

# IR Release 8.03 Notes

## New features for JP1 remotes:

The routines that handle uploading and downloading for JP1 remotes (as distinct from JP1.x remotes) have been substantially rewritten by Dave Reed (3FG). The interface leads for these remotes use a Delcom chip and need an installed driver. The Delcom drivers have proved not to be very reliable with computers that have an OHCI, as distinct from UHCI, USB host controller and they do not support 64-bit versions of Windows. The new library routines support a WinUSB driver for the Delcom chip that works with all host controllers and both 32-bit and 64-bit systems. They also continue to support the Delcom driver but users with an OHCI host controller with 32-bit Windows will get more reliable results by changing to the WinUSB driver.

## New features for Learned Signals:

Two significant new features have been added to the Learned Signals page. These require ExchangeIR.dll, a new DLL that is now included in the IR.exe package. The current version is 0.08.

One feature is the ability to import and export learned signals in Pronto format. Import is available through the Add and Edit buttons of the Learned Signals page, where there are now radio buttons to choose between UEI Learned and Pronto formats, UEI Learned being the standard learned format of UEI remotes. The import routines support all raw and pre-defined Pronto formats other than the 8000 format, i.e. Pronto signals can be imported if they begin with any of the following values: 0000, 0100, 5000, 5001, 6000, 6001, 9000, 9001, 900A, 900B, 900C, 900D, 900E.

Pronto export is available through a new Export Pronto button on the Learned Signals page. Pressing this button will append the Pronto format of each signal selected in the left-hand pane (multiple selections are allowed) to a text file called ExportedPronto.txt. This file is created if it does not exist. The directory for this file can be set by a new entry, Set Export Path, on the File/Set Directory menu.

The other feature is an Analyzer window that now appears below the Decoder window of the Learned Signals page. This shows the signal in IRP form, IRP being a notation developed by John Fine to represent the structure of IR protocols. The Analyzer does not depend on a built-in list of protocols, in the way that DecodeIR does, so it is capable of analyzing signals where the protocol is unknown. If the signal is beyond its capabilities to analyze, it displays as Undetermined. The IRP form is also shown in the Learned Signals section of the Summary listing of the File/Summary menu, and in the extract of this that is shown by the Code Summary button on the Learned Signals page. Note that this button is only present when the Force Learned Timings option is selected on the Advanced menu.

There are two new options on the Advanced menu that are related to these new features. The normal operation of the radio buttons on the Add/Edit dialog box for learned signals is, with one exception, passive. You paste, or otherwise enter, the signal data and select UEI Learned or Pronto format independently. The format choice simply tells IR.exe how to interpret the data when you press the OK button. The new option Active Pronto Switch on the Advanced menu changes the radio buttons to an active mode. In that mode you need to select the format *before* you enter or edit the signal data. Switching the radio buttons to the other format will convert the display in the data window to the new format. You can therefore convert the data between

formats without needing to accept it with the OK button. The one exception in normal operation is when you press Edit for an existing learned signal but have not yet actually edited it. The buttons are active until you actually make an edit. This enables you to switch to, and display, the Pronto format of the existing signal without needing to turn on Active Pronto Switch.

The other new option is IRP Data Format. This enables you to choose whether the data values displayed in the Analyzer window are shown in binary (base 2), quaternary (base 4) or hex (base 16). The default is hex.

### **Minor Modifications:**

Release includes new DecodeIR.dll version, 2.41, and new jp12serial.dll version, 0.16. The new jp12serial.dll version includes support for the JP1 EEPROM Programming Adapter designed by Kevin Timmerman and marketed by Tommy Tyler that enables uploads and downloads of JP1 remotes to be performed with JP1.x interfaces and software. This new version also reduces the dead time before the upload or download starts, during which the software is searching for the port to which the interface is connected.

When Options/Extended Titles is selected on the Tools menu, the version of IR.exe shown on the title bar is now the product name (IR 8.03) instead of the file version (IR 8.0.3.8).

Added number tables up to 485.

## **IR Release 8.02 Notes**

### **Minor Modifications:**

Release includes new DecodeIR.dll version, 2.39.

Changed the handling of the RamAddr RDF parameter. This parameter controls the start address of disassemblies on the Protocols tab. It is now ignored except for the S3C80 and S3F80 processor families. For the S3C80 family it defaults to \$8000 and gives an error if set to anything other than \$8000 or \$FF00. For the S3F80 family it defaults to \$FF00 and gives an error if set to any other value. For other processors, disassemblies start at \$0100 except for the 740 processor for which they start at \$0132. Corrected a bug related to this that caused the wrong start address to be used if the remote was changed to one from a different family but which had the same protocol ID in its first protocol upgrade.

Corrected a bug that caused an Access Violation error if any action was performed that caused the Toolbar to be reset after a row had been selected on the KeyMap grid of the Devices tab.

Corrected a bug that in an unusual set of circumstances caused an error “Unterminated string in RDF”, when there was no such string.

Added a new entry on the Advanced menu, “Always Show Extenders”, unchecked by default. When this is unchecked, the RDF selection offered when you open a .ir file will never contain both extended and unextended RDFs – if any matching RDFs are unextended then any extended ones will be suppressed. If there is only one unextended one, it will open automatically (even if there were matching extended ones). Check this new menu item to revert to the behaviour of earlier versions of IR.exe in which all matching RDFs are listed, regardless of whether or not

they are RDFs for extenders. This menu item has no effect on the list shown by File/New/Select. This will always show all RDFs.

Added number tables up to 475.

## IR Release 8.01 Notes

### Minor Modifications:

Release includes new DecodeIR.dll version, 2.38, and new jp12serial.dll version, 0.14.

The Help/About menu panel now displays the version numbers of both DecodeIR.dll and jp12serial.dll whenever it is accessed. Previously the jp12serial.dll version was only displayed if the About panel was accessed *after* an upload to, or download from, the remote.

