ECAT Interlock Circuitry

FSD 27 July 2015

This document contains the schematics and code that describe/implement the function of the ECAT/LTS Interlock circuitry.

On the schematics, yellow highlighting traces the interlock circuit itself, from its 5 V source through to the transistor that drives the relay to connect +24IL; green highlighting is used to show the interlock fault detection circuitry (the data lines that are read to determine whether there's a fault); and purple (pink?) highlighting marks the indicator LEDs on the 5-1/4" backplane and on the module interface boards that signal faults (ON is normal, OFF is fault).

The function ilock_stat() on page 20 is the function used to read all the detectors. It is called in the periodic interrupt routine (which is executed several times per second). The variable 'interlockData' is actually the address of the fault detection bus transceiver on the Peak Detector board, shown on page 4.

If the value in 'interlockData' indicates a "System" interlock fault (meaning a problem in the module bays, for example an open door), then function istat() is used to read the fault detection bus transceiver on each installed module's Interface board, shown on page 9. Modules are determined to be "installed" using function exist(), which writes to and then reads from a latch on the specified module's Interface board; see the handwritten note with orange highlighting on page 9. Module polling in istat() stops as soon as a fault (valid or otherwise) is found in any module.

Function ilock_stat() returns the value of 'interlockData' for all faults other than "System" (module) interlock faults. For a "System" interlock fault, the module interlock data is combined with the original 'interlockData' from the system level as follows: the least significant byte is the original 'interlockData' masked to 0x7C; the next byte is the module interlock data masked to 0x1F; and the most significant byte is the module number where the fault occurred.

Function ReportInterlockError() is used to display a fault on the front panel display. Because only one fault, the one closest to the start of the interlock circuit, can be resolved at one time, this function tests the detector bits in order starting from the beginning of the circuit, reporting the first fault found; because of the way this circuit works, all subsequent fault detection points will also show faults, whether or not a valid fault exists. The bulk of the function performs "System" (module) interlock fault resolution, because each different module type and some individual modules differ in the way that their fault circuits are implemented. For example, in the ECAT, most Surge modules use IL_ILOCK4 for the left bay interlock, or as the only interlock for single-bay modules; however the E505B uses IL_ILOCK2 for this purpose, and E521 and E522 modules don't distinguish specific faults at all but simply report that one of the side covers on the main cabinet is open.

Under normal operating conditions, the voltage along the entire interlock circuit must exceed the TTL high threshold so that no fault is indicated. The only loads along the interlock circuit are the indicator LEDs and associated circuitry. A fault is indicated by an absence of voltage past one of the detection points. Because all subsequent detection points will also read a low logic level, in the case of multiple faults only the fault closest to the start of the interlock circuit can be resolved.

The interlock circuit proceeds through system circuit boards as follows: from its controller backplane +5 V source, it proceeds down the backplane, passing through every module via their interface boards and whatever interlock circuitry they may have (modules without interface boards have interlock circuitry as well); then back up to the controller backplane and through the microprocessor board (acting as a simple board presence indicator); then back to the backplane and through the peak detector board, where it passes through two relays, one of which is used to actively connect the rest of this interlock circuit during system initialization, which in turn powers up +24IL (the other relay is a vestige of the never-implemented FiberLoop and can be bypassed); then back to the backplane and through its external interlock connector, and finally to the transistor that activates +24IL (the +24 V power supply that is automatically, via this circuit, disconnected upon the occurrence of any interlock fault).

Interlock Circuit

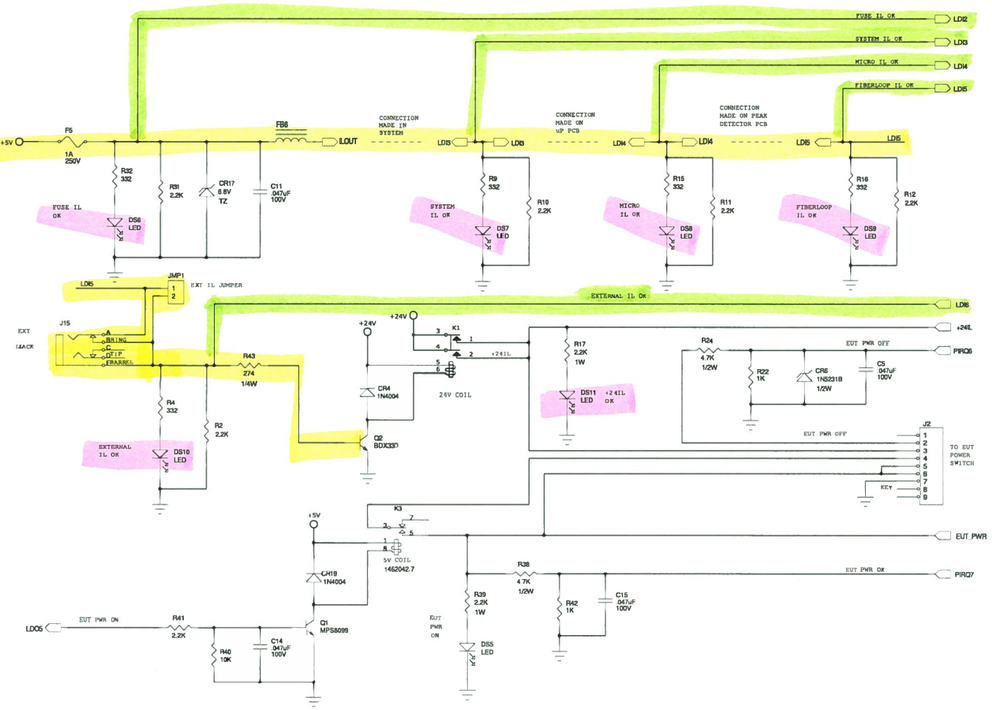
Board	Interlock	Page	Description
E103 5-1/4" Backplane	Fuse	3,5	Interlock circuit +5 V source
8-3/4" Backplane	none	6,7	Pass-through to modules
Module Interface board	System	8,9,10	Source of the 4 module interlocks
through every module interface board via 8-3/4" backplane boards			
E103 5-1/4" Backplane	none	11	Pass-through
Microprocessor board	Micro Board	12	Simple board presence jumper
E103 5-1/4" Backplane	none	13	Pass-through
Peak Detector board	Peak Detector	14,15	Program-controlled +24IL disconnect relay
E103 5-1/4" Backplane	External	3	Activates +24IL connect relay

Detection Circuits

Board	Page	Description
E103 5-1/4" Backplane	3	Source of Fuse, System, Micro, Peak, and External
Peak Detector board	4	Data buffer to read above interlocks
Module Interface board	9	Source and data buffer for all module interlocks

