# CXAR/16 & 32
# COAXIAL SWITCH MATRIX
# with IF-12 or CM-12 Control Module

## Table of Contents

# DRAWINGS

| DRWG/SCHEMATIC # | DESCRIPTION |
|---|---|
| **MISCELLANEOUS** | |
| 11-05-50-1 | Front Panel LED |
| 11-08-50 | CM/32 Relay Driver |
| 11-11-50 | RD-9 Relay Driver Bd |
| 11-17-50 | RD-8/16 Relay Driver Bd |
| 11-21-50 | IF-12 Control Mod |
| 14-20-20 | Top/Bottom Panel |
| 15-60-22 | PB/32 Channel Front Panel |
| 16-16-50 | EIF Module |
| 21-30-21 | CXAR/16 Rear Panel |
| 21-35-21 | CXAR/32 Rear Panel |
| 99-00-30 | 20 Pin Drive Cable |
| 99-02-30 | 16 Pin Data Cable Diagram |
| 99-190-30 | Standard Power Supply Wiring Diagram |
| **CHASSIS (MF AND EXP)** | |
| 21-30-00 | CXAR/16-MF |
| 21-35-00 | CXAR/32-MF |
| 21-23-00 | CXAR/32-EXP |
| **SWITCH MODULE SCHEMATICS** | |
| 21-00-50 | CXR/2x1-C2 |
| 2-101-1 | CXR/2x1-FORM A |
| 21-01-50 | CXR/2x1-GT-50 |
| 21-02-50 | CXR/8x1-2A |
| 21-13-50 | CXR/2(4x1)-1LL |
| 21-14-50 | CXR/8-1T |
| 21-14-51 | CXR/8x1-1S |
| 21-14-52 | CXR/4-1T |
| 21-14-53 | CXR/4x1-1S |
| 21-24-50 | CXR/8x1-1HT |
| 21-24-51 | CXR/8x1-1HS |
| 21-24-52 | CXR/4x1-1HT |
| 21-24-53 | CXR/4x1-1HS |
| 30-10-50 | CXR/2x1-G-75 |
| 30-17-50 | CXR/8x1-G-50 |
| 30-18-50 | CXR/8x1-G-75 |
| 30-19-50 | CXR/8x1-GT-75 |
| 30-21-50 | CXR/4x2-G-50 |
| 30-22-50 | CXR/4x2-G-75 |
| 30-25-50 | CXR/2x1-G-50 |
| 30-29-50 | CXR/8x1-GT-50 |
| 30-30-50 | CXR/2x1-GT-75 |
| 30-31-50 | CXR/2x1-G-75-F |

## 1.0 ADDENDUM

This page is intentionally left blank

## 2.0 GETTING STARTED

Unpack the unit and make sure it has arrived undamaged. Inspect for dents, bent handles, major scratches and missing or loose parts. Note that many of the items listed individually on the packing list are already installed within the chassis, rather than being packed separately.

Compare the Shipped Configuration List on the last page of the Quick Start Guide that shipped with the unit with the included packing slip to verify that all components and ordered parts have been received. If any purchased items are missing please contact your Sales Representative at 1-800-346-3117, 585-381-4740 or sales@cytec-ate.com. Utilize the Shipped Configuration List to identify which drawings and diagrams refer to the specific unit ordered.

Next, set up the chassis on either a bench or rack. The front handles allow the unit to be bolted to a standard 19 inch rack. No special setup tools are needed.

For AC powered units, a Power Cord should be included in the box. Plug one end into the chassis and the other into a three prong commercial AC outlet. The unit will operate from one of two AC voltage ranges: 100/140 or 200/260. There is a fuse holder built into the front panel of the unit. The power supply on AC units is auto-ranging.

Install the appropriate remote control cable to the controlling computer: RS232, IEEE488 (GPIB) or Ethernet. Cytec provides a one to one RS232 D9 cable but does not provide Ethernet or GPIB cables with the unit.

Turn the unit ON via the toggle switch located on the front panel. The front panel Power LED should illuminate.

Study the sections of this manual which deal with your control interface (RS232 or Ethernet), as well as the controlling command syntax. A group of programming examples are included in appendices at the end of the manual and provide a good structure to work from. Example driver programs may be included on a disc if requested. Drivers can also be downloaded from Cytec's web site at: http://www.cytec-ate.com/support

You should now be able to begin writing useful code. **Always write and debug code thoroughly before hooking up live signals to the matrix!** This equipment gives you full control over what is switched to where and will not prevent you from making potentially harmful connections. That is, nothing in the system prevents the switching of excessive power, which can damage or destroy the relay contacts or digital switches.

## 3.0 GENERAL

CXAR/16 **(Drwg.#21-30-00)** and CXAR/32 **(Drwg.#21-35-00)** Mainframes are 19" rack mounting chassis supplying either 16 or 32 switch points respectively. A standard CXAR Mainframe holds two power supplies, one control module, front panel status LED's and up to four CXAR switch modules. Optional manual controls include push buttons. The switch modules are mounted on the rear panel with their signal connectors protruding through the panel. The switch modules are typically arranged so that module #0 is on the top when viewing the panel from the back.

### 3.1 CHASSIS DESCRIPTION

The CXAR/16 and CXAR/32 units include both Mainframe and Expansion chassis. A Mainframe is a single standalone chassis which may be controlled either remotely via computer or locally via the optional manual control. One or more Expansion chassis can be controlled only from the dedicated MESA II control chassis. All chassis are standard 19" rack mounting.

The CXAR/16 supplies 16 switch points, and the CXAR/32 supplies 32 switch points.

The power supply operates from the AC line supply via a fused line cord adapter on the rear panel. This adaptor also houses the ON/OFF switch. Several different switch modules, (**Section 4.0**), of varying sizes and options are available, which allow the system to be custom tailored to the user's specific requirements.

#### 3.1.1 MAINFRAME CHASSIS

A typical CXAR/16 Mainframe can be seen in **Drwg. #21-30-00,** while a CXAR/32 Mainframe can be seen in **Drwg. #21-35-00**.

**<u>Each CXAR Mainframe chassis contains</u>***:*

(One ) Front Panel Display/Relay Driver Modules **(Section 3.4)**
(Up to Four) CXAR Switch Modules **(Section 4.0)**
(One) Control Module **(Section 5.0)**
(One) 12 Volt Power Supply **(Section 3.3)**
(One) 5 Volt Power Supply **(Section 3.3)**

CXAR/16 and CXAR/32 chassis are 15" deep chassis. These are built with the front panel LED Display shown on Drwg. **#11-05-50-1** and include the optional PB/16 or PB/32 Pushbutton Manual Controls described in **Section 3.4.2**.

#### 3.1.2 EXPANSION CHASSIS

An Expansion Chassis differs from a Mainframe in that it cannot operate as a standalone device and must be controlled remotely from a MESA II. An example CXAR/32 Expansion chassis can be seen in **Drwg. #21-23-00.**

*The CXAR Expansion Chassis holds basically the same components as the CXAR Mainframe, with the following exceptions:*

- There is no dedicated control module in the Expansion Chassis. Instead, an expansion cable from the MESA II plugs into the Expansion Chassis' rear panel at the control interface connector.
- There is one CM/32 Relay Driver **Drwg. #11-08-50**.
- Relay power is supplied from a +12V supply in the Expansion chassis.
- Logic +5V power comes from the MESA II only.

For Expansion Chassis, control signals originate in the MESA II Control Mainframe. These signals are routed from the MESA II via the supplied ribbon Expansion Cable to the Interface Connector located on the chassis' rear panel. The Interface Connector in turn wires to the CM/32 Driver Module(s) located inside the Expansion Chassis. The CM/32's drive the coaxial switch modules. The relay coils are wired in parallel with the front panel LEDs.

## 3.2 SPECIFICATIONS

| | |
|---|---|
| **Dimensions:** | 19" Rack Mounting x 3.5" High x 15" Deep |
| **Weight:** | Maximum weight with full complement of modules less than 20 lbs. |
| **Power:** | Less than 60 W @ 100-130 Vac or@ 200-260 Vac |
| **Environment:** | |
| **Operating:** | 0°C to 50°C @ 95% Relative Humidity |
| **Storage:** | -25°C to 65°C @ 95% Relative Humidity |
| **Capacity:** | Up to four CXAR switch modules standard |
| **Expansion Capacity:** | Up to 16 Expansion Units with one MESA II Unit |
| **Display:** | One Power LED and up to 32 Drive Status LED's |
| **Control Mode:** | Ethernet TCP/IP and RS232 standard: IEEE488 Optional |

## 3.3 POWER SUPPLY

The CXAR/16 and CXAR/32 Mainframe Chassis are built with two power supplies. One has a regulated +12 volt output for driving the relays, and the second has a regulated +5 volt output for the logic as shown in **Drwg. # 99-190-30**.

Standard CXAR/16 and CXAR/32 Expansion Chassis are built with power supplies. The +5 volts for the logic is supplied from the MESA II Controller via the Expansion Interface Module as shown in **Drwg. #16-16-50**. All power supplies will operate from 100-140 volts or 200-260 volts at 47-63 Hz.

The CXAR/32 Expansion w/ Power Supply chassis, as shown in **Drwg. #21-23-00**, includes a +12V supply for switch relays.

The supplies are wired out to the Selectable AC Input Module on the rear panel, which also holds the chassis ON/OFF Switch. The user can select one of two AC voltage ranges: 110/120 Volts or 220/240 volts AC. To change the selected voltage, remove the fuse cartridge using a small blade screw driver or a similar tool. Select the desired voltage by matching the arrow on the fuse cartridge to the arrow located on the Input Module's lower right corner. *Replace the fuse cartridge making sure the voltage selection arrow aligns with the arrow located on the Input Module.*

Two fuses are held in the fuse cartridge, with 220/240 VAC fused separately from 110/120 VAC. See the "Shipped Configuration" page that shipped with the unit for fuse sizes.

## 3.4 FRONT PANEL

The CXAR/16 and CXAR/32 front panels have a power LED, and either 16 or 32 switch status LEDs and may include the optional Push Button Manual Control.

### 3.4.1 LABELING AND PROGRAMMATIC REFERENCES

The CXAR/16 and CXAR/32 front panels are typically labeled to match the chassis' built-in switching configuration. Each switch point is assigned a single front panel status LED. Each row of LED's represents one logical switch module, and each LED in that row represents a single switch within that module.

Programmatically, any command references one switch point and its corresponding LED directly. Commands are of the form: ***Command_Module_Switch***, where:

***Command*** - One letter mnemonic indicating function to be performed (L for Latch, U for Unlatch, etc.)
***Module*** - An integer specifying the logical module being address. Matches front panel configuration.
***Switch*** - An integer that specifies the logical switch being addressed. Also matches front panel labeling and chassis configuration.

**NOTE:** Switchpoints in the CXAR/16 and CXAR/32 can be addressed as either ***Module, Switch***, where:

***Module*** = **0, 1** for the CXAR/16 or **0 - 3** for the CXAR/32
***Switch*** = **0 - 7** for both chassis types

Alternately, the Switchpoint can be addresses as simple as ***Switch***, without Module where:

***Switch*** = 0 - 15 for the CXAR/16 or 0 - 31 for the CXAR/32

**Examples**: Assume 2 CXAR/8x1 Switch Modules installed

"L 0 7" - Latches Module 0, Switch 7
"L 7" - Latches Module 0, Switch 7 (performs same function as above)
"U 1 3" - Unlatches Module 1, Switch 3
"U 11" - Unlatches Switch 11" (performs same function as above).

***See Section 5.0 for complete programming information.***

Note that the Latch command causes the assigned front panel LED to illuminate, indicating that the switch has been closed. Similarly, the Unlatch command will extinguish a LED if it is already illuminated (indicating a closed switch has been opened).

# 4.0 SWITCH MODULES

## 4.1 CXR/8x1-G-50, CXR/4x1-G-50, CXR/8x1-G-75 and CXR/4x1-G-75

Each CXR/8x1-G-50 module **(Drwg.#30-17-50**) and CXR/8x1-G-75 module **(Drwg.#30-18-50**) has seven single pole Form C relays arranged as an 8x1 tree switch multiplexer. The characteristic impedance is either 50 or 75 ohms. These modules are used in high frequency applications, up to approximately 1 GHz. and can be built with BNC, SMA or SMB connectors. The following table illustrates common specifications for the switch module:

**CXR/8x1-G-50 and CXR/4x1-G-50 (50 Ohm Modules)**

| | |
|---|---|
| Bandpass (with SMA connectors): | 1.6 GHz (-3 dB) |
| Isolation (with BNC or SMA connectors): | -50 dB @ 1 GHz |
| Bandpass (with BNC connectors): | 1.0 GHz (-3 dB) |

**CXR/8x1-G-75 and CXR/4x1-G-75 (75 Ohm Modules)**

| | |
|---|---|
| Bandpass (with SMB or BNC connectors): | 1.2 GHz (-3 dB) |
| Isolation (with SMB or BNC connectors): | -60 dB @ 1 GHz |

Channel #0 is Normally Closed and is connected to the Common when the module is unenergized or when Drive #0 is latched. Channels #1 through #7 individually connect to Common when latched. Two (or more) channels can never be connected simultaneously. The relay control tables on the drawings indicate which relays are energized when a specific channel is latched (closed). Diodes on the modules or drive adaptor board ensure that the proper relays are selected for each channel. This module is available with SMA or BNC connectors, and it may be cut for 4x1 operation.

The format for the CXR/8x1-G Switch Module part number is:

**CXR/8x1-G-*Impedance-Connector***

| | |
|---|---|
| ***Impedance-*** | Indicates 50 ohms (**-50**) or 75 ohms (**-75**) Characteristic Impedance |
| ***Connector*-** | Indicates BNC (**-N**), SMA (**-SMA**) or SMB (**-SMB**) Signal Connectors |

*Example:* The CXR/8x1-G-50-N has a characteristic impedance of 50 ohms and is built with BNC connectors.

## 4.2 CXR/8x1-GT-50, CXR/4x1-GT-50, CXR/8x1-GT-75 and CXR/4x1-GT-75

Both the CXR/8x1-GT-50 **(Drwg. #30-29-50)** and the CXR/8x1-GT-75 **(Drwg. #30-19-50)** are built with nine single pole and three double pole, high frequency Form C relays arranged as an 8x1 tree switch multiplexer. The characteristic impedance is either 50 or 75 ohms. These modules have a Form C relay at each input that connects unlatched channels to terminating resistors. There is no normally closed path. The following switch module specifications are typical:

**CXR/8x1-GT-50 and CXR/4x1-GT-50 (50 Ohm Modules)**

| | |
|---|---|
| Bandpass (with SMA connectors): | 1.5 GHz (-3 dB) |
| Isolation (with BNC or SMA connectors): | -60 dB @ 1 GHz |
| Bandpass (with BNC connectors): | 1.0 GHz (-3 dB) |

**CXR/8x1-GT-75 and CXR/4x1-GT-75 (75 Ohm Modules)**

| | |
|---|---|
| Bandpass (with SMB or BNC connectors): | 1.0 GHz (-3 dB) |
| Isolation (with SMB or BNC connectors): | -60 dB @ 1 GHz |

Channels #0 through #7 individually connect to the common when energized. Two (or more) channels can never be connected simultaneously. The relay control tables on the drawings show which relays are energized when a specific channel is latched (closed). Diode logic on the modules ensures that the proper relays are selected for each channel. This module is available with SMA, SMB or BNC connectors, and it may be cut for 4x1 operations.

The part number format for CXR/8x1-GT Switch Modules is:

**CXR/8x1-GT-*Impedance-Connector***
***Impedance-*** Indicates 50 ohms (**-50**) or 75 ohms (**-75**) Characteristic Impedance
***Connector*-** Indicates BNC (**-N**), SMA (**-SMA**) or SMB (**-SMB**) Signal Connectors

<u>*Example:*</u> The CXR/8x1-GT-50-N has a characteristic impedance of 50 ohms and is built with BNC connectors.

## 4.3 CXR/8x1-2A-N and CXR/8x1-2A-TBNC

The CXR/8x1-2A switch modules have nine I/O connectors, eight inputs and one common, and are built with either BNCs or Twin BNCS. The modules use double pole relays that switch both conductors of the coaxial connectors as shown in **Drwg #21-02-50**.

When using Twin BNCs, the two inner conductors are switched and the connector shields are bussed to ground. When BNC connectors are used, both the inner conductor and the shield are switched.

Two additional isolation relays switch the conductors of the COM connector. The isolation relays, K8 and K9, are open until one of the two pole relays is energized. These modules are typically used to switch balanced line 100 ohm differential pairs, and may be cut for 4x1 configurations.

## 4.4 CXR/2x1-FORM A-1 and CXR/2x1-FORM A-2

The CXR/2x1-FORM A module **(Drwg.#2-101-1**) consists of two Form A single or double pole relays which switch the common connector between either A, B, or OFF. The A Channel is jumper selectable to be driven as channel 0, 2, 4 or 6 while the B channel may be driven from channels 1, 3, 5 or 7. The following table illustrates common switch module specifications:

**CXR/2x1-FORM A-1 and CXR/2x1-FORM A-2**
Bandpass: 400 MHz (-3dB)
Crosstalk: -60dB @ 5 MHz

The CXR/2x1 is available in single or double pole configurations with standard dry or mercury wetted reeds. The part number format for the CXR/2x1-FORM A Switch Module is:

**CXR/2x1-FORM A-*#PolesRelayType-Connector Type***
These switch modules are built with one of three different relay types, where:
***#Poles*-** Is either **1 or 2**.
***RelayType*-** Indicates Standard Dry (**-S**) or Mercury (**-M**).
***ConnectorType-*** Indicates BNC (**-N**), 3 Pin Headers (**-H**), Twin BNC (**-TBNC**) or SMA (**-SMA**).

<u>*Example:*</u> A CXR/2x1-FORM A-2S-N is built with double pole Standard Dry Reed Relays and BNC

connectors.

Modules built with single pole relays may have either BNC with grounded shields, SMA, or
3 Pin Headers.

Modules built with double pole relays may have either BNC with isolated shields, Twin BNC, or 3 Pin Headers.

## 4.5 CXR/2x1-C2

The CXR/2x1-C2 module, **Drwg#21-00-50**, has one Form C relay. The connector labeled NC is connected to the common when the module is in the un-energized state. When the module is latched, the connector labeled NO is connected to the common. The CXR/2x1-C2 is jumper selectable to be driven from one of eight channels allowing up to eight separate CXR/2x1-C2 modules to be driven from a single eight channel driver module. The CXR/2x1-C2 module is 50 Ohm and available with BNC or Twin-BNC or 3-pin header connectors.

## 4.6 CXR/8x1-1S, CXR/4x1-1S, CXR/8x1-1T and CXR/4x1-1T

The CXR/8x1-1S **(Drwg.#21-14-51)** and CXR/4x1-1S **(Drwg.#21-14-53)** Switch Modules function as single pole 8x1 and 4x1 multiplexers. These modules have Single Pole Form A Dry Reed Relays located at each input. Form A Relays typically have superior isolation when compared to their Form C counterparts. Connector options are limited to either 50 or 75 ohm BNC connectors.

The CXR/8-1T **(Drwg.#21-14-50)** and CXR/4-1T **(Drwg.#21-14-52)** function the same as the two switch modules mentioned above, however these switch modules have Form C relays on each input so that unused inputs are terminated through resistors. Connector options are limited to either 50 or 75 ohm BNC connectors. Both styles have isolation relays at the Common connector to improve bandpass. These switch modules may be built with either 50, 75 or 1K ohm terminating resistors. The part number format for the CXR/8x1-1T Switch Modules is:

**CXR/8x1-1T-*Termination Impedance***
***Termination Impedance-*** Indicates 50 ohms (**-50**), 75 ohms (**-75**) or 1000 ohms (**-1K**)

<u>*Example:*</u> The CXR/8x1-1T-50 is a CXR/8x1-1T built with 50 ohm terminating resistors.

*The part number format of the CXR/4x1-1T Switch Modules is the same as the CXR/8x1-1T Switch Modules.*

## 4.7 CXR/8x1-1HT, CXR/4x1-1HT, CXR/8x1-1HS and CXR/4x1-1HS

The CXR/8x1-1HT and CXR/4x1-1HT Switch Modules are built with eight (or four) single pole Form C relays on the inputs. In the unenergized state, the inputs are connected to terminating impedances of 50, 75 or 1K ohms. A second set of eight (or four) Form C relays make up the Isolation Relays, and in the unenergized state these relays connect the input Form C relay Normally Open contact to ground. This shorts any leakage from the input Form C relay to ground and gives complete isolation to the output. When the relays are energized, the terminating impedance is disconnected and the input is switched through to the isolation relay which is connected to the output. **Drwg. #21-24-50 (was #2-444-2)** shows the Switch Module in the un-energized state. Connector options are limited to either 50 or 75 ohm BNC connectors.

These switch modules may be built with either 50, 75 or 1K ohm terminating resistors. The part number format for the CXR/8x1-1T Switch Modules is:

**CXR/8x1-1HT-*Termination Impedance***
***Termination Impedance-*** Indicates 50 ohms (**-50**), 75 ohms (**-75**) or 1 kOhm (**-1K**)

*Example:* The CXR/8x1-1HT-50 is a CXR/8x1-1HT built with 50 ohm terminating resistors.

*The part number format for the CXR/4x1-1HT Switch Modules is the same as the CXR/8x1-1HT Switch Modules.*

The CXR/8x1-1HS and CXR/4x1-1HS Switch Modules are similar to the CXR/8x1-1HS and CXR/4x1-1HS counterparts. However, eight (or four) single pole Form A relays are placed on the inputs instead of Form C relays, In the un-energized state, the inputs are connected to ground and not a terminating resistor. Form A Relays typically have superior isolation compared to their Form C counterparts.

## 4.8 CXR/2x1 SWITCH MODULES

The CXR/2x1 Series Switch Modules (**Drwg.#30-25-50, 30-10-50 & 30-31-50**) are built with passive reed relays. Both single and two pole switch modules are available and they can be built with Type S Standard Dry. The modules are bidirectional, and input and output connectors can be swapped as needed. The CXR/2x1 jumper selectable to be driven from one of eight channels. Connectors are BNC, SMA or SMB, and F for the **30-31-10**.

**#30-25-10** is a 50 Ohm Module.
**#30-10-10** is a 75 Ohm Module.
**#30-31-10** is a 75 Ohm Module with F connectors.

Common specifications when used as individual switch modules are:

**CXR/2x1**

| | |
|---|---|
| Bandpass: | 400 MHz (-3dB) |
| Isolation: | -60dB @ 5 MHz |

The CXR/2x1-G Switch Module part number format is:

**CXR/2x1-*#PolesRelayType*-BNC**

| | |
|---|---|
| ***#Poles*** | Specifies One (**-1**) or Two (**-2**) Poles Switched |
| ***Connector Type-*** | Indicates BNC (**-N**), 3 Pin Headers (**-H**), Twin BNC (**-TBNC**) or SMA (**-SMA**). |

*Example:* The CXR/2x1-BNC is a two pole switch built with BNC connectors.

## 4.9 CXR/2x1-GT-50 and CXR/2x1-GT-75

The CXR/2x1-GT-50, **Drwg#21-01-50,** and the CXR/2x1-GT-75, **Drwg#30-30-50,** terminated switch modules have two Form C coaxial relays. The following table illustrates common switch module specifications:

**CXR/2x1-GT-50 (50 Ohm Modules)**

| | |
|---|---|
| Bandpass (with SMA, SMB or BNC connectors): | 2 GHz (-3dB) |
| Isolation (with SMA, SMB or BNC connectors): | -50dB @ 1GHz |

**CXR/2x1-GT-75 (75 Ohm Modules)**

| | |
|---|---|
| Bandpass (with SMB or BNC connectors): | 1.5 GHz (-3dB) |

Isolation (with SMB or BNC connectors): -50dB @ 1.6GHz

The connectors labeled A and B are terminated into 50 ohms or 75 ohms when the module is in the un-energized state. The CXR/2x1-G is jumper selectable to be driven from two of eight drive channels. The CXR/2x1-GT is jumper selectable to be driven from two of eight channels. Depending on the jumper settings, connector A is connected to COM when an even numbered drive is energized, and or connector B is connected to COM when an odd numbered drive is energized. Refer to **Drwg#21-01-50** for exact jumper configuration. This module is available with SMA or BNC connectors.

The CXR/2x1-GT Switch Modules part number format is:

**CXR/2x1-GT-*Relay Impedance-Connector Type***
***Impedance-*** (**-50**) indicates 50 ohms or (**-75**) indicates 75 ohms
***Connector Type-*** Indicates BNC (**-N**), SMA (**-SMA**) or SMB (**-SMB**).

<u>*Example:*</u> A CXR/2x1-GT-75-N has a characteristic impedance of 75 ohms and is built with BNC connectors.

## 4.10 CXR/4x2-G

The CXG/4x2-1G module is built with ten single pole, high frequency Form C relays arranged as a 4x2 tree switch as shown in **Drwg.#30-21-50 (50 Ohm Module)** and **Drwg#30-22-50 (75 Ohm Module).** The following table illustrates common specifications:

**CXR/4x2-G-50 (50 Ohm Module)**
Bandpass (with SMA connectors): 2.1 GHz (-3dB)
Isolation (with SMA connectors): -60dB @ 2 GHz

**CXR/4x2-G-75 (75 Ohm Module)**
Bandpass (with SMB or BNC connectors): 1 GHz (-3dB)
Isolation (with SMB or BNC connectors): -60dB @ 1 GHz

Any of four inputs can individually be connected to either of two outputs. Input #0 is connected to Output #0 when no channel is latched or when Channel 0,5, 6, or 7 is latched. Latching any other channel will break the Input #0 to Output #0 connection. The table below shows which input/output pair is selected when a specific drive is energized.

| Channel | Input | Output |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 1 | 0 |
| 2 | 2 | 0 |
| 3 | 3 | 0 |
| 4 | 0 | 1 |
| 5 | 1 | 1 |
| 6 | 2 | 1 |
| 7 | 3 | 1 |

No input can be simultaneously latched to both outputs; however, it is possible to close two separate parallel switch paths. This module is available with 50 or 75 ohm relays and SMA, SMB or BNC connectors.

The CXR/4x2-G Switch Modules part number format is:

**CXR/4x2-G-*Relay Impedance-Connector Type***

***Relay Impedance-*** (**-50**) indicates 50 ohms or (**-75**) indicates 75 ohms
***Connector Type-*** Indicates BNC (**-N**), SMA (**-SMA**) or SMB (**-SMB**).

Example: A CXR/4x2-G-50-SMA has a characteristic impedance of 50 ohms and is built with SMA connectors.

## 4.11 CXR/2(4x1)-1P-LL

The CXR/2(4x1)-1P-LL Switch Module, **Drwg. #21-13-50**, is intended for high insulation resistance measurements or the switching of low currents, down to Femtoamps. The module is built with shielded, low leakage relays, and all signal lines have Faraday Shields. The dual output Commons utilizes isolation relays to further minimize leakage. The two Commons also allow multiple switch modules to be interconnected to supply larger Nx1 multiplexers. Connector options are BNC (CXR/2(4x1)-1P-LL-BNC) or Triax (CXR/2(4x1)-1P-LL-TRIAX). Jumper positions provide for Grounded Guard, Isolated Guard or Driven Guard, as specified by the end user.

**APPLICATIONS:**

**High Speed Driven Guard Testing –** In this application, the outer conductor of the BNC or inner concentric conductor of the Triax connector are driven to the test voltage to reduce the charge time on the capacitance of the coaxial cable and thereby reduce leakage from signal to shield. For these applications, the relay EMI shield is jumpered to ground, the Faraday shield is connected to the outer BNC or inner concentric Triax conductor, and the outer conductor of the Triax is taken to chassis ground.
**If using BNC connectors in a driven guard application, it is critical to remember that there will be exposed voltages on the back of the chassis that could cause severe injury.**

**Low Noise –** Recommended connector is Triax. The inner conductor is used to switch signals. The inner concentric conductor of the Triax connector is isolated from both signal and CYTEC chassis ground. This isolated conductor should only be tied to ground at a single point (usually common to the signal source) and provides a shield from ground loop currents. The inner concentric is tied common across all of the switch module connectors. The outer concentric conductor of the triax is tied to chassis ground everywhere possible. This should be as low a noise and as high quality, a ground as possible. The switch module Faraday shield and relay EMI shield should all be jumpered to chassis ground.

**JUMPERING OPTIONS:**

**Relay EMI shield –** A foil shield protects the relay contacts from noise generated in the relay drive coil. In almost all applications this shield should be jumpered to chassis ground. Jumper holes are provided which would allow this shield to be jumpered to the Faraday shield should test data prove this to be a better option in rare cases.

**Relay commons –** Two relay commons are provided on each switch module. When left isolated, the module operates as two separate 4x1 multiplexers. When jumpered together, the module is a single 8x1 multiplexer with dual commons which allows, multiple module to be cabled together to form larger multiplexers.

**Faraday Shield –** A Faraday shield is provided on the switch module, mainly for driven guard applications, but it may be used for other situations where an isolated shield is beneficial. If the isolated guard from the BNC outer conductor or Triax inner concentric is left unconnected from the Faraday shield, the Faraday shield should be taken to chassis (earth) ground. There are two holes on the top of the Faraday module cover near the connectors. 3/4" long #4 screws may be screwed into these holes to connect the faraday shield to earth ground. A set of screws should have been provided with the chassis. The module will need to be removed from the chassis to install these screws.

## 4.12 CXAR SWITCH MODULE SPECIFICATIONS

**CXAR REED RELAY SWITCH MODULE SPECIFICATIONS**

| Switch Module | Connector Type | Switching Voltage | Switching Current | Switching Power | Life Expectancy | Operate Time | Contact Resistance |
|---|---|---|---|---|---|---|---|
| **CXR 8x1 and 4x1 SWITCH MODULES** | | | | | | | |
| CXR/4x1-1S | BNC | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0 ms | 150 mΩ |
| CXR/8x1-1S | BNC | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0 ms | 150 mΩ |
| CXR/8x1-1T-1K | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/8x1-1T-50 | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/8x1-1T-75 | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/4x1-1T-1K | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/4x1-1T-50 | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/4x1-1T-75 | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/4x1-1HS | BNC | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0 ms | 150 mΩ |
| CXR/8x1-1HS | BNC | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0 ms | 150 mΩ |
| CXR/8x1-1HT-1K | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/8x1-1HT-50 | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/8x1-1HT-75 | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/4x1-1HT-1K | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/4x1-1HT-50 | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/4x1-1HT-75 | BNC | 175 Vdc | 0.25 A | 5 W | 1 E8 Operations | 3.0 ms | 200 mΩ |
| CXR/8x1-2A-N | BNC | 110 Vdc, 125 Vac | 1.0 A | 30 W | 1 E8 Operations | 2.0 ms | 50 mΩ |
| CXR/8x1-2A-TBNC | TBNC | 110 Vdc, 125 Vac | 1.0 A | 30 W | 1 E8 Operations | 2.0 ms | 50 mΩ |
| **CXR-G SERIES SWITCH MODULES** | | | | | | | |
| CXR/8x1-G-50-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/8x1-G-50-SMA | SMA | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/8x1-G-50-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x1-G-50-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x1-G-50-SMA | SMA | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x1-G-50-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/8x1-G-75-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |

| Switch Module | Connector Type | Switching Voltage | Switching Current | Switching Power | Life Expectancy | Operate Time | Contact Resistance |
|---|---|---|---|---|---|---|---|
| CXR/4x1-G-75-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x1-GT-50-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x1-GT-50-SMA | SMA | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x1-GT-50-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/8x1-GT-75-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/8x1-GT-75-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x1-GT-75-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x1-GT-75-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x2-G-50-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x2-G-50-SMA | SMA | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x2-G-50-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x2-G-75-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/4x2-G-75-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |

**CXR 2x1 (A/B) SWITCH MODULES**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| CXR/2x1-FORM A-1S-N | BNC | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0ms | 200 mΩ |
| CXR/2x1-FORM A-1S-H | 3 Pin Header | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0ms | 200 mΩ |
| CXR/2x1-FORM A-1S-TBNC | SMA | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0ms | 200 mΩ |
| CXR/2x1-FORM A-1M-N | BNC | 500 Vdc | 1.0 A | 50 W | 1 E8 Operations | 2.0ms | 150 mΩ |
| CXR/2x1-FORM A-1M-H | 3 Pin Header | 500 Vdc | 1.0 A | 50 W | 1 E8 Operations | 2.0ms | 150 mΩ |
| CXR/2x1-FORM A-1M-TBNC | SMA | 500 Vdc | 1.0 A | 50 W | 1 E8 Operations | 2.0ms | 150 mΩ |
| CXR/2x1-FORM A-2S-N | BNC | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0 ms | 100 mΩ |
| CXR/2x1-FORM A-2S-H | 3 Pin Header | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0 ms | 100 mΩ |
| CXR/2x1-FORM A-2S-TBNC | TBNC | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0 ms | 100 mΩ |
| CXR/2x1-FORM A-2M-N | BNC | 500 Vdc | 1.0 A | 50 W | 1 E8 Operations | 2.0ms | 75 mΩ |
| CXR/2x1-FORM A-2M-H | 3 Pin Header | 500 Vdc | 1.0 A | 50 W | 1 E8 Operations | 2.0ms | 75 mΩ |
| CXR/2x1-FORM A-2M-TBNC | TBNC | 500 Vdc | 1.0 A | 50 W | 1 E8 Operations | 2.0ms | 75 mΩ |
| CXR/2x1-2C-N | BNC | 110 Vdc, 125 Vac | 1.0 A | 30 W | 1 E8 Operations | 2.0 ms | 50 mΩ |
| CXR/2x1-2C-TBNC | TBNC | 110 Vdc, 125 Vac | 1.0 A | 30 W | 1 E8 Operations | 2.0 ms | 50 mΩ |
| CXR/2x1-G-50-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/2x1-G-50-SMA | SMA | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/2x1-G-75-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/2x1-G-75-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |

| Switch Module | Connector Type | Switching Voltage | Switching Current | Switching Power | Life Expectancy | Operate Time | Contact Resistance |
|---|---|---|---|---|---|---|---|
| CXR/2x1-GT-50-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/2x1-GT-75-N | BNC | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| CXR/2x1-GT-75-SMB | SMB | 30 Vdc, 30 Vac | 0.5 A | 10 W @ 900 MHz | 1 E8 Operations | 10.0ms | 100 mΩ |
| **CXR LOW LEAKAGE SWITCH MODULES** | | | | | | | |
| CXR/8x1-1P-LL-N | BNC | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0ms | 100 mΩ |
| CXR/8x1-1P-LL-TRIAX | TRIAX | 200 Vdc | 0.5 A | 10 W | 1 E8 Operations | 1.0ms | 100 mΩ |

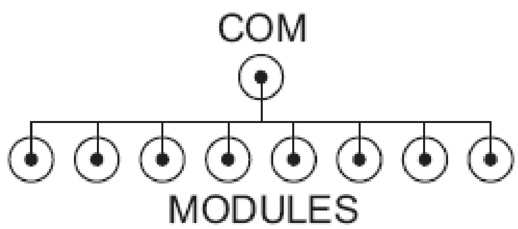
## 4.13 COAXIAL BUS STRIPS

The Coaxial Bus Strips are used to interconnect the Common connectors of CXAR Coaxial Switch Modules, thereby creating larger multiplexers. The Coaxial Bus Strips are designed for maximum bandpass and minimum stub length.

The bus strips are available in the following five sizes:

| | | |
|---|---|---|
| CXBS/2 | (2x1 Bus Bar) | **Drwg. #14-02-29** |
| CXBS/3 | (3x1 Bus Bar) | **Drwg. #14-03-29** |
| CXBS/4 | (4x1 Bus Bar) | **Drwg. #14-04-29** |
| CXBS/8 | (8x1 Bus Bar) | **Drwg. #14-05-29** |
| CXBS/16 | (16x1 Bus Bar) | **Drwg. #14-06-29** |

*CXBS/8 Bus Bar shown in Drawing Below*




**NOTE-** The Coaxial Bus Strips cannot be used with Type G Switch Modules.

## 4.14 CXS/8x1-GT-75-BNC, CXS/8x1-GT-75-F, and CXS/16x1-GT-75-F

The CXS/8x1-GT-75-BNC **(Drwg #64-00-50)** and the CXS/8x1-GT-75-F **(Drwg #64-01-50)** are built with 7 SPDT high frequency, solid state relays arranged as an 8x1 tree switch multiplexer. The CXS/16x1-GT-75-F **(Drwg #64-02-50)** are built with 15 SPDT high frequency, solid state relays arranged as an 16x1 tree switch multiplexer. Unlatched paths are terminated to 75 ohms internal to the relays. The characteristic impedance is 75 ohms. The CXS/16x1-GT-75-F module will not fit into a standard CXAR chassis, and requires a custom chassis. There is no normally closed path. The following switch module specifications are typical:

**CXS/8x1-GT-75-BNC (75 Ohm Modules with BNC connectors)**

| | |
|---|---|
| Isolation: | >50 dB to 1.25 GHz |
| Bandpass (with BNC connectors): | 1.0 GHz (-3 dB) |

**CXS/8x1-GT-75-F and CXS/16x1-GT-75-F (75 Ohm Modules with F connectors)**

| | |
|---|---|
| Isolation: | >50 dB to 1.25 GHz |
| Bandpass (with F connectors): | See plot below in Figure 1 |

Channels #0 through #7, (or #0 through #15), individually connect to the common when energized. Two (or more) channels can never be connected simultaneously. Diode logic on the modules ensures that the proper relays are selected for each channel.

