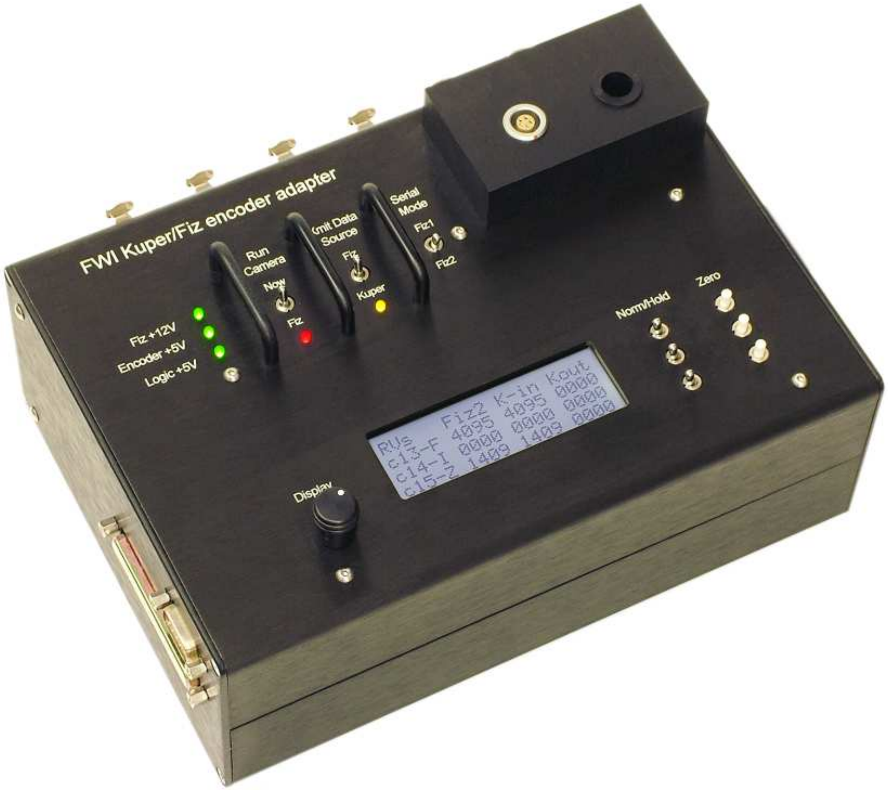
Users Manual – FWI Fi+Z / Kuper interface with Libra–V serial support



Features

Provides support for either Fiz1 or Fiz2 serial modes with the flick of a switch.

Supports Libra-V serial stream for Libra encoding applications.

Displays the status of all data channels, in and out, in real time

Provides ”hold” circuitry that momentarily freezes the data flowing into the Kuper, allowing the operator to zero channels without the cooperation of the camera assistant.

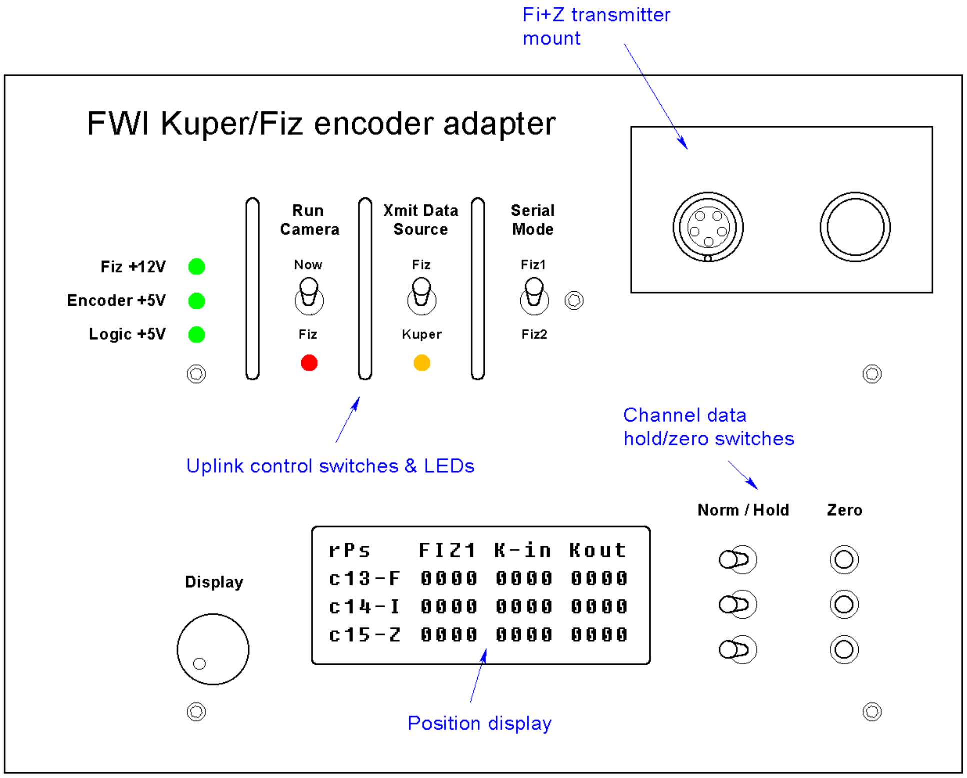
Provides a bypass mode, so the camera assistant can control the MDR directly when the Kuper is busy with file operations.

Powers the Fiz handunit from its own internal power supply, not the transmitter battery, thus extending transmitter battery life

Opto-isolates the Fiz hardware from the Kuper, for hardwired situations.

Provides opto-isolated breakout for the remaining 12 channels of encoders, and powers them from an internal 3amp supply, not from the Kuper supply.

Controls and connectors, Top



```
rPs    FIZ1 K-in Kout
c13-F  0000 0000 0000
c14-I  0000 0000 0000
c15-Z  0000 0000 0000
```

Fi+Z Transmitter Mount

Platform into which the Fi+Z transmitter and battery unit plug. If using a hardwired cable to the MDR, plug it in here. *You can't power a Fi+Z transmitter through the external connector, so the Fi+Z transmitter still powers itself from its own battery*; the user has to monitor this battery in the normal way. The handunit is powered from the box supply.

Position Data Display

An LCD unit which displays the current Fi+Z handunit data, encoder count transmitted to the Kuper and step count received from the Kuper. It also indicates the status of the data stream from the handunit, good (Fiz1 or Fiz2), or bad (NFD – No Fiz Data).

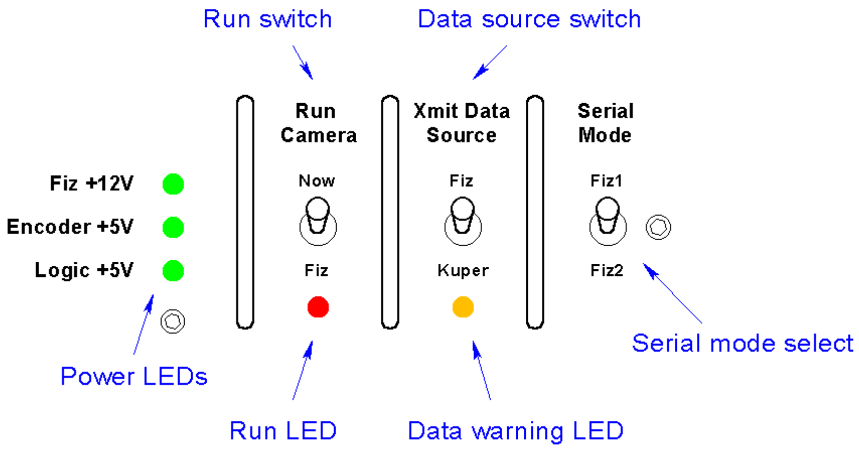
Channel Hold and Zero Switches

There is a hold switch and zero button for each channel. The hold switch temporarily freezes the encoder count going into the Kuper, giving the operator the ability to check his data or zero a channel without coordinating with the AC. The box continues to receive and transmit data in the background, and returns to the correct count when the hold is released. The zero button zeros both the encoder output and the step input count. When a channel is in "hold" the yellow LED under the data source switch will blink as a warning.