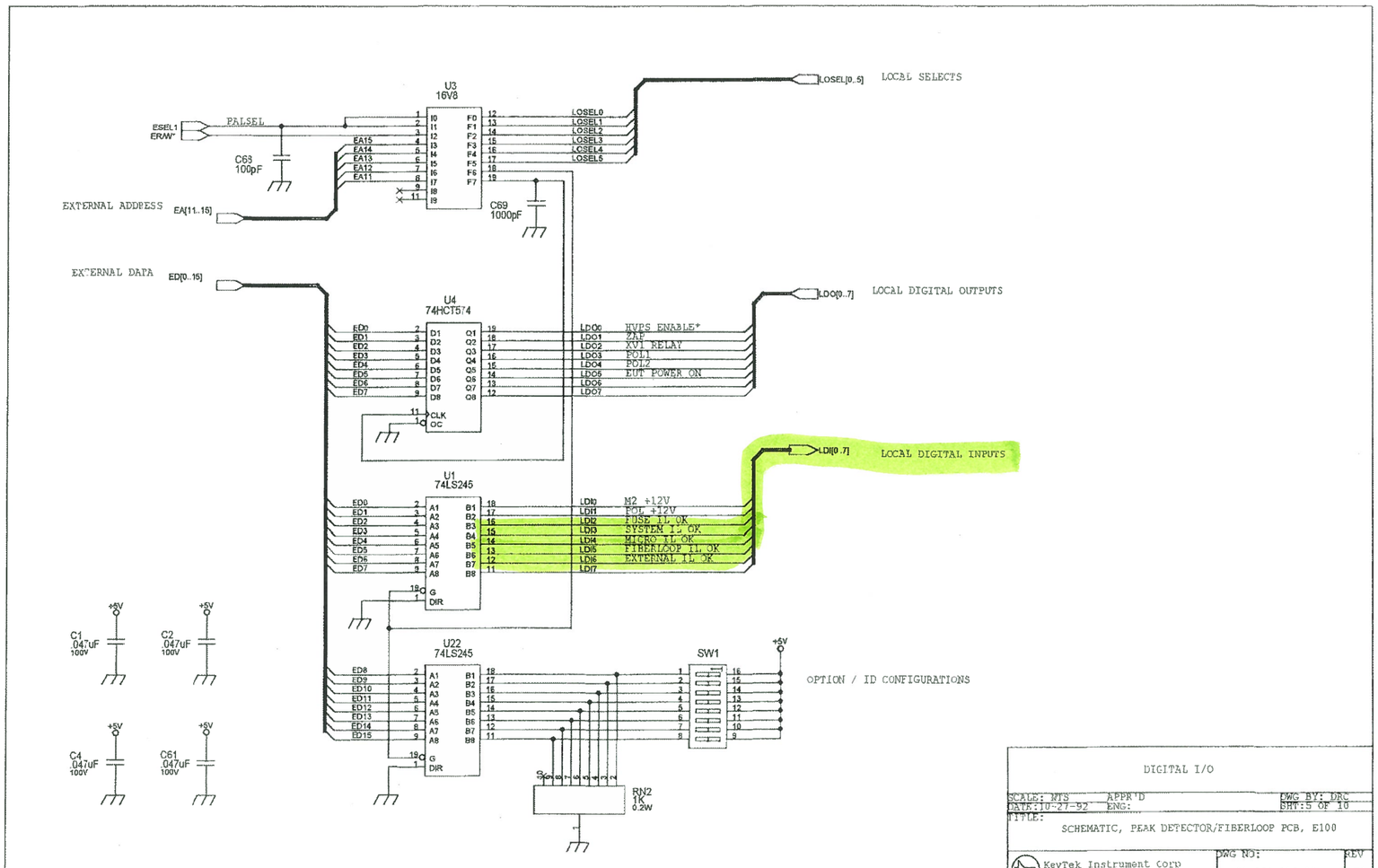
Indicator LEDs

Board	Page	Description
E103 5-1/4" Backplane	3	LEDs for Fuse, System, Micro, Peak, and External
Module Interface board	9	LEDs for module interlocks



INTERLOCK CIRCUITS

DRAWN IEC	DATE 01/08/14	ThermoFisher SCIENTIFIC	
APPROVED MIKE L			
		TITLE: SCHEMATIC, E100 BACKPLANE (5-1/4") PCB	
UNLESS OTHERWISE SPECIFIED ALL RESISTOR VALUES ARE IN OHMS ALL CAPACITOR VALUES ARE IN uF		SIZE C	DRAWING NO. 90-100-100-00 REV D
		SCALE: NONE	SHEET 8 OF 9



DIGITAL I/O

SCALE: NTS	APPR'D	DWG BY: DRC
DATE: 10-27-92	ENG:	SHT: 5 OF 10

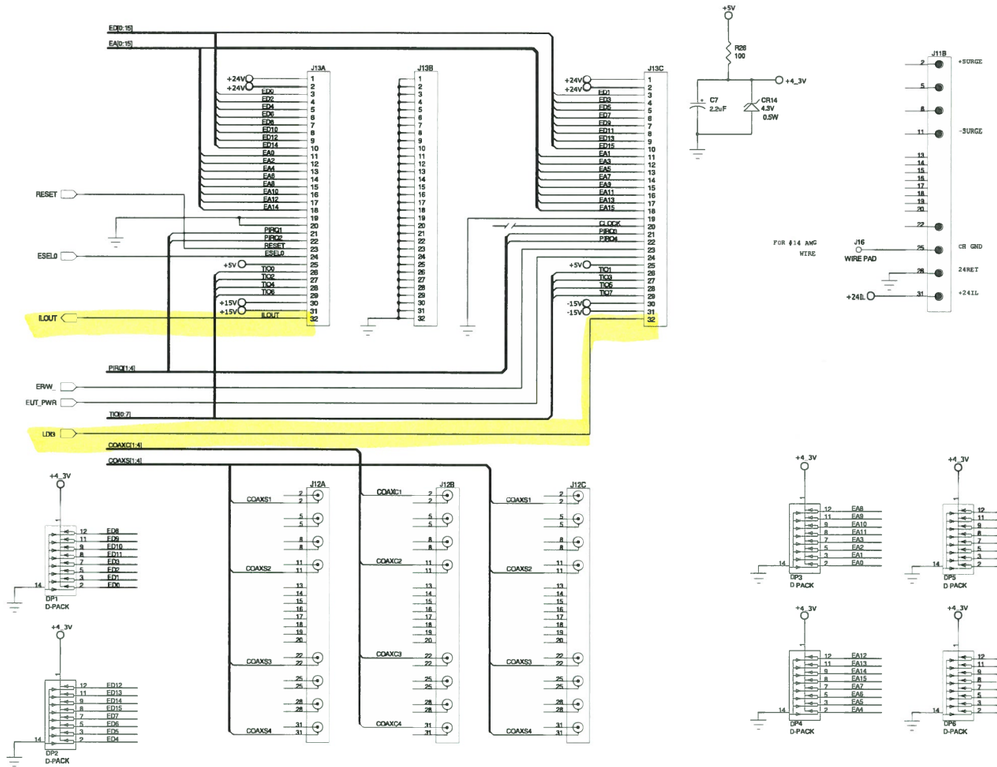
TITLE:
SCHEMATIC, PEAK DETECTOR/FIBERLOOP PCB, E100

KeyTek Instrument Corp
Massachusetts, USA

DWG NO: 90-100-130-00

REV E1

CONNECTORS TO BACKPLANE

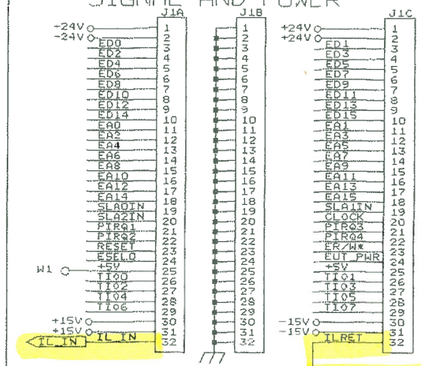


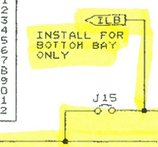
BACKPLANE CONNECTORS

DRAWN IEC	DATE 01/08/14	ThermoFisher SCIENTIFIC	
APPROVED MIKE L			
		TITLE: SCHEMATIC, E100 BACKPLANE (5-1/4") PCB	
UNLESS OTHERWISE SPECIFIED ALL RESISTOR VALUES ARE IN OHMS ALL CAPACITOR VALUES ARE IN uF		SIZE C	DRAWING NO. 90-100-100-00 REV D
		SCALE: NONE	SHEET 6 OF 9

