AN1402
Application Note

KX1400 Software Tools User's Guide

APPLICABLE DEVICES

This Application Note applies to the following Keterex devices: KX1400EW and KX1400EG.

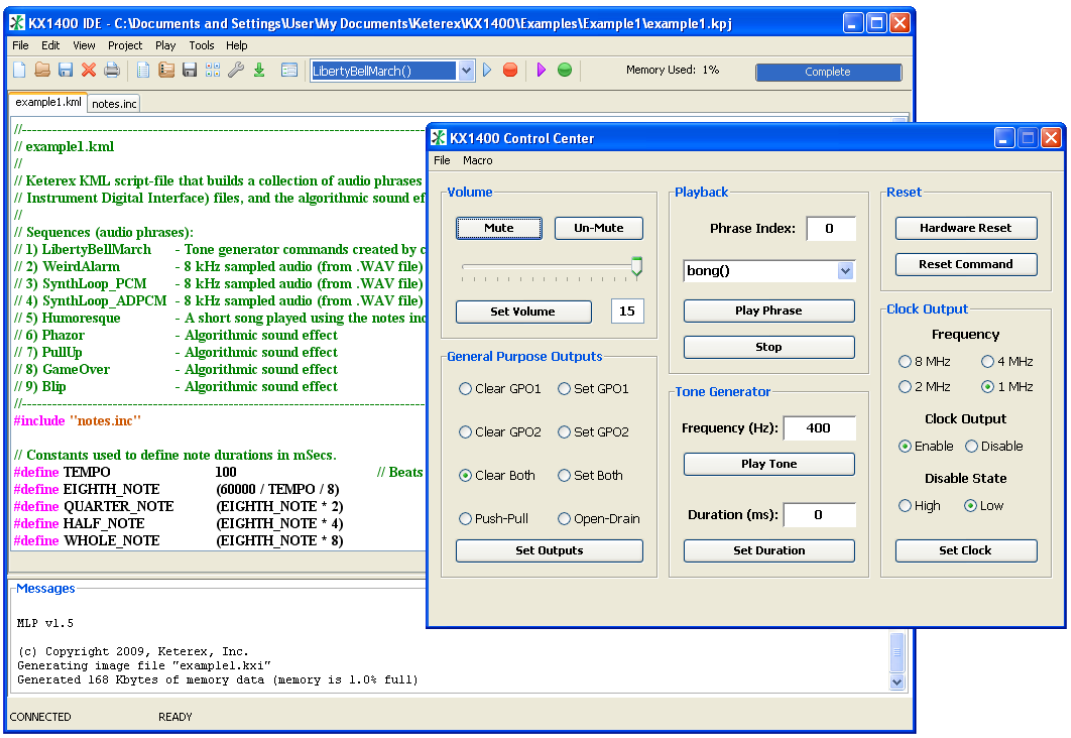
INTRODUCTION

The KX1400 Software Tools Suite provides the tools necessary to quickly and easily program and control the KX1400 Audio Playback IC. The software tools provide features for real-time execution of KX1400 commands, creation of audio data for the KX1400 IC, and converting WAV files into a C Language include file. Support for importing WAV and MIDI files, as well as generating algorithmic sound effects is included.

This guide uses two example projects installed with the Development Kit to illustrate features of the Software Suite and how to develop applications using the KX400 IC.

KEY POINTS

- The Keterex Macro Language Processor supports importing WAV and MIDI files, and generating algorithmic sound-effects.
- Utilities to assist in the creation and evaluation of audio data are included.
- The KX1400 IDE supports programming external Flash or EEPROM memory.
- The KX1400 Control Center provides access to all KX1400 serial commands.
- Macros allow issuing sequences of commands to the KX1400 IC.



1. KX1400 Software Suite

The KX1400 Software Suite include the following utilities:

- KX1400 Integrated Development Environment (IDE)
- KX1400 Macro Language Processor
- KX1400 WAV Encode Utility for Generating C Include Files
- Third-Party Audio Editing Software (Audacity).

2. The KX1400 Integrated Development Environment

The KX1400 IDE provides a framework for importing WAV and MIDI files, generating algorithmic sound effects, building a playlist of phrases, and downloading those phrases to the KX1400 external memory. The IDE maintains what is called a "project". A KX1400 project is simply a configuration stored in a file (a .kpj file). The following items are stored in a project file:

- A set of options used by the Keterex Macro Language Processor (MLP).
- A set of options used by the Keterex WAV Encoder Utility (WAV2INC).
- The type of external serial memory connected to the KX1400.
- A list of files opened when the IDE last closed.
- Various editor defaults (font, tab spacing, etc.).

The primary function of the IDE is to create, edit, compile, and download sets of phrases to the KX1400's external memory. These phrases are generated using a script language called the Keterex Macro Language (KML). KML provides access to all KX1400 serial commands plus a set of programming features which allow the user to generate audio effects, encode and download WAV and MIDI files, and organize KX1400 commands into phrases to be played later in the user's application.