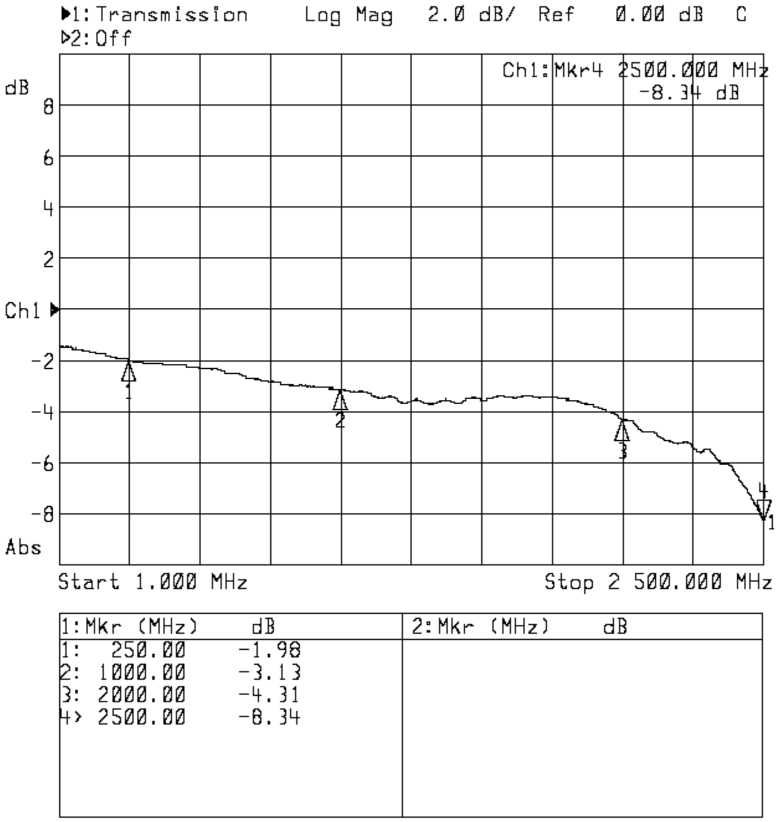



**Figure 1 – Insertion Loss 8x1-GT-**

# 5.0 IF-12 GPIB / RS232 / LAN CONTROL MODULE

## Introduction

CYTEC's IF-12 RS232/LAN Control Module is designed to control single chassis mainframes. Three forms of remote control are available on the module: IEEE488 (GPIB), RS232 and Ethernet LAN. An optional manual control is also available. All four interfaces may be active and used simultaneously.

Interface options: GPIB, RS232 and LAN are standard, but the Manual Control must be specified when purchasing the system. On some systems where panel space is limited, only two of the three interface connectors may be included.

### 5.1 LAN INTERFACE

**Dynamic IP Address (DHCP):** The Cytec IF12 is set at the factory to attempt to obtain an address from a DHCP server when the application boots. If you are connected to a network with a DHCP server, then the device IP address, network mask and gateway should be configured automatically. If your PC is on the same DHCP network, you will be able to communicate with the device after a short boot period of less than 10 seconds.

**Static IP Address:** If the module is plugged in to a network that does not have a DHCP server, you must provide a static IP address, network mask and gateway. These addresses should be provided by your network administrator.

**Auto IP Address:** The factory application contains an auto IP negotiation system. This allows the device to automatically configure its address in the absence of a central DHCP server, and without the need for a static IP address. This scheme is utilized as a fallback that will activate when both dynamic and static IP addresses fail to initialize. In order to communicate with a device in auto IP mode, the host system must support auto IP. Auto IP support is included in both Windows and OS X operating systems. By default, auto IP addressing starts in the 169.XXX.XXX.XXX address range.

**Find Your Device:** Our recommended option to locate the device is to use a local discover utility. You can do this by navigating to the Cytec web site and downloading the tool localdiscover.exe from https://cytec-ate.com/discover-cytec-local. The executable sends out a request to all Cytec devices on the local network. It opens a browser page on the first device to respond that lists all of the discovered devices, or a page that show that no devices were found.

Note: If these options are failing, there may be a firewall issue blocking the applications from sending the UDP broadcast that is used to locate Cytec devices. Always grant Cytec applications the ability to get through your OS firewall and ensure that UDP port 20034 is open for use.

### 5.2 RS232 INTERFACE

**Signal Connections**
The control module is pre-configured at the factory to operate as Data Communications Equipment (DCE) per

the EIA RS232D Standard.  In this configuration, the module transmits on the RxD Pin and receives on the TxD Pin.  RTS is required to be high for the control module to transmit and CTS is output high by the control module to indicate a ready for data state and low when busy.  The RS232 rear panel connector is a D9P (male) and can be run directly from a D9 computer COM port with a straight through (one to one) D9S to D9S cable.  A null modem cable will not work with the factory default settings!  Adaptors are available at any computer store to convert from D25 to D9.  Do not use any adaptor that also acts as a null modem converter.  If you are building your own cables, consult CYTEC Corp., for D25 to D9 pin out conversion.

**D9P (male) PIN OUTS**

| Pin | Signal | Function |
|---|---|---|
| 1 | DCD | Not Used. |
| 2 | RxD | Data out of Control Module. |
| 3 | TxD | Data in to Control Module. |
| 4 | DTR | Not Used |
| 5 | Common | Signal Ground. |
| 6 | DSR | Not Used |
| 7 | RTS | Control Module requires + V to transmit. |
| 8 | CTS | Control Module provides +V when ready |
| 9 | RI | Not Used |

**Find Your Device**

Open Device Manager on Windows computers and navigate to Ports. The COM port number will be bracketed next to the device description.

**Configure Your Device**

The RS232 interface can be accessed using any standard terminal emulation program such as PuTTY which can be downloaded from putty.org. Enter the COM port number in the field for the Serial line to connect to. The default values set at the factory are:

- Speed (baud): 9600
- Data bits: 8
- Stop bits: 1
- Parity: None
- Flow control: RTS/CTS (Hardware)

The first thing you should do is turn on Echo.  This will enable you to see what you are typing.  Make sure you turn Echo back off when you are done with the terminal session.  Echo being left on will normally interfere with programs written specifically to control the switch.

**Echo**

Echos the characters back to your screen while you type them so you can see what you type.

| | | |
|---|---|---|
| Command: | "E 0 73" | Turns Echo Off |
| | "E 1 73" | Turns Echo On |

**Answerback**

Answerback allows the Control Module to return information to the COM port.  Answerback should almost always be left on. If Answerback is enabled, the Answerback byte **must** be read back by the requesting device. Failure to do so could have unpredictable results.

| | | |
|---|---|---|
| Command: | “A 0 73” | Turns Answerback Off |
| | “A 1 73” | Turns Answerback On |

**Verbose**

Verbose causes the system to return more specific information when you request status or read answerback characters. It is sometimes helpful when troubleshooting but it slows the interface down a lot. While there may occasionally be a good reason to turn on Verbose during a puTTY or Hyperterm session, it is almost never used in a programmatic interface. All of the same information can be generated in code based on the non-verbose responses without slowing down the RS232 interface.

| | | |
|---|---|---|
| Command: | “V 0 73” | Turns Verbose Off |
| | “V 1 73” | Turns Verbose On |

**Baud Rate**

Baud rate is set at the factory at 9600 Baud. Change is under software control and the control module must be connected to a serial interface to effect the change.

| Baud | Baud# n |
|---|---|
| 2400 | 4 |
| 4800 | 5 |
| 9600 | 6 |
| 19200 | 7 |
| 38400 | 8 |
| 57600 | 9 |
| 115200 | 10 |
| 230400 | 11 (untested, you should consider LAN) |
| 460800 | 12 (untested, you should consider LAN) |

| | | |
|---|---|---|
| Command: | "P19 n 73" | |
| | "P19 7 73" | sets baud rate to 19200. |

If the Baud rate is inadvertently set to an unknown rate, the default value may be restored. See the section on Setting Defaults for the procedure.

Obviously as soon as you reset the Cytec baud rate you will no longer be able to communicate with the switch until you reset the baud rate on your controlling computer or communication device.

**CTS/RTS Handshake**

The Clear to Send (CTS) and Request To Send (RTS) hardware handshaking functions may be modified by the 'P6' command.

Command: "P6 handshake 73"

| | |
|---|---|
| handshake = 0 | Handshaking off |
| handshake = 1 | Handshaking on (default) |

Example

| | |
|---|---|
| "P6 0 73" | Turn handshaking off. |

## 5.3 IEEE488 INTERFACE

Also known as GPIB (General Purpose Interface Bus), IEEE-488 is the international standard for a parallel interface used for attaching sensors and programmable instruments to a computer. When connecting IEEE-488 cables, some rules apply. The total number of devices should be 15 or less. The total length of all cables should not exceed 2 meters multiplied by the number of connected devices, up to a maximum of 20 meters. And no more than three connectors should be stacked together.

**Find Your Device**

Our recommended option to locate the device is to use NI Measurement & Automation Explorer (NI MAX), which can be downloaded from their website. Search for instruments in the application and the Cytec device should be found at default GPIB address 7.

**Configure Your Device**

**GPIB Address:**

Command syntax: “P14 n 73”.

For example, “P14 8 73” sets the GPIB address to 8.

### 5.3.1 IEEE488.2 SPECIFIC MATRIX COMMANDS

These commands are ignored by the RS232 interface.

***IDN? - Revision Number** (Same as Cytec “N” - Revision Command)

Syntax: “*IDN?”

The 'IDN?' command will cause the matrix to return its current revision number followed by an end of line.

| | | |
|---|---|---|
| Send: | "*idn?" | Request revision number. |
| Receive: | "Cytec VDX/32x32 11-01-13 1.0" eol | Text string indicating rev. |

***RST - Reset** (same as C - Clear command)

The '*RST' command will clear (open all switches) in the matrix.

| | | |
|---|---|---|
| Send: | "*rst" | Reset. |
| Receive: | "0" eol | Returns ‘0'. |

## 5.4 CONFIGURING TCP/IP PARAMETERS FROM A SERIAL CONNECTION

To change parameters you will need to access the serial interface using any standard terminal emulation program from the COM port on your computer. Once you have established a serial connection the following commands can be used for configuration:

**D** command returns a list of current settings:

A1, E1, V0   Answerback = ON, Echo = ON, Verbose = OFF
Baudnumber = 6, RS Handshaking = 1
IP Address = 10.0.0.144
Netmask = 255.255.255.0
Gateway = 0.0.0.0
Port0 = 8080, Port1 = 8081
TCP idle = 60
Telnetlock = 0, Telnet Echo = 0
Battery Ram = 0, Default List = 0

**IFConfig** command is used to set the static IP address. The syntax for this command is:

ifconfig aaa.aaa.aaa.aaa nnn.nnn.nnn.nnn

a = ip address in dotted decimal format n = subnet mask in dotted decimal format

Example: ifconfig 10.0.0.100 255.0.0.0

Typing ifconfig and hitting the enter key will return the current settings.

Since you may be connected via Telnet to do this, **the IP address will not actually change until you reboot the Cytec switch**. This helps prevent anyone from mistakenly setting the IP to an unknown address by accident. It is a good idea to double check the settings with the D command before you reboot.

**HOSTS** command sets the gateway for TCP/IP sockets. The syntax for this command is:

HOSTS xxx.xxx.xxx.xxx

Example: hosts 10.0.0.100

Typing hosts and hitting the enter key will return the current settings.

**SNET TCP PORT** command sets the Port number for TCP/IP sockets. The syntax for this command is:

SNET TCP PORT n m where n = equals one of two sockets and m is the port number

Example:

snet tcp port 0 8088 socket 0 is port #8088
snet tcp port 1 8089 socket 1 is port #8089

Port numbers must be between 1024 and 65535.
The Telnet port (23) may also be available. See TELNETLOCK command.

**SNET TCP Idle** command sets the socket life for the connection. The syntax for this command is:

SNET TCP Idle n (n=seconds) (1 to 3600 sec)
Default = 60 sec
SNET TCP Idle (display)
TCP Idle = 60
SNET TCP Idle 0 Socket never dies until the computer that established the socket kills it.

Setting the TCP Idle to 0 will force the socket to stay alive until the program that established the socket kills it.

**WARNING:** This can lead to issues if there is a network disconnect or the computer that established the socket locks up. If the computer that establishes the socket cannot kill the socket, no one will be able to connect to the switch until the Cytec unit is rebooted.

**TCPAnswerback – Answerback**

Syntax: TCPANSWERBACK n n = 0, 1 or 2

Answerback will enable or disable the transmission of a single character followed by an end of line upon the completion of all commands. The Answerback character will be a 1 or 0 depending on what command is sent. It is used to verify that the command was accepted and can verify completion of relay control commands. See **Section 5.5.2**

| | | |
|---|---|---|
| Eg. | "TCPANSWERBACK 0 " | Turn answerback off. |
| | "TCPANSWERBACK 1 " | Turn answerback on |
| | "TCPANSWERBACK 2 " | Turn answerback plus terminator on |

Note: TCPANSWERBACK 2:
This setting appends a set of square brackets to the answerback byte.

| | | | |
|---|---|---|---|
| Eg. | Send: | "L0 0" | Latch Module 0 Switch 0. |
| | Receive: | "1[]" | End of line follows the terminator |

## 5.5 RUNNING THE CYTEC FACTORY APPLICATION ON THE IF12

The factory application that is included with the IF12 Control module includes:

- System Parameter Settings
- Matrix Parameter Settings
- Remote Switch Control
- List/Config Management
- File Management
- Custom Labeling

The URL request in the browser should look like the following:

http://<Device IP>

Where <Device IP> is replaced with the corresponding IP address. For more information on finding the IP address of your device, please see the device discovery section of this manual.

## 5.6 COMMAND FORMAT/COMPLETION

### COMMAND FORMAT

All commands consist of at least one ASCII character indicating the command followed by optional values. After the command string is sent, an End of Line Character must be sent to affect the command.

If values are included with the command, the first value does not need to be separated from the command; all subsequent values MUST be separated by spaces or commas, eg. L1 2.

Multiple commands may also be sent on one line. Commands must be separated by a semi-colon character. Command line length is limited to 19 characters so avoid abusing this feature.

| | | |
|---|---|---|
| Examples: | "L2 7;C" | Connects Input 2 to Output 7 then clear |
| | "U4 7;L 1 2" | Unlatch Mod 4, Sw 7 then Latch Mod 1, Sw 2 |

### COMMAND COMPLETION

A code representing the last requested switch point status (open or closed) and command completion will be stored by the matrix.

If the LAN or RS232 answerback function is enabled, a single character followed by end of line will be sent upon completion of all commands. Answerback may also include a termination character.

**Note: Command Completion is NOT updated until the matrix finishes the requested operation.**

**Command Completion Codes – See Section 5.8 for error and completion codes**

### 5.6.1 END OF LINE CHARACTER ( EOL )

A received end of line character will cause the control module to execute the ASCII command string. The end of line character may be sent as a carriage return (CR) or New Line / Line Feed (NL/LF) for RS232 interfaces and a New Line / Line Feed (NL/LF) for IEEE488 interfaces or LAN interfaces. The IEEE488 also allows for the END control line being true with the last data character to initiate the command.

Valid end of Lines:

| | |
|---|---|
| CR, LF or NL | LAN, RS232 or IEEE488 |
| CR and END | IEEE488 |
| LF/NL and END | IEEE488 |

Note that the terms New Line and Line Feed are often used to mean the same thing.
Both are expressed as \n in most programming languages and are shown on the ASCII table as “LF”.

LF = Line Feed / New Line represented as \n, on ASCII table it is Decimal 10, or Hex A (0xA).
CR = Carriage Return represented as \r, on ASCII table it is = Decimal 13 or Hex D ( 0xD).

**When any data is returned from the switch, the data will also be followed by an End Of Line character (EOL).**

**Notes - All Interfaces**: Upon requesting status output characters MUST be received by the requesting device. Failure to do this will prevent further use of the matrix.

**Access Code**
Some commands require an access code number to be included with the command. This code prevents inadvertent operation of system modifying commands. The access code is 73.

## 5.7 SETUP COMMANDS

**Matrixsize** command sets the matrix size. The syntax for this command is:

matrixsize mtx# #mods #rlys

mtx#: For mainframes this is 0, for MESA II expansion systems this is the matrix number for the expansion chassis.

#mods: The maximum number of modules for the chassis.

#rlys: The maximum number of relays per module.

Example: matrixsize 0 16 8 Sets the # of modules to 16 and the maximum number of relays per module to 8 for a mainframe chassis (mtx is 0).

Typing matrixsize and hitting the enter key will return the current settings and chassis type (for MESA II systems all of the expansion chassis settings will be returned as a list).

**P Commands** (Except for communications settings these are set at the factory to the correct value for your system and should not need to be altered)

- P0 n 73 Set maximum number of matrices to ‘n’. For Mainframes n = 1, for MESA II Control n = number of expansion chassis
- P6 n 73 n can be 1 or 0. 1 turns RTS/CTS handshaking on, 0 turns RTS/CTS handshaking off. This setting only applies to serial communication.
- P7 n 73 n can be 1 or 0. 1 turns Use RAM on, 0 turns Use RAM off.
- P8 n 73 n can be 0 to 6. Sets the default list (configuration) to load at power up if Use RAM is on.

- P10 n 73   Set maximum number of modules to 'n'  for a Mainframe system. For MESA II Systems sets the maximum number of modules for  Matrix 0;

- P11 n 73   Set maximum number of modules in Matrix 1 of a MESA II System to 'n'.

- P12 n 73   Set maximum number of modules in Matrix 2 of a MESA II System to 'n'.

- P13 n 73   Set maximum number of modules in Matrix 3 of a MESA II System to 'n'.

- P14 n 73  Set GPIB address to 'n'.  n can be 0 to 31.

- P19 n 73  Set the baud number to ' n'.  See RS232 configuration section for corresponding baud rate to baud number.

- P20 n 73   Set maximum number of relays to 'n'  for a Mainframe system. For MESA II Systems sets the maximum number of relays for  Matrix 0;

- P21 n 73  Set maximum number of relays in Matrix 1 of a MESA II System to 'n'.

- P22 n 73  Set maximum number of relays in Matrix 2 of a MESA II System to 'n'.

- P23 n 73  Set maximum number of relays in Matrix 3 of a MESA II System to 'n'.

- P90 n 73  Set the system ID number to 'n'. Used in large systems to differentiate between chassis.

## 5.8 SWITCH COMMANDS

### General Notes

For LAN and RS232, after sending any command the Cytec control will return an integer Answerback character if Answerback (TCPAnswerback for LAN) is ON.  Answerback/TCPAnswerback is turned on by default and is Cytec's preferred operation since it allows you to verify commands are accepted before continuing.

If the command was a switch operation command such as Latch ( L ) or Unlatch ( U ), the character will be a meaningful status response where 1 = switch latched and 0 = switch unlatched.  This may be used to verify that the command was received correctly.

Any other commands sent will also generate an answerback character which may be either a 1 or 0 and either character will indicate the command was received but the value is meaningless so either is acceptable.

Answerback may be turned off when using LAN or RS232 although it is not recommended.
Answerback can also include a termination character for the LAN or RS232 interface.

### Error Characters

If a command is sent incorrectly, an error character will be generated and added to the answerback character.

Since the answerback character may be a 1 or 0, there may be two values for error characters as described below.

Answerback returned:

| Dec | Hex | |
|---|---|---|
| 1 | 30 | Latch completed without errors. |
| 0 | 31 | Unlatch completed without errors. |
| 2 or 3 | 32 or 33 | Unknown command, first character unrecognizable. |
| 4 or 5 | 34 or 35 | Incorrect entries, number or type of entries incorrect. |
| 6 or 7 | 36 or 37 | Entries out of limits, switch point out of usable range. |
| 8 or 9 | 38 or 39 | Invalid access code, number 73 not included when required. |

**Delays to Prevent Errors**

It is important to recognize that with modern computers and control interfaces, it is possible to stream commands to the switch matrix faster than the relays can physically operate. Many electro-mechanical relays may take between 2 to 20 ms to close or open. This can result in unpredictable results if certain operations are streamed together without considering this delay.

A good example of this type of problem occurs if a Latch command is sent and is immediately followed by a status request. Many of Cytec's products actually base status on current flow through the relay drives so it is possible to send a command and request status before the relay has physically operated, resulting in incorrect status feedback.

Typically, a 5 to 20 ms delay between commands requiring feedback can ensure that this is never an issue.

**L,U,X – Latch, Unlatch, Multiplex Commands**

Syntax: Cmd Switch
Cmd Module, Switch
Cmd Matrix, Module, Switch
The specified switchpoint is operated on. Note: For mainframe systems the matrix number will be 0.

(Cmd = 'L', 'U' or 'X')
**L = Latch** = Turn switch ON Closes the specified point, all others unaffected.
**U = Unlatch** = Turn switch OFF Opens the specified point, all others are unaffected.
**X = Multiplex** = Clear + Latch Opens all points, then Latches the specified point.

E.g. "U0 2 3" Matrix 0, Module 2, Switch 3 is opened. (OFF)
"L0 1 3" Matrix 0, Module 1, Switch 3 is closed. (ON)
"L0 1" Module 0, Switch 1 is closed. (ON)
"L2 3 7" Matrix 2, Module 3, Switch 7 is closed. (ON)
"X0 3 0" Clear all switch points ( turn them all OFF ) then Latch Matrix 0, Module 3, Switch 0.

If a single integer value is sent, the control module assumes it is a switch value and defaults to the last module value sent. If two integers are sent, the control module assumes they are a module and switch value and defaults to the last matrix value sent.

E.g. "L3 2 3" Matrix 3, Module 2, Switch 3 is closed (ON). Then, "L1 4" Assumes Matrix 3. Matrix 3, Module 1, Switch 4 is closed (ON). Then,"L5" Assumes Matrix 3, Module 1. Matrix 3, Module 1, Switch 5 is closed (ON).

Some Cytec programming examples may refer to Mod #, Rly # (Relay #). The terms Switch (Sw) and Relay (Rly) mean the same thing. For Unidirectional matrix switches the Module # may be thought of as Input #, and the Switch or Relay # may be thought of as the Output #.

**Multiplex Modes**

Multiplex settings can be configured by accessing the device web page at the IP Address assigned and navigating to the Set System Parameters page.

| | SINGLE 32 (default) | QUAD 8 | DUAL 16 | MATRIX |
|---|---|---|---|---|
| MATRIX SIZE | 1x32 or 4x8(default) | 4x8 | 2x16 | Any |
| ALLOWED CMD | L0 0 0 to L0 0 31 or L0 0 0 to L0 3 7 | L0 0 0 to L0 3 7 | L0 0 0 to L0 1 15 | Configuration Dependent |
| OR | L0 to L31 | | | |
| MUX MODE (X Command) | One relay in the system can be latched at a time | One relay per module can be latched at a time | One relay per module can be latched at a time | One relay in the system can be latched at a time |

Manual Control:

If MODE is set to Matrix on the front panel the user will be able to latch all of the 32 switchpoints at the same time. If MODE is set to Multiplex on the front panel the user will be able to latch one of the 32 switchpoints at a time in SINGLE 32 mode or one switchpoint per module in QUAD 8 or DUAL 16 mode.

Remote Control:

Commands sent remotely will not have the same limitations as the front panel. The coder is able to latch all 32 switchpoints using the Latch command "L". The Multiplex Modes (SINGLE 32, QUAD 8 and DUAL 16) only affect the Multiplex command "X", as detailed in the examples below.

Example 1:

Configuration QUAD 8: The following commands are sent in the respective order

| **Command Sent** | **Result** |
|---|---|
| L0 0 0 | Relay 0 will be latched |
| L0 0 1 | Relay 1 will be latched |
| X0 3 2 | Relay 26 will be latched |
| X0 0 2 | Relays 0 and 1 will be unlatched and relay 2 will be latched Relay 26 will remain latched |

Example 2:

Configuration DUAL 16: The following commands are sent in the respective order

| **Command Sent** | **Result** |
|---|---|
| L0 0 0 | Relay 0 will be latched |
| L0 0 1 | Relay 1 will be latched |

| | |
|---|---|
| X0 1 14 | Relay 30 will be latched |
| X0 0 2 | Relays 0  and 1 will be unlatched and relay 2 will be latched Relay 30 will remain latched |

Example 3:

Configuration SINGLE 32 (1X32 MTX SIZE): The following commands are sent in the respective order

| **Command Sent** | **Result** |
|---|---|
| L0 0 0 | Relay 0 will be latched |
| L0 0 1 | Relay 1 will be latched |
| X0 0 26 | Relays 0  and 1 will be unlatched and relay 26 will be latched |

To recap: The latch command "L" will always latch the relay requested if it is in the range of allowable relays. The multiplex command "X" will unlatch all of the other relays if the Mux Mode is set to SINGLE 32. If Mux Mode is set to QUAD 8 or DUAL 16 the multiplex command "X" will unlatch all of the relays on the same module as the requested relay.

**C - Clear Command**

Syntax: C
All points in the chassis are opened.

E.g. "C" All switches in the chassis are opened.

For IEEE488.2, The C command is the same as the *RST (reset) function.

## 5.9 STATUS AND INTERROGATE COMMANDS

**The Status and Interrogate commands return information to the user so they can determine what state each switch point is in before proceeding. The commands can be used to simply check the switch configuration, to verify connections, or to prevent unwanted connections.**

**The information returned by these commands can be different depending on what type of system you have. Please find the Status or Interrogate section for your specific system before writing code that is dependent on the returned values.**

**S - Status Command**

Syntax: S Returns Status of entire mainframe chassis.
S0 Module# Returns Status of specified Module#.
S0 Module# Switch# Returns Status of specified Switch point.

Status may be requested of a single switch point or for the entire chassis. After receipt of the Status command the Matrix will return a character or string of characters representing the status, open or closed, of a switch point or switch points. A one, '1', signifies a closed switch point (ON) and a zero, '0', an open switch point (OFF).

In the case of a single switch point Status a single character is returned followed by an end of line.

For multiple switch points, a stream of 1's and 0's will be returned. How they are returned and what they mean is dependent on the type of system.

Status return is broken down into four different groups:

- **Systems with 8 to 32 relays drives possible.**
- **Systems with 64 to 512 relay drives possible.**
- **RJV systems which are unique.**
- **Large Matrix systems with over 512 switch points or solid state matrix systems.**

**Please see the sections on the following pages for the information for your particular system**

**Status Returned Values up to 32 drives**

For systems with 8 to 32 relay drives (any model designated as xxx/16 or xxx/32 )
These Models Include:

CXAR/16 and /32
CXM/16 and /32
HXV/16 and HXV/32
RS/8 or /16
GX/4 or /8 or /16 or /32
FO/16 or /32

Model numbers are shown on the rear panel label. If the model no. is simply an integer such as "6754" you should reference the addendum for the system or contact Cytec with questions if you do not know what it is returning.

The S command sent by itself will return a continuous string of 16 or 32 1's and 0's followed by the answerback character (if on) and the end of line character. The string is ordered from lowest value switch to highest value switch.

E.g. 1 CXAR/32 chassis
Send: "S"
Receive: "00010001000000000000110000000010" eol 32 switch points + Answerback + EOL

This is interpreted as Switches 3, 7, 20, 21 and 30 are ON, the rest are OFF.
In Module#, Switch # syntax this is:

Mod 0, Switches 3 and 7 are ON.
Mod 1, no switches are ON.
Mod 2, Switches 3 and 4 are ON.
Mod 3, Switch 7 is ON.
All other switches are OFF.

If the Answerback function is on, the last 1 or 0 before the EOL will be the Answerback character and the value is a "don't care". If Answerback + termination character is ON (TCPAnswerback set to 2 for LAN communications), the termination characters [] will be the last characters returned – this is in order to enable a coder to read the entire response with one chunk of code, (read until char returned equals ']').

**Status Returned Values 64 to 512 drives**

For systems with 64 to 512 relay drives
(any model designated as xxx/64, xxx/128, xxx/256, xxx/512 )
These Models Include:

LXA or LXB ( all )
CXAR/64, /128 or /256
CXB systems ( all )
CXG, CXF, CXS systems ( all )
CXM/64, /128 or /256
HXV/96 or /128, or /256
RSM ( all )
RJM ( all )
JX ( all )
VX ( all )
PX ( all )
GX/64 or /128
FO/64 or /128

Model numbers are shown on the rear panel label. If the model no.is simply an integer such as "6754" you should see the addendum for the system or contact Cytec with questions if you do not know what it is returning.

The S command sent by itself will return a row / column pattern of 1's and 0's that mimic the LED display on the front panel (if equipped ), where the columns are the Module # and the Rows are the Switch #.

Eg. 2 A 16 Module, 8 Switch Matrix ( your configuration may be different):

| | | |
|---|---|---|
| Send: | "S" | Status of chassis |
| Receive: | "0001000100000000" eol | Switch 0, Module 3 and 7 closed |
| | "0000000000000000" eol | Switch 1, none closed |
| | "1111111111111111" eol | Switch 2, all closed |
| | "1000000000000001" eol | Switch 3, Module 0 and 15 closed |
| | "1010101010101010" eol | Switch 4, odd Modules closed |
| | "0101010101010101" eol | Switch 5, even Modules closed |
| | "0110000000000000" eol | Switch 6, Module 1 and 2 closed |
| | "0000000000000110" eol | Switch 7, Module 13 and 14 closed |
| | "0"eol | Answerback character |

If the Answerback function is on, the last 1 or 0 before the EOL will be the Answerback character and the value is a "don't care". For LAN Communication, if Answerback + termination character is ON (TCPAnswerback set to 2), the termination characters [] will be the last characters returned – this is in order to enable a coder to read the entire response with one chunk of code, (read until char returned equals ']').

**Status Returned Values RJV Systems**

RJV systems are unique and the returned status is specific to those systems.
Skip this section if you do not have an RJV/48 or RJV/144 system.

S - Status Command

Syntax: S
Module, Switch

Status may be requested of a single switch point or for the entire chassis. After receipt of the Status command the Matrix will return a character or string of characters representing the status, open or closed, of a switch point or switch points. A one, '1', signifies a closed switch point and a zero, '0', an open switch point.

In the case of a single switch point a single character is returned followed by an end of line.
For multiple switch points a line of ones and zeros will be returned for each switch module in the system. For the RJV/48 there will be four rows of 24 1's or 0's. For the RJV/144 there will be 12 rows of 24 1's or 0's. Each line returned will be followed by an end of line (EOL). For LAN and RS232 interfaces there may also be an Answerback character before the EOL if Answerback is enabled.
At the end of all output rows an end of line will be sent for both *interfaces.* See the command completion and answerback sections for details.

Eg. 1

| | | |
|---|---|---|
| Send: | "S 3,4" | Test Module 3, Switch 4 |
| Receive: | "0" eol | Switchpoint open, end of line. |

Eg. 2 An RJV/48 with four 6x4 Modules:

Send: "S" Status of chassis
Receive:

| | |
|---|---|
| "000100010000000000000000" eol | Module 0, Switches 3 and 7 closed |
| "000000000000000000000000" eol | Module 1, none closed |
| "100000010000000100000001" eol | Module 2, Switches 0, 7, 15 and 23 closed. |
| "000000100000000000100000" eol | Module 3, Switches 6 and 18 closed. |

An RJV/144 will return 12 rows with 24 1's or 0's in each row.

**RJV Module Status table:**
Use this table to relate switch #'s to Input and Output Ports.

**RJV/12x1-X-XX** (any 12x1 module)
These modules only use the first 12 characters of the returned string of 24 1's or 0's.
The second set of 12 characters are "don't cares" and may be returned as 1's or 0's.
They should be ignored.

| Ports 0 to 11 | NA |
|---|---|
| 001000000000 | 000000000000 |

Indicates port 2 is on.

**RJV/6x4-X-XX** (all 6x4 modules)

These modules use all 24 characters and each common port has six associated 1's or 0's:
0 to 23 = Switch #'s

| Port Labels | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| C0 | 0 | 4 | 8 | 12 | 16 | 20 |
| C1 | 1 | 5 | 9 | 13 | 17 | 21 |
| C2 | 2 | 6 | 10 | 14 | 18 | 22 |
| C3 | 3 | 7 | 11 | 15 | 19 | 23 |

So the 24 characters line up as:
C0/0 C1/0 C2/0 C3/0 C0/1 C1/1 C2/1 C3/1 C0/2 C1/2 C2/2 C3/2 C0/3 C1/3 C2/3 C3/3 etc...

**Status Returned Values Large Systems**

For systems with more than relay drives, large matrices, or solid state switches
These Models Include:

VDX/16x16 or /32x32 ( all )
VDM/32x32 (all)
DX/64x64 or /256x256 ( all )
DXM/64x64 or /128x128
FX ( all )
TX ( all )

These systems do not make use of the S command but instead use the I command to return information.

Some of these systems would return up to 65,536 characters if they accepted the S command which becomes very difficult to deal with. Instead, they use the "I" command which only returns the address of switch points which are on. This limits the amount of returned data to a maximum of 256 addresses which is much easier and quicker to deal with.

These systems will interpret any "S" command sent as an "I" command and respond accordingly.

**See the Interrogate Command Section for details.**

## 5.10 I – INTERROGATE COMMAND

Syntax: I

The Interrogate function will return a list of all closed (ON) switch points. Each switch point will be followed by an "end of line" (EOL). The switch point is listed as the Module# and then Switch#. For matrix applications such as a 16x16 this often translates into "Input # then Output #" or "Output # then Input #". Since many systems are bi-directional Input vs Output may be dependent on how you are using it.

| | | | |
|---|---|---|---|
| Eg. | Send | "I" | |
| | Receive | "0, 0" eol | Module 0, Switch 0 Closed. End of line. |
| | | "1, 6" eol | Module 1, Switch 6 Closed. End of line. |
| | | "3, 2" eol | Module 3, Switch 2 Closed. End of line. |
| | | "0" eol | Answerback character (if enabled). End of line. |

For system such as a DX/256x256 the “I” command may return up to 256 addresses. Be sure your buffer size can handle the amount of returned data.

**For Unidirectional matrix switches, specifically DX, DXM, VDX, VDM and TX, the Module # may be thought of as Input #, and the Switch or Relay # may be thought of as the Output #.**

.

## 5.11 OTHER COMMANDS

**F - Front Panel**
Syntax: F n 73 n = 0 or 1

Front panel lock-out will be initiated by the receipt of a 0 character and enabled by the receipt of a 1 character followed by the access code. The access code prevents inadvertent lock-out from occurring. Lock-out will prevent any operation of the system from the front panel until it is terminated from the remote (F 1) or power is turned off then on. Preset to panel enabled at power on.

| | | |
|---|---|---|
| Eg. | "F 0,73" | Lock-out local operation. |
| | "F 1 73" | Enable local operation. |

P - Program
Syntax: P n1,n2,73

The program command allows the operator to setup matrix dependent variables. These include matrix switch configuration and certain interface functions. Use of the P commands is complicated and varies greatly between systems. Your system should have been provided with the correct P command set-up.

If you need to change the matrix configuration, number of allowed modules, or other obscure set-up configurations on your system we recommend you contact Cytec and we can walk you through the P commands needed for your specific system. Please provide the serial # of your system when you contact us.

**N - Revision Number (Same as IEEE488 *IDN? )**

Syntax: N

The 'N' command will cause the matrix to return its current revision number followed by an integer identifier, followed by an end of line.

| | | | |
|---|---|---|---|
| Eg. | Send: | "N" | Request revision Number |
| | Receive: | "Cytec 11-14-10-1, 1.23 0" eol | Text string indicating revision. |

Where: “Cytec” = manufacturer.
“11-14-10-1" = control module board number.
“1.23" = Firmware Revision # (example).
“0" = Integer identifier.

Note: When requesting the Revision number, all characters must be received before the system can be resumed.

**NOTE - The text string received from the 'N' Command will vary depending on the type of system.*

**Integer identifier**

The N command now includes a single byte which can be used as an identifier for Cytec systems. The identifier is a single byte integer so it may 0 to 255. WE do not assign this and it has no meaningful relationship to any product. It is simply a number which may be assigned to a chassis so that the end user can acknowledge that a specific Cytec chassis is communicating. It is up to the customer to assign the number and keep track of it. It allows them to poll multiple chassis and know that the one they are talking to is, for example, the JX/256 that they assigned the identifier “13" to.

Command to enter or change the number:

P90 n 73 where n is the number from 0 to 255

## 5.12 INPUT / OUTPUT vs MODULE / SWITCH NOMENCLATURE

Most of the switching systems sold by Cytec are completely bi-directional and can be used in a variety of ways by the customer so it is impossible for us to use the terms Input and Output, even though it is what probably makes the most sense to the end user when connecting signals to the switch.

We label and control the switches using Module# and Switch# to avoid this confusion since for most systems either can be considered an Input or an Output.

However some Cytec systems are uni-directional and therefore do have assigned Inputs and Outputs:

Uni-directional Systems with defined Input and Output ports:

**DX and DXM high speed digital switch matrix systems**
**VDX and VDM analog or digital switch matrix systems**
**TX analog systems**
**FX fiber optic systems**

**For all of these systems:**

**Module # = Input**
**Switch # = Output**

Note that these systems will all allow multiple Outputs to be connected to a single Input, but will never allow multiple Inputs to be connected to one Output. There are internal controls to prevent this and attempting to do it will simply disconnect a previously set path.

## 5.13 LIST MANAGEMENT

Lists can be set most easily through the device webpage, which can be accessed by typing the IP address for the system into any browser address bar. Currently, Cytec switches allow only nine saved lists and list 0 is always the current latched points. Valid values for n are 1-9.

- BS n 73: Saves the current latched switchpoints in List n
- BL n 73: Clears the switch and loads the switch points in List n.
- BD n 73: Displays the switch points in List n.
- BC n 73: Clears List n.

## 5.14 MATRIX COMMAND SUMMARY

| COMMAND | FUNCTION |
|---|---|
| L sw<br>L mod, sw | Latch switch point. |
| U sw<br>U mod, sw | Unlatch switch point. |
| X sw<br>X mod, sw | Multiplex switch point. |
| C | Clear entire system. |
| S<br>S mod, sw | Return status. |
| I | Interrogate Closed Points. |
| F 0/1 73 | Disable/Enable Front Panel. |
| P parameter value 73 | Program parameter. |
| N | Revision Number |

**<u>RS232 Specific Commands</u>**

| | |
|---|---|
| R baud, RTS/CTS 73 | Baud Rate, RTS/CTS operation. |
| A 0/1 73 | Disable/Enable Answerback. |
| E 0/1 73 | Disable/Enable Echo. |
| V 0/1 73 | Disable/Enable Verbose. |

**<u>TCP/IP Specific Commands</u>**

| | |
|---|---|
| TCPANSWERBACK 0/1/2 | Disable/Enable Answerback/Termination |
| IFCONFIG aaa.aaa.aaa.aaa nnn.nnn.nnn.nnn | a = ip address in dotted decimal format<br>n = subnet mask in dotted decimal format |

SNET TCP PORT n m — Where n = equals one of two sockets and m is the port number

## 5.15 IF-12 (RS232/LAN/GPIB) DEFAULT CONFIGURATION SETTINGS

System parameters can be set most easily through the device webpage, which can be accessed by typing the IP address for the system into any browser address bar.

