



OPERATING MANUAL
for
QUARTZDYNE®
Digital Pressure Transducer

Quartzdyne, Inc.

A DOVER COMPANY

www.quartzdyne.com

Quartzdyne reserves the right to change specifications without notice.

QUARTZDYNE, the Crystal Logo, and DOVER are Registered Trademarks of Dover Corporation and Affiliates



Quartzdyne, Inc.
Digital Pressure Transducer Operating Manual
 2-Wire I²C Serial Interface with Frequency Counter and Coefficient Storage

Revision History

FPGA/ASIC ID	BCF ID	Date	Comments
0D020102	N/A	April 14, 2001	Initial Preliminary Release
0D020103	0D010100	August 24, 2001	Fix overflow problem when gate time > 1.3 seconds Add calibration coefficient sections
0D020201	0D010100	January 3, 2002	Initial Production Release Two-byte addressing for 64k EEPROM Change status bit labels from "Valid" to "Detect" Add Write Protect control for EEPROM Coefficient files provided in MSB first format only (0D01)
0D020201	0D010100	March 19, 2001	Clarify dummy read/trigger command
0D020201	0D010100	December 15, 2003	Add Connector Pin Assignments. Clarify EEPROM Protocol. Describe EEPROM Hung Bus bug. Include general transducer information
0D020301 0D050301	0D010100	February 14, 2006	Change start-up sequence to avoid "stuck status bit" condition.
0D020302 0D050302		July 21, 2006 April 10, 2007 July 26, 2007	Improved power on reset in FPGA Add Section 3.4 concerning sticky SDA line (power-on reset warning) Add Sections 3.5 and 3.6 - recommended start-up procedures. Increase maximum current specifications, Clarify Status/FPGA ID addressing
0D050303		September 1, 2007 February, 23 2009	Reduce startup time with ASIC implementation Remove Electrical Specs/add link to Specs
0D090402		April 16, 2009	Updated to cover digital ASIC 4.02 enhancements. Checksum, 4 byte Control Word, 7.2 MHz REF_OUT option, and A1/T and A2/P as outputs.
		Nov 16, 2009	Include 7-pin universal cabling with reference out EEPROM Warning
		January, 2010	Describe redundant coefficient storage.
		August, 2010	Fix defaults and description for 7.2MHz and A1/T and A2/P pins
0D090403		July, 2011	New EEPROM protocol.
		May, 2012	Minor clean-up
		May, 2017	Alternate color B/W (black/white striped) added for GND. DHB103/104 addressing

Quartzdyne Digital Pressure Transducer Operating Manual

Table of Contents

- 1 GENERAL INFORMATION 4**
 - 1.1 INTRODUCTION4**
 - 1.2 PRESSURE SENSOR DESIGN.....5**
 - 1.3 BELLWS AND DIAPHRAGMS5**
 - 1.4 ASIC KICK-START CIRCUIT.....5**
 - 1.5 SENSOR OUTPUT CHANGES WITH PRESSURE AND TEMPERATURE.....5**

- 2 INTERFACE REQUIREMENTS 7**
 - 2.1 ELECTRICAL CONNECTIONS7**
 - 2.2 ADDRESSING.....8**
 - 2.3 SIGNALING.....8**
 - 2.4 POWER-ON RESET WARNINGS9**
 - 2.5 POWER-ON SEQUENCING9**
 - 2.6 SUGGESTIONS FOR ROBUST I²C COMMUNICATIONS: 10**

- 3 FREQUENCY COUNTER 11**
 - 3.1 READ COUNTER COMMAND.....11**
 - 3.2 CHECKSUM13**
 - 3.3 WRITE CONTROL/STATUS REGISTERS14**
 - 3.4 QUERY STATUS OR CHIP ID 16**

- 4 SERIAL EEPROM 17**
 - 4.1 EEPROM READ - CURRENT ADDRESS17**
 - 4.2 EEPROM READ - SPECIFIC ADDRESS.....17**
 - 4.3 EEPROM WRITE18**
 - 4.4 DATA RETENTION 19**

- 5 CALIBRATION 20**
 - 5.1 TRANSDUCER CALIBRATION DATA.....20**
 - 5.2 TRACEABILITY OF CALIBRATION20**
 - 5.3 TYPICAL VALUES OF CORRECTIONS USED IN CALIBRATION21**
 - 5.4 PRESSURE CONTROL REQUIRED FOR CALIBRATION21**
 - 5.5 TEMPERATURE CONTROL REQUIRED FOR CALIBRATION22**

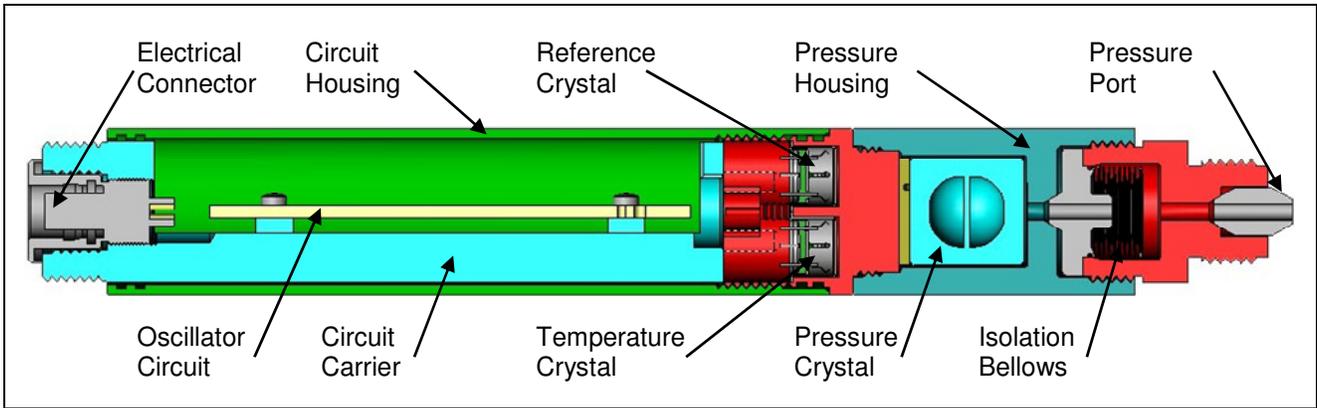
- 6 CALIBRATION COEFFICIENTS 23**
 - 6.1 HEX COEFFICIENT FORMAT23**
 - 6.2 CHECKSUM CALCULATION23**
 - 6.3 PRESSURE AND TEMPERATURE CALCULATIONS.....24**
 - 6.4 EXAMPLE CALCULATION ROUTINE PSEUDO-CODE25**
 - 6.5 INTEL HEX FILE CODING.....26**

1 General Information

1.1 Introduction

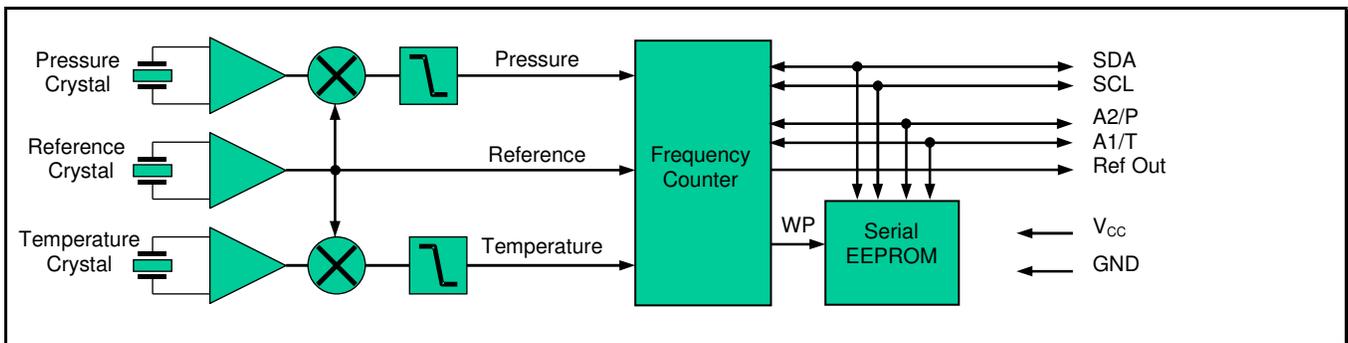
All Quartzdyne Pressure Transducers contain three quartz crystal sensor elements. The first of these is sensitive primarily to exposed pressure, the second responds to temperature, and the third has minimal sensitivity to either pressure or temperature. The crystals are arranged mechanically to provide good thermal coupling. The quartz sensing elements provide high stability and extremely fine resolution for sensing pressure. A bellows is used to protect the pressure crystal from the process fluids. A circuit provides stimulus for the quartz sensing elements, and the corrosion-resistant, high-strength alloy housing provides mechanical support and protection to each of these elements. Various mechanical configurations are available.

Figure 1. Parts of a Quartzdyne Pressure Transducer (Model DMB shown).



The Digital Transducer Circuit contains three quartz crystal oscillators (Pressure, Temperature, and Reference), mixers, a frequency counter, and a serial EEPROM (Figure 2). The pressure and temperature signals are mixed with the reference signal to improve resolution in the counter. The frequency counter outputs two numbers: the ratio of the mixed Pressure Frequency to Reference Frequency and the ratio of the mixed Temperature Frequency to Reference Frequency. Pressure and Temperature are calculated by the end user from these ratios using coefficients stored in the EEPROM. Communication is via an I²C compatible 2-wire serial interface. See our website (www.quartzdyne.com) for detailed Electrical Specifications.

Figure 2. Digital Pressure Transducer Block Diagram



For a complete measurement system, the user must provide power and a means for calculating pressure from the measured frequencies and provided coefficients. Storage and/or transmission of the data must also be considered. The Quartzdyne QCOM provides a USB interface to a PC in a laboratory environment.

1.2 Pressure Sensor Design

The important advantages of the Quartzdyne thickness-shear pressure technology are precision, long-term stability, ruggedness, and rapid transient response.

The pressure sensor is a quartz resonator that changes frequency in response to applied pressure. Crystalline quartz is an ideal material for precision sensors because it is perfectly elastic. The design of the Quartzdyne quartz pressure sensor retains the repeatable performance inherent in the single-crystal material.