Display Adjust Knob

Adjusts LCD contrast.

Uplink switches & LEDs



Uplink control switches & LEDs

Power LED's

There are three green power LEDs, one for each power supply; +5volts for the box logic, +12 volts for the Fi+Z handunit, and +5volts for the encoders

Camera Run switch & run LED

The Run switch has two positions, Run **Now**, which will start the camera regardless of the handunit switch, run from **Fiz**, which follows the operation of the handunit switch

Whatever the source, the red LED under the switch flashes when the camera is commanded to run.

Xmit Data Source switch

Selects whether the Fi+Z transmitter gets the Kuper step data, or data directly from the handunit. Think of it as a "bypass" switch. It may be useful when you're busy doing file operations and the AC wants his lens back.

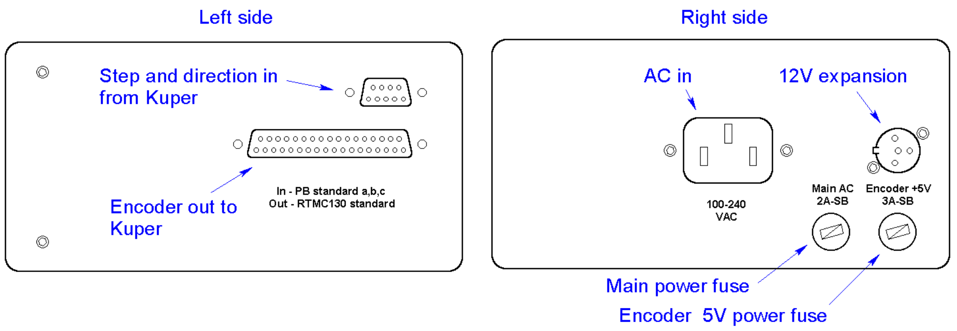
Data warning LED

This yellow LED indicates that the transmitted data is not "normal". It lights whenever the data is source switch is in "bypass" mode and the transmitter is not getting Kuper data, and flashes whenever a channel is in "hold" mode.

Serial Mode Select

Selects the serial packet mode, Fiz1 (31.24Kbaud, 6mS data blocks) or Fiz2 (125Kbaud, 12 mS data blocks).

Side Connections



Step/Dir in

A standard Point Blank current-loop step direction input. Channel A is Focus, B is Iris and C is Zoom. See pinouts on last page

Encoder Out

The encoder output connector is a 1to1 match for a Kuper RTMC48 encoder input fed externally on a 37 pin DB type ribbon connector. Most RTMC48 systems were shipped with this connector and pinout. See pinouts on last page

Channels 1 through 12 are encoder signals passed through from the back panel connectors via opto-isolators. Channel 13 is Focus, 14 is Iris, and 15 is Zoom. Ground is connected, but the +5V pins are ignored. The opto-isolators in channels will respond up to 500KHz, so the box will operate at full bandwidth when feeding a Kuper 2000 board.

AC in

The box has 3 power supplies, one for the logic, one for the FiZ handunit, and one for the Encoders. All are electrically isolated from the input supply and each other. Input is marked as 100 to 240 volts, but the supplies are actually spec'd from 86 to 264VAC, 100 to 260VDC, so it should work anywhere on Earth you can plug in a light bulb.

AC main power fuse

Fuse for the AC input line, 2amp m5x20 slow-blow.

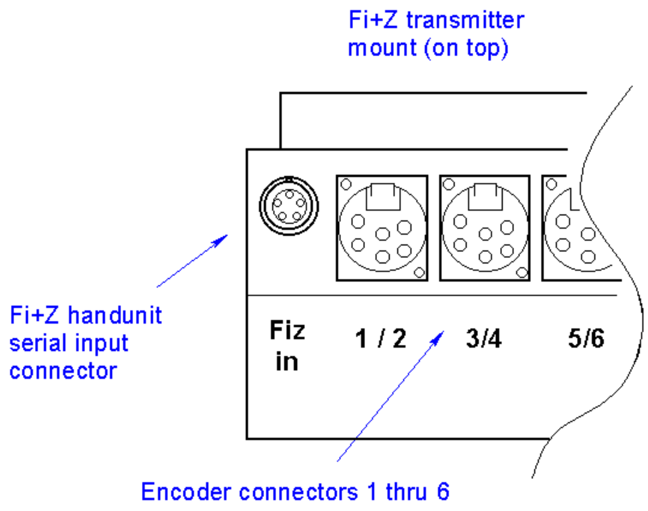
Encoder Fuse

Fuse for the +5V supply to the encoder connectors, 3amp m5x20 slow blow.

Secret mystery expansion plug

It's a secret, that's why it's called the secret mystery expansion plug.

Rear Panel Connections


Fi+Z transmitter mount (on top)
Fi+Z handunit serial input connector
Fiz in
1 / 2
3/4
5/6
Encoder connectors 1 thru 6

Fiz in

Plug in the handunit here. ONLY plug into the handunit. Normally, nothing else can go here since the transmitter will just not fit, but be careful if you're using a hard line to the camera end (the MDR). The backside port is input, for the handunit only, the topside port, on the Fiz platform is output. If you plug the MDR into the input side, the power supplies will fight.

Encoder 1-12 in

These are 6 standard Point-Blank type 6 pin XLR encoder connectors, each of which is wired for two single-ended encoder channels, for a total of 12 available encoder channels. The encoders have their own dedicated 3-amp power supply, which they share among themselves. The supply is fused at 3 amps, and the fuse is located on the right side. The encoders are optically isolated from the Kuper and Fiz circuitry, and the power supply ground.

