2801Prog User and Service Manual

www.6540rom.com

Overview:

The 2801Prog is a compact, low-cost programmer for the Motorola® MCM2801P (2801) EEPROM. The 2801 is a first-generation EEPROM requiring a high voltage (+25V) for programming and thus is not supported by commercial programmers. However, it was widely used in its day and thus a reliable method of reading and programming it is required for the maintenance of legacy equipment. For example, the 2801 is used to store the model differentiation and calibration data for the Mettler® PE series of laboratory balances that are still widely used in laboratories around the world.

Installation

Preparing the 2801Prog for use:

1) Connect the DB9 RS-232 connector to a standard PC serial port using a DB9 male to DB9 female straight-through (not null-modem) cable.
2) Plug the included power adapter into a 120VAC electrical outlet.
3) Plug the included power adapter into the 2801Prog.
4) Insert the MCM2801 IC to be read or programmed into the programmer socket such that pin 1 of the IC is oriented toward the top of the programmer (the end with the power and RS-232 connectors.) Note that the 2801Prog includes internal protection circuitry to avoid damage to either the programmer or the IC under test in the event that the 2801 is inserted backwards.

Using the 2801Prog with a standard terminal emulator

The 2801Prog is designed to allow use on any platform with a standard RS-232 port and terminal emulation software and does not require any special software or drivers. It has been tested in this mode with Minicom on Linux and Mac and with HyperTerminal on Microsoft® Windows® but should work with a wide variety of similar software. Furthermore, for ease of use a GUI based software driver package is provided. See the next section for usage directions for this GUI software.

To use the 2801Prog with a standard terminal emulator, configure the terminal emulator as follows:

```
BAUD: 9600
Data Bits: 8
Parity: None
Stop Bits: 1
Hardware Flow Control: Off
Software Flow Control: Off
```

When power is applied to the 2801Prog, the following text should appear:

```
2801Prog MCM2801 EEPROM Reader and Programmer
(c)2012 Matthew D'Asaro under the terms of the GPL
Type 'h' for help
2801Prog>
```

This indicates that the 2801Prog is correctly connected and ready for use. If trouble is encountered, recheck the serial port settings and connections, then disconnect and reconnect power to the MCM2801 to reset it.

The 2801Prog accepts the following commands. Note that only the first letter of each command is used to differentiate it from other commands, so 'e' 'erase' and 'elkfdgljhsg' will all erase the EEPROM.

'h' or 'help' – Displays a list of commands and their function.
'r' or 'read' – Reads the EEPROM and places its contents into the buffer.
'v' or 'verify' – Verifies that the buffer contents match the EEPROM contents. In the event that they do not, the actual contents of the EEPROM are displayed. Note that this command does not modify the buffer contents in any way.
'p' or 'program' – Programs the contents of the buffer into the EEPROM. Note that the 2801 does not need to be erased before programming. Verification that the EEPROM data matches the buffer after programming is not performed automatically.
'e' or 'erase' – Erases the EEPROM, filling it with zeros. Note that this command does not modify the buffer contents in any way.
'd' or 'display' – Displays the contents of the buffer on the screen in hexadecimal format.
'l' or 'load' – Allows the buffer to be filled. Buffer contents are entered in hexadecimal starting with address zero. Invalid characters are ignored and when the buffer is full (32 bytes) this command exits and the prompt reappears. Note that text can be pasted into the terminal program from a file saved locally on the computer to load the buffer without manually typing the contents. However, be sure to configure the terminal program to allow a delay between bytes, as entering the data too fast will crash the microcontroller in the 2801Prog.
'c' or 'clear' – Clears the buffer, filling it with all zeros.
'a' or 'automation' – Enters automation mode. In this mode all commands are still accepted but entered characters are not echoed back. This mode is used by the GUI interface to directly control the 2801Prog. Exit this mode by resetting the system using the ESC key. Note that entering this mode preserves the buffer contents, but the buffer is cleared when this mode is exited.
ESC – Pressing the ESC key at any time resets the system, clearing the buffer contents.

