

CodeLoader 2

Operating Instructions

Version 2.2.0

PART I: BASIC CODELOADER 2 OPERATION	3
I. Quick Overview.....	3
A. Common Questions Regarding Installation.....	3
B. Purpose of the CodeLoader 2 Program	4
C. CodeLoader 2 Components	4
D. Adding a *.ini (ex. LMX2364.ini) File to CodeLoader	5
II. Using the CodeLoader 2 Software.....	7
A. Selecting the Device Type.....	7
B. Port Setup Tab	7
C. Bits and Pins Tab	9
D. PLL Tabs	9
E. Registers Tab	11
F. Burst Mode Tab	11
G. Loading the Data to the PLL.....	12
H. Additional Bells and Whistles	13
PART II: ACTIVE X CODELOADER 2 FOR AUTOMATED TEST AND MEASUREMENT	14
A. What is Active X CodeLoader 2?	14
B. Registering and Using the Active X CodeLoader 2 Version	14
C. Using the Active X CodeLoader 2.....	14
D. Active X CodeLoader 2 Commands.....	15
E. Sample Application of the Active X CodeLoader 2	16
PART III TROUBLESHOOTING CODELOADER 2 PROBLEMS	17
A. Investigate Potential CodeLoader 2 Issues	17
B. Investigate Potential Windows Issues.....	17
C. Investigate Potential Hardware Issues.....	18

PART I: BASIC CODELOADER 2 OPERATION

I. Quick Overview

A. Common Questions Regarding Installation

- i. **How do I Fit the Install Program on Floppy Disks?**
Indeed, the CodeLoader 2 install program is larger than 1.4 MB and will not fit on a floppy disk as is. However, WinZip, which is freeware available on the web supports a feature called spanning. In order to use this, create an archive, and when you create the *.exe file, choose spanning.
- ii. **What's all this Uninstall and Maintenance Complete Stuff?**
The install program and the uninstall program for CodeLoader 2 are the same program. If you already have CodeLoader 2 on your computer, then running the install program will uninstall the older version. Then you need to run the install program again in order to install the newer version of CodeLoader 2.
- iii. **During Uninstall, which Files are Safe to Delete?**
The codeldr.ini file and the part specific files, like LMX2330.ini, contain part specific information and should be deleted and replaced with the newer versions. However, since these files contain the last state you programmed the part to, there may be some circumstances where you don't want to replace them. The other windows files probably don't have to be replaced, but it's probably good to replace them.
- iv. **The Install Program Says I Need Administrative Access. Why?**
If you have Windows NT or 2000, the computer will not install properly or write to the parallel port correctly without administrative access.
- v. **Where Can I get the CodeLoader 2 Program?**
wireless.national.com
- vi. **What's New for CodeLoader 2.2.0?**
CodeLoader 2.2.0 features include support for 22-bit fractions, long file names, an enhanced menu system, and an enhanced register map.

B. Purpose of the CodeLoader 2 Program

The CodeLoader 2 program is designed to run all of National's wireless products. It has been designed with the intent to provide the user with a user-friendly user interface and also to be flexible to accommodate National's ever expanding selection of wireless products. The CodeLoader 2 runs on Windows 95/98/NT/2000/ME/XP. These instructions apply to CodeLoader 2 version 2.2.0.

C. CodeLoader 2 Components

When CodeLoader 2 is installed, all of the most current components listed below are installed. However, it is also possible to just upgrade certain components as well.

i) Main Executable

This is the CodeLoader 2 executable file. The name is different depending on which windows platform it is running on. For Windows 95/98, the file is named "codeldr95.exe" and for NT/2000/ME/XP, it is "codeldrNT.exe". The current version of the executable file can be found by choosing Help --> About from the menu.

ii) Active X Executable

This Active X executable file can be called from other programs. It is automatically installed when the main executable file is installed, and is only necessary for automated test and measurement applications, described in Part 2 of this document. It is called "codeldr95x.exe" for windows 95/98 and "codeldrNTx.exe" for windows NT/2000/ME/XP.

iii) Menu Initialization File

This is a text file which contains the parts that CodeLoader 2 can run. It can be easily modified by the user in order to add or subtract parts. If a part is on this list, there also needs to be a part-specific initialization file as well. This file is windows independent and is called "codeldr.ini"

iv) Part-Specific Initialization Files

These files contain part specific information for each part including the counter sizes, prescalers, and pins. These files are windows independent and are of the form LMXxxxx.ini. For instance, the part-specific initialization file for the LMX2330 is "lmx2330.ini". Each part-specific initialization file has a version. This can be found by choosing Help ---> About from the menu.

D. Adding a *.ini (ex. LMX2364.ini) File to CodeLoader

Recall that the CodeLoader file has part specific initialization files. The version of CodeLoader that is on the web and widely distributed will not have parts if they are not released yet. If you are evaluating a part like this and were just given a *.ini file, this procedure shows how to add it to the menu on CodeLoader. As a matter of protocol, parts that are not fully released are included under the "ENGINEERING" section of the menu.

Step 1: Find the directory that CodeLoader 2 is in

The default directory is Program Files/National Semiconductor/CodeLoader 2

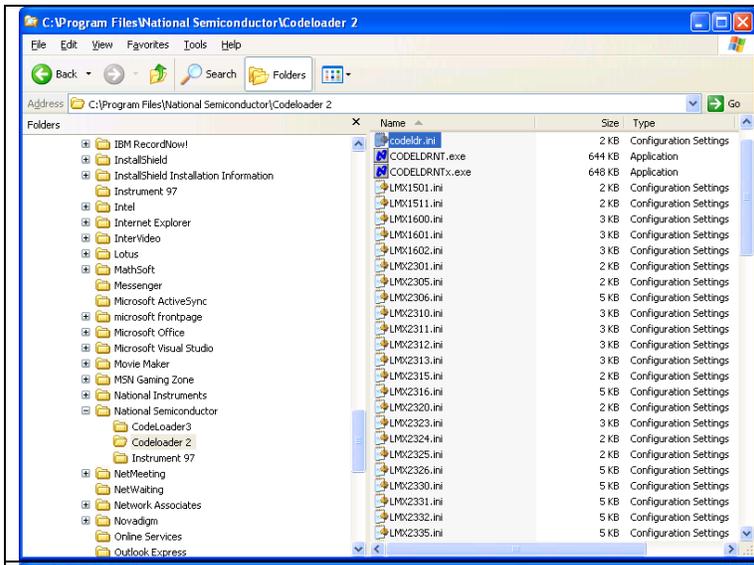
Step 2: Detach the *.ini File

Detach the *.ini file to this default directory.

