Alphalab Inc.

Data Acquisition Communication Protocol

Scope

This document explains in detail the instruction set used to communicate with all data acquisition enabled units. It should be used when building any unit that provides data acquisition capability now and in the future. This shall be the master documentation and will be updated accordingly as improvements and alterations are made to the protocol. It is intended as internal documentation ONLY.

Objective

In order to simplify and streamline our data acquisition software so that only one PC program is necessary a general communication protocol is required that is not only universal but easily adaptable and expandable as new meters with new features are added. Additionally, backwards compatibility will require that these commands change very little over time as new versions are implemented.

General Structure

All communication is initiated by a host (PC) with the attached meter responding. Each initiation starts with a six byte transmission. Most responses involving configuration information will be returned in ASCII format and responses involving measured data will be returned in Binary format. For each meter all configuration information is stored on board the device to keep the universality among devices despite varying firmware/hardware revisions.

Channel Settings

All serial COM port settings are static and unchanging:

- Baud rate: 115200
- Data Bits: 8
- Stop Bits: 1
- Parity: None

USB settings: (Any pertinent USB settings will be added here)

Ethernet settings: (Any pertinent Ethernet settings will be added here)

Command Table

	Command	Subcommand	Command Byte	Page Number
Identification Commands				
	ID_METER_PROP		0x01	4
	ID_METER_SETT		0x02	6
	ID_METER_ADC_SETT		0x1C	7
Stream, Time, and Reset Commands				
	STREAM_DATA		0x03	8
	RESET_TIME		0x04	11
+	BURST_DATA		0x22	11
	RESET_ZERO		0x1B	11
		SAVE_ZERO	0x01	
+	IDENT_SYS_TIME		0x06	12
+	ALTER_SYS_TIME		0x0D	12
Flash Chip Access Commands				
	FILE_DIRECTORY		0x05	13
	DOWNLOAD_FILES		0x0B	15
		BREAK_SUSPEND	0x14	
	DELETE_FILE		0x0C	15
	DELETE_CHIP		0x13	16
	NUKE_CHIP		0x19	16
*	REMOTE_STORE		0x1A	17
*	DUMP_CHIP_TO_PC		0x15	17
*	DUMP_BUFF_TO_PC		0x16	18
*	DUMP_DATA_TO_CHIP		0x21	19
General Communication Commands				
		TERMINATE	0x07	19
		ACKNOWLEDGE	0x08	20
+		CRC_ACKNO	0x09	20
+		CRC_RESEND	0x0A	20
		NULL_ACTION	0x00	20
	KILL_ALL_PROCESS		0xFF	21
Setting Alteration Commands				
	ALTER_METER_SETT		0x0E	21
		ALTER_STREAM_PERIOD	0x0F	
+		ALTER_FILE_HEAD_TYPE	0x10	
		ALTER_ADC_INDEX	0x12	
		ALTER_RECORD_CHANGE	0x17	
		ALTER_DC_PH_COMB	0x18	
Digital Potentiometer Calibration Commands				
*	SET_DIGITAL_POTS		0x1D	22
*	SAVE_DIGITAL_POTS		0x1E	23
*	READ_DIGITAL_POTS		0x1F	23
*	FLSH_WRT_DIG_POTS		0x20	24

* Command for in-house use only

\+ Command is reserved but not implemented

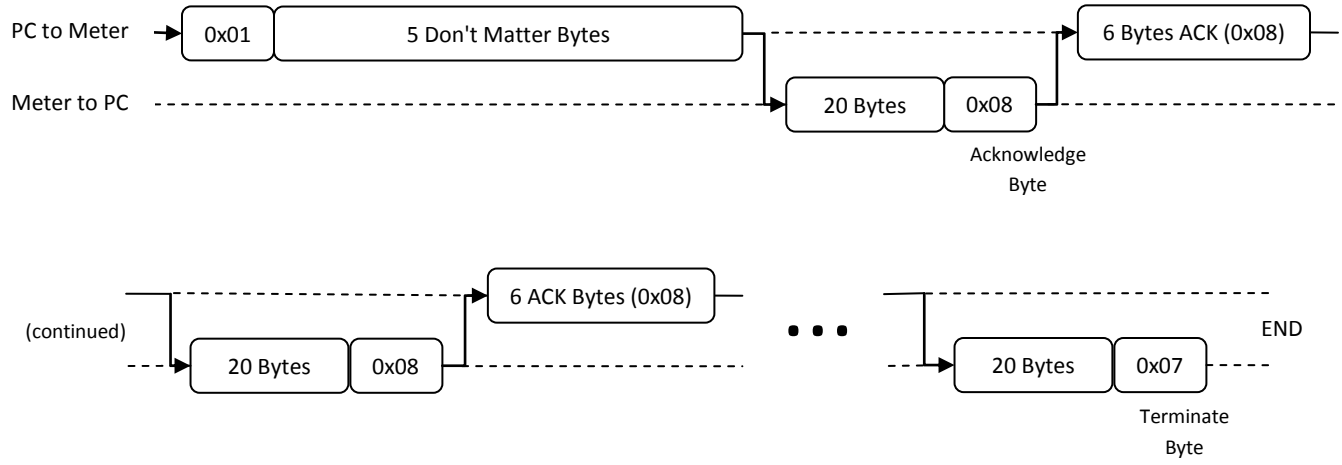
Command Detail

A. Identification Commands

ID_METER_PROP (0x01)

Meter properties are meter characteristics that define what that particular meter is and what it is capable of. They are static values that do not change during the life of the meter but they aren't necessarily the same from meter to meter. In fact, they don't necessarily have to be the same for two meters that are the same model. This means each hardware/firmware revision can be updated to include new features while the data acquisition software remains relatively unchanged.

To initiate this command the meter is sent a command byte (0x01) followed by five bytes whose contents don't matter. The meter will return an ASCII block of data in twenty byte chunks followed by a terminate byte if the end of data has been reached or an acknowledge byte if there is more data to send. The PC will send a six byte acknowledge command back to initiate the next chunk of twenty bytes or, in the case of a terminate byte from the meter, will end the transaction. The total length of the transaction is dependent on what features the meter is equipped with. The final chunk of data may have unused filler bytes to complete the chunk of twenty.



The data output of the ID_METER_PROP command is in ASCII format. Each property field is separated by colons and the property label is further separated from its value using an equal sign:

Property1=Value1:Property2=Value2:……:PropertyN=ValueN:

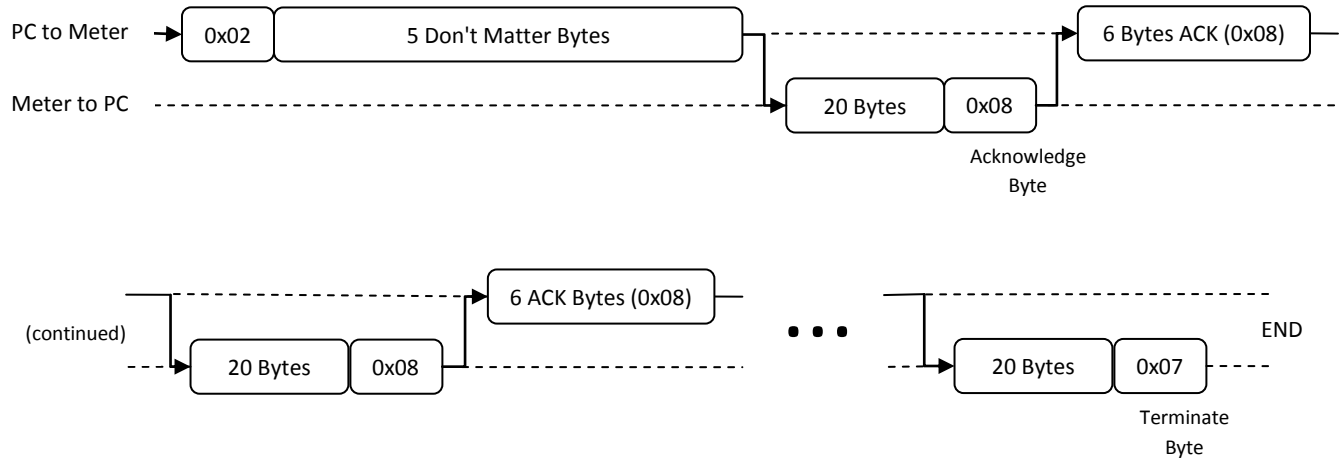
Property Descriptions

METER_NAME	A designation for the meter connected to the PC.
FIRMWARE	A designation for the firmware onboard the meter (this is currently used only for informational purposes).
TABLE_HEADERS	A comma delimited list of the labels for each data point that will be transmitted. This list will include expected units of measure for the measured fields.
TABLE_WIDTH	An integer value that describes how much space (width) should be allotted for each table of data.
MAX_DATA_SETS	An integer value that states how many tables can be open at one time.
BASE_FREQ	This is the base time period at which the unit operates in seconds. This in conjunction with AVBL_FREQS is what defines the rate of streaming or recording of data.
AVBL_FREQS	A comma delimited list of integers that define which periods the unit can be set to record or stream at.
POSS_FILE_HEAD	When this variable is present then the meter has onboard memory that the user can use to store data. Its value designates how the user would like to interpret the file headers. Currently only one option is supported (0000001) which simply numbers the data set. It is open ended so that features can be added, such as, timestamps.
NO_DEC_OVERANGE	A float value that holds the overange value for all data points that have no decimal places.
ONE_DEC_OVERANGE	A float value that holds the overange value for all data points that have one decimal place.
TWO_DEC_OVERANGE	A float value that holds the overange value for all data points that have two decimal places.
THR_DEC_OVERAGE	A float value that holds the overange value for all data points that have three decimal places.
NO_SETTINGS	When this tag is present then the meter has no user selectable settings.
VAR_ADC_SETT	When this tag is present then the meter is capable of user selectable ADC settings.
ADC_CAL	When this tag is present the meter requires calibration of its single ADC. This is only used in a factory environment.
ADC6_CAL	When this tag is present the meter requires a 6 channel calibration of ADC's (it is only used for the 6 channel USB ADC product). This is only used in a factory environment.
REC_CHG	When this tag is present the meter is capable of recording all data or only when changes to the data occur.
DC_WITH_PH	When this tag is present the meter is capable of streaming (but not recording) peakhold data while also streaming DC field data.
REMOTE_ZERO	When this tag is present the meter is capable of being zeroed from the PC toolbar.
DATA_LOAD	When this tag is present the meter requires data from the PC to be loaded to onboard flash memory (such as screen data). This is only used in a factory environment.

ID_METER_SETT (0x02)

Meter settings are the values of all user selectable options that define the desired behavior of the meter. They can be altered through software when the unit is connected to a PC. In some cases they can be altered on the meter itself (when the user interface allows). The list of possible meter setting categories is determined by the units meter properties (i.e., if there is no "REC_CHG" option in the meter property list then there will not be a corresponding option in the meter settings list).

To initiate this command the meter is sent one command byte (0x02) followed by 5 bytes whose contents don't matter. Upon receipt the meter will send an ASCII block of data to the PC in twenty byte chunks followed by a terminate byte if the end of data has been reached or an acknowledge byte if there is more data to send. The PC will send a single acknowledge byte back to initiate the next chunk of twenty bytes or in the case of a terminate byte from the meter will end the transaction. The total length of the transaction is dependent on what features the meter is equipped with. The final chunk of data may have unused filler bytes to complete the chunk of twenty.



The data output of the ID_METER_SETT command is in ASCII format. Each setting field is separated by colons and the setting label is further separated from its value using an equal sign:

Setting1=Value1:Setting2=Value2:.....:SettingN=ValueN:

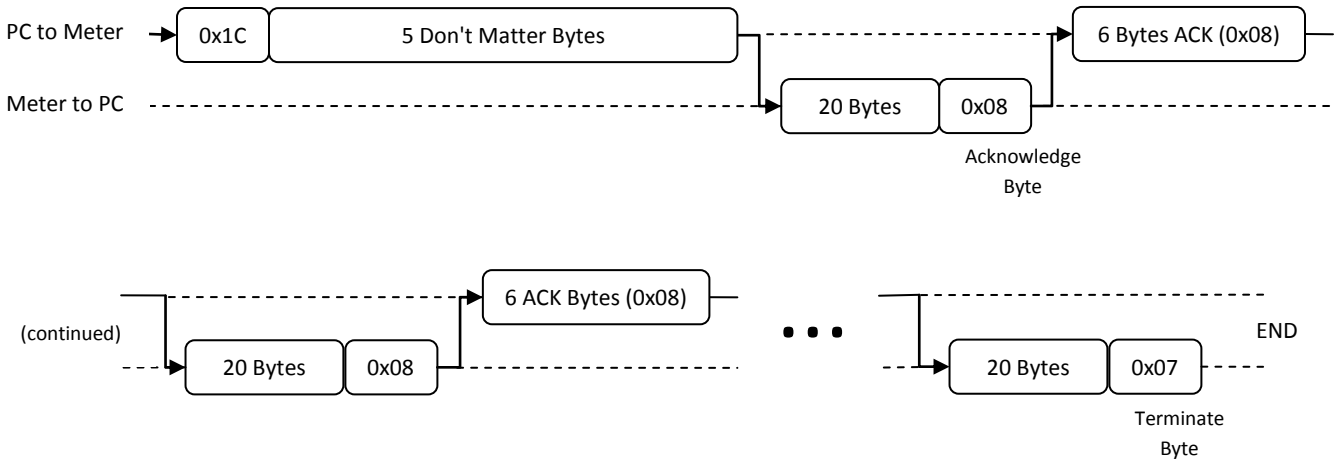
Setting Descriptions

CURR_FREQ	An integer that can only be one of the values that exists in the properties list AVBL_FREQS. This value, in combination with BASE_FREQ, designates what period the meter streams or records data at.
FILE_HEAD	A list of 1's and 0's returned by meters equipped with onboard memory. This data will be one of the possible flags within POSS_HEAD in the meter properties list. This flag will designate how the user would like their files labeled in the PC end application. There is currently only one option implemented with room to offer more.
REC_CHG	An integer value that is either 1 or 0. If the user would like to record or stream only those records where changes exist in the data record then the option will be 1, otherwise 0.
DC_WITH_PH	An integer value that is either 1 or 0. If the user would like to stream both peakhold and DC data simultaneously then the option will be 1, otherwise 0.
VAR_ADC	For meters with VAR_ADC_SETT in property list, this integer value corresponds to the ADC settings that the user has selected.

