USER'S REFERENCE MANUAL

PROTO-8

Prototyping Module for PC/104 Bus

Model No. 100-7549 ("B" and "S" Versions)
Doc. No. M7549 Rev: 1.4 06/21/06

09/08/98

DISCLAIMER: This document contains proprietary information regarding SCIDYNE and its products. The information is subject to change without notice. SCIDYNE makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SCIDYNE shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material. No part of this document may be duplicated in any form without prior written consent of SCIDYNE.

WARRANTY: SCIDYNE warrants this product against defects in materials and workmanship and, that it shall conform to specifications current at the time of shipment, for a period of one year from date of shipment. Duration and conditions of warranty may be superseded when the product is integrated into other SCIDYNE products. During the warranty period, SCIDYNE will, at its option and without charge to Buyer, either repair or replace products which prove defective. Repair or replacement of a defective product or part thereof does not extend the original warranty period.

WARRANTY SERVICE: For warranty service or repair, this product must be returned to a service facility designated by SCIDYNE. The Buyer must obtain prior approval and a Return Material Authorization (RMA) number before returning any products. The RMA number must be clearly visible on the shipping container. The Buyer shall prepay shipping and insurance charges to the service facility and SCIDYNE shall pay shipping and insurance charges to Buyer's facility for products repaired or replaced. SCIDYNE may, at its discretion, bill the Buyer for return shipping and insurance charges for products received for repair but determined to be non-defective. Additionally, the Buyer shall pay all shipping charges, duties and taxes for products returned to SCIDYNE from another country.

LIMITATION OF WARRANTY
The forgoing warranty shall not apply to defects resulting from improper or negligent maintenance by the Buyer, Buyer-supplied products or interfacing, unauthorized modifications or misuse, operation outside the published specifications of the product or improper installation site preparation or maintenance, or the result of an accident. The design and implementation of any circuit using this product is the sole responsibility of the Buyer. SCIDYNE does not warrant the Buyer's circuitry or malfunctions of SCIDYNE products that result from the Buyer's circuitry. In addition, SCIDYNE does not warrant any damage that occurs as a result of the Buyer's circuit or any defects that result from Buyer-supplied products. This Warranty does not cover normal preventative maintenance items such as fuse replacement, lamp replacement, resetting of circuit breakers, cleaning of the Product or problems caused by lack of preventative maintenance, improper cleaning, improper programming or improper operating procedures. No other warranty is expressed or implied. SCIDYNE specifically disclaims the implied warranties of merchantability and fitness for a particular purpose. Some states do not permit limitation or exclusion of implied warranties; therefore, the aforesaid limitation(s) or exclusion(s) may not apply to the Buyer. This warranty gives you specific legal rights and you may have other rights which vary from state to state.

CERTIFICATION
Testing and other quality control techniques are utilized to the extent SCIDYNE deems necessary to support this warranty. Specific testing of all parameters is not necessarily performed, except those mandated by government requirements.

30-DAY PRODUCT EVALUATION POLICY: SCIDYNE offers a no-risk trial for initial, low quantity, evaluation purchases. Items purchased for evaluation can be returned within 30 days for a full refund less shipping charges. The Buyer must obtain a Return Material Authorization (RMA) number before returning any products. The entire package, including hardware, software, documentation, discount coupons and any other accessories supplied must be returned intact and in new and working condition. This policy will not be honored for packages that are not returned complete and intact. The Buyer shall prepay shipping and insurance charges to SCIDYNE. To expedite the return process, the RMA number must be clearly visible on the shipping container. SCIDYNE will cancel the invoice, refund by check or issue credit to your credit card within 10 days after receipt of returned merchandise.

LIFE SUPPORT POLICY
Certain applications may involve the risks of death, personal injury or severe property or environmental damage ("Critical Applications").

SCIDYNE products are not designed, intended, authorized or warranted to be suitable for use in life-support applications, devices or systems or other critical applications without the express written approval of the president of SCIDYNE.

SAFETY AND USAGE CONVENTIONS

NOTE: *Contains important information and useful tips that will assist in the understanding and operation of the product.*

CAUTION: *Calls attention to a procedure, practice or condition that could possibly cause personal injury or damage to equipment.*

⚠ WARNING: *Calls attention to a procedure, practice or condition that could possibly cause severe bodily injury, death or extensive equipment damage.*

Table of contents

Introduction

The PROTO-8 is an extended length prototyping board specifically designed for constructing circuitry for the popular PC/104 bus architecture. It is available in two forms. The "S" version provides a grid of over 2000 pads arranged on 0.1" inch centers. It is suitable for solder or wire-wrap assembly techniques and is intended for permanent and durable circuitry construction. The "B" version comes equipped with a reusable solderless breadboarding area which allows circuitry to be rapidly built and tested. In addition, the PC/104 bus signals are accessible by means of socket strips that directly accept 20-28AWG solid wire and component leads. Both versions feature on board address decoding, buffered data lines, full access to all PC/104 J1/P1 signals and I/O decoding logic contained in a single reprogrammable device. All signals are clearly labeled and easily accessed.

NOTE: *Because the PROTO-8-B and PROTO-8-S are so similar, unless otherwise stated this document will refer to both versions as simply PROTO-8.*

The block diagram for the PROTO-8 is shown in figure 1. A six position jumper sets the base address and allows the PROTO-8 to be placed anywhere in the PC/104 I/O bus region. The PROTO-8 is activated when the state of the address lines emitted by the host computer match the state of the jumpers during I/O operations. The comparison is done by the address decoding logic. When a match occurs the address decoding logic activates the programmable I/O decoder which sub-decodes the address and provides various read and write signals. The data buffer is also activated in accordance to whether the operation is an input (read) or output (write). The data buffer can also be controlled with minor additional circuitry to provide 16 bit I/O and memory read and write operations.