Step 3: Edit the codeldr.ini file

Now open the codeldr.ini file in this default directory. It should look something like this. Note that sometimes windows can hide file extensions.

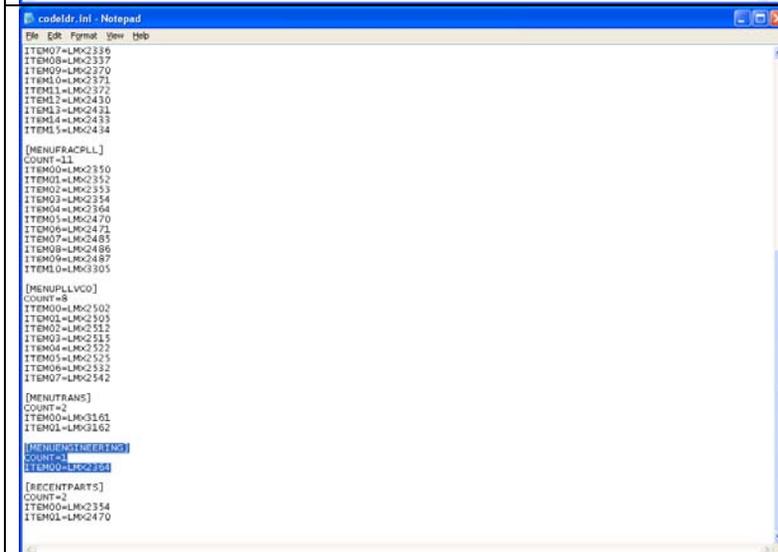
Then the next time that codeloader is run, the new part should appear on the menu and be able to run. The pictures on the next page show step 3 in better detail.



This shows the codeldr.ini file
Double click it to get to the next step.



This shows the top of the file.
Scroll down to the bottom
and it should look something
like the picture below.



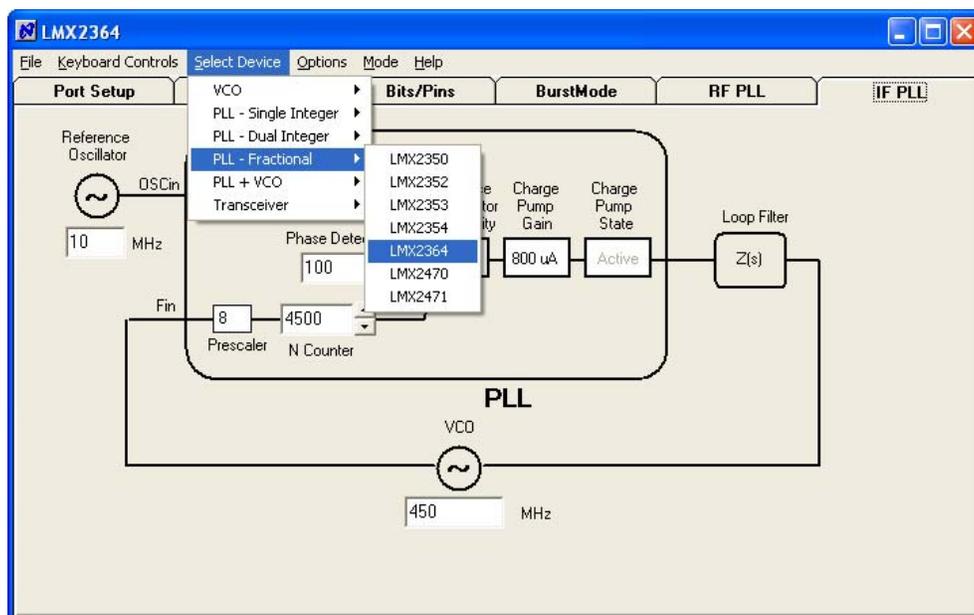
Depending on which parts
were used, this could be
different. Although it can be
added anywhere, it is general
practice to add it to the
ENGINEERING section.
Be sure to update the count
and add the part. The name
must be exactly the same as
the ini file name, minus the
"ini" part. So the
LMX2364.ini file would imply
to put "LMX2364" here.

II. Using the CodeLoader 2 Software

There are three ways to start this program. One can use the icon on the desktop, the icon on the start bar, or launch the program from the windows explorer.

