

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

Bob Furzer K4CY

Eine Serie von Windows-Mitteilungen wurde programmiert, um den Datenaustausch zwischen LOGGER32 und externen, von Nutzern entwickelten Anwendungen zu ermöglichen. Diese Mitteilungen benutzen das ADIF-Format.

Der erste Schritt zu Herstellung der Kommunikation zwischen LOGGER32 und Ihrer Anwendung ist die Identifikation der *hWnd* der **LOGGER32-MDI** :

```
Dim L32hWnd as long
L32hWnd = FindWindow(vbNullString, "Logger32")
```

Läuft LOGGER32 noch nicht, muss Ihre Anwendung die Abfrage periodisch wiederholen, bis LOGGER32 gestartet ist.

Als zweites muss Ihre Anwendung eine ungebrauchte Windowsmitteilung registrieren/identifizieren. LOGGER32 kann simultan bis zu fünf externe Anwendungen unterstützen :

```
Dim L32Msg as long

L32Msg = RegisterWindowMessage("Logger32 1").

PostMessage L32hWnd, L32Msg, 1, GhW ' GhW is the hWnd of your main form
```

LOGGER32 antwortet auf Ihre Anwendung mit einer Mitteilungsnummer 0. Falls die **L32Msg** , die Sie senden, von einer anderen externen Anwendung verwendet wird, wird **IParam** auf 0 gesetzt. Ist die **L32Msg** nicht verwendet, wird **IParam** auf 1 gesetzt. Falls Ihre Verbindungsanforderung zurück gewiesen wurde (**IParam** = 0), muss Ihre Anwendung es erneut mit einer anderen **L32Msg** versuchen:

```
L32Msg = RegisterWindowMessage("Logger32 2").

PostMessage L32hWnd, L32Msg, 1, GhW ' GhW is the hWnd of your main form
```

LOGGER32 unterstützt bis zu fünf externe Anwendungen. Ihre Anwendung kann die Verbindung unter Verwendung einer **RegisteredWindowMessage "Logger32 1" to "Logger32 5"** erneut versuchen.

Hat Ihre Anwendung die Verbindung zu LOGGER32 erfolgreich aufgebaut (Sie erhalten eine Mitteilung (Message) mit der Nummer 0 und einem IParam = 1), müssen Sie nun eine Mitteilung mit der Nummer 2 und einem IParam der TextBox an LOGGER32 senden. hWnd Logger32 sendet ebenfalls Zeichenketten (text strings). Falls Ihre Anwendung keine unangeforderten Daten von LOGGER32 empfangen will, setzen Sie IParam auf 0.

```
PostMessage L32hWnd, L32Msg, 2, TextBox.hWnd
```

Sie können unmittelbar nach der Mitteilung 2 an LOGGER32 eine weitere Mitteilung 3 an LOGGER32 senden. Wenn Sie IParam = 0 oder keine Mitteilung senden, teilen Sie LOGGER32 mit, dass es keine DXSpot-Information an Ihre Anwendung schicken soll. Die meisten Anwendungen werden diese Daten sicher nicht benötigen. Wenn Sie IParam auf 1 setzen :

```
PostMessage L32hWnd, L32Msg, 3, 1
```

teilen Sie LOGGER32 mit, dass es einen **ADIF**-formatierten Text an Ihre Anwendung nach jedem von LOGGER32 empfangenen **DX Spot** senden soll. Falls Sie vorher eine Mitteilung 2 mit einem IParam = 0 (no TextBox hWnd provided to Logger32) und danach eine Mitteilung 3 mit IParam = 1 senden (LOGGER32 soll **DX Spot**-Informationen senden) wird sich LOGGER32 darüber beschweren, eine Warnung will schicken und die Anfrage ableisen.

LOGGER32 wird antworten und diese 3 Dinge tun :

- a) LOGGER32 sendet eine Mitteilung 3 an Ihre Anwendung. IParam dieser Mitteilung ist **hWnd of a TextBox** in LOGGER32 und Ihre Anwendung sendet **ADIF**-formatierten Text
- b) LOGGER32 sendet Ihrer Anwendung eine Mitteilung 100. Der IParam dieser Mitteilung ist die am Transceiver eingestellte Frequenz in Hz.
- c) Falls Ihre Anwendung den Empfang nicht angeforderter Zeichenketten von LOGGER32 bestätigt hat (Sie haben eine **TextBox hWnd** als IParam Ihrer Mitteilung 2 gesendet), sendet LOGGER32 an Ihre Anwendung eine Mitteilung **WM_SETTEXT** mit der am Transceiver eingestellten Sendart in die von Ihnen angegebene **TextBox** . Das Format dieses Textes ist etwa : <APP_RADIO_MODE:3>SSB

Der Verbindungsaufbau und die Synchronisation zwischen LOGGER32 und Ihrer Anwendung ist jetzt hergestellt. LOGGER32 beantwortet die folgenden zusätzlichen Mitteilungen Ihrer Anwendung :

