LP120

USER'S MANUAL

VERSION 1.1g

Lucid Technologies
http://www.lucidtechnologies.info
info@lucidtechnologies.info

CONTENTS

1.0 General Information

1.1 Introduction

The LP120 programmer was designed to be compatible with any computer having an RS-232 serial port. Throughout this manual, your PC will be referred to as the "host system". There is only one software requirement placed on the host system; it must run a communication program that will do terminal emulation and ASCII file transfer. Since all communication programs perform these functions, any one that will run on your machine should be sufficient.

The assembly directions in this manual are intentionally brief. The LP120 kit is intended for people who are knowledgeable in electronics so detailed step-by-step instructions were omitted.

1.2 Theory of operation

The LP120 is an 8-bit microprocessor system with its own RAM, EPROM, and I/O. The LP120 uses a 6803 type CPU. Its operating program starts whenever power is applied. All communication with the host system is via an RS-232 port.

The LP120 tries to meet the various hardware requirements for a wide selection of programmable devices by concentrating the majority of common resources on the main circuit board. Address and data lines to the device being programmed must be steady during any programming pulse. This means the device being programmed can not be connected to the address or data bus of the 6803. Therefore parallel ports are used to set the address and data for the device being programmed. The programming connector (J3) provides four byte-wide I/O ports, two bi-directional control lines and two control inputs. The various programming and supply voltages required by the devices are handled by four switchable voltages. Two of these voltages are fixed (PMVcc and PMVfw) while two are programmable (Vpp and Vps).

If a particular device doesn't use all the digital lines or voltages, it is only connected to those it needs. Getting the required signals to the correct pins for different devices is basically just point to point wiring and is handled with a very simple adapter. The adapter, referred to as a "programming-module", plugs into the "programming connector" (J3) a common 44- pin, 0.156 inch, edge connector.

The LP120's firmware doesn't contain the code to program all the devices the LP120 is capable of programming. The "driver" software for each device must be uploaded to the LP120 before the device can be programmed. Each driver comes with an annex to this manual that explains its operation.

1.3 Hardware description

Page 1 of the schematics shows the functional blocks that make up the LP120 and how they interconnect.

Page 2 is the CPU block with the 6803 microprocessor (U1), address latch (U2), and address decoder (U6). The 6803 provides a 64K byte-wide address space, a UART, and a parallel port, all in one chip. A PAL was used to do address decoding and clock division as discrete logic would have required at least three chips.

Page 3 is the programmer's memory. The LP120's firmware is stored into the 27128 (U4). The 62256 and 6264 chips (U3 and U5) form 40K of continuous RAM.

The serial port is on page 4. The serial connector is a DB- 9 type, wired as a DCE device. This mates directly with the 9-pin DTE connectors found on most IBMs and compatibles. The

MAX232A (U13) contains two RS-232 drivers, two RS-232 receivers, and an on chip charge-pump. The charge-pump uses the 5 volt supply to generate the bipolar voltages needed by the RS-232 drivers. Even though the 6803 has an internal UART, its baud rate selection is limited. A clock divider (U9) provides any standard rate from 1200 to 19200 baud. The baud-rate is selected with jumper JP1.

Page 5 shows U7 and U8, the two 6821 parallel interface adapters (PIAs). The PIA's four bi-directional parallel ports go to the programming connector on page 8. Additionally, the four control lines from U7 are also routed to the programming connector.

Page 6 shows the programmable DC converters. Both converters are powered by the full-wave rectified voltage (Vfw) from the power supply. Both converters are controlled by voltages from the MAX522 dual 8-bit DAC shown on the next page. The DAC's outputs go to the reference inputs of the 78S40 switching voltage regulators (U11 and U12). The 78S40 will try to make its feedback signal (pin 6) equal the reference input (pin 9) by changing the output voltage. The circuit around U11 generates the programming voltage, Vpp. Vpp should be ten times the reference voltage (DACA) since the feedback network (R9) has a gain of 1/10. For example, if DACA is set to 2.10 volts Vpp will go to 21.0 volts. The circuit around U12 generates the supply voltage for the device being programmed, Vps. Vps should be three times the reference voltage (DACB) since the feedback network (R10) has a gain of 0.333.

Page 7 shows the switching circuitry for the fixed voltages going to the programming-module, and the dual DAC. Vcc is switched to the programming-module, where it is called PMVcc, by Q3. Vfw is switched to the programming-module, where it is called PMVfw, by Q2. The MAX522 (U14) is a serial input dual 8- bit DAC. The serial interface is handled by three lines from the parallel port on the 6803. The range for both DAC channels is 0V to the reference voltage on pin 7. The TL431 (U15) provides a 2.5V reference for the DAC.

The programming connector is shown on page 8. The programming-module for the part being programmed plugs into this connector.

The last page of schematics shows a typical external power supply for use with the LP120.

1.4 Specifications

Size:
* 6.75 x 5.75 inches

Power requirement:
* +5VDC @ 750mA maximum from external source
* +15VDC or unregulated +14 to +16VDC @ 1A from external source

Serial interface:
* Asynchronous RS-232, no handshaking
* Female, 9-pin, D connector

Baud rate:
* Jumper selectable; 1200, 2400, 4800, 9600, 19200

Serial data format:
* 8 data bits
* No parity
* 1 stop bit

2.0 Construction

2.1 Preparation

The tools required to build the LP120 can be found on most electronic workbenches. You will need a low wattage soldering iron, lead benders, lead clippers, and a small screwdriver. Be sure to check the errata file before beginning!

2.2 Power

The LP120 kit comes without its own power supply. This was done for two reasons. First, it allows you to enjoy designing and building your own power supply. Second, it puts the liability for construction on you and not Lucid Technologies.

WARNING

BUILDING A POWER SUPPLY FOR THE LP120 WILL INVOLVE A CONNECTION TO HOUSEHOLD AC CURRENT. THIS IS POTENTIALLY DANGEROUS AND SHOULD ONLY BE ATTEMPTED BY KNOWLEDGEABLE AND EXPERIENCED PERSONNEL. EXPOSED AC VOLTAGES ARE A POTENTIALLY LETHAL SHOCK HAZARD! INCORRECT CONSTRUCTION TECHNIQUES CAN RESULT IN A FIRE HAZARD! LUCID TECHNOLOGIES ASSUMES NO LIABILITY FOR DAMAGE OR INJURY RESULTING FROM THE USE OF ANY POWER SUPPLY WITH THE LP120.

Appendix F provides the schematic for a simple low cost power supply that provides a regulated +5VDC and an unregulated +15VDC. You can also use commercial power supplies to provide the required +5 and +15 volts. Note that the +5V supply must be regulated but the +15V supply need not be. If you are uncomfortable about building your own power supply you can return your LP120 for a refund; see the refund policy on the web site.

2.3 Cables

The LP120 is designed as a DCE device. It uses a 9 pin, female, D connector that is directly compatible with 9 pin COM ports found on most IBM compatibles. The pin assignments and signal directions for the serial connector are shown in appendix C. If your computer has a 25 pin D connector you will need to use an adapter cable. Be sure to check the definition of the lines on your computer so they are paired with the correct lines on the LP120.

2.4 Assembly

First, read the errata file on the distribution disk. Check the appendices for the parts list. Sockets for all ICs are strongly recommended. Most of the parts for the LP120 can be found in any electronics store or mail-order catalog. Using common parts helps to keep the cost low and makes it easy to find spare parts when you need them.

Remember, the LP120 is a small computer system, a single fault on the address bus or data bus will render it inoperable. Debugging such a hardware problem can be frustrating, to say the least, therefore it is important you exercise the utmost care during assembly. The square pads on the board

mark pin one for: ICs, resistor networks, and connectors. For polarized capacitors the square pad denotes the positive lead. For resistor networks the square pad is the common pin of the network. For diodes the square pad is the cathode.

Check the circuit board carefully before you start soldering. Hold the board up to a lamp so that the light shines through the board. This backlighting makes it easy to examine the traces on the near side. Look for breaks in the traces or shorts caused by incomplete etching of the copper. Pay particular attention to the areas where the traces run between the pins of the ICs. If you have any doubts use an ohmmeter to double check your visual inspection. Bridge any breaks and cut any shorts you find. Remember to check both sides of the board.

The mounting holes in each corner of the board are tied to ground so be sure your chassis is compatible! Test your mounting hardware for the board and J2 to see if the holes must be enlarged. If drilling is required be sure to do it before beginning the electronic assembly.

