

# **AFRecorder 4800A/R Serial Port Programming Interface Description**

## **For Software Versions 8.6 through 8.9**

### **(Last Revision 1-10-12)**

#### **Background**

The AFRecorder 4800 may be operated under remote control using a standard RS232 serial port. The only function which is not available using the serial port is the "AIR CAL" function. Available remote operating modes include two types of real-time data upload. For examples of remote control operation, try using the AFRemote (MS-DOS PC) software included with the AFRecorder.

The AFRecorder may not be operated both under remote control and "local" control at the same time. If AFRecorder operation is attempted using the front panel keypad while under remote control, the remote control operation will be terminated. Remote control commands are available to enable and disable the front panel keypad in order to avoid accidental termination of remote operation. To initiate remote control of the AFRecorder, a command is issued over the serial port. No operations using the front panel are necessary. The external computer may take control of the AFRecorder at any time except during initialization. The serial port parameters are 4800 baud, no parity, 8 data bits, 1 stop bit.

The AFRecorder stores setup and calibration data in an Electrically Erasable Programmable Read Only Memory (EEPROM). EEPROM values may be altered by the external computer, one at a time, using simple command strings along with the new value. New EEPROM values downloaded from the external computer do not affect the operation of the AFRecorder until an initialization is performed. An initialization is performed during power-on, if real-time data upload is requested, or if a recording session is started.

Data in the EEPROM is stored either as one byte "selections" or as four byte floating point "constants". An example of a "selection" is the choice of AFR units (1 = AFR, 2 = Phi, 3 = Lambda, 4 = %O<sub>2</sub>). The "constants" are values such as the sensor calibration constants. Changing these parameters using the external computer is equivalent to changing them using the front panel keypad.

The serial port remote control code in the AFRecorder was designed to be used with a PC as the external computer. A PC (Intel) stores multi-byte data types with the low byte first. The AFRecorder (Motorola) stores multi-byte data types with the high byte first. In order to make interfacing to a PC easier, the AFRecorder accepts multi-byte data included in download commands in Intel format. However, data uploaded from the AFRecorder to an external computer is in Motorola format. Therefore, a two byte checksum or a floating point number sent by the AFRecorder to a PC must undergo a byte reversal before use as shown in the code fragment below. In most applications, a serial port interrupt routine will be used to create a buffer of received characters. The data bytes may then be taken from the buffer and placed in the union variable in reversed order directly, so as to avoid the need for a "reversing" operation.

```

/* Code showing translation of an AFRecorder floating point data value to PC format */

typedef union {
    float fl;
    unsigned char uc[4];
} FloatData;

FloatData afrfloat;

unsigned char uchar;
.
.
.
/* Assume that a floating point data value from the AFRecorder is contained in "afrfloat" */

uchar = afrfloat.uc[0];
afrfloat.uc[0] = afrfloat.uc[3];
afrfloat.uc[3] = uchar;
uchar = afrfloat.uc[1];
afrfloat.uc[1] = afrfloat.uc[2];
afrfloat.uc[2] = uchar;
.
.
.
/* Use the data */

```

## Serial Port Commands

When the AFRecorder is being operated under remote control, it is said to be "connected". When "connected", the AFRecorder LCD displays will either show the AFR data (while in measure mode) or serial link information. Some commands, such as the status command, do not require "connection".

Many of the commands are acknowledged by the AFRecorder with a two byte code indicating that the command was correctly received. Some commands are not acknowledged since performance of the requested function may be used as an acknowledgement. For example, the command to start the uploading of real-time data is "acknowledged" by the presence or absence of the real-time data.

There are three serial port command formats: "Control", "Change Selection", and "Change Value".

## Control Commands

A "Control" command to the AFRecorder consists of a header and the command number sent twice. A total of 7 bytes must be sent. There is no checksum sent with this type of command. Some Control commands require prior "connection" to the PC while others do not. The Control command format is:

<C8> <C8> <C8> <C8> <D9> <command> <command>

For example, command number 2, the "connect" command (initiate remote control of the AFRecorder), would be sent as:

<C8> <C8> <C8> <C8> <D9> <2> <2>.

