



RPT Head made ready to pack:

- 1 Slip out Buchla transmitter from the holder and turn off the power (press top button on the transmitter body). Unplug the miniplug power cable of the transmitter and coil the wire under the Velcro strip on the rear of the Head. Unscrew two screws of the transmitter holder and screw the screw back to the Head belt.
- 2 Take the camera out by unscrewing the hex screw holding the Camera on the top. Disconnect two BNC cables, secure them with a rubber band.
- 3 Unscrew the Rotation sensor on the front top of the head held by two screws and remove the amp Board next. The Board is held by a single screw. It has a special spacer lifting and holding it from the Head. Disconnect the flat ribbon wire and pack the electronic inside the empty head.
- 4 Release two set hex screws 90 degrees apart at the Tilt Stepper Worm Wheel. This will let the head swing freely on its "Y" axes.
- 5 Release two upper and two lower ring-belt small hex screws and let the rear of the Head slide down towards the mirror at the center of the Head post. Leave a 1/4 inch gap between the Rotation Ring Assembly and the mirror and tighten all four screws. Put a piece of foam between the Ring and mirror since the mirror lays normally free.
- 6 Remove one of the bottom plate hex screws to make place for a screw holding the wood piece which presses and holds the Head assembly to the bottom of the frame.
- 7 Slip the whole apparatus into the metal cage, secure all armature holding it and secure the bottom wooden plate.

Put no other loose stuff in the some box to protect all other Index hardware. Other units regularly sharing the luggage are the Light Dimmer boxes.

That Jeff guy
jeffsc@chips.com
Doschier
3928 Shafter Ave
Oakland Ca 94609
510-653-5825

5/9/95

TO : vasulka@ffa.vutbr.cz
woody @ santafe.edu

Hi there Woody...

Heard from Steina and 'ya want info on that Personal Animation Recorder by DPS. Ed Tannebaum has one and says it works. It is a IBM/PC based interface card (ISA BUS - 16 bit), connected to a dedicated "IDE Hard Disk". It uses hardware based JPEG compression/decompression, to squeeze the images onto the disk. It has no audio provision so it works best for animation and visual sequences.

It has conversion utilities to and from Targa files, and adjustments for compression "quality factor", to trade off size of file versus quality. JPEG compression ratios of 10:1 are are close toll:50AM 5/9/95 the original, ratios of ~6:1 are often called "betacam quality", and 20:1 shows blockiness and contouring.

If you are using it with an SGI box as a "baby Abekas disk", there are issues of how to get the images from the SGI to the PC and back. A generic 486-66 PC/Clone with ISA slots, with 8 MB main memory, 700 MB disk, VGA graphics accelerator and a 17" Super VGA color monitor (0.28mm Dot pitch/ 102*768 Non-interlaced : ex NEC XV17) will do fine for the card. PC/clone price ~ \$1750 (US). Personal Animation Recorder is around \$2K plus \$1K for the TBC IV option, to record composite/S-Video (not component) from live video. Without the TBC option, it has video output only and data comes form the PC. The IDE disk >1.7 Gigabyte (Micropolis 2217A, Conner, Seagate) will cost around \$1K to 1.5K and up. This IDE disk is used exclusively by the PAR, and is independent from the PC/Clone hard disk. Below is a guess at NTSC/US pricing. Computer/Disk prices change daily :

PC 486/66 clone Computer 256K cache,8 MB memory,
750 MB disk, keyboard/floppy/mouse/case,
SVGA graphics accelelartor \$ 1000
Computer-Monitor 17" \$ 750
DPS-PAR (main board) \$ 2000
DPS-PAR : TBCIV (TBC option) \$ 1000
PAR-DISK IDE (2 Gig) \$ 1500
Total -----> \$ 6250

Ethernet Networking Interface and Network software would be extra.
Contact DPS for info for use on Networks. They had bugs
in getting it to work on a network, maybe they fixed it.