The Keterex Macro Language Processor (MLP) interprets a KML file (called building) to generate an image file to be downloaded to the external serial memory (a *.kxi file). The MLP can also generate a standard Intel HEX file for programming serial memories using a third-party programmer.

The MLP builds a single KML file, though this file can include any number of other KML files. Since the MLP only builds a single main or top file, there is no need to explicitly add files to a project. The IDE simply builds the file presently open and selected in its editor. Any other KML files included by the main file need not be open.

2.1. Running the KX1400 IDE

Launch the KX1400 IDE by double-clicking on the "KX1400 IDE" icon on the Desktop (or click "All Programs->Keterex->KX1400->KX1400 IDE" in the Start menu. The IDE should start and automatically load an installed example project.

The IDE shows the connection state of the KX1400 along the bottom of the window. The following status fields may appear:

CONNECTED ..... The USB Adapter is connected.
DISCONNECTED ..... The USB Adapter is not connected. If this is incorrect, try reconnecting the USB cable to the Adapter.
READY..... The KX1400 is ready to accept the next command.
BUSY ..... The KX1400 is busy processing the previous command.

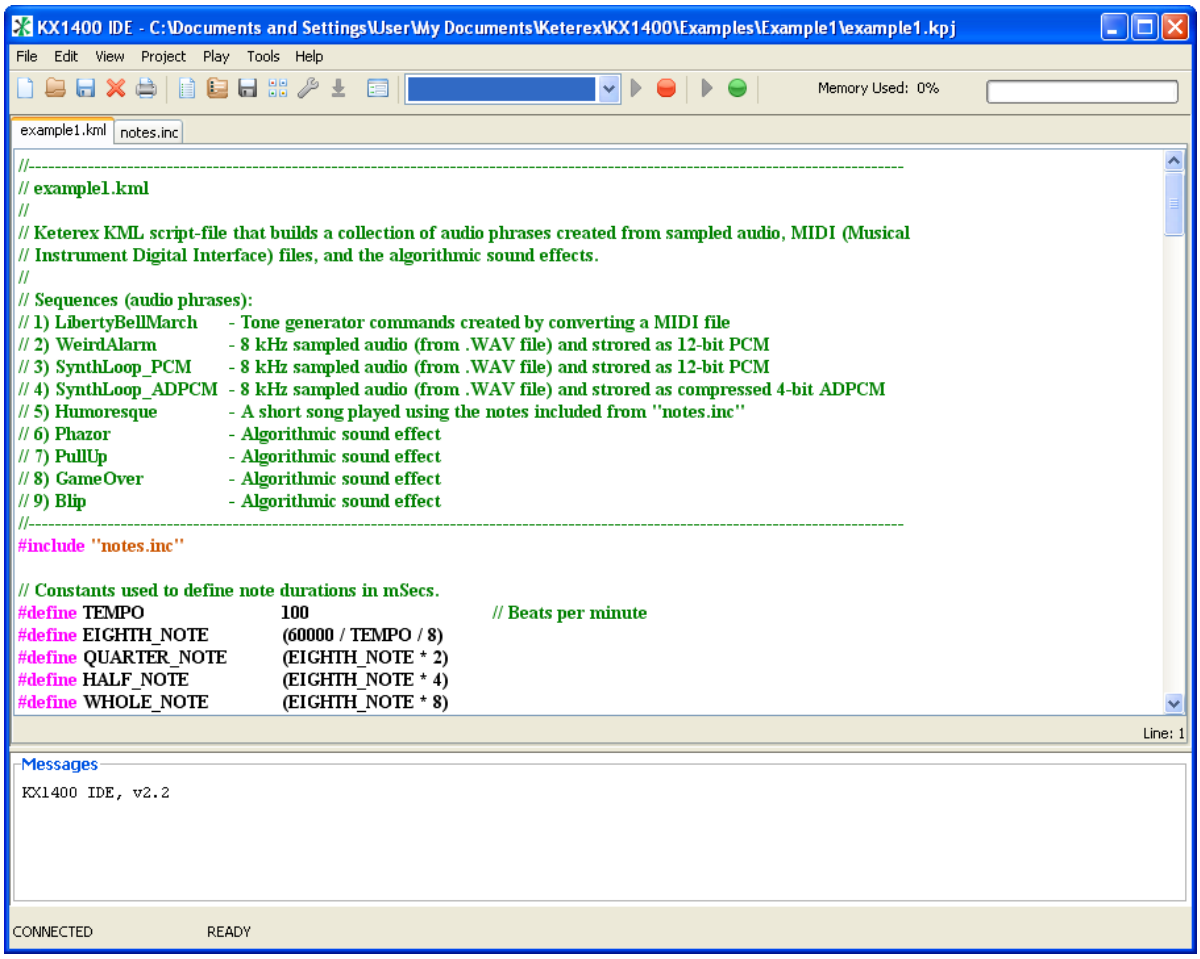
The IDE also provides information in its Messages window. For example, a one-line summary of each serial command sent to the KX1400 is shown as the command is executed.