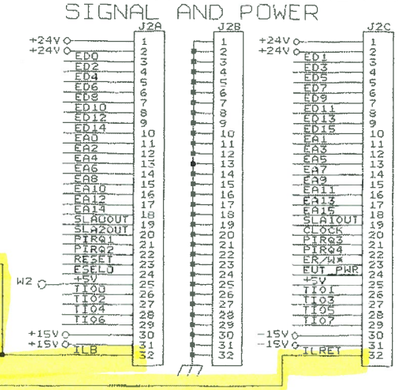
REV	DATE	DESCRIPTION OF CHANGE	ORIG	ELEC
D	15MAY95	CHANGED SHAPE DIAGRAMS FOR J6 - J10 AND BD CHANGES PER ECO #084-95	DRC	
E	4/11/96	NEW CIRCUITRY PER ECO 042-96	BEV	MEW
F	5/29/96	SEE ECO-053-96	GG	MEW
F1	1/12/97	ETCH REPLACED W/ H WIRE SEE ECO-003-97	BEV	
G	1/12/97	ADD ETCHES PER ECO-005-97	BEV	

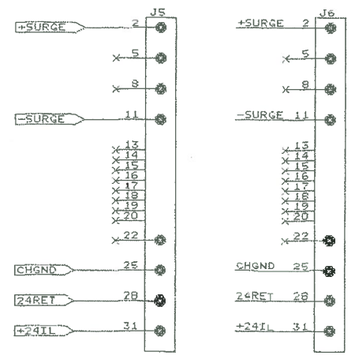
CONNECTORS FROM UPPER BAY SIGNAL AND POWER

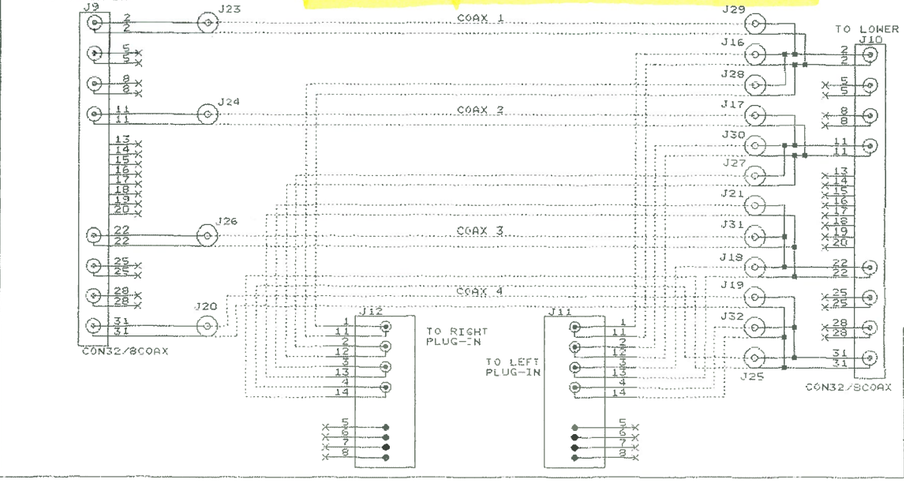


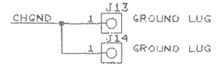


CONNECTORS TO LOWER BAY SIGNAL AND POWER









CONNECTIONS TO UPPER AND LOWER BAYS

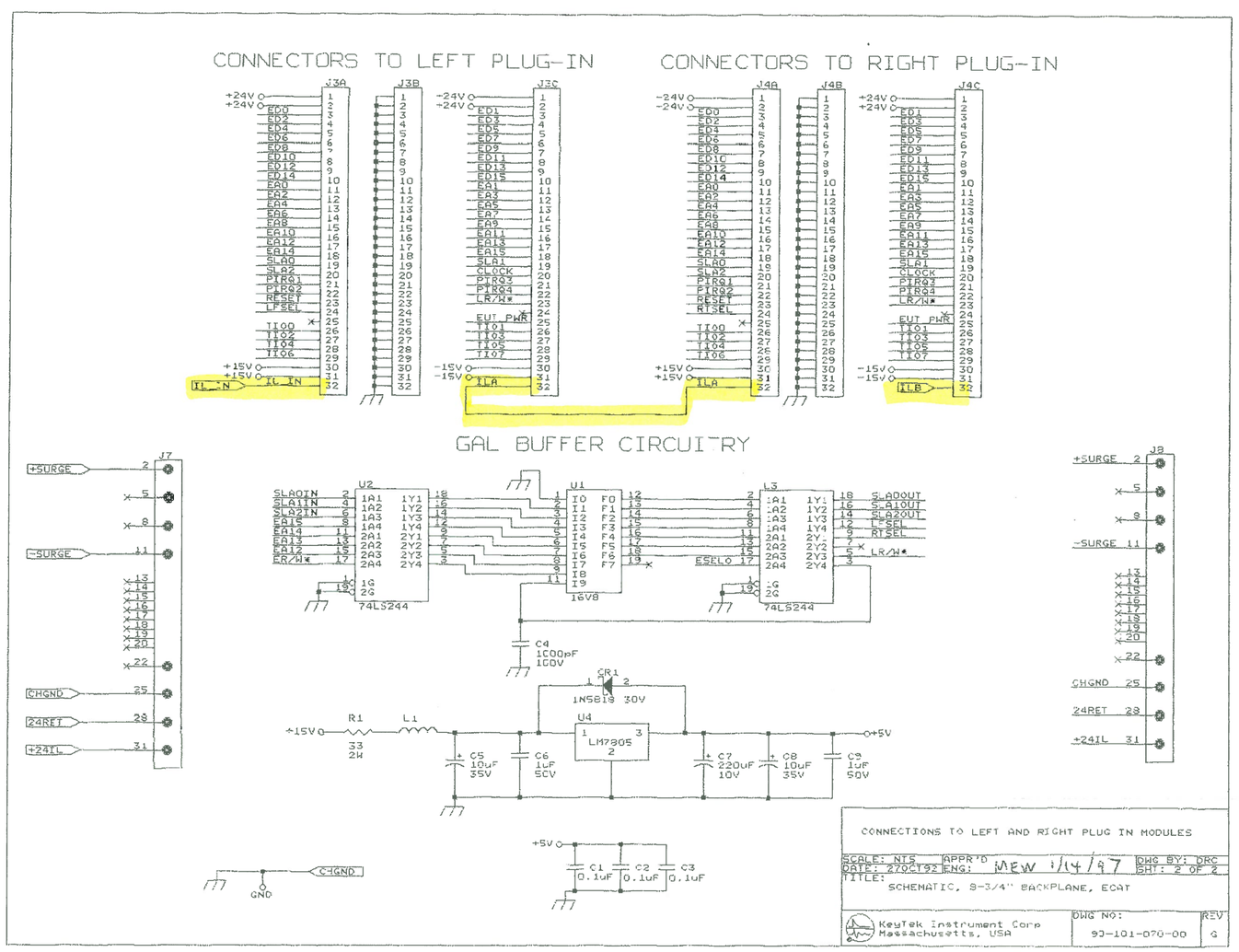
SCALE: NTS	APPR'D		DWG BY: DRC
DATE: 7MAY92	ENG: MEW 1/14/97		SHT: 1 OF 2

TITLE: SCHEMATIC, 8-3/4" BACKPLANE, ECAT

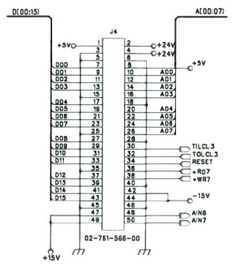
KeyTek Instrument Corp
Massachusetts, USA

DWG NO: 90-101-070-00

REV G

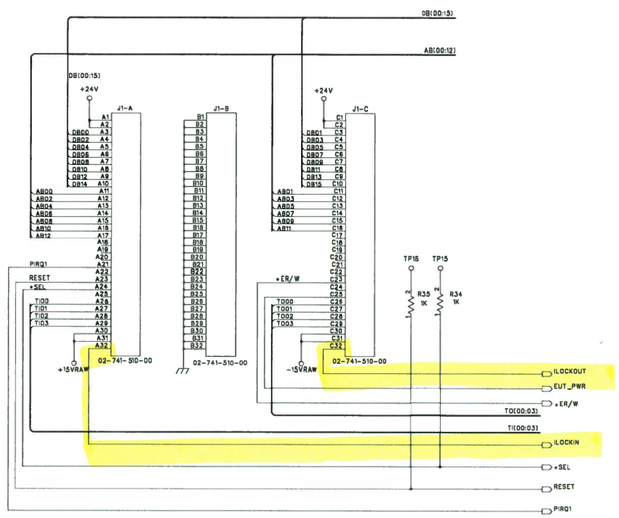

CONNECTORS TO LEFT PLUG-IN
CONNECTORS TO RIGHT PLUG-IN
J3A
J3B
J3C
J4A
J4B
J4C
GAL BUFFER CIRCUITRY
J7
U2
74LS244
U1
16V8
U3
74LS244
C4
1000pF
100V
CR1
1N5818 30V
U4
LM7805
R1
33
2W
L1
C5
10uF
35V
C6
1uF
50V
C7
220uF
10V
C8
10uF
35V
C9
1uF
50V
C1
0.1uF
C2
0.1uF
C3
0.1uF
J8
CONNECTIONS TO LEFT AND RIGHT PLUG IN MODULES
SCALE: NTS
APPR'D
DWG BY: DRC
DATE: 27OCT92
ENG:
MEW 1/14/97
SHT: 2 OF 2
TITLE:
SCHEMATIC, 8-3/4" BACKPLANE, ECAT
KeyTek Instrument Corp
Massachusetts, USA