ID_METER_ADC_SETT (0x1C)

This command is only relevant on meters that have the VAR_ADC_SETT tag in their meter properties list. It returns a list of meter ADC setting that are pre-defined, user selectable, configuration settings for the meter ADC(s). The length of this list will be meter specific and dependent on how many configuration options have been made available to the user.

To initiate this command the meter is sent one command byte (0x1C) followed by five bytes whose contents don't matter. The meter will return an ASCII block of data in twenty byte chunks followed by a terminate byte if the end of data has been reached or an acknowledge byte if there is more data to send. The PC will send a six byte acknowledge command back to initiate the next chunk of twenty bytes or, in the case of a terminate byte from the meter, will end the transaction. The total length of the transaction is dependent on what features the meter is equipped with. The final chunk of data may have unused filler bytes to complete the chunk of twenty.



The output from ID_METER_ADC_SETT is in ASCII format and is composed of a list of configuration information required by the PC to work with different ADC settings. Each configuration option is separated by a semicolon and must contain LABEL, OVERANGE, and TABLE_HEADERS fields. Each of these subfields are separated by colons and these subfields are further separated by equal signs to denote the difference between the field label and the field value:

1:LABEL=label1:OVERANGE=value1:TABLE_HEADERS=headers1;

2:LABEL=label2:OVERANGE=value2:TABLE_HEADERS=headers2;

....

N:LABEL=labelN:OVERANGE=valueN:TABLE_HEADERS=headersN;

The variable VAR_ADC in meter settings is an integer value that corresponds to the first number in the configuration information that is currently in use by the meter. When VAR_ADC is set to a new number then the entire ADC settings block is parsed and the variable values for OVERANGE and TABLE_HEADERS are replaced with their new values.

B. Stream, Time, and Reset Commands

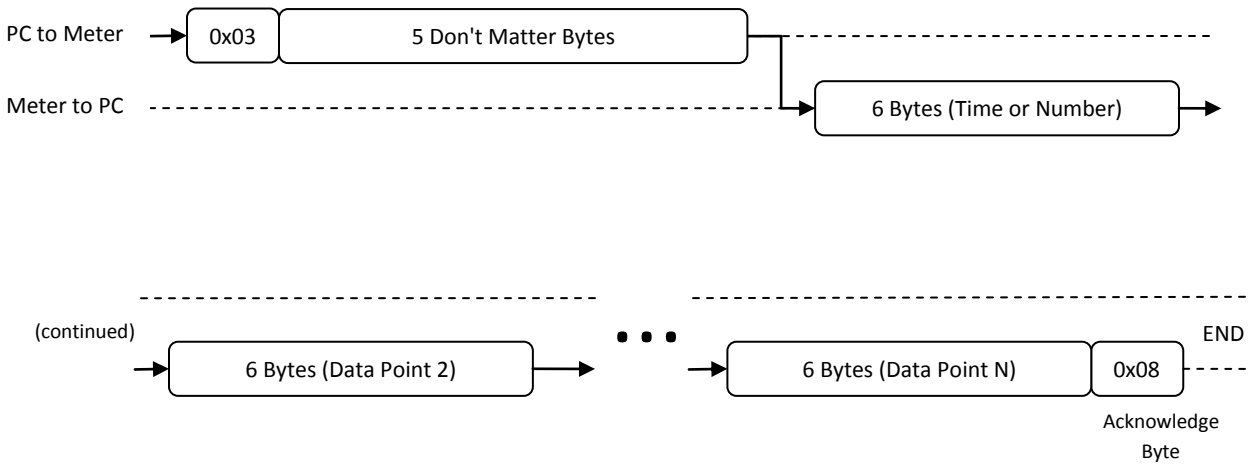
STREAM_DATA (0x03)

Streaming data is one of two methods for getting data from a meter onto a PC (the other is downloading from onboard storage). For each streaming command sent, the meter will reply with a single record that includes a time or numbered data point and the fields of data that that particular meter measures (or calculates). Each specific meter will return unique amounts of data within each record but each data point will always be six bytes in length. Therefore, the total length will be the number of fields provided

by the meters TABLE_HEADERS variable (comma delimited) multiplied by six to give the total number of bytes.

To initiate this command the meter is sent one command byte (0x03) followed by five bytes whose contents don't matter. The meter will receive the command and upon the next appropriate time base will compile and send a record that is composed of a number of bytes equal to the number of data points multiplied by six plus a final acknowledge byte.

NOTE: The final acknowledge byte signifies to the PC that the meter is ready to stream the next sample once the meter receives the next stream command. In some cases it would be important that the meter not add the final acknowledge byte and instead add a terminate byte so that the PC does not send multiple stream commands. This is evident when streaming both peakhold and DC data simultaneously and is the one exception to the "PC initiates all communications" architecture.



The output from STREAM_DATA has already been described as a block of bytes equal to the number of data points multiplied by six plus an acknowledge byte however each six byte data point must be further parsed to provide usable data. Six bytes is equal to 48 bits: the first 12 are used for configuration information, the next 4 are used for sign and decimal information, and the final 32 bits are the actual number.

Data Point Composition

Byte 1	x	I	F	F	H1	H2	C	x
Byte 2	x	x	x	x	S	D	D	D
Byte 3	N	N	N	N	N	N	N	N
Byte 4	N	N	N	N	N	N	N	N
Byte 5	N	N	N	N	N	N	N	N
Byte 6	N	N	N	N	N	N	N	N

