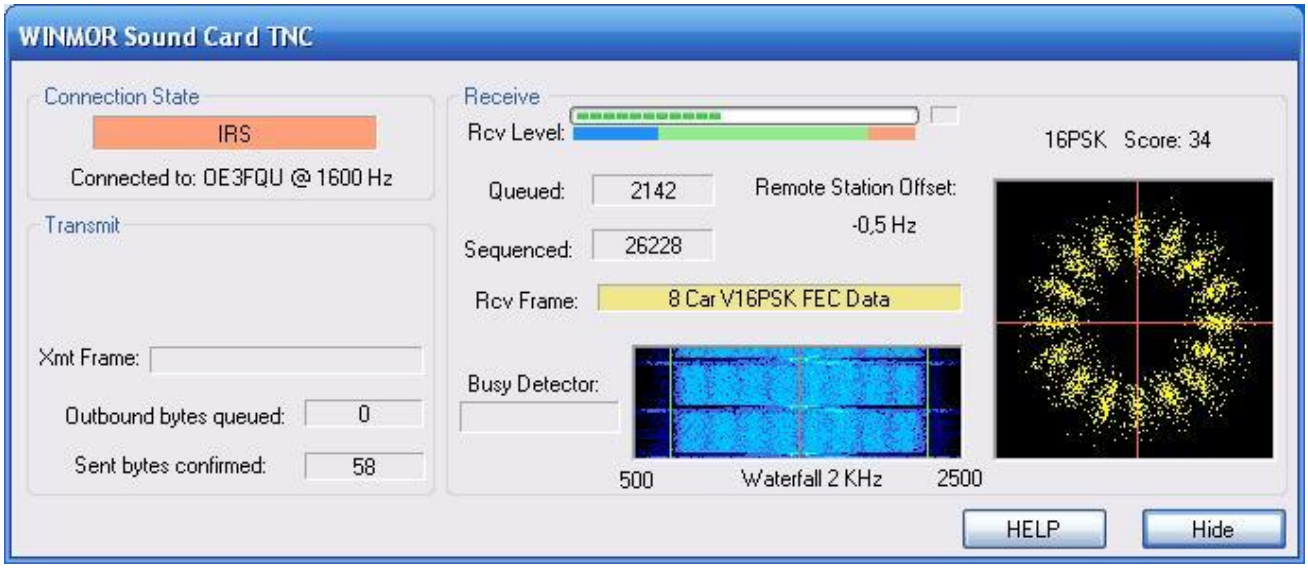
WINMOR

WINMOR (Winlink Message over Radio) - ein Soundkarten TNC im Betatest



WINMOR-Mode im 80m QRM mit 1600 Hz Bandbreite, 8 Träger, 16PSK, FEC - Datendurchsatz etwa 5000 Zeichen pro Minute

Inhaltsverzeichnis

1 Allgemeines 3
2 Protokollbeschreibung 4
2.1 Winmor Timing 6
3 Durchsatzraten 6
4 Software 7
5 Betatest- und Sked-Frequenzen 7

Allgemeines

WINMOR wurde als digitales ARQ Übertragungsprotokoll [1] für die Verwendung mit dem Winlink2000 (WL2K) [2] Netzwerk konzipiert. Mit WINMOR entfällt die Notwendigkeit kostspieliger, externer, PACTOR Modem-Hardware. Allerdings wird WINMOR kaum die Leistungsfähigkeit von PACTOR 3 bez. Datendurchsatz und Betriebssicherheit erreichen können. WINMOR wurde auf der ARRL/TAPR Digital Communications Conference in Chicago, September 26-28, 2008 erstmals präsentiert.

The 27th Annual
ARRL and TAPR Digital Communications Conference
September 26-28, 2008 - Chicago, Illinois

WINMOR…A Sound Card ARQ Mode for Winlink HF Digital Messaging

Rick Muething, KN6KB, AAA9WK
Winlink Development Team
6143 Anchor Lane
Rockledge, FL 32955
rmuething@cfl.rr.com

Abstract:
The improving computational performance of PCs and the near real-time response of PC operating systems now make it feasible to implement reasonable performance HF ARQ messaging protocols suitable for digital messaging. While Pactor (I, II, III) currently dominate and generally represent the best available performance, PC sound cards with appropriate DSP software can now begin to approach Pactor performance at lower cost than dedicated hardware HF modems. This paper covers the on-going development of an optimized sound card mode WINMOR, compatible with the popular Winlink 2000 message system[1,2,3]. This effort leverages a prior feasibility project by the author in the evaluation of SCAMP [4], an adaptation of RDFT for digital messaging systems. The paper reviews the development effort of **WINMOR** (**WIN**link **M**essage **O**ver **R**adio) from motivation through tool development, programming, testing and deployment in the WL2K system.

