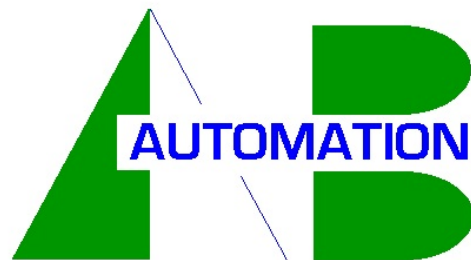


# WOW!<sup>TM</sup> Protocol

Version 1.2



WOW101-SP-01 - Rev. D  
December, 2009

**Able-Baker Automation**  
P.O. Box 6368  
Moraga, California 94570  
1-877-444-2253  
1-510-601-9396

# Revision Notes

---

Able-Baker Automation revisions:

Rev.	Date	Description	Done	Chk'd	App'd
A	Apr 14, 2000	Release - Version 1.0	WCB		
B	Apr 30, 2000	Release - Version 1.1 Added Expanded Response Format	WCB		
C	May 04, 2005	Release - Version 1.2 Added Data Transmission Format	WCB		
D	Dec 10, 2009	Updated Address	WCB		

AUTOMATION

## I. **PROTOCOL**

1. Purpose
2. Specifications
3. Framing
4. Translation Table
5. Parsing Frames
6. Unsolicited Responses

## 1. Purpose

- 1.1 The purpose of the Able-Baker **WOW!**<sup>™</sup> Protocol is to provide a simple method for two devices to share status information. It is not suitable for transferring large amounts of data. It is designed to be easy to implement and test using the Microsoft Windows HyperTerminal program. This is an open protocol.

## 2. Specifications

- 2.1 The Able-Baker **WOW!**<sup>™</sup> Protocol is suitable for any point to point network hardware system. It was primarily developed for RS-232C serial communication ports.
- 2.2 The Able-Baker **WOW!**<sup>™</sup> Protocol is set up to be used in a Master-Slave format. However, unsolicited responses from the Slave unit are supported.
- 2.3 The Able-Baker **WOW!**<sup>™</sup> Protocol is an ASCII protocol. The protocol can be used at any Baud rate with or without hardware parity checking and with or without flow control.
- 2.4 If the Able-Baker **WOW!**<sup>™</sup> Protocol is used with software flow control the XON (DC1 - ASCII 17<sub>10</sub>) and XOFF (DC3 - ASCII 19<sub>10</sub>) characters must be removed from the string before (or during) parsing. This may be implemented in either hardware or software.
- 2.5 The Able-Baker **WOW!**<sup>™</sup> Protocol uses 100% redundancy. Each message command is transmitted twice in a single frame to insure data validity. The Expanded Response format is an exception to this requirement. The expanded response format is intended for non-critical information. (If error checking is desired the query may be transmitted twice and the results compared.)

### 3. Framing

- 3.1 The Able-Baker **WOW!**<sup>TM</sup> Protocol is made up of a normal frame of four ASCII characters or an expanded response frame of eight characters.
- 3.2 The normal frame has a Start Character, two message characters, and a termination character.
  - 3.2.1 Start Character - Always "!" (exclamation point ASCII - 33<sub>10</sub>)
  - 3.2.2 Message Characters - Any character from the translation table repeated twice. (For example "AA")
  - 3.2.3 Termination Character - Always Carriage Return (ASCII - 13<sub>10</sub>)
- 3.3 The expanded response frame has a Start Character, an expanded response format character, three message characters, and a termination character.
  - 3.3.1 Start Character - Always "!" (exclamation point ASCII - 33<sub>10</sub>)
  - 3.3.2 Expanded Response Format Character - Always "." (period ASCII - 46<sub>10</sub>)
  - 3.3.3 Message Characters - Any three alpha-numeric characters. Non alpha numeric characters are not allowed. All three characters must be transmitted.
  - 3.3.4 Termination Character - Always Carriage Return (ASCII - 13<sub>10</sub>)
  - 3.3.5 Optional characters may be appended to the frame. These will be ignored by the protocol but may be helpful in a specific application. (For example appending a line feed character.)
- 3.4 The data transmission format is an extension of the expanded response format. It should be used to transmit information from one device to another. The receiving device should be capable of parsing large strings. The frame is similar to the expanded response frame.
  - 3.4.1 Start Character - Always "!" (exclamation point ASCII - 33<sub>10</sub>)
  - 3.4.2 Expanded Response Format Character - Always "." (period ASCII - 46<sub>10</sub>)
  - 3.4.3 Message Characters - Any three alpha-numeric characters. Non alpha numeric characters are not allowed. All three characters must be transmitted. These characters will determine if additional

data is included.

