Verwendung des externen Interfaces von LOGGER32(Using the Logger32 External Interface)

Bob Furzer K4CY

Programmierer, die ihr Programm direkt mit LOGGER32 verbinden wollen, können das auf zwei Wegen machen :

- Verwendung der CW-Maschine von LOGGER32 als alleinstehend ausführbares Programm oder
- Verwendung der Windowsmeldungen und der Zwischenablage, wie sie von LOGGER32 unterstützt werden

Beide Möglichkeiten sind momentan in ihrer Leistungsfähigkeit noch begrenzt, aber ich werden sie – falls nötig –erweitern- Die Kodierbeispiele zur Erweiterung sind in VB angegeben..

Emulation der CW-Maschine (CW Machine emulation)

Das Interface verwendet eine Kombination von im Speicher abgelegten Daten und Windowsmeldungen. Die Nutzeranwendung muss eine Windowsmeldung als von LOGGER32 kommend registrieren und sie als **hWnd** der LOGGER32-.MDI- Form identifizieren.

```
CWMsg = RegisterWindowMessage("CW")

L32hWnd = FindWindow(vbNullString, "Logger32")
```

Die Nutzeranwendung kann nur Mitteilungen an LOGGER32 unter Verwendung von **SendMessage** oder **PostMessage** senden. Erkannte **wParam-Werte** sind : :

100 Log the QSO with the data currently entered into the Logbook Entry window

150 If lParam = 1 Then stop polling the Radio. If the lParam = 0 then start Radio polling

200 Check the memory mapped file for updates

300 To 310 KeyUP the CW line on a serial port number of wParam - 300

400 To 410 KeyDown of the CW line on a serial port number of wParam - 400

500 To 510 PTTOn of a serial port number of wParam - 500

600 To 610 PTToff of a serial port number of wParam - 600

700 Clear all Logbook Entries

800 Turn the rotor

An LOGGER32 wird eine Mitteilung geschickt, dass LOGGER32 in der im Speicher abgelegten Datei

```
PostMessage L32hWnd, CWMsg, 200, 0
```

nach neuen Einträgen nachsehen soll.
Der Nutzer kann anch Schreib- und Lesedaten (mit einer Null abgeschlossenen Zeichenketten) in folgender Struktur nachsehen :

```
Public Type CWStruct
Callsign as String * 64
Name as String * 64
SerialNumber as String * 64
CurrentOperator as String * 64
Band as String * 64
Logger32Info as String * 64
```

```
End Type
Public CWStructure as CWStruct
```

LOGGER32 beschreibt in alle Felder, wenn sich die Daten in LOGGER32 ändern. Die externe Anwendung muss regelmässig in diesen Felder nachsehen, ob sich etwas geändert hat.

Das Feld SerialNumber enthält die Zeichenkette (null terminated string) "###", wenn die Seriennummer nicht freigegeben ist oder eine numerische Zeichenkett (mit Null abgeschlossen), wenn die Seriennummer freigegeben ist.

Callsign und **Name** sind die einzigen Felder, die von LOGGER32 gelesen werden. LOGGER32 liest den Datenblock, um Änderungen in **CWMsg** mit dem lParam von 200 zu erkennen..

Das **Logger32Info field,** das von LOGGER32 regelmässig beschrieben wird, hat im Moment drei Optionen :

- "**Close**" um der Anwendung mitzuteilen, dass LOGGER32 geschlossen wird
- "**Minimize**"um derAnwendung mitzuteilen, das **WindowState = 1** ist
- "**Norma**l" um den **WindowState = 0** zu setzen

Sie brauchen etwa diesen Code, um in die Datei zu lesen/zu beschreiben :

```
hFilemapCW = OpenFileMapping(FILE_MAP_ALL_ACCESS, False, "CW")

If hFilemapCW = 0 Then

MsgBox "OpenFileMapping API failed, is Logger32"

Exit Sub

Else

pFilemapCW = MapViewOfFile(hFilemapCW, FILE_MAP_ALL_ACCESS, 0&, 0&, 0&) ' &H2 Or &H1
Or &H10)

If pFilemapCW = 0 Then

MsgBox "MapViewOfFile API failed."

Exit Sub

End If

End If
```

Bevor Ihre Anwendung beendet wird, sollte sie hFileMapCW und pFileMapCW etwa so schliessen : is

```
UnmapViewOfFile pFilemapCW

CloseHandle hFilemapCW
```

Wenn Sie ein Rufzeichen von Ihrer Anwendung an die Speicherdatei übergeben wollen, gehen Sie etwa so vor :

```
If pFilemapCW <> 0 Then

With CWStructure
```

```
.Callsign = CallsignField.Text & Chr$(0)

CopyMemory ByVal pFilemapCW, ByVal .Callsign, Len(.Callsign)

End With

PostMessage L32hWnd, CWMsg, 200, 0

End If
```

Zur Übergabe eines Namens gehen Sie so vor :

```
    If pFilemapCW <> 0 Then

    With CWStructure

    .Name = NameField.Text & Chr$(0)

    CopyMemory ByVal pFilemapCW + 128, ByVal .Name, Len(.Name)

    End With

    PostMessage L32hWnd, CWMsg, 200, 0

    End If
```

LOGGER32 als externes Interface verwenden (Using the Logger32 external interface)

Für den Datenaustausch zwischen LOGGER32 udn vom Nutzer entwickelten Anwendunfen wurde eine Reihe von Windowsmitteilungen kodiert und der Datenaustausch über die Zwischenablage vorbereitet.