The quartz pressure sensor is a thick-walled, hollow cylinder with closed ends. A thickness-shear mode resonator divides the central portion of the hollow cylinder. Fluid pressure on the exterior walls hydrostatically compresses the quartz cylinder, producing internal compressive stress in the resonator. The frequency of the resonator changes in response to the internal stress.

1.3 Bellows and Diaphragms

The transducer may be fitted with either a bellows or diaphragm which isolates and protects the pressure crystal from the fluids being measured. The effect of the bellows on the pressure reading is small, but must be compensated for in the calibration. Removal or physical damage to the bellows or diaphragm will change the pressure response and necessitate recalibration of the transducer.

Although the bellows is backfilled under vacuum with oil which has been thoroughly degassed, it is possible that small amounts of gas may be trapped inside the bellows. Normally this gas will be dissolved in the oil and have no measureable effect on pressure performance. At ambient pressures it is possible that some of the gas may be driven out of solution temporarily causing the bellows to expand resulting in a pressure reading as much as 10 psi higher than ambient pressure. The unit will return to normal pressure readings as soon as pressure is applied. Quartzdyne pressure transducers with bellows are specified for accuracy only above 200 psia.

1.4 ASIC Kick-start Circuit

Any contaminant or imperfection on the surface of a quartz resonator can cause its impedance to increase. Usually, the impedance increase will vary depending on the crystal drive level, and is often worse at lower drive levels. This phenomenon is known in the industry as drive level sensitivity (DLS) and can cause a crystal oscillator to fail to start up when power is supplied. While we take every effort to ensure that the crystals included in Quartzdyne pressure transducers are free from such imperfections, it is possible that a contaminant can move in response to shock experienced in the field.

With the introduction of the ASIC oscillator to our hybrid circuits (September 2007), a kick-start feature was added. This feature applies an increased drive level to the crystals during the start-up phase in order to overcome any potential DLS, and then backs off once stable oscillation has been achieved. This significantly reduces the chance that a DLS crystal will cause a field failure. In many cases, the excess drive literally moves the offending contaminant to an area where it no longer affects performance.

A possible, but rare side-effect of this feature is that if a crystal fails for gradually increasing high impedance that is not related to DLS, it may reach a point where the kick-start feature repeatedly engages and disengages. This will be seen as abnormally high jitter in the pressure and/or temperature measurement. If the impedance continues to increase, the jitter will stop as the kick-start feature stays engaged fully, and eventually the oscillator will flat-line. Non-ASIC circuits would have flat-lined long before any of these symptoms were manifested.

1.5 Sensor Output Changes with Pressure and Temperature

Figure 3 **Error! Reference source not found.** through **Figure 6** **Error! Reference source not found.** depict the typical change in the output counts of a DMB20K digital transducer with changes in pressure and temperature. **Error! Reference source not found.** **Figure 3** shows the nominal change in Temperature Output versus temperature. All temperature sensors show very similar sensitivities, but the counts at 25°C will vary from unit to unit. **Figure 4** **Error! Reference source not found.** shows the change in pressure output counts with changes in pressure and temperature. Note that changes in temperature produce changes in both zero and span.

The data from **Error! Reference source not found.** has been re-plotted in **Error! Reference source not found.** for additional examination. In **Figure 5** **Error! Reference source not found.**, the pressure output counts are plotted versus temperature at various pressures. The data has also been normalized at the minimum frequency so that

the temperature sensitivity can be observed more clearly. Note that the temperature sensitivity is positive at ambient pressure, but negative at full-scale pressure, with a zero sensitivity point that occurs at higher temperatures as the pressure is increased.

The transducer has been designed to minimize transient errors in down-hole work. Care has been taken to provide good thermal coupling between the pressure and temperature crystals. In a uniform and static temperature environment, the two crystals will be at the same temperature and the output will be fully compensated. Under transient conditions (temperature ramps, or large pressure steps), a small temperature difference between the two sensors is possible. **Figure 6** Error! Reference source not found. shows the nominal sensitivity of the pressure sensor due to temperature differences that may occur during a transient condition. If the transducer is used in the typical down-hole pressure and temperature ranges where the sensitivity is in the zero to 10psi/°C range, the transient errors will be small. Conversely, if used at the extremes of high temperature - low pressure, or high pressure - low temperature, the transient errors will be more significant. See our web site (www.quartzdyne.com) for a detailed transient report comparing the performance of various pressure transducer configurations.

Figure 3. Typical Temperature Sensor Output as a function of Temperature.

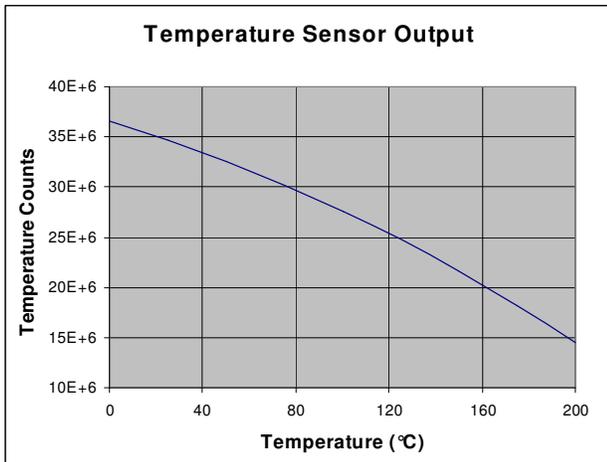


Figure 4. Typical Pressure Sensor Output at various temperatures

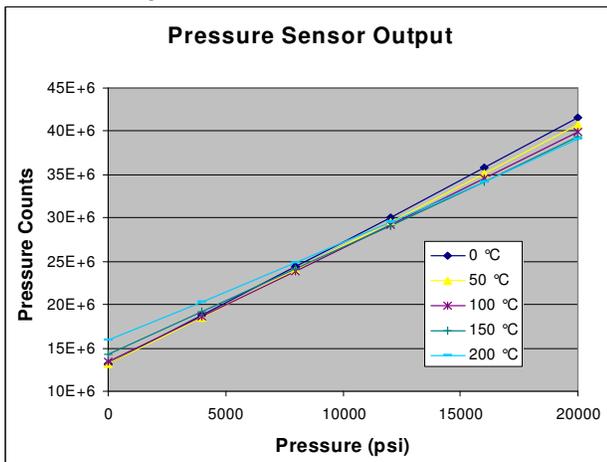


Figure 5. Pressure Sensor Output plotted vs. Temperature and normalized to the minimum (zero - sensitivity) point for each pressure

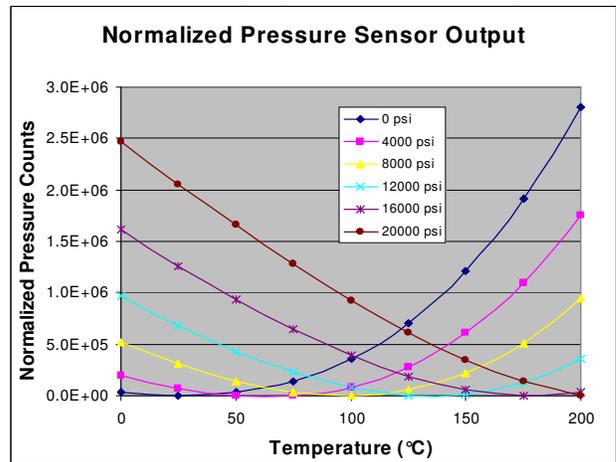
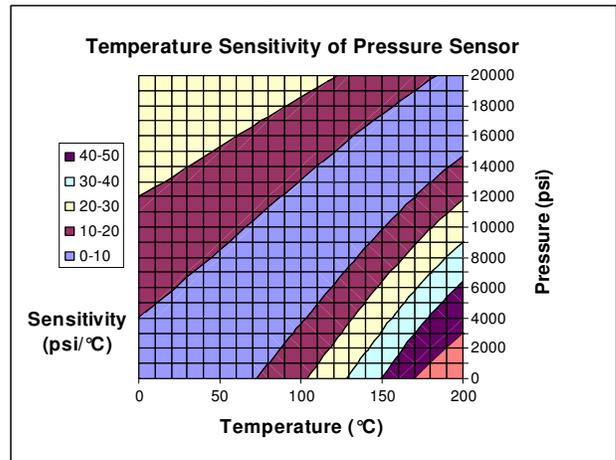


Figure 6. Pressure Sensor Temperature sensitivity gradient plot showing low sensitivity along a typical well gradient. Transient errors will be worse in ranges with high sensitivity



2 Interface Requirements

2.1 Electrical Connections

Each device requires power (2.7 to 5.5 V_{DC}) and two signaling lines SDA and SCL. A1/T, A2/P and R may also be available, but should be connected only if used. The signaling lines are open collector, and should be tied to V_{IN} on the master board using a suitable pull-up resistor. Lower values of R_P are required for longer cables, and higher clock speeds. A1/T and A2/P have weak pull-ups and are susceptible to noise when not connected. R contains a 7.2MHz high-frequency signal, and should be isolated from the other lines to prevent cross-talk. The length of these lines should be minimized (<12" recommended) to avoid noise problems and to minimize power supply current through R_P. A 2-twisted pair cable is recommended, with SDA and V_{IN} twisted together, and SCL and GND twisted together. A 7-conductor cable with an overall shield is recommended when the optional lines are to be used. V_{IN} and GND should be bypassed with a 0.1µF capacitor at the master end of the cable. GND is connected internally to the chassis. Shielding may be required in electrically noisy environments to prevent interference on the communication lines. See Figure 7 for connector pin assignments. Table 1 shows wire colors for units provided with flying leads.

Figure 7. Pin Assignments for Various Digital Transducer Connector Options.

