

NOTE: these instructions are written for Windows XP, some variation in the actual installation dialog may be different for Win7 and Win8.

Installing the Radio Programming Software

Drag and drop the folder found on the CD named "TM742 Programmer - Rev 1 Candidate, July 1, 2014" onto the desktop.

Open that folder from the desktop, folder and Run the **"setup"** file found in that folder. (Double click the setup file) to launch the installation.

A program named **"Kenwood TM742 Programmer"** will now be located in your **Programs** menu of Windows and/or an new icon should appear in the taskbar.

Click and press CTRL and drag to either the desktop or taskbar to create shortcuts if desired.

Once installation is complete, the radio configuration screen appears. Exit the program for now.

Installing the Driver

You may or may not need to install the driver depending on whether you have already installed the driver for another Arduino application.

Plug the USB cable into the Arduino board, and into a USB port of the PC. You should see **"New Hardware Found, communications port"**. The **"Found New Hardware Wizard"** window should appear. **(if not, the driver is likely already installed and you may skip this section)**

In the **"Found New Hardware Wizard"** Select **"No, not this time"**. then **Next**.

Click **"Install from a list of specific location (advanced)"**, then **Next**.

Click the **Search Removable media**, click **Next**

Browse to the **Drivers folder** on the installation CD, select it then click **Next**. After a few seconds, the driver should be found.

NOTE: If you are running Windows 8, be sure to find and run the updated drivers for Windows 8 in the folder named New Arduino Drivers - Win 8. This driver should also work for XP and Win7.

Click **Finish**, and the driver will be installed. After that, a new COM port should appear in the Device Manager window. Note it's port number. If it is higher than COM10 on Rev Beta 3 or earlier, or COM20 on rev Beta rev Beta 4X3 or later (including the Rev 1 candidate), it will need to be remapped (see "Remapping the Com Port" below).

UPDATE: If you have a programmer marked with CH340 (a new serial chip available on very recent Arduino boards), and you are running Windows 8 or later, the driver will be automatically installed.

Driver installation is now complete.

Running the Radio Programming Software

Initially, run the program with the **radio NOT connected** to the DTMF shield.

Start the Kenwood TM742 Programmer Software

Click the appropriate "radio buttons" in the Main Menu (radio configuration) indicating the band modules installed in the radio. The order does not matter.

Below that, find the **TSU-7** checkbox, and click it **ONLY if you have the optional CTCSS decoder** module installed in the radio. This is important because this program has to anticipate the behavior of the radio when the "TONE" button is activated, which is different between radios with that module installed and those without.

You should find the "automatic repeater offset" (ARO) box already checked. This is the normal setting for most radios. Generally this should be selected. The exception is that some radios that have been "unlocked" no longer have the automatic repeater feature, and this box would need to be un-checked for such a configuration. You can test for this feature; see the trouble shooting area below.

Now click the **"Channel Lists"** button to see the window where you build and edit channel lists.

Select the tab at the top left, for the band you want to reprogram. (ex. UT144 for 2m band) You can only program one band at a time.

Build your list using the MHz, kHz, repeater offset, tone frequency drop-downs and tone (CTCSS) options for each channel. Click the **"Set/Update channel"** button to add each new channel to the list. It is not recommended to manually enter the numeric data from the keyboard, use the pulldowns. Errors in entry can cause run-time errors during the programming process. First build a short list to test the system.

NOTE: if your bandplan channel spacing is 12.5 kHz on the 2m band (as in Europe or California) you will need to set the "step size" under the **options tab** for 12.5 kHz. You would also need to reset the "step size" in the transceiver to 12.5 kHz before beginning the programming process. (shown below)

Saving your channel list to the computer

Click **FILE** in the menu bar near the top left of the window, then click **"Save Channel File"**. Each band is saved separately, that is, each band has a different filename. A directory window appears where you can name your file.

Enter the name you wish to identify the channel list you just created by. This file contains only the channels associated with the current band selection (the selected tab at the top of the screen) I like to put the band in the filename. (2m_TM742channels.csv or 70cm_TM742channels.csv for example).

