Kuper – the missing memos (Rev 1.00)

Over the years, many, many, e-mails relating to new Kuper functions or extensions have passed over our desk.

Since the Kuper manual was last updated in 1996, e-mails from Bill sometimes represent the only documentation available on the subject of new commands.

So as a “public service” we have assembled the pile of correspondence into this document. It is in no particular order, and we make no guarantees that the functions described herein still work the way they're described – or, for that matter, if they even still exist at all.

As with all of the Kuper commands which were specially written for one-off applications, many of these commands were rarely used, and consequently, any limitations or bugs are known only to a few operators, if they were found at all. Use these commands at your peril, and TEST, TEST, TEST before you take one into battle on an actual set.

Add yourself to the List

If you happen to have any Kuper info that you want to share with the world please, feel free to drop us a line at steve@flywom.com and we'll add it to the ever growing list of Kuper Lore.

And now... The Lost E-mails....

Integrating Camera
Jan 12, 1996

The "ic" Integrating Camera command requires three parameters:

Starting angle for flash output signal
Ending angle for flash output signal
Length of "grab" strobe in milliseconds, on cap output signal

The flash and cap signals are the ones available on the db15 connector. Do not use the "cap" signal on the db25 RTMC Logic connector.

All signals are true low, although this can be changed by changing jumpers on the card. Call for details.

There is a one millisecond delay between the ending angle timeout on the flash pins, and the beginning of the grab strobe on the cap pins.

Also, the serial framecount now sends at a more favorable time, starting at the beginning of the visual frame.

Autocycle command
August 22, 1996

The Autocycle command (ac) causes the move in memory to be played back repeatedly, with no pause between playbacks. If the start and ending positions do not match, the axes will automatically reposition to the starting position between starting the next playback pass.

Autocycling is started by typing the "ac" two-letter command in lower case, and entering a parameter of "1". Autocycling is turned off by typing "ac" and entering a parameter of "0", or by clicking on the red "STOP" box.

This prototype version of autocycling is not highly robust. The following constraints apply:

1. The "STOP" button only responds while the move is actually running, and not when the axes are repositioning between playback passes.

2. Under certain special circumstances, the emergency stop key may cause a system crash when autocycling is active.

3. Under some circumstances, it may be necessary to move the mouse arrow into the "FWD" box after issuing the two-letter "ac" command.

For the above reasons, autocycling should only be routinely used only for the playback of "canned" moves, rather than for move development.

It is possible to insert delays between motion playbacks by building holds into the move files with the "ah" (add holds) command. "ah 0 240" will build a 240 frame hold at the end of the move, effectively giving a 240 frame waiting period between moving parts of the playback.

Aperture scan command
September 5, 1996

This version of RTMC130 implements a rudimentary camera-speed / aperture control through the "as" Aperture Scam command. The camera is placed on the Jogbox joystick (encoder 16), and the axis named "shutter" compensates for exposure through a control curve.

-There must be axes named "camera" and "shutter". Set "VISUALFPS" to 24.0, and "DATA/VISUAL" to 5

-The shutter axis performs exposure compensation. It must have a move file, where the frame numbers are considered to be camera-fps-values, and the positions are considered to be where the axis should be for each fps value. The shutter axis motor may be connected to shutter blades, f-stop ring, polarizers, or other exposure controlling apparatus. The operator creates the appropriate control curve using ordinary move generation techniques.

-The joystick sensitivity and smoothing are controlled by the values set in Camera row of the "JOYSTICK PARAMS" area of the control panel screen. The camera must be set to VELO mode for this to work correctly. The three-decimal place gain value, and the smoothing value to the right of the "JOYSTICK PARAMS" area control the joystick feel.

-To stop the function, press the "EMER STOP" button on the Jogbox, or press the "~" key on the computer keyboard.

Snorkel Compensation
April 20, 1997

This version compensates for tilting-mirror snorkels, with periscope roll attached to the camera.

To turn on this feature, click "HardSet", "Setup Nodal Parameters", and "Snorkel".

New commands, IH, IS
May 7, 1997

"Insert Holds" Command

The new "ih" (Insert Holds) command inserts hold into existing moves. The parameters are "start frame, number of frames to insert, and an optional smoothing parameter".

```
      100 200 80
```

Inserts 200 frames into the existing move starting at frame 100, with a few passes of smoothing at factor 80 at the splice positions. The move length increases by 200 frames. "ih" always works on all the axes.

This command is only suitable for moves created with joysticks. Use "ks" (Keyframe Spread) to accomplish the same thing with key frame moves.

For holds at the beginning and end of the move, it is preferable to use the "ah" (Add Holds) command.

TIMECODE FPS BUG FIXED

Loading a move no longer resets the timecode FPS to 24.0.

GF COMMAND REMOVED

The "gf" (Goto Feedback position) command has been disabled for systems without position feedback devices, thus preventing a possible software crash on such systems.

INCREMENTAL SAVE FILENAME FIX

The name of the most recent move saved with "is" (Incremental Save) is now displayed on the control panel screen. If the name is <<bracketed>> it means that no move is present in the system, and the name is the name of the last move saved.

Installation notes
May 11, 1997

Most of the files on this site are self extracting .exe files containing recent versions of software designed to be used with Kuper Controls RTMC48, RTMC16, and KRJ32 motion control cards. These programs will not run unless one of these cards is installed in the computer.

The file names are coded as follows:

The first six letters are the year, month, and day the software was posted. Letters 7 and 8 are the number of axes, or the type of card the software is intended to operate.

For example:

```
      97051116.EXE
```

-Was posted on May 11, 1997, and is intended for use with a 16 axis version of the RTMC48 or RTMC16 Card.

```
      970301KR.EXE
```

-Was posted on March 1, 1997, and is intended for use with the KRJ32 Card.

```
      970301HY.EXE
```

-Is for a hybrid combination of RTMC48 and KRJ32 Cards in the same computer.

In general, if you have just a RTMC48 Card, download a file with letters 7 and 8 being "16", "24", or "48" depending on the number of installed axes. If you have a KRJ32 Card, letters 7 and 8 should be "KR". For RTMC48 / KRJ32 hybrid systems, letters 7 and 8 should be "HY".

To upgrade your current program, simply run one of the above programs in your Kuper directory, one time only. The program will overwrite your existing RTMC130.EXE or KRJ130.EXE file. From then on, simply run RTMC130.EXE or KRJ130.EXE. To confirm the update, check the "Compiled on:" date displayed in the sign-on box.

The above files only contain the RTMC130.EXE or KRJ130.EXE program. If you are configuring a system from scratch, you will also need several files contained in UTILARC.EXE. Run utilarc.exe one time only in your current Kuper directory.

If your system hangs up after exiting the Kuper Software, you should download the file DOS4GW.EXE dated 7-11-95 and place it in your Kuper directory, overwriting the previous version.

If you have an old RTMC16 Card, download XXXXXX16.EXE. In order to configure the software for the older card, there must be a file named RTMC16 in the current Kuper directory. The contents of the RTMC16 file do not matter, all that matters is its presence. To make a RTMC16 file, from the DOS prompt, type:

```
      copy rtmc.env rtmc16
```

and press the Enter key. Note that the RTMC16 file must have no extension.

Improved camera head back off technique
November 21, 1997

In stop motion animation, it is often convenient to move the camera head away from its normal shooting position between exposures, to give animators easy access to the model. It has long been possible to do this with the "INDEX" function available on the Animation Setup Screen. The current version of RTMC130.EXE adds a more direct method.