2.2. Example 1

To open the first example project, click on "Project->Open Project" (or the toolbar icon). When the file browser pops up, select the file:

"*Examples\Example1\example1.kpj*"

You should now see the following window (if the KX1400 Tech Tips window appears, you can just close it for now):



You'll notice that two files have been opened in the KML editor. The visible file is the main KML file for Example1. It generates a set of phrases which demonstrate most of the features of the KX1400 IDE.

Now, select the example1.kml file by clicking its tab (it may already be selected), then click "Project->Build Selected File" (or the build button). The MLP will be called to build the file presently selected in the editor pane. Information about the build is shown in the Messages pane. Now click "Project->Download Build Results" (or the download button). The results of the last build will download to the KX1400 Evaluation Board. A download also automatically updates the list of phrases available to play in the selection box in the IDE toolbar.

To play a phrase, select the phrase in the selection box and click the button in the toolbar. You should hear the selected phrase play. To stop the phrase before it completes, click the toolbar button (be aware that you cannot stop ADPCM phrases, such as SynthLoop_ADPCM).

Let's look at how these phrases were generated. Scrolling through the file example1.kml, you'll see that sequences are defined with the same name as the phrases. A sequence in KML is basically a subroutine. In can contain any number of KX1400 commands and/or programming constructs, including calls to other sequences. At the end of the file is the playlist section. A playlist section in KML defines the list of sequences that will form phrases in the KX1400. The sequences called in the playlist section build, in the order they are called, the phrase list to download to the KX1400 IC. Generally, the playlist section will contain a simple list of sequence calls. However, a limited set of programming constructs are allowed, including if and while statements. For more information, see the "*MLP/KML Reference Guide*" by clicking "Help->MLP/KML Reference Guide…".

To illustrate another feature of the KX1400 IC, select and play the phrase "Humoresque(15)" while watching the LEDs on the Evaluation Board (make sure J3 and J4 are populated with jumpers). You should see that the LEDs blink in rhythm with the tune. The phrase combines **Set Output** and audio commands to synchronize the LEDs with the generated audio. In fact, any KX1400 command can be used in a phrase, allowing combinations of audio events, output events, and changes in the general-purpose clock output pin on the KX1400 IC.

2.3. Example 2

Example 2 installed with the KX1400 Development Kit illustrates one of the more power features of the KX1400 IC, specifically the ability to store a dictionary of individual audio clips and concatenate them into sentences using **Play Phrase** commands from a microcontroller. Example 2 generates a playlist of words and sounds used to produce GPS voice directions.

To open Example 2, click on the "Open Project" button and browse to and select:

"*Examples\Example2\example2.kpj*"

If a dialog opens asking to save the previous project (indicating you changed something), just click "No".

Example 2 imports a sequence of WAV files and builds a playlist of individual words and alert sounds. Build the project by clicking the button and when complete click the button to download.

When the download is complete, try playing a few of the individual phrases. Each phrase un-mutes the speaker, plays an individual sound or word, and calls the stop(1) KX1400 command. This command automatically mutes the speaker output and places the KX1400 into its low-power standby mode.

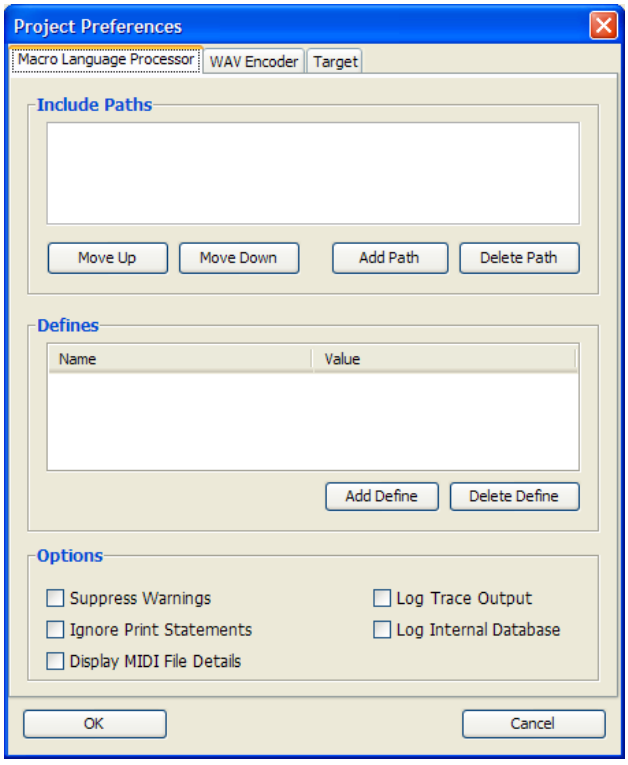
The KX1400 IDE allows the user to string together individual commands into a macro, then play that macro to the KX1400. This simulates how a microcontroller might send commands to the KX1400 in a real application. To use the macro capability to build up a full sentence, perform the following steps:

1. Click the Record Macro button .
2. Select phrase "bong()" and click .
3. Select phrase "turn()" and click .
4. Select phrase "right()" and click .
5. Select phrase "in()" and click .
6. Select phrase "n100()" and click .
7. Select phrase "miles()" and click .
8. Click the Record Macro button again to stop the recording.