Using the 2801Prog with the GUI interface

The 2801Prog application provides a graphical interface to all of the features of the 2801Prog programmer. The source version has been tested on Windows, Linux and OS X, and compiled applications are provided for both Windows and OS X.

The user interface under Windows XP, Redhat Linux 6.0 and OS X 10.6 is shown in Figure 1. The buffer starts out initialized with all zeros, and no serial port is connected. The buffer window can switch between hexadecimal and binary through used of the radio buttons on the left hand side of the window. Note that if an incomplete number of nibbles has been entered in binary mode the input will be padded with zeros when switching to hexadecimal mode. Text entry is limited to 32 bytes of information (64 nibbles in hexadecimal mode, 256 bits in binary mode).

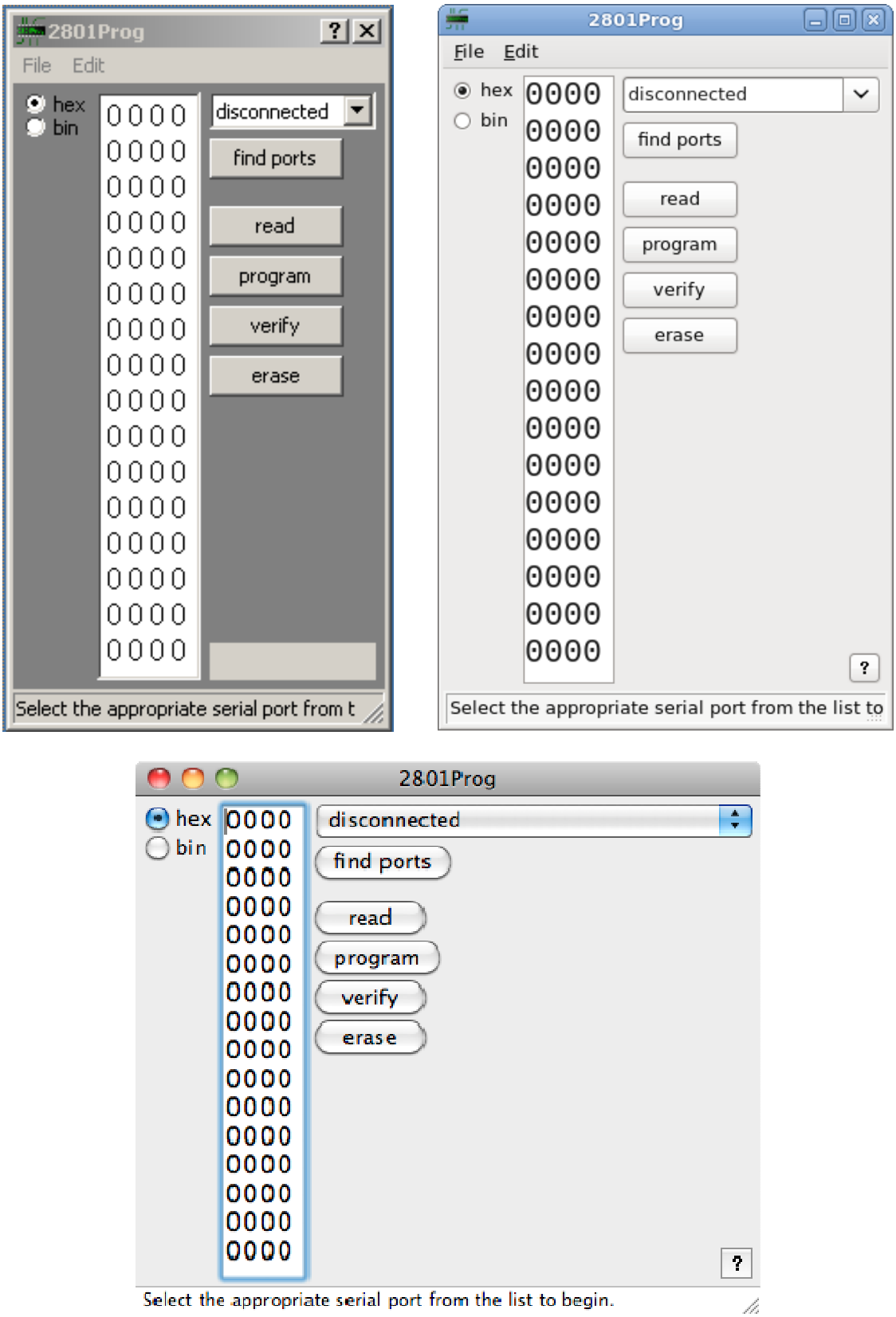


Figure 1: 2801Prog main window in Windows XP, Redhat Linux 6.0 and OS X 10.6

The buffer supports the standard cut, copy, paste and select all actions, accessed from the Edit menu and through standard keyboard shortcuts. Note that illegal characters are stripped when pasting into the buffer and the input is truncated if it would overfill the window. Legal characters are determined by the input mode you are in: the program assumes the clipboard contents match the input mode, so you CANNOT paste hexadecimal input while in binary entry mode.

You can open and save files in both raw binary and human-readable text format using the options under the File menu, as well as standard keyboard shortcuts. You can also drag-and-drop files onto the application window to open them.

When saving a file, the output format is determined by the file extension you give: text for ".txt" and binary for all other extensions. You can override this by explicitly selecting a format with the "Save as type" ("Format" on OS X) menubox in the save dialog. The output will be padded with zeros up to 32 bytes if the contents of the buffer are incomplete.