A. Selecting the Device Type

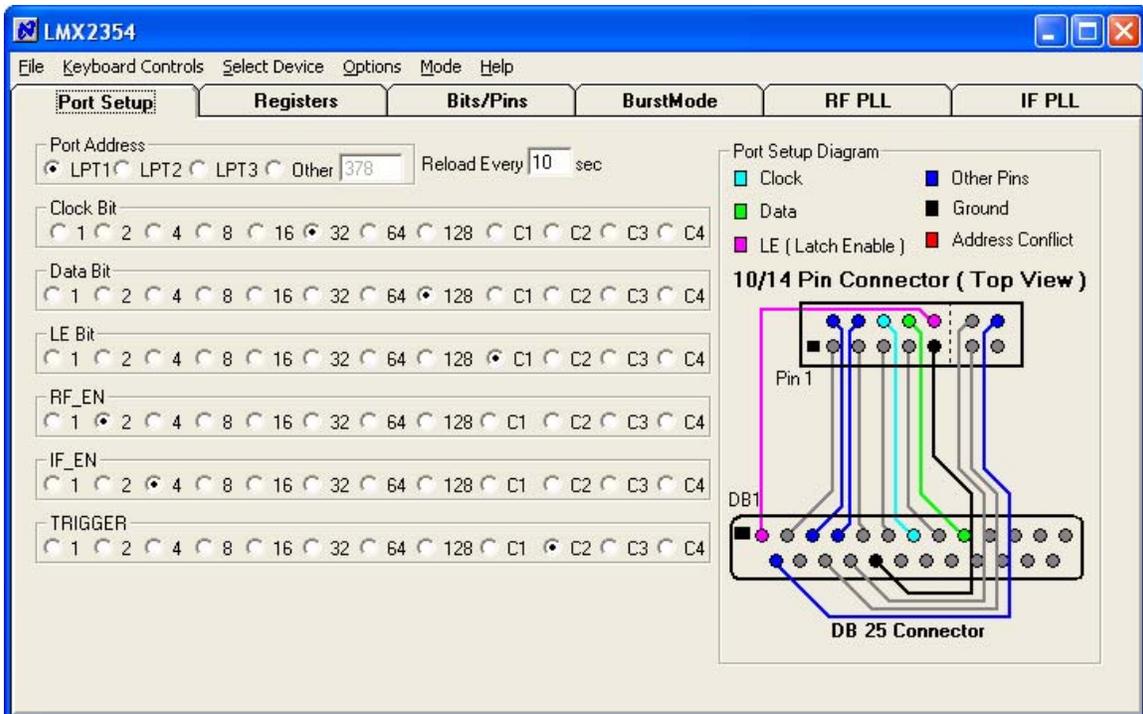
After the program is run, go to the Select Device menu and Select the desired device. After the device is selected, the CodeLoader 2 will take a few moments to configure to the specific settings for that device. Note that the specific devices that are displayed on the menu are specified in the codeldr.ini file. If the desired device is not on this list, then it is necessary to replace the codeldr.ini file with the correct version.



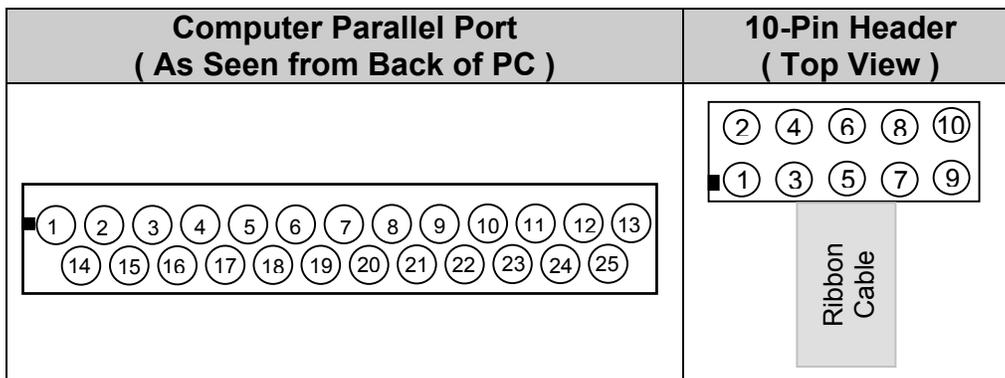
B. Port Setup Tab

The code loader provides the user with the capability of customizing the port configuration and various program signals from the parallel port of the user's computer. The port setup is board specific and specific to each device as well.

If the port setup is incorrect, then it is likely that the part will not program correctly. In addition to the clock, data, Latch Enable (LE), and Chip Enable (CE) Pins, there may be additional pins that are used to output a high or low voltage for various pins on a part. The most common use of these are for the Chip Enable pins on various wireless products. There is also the trigger pin, which can be used to trigger the oscilloscope, if one wants to observe the bits being sent to the part. The numbers for the port setup correlate to the specific pin on the DB25 connector that the signal is sent to. Attached to the computer's parallel port is a ribbon cable with a 10-pin header. The interactive port diagram gives the relationship between the port setup, cable, and 10 pin header.