- 3.4.4 Field Separator - Always “,” (comma ASCII - 44<sub>10</sub>)
- 3.4.5 First Field - ASCII Text as required followed by a Field Separator (if more fields are present) or a termination character. The Field may be null. (Empty)
- 3.4.6 Termination Character - Always Carriage Return (ASCII - 13<sub>10</sub>)
- 3.4.7 Optional characters may be appended to the frame. These will be ignored by the protocol but may be helpful in a specific application. (For example appending a line feed character.)



## 4. Translation Table

- 4.1 The message characters are limited to printable ASCII characters as listed in the translation table. Each message character can be assigned to either the Master or the Slave unit (but not both).
- 4.2 A message character may represent a Query, an Answer, or a Command.
- 4.3 ASCII Characters not in the translation table are not allowed to be used as message characters.
- 4.4 This protocol uses only printable ASCII characters as message characters.

## 5. Parsing Frames

- 5.1 The input buffer should be read and characters flushed until a Start Character "!" (ASCII - 33<sub>10</sub>) is read.
- 5.2 If the next character is a not a "." (ASCII - 46<sub>10</sub>) then the next two characters must be identical. If they match a valid message the receiver should execute the command. If the characters are not identical or the message is not valid these characters should be flushed and the parser should start looking for the next start character.
- 5.3 If the next character is a "." (ASCII - 46<sub>10</sub>) and the system is expecting an expanded response and the next three characters match a valid response the receiver should execute the command. If the message is not valid these characters should be flushed and the parser should start looking for the next start character.
- 5.4 If the next character is a "." (ASCII - 46<sub>10</sub>) and the system is expecting an expanded response and the next three characters match a valid response and this response indicates that this is a data transmission format the receiver should parse the remainder of the string. The receiver should then use this data to update the fields in it's program. If the message is not valid these characters should be flushed and the parser should start looking for the next start character.

## 6. Unsolicited Response from the Slave Unit

- 6.1 A Message command may be sent from the master unit to the slave to allow the slave to send any unsolicited message. The slave will be able to

send an unsolicited message at any time until the master sends any other message. The ability to send an unsolicited message will then be canceled until the master resends the message command to allow unsolicited messages.

- 6.2 When allowed to send unsolicited messages the slave will send the message immediately. This will typically be used to report alarms to the Master unit.

## 7. Registration

- 7.1 The Able-Baker **WOW!**<sup>™</sup> Protocol is an open protocol available for any suitable use. (We accept no liability for any use of this protocol.) Voluntary registration is requested to allow us to advise users of specification updates. Send registration information to:

Able-Baker Automation, Inc.  
P.O. Box 6368  
Moraga, California 94570  
1-877-444-ABLE  
1-510-601-9396

AUTOMATION



Able-Baker WOW!™ Protocol Translation Table

Character	ASCII Code (Base 10)	Source Master/Slave	Message
#	35		
\$	36		
%	37		
&	38		
(	40		
)	41		
*	42		
+	43		
-	45		
/	47		
0	48		
1	49		
2	50		
3	51		
4	52		
5	53		
6	54		
7	55		
8	56		
9	57		
:	58		
;	59		
<	60		
=	61		
>	62		

Able-Baker **WOW!**<sup>TM</sup> Protocol Translation Table

Char-acter	ASCII Code (Base 10)	Source Master/Slave	Message
?	63		
@	64		
A	65		
B	66		
C	67		
D	68		
E	69		
F	70		
G	71		
H	72		
I	73		
J	74		
K	75		
L	76		
M	77		
N	78		
O	79		
P	80		
Q	81		
R	82		
S	83		
T	84		
U	85		
V	86		
W	87		
X	88		

Able-Baker **WOW!**<sup>TM</sup> Protocol Translation Table

Character	ASCII Code (Base 10)	Source Master/Slave	Message
Y	89		
Z	90		
[	91		
\	92		
]	93		
^	94		
_	95		
a	97		
b	98		
c	99		
d	100		
e	101		
f	102		
g	103		
h	104		
i	105		
j	106		
k	107		
l	108		
m	109		
n	110		
o	111		
p	112		
q	113		
r	114		
s	115		

Able-Baker WOW!™ Protocol Translation Table			
Char-acter	ASCII Code (Base 10)	Source Master/Slave	Message
t	116		
u	117		
v	118		
w	119		
x	120		
y	121		
z	122		
{	123		
	124		
}	125		
~	126		
Expanded Responses			
.	46	Slave	Response to Query “___” Acceptable Responses:
			Response to Query “___” Acceptable Responses:
			Response to Query “___” Acceptable Responses:
			Response to Query “___” Acceptable Responses:
Data Transmission Format			
.	46		XXX, (Identification for this string)

Able-Baker WOW!™ Protocol Translation Table			
Character	ASCII Code (Base 10)	Source Master/Slave	Message
			XXX, (First Field, Define format and ranges and acceptable characters.)
			XXX, (Subsequent Fields)