Now, to play the macro click the Play Macro button . You should hear a full sentence. In a real GPS application, a richer set of numbers and words would be generated, allowing a wide range of sentences to be produced without requiring memory to store every possible combination of words. Instead, a microcontroller can instruct the KX1400 IC to assemble sentences from a dictionary of words and sounds. You can abort the macro by clicking the a second time.

3. Configuring the MLP

Each project includes a configuration used when calling the Keterex Macro Language Processor. To edit the configuration, click "Project->Preferences" or the toolbar button. The following window should appear:



The "Macro Language Processor" tab has three main sections. The "Include Paths" section allows the user to set a list of directories which are checked by the MLP, in order, when searching for an include file (the directory containing the KML input file is always checked first). Clicking "Add Path" opens a browser allowing a directory to be selected. If more than one path is added, the "Move Up" and "Move Down" buttons can be used to control the order in which paths are searched (beginning with the top path).

The "Defines" section allows the user to set one or more identifiers to specified values (similar to defines in the C Language). An identifier assigned in this section will override any value assigned in the KML file itself. Click "Add Define" to create a new define assignment. Once created, click on the "Name" and "Value" fields to type in the desired identifier name and value. Be sure to hit "Enter", "Tab", or click outside of the field to actually enter the name or value. To delete a defined value, click on the line to delete and click the "Delete Define" button.

The “Options” section allows the user to manage how the KML file is processed, as described below:

Suppress Warnings ..... Selecting this option causes the MLP to suppress all warning messages.

Ignore Print Statements..... The main purpose of print statements in KML is for showing debug information. Selecting this option causes the MLP to ignore, i.e. not execute, all print statements.

Display MIDI File Details..... Selecting this option causes the MLP to display a brief description of each MIDI track encountered when processing a playMIDI instruction.

Log Trace Output ..... Selecting this option causes the MLP to log a one-line description of each line executed while processing a KML file. The log file is the same name as the KML input file with an extension of *“.log”*. This trace information can be helpful when debugging a KML script problem.

Log Internal Database ..... Selecting this option causes the MLP to generate a representation of its internal translation of the KML input file. The resulting information is stored in the same log file used by the “Log Trace Output” option. This log file may be requested by Keterex when helping to solve a problem experienced by the user.

4. Encoding WAV Files

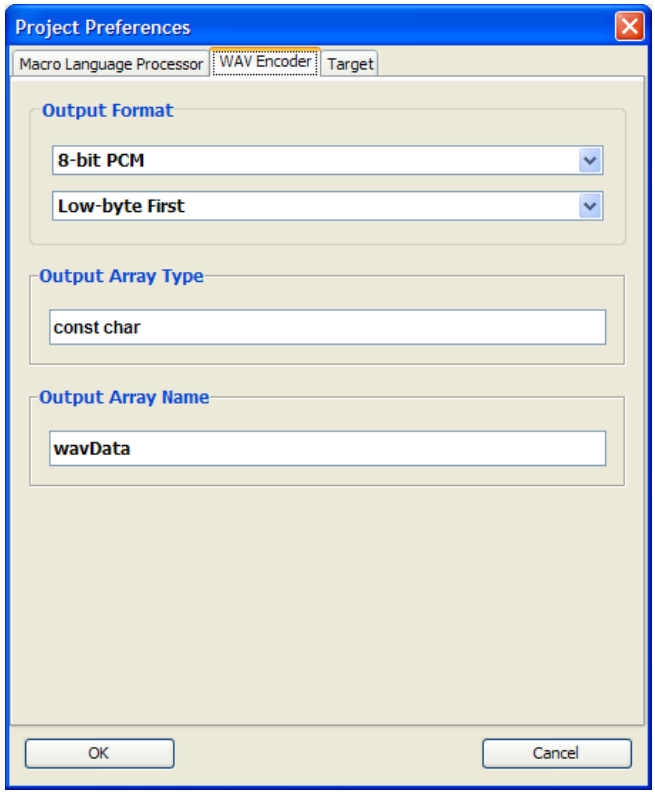
In many applications, audio commands will be sent to the KX1400 IC directly from a microcontroller (as opposed to being stored in a serial memory device connected to the KX1400). To support such applications, the IDE provides access to the Keterex WAV2INC utility. This program converts a WAV file into data formatted as a C Language include file. This include file can be used in a microcontroller program to allow the controller to play the WAV file by sending data directly to the KX1400 IC.

To convert a WAV file to a C include file, perform the following steps:

- In the IDE, click “Tools->WAV Encoder…”.
- Select the WAV file to encode.