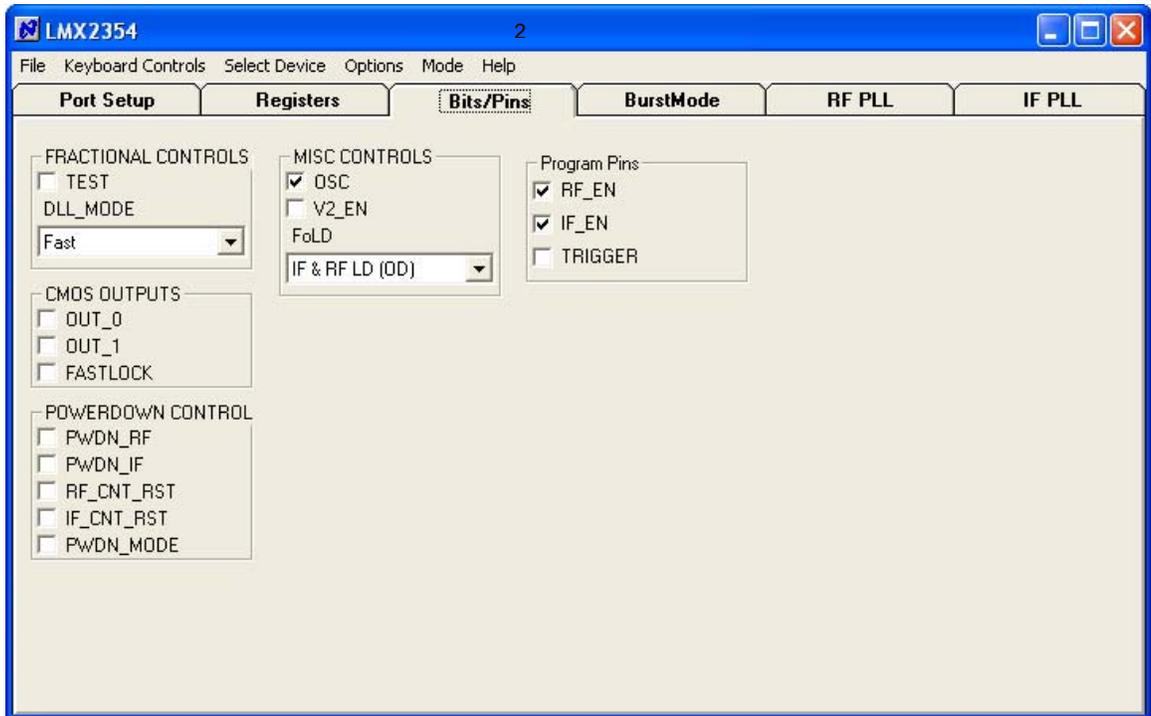


Port Setup Name	Port Setup Column	DB25 Connector Pin	10-Pin Header Pin
1	1	DB2	1
2	2	DB3	2
4	3	DB4	4
8	4	DB5	3
16	5	DB6	5
32	6	DB7	6
64	7	DB8	7
128	8	DB9	8
C1	9	DB1	10
C2	10	DB25	Not Used
N/A	N/A	18 (Ground)	9



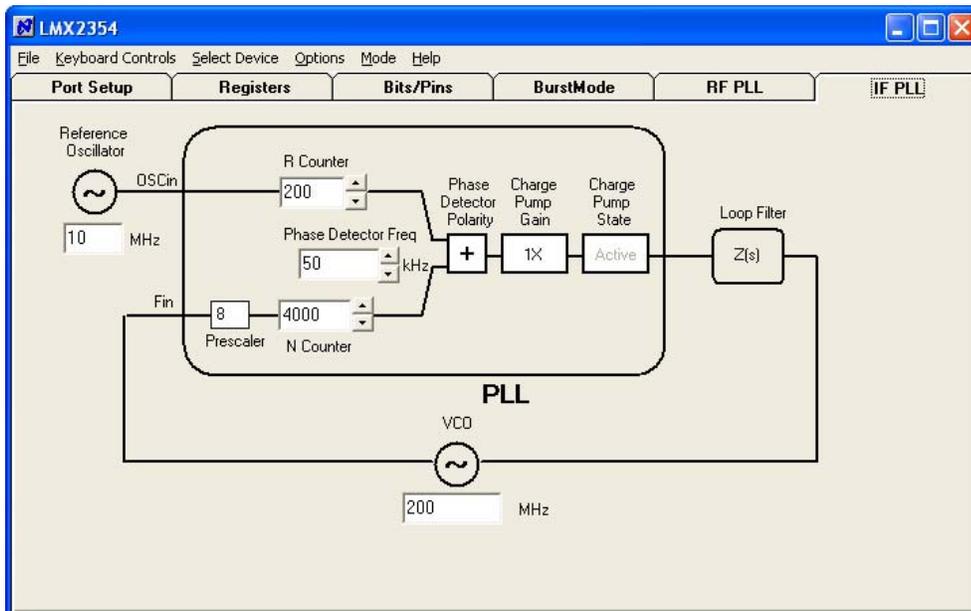
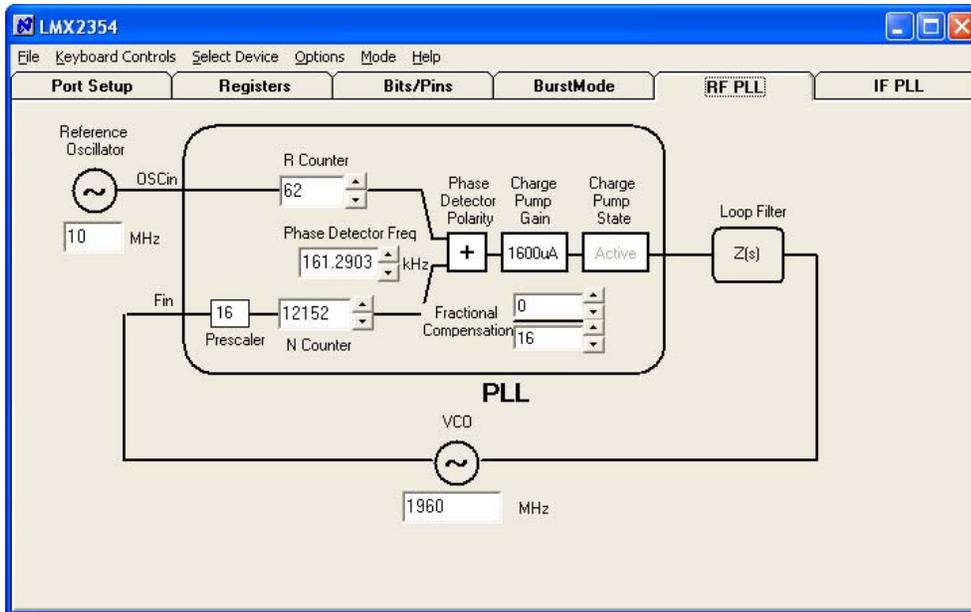
C. Bits and Pins Tab

Two example settings for the Bits/Pins are shown below. Refer to the datasheet for the functions of each bits and pins. For those functions that have a box to the left-hand side of them, an X in the box indicates a high voltage and an unchecked box indicates a low voltage. For some of the bits, a more detailed description is available by right mouse clicking on the description. The proper configuration for this board is shown below:



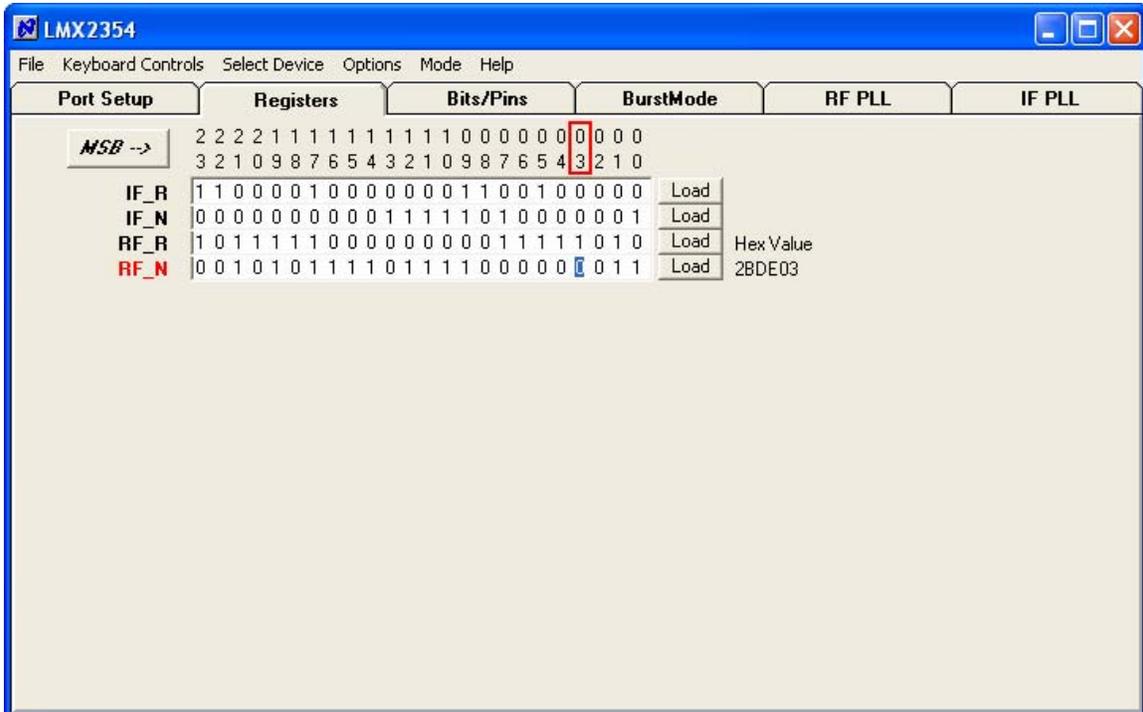
D. PLL Tabs

Depending on the number of PLLs included in this device, there will be a varying number of tabs. For instance, the LMX2330 has 2 PLL tabs. For each tab, the user needs to enter the Reference Oscillator frequency, the VCO frequency, and the Phase Detector Frequency. It is also possible to enter some counter values as well. The phase detector polarity, charge pump gain, and charge pump state can be changed by clicking on the appropriate box. If the writing in the box is grayed, then these states can not be toggled. A sample setting is shown below:



E. Registers Tab

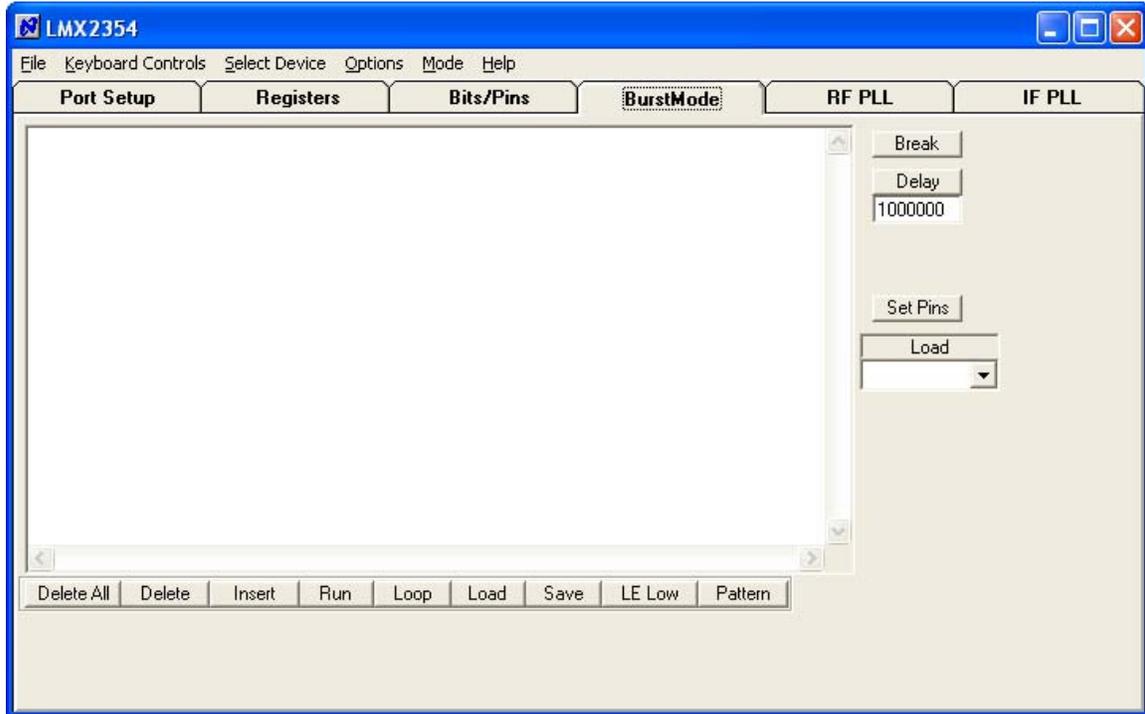
The registers tab shows the bits that the CodeLoader 2 is sending to the device. The names all of the data registers are displayed. For each register, the bit on the right hand side is the MSB and is sent first, while the bit on the left hand side is the LSB and is sent last.. When the Load PLL command is used, all data in all registers is sent. Although the bits on this screen are calculated from the Bits/Pins tab and PLL tabs, the user can modify the bits on this screen. If a bit is selected, the register, number, and hexadecimal value of the register is displayed.