Start by installing the sockets for the ICs. Do not plug the ICs into the sockets until after the initial power check. Next install the resistor networks noting the correct orientation of pin one. Continue by installing the discrete resistors, capacitors, crystal, inductors and jumper pins. Be sure the electrolytic capacitors are properly polarized before soldering them in place. Now the transistors and diodes can be installed. Again, be sure they are properly oriented before soldering any leads.

Before you solder the connectors (J1,J2,J3), give some thought to how you are going to mount the board. You can use standoffs as simple legs at each corner or you can get fancy and mount both the board and power supply in a nice chassis, it's entirely up to you. If you do use a closed chassis you might want to run wires from the board to J2, J3, or both. This would allow you to mount the connectors on the chassis. Be sure to ground the chassis properly.

2.5 Checkout

With no ICs installed attach ground and +5V (Vcc) to J1. Attach the common lead of your voltmeter to a convenient ground point. Turn on the power supply and touch the positive probe of your voltmeter to the Vcc pin of all the IC sockets. The power pins for all ICs are shown on the schematics. If you don't read 5 volts at all these points, power down the circuit and carefully check all wiring, soldering, and component installation. Do not proceed until you have corrected the problem.

Measure the no-load voltage on Vfw at the power supply. Turn the power supply off and add Vfw, the third connection, to J1. Turn the power supply back on and check for Vfw at the following points:

U11 - pins 5,13,14,15
U12 - pins 5,13,14,15,16
Q2 - emitter

If you don't read Vfw at all these points, power down the circuit and carefully check all wiring, soldering, and component installation. Do not proceed until you have corrected the problem.

Remove power and install all the ICs. Make certain each chip is in the correct socket, oriented properly and has no pins folded under the IC. Attach +5V, Vfw, and ground to the designated positions on J1.

Further check-out requires communication between the host system and the LP120. Connect an RS-232 cable from the LP120 to your computer's serial port. Start your host's communications program and insure it is set for the correct COM port and baud rate. Confirm jumper JP1 on the LP120

is set for the same baud rate.

Turn on the power supply and observe your computer screen. If the characters on the screen are gibberish there is probably a baud rate mismatch. If nothing appears, there are several potential problems. Be sure your RS-232 cable is wired correctly for the LP120. If possible check the operation of the host's serial port. Confirm the setting for the communications program are correct. Recheck all the voltages on the LP120, a shorted line may be pulling one of them down. Check the 6803 clock for correct operation, a 921.6 kHz TTL square wave on U1 pin 40.

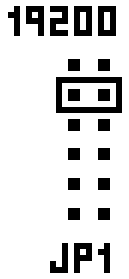
If one of the ICs on the LP120 is exceptionally hot it may be installed backwards, have a shorted data line, or just be a bad chip. Remove all ICs and run board continuity checks for all the pins of the suspect IC. Be sure the pins connect to all of the points they should and nowhere else! If you have some way of testing the chip do so, and replace if necessary.

If the initial screen is a legible menu, it should indicate a calibration option. Select the calibration option. Measure Vpp and Vps at the test points indicated on the LP120. Adjust Vpp to the voltage indicated on the screen with trimmer R9. Adjust Vps to the voltage indicated on the screen with trimmer R10. Press ESCape when both voltages are correct. Turn power off, the LP120 is now ready for use.

3.0 Operation

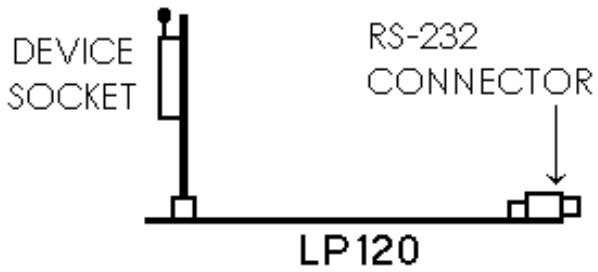
3.1 Baud rate selection

The baud rate is selected via jumper JP1. Any standard baud rate from 1200 to 19200 can be set by selecting the correct jumper position. The top position is 19200, the bottom is 1200. A removable jumper short should be used to set the baud rate. Only one baud rate can be set at a time. An example is shown below.



Jumper for 9600 baud.

3.2 Using programming-modules



Programming-modules should be inserted or removed only when the power to the LP120 is off! As depicted in the diagram below, the programming-module should be inserted so that the programming socket faces away from the LP120 circuit board.

If the connector fingers on the programming-

module get dirty or corroded they can be cleaned by rubbing them with a pencil eraser then washing them with rubbing alcohol.

4.0 Opening Menu

When power is first applied to the programmer it will send an opening menu, similar to the one shown below, to the host. Menu selections are made by typing the character in brackets. Any other character will cause the menu screen to be retransmitted.

```
                LP120 Programmer
              Software Version 2.5
          (C) Lucid Technologies 1990-2005

--------- LP120 OPENING MENU ----------
 [U]pload driver program
 [J]ump to driver at 0100 hex
 [D]isplay system memory
 [T]est static RAM
 [C]alibrate Vpp and Vps
  ?
```

4.1 [U]pload driver program

Use this option to upload the driver for the device you want to program. Drivers are stored as ASCII files in the Motorola S-record format. The upload is accomplished via the ASCII file transfer facility of whatever terminal/communication program you are using on the host. After uploading, control should automatically transfer to the driver program and its menu screen should appear. If the LP120 OPENING MENU reappears, use the [J]ump option to start the driver.

4.2 [J]ump to driver at 0100 hex

Use this option to transfer control to 0100 hex which is the address where driver programs should begin. See the LP120 DEVELOPERS' GUIDE for details.

4.3 [D]isplay system memory

This option is included as a debugging aid for those users who write their own driver programs for the LP120. Any portion of the LP120 memory space can be displayed.

4.4 [T]est static RAM

This test checks all available RAM, from $0020 to $9FFF. Each memory scan writes the same test byte at every address then goes back and reads every address to confirm the data is correct. The test byte is incremented with each new scan. An "*" is printed on the screen every 256 scans. If an error is encountered the test will halt and the address of the bad byte will be displayed. The test may be stopped at any time by pressing the ESCape key.

4.5 [C]alibrate Vpp and Vps

Use this option to calibrate the DC-to-DC converter circuits. This calibration corrects the gain of the DC-DC converters so their outputs match the voltages called for by the microprocessor.

APPENDIX A

S-RECORD INFORMATION

INTRODUCTION

Motorola's S-record format for output modules was devised for the purpose of encoding programs or data files in a printable (ASCII) format. This allows viewing of the object file with standard tools and easy file transfer from one computer to another, or between a host and target. An individual S-record is a single line in a file composed of many S-records.

S-RECORD CONTENT

S-Records are character strings made of several fields which specify the record type, record length, memory address, data, and checksum. Each byte of binary data is encoded as a 2-character hexadecimal number: the first ASCII character representing the high-order 4 bits, and the second the low-order 4 bits of the byte.

The 5 fields which comprise an S-record are:

TYPE - RECORD LENGTH - ADDRESS - CODE/DATA - CHECKSUM

The fields are defined as follows:

Field	**Characters**	**Content**
Type	2	S-record type - S1 or S9
Record length	2	The count of the character pairs in the record, excluding the type and record length.
Address	4	The 2-byte address at which the data field is to be loaded into memory.
Data	0-2n	From 0 to n bytes of executable code, or memory loadable data. n is normally $20 (32 decimal) or less.
Checksum	2	The least significant byte of the one's complement of the sum of the values represented by the pairs of characters making up the record length, address, and data fields.

Each record may be terminated with a CR/LF/NULL. Additionally, an S-record may have an initial field to accommodate other data such as line numbers. Accuracy of transmission is ensured by the record length (byte count) and checksum fields.

S-RECORD TYPES

Eight types of S-records have been defined to accommodate various encoding, transportation, and decoding needs. Lucid programmers use the 8-bit data types, the S1 and S9:

S1 A record containing data and the 2-byte address at which the data is to reside.

S9 A termination record for a file of S1-records. Only one S9-record is allowed per file and it must be the last line of the file. The address field for directly executable code may optionally contain the 2-byte address of the instruction to which control is to be passed. For ROM data the S9 address field is usually 0000. There is no data field.

S-RECORD EXAMPLE

The following is a typical S-record module:

```
S1130000285F245F2212226A000424290008237C2A
S11300100002000800082629001853812341001813
S113002041E900084E42234300182342000824A952
S107003000144ED492
S9030000FC
```

The module consists of four S1 data records and an S9 termination record.

The first S1 data record is explained as follows:

S1 S-record type S1, indicating a data record to be loaded/verified at a 2-byte address.

13 Hex 13 (decimal 19), indicating 19 character pairs, representing 19 bytes of binary data, follow.

00 Four-character 2-byte address field: hex address 0000,
00 indicates location where the following data is to be loaded.