There is a new entry on the Help menu, “Interpreting Decoded IR Signals”, which opens the file DecodeIR.html provided this is in the same folder as IR.exe. This file is now included in the IR distribution; previously it was only available in the separate DecodeIR distribution.

Added number tables up to 474.

There is now a Notes column for the Device Buttons panel of the General page.

Some columns have been re-named to reduce possible confusion.

When uploading and downloading, there is now a message displayed next to the round pseudo-LED at the bottom left corner to remind you this is in progress.

Clean Upper Memory, on the Advanced menu, now displays a confirmation message warning that this cleaning will destroy most extenders.

Dialog forms with a Cancel button can now also be cancelled with the ESCAPE key.

Alignment problems with the Summary display on the File menu have been resolved.

The widths of the main boxes on the Learned Signals page now follow any changes in width made to the overall IR window.

Also on the Learned Signals page, there are now entries on the right-click menu of the list box to show summary extracts of the learned code, or learned timing, data. When the Force Learned Timings option on the Advanced menu is selected, new buttons get added to this page for showing these extracts and a new edit box, “Seed”, becomes visible that acts as a rounding offset. This works in conjunction with the previously existing “RoundTo”. When “Seed” contains a nonzero value, rounding takes place to values of the form “Seed + multiple of RoundTo”.

Selected options, such as those on the Advanced menu and the new Tools/Options menu, are now preserved in a different way in the Windows Registry than they were in earlier versions. Previously there was a separate registry entry for each option. Now that there are 16 such options, this was getting to be unwieldy. There is now a single entry covering all options. When version 8.01 is first used, this new entry is created corresponding to the existing old-style entries, which are then deleted. A side effect is that if you do return to an earlier version, the options there will have been returned to their default settings but when you return to 8.01 your options set in that version will be restored.

There is a new entry on the Advanced menu, “Show RDF Edit Buttons”. When this is selected, two new buttons are shown on the Toolbar, “Open RDF” and “Refresh RDF”, with corresponding new items appearing on the File menu. “Open RDF” opens the current RDF in the default text editor. “Refresh RDF” reloads the current RDF and .ir file from the file system, first prompting you to save changes to the .ir file if any have been made since it was last saved. Changes can be made to the RDF and then put into effect simply by opening the text editor with the “Open RDF” button, making and saving the changes in that editor and then pressing the “Refresh RDF” button to put them into effect.

Interface to DecodeIR now enables it to be passed any bursts that Raw Timing Data reports as “Extra”. Previously these were omitted from the provided data. This enables further checks to be made on protocols that need such bursts, but only if the version of DecodeIR is 2.37 or later.

The package now includes an FAQ that gives more information on new features. DO PLEASE READ THIS.

### **New Features:**

Tools/Options Submenu: There is a new Options submenu under the Tools menu, giving the ability to hide the Toolbar or the “middle buttons” of the General page and to augment the information on the Title Bar to display the full version number of IR.exe and the signature, interface type and processor of the remote.

Selection of multiple rows: It is now possible to select more than one row in the grids and list boxes. Multiple selections can be made in any of the usual ways and they can be moved, edited, cloned and deleted. Moving is performed by dragging with the mouse. In the case of grids, the drag must be started in the left-most (row number) column.

Sorting of grids and list boxes: The various grids can now be sorted by clicking column headers. Most of the headers are active in this way. Sorting on multiple sort keys can be achieved by clicking headers in succession, starting with the most minor sort key. As is usual, successive presses of the same header reverse the sort order. The list boxes of the Devices, Protocols and Learned Signals pages can also be sorted, but as these do not have headers, there are instead entries on the right-click menu to sort them.

Undoing moves and sorts: There are new buttons on the toolbar, and corresponding entries on right-click menus, to undo moves and sorts and also to re-do them after they have been undone. Up to eight levels of moving or sorting can be undone and redone, independently for each grid or list box.

Highlighting: There is now a linkage to show what bytes on the Raw Data page correspond to any particular entry on the other pages. This is called Highlighting and it needs to be turned on through the Tools/Options menu before it can be used. It works with both grids and list boxes. These each gain a Color column when highlighting is turned on. A color can be selected for any item or multiple selection of items with the Set Highlight Color button on the toolbar, from the Set Color entry on the right-click menus or with the Color drop-down box on the editing form that appears when an Edit button is pressed. Double-clicking the Color column, even in list boxes, also opens the color selection form but this only works for single items. The Raw Data bytes corresponding to any item are highlighted in the chosen color on the Raw Data page.

# IR Release 8.00 Notes

## Minor Modifications:

IR should save the JP1/JP1.x interface setting on exit and restore it when IR is next opened. A bug in the restoration code prevented this from working correctly. If JP1.x was saved, IR would still open with JP1 set (unless the computer didn't have a JP1 interface). This has now been corrected.

The headings of the two columns in the Key Moves grid that are both headed "Device Button" have been changed to "From Dev Button" and "To Dev Button" to reduce the risk of confusion.

Certain buttons and labels have been widened to allow the text to be displayed in full when larger fonts are in use.

For those remotes that can support them, support has now been added for device codes in the range 2048-4095. These are the remotes that require the setting "2BytePID=Yes" in the RDF.

In earlier versions "Export to Wav" of Upgrades for certain remotes, in particular URC-7780 and URC-7781, produced .wav files that would not load into the remote. This has now been corrected.

Two bugs have been corrected that led to error messages appearing under certain circumstances when IR.exe is being opened. One caused an error "Unable to set data for "" when IR.exe was being run without administrator rights. In MS Vista this occurred if it had ever been run with administrator rights prior to being run without those rights. The other caused a message "Unable to initialize drivers for any JP1 interface." The exact circumstances causing this error are not clear. Both errors were caused by problems with the Windows Registry.

MS Vista users should note that the automatic association of ".ir" files with IR.exe is only set up if IR.exe is run as administrator, i.e. right click the file and select "Run as administrator". Being logged on as an administrator is not sufficient. Once set up, this association remains in force even when the program is run not as administrator.