Ed Tannenbaum 510-787-1567
Crocket, Ca

USA -
DPS (Digital Processing Systems)
11 Spiral Drive
Suite 10
Florence, Kentucky 41042
Tel : 606-371-5533
Fax : 606-371-3729

Canada/International
DPS
55 Nugget ave
Unit 10
Scarboroug, Ontario Canada
M1S3L1
Tel : 416-754-8090
Fax : 416-754-7046

Europe
DPS Limited
Riverside Business Park
Unit #2
DogFlud Way, Farnham
Surrey U.K. GU9 7SS
tel : (0252)-718300
Fax : (0252) - 718400

'This is a modified DENX.mot for use in Brno starting Sep.
'30, 1993

'introduction: rpt and lights on voice (to bypass voice,
'type "C01 HOME" in console mode

set loop 1
set compass 1

listen home

```
midi 89 24 01      'turn off light 1
midi 89 25 01      'turn off light 2
midi 89 26 01      'turn off light 3
midi 89 27 01      'turn off light 4
midi 89 28 01      'turn off light 5
midi 99 29 5f      'turn on light 6
      switcher 55
video 1 vd pl 12854
say 18 home        'RPT goes Home (North???)
rpt h clear        '"clear" is a label
```

end

rpt_label clear

```
midi 99 24 5f      'turn on light 1
midi 89 25 01      'turn off light 2
midi 89 26 01      'turn off light 3
midi 89 27 01      'turn off light 4
midi 89 28 01      'turn off light 5
midi 89 29 01      'turn off light 6
switcher 55        'Buchla high
rpt h chain2
```

end

rpt_label chain1

```
midi 89 27 01      'turn off light 4
midi 99 24 5f      'turn on light 1
switcher 55
rpt 1 0 0 0 1000 sss chain2
video 19369 se pl
if compass=1
  say 18 north
if compass=2
  say 18 saskatchwann
set loop 2
```

end

rpt_label chain2

```
midi 89 24 01      'turn off light 1
midi 99 25 5f      'turn on light 2
rpt 1 90 0 180 1000 sss chain3
video 20388 se pl
if compass=1
```

```

        say 18 west
    if compass=2
        say 18 ahrrizowna
end

```

```

rpt_label chain3
    midi 89 25 01 'turn off light 2
    midi 99 28 5f 'turn on light 5
    rpt l 45 -90 180 1000 sss chain4
    video 19824 se pl
    if compass=1
        say 18 sky
    if compass=2
        say 18 moon
end

```

```

rpt_label chain4
    midi 89 28 01 'turn off light 5
    midi 99 26 5f 'turn on light 3
    rpt l 180 0 0 1000 sss chain5
    video 21628 se pl
    if compass=1
        say 18 south
    if compass=2
        say 18 ukahtann
end

```

```

rpt_label chain5
    midi 89 26 01 'turn off light 3
    midi 99 27 5f 'turn on the light 4
    if compass=1
        say 18 east
        set compass 3
    if compass=2
        say 18 oklahoohma
        set compass 1
    if compass=3
        set compass 2
    if loop=1
        rpt l 270 0 0 1000 sss chain1
    if loop=2
        rpt l 270 0 0 1000 sss start3
end

```

'----- DENISE2.MOT

```

rpt_label start3
    rpt h move3a
    set compass 1
    set loop 1
    video 1 vd 13986 se 60 sp
end

```

```

rpt_label move3a

```

midi 89 27 01 'turn off light 4
video 13986 se 60 sp
video 14708 mf
rpt l 180 0 0 2000 +++ move3b 24

end

rpt_label move3b
video 60 sp 15069 mf
rpt l 270 0 0 2000 +++ move3c 12

end

rpt_label move3c
video 16104 se
rpt i p 1 111 111 + 5 move3d

end

rpt_label move3d
video 15794 mr
rpt l 270 270 0 2000 +-+ move3e 10.32

end

rpt_label move3e
midi 99 29 5f
rpt i p 1 111 111 + 6 move3f

end

rpt_label move3f
video 16104 mf
rpt l 270 0 0 2000 +++ move3g 10.32

end

rpt_label move3g
video 15069 se
rpt i p 1 111 111 + 3 move3h

end

rpt_label move3h
video 15432 mf
switcher d5
rpt l 0 0 0 2000 +++ move3i 12.1

end

rpt_label move3i
video 30 sp 15252 mr
rpt l 315 0 0 2000 -++ move3j 12.1

end

rpt_label move3j
video 60 sp 15432 mf
switcher 55
rpt l 0 0 0 2000 +++ move3k 6.05

end

rpt_label move3k
video 16463 se 30 sp

video 17184 mf
rpt 1 0 180 0 2000 +++ move31 24

end

rpt_label move31
say 18 all done
switcher d5
video 1 vd 13986 se 60 sp
set compass 1
set loop 1

