PC2400 EEPROM PROGRAMMER

INSTRUCTION MANUAL
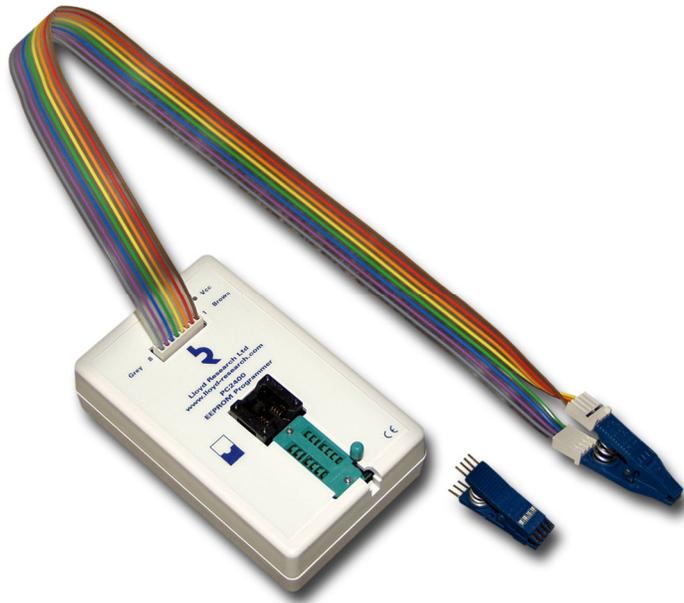
LAST UPDATE 23/09/09

PC2400

OPERATING INSTRUCTIONS

I N D E X

1. GENERAL

It is recommended that these instructions are read before attempting installation!

The PC2400 programs eeproms such as 24Cxx. It is fitted with a dual in line (DIL) ZIF socket and a SOIC socket for surface mount parts. There is also a socket for programming parts 'in circuit' for which a lead is provided.

The programmer has a USB port so that it can be connected to a Windows PC running current operating systems such as XP (Tablet), 2000 or Vista.

The PC2400 programmer is supplied complete with an installation CDROM which contains the code to run on the PC and the USB drivers.

The master data to program a new part can be read from another part or from a part 'in circuit'. *Note that*, unlike eproms, there is no need to erase eeproms before programming.

The PC2400 can also read data from eeproms in, for example, working TV sets placed in standby. This data can be stored on a PC's hard disc. Subsequently, the same data can be used to reprogram a 'corrupt' eeprom in a faulty TV set.

Screen Shot of PC2400 Programmer
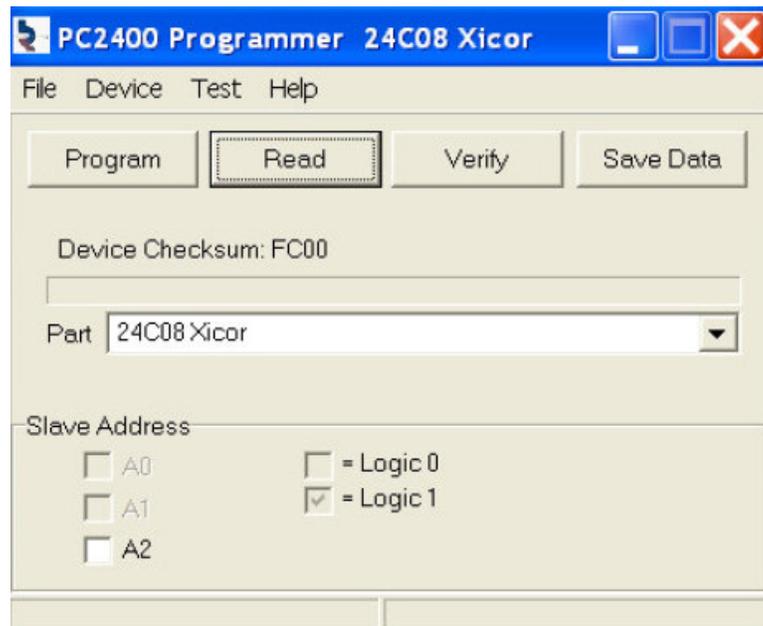
2. INSTALLATION

Do not connect the PC2400 programmer.