Added number tables up to 447 (was 446).

Updated jp12serial.dll driver to version 0.13 (was 0.10).

## New Features:

Re-sizeable Panels: The panels on the General tab are now re-sizeable. There is a moveable vertical divider between the Device Buttons and Other Settings panels and a horizontal one just underneath these panels. The column widths in the grids have always been changeable (drag the dividers in the column headings), as has the size of the IR form itself. With this new capability, the user should now be able to set all the columns to any desired size.

ToolBar: There is a new toolbar below the menu bar, with buttons for New, Open, Save, Download from Remote, Upload to Remote, and Open RM and Open KM buttons that open RemoteMaster and Keymap Master. There is also a button that opens the new Code Selector tool (see below). This is disabled and turned grey if the Code Selector is not supported by the RDF file for the remote.

Interface Display and Selection: The current interface, JP1 or JP1.x, is shown on the new toolbar to the right of the Download and Upload buttons. If the computer supports both types of

interface then this display is in blue and is actually a toggle button. If only JP1.x is supported then the display is in grey and is inactive. As a toggle button it switches between JP1.x and the last used setting for JP1 (i.e. whether Serial, Parallel or USB and whether EEPROM size is auto selected or forced to  $\leq 2K$  or  $\geq 4K$ ). This memory of the last JP1 setting is preserved between invocations of IR, even if JP1.x was selected when IR was closed. (Note that JP1 selection by means of the row of radio buttons does *not* restore the last used JP1 setting.)

Open RM button: This button on the new toolbar opens RemoteMaster, provided that the directory containing RemoteMaster has been set by using the File/Set Directory menu item. If it has not been set, pressing this button gives a message telling you that it needs setting.

Open KM button: This button opens the most recently used version of Keymap Master. Unlike the Open RM button, it is not necessary to specify in advance the directory containing the KM file. To change the version of KM that it opens, open the required version outside of IR. This will then automatically become the version opened by the Open KM button. If KM has never been opened, this button is disabled.

Help: Added *real* help to the Help menu with three new entries. “IR Help” opens IRHelp.hlp (downloadable as irhelp.zip from the from the Help Files section of JP1 Forum File Section). “IR Basic Guide” opens the web page “JP1 – Just How Easy Is It?” in the JP1 Beginners forum. “JP1 Home Page” opens the home web page of the JP1 Forum. Note that users of MS Vista will need to install WinHlp32.exe to use the IR Help item. This is available free of charge at <http://support.microsoft.com/kb/917607>.

New File/Set Directory menu items: There are two new entries here, one for the directory containing IRHelp.hlp and the other for that containing RemoteMaster.

Support for remotes with “soft” device selection: These are remotes such as the URC-7780 and URC-7781 that have device selection through an LCD screen, instead of by dedicated “hard” buttons. Access to this support requires revised RDFs that specify the necessary parameters. For these two remotes and their extenders, suitable RDFs marked “for IR 8.00” have been posted in the Tools/JP1 Files section of the JP1 File Section.

Display of processor, interface and extender versions: The Raw Data page now shows the make and model of the remote’s processor and the version (JP1, JP1.2 etc.) of the JP1 interface it uses. When supported by the appropriate RDF entry, it also displays the extender version for remotes with extender software.

Read-only settings: A new RDF entry now permits some of the “Other Settings” on the General page to be read-only. This is intended for settings that it is useful to initialize and display but which should not normally be changed manually.

User-friendly Pause: There is a new interface for the “Pause” Special Protocol which permits the pause duration to be specified in seconds (decimal point allowed). This is supported by a new entry in the RDF that gives the multiplier to turn seconds into protocol data value. The old Pause interface is of course still available by clicking the Hex check box.

Code Selector: There is a new Code Selector tool that can be opened from the Tools menu, from a button on the toolbar or from a right-click menu within the Device Buttons panel. This enables you to view and select those device setup codes that are valid for any device type for the remote concerned. This is available only if the RDF contains a [SetupCodes] section that provides the required data.

Setup Code validation: For remotes where the RDF supports the Code Selector tool, there is also validation of manually entered device setup codes. Invalid codes are shown in red. If you try to save a setup containing any invalid codes, you are asked to confirm the action. If you try to upload such a setup to the remote, or to export it to a “.wav” file, you are either prevented from doing so or are asked to confirm the action, the choice depending on the SetupValidation entry in the [General] section of the RDF.

Greater flexibility for Special Protocols: The RDF syntax for Special Protocols has been enhanced to support (a) alternative names for Special Protocols, e.g. Long and Double instead of LKP and DKP; (b) two or more instances of the same Special Protocol, e.g. two ToadTogs to provide 16 toggles instead of 8; (c) alternative entries for the same protocol when different implementations need different parameter values but all use the same PID, e.g. to allow automatic selection by IR.exe of the data for either of two Pause protocol implementations that use different multipliers; (d) Special Protocol implementations where there is no corresponding entry in the Device Upgrades section of the remote, e.g. the DSM protocol in URC-6131 and Atlas SA\_7 extenders; (e) Special Protocol implementations internal to an extender that do not in fact use a Protocol ID at all, e.g. the second DSM protocol in the URC-7780 and URC-7781 extenders that could not be used until such support became available in IR.exe. Such internal Special Protocols are specialized macros, as distinct from ordinary Special Protocols that are specialized Key Moves. They are set up and displayed through the Special Protocols tab along with ordinary Special Protocols.

Prioritize Macros: Version 6.15 of IR.exe modified the order in which the Advanced Codes sections are written when rows are moved within any the grids. This lowered the priority given to Macros and is not suitable for everyone. There is now an option “Prioritize Macros” on the Advanced menu tab which when selected, returns the order to that of the pre-6.15 versions. As with other such Advanced options, once selected the setting is permanent until actively deselected – it is even preserved through upgrades to IR.exe.

