

Handheld Project54 ButtonControl Application

Kriste Krstovski

We developed the Handheld Project54 (P54H) ButtonControl Application to utilize all built-in hardware buttons found on Windows CE powered handheld computers by the P54H software system. Through this application, Project54 inter-application messages [1] can be assigned to any of the built-in handheld computer hardware buttons. Pressing and releasing of these buttons could generate different Project54 inter-application messages.

Built in Handheld Computer Hardware Buttons

Different handheld computers have different alignments of the built-in hardware buttons. Shown in Figure 1 are Windows CE powered handheld computers that Project54 currently has available for development purposes.



Figure 1 Windows CE powered handheld computers running P54H software system

One general characteristic of all Windows CE based handheld computers is that they all have five hardware buttons (emphasized with orange circles in Figure 1) assigned by default to activate each of the Personal Information Management (PIM) applications that are part of the Windows CE Operating System (OS). PIM applications include Calendar, Contacts, Email, Notes, and the voice notes recording feature of the Notes application. Hardware buttons can be assigned to different applications using the Button application which can be found under the settings sub-application menu.

Hardware buttons implemented on any Windows CE based handheld computer generate the WM_KEYDOWN Windows CE OS system message when pressed and the WM_KEYUP system message when released [2]. The WPARAM value of these Windows CE system messages is 000000CX where X is an integer from 1 to 5 assigned by the Operating System (OS) for a particular button. For example, the hardware button marked with the Calendar icon (number 1 in Figure 1) has X=1 assigned to it. Shown in Figure 2 is the Windows CE Remote Spy development tool application capturing the Windows CE system messages sent by the handheld computer hardware buttons. The handheld computer used for capturing the Windows CE system messages is HP's iPAQ hx4700. The WM_KEYUP and WM_KEYDOWN system messages sent by each button are highlighted with orange rectangles. The button number is shown to the right of each rectangle. While the button is being pressed, the WM_KEYDOWN message is constantly being sent. This is not apparent from Figure 2 since in this instance all the hardware buttons were pressed and immediately released.