The next 16 character pairs are the ASCII bytes of the actual program data.

2A Checksum of the first S1-record.

The second and third S1 data records also contain $13 character pairs each. The fourth S1 data record contains 7 character pairs.

The S9 termination record is explained as follows:

S9	S-record type S9, indicating a termination record.
03	Hex 03, indicating three character pairs (3 bytes) to follow.
00 00	Four-character 2-byte address field, zeros.
FC	Checksum of S9-record.

APPENDIX B

HEX-RECORD INFORMATION

INTRODUCTION

Intel's Hex-record format allows program or data files to be encoded in a printable (ASCII) format. This allows viewing of the object file with standard tools and easy file transfer from one computer to another, or between a host and target. An individual Hex-record is a single line in a file composed of many Hex-records.

HEX-RECORD CONTENT

Hex-Records are character strings made of several fields which specify the record type, record length, memory address, data, and checksum. Each byte of binary data is encoded as a 2-character hexadecimal number: the first ASCII character representing the high-order 4 bits, and the second the low-order 4 bits of the byte.

The 6 fields which comprise a Hex-record are defined as follows:

Field	**Character**	**Contents**
Start code	1	An ASCII colon, ":".
Byte count	2	The count of the character pairs in the data field.
Address	4	The 2-byte address at which the data field is to be loaded into memory.
Type	2	00, 01, or 02.
Data	0-2n	From 0 to n bytes of executable code, or memory loadable data. n is normally $20 (32 decimal) or less.
Checksum	2	The least significant byte of the two's complement sum of the values represented by all the pairs of characters in the record except the start code and checksum.

Each record may be terminated with a CR/LF/NULL. Accuracy of transmission is ensured by the byte count and checksum fields.

HEX-RECORD TYPES

There are three possible types of Hex-records.

00 A record containing data and the 2-byte address at which the data is to reside.

01 A termination record for a file of Hex-records. Only one termination record is allowed per file and it must be the last line of the file. There is no data field.

02 A segment base address record. This type of record is ignored by the LP120.

HEX-RECORD EXAMPLE

Following is a typical Hex-record module consisting of four data records and a termination record.

```
:10010000214601360121470136007EFE09D2190140
:100110002146017EB7C20001FF5F16002148011983
:10012000194E79234623965778239EDA3F01B2CAA7
:100130003F0156702B5E712B722B732146013421C7
:00000001FF
```

The first data record is explained as follows:

: Start code.

10 Hex 10 (decimal 16), indicating 16 data character pairs, 16 bytes of binary data, in this record.

01 Four-character 2-byte address field: hex address 0100,
00 indicates location where the following data is to be loaded.

00 Record type indicating a data record.

The next 16 character pairs are the ASCII bytes of the actual program data.

40 Checksum of the first Hex-record.

The termination record is explained as follows:

: Start code.

00 Byte count is zero, no data in termination record.

00 Four-character 2-byte address field, zeros.
00

01 Record type 01 is termination.

FF Checksum of termination record.

APPENDIX C

SERIAL INTERFACE CONNECTOR

The table below shows the most commonly implemented signals and their pin assignments in accordance with RS-232D. Host computers are usually DTE (Data Terminal Equipment) and modems are DCE (Data Communications Equipment). Note that circuits are named from the point of view of the DTE. For example, circuit BB (receive data) is actually data transmitted by the DCE. DCE devices normally use a 25 pin, female, D connector.

Pin	Circuit	Description	Direction
1	AA	Protective ground, PG	n/a
2	BA	Transmit data, TD	to DCE
3	BB	Receive data, RD	from DCE
4	CA	Request to send, RTS	to DCE
5	CB	Clear to send, CTS	from DCE
6	CC	Data set ready, DSR	from DCE
7	AB	Signal ground, SG	n/a
8	CF	Data carrier detect, DCD	from DCE
20	CD	Data Terminal ready, DTR	to DCE
22	CE	Ring indicator, RI	from DCE

Many IBM and compatible PCs use a 9 pin, male, D connector instead of the 25 pin connector. The pin assignments for such a DTE device are shown below.

Pin	Circuit	Description	Direction
1	CF	Data carrier detect, DCD	from DCE
2	BB	Receive data, RD	from DCE
3	BA	Transmit data, TD	to DCE
4	CD	Data Terminal ready, DTR	to DCE
5	AB	Signal ground, SG	n/a
6	CC	Data set ready, DSR	from DCE
7	CA	Request to send, RTS	to DCE
8	CB	Clear to send, CTS	from DCE
9	CE	Ring indicator, RI	from DCE

All Lucid Technologies programmers are designed as DCE devices. They use a 9 pin, female, D connector that is directly compatible with 9 pin COM ports found on most IBM compatibles. The pin assignments for this connector are shown below.

Pin	Circuit	Description	Direction
1	CF	Data carrier detect, DCD	from programmer
2	BB	Receive data, RD	from programmer
3	BA	Transmit data, TD	to programmer
4	CD	Data Terminal ready, DTR	to programmer
5	AB	Signal ground, SG	n/a
6	CC	Data set ready, DSR	from programmer
7	CA	Request to send, RTS	to programmer
8	CB	Clear to send, CTS	from programmer
9	CE	Ring indicator, RI	from programmer

None of the handshake lines are actively controlled by the programmer. DTR is not connected and thus is ignored. DSR and DCD are wired to the ON condition (ON = spacing = +voltage) at all times. RTS is received, buffered, and looped back to the host as CTS; thus CTS tracks RTS.

APPENDIX D

COMMUNICATIONS SETUP

Setting up your communications software to work with a new piece of hardware can be a frightening prospect. There are so many options to keep track of, it is often hard to tell what configuration you're using. No matter what communications software you use, the setup requirements, outlined below, will be similar.

1. Terminal emulation - Use a simple terminal like ANSI or TTY
2. Comm port - Select the appropriate serial port
3. Connection type - Select a DIRECT, not a modem, connection
4. Carrier detect - Do not monitor carrier detect
5. Baud rate - Select the programmer's baud rate
6. Data bits - Select 8 data bits
7. Stop bits - Select 1 stop bit
8. Parity - Select no parity
9. Duplex - Select full duplex
10. ASCII transfer - Select line pacing at approximately 100 milliseconds
    Set character pacing at approximately 1 millisecond
    Do not translate CR or LF
    No flow control

To help get you started, two example programs are included on your distribution disk. LUCID.ASP is an aspect command program for ProComm Plus. This is an ASCII file that can be viewed with any text editor. LUCID.TRM is for use with the Terminal program that comes with Windows3.X. If you use either of these programs remember to change the comm port and baud rate settings to match your equipment. Hyperterm that comes with Windows 95/98 will also work.

APPENDIX E

LP120 PARTS LIST (1)

Quan.	Reference	Part
Semiconductors		
2	D1,D2	1N5819, 40V Schottky diode
1	D3	1N5818, 30V Schottky diode
2	Q1,Q2	TIP30, TO-220, PNP
1	Q3	2N2907A, TO-18 or TO-92, PNP
1	Q4	2N3904, TO-92, NPN
1	U1	MC6803, microcomputer (2)
1	U2	74HCT573, octal transparent latch
1	U3	62256, 32k x 8 CMOS RAM
1	U4	27128, 16k x 8 EPROM
1	U5	6264, 8k x 8 CMOS RAM
1	U6	PAL16R4, address decoder
2	U7,U8	6821, peripheral interface adapter
1	U9	74HCT393, dual 4-bit binary counter
1	U10	74HC14, hex inverter
2	U11,U12	78S40, switching regulator
1	U13	MAX232, RS-232 interface
1	U14	MAX522CPA, dual 8-bit DAC
1	U15	TL431ACLP, TO-92, voltage reference
Capacitors		
2	C1,C2	20p disk
1	C3	680p
1	C4	560p
6	C5,C13-C17	0.1u disk
4	C6,C7,C8,C9	1u, 16V radial
1	C10	10u, 16V radial
1	C11	100u, 50V radial
1	C12	100u, 35V radial
1	C18	1u, 50V radial
Resistors (0.25W, 5% unless noted otherwise)		
2	R1,R2	0.33 ohm (or less), 0.5W
2	R3,R4	47k (yellow-violet-orange-gold)
2	R5,R6	1k (brown-black-red-gold)
2	R7,R15	3.3k (orange-orange-red-gold)
1	R8	3.6k (orange-blue-red-gold)
2	R9,R10	1k 3/4" trimmer

