AT-200PC Serial Control Protocol Specification
Revision 1.7

LDG Electronics, Inc.

March 27, 2006

Contents

List of Tables

List of Figures

1. AT-200PC PC-Controlled Tuner Overview

1.1 Document Status

This document reflects the firmware version 1.7 of the AT200PC. Note: Changes to this document from revision 1.3 are noted by underlined, blue text, like this. ~~Text removed from the previous document is redlined, like this.~~

1.2 Brief Description of the AT-200PC

The AT-200PC is an automatic antenna tuner based on the Microchip PIC 16F876 microcontroller. Unlike other LDG automatic tuners, the AT-200PC has no built-in user interface, but rather has a serial port which communicates to a host PC[1] for user interaction.

1.3 Basic Operation of the AT-200PC

The AT-200PC is designed to remain in sleep mode most of the time, so as to reduce radio interference generated by the onboard microcontroller. As such, the microcontroller needs to be awakened before it can perform any task. There are two ways to wake the AT-200PC:

- Interrupt caused by the detection of an RF carrier
- RS232 RTS (active low) signal received from the host PC

If the user keys up the radio, then the tuner will awaken, and begin measuring the incoming RF signal.Periodically, while RF is present, the tuner will send Power, SWR, and Frequency updates to the host PC [2]. After RF disappears for a time, the tuner will re-enter sleep mode.

If a logic-low-going (positive-voltage-going) RTS signal of at least 3 milliseconds duration is received by the AT200PC while in sleep mode, it will wake up, and will listen for a command byte from the PC host. See Figure 1.1. Once the command is completed, the tuner will go back to sleep. Thus, the tuner must be re-awakened before each command. If, after being awakened, no command is received within a period of time, the AT200PC will re-enter sleep mode.

If a valid command is received, then the AT200PC will act upon that command and then return to sleep.

[1] The term "PC" here means any sort of computer with a serial port, not just Windows-based PCs.
[2] Unless this feature is disabled in software. See section 2.19.

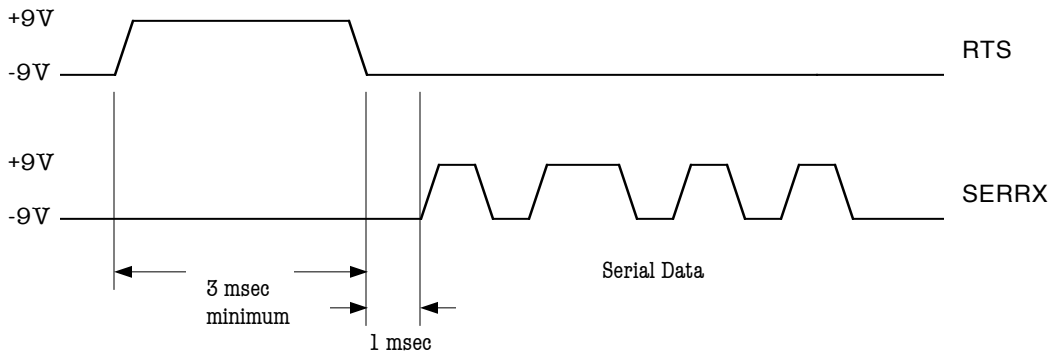

Figure 1.1: Signal Timing For AT200PC Wakeup

1.4 Requirements of the Host PC

The host PC communicates to the AT200PC via an RS232 serial port. The **Transmit Data**, **Receive Data**, **CTS** and **RTS** lines are required. The serial protocol is 9600 baud, 8 data bits, 1 stop bit, no parity, no flow control.

The host PC must be capable of receiving up to 20 characters at a time from the AT200PC without dropping characters. It is recommended that buffering on the host PC is used. The host PC must be capable of of controlling the RTS line, in order to wake the AT200PC.

1.5 General Data Format

Most commands sent from the host PC to the AT200PC are single byte commands. Two-byte commands include the *memory erase* command, which is comprised of a two-byte arming and firing sequence, and the `REQ_SET_IND` and `REQ_SET_CAP` commands, which directly set the inductor or capacitor relay values. The only three-byte command is the `REQ_SET_FREQ` command, which requests setting the relays to the best tune value for a given frequency. Three bytes are required; one for the command, and two bytes for a 16-bit-precision frequency value.

All commands/responses from the AT200PC to the host PC are a fixed four byte format, bytes 0 through 3. Byte 0 is a preamble byte, with a constant value of decimal 165 (hex **$A5**). Byte 1 is a command byte, and the next two bytes are parameters of the command. If the particular command does not have parameters, then the unused bytes are still sent, but are not guaranteed to contain any particular values, and should be ignored by the host PC application. Figure 1.2 shows the general form of a command from the AT200PC to the host PC. Note that bytes 2 and 3 are always present, but may not contain useful data [3], depending on the command.