The IDE will execute the WAV2INC utility, resulting in an include file with the same name as the WAV file (but with a “.h” extension) and stored in the same directory as the WAV file. As with the playWAV KML instruction, the WAV file must be sampled at 8kHz.

The results of the WAV2INC utility can take several forms, depending on the project settings. For example, open the project preferences by clicking “Project->Preferences…” or the  toolbar button. Select the “WAV Encoder” tab in the resulting window to see the following:



Each of the fields in this window control an aspect of the WAV2INC output format. The "Output Format" drop-down list allows the following selections:

8-bit PCM..... The include file will contain an array of 8-bit PCM samples. This format reduces the amount of microcontroller memory required to store the array, but with a loss of audio fidelity. Since the KX1400 expects 12-bit PCM samples, the microcontroller should pad zeros in the lower 4 bits of each 8-bit sample to form 12 bits and add the **Play PCM** command code before sending to the KX1400.

12-bit Packed PCM..... The include file will contain an array of 12-bit PCM samples packed into 16-bit values such that no memory is wasted. The microcontroller must unpack the values and add the **Play PCM** command code before sending to the KX1400.

12-bit PCM with Command Embedded..... The include file will contain an array of 16-bit values (stored as individual high and low bytes). Each value is a 12-bit PCM sample combined with the required KX1400 4-bit **Play PCM** command. The MCU can send the resulting data directly to the KX1400 without any additional work.

16-bit PCM..... The include file will contain an array of 16-bit PCM samples (again, stored as individual high and low bytes). This format is generally used when the microcontroller will be processing the WAV data prior to sending to the KX1400. For example, the microcontroller might provide additional low-pass filtering or equalization of the data.

4-bit IMA ADPCM..... The include file will contain 4-bit IMA ADPCM samples.

The reader should refer to the "*WAV Encoder Reference Guide*" for more information concerning these formats. To read the guide, click "Help->WAV Encoder Reference Guide…" in the IDE. When using a format which produces a multi-byte value, the second selection box in the "Output Format" section can be used to control whether the low-byte or high-byte is stored first (i.e. at the low address).

The "Output Array Type" and "Output Array Name" fields control how the generated array is declared. The default is a constant character array named "wavData". For example, the default declaration will have the form:

```
const char wavData[] = { … };
```

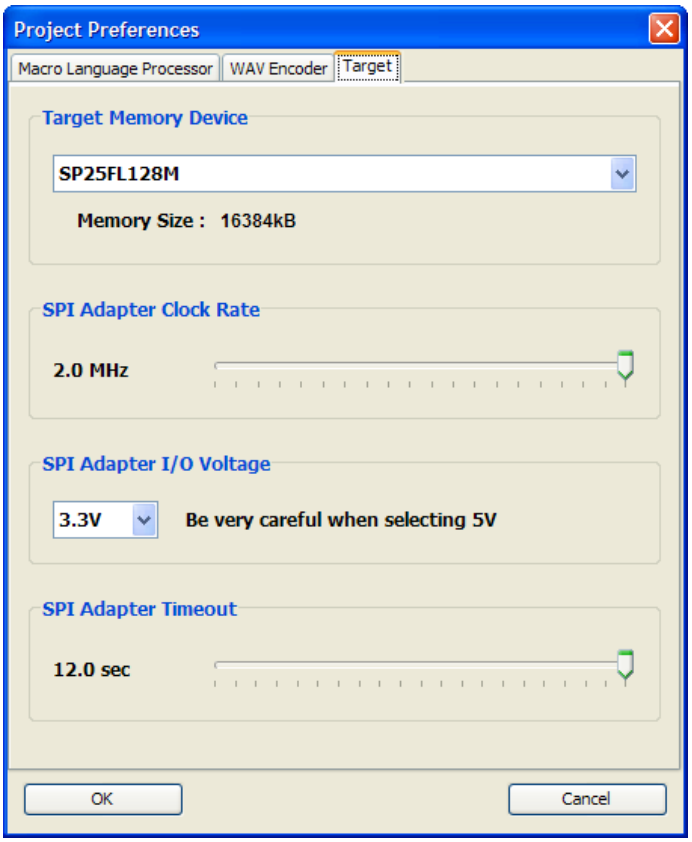
However, it should be noted that the WAV2INC utility always stores the data as a list of 8-bit values. If the output format is set to 16-bit PCM, the microcontroller firmware can access the generated array as 16-bit values using the following code (assuming a short is 16-bits):

```
#include "wavData.h"
short *shortData = (short *)wavData;
```

Be sure to have the byte order set correctly for the C compiler used.

5. Selecting the Target Memory

The IDE supports programming a number of supported serial memory devices. The default is the memory device included on the KX1400 Evaluation Board. To change the target memory, click on the toolbar button and select the "Target" tab. You should see the following:



The target memory is selected from the “Target Memory Device” drop-down list. The size in bytes of the selected memory is shown below the drop-down box. Please contact Keterex if support for a memory not listed is required for an application.