1	R11	3k (orange-black-red-gold)
1	R12	8.2k (gray-red-red-gold)
1	R13	33k (orange-orange-orange-gold)
3	R14,R24,R25	180, 0.5W (brown-gray-brown-gold)
3	R16,R17,R22	1.1k (brown-brown-red-gold)
1	R18	4.7K (yellow-violet-red-gold)
1	R19	390 (orange-white-black-gold)
1	R20	100k (brown-black-yellow-gold)
1	R21	820 (gray-red-brown-gold)
1	R23	2.2K (red-red-red-gold)
3	RN1,RN2,RN3	4.7k, 8-pin SIP, pin 1 common

Sockets

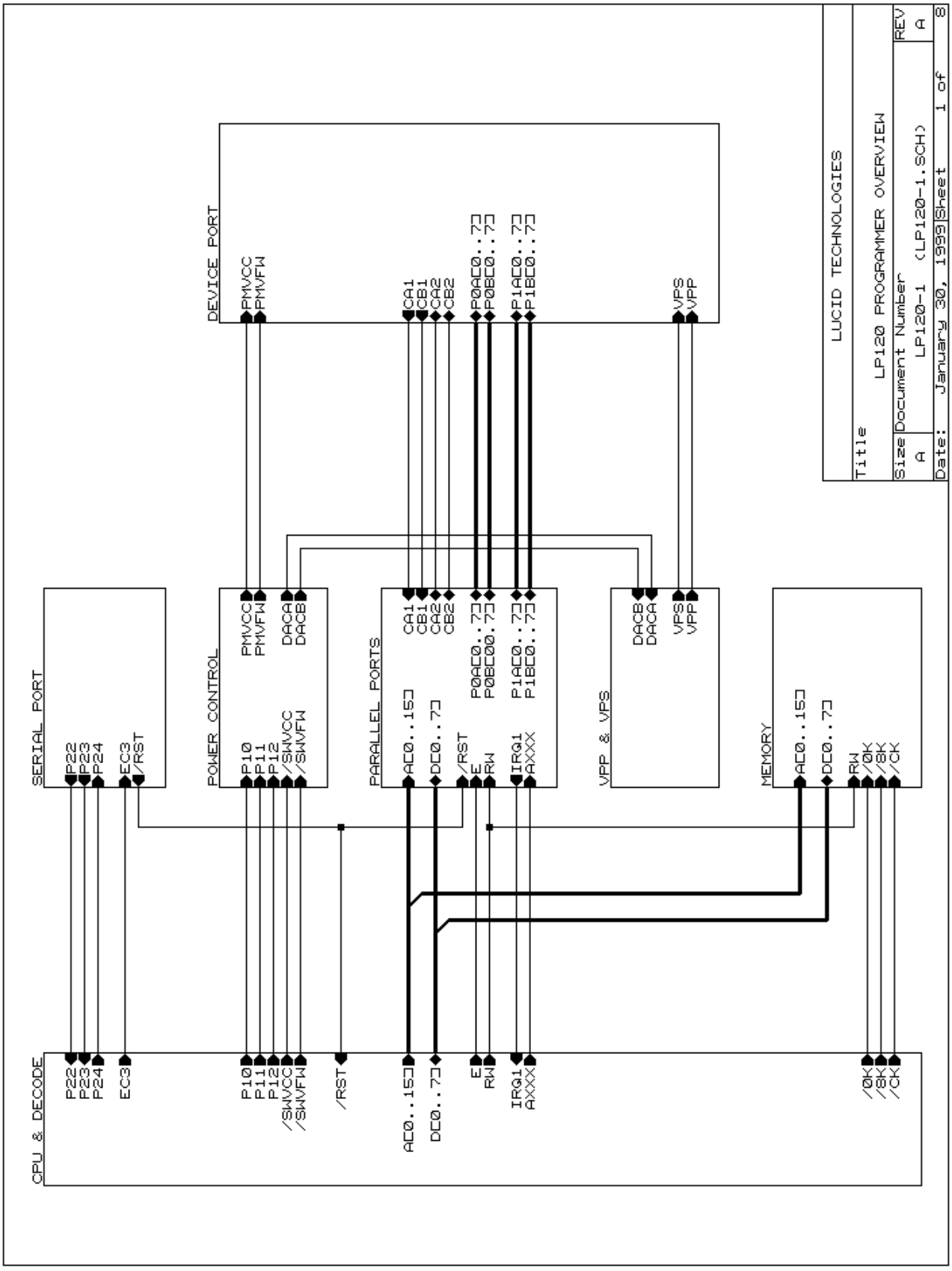
1	U14	8 pin
2	U9,U10	14 pin
3	U11,U12,U13	16 pin
2	U2,U6	20 pin
3	U3,U4,U5	28 pin
3	U1,U7,U8	40 pin