F. Burst Mode Tab

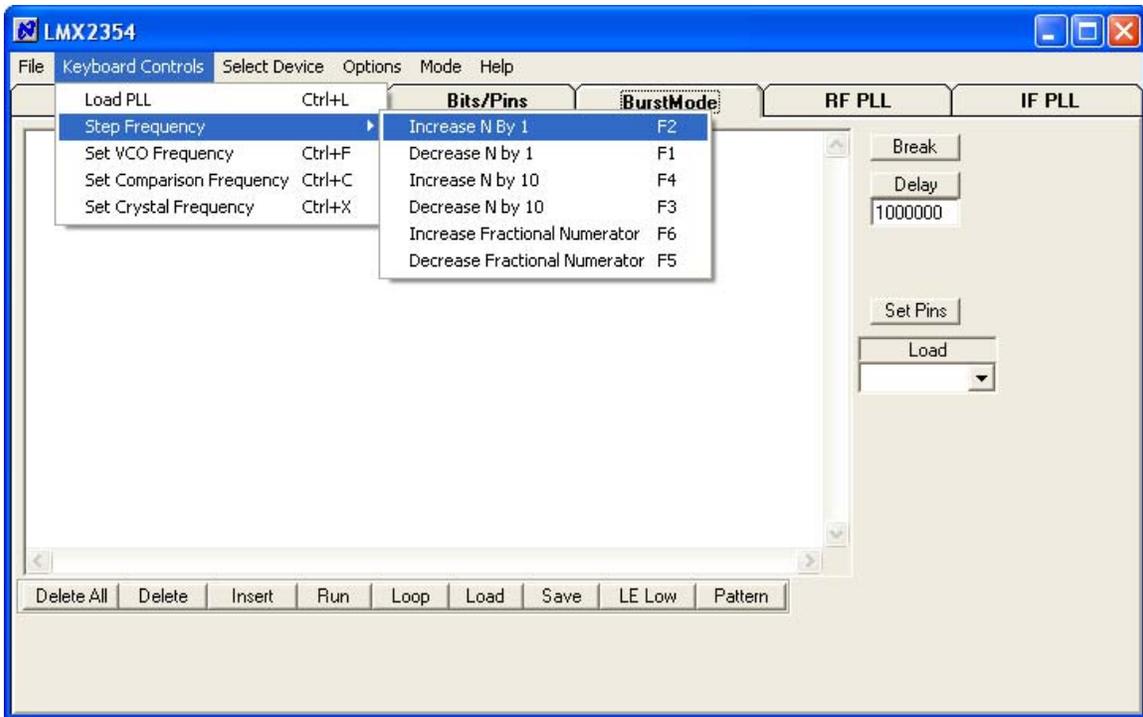
The burst mode allows the user to toggle between various values. A typical application of this is when the user wants to measure the lock time of the PLL. However, any combination of bits and pins can be toggled. The register values can not be changed on the burst mode tab, but instead are changed on the PLL or Bit/Pins Tab. To know what bits corresponding to what functions are on what registers, refer to the datasheet.

For instance, if the user wanted to cause the RF PLL to toggle between 2 different frequencies, he would need to first go to the RF PLL tab and set things to the first desired frequency. Then they would return to the Burst Mode tab and choose the register holding the RF N counter information, which would be the RF_N register. Then a delay would be inserted. The delay is just processor cycles, and is therefore processor dependent. The user then goes back to the RF PLL tab and sets the other desired frequency, returns to burst mode, and inserts the delay.



G. Loading the Data to the PLL

Although this step is pretty trivial, it is still important. To load the data to the PLL, the user needs to go to the Keyboard Controls Menu and select LoadPLL. This will load all of the data to all of the registers. After this step is done once, it is not necessary to do it again because when something is changed, CodeLoader 2 automatically sends out register corresponding to the changed function. Note that after any value or bit is changed, then that register information is automatically sent, however, it is still necessary to use the LoadPLL command one time. For instance, changing the VCO frequency programs the N counter information, but not the R counter information.



H. Additional Bells and Whistles

Since computer monitors produce considerable noise spikes, it is often necessary to turn them off. These keyboard controls allow the user to control the PLL with the computer monitor off. The F2 and F1 function keys are also useful to hand tune the PLL.

There is also a prescaler check which can perform a dual and quadruple modulus prescaler check. If the divide ratio is legal, then the numbers are in green, otherwise, they are in red. Note that CodeLoader 2 does not prevent the user from programming illegal divide ratios. Also realize that this prescaler check is not part-specific. For the majority of the PLLs, this does not matter, but there are some exceptions, which currently include all the fractional parts that require that $A + 2 \leq B$, as opposed to $A \leq B$. The prescaler check is based on the prescaler value that is displayed on the screen.

PART II: ACTIVE X CODELOADER 2 FOR AUTOMATED TEST AND MEASUREMENT

A. What is Active X CodeLoader 2?

Active X is a standard that refers to modules of code that can be accessed from other programs. This allows programs to be more modular and makes it easier to reuse code. The CodeLoader 2 95/NT/98 versions of the program allow the user to access this code so that other programs can use the CodeLoader 2. This is useful in applications where one would want to use CodeLoader 2 for automated test measurement.