rpt i p 1 100 100 - 5 chain1
end

rpt_label start4
video 1 vd 13986 se 60 sp
switcher 55
set loop 1
set compass 1
rpt h chain1

end

' --- lightning data follow

lightning one c4
video 17399 se
end

lightning one c+4
video 17400 se
end

lightning one d4
video 17401 se
end

lightning one d+4
video 17402 se
end

lightning one e4
video 17403 se
end

lightning one f4
video 17404 se
end

lightning one f+4
video 17405 se
end

lightning one g4
video 17406 se
end

lightning one g+4
video 17407 se
end

lightning one a4
video 17408 se
end

lightning one a+4
video 17409 se
end

lightning one b4
video 17410 se
end

lightning one c5
video 17411 se
end

lightning one c+5
video 17412 se
end

lightning one d5
video 17413 se
end

lightning one d+5
video 17414 se
end

lightning one e5
video 17415 se
end

lightning one f5
video 17416 se
end

lightning one f+5
video 17417 se
end

lightning one g5
video 17418 se
end

lightning one g+5
video 17419 se
end

lightning one a5
video 17420 se

end

lightning one a+5
video 17421 se

end

'----- this is the end of the lightning data

Log file friendly.txt
Input Log file
for regular key 'heartbeat' input test
START_TIME Fri Apr 22 18:59:52 1994