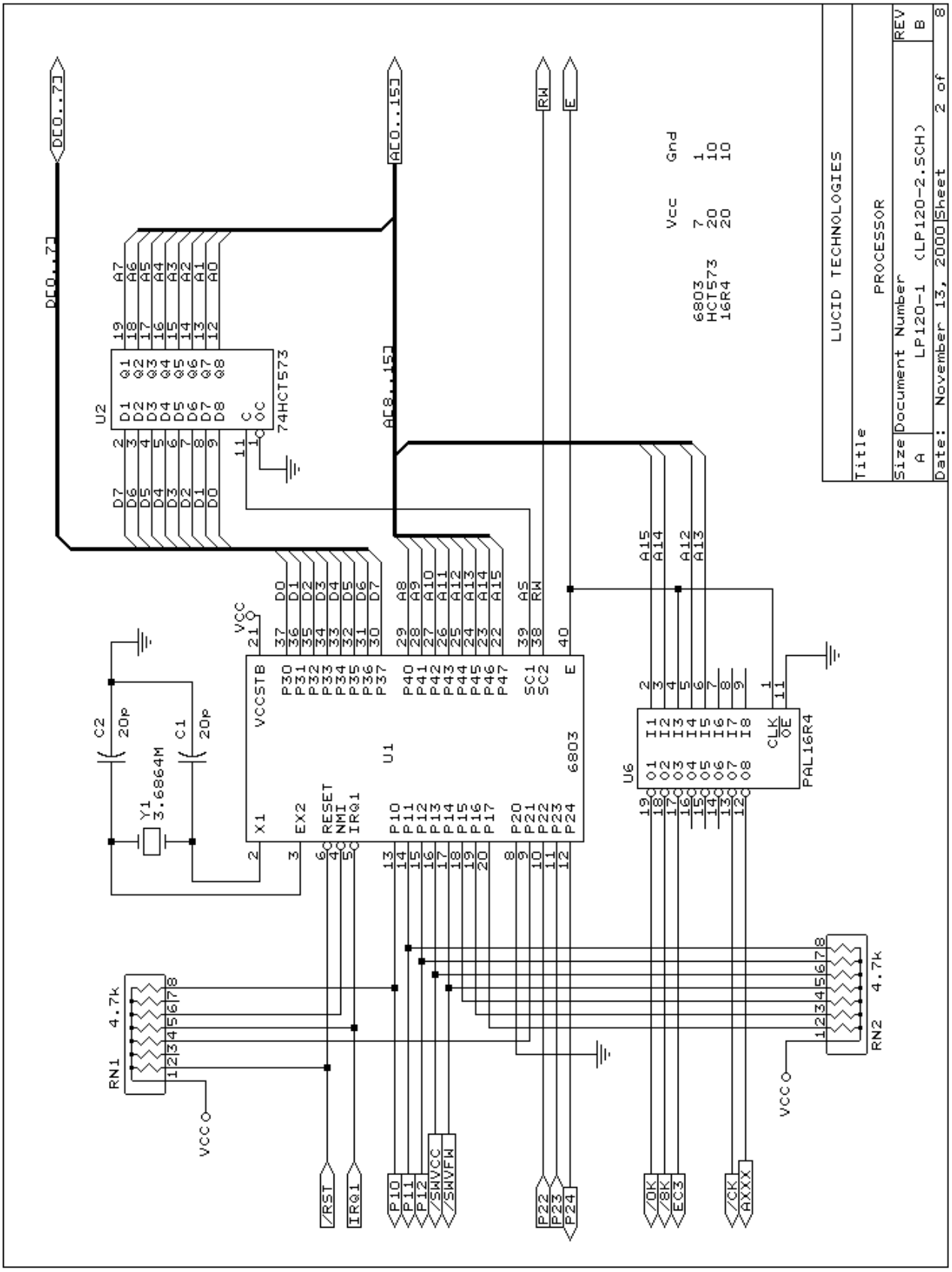
Miscellaneous

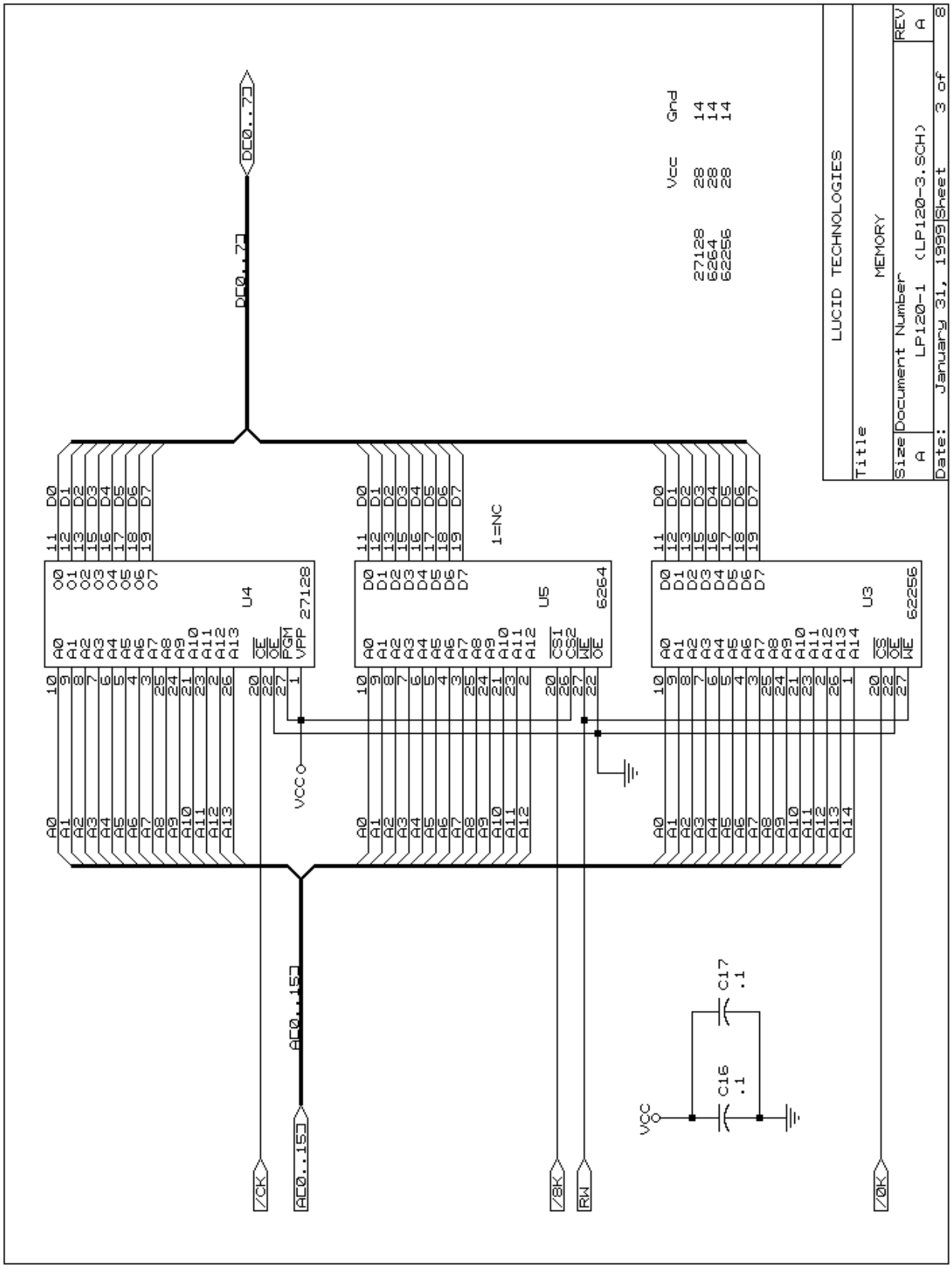
1	J1	3 x 0.2" (5.08mm) terminal strip
1	J2	DB-9S, right angle, PC mount
1	J3	22/44 edge connector, 0.156" contact x 0.2" row spacing
1	JP1	Double row jumper header, 5 x 2
1	L1	100 uH coil
1	L2	47 uH coil
1	Y1	3.6864 MHz crystal, HC-18 or HC-49
1	JP1	Shorting jumper

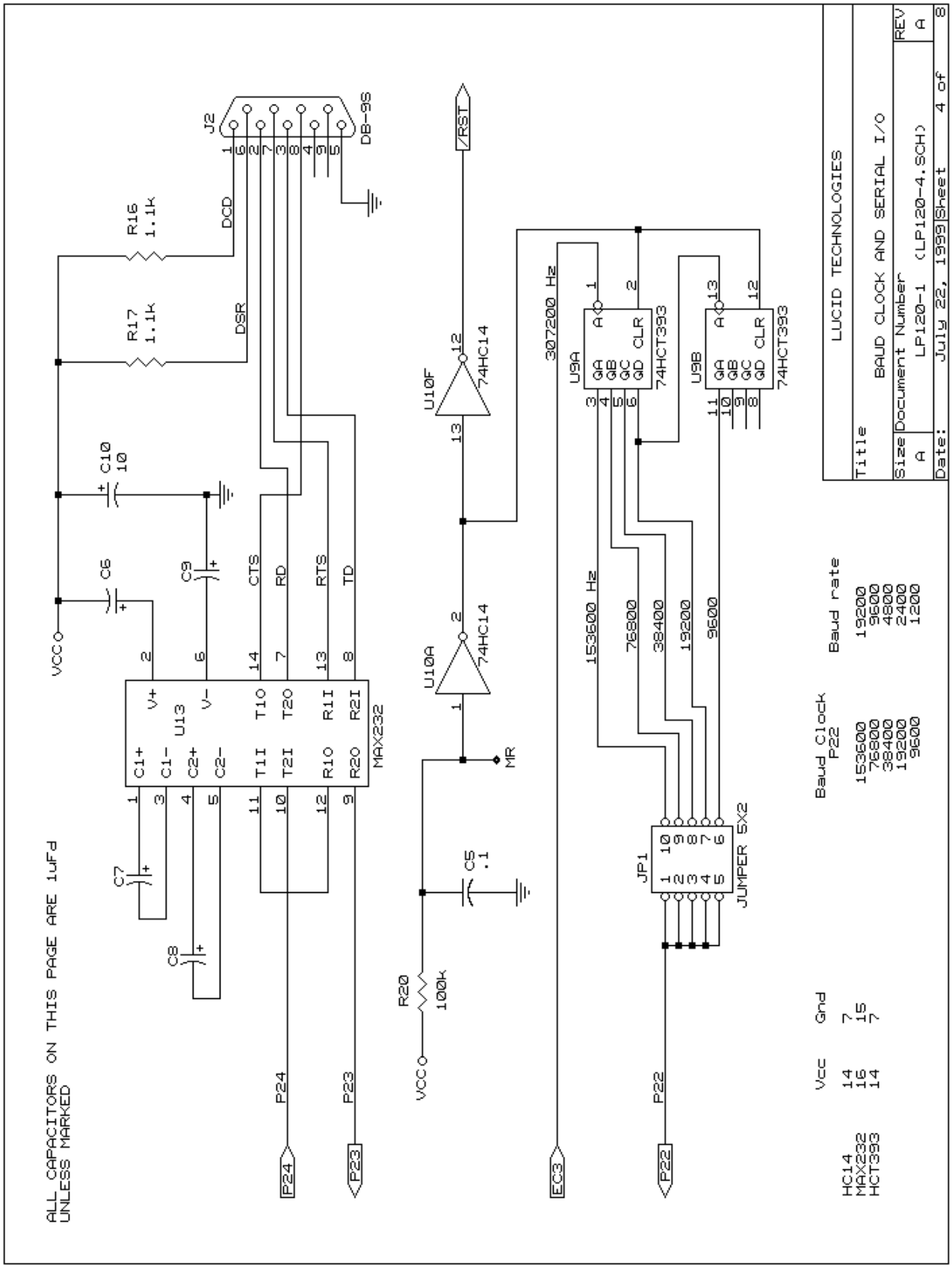
(1) A list of possible sources for these parts is found in the file PARTS.SRC on the distribution disk.