[3] Do not depend on the data in bytes 2 and 3 if they are undefined, as they may contain uninitialized data.

$$Preamble, Command, [< byte2 >], [< byte3 >]$$

Figure 1.2: Format of Command From AT200PC to Host PC

2. Commands Understood By The AT-200PC

Table 2.2 shows commands that the host PC can send to the AT-200PC. The table has a brief description of each of the commands; a fuller explanation of some of the commands is given here:

2.1 REQ_IND[UP|DN], REQ_CAP[UP|DN]: Inductor/Capacitor Up/Down

The AT200PC has 127 possible values of inductance and 127 possible values of capacitance that it can apply to the tune. The Inductor / Capacitor Up/Down commands step through each of the 127 possible values. If the value is already at its maximum or minimum value, no action is taken.

Upon receipt of an Inductor/Capacitor Up/Down command, the appropriate value is incremented or decremented, the relays updated, and a response code is sent back to the host PC informing the host of the new value of inductance or capacitance, either `CMD_INDVAL` or `CMD_CAPVAL`, with the current value reported in Byte 2.

2.2 REQ_MEMTUNE: Request Memory Tune

This requests that the tuner perform a tune from memory. Upon completion, the tuner sends back either a `CMD_TUNEPASS` on success, or `CMD_TUNEFAIL` if the tuner was not able to successfully tune. If `CMD_TUNEFAIL` is sent, byte 2 of the response will contain a code corresponding to the reason for tune failure. See section 3.11 for more details.

2.3 REQ_FULLTUNE: Request Full Tune

Similar to Request Memory Tune, but requests that a full tuning sequence is initiated. `CMD_TUNEPASS` or `CMD_TUNEFAIL` are sent upon completion. As with `REQ_MEMTUNE`, if the tune fails, the `CMD_TUNEFAIL` response contains the reason for failure.

2.4 REQ_HIZ, REQ_LOZ: Set HiLoZ Relay

If either the `REQ_HIZ` or `REQ_LOZ` commands are sent, the tuner switches the HiLoZ relay appropriately, and then sends back a `CMD_HILOZ` response, with byte 2 set to the appropriate value corresponding to the new HiLoZ value. See section 3.4 (pg. 18) for details.

2.5 REQ_ANT1, REQ_ANT2: Set Antenna

If either the `REQ_ANT1` or `REQ_ANT2` commands are sent, the tuner selects the antenna appropriately, and then sends back a `CMD_ANTENNA` response, with byte 2 set to the appropriate value corresponding to the new antenna selection. The value of byte 2 is equal to the antenna number selected.

2.6 REQ_ALLUPDATE: Request All Update

When this command is received by the AT200PC, it sends out a series of commands to the host PC to inform it of its current operating status. The following fields are updated:

- Inductor Value
- Capacitor Value
- HiLoZ switch status
- Current Antenna selection
- Standby / Active status
- Automatic Tuning enabled / disabled
- SWR tuning threshold
- Forward Power
- Reflected Power
- Live Updates On/Off Status
- Last Transmitted Frequency
- SWR value

2.7 REQ_VERSION: Request Version

Upon receipt of this command, the AT200PC responds with a `CMD_VERSION`, and 2 bytes of parameter data. The first byte contains the product ID, and the second byte is a BCD-encoded version number. The most significant nybble of the version number byte is the major revision number, and the least significant nybble is the minor version number. For example, if the version byte is decimal 36 (hex 0x24), then the version is 2.4.

At present, the only valid product ID is 01, indicating the AT-200PC.

2.8 REQ_ARM_CLEAR, REQ_CLEAR_MEM: Clearing EEPROM

The EEPROM of the AT200PC contains the memory tuning data used for fast memory tunes. At time of manufacture, and possibly at the user's request, the EEPROM data can be cleared to initialize the values in it. Because this erases any useful data that may be stored in the memory, and because it is a lenghty operation (it may take up to 30 seconds or so), the command to erase EEPROM is safeguarded by an arming sequence.

First, the memory clear routine must be armed by sending a `REQ_ARM_CLEAR` command. Next, the `REQ_CLEAR_MEM` command must be sent. No commands may be sent in between, or the memory clear routine becomes disarmed.

Once the memory clear sequence is issued, EEPROM data begins clearing. Once completed, the AT200PC will send a `CMD_CLEAR_DONE` code back to the host PC to indicate that it has cleared EEPROM and is ready for the next command.

2.9 REQ_TUNER_STANDBY, REQ_TUNER_ACTIVE: Tuner Standby / Tuner Active

Upon receipt of the `REQ_TUNER_STANDBY` or `REQ_TUNER_ACTIVE` commands, the AT200PC will either enter standby mode (all relays off) or active mode (restore to previous relay settings before standby). A `CMD_INSTANDBY` or `CMD_ACTIVE` response will be sent back to the host PC to acknowledge receipt of the command.