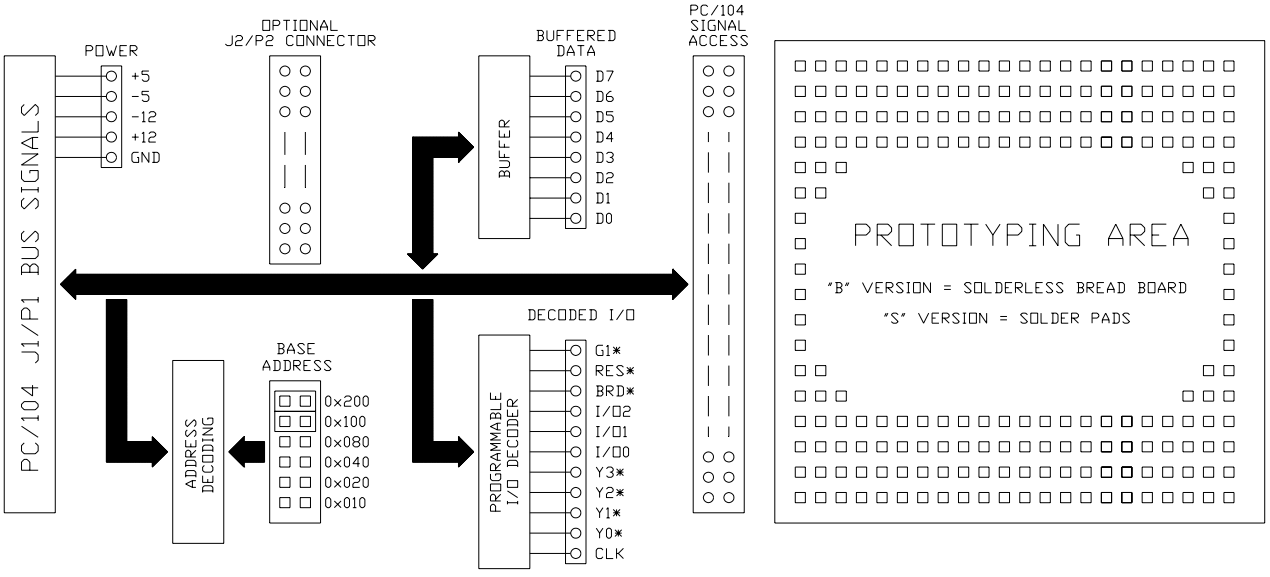


Figure 1 - PROTO-8 Block Diagram

Component Identification

Before using the PROTO-8 the user should become acquainted with the placement and function of its components. They are described below and shown in figure 2.

Component Identification	
Item	Description
1	**I/O Decoding Logic** This socketed device performs the I/O decoding function for the PROTO-8. It is pre-programmed with a general purpose configuration but can be reprogrammed by the user for special applications.
2	**Buffered Data** These eight signals are the buffered PC/104 data bus. They are normally tristate but become active when the host computer is accessing the PROTO-8 through I/O operations. Two default wire jumpers (E-EN*, D-DIR) connect the buffers control signals to the PROTO-8 I/O decoding logic. These jumbers can be removed and additional circuity inserted in their place to support 16 bit I/O and memory read and write operations. The EN* connection activates the buffer and is active low. DIR signal control the direction; A logic 0 will pass data from the PROTO-8 to the PC/104 bus (Read operation); A logic 1 will pass data from the PC/104 bus to the PROTO-8 (Write operation).
3	**Decoded I/O signals** The decoded I/O signals produced by the decoding logic are accessed here.
4	**Power connections** Power from the host computer is routed to these points and is available for the users circuitry. Only those supplies already present on the host will be available.
5	**PC/104 J1/P1 Connector** This connector is the 8 bit PC/104 bus. An optional 20 pin connector (J2/P2) can be installed to upgrade the PROTO-8 for 16 bit stack-through compatibility and signal access.
6	**PC/104 J1/P1 signals** The PC/104 J1/P1 signals are routed directly to these points. The signals are unbuffered
7	**Base Address Jumpers (J3)** This jumper block sets the base address where the PROTO-8 will reside in the hosts I/O map.
8	**Ground** This terminal provides a convenient point for attaching the ground lead of the users test equipment.
9, 10	**Solderless Breadboard** (PROTO-8-B Only) These areas provides an easy way to construct circuits. Experimental circuitry is constructed on item 9. Item 10 is used to distribute the power for the circuits. Each half of item 10 forms a separate power bus.

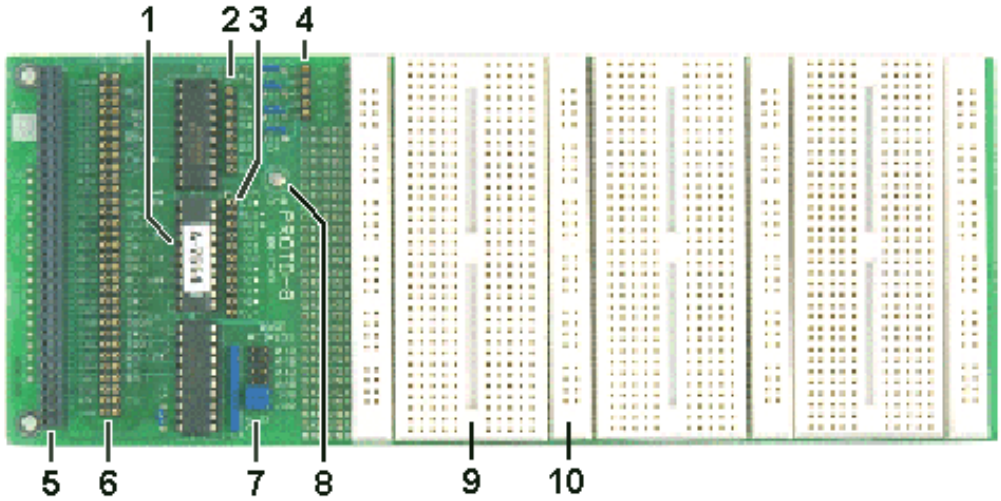


Figure 2 - PROTO-8 Component Identification. (“B” Version shown)