Always Allow RDF Selection: If a file is opened for a remote that has the same signature as the last opened remote, previous versions of IR.exe would always use the RDF used for that last remote, even if there were several RDF files matching the remote in both signature and fixed data. There is now an option “Always Allow RDF Selection” on the Advanced menu tab which, when selected, ensures that IR.exe always offers the full choice of matching RDF files. As with other such Advanced options, once selected the setting is permanent until actively deselected – it is even preserved through upgrades to IR.exe.

New Macro Coding Scheme: The URC-7780 and URC-7781 have introduced a new coding for the storage of macros in their data storage area. This permits Timed Macros to be stored in the same region of the storage area as ordinary (Key) Macros. Prior to this coding, timed macros required a dedicated storage area specified by the TimedMacroAddr RDF parameter. Support has been added for this coding scheme, both for Key Macros and Timed Macros.

## **New RDF Parameters:**

To support these new features, the RDF Specification has been extended by new entries and enhanced syntax for certain existing entries. The RDF Specification version number has been changed from 3 to 4 as a result of these extensions. Means have been provided to enable an RDF to provide alternative data to applications according as an application complies with Version 3 or Version 4 of the Specification. The RDFsync entry in the [General] section should be set to 4 for RDFs that comply only with Version 4, otherwise it should be set to 3.

In outline, the following new entries have been introduced for the [General] section of an RDF:

SoftDev  
SoftHT  
MacroCodingType  
StartReadOnlySettings  
PauseParams  
RDFVersionAddr  
ExtenderVersionAddr  
SetupValidation

In addition the syntax of the Labels and TimeAddr entries has been extended, as has that of entries in the [SpecialProtocols] and [DeviceButtons] sections. There is also a new optional [SetupCodes] section to provide the data needed for setup code validation and for the Code Selector tool. For more details, see the RDF Specification documents included in this package. For ease of use, documents are included for both Versions 3 and 4 of the specification, although the Version 4 document on its own includes a description of the differences between the two.

## **IR Release 7.15 Notes**

Added number tables up to 446.  
Commented out rest of code that references FT232R+PIC JP1 USB interface.  
Updated jp12serial.dll driver to version 0.10.

## **IR Release 7.14 Notes**

Added number tables up to 441.

## **IR Release 7.13 Notes**

Commented out code that uses experimental FT232R+PIC JP1 USB interface.

## **IR Release 7.12 Notes**

Fixed using JP1 USB interface as the default interface.

## **IR Release 7.11 Notes**

Commented out code that uses experimental FT232R JP1 USB interface

## **IR Release 7.10 Notes**

Added number tables up to 434  
Updated jp12serial.dll driver to version 0.09.  
Select Port Address can now display a 16-bit address.

## **IR Release 7.09 Notes**

Added number tables up to 430  
Updated Learned data parsing to better handle format errors. Error msg suggests using Advanced->Clean Upper Memory to clean out old learned data.



## **IR Release 7.08 Notes**

Added number tables up to 410

Updated jp12serial.dll to version 0.07 to handle serial ports greater than COM9 and to handle JP1.3 remotes using the Original JP1.2 design.

## **IR Release 7.07 Notes**

Changed Interface Tab Menu to include JP1 for the Parallel/Serial/USB items.

## **IR Release 7.06 Notes**

Updated jp12serial.dll to version 0.06.

## **IR Release 7.05 Notes**

Added JP12Serial.DLL version Number in About box. Version number appears only after first access with the DLL driver.

## **IR Release 7.04 Notes**

Added support for Labels RDF entry for RS15-100 Remote

Added number tables up to 409

Widened RDF Selection Dialog box

Added -sig option when calling ExtInstall

Fixed MultiMacros on Atlas5 remotes

Removed duplicate DeviceTypes from drop-down selection lists

## **IR Release 7.03 Notes**

Allow Protocol upgrades with no device upgrade entries.

Don't suppress message dialog about the device/protocol upgrade region overflowing.

Fixed disassembly of HCS08 protocol flag and burst bytes.

## **IR Release 7.02 Notes**

Fixed display of defined words in HCS08 and 6805 disassembly of protocol executors.

## **IR Release 7.01 Notes**

Added support for 2BytePid RDF specification for remotes that use 2 bytes for the protocol id.

Added support for S3F80 RDF entry.

## **IR Release 7.00 Notes**

Added support for JP1.1(SST), JP1.2(HCS08), JP1.3(S3F80) remotes and the JP1.x interface(requires use of jp12serial.dll). Fixed various bugs. Search location of gwiopm.sys driver for Parallel port interface has changed to be the current directory and then the directory where IR.exe is located. The IR installation folder should be to a local drive and not a network drive. Otherwise, the loading of gwiopm.sys(Parallel port driver) will fail.

## **New Features:**

Region Overflow: Added ability to overflow device and protocol upgrades into the Learned and/or Advanced Code(KeyMove/Macro) regions. The Free bytes text will be highlighted in yellow if space is used in the Learned and/or Advanced Code Regions.

Learned Signals: Added ability to edit/copy/paste learned data. Added ability(if Forced Learned Timings checked in Advanced Menu) to round the display of the raw learned data timings and also the splitting of the data for bi-phase signals.

Additional Summary Text: Added Learned Raw Data timings.

RAW Tabsheet: Added ASCII Signature display.

File Association: For XP/W2K users with admin rights, .IR file association is done everytime IR.exe is run. For XP/W2K users without admin rights, in order to change which version of IR is invoked when you double click on an IR file, the registry entry that you need to change is

HKEY\_CLASSES\_ROOT\Applications\ir.exe\shell\open\command

## **IR Release 6.20 Notes**

Added the ability to explicitly specify the EEPROM type ( $\leq 2K$  versus  $> 2K$ ). This was necessary to support certain remotes with large EEPROMs that didn't work in conjunction with IR's auto-detect feature. This functionality is available through the Interface menu, and also via radio buttons on the main screen (if "Always Display EEPROM Type" is checked on the Advanced menu). Note that this functionality is not currently supported for the USB interface.