The valid Control command numbers and their functions are as follows:

- 1: Upload status byte. Returns <status byte> <status byte>:  
    <A0> <A0> - Initializing  
    <A1> <A1> - Sensor warm-up period  
    <A2> <A2> - Measure mode  
    <A5> <A5> - Remote mode  
    <A6> <A6> - Recording
- 2: Start remote operation ("connect" command). This command is acknowledged by the AFRecorder sending <D0> <D0> (two bytes). This command should be issued first, before any other commands are used.
- 3: Disable keyboard operation. The keyboard will remain disabled until either Control command 4 or Control command 6 is received or the AFRecorder is switched off then on. This command is not acknowledged.
- 4: Enable keyboard operation. This command is not acknowledged.
- 6: Hard reset of AFRecorder (watchdog reset). This command is not acknowledged.

The following commands require prior "connection" (Control command 2):

- 7: Terminate remote operation and return to local mode ("disconnect" command). This command is acknowledged by the AFRecorder sending <D0> <D0> (two bytes).
- 8: Upload all current AFRecorder EEPROM selections. There are a total of 17 individual one byte selection values in the EEPROM. The 17 bytes of data are uploaded followed by a two byte (16 bit) checksum, for a total of 19 bytes. A listing of the function of the relevant EEPROM values is given below in the "Change Selection Command" section. Remember that the checksum value may have to be reversed before use.
- 9: Upload all current AFRecorder EEPROM constants. There are 77 individual four byte floating point constant values in the EEPROM. The 308 bytes of data are uploaded followed by a two byte (16 bit) checksum, for a total of 310 bytes. An array of the 77 uploaded constants (with the array index starting at zero) may be matched to the descriptions listed in the "Change Value Command" section. The uploaded array index would be the value given in the "Change Value Command" Section - 16. For example, the uploaded H:C value would be located at uploaded array index 43 (59-16). Remember that the floating point data and checksum values may have to be reversed before use.
- 17: Set real-time data upload mode. This command prepares the AFRecorder for real-time mode. In real-time mode, a value is uploaded for both sensors even if one of the sensors is not enabled. The data is sent as <C8> <C8> <C8> <C8> <Left Channel AFR value> <Right Channel AFR value> <Left Channel %O<sub>2</sub> value> <Right Channel %O<sub>2</sub> value> <checksum byte 1> <checksum byte 2>. The sensor data is uploaded in four byte floating point number format. Remember that the data and checksum values may have to be reversed before use.

If the data for one of the sensors is invalid, one of the limit values is uploaded (see the instruction manual for limit values). This command prepares the AFRecorder for real-time data upload but does not start the data transfer. **Use Control command 19 to start the data transfer.**

There are two types of real-time data upload modes: "standard" and "fast". In standard mode, the AFR data is averaged in the same way as the front panel display. In fast mode, the AFR data is not averaged. The range of allowed data upload rates is from 0.24 to 60.0 seconds in standard mode, and 0.10 to 60.0 seconds in fast mode. Fast mode is selected by setting the "fast real-time" flag (Control command 21). The data rate is set using the "Change Value" command described below. For upload rates faster than 0.24 seconds, the AFR displays are disabled during real-time operation.

- 18: Halt real-time upload and return to idle state. This command is acknowledged by the AFRecorder sending <D0> <D0> (two bytes) to the PC.
- 19: Set real-time allowed flag (start/restart real-time data upload). Commands 19 and 20 may be used to temporarily suspend and restart real-time data upload without affecting other AFRecorder functions. These functions may be used to avoid buffer overflow problems in data acquisition or display programs. This command is not acknowledged.
- 20: Clear real-time allowed flag. This command temporarily suspends real-time data upload. This command is not acknowledged.
- 21: Set fast real-time flag. This command allows variable data upload rates from 0.10 seconds to 60.0 seconds. The data uploaded in this mode is not averaged, but is a "snapshot" of the AFR value at the time of data upload. For use, see Control command 17. For upload rates faster than 0.24 seconds, the AFR displays are disabled during real-time operation. This command is not acknowledged.
- 22: Clear fast real-time flag. This command is not acknowledged.
- 23: Reset AFRecorder. Equivalent to pressing the "SYS" key on the front panel during stand-alone operation. Will terminate a recording session. This command is not acknowledged.