The new method allows the animator to press a new hardware "backoff" switch at any time the system is waiting for the normal "shoot" switch. The backoff switch causes the system to move a specified distance away from its programmed shooting position. Pressing the backoff switch a second time returns the axis to its normal shooting position. The axis may be moved back and forth between "backoff" and "shooting" positions any number of times between frames. The shoot switch will not respond when the axis is in its backoff position. Note that the backoff switch is only effective while the system is waiting for the normal shoot switch.

The Animation Setup Screen now has two new input boxes. "BACK OFF AXIS" allows the operator to specify a single axis to exhibit the backoff behavior. "BACK OFF DIST" is how far the axis will move away from its normal shooting position when the backoff switch is pressed. BACK OFF AXIS may be any axis, whether or not it has a move programmed. BACK OFF DIST may be any distance, plus or minus, relative to the programmed shooting position. It is not necessary to program any key frames for this function.

The switch connects between pins 14 and 16 on the RTMC Logic Connector. Pin 14 is ground. The switch should be connected so that pins 14 and 16 are connected when the switch is pressed. As with the shoot switch, the cable should be shielded, and the shield wire should be connected to any screw on the computer case. Two wires inside the shield should connect to pins 14 and 16. Do not connect the shield to any pin on the RTMC Logic Connector since this will conduct excessive electrical noise through the motherboard.

Lower Speed Sync
December 19, 1997

IT IS NOW POSSIBLE TO SYNC TO EXTERNAL CAMERAS DOWN TO AT LEAST 0.5 FPS

Previous RTMC130.EXE software versions could not sync to external camera motors at speeds below 16 FPS. This version extends the sync range down to at least 0.5 FPS.

Please note that most sound speed camera motors are highly optimized to run speeds in the 24 to 30 FPS range. At speeds above and below this range, they may exhibit considerable speed fluctuation. It may not be possible to achieve a satisfactory sync with a very unsteady motor . Please be sure to test the specific camera motor and the specific speeds you plan to use well before the shoot.

The sync procedure is the same as before. The "once-per-frame" shutter pulse from the camera is connected to the DB15 Accessory Connector on the RTMC48 Card (see the manual for pinouts). Change the VISUAL FPS speed on the Control Panel Screen to the same speed as the camera motor. Roll the camera, and the words "SYNC LOCK" will appear in the box just above the REV-STOP-FWD boxes as soon as the camera and the motion control system are synchronized. This may take several frames. Whenever "SYNC LOCK" is displayed, you can run the move in the normal way and be assured of accurate sync provided the "SYNC LOCK" light remains lighted during the entire move. If you need to match elements shot with a stepper motor equipped camera, you will need to do carefully calibrate the shutter phase angle as described in the manual.

The "SYNC FPS" box on the Control Panel Screen displays the actual FPS of the incoming signal. If the camera is running at speed significantly different than the speed claimed by the camera motor, it may not be possible to obtain sync lock. In this case, enter the actual incoming FPS speed in the VISUAL FPS box on the Control Panel Screen.

Model Nodal Compensation
January 14, 1998

This version implements model nodal compensation with model offsets in x, y, and z relative to a pitch / yaw model mover, such as a Worrall head type device. The offset conventions for the model are the same as those for the camera. The nose of the model equals the lens of the camera, so that in the home position the model is flying away from the camera and parallel with the track.

Under "HardSet", the "Nodal Comp Axes" and "Nodal Parameters" boxes have been expanded to include model parameters.

Two model mounting techniques are supported:

1. "ENABLE PITCH & YAW MODEL COMP" assumes the model is mounted so that the mover Pitch (aka Tilt) axis causes the nose of the model to pitch upwards.

2. "ENABLE TILT = ROLL MODEL COMP" assumes the model is mounted so that the mover Pitch (or Tilt) causes the model to roll.

In either case, the model offset parameters are measured from the actual center of rotations, which for a Worrall style device will be located above the camera platform.

Helper buttons
March 5, 1998

This version of RTMC130.EXE includes "Helper Buttons" to aid in positioning when the camera is tilted, panned, and/or rolled. The Helper Buttons are located on the control panel screen, and are also available from the Jogbox.

"PUSH IN / OUT" moves the camera directly towards or away from the center of view, regardless of camera orientation. "LEFT - RIGHT" moves the view screen-left or screen-right. "UP - DOWN" moves the view screen-up or screen-down.

The helper buttons are available only when the virtual axes are turned on. Do not confuse the helper buttons with axes, since helper buttons can not be keyframed or edited. They are only intended to simplify positioning in the course of normal key frame work.

From the Jogbox, the helper function is toggled on and off by holding down "JOG32" and then tapping "OPEN / CLOSE SHUTR". When on, "1+-", "2+-", and "3+-" become PUSH IN/OUT, SCREEN LEFT/RIGHT, and SCREEN UP/DOWN. Other keys retain their normal functions.

The helper functions are not yet available through the encoders.
This marks the beginning of several new "target-tracking" related functions. Stay tuned for updates of this Beta version.

Sorensen Zebra canned setup
March 5, 1998

There is now geometry support for the Sorensen Zebra. Select "HardSet:Nodal Parameters" and click on the "Sorensen Zebra" selection. For the Zebra only, you must also have two special calibration files present in your Kuper directory: "Zebraz.tab" and "Zebray.tab". You may download these files from www.kupercontrols.com. If you are not using a Zebra, we suggest you do not install these two files your Kuper directory, to minimize run-time memory usage.

The "Fulcrum" motor must be calibrated so that one unit of travel in the UP direction moves the Fulcrum post up 1.000 inch. Measure the extension of the Fulcrum post, and not the height of the camera! If you wish to work in centimeters, please contact us or check the web site in a few days for centimeter versions of Zebraz.tab and Zebray.tab.

Model Nodal Compensation
January 14, 1998

This version implements model nodal compensation with model offsets in x, y, and z relative to a pitch / yaw model mover, such as a Worrall head type device. The offset conventions for the model are the same as those for the camera. The nose of the model equals the lens of the camera, so that in the home position the model is flying away from the camera and parallel with the track.

Under "HardSet", the "Nodal Comp Axes" and "Nodal Parameters" boxes have been expanded to include model parameters.

Two model mounting techniques are supported:

1. "ENABLE PITCH & YAW MODEL COMP" assumes the model is mounted so that the mover Pitch (aka Tilt) axis causes the nose of the model to pitch upwards.

2. "ENABLE TILT = ROLL MODEL COMP" assumes the model is mounted so that the mover Pitch (or Tilt) causes the model to roll.

In either case, the model offset parameters are measured from the actual center of rotations, which for a Worrall style device will be located above the camera platform.

Virtual Axes Calibration
September 10 1999

It is now possible for users to calibrate boom lifts with unusual geometries, for use with the Kuper Virtual Axis System. This new scheme is based on the existing scheme originally designed for the Sorensen Zebra device. The user takes measurements which relate camera height to a particular amount of boom lift motor rotation, and camera height to the amount of track movement required to move the camera on a vertical straight line. The user copies the measurements into a file named "NSGEOM.TAB". Each time the Kuper software boots up, it processes the measurements automatically.

