
In CSP-TCP applications, the CRC is ignored

Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- \rightarrow</th>
<th>G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x0001</td>
<td>0x00</td>
</tr>
<tr>
<td>Instruction body</td>
<td>0 Bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>air</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CRC Description:
In CSP-TCP applications, the CRC is ignored
3.2 Time synchronization
Time synchronization for other systems to synchronize the camera time.

Request instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- →</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x0002</td>
<td>0x08</td>
</tr>
</tbody>
</table>

Instruction body
8 Bytes

Instruction body format:
Struct {
  String strDateTime [20]; // DT time format, such as 2017-01-01 01:01
};

CRC Description:
In CSP-TCP applications, the CRC is ignored

Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- →</th>
<th>G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x0003</td>
<td>0x00</td>
</tr>
</tbody>
</table>

Instruction body
0 Bytes

CRC Description:
In CSP-TCP applications, the CRC is ignored

Information on this page is subject to change without notice
### 3.3 GPS information

Set the camera GPS information for the camera picture metadata to fill.

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x0004</td>
<td>0x0C</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body format:**

```c
Struct {
    Int32_t s32 Longitude; // Longitude * 1E7
    Int32_t s32 Latitude; // latitude * 1E7
    Int32_t s32 Altitude; // absolute elevation * 1000
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

**Response command:**

no

### 3.4 Image display mode

#### 3.4.1 query

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x1000</td>
<td>0x00</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body**

0 Bytes

**CRC2 / reserved**

---

Information on this page is subject to change without notice
CRC Description:
In CSP-TCP applications, the CRC is ignored

### Data flow:

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1001</td>
<td>0x04</td>
<td>CRC</td>
<td>CRC</td>
</tr>
</tbody>
</table>

Instruction body format:

```c
Struct {
    Uint8_t u8DisplayMode; // image display format,
    Byte Reserve [3]; // Reserved
};
```

**VIDEO_DISPLAY_MODE Definition:**

```c
typedef enum {
    VIDEO_DISPLAY_PIC_IN_PIC = 0,
    VIDEO_DISPLAY_VISIBLE = 1,
    VIDEO_DISPLAY_IR = 2,
} VIDEO_DISPLAY_MODE;
```

CRC Description:
In CSP-TCP applications, the CRC is ignored

### 3.4.2 settings

Request instruction

---

*Information on this page is subject to change without notice*
### Instruction Application:

**UART / TCP**

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x1002</td>
<td>0x04</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body 4 Bytes**

Instruction body format:

```c
struct {
    uint8_t u8DisplayMode; // image display format,
    byte Reserve [3]; // Reserved
};
```

**VIDEO_DISPLAY_MODE Definition:**

```c
typedef enum {
    VIDEO_DISPLAY_PIC_IN_PIC = 0,
    VIDEO_DISPLAY_VISIBLE = 1,
    VIDEO_DISPLAY_IR = 2,
} VIDEO_DISPLAY_MODE;
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

### Response command:

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1003</td>
<td>0x00</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body 0 Bytes**

CRC2 / reserved

---

436-0100-01-10, Rev. 200  Duo User Guide

Information on this page is subject to change without notice
3.5 Recording status
3.5.1 query
   Request instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x1004</td>
</tr>
</tbody>
</table>

**Instruction body**
0 Bytes

| Description: |
| In CSP-TCP applications, the CRC is ignored |

**Response command:**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1005</td>
</tr>
</tbody>
</table>

**Instruction body**
4 Bytes

**Instruction body format:**

```
Struct {
    CRC
}
```
3.6 IR image color
3.6.1 query
Request instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- CS</th>
<th>CRC Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x1008</td>
<td>0x00</td>
</tr>
</tbody>
</table>

Instruction body 0 Bytes

CRC Description:
In CSP-TCP applications, the CRC is ignored

Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- G / A</th>
<th>CRC Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1009</td>
<td>0x04</td>
</tr>
</tbody>
</table>

Instruction body 4 Bytes

Instruction body format: CRC

Information on this page is subject to change without notice
**IR_COLOR Definition:**

Typedef enum {
    IR_COLOR_WHITEHOT = 0,
    IR_COLOR_HOTMETAL = 1,
    IR_COLOR_RAINBOW = 2,
} IR_COLOR;

CRC Description:
In CSP-TCP applications, the CRC is ignored

### 3.6.2 settings

**Request instruction**

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x100A</td>
<td>0x04</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body format:**

Struct {
    Uint8_t u8IRColor; // IR color, see IR_COLOR definition
    Byte Reserve [3]; // Reserved
};

**IR_COLOR Definition:**

Typedef enum {
    IR_COLOR_WHITEHOT = 0,
    IR_COLOR_HOTMETAL = 1,
    IR_COLOR_RAINBOW = 2,
} IR_COLOR;

CRC Description:
In CSP-TCP applications, the CRC is ignored

### Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x100B</td>
</tr>
<tr>
<td>Instruction body</td>
<td>0 Bytes</td>
<td>CRC2 / reserved 2 Bytes</td>
</tr>
<tr>
<td>air</td>
<td></td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Description:**
In CSP-TCP applications, the CRC is ignored

### 3.7 HDMI output resolution

#### 3.7.1 query

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x100C</td>
</tr>
<tr>
<td>Instruction body</td>
<td>0 Bytes</td>
<td>CRC2 / reserved 2 Bytes</td>
</tr>
<tr>
<td>air</td>
<td></td>
<td>CRC</td>
</tr>
</tbody>
</table>