Setting the I/O Base Address

The PROTO-8 occupies 16 consecutive bytes (8 words) and can be placed on any 16 byte boundary within the hosts I/O map. Generally addresses between 0x000 through 0x0ff are used by the host and should be avoided. The factory default I/O address is 0x300 (768_{10}) but can be easily changed to accommodate any special requirements. The six position jumper block, J3, determines the base address. Each jumper position corresponds to a "weighted" I/O address as shown in the following table. The actual starting I/O address where the PROTO-8 resides is calculated by simply adding together the "weight" for each jumper that is installed.

Example: The factory default address is set by placing jumpers in positions 0x100 and 0x200.

Installed jumper	Address value "weight"
J3-5	0x100
J3-6	+ 0x200

	$0x300_{16}$ = 768_{10} = BASE ADDRESS

J3 Base Address Jumper Settings						
J3	6	5	4	3	2	1
Weight (Dec)	0x200 (512)	0x100 (256)	0x080 (128)	0x040 (64)	0x020 (32)	0x010 (16)

Shaded area represents J3 factory default installed jumper positions. All other positions are left open.

Hardware Interface

The PROTO-8 provides easy access to all PC/104 J1/P1 signals as well as generating four decoded I/O strobes. The decoding logic is contained in a single GAL16V8 device. The device is socketed and can be reprogrammed by the user to customize the PROTO-8 in specific applications. Other programmable devices such as PEELS are also supported. The factory default signals generated by the I/O decoding logic are described below.

NOTE: *Additional hardware and software tools are required to reprogram the decoding logic. To successfully reprogram the decoding logic, the user must have prior experience designing with programmable logic and access to a logic compiler and device programmer. These tools are available from a variety of third party vendors.*

Factory Default I/O Decoding Signal Descriptions			
Name	I/O	Enabled	Description
CLK	I	No	**Clock** This input signal is used to provide an asynchronous clock to the decoding logic. It is not enabled on the factory supplied PROTO-8 but can be activated by reprogramming the decoding logic chip.
G1*	I	No	**Gate** This input provides an optional gate enable signal to the decoding logic. It is not enabled on the factory supplied PROTO-8 but can be activated by reprogramming the decoding logic chip.
RES*	O	Yes	**System Reset** This is the inverted system reset. It is active low.
BRD*	O	Yes	**Board Decode** This signal is active whenever Input or Output operations are occurring within the range of the PROTO-8. It is active low.
IO1, IO2	I/O	No	**Input / Output** These signals are provided by the GAL for custom input and output functions. They are not enabled on the factory supplied PROTO-8 but can be activated by reprogramming the GAL.
Y0* Y1* Y2* Y3*	O	Yes	**Decoded Input and Outputs** These signals are generated by the I/O decoding logic. They are normally high but go low during I/O operations within their respective address range and in conjunction with the PC/104 Bus IOR* and IOW* signals. Decoded I/O strobe Active: BASE ADDRESS 0-3 Use with peripheral devices which provide both input and output controls or with additional logic to create sub-decoded input and output strobes. Decoded I/O strobe Active: BASE ADDRESS 4-7 Use with peripheral devices which provide both input and output controls or with additional logic to create sub-decoded input and output strobes. Decoded Output strobe Active: BASE ADDRESS 8-B and IOW* Use with output only devices such as latches. Can be sub-decoded to create four separate one byte output strobes. Decoded Input strobe Active: BASE ADDRESS C-F and IOR* Use with input only devices such as buffers. Can be sub-decoded to create four separate one byte input strobes.

The PC/104 J1/P1 signals are conveniently routed to a header (Solder pads on the "S" version). A brief description of each signal is given below.

NOTE: *An in depth explanation for each signal's purpose and its use is beyond the scope of this manual. Many excellent books exist which fully describe the operation of the PC/104 and ISA bus architecture. The reader is encouraged to seek one of these for a definitive reference.*