Here is an example of the required user measurement file:

```
  ;Acme Boom Camera
  -42      -9.725      -6.835
  -36      -8.444      -4.931
  -30      -7.128      -3.372
  -24      -5.776      -2.141
  -18      -4.390      -1.191
  -12      -2.963      -0.536
  -6       -1.502      -0.142
   0        0           0
   6        1.525      -0.111
   12       3.081      -0.473
   18       4.667      -1.087
   24       6.280      -1.946
   30       7.910      -3.052
   36       9.566      -4.416
   42      11.242      -6.060
   48      12.924      -7.980
   54      14.612     -10.243
   60      16.292     -12.828
   63      17.135     -14.262
```

The line "Acme Boom Camera" is the first line in the file. The line "63 17.134 -14.262" is the last line. When the Kuper Software boots up, it will display "Acme Boom Camera" in a sign-on dialogue box, and read the subsequent numbers as data.

The numbers are arranged in three vertical columns, and several rows. The first vertical column is camera height. When the boom arm is level, camera height is 0.000. The second vertical column is the amount of boom lift motor travel required to reach the height in the first column. The third vertical column is the amount of track movement required to keep the camera moving on a vertical straight line. In order to place the camera 48 inches exactly above its "0.000" position, the boom lift motor must move to a position of 12.924, and the track motor must move to a position of -7.98. The track number assumes the boom is lined up with the track, as it must be when taking the measurements for the table. In actual use, the Software modifies the track number based on the left and right swing of the boom. These examples are in inches, but centimeters, millimeters, or any arbitrary unit of measurement may be employed.

Note that approximately in the middle of the list, there is a line which reads "0 0.000 0.000". This is the "home position" for the vertical lift system. This "zero line" must always be present in the file. The boom must be completely level at the home position.

HOW TO TAKE THE MEASUREMENTS:

Here is one steadfast way to get your measurements. There may be others.

One of the big advantages of this scheme is that it automatically compensates for mechanical bending and other non-linear effects due to gravity, relativistic space warpage, and other effects. To get the best possible data, configure the system the way it would normally be used. Make sure the weight and load distribution of the boom is the same as it would be when actually shooting film. Mount a typical magazine, preferably with a dummy load to simulate the weight of the film. All the acessories, wires, cables, video taps, and so on normally attached to the boom should be in place. Dont forget the lens.

Attach a plumb bob to the camera. One scheme is to attach the string to a lens cap mounted on the lens. The length of the string should be easily adjustable, without changing the attachment point. The plumb bob will be used to measure track travel.

Be sure all the "real" motorized axes are accurately calibrated in the desired units. The boom lift axis should be calibrated in units of extension, rather than how much it lifts the camera. It is important to note that if you change the "Pulses Per Unit" of the track or lift axes after you have taken your measurements, the measurements will no longer be valid.

Jog the system around for a few minutes so all the axes are settled in.

We're going to take our numbers with the virtual axes turned off. Make sure you have an accurate measuring device.

Place the boom in the desired home position. Be sure the boom is perfectly aligned with the track. Point the camera line of sight directly parallel with the track.

We need to take all our measurements at integer heights, without decimal places.

Zero all the motors. The screen positions must all read "0.000".

Exactly measure the height of the camera. Make an accurate mark on the floor, exactly under the point of the plumb bob. Of course, the height measurement won't be zero in real world units, even though the motor position reads zero. After you have taken all your height measurements, you'll have to subtract this inital height measurement from each height measurement, to offset the original measurement to 0.000. Somewhere in the middle of your data sheet, write down "0.000" (the height of the camera lens), another "0.000" (the position of boom lift motor), and a final "0.000" (the track position).

Of course, write everything down so you get a table like the example table above. Now move the camera down as low as it will go. Move the track so the plumb bob is back hanging over its mark. Measure the lens height. Remember that the height measurements must have all zero's in the decimal places -- this means that for this lowest measurement, you will probably have to cheat the boom slightly higher than it can acutally go. Record your numbers -- lens height, boom lift motor position, track position. Now move up through the available range of movement at approximately even intervals. It is strongly recommended that you do not take height measurements too close together. Experience has shown that intervals of about 5 or 6 inches (125 to 150mm) are about right. The pattern is:

1. Adjust the height of the boom upwards by five or six inches.

2. Move the plumb bob so it's over its original mark.

3. Record the lens height, the position displayed for the boom lift, and the position of the track.

When you've got all your measurements, subtract the tape-measured lens height distance for the axis "0.000" position from all the lens height measurements. This will make the "0.000" position lens height measurement 0.000.

Enter your numbers into a text file to look like the one above. At the top of the list, put a ';' followed by a very brief description of what system this table is for. This short description will be displayed when the system boots up. Make sure there are no blank lines. Be sure to save this file as a pure ascii text file, with no formatting, fonts, or other foolishness.

For this table to effective, you must have selected "Table N/S, like GenuFlex and Zebra" in "HardSet:Setup Nodal Paramters". When the Kuper Software boots up, it looks for "NSGEOM.TAB" in the Kuper directory. If "NSGEOM.TAB" is not found, it looks for "ZEBRAZ.TAB" and "ZEBRAY.TAB" which are equivalent canned files for the Sorensen Zebra. Since the Kuper Software only looks for these files at boot up, you must have selected "Table N/S, like GenuFlex and Zebra" the last time Software was running, and then saved the change with "UtilFiles:Save Default Setup" -- otherwise none of these files can be loaded.

Be sure that the proper boom length and diving board offset are filled in in the "HardSet:Nodal Parameters" dialogue box.

If you are using a single computer to control several systems, you can switch between different geometry tables by saving each table under a unique name, and then copying (not re-naming) the appropriate table to "NSGEOM.TAB" as needed. The boot up dialogue box will tell you if the right table is being loaded.

LIMITATIONS

In the current design, the height of "0.000" must be included in the data file.

You need to have a least 8 or so data points. The above example has 19 entries, which is more than adequate. Internally, the Kuper Software curve fits your sample set into long, high resolution tables.

The system does not like blank lines in the data file, especially at the end.

When the system reaches the limit of travel programmed in your NSGEOM.TAB table, it will come to a sudden stop, without eases.

ADVANCED TOPICS

If you would like to view the shape of the internally generated tables, include the word "VIEW" at the top of your data file, on its own line with no leading characters. Just after you boot up, the tables can be viewed on axes 1 and 2. Pay special attention to the velocity curves. If the curves aren't smooth, your data may be faulty. Be careful! There will be data on these axes which could lead to accidents during move building. After examining the curves, us "if 1 0 2 0" to pre-clear the axes to 0.000. Edit out the "VIEW" line for general usage.

You can also pre-process you data by typing in your measurements as key frames into the Kuper Software. Type in the lens height numbers as key frames, and either the boom lift motor position, or track position as positions. You will have to offset your frame numbers, so no frame number is less than 0.000. If your bottom position is "-42.00", you must add 42.00 to all your keyframe numbers before typing them in. Examining the curves for smoothness, especially on the velocity graph, will help you sort out bad data. Once you are finished tweaking the data, save it as an ASCII File, centered on shutter closed. You'll have to then text edit the ascii file, adding keyframe numbers and the ";Banner" line.

Updated Flash and Cap functions
Dec 29, 1999

The "f2" command allows the Flash and Cap ouputs on the Kuper RTMC48 Card Accessory Connector to be programmed as two independent strobe ouputs, in units of degrees of camera rotation.

For examples:

```
      f2 170 190 340 350
```

would causes the "Flash" strobe to assert between the shutter angles of 170 and 190 degrees, and the "Cap" strobe to assert between 340 and 350 degrees.