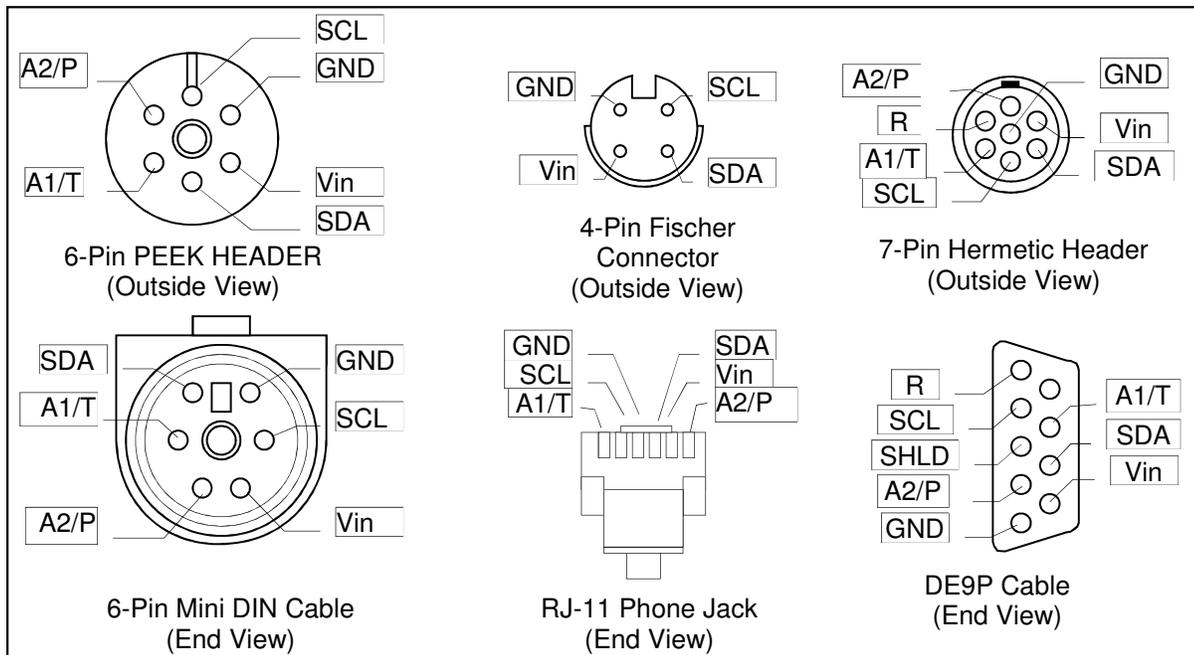


Table 1. Pin and wire labels and their functions

Label	Function	Wire Color (Digital ASIC)	Alternate Colors (FPGA)
Vin	Power Supply (Pos)	Blue	Red
GND	Power Supply (Neg)/Chassis	Black or B/W	Black
SHLD	Shield Termination/Chassis	-	-
R	Reference Frequency Output	White	-
A2/P	Address 2 / Pressure Frequency Output	Purple	Blue
A1/T	Address 1 / Temperature Frequency Output	Yellow	Orange
SCL	SCL (Serial Clock)	Slate	White
SDA	SDA (Serial Data)	Green	Green

2.2 Addressing

Up to four transducers may be wired together in a single system. The individual transducers are distinguished using the A2/P and A1/T inputs. If connected to GND, these are interpreted as '0'; if left floating, A2/P and A1/T are interpreted as '1'. **Warning!** Do not connect these inputs to V_{CC}, as this may cause internal device damage. For multiple transducer systems, A1/T and A2/P must be connected appropriately ("00", "01", "10", "11") to provide a unique address for each device. The device addresses are shown in **Table 2**. For transducers with four pin connectors, the address lines A2/P and A1/T are accessible inside the transducer. These are normally open when shipped (address "11"). DHB103/104 transducers will only have A1/T available for address modification i.e. transducers are shipped with a default address "11", where the user has the option to change the A1/T bit to "0", resulting in an address of "10".

Carefully consider the reliability implications when connecting multiple transducers on the same SDA/SCL lines as a single failure may disable communications for all interconnected devices.

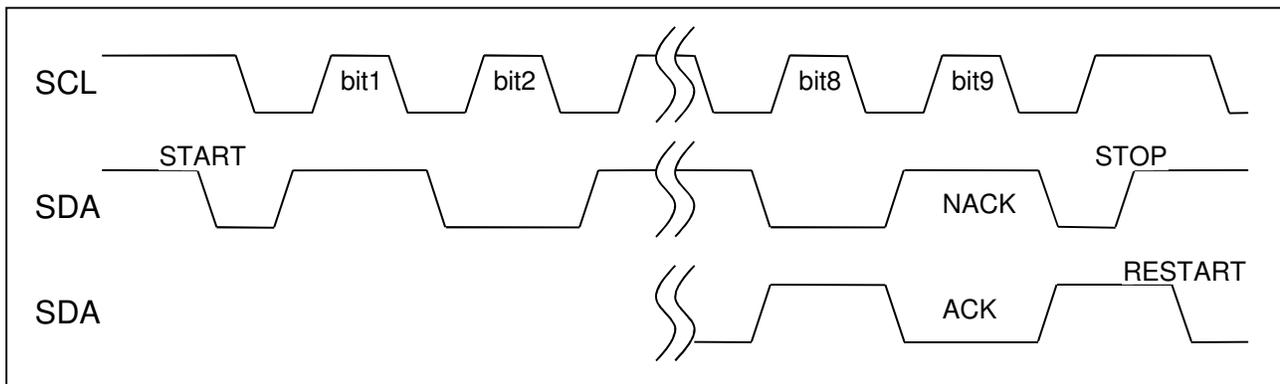
Table 2. Device Addresses

	MSB							LSB
Frequency Counter	1	0	0	1	A2	A1	T/P	R/W
Status and Control Registers	1	0	0	1	A2	A1	Status/ID	R/W
EEPROM	1	0	1	0	A2	A1	0	R/W

2.3 Signaling

The transducer has been designed for compatibility with the NXP I²C interface. The SDA and SCL lines are normally pulled high by a resistor on the master device. Data transitions of SDA are allowed only while SCL is low. A START condition is signaled by the master asserting SDA low while SCL is high, and a STOP condition is signaled by a low to high transition of SDA while SCL is high. Data is transferred eight bits at a time, followed by an ACK (SDA low) or NACK (SDA high) which is generated by the receiving device. A message consists of a START condition followed by an Address Byte (MSB first) generated by the master, followed by an ACK generated by the slave. If the LSB of the Address Byte is low (WRITE), the master continues to send data, the slave generating ACK as appropriate, until the end of message is indicated by a STOP condition generated by the master. If the LSB bit of the Address Byte is high (READ), the bus changes direction, and further data is supplied by the slave, with ACK generated by the master. The master indicates receipt of the last byte with a NACK followed by a STOP condition.

Figure 8. I2C bus timing



2.4 Power-on Reset Warnings

Some FPGA and EEPROM devices contained in transducers do not come out of power-on reset gracefully. Two separate problems have been observed.

Transducers with FPGA ID's lower than 3.02 will occasionally hold SDA low during a Write Control, or Read FPGA ID command (3.4) immediately following a power-up condition. This condition has been remedied in versions 3.02 and higher. When this occurs the slave holds the SDA line low, preventing further communication. The master can clear this condition by issuing 9 clock pulses while allowing the SDA line to float, followed by a STOP bit. This clears all devices on the I²C bus, and is applicable any time the master observes a stuck SDA line.

A similar problem existed in the EEPROM in versions 4.02 and earlier. A Read-Specific Address command (**Error! Reference source not found.**) issued after a power-on reset will cause some devices to latch the SDA line low at certain temperatures, preventing further communication. Unfortunately, this condition is not cleared by issuing 9 clock pulses. The only remedy is to cycle the power. This problem can be avoided by issuing a Read-Current Address command (4.1) prior to sending any other commands to the EEPROM.

Both of the above problems have been solved in versions 4.03 and higher, but caution is warranted when designing systems that may communicate with older devices.

2.5 Power-on Sequencing

To assure reliable data at power-on, the following **minimum wait-time** sequence is recommended:

1. Apply power to devices.
2. Wait at least 0.1 seconds for internal power supply to stabilize.
3. Issue an EEPROM Read-Current Address command if the EEPROM is to be read anytime during this power cycle (1 byte is sufficient – not necessary for Versions 4.03 and higher).
4. Upload coefficients if needed. **WARNING:** Verify the coefficient checksum before using coefficients to calculate P/T.
5. Query Pressure and Temperature counters successively until both respond with ACK. (To avoid possible start-up anomalies you may choose to wait an additional 0.1 seconds after both counters respond and then re-trigger.)
6. Wait desired gate time.
7. Read Pressure and Temperature counters and repeat 6-7 as desired.

For systems where **simplicity** is preferred over minimum start-up time:

1. Apply power to devices.
2. Wait 1.0 seconds for the internal power supply to stabilize, for the frequencies to come up, and in addition allow for a reasonable gate time.
3. Issue an EEPROM Random Read Command if the EEPROM is to be read anytime during this power cycle (1 byte is sufficient – not necessary for Versions 4.03 and higher).
4. Upload coefficients if needed. **WARNING:** Verify the coefficient checksum before using coefficients to calculate P/T.
5. Query Pressure and Temperature counters – both should be ready with good data. (First gate time will vary depending on startup time.)
6. Wait desired gate time and repeat step 5-6 as desired.

2.6 Suggestions for Robust I²C Communications:

1. Assure that SDA and SCL lines are both high before beginning communications
2. Wait at least 2.5uS (one quarter of 100 Khz SCL clock) between each transition of either SDA or SCL - Do not change SDA and SCL simultaneously!
3. Verify that the desired data actually takes before moving on (set a reasonable timeout period).
4. If conflicts are detected (timeout exceeded), issue 9 clock pulses followed by a STOP before proceeding.
5. If a transducer fails to respond, Issue a Trigger (Dummy Read) Command and try again after 1 gate time.
6. Couple SDA and SCL lines to V_{IN} and GND respectively, and shield from any possible electrical noise sources.
7. Slow the host clock speed to allow time for voltages to stabilize with high-capacitive loads (long cables and/or multiple devices on the same bus).
8. Do not use edge-transitions to run internal state machines. SCL and SDA should be sampled with a local system clock to avoid inadvertent double-clocking on noisy edges.

3 Frequency Counter

The Frequency Counter is an ASIC (Chip) that provides an interface to the crystal oscillators in a Quartzdyne transducer. It provides serial communications compatible with the NXP I²C bus. Temperature and Pressure frequencies may be queried as well as Chip ID and Status. Control bits may be written to select certain features of the transducer. The Chip controls the Write Enable bit for the memory.