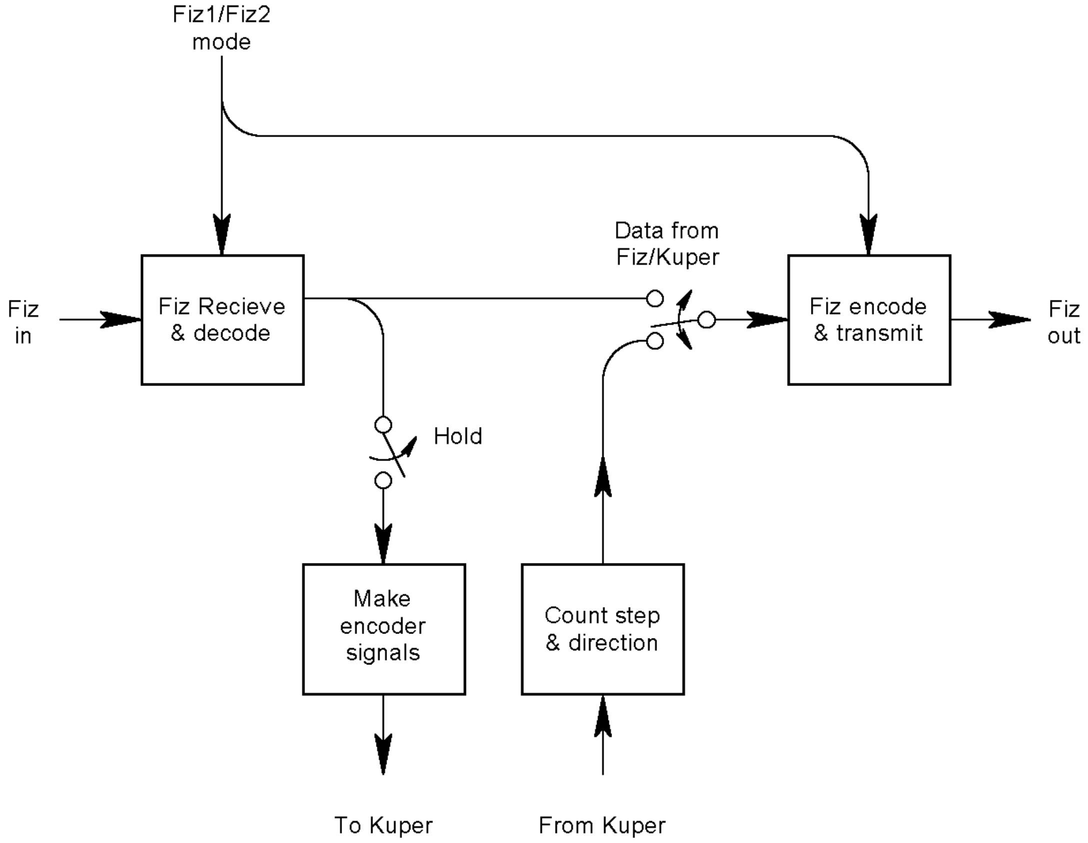
Initial hookup

Basic theory

The FiZ/Kuper interface sits between the Preston handunit and the Preston transmitter. It intercepts the handunit signals, which it conditions to resemble encoder signals. In this form they can be fed to, and recorded by, the Kuper as though a conventional focus encoder was attached.

The box also accepts step/direction signals from the Kuper, which it translates back into Preston format signals which get sent to the Digital MDR lens motors. In this way the Kuper can control the Preston lens motors as though they were conventional step motors.

The block diagram of the adapter box looks like this.

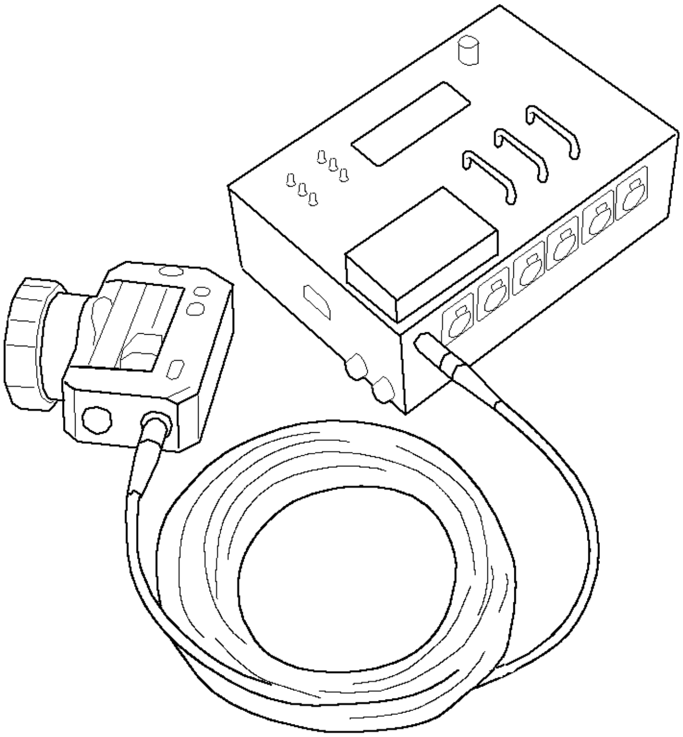


Don’t worry about the details, we’ll come back to it later.

In order to insert ourselves into the system, we have to make 4 connections, 1) Data out of the Preston handunit, 2) encoders into the Kuper 3) step/dir out of the Kuper, and 4) data to the Preston transmitter.

Connection to the Preston parts

1) Get the handunit (or Libra-V) data in



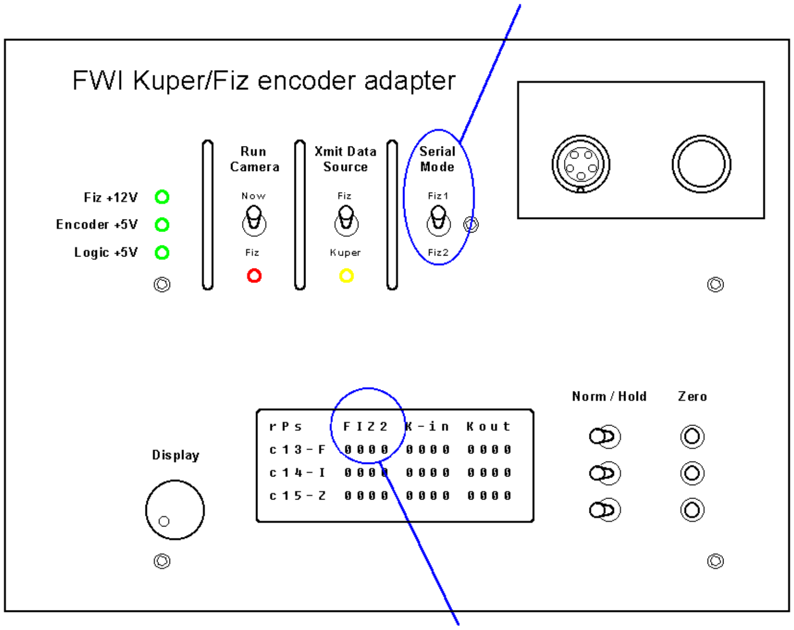
Connect the handunit to the Fiz data input port on the back of the unit. Use a Preston hardwire cable. This cable can be identified by its 5 pin Lemo connectors on each end. Original Preston cables are color coded with a white strain relief boot.

The handunit can be hot-plugged and unplugged without damage. The green status LED on the front of the handunit should come on, indicating power.

Of course, if you’re in the Libra-V digitizing mode, there is no handunit, but you’ll make the same connections to the Libra head. Providing you do not make a inadvertent ground connection through the connector shells and cable shields, the input port (and therefore the Libra head) is electrically isolated from both the Kuper and AC ground.

Preston MDRs were produced over the years using two distinct serial protocols, call them FiZ1 and FiZ2. This adapter speaks both “languages” but it is important to tell it which one we’re using. This is easy.