When opening a file, the type is again determined by the file extension -- text for ".txt" and binary for all other extensions. Reading a text file is similar to pasting into the buffer: all non-hexadecimal characters are stripped and only the first 64 nibbles are read. If the file contains less than 64 valid hexadecimal digits, the status indicator will turn yellow to warn you that the file came up short and blank spaces will be left in the buffer. Note that text files are always read assuming they contain hexadecimal representations, even when in binary entry mode.

You select the port to connect to using the menubox in the top right. If you connected your USB-to-serial converter after starting the program, press the "find ports" button to look for available serial ports again. Upon connecting to a port, the program checks to make sure it is talking to the right device by commanding the 2801Prog programmer to enter automation mode. Once a valid response is received, the status indicator will turn green and the status bar will report that you have successfully connected. See figure 2 below. To disconnect, select "disconnected" from the menubox.

Once connected, you can use the "read," "program," "verify" and "erase" buttons to program the chip you have loaded into the 2801Prog programmer.

“Read” will read the data on the currently-loaded chip and place the contents into the buffer, replacing the previous contents. If the read fails, the contents of the buffer will still be cleared.

“Program” will program the chip with the present contents of the buffer. If the entry into the buffer is incomplete, it will be padded with zeros. You will be notified if the contents of the chip following the read do not match the contents of the buffer.

“Verify” will check that the contents of the chip match the contents of the buffer. If the entry into the buffer is incomplete, it will be padded with zeros. You will be notified of the present contents of the chip if the contents differ.

“Erase” will erase the chip. You will be notified if the chip’s contents do not go to zero.