**CRC Description:**
In CSP-TCP applications, the CRC is ignored

### Response command:

---

Information on this page is subject to change without notice

436-0100-01-10, Rev. 200  Duo User Guide

The World's Sixth Sense™
### Instruction Application:

**UART / TCP**

Data flow: CS --- [ ] G / A

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
</tr>
</tbody>
</table>

Instruction body format:

```c
struct {
    Ureal8_t u8Resolution; // output resolution, see HDMI_RESOLUTION definition
    Byte Reserve [3]; // Reserved
};
```

**HDMI_RESOLUTION Definition:**

```c
typedef enum {
    HDMI_RESOLUTION_720P30 = 0,
    HDMI_RESOLUTION_720P60 = 1,
    HDMI_RESOLUTION_1080P30 = 2,
    HDMI_RESOLUTION_1080P60 = 3,
} HDMI_RESOLUTION;
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

### 3.7.2 settings Request instruction

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
</tr>
</tbody>
</table>

Instruction body format:

Instruction body format:

```c
struct {
    Ureal8_t u8Resolution; // output resolution, see HDMI_RESOLUTION definition
    Byte Reserve [3]; // Reserved
};
```
**FLIR Duo User Guide**

```c
typedef enum {
    HDMI_RESOLUTION_720P30 = 0,
    HDMI_RESOLUTION_720P60 = 1,
    HDMI_RESOLUTION_1080P30 = 2,
    HDMI_RESOLUTION_1080P60 = 3,
} HDMI_RESOLUTION;
```

**CRC Description**: In CSP-TCP applications, the CRC is ignored.

### Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x100F</td>
</tr>
</tbody>
</table>

**Instruction body**: 0 Bytes

**CRC Description**: In CSP-TCP applications, the CRC is ignored.

#### 3.8 MSX edge enhancement

**3.8.1 query**

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
</tbody>
</table>

---

436-0100-01-10, Rev. 200 Duo User Guide

Information on this page is subject to change without notice.
<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1011</td>
<td>0x04</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body format**:
```c
Struct {
    BOOL bEnable; // the MSX enabled, 1- open, 0- off
    UX8S trength; // MSX strength, 0 ~ 100
    Int8_t s8PosX; // MSX X coordinates, -100 ~ 100
    Int8_t s8PosY; // MSX Y coordinates, -8 ~ 8
};
```

**CRC Description**:
In CSP-TCP applications, the CRC is ignored.

**3.8.2 settings**

**Request instruction**

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
</tr>
</thead>
</table>

**CRC Description**:
In CSP-TCP applications, the CRC is ignored.
**Information on this page is subject to change without notice**
**FLIR Duo User Guide**

**Instruction body format:**

```cpp
struct {
    uint8_t u8SDcardNo; // SD card number, from 1 to start
    byte Reserve [3]; // Reserved
} ;
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

---

**Response command:**

**Instruction body format:**

```cpp
struct {
    uint8_t u8SDcardNo; // SD card number, from 1 to start
    byte Reserve [3]; // Reserved
    uint32_t u32Total; // total capacity in megabytes
    uint32_t u32Free; // remaining capacity in megabytes
} ;
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.
### 3.9.2 Formatting

Format the specified SD card.

#### Request Instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A ---</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x1016</td>
<td>0x04</td>
</tr>
</tbody>
</table>

**Instruction body format:**

```c
struct {
    uint8_t u8SDcardNo; // SD card number, from 1 to start
    Byte Reserve [3]; // Reserved
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

#### Response Command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS ---</th>
<th>G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1017</td>
<td>0x00</td>
</tr>
</tbody>
</table>

**Instruction body:**

```c
struct {
    Byte Reserve [1]; // Reserved
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.
In CSP-TCP applications, the CRC is ignored

### 3.10 LED enabled

#### 3.10.1 query

**Request instruction**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x101A</td>
</tr>
</tbody>
</table>

**Instruction body**

- 0 Bytes

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

**Response command:**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x101B</td>
</tr>
</tbody>
</table>

**Instruction body format:**

```c
struct {
    BOOL  bEnable; // Close 1 - opening
    Byte Reserve [3]; // Reserved
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored
### 3.10.2 settings

#### Request instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x101C</td>
</tr>
</tbody>
</table>

**Instruction body format:**

```c
struct {
    BOOL bEnable; 0- // Close 1- opening
    Byte Reserve [3]; // Reserved
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

#### Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x101D</td>
</tr>
</tbody>
</table>

**Instruction body:**

- 0 Bytes

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

---

3.11 TONES enabled

---

Information on this page is subject to change without notice
### 3.11.1 query

**Requests instruction**

<table>
<thead>
<tr>
<th>Instruction Application: UART / TCP</th>
<th>Data flow: G / A --- → CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
</tr>
</tbody>
</table>

**Instruction body**
- 0 Bytes
- **CRC Description:**
  - In CSP-TCP applications, the CRC is ignored

**Response command:**

<table>
<thead>
<tr>
<th>Instruction Application: UART / TCP</th>
<th>Data flow: CS --- → G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
</tr>
</tbody>
</table>

**Instruction body format:**

```c
struct {
    BOOL bEnable; // Close 1 - opening
    Byte Reserve [3]; // Reserved
};
```

**CRC Description:**
- In CSP-TCP applications, the CRC is ignored

### 3.11.2 settings

**Requests instruction**
## Instruction Application: UART / TCP

### Data flow: G / A --- → CS

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x1020</td>
<td>0x04</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body**

**4 Bytes**

### Instruction body format:

```c
Struct {
    BOOL bEnable; // Close 1 - opening
    Byte Reserve [3]; // Reserved
};
```

**CRC Description**: In CSP-TCP applications, the CRC is ignored

---

### Response command:

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1021</td>
<td>0x00</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body**

**0 Bytes**

### CRC Description:

In CSP-TCP applications, the CRC is ignored

---

3.12 PIP position
3.12.1 query

Request instruction
### Instruction Application: UART / TCP

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x1022</td>
<td>0x00</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body 0 Bytes**

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

### Response command:

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1023</td>
<td>0x04</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body 4 Bytes**

**Instruction body format:**

```c
typedef enum {
    LEFT_UP = 0,
    MIDDLE_UP,
    RIGHT_UP,
    LEFT_MIDDLE,
    CENTRAL,
} PIP_POSITION;
```

**PIP_POSITION Definition:**

```c
struct {
    u8P uint8_t position; // see PIP_POSITION defined
    Byte Reserve [3]; // Reserved
};
```
3.12.2 settings
Request instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A ---→ CS</th>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
<th>CRC2 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0x64</td>
<td>0x00</td>
<td>0x1024</td>
<td>0x04</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>

Instruction body format:
Struct {
    u8P uint8_t position; // see PIP_POSITION defined
    Byte Reserve [3]; // Reserved
};

PIP_POSITION Definition:
typedef enum {
    LEFT_UP = 0,
    MIDDLE_UP,
    RIGHT_UP,
    LEFT_MID,
    CENTRAL,
    RIGHT_MID,
    LEFT_DOWN,
    MIDDLE_DOWN,
    RIGHT_DOWN
} PIP_POSITION;

CRC Description:
In CSP-TCP applications, the CRC is ignored.
**CRC Description:**
In CSP-TCP applications, the CRC is ignored

**Response command:**

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
<th>CRC2 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1025</td>
<td>0x00</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>

**Instruction body**
0 Bytes

**CRC Description:**
In CSP-TCP applications, the CRC is ignored

---

### 3.13 storage stream format

#### 3.13.1 query

**Request instruction**

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
<th>CRC2 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x1026</td>
<td>0x00</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>

**Instruction body**
0 Bytes

**CRC Description:**
In CSP-TCP applications, the CRC is ignored
### Response command:

<table>
<thead>
<tr>
<th>Instruction code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1027</td>
<td>0x04</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body format:**

```c
Struct {
    Uint8_t u8StoreStream; // see STORE_STREAM definition
    Byte Reserve [3]; // Reserved
};
```

**STORE_STREAM Definition:**

```c
typedef enum {
    VISIBLE_ONLY = 0,
    IR_ONLY,
    IR_AND_VISIBLE,
} STORE_STREAM;
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

### Request instruction

<table>
<thead>
<tr>
<th>Instruction code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x1028</td>
<td>0x04</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body format:**

```c
Struct {
    uint8_t u8StoreStream; // see STORE_STREAM definition
    uint8_t u8Reserved [3]; // Reserved
};
```

**CRC**

3.13.2 settings

**Request instruction**
Struct {
    Uint8_t u8StoreStream; // see STORE_STREAM definition
    Byte Reserve [3]; // Reserved
};

STORE_STREAM Definition:
Typedef enum {
    VISIBLE_ONLY = 0,
    IR_ONLY,
    IR_AND_VISIBLE,
} STORE_STREAM;

CRC Description:
In CSP-TCP applications, the CRC is ignored

Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- → G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x1029</td>
</tr>
</tbody>
</table>

Instruction body
0 Bytes

<table>
<thead>
<tr>
<th>CRC Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In CSP-TCP applications, the CRC is ignored</td>
</tr>
</tbody>
</table>

3.14 take pictures
3.14.1 start
Take pictures start, take pictures at the end, you can specify a single shot or timer shot, if it is time to shoot can specify the camera interval, in seconds.

Request instruction
### Instruction Application: UART / TCP

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 2 Bytes</th>
<th>CRC2 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x2000</td>
<td>0x08</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Instruction body format:**

```c
struct {
    uint8_t capture; // capture interval, 0 ~ 60s; 0 refers to a single capture
    uint8_t u8Format; // file format, 0-RJPEG, 1-JPEG & TIFF
    uint32_t u32ID; // ID unique
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

### Response command:

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 2 Bytes</th>
<th>CRC2 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x2001</td>
<td>0x04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Instruction body format:**

```c
struct {
    uint32_t u32ID; // ID unique
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

---

Information on this page is subject to change without notice
**3.14.2 stop**

Take pictures start, take pictures at the end, you can specify a single shot or timer shot, if it is time to shoot can specify the camera interval, in seconds.

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- G</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x2002</td>
<td>0x04</td>
</tr>
</tbody>
</table>

**Instruction body**

4 Bytes

Instruction body format:

```c
struct {
    uint32_t u32ID; // ID unique, and should begin taking pictures with ID consistent
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

**Response command:**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x2003</td>
</tr>
</tbody>
</table>

**Instruction body**

0 Bytes

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.
### 3.15 Video

#### 3.15.1 start

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- (\rightarrow) CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x2004</td>
</tr>
</tbody>
</table>

**Instruction body**

8 Bytes

Instruction body format:

```c
Struct {
    Uint8_t u8Format; // file format, 0-H264, 1- tIFF
    Byte Reserve [3]; // Reserved
    ui nt32_t u32ID; // ID unique
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

---

**Response command:**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- (\rightarrow) G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x2005</td>
</tr>
</tbody>
</table>

**Instruction body**

4 Bytes

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

---

Information on this page is subject to change without notice
### 3.15.2 stop Request instruction

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x2006</td>
<td>0x04</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body format:**
```c
struct {
    uint32_t u32ID; // ID unique, and should start with the video ID is consistent
};
```

**CRC Description:**
In CSP-TCP applications, the CRC is ignored.

### Response command:

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x2007</td>
<td>0x00</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body format:**
```c
struct {
    uint32_t u32ID; // ID unique
};
```

**CRC Description:**
In CSP-TCP applications, the CRC is ignored.
### CRC Description:
In CSP-TCP applications, the CRC is ignored.