START_ENTRY
DELTA_TIME 0.000
CHANNEL 0
\90\4e\50
END_ENTRY

START_ENTRY
DELTA_TIME 0.000
CHANNEL 0
\90\52\50
END_ENTRY

START_ENTRY
DELTA_TIME 0.000
CHANNEL 0
\90\60\10
END_ENTRY

START_ENTRY
DELTA_TIME 59.000
CHANNEL 0
\90\61\10
END_ENTRY

START_ENTRY
DELTA_TIME 105.00
CHANNEL 0
\90\62\10
END_ENTRY

START_ENTRY
DELTA_TIME 181.00
CHANNEL 0
\90\63\10
END_ENTRY

START_ENTRY
DELTA_TIME 261.00
CHANNEL 0
\90\64\10
END_ENTRY

START_ENTRY
DELTA_TIME 338.00
CHANNEL 0
\90\48\10
END_ENTRY

END_ENTRY
START_ENTRY
DELTA_TIME 338.00
CHANNEL 0
\90\70\10
END_ENTRY

START_ENTRY
DELTA_TIME 348.00
CHANNEL 0
\90\49\10
END_ENTRY

START_ENTRY
DELTA_TIME 350.00
CHANNEL 0
\90\6e\6f
END_ENTRY

START_ENTRY
DELTA_TIME 350.00
CHANNEL 0
\90\44\10
END_ENTRY

START_ENTRY
DELTA_TIME 358.00
CHANNEL 0
\90\49\10
END_ENTRY

START_ENTRY
DELTA_TIME 365.00
CHANNEL 0
\90\3b\10
END_ENTRY

START_ENTRY
DELTA_TIME 368.00
CHANNEL 0
\90\49\10
END_ENTRY

START_ENTRY
DELTA_TIME 378.00
CHANNEL 0
\90\49\10
END_ENTRY

START_ENTRY
DELTA_TIME 380.00
CHANNEL 0
\90\40\10
END_ENTRY

START_ENTRY
DELTA_TIME 388.00
CHANNEL 0
\90\49\10
END_ENTRY

START_ENTRY
DELTA_TIME 395.00
CHANNEL 0
\90\39\10
END_ENTRY

START_ENTRY
DELTA_TIME 398.00
CHANNEL 0

\90\49\10	DELTA_TIME 625.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 822.00
START_ENTRY	\90\68\35	CHANNEL 0
DELTA_TIME 408.00	END_ENTRY	\90\49\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\49\10	DELTA_TIME 674.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 832.00
START_ENTRY	\90\69\35	CHANNEL 0
DELTA_TIME 410.00	END_ENTRY	\90\49\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\42\10	DELTA_TIME 691.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 835.00
START_ENTRY	\90\6a\35	CHANNEL 0
DELTA_TIME 418.00	END_ENTRY	\90\3a\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\49\10	DELTA_TIME 725.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 842.00
START_ENTRY	\90\6b\35	CHANNEL 0
DELTA_TIME 428.00	END_ENTRY	\90\49\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\49\10	DELTA_TIME 744.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 845.00
START_ENTRY	\90\6c\35	CHANNEL 0
DELTA_TIME 438.00	END_ENTRY	\90\41\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\49\10	DELTA_TIME 792.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 852.00
START_ENTRY	\90\48\10	CHANNEL 0
DELTA_TIME 448.00	END_ENTRY	\90\49\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\49\10	DELTA_TIME 792.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 862.00
START_ENTRY	\90\70\10	CHANNEL 0
DELTA_TIME 458.00	END_ENTRY	\90\49\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\49\10	DELTA_TIME 802.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 865.00
START_ENTRY	\90\49\10	CHANNEL 0
DELTA_TIME 468.00	END_ENTRY	\90\33\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\49\7f	DELTA_TIME 805.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 872.00
START_ENTRY	\90\6f\10	CHANNEL 0
DELTA_TIME 475.00	END_ENTRY	\90\49\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\66\10	DELTA_TIME 812.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 882.00
START_ENTRY	\90\49\10	CHANNEL 0
DELTA_TIME 564.00	END_ENTRY	\90\49\10
CHANNEL 0	START_ENTRY	END_ENTRY
\90\67\10	DELTA_TIME 820.00	START_ENTRY
END_ENTRY	CHANNEL 0	DELTA_TIME 892.00
START_ENTRY	\90\3e\10	CHANNEL 0
DELTA_TIME 625.00	END_ENTRY	

\90\49\10
END_ENTRY

START_ENTRY
DELTA_TIME 902.00
CHANNEL 0
\90\49\10
END_ENTRY

START_ENTRY
DELTA_TIME 912.00
CHANNEL 0
\90\49\10
END_ENTRY

START_ENTRY
DELTA_TIME 922.00
CHANNEL 0
\90\49\7f
END_ENTRY

START_ENTRY
DELTA_TIME 927.00
CHANNEL 0
\90\49\35
END_ENTRY

rewind

```

' Rotterdam sleep mode set tsleep 450 slide down 1
end end

set tsleep 1
set video_mode 1 'c5
set cylinder 1 midiin one x48 'c+5
set light 1 if tsleep=1 midiin one x49 vel 30 7e
set slide 1 midi bf 13 1 slide up 1
end end

'g+0 if tsleep=1 air 1 on 'f+5
midiin one x15 air 2 on midiin one x4e vel 1 2f
if tsleep=1 air 3 on video 0 ad 0 vd
air 5 on end end

'a0 if tsleep=1 air 4 on 'f+5
midiin one x15 air 5 on midiin one x4e vel 30 5f
if tsleep=1 air 5 off dimmer F1\rD1@100\rG midi bf 33 40
end end dimmer F5\rD3@0\rG midi bf 10 64
midi bf 11 64
midi bf 12 64
midi bf 13 7f
video 64 rb
end end

'c2 INITIATE SLEEP 'd5
midiin one x24 midiin one x4a
if tsleep=0 if tsleep=1
dimmer F5\rD1-3@0\rG air 1 on
if tsleep=0 air 2 on
dimmer F5\rD4@75\rG air 3 on
if tsleep=0 air 4 on
video 1se air 5 on
if tsleep=0 air 1 on
if tsleep=0 air 2 on
if tsleep=0 air 3 on
if tsleep=0 air 4 on
if tsleep=0 air 5 on
midi bf 13 7f end
end

'c+ AWAKEN 'c+5
midiin one x25 midiin one x49 vel 7
if tsleep=0 air 1 off
dimmer F5\rD2@40\rG if tsleep=1 air 2 off
if tsleep=0 air 3 off
dimmer F5\rD3@65\rG if tsleep=1 air 5 off
if tsleep=0 air 5 off
dimmer F5\rD4@0\rG if tsleep=1 dimmer F5\rD1@0\rG
end end
video 5sepl
if tsleep=0 'c+5
air 1 off midiin one x49 vel 3
if tsleep=0 if tsleep=1
air 2 off air 4 off
if tsleep=0 air 3 off end
if tsleep=0 air 4 off
air 4 off 'c+5
if tsleep=0 midiin one x49 vel 0
air 5 off if tsleep=1
end end