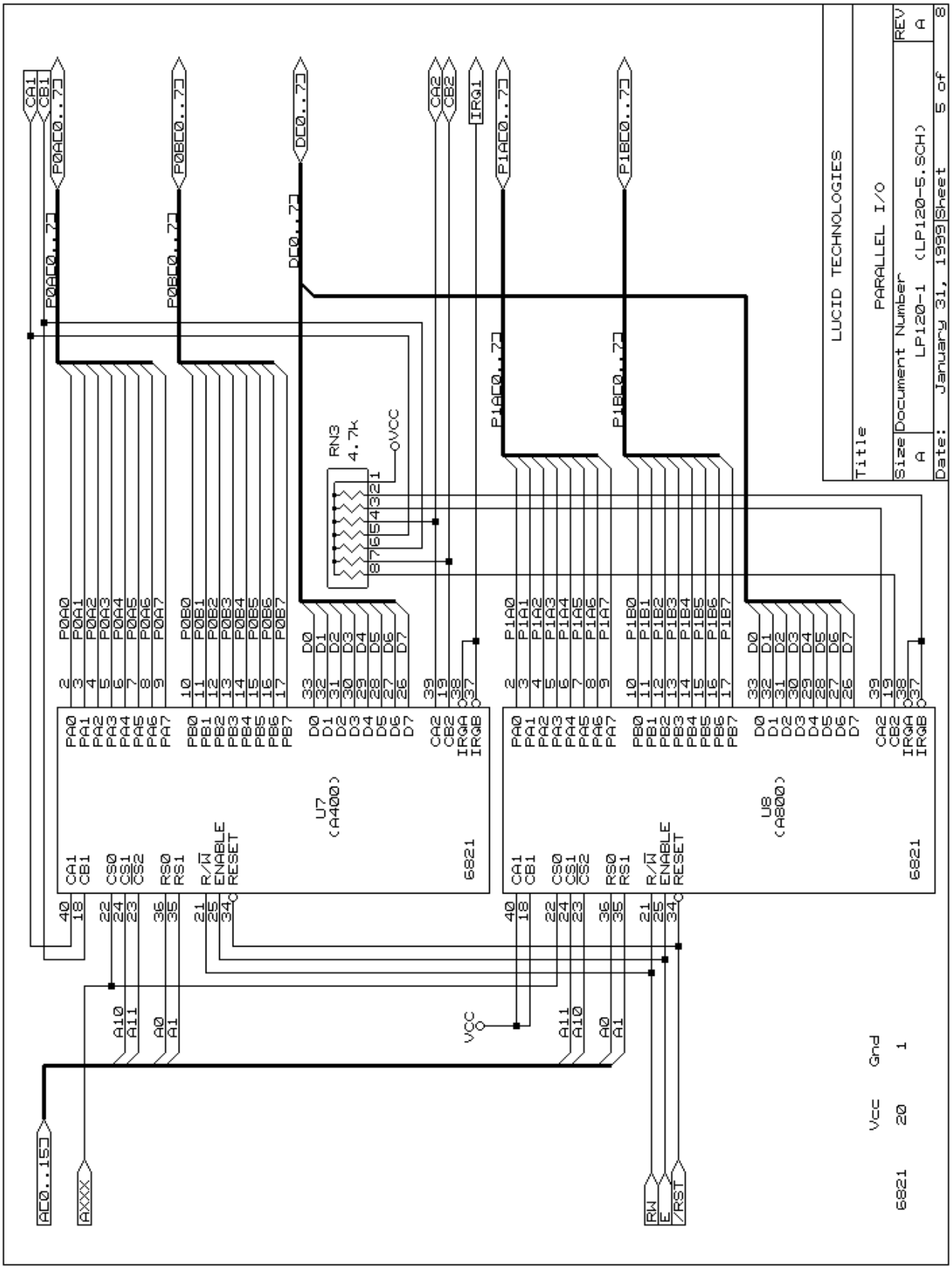
(2) The MPU chip may be either a 6801, a 6803, or a 6303R.

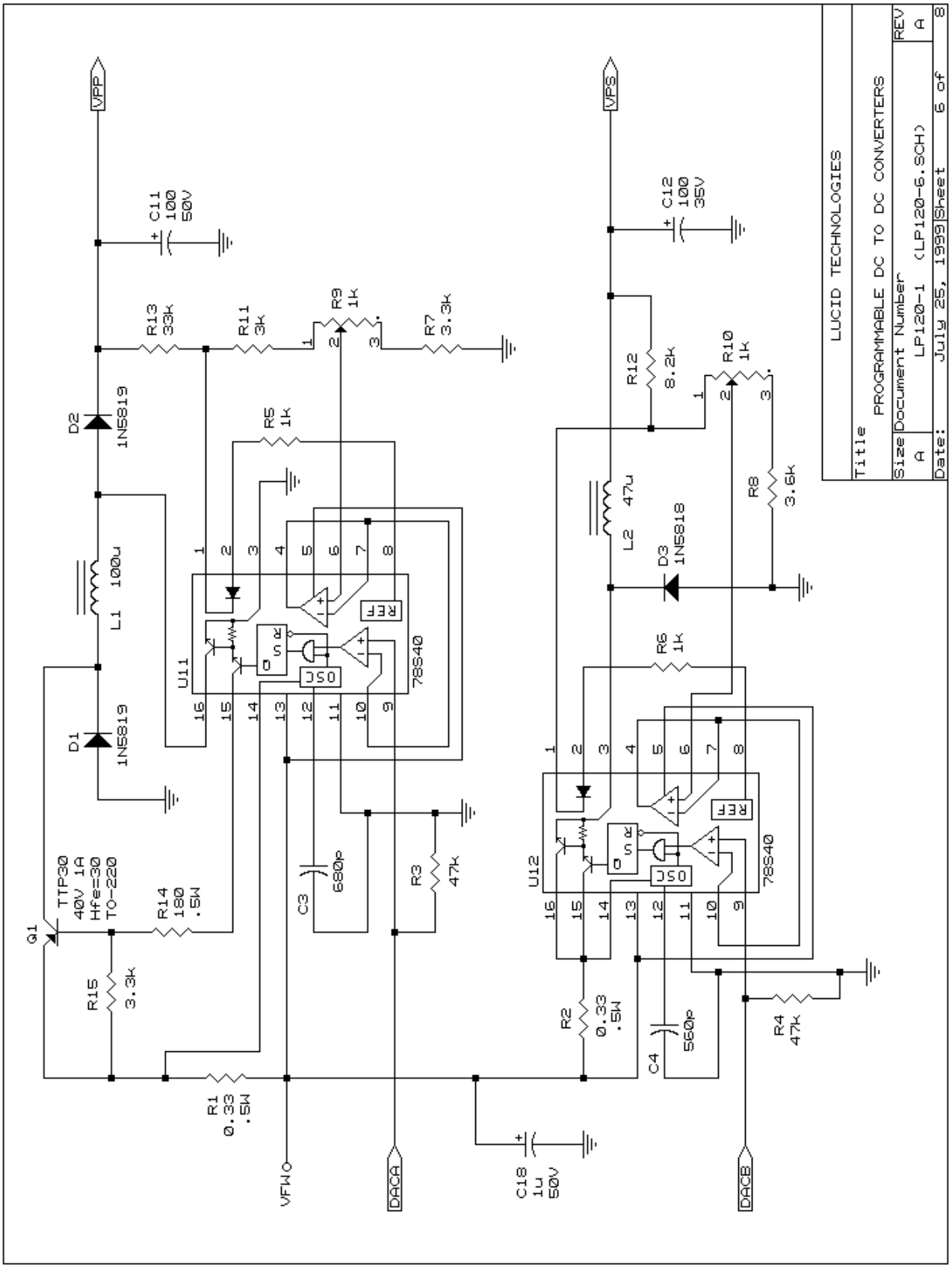

LUCID TECHNOLOGIES
Title
LP120 PROGRAMMER OVERVIEW
Size
A
Document Number
LP120-1 (LP120-1.SCH)
REV
A
Date: January 30, 1999
Sheet 1 of 8
DEVICE PORT
PMVCC
PMVFW
CA1
CB1
CA2
CB2
P0AC0..7
P0BC0..7
P1AC0..7
P1BC0..7
VPS
VPP
SERIAL PORT
P22
P23
P24
EC3
/RST
POWER CONTROL
P10
P11
P12
/SWVCC
/SWVFW
DACA
DACB
PARALLEL PORTS
AC0..15
DC0..7
/RST
E
RW
IRQ1
AXXX
VPP & VPS
MEMORY
/OK
/BK
/CK
CPU & DECODE