#### 3.16 capture interval

##### 3.16.1 query

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- → CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x2008</td>
</tr>
</tbody>
</table>

**Instruction body**
0 Bytes

<table>
<thead>
<tr>
<th>CRC Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In CSP-TCP applications, the CRC is ignored</td>
</tr>
</tbody>
</table>

**Response command**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- → G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x2009</td>
</tr>
</tbody>
</table>

**Instruction body**
4 Bytes

**Instruction body format:**
```c
struct {
    uint8_t, use the setting; // capture interval, 0 ~ 60s; 0 refers to a single capture
```
CRC Description:
In CSP-TCP applications, the CRC is ignored.

### 3.16.2 settings
#### Request instruction

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>2 Bytes reserved</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x200A</td>
<td>0x04</td>
<td></td>
</tr>
</tbody>
</table>

Instruction body:
Struct {
    U8i uint8_t, use the setting; // capture interval, 0 ~ 60s; 0-refer to a single capture
    Byte Reserve [3]; // Reserved
}

CRC Description:
In CSP-TCP applications, the CRC is ignored.

### Response command:

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>2 Bytes reserved</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x200B</td>
<td>0x00</td>
<td></td>
</tr>
</tbody>
</table>

Instruction body:
0 Bytes

CRC Description:
In CSP-TCP applications, the CRC is ignored.
## CRC Description:

In CSP-TCP applications, the CRC is ignored

### 3.17 Capture file format

#### 3.17.1 query

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction TCP</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- =&gt; CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x200C</td>
</tr>
</tbody>
</table>

**Instruction body**

0 Bytes

<table>
<thead>
<tr>
<th>Instruction body</th>
<th>0 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>air</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

#### Response command:

<table>
<thead>
<tr>
<th>Instruction TCP</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- =&gt; G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x200D</td>
</tr>
</tbody>
</table>

**Instruction body**

4 Bytes

<table>
<thead>
<tr>
<th>Instruction body</th>
<th>4 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>air</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body format**

```c
Struct {
    Uint8_t u8Format; // file format, 0-RJPEG , 1-JPEG & TIFF
    Byte Reserve [3]; // Reserved
} CRC
```
### FLIR Duo User Guide

#### 3.17.2 Settings

**Request Instruction**

| Instruction | Application: UART / TCP | Data flow: G / A --- CS | CRC1 | CRC2/
---|---|---|---|---
| **Identification code** | **status code** | **Instruction number** | **Instruction length** | **reserved** |
| 1 Byte | 1 Byte | 2 Bytes | 2 Bytes | 2 Bytes |
| 0x64 | 0x00 | 0x200E | 0x04 | CRC |

**Instruction body format**

```c
Struct {
    Uint8_t u8Format; // file format, 0-RJPEG , 1-JPEG & TIFF
    Byte Reserve[3]; // Reserved
};
```

**CRC Description**

In CSP-TCP applications, the CRC is ignored.

#### Response Command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identification code</strong></td>
<td><strong>status code</strong></td>
<td><strong>Instruction number</strong></td>
</tr>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x200 F</td>
</tr>
</tbody>
</table>

**Instruction body**

```c
0 Bytes
```

**CRC Description**

In CSP-TCP applications, the CRC is ignored.
## FLIR Duo User Guide

In CSP-TCP applications, the CRC is ignored

### 3.18 video file format

#### 3.18.1 query

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x2010</td>
<td>0x00</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body**

<table>
<thead>
<tr>
<th>0 Bytes</th>
<th>CRC2 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>air</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

### Response command:

<table>
<thead>
<tr>
<th>Instruction code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x2011</td>
<td>0x04</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body**

<table>
<thead>
<tr>
<th>4 Bytes</th>
<th>CRC2 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body format**

```c
struct {
    Uint8_t u8Format; // file format, 0-H264, 1-.tiff
    Byte Reserve [3]; // Reserved
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

---

Information on this page is subject to change without notice
### 3.18.2 settings

#### Request instruction

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x2012</td>
<td>0x04</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>

**Instruction body format**
```
struct {
    uint8_t u8Format; // file format, 0-H264 , 1- tIFF
    byte Reserve [3]; // Reserved
};
```

**CRC Description**
In CSP-TCP applications, the CRC is ignored

#### Response command:

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x2013</td>
<td>0x00</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>

**Instruction body**

<table>
<thead>
<tr>
<th>0 Bytes</th>
</tr>
</thead>
</table>

**CRC Description**
In CSP-TCP applications, the CRC is ignored
3.19 firmware

3.19.1 Firmware information query

Check the current firmware version of the device and the firmware version that can be upgraded in the SD card.

Request instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A --- CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x3000</td>
</tr>
</tbody>
</table>

Instruction body 0 Bytes

CRC Description:
In CSP-TCP applications, the CRC is ignored

Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x3001</td>
</tr>
</tbody>
</table>

Instruction body 128 Bytes

Instruction body format:

```c
Struct {
    String strCurFirmware [64]; // device current firmware version information
    String strUpgFirmware [64]; // Upgradeable firmware name
};
```

CRC Description:
In CSP-TCP applications, the CRC is ignored

3.19.2 Firmware local upgrade
## FLIR Duo User Guide

### Request instruction

<table>
<thead>
<tr>
<th>Instruction Application: UART / TCP</th>
<th>Data flow: G / A → CS</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
<td></td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
<td></td>
</tr>
<tr>
<td>0x3002</td>
<td>0x40</td>
<td></td>
</tr>
<tr>
<td>Instruction body 64 Bytes</td>
<td>CRC1 / reserved 2 Bytes</td>
<td></td>
</tr>
</tbody>
</table>

Instruction body format:
```c
Struct {
    String strUpgFirmware [64]; // the firmware name to be upgraded
};
```

**CRC Description:**
In CSP-TCP applications, the CRC is ignored

### Response command:

<table>
<thead>
<tr>
<th>Instruction Application: UART / TCP</th>
<th>Data flow: CS → G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
</tr>
<tr>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x3003</td>
<td>0x00</td>
</tr>
<tr>
<td>Instruction body 0 Bytes</td>
<td>CRC1 / reserved 2 Bytes</td>
</tr>
</tbody>
</table>

**CRC Description:**
In CSP-TCP applications, the CRC is ignored

### 3.19.3 Firmware Remote Upgrade
Remote upgrade the device.

**Request instruction**
### Instruction Application: UART / TCP

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>bytes</td>
<td>bytes</td>
</tr>
</tbody>
</table>

*Data flow: G / A --- \(\rightarrow\) CS*

**Instruction body**

- **1 Byte**
- **2 Bytes**
- **2 Bytes**

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

**Response command:**

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>bytes</td>
<td>bytes</td>
</tr>
</tbody>
</table>

*Data flow: CS --- \(\rightarrow\) G / A*

**Instruction body**

- **0 Bytes**

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

**data transmission:**

The data is transmitted in multiple slices.
3.20 equipment
3.20.1 Device model
Check the device model.
Request instruction

Instruction Application: UART / TCP
Data flow: G / A → CS

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x3006</td>
<td>4 + n</td>
<td>CRC</td>
</tr>
</tbody>
</table>

Instruction body
4 + n Bytes

Instruction body format:
Struct {
    uint32_t u32Size; // data block size
    Uint16_t u16TotalSlices; // total number of slices
    uint16_t u16Seq; // packet number, numbered from a beginning
    Byte Data[n]; // data, length n, n = u32Size
};

CRC Description:
In CSP-TCP applications, the CRC is ignored

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x4000</td>
<td>0x00</td>
<td>CRC</td>
</tr>
</tbody>
</table>

Instruction body
0 Bytes

air

CRC Description:
In CSP-TCP applications, the CRC is ignored
### Response command:

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td></td>
<td>0x4001</td>
<td>0x2C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Instruction body**: 44 bytes

Instruction body format:
```
Struct {
    Negative temperature type, 1 - temperature type
    Byte    Reserve [3]; // Reserved
    String strPN [20]; // part no
    String strSN [20]; // serial no
};
```

**CRC Description**: In CSP-TCP applications, the CRC is ignored

### 3.20.2 Restore factory settings

**Request instruction**

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x4002</td>
<td>0x00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Instruction body**: 0 bytes

**CRC Description**: In CSP-TCP applications, the CRC is ignored
### Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS → G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x4003</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction body</th>
<th>0 Bytes</th>
<th>CRC2 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>air</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>

**CRC Description:**
In CSP-TCP applications, the CRC is ignored

### 3.20.3 Restart the device

#### Request instruction

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: G / A → CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0x4004</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instruction body</th>
<th>0 Bytes</th>
<th>CRC2 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>air</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>

**CRC Description:**
In CSP-TCP applications, the CRC is ignored

#### Response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS → G / A</th>
</tr>
</thead>
</table>
## Identification code

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x4005</td>
<td>0x00</td>
<td>CRC</td>
</tr>
</tbody>
</table>

### Instruction body

- 0 Bytes

<table>
<thead>
<tr>
<th>CRC2 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC</td>
</tr>
</tbody>
</table>

### CRC Description:

In CSP-TCP applications, the CRC is ignored

### 3.21 temperature measurement

#### 3.21.1 Radiation Coefficient

**Query request command:**

<table>
<thead>
<tr>
<th>Instruction Application: UART / TCP</th>
<th>Data flow: G / A --- ➔ CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
</tr>
<tr>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
</tr>
<tr>
<td>0x5000</td>
<td>0x00</td>
</tr>
</tbody>
</table>

### Instruction body

- 0 Bytes

<table>
<thead>
<tr>
<th>CRC2 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC</td>
</tr>
</tbody>
</table>

### CRC Description:

In CSP-TCP applications, the CRC is ignored

**Query response command:**

<table>
<thead>
<tr>
<th>Instruction Application: UART / TCP</th>
<th>Data flow: CS --- ➔ G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
</tr>
<tr>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
</tr>
<tr>
<td>0x5001</td>
<td>0x08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC</td>
</tr>
</tbody>
</table>

Information on this page is subject to change without notice
Instruction body format:
Struct {
    Uint8_t u8 TempUnit; // temperature unit, 0- ° C, 1 ° F
    BOOL   bSpotMeter; // measurement points enabled, 0- closed, 1- opening
    Uiss8_t u8 Emissivity; // emissivity, range 50 to 100
    Int8_t s8 AirTemp; // atmospheric temperature, range from -50 to 127
    U8 uint8_t SkyCond; // sky conditions, see SKY_CONDITION defined
    U8 uint8_t Humidity viewer; // humidity range of 0 to 100
    U16 t u16 SubjectDistance; // target distance, range 0 ~ 2000 meter
};

SKY_CONDITION Definition:
Typedef enum {
    CLEAR_SKIES = 0,
    SCATTERED_SKIES = 25,
    CLOUDY_SKIES = 75,
} SKY_CONDITION;

CRC Description:
In CSP-TCP applications, the CRC is ignored.

Set request command:

<table>
<thead>
<tr>
<th>Instruction Application: UART / TCP</th>
<th>Data flow: G / A --- CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
</tr>
<tr>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
</tr>
<tr>
<td>0x5002</td>
<td>0x08</td>
</tr>
<tr>
<td>Instruction body 8 Bytes</td>
<td>CRC</td>
</tr>
</tbody>
</table>

Information on this page is subject to change without notice.
UTF8_t u8 TempUnit;  // temperature unit, 0° C, 1° F
BOOL bSpotMeter;  // measurement points enabled, 0- closed, 1- opening
Uiss8_t u8 Emissivity;  // emissivity, range 50 to 100
Int8_t s8 AirTemp;  // Atmospheric temperature, range from -50 to 127
U8 uint8_t SkyCond;  // sky conditions, see SKY_CONDITION defined
U8 uint8_t Humidity viewer;  // humidity range of 0 to 100
Uint16_t u16 SubjectDistance;  // target distance, range 0 ~ 2000 meter

};

SKY_CONDITION Definition:
TypeDef enum {
    CLEAR_SKIES = 0,
    SCATTERED_SKIES = 25,
    CLOUDY_SKIES = 75,
} SKY_CONDITION;

CRC Description:
In CSP-TCP applications, the CRC is ignored

Set the response command:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Application: UART / TCP</th>
<th>Data flow: CS --- → G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0x5003</td>
</tr>
</tbody>
</table>

Instruction body
0 Bytes

CRC Description:
In CSP-TCP applications, the CRC is ignored

The first 4 chapters event notification

4.1 Video or capture notification
The device generates a new capture or video file, a push event notification, which contains the ID of the camera that triggered the camera or the ID of the recording start command and the path of the generated picture.

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xE000</td>
<td>0x08</td>
<td>0x64</td>
<td>0xE000</td>
</tr>
</tbody>
</table>

**Instruction body**

8 Bytes

Instruction body format:

```c
typedef struct {
  uint8_t U; // 8-video file, 1-still image
  uint8_t U8_ErrorCode; // 0-success, 1-fails
  Uint8_t reserve[2];
  Uint32_t u32ID;
  Uint8_t strPath[64];
} SendRecordEventNotifyParam;
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

**Response command:**

no

**4.2 Firmware upgrade result notification**

Request instruction

<table>
<thead>
<tr>
<th>Instruction code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xE002</td>
<td>0x14</td>
<td>0x64</td>
<td>0xE002</td>
</tr>
</tbody>
</table>

**Instruction body**

20 Bytes

Information on this page is subject to change without notice
Instruction body format:
Struct {
    ui NT 32_t u32Errorcode; // update error codes, see UPGRADE_ERRORCODE defined
};

UPGRADE_ERRORCODE Definition:
Typedef enum {
    UPGRADE_SUCCESS = 0,
    UPGRADE_FILE_NO_EXIST = 1,
    UPGRADE_CRC_ERROR = 2,
    UPGRADE_VERSION_ERROR = 3,
    UPGRADE_FWSIZE_ERROR = 4,
    UPGRADE_FAIL = 5,
} UPGRADE_ERRORCODE;

CRC Description:
In CSP-TCP applications, the CRC is ignored

4.3 SD card formatting result notification
Request instruction

Instruction Application: UART / TCP
Data flow: CS --- $\rightarrow$ G / A

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
<th>CRC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xE00 4</td>
<td>0x14</td>
<td>CRC</td>
<td>CRC</td>
</tr>
</tbody>
</table>

Texture:

Instruction body format:
Struct {
    ui NT 32_t u32Errorcode; // 0- successful, non-0- unsuccessful
}

CRC Description:
### 4.4 Recording or capturing stop notification

**Request instruction**

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 (1/2) Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xE006</td>
<td>0x14</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body**

20 Bytes

**Instruction body format:**

```c
typedef struct {
    Uint32_t u32ID;
} SendRecord Stop EventNotifyParam;
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

### 4.5 Recording or capturing start notification

**Request instruction**

<table>
<thead>
<tr>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>CRC1 (1/2) Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xE008</td>
<td>0x14</td>
<td>CRC</td>
</tr>
</tbody>
</table>

**Instruction body**

20 Bytes

**Instruction body format:**

```c
typedef struct {
    Uint32_t u32ID;
} SendRecord Start EventNotifyParam;
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.
5.1 file list query

Query the file list by the suffix of the file name.