Click the **SAVE** button, and the file will be saved. (you may put the file in any folder you wish, the program will remember where it is)

Repeat these steps for other bands as needed. **Remember the filename will need to be different for each band.**

Loading a previously saved channel list

Select the band you wish to program from the tabs at the top of the screen. (UT144, or UT440 for example)

Click "**FILE**" in the menu bar near the top left of the Channel List window

Click "**Open Channel File**" then select the file you wish to load, and double click it. (you may need to Browse to locate your file if you have moved them)

You will see the channel list on the right in the Channel Lists window

Edit the list as desired, and program the radio as described below. The frequencies represent the receive frequency. The transmit frequency is controlled by the repeater offset. Odd offsets are not supported. They must be programmed manually after the program has executed the rest of the list.

Testing the Programming System for the first time

Initially, for this test, run the program with the **radio NOT connected** to the DTMF shield.

With a short channel list loaded as above, and the **radio (CAT 5 cable) NOT connected, but the Arduino board connected to the PC (USB Cable)**, click the **Program Button**, **ignore the dialog box**, and click **OK**. (to save time you may want to test with a short list).

System will search the Com Ports up to COM20 (COM10 for revision Beta 3 and earlier), and identify which one the Arduino is found on. (Note the com port number for future reference)

(If it fails to locate the com port, see Note 2 below.)

Click OK, and you should see a progress bar, a green bar moving down the frequency list, and the lights on the Arduino flashing rapidly. (to see the lights, you will need to open the enclosure) **If so, you are good to go!!**

Let the process finish, then exit the program. (On revision Beta 4X3 and later, and on version 1.0 and later, you can hit the "stop programming" button, then exit the program directly.)

Note: stopping the program using the "X" box in the upper right hand corner will likely hang the USB port up, requiring a reboot, so do not exit that way without first "stopping" the programming.

Uploading the Channel List to the radio

Start with the radio turned OFF.

Restart the Kenwood TM742 Programmer. After checking the radio configuration settings on the main menu, click the **Channel Lists** button.

Click the tab at the top of the page to **select the band** you want to program. (e.g. UT144)

Load a previously created channel list or build and save a new list as described above for the band you have selected.

Plug in the USB cable between the Arduino board and the PC then connect the CAT 5 cable between the DTMF board and the microphone jack of the radio. **Leave the radio turned off for now.**

Click the **"Program"** button. The program searches the COM ports for the Arduino connection.
(See NOTE 2 below concerning Comm Ports if the Com port is not found.)

When the COM port is found click **"OK"** and a dialog box appears with some instructions. They are more fully described below:

- On the radio, **starting with the power turned off, reset the band** you wish to reprogram by **pressing and holding the F BUTTON AND the desired BAND SELECTOR button simultaneously while turning the radio power ON**, holding both buttons until all segments of the display for that band are shown (about 3 seconds), at which time you release the buttons. The default frequency for that band will now be displayed. (e.g. 144.000 MHz)
- On the radio, be sure to **select the band you just reset** before launching the programming cycle. The green LED indicates the selected band.
- Now, press the **RC** button on the radio to place the radio into **"remote control"** mode. A small RC icon appears in the upper left hand corner of the display to indicate the RC mode. (some, but not most, radios use the F key then the RC button to get into RC mode)
- Verify that the **step size** selection in the OPTIONS menu (top left of screen) and the Step Size selection in the radio are **both** compatible with your frequency list per your local bandplan. Default for 2m in the US/Canada is 5,10,15, or 20 kHz, The 5 kHz setting in the program is compatible. If your bandplan is on 12.5 kHz spacing, as is the case in parts of California, and Europe, you must set the program step size to 12.5 kHz, and set the radio step size to 12.5kHz. (note that clearing the channel list in the first step above, also resets the radio to the default step size of 5 kHz).
- **Set the radio to VFO mode.** (press the VFO button if necessary on the front panel of the radio; the radio cannot accept frequency data into memory in the MR mode.) Version 1.0 and later automatically sets the VFO mode at the start of the programming cycle, so this step is generally redundant.