4. IParam ist 0. Ihre Anwendung teilt LOGGER32 mit, dass sie angehalten wurde. LOGGER32 gibt die **RegisteredWindowMessage** frei, die von Ihrer Anwendung verwendet wurde.
5. IParam ist 0. Ihre Anwendung teilt mit, dass LOGGER32 beendet werden soll.
6. IParam ist 0. Ihre Anwendung gibt die PTT-Steuerung frei. LOGGER32 übernimmt die PTT-Steuerung wieder und initialisiert die zugehörige Schnittstelle und Leitung.
7. IParam ist 0. Ihre Anwendung übernimmt die Steuerung der PTT-Leitung.
8. IParam ist 0. Ihre Anwendung fordert LOGGER32 auf, die PTT-Leitung zu aktivieren.
9. IParam ist 0. Ihre Anwendung fordert LOGGER32 auf, die PTT-Leitung zu deaktivieren
10. IParam ist **hWnd of a TextBox** ihrer Anwendung, in die geschrieben werden soll. LOGGER32 antwortet mit einem ADIF-formatierten Text im folgenden Format : <APP_SET_FREQ_MODE:27>18132.012|CW . Die Frequenz wird in kHz angegeben. Der Dezimaltrenner hat das Format der Ländereinstellung Ihres PC.
11. IParam ist 0. Ihre Anwendung teilt LOGGER32 mit, dass es die CAT-Steuerung des Transceivers freigeben soll.
12. IParam ist 0. Ihre Anwendung teilt LOGGER32 mit, dass es die CAT-Steuerung des Transceivers wieder übernehmen soll.
13. IParam ist 0. Ist bei LOGGER32 eingestellt, dass die Sendart von dem Soundkartenprogramm übernommen werden soll, wird mit <APP_FORCE_MODE:x> die Sendart von Ihrer Anwendung bei LOGGER32 eingetragen.
14. IParam ist 0. Die unter Punkt 13 eingestellte Freigabe wird zurückgenommen. Achten Sie darauf, dass Ihre Anwendung diese Freigabe abschaltet, wenn Sie sie ihre Anwendung beenden.
15. Falls IParam = 1 ist, dann wird eine Flagge gesetzt, dass geloggte QSOs für Papier-QSLs vorgemerkt werden. Ist IParam = 0 wird die Flagge abgeschaltet.
16. Falls IParam = 1 ist, dann wird eine Flagge gesetzt, dass geloggte QSOs für eQSLs vorgemerkt werden. Ist IParam = 0 wird die Flagge abgeschaltet
17. Falls IParam = 1 ist, dann wird eine Flagge gesetzt, dass geloggte QSOs für LoTW-QSLs vorgemerkt werden. Ist IParam = 0 wird die Flagge abgeschaltet.

LOGGER32 sendet die folgende ergänzende Mitteilung an Ihre Anwendung :

100 IParam enthält die Transceiverfrequenz in Hz. Diese Mitteilung wird immer dann gesendet, wenn sich die Transceiverfrequenz ändert.

Mehr ist bisher nicht programmiert > fragen Sie nach, wenn Sie weitere Mitteilungen benötigen.

Die Daten zwischen LOGGER32 und Ihrer Anwendung werden mit Hilfe von ADIF-formatiertem Text ausgetauscht, der in die definierten **TextBoxes** geschrieben wird. Es wird empfohlen, die Funktion **TextBox_Change event** als Trigger zu verwenden, um die empfangenen ADIF-Daten zu verarbeiten.

LOGGER32 erkennt diese ADIF-Zeichenketten :

<EOR> gelöscht	Wenn allein empfangen, werden die Einträge im Logeingabefenster
<APP_TAB>	ohne zusätzlichen Text empfangen, wird vom Rufzeichenfeld im Logeingabefenster weitergeschaltet. Automatische Funktionen, wie die Suche nach früheren QSOs oder die Nachfrage bei QRZ.com werden ausgelöst.
<CALL:x>	Das Rufzeichen wird im Logeingabefenster eingetragen
<RST_RCVD:x>	Das empfangene RST wird im Logeingabefenster eingetragen
<RST_SENT:x>	Das gesendete RST wird im Logeingabefenster eingetragen
<NAME:x>	Der Name wird im Logeingabefenster eingetragen
<APP_TIME_ON:x> QSO_DATE und TIME_ON-Feld	der Microsoft-Zeitstempel (Beispiel : 39470.6737384259) setzt das ADIF
<APP_TIME_OFF:x> ADIF TIME_OFF field	der Microsoft-Zeitstempel (Beispiel : 39470.6737384259) setzt das Feld
<APP_QSL:1>Y schaltet sie ab.	setzt die LOGGER32-Papier-QSL-Flagge auf EIN. Ein anderer Buchstabe
<APP_eQSL:1>Y schaltet sie ab.	setzt die LOGGER32-eQSL-Flagge auf EIN. Ein anderer Buchstabe
<APP_LoTW:1>Y schaltet sie ab.	setzt die LOGGER32-LoTW-Flagge auf EIN. Ein anderer Buchstabe
<COMMENT:x>	der Kommentar wird im Logeingabefenster eingetragen
<QTH:x>	das QTH wird im Logeingabefenster eingetragen
<ADDRESS:x>	die Adresse wird im Logeingabefenster eingetragen
<STATE:x>	falls im Logeingabefenster ein STATE-Feld eingerichtet wurde, wird hier die Primary Administrative Subdivision eingetragen
<CNTY:x>	falls im Logeingabefenster ein CNTY-Feld eingerichtet wurde, wird hier die Secondary Administrative Subdivision eingetragen
<GRIDSQUARE:x>	falls im Logeingabefenster ein Feld für den WWLokator (GRIDSQUARE) eingrichtet wurde, wird der WWLokator eingetragen
<IOTA:x>	dito für IOTA-Nummer
<STX:x>	dito für Contestseriennummer STX
<SRX:x>	dito für Contestseriennummer SRX
<QSL_VIA:x>	dito für den QSL-Manager
<SAT_NAME:x>	dito für den Satellitennamen SAT_NAME
<SAT_MODE:x>	dito für den Satellitenmodus SAT_MODE
<PROP_MODE:x>	dito für die Ausbreitungsart PROP_MODE
<FREQ_RX:x>	dito für die Empfangsfrequenz FREQ_RX
<TEN_TEN:x>	dito für die TEN-TEN-Nummer
<USER_1:x>	dito für das Feld USER_1