Serial mode switch



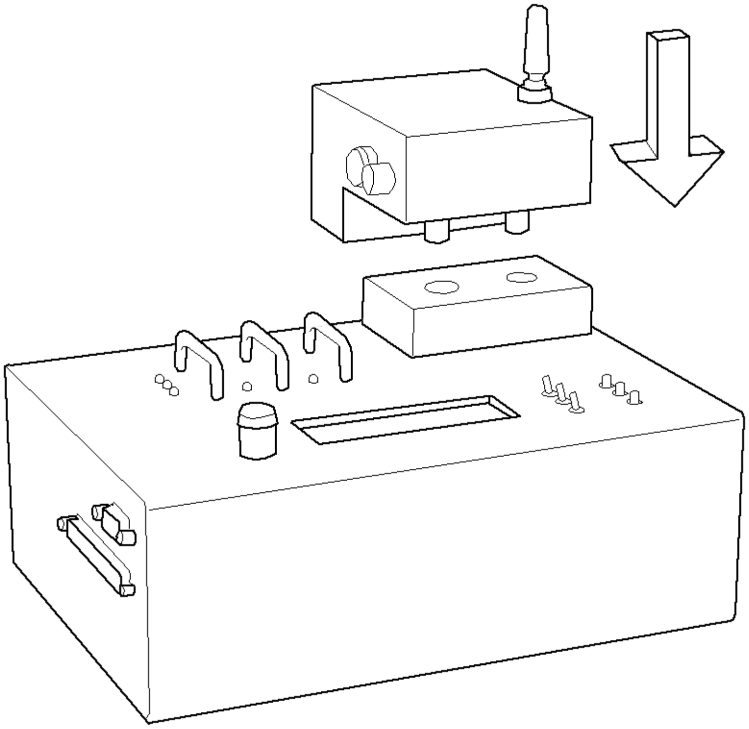
```
rPs    FIZ2 K-in Kout
c13-F  0000 0000 0000
c14-I  0000 0000 0000
c15-Z  0000 0000 0000
```

Status of incomming serial signal from handunit

Identify the incoming serial status display. It’s on the top line of the LCD display, second from the left. It will be displaying FiZ1, FiZ2, or flashing No Fiz Data (NFD).

If it’s flashing NFD, that likely means that the adapter and the handunit are using different serial protocols. If this is the case, identify the serial mode switch on the top of the adapter, and select the other serial mode. When the adapter and handunit are speaking the same language, the display will lock and the input data column will start displaying handunit data.

2) Get the transmit data out



Assuming we've got the Preston data flowing into the adapter box, we have to get data out to the transmitter so it can be uplinked to the MDR. This is fairly straightforward, as the transmitter plugs directly into the top of the adapter box, automatically making all the necessary electrical connections.

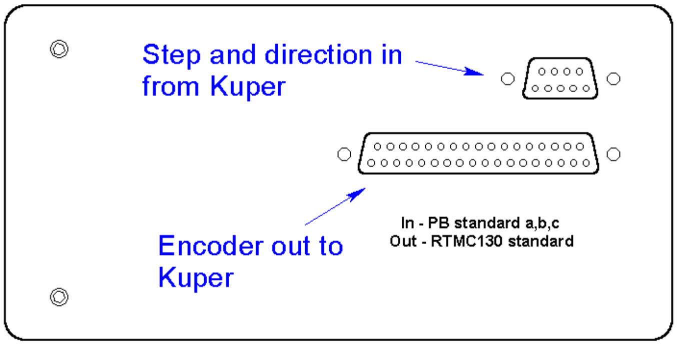
Note that a Preston transmitter cannot be powered through the exterior connector. It has to run on its own battery, and it has its own power switch. You must make sure the transmitter is turned on and does not run out of battery power. Helpfully, there's a power LED right on the front of the transmitter, which goes red to indicate a low battery.

The transmitter can be hot-plugged and unplugged into its socket without damage. The battery can be replaced while the transmitter is in the socket without damage, though Preston does recommend shutting off the power switch, as a brief burst of spurious transmission may result.

Note that in the Libra mode, the box does not transmit any data (since the current Libra standard does not support external position data input). When using a Libra head, make no connection to the transmitter port.

Connection to the Kuper parts

Connecting to the Kuper side tends to be more tricky than connecting to the Preston side, since there is so much variation in how Kuper systems are configured. Many different connector standards exist and most field systems already have at least some dedicated step lines and encoder lines spoken for, so the user may have to resort to cable adapters.



Connectors for Kuper encoder and Kuper step signals are located on the left hand side of the box.

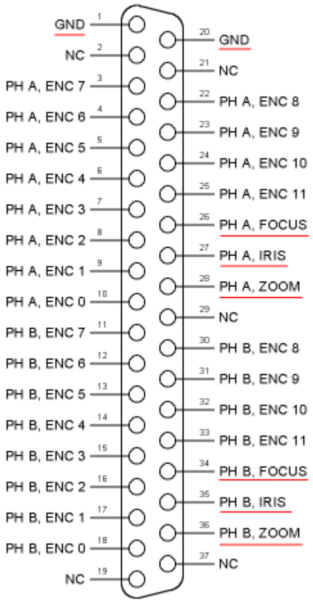
1) Get the encoder data out

The adapter box has a 37 pin encoder connector. The pinouts on this connector are the same as the encoder pinouts on a Kuper RTMC48 board brought out using a standard D37 ribbon connector.

If you have such a Kuper, you can use a 1:1 cable to connect the two boxes, and the Preston channels will come out on 13 (focus), 14 (iris), and 15 (zoom). The remaining encoder channels will be broken out to individual encoder connectors on the back of the unit. Please see page 22 for notes on known encoder conflicts on some RTMC48 cards.

If you have to make a cable to adapt the box to an existing application, you will have to connect at least these 7 wires…

• Ground	(both pins 1 and 20)
• Focus "A"	(pin 26)
• Focus "B"	(pin 34)
• Iris "A"	(pin 27)
• Iris "B"	(pin 35)
• Zoom "A"	(pin 28)
• Zoom "B"	(pin 36)



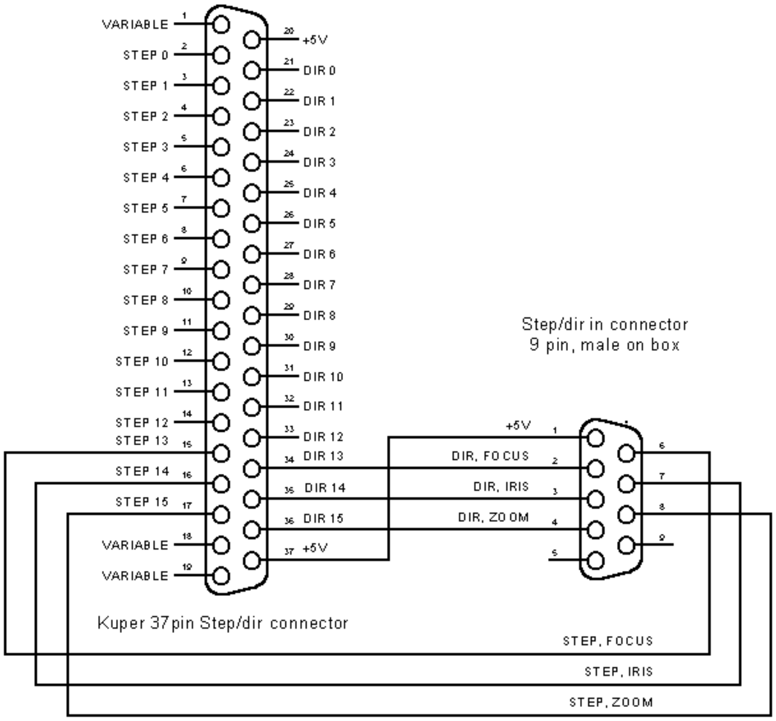
At this point, it is often expedient to cycle power on the adapter box, and use the startup menu to put the box into self-test mode. In this mode the adapter box will continuously generate encoder signals as though someone is manipulating a handunit.

The Kuper operator can then methodically search the encoder channels, one at a time, to figure out where the encoder signals are going inside the system cabling.

Note that the Kuper system uses two distinct numbering systems for the encoder channels. Within the software, encoder channels are numbered 1 through 15 (with 16 being the default jogbox encoder) but the hardware documentation refers to the encoder channels as 0 through 14, (with 15 being the jogbox).