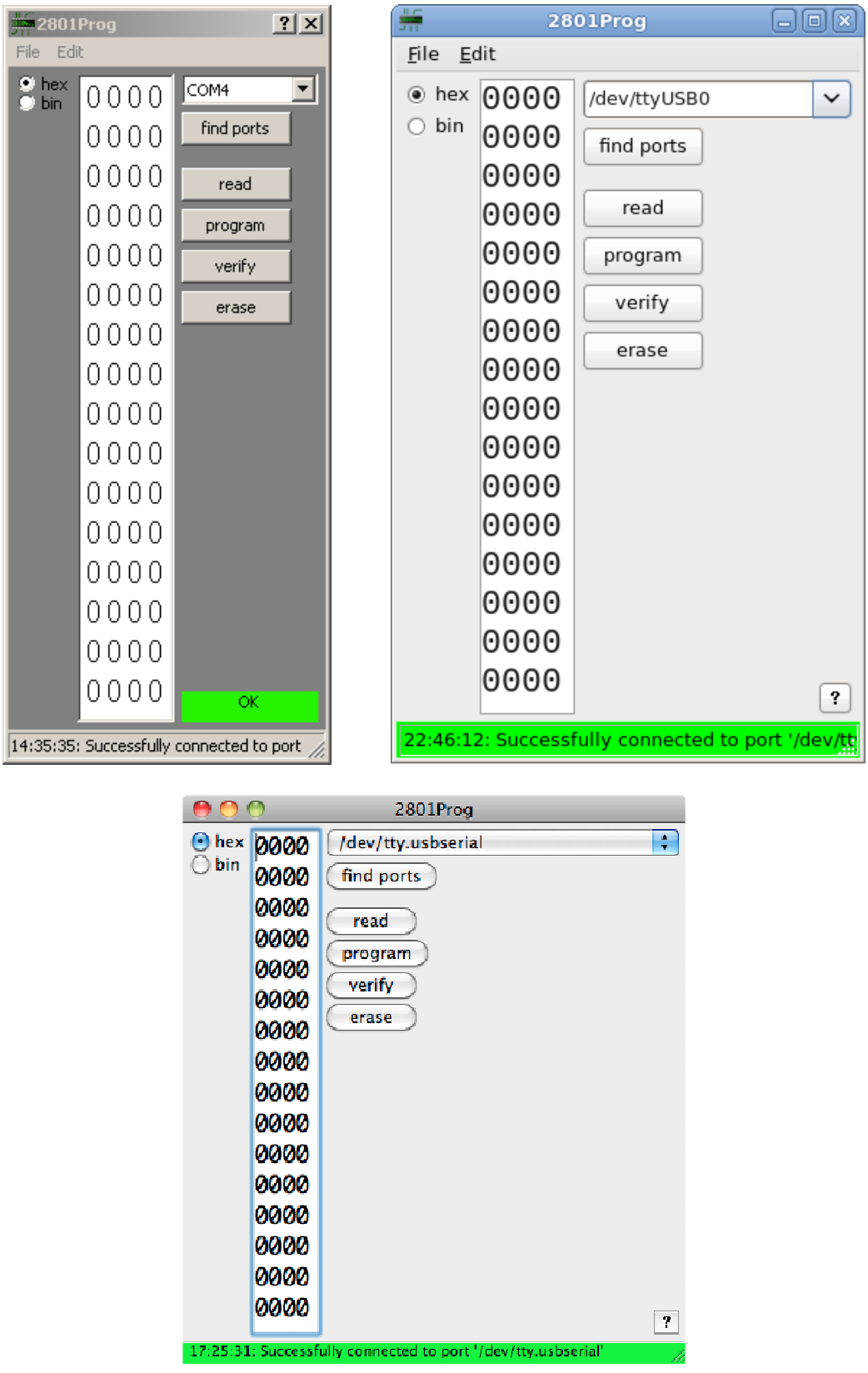


Figure 2: 2801Prog connected to programmer in Windows XP, Redhat Linux 6.0 and OS X 10.6

In all cases, the color of the status indicator indicates the results of the operation.

If you need help on any of the items, you can use the contextual help tool (question mark button) located in the top right of the Windows title bar and located in the bottom right corner of the other versions. To use, simply click the question mark button then click the item you want information on with the question mark cursor that appears.

Known Issues
Because of an apparent bug in wxPython on Ubuntu, the status bar may end up being drawn over the buffer when the program first starts. To fix this problem, drag the corner of the window to resize until the status bar is redrawn in the correct position, then drag the corner back to the desired size.