2.10 REQ_MANUAL_STORE: Store Tune

Upon receipt of the `REQ_MANUAL_STORE` command, the AT200PC will attempt to store the current relay settings in the memory that corresponds to the most recently transmitted frequency. A `CMD_STORE_OK` will be returned when completed.

2.11 REQ_SWRxx: Set SWR Threshold

The SWR Threshold is the value of SWR above which the AT200PC will not store a tune. In other words, if the lowest SWR attained while tuning was not below this value, the tune will be considered a failed tune.

The value of SWR Threshold is also used by the autotuning algorithm. When automatic tuning is enabled (see section 2.13), any time RF is present, the SWR is checked against the SWR Threshold. If the SWR exceeds the SWR Threshold value, then an automatic tuning sequence begins.

In memory tuning mode, if the SWR is not below this value after trying the memory locations, a full tune will be initiated. Upon receipt of a `REQ_SWRxx` command, the AT200PC will set the SWR threshold to the indicated value, and then send back a `CMD_SWRTHRESH`, with byte 2 indicating the current value of SWR Threshold. Possible values of SWR Threshold returned are shown in table 2.1.

Byte 2 value	SWR Threshold
0	1.1:1
1	1.3:1
2	1.5:1
3	1.7:1
4	2.0:1
5	2.5:1
6	3.0:1

Table 2.1: SWR Threshold Values in `CMD_SWRTHRESH`

2.12 REQ_RESET: Reset Tuner

This request to the tuner causes the Inductor and Capacitor relays to be set to their zero value (no added inductance or capacitance) and causes the HiLoZ relay to assume its default setting. The tuner will respond to this request with commands updating the status of the inductor, capacitor, and HiLoZ relay settings.

2.13 REQ_AUTO_ON: Enable Automatic Tuning

Sending this request to the tuner enables automatic tuning. Automatic tuning means that the tuner will automatically begin a memory tuning sequence if it detects an SWR value greater than the SWR Threshold (see section 2.11 for details).

The tuner responds to this request with a `CMD_AUTO_STATUS` command with a 1 as its parameter byte.

As with normal memory tunes, if the memory tune fails to acheive a good match, the tuner will proceed on to a full tune sequence.

A `CMD_TUNE_FAIL` or `CMD_TUNE_PASS` will be sent at the end of the tuning sequence.

During the time that the automatic tuning is occurring, the tuner will assert the CTS line until the tune has completed. The AT200PC is not capable of responding to serial commands while tuning, and so this line can be used as a hardware handshaking line on the host PC. The host PC application should not send data while the CTS line is asserted.

2.14 REQ_AUTO_OFF: Disable Automatic Tuning

Sending this request to the tuner disables the automatic tuning feature. Tuning sequences will only be initiated when a tuning request is received.

The tuner responds to this request with a `CMD_AUTO_STATUS` command with a 0 as its parameter byte.

2.15 REQ_FWD_PWR: Request Forward Power Value

Sending this request to the tuner causes the tuner to reply with a CMD_FWDPWR command, indicating the current forward power. See section 3.6.

2.16 REQ_REV_PWR: Request Reflected Power Value

Sending this request to the tuner causes the tuner to reply with a CMD_REVPWR command, indicating the current reflected power. See section 3.7.

2.17 REQ_SWR: Request SWR Value

Sending this request to the tuner causes the tuner to reply with a CMD_SWR command, indicating the current value of SWR. See section 3.8.

2.18 REQ_UPDATE_ON: Request Live Updates On

At power-up, the tuner is already in Live Updates On mode. This request can be sent by the host PC to re-enable live updates if it had previously sent a REQ_UPDATE_OFF command.

Live Updates mode is when the tuner automatically sends the current forward power, reflected power, frequency, and SWR data to the host PC any time transmitted RF is present. In a graphically-oriented software control package, it may be useful to have the tuner give live updates to the PC, rather than have the PC continually poll the tuner for this information.

Upon receipt of a REQ_UPDATE_ON request, the AT200PC will enable Live Updates mode, and will send a CMD_UPDATE_STATUS command, with byte 2 set to 1.

2.19 REQ_UPDATE_OFF: Request Live Updates Off

Sending this request disables live updates during the presence of transmitted RF. Send this code to the tuner if you do not wish to have the tuner send automatic updates in the presence of transmitted RF.

Upon receipt of a REQ_UPDATE_OFF request, the AT200PC will disable Live Updates mode, and will send a CMD_UPDATE_STATUS command, with byte 2 set to 0.