DWG NO:
93-101-070-00
REV
G

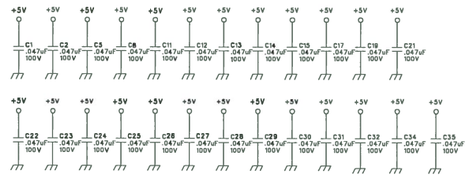
REVISIONS				
REV	DESCRIPTION	ORIG	EE	DATE
F	REDRAW IN PADS POWER-LOGIC	M.B	G.C.	9-29-00
G	ECO-006-04	M.B.	G.C.	1-29-04
G1	ECN-198-07, UPREV ONLY.	M.B.	G.C.	7-24-07
H	ECN-025-10	R.P.	FSD	4/29/10
J	ECN-104-11	R.P.	FSD	7/21/11

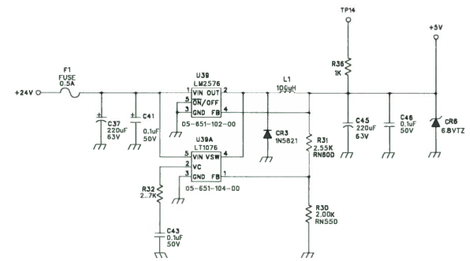


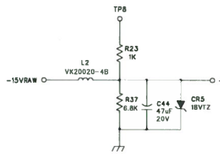
JUMPER CONFIGURATION CHART

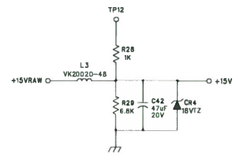
	PINS 1 & 2	PINS 2 & 3
JP1	INSTALLED	———
JMP1	COUPLER / DECOUPLER & PQF	SURGE & EFT
JMP2	COUPLER / DECOUPLER & PQF	SURGE & EFT
JMP3	RELAY DRIVER OUTPUT FOR ALL OTHERS	EFT DRIVE OUTPUT
JMP4	ALL OTHERS (EXCEPT E505)	E505
JMP5	E571	ALL OTHERS (EXCEPT E571)
JMP6	PQF	ALL OTHERS (EXCEPT PQF)
JMP7	E571	ALL OTHERS (EXCEPT E571)
JMP8	LTS	ECAT











UNLESS OTHER SPECIFIED		DRAWN BY MB	09/29/00	DATE	ThermoFisher SCIENTIFIC
TOLERANCES				DATE	
.XX	± 0.01	ELEC ENG		DATE	
.XXX	± 0.005	MECH ENG		DATE	TITLE: SCHEMATIC, PCB ECAT INTERFACE
ANGLES	± 1°	MFG ENG		DATE	
FRACTIONS	± ———				
DIMENSION ARE IN INCHES					
MACHINE FINISH ✓	FABRICATION SPEC	SCALE 1:1	SHEET 2 OF 7	DWG.SIZE B	DWG.NO. 90-500-110-00 REV J

REVISIONS				
REV	DESCRIPTION	ORIG	EE	DATE
F	REDRAW IN PADS POWER-LOGIC	M.B.	G.C.	9-29-00
G	ECO-006-04	M.B.	G.C.	1-29-04
G1	ECN-198-07, UPREV ONLY.	M.B.	G.C.	7-24-07
H	ECN-025-10	K.P.	FSD	6/29/10
J	ECN-104-11	K.P.	FSD	7/21/11