A 7.2MHz reference clock is available on some models and can be used to clock user circuits. The output can be switched to 1kHz using a software command to reduce power consumption.

3.1 Read Counter Command

The Counter Chip includes two completely independent frequency counters providing simultaneous counting of both frequencies with no dead time. The Read request triggers the Chip to transfer count data to a serial divide circuit and the counter starts counting again while the divide is taking place.

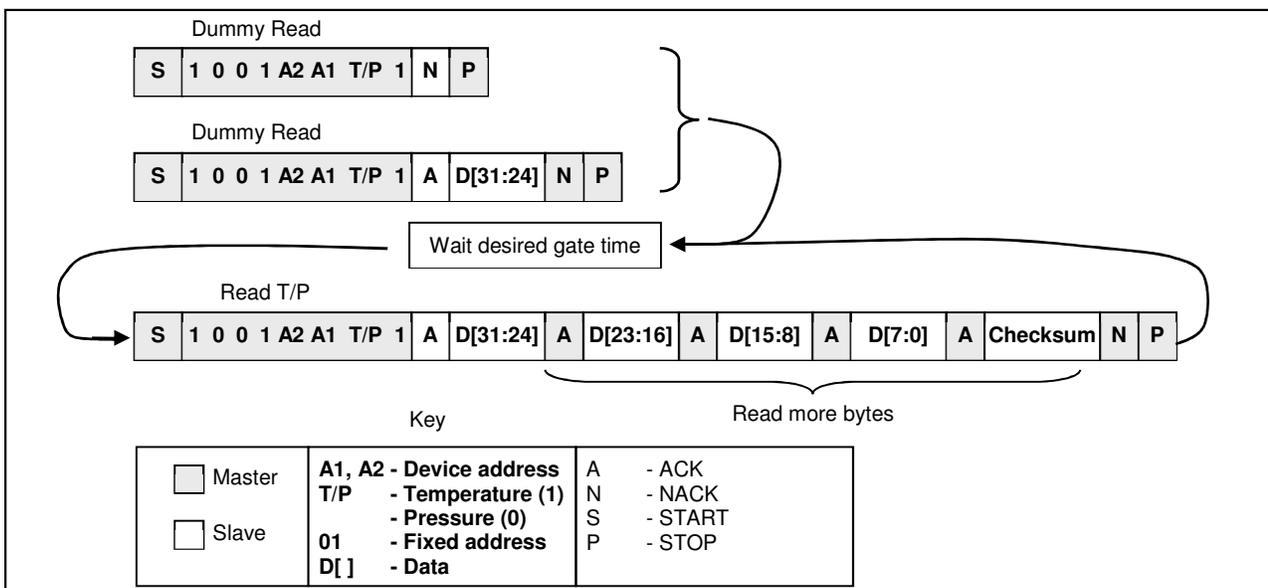
The counters can be queried by issuing a Dummy Read (see **Figure 9**) to each counter successively. If the counter is ready it will acknowledge (ACK) the request; otherwise it will not acknowledge (NACK). If the counter acknowledges, the master must read at least one byte followed by NACK and STOP to clear the bus cycle. If it is not ready, the master should issue a STOP. In either case, the counter that has just been addressed will be re-triggered. Reading one counter does not affect the other counter.

An alternative to the above method is to use the Read Status command and examine the TDetect and PDetect bits. This method returns the status of both counters at the same time while they continue to count. Additionally, the two counters can be simultaneously triggered by writing to the Control Register (see section **Error! Reference source not found.** and 3.4 for more details.)

Once it is determined that a counter is working it can be queried continuously using the desired gate time interval. If a counter is not queried within 2.3 seconds, it will overflow and go into a reduced power mode until re-triggered.

Normally, the master will read four bytes of data. For Chip versions 4.02 and greater, an optional fifth-byte checksum may be read (See Section 3.2).

Figure 9. Read Counter Command



Frequency is reported as the ratio of Pressure (X_P) or Temperature (X_T) counts divided by Reference Counts. It is expressed as an unsigned fixed-point number with the MSB representing 2^{-1} . Frequencies in the range of 10kHz to 100kHz can be read as an integer up to 32 bits in length with the most significant byte first. The first five bits will always be zero.

The gate time (T_G) is determined by how often the master queries each counter and can range from .001 to 2.3 seconds. The effective limit of resolution of the result is determined by the number of counts of the reference frequency $N_R = T_G * F_R$ (approximately 1 part per 7.2 million per second). Increased gate time will give better precision. The precision of the result with a one second gate time is approximately 23 bits starting with the first non-zero bit. This translates to pressure resolution on the order of 0.003 psi (varies with model and actual pressure).

X_P and X_T are used directly to calculate pressure and temperature, and are related to counts (N_P , N_T , N_R) and frequency (F_P , F_T , F_R) as follows:

$$X_P \text{ [n bits]} = N_P * 2^n / N_R = F_P * 2^n / F_R$$

$$X_T \text{ [n bits]} = N_T * 2^n / N_R = F_T * 2^n / F_R$$

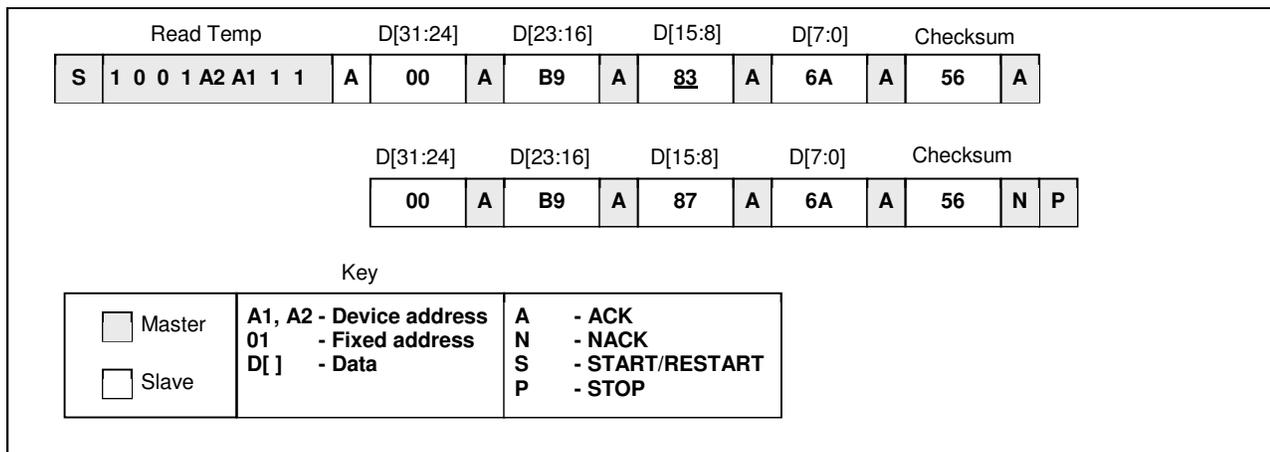
3.2 Checksum

A checksum feature was added to Counter Chip version 4.02. When reading any data including Frequency, Status or Chip ID, a checksum is output as the fifth byte (see example calculation below). If a checksum error is detected the master may re-read the data by continuing to acknowledge (ACK) bytes. As long as the master acknowledges, the Chip re-sends the original data including the checksum. A NACK followed by STOP terminates reading of data. Once the command is terminated the original data cannot be retrieved.

Compatibility Note: Reading the first four bytes is backward compatible to previous versions. Additional bytes in prior versions can be read, but should be considered noise.

Warning: The first time data is read it is new data. Continued reading without terminating the read command simply re-transmits old data. Thus, continued reading of the counters or status without issuing a STOP will not reflect new data.

Figure 10. Read data with checksum showing transmission error in 3rd byte.



Example checksum calculation:

(A)	(B)
0000 0000	0000 0000
1011 1001	1011 1001
1000 0 <u>0</u> 11 bad bit [2]	1000 0111 correct data
0110 1010	0110 1010
0101 0110 checksum	0101 0110 checksum
-----	-----
01 1111 1100 non zero	10 0000 0000 discard two carry-out bits, the rest is zero

Note: The checksum feature does not apply to writing to the Control Register or accessing external Memory (EEPROM)

3.3 Write Control/Status Registers

The control word allows for enabling or disabling various features in the frequency counter ASIC. Reading and writing the first byte of the control word is backward compatible with FPGA versions 3.03 or earlier. Additional control bits have been added for a total of 32 bits described in **Figure 11** and **Table 3**. The “Control” register and “Status” register refer to the same digital bits. “Control” is used in the context of writing to the bits and controlling features. “Status” is used in the context of reading and querying. For example the TDetect and PDetect bits can only be read as “Status” bits. However, dummy values are needed to preserve bit position when writing to the “Control” register.