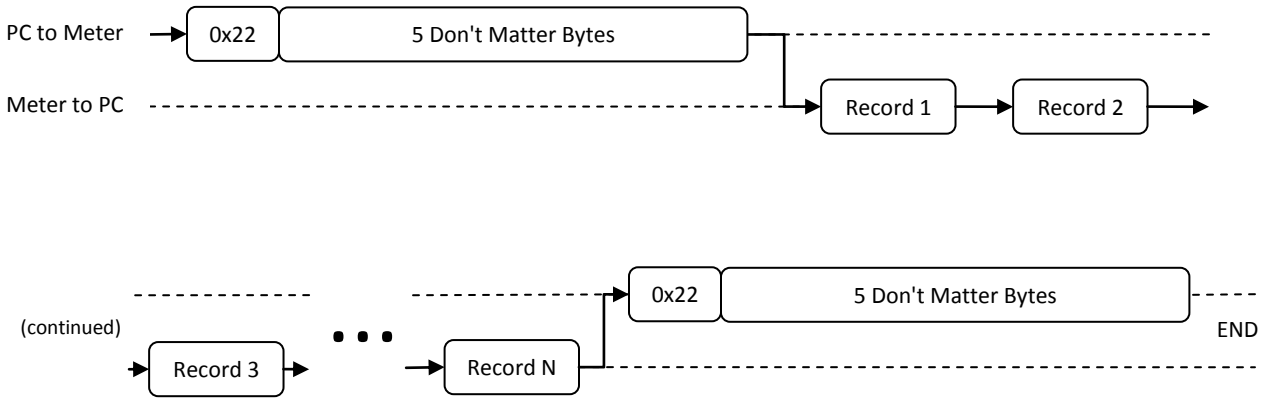
I	1 bit	Flag will show if current 6 byte sample is included in the record. In some designs (such as a 3-Axis meter measuring in 1-Axis mode) some data points will be null data. These data points should be read in to count the proper number of bytes received but should be excluded from data output on the PC. If the bit is 1 then the data is null and thus ignored; if the bit is 0 then it is parsed as usable data.
F	2 bits	These bits define field type of input data • 00 = DC Field • 01 = AC Field • 10 = PH Field • 11 =
H1	1 bit	Flag defines if data should be placed in continuous data recording stream. If bit is 1 then data is displayed; if 0 it is not placed in the continuous data stream (the data won't be recorded). This is primarily useful for samples meant for instantaneous data stream update only.
H2	1 bit	Flag defines if a record should be hidden from the instantaneous data stream. If bit is 1 then the data is not sent to the instantaneous data stream; if 0 then the data is.
C	1 bit	Flag is a real time warning when meter settings have been changed at the meter since the last record was sent. This signals PC to stop streaming and update settings. If bit is 1 then meter settings will need to be re-queried.
S	1 bit	This is the sign bit for the data point. If the bit is 1 then the data is negative; if the bit is 0 then the data is positive.
D	3 bit	These bits define the decimal place of the data point. The decimal is calculated by converting these 3 bits to an integer (0 - 7) and shifting the decimal to the left by that many places. • 000 = 0 (Shift decimal left 0 places or divide number by 10^0) • 001 = 1 (Shift decimal left 1 places or divide number by 10^1) • 010 = 2 (Shift decimal left 2 places or divide number by 10^2) • 011 = 3 (Shift decimal left 3 places or divide number by 10^3) • 100 = 4 (Shift decimal left 4 places or divide number by 10^4) • 101 = 5 (Shift decimal left 5 places or divide number by 10^5) • 110 = 6 (Shift decimal left 6 places or divide number by 10^6) • 111 = 7 (Shift decimal left 7 places or divide number by 10^7)
N	32 bits	These bits comprise the numerical value of the data point. They are a 32 bit unsigned integer. Byte 3 is the most significant byte and byte 6 is the least significant. Convert by using (Byte3 << 24) | (Byte 4 << 16) | (Byte 5 << 8) | (Byte 6)
x	6 bits	These bits are currently unused.

RESET_TIME (0x04)

This command returns exactly the same way as STREAM_DATA (0x03) does except that it resets the time data point (or the sample count data point depending on the meter) to zero. The only difference in execution is that the first bytes sent to the meter is 0x04 (instead of 0x03). Generally, this command is used once at the beginning of a streaming session followed by subsequent uses of the STREAM_DATA command until the PC ends the streaming session.

BURST_DATA (0x22)

This command returns data without regard to host response. The general time base is 250 ms however some designs may be capable of much faster streaming rates. Regardless of whether or not a meter can keep accurate time at periods smaller than 250 ms this command will stream data repeatedly without waiting for acknowledge handshakes.

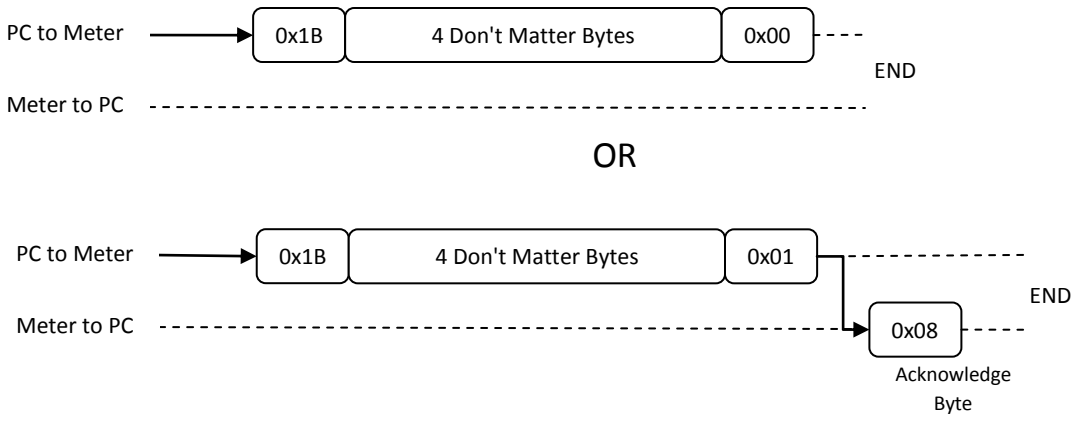


This command is in early development and this section should be expanded on as the command is implemented.

RESET_ZERO (0x1B)

This command is only relevant to meters that possess the REMOTE_ZERO tag within their meter properties list. When sent to the meter the meter will apply a zeroing function that removes all current offset from the record. Current AlphaApp software bypasses this function by maintaining meter offsets onboard the PC without consulting the meter. This was implemented because zeroing onboard the meter presented slow responses.

To initiate this command the meter is sent a one byte command (0x1B) followed by four bytes whose contents don't matter and one byte that is either 0x00 or 0x01 (SAVE_ZERO). If the final byte is 0x00 the meter does not send any response. If the final byte is 0x01 the meter sends an acknowledge byte back to the PC. In some designs (notably the ADC Stick) the 0x01 on the final byte can be used as an in house command to save the offset to onboard flash. The saving feature should not be used by the end user and is not built in to any software publicly released.



IDENT_SYS_TIME (0x06)

This command is unused in any current design it's intent was to provide clock access to onboard real time clocks in meters, however, we have not implemented this feature. When used it should return the onboard time to the PC in a yet to be determined format.

ALTER_SYS_TIME (0x0D)

This command is unused in any current design it's intent was to provide clock access to onboard real time clocks in meters, however, we have not implemented this feature. When used it should allow a controlling host to set the system time onboard a meter.

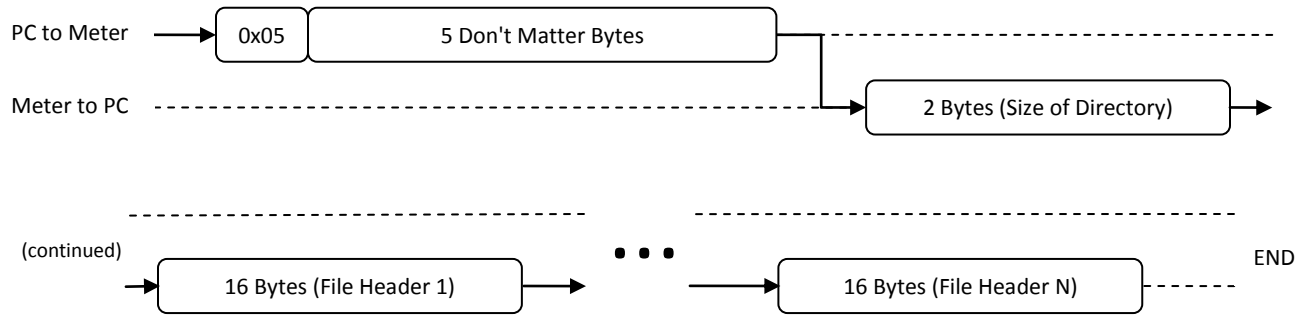
C. Flash Chip Access Commands

The flash chip access commands are only relevant to meters that possess onboard memory for remote storage. Meters equipped with onboard memory will have specific tags in their meter properties list as described in their command detail below.