ILOCKIN

ILOCKOUT

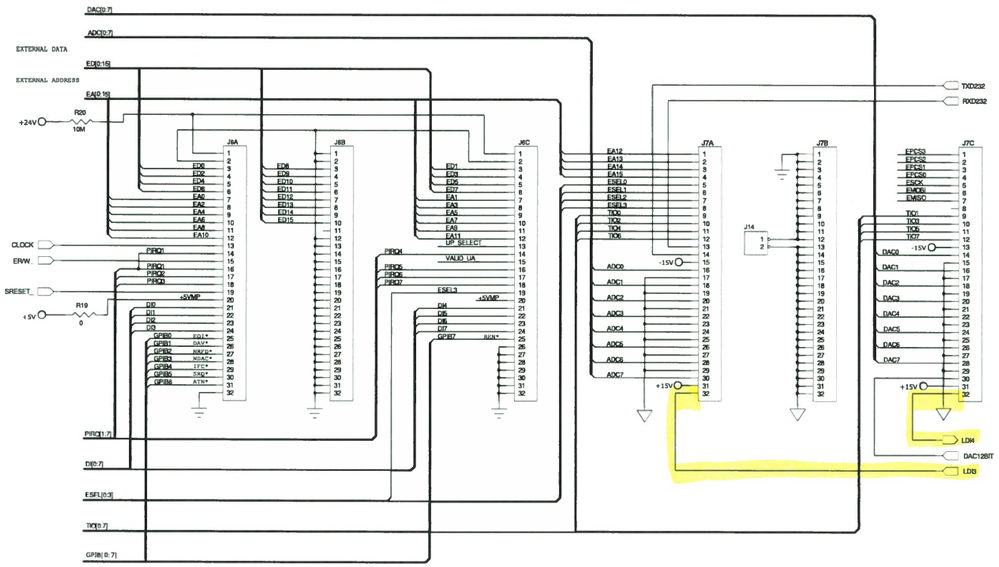
exist() function write/read

UNLESS OTHER SPECIFIED	DRAWN BY MB 09/29/00	DATE	ThermoFisher SCIENTIFIC
TOLERANCES		DATE	
.XX ± 0.01	ELEC ENG	DATE	
.XXX ± 0.005	MECH ENG	DATE	TITLE: SCHEMATIC, PCB ECAT INTERFACE
ANGLES ± 1°	MFG ENG	DATE	
FRACTIONS ±			
DIMENSION ARE IN INCHES			
MACHINE FINISH	FABRICATION SPEC	SCALE 1:1 / SHEET 6 of 7 / DWG.SIZE B	DWG.NO. 90-500-110-00 / REV J

REVISIONS				
REV	DESCRIPTION	ORIG	EE	DATE
F	REDRAW IN PADS POWER-LOGIC	M.B	G.C.	9-29-00
G	ECO-008-04	M.B.	G.C.	1-29-04
G1	ECN-198-07, UPREV ONLY.	M.B.	G.C.	7-24-07
H	ECN-023-10	R.P.	FSD	8/29/10
J	ECN-104-11	R.P.	FSD	7/21/11

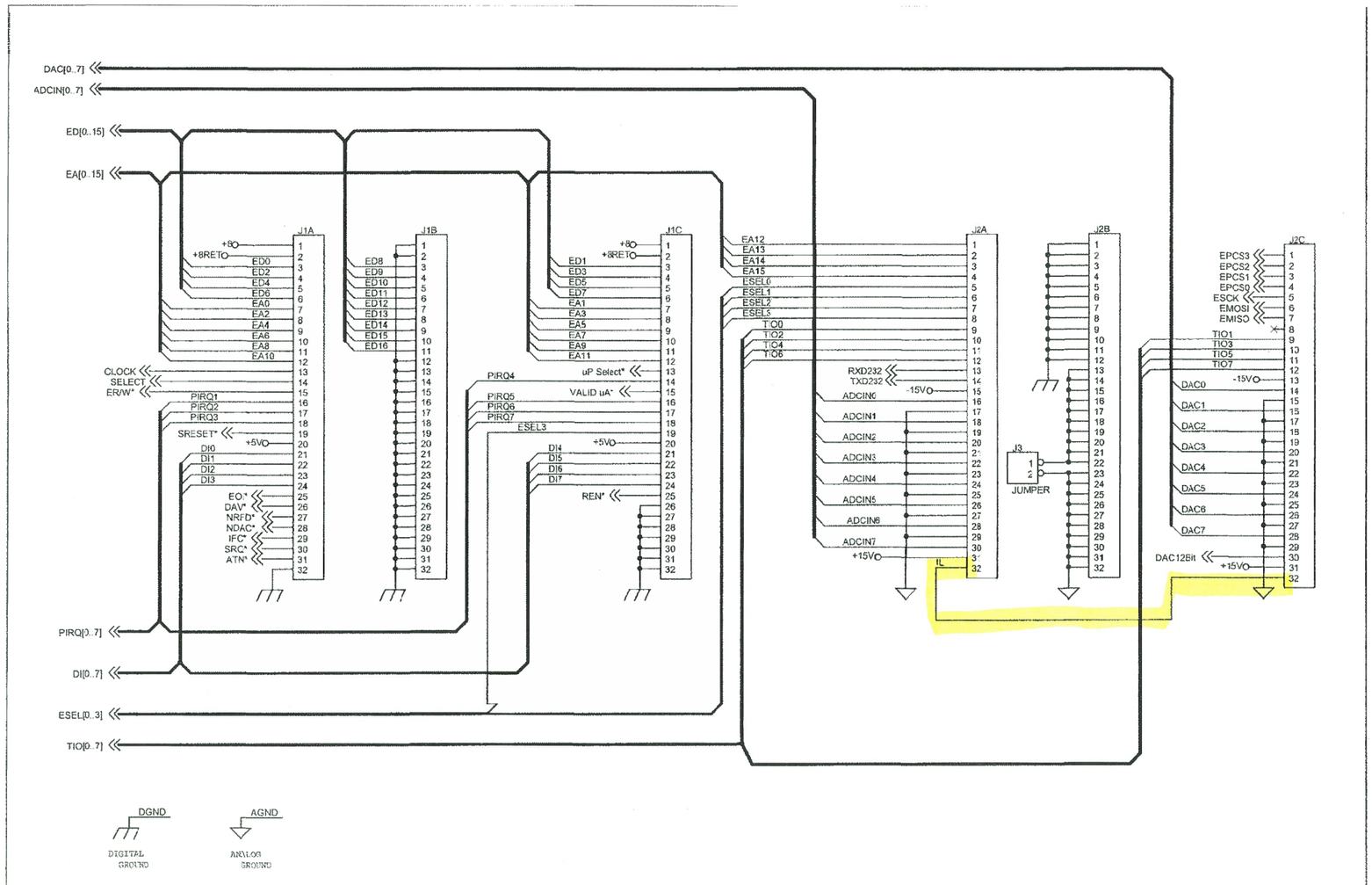
JUMPER JMP3 PINS 1 TO 2 FOR RELAY DRIVE OUTPUT.
JUMPER JMP3 PINS 2 TO 3 FOR EFT DRIVE OUTPUT.