Figure 11. Control/Status Register bits

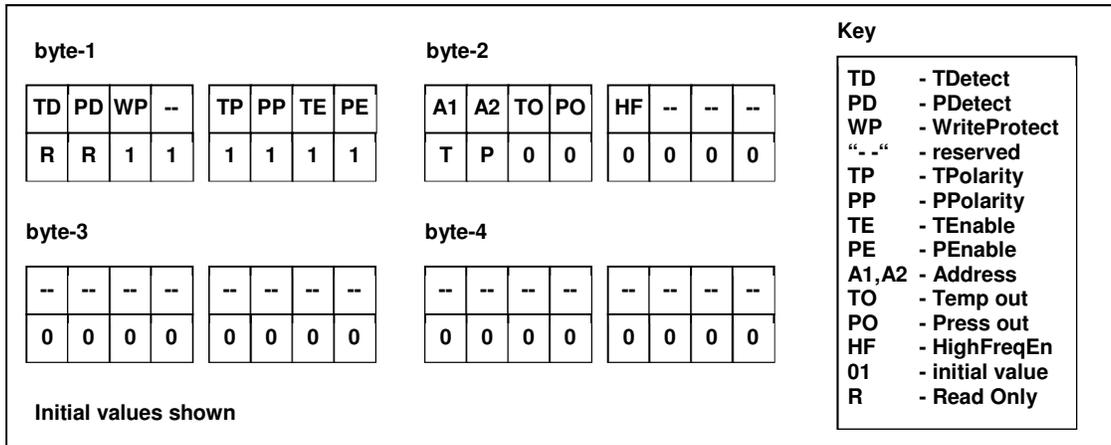


Table 3. Control/Status bit descriptions

BIT	SIGNAL	INIT	RW	DESCRIPTION
31	TDetect	na	RO	Valid temperature data available. This bit is ignored during write.
30	PDetect	na	RO	Valid pressure data available. This bit is ignored during write.
29	Write Protect	1	RW	Write protect memories.
28	-reserved-	1	RW	
27	TPolarity	1	RW	Specify polarity of temperature input signal 1-pos 0-neg
26	PPolarity	1	RW	Specify polarity of pressure input signal 1-pos 0-neg
25	TEnable	1	RW	Enable temperature counters.
24	PEnable	1	RW	Enable pressure counters.
23	A1 Address	A1/T	RO	Address on pin A1/T during power-on reset is stored in this bit.
22	A2 Address	A2/P	RO	Address on pin A2/P during power-on reset is stored in this bit.
21	Temp out	0	RW	Enable pin A1/T as a high drive mixed temperature out If address A1/T is 0 this cannot be set.
20	Pres out	0	RW	Enable pin A2/P as a high drive mixed pressure out If address A2/P is 0 this cannot be set.
19	HighFreqEn	1	RW	Sets pin RO to be 7.2 MHz(1) or 1Khz (0)
18	-reserved-	0	RW	
17	-reserved-	0	RW	
16	-reserved-	0	RW	
15-8	-reserved-	0	RW	
7-0	-reserved-	0	RW	

General: Writing to byte-1 resets both counters simultaneously. To select any of the features in byte-2 of the control word, byte-1 must also be written and care must be taken when doing this to rewrite the intended values.

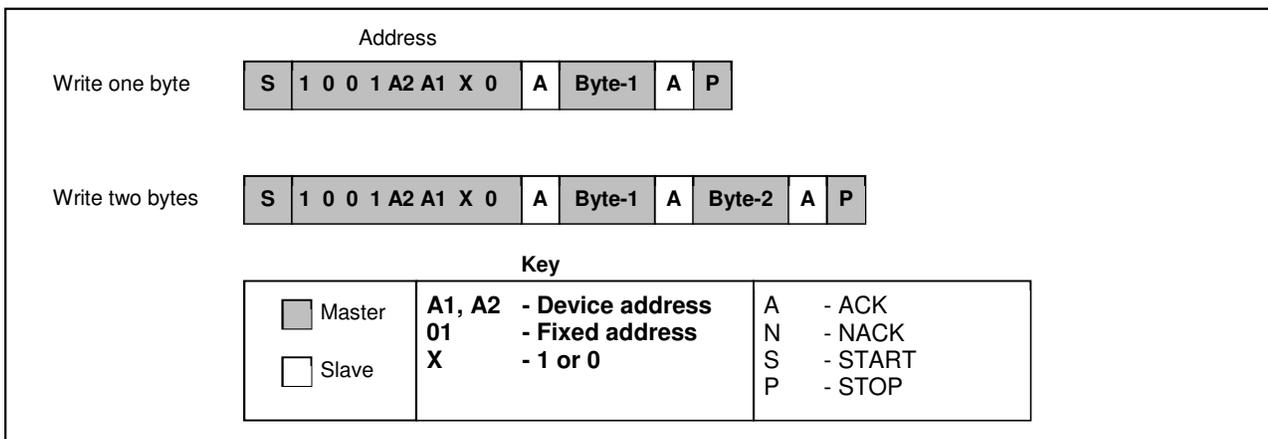
Bits 30-31: Prior to version 3.01 (0D020301, 0D050301) it was possible that Temperature or Pressure Detect remained inactive even though a signal was present. This occurs during the startup sequence, when the Pressure or Temperature signal starts sporadically after the reference frequency is stable. The counter detects a sporadic signal as an under-frequency condition (overflow) and shuts down the counter. When this occurs, the counter can be restarted by executing a Dummy Read to trigger the counter. In order to avoid this situation in versions prior to 3.01, it is recommended that the Query Counter command be used to detect startup, rather than the Query Status command. For Chip or FPGA versions 3.01 and greater, this problem has been resolved; the counter will ignore sporadic startup conditions, and continually retry until a valid frequency is detected. In order to guaranty accurate readings, the host system should wait at least 0.1 seconds after signal detection before triggering the counter for the first read after power is applied.

Bits 20-23: The A1/T and A2/P pins have tristate drivers with internal pullups. During power-on reset, the logical values on these pins are written into control bits 23 and 22 and become part of the device address. Writing different values after power-on reset are ignored but their current value is reported when querying the status register. If nothing is pulling them down during startup then they will be "1" because of the internal pullup. After startup they may be programmed as Temperature and Pressure outputs through the I2C interface only if the startup value is "1". This is accomplished by writing a "1" to TO (bit 21) for Temperature or PO (bit 20) for Pressure. They may be turned off by writing a "0". These pins are independent and may be controlled separately. If either pin is tied to Ground at startup this feature is disabled. Driving the pins with an external digital device is not recommended since the chip and the external device could end up driving the pin at the same time and cause permanent damage. As long as these pins are either floating (externally) or tied to ground during power-on reset there is no problem.

Writing: The control register is written to as follows (see **Figure 12**). A START transition begins the command. The correct address is sent with R/W=1 and the Chip responds with an ACK. The master then writes control Byte-1. After receiving the first control byte, the Chip responds with an ACK and the master can terminate the command with a STOP transition. The master may continue to write bytes and the Chip will ACK after each byte. After four bytes are written, subsequent bytes are ignored, but the Chip will still ACK until the master sends a STOP transition. The STOP must immediately follow a Chip ACK.

After writing any number of bytes and following a Chip ACK the master may send a RESTART instead of a STOP followed by a second address byte and read either the Status or Chip ID. This is described in further detail in "Query Status or Chip ID Command" section 3.4.

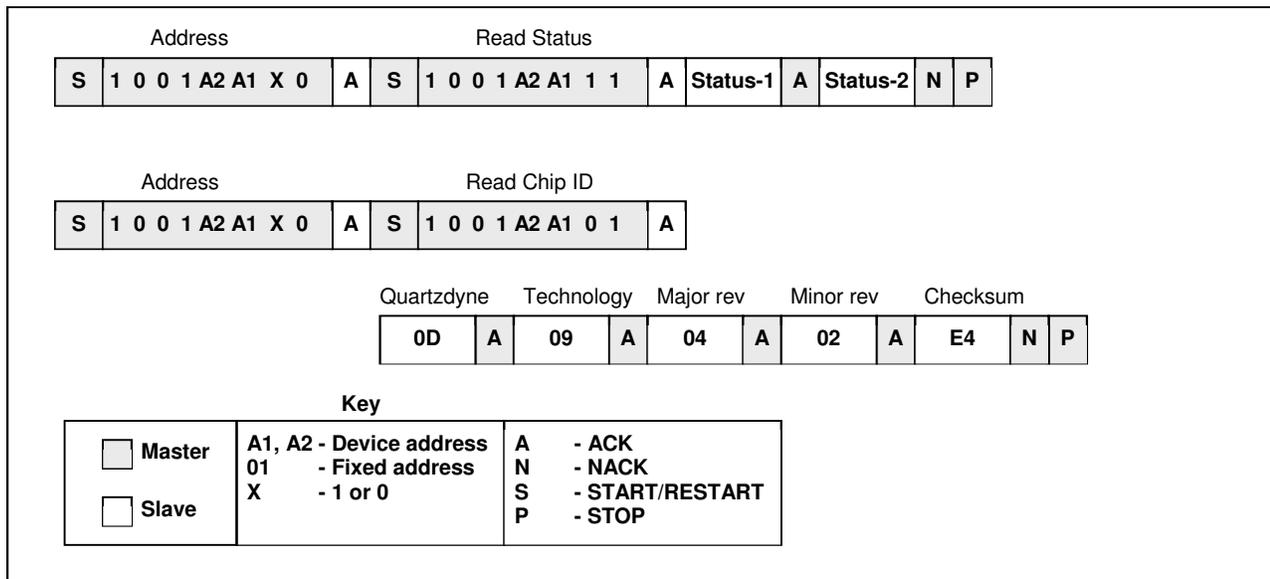
Figure 12. Write Control



3.4 Query Status or Chip ID

Information about the internal Chip is accessed by querying the Status or Chip ID registers (see **Figure 11** and **Table 3**). Start by issuing a Write Control Command (R/W=0) with or without a data to be written. Immediately following a Chip ACK send a repeated START transition (no STOP after ACK). The address is again repeated, this time with R/W=1 and S/V=1 for Status or S/V=0 for Chip ID. The master may then read as many bytes as desired. When reading Status the first byte is compatible with earlier versions. Additional bytes are returned as defined in **Figure 11**. The fifth byte is a checksum and the data is repeated if the master continues to query (ACK bytes). To terminate the Query command the master responds with no-acknowledge (NACK) followed by a STOP.

Figure 13. Query Status or Chip ID



The Chip ID code is defined as follows:



Table 4. Counter Chip ID bits

Bits	Value (Hex)	Description
QQ	0D	Quartzdyne
CC	02	SMT FPGA
	05	Hybrid FPGA
	09	ASIC
V V V V	0103	FPGA version 1.03
	0402	ASIC version 4.02

Example version codes are:

- 0x0D020103
- 0x0D090402

4 Serial EEPROM

The EEPROM contains 8192 bytes of non-volatile EEPROM and is compatible with standard 24C64 devices. Data is addressed using two Data Address bytes following the Device Address. Only 13 bits are significant. The remaining bits should be programmed as '0' for compatibility with future devices. The Serial EEPROM should only be used to store coefficients. Unused sections of the EEPROM are reserved for use by Quartzdyne only. The presence of the extended EEPROM is not guaranteed in future versions.