FILE_DIRECTORY (0x05)

This command is only relevant for meters with the POSS_FILE_HEAD tag in their meter properties list. The output from this command is the entire file directory including all files that are currently being stored on the meter. When using this command most of the meter functions are suspended so that data transfers are not hampered by interruptions and timing delays. After using this function it is important to use BREAK_SUSPEND (0x14) to return the meter to its fully functional state.

To initiate this command the meter is sent a command byte (0x05) followed by five bytes whose contents don't matter. The meter will reply with two bytes that signify the size of the directory and then begin sending each 16 byte entry in the directory until all have been sent.



The data output of the FILE_DIRECTORY command is a binary formatted list of file headers. The format of each directory entry (file header) will include all pertinent details to describe each file so that host software can appropriately handle file downloads.

File Header Composition

	Byte 1	IV	T	T	T	T	T	T	T
	Byte 2	Y	Y	Y	Y	Y	Y	Y	Y
	Byte 3	Y	Y	Y	Mo	Mo	Mo	Mo	D
	Byte 4	D	D	D	D	H	H	H	H
	Byte 5	H	Mi	Mi	Mi	Mi	Mi	Mi	S
	Byte 6	S	S	S	S	S	N	N	N
	Byte 7	N	N	N	N	N	N	N	N
	Byte 8	N	N	Dt	Dt	Ft	Ft	Ft	x
	Byte 9	x	x	x	x	x	x	x	x
	Byte 10	x	x	x	x	P	P	P	P
	Byte 11	P	P	P	P	P	P	P	P
	Byte 12	P	P	P	P	A	A	A	A
	Byte 13	A	A	A	A	A	A	A	A
	Byte 14	A	A	A	A	A	A	A	A
	Byte 15	F	F	F	F	F	F	F	F
	Byte 16	F	F	F	F	F	F	F	F

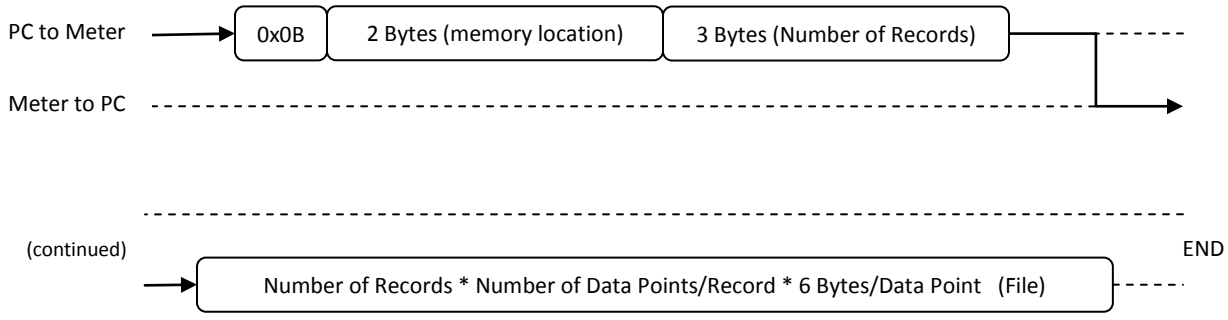
IV	1 bit	Flag will show if data is a valid file header.
T	7 bits	These bits define the header type of the file header. Currently, this field is not used but was planned so that users could change the labels on their data to something more meaningful to them. Only one option is populated in meters. • 0000001 = Numbered Data Sets
Y	11 bits	This field is a 4 digit numerical representation of years in Common Era (CE) format (i.e., 2014). Most significant bit is first.
Mo	4 bits	This field designates a numerical representation of months from 1 to 12 (Gregorian Calendar). Most significant bit is first.
D	5 bits	This field designates the day of the month (Gregorian Calendar). Most significant bit first.
H	5 bits	This field designates the hours in the day from 0 to 23 (24-Hour Clock format). Most significant bit first.
Mi	6 bits	This field designates the minutes in the hour from 0 to 59. Most significant bit first.
S	6 bits	This field designates the seconds in the minute from 0 to 59. Most significant bit first.
N	13 bits	This field is standard unsigned integer that designates the data set number. Most significant bit first.
DT	2 bits	This field designates the data type of the data in the file. • 00 = DC data type • 01 = AC data type • 10 = PH data type • 11 = No Data Type
FT	3 bits	This field designates the field type of the data in the file. It is used for meters that measure multiple field types or ranges. The value of the 3 bits is an integer conversion (most significant bit first) and corresponds to the ADC index that the unit is recorded in.
P	16 bits	This field designates the location in memory of the file (starting page number). Most significant bit first.
A	20 bits	This field designates the total amount of data in the file (number of records). Most significant bit first.
F	16 bits	This field designates the stream period at which the data in the file was recorded. Most significant bit first.
x	13 bits	These bits are unused

DOWNLOAD_FILES (0x0B)

This command is only relevant to meters that possess the POSS_FILE_HEAD tag in their meter properties list. When used this command downloads a file from the onboard memory to the host. Given the location in memory and the number of records sent to the meter the meter will then send a file back to

the PC from that location that is that length. When using this command most of the meter functions are suspended so that data transfers are not hampered by interruptions and timing delays. After using this function it is important to use BREAK_SUSPEND (0x14) to return the meter to its fully functional state.

To initiate this command the meter is sent one command byte (0x0B) followed by two address bytes and three number of records bytes. The address and number of records bytes correspond to fields in the file header for that file (See FILE_DIRECTORY section for details). The address is a 16 bit unsigned integer and is the same as the location in the memory file (denoted by P in the file header description) and the number of records is a 24 bit unsigned integer and is the same as the amount of data in the file (denoted by A in the file header description).

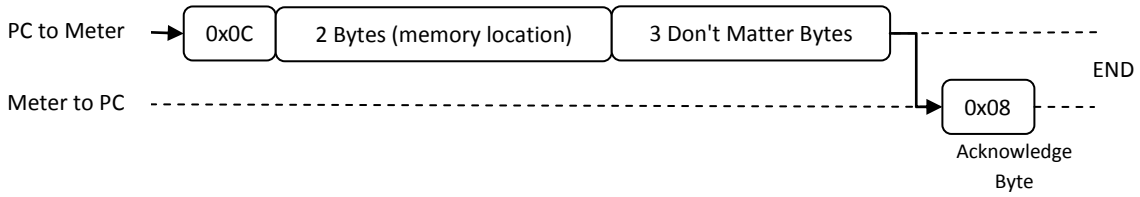


The output data is unbroken by any meter host handshaking and can be converted exactly as illustrated in the STREAM_DATA section.

DELETE_FILE (0x0C)

This command is only relevant to meters that possess the POSS_FILE_HEAD tag in their meter properties list. When used this command will delete a file header from the file directory. Note that this command does not delete the actual file, it only removes the file header so that the file can no longer be accessed.