2.20 REQ_SET_IND: Set Inductor and HiLoZ Relay Value Directly

This is a two-byte request. The second byte of the request contains the requested inductor relay value (0 to 127) and the value of the HiLoZ relay being requested. Bits 0-6 of the second byte of the request indicate

the inductor relay value, and bit 7 indicates the setting of the HiLoZ relay. A value of 0 indicates high impedance (antenna side caps), and a value of 1 indicates low impedance (transmitter side caps).

The tuner responds to this request by setting the relays as requested, and then sending a `CMD_INDVAL` command followed by a `CMD_HILOZ` command, indicating the new values of inductor and HiLoZ.

2.21 REQ_SET_CAP: Set Capacitor Relay Value Directly

This is a two-byte request. The second byte of the request contains the requested capacitor relay value (0 to 127). Values outside this range are truncated to 0-127.

The tuner responds to this request by setting the relays as requested, and then sending a `CMD_CAPVAL` command indicating the new value of capacitor.

2.22 REQ_SET_FREQ: Request Tuner Recall for Frequency

Sending this command to the AT200PC causes the AT200PC to look up the specified frequency in its tuning memory, and recall the relay setting for that frequency if there is a relay setting stored for the specified frequency. If no setting is found, the memory locations on either side of that frequency are also checked. If no data is found, no action is taken. Otherwise, the memorized relay settings corresponding to the requested frequency are recalled, and the relays set accordingly. Note that if more than one set of relay data is available for the selected frequency, the first one recalled is used. Thus, if the tuner has been used with more than one antenna on the selected port, it is possible that the data recalled is for a different antenna.

In version 1.7 of the firmware, some debugging information was left in the code. This causes the tuner to send an invalid code (code 0x64 hex, 100 decimal), followed by three additional dummy bytes, after a `REQ_SET_FREQ` request. Third-party software developers should receive this four-byte message, but ignore it. This "dummy" packet may be removed in a future release of the firmware.

~~After recalling the memory setting,~~ After sending this dummy packet, the tuner responds in the same manner as if it had received a `REQ_ALLUPDATE` request. See section 2.6 for a description of the `REQ_ALLUPDATE` request.

The format for the frequency data is the same as that of the `CMD_TXFREQ`[1] command; the frequency is given as a period, in ticks. The second byte of the `REQ_SET_FREQ` command is the MSB of the period, and the third byte is the LSB of the period. Valid values of period range from 370 ticks (55.3 MHz) to 11593 ticks (1.76 MHz).

A shortcut for determining the period value from the desired operating frequency is

$$Period = \frac{20,480}{Freq(MHz)}$$

Here's an example of setting the frequency to 14.230 MHz:

$$Period = \frac{20,480}{14.230} = 1439$$

[1] See section 3.9, page 20.

$$MSB(1439) = \texttt{0x05}$$
$$LSB(1439) = \texttt{0x9F}$$

So the command would be sent as:

```
0x43 0x05 0x9F
```

Note that the accuracy of the frequency counter in the AT200PC is somewhat limited. The frequency reported by the AT200PC may differ slightly from the actual transmitted frequency; bear this in mind when sending a `REQ_SET_FREQ` request.

2.23 REQ_MEM_DUMP: Memory Dump

Upon receipt of the `REQ_MEM_DUMP` command, the AT200PC begins transmitting the contents of EEPROM. The data format is raw binary. Sixteen bytes are transmitted at a time, followed by a 50 millisecond pause to allow the host to catch up. 32,768 bytes are transmitted. The layout of the memory dump is in section 4.1.