The two strobes are of equal capabilities and performance. All combinations are possible within the following constraints: For each output, the ending angle must be larger than the starting angle; The "Startin Angle" parameter for each strobe must occur with the range of 0.1 and 359.7 degrees; The "Ending Angle" parameter must be between 0.2 and 719.3 degrees. Thus, while each strobe must start within the current frame, it may carry over into most of the next frame. For instance, the following is permissible:

```
    f2 170 190 350 370
```

in which case the "Cap" signal is asserted on both sides of the shutter closed home position.

Some example combinations:

```
    f2 340 350 10 90    //within each frame, cap fires before flash
    f2 10 90 340 350    //with each frame, flash fires before cap
    f2 90 270 100 280   //cap and flash can overlap
    f2 90 370 170 450   //overlap and last into next frame
```

DAC outputs as velocities
March 8, 2000

Dac outputs can be made to output a voltage relative to the axis velocity rather than the axis position.

Create a pragma file named "veldac" with no extension. Veldac contains a list of characters which are all either '1' or '0'. Each character corresponds to an axis. If the character is '0', the dac output is proportionate to position. If the character is '1', the dac output is equal to the absolute value of the velocity of the axis. The characters must have no spaces in between, and the very first character must be a '1' or a '0'.

In MouseJog, select "PatchDac" and then click on the numeral to your left of the axis name, so the numeral is highlighted yellow. Click ok. The axis will now output to the dac, as setup with the normal "DacAddr" and "DA720" files.

In order to scale the velocity dac output, in AxisSetup program the axis Slew Speed to 491400 if the Data/Visual is 5, or 393120 if the Data/Visual is 4. This will cause the screen Velocity column to display "4095" when the axis is at maximum speed. Test the actual shooting FPS speeds you plan to use.

Various new commands
March 23, 2000

In stop motion, it is possible to produce extra gentle whip-open camera ramps by adding a pragma file named "softwhip" to the Kuper directory. Softwhip should contain an "exposure time" which characterises the desired whip-open ramping characteristics. The exposure time in the softwhip file in no way controls the actual exposure time, but rather just the character of the ramps which will be applied when the camera whips-open. For instance, the default whip-open ramps are equivalent to the ramps for a 0.125 second exposure, regardless of what the actual whip-open exposure time is. If the softwhip file contains the number 0.250, the whip open ramps will be typical of the ramps for a 0.250 second exposure, or half as abrupt as the standard 0.125 second equivalent ramps.

IMPORTANT NOTE: if the softwhip file contains a value greater than 0.125, the actual exposure time will be slightly longer than specified in the animation setup screen. For ballpark estimates, the additional exposure time will be roughly equivalent to:

```
      (contents_of_softwhip_file - 0.125) / 0.3
```

So if the softwhip file contains the value 0.250, the actual whip-open exposures will be about 0.04 second longer than actually specified in the animation setup screen.

****

The "pv" command controls the overall system PUSHIN speed. "pv 2" doubles the speed. You can use any positive number. Note that the "PUSH" controls only work when the virtual axes are operating.

****

The attached RTMC130.exe file expands the old move browsing function, modified to allow parts of the FPS control file to be cue'd with the shoot switch.

Suppose we had a shot which had to start shooting at 60 FPS, and then ramp down to 24 FPS on cue.

Program a FPS Control Axis with the following keyframes
0 1
30 60
45 24

and do an EasyFit. Also program the Exposure Comp Axis as before, where the frame numbers are taken to be the current browsing FPS, and the positions the desired compensator axis position for the current browsing FPS. EasyFit is the best curve fit method for this, because it always hits the exact speed. Do not use LineFit, which was recommended for the older method.

Click ModeSet:Start Browing.

Click the new "Enable Shoot-Switch Cueing" checkbox.

Select the FPS CONTROL AXIS and EXP COMP AXIS.

Make "Start Frame for Cue 1" = 0
Make "Start Frame for Cue 2" = 30
Make "Start Frame for Cue 3" = 45

Release the shoot switch button. Click on BEGIN. The systems start turning over at 1 fps, the value for Cue frame 1. On the first switch hit, the system ramps up to 60 FPS in 30 frame periods. One the second switch hit, the system ramps down to 24 fps in 15 frame periods. Up to 8 speed transitions can be programmed for a single run.

So basically the Exposure Compensation Axis is programmed as a series of speed transitions butted up against each other, minus the constant speed holds. The "Start Frame for Cue XX" values control how far into the Exp. Comp. Axis move the system will advance on each key press.

Prototype Genlock
December 3, 2000

This version contains a prototype genlock ability. The motion control can sync lock to video inputs at the basic video frame rate, divided by an integer value. For instance, a 24.000 fps video signal can be divided by 16 to obtain a motion control sync lock with the motion control running at 1.5 frames per second, with 16 video frames per each motion control move frame, i.e. (24.000 / 16 = 1.5)

The divisor is programmed with the 'gl' (GenLock) two letter command. For the above example, "gl 16" would cause the motion control to sync lock to a 24.000 fps video signal, with the motion control fps set to 1.5.

The "SYNC FPS" box on the Kuper Software screen will display the incoming video fps divided by the "gl" value.

To turn off the genlock function, issue the command "gl 1".

At present, the following limitation apply:

--The signal source must be extremely steady, such as a video signal. This version can not track changing frame rates.

--After programming the "gl" value, the operator must enter the frame rate into the "VISUAL FPS" box on the Kuper screen. If the camera is running, he can simply copy the displayed "SYNC FPS" value to the "VISUAL FPS"box.

--For large "gl" values, it may take up to a minute for the motion control and camera signal to come into sync.

Cue Axis
December 21, 2000

This capability is only available in 48 axis versions of RTMC130.exe.

It is now possible to pre-store a canned move on the Virtual axes and Real axes 7 through 16, so that any time during a normal move playback the canned section is inserted into the move and played back when the operator issues the "qa" command at the keyboard. This allows the canned section to start playback when some unpredictable event occurs during the time the motion control move is running. This is a true "add" function, so that the canned move data is added into any existing data on the axes.

In practice, move making is broken down into two phases:

1. Making and storing the canned section.

2. Making and running the production move, with the canned section inserted as needed.

MAKING AND STORING THE CANNED SECTION

Make the "canned" move on any of the Virtual axes, and any of axes 7 through 16.

Save the move for future reference.

Issue the "qd" command. This copies the move into the storage axes, and initializes the original Virtual and Real axes to the first frame of the canned move. The operator should only store the canned move using the "qd" command, rather than using any other method such as "cd".

Use "st" SaveTemp command to save a quick copy of the canned move. You will probably have to reload the temp move several times during shooting.

MAKING THE PRODUCTION MOVE

With the canned move stored as above, start the motion control in the normal way. You may be recording joystick axes, and playing back moves on any axes, including axes which will be affected by the canned data.

To start playing the canned move, issue the "qa" command at the keyboard any time after the normal move playback starts. The instant the final "a" is entered, the canned move will start playing back, and will be ADDED into the move file for the Virtual axes, and axes 7 through 16.

When the move is stopped, the inserted canned data will be reflected in the move files. Subsequent playbacks will contain the canned portion of the move without using the "qa" command.

In normal production there will be several false starts and unsuccessful takes. For each unwanted take where the "qa" command was issued, the operator MUST reload the temp file ("lt"), as the only way of undoing the unwanted canned data inserted into the file. Failing to reload the temp file will leave portions of the canned data in the main move file -- this will lead to unwanted motion on the next playback, and at worst may cause major body slams to the system. Each time a successful take is achieved, save the move in the normal way, and then reload the temp file.