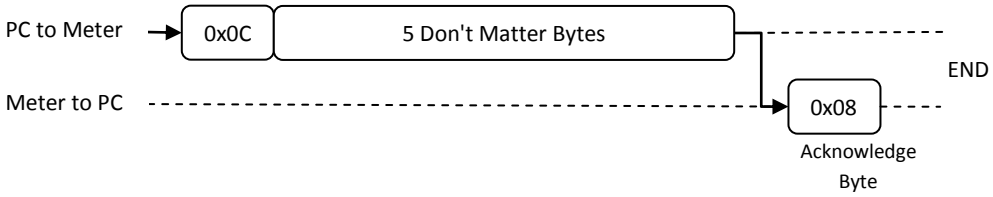
To initiate this command the meter is sent one command byte (0x0C) followed by two address bytes and three bytes whose contents don't matter. The meter will perform the delete action and respond with an acknowledge byte when the process is complete. The address is a 16 bit unsigned integer and is the same as the location in the memory file (denoted by P in the file header description - see FILE_DIRECTORY section for details).



DELETE_CHIP (0x13)

This command is only relevant to meters that possess the POSS_FILE_HEAD tag in their meter properties list. When used this command will delete the entire file directory in onboard memory. Note that this will not delete the actual file but will remove all the file headers so that those files can no longer be accessed. This will also reset all the pointers in the meter so that data writes begin from the beginning of memory. It is important to do this regularly otherwise the meter will continue writing to the end of memory and run out of room.

To initiate this command the meter is sent one command byte (0x13) followed by five bytes whose contents don't matter. The meter will perform the deletion process and respond with an acknowledge byte when the process is complete.



D. Flash Chip Access Commands (In-House ONLY)

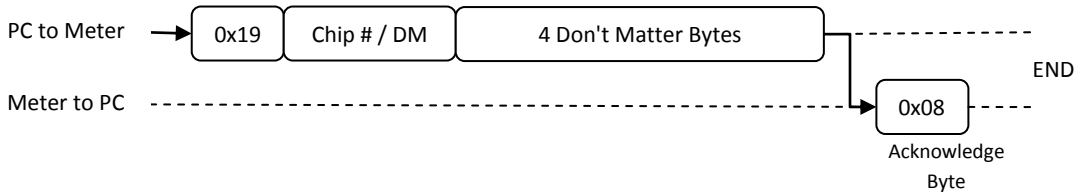
The commands in this section are flash chip access commands that are only used for testing and development of memory equipped meters. They are only relevant to meters with onboard memory and should not be accessible by end user applications.

NUKE_CHIP (0x19)

This command will delete the entire onboard memory, resetting every register.

To initiate this command the meter is sent one command byte (0x19) followed by one byte that can be used to designate which memory chip to delete and four bytes whose contents don't matter. In meters

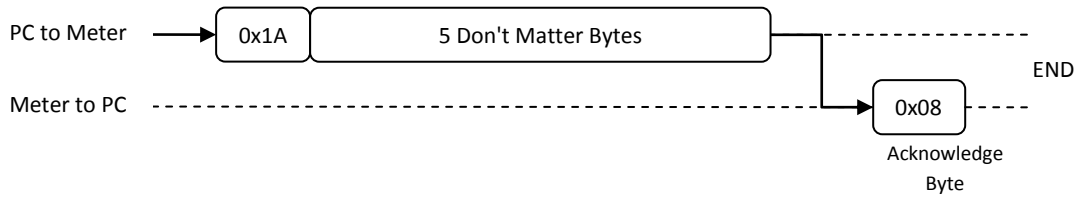
that contain more than one memory chip it's a simple integer designation (1,2,... etc.). In meters where there is no more than one memory chip the second byte becomes a don't matter byte.



REMOTE_STORE (0x1A)

This command is only relevant to meters that possess the POSS_FILE_HEAD tag in their meter properties list. When used this command will toggle the data capture on/off remotely so that the meter will store data to its onboard flash. Each execution of this command toggles only once so to turn the unit on and then off the command will need to be sent twice.

To initiate this command the meter is sent one command byte (0x1A) followed by five bytes whose contents don't matter. The meter will then send an acknowledgement byte in response.



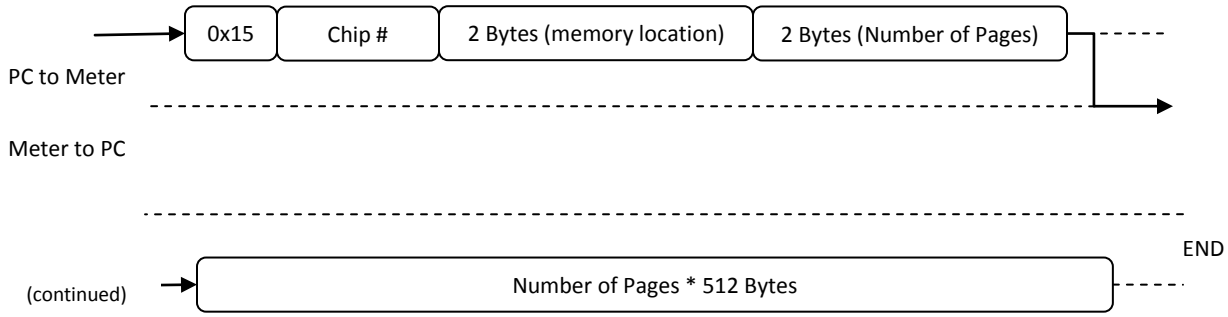
DUMP_CHIP_TO_PC (0x15)

This command downloads a number of predefined pages from the onboard memory chip. When using this command most of the meter functions are suspended so that data transfers are not hampered by interruptions and timing delays.

To initiate this command the meter is sent one command byte (0x15) followed by one memory chip designation byte, two bytes that designate the starting page, and two bytes that designate the number of pages to download. The memory chip designation is a simple integer reference (1,2,...etc.) and 0 is always reserved for the microcontroller. The starting page is a 16 bit unsigned integer that designates where in memory the read will begin. The number of pages to download is a 16 bit unsigned integer that

designates the number of 512 byte pages that will be downloaded. The meter responds by sending the corresponding data back to the PC.

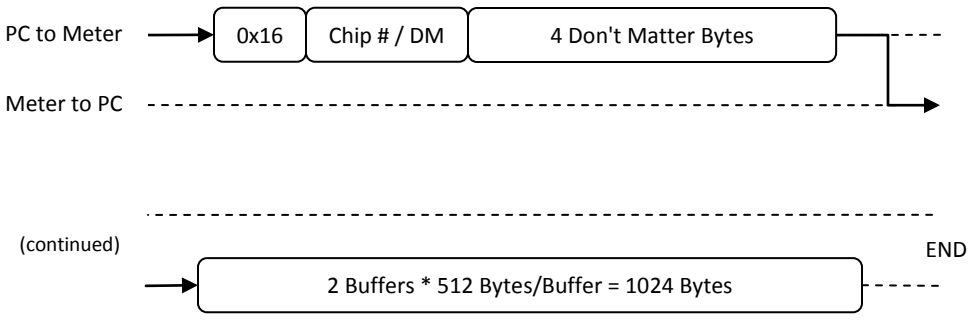
NOTE: This command does not have complete backwards compatibility but is only used in-house so it doesn't negatively impact AlphaApp.



DUMP_BUFF_TO_PC (0x16)

This command downloads the buffers from the onboard memory of the meter. When using this command most of the meter functions are suspended so that data transfers are not hampered by interruptions and timing delays.

To initiate this command the meter is sent one command byte (0x16) followed by one memory chip designation byte and four bytes whose contents don't matter. The memory chip designation is a simple integer reference (1,2,...etc.) and becomes a don't matter byte in meters that have only one memory chip onboard. The meter responds to the command by sending the contents of two 512 byte buffers back to the host with no handshaking or acknowledgement.