PC/104 J1/P1 Signal Descriptions (Listed Alphabetically)		
Signal Name	I/O	Description
AEN	O	**Address Enable** This line is used to degate the processor and other devices from the I/O channel to allow DMA transfers to take place. When this line is active (high), the DMA controller has control of the address bus, data bus, read command lines (memory and I/O), and the write command lines (memory and I/O)
BALE	O	**Bus Latch Enable** This line is provided by the bus controller and is used on the system board to latch valid addresses from the processor. It is used on the I/O channels as an indicator of a valid processor address with AEN. Processor addresses are latched with the falling edge of ALE.
BCLK	I/O	**Bus Clock** Refer to host computer documentation for actual timing information.
DACKx*	O	**DMA Acknowledge** These lines are used to acknowledge DMA requests (DRQ1-DRQ3) and to refresh system dynamic memory (DACKO). They are active low.
DRQx	I	**DMA Request** These lines are asynchronous channel requests used by peripheral devices to gain DMA service. They are prioritized with DRQ3 being the lowest and DRQ1 being the highest. A request is generated by bringing a DRQ line to an active level (high). The DRQ line must be held high until the corresponding DACK line goes active.
IOCHK*	I	**IO Channel Check** This line provides the processor with parity (error) information on memory or devices in the I/O channel. When this signal is active low, a parity error is indicated by typically sending NMI to the CPU.
IOCHRDY	I	**I/O Channel Ready** Driven inactive by a bus resource that needs additional cycles. Must not be held inactive for more than 15.6µs or refresh may be missed.
IOR*, IOW*	O	**Input/Output Read** This signal commands an I/O device to drive its data onto the I/O Channel data bus. It may be driven by the System Control Unit or DMA controller. This signal is active low. **Input/Output Write** This command line instructs an I/O device to read the data on the data bus. It may be driven by the processor or the DMA controller. This signal is active low.
IRQx	I	**Interrupt Request** These lines are used to signal the processor that an I/O device requires attention. They are prioritized with IRQ2 as the highest priority and IRQ7 as the lowest. An Interrupt Request is generated by raising an IRQ line (low to high) and holding it high until it is acknowledged by the processor (interrupt service routine).
OSC	O	**Oscillator** High speed clock. It typically has a 50% duty cycle. Refer to host computer documentation for actual timing information.
REFRESH*	O	**Memory Refresh** Indicates a dynamic memory refresh cycle is in progress. Memory is refreshed every 15.6µs.
RESET	O	**System Reset** Used to reset or initialize system logic on power up, during low line voltage or after a bus timeout.
SA0 - SA19	I/O	**System Address** These lines are used to address memory and I/O devices within the system. The 20 address lines allow access of up to 1 megabyte of memory (2^{20}). AO is the least significant bit (LSB) and A19 is the most significant bit (MSB). These lines are generated by either the processor or DMA controller. They are active high.
SD0 - SD7	I/O	**System Data Bus** SD0 is the Least Significant Bit.
SDRY	I	**Synchronous Ready** (No wait state) Asserted by accessed device to indicate that a shorter access cycle can be executed.
SMEMR*, SMEMW*	O	**System Memory Read** This command line instructs the memory to drive its data on to the data bus. It may be driven by the processor or the DMA controller. This signal is active low. **System Memory Write** This command line instructs the memory to store the data present on the data bus. It may be driven by the processor or the DMA controller. This signal is active low.
TC	O	**Terminal Count** This line provides a pulse when the terminal count for any DMA channel is reached. This signal is active high.

Application Information

The onboard decoding logic makes PC/104 circuitry evaluation quick and easy. The factory supplied configuration should prove sufficient for most designs. For more advanced designs the basic I/O decoding scheme is easily expanded by adding external logic or by reprogramming the I/O decoding chip. Some of the more common circuits arrangements are shown in figure 3.