```

```
midiiin one x52 vel 25 39
if video_mode=1
video 30sp mf
end
```

```
'a+5
midiiin one x52 vel 3a 5f
if video_mode=1
video pl
end
```

```
'a+5
midiiin one x52 vel 60 6f
if video_mode=1
video 120sp mf
end
```

```
'a+5
midiiin one x52 vel 70 7f
if video_mode=1
video 240sp mf
end
```

```
'b5
midiiin one x53
if video_mode=1
video mr
end
```

```
'c+6
midiiin one x55
if tsleep=1
air 1 on
if tsleep=1
air 2 on
if tsleep=1
air 3 on
if tsleep=1
air 4 on
if tsleep=1
air 5 on
if tsleep=1
dimmer F5\rD1@100\rG
end
```

```
'f6 117 sec
midiiin one x59
if tsleep=1
dimmer F6\rD1@0\rG
if tsleep=1
midi 90 31 10
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 12608se16141pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
```

```
air 4 off
if tsleep=1
air 5 off
end
```

```
'g6
midiiin one x5b
if tsleep=1
air 1 on
if tsleep=1
air 2 on
if tsleep=1
air 3 on
if tsleep=1
air 4 on
if tsleep=1
air 5 on
if tsleep=1
video pa
if tsleep=1
dimmer R
if tsleep=1
dimmer D1@100\rG
if tsleep=1
slide down 30
end
```

```
'b6 496 sec
midiiin one x5f
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 29082se43969pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'c7 59 sec
midiiin one x60
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 5se1801pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
```

```
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'c+7 46 sec
midiiin one x61
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 1802se3200pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'd7 76 sec
midiiin one x62
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 3261se5551pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'd+7 80 sec
midiiin one x63
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 5552se7973pl
if tsleep=1
air 1 off
```

```
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'e7 77 sec
midiin one x64
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 7974se10303pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'f7 41 sec
midiin one x65
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 11380se12607pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'f+7 95 sec
midiin one x66
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 16142se19010pl
```

```
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'g7 161 sec
midiin one x67
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 19011se23851pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'g+7 48 sec
midiin one x68
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 23852se25287pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'a7 16 sec
midiin one x69
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
```

```
video 25288se25775pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'a+7 33 sec
midiin one x6a
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 25776se26775pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'b7 28 sec
midiin one x6b
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 26776se27638pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'c8 47 sec
midiin one x6c
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
```

```
video 27639se29043pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'c+8 207 sec
midiin one x6d
if tsleep=1
dimmer F5\rD1@0\rG
if tsleep=1
dimmer F5\rD2@40\rG
if tsleep=1
dimmer F5\rD3@65\rG
if tsleep=1
video 44271se50477pl
if tsleep=1
air 1 off
if tsleep=1
air 2 off
if tsleep=1
air 3 off
if tsleep=1
air 4 off
if tsleep=1
air 5 off
end
```

```
'd8
midiin one x6e
if tsleep=1
video 29082sepl
end
```

```
'd+8
midiin one x6f
if tsleep=1
video 34000sepl
end
```

```
'e8
midiin one x70
set video_mode 0
video 1se
end
```

```
'd3-d+3 496 sec
midiin two x32-x33
if video_mode=1
video 29082se43969pl
end
```

```
'e3-f3
midiin two x34-x35
if video_mode=1
video 7974sepl
end
```

```
'f+3-g3
midiin two x36-x37
if video_mode=1
video 3261sepl
end
```

```
'g+3-a3
midiin two x38-x39
if video_mode=1
video 5sepl
end
```

```
'g+2-a2
midiin two x2c-x2d
if video_mode=1
video 16142sepl
end
```

```
'a+2-b2
midiin two x2e-x2f
if video_mode=1
video 26776sepl
end
```

```
'c3-c+3
midiin two x30-x31
if video_mode=1
video 23852sepl
end
```

```
'midi bf 10 1 'channel 1 volume 1
'midi bf 10 7f 'channel 1 volume 127
'midi bf 11 1 'channel 2 volume 1
'midi bf 30 1-7f 'smoothing fast-slow
'midi bf 35 1-7f 'smoothing enabled
```

MINICOM Release Notes

Version 2.12

10/1/94

Russ Gritzso,
Jericho Data Systems

Version 2.12, 10/1/94

Version 2.12 is an enhancement release of MINICOM that changes the way the system device labels work.