Put the installation CD into your CD drive and run the *SETUP.EXE* program. The software will by default be installed into the sub-directory –

c:\Program Files\Lloyd Research Ltd\PC2400

After installing the software, connect the programmer using the standard USB A to B lead supplied with the programmer. The Windows operating system will display a message to the effect that new hardware has been found. Windows should be able to automatically locate the first device driver on the installation disc. The PC2400 needs two device drivers. Therefore, after locating the first driver, the first process will be repeated to load the second one. The device drivers only need to be loaded once by Windows.

After the device drivers have been loaded, it should be possible to start the programmer. The screen will show that the programmer has been connected after a few seconds.

3. MAIN FUNCTIONS

a) Device Selection

The user can select a wide range of 24Cxx parts from the drop down menu. If the required part cannot be found, it is worth trying a similar part from another manufacturer. Unlike eproms, there is no danger of damaging a device because all of the currently supported devices can be programmed at 5V. However, some can be programmed and read at a lower voltage.

New devices can be added by editing the device text file (*TypeDef.txt*).

b) File Menu

Use this menu to select the file containing the data to be programmed into the device. The data in the file *MUST* be in one of the file formats recognised by the PC2400. The file formats supported are Intel hex, Motorola S19 or S28, binary and E2P. There is also an 'automatic' setting which, in most cases, is capable of distinguishing the file type. However, if this is unsuccessful, select the file type from the 'drop down' list. As the PC2400 is intended to be used for programming relatively small devices, it only understands Intel record types 00 and 01. Other record types are ignored. *Note that* the colon at the start of each Intel hex record must be in column 1. Other colons are ignored.

Extra information or notes can be included at the start of the file. When a file is 'opened', the PC2400 looks at the beginning of the file to see if there is a recommended device type and slave address. The device type is indicated as *<<24C16 National>>* or just *<<24C16>>* if the manufacturer is not important. The part number and manufacturer are case sensitive and *MUST* be exactly the same as they appear on the device list. The eprom slave address is indicated as *[[2]]* where 2 is the slave address. If a file with this information is opened, the program tells the user the recommended device type/slave address. In most cases, the slave address defaults to 0. *Note that* this facility is not available with binary files.

The file data is stored in the PC's memory. *Note that* the checksum of the file data is automatically displayed after opening and storing the file data. The checksum is calculated by adding each data byte into a 16-bit accumulator and ignoring the 'carries out'.

The file menu also contains a facility to view a file using Notepad.

The file menu can also be used to end the program.

c) Device Menu

Program or write: This function writes data from the PC's memory to the part in either one of the ZIF sockets or a part connected to the programmer via the remote cable.

Parts do *NOT* need to be erased before programming.

Parts should be positioned in the front section of either the DIL or SO socket.

Note that, when programming remote parts, the slave address can be specified (see *Slave Address* below).

After programming, a separate verify is performed automatically. There is no need to perform this function as a separate operation.

Note that, after programming, the device checksum is displayed. This can be different to the file checksum unless both the device and the file are identical sizes.

It is *NOT* possible to program an eeprom unless another eeprom has been read or a file has been opened.

Read: This function reads a device into the PC's memory with the device checksum being displayed. *Note that* a device wrongly inserted will probably read as 00 and that a missing device will probably be read as FF. The device checksum is displayed after a device has been read.

If a device has already been read, the PC software will display a message to this effect and allow the operator to either continue and, hence, overwrite the existing data or abort the operation.

Save data: This function allows the user to save data from a device to a disc file. The user can either upload to an existing file or create a new file. All saved files are in Intel hex format.

The data is taken from the PC's memory – not from the device. It is, therefore, essential to read the device data into the PC first.

Note that a warning message appears if a second device is read without saving the data from the first to disc.

Verify: This function just compares the data in the PC's memory to the device.