<USER_2:x>	dito für das Feld USER_2
<USER_3:x>	dito für das Feld USER_3
<APP_LOGQSO:x>	loggt das im Logeingabefenster eingetragene QSO. Dieses Kommando kann allein oder zusammen mit den kompletten QSO-Daten gesendet werden
<FREQ:x>	falls unter LOGGER32 kein Transceiver eingetragen ist, erzeugt dieses Kommando (Daten in kHz) eine Änderung der Frequenzangabe im Logeingabefenster
<MODE:x>	falls die gesendeten Daten nicht mit der im Logeingabefenster stehenden Sendart übereinstimmen, wird eine Warnung erzeugt
<APP_FORCE_MODE:x>	Die Sendart im Logeingabefenster wird geändert. Das Kommando ist vorgesehen, wenn LOGGER32 nicht die Herrschaft über die Transceiversteuerung hat. Hat LOGGER32 die Herrschaft, wird der Eintrag bei der nächsten Transceiverabfrage überschrieben-
<APP_SET_FREQ_MODE:x>	LOGGER32 setzt Frequenz und Sendart auf die empfangenen Daten. Das Format ist z.B. <APP_SET_FREQ_MODE:14>14003.451 CW-R > die Frequenz wird in kHz angegeben !
<APP_CLICK_DXSPOT:x>	Es wird der Klick auf einen DXSpot in LOGGER32 simuliert. Ihre Anwendung muss dazu die Frequenz in kHz und das gespottete Rufzeichen übertragen, z.B.<APP_CLICK_DXSPOT:13>14003.01 K4CY.

LOGGER32 sendet die folgenden ADIF-Zeichenketten an Ihre Anwendung, falls sie unaufgeforderte Mitteilungen zulassen :

<CALL:x> geändert wurde	wird gesendet, wenn das Rufzeichen im Logeingabefenster
<RST_SENT:x> geändert wurde	wird gesendet, wenn das Feld RST_SENT im Logeingabefenster
<RST_RCVD:x> geändert wurde	wird gesendet, wenn das Feld RST_RCVD im Logeingabefenster
<NAME:x> geändert wurde	wird gesendet, wenn das Feld NAME im Logeingabefenster
<MODE:x> geändert wurde	wird gesendet, wenn das Feld MODE im Logeingabefenster
<APP_RADIO_MODE:x>	wird gesendet, wenn die Sendart geändert wurde
<APP_DXSPOT_CALLSIGN:x>	wird gesendet, wenn ein DXSpot empfangen wurde
<APP_DXSPOT_FREQ:x>	wird gesendet, wenn ein DXSpot empfangen wurde
<APP_DXSPOT_BAND:x>	wird gesendet, wenn ein DXSpot empfangen wurde, Das Band wird dem Bandplan entnommen.
<APP_DXSPOT_MODE:x>	wird gesendet, wenn ein DXSpot empfangen wurde, Die Sendart wird dem Bandplan entnommen.
<APP_DXSPOT_COLOR:x>	wird gesendet, wenn der DXSpot hellgetastet wird (neues Land usw.)
<EOR>	wird gesendet, wenn LOGGER32 das Logeingabefenster löscht

Wenn Sie eine Anwendung oder ein Interface für LOGGER32 schreiben, beachten Sie vor allem die Dezimaltrenner (Komma oder Punkt), die von der regionalen Ländereinstellung abhängen. Beachten Sie weiterhin die korrekte Abfolge bei der Verbindungsaufnahme, so dass nicht mehr als eine **RegisteredWindowMessage** unbearbeitet bleibt. Das Programmbeispiel zeigt einen arbeitsfähigen Ablauf.

Achten Sie auch darauf, dass sie alle Ressourcen nach Ende Ihrer Anwendung einschliesslich der **RegisteredWindowMessage** wieder freigeben.