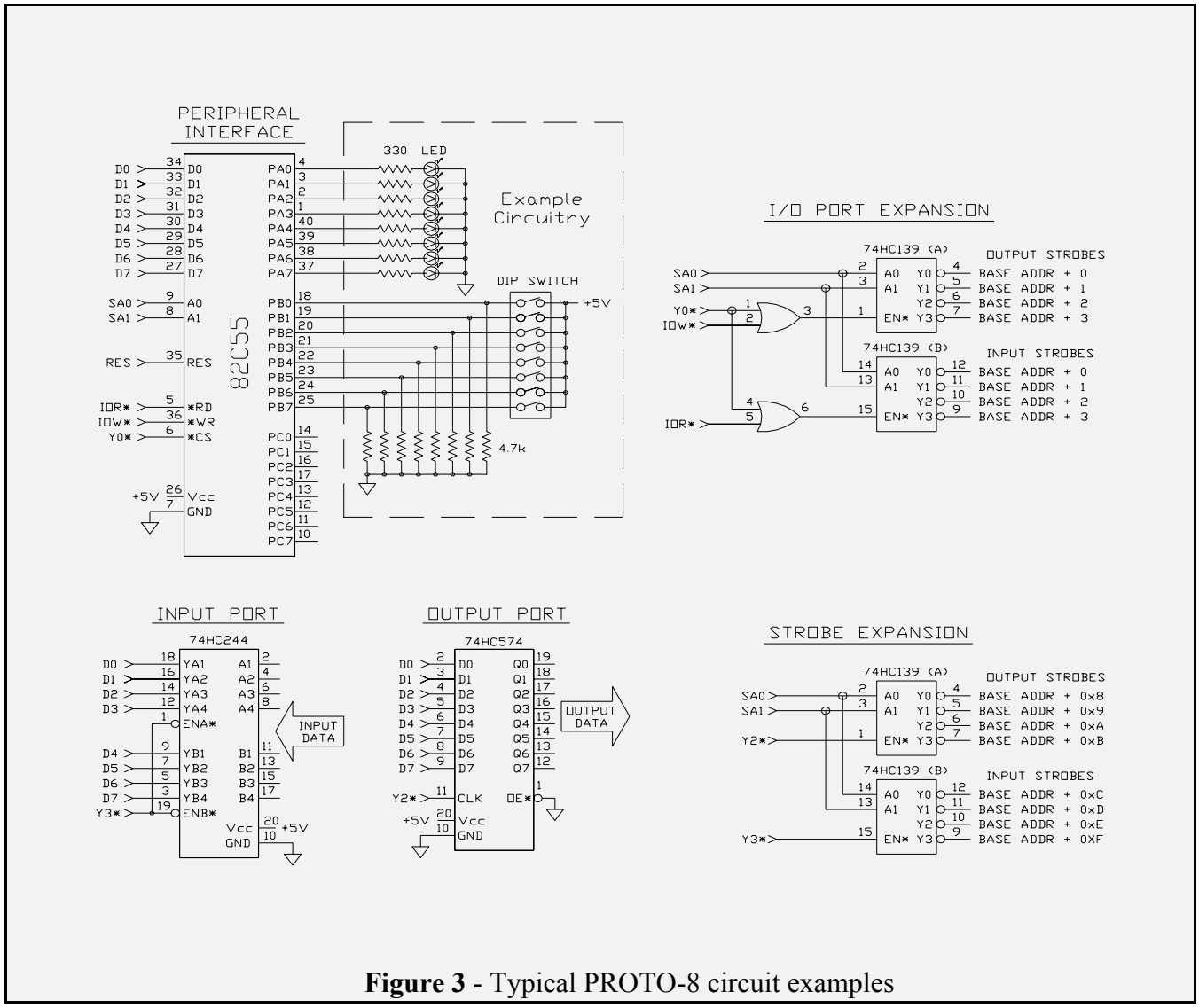


Figure 3 - Typical PROTO-8 circuit examples

The PROTO-8 can be used with any programming language which supports I/O operations such as C, BASIC or Assembler. Some languages such as Microsoft® Visual basic do not inherently support I/O operations and require additional software components to provide these functions.

The following software examples all perform the same function. An 82C55 is wired to the PROTO-8 using the example circuitry depicted in figure 3. The program reads the DIP switches connected to PORTB and displays the resulting hexadecimal value on the computer screen. The user is prompted to enter a decimal number which is then written to PORTA and displayed in native binary on the LEDS.

QuickBASIC Programming Example

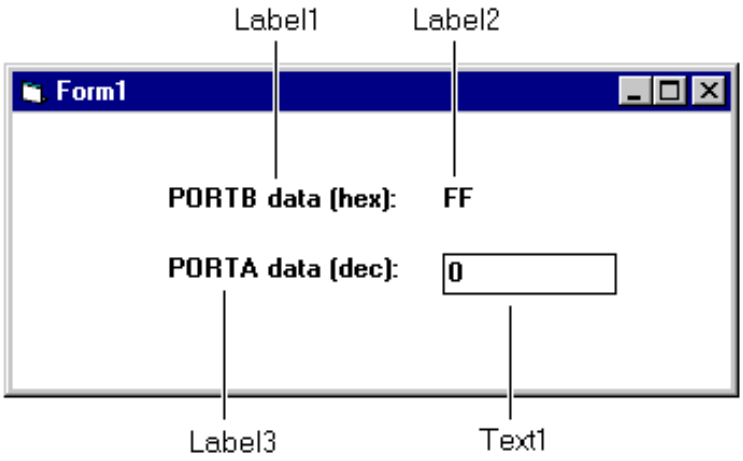
```
'This example assumes an 82C55 connected to PROTO-8 using Y0*

DEFINT A-Z

'-- Define program constants
CONST BASE.ADDR = &H300              'Base IO address of PROTO-8
CONST PORTA = BASE.ADDR + 0          'PORTA Address
CONST PORTB = BASE.ADDR + 1          'PORTB Address
CONST PORTC = BASE.ADDR + 2          'PORTC Address
CONST CTRLREG = BASE.ADDR + 3        'Control Register

'-- Initz the hardware and program
OUT CTRLREG, &H82          'PORTA, PORTC = Outputs   PORTB=Input
OUT PORTA, &H0             'Assure PORTA = 0
OUT PORTC, &H0             'Assure PORTC = 0

'-- Top of program
main:
CLS

PORTB.IMAGE = INP(PORTB)                          'Read PORTB data
PRINT "PORTB (hex): "; HEX$(PORTB.IMAGE)          'Print the result

INPUT "Enter data for PORTA (dec)"; PORTA.DATA    'Ask user for data
PORTA.DATA = PORTA.DATA AND 255                   'Keep it in a byte
OUT PORTA, PORTA.DATA                             'Send out the data

GOTO main                          'Repeat
```