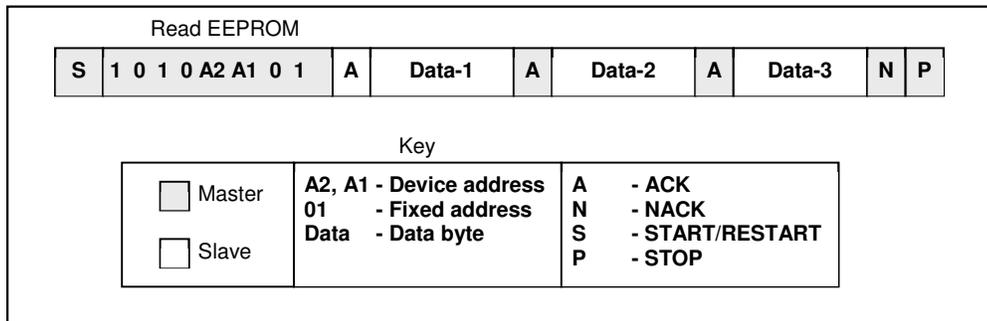
In systems where Pressure and Temperature will be calculated locally, it is important that the checksum in the coefficient file be verified before using the data. As of January 2010, four redundant copies of the coefficients are provided in successive memory locations. The system may use this redundancy to re-create the true coefficients in case of corruption.

Transducers with Chip versions prior to 2.01 used a smaller EEPROM that used a one byte addressing scheme, and had no write protect feature (ATMEL AT24C04). Chip versions prior to 4.03 used the RAMTRON FM24C64 and coefficient storage was limited to 180°C. Above this temperature, data was likely to become corrupt. Data retention is significantly improved in versions 4.03 and higher.

4.1 EEPROM Read - Current Address

The internal Data Address counter maintains the last address accessed during the last read or write operation, incremented by one. This address stays valid between operations as long as power is maintained. The address rollover during read is from the last byte of the device to the first byte (0x1FFF to 0x0000). To read data beginning with the current address, the device is addressed with the R/W bit set high (READ). The EEPROM acknowledges, and will transmit sequential data as long as the master acknowledges each byte. After the last byte is read, the master terminates the read with NACK followed by STOP. The checksum is not available when reading from external memory.

Figure 14. EEPROM Current Address Read

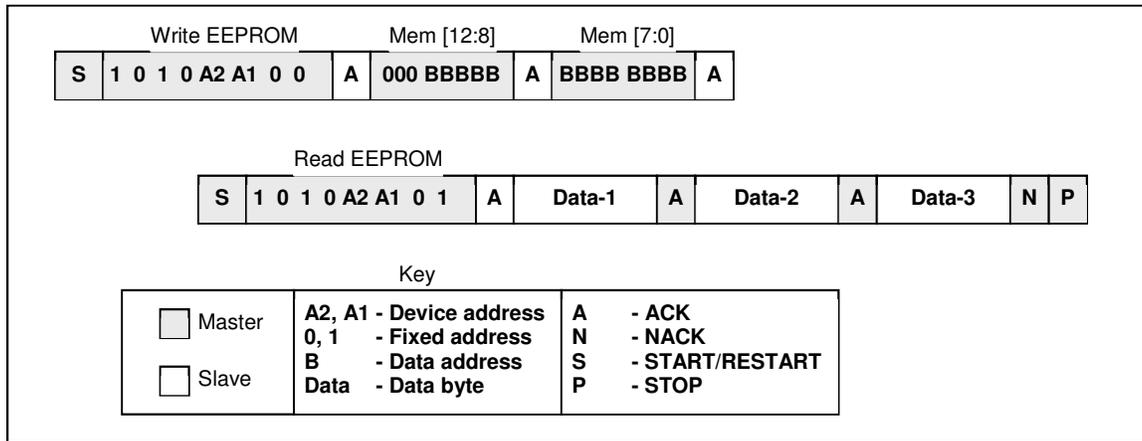


4.2 EEPROM Read - Specific Address

The Specific-Address Read requires a Write command to load the Data Address. The device address byte is sent with the R/W bit set low (WRITE) and is acknowledged by the EEPROM. The two Data Address Bytes are then sent and acknowledged by the EEPROM, following which the master generates a repeated START condition. The device is addressed again, this time with the R/W bit set high (READ). The EEPROM acknowledges, and transmits sequential data as long as the master acknowledges each byte. After the last byte is read, the master terminates the read with NACK followed by STOP.

Warning: Prior to version 4.03 this command could fail if it were the first command issued after a power-on reset. See Section 2.4.

Figure 15. EEPROM Specific Address Read



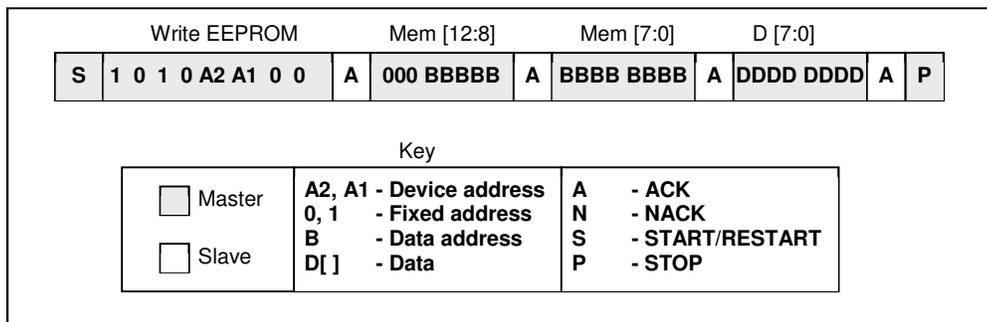
4.3 EEPROM Write

Coefficients are written to the device by Quartzdyne. The EEPROM is not intended to be written by the customer, and this functionality is not guaranteed in future versions. On power on, the device is write-protected. Before programming, the Write Protect bit must be cleared using the Write Control Byte command described earlier. To overwrite data in the EEPROM, the device is addressed with the R/W bit set low (WRITE). The EEPROM acknowledges, and the Two Data Address Bytes are then sent from the master. This is also acknowledged by the slave. The next byte transmitted will be written to the memory location defined by the Address Bytes. Subsequent data will be written into consecutive memory locations within 32 byte pages. If a page boundary is reached, the internal address wraps to the beginning of the current 32 byte page with the potential for data to be overwritten. After the data has been transmitted, the master terminates the write sequence with STOP. The write cycle can take several milliseconds (5-10) during which the device will not respond. Acknowledge polling is necessary to determine when the device is ready to receive additional data.

Note that prior to version 4.03 the write functionality was different, but this algorithm works with the older devices. Customers using a Q-Link or QCOM to write coefficients to a transducer should download the most current firmware to insure compatibility with the newer EEPROM.

Warning: Prior to version 4.03 this command could fail if it were the first command issued after a power-on reset. See Section 2.4.

Figure 16. EEPROM Write Command



4.4 Data Retention

Data retention testing is now in progress by Quartzdyne for the EEPROM used in versions 4.03 and higher. Initial results suggest at least 5000 hours at 225°C. The customer should be warned, however to always verify the checksum before relying on coefficients read from the device.

For versions prior to 4.03, original testing at Quartzdyne confirmed data published by Ramtron regarding data retention at elevated temperatures. The data retention time was found to be limited to approximately 400 hours at 200°C, with an activation energy of 0.94 eV (1000 hours at 180°C). Unfortunately, more recent testing has produced contrary results above 180°C. Corruption can be expected after a few minutes at 200°C. Reading the device at temperatures above 180°C may also contribute to data corruption. The user is advised to provide alternate storage for coefficients once the unit has been deployed. Always verify the checksum prior to using coefficients. Coefficients may be retrieved from our website.

Beginning January, 2010, four redundant copies of the coefficient file are stored in memory locations 0x000, 0x100, 0x200 and 0x300. If an invalid checksum is detected in the first block, the host may read successive blocks to improve the probability of obtaining valid coefficients. This feature has been established as a temporary stop-gap measure while we pursue alternate options for coefficient storage.

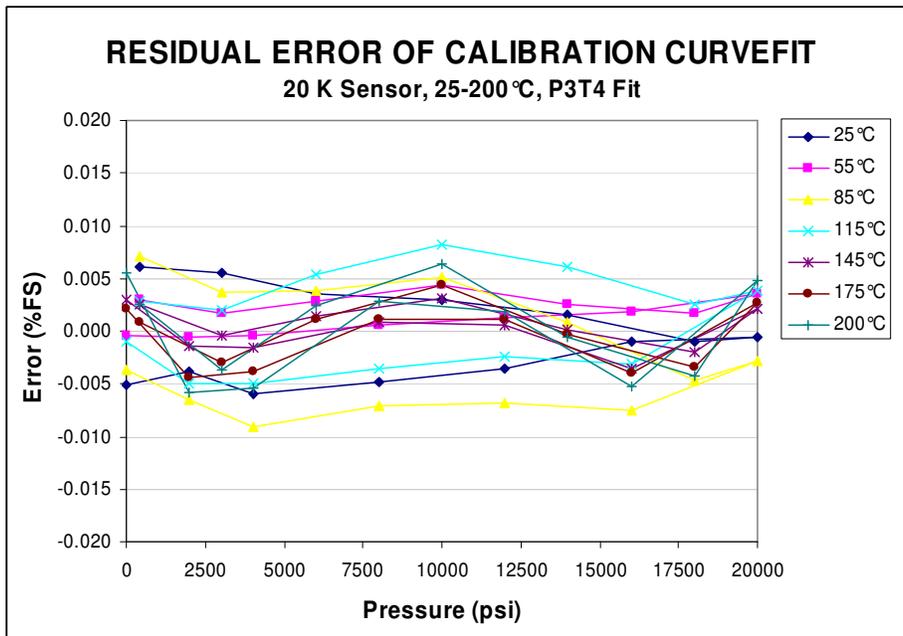
5 Calibration

The unique coefficients for each transducer are determined in calibration. The pressure and temperature counts are read at a number of known pressures at several different temperatures. The coefficients are computer-generated using an LMS (least mean squared) algorithm from the counts and the pressure data recorded during the calibration. Calibration coefficients are then stored in the transducer's EEPROM. Additionally, coefficients may be downloaded from our website (www.quartzdyne.com). Utility programs for generating coefficients are also available.