B. Registering and Using the Active X CodeLoader 2 Version

Although a trivial step, it is important nevertheless. To register the program, double click on the Active X version of the program. It is only necessary to register the program once. For Windows 95/98, the filename is `Codeldr95x.exe` and for Windows NT/2000/ME/XP it is `CodeldrNT.exe`. There is no Active X version for Windows 3.1, since this platform does not support Active X. Usually, when the file is double clicked, an hourglass will appear for a fraction of a second and then go away.

C. Using the Active X CodeLoader 2

Once the program is registered, then it can be called from other programs. To do this, follow the procedure below:

i. Create an object pointer.

Example `Dim PLLObject as Object`

ii. Set the object pointer to point to the CodeLoader 2. Use the syntax shown below:

Example `Set PLLObject = CreateObject("CodeLoader2x.Application")`

iii. Control CodeLoader 2 by calling subroutines

The functions in this code are in Appendix A. The context used is "`PLLObject.function`", where *function* is one of the defined in Appendix A.

iv. Use of the SetPrgmBits and SetPrgmPins Command

These two commands are very powerful general purpose commands. The name of all of the program pins and some of the program bits are on the bits/pins page. However some bits, such as the charge pump magnitude bit usually are not on this page. It is therefore necessary to open the part initialization file to find out what the name is. The naming for most bits matches the datasheet.

D. Active X CodeLoader 2 Commands

Command	Command Description	Example
GetPrgmBits <i>BIT</i>	For the BIT specified, gets the VALUE specified. Some Bits are displayed on the Bits/Pins page, while others are not displayed, but are in the part-specific initialization file.	<i>PLLObject.SetPrgmBits("FoLD")</i>
GetPrgmBitValue <i>BIT</i>	Very similar to SetPrgmBits, but instead of setting the binary value, it gets the actual value, if different. For instance to set the prescaler to 16, you would get 16 with this command, but 0 or 1 with SetPrgmBits, depending on the PLL used.	<i>PLLObject.SetPrgmBits("RF_A")</i>
GetPrgmPins <i>PIN</i>	Gets program PIN State State . Valid states are 0 for logic low and 1 for logic high. The pin name is visible on the Bits/Pins page.	<i>PLLObject.SetPrgmPins("EN_RF")</i>
GetVCOFrequency <i>PLL</i>	For the PLL specified, gets the VCO operating FREQUENCY in MHz.	<i>PLLObject.SetVCOFrequency("RF PLL")</i>
GetOSCinFrequency <i>PLL</i>	For the PLL specified, gets the Crystal Reference operating FREQUENCY in MHz.	<i>PLLObject.SetOSCinFrequency("RF PLL")</i>
LoadPart	Loads all register values into the PLL.	<i>PLLObject.Loadpart</i>
Maximize	Maximizes CodeLoader Window	<i>PLLObject.Maximize</i>
Minimize	Minimizes CodeLoader Window	<i>PLLObject.Minimize</i>
RestoreSetup <i>SETUP</i>	Loads a Saved SETUP	<i>PLLObject.RestoreSetup "MySetup.mac"</i>
SelectPart <i>PART</i>	Selects the specified PART from the menu.	<i>PLLObject.SelectPart "LMX2330"</i>
SelectTab <i>TAB</i>	Sets the TAB specified as the active tab.	<i>PLLObject.SelectTab "RF PLL"</i>
SetPDFrequency <i>PLL</i> , <i>FREQUENCY</i>	For the PLL specified, this command sets the phase detector FREQUENCY in KHz.	<i>PLLObject.SetPDFrequency "RF PLL" , 30</i>
SetMode <i>INDEX</i>	Selects a mode that has been previously saved. The INDEX is an integer.	<i>PLLObject.SetMode 1</i>
SetPrgmBits <i>BIT</i> , <i>VALUE</i>	For the BIT specified, sets the VALUE specified. Some Bits are displayed on the Bits/Pins page, while others are not displayed, but are in the part-specific initialization file.	<i>PLLObject.SetPrgmBits "FoLD",2</i>
SetPrgmBitValue <i>BIT</i> , <i>VALUE</i>	Very similar to SetPrgmBits, but instead of setting the binary value, it sets the actual value, if different. For instance to set the prescaler to 16, you would use 16 with this command, but 0 or 1 with SetPrgmBits, depending on the PLL used.	<i>PLLObject.SetPrgmBits "FoLD",2</i>
SetPrgmPins <i>PIN</i> , <i>STATE</i>	Sets program PIN to the selected State . Valid states are 0 for logic low and 1 for logic high. The pin name is visible on the Bits/Pins page.	<i>PLLObject.SetPrgmPins "RFen", 1</i>
SetVCOFrequency <i>PLL</i> , <i>FREQUENCY</i>	For the PLL specified, sets the VCO operating FREQUENCY in MHz.	<i>PLLObject.SetVCOFrequency "RF PLL", 900</i>
SetOSCinFrequency <i>PLL</i> , <i>FREQUENCY</i>	For the PLL specified, sets the Crystal Reference operating FREQUENCY in MHz.	<i>PLLObject.SetOSCinFrequency "RF PLL", 900</i>

E. Sample Application of the Active X CodeLoader 2

This appendix does a sample application of the Active X CodeLoader 2 that uses Microsoft Excel in order to program an LMX2330 PLL from 800 to 900 MHz in 200 KHz increments. It then writes various parameters it reads onto the excel spreadsheet.

Step 1: Register the Active X CodeLoader 2

Locate codeldr95x.exe (Windows 95/98) or codeldrNTx.exe (Windows NT/2000/ME/XP) and double click on this. This should be in the same directory as the main executable. An hourglass may appear for a fraction of a second while this program gets registered.