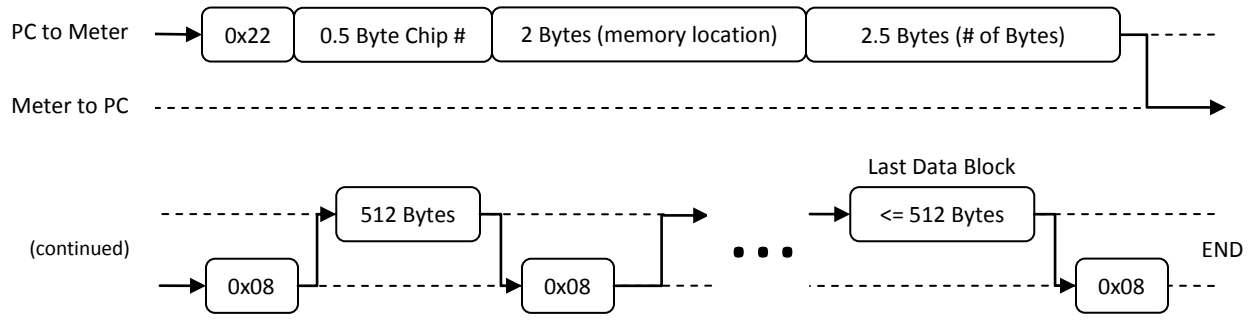
DUMP_DATA_TO_CHIP (0x22)

This command allows the host to load large amounts of data to the onboard memory of a meter. When using this command most of the meter functions are suspended so that data transfers are not hampered by interruptions or timing delays.

To initiate this command the meter is sent one command byte (0x22) followed by one half byte for the memory chip designation, two bytes that designate the starting page, and two and a half bytes that designate the number of bytes that are being uploaded. Then, to signal ready for the next block (equal to or less than 512 bytes) the meter will respond with an additional acknowledge byte. The meter will continue sending acknowledge bytes to receive the next block until all blocks have been received. Note that the last block does not have to be the full 512 bytes in length.

The half byte holding the chip identification information is a 4 bit integer (0 - 15). The 0th chip id is always reserved for the meter microcontroller to allow for external flash programming. Subsequent numbers in this half byte can be assigned to additional memory chips as desired. When data is being written to the meter then the meter will write that data to the memory target that corresponds to the chip id.

NOTE: This command does not have complete backwards compatibility.



D. General Communication Commands

TERMINATE (0x07)

This is not a full command. It is a response to action used in various functions. The value 0x07 is reserved from use as a command to avoid confusion and backwards compatibility problems.

ACKNOWLEDGE (0x08)

This is not a full command. It is a response to action used in various functions. The value 0x08 is reserved from use as a command to avoid confusion and backwards compatibility problems.

CRC_ACKNO (0x09)

This is not a full command. It is a response to action planned for future implementations but has not been deployed. The value 0x09 is reserved from use as a command to avoid confusion and backwards compatibility problems.

CRC_RESEND (0x0A)

This is not a full command. It is a response to action planned for future implementations but has not been deployed. The value 0x0A is reserved from use as a command to avoid confusion and backwards compatibility problems.

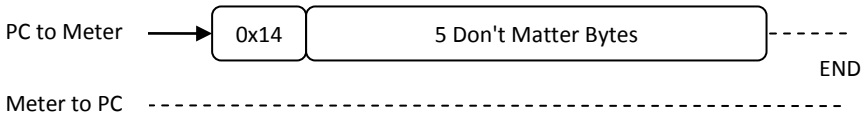
NULL_ACTION (0x00)

This is not a full command. It is a response to action used in various functions. The value 0x00 is reserved from use as a command to avoid confusion and backwards compatibility problems.

BREAK_SUSPEND (0x14)

This command will return a meter from program suspension. This is necessary after downloading large amounts of data from the meter.

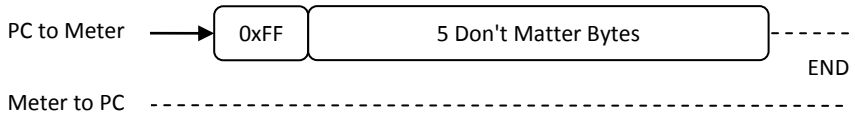
To initiate this command the meter is sent one command byte (0x14) followed by five bytes whose contents don't matter (DM). There is no response from the meter.



KILL_ALL_PROCESS (0xFF)

This command provides a means of shutting down all current processes that the meter is currently running. It immediately ends all data transmission processes and clears the buffers.

To initiate this command the meter is sent one command byte (0xFF) followed by five bytes whose contents don't matter (DM). There is no response from the meter.

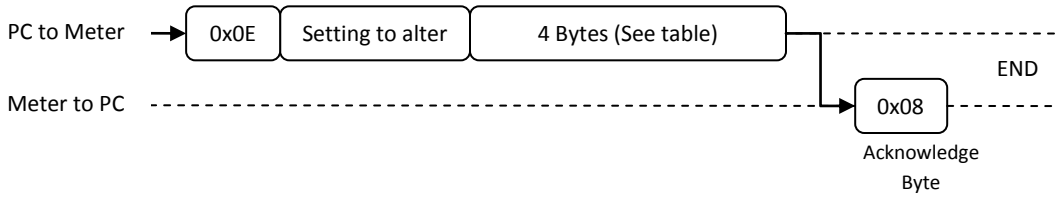


E. Setting Alteration Commands

ALTER_METER_SETT (0x0E)

This command is relevant to all meters that do NOT possess the NO_SETTINGS tag in their meter properties list. When used, this command allows the PC to manipulate the meter settings onboard the meter. All settings that the meter has available are alterable using this command.

To initiate this command the meter is sent one command byte (0x0E) followed by a register byte that designates which setting is being changed and four bytes that are determined by the details of each setting. After altering the desired setting the meter then responds with an acknowledge byte.



Setting Alteration Composition

Setting to alter		Byte 3	Byte 4	Byte 5	Byte 6
ALTER_STREAM_PERIOD	0x0F	Doesn't Matter	Doesn't Matter	Stream period	
		Stream period is a 16-bit unsigned integer value that must be one of the options given by the AVBL_FREQS tag in the meter properties list.			
ALTER_FILE_HEAD_TYPE	0x10	Doesn't Matter	Doesn't Matter	Doesn't Matter	File header type
		File header type is an 8-bit unsigned integer value of 1's and 0's that must adhere to the bit flags given by the POSS_FILE_HEAD tag in the meter properties list.			
ALTER_ADC_INDEX	0x12	Doesn't Matter	Doesn't Matter	Doesn't Matter	ADC index
		VAR_ADC_SETT is a flag found in the meter properties list of certain meters. This is an 8-bit unsigned integer value that provides an index to what internal ADC settings the meter should be operating with. The index will correspond to the meter ADC properties list (see ID_METER_ADC_SETT)			
ALTER_RECORD_CHANGE	0x17	Doesn't Matter	Doesn't Matter	Doesn't Matter	0x00 or 0x01
		REC_CHG is a flag found in the meter properties list of certain meters. This is an on/off variable flag that can be turned on with 0x01 and off with 0x00			
ALTER_DC_PH_COMB	0x18	Doesn't Matter	Doesn't Matter	Doesn't Matter	0x00 or 0x01
		DC_WITH_PH is a flag found in the meter properties list of certain meters. This is an on/off variable flag that can be turned on with 0x01 and off with 0x00			