2) Get the step/direction data in

The adapter box uses a common D9 Point-Blank style step/direction connector. Depending on your system configuration, you may have such a pinout available, or have to make up a cable to work around existing equipment mapping.



A typical step/direction cable using Kuper axis 14 for Focus, 15 for Iris and 16 for Zoom.

(note that in the Kuper software the axes are labeled 1- 16, while the actual hardware channels are labeled 0 -15)

At this point it may be expedient to jog Kuper axes one at a time, and watch for activity in the Kuper out data column, in order to figure out which step/direction signals are going where inside the system cabling

Fiz data format and block diagram

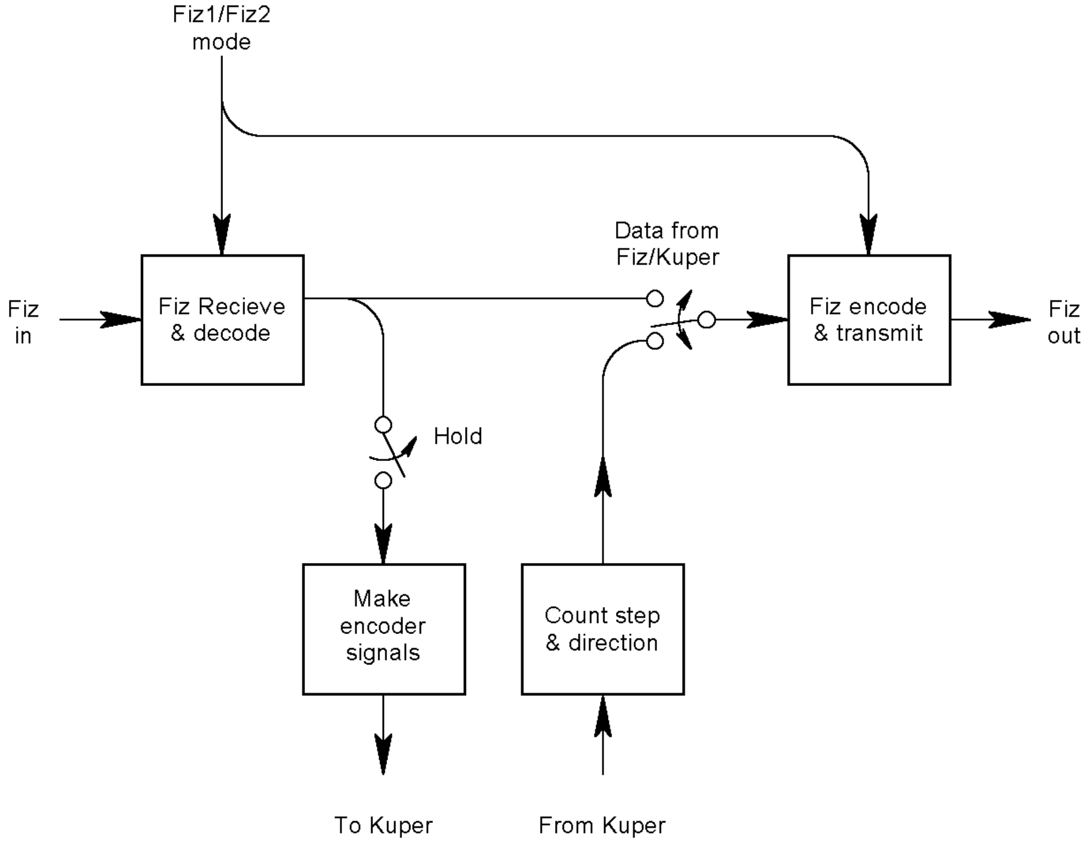
Fi+Z Data

The Preston Fi+Z format uses a 16-bit data word for each channel. In theory, this means that each channel counts from 0 to 65536. This is, of course, much more accuracy than we can really use, especially with a Kuper 48 card which is bandwidth limited to just over 8000 encoder counts per second.

So for the purposes of this box, we internally divide the count by 16, and deal with it on the Kuper side as though the Fi+Z system operated with a range of 0 to 4095. The interface box takes care of all the math going in and out, so this translation is transparent to the operator.

Block Diagram

We saw the block diagram earlier



There are 4 basic blocks, 1) Fi+Z Receive and decode, 2) Fi+Z encode and transmit, 3) Make Kuper encoder signals, and 4) Count Kuper step pulses.

Most of the time, operation is straightforward; the Fi+Z handunit signal (or Libra serial stream) is received and decoded. It is displayed on the front panel.

Next, the data is passed to the Kuper encoder block, where a circuit translates the count into signals that look like encoder pulses, and transmits them out to the Kuper. The data passing through the Kuper encoder block is also displayed.

In normal operation, the Kuper responds with step/direction signals, which are counted and displayed. (in Libra mode there is no data output and no data displayed)

This count is then passed to the Fi+Z encode and transmit block which translates the data back into a format that the Fi+Z transmitter and MDR can understand.

Note that it's possible to send too many count pulses into the box and overshoot or undershoot the 0 to 4095 target range, especially during preroll or postroll. In these cases, you'll see a display that says "over" or "undr" indicating overshoot or undershoot, respectively. That's OK, the output gets clipped at 0 or 4095, as necessary, and circuitry still keeps track of the real count. As soon as you get back into range, the count returns to normal operation.

It is possible, by the way, to overflow the step counters but it's unlikely, since there's a large buffer (on the order of +/- 500,000 counts). If somehow you do manage to overflow, you'll get an error message and that channel will lock itself out until you reboot the box.

There are also a few routing switches in the data stream, which allow the Kuper operator some convenient features.

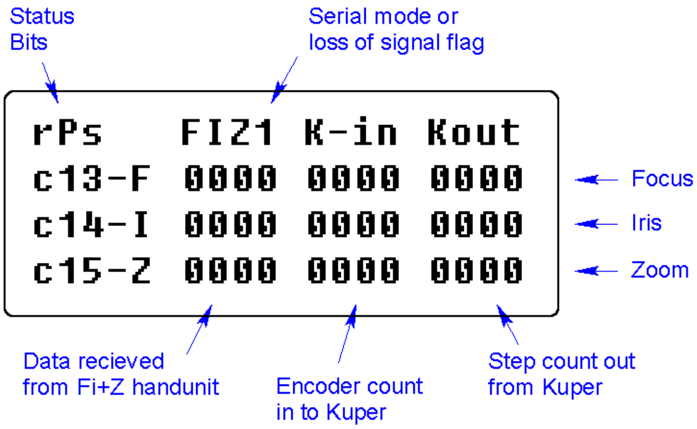
A "hold" switch in the path before the encoder block temporarily freezes the data fed to the Kuper, allowing the operator to change or zero a channel without coordinating with the camera assistant. Internally, the box still keeps track of the incoming data, and when the hold is released, the encoder count returns to where it's supposed to be.

A data select switch allows the Kuper to be bypassed entirely, with the Fi+Z handunit sending data directly to the transmitter. In this mode the camera assistant can maintain control of the lens for focusing and slating while the Kuper operator is busy doing things that would otherwise hold up the lens or interfere with the data stream.

And a Fiz1 / Fiz2 mode switch allows the operator to use the unit with both revisions of Preston Fi+Z products without having to change any hardware.