**Default Values**

**TCP Settings:**

| | |
|---|---|
| Port 0 | 8080 |
| Port 1 | 8081 |
| Socket Timeout | 60 seconds |
| TCPAnswerback | 1 (on) |
| TelNet Lock | 0 (off) |

**Serial Settings:**

| | |
|---|---|
| Answerback | 1 (on) |
| Verbose | 0 (off) |
| Echo | 0 (off) |
| Baudrate | 9600 |
| RS Handshake | 1 (RTS/CTS) |

**GPIB Settings:**

| | |
|---|---|
| GPIB Address | 7 |

**Front Panel Settings:**

| | |
|---|---|
| Mux Config | 0 (Single 32) only for 16 or 32 channel systems |
| Front Panel | 1 (on) |

**Miscellaneous Settings:**

| | |
|---|---|
| Use RAM (startup) | 0 (off) |
| Default List | 0 (currently latched switchpoints) |
| Sys ID Number | 0 |

## 5.16 LCD DISPLAY/KEYPAD MANUAL CONTROL OPTION

The Keypad/Display option allows manual control of the matrix from the front panel. Keypad operation is always enabled at power on but may be disabled by the remote command, 'F 0 73'.

**Display**

The display contains two lines with sixteen characters per line. The top line displays matrix commands and numeric entry. The bottom line displays the status of the entry or operation. The display will also show the last command entered from the remote computer interface when the front panel is enabled.

**Keypad**

The keypad consists of ten numeric keys, four function keys, a space key and an enter key.

| Key | Function |
|---|---|
| 0-9 | Numeric entries. |
| space | Delimits between numeric entries. |

| | |
|---|---|
| L | Latch operation. |
| U | Unlatch operation. |
| X | Multiplex operation. |
| C | Clear operation. |
| ENTR | Execute displayed operation. |

**Operation**

A matrix command key, **L**, **U**, **X** or **C**, MUST be pressed before numeric entry keys. Pressing any key except a matrix command key causes the message **Enter Cmd First** to be displayed. After pressing a matrix command key the command and a cursor are displayed. The switch point to be operated on may now be entered with the numeric and space keys. The entry format is the same as described in the MATRIX OPERATION section and described briefly by the following table:

| Command Key | Display Line 1 | Line 2 |
|---|---|---|
| L | Lat _ | Enter Point |
| U | Unl _ | Enter Point |
| X | Mux _ | Enter Point |
| C | Clr _ | Enter Matrix |

The numeric keypad now allows selection of the Module and Relay ( Input and Output ) to be operated on. Each entry may be multiple digits and a space must be pressed between selections.

| Key | Line 1 Display | Line 2 Message |
|---|---|---|
| L | Lat _ | Enter Point |
| 1 | Lat 1_ | |
| Space | Lat 1 _ | |
| 2 | Lat 1 2_ | |
| 3 | Lat 1 23_ | |
| Enter Key | Lat 1 23_ | 1 |

The **ENTR** key may now be pressed to execute the displayed operation. If the displayed entry is incorrect or the operation is not desired, pressing any matrix command key will clear the display and restart the entry.

**Status Display**

After the **ENTR** key is pressed, the displayed operation is attempted to be executed by the control module. If the execution is successful, a **Point Closed** or **Point Open** message will be displayed on line 2. If the operation cannot be executed, an error message will be displayed.

| Line 2 Message | Status |
|---|---|
| Ready | Displayed after power on. |
| Enter Point | The ENTR key has not been pressed, command and selection mode. |
| Point Closed | The selected point was closed. |
| Point Open | The selected point was opened. |
| Points Open | All points opened, Clear operation. |
| ***Err: limits | The selected point is outside the programmed size of the matrix. |
| ***Err: entry | An incorrect entry was selected. |

**Front Panel Disable**

The ‘F’ command allows enabling or disabling front panel operation. If the front panel is disabled, no operation can be performed from the keypad.

| Remote Command | Line 1 | Line 2 |
|---|---|---|

| | | |
|---|---|---|
| F 0 73 | Panel | Disabled |
| F 1 73 | Panel | Enabled |

**Contrast and LED Backlight Adjustment**
Controls are provided to adjust the LCD contrast and LED backlight level. These controls should only need adjustment in extremely bright or dim environments or for acute viewing angles. Both LCD and LED circuits have temperature sensing elements that will automatically adjust the output level for changes in the ambient temperature.

## 5.17 MEMORY SANITATION PROCEDURE

Cytec's IF-12 uses the NXP MCF54415 microprocessor with 32 MB of non-volatile flash memory which is used to store user lists and labels for the webpage factory application and 128kB of user parameter storage. If the unit is ever removed from service or needs to be sanitized for disposal the memory can be erased using one of the following methods.

1) Easiest with least damage. Contact Cytec for factory application .bin file at: sales@cytec-ate.com or 1-585-381-4740.

2) Permanent. Remove cover. Locate IF-12. Remove the NetBurner core board by disconnecting the LAN cable and prying the module off the IF-12 board. Destroy the NetBurner board. Unit is non functional until IF-12 has been replaced.

**APPENDIX - EXAMPLE PROGRAMS**

**Java LAN Programming Example:**

```
import java.net.*;  // for Socket
import java.io.*;   // for IOException and Input/OutputStream

public class if12_lantester
{
        static final int N_MODS = 4;
        static final int N_RLYS = 12;

        /**-----------------test if12 utility functions---------------------------------------*/

        public static void main(String[] args) throws IOException, InterruptedException
        {
            if  (args.length != 2) // Test for correct # of args. IP Address and Port
                throw new IllegalArgumentException("Parameter(s): <IP Address> [<Port>]");

            String server = args[0];       // Server name or IP address
            int servPort = Integer.parseInt(args[1]);  // Port Number

            // Create socket that is connected to server on specified port
            Socket socket = new Socket(server, servPort);
            System.out.println("Connected to server...sending string");
            InputStream in = socket.getInputStream();
            OutputStream out = socket.getOutputStream();

            if12_lan if12 = new if12_lan();

            // Initialize Device: Turn Verbose & Echo off, Answerback on
            if (if12.init_LAN(in,out) < 0)
                throw new SocketException("Error Initializing Device");

            // Clear Device: Unlatch all relays
            if (if12.matrix_clear(in,out) != 48)
                throw new SocketException("Error clearing Device");

            // Latch and Unlatch Relays
            for (int mod =0; mod < N_MODS; mod++)
            {
                for (int rly=0;rly<N_RLYS;rly++)
                {
                    if (if12.point_ops(in,out,'L',0,mod,rly) != 49)
                    {
                        System.out.printf("Error latching Mod %d Rly %d\n",mod,rly);
                        break;
                    }
                    System.out.printf("Latched Mod %d Rly %d\n",mod,rly);
                    if (if12.point_ops(in,out,'U',0,mod,rly) != 48)
                    {
                        System.out.printf("Error unlatching Mod %d Rly %d\n",mod,rly);
```

```
                    break;
                }
                System.out.printf("Unlatched Mod %d Rly %d\n",mod,rly);
            }
        }

        socket.close();  // Close the socket and its streams
    }
}

public class if12_lan
{
    private int bytesRcvd,bytesSent;
    private byte[] rcvBuffer = new byte[256];

    public if12_lan()
    {
        bytesRcvd = 0;
        bytesSent = 0;
    }
    /**------------------Initialize Device----------------------------------*/
    public int init_LAN(InputStream in, OutputStream out) throws IOException,
            InterruptedException
    {
        String str =new String("E0 73;V0 73;TCPANSWERBACK 1\n");
        // Convert string to bytes for writing to output stream
        byte[] byteBuffer = str.getBytes();
        // Send the encoded string to the if12
        out.write(byteBuffer);
        Thread.sleep(1000); //Wait one second
        // Receive the response from the device
        if ((bytesRcvd = in.read(rcvBuffer,0,9)) != 9)
            return -1;
        return 0;
    }
    /**---------------------clear matrix------------------------------------*/
    public int matrix_clear(InputStream in,OutputStream out) throws IOException
    {
        String str = new String("C\n");
        // Convert string to bytes for writing to output stream
        byte[] byteBuffer = str.getBytes();
        // Send the encoded string to the if12
        out.write(byteBuffer);
        // Receive the response from the device
        if ((bytesRcvd = in.read(rcvBuffer,0,3)) == -1)
            return -1;
        return rcvBuffer[0];
    }
    /**------------------switchpoint operations-----------------------------*/
    public int point_ops(InputStream in, OutputStream out,char cmd,int mtx,
        int mod,int rly) throws IOException, InterruptedException
    {
```

```
            //Format command string to send to device
            String cmd_line = String.format("%c%d %d %d\n",cmd,mtx,mod,rly);
            // Convert string to bytes for writing to output stream
            byte[] byteBuffer = cmd_line.getBytes();
            // Send the encoded string to the if12
            out.write(byteBuffer);
            Thread.sleep(100); //Wait 1/10 second
            // Receive the response from the device
            if ((bytesRcvd = in.read(rcvBuffer,0,3)) == -1)
                return -1;
            return rcvBuffer[0];
        }
}
```

**C LAN Programming Example**

```
/* Cytec Matrix Test Program for LAN  */
/* This program uses Microsoft's WS2_32 Library */
/* and winsock2.h. These are available in the */
/* Microsoft SDKs and can be downloaded from */
/* Microsoft's Developer Network */
/* https://msdn.microsoft.com/en-us/default.aspx */

#include <stdio.h>
#include <winsock2.h>
#include <stdlib.h>   /* for exit() */

int init_LAN(int sock);
int point_ops(int sock,int cmd, int mtx, int mod, int rly);
int matrix_clear(int sock);
void DieWithError(char *errorMessage);


#define MAX_MTX   1
#define MAX_MOD  4
#define MAX_RLY 12

int main(int argc, char *argv[])
{
        int sock;
        char *servIP = "10.0.0.144";  /*Default IP Address*?
        struct sockaddr_in servAddr; /* IP address */
        unsigned short servPort = 8080;  /* Port */
        int mtx, mod, rly,status;

        if (argc == 3)
        {
            servIP = argv[1];
            servPort = atoi(argv[2]);
        }
```

```
	WSADATA wsaData;		/* Structure for WinSock setup communication */
	WSAStartup(0x202, &wsaData);	/* Load Winsock 2.2 DLL */

	/* Create a reliable, stream socket using TCP */
	if ((sock = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP))<0)
	     DieWithError("socket() failed");

	/* Construct the server address structure */
	memset(&servAddr, 0, sizeof(servAddr));     /* Zero out structure */
	servAddr.sin_family      = AF_INET;         /* Internet address family */
	servAddr.sin_addr.s_addr = inet_addr(servIP);   /* Server IP address */
	servAddr.sin_port        = htons(servPort); /* Server port */

	/* Establish the connection to the server */
	if (connect(sock, (struct sockaddr *) &servAddr, sizeof(servAddr))<0)
		DieWithError("connect() failed");

	/* Initialize Device using init_LAN Function */
	init_LAN(sock);

	/* Send Clear Command to Device with matrix_clear Function*/
	if ((status = matrix_clear(sock)) != 48)
	     printf("Error clearing device/n");

	/* Simple looping through switchpoints */

	for(mtx=0; mtx<MAX_MTX; mtx++)
	{
	  for (mod=0; mod<MAX_MOD; mod++)
	  {
	     for (rly=0; rly<MAX_RLY; rly++)
	     {
	     if (((status = point_ops(sock,'L',mtx,mod,rly))) !=49)
		printf("Error point %d %d %d not closed\n",mtx,mod,rly);
		else
		printf("Latched point %d %d\n",mod, rly);

		if (((status = point_ops(sock,'U',mtx,mod,rly))) !=48)
		printf("Error point %d %d %d not open\n",mtx,mod,rly);
		else
		printf("Unlatched point %d %d\n",mod, rly);
	     }
	  }
	}
	closesocket(sock);
	WSACleanup();  /* Cleanup Winsock */
	return 0;
}
/*--------------Initialize---------------------------*/

int init_LAN(int sock)
```

```
{
        char rcvString[40];     /* Buffer for device response */
        int rcvStringLen;      /* Length of device response */
        void DieWithError(char *errorMessage);
        /* Initialize Device */
        if ((send(sock, "E0 73;V0 73;TCPANSWERBACK 1\n", 28, 0)) != 28)
        {
            DieWithError("send() failed");
        }
        Sleep(1000);     /* Wait for Response from Device */
        if ((rcvStringLen = recv(sock, rcvString, 9, 0)) < 9)
        {
            DieWithError("recv() failed or connection closed prematurely");
        }
        rcvString[rcvStringLen] = '\0';
        return 0;
}

/*------------------Clear Matrix-----------------------*/

int matrix_clear(int sock)
{

        char rcvString[8];     /* Buffer for device response */
        int rcvStringLen;      /* Length of device response */

        if ((send(sock,"C\n",2,0)) != 2)
            DieWithError("send() failed");

        Sleep(200);   /* Wait for Response

        /* Receive Response from Device */
        if ((rcvStringLen = recv(sock, rcvString, 10, 0)) <= 0)
            DieWithError("recv() failed or connection closed prematurely");
        rcvString[rcvStringLen] = '\0';
        int status = rcvString[0] & 0x3f;

        return status;
}
/*-----------Switchpoint Operation------------------  */

int point_ops(int sock,int cmd, int mtx, int mod, int rly)
{
        char cmd_str[40];          /* Formatted command string */
        char rcvString[8];     /* Buffer for device response */
        int rcvStringLen;      /* Length of device response */

        /* Format String */
        sprintf(cmd_str,"%c%d %d %d\n",cmd,mtx,mod,rly);

        /* Send Command to Device */
        if ((send(sock,cmd_str,strlen(cmd_str),0)) != strlen(cmd_str))
```

```
                DieWithError("send() failed");

        Sleep(200);   /* Wait for Response

        /* Receive Response from Device */
        if ((rcvStringLen = recv(sock, rcvString, 10, 0)) <= 0)
            DieWithError("recv() failed or connection closed prematurely");
        rcvString[rcvStringLen] = '\0';
        int status = rcvString[0] & 0x3f;

        return status;
}

/*------------Error Handling Function---------------------*/

 void DieWithError(char *errorMessage)
{
        fprintf(stderr,"%s: %d\n", errorMessage, WSAGetLastError());
        getchar();
        exit(1);
}
```

**LabWindows RS232 Programming Example**

```
*== Cytec Main Frame Control Include File rs232.h ====================================*/

int RS232port;

/*== GLOBAL FUNCTION DECLARATIONS ==========================================*/

int CYRS232Initialize (int com_port, int baud_rate);
int CyIf3_read (char *buf);
int CyIf3_write (char *buf);
int CyIf3_close (void);


/*================================= END */

#include <ansi_c.h>
#include <utility.h>
/*======================================================================*/
/* Cytec Main Frame RS232 LabWindows/CVI Driver Module                  */
/*======================================================================/*

#include <rs232.h>
#include <formatio.h>
#include "CYRS232.h"

/*= STATIC VARIABLES =====================================================*/
/* port contains the number of the port opened for the instrument module.  */
/* cmd is a buffer for RS-232 I/O strings.                                 */
/* rscnt contains the number of bytes transferred during a read or write.  */
```

```
/* CyIf3_err: the error variable for the instrument module               */
/*=========================================================================*/
//static int port;
static char cmd[26];
static int rscnt;
static int CyIf12_err;

/*= UTILITY ROUTINES ======================================================*/
int CyIf12_invalid_short_range (short val, short min, short max, int err_code);
int CyIf12_invalid_integer_range (int val, int min, int max, int err_code);
int CyIf12_invalid_longint_range (long val, long min, long max, int err_code);
int CyIf12_invalid_real_range (double val, double min, double max, int err_code);
int CyIf12_read_data (char *buf, int cnt, int term);
int CyIf12_write_data (char *buf, int cnt);
int CyIf12_device_closed (void);
void CyIf12_setup_arrays (void);

int main()
{
	CYRS232Initialize(12,9600);

}

/*=========================================================================*/
/* This function opens a com port for the instrument module, queries for   */
/* ID, and initializes the instrument to a known state.                    */
/*=========================================================================*/
int CYRS232Initialize(int com_port, int baud_rate)
{
	char s[40];

   if (CyIf12_invalid_integer_range (baud_rate, 110, 19200, -2) != 0)
      return -14;


   CyIf12_err = OpenComConfig (com_port, "", baud_rate, 0, 8, 1, 512, 512);
   if (CyIf12_err<0) {
      return CyIf12_err;
   }
   CyIf12_err = SetComTime (com_port, 1.0);
   if (CyIf12_err<0) {
      return CyIf12_err;
   }


/*
   Set port to the number of the port just opened.
*/

   RS232port = com_port;

/* Initialize communication, Answerback ON, Verbose, Echo OFF */
```

```
        Fmt (s, "A1,73;V0,73;E0,73\r");
        CyIf12_err = (ComWrt (RS232port, s, StringLength(s)));
        if (CyIf12_err<0)  {
            return CyIf12_err;
        }
        Delay(.1);

        CyIf12_err = ComRdTerm(RS232port, s, 40, '\r');
        if (CyIf12_err<0) {
            return CyIf12_err;
        }
        Delay (1.0);

        FlushInQ (com_port);
        FlushOutQ (com_port);

   return CyIf12_err;
}

/*=========================================================================*/
int CyIf12_read (char *buf)
{
        return(CyIf12_read_data (buf, 40, '\r'));

}

int CyIf12_write (char *buf)
{
        return (CyIf12_write_data (buf, StringLength(buf)));

}


/*=========================================================================*/
/* This function closes the port for the instrument module and sets the    */
/* port to zero.                                                           */
/*=========================================================================*/
int CyIf12_close (void)
{
/*  Check for device closed */

   if (CyIf12_device_closed())
      return CyIf12_err;

/*
   Close the com port.  If error, set CyIf3_err = rs232err+300.
*/

   CloseCom(RS232port);
   if (rs232err != 0) {
      CyIf12_err = rs232err+300;
      return CyIf12_err;
```

```
    }

    RS232port = 0;
    return CyIf12_err;
}

/* = UTILITY ROUTINES ============================================*/

/*================================================================*/
/* Function: Invalid Short Range                                  */
/* Purpose:  This function checks a short to see if it lies between a      */
/*          minimum and maximum value.  If the value is out of range, set */
/*          the global error variable to the value err_code.  If the      */
/*          value is OK, error = 0.                                */
/*================================================================*/
int CyIf12_invalid_short_range (short val, short min, short max, int err_code)
{
  if ((val < min) || (val > max)) {
    CyIf12_err = err_code;
    return -1;
  }
  return 0;
}
/*================================================================*/
/* Function: Invalid Integer Range                                */
/* Purpose:  This function checks an integer to see if it lies between a   */
/*          minimum and maximum value.  If the value is out of range, set */
/*          the global error variable to the value err_code.  If the      */
/*          value is OK, error = 0.                                */
/*================================================================*/
int CyIf12_invalid_integer_range (int val, int min, int max, int err_code)
{
  if ((val < min) || (val > max)) {
    CyIf12_err = err_code;
    return -1;
  }
  return 0;
}
/*================================================================*/
/* Function: Invalid Long Integer Range                           */
/* Purpose:  This function checks a long integer to see if it lies between */
/*          a minimum and maximum value.  If the value is out of range,   */
/*          set the global error variable to the value err_code.  If the  */
/*          value is OK, error = 0.  The return value is equal to the     */
/*          global error value.                              */
/*================================================================*/
int CyIf12_invalid_longint_range (long val, long min, long max, int err_code)
{
    if (val < min || val > max) {
        CyIf12_err = err_code;
        return -1;
    }
```

```
    return 0;
}
/*=====================================================================*/
/* Function: Invalid Real Range                                      */
/* Purpose:  This function checks a real number to see if it lies between  */
/*           a minimum and maximum value.  If the value is out of range,   */
/*           set the global error variable to the value err_code.  If the  */
/*           value is OK, error = 0.                                  */
/*=====================================================================*/
int CyIf12_invalid_real_range (double val, double min, double max, int err_code)
{
  if ((val < min) || (val > max)) {
    CyIf12_err = err_code;
    return -1;
  }
  return 0;
}
/*=====================================================================*/
/* Function: Device Closed                                           */
/* Purpose:  This function checks to see if the module has been            */
/*           initialized.  If the device has not been opened, a 1 is       */
/*           returned, 0 otherwise.                                   */
/*=====================================================================*/
int CyIf12_device_closed (void)
{
    if (RS232port == 0)  {
        CyIf12_err = 232;
        return -1;
    }
    return 0;
}

/*=====================================================================*/
/* Function: Read Data                                               */
/* Purpose:  This function reads a buffer of data from the instrument. The */
/*           return value is equal to the global error variable.           */
/*=====================================================================*/
int CyIf12_read_data (char *buf, int cnt, int term)
{
    rscnt = ComRdTerm(RS232port, buf, cnt, term);
            FlushInQ (RS232port);

    return rscnt;
}

/*=====================================================================*/
/* Function: Write Data                                              */
/* Purpose:  This function writes a buffer of data to the instrument. The  */
/*           return value is equal to the global error variable.           */
/*=====================================================================*/
int CyIf12_write_data (char *buf, int cnt)
{
```

```
   rscnt = ComWrt (RS232port, buf, cnt);

   return rscnt;
}

/*=====================================================================*/
/* This function is called by the init routine to initialize global arrays  */
/* This routine should be modified for each instrument to include          */
/* instrument-dependent commmand arrays.                                   */
/*======================================================================*/
void CyIf12_setup_arrays (void)
{
}
/*= THE END ==============================================================*/
```

**LabWindows GPIB Programming Example**

```
/*=====================================================================*/

/*= Cytec IF-11 IEEE488 Control Module Include File ========================*/

/*== GLOBAL CONSTANT DECLARATIONS ==========================================*/

/* Replace 10 with the maximum number of devices of this type being used.  */
#define IF12_MAX_INSTR    10

/*== GLOBAL FUNCTION DECLARATIONS ==========================================*/
int if12_init (int, int, int *);

/** INSERT INSTRUMENT-DEPENDENT FUNCTION DECLARATIONS HERE **/

int if12_operate(int, int, int, int, int *);
int if12_write (int, char *);
int if12_read (int, int, char *, int *);
int if12_close (int);

/*=== END INCLUDE FILE =====================================================*/


/*=====================================================================*/

#include <gpib.h>
#include <utility.h>
#include <formatio.h>
#include "cy_if12.h"


/*= INSTRUMENT TABLE ========================================================*/
/* address array: contains the GPIB addresses of opened instruments.       */
/* bd array: contains the device descriptors returned by OpenDev.          */
/* instr_cnt: contains the number of instruments open of this model type.  */
```

```
/*=================================================================*/
static int address[IF12_MAX_INSTR + 1];
static int bd[IF12_MAX_INSTR + 1];
static int instr_cnt;

/*= STATIC VARIABLES ===============================================*/
/* cmd is a buffer for GPIB I/O strings.                          */
/* if12_err: the error variable for the instrument module          */
/* ibcnt: contains the number of bytes transferred by GPIB reads and     */
/*       writes.  See the GPIB library I/O Class for more information    */
/*=================================================================*/
static char cmd[50];
static int if12_err;

/*= UTILITY ROUTINES ===============================================*/
int if12_open_instr (int, int *);
int if12_close_instr (int);
int if12_invalid_integer_range (int, int, int, int);
int if12_device_closed (int);
int if12_read_data (int, char *, int);
int if12_write_data (int, char *, int);
int if12_set_timeout (int, int, int *);
void if12_setup_arrays (void);

/*=================================================================*/
/* Function: Initialize                                            */
/* Purpose:  This function opens the instrument, queries the instrument    */
/*          for its ID, and initializes the instrument to a known state.  */
/*=================================================================*/
int if12_init (addr, rest, instrID)
int   addr;
int   rest;
int * instrID;
{
        int  ID;

        if (if12_invalid_integer_range (addr, 0, 30, -1) != 0)
            return if12_err;
        if (if12_invalid_integer_range (rest, 0, 1, -3) != 0)
            return if12_err;

        if (if12_open_instr (addr, &ID) != 0)
            return if12_err;

        if (rest) {
            if (if12_write_data (ID, "C", 1) != 0) {
                if12_close_instr (ID);
                return if12_err;
            }
            Delay(0.01);
        }
        if12_setup_arrays ();
```

```
        *instrID = ID;

        return if12_err;
}

/*=========================================================================*/
/* - Operations: Latch, Unlatch, Multiplex, Clear and Status --- */

int if12_operate (instrID, Operation, Module, Relay, Status)
int instrID, Operation, Module, Relay;
int *Status;
{
        char s[20];

        *Status = -1;
        if (Operation == 'C') {
            Fmt(s,"C");
            if (if12_write_data(instrID, s, StringLength(s)) != 0)
                return if12_err;
            Delay(0.01);
        }
        else {
            Fmt(s,"%c %d %d", Operation, Module, Relay);
            if (if12_write_data(instrID, s, StringLength(s)) != 0)
                return if12_err;
        }
        if (if12_read_data(instrID, s, 2) != 0)
            return if12_err;
        *Status = s[0] & 0xf;
        return if12_err;
}

/*=========================================================================*/
/* Function: Write To Instrument                                  */
/* Purpose:  This function writes a command string to the instrument.     */
/*=========================================================================*/
int if12_write (instrID, cmd_string)
int   instrID;
char *cmd_string;
{
        if (if12_invalid_integer_range (instrID, 1, IF12_MAX_INSTR, -1) != 0)
            return if12_err;
        if (if12_device_closed(instrID) != 0)
            return if12_err;

        Fmt (cmd, "%s<%s", cmd_string);
        if (if12_write_data (instrID, cmd, NumFmtdBytes()) != 0)
            return if12_err;

        return if12_err;
}
```

```
/*=================================================================*/
/* Function: Read Instrument Buffer                                */
/* Purpose:  This function reads the output buffer of the instrument.      */
/*=================================================================*/
int if12_read (instrID, numbytes, in_buff, bytes_read)
int   instrID;
int   numbytes;
char *in_buff;
int  *bytes_read;
{
        if (if12_invalid_integer_range (instrID, 1, IF12_MAX_INSTR, -1) != 0)
            return if12_err;
        if (if12_device_closed(instrID) != 0)
            return if12_err;

        *bytes_read = 0;
        if (if12_read_data (instrID, in_buff, numbytes) != 0)
            return if12_err;

        *bytes_read = ibcnt;

        return if12_err;
}

/*=================================================================*/
/* Function: Close                                                 */
/* Purpose:  This function closes the instrument.                  */
/*=================================================================*/
int if12_close (instrID)
int instrID;
{
        if (if12_invalid_integer_range (instrID, 1, IF12_MAX_INSTR, -1) != 0)
            return if12_err;
        if (if12_device_closed (instrID))
            return if12_err;

        if12_close_instr (instrID);

        return if12_err;
}

/*= UTILITY ROUTINES ==============================================*/

/*=================================================================*/
/* Function: Open Instrument                                       */
/* Purpose:  This function locates and initializes an entry in the         */
/*           Instrument Table and the GPIB device table for the            */
/*           instrument.  The size of the Instrument Table can be changed  */
/*           in the include file by altering the constant                  */
/*           IF12_MAX_INSTR.  The return value of this function is equal */
/*           to the global error variable.                          */
/*=================================================================*/
```

```
int if12_open_instr (addr, ID)
int  addr;
int *ID;
{
	int i, instrID;

	instrID = 0;
	if12_err = 0;

/* Check to see if the instrument is already in the Instrument Table. */

	for (i = 1; i <= IF12_MAX_INSTR; i++)
	    if (address[i] == addr) {
	        instrID = i;
	        i = IF12_MAX_INSTR;
	     }

/* If it is not in the instrument table, open an entry for the instrument. */

	if (instrID <= 0)
	    for (i = 1; i <= IF12_MAX_INSTR; i++)
	        if (address[i] == 0) {
	            instrID = i;
	            i = IF12_MAX_INSTR;
	         }

/* If an entry could not be opened in the Instrument Table, return an error.*/

	if (instrID <= 0) {
	    if12_err = 220;
	    return if12_err;
	}

/* If the device has not been opened in the GPIB device table (bd[ID] = 0),*/
/*  then open it.                                                  */

	if (bd[instrID] <= 0) {
	    if (instr_cnt <= 0)
	        CloseInstrDevs("if12");
	    bd[instrID] = OpenDev ("", "if12");
	    if (bd[instrID] <= 0) {
	        if12_err = 220;
	        return if12_err;
	    }
	    instr_cnt += 1;
	    address[instrID] = addr;
	 }

/*  Change the primary address of the device    */

	if (ibpad (bd[instrID], addr) < 0) {
	    if12_err = 233;
```

```
        return if12_err;
    }

    *ID = instrID;
    return if12_err;
}

/*===========================================================================*/
/* Function: Close Instrument                                        */
/* Purpose:  This function closes the instrument by removing it from the   */
/*          GPIB device table and setting the address and bd[instrID] to  */
/*          zero in the Instrument Table.  The return value is equal to   */
/*          the global error variable.                              */
/*===========================================================================*/
int if12_close_instr (instrID)
int instrID;
{
    if (bd[instrID] != 0) {
        CloseDev (bd[instrID]);
        bd[instrID] = 0;
        address[instrID] = 0;
        instr_cnt -= 1;
    }
    else
        if12_err = 221;

    return if12_err;
}

/*===========================================================================*/
/* Function: Invalid Integer Range                                   */
/* Purpose:  This function checks an integer to see if it lies between a   */
/*          minimum and maximum value.  If the value is out of range, set */
/*          the global error variable to the value err_code.  If the      */
/*          value is OK, error = 0.  The return value is equal to the     */
/*          global error value.                                     */
/*===========================================================================*/
int if12_invalid_integer_range (val, min, max, err_code)
int val;
int min;
int max;
int err_code;
{
    if (val < min || val > max)
        if12_err = err_code;
    else
        if12_err = 0;

    return if12_err;
}

/*===========================================================================*/
```

```
/* Function: Device Closed                                        */
/* Purpose:  This function checks to see if the module has been          */
/*           initialized.  If the device has not been opened, set the     */
/*           global error variable to 232, 0 otherwise. The return value   */
/*           is equal to the global error value.                          */
/*=====================================================================*/
int if12_device_closed (instrID)
int instrID;
{
          if (bd[instrID] <= 0)
               if12_err = 232;
          else
               if12_err = 0;

          return if12_err;
}

/*=====================================================================*/
/* Function: Read Data                                            */
/* Purpose:  This function reads a buffer of data from the instrument. The */
/*           return value is equal to the global error variable.          */
/*=====================================================================*/
int if12_read_data (instrID, buf, cnt)
int   instrID;
char *buf;
int   cnt;
{
          if (ibrd(bd[instrID], buf, (long)cnt) <= 0)
               if12_err = 231;
          else
               if12_err = 0;

          return if12_err;
}

/*=====================================================================*/
/* Function: Write Data                                           */
/* Purpose:  This function writes a buffer of data to the instrument. The  */
/*           return value is equal to the global error variable.          */
/*=====================================================================*/
int if12_write_data (instrID, buf, cnt)
int   instrID;
char *buf;
int   cnt;
{
          if (ibwrt(bd[instrID], buf, (long)cnt) <= 0)
               if12_err = 230;
          else
               if12_err = 0;

          return if12_err;
}
```

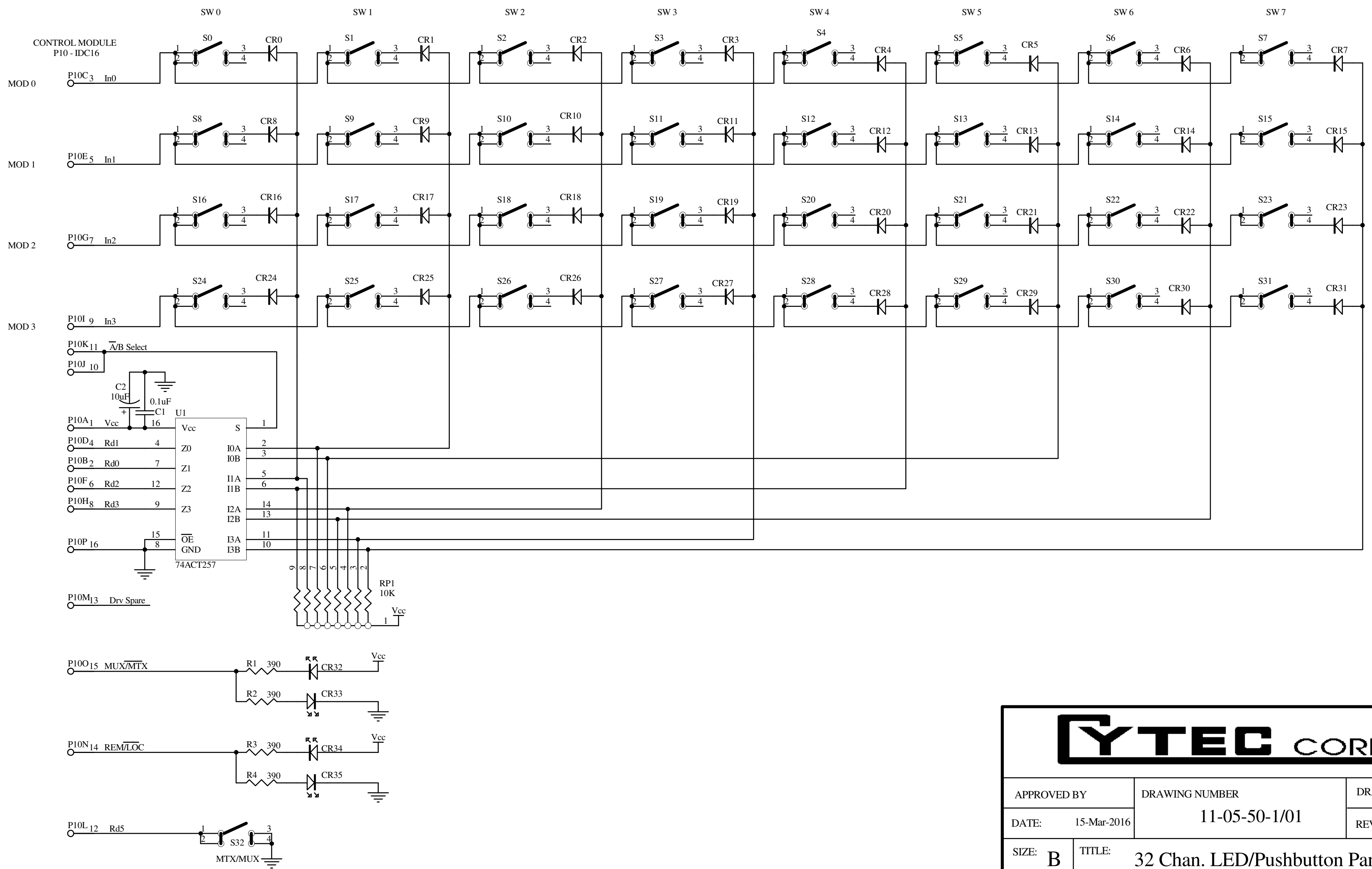
```
/*=====================================================================*/
/* Function: Set Timeout                                              */
/* Purpose:  This function changes or disables the timeout of the device.  */
/*          Refer to the LabWindows Standard Libraries Reference Manual   */
/*          for timeout codes. The return value is equal to the global    */
/*          error variable.                                           */
/*=====================================================================*/
int if12_set_timeout (instrID, tmo_code, old_timeout)
int  instrID;
int  tmo_code;
int *old_timeout;
{
        *old_timeout = ibtmo (bd[instrID], tmo_code);
        if (ibsta <= 0)
            if12_err = 239;
        else
            if12_err = 0;

        return if12_err;
}

/*=====================================================================*/
/* Function: Setup Arrays                                             */
/* Purpose:  This function is called by the init routine to initialize     */
/*          static arrays.                                            */
/*          This routine should be modified for each instrument to        */
/*          include instrument-dependent commmand arrays.              */
/*=====================================================================*/
void if12_setup_arrays ()
{
}
/*=== THE END ===================================================*/
```

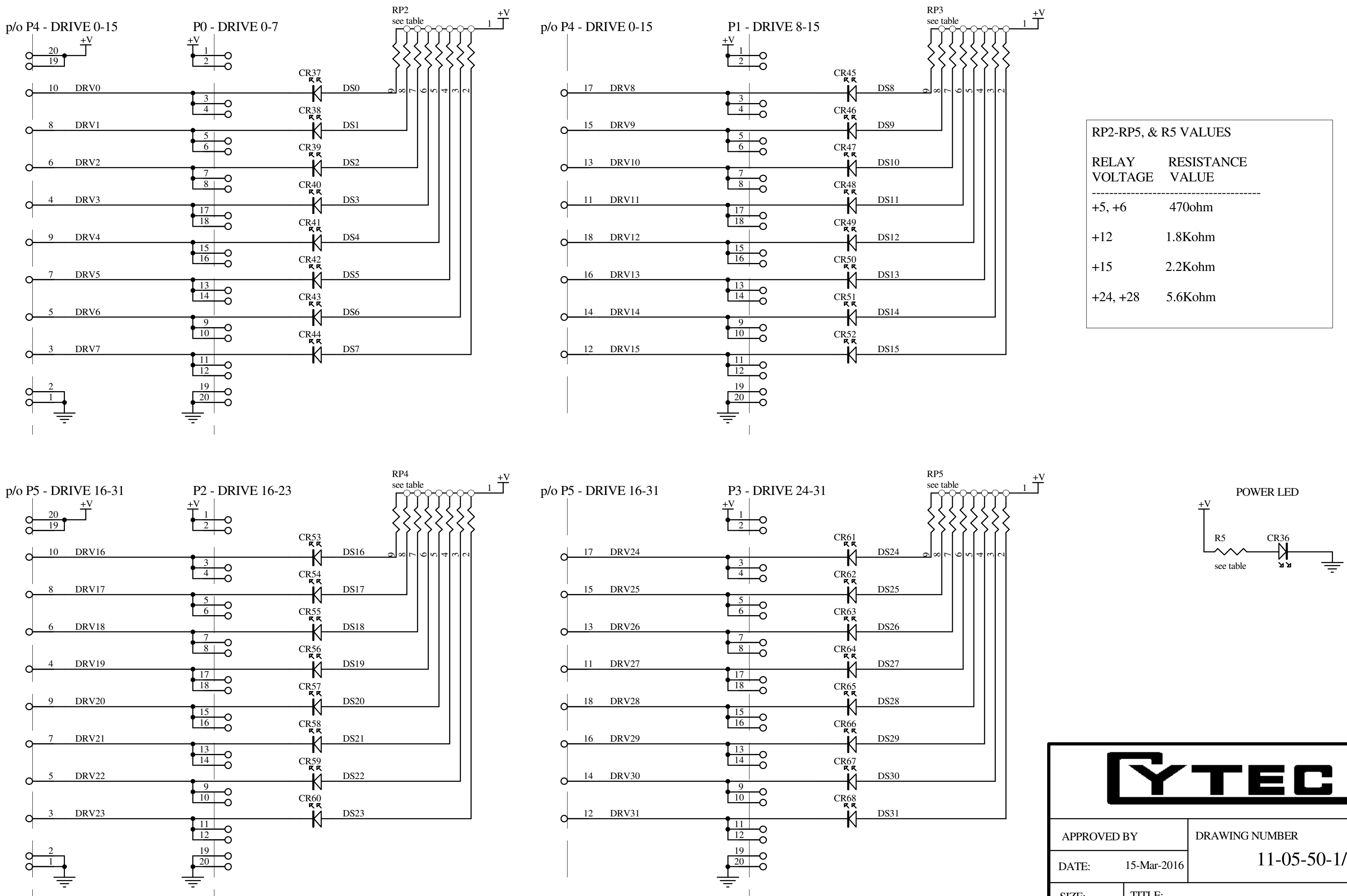
**LabView Drivers**

Labview Drivers are available for download at https://cytec-ate.com/downloads/drivers/




CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
| DATE: 15-Mar-2016 | 11-05-50-1/01 | REVISION: |
| SIZE: B | TITLE: 32 Chan. LED/Pushbutton Panel | |
| 11-05-50-1-01.sch | SHEET 1 OF 2 | |

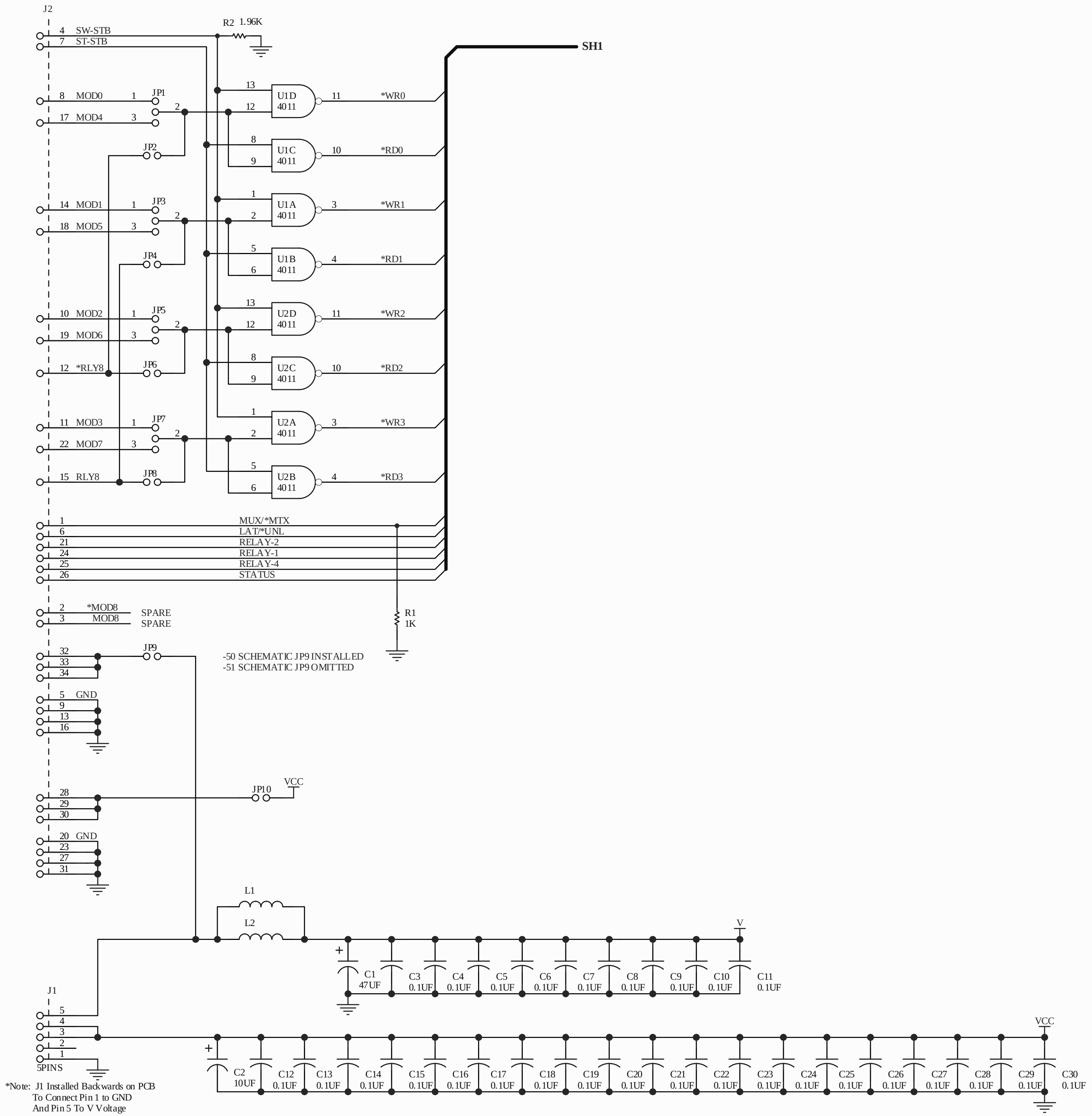



RP2-RP5, & R5 VALUES

| RELAY VOLTAGE | RESISTANCE VALUE |
|---|---|
| +5, +6 | 470ohm |
| +12 | 1.8Kohm |
| +15 | 2.2Kohm |
| +24, +28 | 5.6Kohm |

CYTEC CORP.