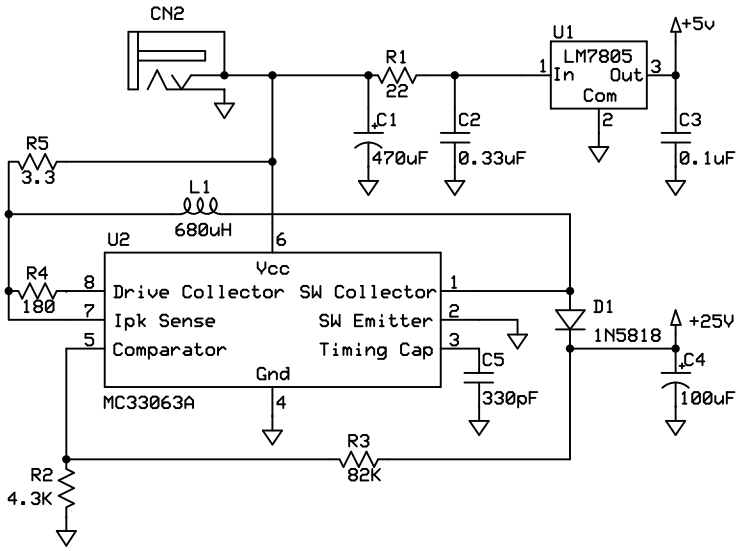
Because of an apparent bug in wxPython.TextCtrl on Linux, the Linux version will not be formatted properly upon entering binary editing mode. To fix the window, drag the corner of the window to resize until the scrollbar disappears, then drag the corner to put the window back to your desired size.

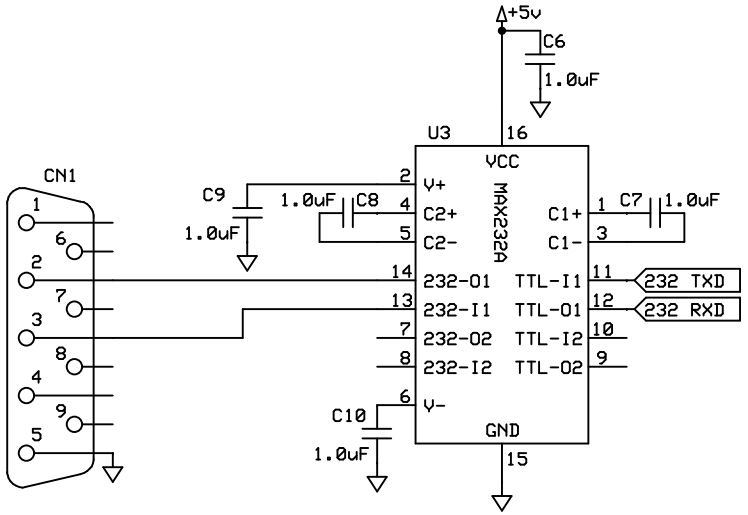
Running from Source
The program has the following dependencies, with the versions used for testing (under OS X 10.6.8) indicated:
Python 2.7.3
PySerial 2.6
wxPython 2.8.12.1

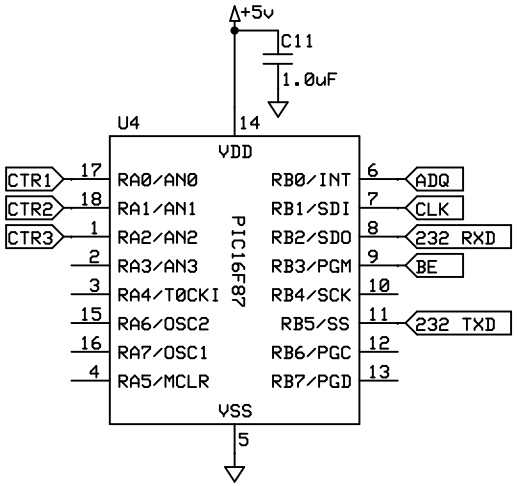
The software may work with other versions, but has not been tested. Once the dependencies have been installed, you can simply launch the program as:
python prog2801GuiWX.py

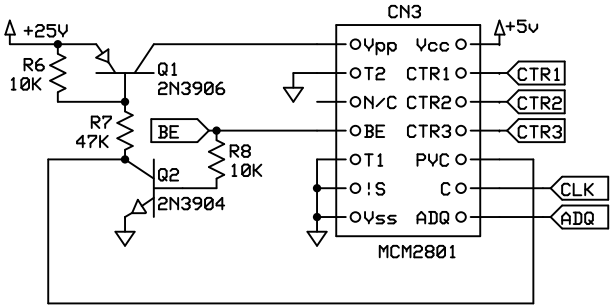
The program was built for distribution using py2app 0.7.2 and pyInstaller.