Data display

During normal operation, the data display on the front of the unit provides complete information about the data flowing through. (this is the Preston mode screen, the Libra mode screen is similar). It contains several sections…



```
rPs    FIZ1 K-in Kout
c13-F  0000 0000 0000
c14-I  0000 0000 0000
c15-Z  0000 0000 0000
```

Status Bits
Serial mode or loss of signal flag
Focus
Iris
Zoom
Data recieved from Fi+Z handunit
Encoder count in to Kuper
Step count out from Kuper

Data recieved from Fi+Z handunit

Channel data as received from the handunit. If communication is lost, the display indicates the last good data block from the handunit.

Encoder count into Kuper

Channel data being output from the adapter box into the Kuper as encoder counts. This data typically chases the Fi+Z input data unless you're holding or zeroing a channel.

Step count out from Kuper

Channel data being output from the Kuper into the adapter box as step counts. This is the data that will be uplinked to the Fi+z transmitter, unless you're in bypass mode.

Note that these position displays will either read 0 to 4095 counts or 0 to 100%, depending on how the display configuration is set.

Serial mode / LOS flag

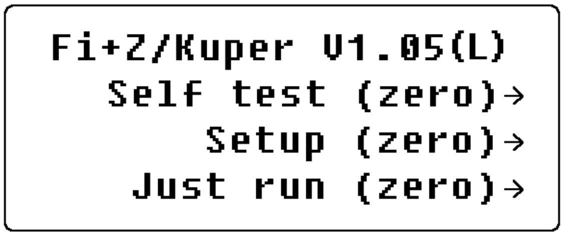
Displays "Fiz1" or "Fiz2" to indicate a good serial link up from the handunit using either Fiz 1 or 2communication protocols. If the link is lost this display blinks "NFD*" to indicate No Fiz Data is being received.

Status bits

Displays the current state of the command bits being uplinked to the transmitter. Status bits include R/r (for run command on/off), P/V (for position/velocity mode) and S/s (for snorkel mode on/off) normal display is rPs for no-run / position / no-snorkel.

Power up-screen

Upon powering up, the adapter displays a setup screen, which allows you to configure various parameters or enter a self-test mode. If you make no selection, the box automatically escapes into normal run mode after three seconds.



```
Fi+Z/Kuper V1.05(L)
  Self test (zero)→
      Setup (zero)→
   Just run (zero)→
```

Throughout this section, you select options by pressing the zero buttons opposite the arrow by your choice.

Self Test

Places the box in run mode, but instead of waiting for handunit data on the serial line, the box generates it’s own data, mimicking a handunit cycling though channels. Since all other box and Kuper operations are the same, this mode is useful for setting up the remainder of the system without requiring an actual handunit, which are often in short supply on a working film set. Self-test mode runs until you cycle power to the box.

Just Run

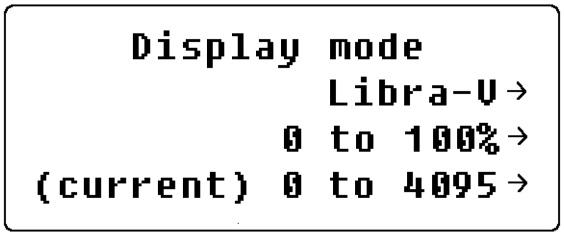
Immediately escapes to normal run mode.

Setup

Access configuration menus, including display mode and default status bits.

If you select setup, you get the following menus…

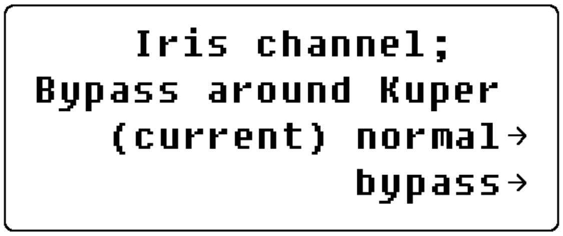
Display mode menu



```
     Display mode
             Libra-V→
          0 to 100%→
(current) 0 to 4095→
```

This screen selects the box operating mode (Kuper or Libra) and the data display format within Kuper mode, either 0 to 4095 counts, or 0.0 to 100.0%. The Libra display is always given in terms of degrees. If you pick Libra mode, you go right into operation and skip the following screens (which have no meaning to the Libra head)

Iris channel bypass

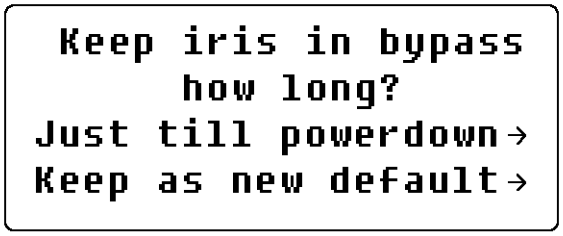


```
    Iris channel;
Bypass around Kuper
   (current) normal→
             bypass→
```

Since it is often expedient to leave the iris channel under the direct control of the camera assistant at all times, this menu allows you to put the iris channel in full bypass mode and allows the Kuper operator to totally ignore it (though you can still record data).

In this mode the Kuper has no influence on the channel at all, the box simply passes along whatever iris data it receives from the handunit. If the data stream from the handunit is interrupted, the iris data output to the transmitter freezes on the last good data frame.

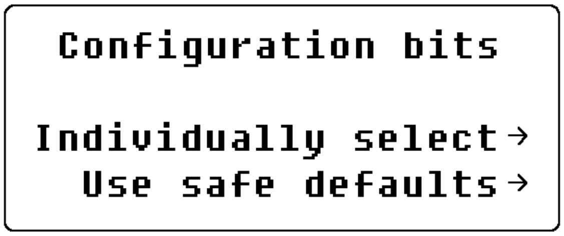
If you choose the “bypass” option, you get a second selection window



```
 Keep iris in bypass
      how long?
Just till powerdown→
Keep as new default→
```

Asking if you want iris bypass mode to apply temporarily (until the next power cycle) or to become the default mode until further notice.

Configuration bits



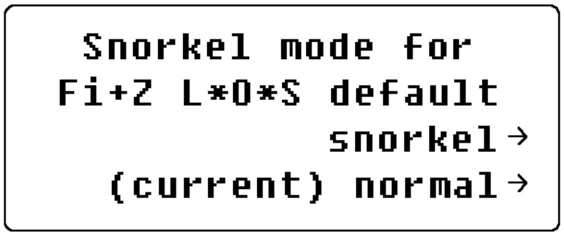
```
 Configuration bits

Individually select→
  Use safe defaults→
```

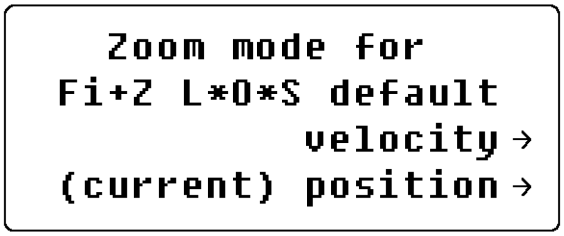
Allows you to adjust the individual status bits, or escape using safe defaults. If you choose the “safe defaults” menu, the machine selects position mode / no snorkel / no tristate as default values.