Command Name	Value (dec)	Description
REQ_NOOP	0	No Operation
REQ_INDUP	1	Inductor Up. Increase the current inductor value by one.
REQ_INDDN	2	Inductor Down. Decrease the current inductor value by one.
REQ_CAPUP	3	Capacitor Up. Increase the current capacitor value by one.
REQ_CAPDN	4	Capacitor Down. Decrease the current capacitor value by one.
REQ_MEMTUNE	5	Request Memory Tune. Attempt to tune via a built-in stored tune value for this frequency. Will fall through to a full tune if the memory tune fails.
REQ_FULLTUNE	6	Request Full Tune. Do a complete tuning sequence.
REQ_HIZ	8	Set the HiLoZ relay to High Impedance.
REQ_LOZ	9	Set the HiLoZ relay to Low Impedance.
REQ_ANT1	10	Select Antenna 1
REQ_ANT2	11	Select Antenna 2
REQ_ALLUPDATE	40	Request that the tuner send an update of current relay settings.
REQ_VERSION	41	Ask the tuner for its product ID and version number.
REQ_ARM_CLEAR	42	Arm the memory clear routine. This must be sent immediately before a REQ_CLEAR_MEM command.
REQ_CLEAR_MEM	43	Clear EEPROM memory data. The memory clear routine must first be armed by sending a REQ_ARM_CLEAR command. If any intervening commands are sent, the memory clear armed status is unset, and the memory clear will not take place.
REQ_TUNER_STANDBY	44	Place the tuner in standby mode.
REQ_TUNER_ACTIVE	45	Take the tuner out of standby mode.
REQ_MANUAL_STORE	46	Store Tune. Stores the current inductor and capacitor relays settings at the memory location corresponding to the last frequency transmitted on.
REQ_SWR11	50	Set SWR threshold for “good tune” to 1.1:1 or lower.
REQ_SWR13	51	Set SWR threshold for “good tune” to 1.3:1 or lower.
REQ_SWR15	52	Set SWR threshold for “good tune” to 1.5:1 or lower.
REQ_SWR17	53	Set SWR threshold for “good tune” to 1.7:1 or lower.
REQ_SWR20	54	Set SWR threshold for “good tune” to 2.0:1 or lower.
REQ_SWR25	55	Set SWR threshold for “good tune” to 2.5:1 or lower.
REQ_SWR30	56	Set SWR threshold for “good tune” to 3.0:1 or lower.
REQ_RESET	57	Reset L and C relays, HiLoZ relay
REQ_AUTO_ON	58	Request Automatic Tuning On
REQ_AUTO_OFF	59	Request Automatic Tuning Off
REQ_FWD_PWR	60	Request Current FWD Power
REQ_REV_PWR	61	Request Current Reflected power
REQ_SWR	62	Request Current SWR
REQ_UPDATE_ON	63	Turn live updates on
REQ_UPDATE_OFF	64	Turn live updates off
REQ_SET_IND	65	Directly set Inductor (and HiLoZ) relay value
REQ_SET_CAP		66 Directly set Capacitor relay value
REQ_SET_FREQ	67	Request that tuner recalls tuning memory relay setting for the specified frequency
REQ_MEM_DUMP	68	Request memory dump of all of EEPROM data.

Table 2.2: Commands Understood by AT200PC

3. Commands (Responses) To The Host PC From The AT-200PC

Table 3.6 shows commands that the AT200PC can send to the host PC. The table has a brief description of each of the commands; a fuller explanation of some of the commands is given here:

3.1 CMD_NOOP: No-Op

No Operation. This is also sent to the host PC upon waking from an RF-detect interrupt, so that the host PC knows that the AT-200PC has awakened.

3.2 CMD_INDVAL: Report Inductor Value

This command is issued from the AT200PC in response to the host PC sending a `REQ_INDDN` or `REQ_INDUP` command, or when the host PC sends a `REQ_ALLUPDATE` command. Byte 2 of the response contains the integer inductor value, from 0 to 127.

3.3 CMD_CAPVAL: Report Capacitor Value

This command is issued from the AT200PC in response to the host PC sending a `REQ_CAPDN` or `REQ_CAPUP` command, or when the host PC sends a `REQ_ALLUPDATE` command. Byte 2 of the response contains the integer capacitor value, from 0 to 127.

3.4 CMD_HILOZ: Report HiLoZ Relay Status

This command is issued from the AT200PC in response to the host PC sending a `REQ_HIZ` or `REQ_LOZ` command, or when the host PC sends a `REQ_ALLUPDATE` command. Byte 2 of the response contains either a zero or one, depending on the state of the HiLoZ relay. Table 3.1 shows the values of byte 2 and the corresponding impedance selection.

Byte 2	Impedance
00	High (Antenna side Caps)
01	Low (Transmitter side Caps)

Table 3.1: HiLoZ Values in `CMD_HILOZ`

3.5 CMD_ANTENNA: Report Antenna Selection

This command is issued from the AT200PC in response to the host PC sending a `REQ_ANT1` or `REQ_ANT2` command, or when the host PC sends a `REQ_ALLUPDATE` command. Byte 2 of the response contains either a zero or one, depending on which antenna is selected. Table 3.2 shows the values of byte 2 and the corresponding antenna selection.

Byte 2	Antenna Selection
00	Antenna 1
01	Antenna 2

Table 3.2: Antenna Values in `CMD_ANTENNA`

3.6 CMD_FWDPWR: Report Forward Power

If Live Update mode is on, this command is sent periodically while RF power is detected. Byte2 and Byte3 form a 16-bit value equal to 100 times the actual forward power, in watts. In this manner, more resolution is acheived than if an integer number of watts were reported. Byte2 is the MSB and Byte3 is the LSB of the 16-bit value. Valid vales range from zero to 25,000 (250 watts).