The canned data is stored as follows:

AXIS	STORED_ON
VTrack,	Parm1
VNS,	Parm2
VEW,	Parm3
VPan,	Parm4
VTilt,	Parm5
VRoll,	Parm6
Axes 7 through 16,	Parm7 through Parm 16

Serial Focus Data
January 21, 2001

The Kuper software can now transmit position data for an external focus axis and "exposure control" axis over a serial port.

The data is in ascii format. A typical stream is:

The first four characters are hexadecimal numbers in the range of 0000 to ffff, which is scaled by the receiving device to the maximum available focus displacement. This number is always padded to four characters.

The fifth character is an ascii space

The sixth and seventh characters are hexadecimal numbers in the range 00 to ff, representing the available range of exposure control. This number is always padded to 2 characters.

The 8 eighth character is a binary "0x0a"

This function is turned on by the presence of a pragma file named "serfocus" located in the same directory as the Kuper software. There are four possible parameters, the first two of which are required. For example:

```
13 14 19200 2f8
```

--Axis 13 is the focus axis. A value of 0 indicates no focus axis. This a required parameter.

--Axis 14 is the exposure control axis. A value of 0 indicates no exposure axis. This a required parameter.

--The baud rate is 19200. This is an optional parameter. If not specified, the default baud rate is 19200, and the default serial port base address is 2f8.

--The base address for the serial port, in hexadecimal format. This is an optional parameter, but if specified you must also specify the baudrate, above. The default is hex 2f8, the standard IBM PC address for COM2.

The serial port must contain at least a 16c550 equivalent modem. Most PC's with a PCI bus will satisfy this requirement.

In the Kuper software, you can map the Kuper axis position numbers to the available range of the focus and exposure control data stream. Click "HardSet" and then "Software Position Limits". In the two columns labeled "--Software Limits--" set the left column to the Kuper motor position you want to map to focus position "0", and the right column to the Kuper motor position you want to map to focus maximum (ie. hex ffff). If you select "0.000" and "100.000", the focus data stream will send "0" when the Kuper axis is at "0.000" on the Kuper screen, and "ffff" when the Kuper axis is at "100.000" on the Kuper screen. You can map any range of screen values you find pleasing.

In the present version, the serial stream is transmitted every 24 data frames. This will be 4 times per second when Kuper "DATA/VISUAL" = 4, or 5 times per second when DATA/VISUAL = 5.

Timecode functions
April 21, 2003

ti Timecode I/O. Enter a 0 to receive timecode from an external source, enter a 1 to generate timecode. Timecode will start transmitting with the value in the computer's hardware time-of-day clock. Use "UtilFiles: Save Default Setup" to save this setting.

th Timecode Hours. Enter HOURS MINUTES SECONDS FRAMES separated by spaces. Only valid when transmitting timecode, see the 'ti' command. The The transmitted timecode is updated to the new values when the the Enter key is pressed. 'th' does not update the computer's hardware time-of-day clock. To adjust the computer's time-of-day clock, use the TIME and DAY commands at the DOS prompt.

DMX and FiZ streams from K2001 board
January 29, 2004

Kuper RTMC130X.exe now supports DMX512 devices (only on a K2001 board - ed.). Axes 33 through 48 are sent out as a DMX512 stream, controlling up to 16 DMX devices. You must use the 48 axis version of the Kuper software -- this version may be used irregardless of how many axes you actually have.

HARDWARE

The DMX512 signal is sent through the Kuper2001-I serial connector, please see the .pdf drawing for pinouts. Supposing that the DMX channel has been brought out via a DB9 insulation displacement connector...

DB9 pin 1 connectors to DMX common, XLR-5 pin 1
DB9 pin 5 connects to DMX-, XLR-5 pin 2
DB9 pin 4 connects to DMX+, XLR-5 pin 3

The last (and only the last) device in the DMX512 network daisy chain must be terminated with a 120 ohm resistor. No more than 32 physical devices should be plugged into the same daisy chain. Only 16 devices can be addressed by the Kuper software.

SOFTWARE

Create a "pragma" file in the same directory as your Kuper software. The file is named "dmxcard". Use the "copy con" command:

```
   C:/Kuper>copy con dmxcard {press the Enter key}
   e800 {press the Enter key}
   {press the single key labeled "F6"}
  "1 File(s) copied"
   C:/Kuper>
```

The dmxcard pragma file can contain a second parameter to set the bit resolution of the DMX signal. The default resolution without a parameter is 8 bits. For 16 bit resolution, the dmxcard pragma would look like this:

```
   e800 16
```

Run the Kuper RTMC130.EXE software. You must first calibrate the range of axes 33 through 48, the sixteen axes that will be sent to the DMX512 network:

1. At screen lower right, click on "HardSet"

2. Click on "Software Position Limits"

3. Press the "F3" key on the keyboard so that axes 33 through 48 are displayed.

4. In the two "--Software Position Limits--" columns, make the left number be "0.000" and the right number be "100.000"

5. Click "OK" down at screen lower right.

6. Click "UtilFiles"

7. Click "Save Default Setup"

8. Click "OK"

This causes the DMX512 device to be at their lowest level (ie. off) when a Kuper axis is at 0.000, and at their maximum level when the Kuper axis is at 100.000. Kuper axis positions less than 0.000 will be truncated to DMX512 minimum, and Kuper axis positions greater than 100.000 will be truncated to DMX512 maximum. You can select other ranges if you want.

In normal 8 bit resolution, The Kuper software sends axes 33 through 48 out as DMX channels 1 to 16.

HOWEVER, IF THE "DMXCARD" PRAGMA FILE CONTAINS A "16" ENTRY:

The Kuper software sends "16 bit, high byte" DMX512 data. This gives enough resolution to work with certain high resolution DMX512 devices such as motion controlled lights. However, most DMX512 lights are only "8 bit" devices. The Kuper generated DMX512 data can work with both 16 bit and 8 bit devices by simply assigning the DMX512 device addresses in the appropriate way. For 8 bit devices:

Kuper Axis	DMX512 8 bit Address
33	01
34	03
35	05
36	07
37	09
38	11
39	13
40	15
41	17
42	19
43	21
44	23
45	25
46	27
47	29
48	31

If you connect a 16 bit DMX512 device, assign the "High" and "Low" bytes as follows:

Kuper Axis	DMX512 "High"	DMX512 "Low"
33	01	02
34	03	04
35	05	06
36	07	08
37	09	10
38	11	12
39	13	14
40	15	16
41	17	18
42	19	20
43	21	22
44	23	24
45	25	26
46	27	28
47	29	30
48	31	32

No software changes need to be made to change from 16 bits to 8 bits, it's strictly a matter of how the DMX512 address are assigned.

NOTE: while only 16 DMX512 channels are sent by the Kuper software, you can plug in multiple devices with the same address. You could, for instance, assigned 32 lights to Kuper axis 33 by giving all the light the DMX512 address of "1". All the lights would fade up and down in sync with the movement of axis 33. Just remember that a maximum of only 32, DMX512 devices can be physically plugged into the network at the same time.

Serial Position Streaming
January 29, 2004

The Kuper2001-I card can now stream serial position data throught its RS232 port. The position data is in ASCII format. Up to 12 axes are transmitted in two packets per frame. The first packet is sent at shutter open, the second packet at shutter close.