## **IR Release 6.15 Notes**

Modified the order in which the KeyMove section is written when rows are moved within any of the AdvCode-related grids. In the prior version the order was: KeyMoves, Macros, SpecialProtocols, Favs. The new order is: KeyMoves, SpecialProtocols, Favs, Macros. This change will allow people with certain extenders to more easily use the special protocol functions.

## **IR Release 6.10 Notes**

### **New Features:**

Support For New Learned Command Format: Added LearnedDevBtnSwapped item to the General section of the RDF. Setting this to Y, Yes, T, True, or 1 will cause IR to interpret the second nibble (rather than the first nibble) of the second byte as the device index. The first

nibble will always be set to 2. This format is used by the URC-9960B01, and will probably be used by others.

Modified Parallel Port Initialization: Changed the control byte used to initialize the parallel port. This will likely have no effect on anyone, but *might* cause the interface to work in conjunction with old, non-standard ports.

## IR Release 6.00 Notes

### New Features:

New File Format: Notes are now tied directly to the entries (rather than to keystrokes, etc.). This means that if you change the key that a KeyMove is bound to, the note will not disappear. It also means that notes can be correctly associated with MultiMacros. Also, the [Buffer] tag has been removed and the buffer is in the same format as the old .TXT files. (Existing .IR files can still be read, but they will be converted to the new format when saved.)

File Save: The program will no longer save to both .TXT and .IR formats simultaneously. You can choose the format. (.TXT files do not contain note entries.)

Additional Note Support: Notes are now supported in TimedMacros and FavLists.

Auto File-Association: If no application is associated with .IR files, the program will automatically create the association.

Enhanced Support for Special Protocol-Related Functions: A new Spcl Prot Fns tab has been added to the main form, and functions made available via special protocols will be listed there (instead of in the Key Move page). This tab can display the values and accept input in a much more user-friendly way than the Key Move tab. Also, special protocols are now identified by their ProtocolIDs rather than device type/code (which are much more easily modified by the user.) IR currently supports the following special protocols:

- DSM: Extender-based device-specific macro
- LDKP: Extender-based long keypress (LKP) and double keypress (DKP)
- ToadTog: Extender-based toggle utility
- UDSM: Non-extender-based device-specific macro
- ULDKP: Non-extender-based device-specific macro (DSM), long keypress (LKP), and/or double keypress (DKP)
- Multiplex: Device multiplexer

To enable any of these within IR, simply add a [Special Protocols] section to your RDF and enter the special protocol (SP) identifier along with the ProtocolID associated with it for the given remote. (For example, ULDKP=01F9) The protocol must be installed in your remote, and a device upgrade must reference it in order for the Special Protocol Function to be usable. If IR does not yet support a Special Protocol, that protocol can still be displayed in the Special Protocols tab (if defined as such in the RDF), but its value(s) will be entered/displayed as hex.

Support for 5-Digit EFCs: IR now supports EFCs that can encode 2-byte HexCmds. These EFCs can be keyed directly into some newer remotes, but even if your remote doesn't support them natively, IR will recognize them and translate them into the correct format you're your remote.

EFC/HexCmd Translation: After bringing up the KeyMove dialog to edit an entry, the user can click between the EFC and HexCmd radio buttons in order to have IR translate between the values. Once the user has begun editing either of these values, clicking on the radio buttons will no longer invoke the translation (until the dialog is closed.)

Enhanced AdvCode Formats: Remotes that store KeyMoves in the new "EFC" format are now properly supported. (Specify AdvCodeFormat=EFC in the RDF.) This version also supports remotes that...

Use the "long" format. (Specify AdvCodeBindFormat=LONG in the RDF.)

Terminate the AdvCode section with \$FF (rather than \$00). (Specify SectionTerminator=\$FF in the RDF.)

Use 5-digit EFCs. (Specify EFCDigits=5.)

Enhanced EFC Calculator: The EFC calculator now supports 2-byte HexCmds and 5-digit EFCs. It will consider all 3-digit EFCs to be single-byte EFCs unless explicitly asked to convert them to 2-byte EFCs. For 2-byte values, only the second byte will be used for the OBC translation.

Enabling/Disabling Key Moves and Spcl Prot Fns Tabs: The Key Moves and Spcl Prot Fns tabs can now be explicitly disabled within the RDF. (Set KeyMoveSupport to N, No, F, False, or 0 in the RDF to disable KeyMoves, or omit the [SpecialProtocols] section to disable Spcl Prot Fns.)

Two-Byte MultiMacro Support: A second address can now be specified in the MultiMacro section of the RDF. If one address is specified then it will work as it always has. (The high order nibble of the byte at that address will contain the number of macros and the low order nibble will contain information about the next macro to fire.) If a second address is specified, then the first address will contain the number of macros assigned to the button (\$00 = 0 or 3, \$40 = 1, \$80 = 2) and the second address will contain the next macro to fire (which IR will always set to 0 so the sequence starts fresh after each download).

Enhanced Summary Support: The summary screen now supports multi-line notes and wrapping at the column level. This should allow the information to be displayed correctly on the screen. It can also be saved as a .TXT file or as an .RTF (rich text) file.

Enhanced Integration with External Tools: ExtInstall and IRTToWav can now be located in any directory, and they can be invoked from within IR without regard to whether the current image is a .TXT or .IR file, or a download that hasn't yet been saved.

Enhanced Wav Export: The user can now specify an output file name to be used when exporting to a file using IRTToWav. The export operation is available by selecting either "Export to Wav" or "Save As" from the File menu. When using "Export to Wav," users can now

generate .WAV files for all memory sections independently (including the TimedMacro and Learned sections.) IR can also play sound files directly for remotes that are sound-upgradable.

Direct Wav Upload: IR can now play the contents of a .WAV file so it can be directly uploaded to a remote that supports modem-based upgrades. (For this option to be enabled, specify WavUpgrade=1 in the RDF.)

Wav Import: IR can now import .WAV files created by IRTToWav. This can be accomplished by selecting either “Open” or “Import from Wav” on the File menu.