3.7 CMD_REVPWR: Report Reflected Power

If Live Update mode is on, this command is sent periodically while RF power is detected. Byte2 and Byte3 form a 16-bit value equal to 100 times the actual reflected power, in watts. In this manner, more resolution is acheived than if an integer number of watts were reported. Byte2 is the MSB and Byte3 is the LSB of the 16-bit value. Valid vales range from zero to 25,000 (250 watts).

3.8 CMD_SWR: Report SWR

If Live Update mode is on, this command is sent periodically while RF power is present. Byte2 and Byte3 form a 16-bit value, although this value is never over 255, so Byte3, the LSB, may be used solely by the host application. The value reported represents $256 \times \rho^2$ in the classic impedance calculation:

$$SWR = \frac{(1+\rho)}{(1-\rho)}$$

So, if $256 \times \rho^2$ is, for example, 64, then

$$SWR = \frac{(1+\sqrt{\rho^2})}{(1-\sqrt{\rho^2})}$$

$$SWR = \frac{(1 + \sqrt{\frac{64}{256}})}{(1 - \sqrt{\frac{64}{256}})}$$

$$SWR = \frac{(1 + \sqrt{0.25})}{(1 - \sqrt{0.25})}$$

$$SWR = \frac{(1 + 0.5)}{(1 - 0.5)}$$

$$SWR = \frac{1.5}{0.5}$$

$$SWR = 3.0 : 1$$

Valid values of SWR range from 0 (1:1 SWR) to 255 (1022:1 SWR).

3.9 CMD_TXFREQ: Report Transmit Frequency

If Live Update mode is on, this command is sent periodically while RF power is present. Bytes 2 and 3 form a 16-bit value equal to half the period of the transmitted RF carrier, in ticks. Byte2 is the MSB, Byte3 is the LSB. One tick is equal to 0.8 microseconds. Thus, if the value reported is 471, then

$$Period = 471 \times 0.8\mu Sec$$

$$Period = 3.768 \times 10^{-4} Sec$$

But this is the half-period, so the actual period is:

$$3.768 \times 10^{-4} Sec \times 2 = 7.536 \times 10^{-4} Sec$$

And thus, the frequency is:

$$Freq = \frac{1}{7.536 \times 10^{-4}} = 1326.9\ Hz$$

However, the frequency counter on the AT200PC internally divides the incoming frequency by 32,768 before measuring it, so you must then multiply the result by 32,768 to get the actual frequency value, thus

$$Freq = 1326.9 \times 32,768 = 43,481,953\ Hz$$

A simplification is:

$$Freq(Mhz) = \frac{20,480}{Period}$$

$$Freq(Mhz) = \frac{20,480}{471} = 43.48\ MHz$$

Valid values of transmit frequency range from 370 (55.35 MHz) to 11,594 (1.766 MHz).

3.10 CMD_TUNEPASS: Tune Succeeded

This command is sent to the host PC upon successful completion of a tune request, either memory tune or full tune. It is also sent at the completion of an automatic tune sequence.

3.11 CMD_TUNEFAIL: Tune Failed

This command is sent to the host PC if a tune request has failed. It is also sent at the end of an automatic tuning sequence, if the automatic tuning sequence fails. A tune request can fail for a number of reasons, including:

- No RF carrier detected.
- RF carrier was lost before the tune completed.
- A good match was not able to be made (The antenna impedance was too far off for the tuner to make a good match).

Byte 2 of the `CMD_TUNEFAIL` commands indicates the reason for the failure to tune. Table 3.3 shows the valid values for the failure reason.

Byte 2 Value	Failure Reason
00	No RF was detected.
01	RF Carrier was lost before the tune completed.
02	The tuner was unable to bring the SWR down below the SWR Threshold.

Table 3.3: Tuning Failure Reason Codes

3.12 CMD_VERSION: Report Product ID and Version

This command is sent in response to a `REQ_VERSION` command from the host PC. Byte2 of the response contains the product ID. Byte 3 is a BCD-encoded version number. The most significant nybble contains the major version number, and the least significant nybble contains the minor revision number.

Presently, the product ID is fixed at a value of 01, and the version number may range from 0 to 255. (0.0 to 15.15)

3.13 CMD_CLEAR_DONE: EEPROM Erase Cycle Complete

This command is sent to the host PC upon completion of an erase EEPROM request.

3.14 CMD_INSTANDBY: Indicate Tuner is in Standby Mode

This command is issued to the host PC in response to either a `REQ_TUNER_STANDBY` request, or a `REQ_ALLUPDATE` request if the tuner is in the standby mode when the `REQ_ALLUPDATE` request is issued.

3.15 CMD_ACTIVE: Indicate Tuner is in Active Mode

This command is issued to the host PC in response to either a `REQ_TUNER_ACTIVE` request, or a `REQ_ALLUPDATE` request if the tuner is in the active mode when the `REQ_ALLUPDATE` request is issued.