Click **OK** when ready, and the programming begins. You should see the radio programming on it's display, and hear tones on the radio indicating it is being programmed.

At completion, you may select another band, and repeat these steps if desired.

When all programming is complete, **exit the RC mode** on the transceiver (press RC) , then **turn the radio off before disconnecting cables.** (see NOTE 1 below) Do not disconnect the box from the computer or shut the computer down while the radio is turned on and connected. (this can corrupt the programming data)

NOTE 1:

After programming the radio, If the Arduino is disconnected from the computer while the radio is powered up, glitches on the control lines can corrupt programming or put the radio into undefined modes. This is of no harm to the radio, but has the potential to reprogram the PF buttons (VFO, MEM, PF, or CALL buttons on the microphone), changing their function or leaving the radio in some strange mode. For this reason, it is best to turn the radio off first, or disconnect it from the programming boards first before disconnecting from the computer or shutting the computer down.

NOTE 2:

If the COM port cannot be found, it may be necessary to install the driver for the Arduino UNO communications port, or possibly necessary to remap the port to a COM port to a number below COM10 (COM20 for Beta 4X3 version and Version 1.0 or 1.0.1. and later) . Occasionally, when re-assigning COM ports, the USB/COM driver or port gets hung and the COM port cannot be found when running the program. Rebooting the computer or moving the cable to a different USB port resolves this situation.

Determining whether the Com Driver is installed

With the Arduino **not plugged in**, (USB cable) go to **Settings - Control Panel - Hardware - Device Manager**
Double click **Ports(COM & LPT)**

- Note the presence and port numbers of existing COM ports.
- Plug the Arduino USB cable into a USB port
- After a few seconds, the Device Manager screen should update and show the new Com Port. If a yellow question mark appears on the new device, the driver will need to be re-installed.
- If it appears, the driver has been correctly installed. Note the COM Port number. If it doesn't appear in the Ports list, but does appear in "Other Devices" with a question mark, it will be necessary to install the driver. (see instructions above to install the driver)
- If the new Com Port number is higher than COM10 (COM20 for version Beta 4X3 and later or Version 1.0 or later), you will need to reassign the Com Port for the Arduino Uno.

Remapping (reassigning) the Arduino Uno Com Port.

If the driver is successfully installed, (see above) determine the Com Port number (also see above)

If the Com Port is higher than COM10 (COM20 for revision Beta 4X3 and later and Version 1.0 and later), it will need to be reassigned to a lower Com Port number as the program will not recognize it.

In the Device Manager, right click the port, and select Properties, then Port Settings, click the Advanced button, and in the pull-down you will see the list of COM Ports. You may see a few that are "in use." Select one that is not in use at or below COM10 (COM20 or below for Beta 4X3 revision and later). If all are in use, you may need to determine one that isn't active, and assign the port to that value. Click OK, then OK on the Com Port Properties window, then exit the System Properties, Device Manager, the Control Panel windows.

The driver information (.inf files) can be found in the Drivers folder on the CD if needed.

TROUBLESHOOTING HINTS

PROBLEM: When I program the radio some of the repeater offsets do not program correctly

There are three likely causes of this anomaly. They can all be resolved by either changing the "automatic repeater offset" selection in the main menu or the automatic repeater offset selection in the radio so they match.

1. Your selection with respect to "Automatic Repeater Offset" in the main menu does not agree with the selection in the radio. The default in the radio is **enabled**, but it **can be disabled** (see page 46 of the manual). Doing a master reset of the radio will also restore the default setting to **enabled**. (see page 34 of the operators manual) To resolve this, change either the Automatic Repeater Offset in the main menu or the radio so they agree. (also see tests below)
2. The radio is an E type (European bandplan) and your setting in the main menu isn't in agreement, or vice-versa. To resolve this, select or de-select "E type" as necessary in the main menu to match the radio type. (see tests below to determine Radio Type)
3. The radio has been modified to extend the TX frequency capability beyond the ham bands. There is more than one way to do this, and one of them disables the ARO (automatic repeater offset) feature. To resolve this, select or de-select "Automatic Repeater Offset" as necessary in the main menu to match the radio configuration. (see tests below to determine whether Automatic Repeater Offset is operating)
 - *You can test for Extended TX frequency modification by attempting to transmit into a dummy load outside the ham bands, 150 MHz in a US 2m band for example. A "dumb beep" indicates your radio has not been "unlocked" or "wide-banded."*
 - *You can test for the Radio Type and Automatic Repeater Offset feature in the following way. Set the radio for VFO mode then vary the frequency starting at 145.500 MHz and finishing at 146.000 MHz in the two meter band..*