The pragma file "k2ser" must be present in the Kuper directory to turn on this feature. "k2ser" can be created using the copy con command...

```
   C:\KUPER>copy con k2ser {press the Enter key}
   3 115200 {press the Enter key}
   19 18 17 22 20 21 1 2 3 4 5 6 {press the Enter key}
   {press the "F6" key}
      {press the Enter key}
```

The first line, "3 115200", sets the RS232 channel and baudrate. The channel must always be "3". Other baudrates may be selected, but baudrates lower than 115200 may not be able to completely send one packet before the next one starts. Slower baudrates will require that smaller numbers of axes be sent.

The second line specifies which axes will be sent. The axes can be in any order, and will sent in the order specified.

When the camera is rolling, an "R" will appear at the start of each line. When the system is pre-rolling, a "P" will appear at the start of each line. Otherwise, an "S" will be the first character in each line. The first parameter after this letter is the current frame number.

The datastream may be viewed with the Windows "Hyperterminal" program for familiarization.

Preston Format Streaming
January 25, 2004

Kuper RTMC130.exe can send any three axes over an RS422 port in Howard Preston's "FI+Z" format. Many thanks to Howard for providing the FI+Z technical data.

Create a "pragma" file in the same directory as your Kuper software. The file is named "fiz". Suppose we want to send axes 17, 18, and 19...

```
      C:/Kuper>copy con fiz {press the Enter key}
      e800 17 18 19 1
      {press the Enter key}
      {press the "F6" key}
      {press the Enter key}
```

```
"1 File(s) copied" C:/Kuper>
```

The "e800" is a constant, and should always be the first item on the line.

At this writing it is not known whether the protocol uses "hi byte first" or "lo byte first" sequencing. The "1" at the end of the "fiz" data above forces "hi byte first" format. If the axes seem to lurch around, change the "1" to a "0" for "lo byte first" format.

Run the Kuper RTMC130.EXE software. You must first calibrate the range of the three FIZ axes:

1. At screen lower right, click on "HardSet"

2. Click on "Software Position Limits"

3. For each of the three axes, in the two "--Software Position Limits--" columns, make the left number be the minimum position for the axis, and the right column be the maximum position. If you want to map the range of a FIZ

axis to -100.000 to 100.000 on the Kuper screen, enter "-100.0" in the left column, and "100.0" in the right column.

4. Click "OK" down at screen lower right.

5. Click "UtilFiles"

6. Click "Save Default Setup"

7. Click "OK"

The normal convention for FI+Z axes is:

```
      First axis = Iris
      Second axis = Focus
      Third axis = Zoom
```

However, for certain FI+Z compatible devices these axes may be considered as having general purpose functions.

Kuper handles all axes in FI+Z "position mode" only.

Whenever the "Camera" axis on the Kuper screen is moving, the "Camera Run/Stop bit" in the FI+Z date stream will be set.

New Timecode commands
February 4, 2004

NEW TWO LETTER COMMANDS, KUPER2001 CARDS

ti	Sets Kuper2001 timecode direction to input or output
th	Sets the Hours, Minutes, Seconds, Frames when Kuper2001 is generating timecode.

Kuper 2001 Serial Commands
and Serial File Loading
July 8, 2004

Kuper RTMC130x.exe for the Kuper2001 Card now accepts many standard Kuper two letter commands through the PC's COM2 port. Also, it is possible to download Kuper ASCII move tiles directly to the Kuper2001 system through the same COM2 port.

Almost any 2 letter command that can be typed from the keyboard can be issued over the serial port. For instance,

```
   gm track 10.333 pan -45.334 tilt 0.000
   ms {until the axes are all in position}
   wr1 0.125
```

would position the axes at the above positions, wait for all the axes to reach position, and shoot one frame forward at 0.125 seconds.

Kuper and the host computer must be connected through a null modem connector. Pins 2 and 3 swap between Kuper and host, and at both ends.
1,4,6 are wired together, pins7 and 8 are wired together.

The special 2 letter command "ms" requests motor busy status. If you send "ms", Kuper returns

```
   mS0010010000000000
```

which means that the third and sixth motors are busy, in this case not yet at their slew destination positions.

In the above "gm", "wr" sequence, you would poll for axis motion completion just after issuing the "ms‟ command, and only "wr" the frame when you have received

```
   "ms0000000000000000".
```

Turn the serial function on with the "so" command.

```
  so1
```

turns the serial stream on,

```
   so0
```

turns the serial stream off.
For diagnostics only,

```
   so1 1
```

will tum the serial stream on, and place each incoming command line received in a dialog box.

A Kuper ASCII data tile can be downloaded into Kuper by simply sending the data file in over the PC's COM2: port, provided the Kuper system is already in serial mode via the "so1" command having been typed at the keyboard, or the presence of a "extcom" pragma file in the Kuper directory at boot up time.

At present the ASCII tile must comply rigorously to the Kuper standard, although Kuper 2-letter commands can be placed in front of the ASCII file formatted data. During download the “VisualFrame:Total" box will increment with each new line received. Once the tile has been completely downloaded, the Kuper system will process the input data with "centered on blur" conversion rules.

lt is recommended that a "nm" command always be issued before loading a new ASCII tile to avoid mixing parts of old and new moves. In practice, the ASCII tile may be moditied with a "nm" command placed on a line just above the "Axes =" line.

Please note in the "Axes =" line, the 'A' is upper case, 'xes' is in lower case, and
there is a space between the 's' and the '='. Kuper uses this format to identify an arriving ASCII tile. The present Kuper ASCII file deinition does not require an "end of tile" character to be present at the end of the tile, so the receiving Kuper
system uses a 1.5 second timeout with no new ASCII data to tell when the file has tinished transferring. So the sending device must be sure to pause at least 2 seconds between sending the ASCII file and issuing any new commands. Also, the sending device must not insert any waits greater than 1 second in its data stream until it is finished sending the entire tile. If desired, the sending device may append two characters "lq" to identify the end of ASCII tile, which will skip the timeout period.

Only the PC's COM2 port may be used, only 9600 baud, commands must be in lower case only, you must use null modem adapters both ends, and you need to have at least a 16550 UART with a 16 character or larger buffer, which most motherboards have. The Kuper system will probably lock up if you don't have a null modem connector on the Kuper serial port.

The Kuper system will automatically boot up with serial communications turned on if there is a tile named "extcom" in the Kuper directory. No extension, tile content doesn't matter.

The Vlhndows Hyperterminal program can be used to exercise the 2-letter serial commands. However, it is unfortunate that the Hyperterminal "Transfer: Send Text FiIe" menu item can not be used to transfer ASCII data tiles, since it inserts additional characters into the data stream, and is glacially slow. For testing from Windows, invoke a DOS command box, and tirst issue the command "mode comX:9600,n,8,1,p" command (where X is the Windows system serial port number). Then you can type "copy xxx.xxx comX:" where xxx.xxis the Kuper format ASCII tile, and X is the Windows com port number.