Servicing the Hardware
The 2801Prog is designed for long life and built with only top-quality components. Furthermore, it is backed by a simple lifetime guarantee – if it ever fails, just return it for a free repair. However, in the event that field service is required, a component layout diagram, schematic, and a complete parts list are included on the following pages.

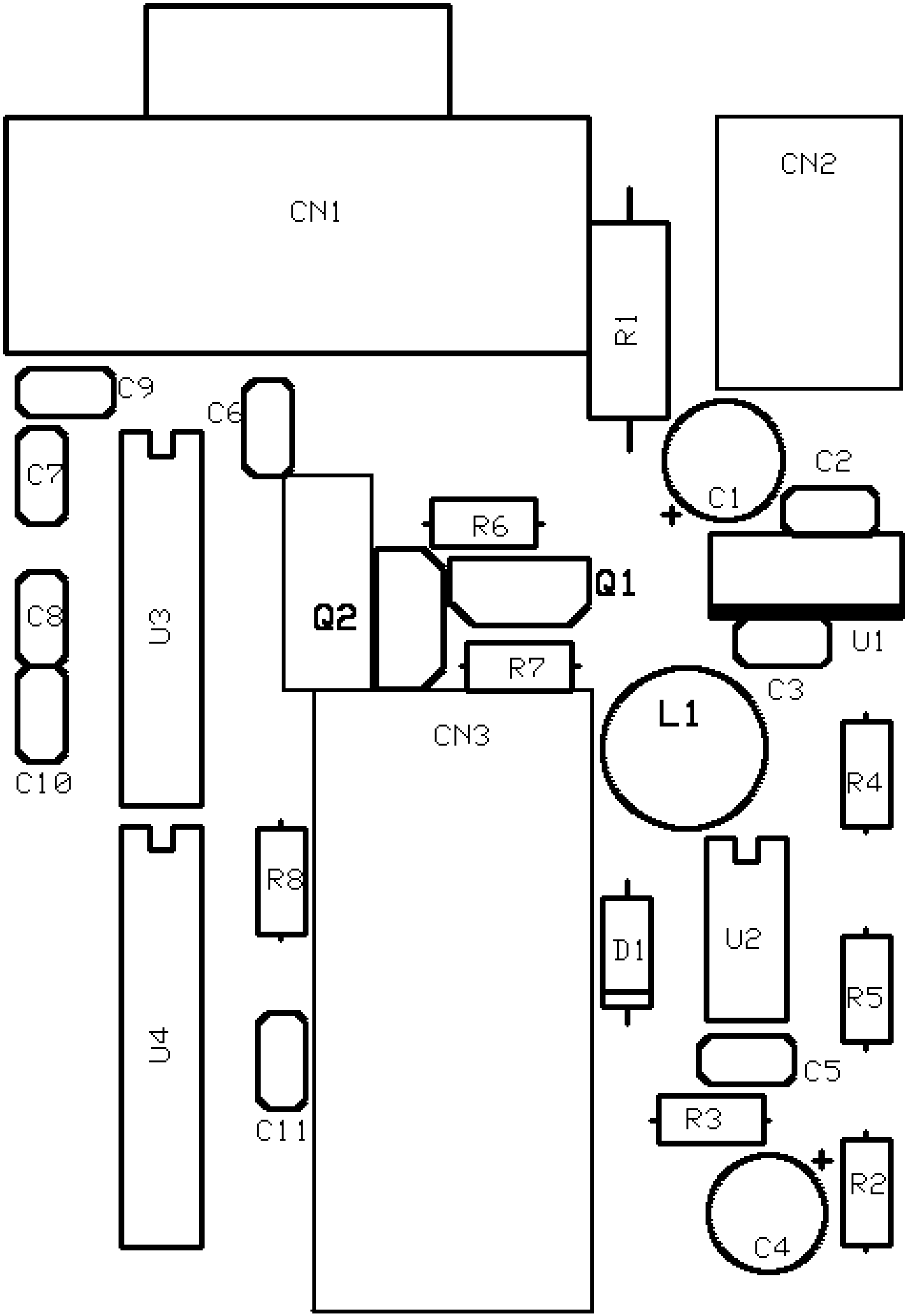








6540rom.com		
8601Prog EEPROM Programmer		
Matthew D'Asaro	Rev 0 12/3/2012	Page 1/1