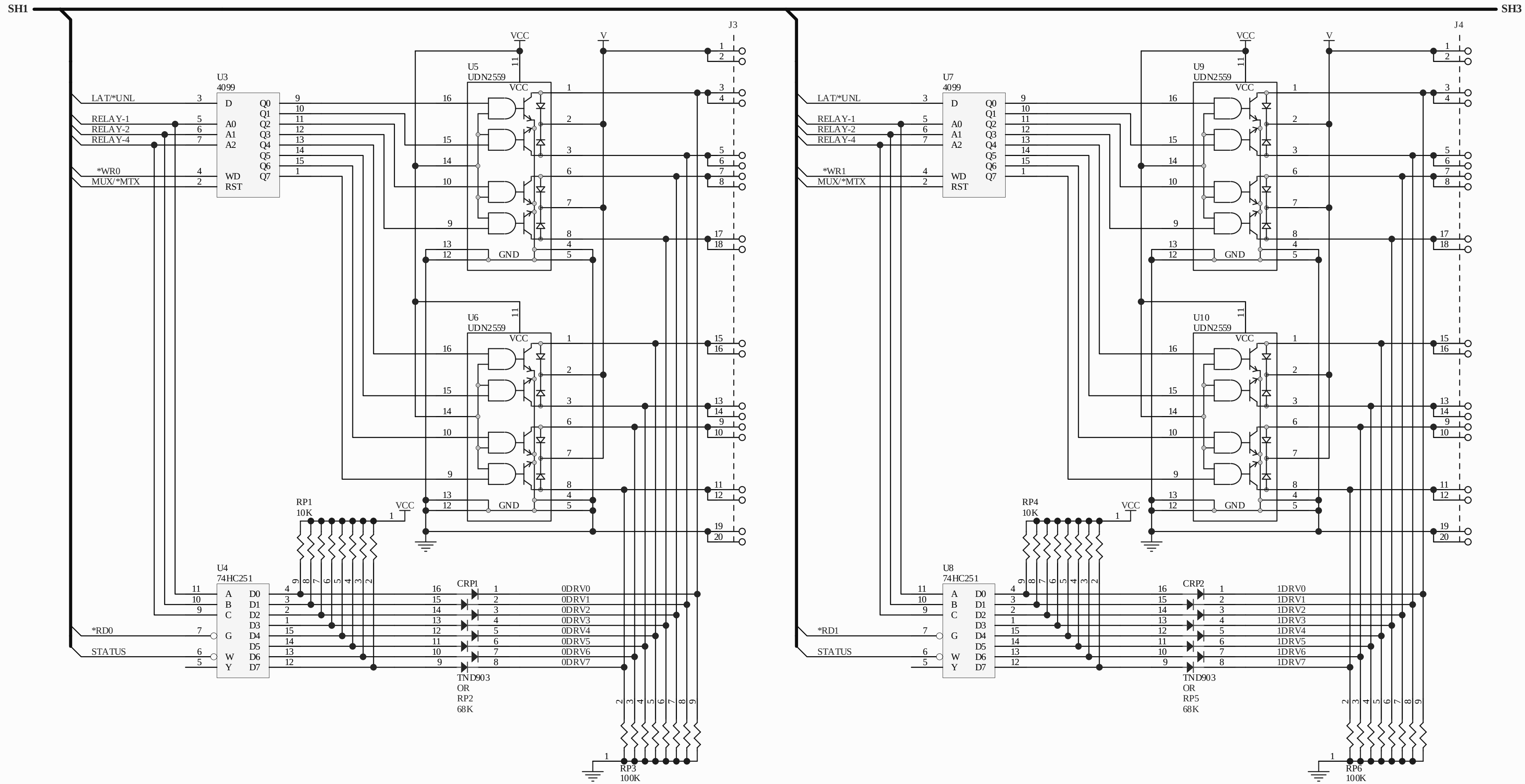
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
| DATE: 15-Mar-2016 | 11-05-50-1/02 | REVISION: |
| SIZE: B | TITLE: 32 Chan. LED/Pushbutton Panel | |
| 11-05-50-1-02.sch | SHEET 2 OF 2 | |




REVISED FROM:

CYTEC CORP.

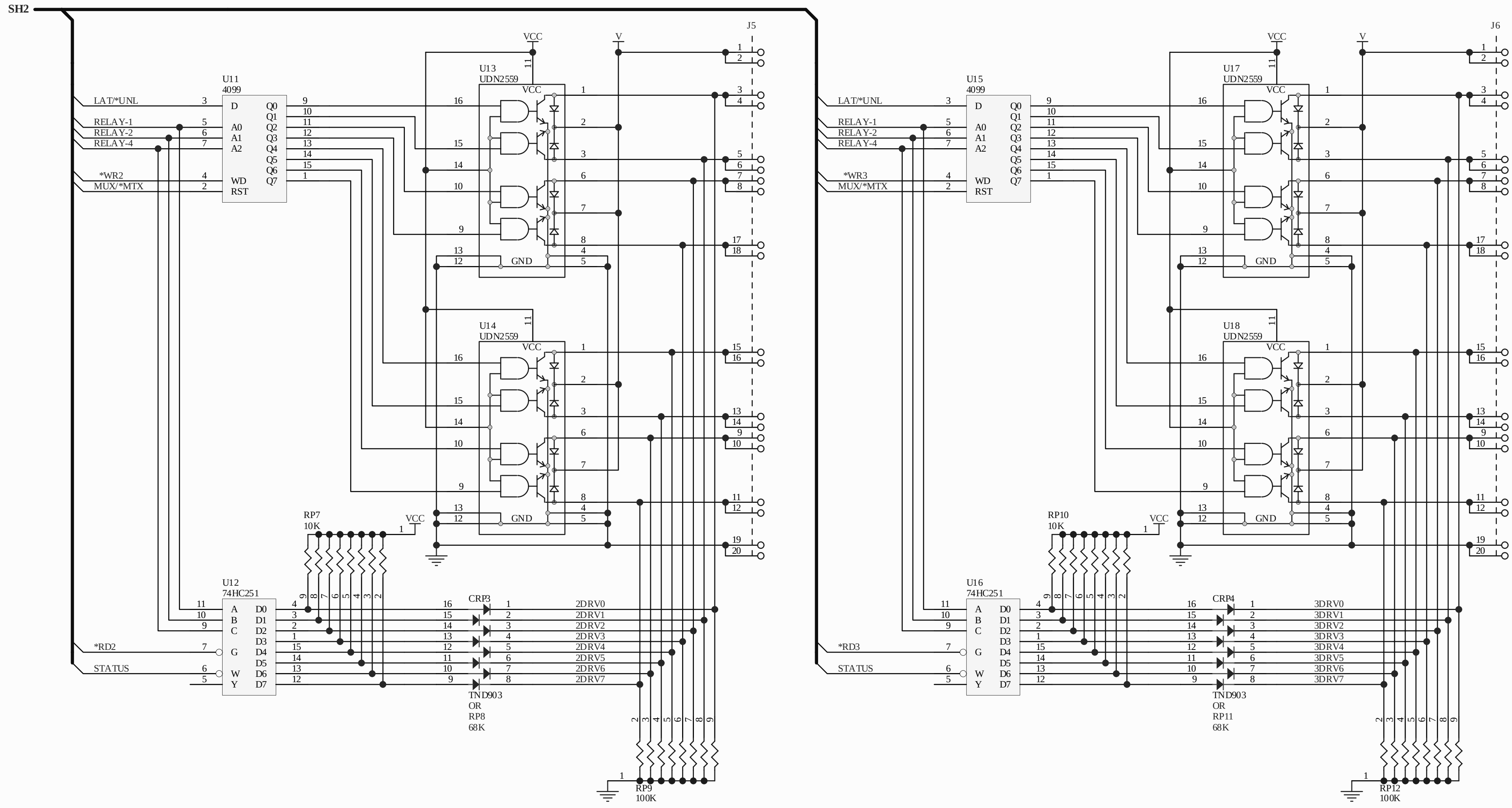
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
| DATE: 11/2/2007 | 11-08-50/1 | REVISION: |
| SIZE: A3 | TITLE: CM/32-VHP | |
| 11-08-50/1 | | SHEET 1 OF 3 |




REVISED FROM:

CYTEC CORP.

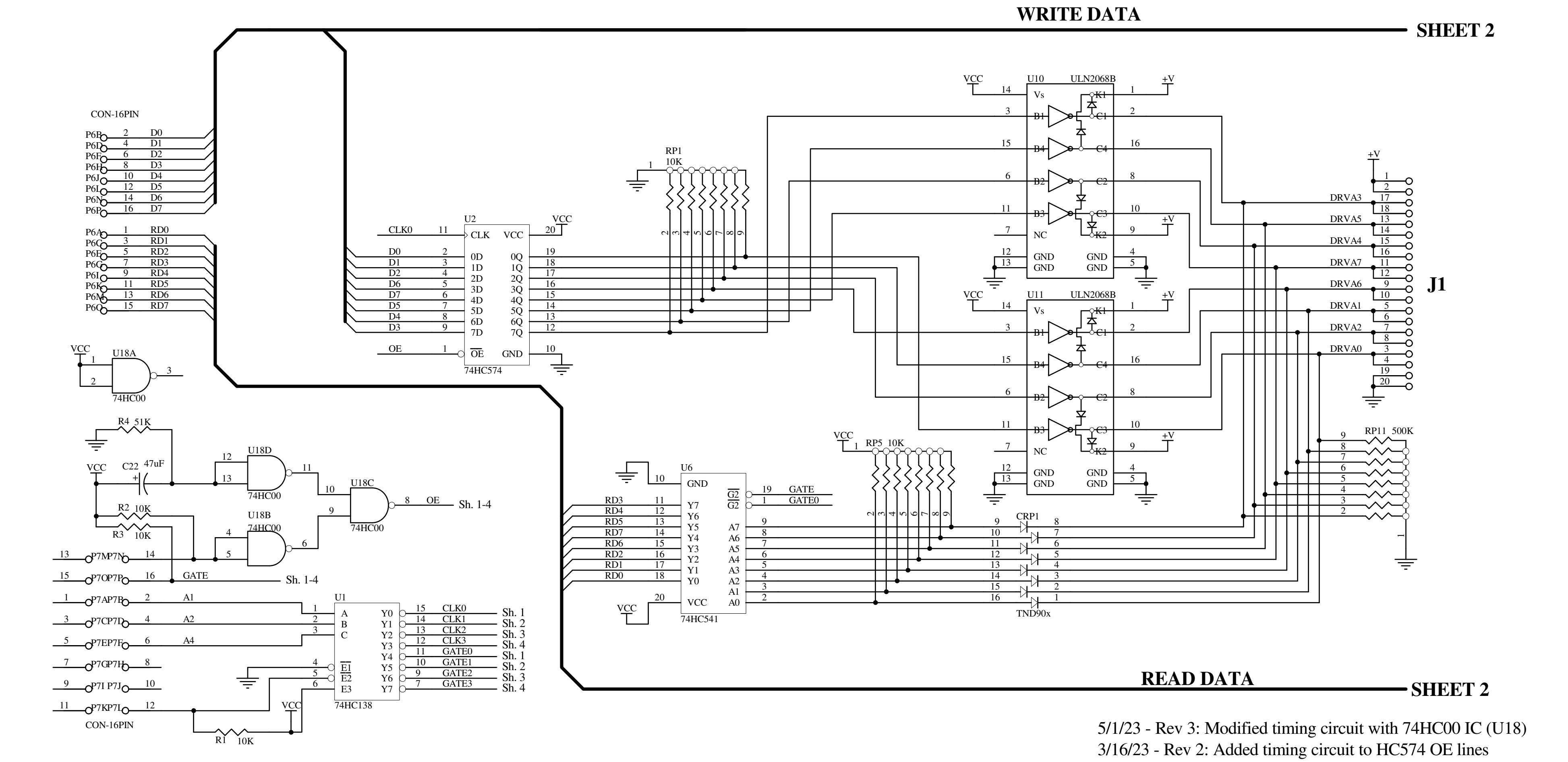
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
| DATE: 11/2/2007 | 11-08-50/2 | REVISION: |
| SIZE: A3 | TITLE: CM/32-VHP | |
| 11-08-50/2 | | SHEET 2 OF 3 |




REVISED FROM:

CYTEC CORP.

| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
|---|---|---|
| DATE: 11/2/2007 | 11-08-50/3 | REVISION: |
| SIZE: A3 | TITLE: CM/32-VHP | |
| 11-08-50/3 | | SHEET 3 OF 3 |

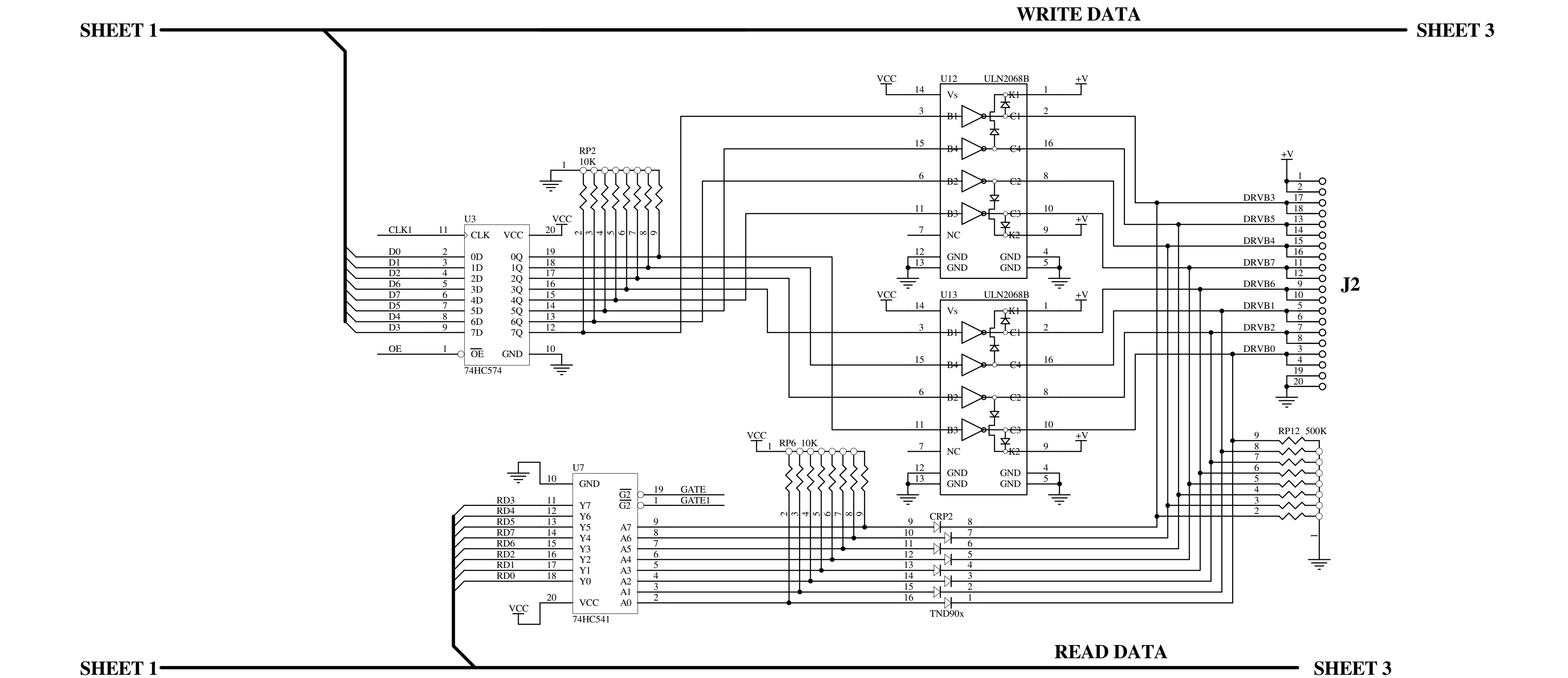



5/1/23 - Rev 3: Modified timing circuit with 74HC00 IC (U18)

3/16/23 - Rev 2: Added timing circuit to HC574 OE lines

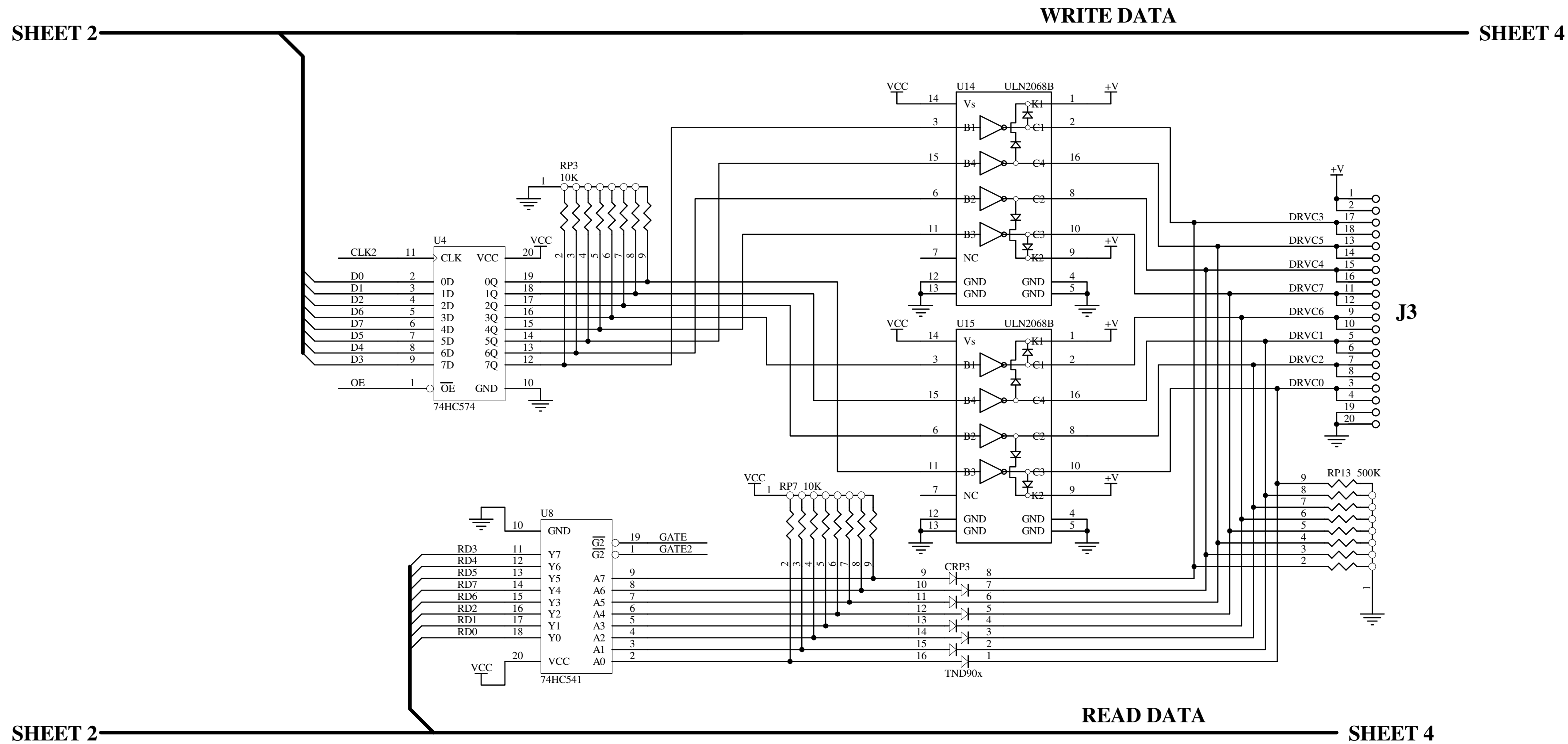
CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER 11-11-50/01 | DRAWN BY: KJA |
| DATE: 26-May-2023 | | REVISION: 3 |
| SIZE: B | TITLE: RD-9 Relay Driver Board | |
| 11-11-50-3-01.sch | | SHEET 1 OF 4 |




CYTEC CORP.

| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
|---|---|---|
| DATE: 26-May-2023 | 11-11-50/02 | REVISION: 3 |
| SIZE: B | TITLE: RD-9 Relay Driver Board | |
| 11-11-50-3-02.sch | | SHEET 2 OF 4 |

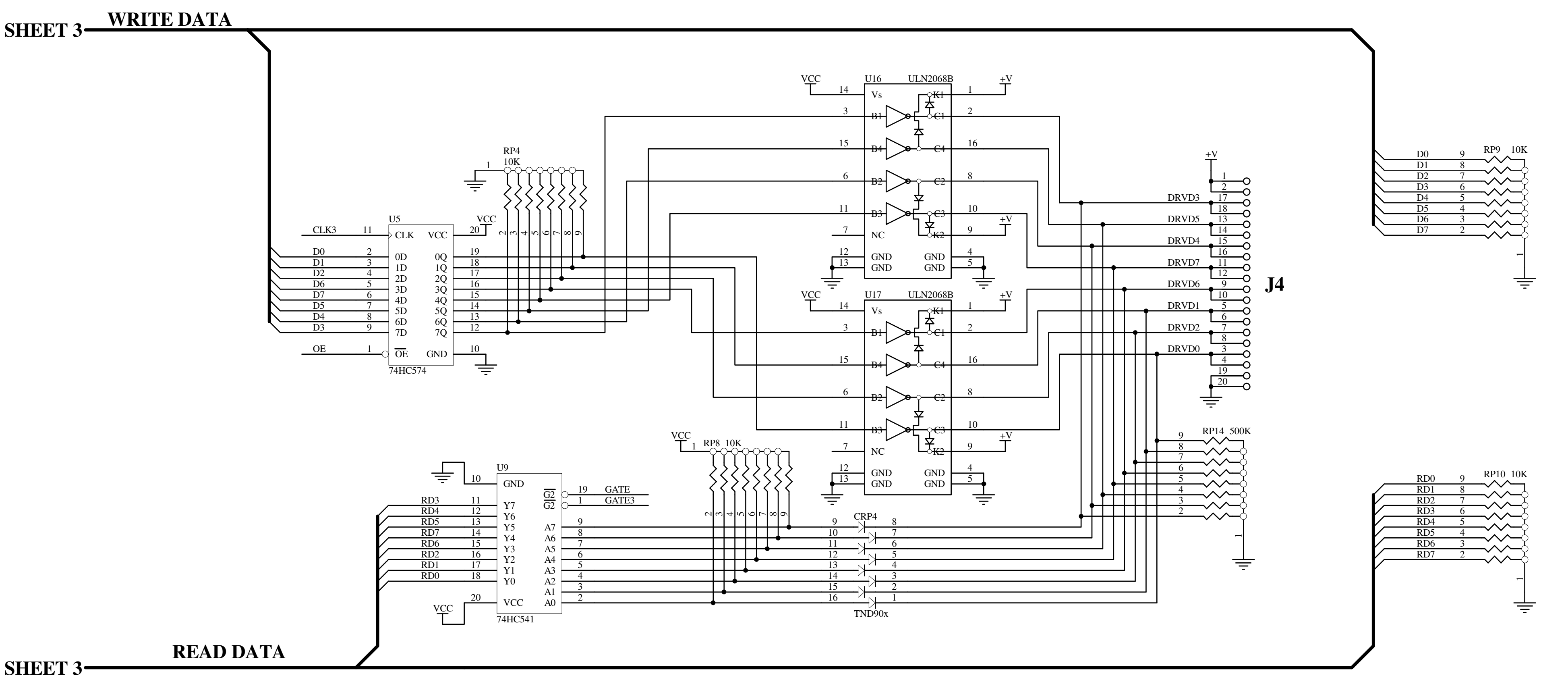



CYTEC CORP.

| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
|---|---|---|
| DATE: 26-May-2023 | 11-11-50/03 | REVISION: 3 |

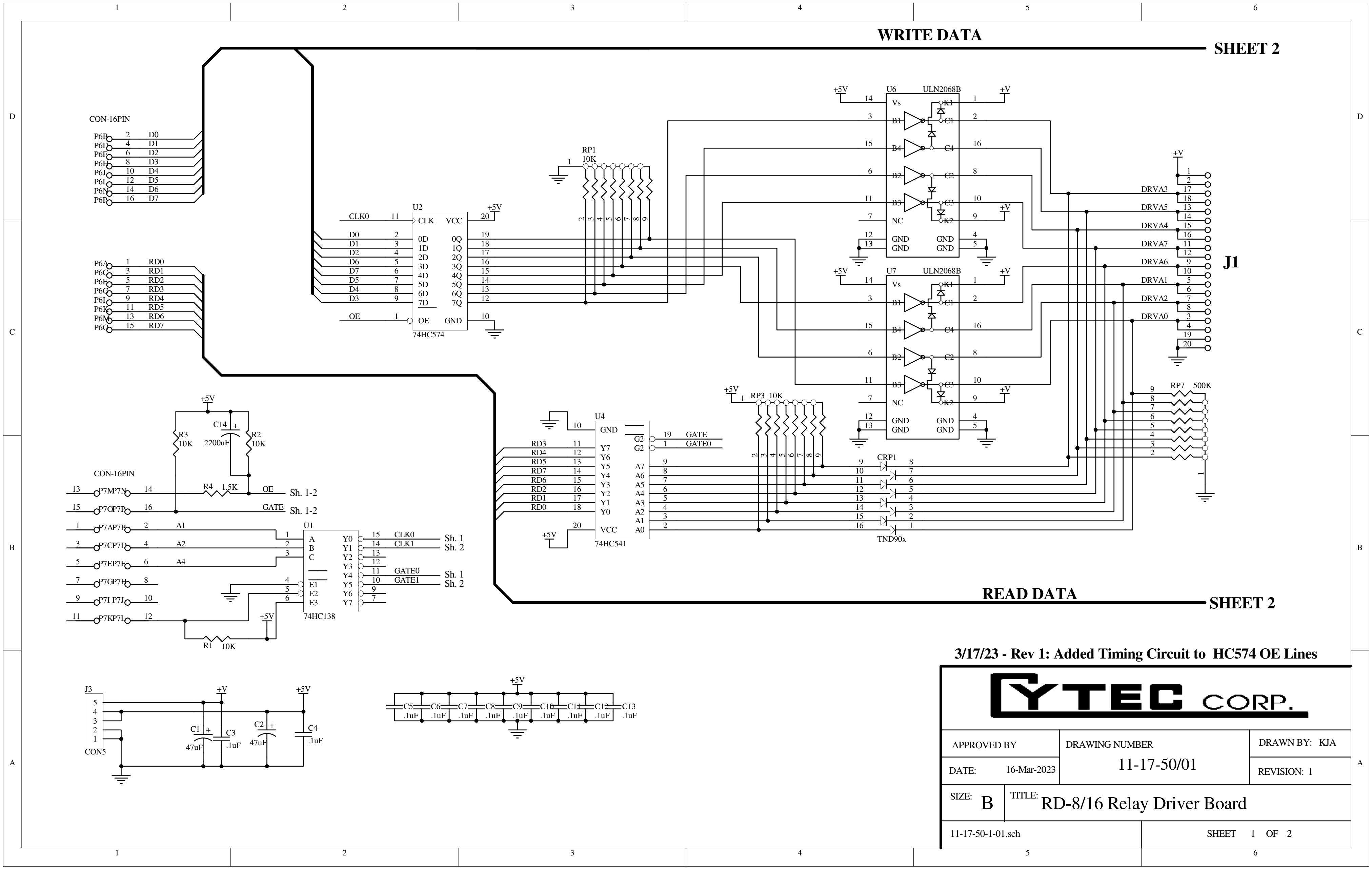
SIZE: B TITLE: RD-9 Relay Driver Board

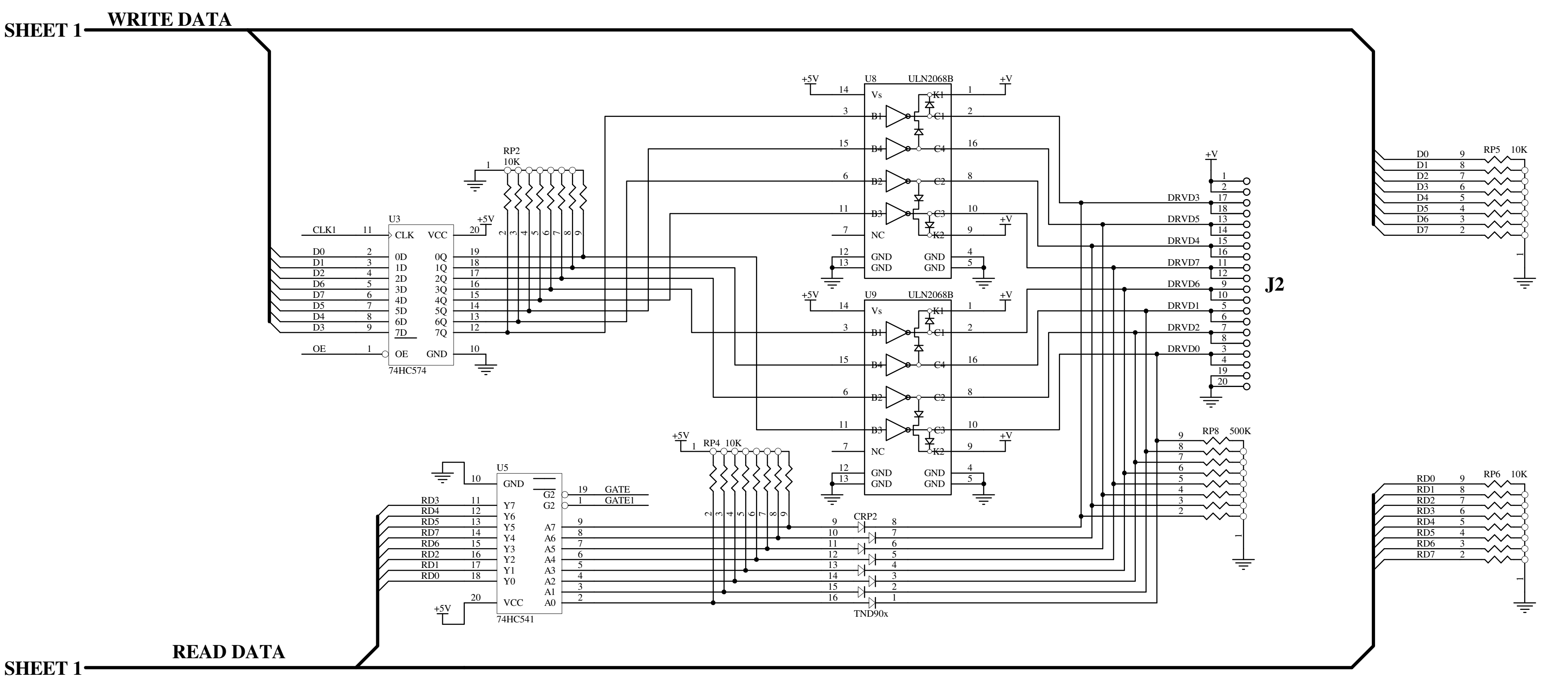
11-11-50-3-03.sch SHEET 3 OF 4




CYTEC CORP.