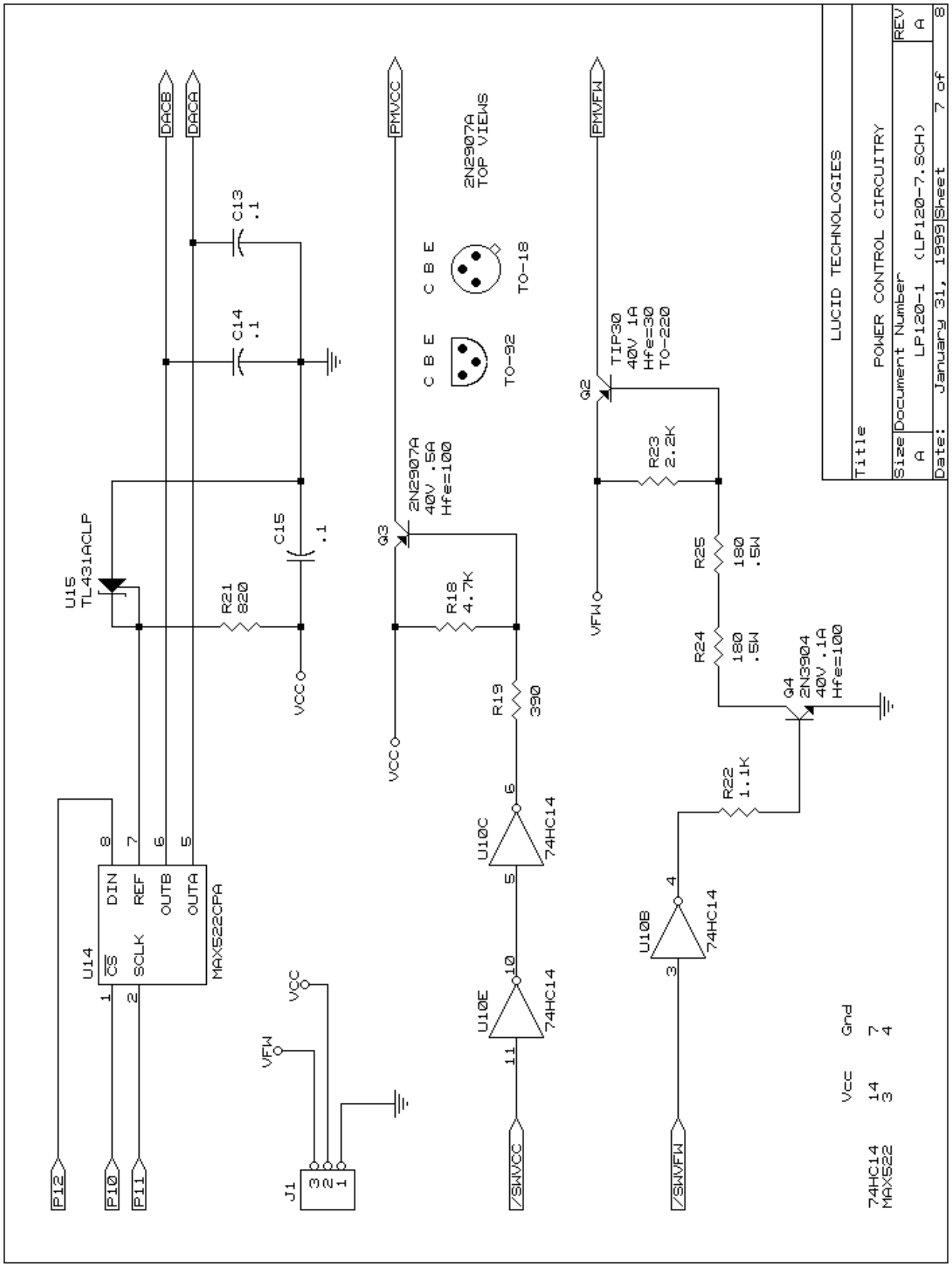

LUCID TECHNOLOGIES
Title
PROCESSOR
Size A
Document Number LP120-1 (LP120-2.SCH)
REV B
Date: November 13, 2000 Sheet 2 of 8
U1
6803
U2
74HCT573
U6
PAL16R4
RN1 4.7k
RN2 4.7k
Y1 3.6864M
C1 20p
C2 20p
VCC
Gnd
RW
E
/RST
IRQ1
/SWVCC
/SWVFW
/CK
AXXX
/8K
EC3
/OK

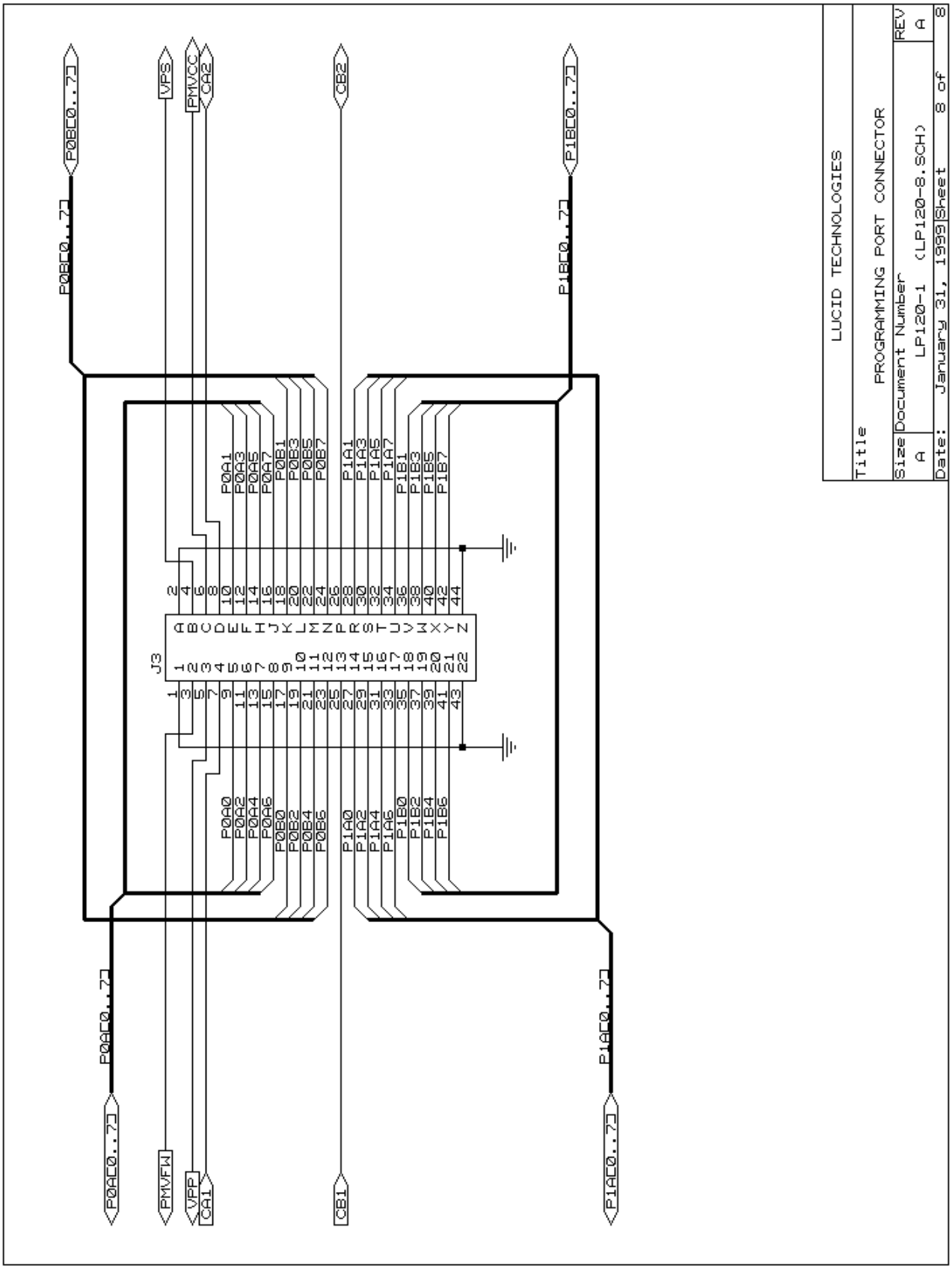

DC0..7
AC0..15
/CK
/8K
RW
/0K
VCC
C16 .1
C17 .1
U4
U5
U3
27128
6264
62256
1=NC
Vcc Gnd
27128 28 14
6264 28 14
62256 28 14
LUCID TECHNOLOGIES
Title
MEMORY
Size A
Document Number LP120-1 (LP120-3.SCH)
REV A
Date: January 31, 1999 Sheet 3 of 8