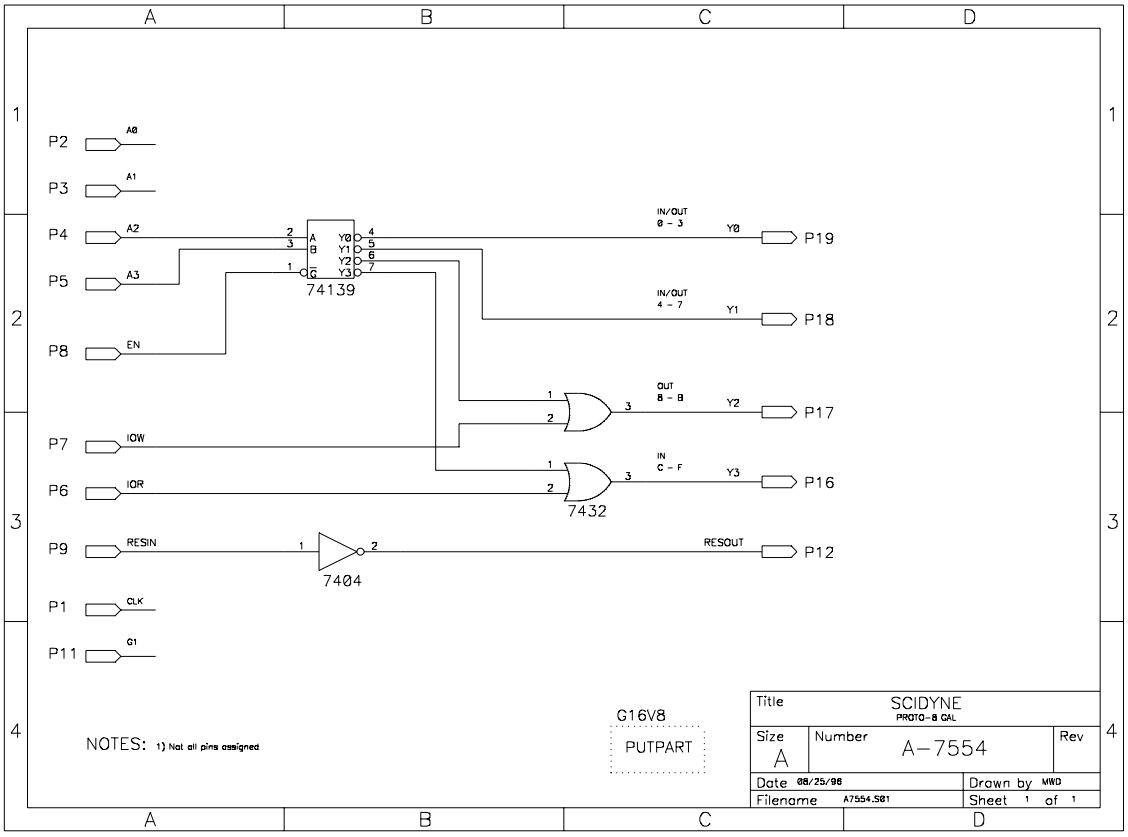
TURBO C Programming Example

```
/****
This program assumes an 82C55 connected to PROTO-8 using Y0*
*****/

/*Includes */
#include <stdio.h>
#include <dos.h>
#include <conio.h>

/* Define global program constants */
#define BASE_ADDR  0x300                    /* Base address of PROTO-8  */
#define PORTA      BASE_ADDR + 0x00         /* Address of porta         */
#define PORTB      BASE_ADDR + 0x01         /* Address of portb         */
#define PORTC      BASE_ADDR + 0x02        /* Address of portc          */
#define CTRLREG    BASE_ADDR + 0x03        /* Address of control register   */

void main(void)
{
   unsigned char portb_image, porta_data;

  /* Initz the hardware and program */
  outp (CTRLREG, 0x82);    /* PORTA, PORTC = Outputs PORTB = Input */
  outp (PORTA, 0x0);        /* Assure PORTA = 0 */
  outp (PORTC, 0x0);       /* Assure PORTC = 0 */

   while(1)
   {
          clrscr();
          portb_image = inp(PORTB);                    /* Read PORTB data */
          printf("PORTB (hex) %X \n",portb_image);     /* Print the result */

          printf("Enter data for PORTA (dec) ");       /* Ask user for data */
          scanf("%d",&porta_data);
          porta_data &= 255;                           /* Keep it in a byte */
          outp (PORTA,porta_data);                     /* Send out the data */
    }
}
```

Visual Basic 3.0 Programming Example

```
[GENERAL DECLARATIONS]
Option Explicit

' This program assumes an 82C55 using Y0*
' NOTE:  The ISACOMM.DLL is available on the SCIDYNE web page and must be present for this program to work

'-- Declare dynamic link libraries
Declare Sub isawrite8 Lib "C:\WINDOWS\SYSTEM\ISACOMM.DLL" (ByVal BusAddr As Integer, ByVal busvalue As Integer)
Declare Function isaread8 Lib "C:\WINDOWS\SYSTEM\ISACOMM.DLL" (ByVal BusAddr As Integer) As Integer

'-- Define program constants
Const BASE_ADDR = &H300              'Base IO address of PROTO-8
Const PORTA = BASE_ADDR + 0          'PORTA Address
Const PORTB = BASE_ADDR + 1          'PORTB Address
Const PORTC = BASE_ADDR + 2          'PORTC Address
Const CTRLREG = BASE_ADDR + 3    'Control Register

'-- Declare global variables
Dim PORTA_DATA As Integer
Dim PORTB_IMAGE As Integer

[FORM LOAD]
Sub Form_Load ()

'-- Initz the hardware and program
Call isawrite8(CTRLREG, &H82)      'PORTA, PORTC = Outputs PORTB=Input
Call isawrite8(PORTA, &H0)           'Assure PORTA = 0
Call isawrite8(PORTC, &H0)            'Assure PORTC = 0

text1.Text = "0"
PORTB_IMAGE = isaread8(PORTB)      'Read PORTB data
label2.Caption = Hex$(PORTB_IMAGE)  'Print the result

End Sub

[TEXT1 CHANGE]
Sub Text1_Change ()

PORTB_IMAGE = isaread8(PORTB)         'Read PORTB data
label2.Caption = Hex$(PORTB_IMAGE)     'Print the result

PORTA_DATA = Val(text1.Text)                  'Ask user for data
PORTA_DATA = PORTA_DATA And 255       'Keep it in a byte
Call isawrite8(PORTA, PORTA_DATA)            'Send out the data

End Sub
```





Y0.oe = 1;

!Y0 = !EN.i & !A2.I & !A3.I ;

Y1.oe = 1;

!Y1 = !EN.i & A2.I & !A3.I ;

Y2.oe = 1;

!Y2 = !EN.i & !A2.I & A3.I & !IOW.i ;

Y3.oe = 1;

!Y3 = !EN.i & A2.I & A3.I & !IOR.i ;