The following commands relate to recording functions:

- 12: Start a recording session. This command is equivalent to placing the unit in measure mode and then pressing the "REC" button on the front panel. This command is acknowledged by the AFRecorder sending <D0> <D0> (two bytes) to the PC.
- 13: Upload recorded data units used in the last recording session. The response to this command is one floating point number followed by a two byte checksum. Responses are 1 for AFR units, 2 for Phi, 3 for Lambda and 4 for %O<sub>2</sub>.
- 14: Upload recorded data interval used in the last recording session. The response to this command is one floating point number followed by a two byte checksum.
- 15: Upload number of recorded data points. The value is a single long (32bit) integer followed by a two byte checksum. If there is no recorded data, the AFRecorder will return a value of zero.

24: Upload recorded data channels choice. The response to this command is a single long (32bit) integer followed by a two byte checksum. Responses are 1 for left channel only, 2 for right channel only, and 3 for both.

16: Upload recorded AFR data. The data is sent in floating point format as:

<all Left Channel AFR data> <all Right Channel AFR data>  
<checksum byte 1> <checksum byte 2>.

If a sensor was not enabled, the data corresponding to that sensor will be missing, so that the entire buffer (5000 points) may be used for the other sensor. The total number of points recorded (sum from both sensors) is available using command 15.

### Change Selection Command

A "Change Selection" command to the AFRecorder consists of a header, an index, and the one byte value sent twice. A total of 8 bytes must be sent. There is no checksum sent with this type of command. The index is an array index, **starting at zero**, in the 17 byte selection list in the EEPROM. Any indices not listed below should never be changed by the external computer. The Change Selection command requires prior "connection" to the PC. Values successfully received by the AFRecorder are acknowledged by <D0> <D0>. A checksum failure is reported by <D1> <D1>.

The "Change Selection" command format is:

<C8> <C8> <C8> <C8> <DA> <index> <selection> <selection>

The valid index values and selections are:

Index	Function	Valid Values
3	AFR Display units	1=AFR, 2=Phi, 3=Lambda, 4=%O <sub>2</sub>
4	AFR Analog output units	1=AFR, 2=Phi, 3=Lambda, 4=%O <sub>2</sub>
5	Simulated EGO units	1=AFR, 2=Phi, 3=Lambda
8	Left display size	1=Small, 2=Large
9	Right display size	1=Small, 2=Large
10	Display update rate	1=Slow, 2=Med, 3=Fast
11	Left large display type	1=AFR, 2=DEV, 3=DIF, 4=AVG
12	Right large display type	1=AFR, 2=DEV, 3=DIF, 4=AVG
13	ICC Left	0=Disabled, 1=Enabled
14	ICC Right	0=Disabled, 1=Enabled
16	Key beeping enable	0=Disabled, 1=Enabled

## Change Value Command

A "Change Value" command to the AFRecorder consists of a header, an index, the four byte floating point value and a two byte checksum. The checksum is calculated from the four-byte floating point value and does not include the index or the header. A total of 12 bytes must be sent. **Remember to present the floating point and checksum values to the AFRecorder in Intel format.** Any indices not listed below should never be changed by the external computer. The Change Value command requires prior "connection" to the PC. Values successfully received by the AFRecorder are acknowledged by <D0> <D0>. A checksum failure is reported by <D1> <D1>.

The "Change Value" command format is:

<C8> <C8> <C8> <C8> <DB> <index> <value 1> <value 2> <value 3> <value 4>  
<checksum 1> <checksum 2>.