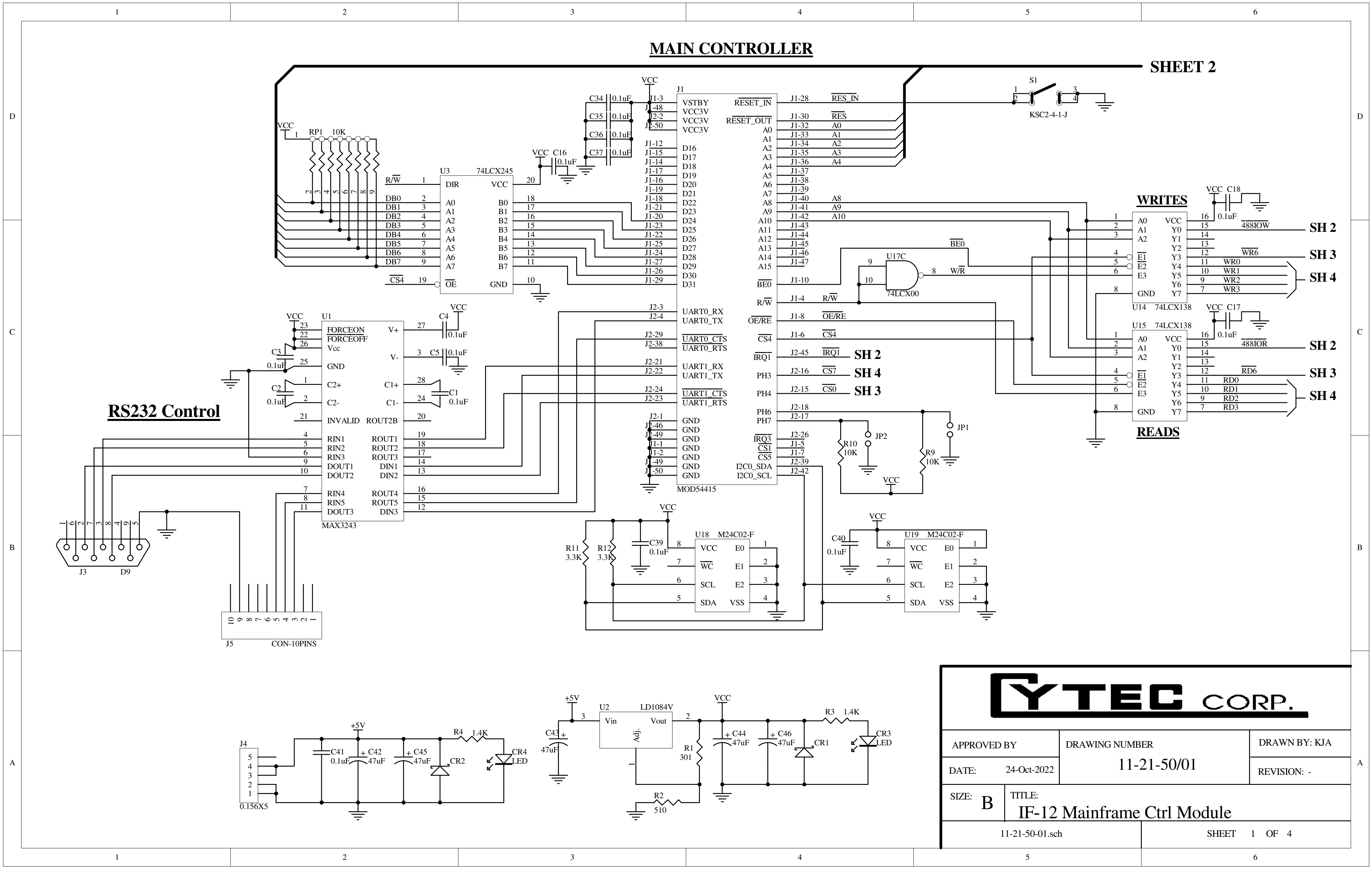
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
| DATE: 26-May-2023 | 11-11-50/04 | REVISION: 3 |
| SIZE: B | TITLE: RD-9 Relay Driver Board | |
| 11-11-50-3-04.sch | SHEET 4 OF 4 | |


WRITE DATA
SHEET 2
READ DATA
SHEET 2
CON-16PIN
U2
74HC574
U4
74HC541
U1
74HC138
U6
ULN2068B
U7
ULN2068B
RP1
10K
RP3 10K
RP7 500K
CRP1
TND90x
J1
J3
CON5
C14
2200uF
R3
10K
R2
10K
R4 1.5K
R1 10K
OE Sh. 1-2
GATE Sh. 1-2
CLK0 Sh. 1
CLK1 Sh. 2
GATE0 Sh. 1
GATE1 Sh. 2
C1 47uF
C2 47uF
C3 .1uF
C4 .1uF
C5 C6 C7 C8 C9 C10 C11 C12 C13 .1uF
DRVA3
DRVA5
DRVA4
DRVA7
DRVA6
DRVA1
DRVA2
DRVA0
3/17/23 - Rev 1: Added Timing Circuit to HC574 OE Lines
CYTEC CORP.
APPROVED BY
DRAWING NUMBER
11-17-50/01
DRAWN BY: KJA
DATE: 16-Mar-2023
REVISION: 1
SIZE: B
TITLE: RD-8/16 Relay Driver Board
11-17-50-1-01.sch
SHEET 1 OF 2

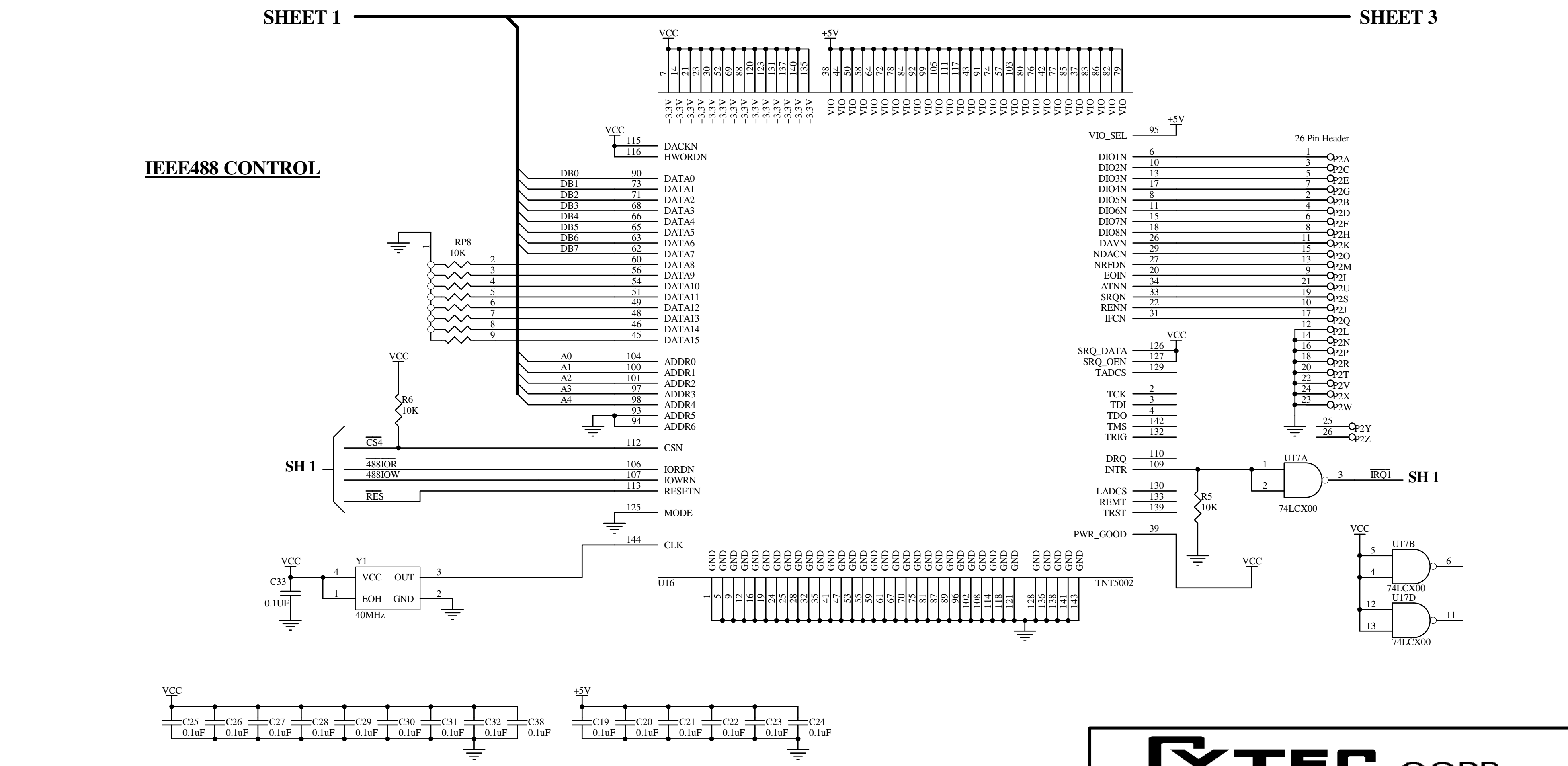



Error : LOGO2.BMP file not found.

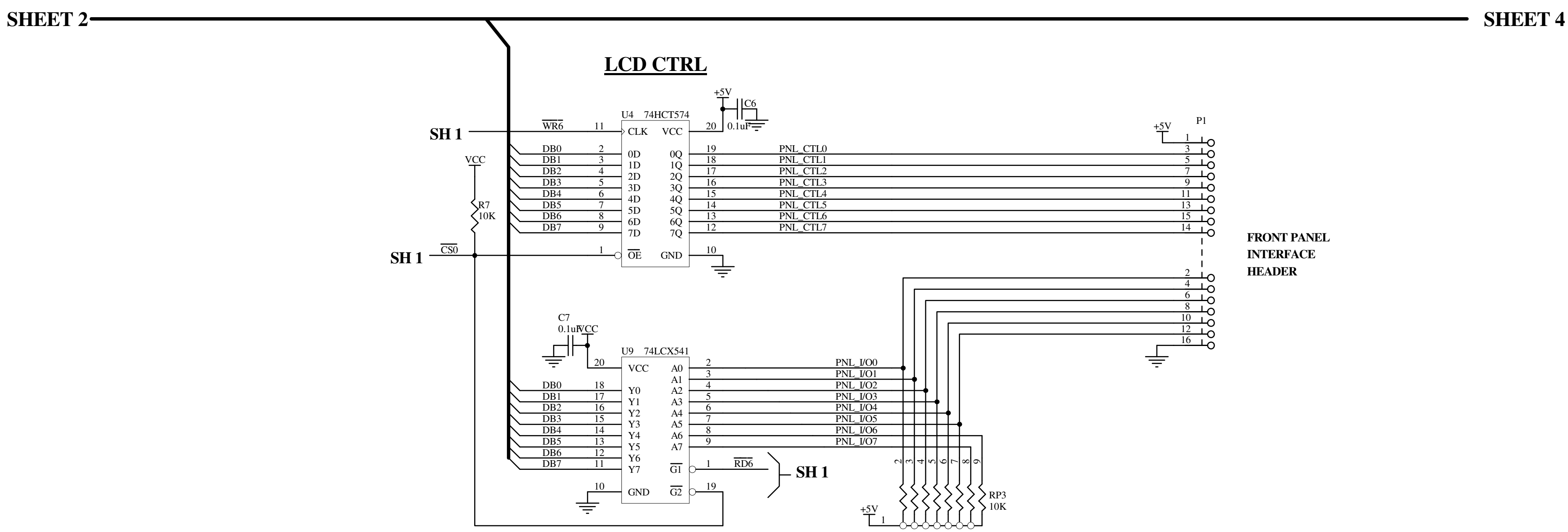
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
| DATE: 16-Mar-2023 | 11-17-50/02 | REVISION: 1 |
| SIZE: B | TITLE: RD-8/16 Relay Driver Board | |
| 11-17-50-1-02.sch | | SHEET 2 OF 2 |



MAIN CONTROLLER
SHEET 2
RS232 Control
WRITES
READS
SH 2
SH 3
SH 4
J1 MOD54415
U1 MAX3243
U3 74LCX245
U14 74LCX138
U15 74LCX138
U17C 74LCX00
U18 M24C02-F
U19 M24C02-F
U2 LD1084V
S1 KSC2-4-1-J
RP1 10K
J3 D9
J5 CON-10PINS
J4 0.156X5
CYTEC CORP.
APPROVED BY
DRAWING NUMBER
11-21-50/01
DRAWN BY: KJA
DATE: 24-Oct-2022
REVISION: -
SIZE: B
TITLE:
IF-12 Mainframe Ctrl Module
11-21-50-01.sch
SHEET 1 OF 4

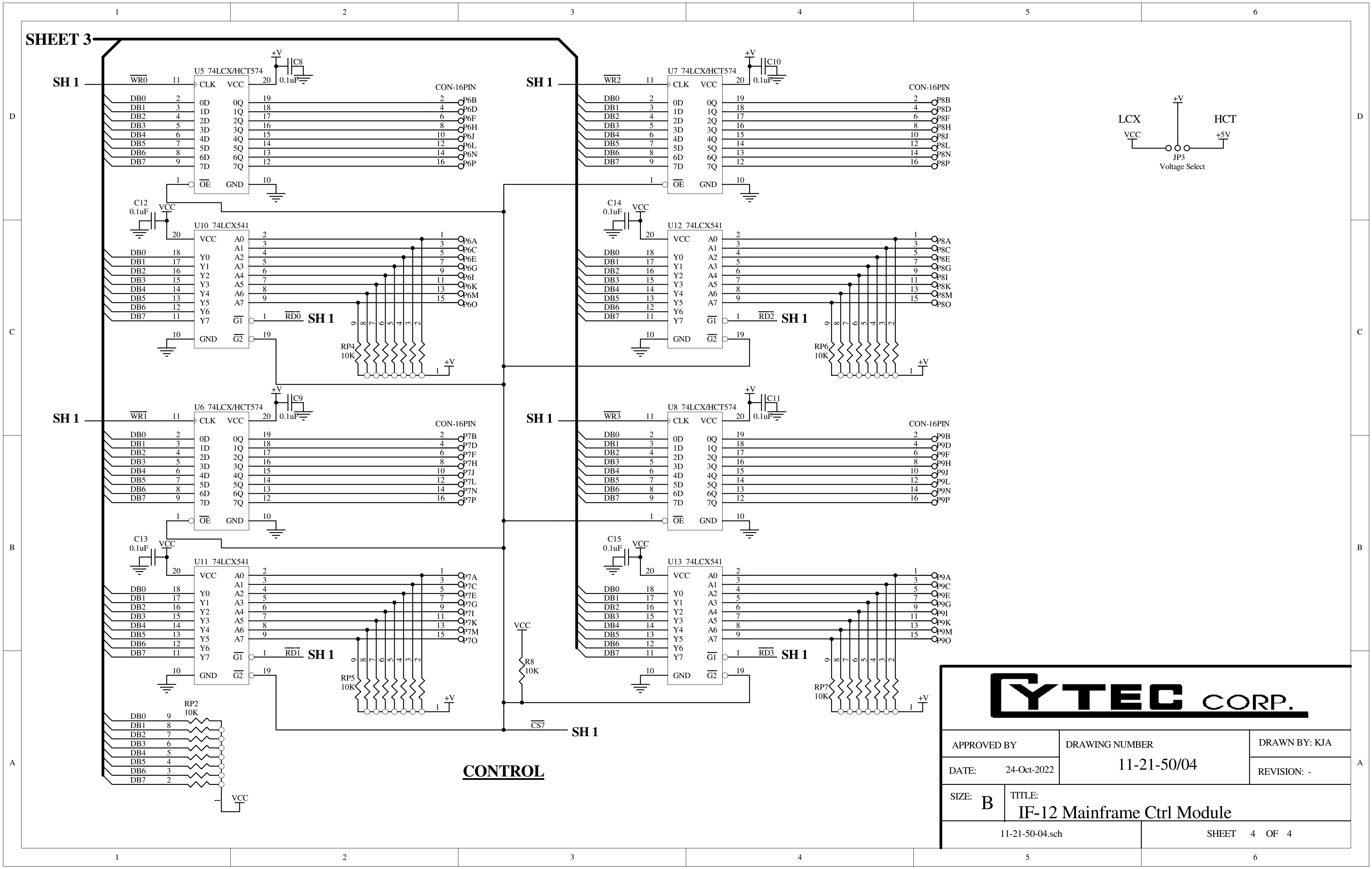



| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
| DATE: 24-Oct-2022 | 11-21-50/02 | REVISION: - |
| SIZE: B | TITLE: IF-12 Mainframe Ctrl Module | |
| | 11-21-50-02.sch | SHEET 2 OF 4 |

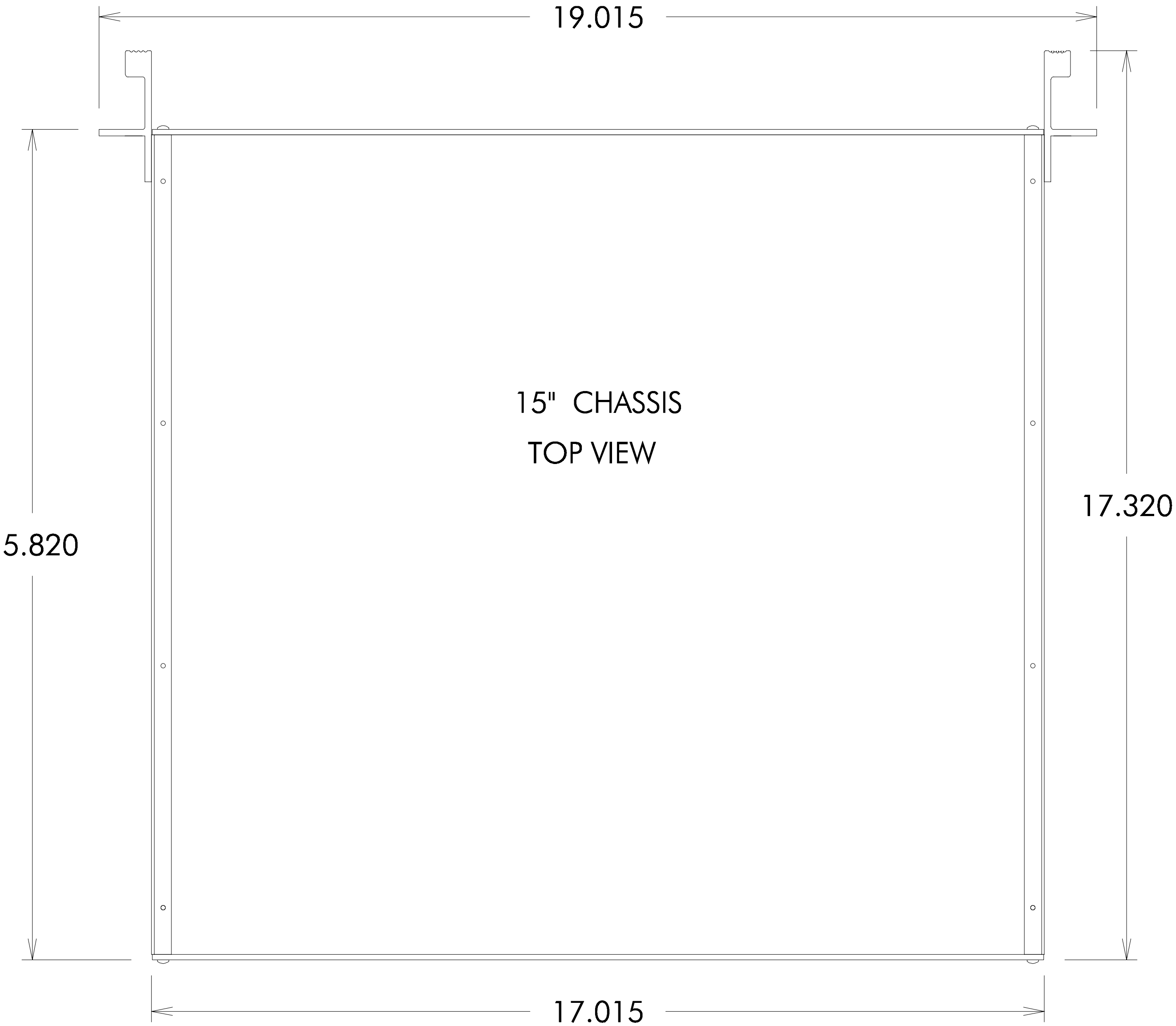



CYTEC CORP.

| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
|---|---|---|
| DATE: 24-Oct-2022 | 11-21-50/03 | REVISION: - |
| SIZE: B | TITLE: IF-12 Mainframe Ctrl Module | |
| 11-21-50-03.sch | SHEET 3 OF 4 | |



SHEET 3
SH 1
U5 74LCX/HCT574
U6 74LCX/HCT574
U7 74LCX/HCT574
U8 74LCX/HCT574
U10 74LCX541
U11 74LCX541
U12 74LCX541
U13 74LCX541
CON-16PIN
C8 0.1uF
C9 0.1uF
C10 0.1uF
C11 0.1uF
C12 0.1uF
C13 0.1uF
C14 0.1uF
C15 0.1uF
RP2 10K
RP4 10K
RP5 10K
RP6 10K
RP7 10K
R8 10K
LCX
HCT
VCC
+V
+5V
JP3
Voltage Select
CONTROL
CYTEC CORP.
APPROVED BY
DRAWING NUMBER
11-21-50/04
DRAWN BY: KJA
DATE: 24-Oct-2022
REVISION: -
SIZE: B
TITLE: IF-12 Mainframe Ctrl Module
11-21-50-04.sch
SHEET 4 OF 4




1U = 1.75" HIGH

2U = 3.5" HIGH

3U = 5.25" HIGH

4U = 7" HIGH

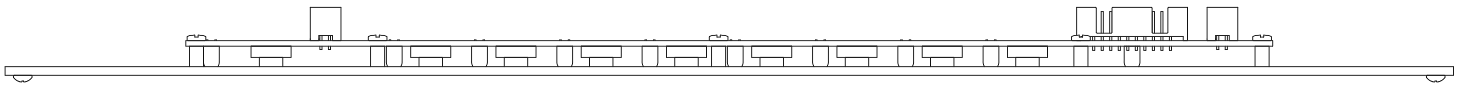
5U = 8.75" HIGH

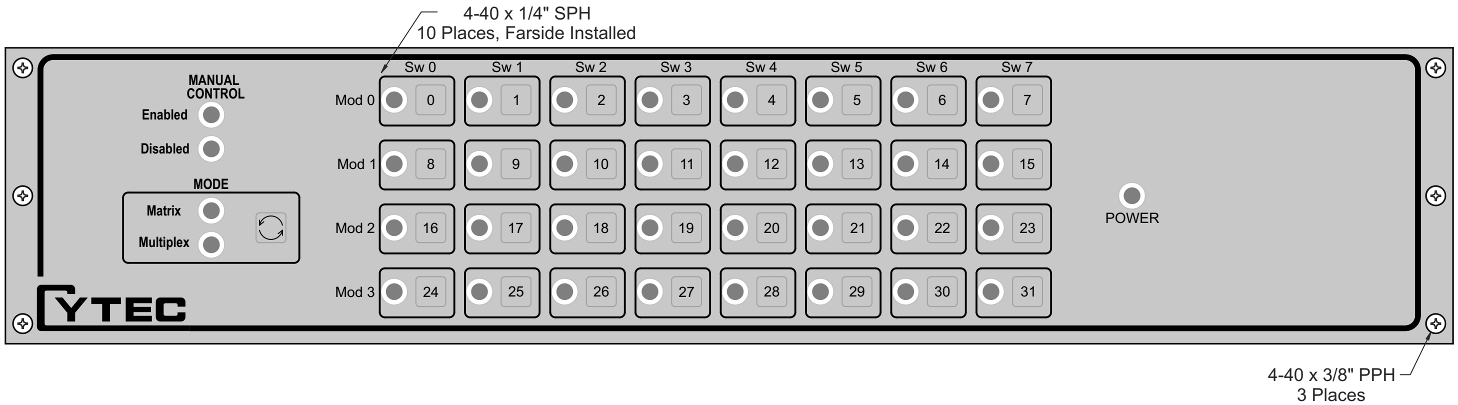
6U = 10.5" HIGH

CYTEC CORP.

TOL. +/- .010" U.O.S.

| | | | | | |
|---|---|---|---|---|---|
| DATE: 3/26/03 | CAD FILE: 14-20-20.dw2 | | DRAWN BY: DR | | |
| SIZE: D | TITLE: BASIC CHASSIS DIM's | | | | |
| COMMENTS: 15" CHASSIS | | | DRAWING NUMBER: 14-20-20 | | |





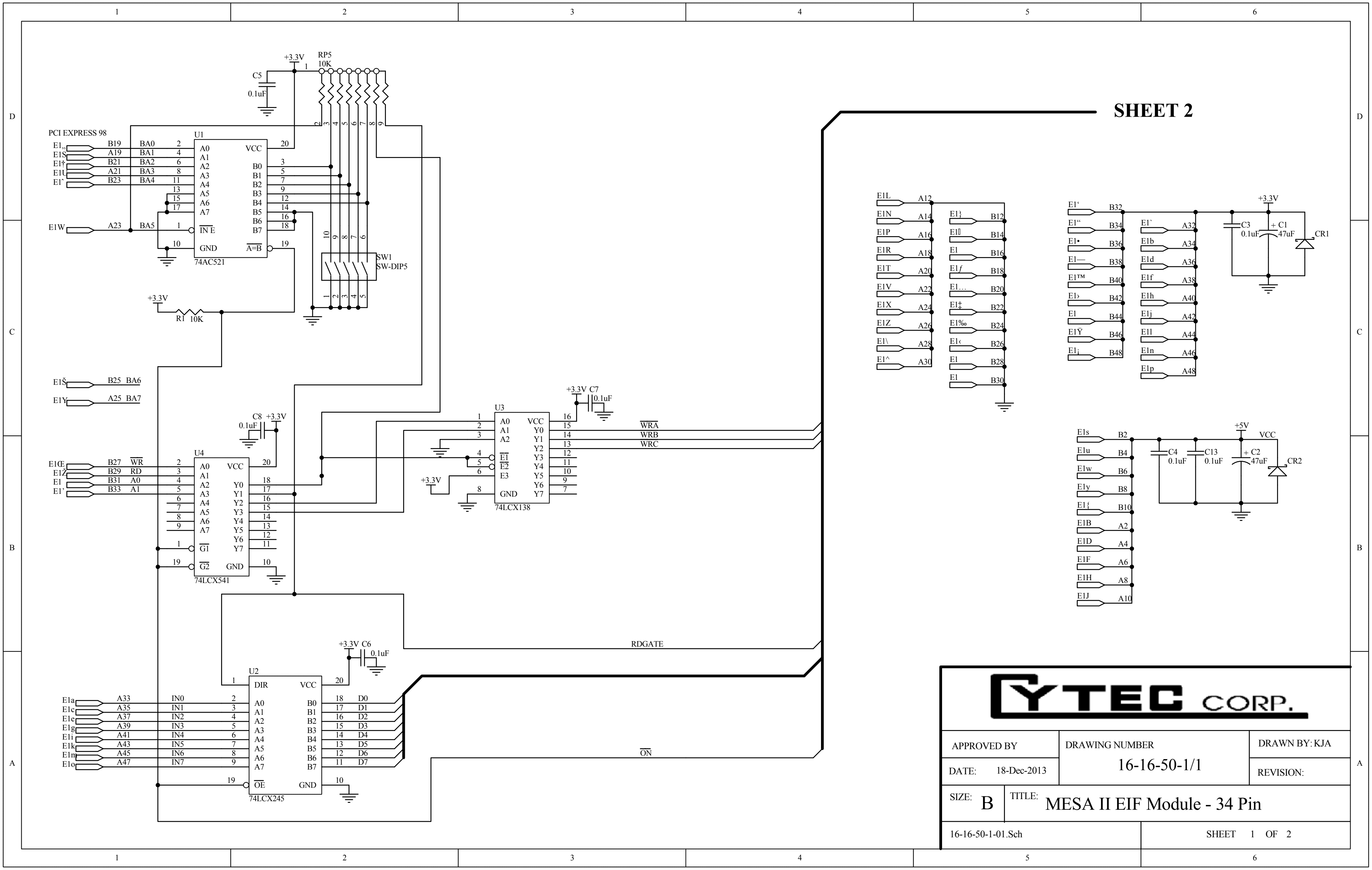

| Cytec P/N | Description | Qty |
|---|---|---|
| FAS1014 | 4-40 x 3/8" Phillips Panhead | 6 |
| FAS1022 | 4-40 x 1/4" Slotted Panhead | 10 |

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| A | 10/21/19 | PV | Added Ruggedization Brk Mounting |
| - | 6/24/04 | DR | Drawing Start |

CYTEC CORP

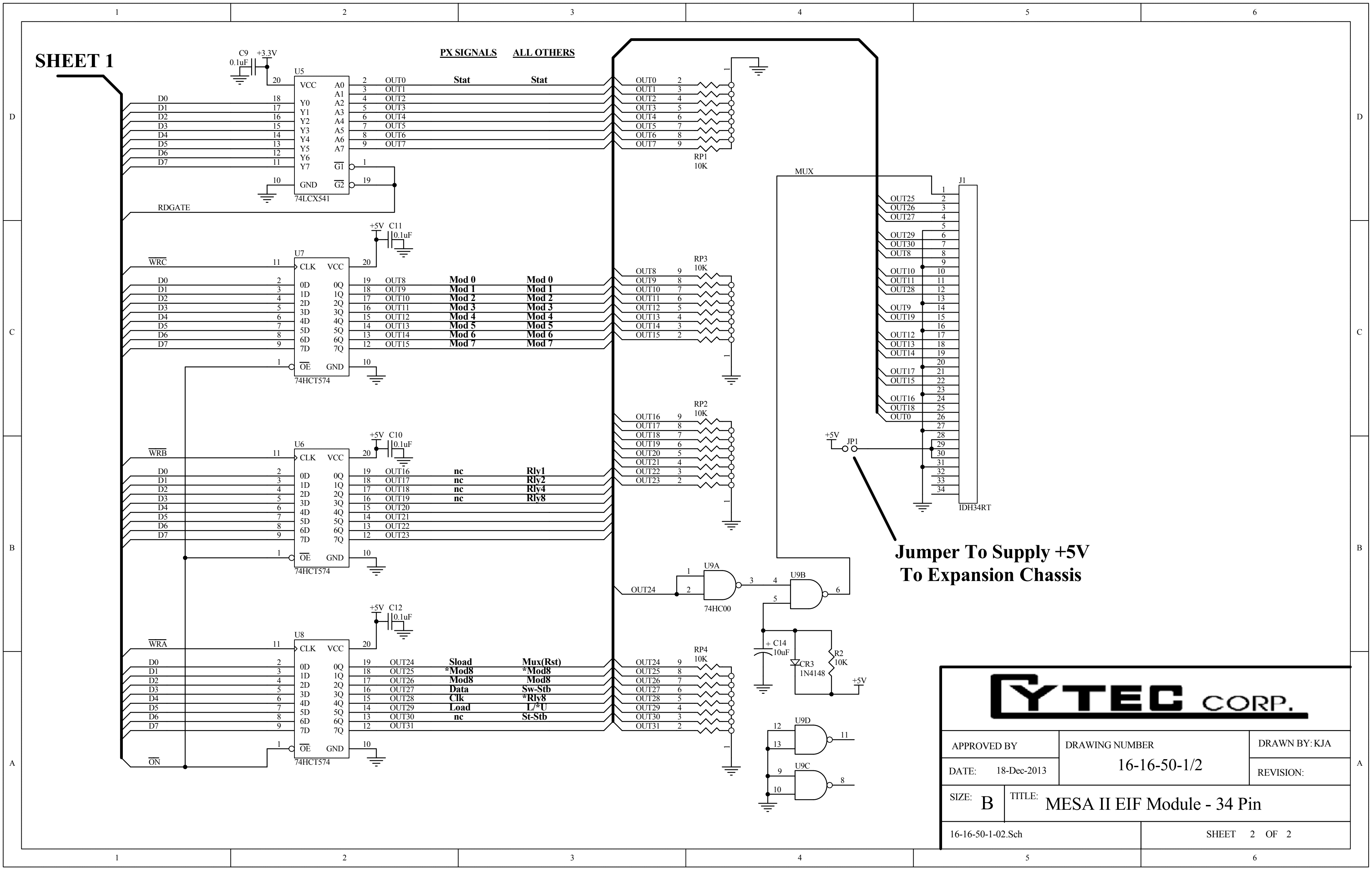
TOL. +/- 0.01" U.O.S.

| | | |
|---|---|---|
| Date: 6/24/04 | CAD File: 15-60-22 Rev A.dxf | Drawn By: DR |
| Size: A | Title: 16/32 Channel Front Panel | |
| Comments: AFP1049 | | Drawing Number: 15-60-22 |

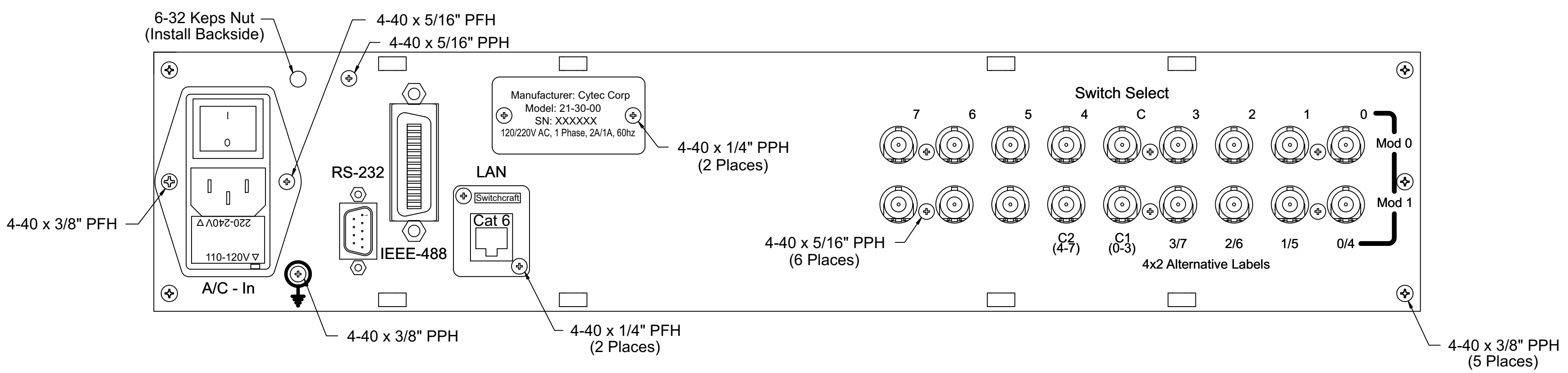



CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER 16-16-50-1/1 | DRAWN BY: KJA |
| DATE: 18-Dec-2013 | | REVISION: |
| SIZE: B | TITLE: MESA II EIF Module - 34 Pin | |
| 16-16-50-1-01.Sch | | SHEET 1 OF 2 |



SHEET 1
PX SIGNALS
ALL OTHERS
C9
0.1uF
+3.3V
U5
VCC
GND
74LCX541
D0
D1
D2
D3
D4
D5
D6
D7
RDGATE
OUT0 Stat Stat
OUT1
OUT2
OUT3
OUT4
OUT5
OUT6
OUT7
RP1
10K
U7
74HCT574
WRC
CLK
OE
C11
0.1uF
+5V
OUT8 Mod 0 Mod 0
OUT9 Mod 1 Mod 1
OUT10 Mod 2 Mod 2
OUT11 Mod 3 Mod 3
OUT12 Mod 4 Mod 4
OUT13 Mod 5 Mod 5
OUT14 Mod 6 Mod 6
OUT15 Mod 7 Mod 7
RP3
10K
U6
74HCT574
WRB
C10
0.1uF
OUT16 nc Rly1
OUT17 nc Rly2
OUT18 nc Rly4
OUT19 nc Rly8
OUT20
OUT21
OUT22
OUT23
RP2
10K
U8
74HCT574
WRA
ON
C12
0.1uF
OUT24 Sload Mux(Rst)
OUT25 *Mod8 *Mod8
OUT26 Mod8 Mod8
OUT27 Data Sw-Stb
OUT28 Clk *Rly8
OUT29 Load L/*U
OUT30 nc St-Stb
OUT31
RP4
10K
MUX
J1
IDH34RT
JP1
U9A
U9B
U9C
U9D
74HC00
C14
10uF
CR3
1N4148
R2
10K
Jumper To Supply +5V
To Expansion Chassis
CYTEC CORP.
APPROVED BY
DRAWING NUMBER
16-16-50-1/2
DRAWN BY: KJA
DATE: 18-Dec-2013
REVISION:
SIZE: B
TITLE: MESA II EIF Module - 34 Pin
16-16-50-1-02.Sch
SHEET 2 OF 2




| Cytec P/N | Description | Qty |
|---|---|---|
| FAS1017 | 4-40 x 1/4" Phillips Panhead | 2 |
| FAS1041 | 4-40 x 1/4" Phillips Flathead | 2 |
| FAS1010 | 4-40 x 5/16" Phillips Flathead | 1 |
| FAS1016 | 4-40 x 5/16" Phillips Panhead | 7 |
| FAS1081 | 4-40 x 3/8" Phillips Flathead | 1 |
| FAS1014 | 4-40 x 3/8" Phillips Panhead | 6 |
| FAS1012 | 6-32 Keps Nut | 1 |

| | | | |
|---|---|---|---|
| - | 9/27/22 | PV | Drawing Start |

CYTEC CORP

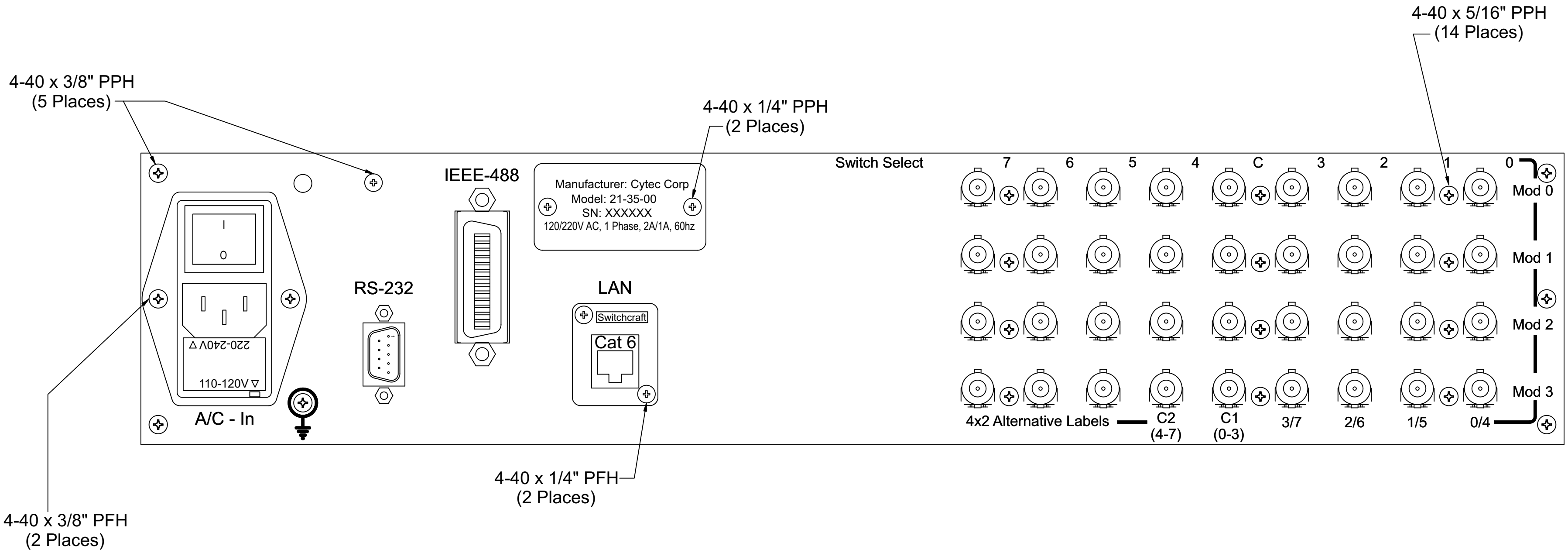
TOL. +/- 0.01" U.O.S.