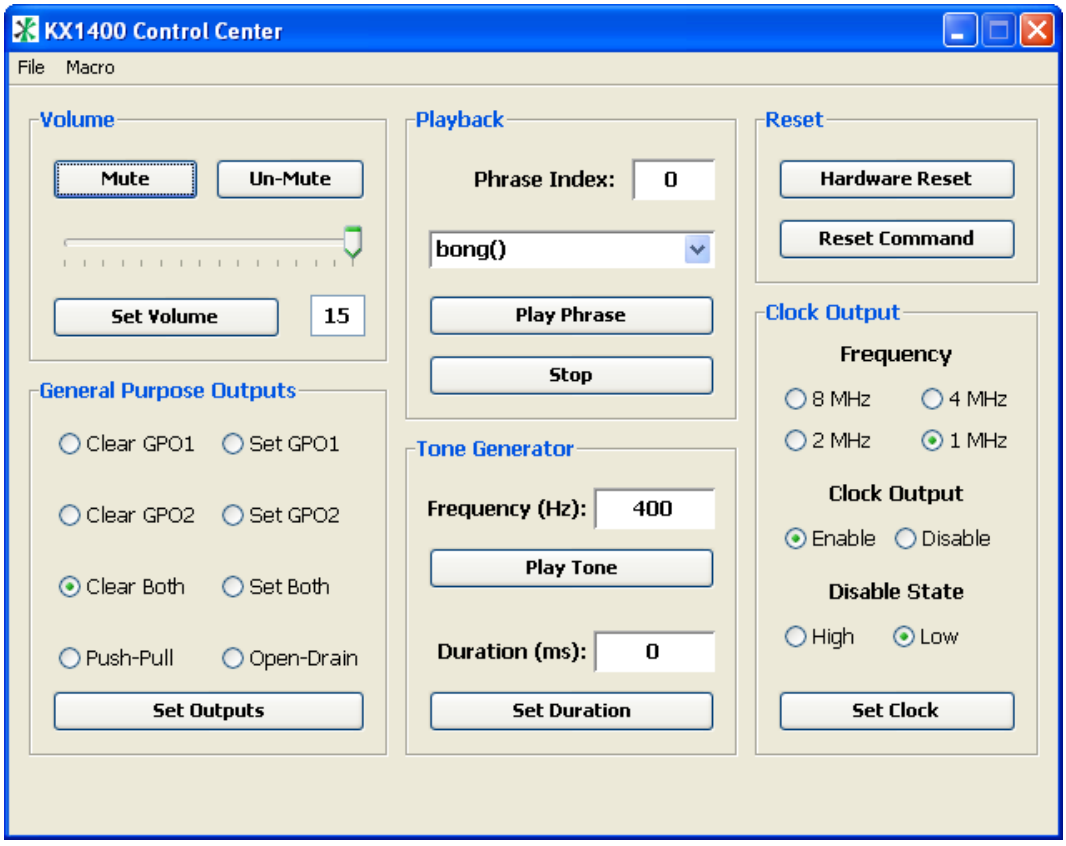
Programming of the serial memory device occurs through the KX1400 IC (which is automatically placed in pass-thru mode) using the Keterex USB Adapter. The SPI interface to the KX1400 can be configured using the “SPI Adapter Clock Rate” slider and the “SPI Adapter I/O Voltage” box. The frequency of the SPI clock is controlled by the slider. However, the default of 2MHz is generally acceptable to most serial memory devices.

The USB Adapter can be configured to communicate with a KX1400 IC operating at either 3.3V or 5V. ***Be very careful with changing this selection to 5V. This setting could damage a KX1400 IC operating at 3.3V.*** The KX1400 IC on the Evaluation Board operates at 3.3V (powered from the USB Adapter).

The USB Adapter can also be configured to timeout and return an error if the serial memory device does not complete a write or erase operation within a given amount of time. This timeout period is configured using the “SPI Adapter Timeout” slider. Moving the slider to the far left disables the timeout feature.

6. KX1400 Control Center

The KX1400 Control Center provides low-level access to all KX1400 serial commands. The Control Center is run by clicking “Tools->KX1400 Control Center…” in the IDE window or by clicking the button. The following screen should appear:



In general, each of the buttons available in the Control Center corresponds to a single serial command for the KX1400 Audio Playback IC. For example, the "Mute" button sends the KX1400 Mute command. Below is a brief description of each button in the Control Center:

Mute.....Sends the **Mute** command to the KX1400.
Un-Mute.....Sends the **Un-Mute** command to the KX1400.
Set Volume.....Sends the **Set Volume** command to set the volume indicated in the text box to the right of the "Set Volume**"** button. The user can either type a volume number into this box (0 through 15), or use the slider above the button to set the volume. Regardless, the "Set Volume" button must be clicked to actually send the KX1400 command.
Set Outputs.....Changes the KX1400 GPIO outputs according to the selection above the button.
Play Phrase.....Plays the phrase programmed into the KX1400 Evaluation Board's memory indicated by the "Phrase Index" box. The user can either enter an index into the box, or select a named phrase from the pull-down list above the "Play Phrase**"** button. Again, the "Play Phrase" button must be clicked to send the actual **Play Phrase** command.
Stop.....Sends the KX1400 **Stop** command to stop either an active phrase or tone.
Play Tone.....Plays the tone indicated by the "Frequency**"** box.
Set Duration.....Sets the duration of future **Play Tone** commands to the number of milliseconds in the "Duration**"** box (0 means play until stopped or replaced by another audio command).
Hardware Reset.....This button does not issue a KX1400 command, but rather instructs the USB Adapter to perform a hardware reset of the KX1400.
Reset Command.....Sends the KX1400 **Reset** command.
Set Clock.....Sets the **Clock Output** state of the KX1400 according to the selection above the "Set Clock" button.

It is important to understand that the various controls and buttons in the Control Center do not necessarily represent the present state of the KX400 IC. For example, although the "Clock Output**"** may have been set to 1 MHz and the command sent, a phrase played by the KX1400 (or a reset) may have changed the "Clock Output**"** state since the command was sent.

6.1. Playing Tones

The KX1400 can play a single tone ranging from 0Hz to nearly 4000 kHz in approximately 0.25 Hz steps. To play a tone, perform the following steps:

- Click "Stop" in case a tone is playing (but muted). You should see a report in the Messages window in the IDE indicating what action is taken, followed by the actual hexadecimal command sent to the KX1400.
- Click the "Un-Mute**"** button to un-mute the KX1400 output.
- Enter 0 into the "Duration**"** box and Click "Set Duration**"**.
- Enter 300 into the "Frequency**"** box and click "Play Tone**"**. You may see that the frequency value changes slightly. The Control Center replaces the entered value with the closest frequency producible by the KX1400.

You should hear the requested tone (assuming a speaker is connected). If not, try increasing the volume by sliding the volume to the far right and clicking the "Set Volume**"** button. Try different volume settings by moving the slider around and clicking the "Set Volume**"** button at each desired setting. To stop the tone, click the "Stop**"** button.

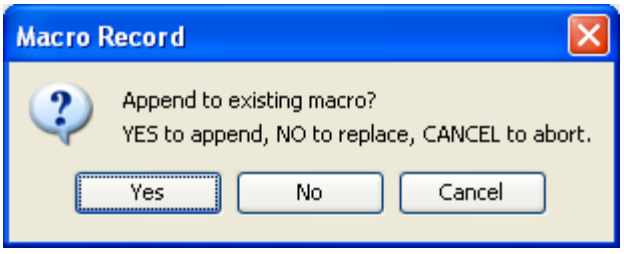
Now, set the "Duration**"** box to 4000, click "Set Duration**"**, and then click "Play Tone**"**. This should play the tone for 4 seconds (i.e. 4000 msec) then stop automatically. You can also stop the tone early by clicking the "Stop**"** button.

Once the tone is finished, try clicking "Play Tone" twice quickly. Since the KX1400's internal command buffer can only queue up two commands at a time, after the second click the Control Center will disable all command buttons until the KX1400 is finished with the first tone command and is ready to accept another command into its command queue.

The "Frequency" box also accepts named piano-style notes. For example, enter a "C" into the "Frequency" box and hit "Enter" (the case of the letter doesn't matter). The Control Center will change the "C" into the frequency for a middle-C on a piano. You can follow the note with an octave number (0 through 7). For example, "C5" will generate the frequency for a C-note in the fifth octave, i.e. one octave above middle-C. You can also enter flat and sharp notes by following a note (and optional octave number) with a "b" or "#", for example "C#" or "A3b". Also, a "Frequency" of 0 can be used to play a period of silence.

6.2. Playing a Macro

The macro feature available in the IDE is also available in the KX1400 Control Center. To play a short tune, click "Macro->Record Macro…" from the Menu bar across the top of the Control Center. The following Control Center dialog may open:



This indicates that a macro recording already exists. For now, just click "No" to replace the existing macro (clicking "Yes" appends to the existing macro).

Now, click "Un-Mute" and then enter a series of frequency values followed by clicking the "Play Tone" button after each entry. For example, to play an eight note scale enter the frequencies: C,D,E,F,G,A,B,C5 (remember to click "Play Tone" after each entry). When finished, click "Macro->Wait for not BUSY (this is explained below), click "Mute" and finally click "Macro->Stop Recording".

To play the macro:

- Enter a duration setting of 100 and click "Set Duration".
- Click "Macro->Play Macro…"
- You should hear your macro play from the KX1400.
- Click "Macro->Repeat Macro…" to play the same macro in an infinite loop. Click "Macro->Stop Playing Macro" to stop the macro playback.