A feature was added to allow a system device (sysdev) label to be set such that it persists for some period of time, and then expires on its own. This allows the code to set up a condition that will last for some period of time and then go away automatically. This condition can be continuously overwritten or updated, so it has the effect of being a 'watchdog timer'. This modification was done in such a way as to not require a mod in the DAD code in order to support the use of this feature. As a result this feature has the following behavior:

Labels that can assume this timing property must start with the letter t (lower case). This is an indicator to the code to treat these labels special. For labels that start with t, the value is treated differently. If the value is initialized to a 0 or a 1, the code will treat the label as a regular label, testing for the conditions 0 or 1. If, however, the value is initialized to a 15, the label will assume the value of 1 for 15 seconds, and then default to 0. This initialization works the same from either the SYSDEV.CFG file, or from sending a message directly to the system device.

For example, assume the MIDI.CFG file has the following key entries:

```
MIDI_KEY 41
LOW 0x29
HI 0x2f
NOTE_ON
MIDI_CHANNEL 00
CONDITION test
VALUE 1
DEVICE 0
TEST_SET
END_KEY
```

```
MIDI_KEY 42
LOW 0x29
HI 0x2f
NOTE_ON
MIDI_CHANNEL 00
CONDITION test
VALUE 0
DEVICE 0
TEST_CLEAR
END_KEY
```

And assume that the SYSDEV.CFG file contains the following label declaration:

```
SYSTEM_LABEL 3
LABEL test
VALUE 60
END_LABEL
```

When the code first starts up, anytime the midi channel gets a NOTE_ON of 0x2a the console will display the string TEST_SET. Once 60 seconds have gone by, the label value will time out and default to 0. From then on the console will display TEST_CLEAR each time the midi note is input. The label can be reset to time out again by sending the sysdev a 'test 90' message (set test to 1 for 90 seconds) or can be set on permanently by sending a 'test 1' message. Remember that this can be tested by typing 'd15 test 90' at the keyboard to send a message to the system device directly from the keyboard. If a new time value is sent to the label prior to the timeout, the timeout duration will be restarted.

The timed labels can only assume the values of 0 or 1, but the code can have multiple timed labels (up to the 32 label maximum). The timing value is in seconds, up to a maximum of 32000 (about 8 hours).

The behavior of the other labels (ones that don't start with a t in the first letter of the name) stays the same, with one minor difference. In previous versions of the code having a condition value of 0 (say in a midi key entry) caused that key entry to execute unconditionally. This forced the user to avoid using 0, and use non zero numbers for most conditions. This has changed now, and 0 is a legitimate state. If the user specifies a key entry with a condition of 0, that key will only execute if the sysdev label value is 0. The only way now to make a key entry unconditional is to not specify any label in the key entry. This makes the key execute unconditionally. If the user has a condition label on every key entry, the label values will have to be setup explicitly in the sysdev initialization in order to work. Unless the user has set all midi keys to have conditions with non-zero length label names, this release of the code should be compatible with earlier .CFG files.

/*****
Version 2.11, 4/22/94

Version 2.11 is an enhancement release of MINICOM that changes the way that file input works and increases the number of midi key values supported.

MINICOM now supports 256 MIDI key values, numbered 0-255.

The file input has been enhanced. The input file command string now supports the string conversion referred to in the INTERCOM manual. This allows the user to, say, send a midi string to the midi channel by specifying ' \90 \3c \40 ' rather than having to figure out what obscure ascii characters these are and how to input them in an editor or having to generate them by a file output first.

The file input now also supports the keyword 'rewind'. If this keyword appears at the end of the input file specified in file.cfg, then the software will rewind back to the beginning of the file and reset the start time value. Thus, the following input file simulates a midi NOTE IN, once each 10 seconds. The first note, at 10 seconds, is a note value of 0x30, and the second, at 20 seconds, is a 0x3c. This repeats until the code is stopped.

FILE INPUT SAMPLE FILE:

```
Log file infile.txt
Input Log file

for regular key 'heartbeat' input test

START_TIME Fri Apr 22 18:59:52 1994
```

```

START_ENTRY
  DELTA TIME 10.000000
  CHANNEL 0
É0@
END_ENTRY

```

```

START_ENTRY
  DELTA TIME 20.000000
  CHANNEL 0
\90\3c\41
END_ENTRY

```

rewind

:FILE INPUT SAMPLE FILE

In order for this input to work, the user must have entered the file name of the input file in FILE.CFG, under the INPUT_FILE entry. Also, the user should know what channels are used on the input stream of the devices, as it is these channels that the code inputs the commands to from the file device.