New Command-Line Parameters: The command-line parameters have been changed to be more in line with standard conventions. The syntax is:

IR.EXE [<File Name> [/O <Output File Name>]]

If a file name is specified then that file will be loaded. If an output file is specified then that file will be generated (in the same syntax as the summary screen). If the output file has a .RTF extension then it will be written to in rich-text format. Otherwise it will be written to as a standard text file. After the file is written, the program will automatically terminate.

Enhanced UI: The program now saves all window attributes for each resizable window. It also remembers the column widths of all grids within the main form. (Some additional cleanup was done to align objects a bit better.)

Suppression of Device Code Offsets: You can now disable IR’s automatic device code correction (used in conjunction with remotes such as the RCU-810 that display values that are different from the ones stored in memory). If the offset is non-zero, then it will normally be displayed on the General page above the Device Buttons grid. If it is suppressed, then a warning message will be displayed there instead.

Support for Nested Macro Display: Because some extenders and special protocols support the nesting of macros, IR can now display the entire list of commands (keys) that will be sent when a given macro is invoked. This is currently only supported for certain special protocol functions, but can be expanded in the future for extenders as well.

Suppression of Learned Timings: A “Force Learned Timings” menu item has been added to the Advanced menu. If left unchecked (the default), then the raw timing information about learned signals will not be displayed if the decoder identifies the signal. If the menu item is checked, then the raw timings will be displayed regardless of whether or not the decoder identifies the signal.

Support for MaxProtocolLength: IR will now enforce the MaxProtocolLength value in the RDF.

Support for OmitDigitMapByte: IR will now recognize the OmitDigitMapByte value in the RDF and support remotes whose upgrade device entries don’t contain a digit map byte.

Added [AutoSet] Section to RDF: A new [AutoSet] section has been added to the RDF which allows the RDF designer to specify addresses that will always be set to a given value. This is very similar to the [FixedData] section, except that FixedData is considered to be invariant. (If

IR sees a FixedData conflict, it considers it an error and warns the user.) IR makes no assumptions about AutoSet data. It simply sets the specified bytes automatically.

Added Unsafe Upload for Advanced Users: An “Enable Unsafe Upload” option was added to the Advanced menu. This will put a button on the General tab that will allow IR to upload to the remote without first checking the remote’s signature. This is useful for people who add EEPROMs to remotes and for people with remotes that don’t have standard signatures, but should be avoided by all other users.

Added Raw Download for Advanced Users: An “Enable Raw Download” option was added to the Advanced menu. This will put a button on the General tab that will allow IR to download from the remote without looking for a matching RDF. (At that point IR will display only the General and Raw tabs.) This is useful for people with remotes that don’t have standard signatures.

## IR Release 5.15 Notes

Fixed bug with history.

## IR Release 5.14 Notes

Recalled.

## IR Release 5.13 Notes

Added the ability to run IR from the command line and automatically create a text file output containing the same information as the summary screen.

The command line parameters are as follows:

`ir.exe inputfilename p destinationfilename`

The **p** flag is to instruct IR to print the summary out the destination file.

Notice that the **inputfilename** and **destinationfile** names should be fully qualified. The resulting file will be saved as a .txt

Also added an Export To File button from the summary page that will create the same output as above.

## IR Release 5.12 Notes

Fixed bug when importing Upgrades from KM that contain notes.

## **IR Release 5.11 Notes**

Fixed bug with MRU and Learned Tab focus

## **IR Release 5.10 Notes**

Added support for notes in learned signals;

## **IR Release 5.09 Notes**

Reversed changes from 5.08 since it didn't work

## **IR Release 5.08 Notes**

Fixed bug with "." In directory names.

## **IR Release 5.07 Notes**

Added support for USB interfaces that connect the RESET line to Delcom's pin 5 (port 1 bit 0) rather than pin 3 (port 0 bit 0).

## **IR Release 5.06 Notes**

Added more checking for ExtInstall.

## **IR Release 5.05 Notes**

Fixed bug when selecting File|Save As.

## **IR Release 5.04 Notes**

This release supports converting standard IR files to a WAV file. Note that this is only available with certain remotes.

To use this feature open the .TXT version of the IR file (currently IRToWav only supports this version) then select File | Covert File to Wav.

## IR Release 5.03 Notes

This release supports converting standard IR files to the extender version using the ExtInstall. For this to work the ExtInstall.exe and the RDF's must be in the same directory. This includes the normal and the extended version of the RDF's in question.

To use this feature open the .TXT version of the IR file (currently ExtInstall only supports this version) then select File | Covert File for extender. Follow the instructions for selecting the Hex definition file and the location and name for the output file. The new file will be automatically loaded in IR upon completion of the conversion.

## IR Release 5.02 Notes

Fixed bug when copying upgrades from RM.

## IR Release 5.01 Notes

This version of IR supports the embedded notes function from KeyMaster 8.21. Any notes created in KM 8.21 (see KM for specifics) will be carried over into IR when pasting Device or Protocols Upgrades. It will also carry over any notes associated with KeyMoves that are pasted in as part of a Device Upgrade.

## IR Release 5.00 Notes

This new version of IR has significant enhancements in the area Special Protocols, notes for Macros, KeyMoves, Devices, Protcols and a Global note on the general panel. It is also different than the prior IR in that it creates the notes and traditional IR upgrade information in one file, named exactly the same way the previous version names the file but uses an [.IR](#) extension instead. Once you have saved the file as an [.IR](#) you can open and use it in IR5.00 in a normal way. IR 5.00 also saves the buffer info in the traditional TXT file in case you need it to use with other apps that do no support the new file format at this time. However, remember that if you open the .TXT file and select an explicit save any notes or other changes in the .IR file will be overwritten. Therefore, think carefully before answering YES.

### **WARNING: The .IR format has the following limitations**

- 1) This .IR version of the file has a significantly different format to incorporate the notes.
- 2) It is not compatible with the previous versions of IR.
- 3) It is not compatible with the current version of ExtInstall for extender installation



4) It is not compatible with the current version of IRtoWav.