Key Words:
Winlink 2000, WINMOR, ARQ protocols, multi-carrier PSK, Sound Card Modes, Pactor, SCAMP, HF Channel Simulators

Motivation:
The PC, widely available DSP tools, well designed sound card/radio interfaces and improving amateur software skills have yielded a variety of sound card modes over the last several years. These modes range from simple DSP/software implementation of RTTY through complex streaming applications like Win DRM. It is one of the few remaining areas where amateurs can and do experiment. Many of the modes developed however are a replacement of existing "chat" modes or "broadcast" modes where absolute accuracy is not a requirement or data is limited to plain ASCII text. Today, however, a viable message system (with the need for compression and binary attachments) requires true "error-free" delivery of binary data. To achieve this there must be some "back channel" or *ARQ* (*A*utomatic *R*etry re*Q*uest) so the receiving station can notify the sender of lost or damaged data and request retransmission or repair. HF Pactor (I, II, III) has served us well in this regard providing good performance (net bits/sec/ Hz bandwidth) and robustness. However the proprietary nature of high performance Pactor modems (Pactor II, III) can be cost prohibitive especially in applications such as emergency communications where wide deployment coupled with low average usage make it difficult to justify the investment in high performance but costly hardware. As developers of Winlink 2000 we are continually asked to supply a lower cost of entry than Pactor for those needing to access the WL2K system on HF.

WINMOR ist keine Software, sondern ein Protokoll, es gibt jetzt zwei Programme in der Entwicklung, die dieses Protokoll verwenden:

- **RMS Express** ein Benutzer-Client-Programm
- **RMS Winmor**, ein Radio-Message-Server als Teil des WL2K Systems.

Protokollbeschreibung

Die Protokollbeschreibung (englisch)

WINMOR Protocol Specification (Preliminary)

Revision: 1.0.0 Oct 13, 2009

Rick Muething, KN6KB, AAA9WK

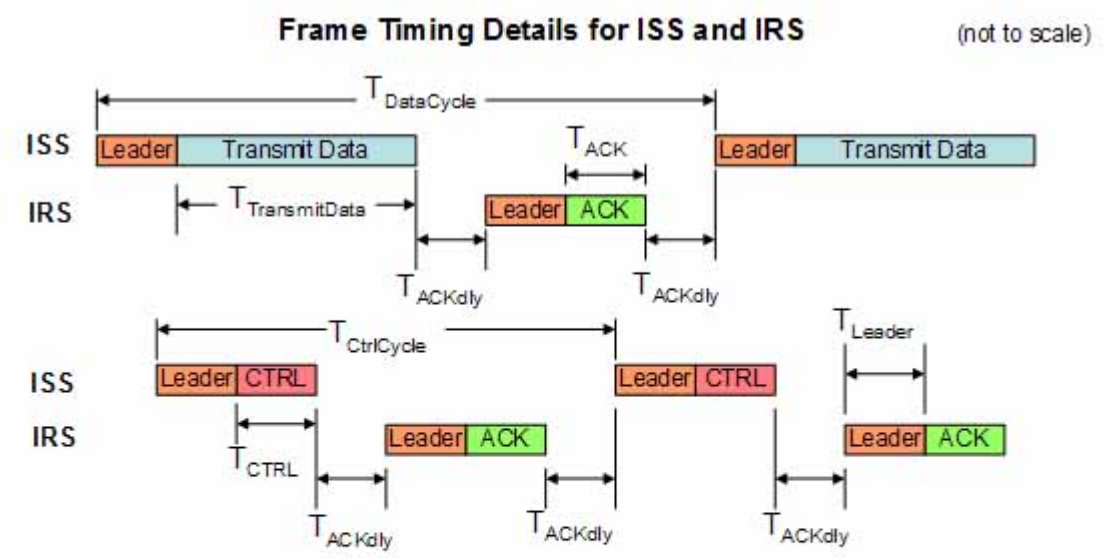
1.0 Scope:
This document describes the preliminary WINMOR sound card protocol at the physical and data link levels. It is the complete specification of the WINMOR protocol. It does not address higher level protocol layers. The WINMOR protocol is not proprietary and is released to the public domain. This document describes the 500 Hz and 1600 Hz bandwidth modes using 93.75 Baud (PSK) and 46.875 baud FSK modulation.