If the file output is used, the file device must be specified as a destination in the channels for which you want to record an input.

While this feature is useful, it does have some limitations. Currently there is no provision to stop the looping, short of stopping the code. If the user wanted to cease the execution of these inputs, he would have to use system labels and set a mode such that the inputs of these values are ignored. Also, there is no way to dynamically alter the timing using this method. The commands will be input at fixed times, no matter what.

Of course, the real-time input is not suppressed during this time, and the code should work normally. Examples of how this feature could be used include: regular timing, full-blown scripting of a performance, and a periodic reset to a known state.

To elaborate on the regular timing idea, if the input file shown above were used, a midi note would come in every 10 seconds, no matter what else is going on. The other conditions, like label values, etc. could dictate how you used this note in, but at least it would come in regularly, mixed with any real-time input.

If you fully script a performance, and add the rewind keyword, then a performance could take on an underlying tone or set of actions driven by the file input, with the ability of the user to make real-time inputs, and at the end of the script reset itself and repeat.

If all you wanted was a reset, you could have a set of commands that set the machine to a known state once each 30 minutes or so, and do that repeatedly.

In a future version of MINICOM a device called 'timer' will be added to allow the posting of a command to execute at some specific time.

/*****
Version 2.1, 4/15/94

Version 2.1 is an enhancement release of MINICOM to support two new devices and two new ports. Notes on these two devices appear below.

4

Access to a lighting unit is provided. It is called Device 9, and is connected to port 9. It uses COM 6 of the COM x/i board, which is labeled number 2 on the octopus cable set. Currently it is output only, but it will accept input in the form of a reply from the light unit. Any string returning from the lighting unit will be displayed on the screen if the screen diags (in the channel.cfg file) is set to a number greater than one. This allows the user to see what the replies the light unit is sending back, a useful feature for command debugging. MINICOM is not able to take a specific action on the light input, and so it does not require a CFG file of it's own.

The light unit output behaves in exactly the same manner as the disk output, with a carriage return (0x0d) appended to the end of all outputs automatically. Only ASCII strings are allowed to be put out.

Sound unit.

The sound unit is very similar to the lighting unit. The sound unit is called Device 10, and is connected to port 10. It uses COM 7 of the COM x/i board, which is labeled number 3 on the octopus cable set. Currently it is output only, but it will accept input in the form of a reply from the sound unit. Any string returning from the sound unit will be displayed on the screen if the screen diags (in the channel.cfg file) is set to a number greater than one. This allows the user to see what the replies the sound unit is sending back, a useful feature for command debugging. MINICOM is not able to take a specific action on the sound input, and so it does not require a CFG file of it's own.

The sound unit output behaves in exactly the same manner as the disk output, with a carriage return (0x0d) appended to the end of all outputs automatically. Only ASCII strings are allowed to be put out.

Altering the CHANNEL.CFG file:

CHANNEL.CFG must be modified to support the new devices. The first thing to do is to add the following line to any channels that must send anything out to the new devices:

```
DEST 9 (For the light unit)
DEST 10 (For the sound unit)
```

This line should appear after the PARAM lines and must appear before the END CHANNEL line. Most likely it will be added to the channel for the midi device and the console device as a minimum. Without this line the channels will not be able to talk to the devices.

Channel descriptions for the two devices must be added to accept the input reply information, and to configure the baud rate, etc. Example channel descriptions for the light unit and the sound unit are shown below:

```
CHANNEL_NUMBER 7
NAME Light Unit
DEVICE 9 light unit
PORT 9 Com6, comx/i board
IO 2
TERM end
INT 0x0
ADDR 0x0
PARAM 0 9600 Baud
PARAM 1 8 Bits
PARAM 2 0 Parity
PARAM 3 1 Stop
DEST 0
END_CHANNEL
```

```
CHANNEL_NUMBER 8
NAME Sound Unit
DEVICE 10 sound unit
PORT 10 Com7, comx/i board
```

5

```

IO      2
TERM    end
INT     0x0
ADDR    0x0
        PARAM 0 9600 Baud
        PARAM 1 8 Bits
        PARAM 2 0 Parity
        PARAM 3 1 Stop
DEST    0

```

END_CHANNEL

The channel number are somewhat arbitrary, but they must not be in conflict with any existing channel numbers.

```

/*****
Version 2.0, 4/8/94

```

Version 2.0 is a major rewrite of MINICOM to both bring it up to date with new changes in INTERCOM (version 1.71) and to support the Brotherhood Table III project. Notes on this version appear in several categories, as listed below.