4. 'IN CIRCUIT' PROGRAMMING AND SLAVE ADDRESSES

The PC2400 can usually reprogram eeproms 'in circuit'. A typical application is to reprogram a 24Cxx eeprom in a TV set. In order to do this, place the set in 'standby' mode and fit the 'in circuit' lead using the DIL or surface mount clips. The chips should be fitted so that the wire colours follow the colour code, i.e. brown = 1, green = 5, etc.). When the chip has been fitted correctly, a red LED on the programmer should light indicating the presence of Vcc and Gnd. The PC2400 uses this measurement to apply correct logic signals (5V, 2.5V or 3.3V, etc.). *Note that* the red LED will be dimmer if the Vcc is 2.5V or 3.3V.

The 24Cxx family uses a serial bus known as I²C. A feature of this bus structure is the ability to read or write (program) different devices on the bus. Each device has a unique address which is determined by the connection of up to three address lines A0, A1 and A2. Normally, these lines are connected to GND giving an address of 0. However, with three address lines, there is a maximum of eight possible addresses as follows:-

ADDRESS LINE			ADDRESS
A2	A1	A0	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

When the programmer writes or reads data, it requests data from the appropriate device. Therefore, it is possible to fit a clip over an accessible part and then program another part.

Note that, in some configurations, it is not possible to use all eight addresses. The user must select the appropriate slave address when programming a remote device. When programming a remote part, the address lines A0, A1 and A2 are allowed to float high but can be pulled low.

The slave address can be selected on the menu.

5. FAQS

How do I copy an eeprom?

It is only possible to copy devices of the same size. For example, one 24C04 can be copied into another 24C04. However, the devices do not need to be the same make and some devices may have a suffix after the part number such as 24C04A. The 'A' version *MAY* have a different programming algorithm. In general, the manufacturer does not matter.

The device I want to read does not appear on the menu.

Look for a device of the same size and try that setting. If possible, check the checksum otherwise there is no way of being sure you have read the correct data.

Arrange for the device to be formally added.

The device I want to program does not appear on the menu.

Arrange for the device to be formally added.

Provided that I select the correct size, does it matter if the manufacturer is not correct?

If the device programs and verifies, it probably does not matter. However, when reading devices, you may get invalid data if the manufacturer is wrong because algorithms do vary between manufacturers.

Can I use the PC2400 without a PC?

No.

Can I add a new device setting?

Yes but the manufacturer does not accept responsibility for mis-programming, etc. The programming parameters for each device are held in a text file. From time to time, this file is updated.

I have received an error message saying 'Unknown programming type'. What does this mean?

The PC2400 device parameter file contains a parameter which tells the PC2400 how to program or read a particular family of devices. As new devices are added, new methods may be required. The software for the new method is contained in the main installation program. You will require a new version of the program.

Does the PC2400 need batteries?

As power is taken from the USB bus, no batteries are required.

What does an Intel hex file look like?

The first record begins with a ':'. Usually the file has a number of records which all begin with ':'.

What does a Motorola S19 or S28 file look like?

All records start with 'S' or 's'.

What does a binary file look like?

Binary files contain printable and unprintable characters. If a binary file is opened with a program such as Notepad, odd characters and punctuation marks may appear in the file.

What does an E2P file look like?

The first few characters are E2P. After the first 152 characters, the file is binary.

I do not think the PC2400 is working. How can I test it?

After power up, the PC2400 should display '*Programmer connected*'.   If it does not, check the USB lead. If this appears to be satisfactory, try another USB device to check the USB port.

Run the self-test facility and note down any error(s) reported. If the self-test fails, the 'r*emote*' LED will turn on and stay on. Even if the self-test does fail, it may be possible to use the programmer.

My PC2400 programs components but the 'in circuit' facility does not work.

Assuming that the PC2400 passes 'self test', there is unlikely to be a fault. The usual cause of this symptom is that the target 24Cxx must be powered. When the 'in circuit' lead is connected, the '*Remote Vcc*' LED will be lit.

I have some files in another format. Is there a file conversion facility?

No. If you have access to a programmer which can load this file type, an eeprom can be programmed. The programmed part can then be read by the PC2400 and, subsequently, be saved on a PC's hard disc in Intel hex format by the PC2400.