2.0 Purpose:
The intent of this document is two fold:
a) To serve as a working document during protocol development and testing
b) To serve as a template to allow others familiar with the art to build compatible drivers that support the data link protocol layer.

3.0 Definitions and Syntax:
Several specific terms and syntax are used in this document:
Definitions: A term or item is defined using the := symbol. This symbol can be read as "is defined as"
Implementation directives: These are key words that indicate how an item is to be implemented or recommend a method of implementation. They are always indicated by capitalized italic words. These are:
MUST := this must be followed to implement the protocol
MUST NOT := this must not be done to implement the protocol
SHOULD := this is the recommended way to implement the protocol
MAY := this is alternative way to implement the protocol.
The syntax above is always used to distinguish between the common use of the same words.
& is used to indicate catenation. E.g. Frame := Pilot & Data

4.0 Overview of the Protocol:
The WINMOR protocol is intended to be used for sending messages and binary data error free over a HF radio link. It is a Selective Repeat Automatic Retry reQuest (SRARQ) protocol where the Information Receiving Station (IRS) acknowledges receipt of the data to the Information Sending Station (ISS). Normally during a connection session the IRS and ISS exchange roles multiple times. The protocol is designed to handle the type conditions normally encountered in amateur radio transmission.

Winmor Timing



T_{leader} 28 Symbols + optional 12 symbol VOX extenstion (298.6 – 426.6 ms)

T_{ACKdly} 100ms Min, 500 ms max

$T_{DataCycle}$ Repeat interval if NO ACK received >= $T_{Transmit\ Data} + T_{ACK} + 2(T_{Leader} + T_{ACKdly})_{max}$

$T_{CtrlCycle}$ Repeat interval if NO ACK received >= $T_{CTRL} + T_{ACK} + 2(T_{Leader} + T_{ACKdly})_{max}$

Durchsatzraten

Tabelle erzielbarer Daten-Durchsatzraten

'WINMOR Mode Rate Worksheet (200, 500, 1000, 2000Hz B Revised: 9/17/2009 Rick Muething, KN6KB

Mode Description	Info b/Sym	Samp/ Sym	Baud	# of car	~BW (Hz)	Raw bps/Hz	Leader (Sym)	OH/Car (Sym)	PL/Car (sym)	RS-FEC/ Car (sym)	Frame Length (sym)	Frame Length (sec)	ACK (sec)	Rx+Tx +O (sec)	Tot Cycle Len (sec)	Net max Throughput (bits/sec)	(byt/min)	(wrds/min)7
15 Car 16PSK Prag TCM + RS	3	128	62,50	15	2000	1,41	27	16	256	56	355	5,680	1,056	0,3	7,036	1637,3	12280	4093
15 Car 8PSK Prag TCM + RS	2	128	62,50	15	2000	0,94	27	24	256	48	355	5,680	1,056	0,3	7,036	1091,5	8186	2729
15 Car 4PSK Prag TCM + RS	1	128	62,50	15	2000	0,47	27	48	192	80	347	5,552	1,056	0,3	6,908	416,9	3127	1042
15 Car 4FSK +RS	2	256	31,25	15	2000	0,47	13	24	64	64	165	5,280	1,056	0,3	6,636	289,3	2170	723
7 Car 16PSK Prag TCM + RS	3	128	62,50	7	1000	1,31	27	16	256	56	355	5,680	1,056	0,3	7,036	764,1	5731	1910
7 Car 8PSK Prag TCM + RS	2	128	62,50	7	1000	0,88	27	24	256	48	355	5,680	1,056	0,3	7,036	509,4	3820	1273
7 Car 4PSK Prag TCM + RS	1	128	62,50	7	1000	0,44	27	48	192	80	347	5,552	1,056	0,3	6,908	194,6	1459	486
7 Car 4FSK +RS	2	256	31,25	7	1000	0,44	13	24	64	64	165	5,280	1,056	0,3	6,636	135,0	1013	338
3 Car 16PSK Prag TCM + RS	3	128	62,50	3	500	1,13	27	16	256	56	355	5,680	1,056	0,3	7,036	327,5	2456	819
3 Car 8PSK Prag TCM + RS	2	128	62,50	3	500	0,75	27	24	256	48	355	5,680	1,056	0,3	7,036	218,3	1637	546
3 Car 4PSK Prag TCM + RS	1	128	62,50	3	500	0,38	27	48	192	80	347	5,552	1,056	0,3	6,908	83,4	625	208
3 Car 4FSK +RS	2	256	31,25	3	500	0,38	13	24	64	64	165	5,280	1,056	0,3	6,636	57,9	434	145
1 Car 16PSK Prag TCM + RS	3	128	62,50	1	200	0,94	27	16	256	56	355	5,680	1,056	0,3	7,036	109,2	819	273
1 Car 8PSK Prag TCM + RS	2	128	62,50	1	200	0,63	27	24	256	48	355	5,680	1,056	0,3	7,036	72,8	546	182
1 Car 4PSK Prag TCM + RS	1	128	62,50	1	200	0,31	27	48	192	80	347	5,552	1,056	0,3	6,908	27,8	208	69
1 Car 4FSK +RS	2	256	31,25	1	200	0,31	13	24	64	64	165	5,280	1,056	0,3	6,636	19,3	145	48
1 Car Connect Request (V4PSK) +RS	1	128	62,50	1	200	0,31	27	0	104	32	163	2,608						
1 Car Coded Control 4FSK + RS	2	256	31,25	1	200	0,31	13	0	8	8	29	0,928						
1 Car ACK 4FSK +RS	2	256	31,25	1	200	0,31	13	0	12	8	33	1,056						
Leader extension (symbols 0-16)	0	See Note 2																
Calculated Leader extension (ms)	0																	