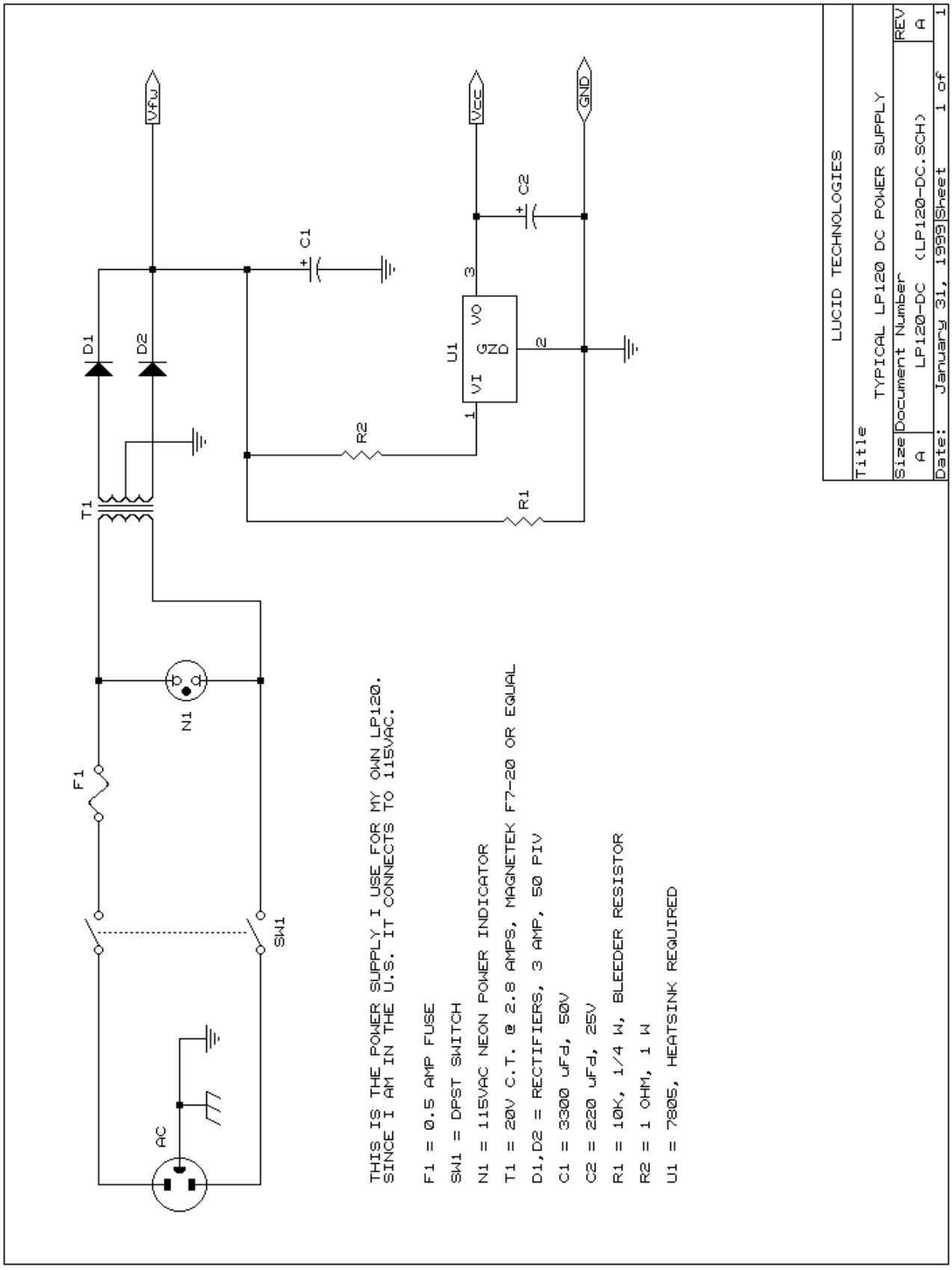

LUCID TECHNOLOGIES
Title
BAUD CLOCK AND SERIAL I/O
Size A
Document Number LP120-1 (LP120-4.SCH)
REV A
Date: July 22, 1999 Sheet 4 of 8
ALL CAPACITORS ON THIS PAGE ARE 1uFd UNLESS MARKED
J2
DB-9S
R16 1.1k
R17 1.1k
DCD
DSR
C10 10
C6
C9
C7
C8
U13
MAX232
V+
V-
C1+
C1-
C2+
C2-
T1I
T2I
R1O
R2O
T1O
T2O
R1I
R2I
CTS
RD
RTS
TD
VCC
P24
P23
U10F
74HC14
U10A
74HC14
/RST
MR
C5 .1
R20 100K
307200 Hz
153600 Hz
U9A
U9B
74HCT393
QA
QB
QC
QD
CLR
76800
38400
19200
9600
JP1
JUMPER 5X2
P22
EC3
Baud Clock P22
Baud rate
153600 19200
76800 9600
38400 4800
19200 2400
9600 1200
Vcc Gnd
HC14 14 7
MAX232 16 15
HCT393 14 7


LUCID TECHNOLOGIES
Title
PARALLEL I/O
Size A
Document Number LP120-1 (LP120-5.SCH)
REV A
Date: January 31, 1999 Sheet 5 of 8
U7 (A400)
U8 (A800)
6821
RN3 4.7K
VCC
Gnd
CA1
CB1
CA2
CB2
IRQ1
P0AC0..7
P0BC0..7
DC0..7
P1AC0..7
P1BC0..7
AC0..15
AXXX
RW
E
/RST


LUCID TECHNOLOGIES
Title
PROGRAMMABLE DC TO DC CONVERTERS
Size A
Document Number LP120-1 (LP120-6.SCH)
REV A
Date: July 25, 1999 Sheet 6 of 8
VPP
VPS
VFWD
DACA
DACB
C11 100 50V
C12 100 35V
R13 33K
R11 3K
R9 1K
R7 3.3K
R12 8.2K
R10 1K
R8 3.6K
R5 1K
R6 1K
D2 1N5819
D1 1N5819
D3 1N5818
L1 100u
L2 47u
U11
U12
78S40
OSC
REF
Q1 TIP30 40V 1A Hfe=30 TO-220
R14 180 .5W
R15 3.3K
R1 0.33 .5W
R2 0.33 .5W
R3 47K
R4 47K
C3 680p
C4 560p
C18 1u 50V


LUCID TECHNOLOGIES
Title
POWER CONTROL CIRCUITRY
Size A
Document Number LP120-1 (LP120-7.SCH)
REV A
Date: January 31, 1999 Sheet 7 of 8
U14 MAX522CPA
DIN
REF
OUTB
OUTA
SCLK
CS
U15 TL431ACLP
U10B 74HC14
U10C 74HC14
U10E 74HC14
Q2 TIP30 40V 1A Hfe=30 TO-220
Q3 2N2907A 40V .5A Hfe=100
Q4 2N3904 40V .1A Hfe=100
2N2907A TOP VIEWS
C B E C B E
TO-92
TO-18
R18 4.7K
R19 390
R21 820
R22 1.1K
R23 2.2K
R24 180 .5W
R25 180 .5W
C13 .1
C14 .1
C15 .1
DACB
DACA
PMVCC
PMVFW
VCC
VFW
/SWVCC
/SWVFW
P12
P10
P11
J1
74HC14 Vcc 14 Gnd 7
MAX522 3 4


LUCID TECHNOLOGIES
Title
PROGRAMMING PORT CONNECTOR
Size
A
Document Number
LP120-1 (LP120-8.SCH)
REV
A
Date: January 31, 1999
Sheet 8 of 8
J3
P0BC0..7
P1BC0..7
P0AC0..7
P1AC0..7
VPS
PMVCC
CA2
CB2
PMVFW
VPP
CA1
CB1


LUCID TECHNOLOGIES
Title
TYPICAL LP120 DC POWER SUPPLY
Size
A
Document Number
LP120-DC (LP120-DC.SCH)
REV
A
Date: January 31, 1999
Sheet 1 of 1
THIS IS THE POWER SUPPLY I USE FOR MY OWN LP120.
SINCE I AM IN THE U.S. IT CONNECTS TO 115VAC.
F1 = 0.5 AMP FUSE
SW1 = DPST SWITCH
N1 = 115VAC NEON POWER INDICATOR
T1 = 20V C.T. @ 2.8 AMPS, MAGNETEK F7-20 OR EQUAL
D1,D2 = RECTIFIERS, 3 AMP, 50 PIV
C1 = 3300 uFd, 50V
C2 = 220 uFd, 25V
R1 = 10K, 1/4 W, BLEEDER RESISTOR
R2 = 1 OHM, 1 W
U1 = 7805, HEATSINK REQUIRED
AC
F1
N1
SW1
T1
D1
D2
C1
R2
R1
U1
VI
GND
VO
C2
Vfw
Vcc
GND

APPENDIX G

LP120 circuit board