Date: 9/27/22 CAD File: 21-30-21.dxf Drawn By: PV

Size: A Title: CXAR/16 Mainframe Rear Panel

Comments:

Drawing Number: 21-30-21




| Cytec P/N | Description | Qty |
|---|---|---|
| FAS1014 | 4-40 x 3/8" Phillips Panhead | 5 |
| FAS1016 | 4-40 x 5/16" Phillips Panhead | 14 |
| FAS1041 | 4-40 x 1/4" Phillips Flathead | 2 |
| FAS1081 | 4-40 x 3/8" Phillips Flathead | 2 |
| FAS1017 | 4-40 x 1/4" Phillips Panhead | 2 |

MAKE FROM ABL1002
.100" THICK ALUM.
TIME-SAVE IN-HOUSE
DEEP ETCH & CLEAR ANODIZE

| Deburred | | Initial | |
|---|---|---|---|

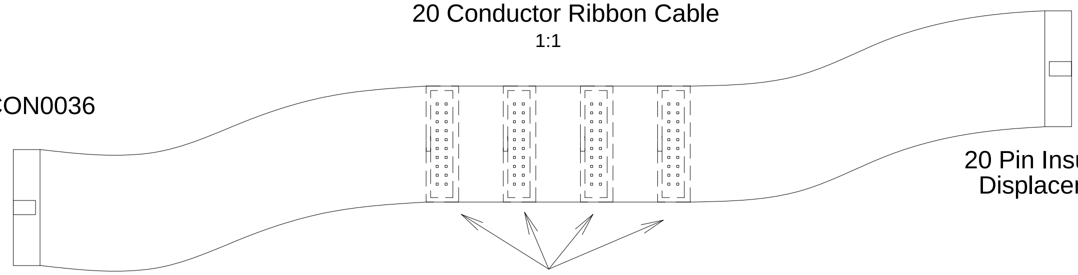
| | | | |
|---|---|---|---|
| - | 10/2/2024 | ST | Drawing Start |

CYTEC CORP

TOL. +/- 0.01" U.O.S.

| | | |
|---|---|---|
| Date: 10/2/2024 | CAD File: 21-35-21.dxf | Drawn By: ST |
| Size: A | Title: CXAR/32 MF Rear Panel w/ IF-xx | |
| Comments: ARP1168 | | Drawing Number: 21-35-21 |

0 0.5 1



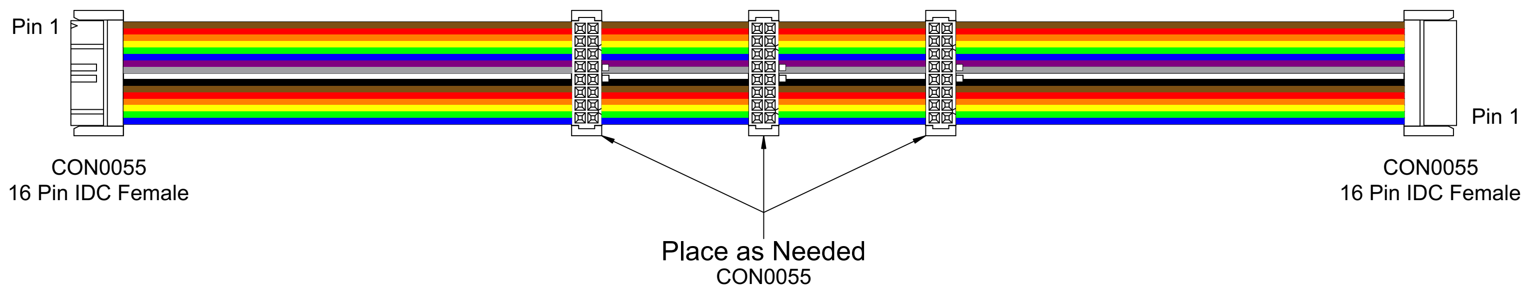

CYTEC CORP.

TOL. +/- .010" U.O.S.

| | | | | | |
|---|---|---|---|---|---|
| DATE: | 5/4/15 | CAD FILE: | 99-00-30.dcd | DRAWN BY: | PV |
| SIZE: C | TITLE: 20 Pin Drive Cable | | | | |
| COMMENTS: | | | | DRAWING NUMBER: | 99-00-30 |

# CBL0004

16 Pin Conductor Ribbon Cable
1:1




| | | | |
|---|---|---|---|
| - | 12/4/18 | PV | Drawing Start |

CYTEC CORP

TOL. +/- 0.01" U.O.S.

| | | | | | |
|---|---|---|---|---|---|
| Date: | 12/4/18 | CAD File: | 99-02-30.dxf | Drawn By: | PV |

Size: A

Title: 16 Pin Data Cable

Comments:

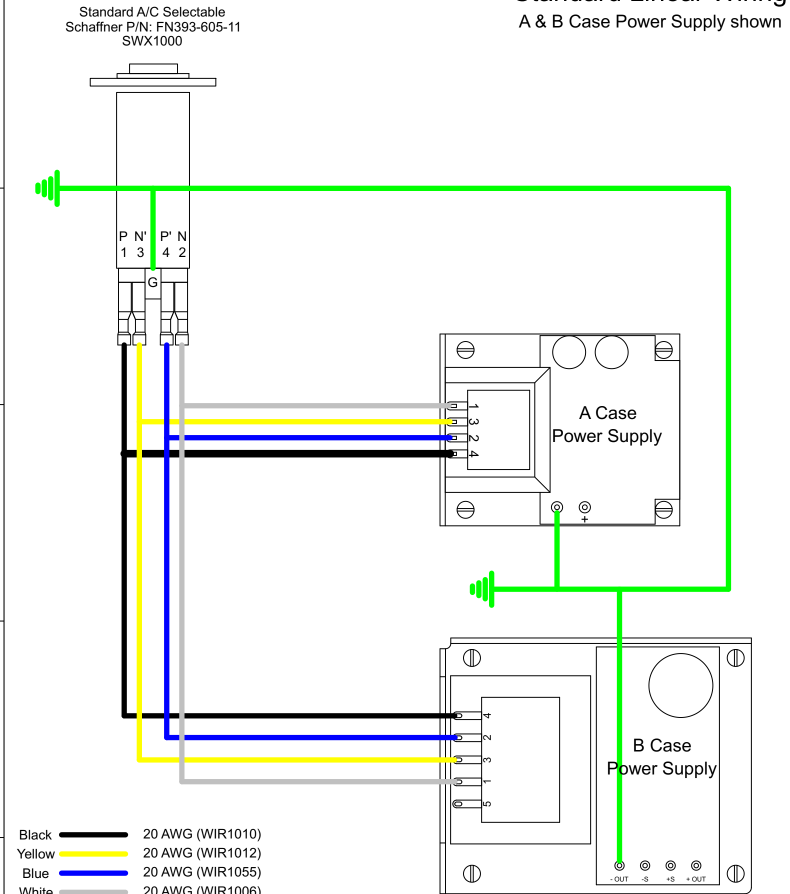
Drawing Number: 99-02-30

| | | | |
|---|---|---|---|
| Deburred | | Initial | |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|

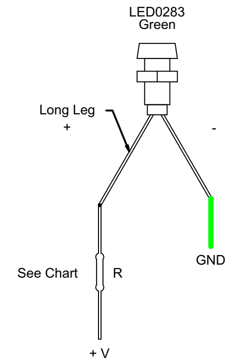
## Standard Linear Wiring

A & B Case Power Supply shown




Black 20 AWG (WIR1010)
Yellow 20 AWG (WIR1012)
Blue 20 AWG (WIR1055)
White 20 AWG (WIR1006)
Green 18 AWG (WIR1040)

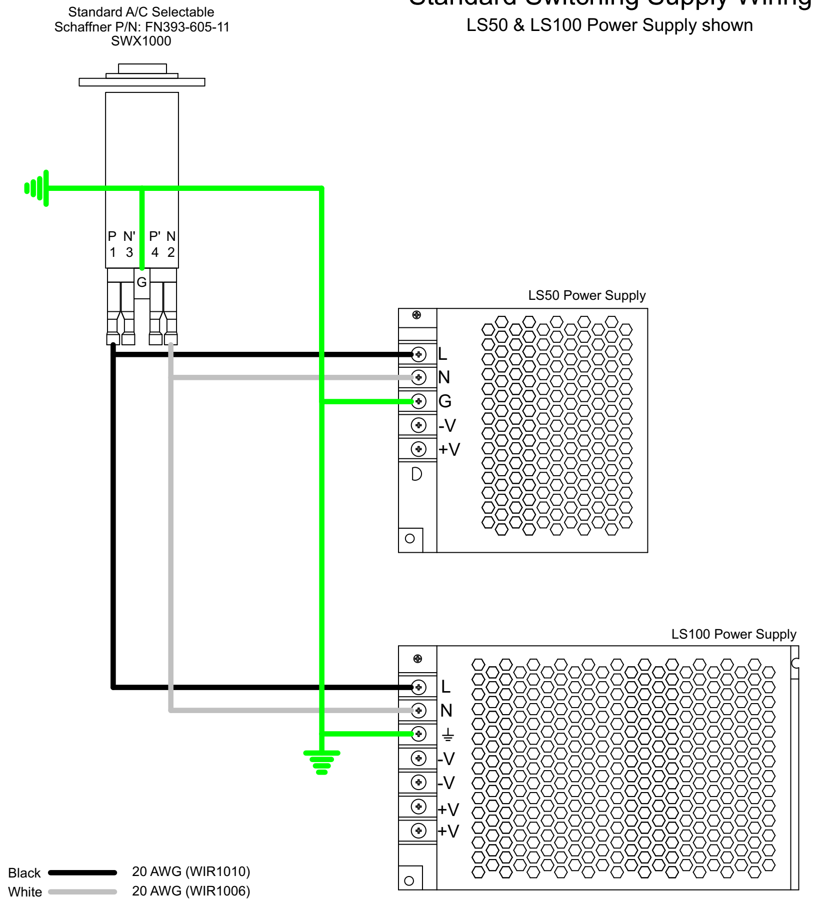
## Front Panel LED Wiring




| Voltage | Ohms | P/N |
|---|---|---|
| 2V | 10.7Ω | RES1066 |
| 2.5V | 100Ω | RES1085 |
| 3.3V | 200Ω | RES1141 |
| 5V | 390Ω | RES1027 |
| 12V | 1KΩ | RES1025 |
| 15V | 2KΩ | RES1049 |
| 24/28V | 5.1KΩ | RES1172 |

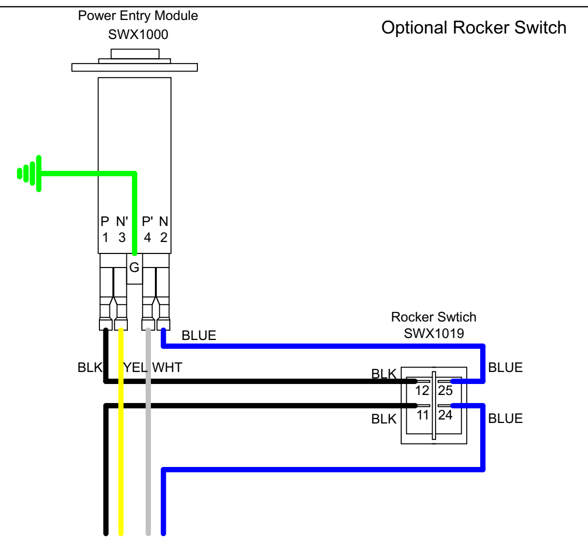
## Standard Switching Supply Wiring

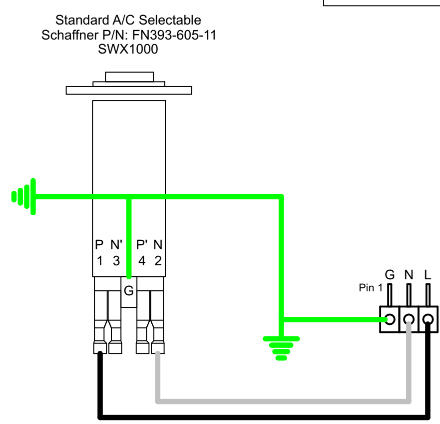
LS50 & LS100 Power Supply shown




Black 20 AWG (WIR1010)
White 20 AWG (WIR1006)
Green 18 AWG (WIR1040)

## Optional Rocker Switch




Black 20 AWG (WIR1010)
White 20 AWG (WIR1006)
Green 18 AWG (WIR1040)

| Rev | Date | By | Description |
|---|---|---|---|
| A | 1/27/20 | PV | Tied Gnd on Dual Supplies |
| - | 5/2/18 | PV | Drawing Start |

CYTEC CORP

TOL. +/- 0.01" U.O.S.

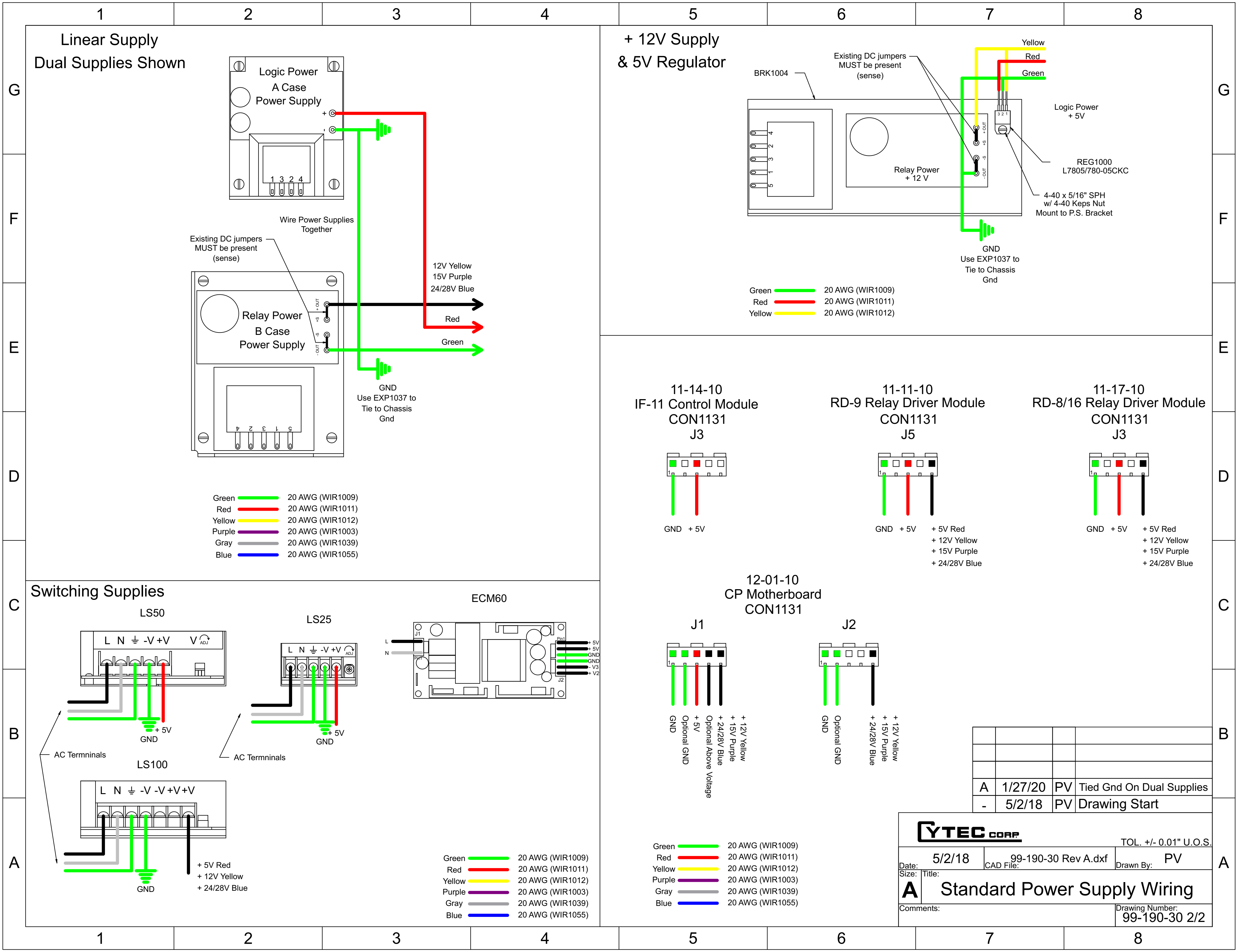
Date: 5/2/18 CAD File: 99-190-30 Rev A.dxf Drawn By: PV

Size: A Title: Standard Power Supply Wiring

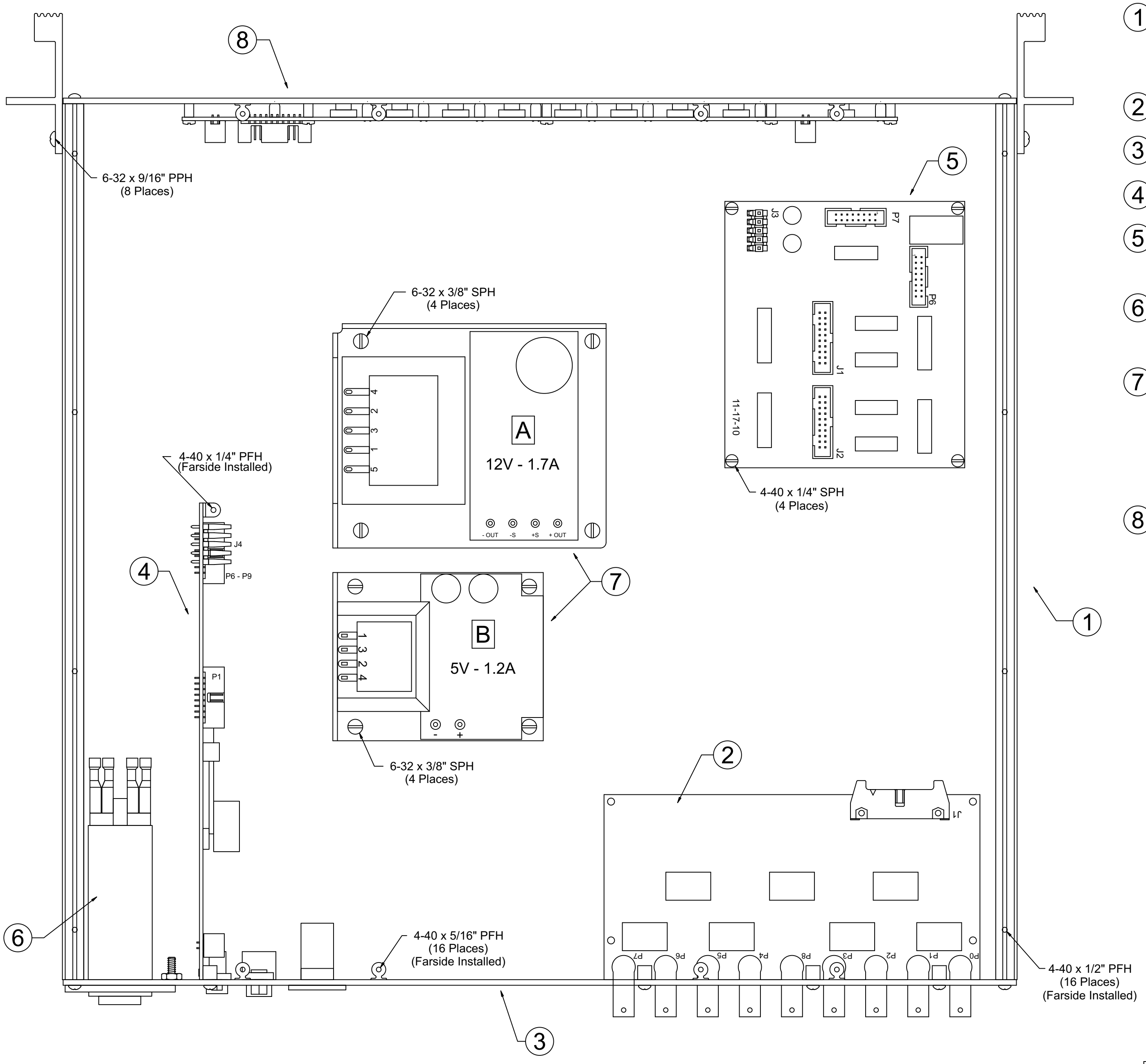
Comments:

Drawing Number: 99-190-30 1/2

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|



Linear Supply
Dual Supplies Shown
Logic Power
A Case
Power Supply
Wire Power Supplies
Together
Existing DC jumpers
MUST be present
(sense)
Relay Power
B Case
Power Supply
12V Yellow
15V Purple
24/28V Blue
Red
Green
GND
Use EXP1037 to
Tie to Chassis
Gnd
Green 20 AWG (WIR1009)
Red 20 AWG (WIR1011)
Yellow 20 AWG (WIR1012)
Purple 20 AWG (WIR1003)
Gray 20 AWG (WIR1039)
Blue 20 AWG (WIR1055)
+ 12V Supply
& 5V Regulator
BRK1004
Existing DC jumpers
MUST be present
(sense)
Yellow
Red
Green
Logic Power
+ 5V
REG1000
L7805/780-05CKC
4-40 x 5/16" SPH
w/ 4-40 Keps Nut
Mount to P.S. Bracket
Relay Power
+ 12 V
GND
Use EXP1037 to
Tie to Chassis
Gnd
Green 20 AWG (WIR1009)
Red 20 AWG (WIR1011)
Yellow 20 AWG (WIR1012)
11-14-10
IF-11 Control Module
CON1131
J3
GND + 5V
11-11-10
RD-9 Relay Driver Module
CON1131
J5
GND + 5V + 5V Red
+ 12V Yellow
+ 15V Purple
+ 24/28V Blue
11-17-10
RD-8/16 Relay Driver Module
CON1131
J3
GND + 5V + 5V Red
+ 12V Yellow
+ 15V Purple
+ 24/28V Blue
Switching Supplies
LS50
L N -V +V V ADJ
+ 5V
GND
AC Termninals
LS25
L N -V +V ADJ
+ 5V
GND
AC Termninals
LS100
L N -V -V +V +V
GND
+ 5V Red
+ 12V Yellow
+ 24/28V Blue
ECM60
L
N
J1
J2
Pin1
+ 5V
+ 5V
GND
GND
- V3
+ V2
12-01-10
CP Motherboard
CON1131
J1
GND
Optional GND
+ 5V
Optional Above Voltage
+ 24/28V Blue
+ 15V Purple
+ 12V Yellow
J2
GND
Optional GND
+ 24/28V Blue
+ 15V Purple
+ 12V Yellow
Green 20 AWG (WIR1009)
Red 20 AWG (WIR1011)
Yellow 20 AWG (WIR1012)
Purple 20 AWG (WIR1003)
Gray 20 AWG (WIR1039)
Blue 20 AWG (WIR1055)
A 1/27/20 PV Tied Gnd On Dual Supplies
- 5/2/18 PV Drawing Start
CYTEC CORP
TOL. +/- 0.01" U.O.S.
Date: 5/2/18
CAD File: 99-190-30 Rev A.dxf
Drawn By: PV
Size: A
Title: Standard Power Supply Wiring
Comments:
Drawing Number: 99-190-30 2/2




1. Chassis Assembly
   Fabrication: 21-30-140
   Assembly: 21-30-240
2. Switch Modules Per Order
3. CXAR/16 Rear Panel (21-30-21)
4. IF-11 Control Module (11-14-10)
5. RD-8/16 Relay Driver Module (11-17-10)
6. AC Selectable Power Entry (SWX1000)
7. Power Supplies:
   - A 12V - 1.7A (PWS0295)
   - B 5V - 1.2A (PWS0296)
8. 16 Channel Front Panel w/ Optional Pushbutton Manual Control

| Cytec P/N | Description | Qty |
|---|---|---|
| FAS1041 | 4-40 x 1/4" Phillips Flathead | 1 |
| FAS1022 | 4-40 x 1/4" Slotted Panhead | 4 |
| FAS1010 | 4-40 x 5/16" Phillips Flathead | 16 |
| FAS1006 | 4-40 x 1/2" Phillips Flathead | 16 |
| FAS1004 | 6-32 x 3/8" Slotted Panhead | 8 |
| FAS1007 | 6-32 x 9/16" Phillips Panhead | 8 |

| | | | |
|---|---|---|---|
| D | 10/4/22 | PV | Changed to Hemless Base |
| C | 9/16/20 | PV | Changed RD9 to RG8/16 |
| B | 6/8/18 | PV | Updated Power Wiring |
| A | 1/18/16 | PV | Added Control Wiring Diagram |
| - | 11/13/13 | PV | Drawing Start |

CYTEC CORP

TOL. +/- 0.01" U.O.S.

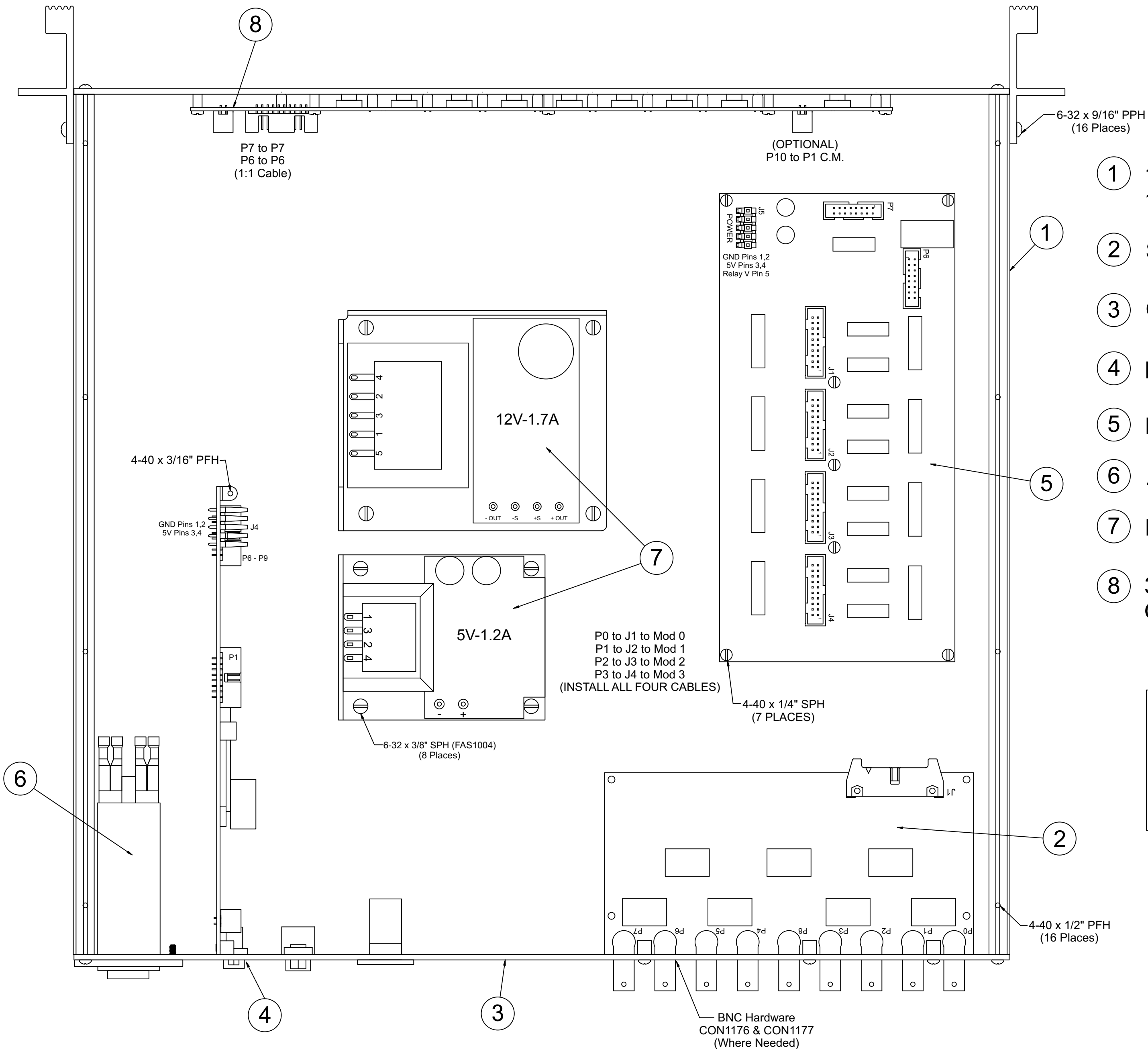
Date: 11/13/13 | CAD File: 21-30-00 Rev D.dxf | Drawn By: PV

Size: A | Title: CXAR/16-MF Mainframe Chassis

Comments:

Drawing Number: 21-30-00

Make From TBP1017




1. 14-20-20 CHASSIS
   15.6" DEEP x 19" WIDE x 3.5" HIGH
2. SWITCH MODULES PER ORDER
3. CXAR/32 REAR PANEL
4. IF-9, IF-11 CONTROL INTERFACE
5. RD-9 RELAY DRIVER BOARD
6. AC SELETABLE POWER ENTRY
7. POWER SUPPLIES
8. 32 CHANNEL FRONT PANEL
   OPT. PUSHBUTTON MANUAL CONTROL

| Cytec P/N | Description | Qty |
|---|---|---|
| FAS1022 | 4-40 x 1/4" Slotted Panhead | 7 |
| FAS1006 | 4-40 x 1/2" Phillips Flathead | 16 |
| FAS1004 | 6-32 x 3/8" Slotted Panhead | 8 |
| FAS1007 | 6-32 x 9/16" Phillips Panhead | 8 |
| FAS1040 | 4-40 x 3/16" Phillips Flathead | 1 |

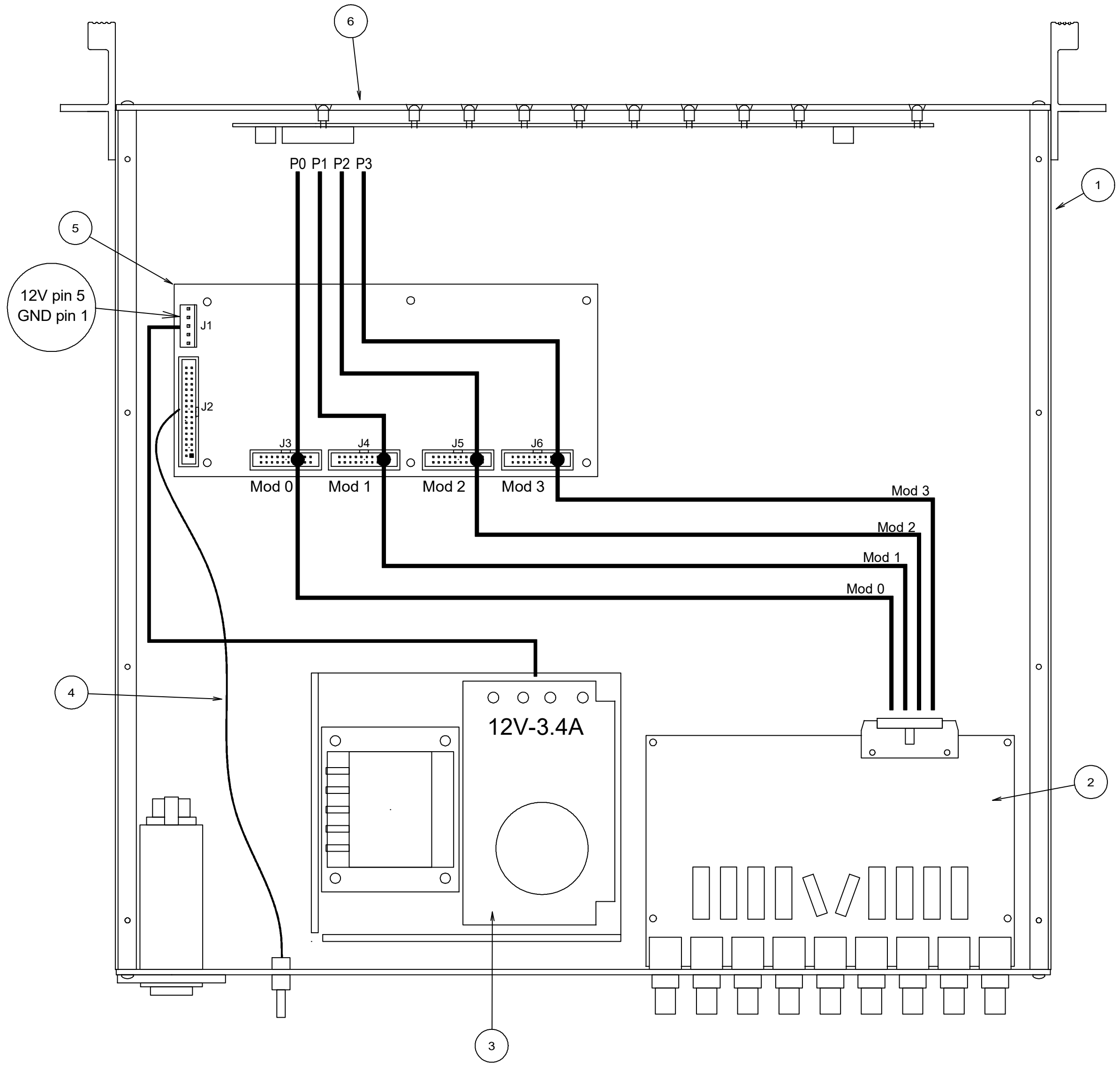
| | | | |
|---|---|---|---|
| A | 5/10/2016 | PV | Updated for IF-11 Control |
| - | 11/8/2013 | PV | Drawing Start |

CYTEC CORP

TOL. +/- 0.01" U.O.S.

| | | |
|---|---|---|
| Date: 11/8/2013 | CAD File: 21-35-00 Rev A.dxf | Drawn By: PV |
| Size: A | Title: CXAR/32 - MAINFRAME | |
| Comments: IF-9, or IF-11 Control | | Drawing Number: 21-35-00 |

| | | | |
|---|---|---|---|
| Deburred | | Initial | |




1. 14-20-20 CHASSIS ASSY
2. SWITCH MODULES PER ORDER
3. 12V POWER SUPPLY
4. 34 PIN CABLE ASSY
5. 11-08-11 CM/32-VHP MODULE
6. 32 CHANNEL FRONT PANEL

2-190-5B POWER WIRING

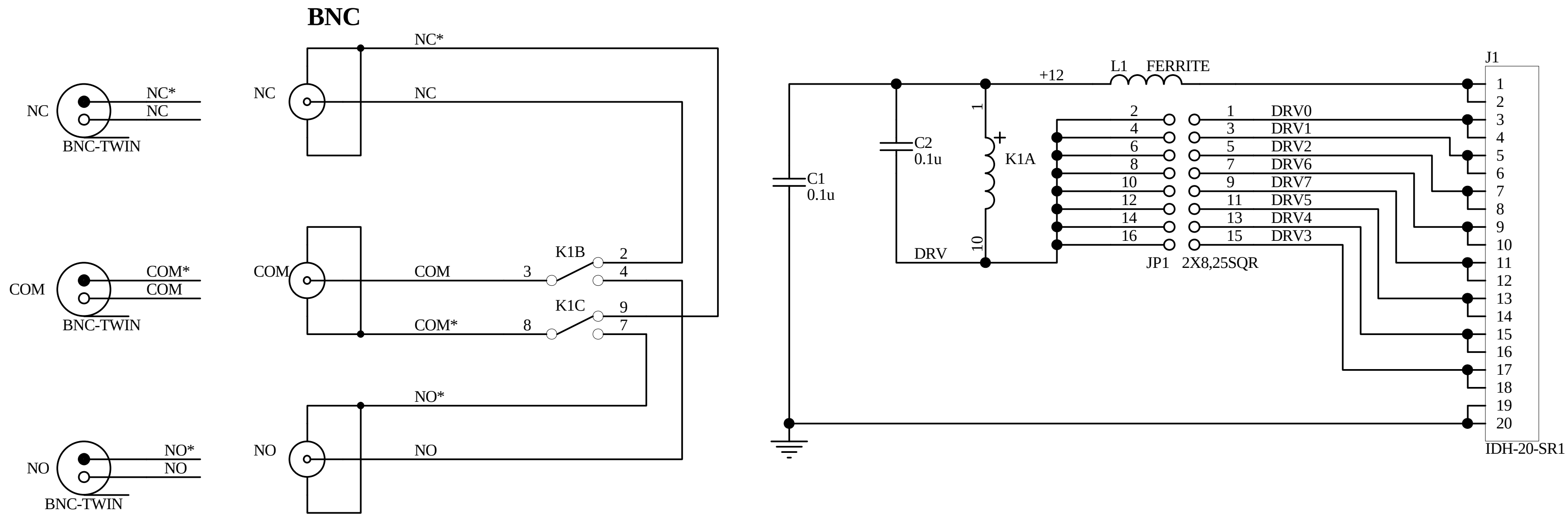
6-015-3 MESA INTERFACE ASSY

99-00-30 DRIVE/CONTROL WIRING

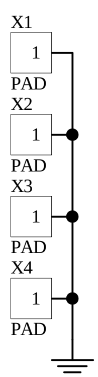
99-51-30 34 PIN CABLE ASSY

CYTEC CORP.

| | | | |
|---|---|---|---|
| DATE: 10/25/11 | CAD FILE: 21-23-00.dcd | | DRAWN BY: DR |
| SIZE: D | TITLE: CXAR/32 - E-PS | | |
| COMMENTS: | | | DRAWING NUMBER: 21-23-00 |




MOUNTING PADS

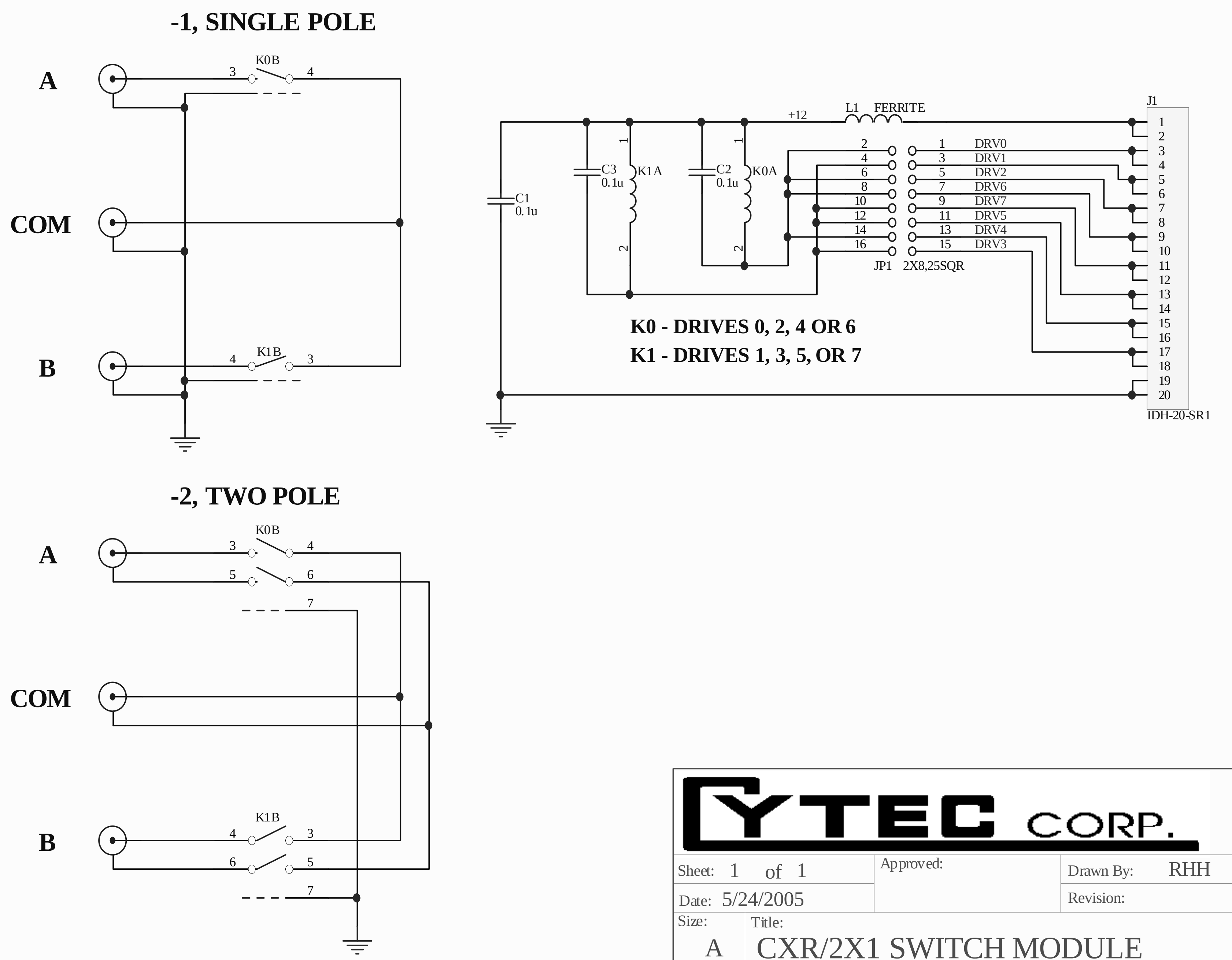



REVISED FROM:

CYTEC CORP.