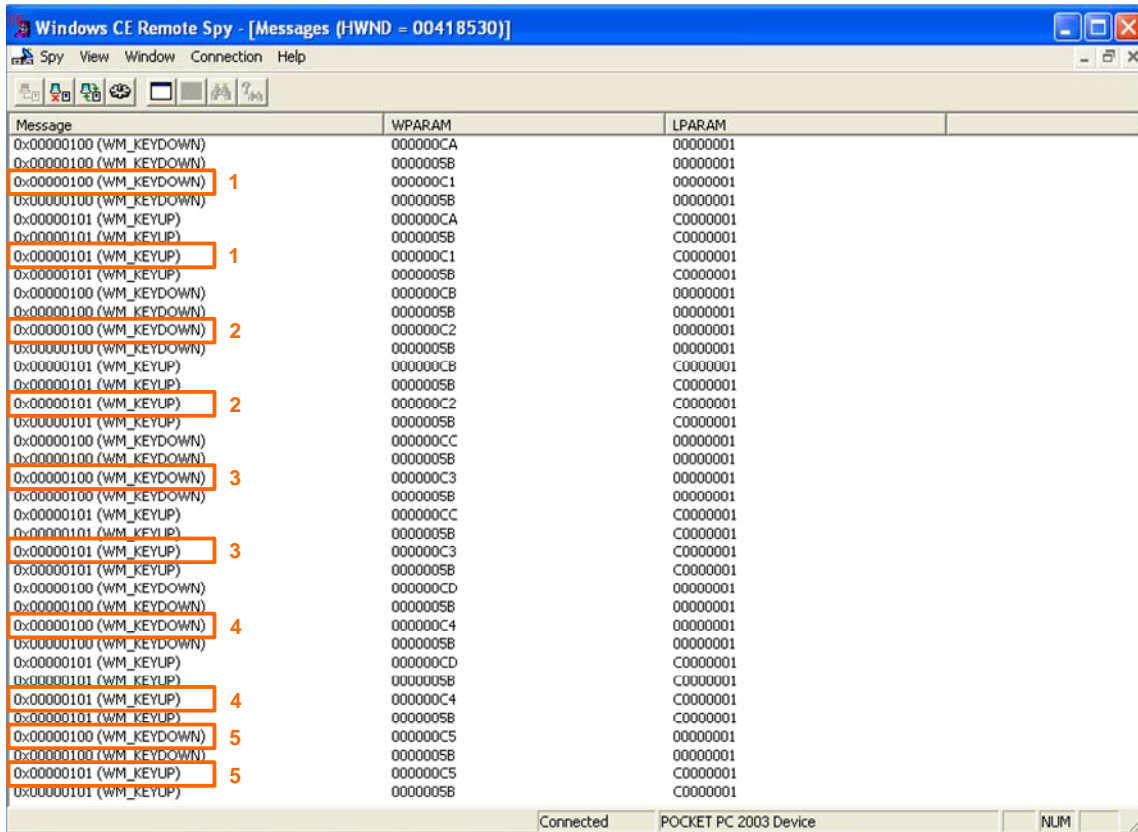


Figure 2 Windows CE Remote Spy Developer Tools application capturing system messages sent by the five hardware buttons found on HP's iPAQ hx4700 handheld computer.

Project54 Inter-Application Messages

The ButtonControl Application captures all Windows CE system messages sent by hardware buttons implemented on the handheld computer. It uses the P54H Graphical User Interface (GUI) Component [1] to take control of all the hardware buttons while the P54H application is running. Whenever a hardware button is pressed the P54H GUI Component redirects the WM_KEYUP and WM_KEYDOWN system messages to the ButtonControl Application. Within the ButtonControl Application these Windows CE system messages are being processed and Project54 inter-application system messages

are being generated. Project54 inter-application messages are assigned for each button press and release state using the registrysettings.txt setup file.

Shown in Figure 3 is a sample content of the registrysettings.txt setup file, specifically the ButtonControl Application section.

```

registrysettings.txt - Notepad
File Edit Format View Help
{ButtonControl}
  <Messaging>
    [self] pdabuttoncontrol
    [Appmanager] pdamainscreen
    [Speechio] pdaspeechio
    [Logger] pdalogger
  <ButtonAssignment>
    [C50N] pdaspeechio LISTEN
    [C50FF] pdaspeechio ENDLISTEN
    [C40N] pdalights SHOW WINDOW
    [C40FF] pdalights SHOW WINDOW
    [C30N] pdaradar FRONT ANTENNA
    [C30FF] pdaradar REAR ANTENNA
    [C20N] NONE NONE
    [C20FF] NONE NONE
    [C10N] pdamainscreen SHOW WINDOW
    [C10FF] pdamainscreen SHOW WINDOW
  
```

Figure 3 ButtonControl Application parameters within the registrysettings.txt file

Listed under the “ButtonAssignment” subsection are the five hardware buttons’ press and release states. Next to each of the five hardware buttons’ states are the names of the destination applications and message body content that is to be sent. The ButtonControl Application uses these parameters to compose the Project54 inter-application system message that will be sent to the destination application specified for that hardware button. When a hardware button press and release states have the “NONE” string specified as the destination application and message body, the ButtonControl Application sends the “BUTTON CXON” and “BUTTON CXOFF” messages to the P54H application whose Graphical User Interface (GUI) is currently displayed. “X” in these messages is an integer from 1 to 5 assigned by the OS to the particular button. In the above example the second button, which is originally used for instantiation of the “Contacts” PIM application, is utilized in this fashion. It is up to the developer of the

P54H application to decide on how it will process the “BUTTON CXON” and “BUTTON CXOFF” messages. The ButtonControl Application keeps track of the P54H application whose GUI is currently being displayed by receiving the “INFOCUS” Projec54 inter-application message. This message is being sent by each P54H application when it displays its GUI.

Using this approach we can utilize hardware buttons found on different handheld computer models for different purposes in the context of the application whose GUI is currently displayed or for one particular purpose, regardless of the type of application GUI displayed.

The current version of the ButtonControl Application processes the five built-in hardware buttons commonly found on every Windows CE powered handheld computer. Other built-in hardware buttons could be processed and utilized by simply specifying their press and releases states in the registrysettings.txt file and assigning Project54 inter-application messages to these states. The ButtonControl Application could be extended to utilize combinations of hardware button presses and releases following the same approach.

Reference List

- [1] Pelhe, A, "One-to-One Communication between Objects in the Project54 System Software." Master Thesis, University of New Hampshire, 2003
- [2] Boling, D., "Programming Microsoft Windows CE", Microsoft Press, Redmond, WA, 2001