Die Nutzeranwendung muss eine Windowsmeldung als von LOGGER32 kommend registrieren und sie als **hWnd** der LOGGER32-.MDI- Form identifizieren.

```
L32Msg = RegisterWindowMessage("Logger32")

L32hWnd = FindWindow(vbNullString, "Logger32")
```

Die Nutzeranwendung kann nur Mitteilungen an LOGGER32 unter Verwendung von **SendMessage** oder **PostMessage** senden. Erkannte Mitteilungen sind : :

```
SendMessage L32hWnd, L32Msg, 3, Me.hWnd
```

LOGGER32 erkennt folgende **wParam-Werte**

- 3 An external application has started. The hWnd of the external application must be in the lParam. Logger32 will respond
  to this message with a message to you with a message wParam 100 and the Logger32 radio frequency in the lParam field.
- 4 The external program has stopped. Send this before you shut down.
- 5 Tell Logger 32 to shut downLogger32
- 6 Logger32 has PTT control. Your application hWnd must be in the lParam field
- 7 Logger32 does not have PTT control. Your application must be in the lParam field.
- 8 PTT on. If Logger32 has PTT control, Logger32 will turn PTT on either by Radio Command or

Shared Serial port
(based in internal Logger32 setup parameters).

- 9 PTT off
- 10 Populate the Logbook Entry window with the data currently in the Clipboard, and log the QSO.
- 11 Populate the Logbook Entry window with the data currently in the Clipboard, and do not log the QSO.
- 12 Log the QSO with the data already populated in the Logbook Entry window.
- 13 If lparam = 0 then read the Clipboard for the frequency. If lParam = 1 then set the radio frequency to the frequency
in the Clipboard. Logger32 will then clear the Clipboard
- 14 If lParam = 0 then read the clipboard for the Mode. If lParam = 1 then set the radio mode to the mode in the Clipboard.
Logger32 will then clear the Clipboard.
- 15 If lParam = 1 then set the Logger32 flag to mark this QSO for (paper) QSLing. If lParam = 0 then turn off the QSLing flag
- 16 If lParam = 1 then set the Logger32 flag to mark this QSO for eQSLing. If lParam = 0 then turn off the eQSL flag.
- 17 If lParam = 1 then set the Logger32 flag to mark this QSO for LoTW. If lParam = 0 then turn off the LoTW flag.
- 18 If lParam = 1 then set the radio frequency and mode from the data in the Clipboard (in the format 14003.45|CW).

Für die wParam – Werte 10 und 11 müssen Sie formatierten Text im ADIF-Format in der Zwischenablage bereitstellen. LOGGER32 liest die folgenden ADIF-Parameter und trägt sie in das Logeingabefenster ein :

- CALL
- COMMENT
- RST_RCVD
- RST_SENT
- NAME
- QTH
- APP_TIME_ON
- APP_TIME_OFF
- APP_QSL "Y" ist der einzige mögliche Wert
- APP_eQSL “Y” ist der einzige mögliche Wert
- APP_LoTW “Y” ist der einzige mögliche Wert
- MODE
- FREQ Dieses Feld wird nur gelesen, wenn kein Transceiver angeschlossen ist.

Die folgenden Felder werden nur übernommen, wenn im Logeingabefenster von LOGGER32 ein Feld dafür vorgesehen ist, andernfalls werden sie ignoriert.

- STATE
- GRIDSQUARE

- IOTA
- CNTY
- STX
- SRX
- QSL_VIA

LOGGER32 löscht die Zwischenablage, nachdem die Daten gelesen wurden. Falls die Zwischenablage nur ein <EOR> enthält, wird das Logeingabefenster gelöscht. clears the Clipboard after reading the data.

Die Mitteilungen von LOGGER32 an Ihre Anwedung verwendet folgende **wParam-Werte** : :

16 To tell your application to terminate (sent when Logger32 shuts down)

100 The frequency is in the lParam. This message is sent when the radio frequency changes.

101 To tell your application to read the Clipboard for ADIF formatted fields CALL, RST_SENT, RST_RCVD, NAME and MODE (there is no ADIF header, or <EOR> included in the format.

102 This is sent when Logger32 logs a QSO (maybe you want/need to clear some fields in your application)

Zusätzlich teilt LOGGER32 Windows (nicht Ihrer Anwendung) über ShowWindow API mit, dass Ihre Anwendung versteckt (hide) oder angezeigt (show) werden soll, wenn LOGGER32 minimiert oder wiederhergestellt worden ist.