## **IR Release 4.02 Notes**

### **Bug Fixes:**

USB interface wasn't working for remotes with EEPROMs smaller than 2K. This has been repaired.

### **New Features:**

IR can now load the I/O port driver (gwiopm.sys) from the Windows or Windows\System directory. This allows users to run IR from any drive (not just the boot drive) as long as they copy gwiopm.sys to one of these places.

Several new DigitMap entries have been added.

## **IR Release 4.01 Notes**

### **New Features:**

A "Set RDF Path" option has been added to the File menu to allow users to specify where the RDF files are stored. This will allow users to more easily share RDFs across applications, and will help to keep the IR directory less cluttered.

## **IR Release 4.00 Notes**

### **New Features:**

Support has been added for two new interfaces. IR now supports a serial interface and a USB interface (using the Delcom USB chip) in addition to the parallel interface.

The Interface | Transmission Delay menu item has been moved to the Advanced menu.

A "Configure Interface" option has been added to the Advanced menu for both the parallel and the serial interfaces. This allows interface hardware designers to indicate which ports/offsets/pins a new interface is connected to. To access these menu options, a value named "SpecialHardware" must be added to the registry under HKEY\_CURRENT\_USER\Software\UEIC. The type and value of this entry doesn't matter. (Note: Manually modifying the registry is not recommended for normal users. This option is only intended for experts.)

Read-verification has been made optional. In prior versions, IR always read back the contents of the EEPROM after writing to it in order to verify that the write successful. This re-reading is probably overkill since problems encountered while writing should be identified as they occur, so users can now go to the Interface menu to disable the read-after-write verification.

IR now allows buttons to be restricted within the RDF. For each button, the RDF developer can now indicate where the button can be used and where it can't. For example, a device button (such as VCR) can be defined so a user can't *bind* a macro to it, but it can be *used* within a macro. These button restrictions should allow all of the P8-style remotes to be defined correctly with respect to their Shift and XShift capabilities as well.

The user can now specify both Shifted and XShifted keys. The names used by IR to label the "Shift" and "XShift" states can now be specified in the RDF. This will hopefully help to reduce confusion for remotes that contain an actual Shift key.

Support has been added for the URC-6131, and any other remote that stores KeyMoves in EFC format rather than as Hex Commands. (In this format, single-byte entries are interpreted as KeyCodes and two-byte entries are interpreted as 16-bit EFCs. Longer entries are preserved as is, but are interpreted as KeyCodes.)

An optional *Identification* value is now available for RDFs. If specified, then this will be displayed for each matching RDF when multiple RDFs match a given signature. The identification string should provide the information needed by the user to allow him/her to determine which RDF is appropriate to the given remote.

RDFs now recognize quoted strings, so key names can contain spaces.

Boolean values in RDFs can now be set to Y, N, Yes, No, T, F, True, False, 1, or 0.

## IR Release 3.20 Notes

### Bug Fixes:

Various minor bugs have been fixed. Unfortunately, version control went down so I don't have a good list.

### New Features:

The DecodeIR DLL has been integrated into IR to allow for better analysis of learned signals.

Several RDFs have been added to the distribution zip.

Several entries have been added to the digit map tables to support additional remotes.

## **IR Release 3.12 Notes**

### **Bug Fixes:**

Repaired a problem that caused a Privileged Instruction error when changing the port address on NT-based systems.

## **IR Release 3.11 Notes**

### **Bug Fixes:**

Import of key moves from KM had been discarding the first key move if the device the key was bound to had not yet been defined.

Registry key for ExeFile entry has been changed to HKEY\_CURRENT\_USER\Software\VB and VBA Program Settings\Keymap-Master\IR.

### **New Features:**

Several RDFs have been added to the distribution zip.

Several digit map entries have been added.

## **IR Release 3.10 Notes**

### **Bug Fixes:**

Modified the handling of “empty” signatures to allow uploads/downloads to complete when the signature is all \$00s. (Previous versions assumed that an empty signature was necessarily an indication of a bad interface and did not allow the operation to continue.)

A “List Index” error (which occurred when a bad EEPROM image was loaded) has been fixed. IR will now issue a warning.

Several minor problems have been fixed in the protocol disassembler.

### **New Features:**

IR now supports remotes with EEPROMs larger than 2K bytes.

The EEPROM size is now auto-detected. A warning message will be displayed if the detected size differs from the size listed in the RDF.

The list of RDFs displayed in the Select Remote dialog is now displayed in sorted order, either by signature or description.

If multiple RDFs match the remote's signature, IR tries to automatically select the correct one.

Key moves and macros can now be "imported." This allows device upgrades to be fully defined within (an upgraded) KM, even if the keys that you want to assign functions to are not mapped for the current device.

IR now writes the full file name (including path) of the program to HKEY\_CURRENT\_USER\Software\VB and VBA Program Settings (in the registry) each time it is started.

## **IR Release 3.07 Notes**

### **Bug Fixes:**

The items in the Learned Tab's scroll box should now be positioned correctly regardless of the position of the scroll bar.

### **New Features:**

IR now has the capability of setting the clock when uploading to certain P8-style remotes. (Downloading will still reset the clock to 12:00AM.) This functionality must be supported in the remote's RDF.

The home theater settings on certain P8-style remotes can now be stored in the EEPROM. (Again, this functionality is dependent on RDF support.) This means that HT settings programmed via IR will be persistent. One limitation is that IR cannot read the HT settings from the remote, so any changes you make directly to the remote (without using IR) will be lost if the remote is reset.

When using the updated RDF in conjunction with an existing dump, you will get a message that the checksum is incorrect in the \$700 region. This message will be followed by a dialog informing you that some fixed data is incorrect. Both of these messages are normal. Simply tell IR to replace the existing data with the fixed data and then fill in your home theater settings on the General tab.