Configuration.

This version of MINICOM supports the following configuration:

Computer = Toshiba T1200XE with expansion chassis, Add-in board = COM/Xi board with DOS/BIOS driver, COM1 = MIDI via the Portman/S midi interface, COM5 = the laser disk player via the COM/Xi board, and interface to the air and slide system using the Heckel parallel port interface board.

Note: The laptop must have the following statement in CONFIG.SYS:

device = xidos5.sys, assuming that the file is in the root path. Prior to useage the first time, the program XIDOSCFG.EXE (in the DIGIBOARD directory) must be run to set the configuration. Set the following parameters to start with:

```

Board#      1
Type        COM/Xi
Window      32K
Memory      D0000h
I/O Port    0300
IRQ         12
# Channels   8
Start Channel 4
Driver      INT 14h

```

Under channel parameters, set all channels to 9600 Baud, Mode of 8,N,1 and TX and RX flow of NONE. When changing any of these, a new version of XIDOS5.SYS will be written in the DIGIBOARD directory, and must be copied to the root to be picked up by the CONFIG.SYS. The COM/Xi board will now be supported by the BIOS INT 14 calls.

New Devices:

MINICOM version 2.0 adds two new devices, number 7 is the air cylinders, and number 8 is the slide projector.

F A X

THE VASULKAS

F A X

TO: _____
 To: Baley
 DATE: _____
 From: Woody Vasulka
 RE: Shipment to Czech Republic

CRATES	height	width	length	weight(Kg)
2 - 2200s	69 cm	61 cm	64 cm	43
4 - 2200s	69 cm	75 cm	117 cm	70
2 - 8000s	64 cm	62 cm	64 cm	56
5 - 8000s	69 cm	66 cm	122 cm	115
5 - 8000s	69 cm	66 cm	122 cm	115
Lyon Wood	74 cm	52 cm	169 cm	116
Metal	61 cm	83 cm	12 cm	76
Wood (Mirror)	54 cm	15 cm	86 cm	16
Wood (Mirror)	54 cm	15 cm	86 cm	20
Plastic Tube	11.5 cm diameter x 191 cm long			10
Plastic Tube	11.5 cm diameter x 191 cm long			10
TOTAL				647

(We may add to the existing crates additional material of about 10Kg)

Contents:

#2200s and 8000s refers to two different models of Pioneer Laserdisk players, each estimated for a replacement value of \$300 for a total of

\$5,400

"Lyon wood" and "Metal" contains an art exhibit titled "Machine vision" with replacement value for both crates.

\$22,000

Two "Mirror" boxes contain split beam mirror assemblies @ \$400 each

\$800

Plastic Tubes @ \$300 each (screens & frames)

\$600

Disk Synchronizer, replacement value:

\$900

Media, (7 Laser disks) replacement value

\$2,100

TOTAL value estimated at US \$31,800

Destination:**Prague Airport, then:****Dum Umeni mesta Brna (Art Galery of city of Brno)****616 00 Brno,****Czech republic, (second size city in the republic, about 200 miles SE of Prague)****Sponsor:****Silicon Graphios s.r.o.****Michal Klimes****Czech Technology Park****Technicka 15****616 00 Brno****Czech Republic****vmail: 58420****michalk@brno.sgi.com****Central Phone: +42-5-4119 1919****Dir. Phone: +42-5-4119 1931****Fax: +42-5-4119 1915****Date for arrival:****October 5 1996**

Triennale Ruhr Project
The Vasulkas/Dunn preliminary budget:

Fees:

3 - Artistic fees
2 - Technical personnel fees

SubTotal \$60,000

Travel:

All personnel \$8,000

Lab and Production
Period: Fall 1995
Length: 3 Months

Option 1 (Santa Fe) \$30,000

Option 2 (European Institution) ?

Option 3 (Oberhausen, on site) ?

Additional expenses:

Administrative

**Media - Tapes, data disks, Laserdisks,
data preparation, data compression,
CD/Rom and audio CD formatting etc.**

Custom engineering (hardware parts)

\$10,000

TOTAL = \$108,000