Step 2: Launch the Microsoft Excel Visual Basic Editor

Launch Microsoft Excel, then go Tools → Macro → Visual Basic Editor
Then go Insert → Module

Step 3: Type in the following Code (or cut and paste)

```
Sub main()  
    ' This syntax below is critical  
    'The Dim and Set Keywords are very necessary  
    Dim PLLObject As Object  
    Set PLLObject = CreateObject("CodeLoader2x.Application")  
  
    ' Chooses the LMX2330 Part  
    PLLObject.SelectPart "LMX2330"  
  
    ' Programs FoLD Pin Output to RF Lock Detect  
    PLLObject.SetPrgmBits "FoLD", 4  
  
    ' Set the Xtal Frequency to 14.4 MHz  
    PLLObject.SetOSCinFrequency "RF PLL", 14.4  
  
    ' Set the Phase Detector Frequency to 30 KHz  
    PLLObject.SetPDFrequency "RF PLL", 30  
  
    ' Selects the RF PLL Tab  
    PLLObject.SelectTab "RF PLL"  
  
    'Tunes the PLL to 800 to 900 MHz  
    For i = 800 To 900  
        PLLObject.SetVCOFrequency "RF PLL", i * 1  
    Next i  
  
    Range("A1").Value = PLLObject.GetOSCinFrequency("RF PLL")  
    Range("A2").Value = PLLObject.GetPDFrequency("RF PLL")  
    Range("A3").Value = PLLObject.GetVCOFrequency("RF PLL")  
    Range("A4").Value = PLLObject.GetPrgmBits("FoLD")  
    Range("A5").Value = PLLObject.GetPrgmBitValue("RF_A")  
  
    MsgBox ("PLL Active X Exercise Finished")  
  
End Sub
```

Step 4 Run the Program

Select Run → Run Sub/User Form

PART III TROUBLESHOOTING CODELOADER 2 PROBLEMS

A. Investigate Potential CodeLoader 2 Issues

1. If the part programs to the correct frequency, then this proves that there is no programming problem. If the PLL does not lock to the proper frequency, do NOT assume that this is necessarily a CodeLoader issue, although that is a certainly a possibility.
2. If there are any chip enable pins, ensure that these are high and also that they are at a sufficient voltage to power up the part. If not, then this indicates a programming problem.
3. One of the most basic tests is the power up/power down test. Toggle a power down bit. The PLL should power up and down. You can observe this as a change in current, or observe the bias level on the fin pin change from 1.6 volts (when powered up) to 0 volts (when powered down). Note that since many parts have synchronous powerdown, ensure that there is a signal going to the OSCin and Fin pins.
4. Check the port setup page. Ensure that the proper bits are being sent to the right pins by observing the diagram on this screen. Note that if you change the port setup, your chip enable pins may be disabled, so you need to go back to the Bits/Pins tab to ensure that they are enabled. If any line is in red, this indicates that two signals are assigned to the same port, and this presents an address conflict.
5. Any pins, such and Chip Enable are easy to test. If the part has no chip enable pins, then perhaps the best pin to test this with is the Trigger. Another possibility is to observe the clock output. Note that this program does NOT continually talk to the part. Do NOT count on AutoReload either. The best way to test this is to go to the Burst Mode tab, select a register, and hit run to continuously reload it. Look for the clock, since it is easiest to see. 5 uS/division is a reasonable time setting on the oscilloscope to use.
6. Tinker with the port address. Code Loader does not automatically detect the correct parallel port. For most computers, it is LPT1, which is the default For others, particularly laptop computers, it is often LPT3. Again, be aware that changing this may disable the Chip Enable Pins, so be sure to re-enable them on the bits/pins page.

B. Investigate Potential Windows Issues

1. Windows Platforms

Tested: Windows 95, Windows 98, Windows NT, Windows 2000, Windows NT, Windows ME, Windows XP

Not Tested: Windows XP

Not Supported: Windows 3.1 (although old version does).

2. Administrative Privileges
Note that for Windows NT, Windows 2000, and Windows ME it is necessary to have administrative privileges to write to the parallel port. If you do not have these, or if the software was installed by a user without administrative privileges, then Code Loader will appear to function as normal but will not write to the parallel port.
3. System Reboot ("Bill Gates Salute")
For most installations it is not necessary to reboot the machine. However, on some machines (Particularly Windows 2000), it is necessary. On these machines, Code Loader appears to function as normal, but no bits come out. This is sometimes necessary to install some of the *.dll files that write to the parallel port.
4. Shared Printer with CodeLoader
Although not totally understood, if a printer is installed in the same port as CodeLoader is trying to use, windows may block CodeLoader from using this parallel port, but no error message is given. For instance, if both the printer and CodeLoader use LPT1, this could occur. Perhaps this is why changing the BIOS setting in Windows 2000 from "AUTO" to "ENABLED" works for some customers.

C. Investigate Potential Hardware Issues

1. Verify Proper PC Voltage Levels
Note that most PCs put out 5 V, but some put out 3. The evaluation boards have resistive dividers that divide down this voltage. Many evaluation boards divide this from 5 V to 3 V, which violates the $0.8 \cdot V_{cc}$ spec for a logic high. This gives unpredictable results. Confirm that the PC is not sending out lower voltage levels than expected. Although this is very rare, every now and then, one will get 200 mV signals out of the parallel port from Code Loader. If this is the issue, close down Code Loader and restart. If you still get the 200 mV volt signals from the parallel port, a "Bill Gates Salute" may be necessary. Again, this is very rare.
2. Verify the Parallel Port Works With Some Other Software
The DOS version of the PLL driver software is also available at wireless.national.com. Test the parallel port with this, or some other software to ensure that it indeed works at all.
3. Investigate Parallel Port Setup
The parallel port has different modes. For the parallel port, there is <enabled> and <disabled>. If the port is set to <disabled>, it will not work. Windows 2000 adds another configuration called <auto>, which does not seem to work either. These settings are accessible by pressing F2 on Windows 2000, or sometimes on the start menu for other Windows Platforms. It is not accessible through the device manager in Windows.