5.1 Transducer Calibration Data

The performance demonstrated during calibration of each transducer is shown in the Transducer Calibration Chart supplied with each transducer. The chart shows the deviation of each point from the calibration equation using the coefficients calculated for the transducer. **Figure 17** shows the performance typical of Quartzdyne Pressure Transducers. The residual error at all temperatures is less than 0.01% of full scale (FS). This includes any errors in the linearity correction, repeatability, hysteresis, and temperature errors. Note that the residual error compares the transducer to the calibration standard; the error of the standard must be added to the residual error shown. Quartzdyne's calibration standard is a deadweight tester that is accurate to $\pm 0.01\%$ of reading.

Figure 17 . Calibration curve typical of Quartzdyne Pressure Transducers.



5.2 Traceability of Calibration

Calibration of Quartzdyne Pressure Transducers is traceable to the U.S. National Pressure Standards maintained by the National Institute of Standards and Technology (NIST), formerly known as the National Bureau of Standards (NBS).

5.3 Typical Values of Corrections Used in Calibration

Achieving the 0.01% of reading accuracy specification of our pressure calibration standards requires that all potential sources of calibration error are identified. In the Quartzdyne calibration laboratory the atmospheric pressure reference is either a Mensor 2104 or DH Instruments PG-7302-M Digital Pressure Gauge. The height of the head is measured to within 0.25 inches [6.35 mm], providing a head pressure correction to 0.008 psia [0.570 mbar]. The pressure generated by our DH Instruments Model 50300-01 and Model 50000 Class S fundamental (deadweight) pressure standards includes corrections for local gravity, air buoyancy, and pressure and temperature effects on the cross-sectional area of the pistons. With these corrections, the pressure generated by the pressure standards is accurate to 0.01% of reading. The typical values used in calibration at our facility are "non-standard" values due to our 4251 ft [1296 m] elevation.

P_{atm}	= 12.6 psi [869 mbar] (measured continuously during calibration)
P_{head}	= 0.033 psi/in [0.897 mbar/cm]
Local gravity	= 9.797930 m/s ² (interpolated from National Geodetic Survey data http://www.ngs.noaa.gov/)
air density	= 1.015 gm/cm ³ (buoyancy correction is 23 ppm less than standard)
Piston pressure effect	= 60 ppm at 10,000 psi [689 bar]
Piston temperature effect	< 50 ppm typical

5.4 Pressure Control Required for Calibration

Quartzdyne Pressure Transducers measure absolute pressure. The pressure that the transducer senses during calibration is the sum of the atmospheric pressure, head pressure, and the pressure generated by the (gauge) pressure source.

The calibration equation is a polynomial in temperature and pressure. The order of the equation determines the number of different pressures and temperatures required in the calibration. As a minimum, at least one more data point is required than the order of the fit for each variable. Some redundancy in the pressure and temperature measurement is recommended. For example, in our calibrations we typically apply a sequence of pressures at each of five or six temperatures. Four temperatures are required for a third order fit, five for a fourth order. At each temperature, measurements are made at ambient atmospheric pressure, at 400 psia [27.6 bar], and at least every 20% of range up to full scale. The pressures are typically applied in the sequence:

ambient, 20%, 60%, 80%, 100%, 90%, 70%, 40%, 400 psia.

It is not necessary to follow this exact procedure, but this sequence has several advantages:

- (1) It provides a large enough sample of different pressures (four are required for a third order curve fit; nine unique pressures are used) without excessive repetition.
- (2) Two low pressure points show the zero return. (The lowest pressure at which our automated fundamental pressure standard provides 0.01% of reading accuracy is 400 psia [27.6 bar]; at lower pressures the error is a constant 0.02 psia [0.00138 bar].)
- (3) It provides the same number of increasing and decreasing pressure data points; the curve fit will be forced through the center of any apparent hysteresis loop.

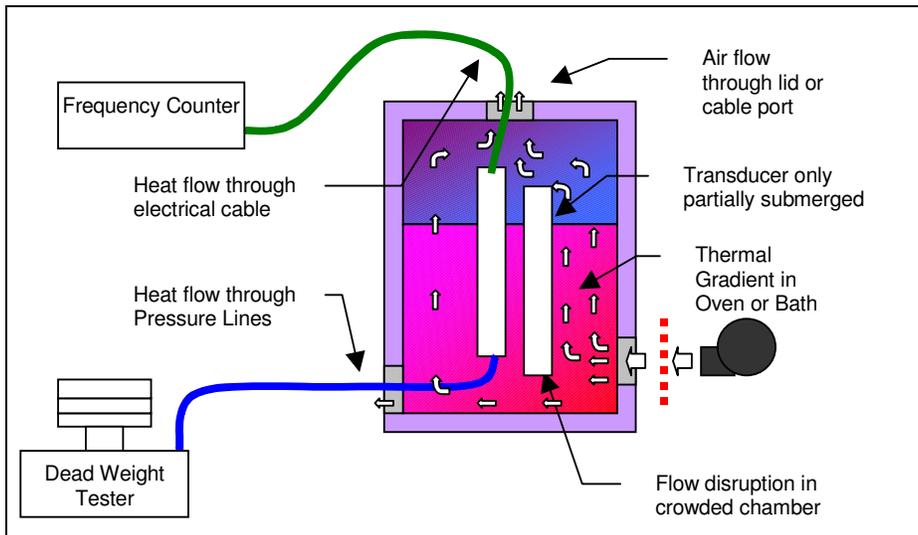
5.5 Temperature Control Required for Calibration

The pressurization process described above must be performed at several temperatures as dictated by the fit order. The temperature need not be measured (unless a calibrated temperature output is desired), but it must be stable. A temperature chamber with stability of at least 0.25°C is recommended. Monitoring the temperature sensor counts will provide an indication of the thermal stability of the transducer; when the temperature sensor has stabilized, a pressure calibration cycle may begin. Note that the temperature sensitivity varies from approximately 68000 counts/°C at -40°C to 150000 counts/°C at 200°C.

It is desirable that the temperature of the three crystals in the transducer be as uniform as possible. Beware of thermal gradients that can develop along the length of the transducer as these may cause significant errors in the calibration. **Figure 18** shows possible sources of temperature gradients that may occur in a calibration oven or bath. The error will be most pronounced in the areas of high thermal sensitivity as shown in **Figure 6**. **Error! Reference source not found..**

At least four temperatures are dictated by a third-order temperature dependence. At least five temperatures are required for a fourth order dependence. The six- or seven-temperature calibration done by Quartzdyne provides adequate redundancy. It is recommended that the entire temperature range of use be covered in the calibration. Specifications of the transducer outside the factory calibrated temperature range are not guaranteed.

Figure 18. Possible sources of temperature gradients in a calibration oven or bath.



6 Calibration Coefficients

6.1 Hex Coefficient Format

Quartzdyne Digital Pressure Transducers include binary coefficients stored in EEPROM. The coefficients occupy the lower 256 bytes of EEPROM with three redundant copies in the first 1024 bytes. Coefficient files may also be supplied separately using Intel HEX coding as described in section 6.5. For compatibility with legacy systems, text coefficient files (.CFF/.CFT) as described in our Frequency Manual may also be supplied. Hex coefficients are preferred and should be used for new systems.

Table 5. Binary Coefficient File Format

Address	Bytes	Coding	Symbol	Description	Typical Value	Interpretation
000 - 001	2	BCD	Type	File Type	0x0D01	QD BCF
002 - 003	2	BCD	Version	File Version	0x0123	1.23
004 - 007	4	BCD	SerNo	Transducer Serial Number	0x0D062351	SN 062351
008 - 00F	8	ASCII	PartNo	Root Part Number (Left Justified)	QSB001	QSB001
010 - 013	4	BCD Date	CalDate	Calibration Date (yyyymmdd)	0x20011231	December 31, 2001
014 - 014	1	signed[8]	Pmin	Min Pressure (kpsi)	0x00	0
015 - 015	1	signed[8]	Pmax	Max Pressure (kpsi)	0x10	16000
016 - 016	1	signed[8]	Tmin	Min Temperature (x 5°C)	0xF8	-40
017 - 017	1	signed[8]	Tmax	Max Temperature (x 5°C)	0x10	80
018 - 018	1	code[8]	Cal1.Type	Output #1 Calibration Type - see coding	0x01	Pressure psi, bar
019 - 019	1	code[8]	Cal1.Prescale	Output #1 Prescale Type - see coding	0x00	0
01A - 01A	1	signed[8]	Cal1.N1	Output #1 Fit Order in X1 (Pressure)	0x03	3
01B - 01B	1	signed[8]	Cal1.N2	Output #1 Fit Order in X2 (Temperature)	0x03	3
01C - 01F	4	float[32]	Cal1.S1	Output #1 Scale Factor Standard Units	0x39800000	1 psi / 4096
020 - 023	4	float[32]	Cal1.S2	Output #1 Scale Factor Alternate Units	0x378D3466	0.0689476 bar / 4096
024 - 027	4	signed[32]	Cal1.OFS2	Output #1 Offset for Alternate Units	0x00000000	0 psi / 4096
028 - 08B	100	signed[32]	Cal1.C(i,j)	Output #1 Coefficients (Max 25)		C00,C01..Cij
08C - 08C	1	code[8]	Cal2.Type	Output #2 Calibration Type - see coding	0x02	Temperature °C, °F
08D - 08D	1	code[8]	Cal2.Prescale	Output #2 Prescale Type - see coding	0x03	0
08E - 08E	1	signed[8]	Cal2.N1	Output #2 Fit Order in X1 (Pressure)	0x00	0
08F - 08F	1	signed[8]	Cal2.N2	Output #2 Fit Order in X2 (Temperature)	0x03	3
090 - 093	4	float[32]	Cal2.S1	Output #2 Scale Factor Standard Units	0x39800000	1 °C / 4096
094 - 097	4	float[32]	Cal2.S2	Output #2 Scale Factor Alternate Units	0x39E66666	1.8 °F / 4096
098 - 09B	4	signed[32]	Cal2.OFS2	Output #2 Offset for Alternate Units	0x00011C72	(32 / 1.8) °C * 4096
09C - 0FB	96	signed[32]	Cal2.C(i,j)	Output #2 Coefficients (Max 24)		C00,C01..Cij
0FC - 0FE	3	Unsigned[8]	EOF	End of File	0xFF0000	
0FF - 0FF	1	Unsigned[8]		Checksum		0x00-SUM(000..0FE)

Table 6. Calibration Type Coding

Value	Meaning
0	None
1	Pressure in psi or bar
2	Temperature in °C or °F
255	End of File

Table 7. Prescale Type Coding

Value	Meaning
3	$Z = \sum C_{ij} \cdot (X_P/2^{24})^i \cdot (X_T/2^{24})^j$

6.2 Checksum Calculation

The sum of all bytes in the coefficient file should add to 0x00 using 8-bit addition. The checksum may be calculated as the two's complement of the sum of all previous bytes.