You will probably want to delve no deeper into the inner workings of the Fi+Z data stream, but if you choose the “individually select” menu, you get the following three screens…

Snorkel Mode and Zoom mode



```
  Snorkel mode for
 Fi+Z L*O*S default
            snorkel→
   (current) normal→
```

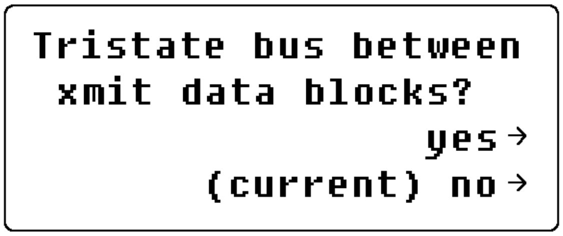


```
   Zoom mode for
 Fi+Z L*O*S default
           velocity→
 (current) position→
```

The adapter box always follows the handunit status bits whenever a handunit is attached, however, in the event of a loss of signal from the outside world, it has to know what to do with these bits. These two screens allow you to select the default states transmitted as the MDR operating mode. They are analogous to the corresponding mode setting switches in the handunit.

Note that in normal operation you will almost always use normal mode for the snorkel bit and position mode for the velocity bit. If you get the mode wrong, the lens motors will behave in weird and mysterious ways.

Tristate bus between packets



```
Tristate bus between
 xmit data blocks?
                yes→
       (current) no→
```

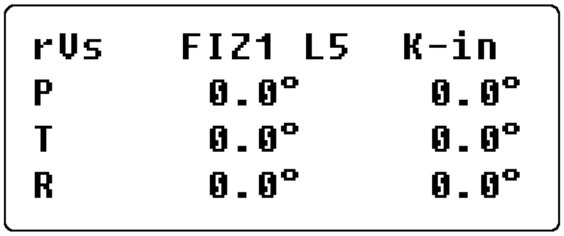
The FIZ-2 communication protocol allows up to 7 units to share a single communication channel. To this end, individual “clients” will timeshare the bus and relinquish it between data packets. This adapter box has software and hardware that allows it to tristate its uplink bus between transmission packets so that other devices may share.

How this communication protocol will operate in real-life is uncertain, since the new format is mostly a future expansion spec. The only Preston device currently configured to use it is the F/X unit, which tends to be seen infrequently, and it is unclear if original transmitters designed under FIZ-1 appreciate having their data buses tri-state.

Generally, we'd suggest that you leave the box in "do not tristate" mode unless you find yourself in a network application where it becomes an issue.

Libra mode

During Libra mode operation, the data display is similar to the normal Preston display, but has a few significant differences, since the Libra head produces a "hard coded" output representing pan, tilt and roll angles.



```
rVs    FIZ1 L5   K-in
P        0.0°      0.0°
T        0.0°      0.0°
R        0.0°      0.0°
```

Foremost in the differences is the lack of a transmit data column. The Libra head doesn't accept positional data, therefore the box doesn't supply it.

The Libra head handles rollovers from 360° to 0° and 0° to 360° in CGI style discontinuous mode. Since the Kuper works in physical continuous mode, the adapter box provides "graceful" transitions, where the rollover through 360° to 0° is provided to the Kuper as continuous counts; 359°, 360°, 361°, 362°, etc. Likewise, the transitions through zero are provided as 2°, 1°, 0°, -1°, -2°, etc.

The adapter box can keep track of about 4 transitions ($^{+}/_{-}$ 1440°) in either direction of zero.

Zeroing works like it does with a Preston, you "hold" a channel, Zero the output data, zero the Kuper, and then release the hold to chase the input again.

Note that since the Libra can output negative directions, adapter box automatically picks the shorter path when returning from zero. For example, if the head is reporting 350 degrees, the adapter box will interpret that as being a slight pan to the left (-10°) rather than almost one full turn to the right (+350°) .

When using a Libra head, Pan is provided on encoder channel 13, Tilt on 14, and Roll on 15.

To encode real degrees, use 11.375ppu in the Kuper (11.375 = 4095 counts / 360°).

First Time Out

There are as many ways to set up a Kuper as there are operators, but while familiarizing yourself with the unit, try the following settings…

First, connect the box to the Kuper

See the previous section about hooking up to the Kuper and Preston. If you use a 1:1 cable into an RTMC48 style 37 pin encoder connector on the Kuper, the Preston channels will come out on 13,14, and 15.

Likewise, select three channels for Kuper step/direction and connect them to the box using a Point Blank type standard connection.

On the box

Cycle the power on the box and when it displays the startup screen enter the setup menu and select 0 to 4095 display mode, no Iris bypass and safe defaults for the configuration bits. Cycle the power on the box again and when it displays the startup screen put it into self test mode.

In the Kuper

Set up channels for Focus, Iris and Zoom thusly…

Pulses per unit:	1.000 ppu (11.375 for Libra mode)
Slew speed:	10000 pps (use 2000 pps if you're going to mouse jog a lot)
Accel/Decel:	0.10 s
Encoder sensitivity:	1.000
Encoder damping:	0
Encoder mode:	position

--- these may vary on your system, but if you use a straight thru cable… ---

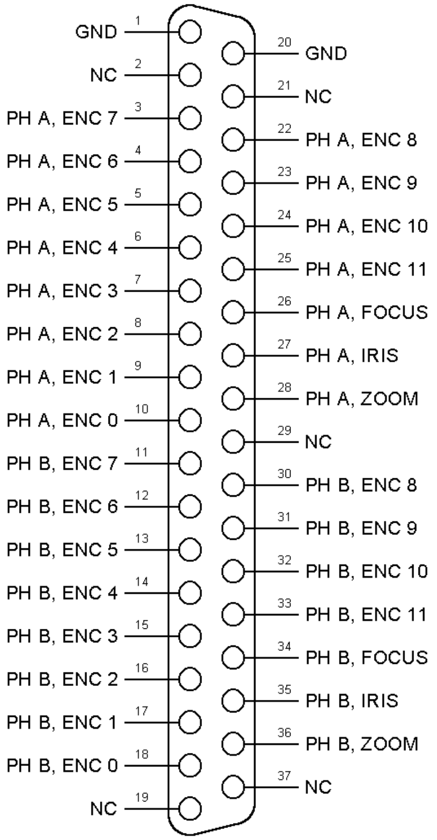
Kuper encoder channel for Focus	ch 13	(hardware ch 12)
Kuper encoder channel for Iris	ch 14	(hardware ch 13)
Kuper encoder channel for Zoom	ch 15	(hardware ch 14)

These damping and sensitivity numbers seem weird, but remember that the MDR unit applies its own damping, and can track changes as fast as you can output them. The limiting factor, from the Kuper side, is the encoder pulse count rate of an RTMC130 card. Currently, the software in the adapter box clocks out about 3000 encoder pulses per second, which allows a full range move in 1 1/3 seconds, with about a 2.5x safety margin for noisy environments. Later revisions of the software will probably count faster as reliability is established in the field.

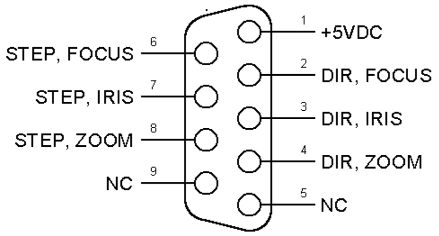
When you are happy that signals are going back and forth between the box and the Kuper, take the box out of self-test by cycling power and try a real Fi+Z.