The valid index values and allowed ranges for the data are:

Index	Function	Valid Values
<b>Real-time Data Rate</b>		
68	Real-time data upload rate	0.10 to 60.0 sec.
<b>Fuel Constants</b>		
59	Fuel H:C	1.0 to 10.0
60	Fuel O:C	0.0 to 1.0
61	Fuel N:C	0.0 to 1.0
<b>Simulated EGO Output Threshold Levels</b>		
45	Left simulated EGO (AFR)	0.00 to 400.00
46	Left simulated EGO (Phi)	0.00 to 10.00
47	Left simulated EGO (Lambda)	0.00 to 10.00
48	Right simulated EGO (AFR)	0.00 to 400.00
49	Right simulated EGO (Phi)	0.00 to 10.00
50	Right simulated EGO (Lambda)	0.00 to 10.00
<b>Recording Functions</b>		
69	Recording period (interval)	0.02 to 60 sec
70	Recording duration (minutes)	0 to 5000
71	Recording duration (seconds)	0 to 1000
<b>AFR Sensor Calibration Constants</b>		
83	Left sensor I <sub>1</sub>	0.10 to 5.00
84	Left sensor I <sub>O2</sub>	0.01 to 1.00
85	Left sensor I <sub>CO</sub>	0.01 to 1.00
86	Left sensor I <sub>H2</sub>	0.01 to 1.00
87	Left sensor I <sub>2</sub>	-1.00 to 1.00
88	Right sensor I <sub>1</sub>	0.10 to 5.00
89	Right sensor I <sub>O2</sub>	0.01 to 1.00
90	Right sensor I <sub>CO</sub>	0.01 to 1.00

91	Right sensor $I_{H2}$	0.01 to 1.00
92	Right sensor $I_2$	-1.00 to 1.00
<b>Engine Calibration Constants</b>		
62	Left sensor ICC lean	-10.00 to 10.00
63	Left sensor ICC stoic	-10.00 to 10.00
64	Left sensor ICC rich	-10.00 to 10.00
65	Right sensor ICC lean	-10.00 to 10.00
66	Right sensor ICC stoic	-10.00 to 10.00
67	Right sensor ICC rich	-10.00 to 10.00
<b>Analog Output Constants</b>		
17	Left Channel at 0.0V (AFR)	0.00 to 400.00
18	Left Channel at 0.0V (Phi)	0.00 to 10.00
19	Left Channel at 0.0V (Lambda)	0.00 to 10.00
20	Left Channel at 0.0V (%O <sub>2</sub> )	0.00 to 100.00
21	Left Channel %O <sub>2</sub> at 0.0V	0.00 to 100.00 (Version 8.6 only)
22	Left Channel at 5.0V (AFR)	0.00 to 400.00
23	Left Channel at 5.0V (Phi)	0.00 to 10.00
24	Left Channel at 5.0V (Lambda)	0.00 to 10.00
25	Left Channel at 5.0V (%O <sub>2</sub> )	0.00 to 100.00
26	Left Channel %O <sub>2</sub> at 5.0V	0.00 to 100.00 (Version 8.6 only)
27	Right Channel at 0.0V (AFR)	0.00 to 400.00
28	Right Channel at 0.0V (Phi)	0.00 to 10.00
29	Right Channel at 0.0V (Lambda)	0.00 to 10.00
30	Right Channel at 0.0V (%O <sub>2</sub> )	0.00 to 100.00
31	Right Channel %O <sub>2</sub> at 0.0V	0.00 to 100.00 (Version 8.6 only)
32	Right Channel at 5.0V (AFR)	0.00 to 400.00
33	Right Channel at 5.0V (Phi)	0.00 to 10.00
34	Right Channel at 5.0V (Lambda)	0.00 to 10.00
35	Right Channel at 5.0V (%O <sub>2</sub> )	0.00 to 100.00
36	Right Channel %O <sub>2</sub> at 5.0V	0.00 to 100.00 (Version 8.6 only)
<b>AFR Offset Values</b>		
51	Left Channel offset (AFR)	-2.00 to 2.00
52	Left Channel offset (Phi)	-0.20 to 0.20
53	Left Channel offset (Lambda)	-0.20 to 0.20
54	Left Channel offset (%O <sub>2</sub> )	-2.00 to 2.00
55	Right Channel offset (AFR)	-2.00 to 2.00
56	Right Channel offset (Phi)	-0.20 to 0.20
57	Right Channel offset (Lambda)	-0.20 to 0.20
58	Right Channel offset (%O <sub>2</sub> )	-2.00 to 2.00