A US/Canada radio will have no offset at 145.000 MHz, and changes to a **negative (-) offset at 145.100 MHz**, then back to **no offset at 145.500 MHz**, then to a **positive (+) offset at 146.000 MHz if it has Automatic Repeater Offset enabled**.

An E-type radio (European) will have no offset at 145.000 and changes to a **negative (-) offset at 145.600 MHz** and back to **no offset at 145.800 MHz if it has Automatic Repeater Offset enabled**.

E type radios cannot be tuned outside the ham bands (even for receive) at all, unless they have been modified to extend the TX frequency range.

Some, **but not all** radios that have been **modified to extend the TX frequency range** outside the ham bands will have **Automatic Repeater Offset totally disabled**. You will have to **de-select "automatic repeater offset"** in the main menu of the program for these radios.

NOTE: Check the manual on page 46, **concerning how to enable/disable ARO**. My recommendation is to enable it if possible, then select "Automatic Repeater Offset" in the program. **Whatever the case, the selection of ARO in the main menu of the program must agree with the actual radio configuration.**

PROBLEM: When I program the radio, sometimes the TONE function is enabled/disabled improperly on the radio or CTCSS is invoked improperly.

This is caused by having the TSU-7 box in the main menu checked inappropriately. If you have a TSU-7 (CTCSS module) installed, you must select TSU-7 in the main menu of the program. If not, you must de-select it. (the radio does not send feedback to the programmer, and the radio behavior is different depending on whether or not the TSU-7 is installed.)

You can test for the TSU-7 (CTCSS) module by repeatedly pressing the TONE button. If it goes from T to blank, then back to T (only 2 states) you do NOT have the CTCSS (TSU-7) installed. If you have the TSU-7 installed it will cycle from blank to T to CTCSS then back to blank (3 states)

In some cases, this can also be caused by frequency step size in the radio being inconsistent with the frequencies in the list. This is because the number of digits sent to the radio is different for the 12.5/25 kHz case than for the 5/10/15/20 kHz case and the program can get "out of sync"

PROBLEM: Some or all of my frequencies come out wrong and disagree with the channel list.

There are three likely causes of this condition.

1. The radio **step size** selection does not match the programmer selected step size in the **options** menu. **Both must be compatible** with the frequency list you build. The radio will "round" to the nearest step in this case. (see page 51 of the operators manual) **Set the stepsize** in the radio to be compatible with the frequency list (F then Rev<step>) and verify that the step size in the options menu of the program agrees.
2. This can also result from having a radio that has been modified to allow transmission outside the amateur bands. To resolve this, **select "wide band"** in the main menu. The reason is that these "wide band" modifications require an extra digit to be sent in the 1200MHz band in A and E type radios, and also in the 2m and 70cm bands in E type radios. This can cause the radio and programmer to get "out of sync".
3. The "E-type" selection in the main menu could be inconsistent with the actual radio type. (E type radios have one fewer digit sent to the frequency memories in the 2m and 70 cm bands) US/Canada radios are less restrictive as to allowed frequencies. E type radios do not need to send the Tens digit of MHz entries. To resolve this, **change the "E type" setting** so that it agrees with the radio type.

(in the 2m band, E type radios assume 14 (hundreds and tens) for the first two digits of the frequency. In the US only a 1 (hundreds) is assumed. Similarly, for the 70 cm band, E type radios assume a 43 for the first two frequency digits, the US/Canada configuration assumes only a 4)

PROBLEM: When I'm done programming, and disconnect the programmer, I see the radio go "weird" like it is in scan mode or other strange behavior.