There are 2 letter command aliases for many of the click boxes on the Kuper screen. Here are a few useful commands...

```
"as" All axes Stop. Use sparingly.
"vp0" sets the current "Visual Frame:Present" to the start of the move, frame 0.
"nm" clear out the move in memory, set the move length to 0.0. "new move"
"fd" same as clicking the FWD button.
"sp" same as clicking the STOP button.
"rv" same as clicking the REV button.
"re1 3 4" places axes 1,3, and 4 in record.
"pl2 6" places axes 2 and 6 in playback.
"ne7 8 9 10 11" places axes 7 and 11 in neutral.
"en1 2 3 4 5 6" enables axes 1 through six.
"ea2 10" assigns encoder 10 to axis 2.
"es3 2.5" set the joystick gain (sensitivity) for axis 3 to a value of 2.5.
"ev6 v" puts axis 6 in joystick velocity mode.
"ev8 p" puts axis 8 in joystick position mode.
"ef1 4" sets the smoothing for axis 1 to a value of 4.
"f+" moves one frame forward in stop motion mode.
"f-" moves one frame reverse in stop motion mode.
"gs" moves all enabled axes to the ramped move start position for the present
"Visual Frame."
"za" sets all axis move positions to 0.000.
"im3 333 4 444" sets the position for axis 3 to 333.000 and axis 4 to 444.000.
```

plus many of the old favorites like "mm‶ "gm" "pf‵ etc.

Linear Ratio command
December 7, 2004

The new "lr" Linear Ratio command slaves one axis to another by a constant linear ratio. It is similar to the Parallel Axis "pa" series of commands, however Linear Ratio locks the axes together by their calibrated units, whereas "pa" Parallel Axes were locked together in terms of pulses. Linear Ratio can perfectly lock together axes with different pulses-per-unit.

Linear Ratio can not be turned off accidentally through emergency stops, or disabled in any of the ways axes are normally affected by the Kuper software. Linear Ratio can only be engaged and disengaged by use of the "lr" command.

It is important that the slave and master have correctly calibrated pulses per units. For instance, if the slave is acting as a parallelogram compensator on a boom system, both slave and master axes should be calibrated for degrees.

The "lr" command has two parameters: the number of the slave axis, and the number of the master axis. To slave axis 7 to axis 51, enter...

```
lr 7 51
```

at the computer keyboard and press the Enter key.

To disengage axis 7 from its master axis, enter a master axis value of "0"...

```
lr 7 0
```

As soon as you press the Enter key with a non-zero master axis, the slave axis will simply "pick up" from wherever it is without any initial movement.

It is safe to initialize the position when it is being tracked by a slave axis. The only operation that should be avoid while a slave / master pair is in operation is changing the pulses-per-unit of either one using the AxisSetup command. Also disengage the slave and master before changing either one's pulses-per-unit.

If you are using the slave as a parallelogram compensator on a boom lift with a cylinder driven lift, slave the compensator to the Virual NS axis, which will automatically compensate for the non linear motivation of the lift. However, this scheme will not work when the virtual axes are turned off. On systems with a "linear" boom lift axis, it is probably best to slave the compensator to the real boom lift axis.

Quad encoders in Absolute mode
January 6, 2005

The "ae" command allows quadrature encoders joysticks to have the same absolute response as analog joystick axes. A given encoder position always results in a specific axis position. The "absjoy" pragma file must be present in the Kuper directory to enable absolute position response.

```
ae 3 4 5
```

Creates an absolute position response for any axes connected to encoder 3, 4, or 5. When an axis is assigned to an absolute quadrature encoder, its "POSN" button is displayed in orange instead of the normal bluegreen color.

```
ae 0
```

turns off absolute position response for all axes.

It is important that the encoders always be in the same position when the encoders are engaged the first time after the software starts. However, for subsequent engagements or disengagements of the encoder and axis, the encoder can be in any position. But it is essential that the encoder be in a consistent position for the first engagement of axis and joystick after booting the software.

PRELIMINARY PROCEDURE FOR USING QUADRATURE ENCODERS IN ABSOLUTE POSITION MODE

1. Make certain the absolute encoder function is turned off by typint "ae 0".

2. Setup the encoder-to-axis assignments.

3. Make sure the axes with absolute encoders are in PLAY, and not in REC or NEUT.

4. Place the absolute quadrature encoders in their physical home position, disregarding any numbers displayed in the VELO column.

5. Invoke the "ae" command to assign the encoders that will operate in absolute position mode. Remember the encoder numbers must be preassigned in the encoder column.

Reassign the "j1" and "j2" positions for the absolute encoder axes. Save the default setup.

From then on, you should only have to go through the five steps above each time the software is started.
If you must reassign encoders, first turn off the absolute encoder function with "ae 0", then go through the five steps above and also reassign the j1 and j2 positions.

Using Editbits to Control Camera Triggering
January 5, 2007

Due to hardware limitations, this function only exists in the Kuper2001 version of RTMC130.

Make a pragma file in the Kuper directory named "stillcam"

For example, the following "stillcam" file sets RTMCLogic connector bit 3 low between 40 and 220 degrees of camera rotation (which allows for some shutter delay)...

```
3 40 220
```

To get a short strobe near the start of the exposure on bit 5, you would have something like

```
5 45 90
```

Caveats...

Only works if at least one axis is in Blur mode.
One works if an axis is named "Camera" exists and is enabled. Can be a dummy axis with no motor.
Can be interfered with by other bit programming, don't use this bit for anything else.
The bit is normally high, goes low only between the angles specified.

More Special Serial Commands
February 22, 2007

As of release 070221k2.zip

Three new two-letter commands are implemented to allow bit control via the serial connection. The parallel printer port and Kuper RTMC LOGIC output pins are mapped in order as pins 2,3,4,5,6,7,8,9. For instance:

p101000000 (or p1 01010000) sets pins 3 and 5 of the motherboard parallel printer port high, and pins 2,4,6,7,8,9 low.

```
    p1xxxxxxxx sends to LPT1:, the motherboard parallel printer port
    p2xxxxxxxx sends to LPT2:, or any parallel port card addressed to 0x278 hex.
    p3xxxxxxxx sends to the RTMC LOGIC port on the Kuper2001 card
```

Important points:

--The 'x" fields above must contain only ASCII 1's or 0's

--There must be either zero or one spaces between the command and the bit string.

--Most of the time the p1 and p2 patterns are applied as soon as the full bit string is received. However, during the time the Kuper system's motion control interrupt is being serviced the ports can not be updated, which means there may be an update delay of up to 1 millisecond in each 8.33 millisecond time frame. Characters receipt is not affected by the Kuper interrupt, characters may be sent at any time.

--The p3 bit pattern must be applied synchronous with the motion control tick, so for p3 bit patterns a delay of up to 8.33 milliseconds may occur between the full command arriving and the bit pattern being applied.

--For the p3 command, only the first 5 bits should be used, the system may use the last three bits for blooping, work lights etc. On some systems with Kuper controlled work lights, p3 bit patterns may interact with the worklight. In that case, the appropriate bit may have to always be set high or low by the external controller software.

--If using the p1 and p2 commands, the "EditBits" box should be set to "BITS OFF" on the main Kuper screen.

--If less than eight bits are specified, unspecified bits are taken to be 0's. For instance:

```
p211
```

sets LPT2: pins 2,3 high, and pins 4,5,6,7,8,9 low. This may be useful for rapid toggling.

Sending an "s1" command is the same as pressing the shoot switch.

There is no corresponding "s1" command that can be typed from the keyboard, this command can be only sent serially. The serial link must be active.