6.3 Pressure and Temperature Calculations

Pressure and Temperature (Z) are calculated using a polynomial expansion of counts Xp and Xt as read from a digital transducer or a QCOM interface according to the following equations:

$$Z_{out}(Standard\ Units) = S1 \sum_{i=0}^{Np} \sum_{j=0}^{Nt} C_{ij} \left(\frac{Xp}{2^{24}}\right)^i \left(\frac{Xt}{2^{24}}\right)^j$$

$$Z_{out}(Alternate\ Units) = S2 \left\{ OFS2 + \sum_{i=0}^{Np} \sum_{j=0}^{Nt} C_{ij} \left(\frac{Xp}{2^{24}}\right)^i \left(\frac{Xt}{2^{24}}\right)^j \right\}$$

Standard units are pressure in psi and temperature in °C. Alternate units are bar and °F respectively.

In systems with limited processing power, the polynomial may be computed efficiently using 32-bit signed integer math if it is factored as shown below (example is a 3x4 fit – actual fit order may vary):

$$Z = Temp_0 + \frac{Xp}{2^{24}} \times \left(Temp_1 + \frac{Xp}{2^{24}} \times \left(Temp_2 + \frac{Xp}{2^{24}} \times (Temp_3) \right) \right)$$

where

$$Temp_0 = C_{00} + \frac{Xt}{2^{24}} \times \left(C_{01} + \frac{Xt}{2^{24}} \times \left(C_{02} + \frac{Xt}{2^{24}} \times \left(C_{03} + \frac{Xt}{2^{24}} \times (C_{04}) \right) \right) \right)$$

$$Temp_1 = C_{10} + \frac{Xt}{2^{24}} \times \left(C_{11} + \frac{Xt}{2^{24}} \times \left(C_{12} + \frac{Xt}{2^{24}} \times \left(C_{13} + \frac{Xt}{2^{24}} \times (C_{14}) \right) \right) \right)$$

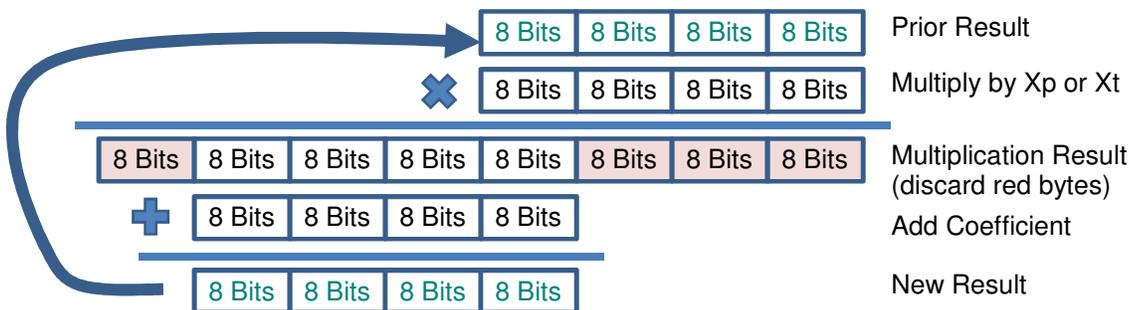
$$Temp_2 = C_{20} + \frac{Xt}{2^{24}} \times \left(C_{21} + \frac{Xt}{2^{24}} \times \left(C_{22} + \frac{Xt}{2^{24}} \times \left(C_{23} + \frac{Xt}{2^{24}} \times (C_{24}) \right) \right) \right)$$

$$Temp_3 = C_{30} + \frac{Xt}{2^{24}} \times \left(C_{31} + \frac{Xt}{2^{24}} \times \left(C_{32} + \frac{Xt}{2^{24}} \times \left(C_{33} + \frac{Xt}{2^{24}} \times (C_{34}) \right) \right) \right)$$

The innermost loop is the last coefficient (C₃₄ in the example). Calculation proceeds by repeatedly adding each coefficient to the prior result, multiplying by either Xp or Xt and then dividing by 2²⁴. The inner loop multiplies by Xt creating a temperature-compensated coefficient for the outer loop which multiplies the temporary result by Xp. See the example code in Section 6.4

In systems without floating point processors, the multiplication and 2⁻²⁴ scaling may be accomplished by computing the 64 bit result of X_P or X_T times the previously accumulated value, and then selecting bytes 7 thru 4 as the result as shown in **Figure 19**. If the eighth byte is non-zero, an overflow has occurred.

Figure 19. Polynomial computation using integer math.



6.4 Example Calculation Routine

The following VBA code snippet is taken from Calcfunctions.xls which may be downloaded from our website (www.quartzdyne.com). It demonstrates calculation of Pressure or Temperature from a Quartzdyne Hex coefficient file. Before calling this function, the Hex file is first parsed into a single Hex string using the function qdHexFileData(). The resultant string is then passed to this function along with the counts as read from a Quartzdyne Pressure Transducer. Within the function, Hex data is further parsed using calls to qdHexGetInt(), qdHexGetLong and qdHexGetFloat, also defined in CalcFunctions.xls.

```
Public Function qdHexCalc(xp As Double, xt As Double, hexData As String, Optional
  calcT As Boolean = False, Optional altUnits As Boolean = False) As Double
  ' Calculates pressure from I2C Digital Transducer Readings
  ' using a Quartzdyne Hex coefficient file.
  ' The file contents should be pasted into the spreadsheet at a convenient location
  ' The data portion must then be extracted using the qdHexFileData function.
  ' Presssure is calculated from the extracted data in the hexData string.
  '
  Dim xpp As Double, xtp As Double
  Dim Z As Double, Temp As Double
  Dim i As Integer, j As Integer, n As Integer
  Dim np As Integer, nt As Integer
  '
  ' n points to last coefficient
  '
  xpp = xp * 2 ^ -24 ' Only prescale type 3 supported for digital coefficients
  xtp = xt * 2 ^ -24
  If calcT Then base = 140 Else base = 24

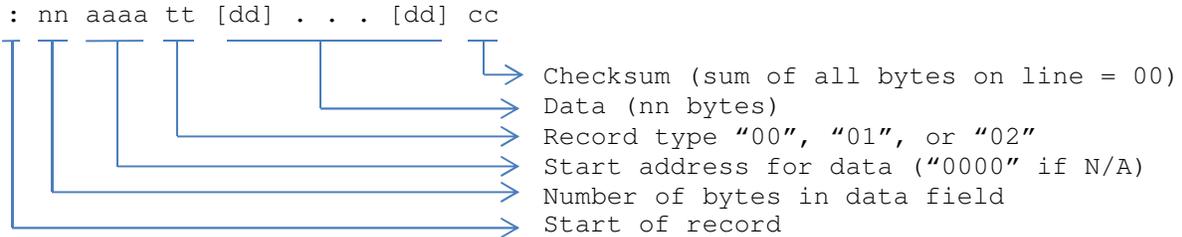
  n1 = qdHexGetInt(hexData, base + 2) ' Fit order in Xp
  n2 = qdHexGetInt(hexData, base + 3) ' Fit order in Xt
  n = 4 * (n1 + 1) * (n2 + 1) + base + 12 ' point to last coefficient
  Z = 0
  For i = 0 To n1 ' outer loop for np, xp polynomial
    Z = Z * xpp ' multiply by xp (0 first time through loop)
    Temp = 0
    For j = 0 To n2 ' inner loop for nt, xt polynomial
      Temp = xtp * Temp ' process factored using coefficients from last to first
      Temp = Temp + qdHexGetLong(hexData, n)
      n = n - 4
    Next j
    Z = Z + Temp ' add temperature compensated pressure coefficient
  Next i

  If (altUnits) Then
    Z = Z + qdHexGetLong(hexData, base + 12) ' OFS2 placed just before coefficients
    qdHexCalc = Z * qdHexGetFloat(hexData, base + 8) ' alternate scale factor S2
  Else
    qdHexCalc = Z * qdHexGetFloat(hexData, base + 4) ' primary units scale factor S1
  End If
End Function
```

6.5 Intel HEX File Coding

Quartzdyne Digital Transducer Coefficient files are distributed using the Intel HEX file format. This allows viewing of the object file with standard tools and makes for easy file transfer from one computer system to another. Binary data stored in these files is stored MSB first (Big Endian). As some programming environments expect integer and floating point numbers in an LSB first format, translation may be required.

The Intel Hexadecimal Object file record format has a four-field prefix that defines the start of record, byte count, load address and record type. This is followed by the actual data and a 2-character checksum. Each 2-character pair is a Hex representation of an eight-bit byte with the most significant nibble (4-bits) first. The checksum is the 2's complement of the sum of the preceding bytes in the record (excluding the start character). The sum of all bytes, including the checksum, will be 00. Records are separated by an end-of-line delimiter (CR/LF).



Record Types

Record type 00, the data record, is the record that contains the data of the file. The data record begins with the colon start character (":") followed by the byte count, the address of the first byte and the record type ("00"). Following the record type are the data bytes and the checksum. The following are examples of data records (spaces are included for clarity only and are not included in the actual file).

```
:10 0040 00 137DF401CFADAAFFA38FB5FEDF13E001 4E
:05 0010 00 0102030405 AA
```

Record type 01, the end record, signals the end of the data file. The end record starts with the colon start character (":") followed by the byte count ("00"), the address ("0000"), the record type ("01") and the checksum ("FF").

```
:00 0000 01 FF
```

Record type 02, the extended segment address record, defines bits 4 through 19 of the segment base address. It can appear anywhere within the object file and it affects the absolute memory address of all subsequent data records in the file until it is changed. The extended segment address record starts with the colon start character (":"), followed by the byte count ("02"), the address ("0000"), and the record type ("02"), followed by bits 4 through 19 of the segment base address and the 2 character checksum. Record type 02 is not used in Quartzdyne HEX coefficient files

```
:02 0000 02 1000 55
```