CN1
CN2
R1
C9
C6
C7
C8
C10
U3
U4
C1
C2
U1
C3
R6
Q1
Q2
R7
CN3
L1
R4
R8
C11
D1
U2
R5
C5
R3
R2
C4

2801Prog Electrical Parts List, Page 1

Reference Designation	Quantity	Mfg.	Mfg. P/N	Mouser P/N	DigiKey P/N	Description / Notes
C1	1	Nichicon	UVZ1C221	647-UVZ1C221MED	493-1284-ND	220uF, 16V
C2	1	TDK	FK18X7R1E334K	810-FK18X7R1E334K	445-8424-ND	0.33uF, 25V
C3	1	Vishay	K104K15X7RF5TL2	594-K104K15X7RF5TL2	BC1084CT-ND	0.1uF, 50V
C4	1	Nichicon	UHV1V101MED	647-UHV1V101MED	UHV1V101MED-ND	100uF, 35V
C5	1	Vishay	K331K15X7RF5TL2	594-K331K15X7RF5TL2	BC1069TR-ND	330pF, 50V
C6 – C11	6	Vishay	K105Z20Y5VF5TL2	594-K105Z20Y5VF5TL2	BC1162CT-ND	1.0uF, 50V
CN1	1	TE Connectivity	5747844-5	571-5747844-5	A32118-ND	DB-9 Female
CN2	1	Kobiconn	163-179PH-EX	163-179PH-EX	N/A	Power Connector 2.1mm x 5.5mm
CN3	1	N/A	N/A	OBD	OBD	14-pin zif socket.
D1	1	STMicroelectronics	1N5818	511-1N5818	497-4548-3-ND	Schottky Diode, 1A, 30V
L1	1	Bourns	RLB0914-681KL	652-RLB0914-681KL	RLB0914-681KL-ND	Inductor, 680uH
Q1	1	Fairchild	2N3906TAR	512-2N3906TAR	2N3906TARFSCT-ND	PNP Transistor
Q2	1	Fairchild	2N3904TA	512-2N3904TA	2N3904TAFSCT-ND	NPN Transistor
R1	1	KOA Speer	MOS2CT52R220J	660-MOS2CT52R220J	OBD	22Ohms 2W 5% Metal Oxide
R2	1	KOA Speer	MF1/4DC4301F	660-MF1/4DC4301F	OBD	4.3K, 1/4W 1% Metal Film
R3	1	KOA Speer	MF1/4DCT52R8202F	660-MF1/4DCT52R8202F	OBD	82K, 1/4W 1% Metal Film
R4	1	Xicon	291-180-RC	291-180-RC	OBD	180Ohms 1/4W 5% Carbon Film
R5	1	Xicon	291-3.3-RC	291-3.3-RC	OBD	3.3Ohms 1/4W 5% Carbon Film

2801Prog Electrical Parts List, Page 2

Reference Designation	Quantity	Mfg.	Mfg. P/N	Mouser P/N	DigiKey P/N	Description / Notes
R6, R8	2	Xicon	291-10K-RC	291-10K-RC	OBD	10K 1/4W 5% Carbon Film
R7	1	Xicon	291-47K-RC	291-47K-RC	OBD	47K 1/4W 5% Carbon Film
U1	1	TI	LM7805CT/NOPB	926-LM7805CT/NOPB	OBD	Standard 7805 +5V Linear Voltage Regulator
U2	1	TI	MC33063AP	595-MC33063AP	296-17764-5-ND	Switching Regulator Controller
U3	1	TI	MAX232NE4	595-MAX232NE4	296-1402-5-ND	TTL to RS232 Converter
U4	1	Microchip	PIC16F87-I/P	579-PIC16F87-I/P	PIC16F87-I/P-ND	Microcontoller