Whether or not you send a s1 command, when shooting stop motion the Kuper system now sends the frame number and exposure number over the serial link. This information is sent just before waiting for the shoot switch. If the STOP button is pressed by the animators, or we simply reach the end of the programmed section, a "STOP" is sent. The sent data stream for a 3 frame shot with 2 exposures per frame would be:

```
fr:0 ex:0
fr:0 ex:1
fr:1 ex:0
fr:1 ex:1
fr:2 ex:0
fr:2 ex:1
STOP
```

Each line is terminated with CR, LF (0x0d, 0x0a). If practice, the controlling software would wait for "fr:x ex:x" which indicates the system is waiting for the switch. The controlling software then sends "s1" to shoot the frame. It is still possible for the animators to shoot a frame the old-fashioned way, so they must be well behaved boys and girls.
Will any of this have unexpected side effects? Only time will tell.

Kuperboy Remote Hookup
Date unknown

You must connect Kuperboy and Kuper Remote with a serial cable. Connect "RS232-2 H7" on the two boards. Connect pins 3 and 2, 2 and 3, 5 and 5. No other connections or jumpers are necessary.

On Kuperboy, there is a file called "kboyser" with no extension. Kboyser sets the communications port, the baud rate, and the number of axes. The standard kboyser is:

```
1 38400 8
```

This gives communications port 1, 38400 baud, and 8 axes. Port 1 is actually "H7". This is the only port which will work for connecting the two boards. Kuperboy numbers communication ports differently than DOS.

Kuper Remote automatically wakes up at 38400 baud. It can receive any number of axes that Kuperboy sends.

If it is necessary to reduce the baudrate for very long serial cables:

   On Kuperboy, create a file named "kboyser":

```
   C:\KUPER>copy con kboyser {press the Enter key}
      1 19200 5 {press the "F6" key}
```

Then run the RTMC130 software.

On Kuper Remote:

On "H18 RTMC LOGIC" connect pins 2 and 4. Pins 1 and 2 are labeled - the next two pins down are three and 4. The jumper which connects pins 2 and 4 is near the letters "H18", and in the same direction. Kuper Remote checks these pins only when it wakes up. If the pins are connected, it receives at 19200. If the pins are not connected, it receives at 38400.

"H14 ANALOG OUT" is the voltage output. Pins 1 to 8 are the voltages for axes 1 to 8. Pins 9 to 15 are all ground. You can change the voltage range for axes 1 to 4, but axes 5 to 8 are always -10 volts to +10 volts. I set the voltage range for axes 1 to 4 to 3 volt to 8 volts. I also set "Zoom" and "Focus" for analog axes.

You can set the range to axes 1 to 4 to any range within -10 volts and +10 volts:

--In "AxisSetup", set the Pulses Per Unit to 40.95

--In "Hardset:Setup Software Position Limits", set the "Software Position Limits" for the analog axes to 0 and 100. You may also want to set the Joystick Position Limits, but this is not necessary. This makes the axes have the minimum volage at the position 0.000, and the maximum voltage at the position 100.00, like percentages.

--In "MouseJog", use the "PatchDAC" command to select the analog axes. Click on the number (not the name) of the axis so the number turns from grey to yellow.

--Move the analog axes to position 0.000. On Kuper Remote, adjust R31 to the desired minimum voltage (for instance, 2.000 volts). Move the analog axes to position 100.00. Adjust R32 for the desired maximum voltage (for instance, 8.000 volts).

--Don't forget to use "UtilFiles:Save Default Setup".

I have already set "Zoom" and "Focus" for analog. If you want to change the voltages, just move the axes to 0.000 and adjust R31 for the minimum voltage. Move the the axes to 100.000, and adjust R32 for the maximum voltage. Leave a little bit of motion, so it is not possible to crash the lens.

All the axes always are available both on step-and-direction "H1" and analog voltages "H14". It is simply a matter of which signal you want to use.

"DC1" is the power supply input. The power should be between 9 and 18 volts, but you can use up to 30 volts if necessary. The boards will start to have problems if the voltage goes below 7.7 volts.

If you look very closely at the connector, you can see "1" and "+12" near the post -- the screw may partly cover the number. The middle pin is ground. Do not use the third pin, labelled "3". There is a big diode on the board, in case the connections are not correct. If you have a meter, there should be continuity between pin 1 (+12 volts), and the wire labelled "F1".

```
|
|   O
| DC1 GND
|O  O  O  O
|+12       3
--------------------
```

I included inputs for encoders on Kuper Remote, but there is no software yet.

Kuperboy will automatically "wake up" with certain axes enabled and in "REC". This makes it possible to use the system with a keyboard or monitor. If you want to change the setup, use the "mg" (Memorize Group) command. This causes Kuperboy to remember the combination of axes which are enabled and in "REC". These axes will automatically boot up.

I still need to send the keypad, the LCD display. I will try to send these in the next few days.

I want to do one more variation on the Kuperboy board, in about 3 weeks. This will eliminate the wires. I will send replacements for you present boards when the new boards. We can perhaps use regular mail to avoid customs problems.

Everything should work very well. The only problem might be with very long cables, in which cause you must reduce the baud rate. I think there should be no problem with 38400 baud with a 50 meter cable.

Kuper Ascii File notes
Date Unknown

This is a sample Kuper ascii file. The first line in the file names which axes are saved in the data. This selection of axes is made by the Kuper operator when he saves the ascii file. Any number of axes may be saved. The rows are frame numbers, the columns are axes.

In the example move, the Z_Track axis moves forward -100 inches. X_EW and Z_NS both move 100 inches in the + direction. Cam_Pan and Cam_Tilt move as necessary to stay pointed at an object 100 inches in front of the camera's starting position. In this example, the operator also elected to save the data on the axis names ZoomLens.

Note that the recorded data shows the "center of blur" positions. In the first frame, Z_Track moves from 0.000 to -1.000 inches. Since the camera shutter winds from closed, through open, and then on to closed again during this sequence, the actual data for the center of the resultant blurred streak is at -0.500 inches. When superimposing computer graphic images, it is usually best to attach them to the center-of-blur position.

If the operator has calibrated his system properly and activated the automatic nodal compensation features of the Kuper software, the data can represent the move data in true XYZ and rotation coordinates. In this case, the operator would choose to save the axes named VTrack, VEW, VNS, VPan, Vtilt, and VRoll. This makes the move data independent of the specific geometry of the motion control rig.

```
{the first line in the Kuper ascii file is the line just below this one}
Axes = Z_Track, X_E/W, Z_N/S, Cam_Pan, Cam_Tilt, ZoomLens
     -0.5000      0.5000      0.5000     -0.2880     -0.2880     50.1500
     -1.5000      1.5000      1.5000     -0.8725     -0.8724     50.4500
     -2.5000      2.5000      2.5000     -1.4689     -1.4684     50.7500
     -3.5000      3.5000      3.5000     -2.0772     -2.0759     51.0500
     -4.5000      4.5000      4.5000     -2.6979     -2.6949     51.3500
```

Pulse Width command
date unknown

The pwidth file sets the default pulse width to 1, 2, 3, or 4 microseconds. The file contains a series of characters in the range of 1 to 4, one character for each axis. The characters should not be separated. If no pwidth file is present, the default width is 1 microsecond.

Note that pwidth must have no extension (ie. .doc, .txt, etc.).

In the following example, the first 4 axes would have 1, 2, 3, and 4 microsecond pulse widths, all other axes would default to 1 microsecond:

```
      1234
```

For some models of Lynx and Centent CNO-143 drivers, it may be necessary to set the the default pulse widths to 4 microseconds to help penetrate noise filters on the inputs. The appropriate pwidth file would be:

```
      4444444444444444
```

which would set the first sixteen axes to 4 microsecond pulse widths.