Notes: 1) The 4FSK mode was modified for lower user data and higher RS FEC and now runs slower than the 4PSK mode.
2) Leader extension up to 16 symbols (256 ms) may be used for slow switchover Trancievers or VOX operated PTT
3) The above modes yield the following speed ranges depending on session bandwidth:
note these selections may change based on testing!
2000 Hz BW Sessions: 15x16PSK, 15x8PSK, 15x4PSK, 15x4FSK, 7x4FSK ~ 12:1 speed range
1000 Hz BW Sessions: 7x16PSK, 7x8PSK, 7x4PSK, 7x4FSK, 3x4FSK ~ 13:1 speed range
500 Hz BW Sessions: 3x16PSK, 3x8PSK, 3x4PSK, 3x4FSK,1x4FSK ~ 18:1 speed range
200 Hz BW Sessions: 1x16PSK, 1x8PSK, 1x4PSK , 1x4FSK ~ 6:1 speed range
4) Session BW is set by Server (answering) station using one of 4 coded Idle frames (200, 500, 1000, or 2000 Hz)
5) All PSK modes use pragmatic Trellis Code Modulation (one redundancy bit/symbol) and use the standard R=1/2, K=7
(NASA Voyager) Viterbi Encoder/Decoder based on Phil Karns Code.
6) Rx + Tx + O refers to the receive to transmit, transmit to receive and software overhead delays and is typical for modern hardware.
The protocol acutally measures the latency due to RX>TX switchover, sound card and CPU processing latency.
7) Word per minute calculation based on average word of 5 char + space and a 50% compression ratio (typical using B2 compression on
mid to large messages). Short messages will be less.

Software

Wer beim Betatest mitmachen möchte, Englischkenntnisse sind vorausgesetzt, muss sich im Yahoo Forum [3] anmelden.

Voraussetzungen: WIN-XP oder -Vista, .NET3.5, höherwertige Soundkarte z.B. [4], KW-Transceiver

WINMOR unterstützt direkt den USB-Soundkartentreiber (USB Audio Codec) für die ICOM IC-7200 und IC-7600 KW-Transceiver.

Die Software RMSExpress mit den Betriebsarten WINMOR, Pactor, Packet, Telnet wird anlässlich der HAMVENTION im Mai 2010 freigegeben.

Betatest- und Sked-Frequenzen

RMS WINMOR (Server) sind im Betatest aktiv. Die aktiven RMS Winmor Stationen sind hier [5] zu sehen.

Peer-to- Peer Skeds können hier [6] vereinbart werden.

MHeard Log (@ OE3ZK)

2010.01.26 20:45:27 UA6DX-5 41 (KN95NA)

2010.01.26 20:05:16 UR5VIB 69

2010.01.26 20:04:56 ON4KCY 64 (JO21IA)

2010.01.26 19:30:19 DL8RCB-6 51 (JN68RS)

2010.01.26 19:10:55 OH8GKP 52

2010.01.26 19:05:07 OH7JJT 44

2010.01.26 19:04:34 RU3AT 60 (KO85VT)

2010.01.26 18:52:39 OH5RM 51 (KP30JR)

2010.01.26 16:23:00 PD4U-5 80

2010.01.26 15:56:37 F5FJ 39

2010.01.26 15:12:41 I3FUE 81 ()