RESOUT.oe = 1;

!RESOUT = RESIN.i ;

Changing the decoding logic

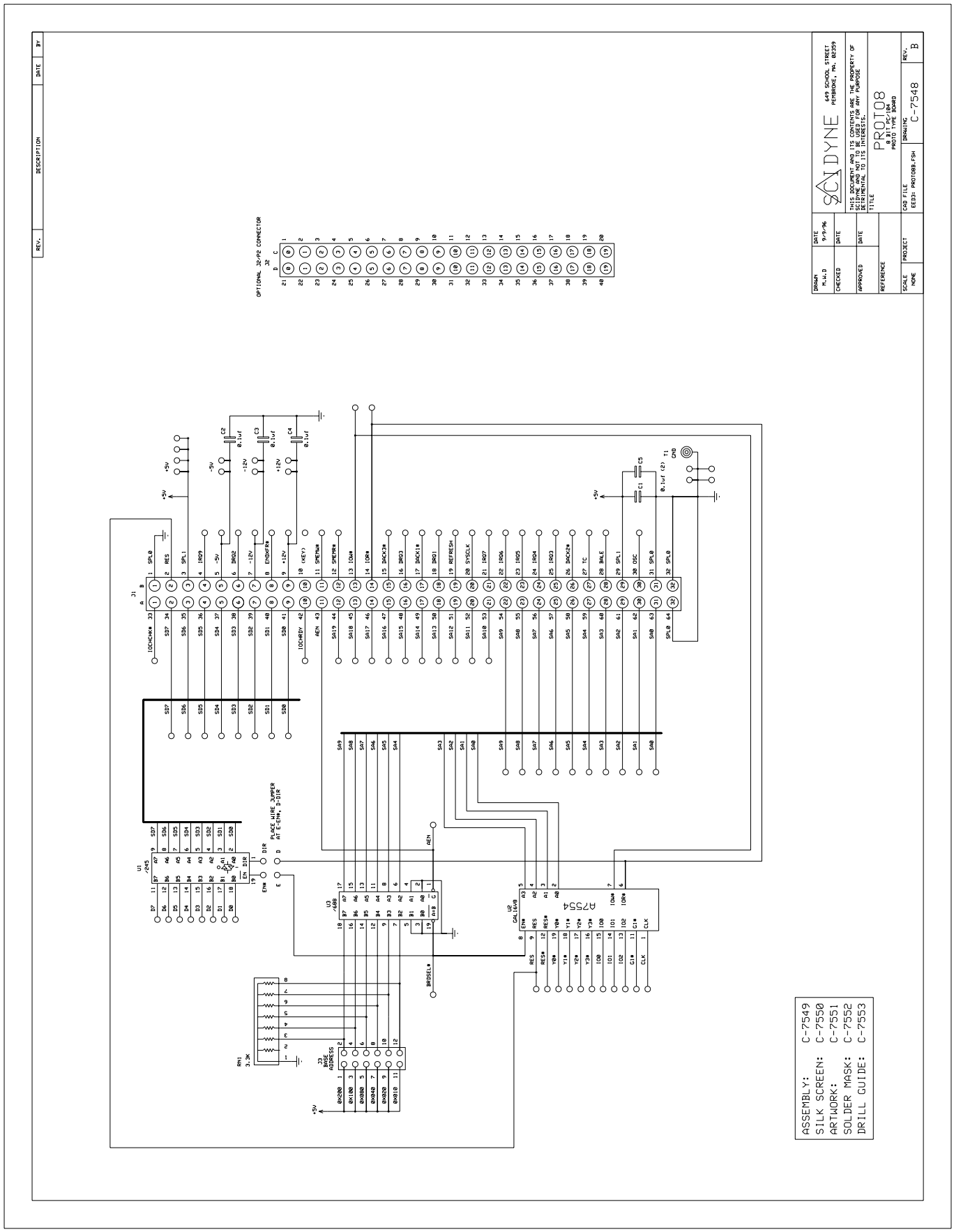
Certain PC/104 signals are hardwired to specific pins of the I/O decoding logic chip. When creating new decoding logic these pins can only be used to perform the same function. Other pins are simply connected to positions on the DECODE socket strip (solder pads on the "S" version) and arbitrarily named for their originally intended purpose. These signals may be freely reassigned as required by the user.