Request instruction

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xF000</td>
<td>0x0C</td>
<td>0x00</td>
</tr>
</tbody>
</table>

Instruction body format:

```c
struct {
    Uint32_t u32Index;  // file start index
    Uint32_t u32Count;  // number of files
    String strSuffix[32];  // file suffix, support multiple suffixes such as " .jpg |
    .mov ");
}
```

CRC Description:

In CSP-TCP applications, the CRC is ignored

Response command:

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0xF001</td>
<td>8 + 88 * N</td>
<td>0x00</td>
</tr>
</tbody>
</table>

Instruction body format:

```
(8 + 88 * N) Bytes
```

Reserved 2 Bytes

0x00
**FLIR Duo User Guide**

Struct {
    Uint32_t u32Total; // The total number of currently retrieved files
    Uint32_t u32Count; // number of file lists
}

Struct {
    Uint32_t u32Size; // file size
    String strTime [20]; // file creation time
    String strPath [64]; // file path
}

List [n]; // file list, n = u32Count

Struct {
    Uint32_t u32Flag; // 0 - can continue to query, 1 - last file
}

**CRC Description:**
In CSP-TCP applications, the CRC is ignored.

【Remarks】
The device is querying and can not respond to another query event. The response command status code is returned to **0x08 “The device is busy and can not respond”**.

### 5.2 file details query
**Request instruction**

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xF002</td>
<td>0x40</td>
<td>0x00</td>
</tr>
</tbody>
</table>

**Instruction body format:**
Struct {
    String strPath [64]; // file path
}

**CRC Description:**
In CSP-TCP applications, the CRC is ignored.

**Response command:**

<table>
<thead>
<tr>
<th>Instruction Application: TCP</th>
<th>Data flow: G / A --- →</th>
<th>CS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xF002</td>
</tr>
</tbody>
</table>

Instruction body format:
Struct {
    String strPath [64]; // file path
}

**CRC Description:**
In CSP-TCP applications, the CRC is ignored.
Identification code | status code 1 Byte | Instruction number 2 Bytes | Instruction length 2 Bytes | Reserved 2 Bytes
--- | --- | --- | --- | ---
0x64 | 0xF003 | 0x20 | 0x00

Instruction body format:
Struct {
  uint8_t u8Type; // file type, 0 - video, 1 - picture, 3 - other
  uint8_t u8HaveThumb; // This file supports thumbnails
  uint16_t u16ThumbSize; // thumbnail size
  uint16_t u16Width; // video or picture resolution, image width
  uint16_t u16Height; // video or picture resolution, image height
  uint32_t u32Size; // file size
  string strTime[20]; // file creation time
};

5.3 file download
Request instruction

Instruction Application: TCP | Data flow: G / A --- → CS | CRC1 / reserved 2 Bytes
--- | --- | ---
Identification code 1 Byte | status code 1 Byte | Instruction number 2 Bytes | Instruction length 2 Bytes | CRC1 / reserved 2 Bytes
0x64 | 0x00 | 0xF004 | 0x50 | 0x00

Instruction body format:
Struct {
  uint8_t u8Type; // file type, 0 - original file 1 - thumbnail
  Byte Reserve [3]; // Reserved
  uint32_t u32SesID; // file download session ID
  uint32_t u32Offset; // file offset, in bytes
};
436-0100-01-10, Rev. 200 Duo User Guide

Information on this page is subject to change without notice
5.5 file download control
Request instruction

<table>
<thead>
<tr>
<th>Instruction Application: TCP</th>
<th>Data flow: G / A ---→ G / CS</th>
<th>Identification code 1 Byte</th>
<th>status code 1 Byte</th>
<th>Instruction number 2 Bytes</th>
<th>Instruction length 2 Bytes</th>
<th>Reserved 2 Bytes</th>
<th>CRC1 / reserved 2 Bytes</th>
<th>CRC2 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xF008</td>
<td>0x08</td>
<td>0x00</td>
<td></td>
<td></td>
<td>0x00</td>
<td></td>
</tr>
</tbody>
</table>

Instruction body format:

Struct {
    0x00
}

CRC Description:
In CSP-TCP applications, the CRC is ignored
5.6 Specify the path target query

Is the "file list query" the extended instruction. You can specify query the directory, and the path under the folder or file list query.

Request instruction

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>Reserved</th>
<th>CRC1</th>
<th>reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xF00</td>
<td>0x00</td>
<td>0x00</td>
<td>0x00</td>
<td>0x00</td>
</tr>
</tbody>
</table>

Instruction body format:
Struct {
);

Response command:

<table>
<thead>
<tr>
<th>Instruction Application</th>
<th>Data flow</th>
<th>G / A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code</td>
<td>status code</td>
<td>Instruction number</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xF00</td>
</tr>
</tbody>
</table>

Instruction body
0 Bytes

Reserved 2 Bytes
0x00
Uint32_t u32Type;  // target type 0 - folder 1 - file  
Uint32_t u32Index;  // target start index  
Uint32_t u32Count;  // number of targets  
String strPath [32];  // query the directory path, fill in the root directory;  
String strSuffix [32];  // file suffix, support multiple suffixes such as " .jpg | .mov "; u32Type is 0 can be ignored ;  
}

**CRC Description:**
In CSP-TCP applications, the CRC is ignored

**Response command:**

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Byte</td>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0xF00 B</td>
<td>8 + 88 * N</td>
<td>0x00</td>
</tr>
</tbody>
</table>

**Instruction body format:**

Struct {
    Uint32_t u32Total;  // total number of targets  
    Uint32_t u32Count;  // number of target lists  
    Struct {
        Uint32_t u32Size;  // target size, the target for the folder can be ignored  
        Sting strTime [20];  // target creation time  
        Sting strPath [64];  // target path  
    } List [n];  // target list, n = u32Count  
};

**CRC Description:**
In CSP-TCP applications, the CRC is ignored

**5.7 file start query**

It is mainly used for continuous queries on file lists. The application conditions are as follows:
1. The device at the same time only allowed to execute a file query event, that is, the file query must wait for the end of the file query event before the implementation of the next query event.