3.16 CMD_STORE_OK: Indicate Manual Memory Store Complete

This command is issued to the host PC in response to a `REQ_MANUAL_STORE` request, after the store operation has completed.

3.17 CMD_SWRTHRESH: Report Current SWR Threshold

Byte 2 of this command indicates the current SWR Threshold. See table 2.1 on page 13 for valid values of Byte 2. This command is sent in response to a `REQ_SWRxx` command, or a `REQ_ALLUPDATE` command.

3.18 CMD_AUTO_STATUS: Report Status of Auto Tune Setting

Byte 2 of this command indicates whether automatic tuning is enabled or disabled. This command is sent in response to a `REQ_AUTO_ON` or `REQ_AUTO_OFF` request, or in response to a `REQ_ALLUPDATE` request. Valid return values are 0 or 1, shown in table 3.4.

Byte 2 Value	Meaning
00	Automatic tuning disabled.
01	Automatic tuning enabled.

Table 3.4: `CMD_AUTO_STATUS` Byte 2 Parameter Values

3.19 CMD_UPDATE_STATUS: Report Status of Live Update on/off

Byte 2 of this command indicates whether Live Updates are enabled or disabled. This command is sent whenever the Live Update status is changed, or when a `REQ_ALLUPDATE` is sent. Table 3.5 shows the possible values of byte 2 of the `CMD_UPDATE_STATUS` response.

Byte 2 Value	Meaning
00	Live Updates Disabled.
01	Live Updates Enabled.

Table 3.5: `CMD_UPDATE_STATUS` Byte 2 Parameter Values

Command Name	Value (dec)	Parm1	Parm2	Description
CMD_NOOP	00	none	none	No Operation - Also sent to indicate AT200PC has woken up
CMD_INDVAL	01	Value 0-127	none	Inductor Value
CMD_CAPVAL	02	Value 0-127	none	Capacitor Value
CMD_HILOZ	03	0 or 1	none	HiLoZ relay value
CMD_ANTENNA	04	0 or 1	none	Antenna Selection
CMD_FWDPWR	05	MSB	LSB	Forward Power, in hundredths of watts. I.e. a value of 1000 indicates 10.0 watts
CMD_REVPWR	18	MSB	LSB	Reflected Power, in hundredths of watts.
CMD_SWR	06	none	LSB	SWR (see text)
CMD_TXFREQ	07	MSB	LSB	Transmit period (see text)
CMD_TUNEPASS	09	none	none	Indicates that the requested tuning operation succeeded.
CMD_TUNEFAIL	10	Reason 0-2	none	Indicates that the requested tuning operation failed. Byte 2 indicates reason for failure. See table 3.3.
CMD_VERSION	11	ProdID	MajMin	Indicates product and major and minor version number.
CMD_CLEAR_DONE	12	none	none	Indicates that the EEPROM erase operation has completed.
CMD_INSTANDBY	13	none	none	Indicates that the tuner has been placed in standby mode.
CMD_ACTIVE	14	none	none	Indicates that the tuner has restored the settings that were saved prior to entering standby mode.
CMD_STORE_OK	15	none	none	Indicated that the requested store to memory operation has completed. Note that this will be indicated whether the store operation failed or succeeded.
CMD_SWRTHRESH	16	Thresh 0-6	none	Indicates the currently active SWR threshold used for tuning. See table 2.1 for values.
CMD_AUTO_STATUS	17	0 or 1	none	Report Auto Tune setting. See table 3.4 for values.
CMD_UPDATE_STATUS	19	0 or 1	none	Indicate Live Update on/off

Table 3.6: Commands Sent to the Host PC by the AT200PC

4. AT200PC Memory Dump

4.1 AT200PC Memories

The AT200PC stores inductor, capacitor, and HiLoZ relay settings for successful tunes in memory. The memory slot number is determined by the transmitted frequency. Since there are not an infinite number of memories, the memories are organized into "buckets", each of which corresponds to a small range of frequencies. If the transmitted frequency falls anywhere within the range of frequencies contained in a "bucket", then that bucket is used for either store or recall, depending on the operation.

Additionally, there are separate banks of memories for each antenna port (2 in total), and for each antenna port there are four groups of memories which are searched sequentially when looking up a tune value for a particular frequency on a particular antenna port. So for a given frequency, there are a possible 8 locations where that frequency's data may be stored.

4.2 AT200PC Memory Map

As stated above, the AT200PC's memories are divided into banks, two per-antenna banks, and four per-frequency banks in each of the two antenna banks. Each entry is two bytes long. Table 4.1 shows the format for each memory entry. Figure 4.1 shows how the memories are laid out in physical memory.