Now, about that "Wait for not Busy" instruction – the KX1400 supports two basic types of commands, audio and non-audio. Commands which generate audio data (**Play Tone**, **Play Phrase**, etc.) will not execute until the previous audio command completes. This allows sounds to queue up and play in order as expected. The KX1400's **BUSY** pin indicates when the KX1400 is executing an audio command. However, the KX1400 will execute a non-audio command while still executing an audio command. For example, the **Mute** command will execute while a **Play Tone** command is active. The **Play Tone** command is not cancelled; you simply cannot hear it since the output is muted. To ensure we hear the last tone played in the macro above, we instruct the Control Center to wait for BUSY to de-assert before sending the "Mute" command. Note that this instruction is not a KX1400 command – it simply tells the Control Center to wait until not **BUSY** before continuing, much like a microcontroller might poll **BUSY** in a real application.

A macro can contain any KX1400 serial command. For example, the "Set Outputs" button can be used to embed changes in the general-purpose outputs into a sequence of audio commands, as illustrated in Example 2 above. Try the following:

- Click "Macro->Record Macro…".
- Click "No" if the append dialog windows pops up.
- Click "Un-Mute".
- Enter a frequency and click "Play Tone".
- Click "Macro->Wait for not Busy".
- Select "Set Both" under "General Purpose Outputs" and click "Set Outputs".
- Enter another frequency and click "Play Tone".
- Click "Macro->Wait for not Busy".
- Select "Clear Both" under General Purpose Outputs and click "Set Outputs".
- Click "Mute"
- Click "Macro->Stop Playing Macro".
- Enter 1000 as the duration and click "Set Duration".
- Click "Macro->Repeat Macro…".

You should hear your two tones repeating as the LEDs on the Evaluation Board blink between each note.

In the above example, we wanted the LEDs to change state after each note is finished playing. To ensure the **Set Outputs** command waited until the tone was complete, the "Wait for not Busy" instruction was embedded in the macro. If we had wanted the outputs to change immediately, this instruction would be left out.

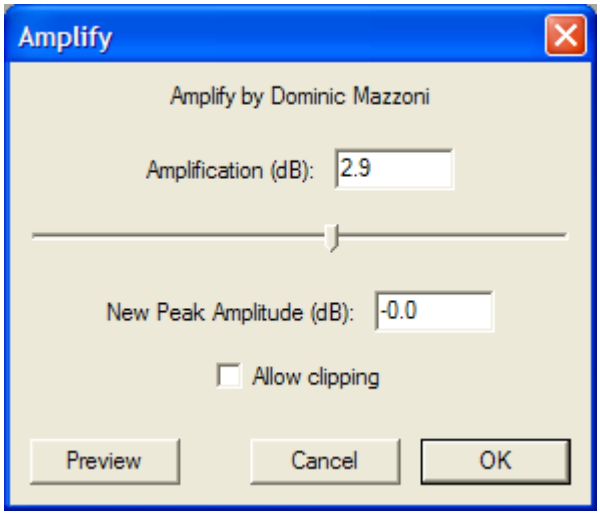
7. Using Audacity

Audacity is an audio editing utility. Among its many capabilities, it provides a means of editing and converting WAV files to a format accepted by MLP. For example, to convert a WAV file sampled at another frequency to 8kHz sampled data, perform the following steps:

1. Run Audacity.
2. Click "File->Open" and select a WAV file to convert.
3. In the lower left corner, click on the number field beside the "Project rate:" label and select "8000Hz".
4. Click "File->Export As WAV" and select an output WAV filename.

The resulting WAV file will be sampled at 8kHz. In Audacity, a segment of a WAV file can be exported by selecting a portion of the waveform (click left in the wave field and drag to select a section). Once selected, click "File->Export Selection As WAV" to export the selected section of the opened file. You can also click the Play arrow in Audacity to hear the selection.

Audacity has many other powerful features which can be used to prepare WAV files. A particularly useful effect is the ability to adjust the maximum volume of an audio clip. To try this effect, open a WAV file, select a portion of it (click Ctrl-A to select the whole waveform) and click "Effect->Amplify…". The following dialog window should appear:



The default value in the "Amplification (dB)" field is the amplification that will maximize the volume without clipping. However, for many audio clips (particularly recorded voice), up to 6dB of clipping can be tolerated without noticeable degradation in the sound quality. This excessive gain provides more volume from the KX1400 IC. To force the WAV file to clip, click the "Allow clipping" box and set the "Amplification" to about 6dB more than the default. After clicking the "OK" button, export the resulting WAV file (remember to set the Project Rate to 8000Hz). This WAV file is now ready to load into an MLP project.

You can also adjust the volume of a WAV file using the KML playPCM or playADPCM instructions. Both commands accept an optional gain parameter in dB, allowing the WAV file volume to be adjusted as it is imported by the MLP. See the "*MLP/KML Reference Guide*" for more details.

Other useful effects in Audacity include "Fade In", "Fade Out", and "Change Pitch". Note that most effects within Audacity operate on a selected portion of the waveform. For more information about Audacity, go to "Help->Contents" within Audacity.

Contact Information

Keterex, Inc.
7320 N. MoPac Expressway
Suite 201
Austin, Texas 78731
Tel: 512-346-8800
www.keterex.com
Email: support@keterex.com