F. Digital Potentiometer Calibration Commands

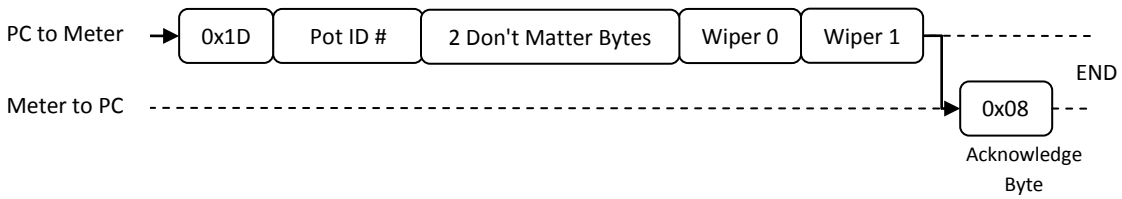
These command are only relevant to meters who possess onboard digital pots that require in factory calibration. These meters will have the VAR_ADC_SETT tag in their meter properties list. These meters will also have specific tags that describe a certain calibration routine to be used (i.e., ADC_CAL, ADC6_CAL, etc.). The only exception to this is the USB-DL1 which does not have a unique tag to identify the calibration routine to use. The commands in this section should NOT be available to the end user.

SET_DIGITAL_POTS (0x1D)

This command will send digital potentiometer values to the meter. These values will be set on board the meter and then the resulting value can be measured using the STREAM_DATA command.

To initiate this command the meter is sent one command byte (0x1D) followed by a digital pot identification byte, two bytes whose contents don't matter, and two bytes that are the value of the digital pot wipers. In designs where only one onboard digital pot exists the digital pot identification byte

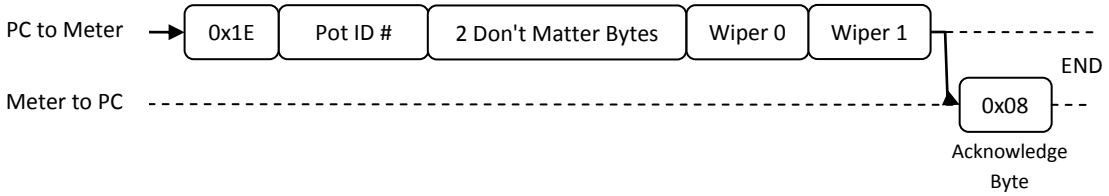
becomes a don't matter byte. The digital pot identification byte is an 8-bit unsigned integer that corresponds to the number (1,2,...etc.) of digital pot onboard the meter. The fifth and sixth bytes are 8-bit unsigned integers that are the wiper values to set each digital pot channel to (wiper 0 corresponds to the fifth byte and wiper 1 corresponds to the sixth byte). Numbers expected are 0 to 128; 0 being the least resistance in the pot and 128 being the most. The meter responds with an acknowledge byte.



SAVE_DIGITAL_POTS (0x1E)

This command will save wiper settings to the RAM of the meter saving them temporarily while any other processes are completed. These settings will be lost if the meter loses power (for permanent saving see FLSH_WRT_DIG_POTS).

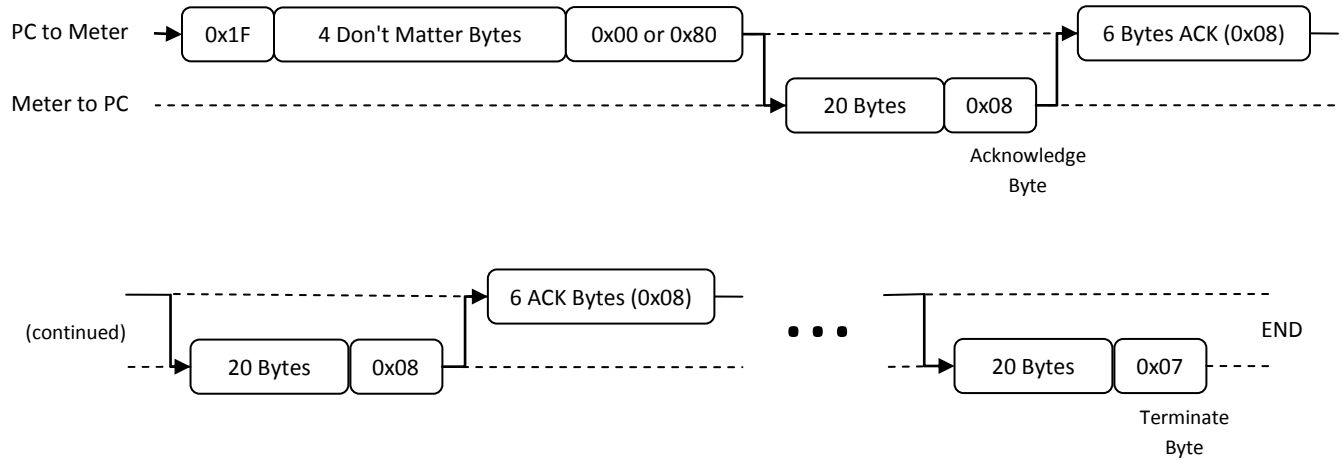
To initiate this command the meter is sent one command byte (0x1E) followed by a digital pot identification byte, two bytes whose contents don't matter, and two bytes that are the value of the digital pot wipers. In designs where only one onboard digital pot exists the digital pot identification byte becomes a don't matter byte. The digital pot identification byte is an 8-bit unsigned integer that corresponds to the number (1,2,...etc.) of digital pot onboard the meter. The fifth and sixth bytes are 8-bit unsigned integers that are the wiper values to set each digital pot channel to (wiper 0 corresponds to the fifth byte and wiper 1 corresponds to the sixth byte). Numbers expected are 0 to 128; 0 being the least resistance in the wiper and 128 being the most. The meter responds with an acknowledge byte.



READ_DIGITAL_POTS (0x1F)

This command will return the meter settings from the meter. There are two forms of this command, in the first the meter returns only the current settings of all the pot(s) in the meter, in the second the meter returns all wiper values saved to the meters' onboard flash.

To initiate this command the meter is sent a command byte (0x1F) followed by four bytes whose contents don't matter and a sixth byte that defines how the command should work. To return all the wiper settings from permanent memory the sixth byte should be 0x80. To return only the current wiper settings the sixth byte should be 0x00.



The output of this command will be in ASCII format and can be separated however the designer deems fit. The method should be selected to best suit the design but commas should separate wiper values and vertical bars should be used to separate digital pots, different ADC settings, etc. The final 20 byte block may have filler bytes in it to round out the total transmission. These filler bytes should be semi-colons.

FLSH_WRT_DIG_POTS (0x20)

This command will instruct the meter to write ALL wiper settings currently in RAM directly to flash so that the meter can use them indefinitely. Upon every subsequent startup the meter will load these wiper values and set the digital pots accordingly. Note that wiper values must be saved to RAM using SAVE_DIGITAL_POTS first before using this command.

To initiate this command the meter is sent a command byte (0x20) followed by five bytes whose contents don't matter. The meter will respond with an acknowledge byte.