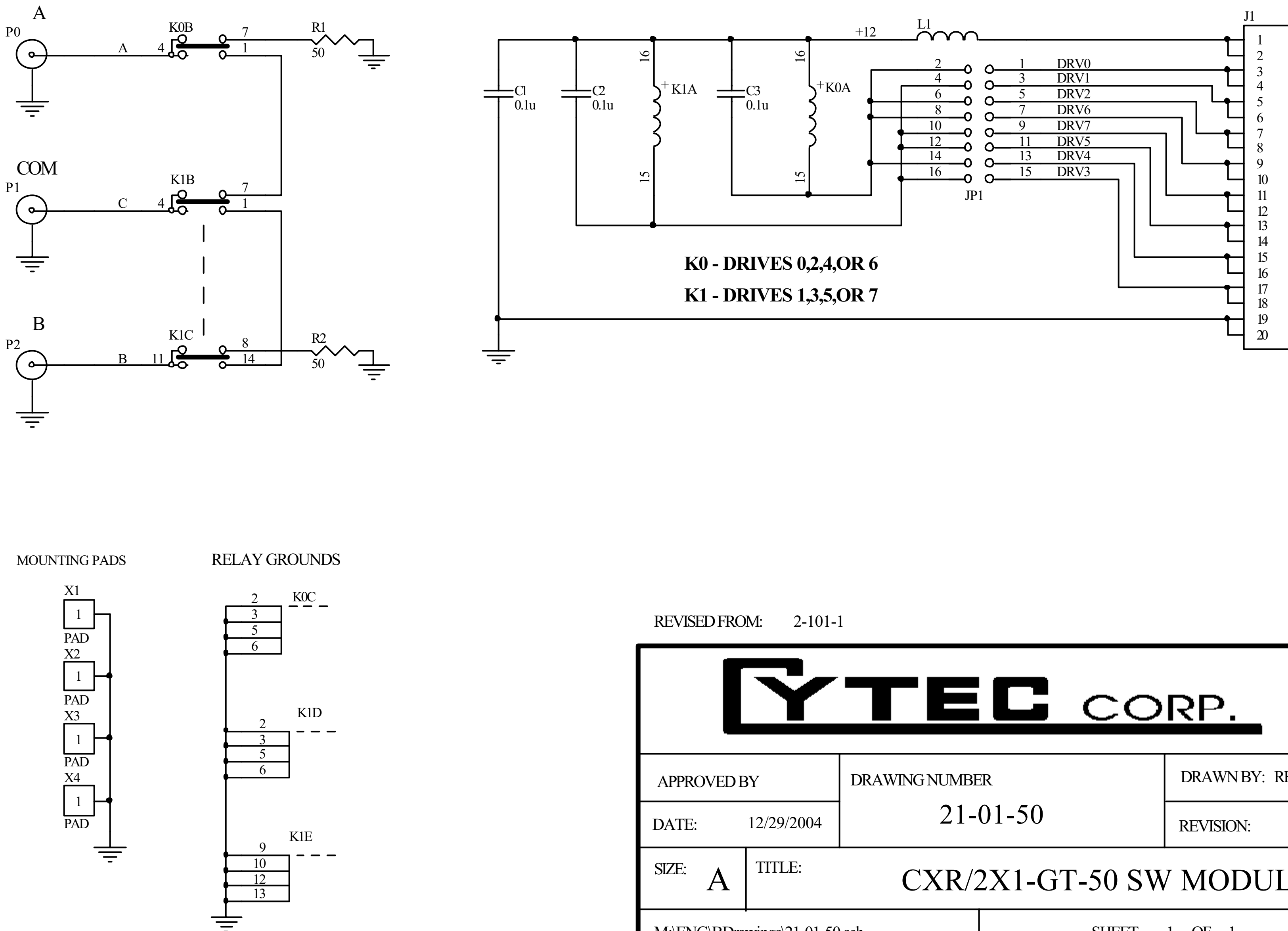
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
| DATE: 27-Jan-2004 | 21-00-50 | REVISION: |
| SIZE: A | TITLE: CXR/2X1-C2 | |

M:\ENG\BDrawings\2100_99se\21-00-50.ddb - Documents\21-00-50.SCH SHEET 1 OF 1




CYTEC CORP.

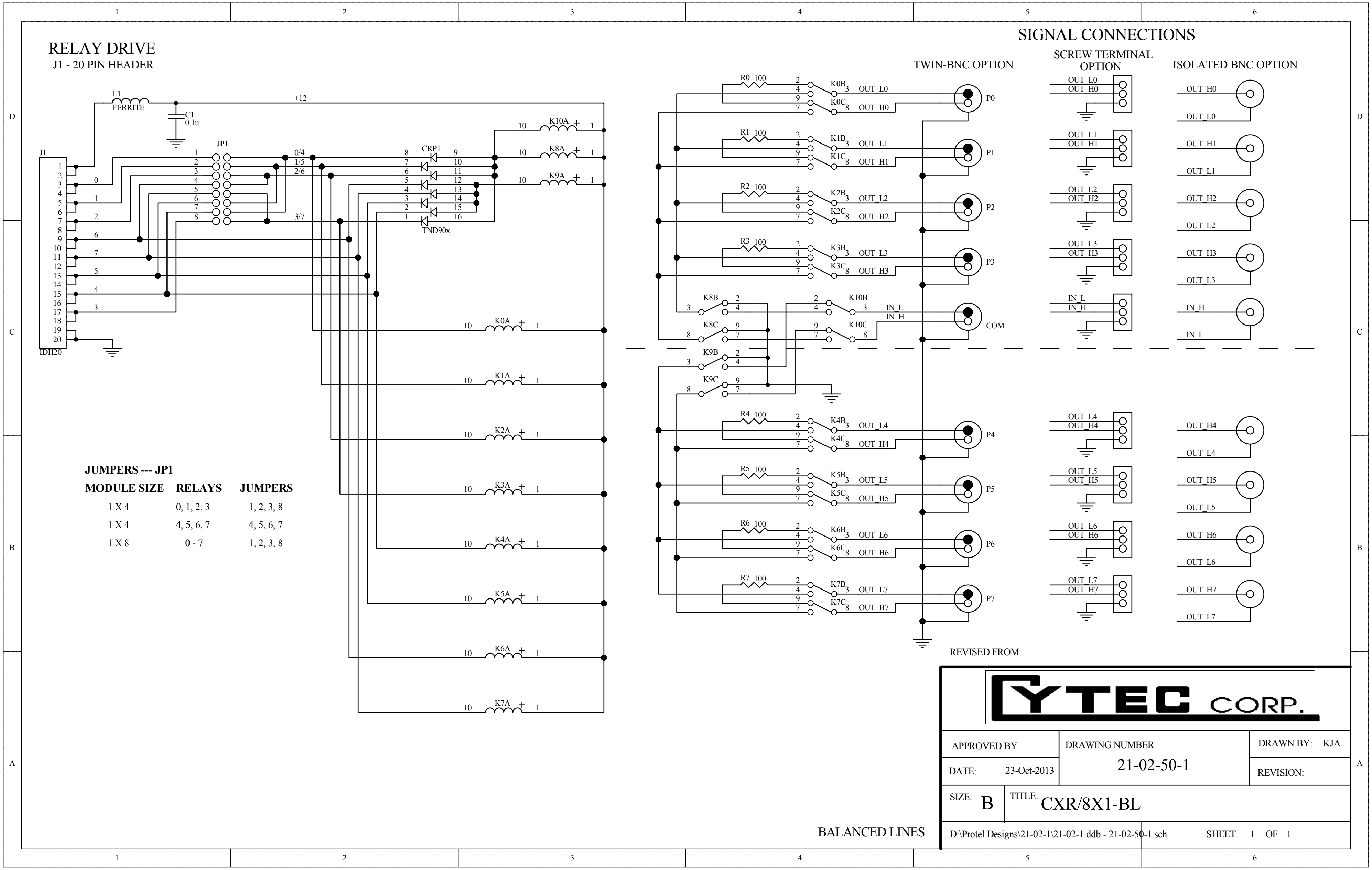
| | | | |
|---|---|---|---|
| Sheet: 1 of 1 | Approved: | Drawn By: RHH | |
| Date: 5/24/2005 | | Revision: | |
| Size: A | Title: CXR/2X1 SWITCH MODULE | | |
| Comments: CT/2 MODULE | | Number: 2-101-1 | |



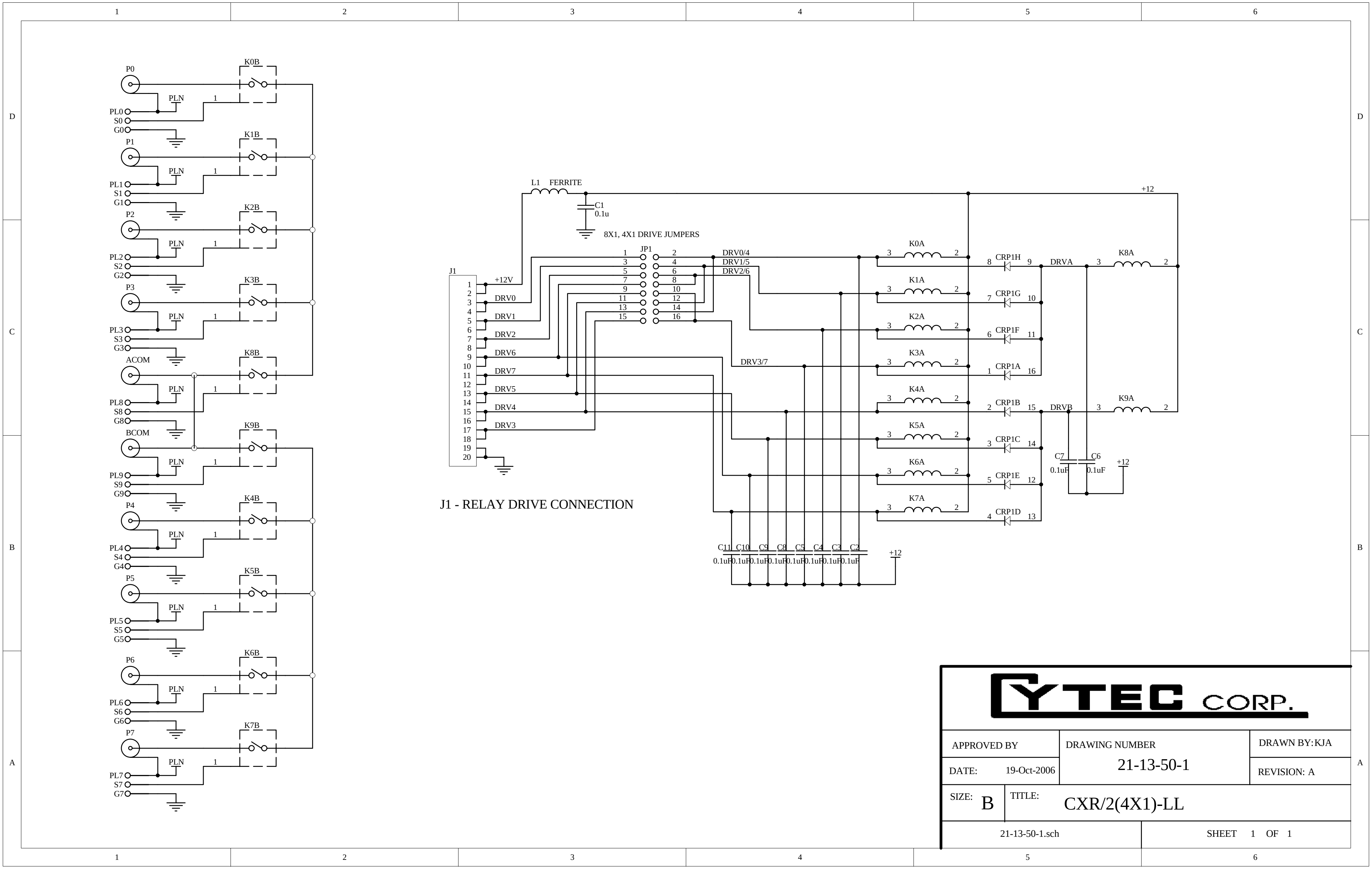

REVISED FROM: 2-101-1

CYTEC CORP.

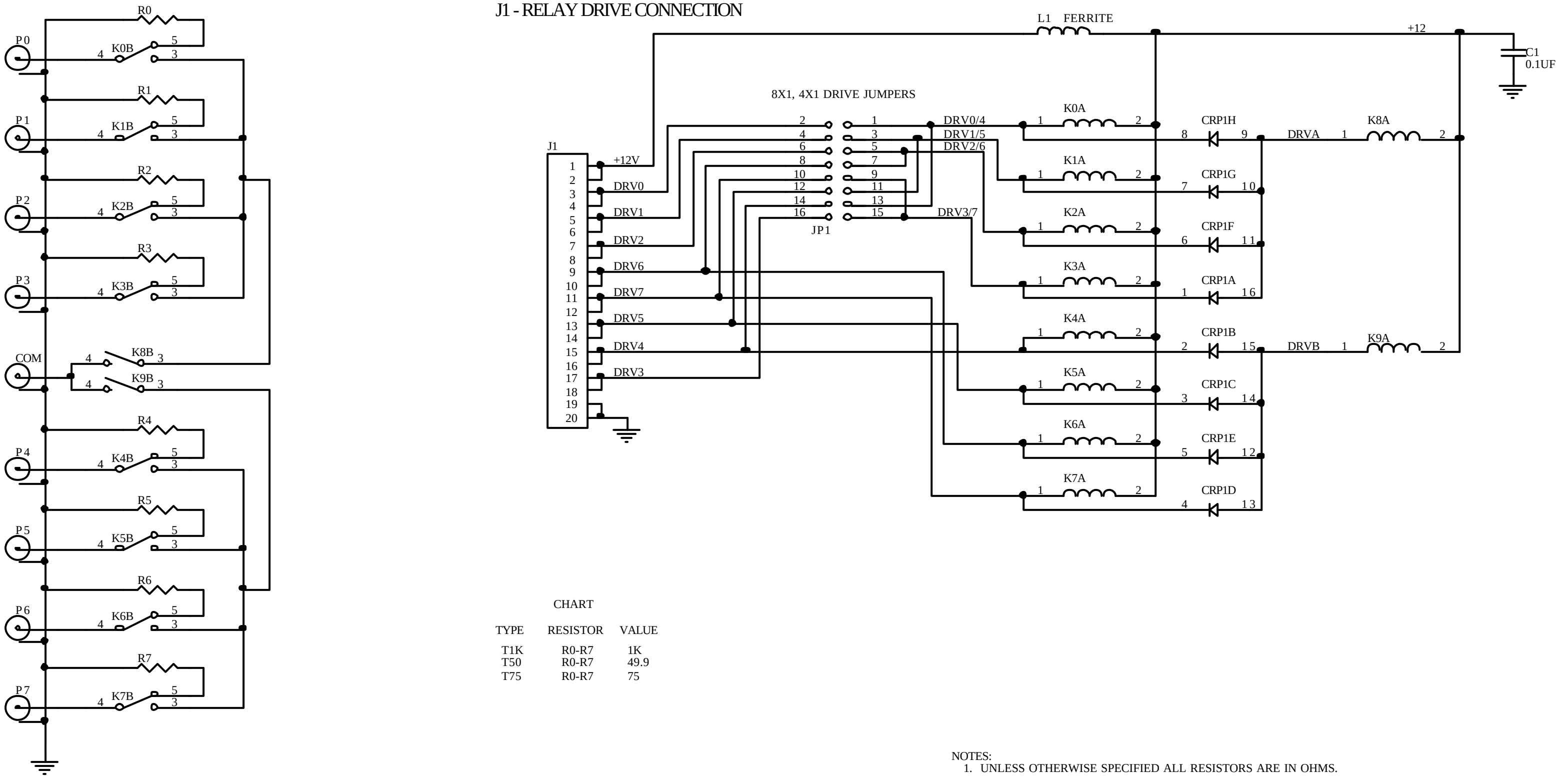
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
| DATE: 12/29/2004 | 21-01-50 | REVISION: |
| SIZE: A | TITLE: CXR/2X1-GT-50 SW MODULE | |
| M:\ENG\BDrawings\21-01-50.sch | SHEET 1 OF 1 | |



RELAY DRIVE
J1 - 20 PIN HEADER
L1 FERRITE
C1 0.1u
+12
JP1
CRP1
TND90x
IDH20
K10A
K8A
K9A
K0A
K1A
K2A
K3A
K4A
K5A
K6A
K7A
SIGNAL CONNECTIONS
TWIN-BNC OPTION
SCREW TERMINAL OPTION
ISOLATED BNC OPTION
JUMPERS --- JP1
MODULE SIZE RELAYS JUMPERS
1 X 4 0, 1, 2, 3 1, 2, 3, 8
1 X 4 4, 5, 6, 7 4, 5, 6, 7
1 X 8 0 - 7 1, 2, 3, 8
BALANCED LINES
REVISED FROM:
CYTEC CORP.
APPROVED BY
DRAWING NUMBER
21-02-50-1
DRAWN BY: KJA
DATE: 23-Oct-2013
REVISION:
SIZE: B
TITLE: CXR/8X1-BL
D:\Protel Designs\21-02-1\21-02-1.ddb - 21-02-50-1.sch
SHEET 1 OF 1


J1 - RELAY DRIVE CONNECTION
8X1, 4X1 DRIVE JUMPERS
L1 FERRITE
C1 0.1u
+12V
DRV0
DRV1
DRV2
DRV6
DRV7
DRV5
DRV4
DRV3
DRV0/4
DRV1/5
DRV2/6
DRV3/7
DRVA
DRVB
JP1
K0A
K1A
K2A
K3A
K4A
K5A
K6A
K7A
K8A
K9A
CRP1H
CRP1G
CRP1F
CRP1A
CRP1B
CRP1C
CRP1E
CRP1D
C7 0.1uF
C6 0.1uF
C11 C10 C9 C8 C5 C4 C3 C2
+12
P0
P1
P2
P3
P4
P5
P6
P7
ACOM
BCOM
K0B
K1B
K2B
K3B
K8B
K9B
K4B
K5B
K6B
K7B
PLN
PL0 S0 G0
PL1 S1 G1
PL2 S2 G2
PL3 S3 G3
PL8 S8 G8
PL9 S9 G9
PL4 S4 G4
PL5 S5 G5
PL6 S6 G6
PL7 S7 G7
CYTEC CORP.
APPROVED BY
DRAWING NUMBER
21-13-50-1
DRAWN BY:KJA
DATE: 19-Oct-2006
REVISION: A
SIZE: B
TITLE: CXR/2(4X1)-LL
21-13-50-1.sch
SHEET 1 OF 1

J1 - RELAY DRIVE CONNECTION




CHART

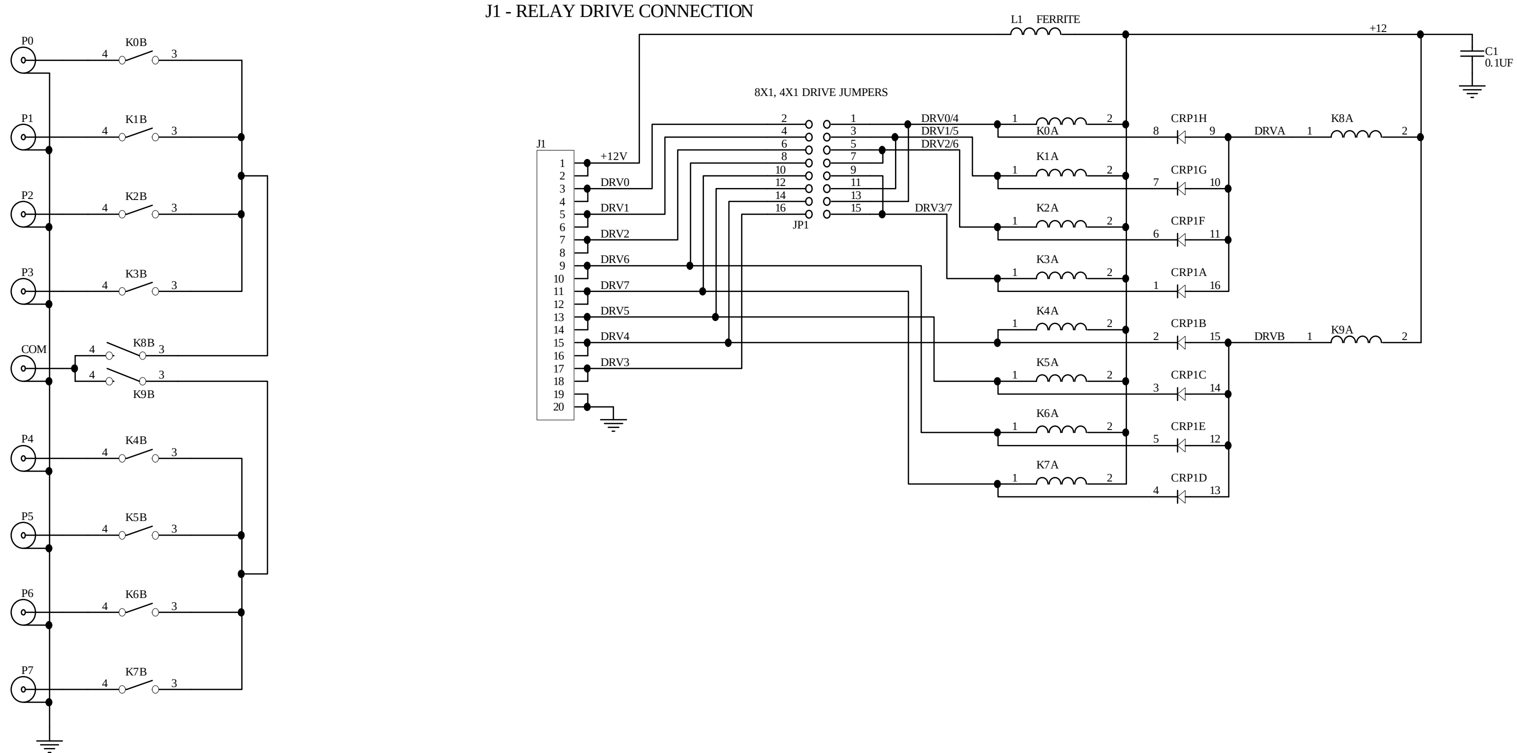
| TYPE | RESISTOR | VALUE |
|---|---|---|
| T1K | R0-R7 | 1K |
| T50 | R0-R7 | 49.9 |
| T75 | R0-R7 | 75 |

NOTES:
1. UNLESS OTHERWISE SPECIFIED ALL RESISTORS ARE IN OHMS.

REVISED FROM:2-442-2

CYTEC CORP.

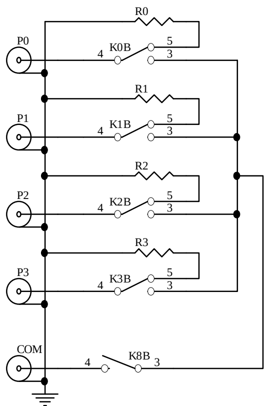
| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
|---|---|---|
| DATE: 4/14/2005 | 21-14-50 | REVISION: |
| SIZE: A3 | TITLE: CXR/8X1-1T SWITCH MODULE | |
| 21-14-50 | | SHEET 1 OF 1 |



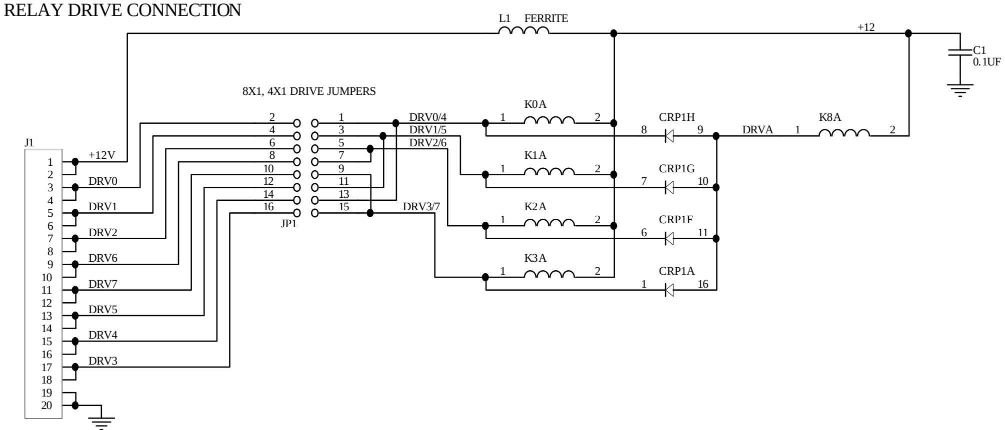

REVISED FROM:2-442-2

CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
| DATE: 4/14/2005 | 21-14-51 | REVISION: |
| SIZE: A3 | TITLE: CXR/8X1-1S SWITCH MODULE | |
| 21-14-51 | | SHEET 1 OF 1 |




## J1 - RELAY DRIVE CONNECTION




CHART

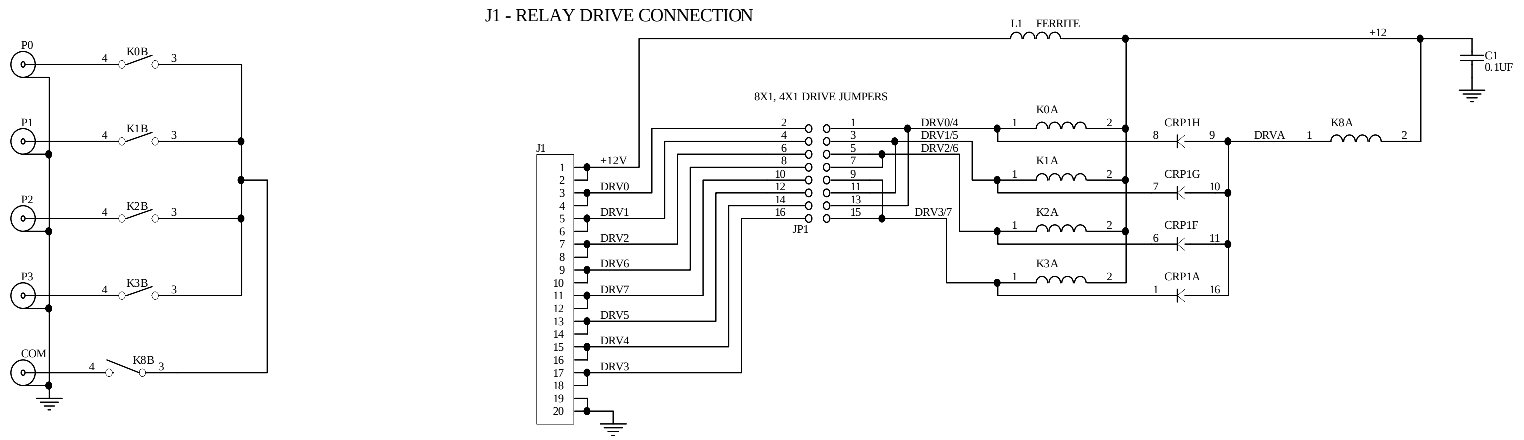
| TYPE | RESISTOR | VALUE |
|---|---|---|
| T1K | R0-R3 | 1K |
| T50 | R0-R3 | 49.9 |
| T75 | R0-R3 | 75 |

NOTES:
1. UNLESS OTHERWISE SPECIFIED ALL RESISTORS ARE IN OHMS.

REVISED FROM:2-442-2

CYTEC CORP.

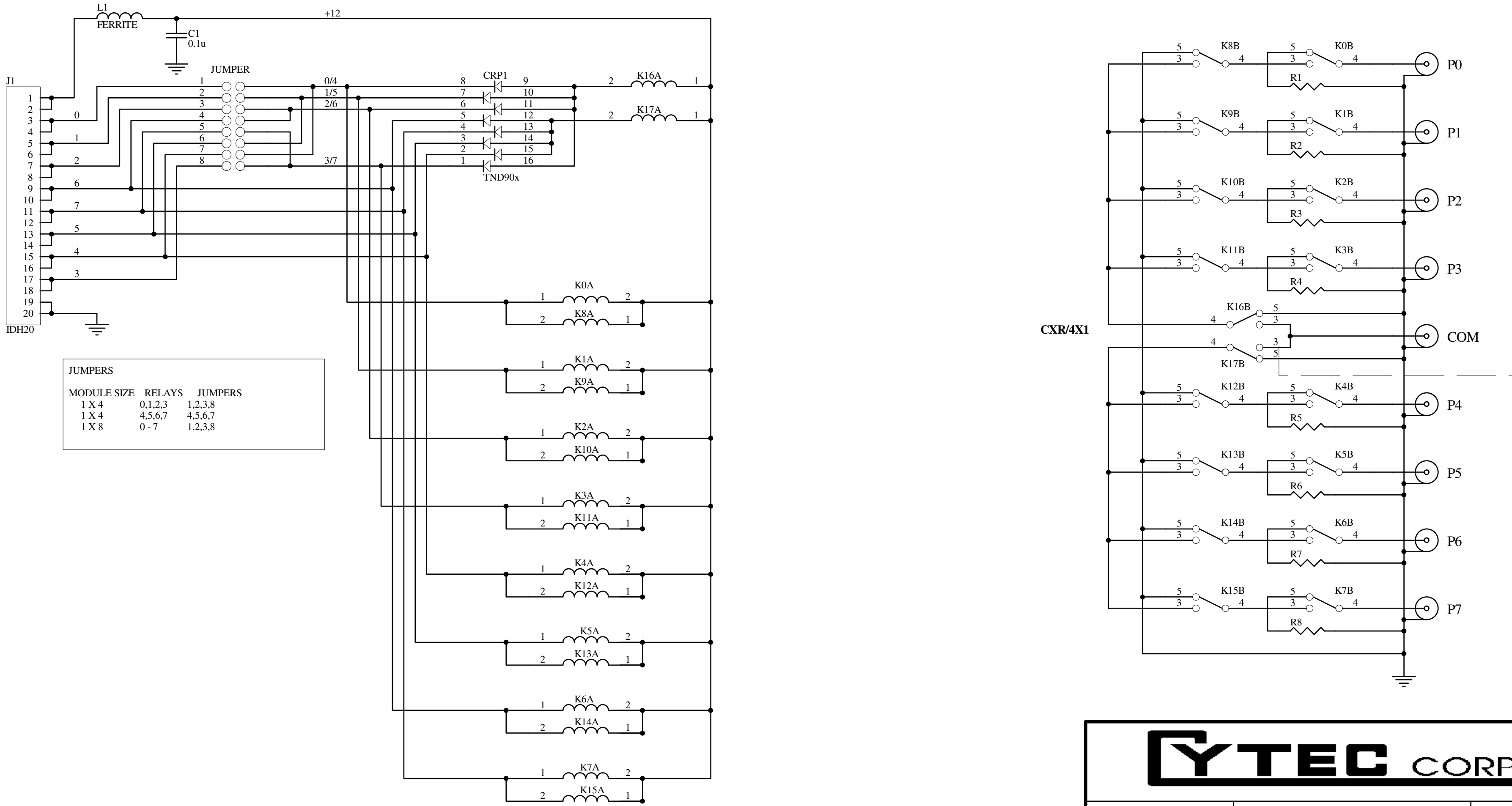
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER 21-14-52 | DRAWN BY: RHB |
| DATE: 4/14/2005 | | REVISION: |
| SIZE: A3 | TITLE: CXR/4X1-1T SWITCH MODULE | |
| **21-14-52** | | SHEET 1 OF 1 |




REVISED FROM:2-442-2

CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
| DATE: 4/14/2005 | 21-14-53 | REVISION: |
| SIZE: A3 | TITLE: CXR/4X1-1S SWITCH MODULE | |
| 21-14-53 | | SHEET 1 OF 1 |



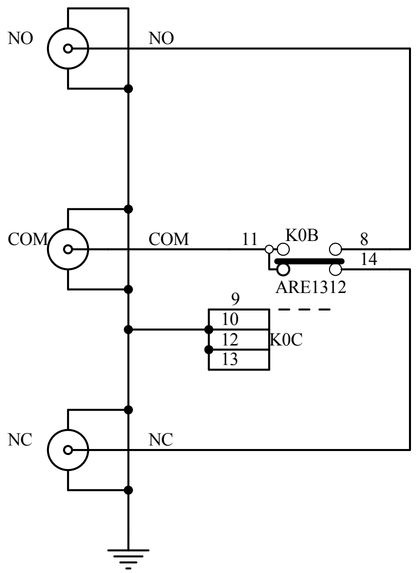

JUMPERS

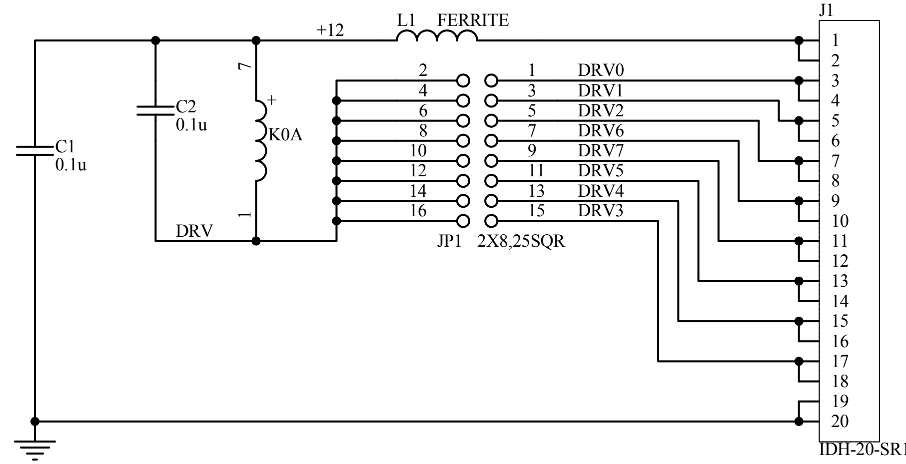
| MODULE SIZE | RELAYS | JUMPERS |
|---|---|---|
| 1 X 4 | 0,1,2,3 | 1,2,3,8 |
| 1 X 4 | 4,5,6,7 | 4,5,6,7 |
| 1 X 8 | 0 - 7 | 1,2,3,8 |

CYTEC CORP.

| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
|---|---|---|
| DATE: 2-Feb-2018 | 21-24-50 | REVISION: |
| SIZE: B | TITLE: CXR/8x1-1H SWITCH MODULE | |
| 21-24-50.SCH | SHEET 1 OF 1 | |