Note that in order to achieve the new functionality, the size of the timed macro section has decreased slightly. It's possible that you'll get an error if you have too many timed macros defined.

Badly learned signals are now broken down into two categories. The first category is for signals that we don't currently know how to decode, and the second is for signals that contain a portion that fires only when the button is released. Signals that we can't decode are often learned properly, but IR doesn't yet know how to deal with them. Signals of the other type are most likely learned badly and need to be relearned.

IR will now check to make sure that the EEPROM size is set correctly (in the RDF) every time it downloads from the remote. It will issue a warning message if it encounters a conflict.

## **IR Release 3.06 Notes**

### **Bug Fixes:**

The device-specific upgrade section was not properly being loaded. Only the first upgrade in the section was being loaded, so any subsequent save would wipe out all but the first one. This has been fixed so you can now store (and load) multiple device-specific upgrades.

An access violation was occurring sporadically when the mouse cursor moved over a combo box's open drop-down list.

The forms have been shifted around slightly so data is no longer cut off when using large fonts.

### **New Features:**

A "Purge Memory" feature has been added to the Advanced menu which sets all bytes in the buffer to \$FF. This can be used as an aid to create pristine dumps, but should only be used by people who know what they're doing.

## **IR Release 3.05 Notes**

### **Bug Fixes:**

A few timing problems have been corrected in the interface check and the upload/download routines.

### **New Features:**

Multiple instances of IR can now be running at the same time.

IR now supports the ability to store multiple macros on a single button (for remotes which support that functionality).

# IR Release 3.00 Notes

## Bug Fixes:

Burst timings in the "learned" tab were sometimes being displayed as double their actual length.

A few machine instructions were being disassembled improperly.

## New Features:

Brand new interface routines -- All code that talks to the remote has been rewritten. It should be more stable, less likely to provide false errors, and more helpful when it encounters an error. Also, explicitly invoking the interface check is no longer required because it is built into the upload/download routines. (Thanks to Tommy for all his help!)

Modified the way IR deals with bit-mapped settings such as volume punch thru, sleep status, etc. All of these settings can now be supported through generically through the RDF, which means that IR can now support all of the different implementations of VPT (including those that define VPT in multiple bytes). It can also support less common features such as foreign language selection, etc. And as more of these settings are discovered, they will be able to be supported with a simple addition to the RDF rather than requiring an upgrade to IR.

Supports "device-specific" upgrades for P8-style remotes. (This is the memory area from \$500..\$6FF.) This is especially useful in conjunction with older P8 models which have a bug that prevents normal upgrades (in the \$100 section) from working.

Supports "timed macros" for P8-style remotes. (This is the memory area from \$700..\$7FD.)

Supports Fav/Scan function of the 15-1995 remote, and most likely any P8-style remote with a Fav/Scan button. (The RDF will need to be updated for other P8 remotes, as I don't know which ones they are.)

Supports OEM device types for remotes (such as the Replay and the Outlaw) that contain this sort of device type.

Prompts you to save the buffer before any operation that will cause unsaved changes to be lost.

A "Revert to Saved" option (which throws away changes) has been added to the File menu.

The data in the Raw tab now displays data that has changed in bold. Also, the "Baseline" checkbox has been converted into a radio button that allows you to switch between viewing the data in the buffer, the data in the baseline, and the unmodified data.

The application's size and state is now preserved across sessions. This should be helpful for people running either very low or very high resolution.

Changed some terminology: Advanced codes are now known as Key Moves, etc.

A "clone" feature has been added for key moves, macros, and timed macros, which allows the user to add a new item based on an existing one.

Added a "device grid" to the Key Move add/edit dialog that displays all device type/code combinations that have been used, allowing you to more easily specify devices that have already been specified elsewhere. This dialog is also resizable, and its height will be preserved across sessions.

Added a Summary option to the main menu that displays a textual summary of all the settings in the remote. (Also added buttons next to each display grid that copy the contents of the grid into the clipboard.)

Added code to set ECP ports to operate in SPP mode. Also added code to set the port to output mode, if for some reason it had been set to input mode. This should theoretically remove the need for people to play with their port settings in Windows and/or the BIOS.

Error messages have been updated to (hopefully) be more informative.

Added a number of RDFs, and updated several others. (Thanks Rob!)

Supports multiple RDFs for remotes whose out-of-the-box signatures may not be distinct. In this case a dialog box will prompt you to select the RDF file appropriate to your remote. This will likely become annoying pretty fast, so it is assumed that users who own one of these remotes will rename the RDF that isn't applicable to their remote, and/or potentially write a new signature to their remote to make it unique.

Added a "Clean Memory" function (available on the Tools menu) that sets all unused bytes from \$100 and above to \$FF. This will not effect the data from the currently loaded dump. (This was added to aid in manual analysis of these sections, and is not expected to be of much value to the typical user.)

Added a check when adding/editing key moves to make sure that a device/key combination only has one key move assigned to it. Similarly, added a check when adding/editing macros to make sure that a key has only one macro assigned to it.

The meaning of AutoCheck has changed. There is no longer a need to check the interface with the remote unplugged, so the interface check is now done (transparently) before every read/write operation. In extremely rare circumstances (with "marginal" remotes, interfaces, and/or parallel ports) the interface test fails but the interface will still work. The AutoCheck menu item allows users with this problem to disable the automatic test. Note that regardless of the AutoCheck setting in previous versions, AutoCheck with default to "On" the first time you run 3.0.

The behavior has changed when the user enters an edit region by clicking on it. Previously the region defaulted to unselected, but now it defaults to selected. (This is consistent with what happens when the user tabs into an edit region.)

Added a space at the end of each line of data in the Device and Protocol edit dialogs. This is helpful for those people who want to cut and paste existing protocols and manipulate them.

Users can now paste upgrade devices and/or protocols into the device and protocol edit boxes, even if they have commas embedded in them. This is helpful for users who want use the new CSV files created when you hit SAVE in keymap-master

The look-and-feel of the user interface has changed in certain places to support the new functionality.