If you prefer to think in terms of 0 to 100%, now is a good time to adjust your axis ppu settings to 40.95. (Leave at 11.375 for Libra mode 0-360°)

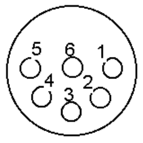
Pinouts



Encoder out connector
37pin, female on box



Step/dir in connector
9 pin, male on box



Encoder connector, 6 pin
Switchcraft style, female on box

1: GND

2: Ch1-A

3: Ch1-B

4: Ch2-A

5: +5V

6: Ch2-B

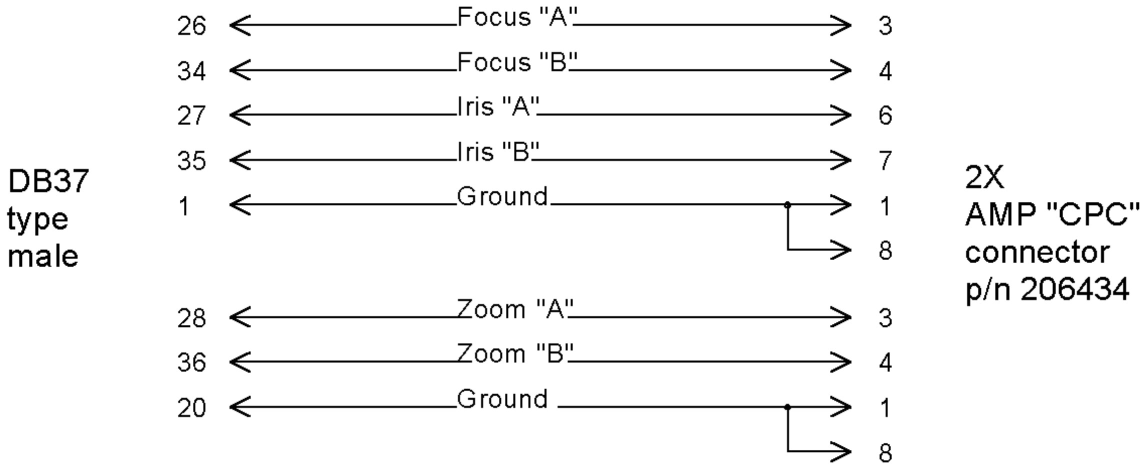
Note when referring to encoder channels that the Kuper uses two different counting systems, one inside the software, where channels are numbered 1 through 16, and one for the hardware, where channels are numbered 0 through 15. *Also see the note on page 22 regarding known bug issues on Kuper hardware channels 13 and 14.*

Channel	hardware	software
Focus (LM pan)	c12	c13
Iris (LM tilt)	c13	c14
Zoom (LM roll)	c14	c15

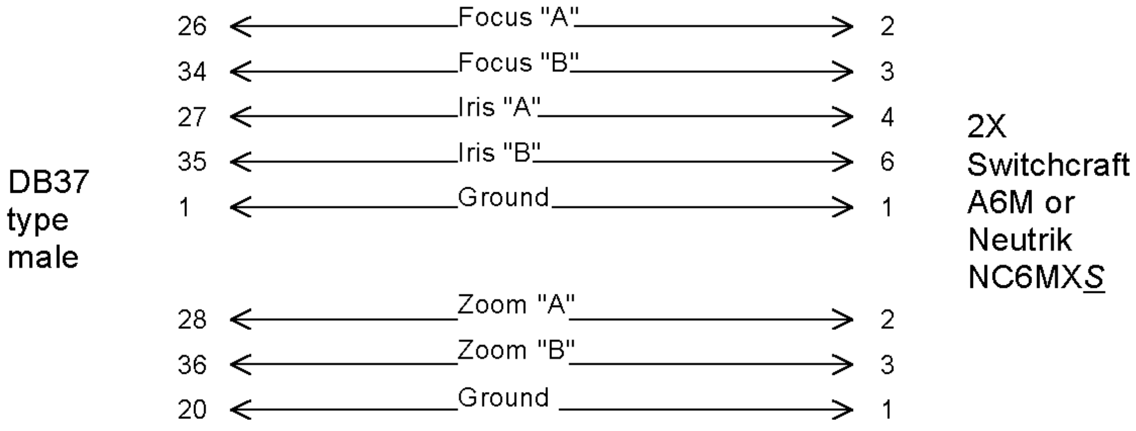
Bypassing the encoder stuff

The Fi+Z interface is designed to also handle all your encoder interfacing needs, providing breakout, opto-isolation, and a dedicated 5V encoder supply, all in a convenient RTMC48-friendly package, but if you already have all this stuff, and you just want to plug it in as a Fi+Z interface device, that doesn't make you a bad person. Really.

You will, however, have to build an interface cable. In the event that you're wired for the popular Lynx or Point Blank standards, your cable will look like this…



Breakout to Lynx MBP-01 standard



Breakout to Point Blank 6 pin XLR standard

Known encoder hardware issues on K48 boards

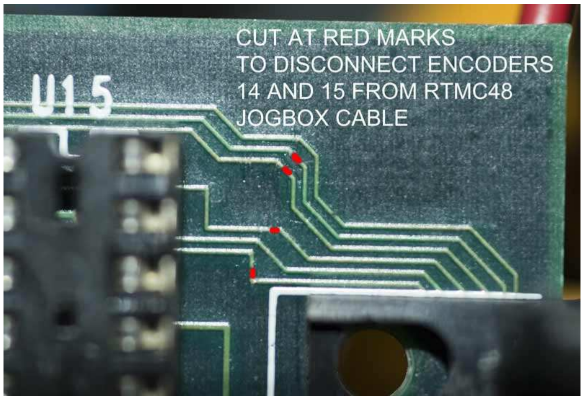
Some users report problems with encoder channels 14 and 15 on Kuper 48 cards when used with some Kuper jogboxes. This relates to a known wiring issue between the two components.

When the RTMC48 board was originally designed, provision was made for future expansion through smart peripherals plugged into the jogbox port. To facilitate this, the three high encoder channels were brought to the jogbox connector on the K48 card, in addition to the usual encoder connector.

Encoder channel 16 became, of course, the jogbox jogwheel, but channels 14 and 15 were never implemented. However, bringing these two signals to the jogbox connector produced a hardware incompatibility with older jogboxes which used these pins for other purposes.

Basically, the act of plugging in an older jogbox shorts out these two encoder signals, producing erratic operation on channels 14 and 15.

If this is the case on your system, there are several options. Bill Tondreau posts information on the Kuper website advising clients to simply cut the offending traces, since it solves the problem forever. He posts this picture of which traces are affected.

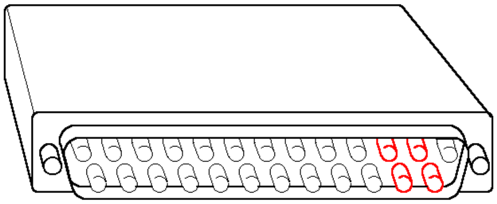


“Jogboxcuts.jpg” from Kupercontrols.com

Alternately, FWI can supply a dongle, which connects between the jogbox port and the jogbox cable and passes all but the offending traces, thus eliminating the signal conflict.

Cutting traces on the board seems a little drastic, and most people hate easy-to-loose dongles, so we advise that you solve the problem by simply breaking off the offending pins in your jogbox cable; those signals don’t affect anything else, and you’ve probably already got a dedicated jogbox cable anyway since they usually have to be modified a bit to clear the accessory connector.

The pins in question are Pins 11, 24, 12 and 25, which are encoder 13, phase A&B, and encoder 14, phase A&B, respectively. They're at the very end of the connector on the male end.



remove pins 11,12, 24 & 25 on
male end of jogbox cable

These pins can be removed easily with a pair of fine needle-nosed pliers , a hemostat, or any similar tool. Grab them down near the base and twist and snap them off, being careful to leave them flush with the connector face so they can't make any contact with the mating side.

This is a permanent fix, which does not alter the operation of the jogbox in any way.