**BNC**

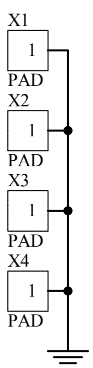



**Place Single Jumper Across JP1 At Desired Drive Location To Actuate Relay**

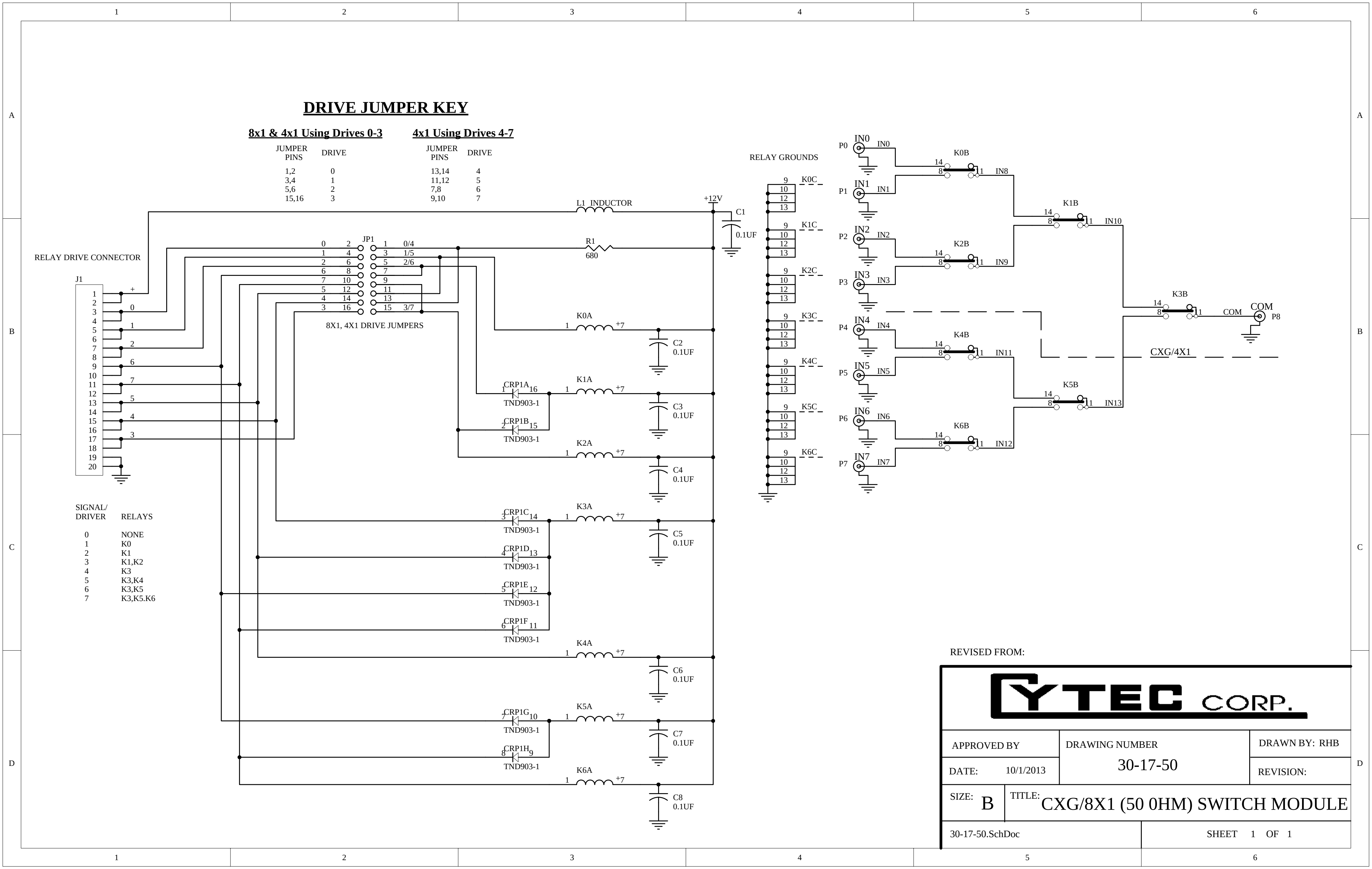
**Drive Numbers Are Labeled On PCB**

MOUNTING PADS




CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER 30-10-50-1 | DRAWN BY: KJA |
| DATE: 1-Oct-2013 | | REVISION: |
| SIZE: A | TITLE: CXR/2X1-75 SW MODULE | |
| | SHEET 1 OF 1 | |


DRIVE JUMPER KEY
8x1 & 4x1 Using Drives 0-3
JUMPER PINS DRIVE
1,2 0
3,4 1
5,6 2
15,16 3
4x1 Using Drives 4-7
JUMPER PINS DRIVE
13,14 4
11,12 5
7,8 6
9,10 7
RELAY DRIVE CONNECTOR
J1
JP1
8X1, 4X1 DRIVE JUMPERS
0/4
1/5
2/6
3/7
L1 INDUCTOR
+12V
C1
0.1UF
R1
680
K0A
K1A
K2A
K3A
K4A
K5A
K6A
CRP1A
CRP1B
CRP1C
CRP1D
CRP1E
CRP1F
CRP1G
CRP1H
TND903-1
C2 0.1UF
C3 0.1UF
C4 0.1UF
C5 0.1UF
C6 0.1UF
C7 0.1UF
C8 0.1UF
SIGNAL/ DRIVER RELAYS
0 NONE
1 K0
2 K1
3 K1,K2
4 K3
5 K3,K4
6 K3,K5
7 K3,K5.K6
RELAY GROUNDS
K0C
K1C
K2C
K3C
K4C
K5C
K6C
P0 IN0
P1 IN1
P2 IN2
P3 IN3
P4 IN4
P5 IN5
P6 IN6
P7 IN7
K0B
K1B
K2B
K3B
K4B
K5B
K6B
IN8
IN9
IN10
IN11
IN12
IN13
COM
P8
CXG/4X1
REVISED FROM:
CYTEC CORP.
APPROVED BY
DRAWING NUMBER
30-17-50
DRAWN BY: RHB
DATE: 10/1/2013
REVISION:
SIZE: B
TITLE: CXG/8X1 (50 0HM) SWITCH MODULE
30-17-50.SchDoc
SHEET 1 OF 1

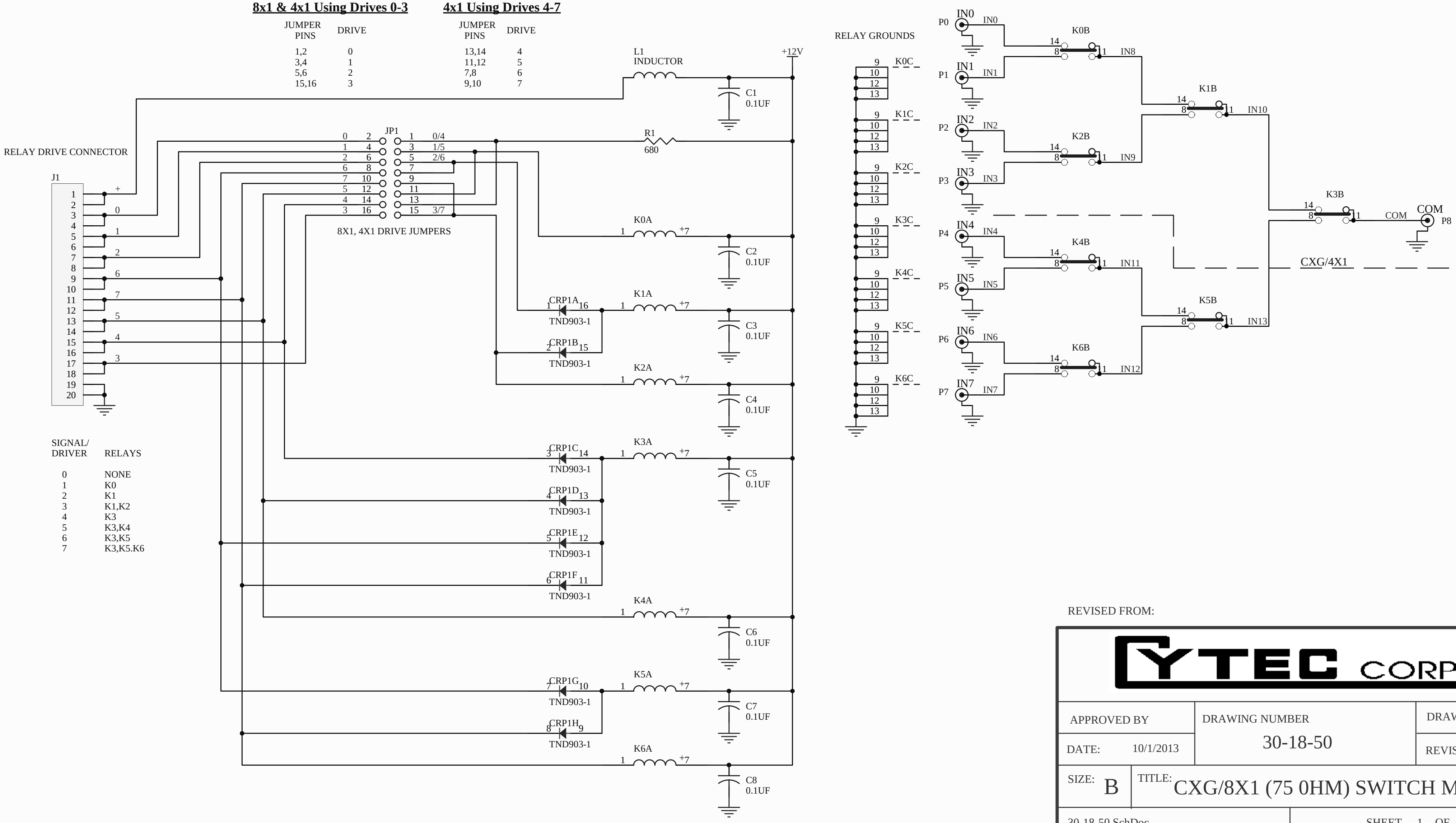
# DRIVE JUMPER KEY

**8x1 & 4x1 Using Drives 0-3**

| JUMPER PINS | DRIVE |
|---|---|
| 1,2 | 0 |
| 3,4 | 1 |
| 5,6 | 2 |
| 15,16 | 3 |

**4x1 Using Drives 4-7**

| JUMPER PINS | DRIVE |
|---|---|
| 13,14 | 4 |
| 11,12 | 5 |
| 7,8 | 6 |
| 9,10 | 7 |




| SIGNAL/ DRIVER | RELAYS |
|---|---|
| 0 | NONE |
| 1 | K0 |
| 2 | K1 |
| 3 | K1,K2 |
| 4 | K3 |
| 5 | K3,K4 |
| 6 | K3,K5 |
| 7 | K3,K5.K6 |

REVISED FROM:

CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
| DATE: 10/1/2013 | 30-18-50 | REVISION: |
| SIZE: B | TITLE: CXG/8X1 (75 0HM) SWITCH MODULE | |
| 30-18-50.SchDoc | | SHEET 1 OF 1 |

# DRIVE JUMPER KEY

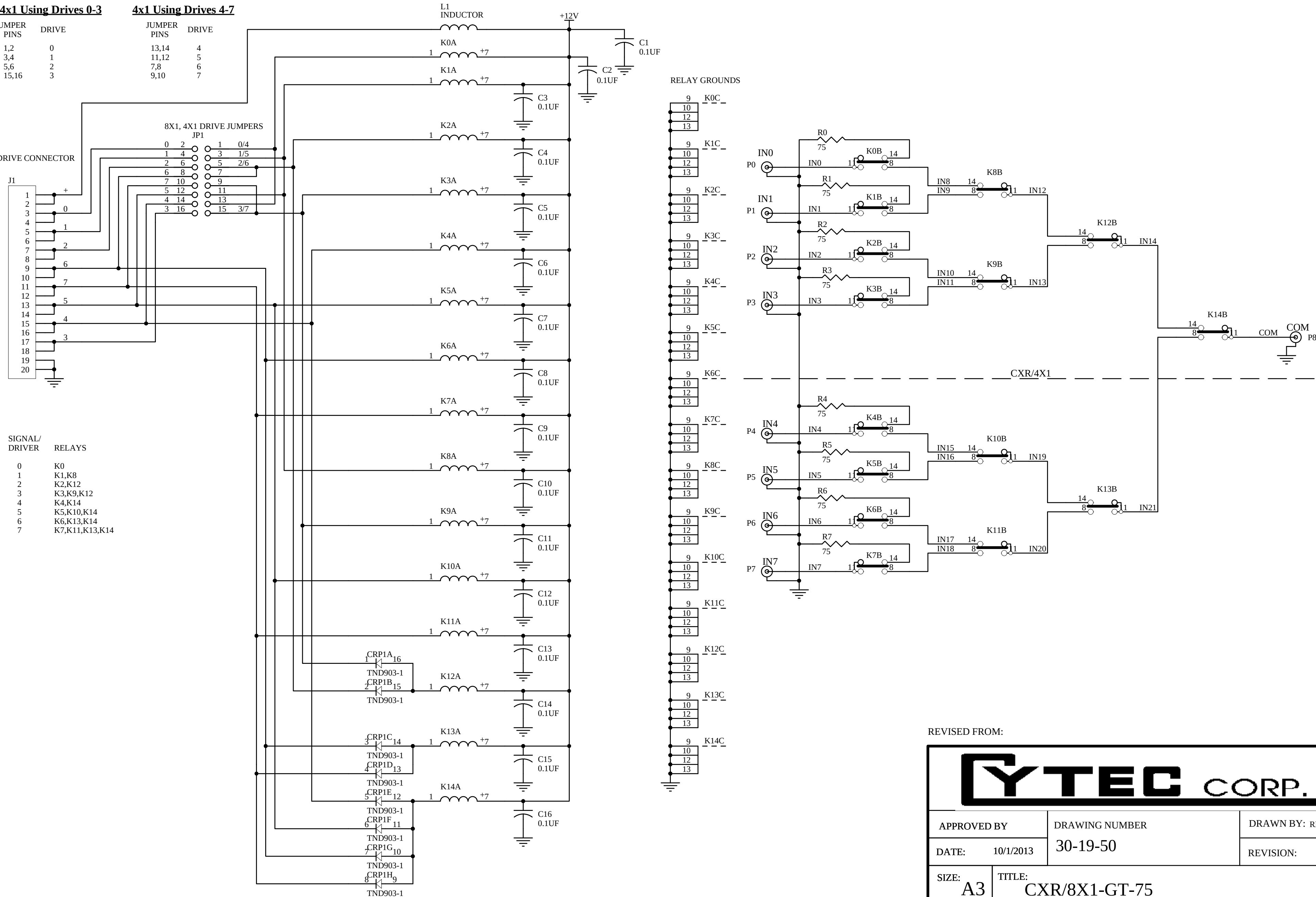
**8x1 & 4x1 Using Drives 0-3**

| JUMPER PINS | DRIVE |
|---|---|
| 1,2 | 0 |
| 3,4 | 1 |
| 5,6 | 2 |
| 15,16 | 3 |

**4x1 Using Drives 4-7**

| JUMPER PINS | DRIVE |
|---|---|
| 13,14 | 4 |
| 11,12 | 5 |
| 7,8 | 6 |
| 9,10 | 7 |

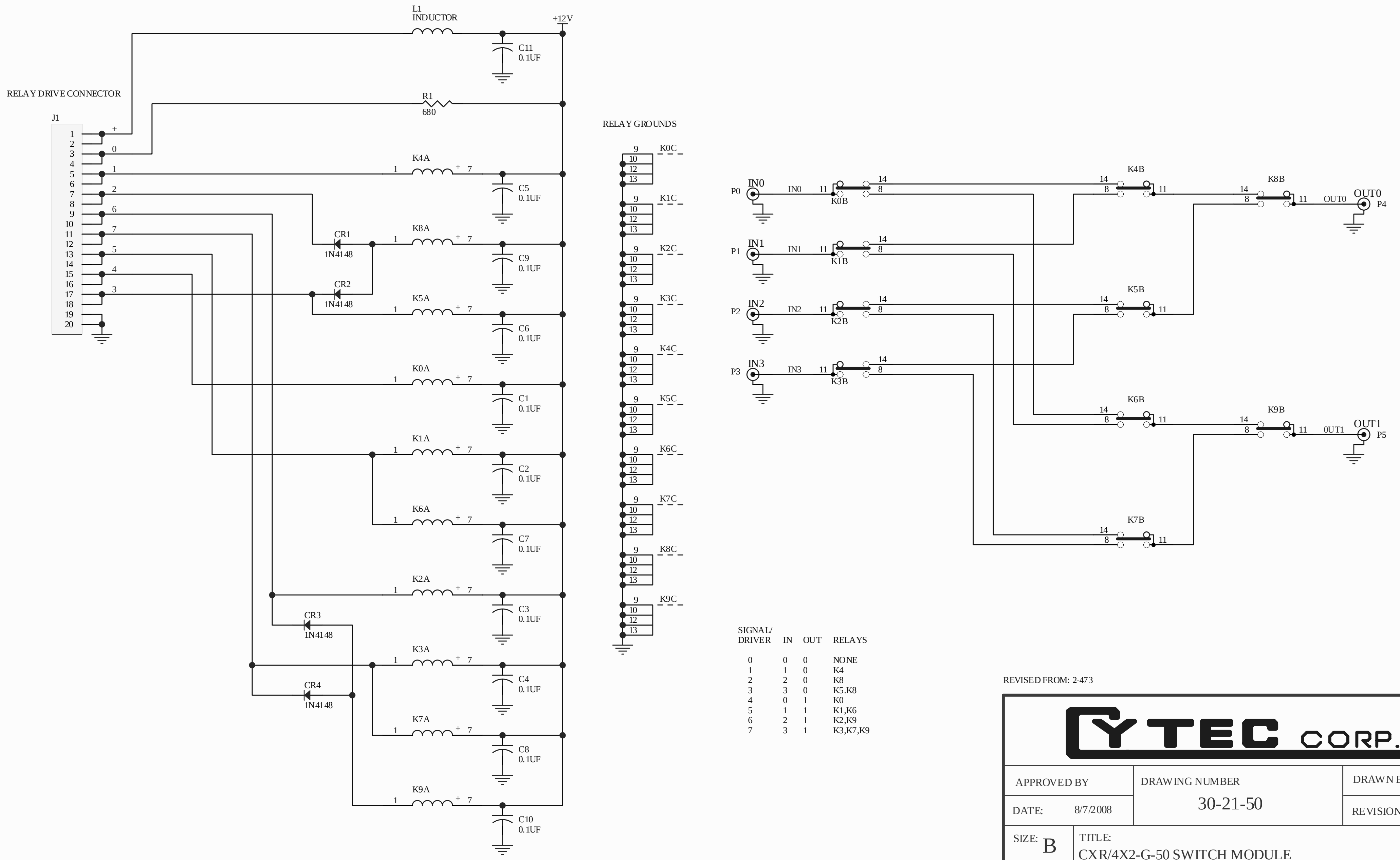
| SIGNAL/ DRIVER | RELAYS |
|---|---|
| 0 | K0 |
| 1 | K1,K8 |
| 2 | K2,K12 |
| 3 | K3,K9,K12 |
| 4 | K4,K14 |
| 5 | K5,K10,K14 |
| 6 | K6,K13,K14 |
| 7 | K7,K11,K13,K14 |




REVISED FROM:

CYTEC CORP.

| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
|---|---|---|
| DATE: 10/1/2013 | 30-19-50 | REVISION: |
| SIZE: A3 | TITLE: CXR/8X1-GT-75 | |
| 30-19-50 | SHEET 1 OF 1 | |

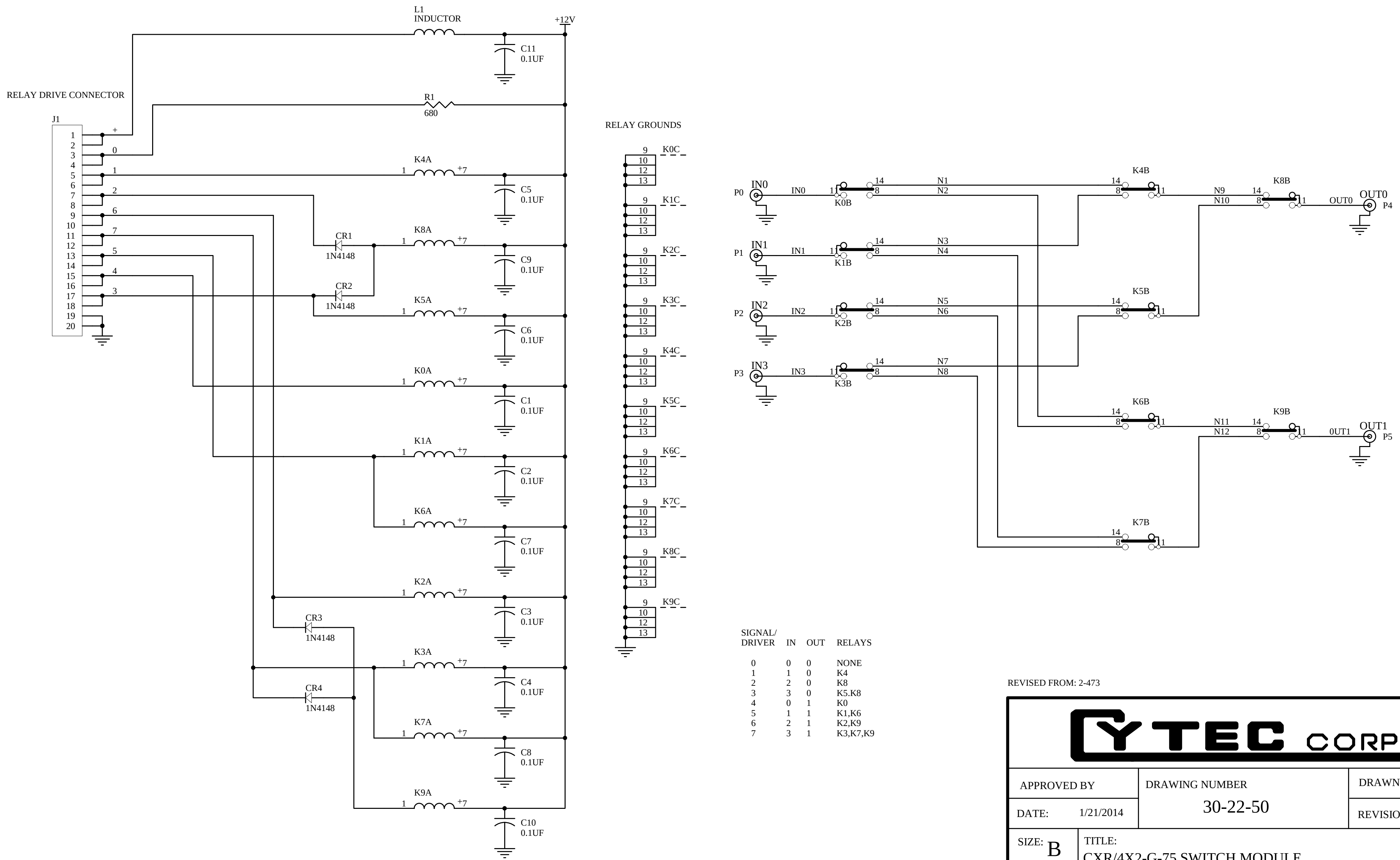



| SIGNAL/ DRIVER | IN | OUT | RELAYS |
|---|---|---|---|
| 0 | 0 | 0 | NONE |
| 1 | 1 | 0 | K4 |
| 2 | 2 | 0 | K8 |
| 3 | 3 | 0 | K5,K8 |
| 4 | 0 | 1 | K0 |
| 5 | 1 | 1 | K1,K6 |
| 6 | 2 | 1 | K2,K9 |
| 7 | 3 | 1 | K3,K7,K9 |

REVISED FROM: 2-473

CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER 30-21-50 | DRAWN BY: RHB |
| DATE: 8/7/2008 | | REVISION: |
| SIZE: B | TITLE: CXR/4X2-G-50 SWITCH MODULE | |
| 30-21-50.SchDoc | | SHEET 1 OF 1 |



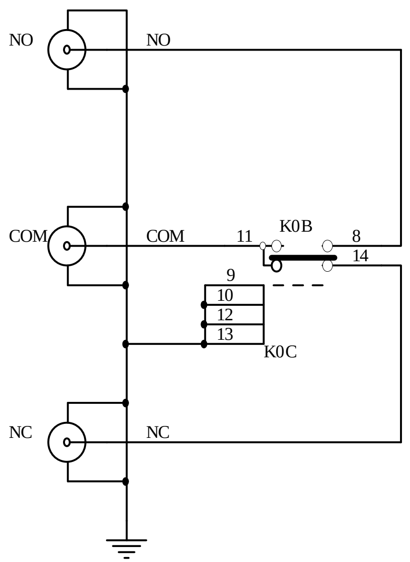

| SIGNAL/ DRIVER | IN | OUT | RELAYS |
|---|---|---|---|
| 0 | 0 | 0 | NONE |
| 1 | 1 | 0 | K4 |
| 2 | 2 | 0 | K8 |
| 3 | 3 | 0 | K5,K8 |
| 4 | 0 | 1 | K0 |
| 5 | 1 | 1 | K1,K6 |
| 6 | 2 | 1 | K2,K9 |
| 7 | 3 | 1 | K3,K7,K9 |

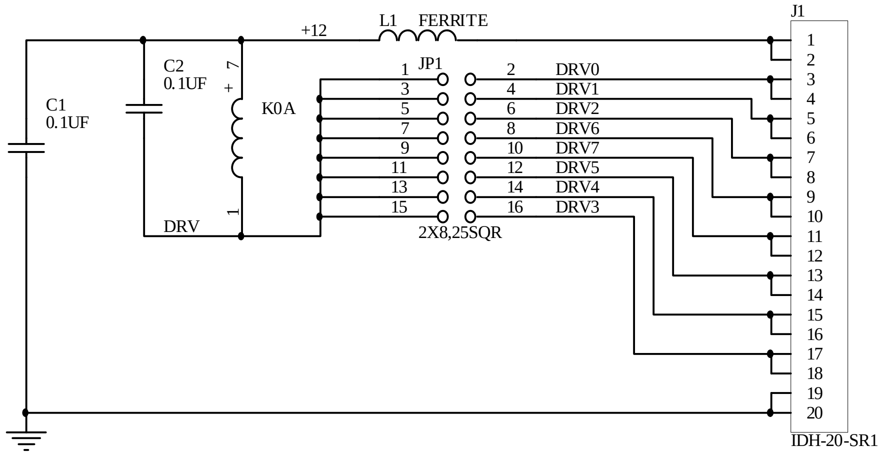
REVISED FROM: 2-473

CYTEC CORP.

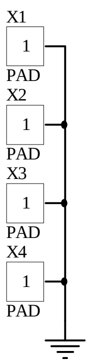
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: RHB |
| DATE: 1/21/2014 | 30-22-50 | REVISION: |
| SIZE: B | TITLE: CXR/4X2-G-75 SWITCH MODULE | |
| 30-22-50.SchDoc | | SHEET 1 OF 1 |

# BNC



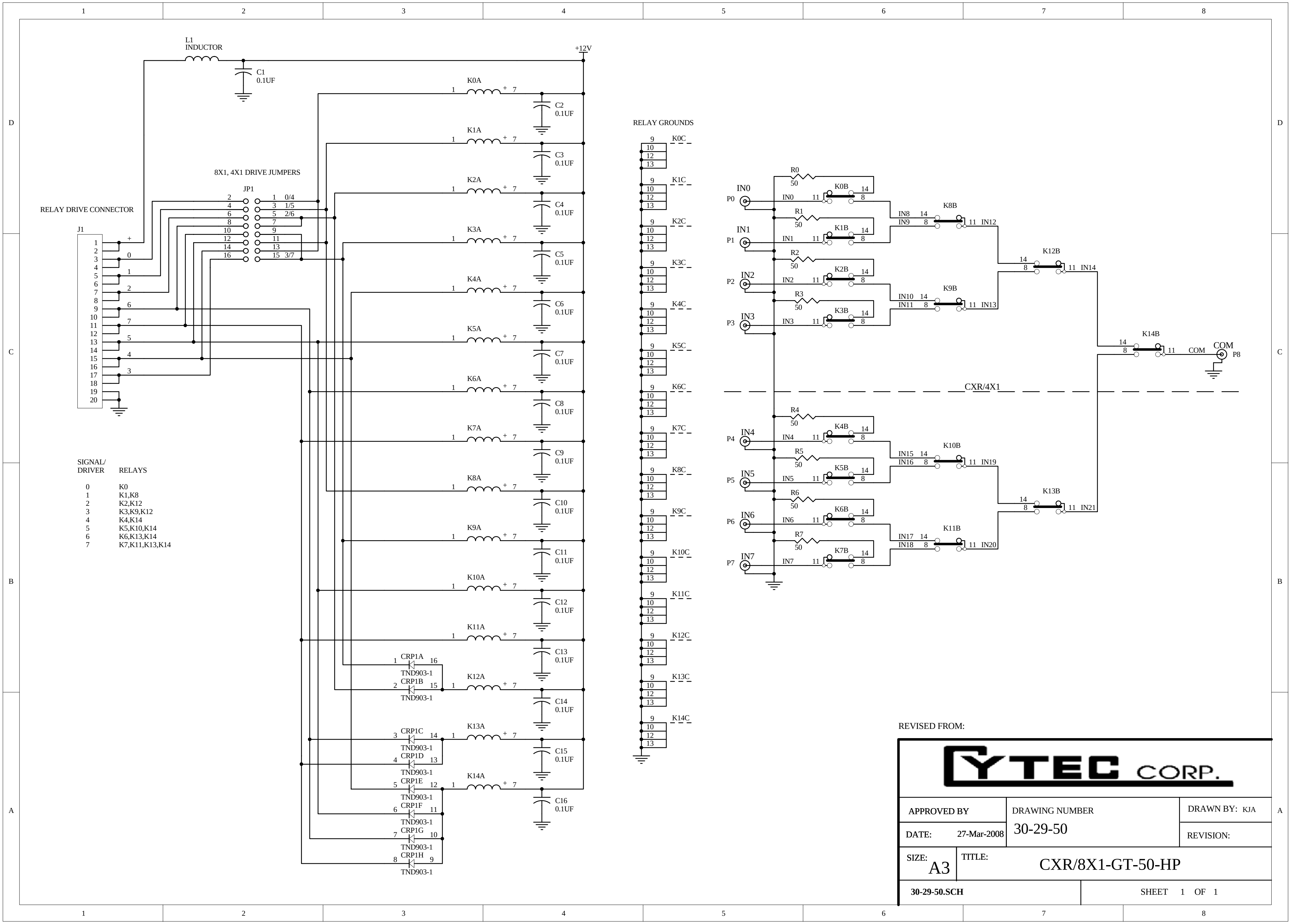

MOUNTING PADS

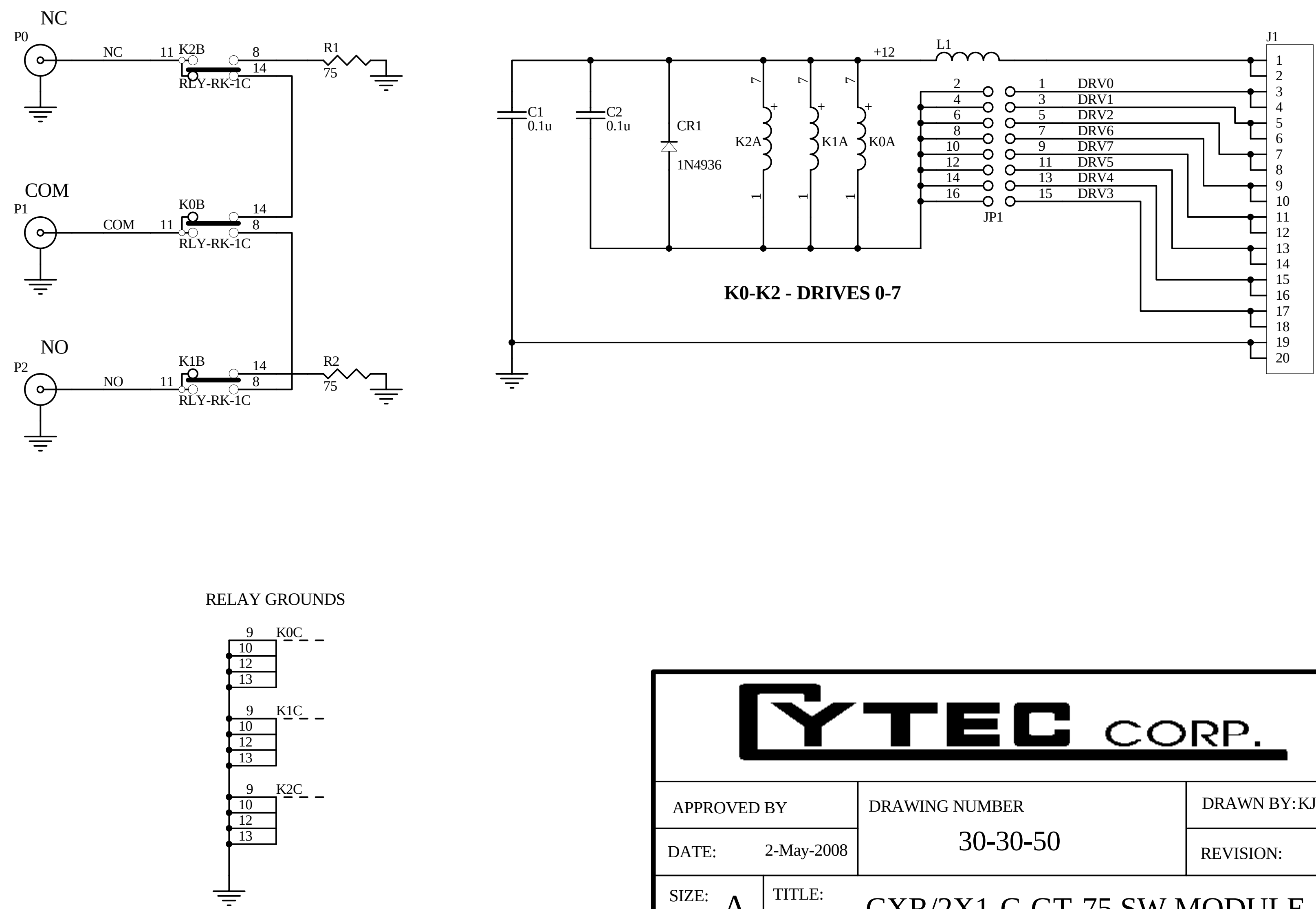



REVISED FROM:

CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER<br>30-25-50 | DRAWN BY: RHB |
| DATE: 10/25/2005 | | REVISION: |
| SIZE: A | TITLE: CXR/2X1-G-50 SW MODULE | |
| 30-25-50.schdoc | | SHEET 1 OF 1 |

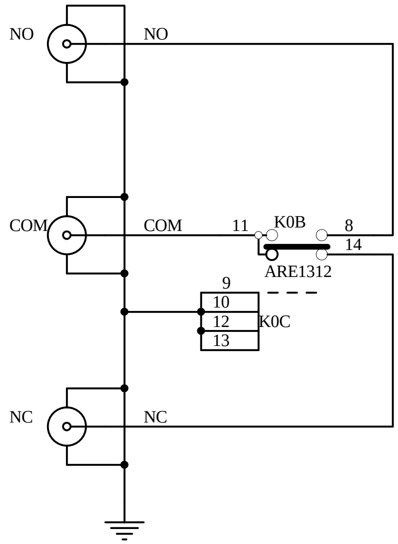

L1
INDUCTOR
C1
0.1UF
+12V
RELAY DRIVE CONNECTOR
J1
8X1, 4X1 DRIVE JUMPERS
JP1
1 0/4
3 1/5
5 2/6
15 3/7
K0A
K1A
K2A
K3A
K4A
K5A
K6A
K7A
K8A
K9A
K10A
K11A
K12A
K13A
K14A
C2 0.1UF
C3 0.1UF
C4 0.1UF
C5 0.1UF
C6 0.1UF
C7 0.1UF
C8 0.1UF
C9 0.1UF
C10 0.1UF
C11 0.1UF
C12 0.1UF
C13 0.1UF
C14 0.1UF
C15 0.1UF
C16 0.1UF
CRP1A TND903-1
CRP1B TND903-1
CRP1C TND903-1
CRP1D TND903-1
CRP1E TND903-1
CRP1F TND903-1
CRP1G TND903-1
CRP1H TND903-1
SIGNAL/
DRIVER RELAYS
0 K0
1 K1,K8
2 K2,K12
3 K3,K9,K12
4 K4,K14
5 K5,K10,K14
6 K6,K13,K14
7 K7,K11,K13,K14
RELAY GROUNDS
K0C
K1C
K2C
K3C
K4C
K5C
K6C
K7C
K8C
K9C
K10C
K11C
K12C
K13C
K14C
IN0 P0
IN1 P1
IN2 P2
IN3 P3
IN4 P4
IN5 P5
IN6 P6
IN7 P7
R0 50
R1 50
R2 50
R3 50
R4 50
R5 50
R6 50
R7 50
K0B
K1B
K2B
K3B
K4B
K5B
K6B
K7B
K8B
K9B
K10B
K11B
K12B
K13B
K14B
IN8
IN9
IN10
IN11
IN12
IN13
IN14
IN15
IN16
IN17
IN18
IN19
IN20
IN21
COM
P8
CXR/4X1
REVISED FROM:
CYTEC CORP.
APPROVED BY
DRAWING NUMBER
DRAWN BY: KJA
DATE: 27-Mar-2008
30-29-50
REVISION:
SIZE: A3
TITLE:
CXR/8X1-GT-50-HP
30-29-50.SCH
SHEET 1 OF 1

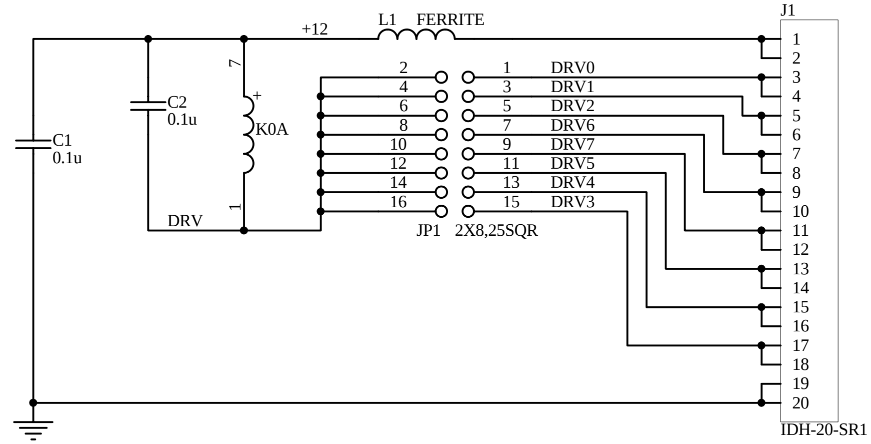



CYTEC CORP.

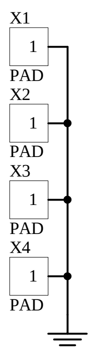
| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER 30-30-50 | DRAWN BY: KJA |
| DATE: 2-May-2008 | | REVISION: |
| SIZE: A | TITLE: CXR/2X1-C-GT-75 SW MODULE | |
| 30-30-50.sch | SHEET 1 OF 1 | |

## F CONNECTORS




MOUNTING PADS




CYTEC CORP.

| | | |
|---|---|---|
| APPROVED BY | DRAWING NUMBER | DRAWN BY: KJA |
| DATE: 17-Apr-2009 | 30-31-50 | REVISION: |
| SIZE: A | TITLE: CXR/2X1-G-75-F SW MOD. | |
| | | SHEET 1 OF 1 |