2. The file began to query, follow-up file list query file suffix must be consistent with the file start query, or as invalid query.

**Request instruction**

<table>
<thead>
<tr>
<th>Instruction Application: TCP</th>
<th>Data flow: G / A --- →</th>
<th>CS</th>
<th>CRC1 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xF00 C</td>
<td>0x20</td>
</tr>
</tbody>
</table>

**Instruction body 32 Bytes**

Instruction body format:

```c
Struct {
    String strSuffix [32]; // file suffix, support multiple suffixes such as " .jpg | .mov ";
};
```

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

**Response command:**

<table>
<thead>
<tr>
<th>Instruction Application: TCP</th>
<th>Data flow: CS --- →</th>
<th>G / A</th>
<th>CRC2 / reserved 2 Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification code 1 Byte</td>
<td>status code 1 Byte</td>
<td>Instruction number 2 Bytes</td>
<td>Instruction length 2 Bytes</td>
</tr>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0xF00 D</td>
<td>0x00</td>
</tr>
</tbody>
</table>

**Instruction body 0 Bytes**

**CRC Description:**

In CSP-TCP applications, the CRC is ignored

【Remarks】

The device is querying and can not respond to another query event. The response command status code is returned to 0x08 "The device is busy and can not respond".

436-0100-01-10, Rev. 200 Duo User Guide

Information on this page is subject to change without notice
### 5.8 file stop query

**Request instruction**

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>CRC1 / reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>0x00</td>
<td>0xF00E</td>
<td>0x00</td>
<td>0x00</td>
</tr>
</tbody>
</table>

**Instruction body**

<table>
<thead>
<tr>
<th>0 Bytes</th>
</tr>
</thead>
</table>

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.

### Response command:

<table>
<thead>
<tr>
<th>Identification code</th>
<th>status code</th>
<th>Instruction number</th>
<th>Instruction length</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x64</td>
<td>Status Code</td>
<td>0xF00F</td>
<td>0x00</td>
<td>0x00</td>
</tr>
</tbody>
</table>

**Instruction body**

<table>
<thead>
<tr>
<th>0 Bytes</th>
</tr>
</thead>
</table>

**CRC Description:**

In CSP-TCP applications, the CRC is ignored.
FLIR Duo User Guide

FLIR Commercial Systems, 2016. All rights reserved worldwide. No parts of this manual, in whole or in part, may be copied, photocopied, translated, or transmitted to any electronic medium or machine readable form without the prior written permission of FLIR Commercial Systems.

Names and marks appearing on the products herein are either registered trademarks or trademarks of FLIR Commercial Systems and/or its subsidiaries. All other trademarks, trade names, or company names referenced herein are used for identification only and are the property of their respective owners.

This product is protected by patents, design patents, patents pending, or design patents pending.

If you have questions that are not covered in this manual, or need service, contact FLIR Commercial Systems Customer Support at 805.964.9797 for additional information prior to returning a camera.

This documentation and the requirements specified herein are subject to change without notice.

This equipment must be disposed of as electronic waste. Contact your nearest FLIR Commercial Systems, Inc. representative for instructions on how to return the product to FLIR for proper disposal.

FCC Notice. This device is a subassembly designed for incorporation into other products in order to provide an infrared camera function. It is not an end-product fit for consumer use. When incorporated into a host device, the end-product will generate, use, and radiate radio frequency energy that may cause radio interference. As such, the end-product incorporating this subassembly must be tested and approved under the rules of the Federal Communications Commission (FCC) before the end-product may be offered for sale or lease, advertised, imported, sold, or leased in the United States. The FCC regulations are designed to provide reasonable protection against interference to radio communications. See 47 C.F.R. §§ 2.803 and 15.1 et seq.

Industry Canada Notice. This device is a subassembly designed for incorporation into other products in order to provide an infrared camera function. It is not an end-product fit for consumer use. When incorporated into a host device, the end-product will generate, use, and radiate radio frequency energy that may cause radio interference. As such, the end-product incorporating this subassembly must be tested for compliance with the Interference-Causing Equipment Standard, Digital Apparatus, ICES-003, of Industry Canada before the product incorporating this device may be manufactured or offered for sale or lease, imported, distributed, sold, or leased in Canada.

Avis d’Industrie Canada. Cet appareil est un sous-ensemble conçu pour être intégré à un autre produit afin de fournir une fonction de caméra infrarouge. Ce n’est pas un produit final destiné aux consommateurs. Une fois intégré à un dispositif hôte, le produit final va générer, utiliser et émettre de l'énergie radiofréquence qui pourrait provoquer de l'interférence radio. En tant que tel, le produit final intégrant ce sous-ensemble doit être testé pour en vérifier la conformité avec la Norme sur le matériel brouilleur pour les appareils numériques (NMB-003) d’Industrie Canada avant que le produit intégrant ce dispositif puisse être fabriqué, mis en vente ou en location, importé, distribué, vendu ou loué au Canada.

EU Notice. This device is a subassembly or component intended only for product evaluation, development or incorporation into other products in order to provide an infrared camera function. It is not a finished end-product fit for general consumer use. Persons handling this device must have appropriate electronics training and observe good engineering practice standards. As such, this product does not fall within the scope of the European Union (EU) directives regarding electromagnetic compatibility (EMC). Any end-product intended for general consumer use that incorporates this device must be tested in accordance and comply with all applicable EU EMC and other relevant directives.

436-0100-01-10, Rev. 200 Duo User Guide

Information on this page is subject to change without notice.

FLIR The World’s Sixth Sense™