4.3 Nonlinearity of Memory Map

One additional detail is the mapping of memory address to frequency. There is not a 1:1 mapping of frequency to memory address, as this would not give good frequency resolution where it is needed. Thus, a provided include file that maps memory address to center frequency for that memory's bucket is needed in order to determine the center frequency that is represented by each memory location.

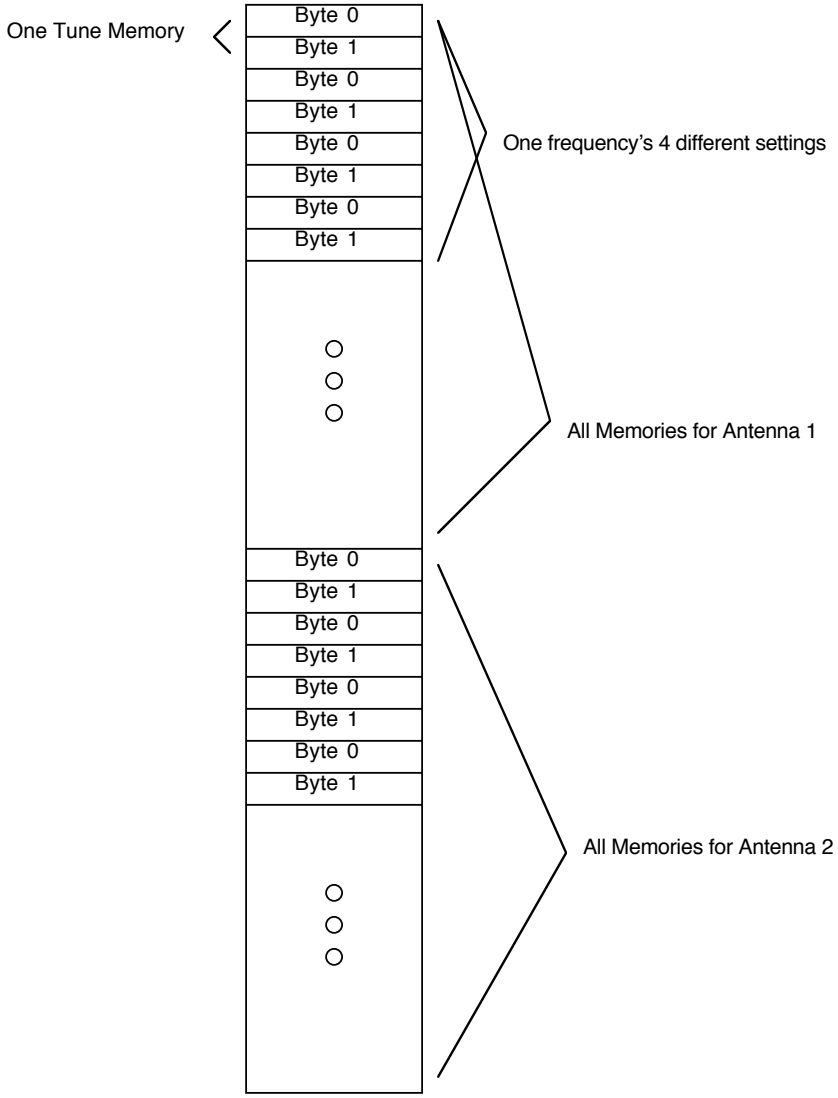

Figure 4.1: AT200PC EEPROM Memory Map

Byte 0 Format

Bit	Name	Description
7	HiLoZ	HiLoZ Relay setting
6	CAP6	Capacitor relay 6
5	CAP5	Capacitor relay 5
4	CAP4	Capacitor relay 4
3	CAP3	Capacitor relay 3
2	CAP2	Capacitor relay 2
1	CAP1	Capacitor relay 1
1	CAP0	Capacitor relay 0

Byte 1 Format

Bit	Name	Desctiption
7	MemFree	Set if memory location unused
6	IND6	Inductor relay 6
5	IND5	Inductor relay 5
4	IND4	Inductor relay 4
3	IND3	Inductor relay 3
2	IND2	Inductor relay 2
1	IND1	Inductor relay 1
0	IND0	Inductor relay 0

Table 4.1: Memory Format

A. Document History

Revision	Date	Description
1.2	8/11/2005	Initial release
1.21	8/12/2005	Updated section 4.3 to reflect that frequency coverage is continuous, but frequency-to-memory-address mapping is nonlinear.
1.3	8/18/2005	Updated to reflect version 1.3 of firmware. Notably, changed to reflect that RTS signal is used instead of BREAK signal, plus added features.
1.7	2/23/2006	Updated to reflect version 1.7 of firmware. Added note about unused response code 100 (0x64), and clarified operation of RTS and sleep. Added underline and ~~strikethru~~ to indicate changes from previous documents. See updates in sections 1.3 and 2.22.