UNLESS OTHER SPECIFIED	DRAWN BY MB	09/29/00 DATE	ThermoFisher SCIENTIFIC		
TOLERANCES		DATE			
.XX ± 0.01	ELEC ENG	DATE			
.XXX ± 0.005	MECH ENG	DATE	TITLE	SCHEMATIC, PCB ECAT INTERFACE	
ANGLES ± 1°	MFG ENG	DATE			
FRACTIONS ± ———					
DIMENSION ARE IN INCHES					
MACHINE FINISH ✓	FABRICATION SPEC	SCALE 1:1	SHEET 7 OF 7	DWG.SIZE B	DWG.NO. 90-500-110-00 REV J



MICROPROCESSOR CONNECTORS

DRAWN IEC	DATE 01/08/14	ThermoFisher SCIENTIFIC	
APPROVED MIKE L			
		TITLE: SCHEMATIC, E100 BACKPLANE (5-1/4") PCB	
UNLESS OTHERWISE SPECIFIED ALL RESISTOR VALUES ARE IN OHMS ALL CAPACITOR VALUES ARE IN uF		SIZE C	DRAWING NO. 90-100-100-00 REV D
		SCALE: NONE	SHEET 2 OF 9



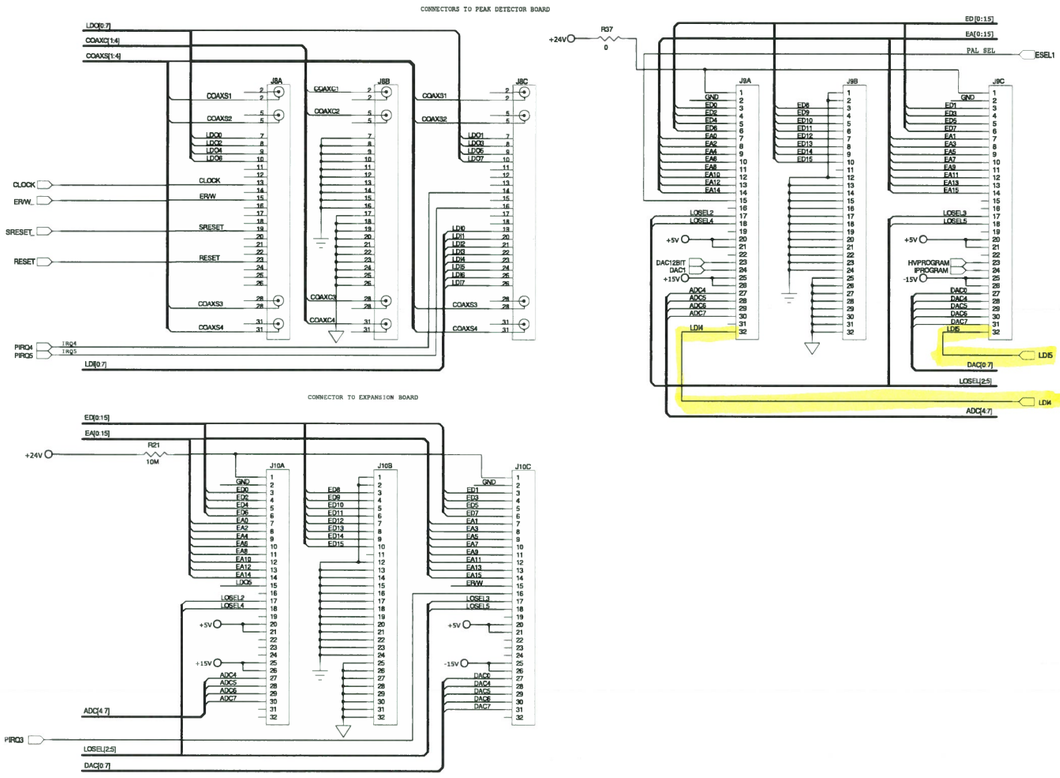
SCALE: NTS	APPR'D	DWG BY: DPC
DATE: 19MAY92	ENG:	SHT: 8 OF 10

TITLE: SCHEMATIC, MICRO-PROCESSOR CONTROLLER PCB

KeyTek Instrument Corp
Massachusetts, USA

DWG NO: 8C-144-160-XX

REV: F



PEAK DETECTOR/ EXPANSION BOARD CONNECTORS

DRAWN IEC	DATE 01/08/14	ThermoFisher SCIENTIFIC	
APPROVED MIKE L			
		TITLE: SCHEMATIC, E100 BACKPLANE (5-1/4") PCB	
UNLESS OTHERWISE SPECIFIED ALL RESISTOR VALUES ARE IN OHMS ALL CAPACITOR VALUES ARE IN uF		SIZE C	DRAWING NO. 90-100-100-00 REV D
		SCALE: NONE	SHEET 7 OF 9

INPUT / OUTPUT CONNECTORS

SCALE: NTS	APPR'D	DWG BY: DRC
DATE:10-2-92	ENG:	SHT:2 OF 10

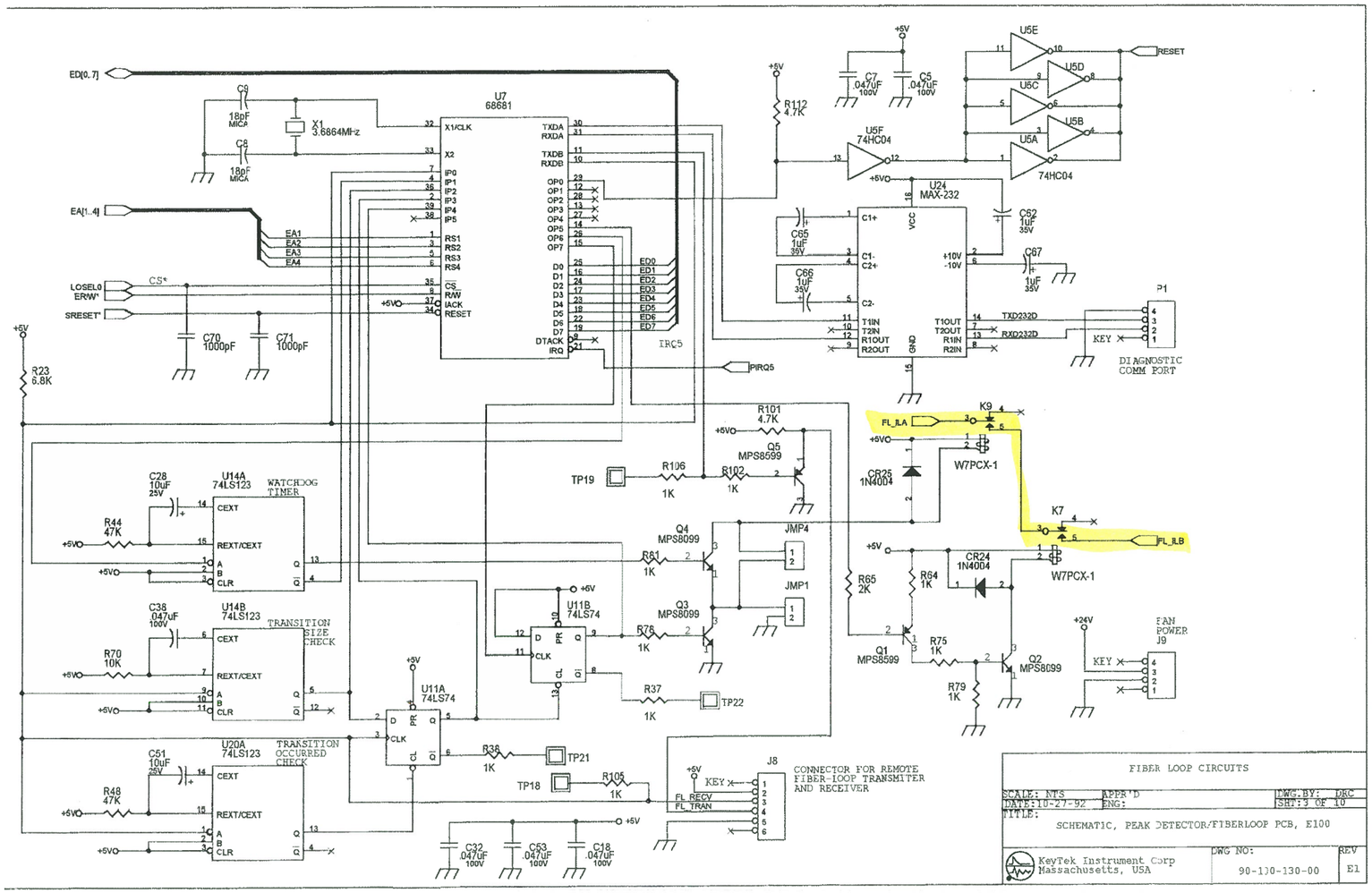
TITLE:

SCHEMATIC, PEAK DETECTOR/FIBERLOOP PCB, E100

KeyTek Instrument Corp
Massachusetts, USA

DWG NO: 90-100-130-00

REV E1


ED[0..7]
EA[1..4]
LOSEL0
ERW*
SRESET*
CS*
U7
68681
X1/CLK
X2
IP0
IP1
IP2
IP3
IP4
IP5
RS1
RS2
RS3
RS4
CS
R/W
IACK
RESET
TXDA
RXDA
TXDB
RXDB
OP0
OP1
OP2
OP3
OP4
OP5
OP6
OP7
D0
D1
D2
D3
D4
D5
D6
D7
DTACK
IRQ
ED0
ED1
ED2
ED3
ED4
ED5
ED6
ED7
IRC5
PIRQ5
C9
18pF
MICA
C8
18pF
MICA
X1
3.6864MHz
C70
1000pF
C71
1000pF
R23
6.8K
+5V
+5VO
R112
4.7K
C7
.047UF
100V
C5
.047UF
100V
U5E
U5D
U5C
U5B
U5A
U5F
74HC04
74HC04
RESET
U24
MAX-232
VCC
GND
C1+
C1-
C2+
C2-
T1IN
T2IN
R1OUT
R2OUT
T1OUT
T2OUT
R1IN
R2IN
+10V
-10V
C65
1uF
35V
C66
1uF
35V
C62
1uF
35V
C67
1uF
35V
TXD232D
RXD232D
P1
KEY
DIAGNOSTIC
COMM PORT
R101
4.7K
Q5
MPS8599
TP19
R106
1K
R102
1K
K9
FL_ILA
W7PCX-1
CR25
1N4004
K7
FL_ILB
W7PCX-1
JMP4
JMP1
Q4
MPS8099
Q3
MPS8099
R81
1K
R78
1K
R37
1K
TP22
U11B
74LS74
U11A
74LS74
C28
10uF
25V
U14A
74LS123
WATCHDOG
TIMER
R44
47K
C38
.047uF
100V
U14B
74LS123
TRANSITION
SIZE
CHECK
R70
10K
C51
10uF
25V
U20A
74LS123
TRANSITION
OCCURRED
CHECK
R48
47K
R36
1K
TP21
TP18
R105
1K
C32
.047UF
100V
C93
.047UF
100V
C18
.047UF
100V
J8
KEY
FL_RECV
FL_TRAN
CONNECTOR FOR REMOTE
FIBER-LOOP TRANSMITTER
AND RECEIVER
R65
2K
R64
1K
CR24
1N4004
Q1
MPS8599
R75
1K
Q2
MPS8099
R79
1K
+24V
FAN
POWER
J9
KEY
FIBER LOOP CIRCUITS
SCALE: NTS
APPR'D
DWG.BY: DKC
DATE:10-27-92
ENG:
SHT:3 OF 10
TITLE:
SCHEMATIC, PEAK DETECTOR/FIBERLOOP PCB, E100
KeyTek Instrument Corp
Massachusetts, USA
DWG NO:
90-130-130-00
REV
E1

```
void    ReportInterlockError(
uint    state,
FILE    *dest)
{
    char    faultmodule;
    uchar   faultlock;

    if(dest == NULL)
        return;

  if(!(state & 4))
    {
        fputs("The Interlock FUSE is OPEN",dest);
        return;
    }
    if(!(state & 8))
    {
        faultmodule = (char)((state >> 16) & 0x0F);
        faultlock   = (state >> 8) & 0x1f;
        /* Surge Module Interlocks */
        if(moddata[faultmodule].modinfo.typePrimary & TYPE_SRG)
        {
            /*
             * Faultlock = IL0 - IL1 (0x1E) ::= RIGHT Door
             * FaultLock = IL2 - Il3 (0x1C) ::= Not Used
             * Faultlock = IL4 - IL5 (0x18) ::= LEFT Door (FULL-WIDE Only)
             * FaultLock = IL6 - IL7 (0x10) ::= Not Used
             */
    if((moddata[faultmodule].modinfo.id == E522S_BOX) || (moddata[faultmodule].modinfo.id == E521S_BOX))
    {
        fputs("One of the SIDE COVERS on the\n\r",dest);
        if(moddata[faultmodule].modinfo.id == E522S_BOX)
            fputs("     E522 Main Cabinet is OPEN",dest);
        else
            fputs("     E521 Main Cabinet is OPEN",dest);
        return;
    }
    else
    {
        uchar left_interlock = IL_ILOCK4;  // all modules except E505B
        if ( moddata[faultmodule].modinfo.id == E505B_BOX )
            left_interlock = IL_ILOCK2;  // E505B is different because it has a built-in CDN
        if ( ( faultmodule & 0x01 ) || ( faultlock == left_interlock ) )
            fputs( IL_LDOOR, dest );
        else
            fputs( IL_RDOOR, dest );
        fprintf( dest, IL_GDOOR, ( (state>>17) & 0xff ) + 1 );
    }
    return;
}
/* Coupler Module Interlocks (Except EFT/COUPLER combinations) */
if((moddata[faultmodule].modinfo.typePrimary & TYPE_CPLGEN) &&
    (!(moddata[faultmodule].modinfo.typePrimary & TYPE_EFT)) &&
    (!(moddata[faultmodule].modinfo.typePrimary & TYPE_EMC)))
{
    /*
     * Faultlock = IL0 - IL1 (0x1E) ::= RIGHT Door
     * FaultLock = IL2 - Il3 (0x1C) ::= LEFT Door (FULL-WIDE Only)
     * Faultlock = IL4 - IL5 (0x18) ::= Not Used
     * FaultLock = IL6 - IL7 (0x10) ::= Not Used
     */
    if((moddata[faultmodule].modinfo.id == E4556_BOX) && (faultlock == IL_ILOCK0))
    {
        LCD_gotoxy(2,4);
        fputs(IL_RearDoor,dest);
    }
    else
    {
        if((faultmodule & 0x01) || (faultlock == IL_ILOCK2))
            fputs(IL_LDOOR,dest);
        else
            fputs(IL_RDOOR,dest);
        fprintf(dest,IL_GDOOR,((state>>17) & 0xff)+1);
    }
    return;
}
/* PQF Module Interlocks */
if(moddata[faultmodule].modinfo.typePrimary & TYPE_PQF)
{
    switch(moddata[faultmodule].modinfo.id)
    {
        case EP61_BOX:
        case EP62_BOX:
            switch(faultlock)
            {
                case IL_ILOCK0: /* Door */
                    fputs(IL_PQFDOOR,dest);
                    break;
```

```
                case IL_ILOCK2: /* Snub */
                    fputs(IL_PQFSNUB,dest);
                    break;
                case IL_ILOCK4: /* Temp */
                    fputs(IL_PQFTEMP,dest);
                    break;
                case IL_ILOCK6: /* Power*/
                    fputs(IL_PQFPWR,dest);
                    break;
            }
            break;
        case EP3_BOX:
            if((faultmodule & 0x01) || (faultlock == IL_ILOCK2))
                fputs(IL_LDOOR,dest);
            else
                fputs(IL_RDOOR,dest);
            fprintf(dest,IL_GDOOR,((state>>17) & 0xff)+1);
            break;
    }
    return;
}
/* EFT Module Interlocks */
if(moddata[faultmodule].modinfo.typePrimary & TYPE_EFT)
{
    switch(moddata[faultmodule].modinfo.id)
    {
        case E421_BOX:
        case E422_BOX:
            switch(faultlock)
            {
                case IL_ILOCK0: /* Right Hand Door  */
                case IL_ILOCK2: /* Left Hand Door       */