This is caused by disconnecting the Arduino (programmer) from the computer before either turning off the radio or disconnecting the radio. Always disconnect the RADIO FIRST, or turn it off before disconnecting the programmer from the computer or turning the computer off.

Not doing so can have unpredictable results, doesn't damage the transceiver, but can reprogram things that are hard to figure out. (such as re-assign the four function buttons on the microphone, **after which the programmer may not function correctly**) A master-reset (Page 34) will always clear up the rogue programming, but doing a master reset clears all frequencies, meaning you will have to start over with your programming effort.

USE THIS SECTION ONLY IF YOU NEED TO PROGRAM THE ARDUINO UNO BOARD - IF YOU HAVE A PREVIOUSLY PROGRAMMED ARDUINO UNO BOARD, SKIP THIS SECTION

WARNING: The unit comes to you pre-programmed. Do not execute the procedures below unless it is clear the Arduino has lost its program. A more likely scenario is that the driver isn't correctly installed. Check in Control Panel > System > Device Manager under COM ports and verify that the device driver is installed (No yellow question mark) before attempting to reprogram the unit.

Installing the Arduino Environment

NOTE: If you have the Radio Programming installation CD, I have already unzipped the Arduino environment and loaded the Messenger file into the libraries. Run the Arduino setup from the CD and skip the following instructions (steps 1 through 7) and proceed directly to "You are now ready to program the Arduino Uno board."

If you do not have the TM742 programmer installation CD, do the following:

1. Download the Arduino file from www.arduino.cc for your type of computer (Windows)
2. It is a zipped file, so extract all files. This is accomplished by double clicking the file if your system has a run-time unzipping utility.
3. In the extract files, you will find a file named **arduino**; double click that icon, and it will prompt you to extract more files, click this option.

At this point, the icon for the file named arduino will change to the arduino logo.
4. Double click this icon, and the Arduino utility will be installed.
5. You will find the icon for the **arduino** utility within the arduino folder (installed at the root directory as a default, though you could have chosen to install it at another location)
6. Right click this icon, and choose "create shortcut". The shortcut will then appear on the desktop, again showing the Arduino icon.
7. Click this icon to test-run the utility. Exit the program and complete the steps below.

Programming the Arduino Uno

Before attempting to program the Arduino Uno pcb, it is necessary to copy the unzipped "**Messenger**" folder into the "**libraries**" folder, under the arduino folder you found earlier. This adds some crucial library functions needed to support the DTMF shield program compilation and execution. This file is found on the CD, and also in the programming files on the TM742 Memory Management Yahoo Groups site in the FILES area.

To do this, first use Windows Explorer to locate and open the **arduino** folder. Inside this folder you will find the "**Arduino - 1.0.1**" folder. Open that folder you will find several folders, one of which is a folder named "**libraries**" (not lib)

Find the find a zipped **Messenger** folder (see above). Drag it to the desktop, then double click it. This unzips and reveals the **Messenger** folder. Drag and drop this folder onto the "**libraries**" folder mentioned above. This places the entire folder with its contents into (under) the libraries folder.

You are now ready to program the Arduino Uno board.

Connect the Arduino Uno board to a USB port of the computer, press the reset button on the Arduino board. Then run the Arduino utility by clicking the new icon on your desktop (created in step 6 above)

Under the file tab, select the DTMF Shield file (the only one in the list at this point) and you will observe another window showing the source code for programming the Arduino Uno board. (The **DTMF Shield.pde** file is located within a folder named "**DTMF Shield**" in the "**libraries\Messenger**" folder if it doesn't appear at first. You can browse to that folder within the program if necessary.)

At this point, you should see the source code for the Arduino board in a new window. It will need to be compiled, which you do in the next two steps.

Click the left most icon button (verify) and note that in a few seconds, it shows "compilation complete" in the lower window with no errors.

Find the icon button that Verifies & Compiles the code, and click it. When complete, go to next step.

At this point, find the icon that "uploads" the file to the Arduino board. Click this to launch upload to the board.

When complete, there is a message stating so.

YOU ARE DONE!! Now proceed to install/run the **Kenwood TM742 Programmer** program on the laptop side.