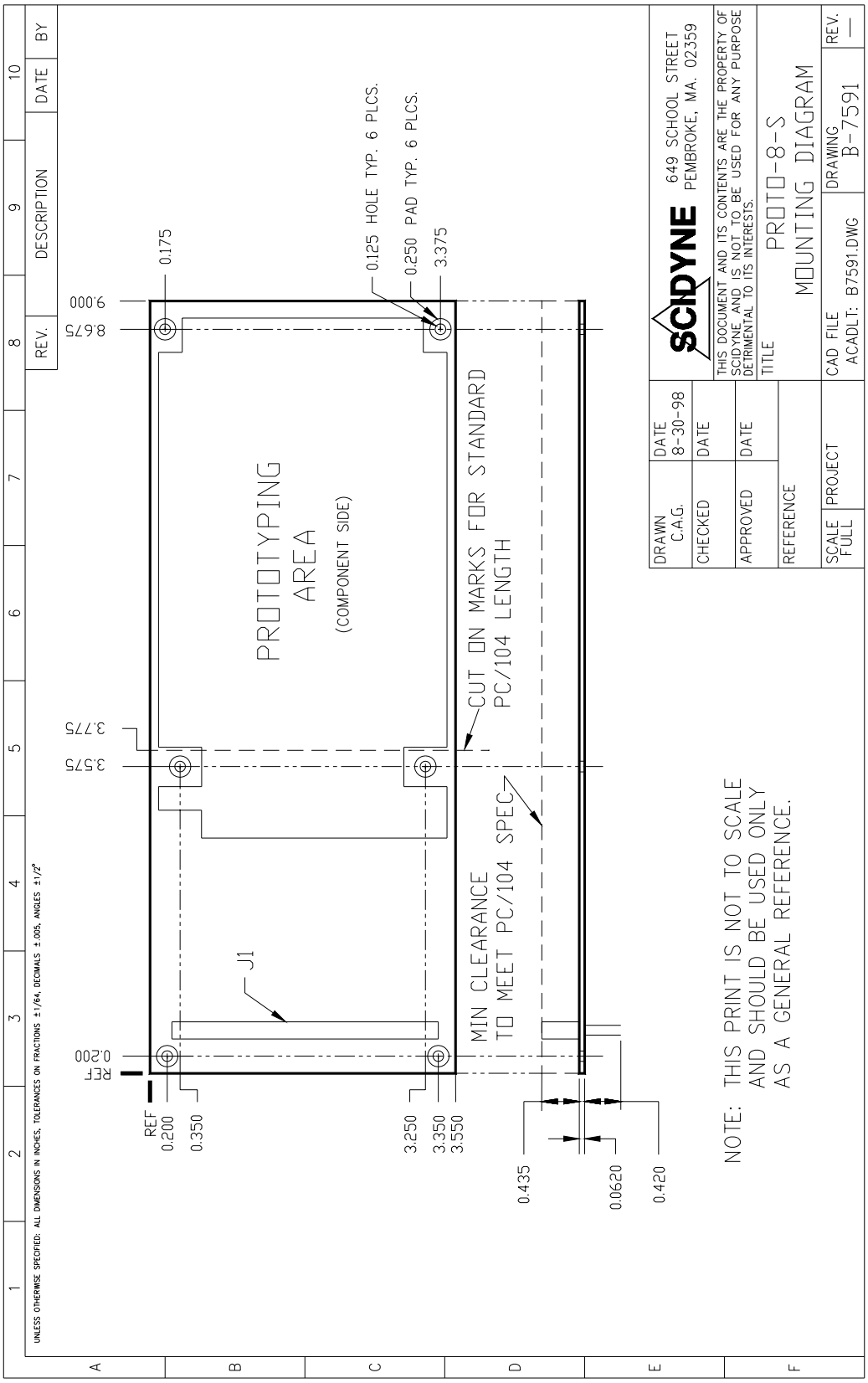
PROTO-8 Decoding logic		
IC Pin #	Default Name	Function / Commitment
2 3 4 5	A0 A1 A2 A3	Address inputs, must remain as A0-A3
6 7	IOR* IOW*	I/O control inputs, must remain as IOR* and IOW*
8	EN	Decoder enable input, must remain as EN
9	RES	Reset input, must remain as RES
10 20	Vss Vdd	These are power pins and cannot be changed
1 11 12 13 14 15 16 17 18 19	CLK G1* RES* IO2 IO1 IO0 Y3* Y2* Y1* Y0*	Can be changed. These signals are routed to the DECODE access points.

Replacing the chip

The decoding logic chip is located at position U2 on the PROTO-8 printed circuit board. A socket is used to mount the chip which permits easy extraction and replacement. The chip is best removed using an IC extraction tool. If this tool is not available the chip can also be removed by gently prying up each end in an alternating fashion. Be sure to pry only the chip and not the socket. The replacement chip is installed by pressing its leads into the socket. Be sure its leads are accurately located over the socket to prevent bending and possible damage.

CAUTION: *This procedure should be performed at a static safe work area to prevent damage to the decoding logic chip. When installing a chip pay close attention to its orientation. Be sure to match pin #1 of the socket to that of the chip .*


SCIDYNE
PROTO8
8 BIT PC/104
PROTO TYPE BOARD
C-7548
ASSEMBLY: C-7549
SILK SCREEN: C-7550
ARTWORK: C-7551
SOLDER MASK: C-7552
DRILL GUIDE: C-7553


PROTOTYPING AREA
(COMPONENT SIDE)
CUT ON MARKS FOR STANDARD PC/104 LENGTH
MIN CLEARANCE TO MEET PC/104 SPEC
J1
0.175
0.125 HOLE TYP. 6 PLCS.
0.250 PAD TYP. 6 PLCS.
3.375
9.000
8.675
3.775
3.575
REF 0.200
REF
0.200
0.350
3.250
3.350
3.550
0.435
0.0620
0.420
NOTE: THIS PRINT IS NOT TO SCALE AND SHOULD BE USED ONLY AS A GENERAL REFERENCE.
UNLESS OTHERWISE SPECIFIED: ALL DIMENSIONS IN INCHES, TOLERANCES ON FRACTIONS ±1/64, DECIMALS ±.005, ANGLES ±1/2°
REV. DESCRIPTION DATE BY
SCIDYNE 649 SCHOOL STREET PEMBROKE, MA. 02359
THIS DOCUMENT AND ITS CONTENTS ARE THE PROPERTY OF SCIDYNE AND IS NOT TO BE USED FOR ANY PURPOSE DETRIMENTAL TO ITS INTERESTS.
TITLE PROTO-8-S MOUNTING DIAGRAM
DRAWN C.A.G. DATE 8-30-98
CHECKED DATE
APPROVED DATE
REFERENCE
SCALE FULL PROJECT
CAD FILE ACADLT: B7591.DWG
DRAWING B-7591
REV. —