2010.01.26 11:24:45 PY2LIN 44

2010.01.26 11:17:25 EA2AFR 85

2010.01.26 11:11:18 OH1IU 90 (KP01XK)

2010.01.25 14:11:13 SM6ESQ 58 (JO67KI)

2010.01.21 13:18:54 F1AFZ 26

2010.01.20 16:09:25 OE3ZK-3 98 (JN88DH)

2010.01.17 14:31:42 OE1KBC 76 (JN88EG)

2010.01.15 09:53:08 OE3FQU 38 (JN78TE)

2010.01.06 12:59:35 VE1YZ-5 36 (FN84BQ)

2009.12.27 18:41:58 DG6ABK 82

2009.12.27 18:01:41 IW7DGY 69 (JN81KC)

2009.12.21 13:17:35 KD4XV 28 (EM75EB)

2009.12.21 13:15:44 LZ1OE 42 (KN32QM)

2009.12.21 12:56:25 PD4U 28 (JO22XE)

2009.12.21 12:55:26 OH7TE 74 (KP20IE)

2009.12.20 10:01:49 OH6XI 58 (KP13IQ)

2009.12.18 10:31:17 OE5RPP 52 (JN67RW)

2009.12.13 12:20:22 LA3RIA 0 (JO59HW)

2009.12.13 11:53:58 OH3NJC 85 (KP11OP)

2009.12.08 12:04:11 9Z4BM 72 (FK90GG)

2009.12.07 14:31:54 V31AE 32 (EK58UK)

2009.12.07 11:30:53 F6DKQ 64

2009.12.05 16:12:58 ON4MS 56 (JO20KR)

2009.12.04 15:20:21 WA4BDS 30 (EM85AQ)

2009.12.04 13:23:40 KB2BHW 0 (FN13VA)

2009.11.20 13:07:16 LZ1ZM 77

2009.11.19 12:12:25 KB1LIH 35

2009.11.17 14:59:18 W1EO 38

2009.11.17 14:39:11 W4NTG 59

2009.11.16 20:33:39 OE9FWV 79

2009.11.16 11:57:00 PA7HPH 30

2009.11.14 11:28:05 M6DJE 86 (IO81RM)

2009.11.13 14:02:30 KC9LET 24 (EM69XX)

2009.11.12 11:55:17 C56DL 38

2009.11.08 14:28:54 NY3J 70

2009.10.29 17:30:56 DL1ZG 30

2009.10.29 17:26:15 DL0NOT 66

2009.10.24 17:58:00 IZ1GCL 67

2009.10.23 15:30:28 KB0E 28

2009.10.22 16:38:07 W9IB 44 (EN62AC)

2009.10.22 14:24:50 ON5VW 27

2009.10.20 15:50:21 AFA7HZ 29

2009.10.20 15:14:39 GM8BPQ 32 (IO68VL)

2009.10.20 15:06:28 W0RLI 35

2009.10.19 14:09:13 W4BGH 35 (EL87VT)

2009.10.19 13:55:17 WS5H 29 (EL29FQ)

2009.10.19 13:44:48 W5SEG 56 (EL19AN)

2009.10.19 13:44:25 K4JCC 30

2009.10.15 18:16:03 KA2UOL

2009.10.15 18:13:34 AK5U

2009.10.15 17:52:36 K0TER

2009.10.15 16:38:04 WB2HVF

2009.10.15 16:03:22 K4DMU

2009.10.14 16:51:24 VE7CRT

2009.10.14 15:52:40 AA6FT

2009.10.13 16:34:56 DB7EM

2009.10.13 14:14:27 KC4QYM

2009.10.12 17:26:08 VE3LKI

2009.10.12 16:04:38 LA5VNA

2009.10.12 16:02:05 VE3IRI

2009.10.11 11:10:30 AD4RG

2009.10.10 16:32:52 VE2HAR

2009.10.08 18:36:56 K2MO

2009.10.06 19:28:10 KB3CS

2009.10.06 18:54:15 N2UBP

2009.10.06 18:53:37 9Y4LT

2009.10.06 18:22:02 K4PPK

2009.10.06 16:23:40 EA5GF

2009.10.05 17:58:52 KD4NUE

2009.10.05 16:31:11 KR4XN

2009.10.04 17:51:26 PA0LSK

2009.10.04 17:34:09 IZ3LEF

2009.10.03 15:49:00 VE2AEY

2009.10.03 15:26:10 VA3LKI

2009.10.03 13:37:45 N1CPE

2009.10.03 09:41:43 S57MK

2009.10.01 18:12:13 LZ1CWK