                    if((faultmodule & 0x01) || (faultlock == IL_ILOCK2))
                        fputs(IL_LDOOR,dest);
                    else
                        fputs(IL_RDOOR,dest);
                    fprintf(dest,IL_GDOOR,((state>>17) & 0xff)+1);
                    break;
                case IL_ILOCK4: /* Cable */
                    if(faultmodule & 0x01)
                        fputs(IL_LCABLE,dest);
                    else
                        fputs(IL_RCABLE,dest);
                    fprintf(dest,IL_GCABLE,((state>>17) & 0xff)+1);
                    break;
                case IL_ILOCK6: /* UNUSED */
                default:
                    break;
            }
            return;
        default:
            switch(faultlock)
            {
                case IL_ILOCK0: /* Right Hand Door  */
                case IL_ILOCK4: /* Left Hand Door       */
                    if((faultmodule & 0x01) || (faultlock == IL_ILOCK4))
                        fputs(IL_LDOOR,dest);
                    else
                        fputs(IL_RDOOR,dest);
                    fprintf(dest,IL_GDOOR,((state>>17) & 0xff)+1);
                    break;
                case IL_ILOCK2: /* Cable */
                    if(faultmodule & 0x01)
                        fputs(IL_LCABLE,dest);
                    else
                        fputs(IL_RCABLE,dest);
                    fprintf(dest,IL_GCABLE,((state>>17) & 0xff)+1);
                    break;
                case IL_ILOCK6: /* UNUSED */
                default:
                    break;
            }
            return;
    }
}
if(moddata[faultmodule].modinfo.typePrimary & TYPE_EMC)
{
    switch(moddata[faultmodule].modinfo.id)
    {
        case ERI1_BOX:
            switch(faultlock)
            {
                case IL_ILOCK0: /* Right Hand Door  */
                    if(faultmodule & 0x01)
                        fputs(IL_LDOOR,dest);
                    else
                        fputs(IL_RDOOR,dest);
                    fprintf(dest,IL_GDOOR,((state>>17) & 0xff)+1);
```

```
            break;
        case IL_ILOCK4: /* Thermal Interlock        */
            fputs(IL_PQFTEMP,dest);
            break;
    }
    return;
case ERI3_BOX:
    switch(faultlock)
    {
        case IL_ILOCK0: /* Right Hand Door  */
        case IL_ILOCK2: /* Left Hand Door       */
            if(faultlock == IL_ILOCK2)
                fputs(IL_LDOOR,dest);
            else
                fputs(IL_RDOOR,dest);
            fprintf(dest,IL_GDOOR,((state>>17) & 0xff)+1);
            break;
        case IL_ILOCK4: /* Thermal Choke Interlock      */
            fputs(IL_AMPTEMP0,dest);
            break;
        case IL_ILOCK6: /* Thermal Resistor Interlock   */
            fputs(IL_AMPTEMP1,dest);
            break;
    }
    return;
case EP71_BOX:
case EP72_BOX:
case EP73_BOX:
case EP91_BOX:
case EP92_BOX:
case EP93_BOX:
case EP94_BOX:
    switch(faultlock)
    {
        case IL_ILOCK0: /* Right Hand Door  */
        case IL_ILOCK2: /* Left Hand Door       */
            if(faultlock == IL_ILOCK2)
                fputs(IL_LDOOR,dest);
            else
                fputs(IL_RDOOR,dest);
            fputs(IL_BDOOR,dest);
            break;
        case IL_ILOCK4: /* Panel Covers     */
            fputs("One of the SIDE COVERS on the\n\r",dest);
            fputs("     BASE Main Module is OPEN",dest);
            break;
    }
    return;
}
}
if(moddata[faultmodule].modinfo.typePrimary & TYPE_CLAMP)
{
    switch(faultlock)
    {
        case IL_ILOCK0: /* Right Hand Door  */
        case IL_ILOCK2: /* Left Hand Door       */
            if(faultlock == IL_ILOCK2)
                fputs(IL_LDOOR,dest);
            else
                fputs(IL_RDOOR,dest);
                fprintf(dest,IL_GDOOR,((state>>17) & 0xff)+1);
                break;
        }
        return;
    }
    /* Unknown Interlock */
    fputs("UNKNOWN Interlock Condition\n\r",dest);
    fprintf(dest,"     Interlock Code: %#05X",state);
}
else
    if(!(state & 16))
        fputs("The Micro Board is LOOSE",dest);
    else
        if(!(state & 32))
            fputs("The Peak Detector Board is LOOSE",dest);
        else
            if(!(state & 64))
                fputs("The EXTERNAL Interlock is OPEN",dest);
return;
}
```

```
int exist(
int d)
{
    int base;
    unsigned short  temp;

    base = 0x60000+(d*0x1000);
    *((unsigned short *)(base+0x600)) = 0x00;
    temp = *((unsigned short *)(base+0x600));
    if(temp & 0x80)
        return(0);
    *((unsigned short *)(base+0x600)) = 0x01;
    temp = *((unsigned short *)(base+0x600));
    if(!(temp & 0x80))
        return(0);
    return(1);
}

int istat(
int d)
{
    int base;
    unsigned short  temp;

    base = 0x60000+(d*0x1000);
    temp = *((unsigned short *)(base+0x600));
    temp = (temp & 0x1f) ^ 0x1f;
    return(temp);
}

int ilock_stat(void)
{
    int il;
    int is;
    int i;

    il = interlockData;
    il &= 0x7C;
    if((il & 0x04) && (!(il & 0x08)))
    {
        for(i = 0; i < 16; i++)
        if(exist(i))
        {
            is = istat(i);
            if((is) && (!(is & 0x01)))
            {
                il = (il & 0xff) | (i << 16) | (istat(i) << 8);
                i = 20;
            }
        }
    }
    return(il);
```